UNIVERSITY OF NEW SOUTH WALES
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

WebPMS: A WebCMS-Based Software Project Management System

Thesis B Report

Student Name	Student Number
Mark Lockery	3187890
John Youkhana	3187956

Degree: Software Engineering

Supervisor: Albert Nymeyer

Assessor: John Shepherd

13 October 2009

Contents

1 Introduction

The aim of this project was to continue the development of a set of easy to use project management tools to aid in the project work taking place in the software engineering workshops offered at CSE. These tools are to be accessible to both students and staff (for tracking and assessment purposes) and are to provide functionality for groups to share files in a repository, schedule and assign tasks to group members, carry out basic quality assurance, and generally aid the students in managing their projects.

A large part of what the software engineering workshops aim to teach the students is the importance of good project management practices. Presently however, there are no standard tools available to all of the students, making formal project management difficult and also making it very difficult for the lecturer-in-charge to assess each group's effort and success in the area of project management. With a standard set of accessible tools for the students to use, it would be much easier for the lecturer-in-charge to track each group's progress and would make the management processes easier for the students.

There are plenty of project management tools in existence presently, but, as demonstrated later in the report (specifically in the Existing Systems sub-section of the Background section), none of them meet our needs in terms of their usability, functionality and availability. For this purpose we need an easy to use, integrated set of project management tools that are easily available to all software engineering students at UNSW, and these requirements show the need for a new system to be developed.

This remainder of this report will outline the scope of our system including the functionality it provides and what is out of scope. Then we have gone into detail about the problem we face (i.e. the need for this project) and evaluated alternative solutions and partial solutions that are currently available using an evaluation framework we have come up with to ensure our critiques are well-focused.

Following that we have provided a programmer's manual that contains implementation details on the system; a user manual that provides explanations of the system's functionality; and an evaluation and analysis section in which we provide information on the testing and evaluation work we have done to ensure that our system is functioning as we desired (via requirements tracing), reliable, and most importantly, useful to the target users.

The report concludes with conclusions drawn from the project and an outline on the enhancements and future work planned (via other thesis projects) on the system we have created.

1.1 System Scope

WebPMS will focus on providing project management functionality, primarily based around group management functions, tasks that are assigned to students, the quality assurance procedures associated with those tasks, and reporting functionality derived from tasks created and worked on, i.e. task schedules and associated statistics. The system will build around the existing functionality of the 2008 thesis project WebTMT, whilst incorporating the functionality of WebGMS. The whole system will function within the constraints of WebCMS.

Functionality provided by the system will include:

- Group management, i.e. create/delete groups, add/remove students, assign roles to students.
- Task management, i.e. create/remove tasks, assign students to tasks, assign and enforce quality assurance, create task schedules, store and submit task output files, etc.
- Automatic production of task schedules and relevant statistics (both group- and individual-based).

Functionality outside of the scope of the system includes:

- Assessment of student tasks and group project management, i.e. assessment and annotation of task output, task schedules, and member contributions. Note that this was de-scoped during Thesis B and will be a future enhancement.
- File repository features, i.e. those provided by DoCo (storage of input and output files for tasks). This functionality is there and remains unbroken by any of our work, but is not included as part of our deliverable.
- Group communication, i.e. any kind of forum, instant messenger, or other online communication tool.

- Development tools. The system is solely for project management assistance, and does not provide tools to directly aid in the actual production of software.

2 Background

2.1 Background Introduction

This chapter mainly focuses on setting the context for this project and evaluating alternative solutions, as well as what systems/tools we currently have access to. This chapter defines and examines the problem in detail, and sets out an evaluation framework. We then use this evaluation framework to analyse and evaluate the systems that are already out there, and demonstrate the necessity for our project.

Our evaluation framework is primarily based on the usability and availability of the systems, as well as how well they meet the specific functional requirements we have for the system (i.e. what the system is capable of).

The existing systems and tools we evaluate include Microsoft Project, WebCMS2 in its current state, WebGMS, WebTMT, and DoCo. These systems are critiqued for their suitability based on the evaluation framework we have developed.

2.2 Project Context

A very important component of the software engineering workshops at CSE is project management. To do project management properly you need to have tools for assistance. Ideally, these tools allow you to assign and schedule tasks, monitor progress, document the development, do configuration management, carry out basic QA, keep a software repository, and generally work together as a team.

There are tools available that do these things and lots more, but they are usually large and difficult to master (particularly in a short time). In this project we seek a light-weight tool that is tailored to the needs of the software engineer workshops run at UNSW, and easy to use.

WebCMS is a web-based course management system that is used throughout the school. It offers many basic facilities. One facility that it does not offer however is support for project management and the features listed above. Via the WebGMS module, it has support for assigning people to "groups", but does not have functionality beyond that. In order to achieve the simplest distribution of the project management tools, we are to integrate the system with WebCMS.

2.3 Evaluation Framework

The framework which we will use to evaluate existing systems in their usefulness and suitability for our needs is based around three primary categories: usability, availability, and functionality.

2.3.1 Usability

For a system to meet our demands in this area requires that the solution is not overly-complicated and is generally easy to use without significant time spent on training. The user interface should be intuitive and uncluttered. Complex software will have a steeper learning curve and require more time to learn before completing tasks for the course, which can be off-putting for students, and in semesters lasting only twelve/thirteen weeks, can take away a significant portion of time that could be spent furthering current projects.

2.3.2 Functionality

A complete solution must meet the functional requirements we have decided upon in order to provide the students and software engineering staff with useful tools that will help them manage and track the progress of their projects more efficiently.

For a complete list of our requirements, see appendix A: WebPMS Requirements.

Note that the total functionality can be pieced together by several different programs and tools, provided that we can integrate them together into a single application. So a tool is not necessarily void of usefulness just because it does not meet *all* of these functional requirements.

2.3.3 Availability

The project management tools provided must be accessible to all students in a group if it is to be considered useful to that group, as all students must be able to track their project's progress, access their file repository, view assigned tasks, etc. As such, we require that the solution be easily available to all software engineering students.

Essentially this immediately eliminates any solutions that require purchase, are operating system exclusive, or are otherwise difficult for students to obtain. An ideal solution would be for the system to be based online.

2.4 Existing Systems

2.4.1 Microsoft Project

(Information from Microsoft 2009 (see references) and personal experience.)

Overview:

Microsoft Project is a full-featured commercial project management application including tools for creating and assigning resources to tasks, managing budgets, developing plans, tracking progress, and analysing workloads.

Evaluation:

- **Usability**

  One reason why MS Project is not an ideal solution for our situation is the overall complexity of the program. We are really seeking a lightweight solution that only contains features that the software engineering students will find useful and/or necessary. We are only after tools to schedule tasks for different group members, monitor their progress, document the development of the project, carry out some basic QA on the tasks, and keep a software and file repository with appropriate versioning. Whilst MS Project provides tools for a lot of these, it also provides a lot of other (unnecessary, in our case) functionality (e.g. budgeting tools) that generally just clutters up the interface and makes the system more complicated to use. This would generally mean that the students would need to spend significant time learning how to actually use the application, and in the 12/13-week semesters that a software engineering workshop takes place, this would not be acceptable.

- **Functionality**

  Another point is that, despite the large feature set, MS Project would still not be an "all-in-one" solution for us as it does not provide all of the functionality required. Due to its intended use in industry, it does not have features to allow group-based viewing and editing of data, or functionality relating to managing groups (roles, leaders, etc). I.e. the program does not allow you to store all data in one area, and only allow groups to view and edit their own files. It also does not support any separation of staff and student views.

- **Availability**

  Another significant issue we have with any solution involving MS Project is that it simply does not meet our availability requirements. For use in the software engineering workshops, we need the tools to be easily available to all participating students. There are three factors keeping MS Project from being adequately available for this purpose:

  1. The software is not free, and would require either individual purchases by students, or a bulk purchase by the university in order to provide each student with a copy.

  2. The software is operating-system exclusive to Windows operating systems, essentially meaning that it will be difficult for students who prefer Linux or Macintosh operating systems to gain access to the software.

  3. The software is not hosted online. This would be the ideal solution for distribution, as it means students can access it anywhere they have access the internet. Requiring local installs means that the university will need to install the software on the lab machines, and students will have to install it on all of their personal computers they wish to use.

Conclusion: not a suitable solution. Furthermore, the study of MS Project indicates to us that any ready-made solution is not going to be able to provide what we need here, specifically in terms of availability and accessibility to students and staff and the group-based and staff vs. student nature of the system that we require. It has become clear that we would get a better result by piecing together our own solution.

2.4.2 WebCMS2 in its Present State

Overview:

WebCMS is a web-based course management system that is used for a large portion of the courses taught by CSE. Students are already required to use WebCMS to perform basic functions such as viewing course notices, obtaining assignment specifications (and sometimes submitting assignments), and using the message board. However WebCMS, as it is now, has no project management features. It provides the functionality to create groups (assign members) via WebGMS but there are no group management, task scheduling or version-control features built-in.

Evaluation:

- **Usability**

  WebCMS is a simple and user-friendly system that students will be exposed to via courses other than the software engineering workshops. It does not offer project management tools, but its simple interface makes it a suitable option as a framework to which we could attach project management functionality (similar in application to providing a ”plugin” type application (or set of applications) to it).

- **Functionality**

  WebCMS itself only provides functionality for course management, though tools can be created and plugged in to take advantage of WebCMS2's availability.

- **Availability**

  WebCMS is based online and logins are provided for all students enrolled in relevant CSE courses, so the system is perfectly available for all desired students, at no cost, and with no operating system or other machine restrictions. These points make WebCMS an ideal platform for us to attach our project management tools to, despite its current lack of features.

Conclusion: a good platform to add functionality to.

2.4.3 WebGMS

Overview:

WebGMS is a component module in WebCMS that controls student groups. The primary function of WebGMS is (limited) management of student groups; this includes create/remove groups and add/delete students from groups. This module is integrated in WebCMS2 and is deployed with the current WebCMS2 system.

Evaluation:

- **Usability**

  WebGMS provides a simple and suitable interface that lets users manage groups easily. The GUI has buttons/links to create/remove groups and to add/delete students from these groups.

- **Functionality**

  As stated in the overview the WebGMS provides only basic group management functionality, i.e. to create groups of students. A desired feature for group management in our system is to have the concept of students taking on roles within their group, though this is presently missing from WebGMS.

WebGMS provides the following functions:

  - Create and remove student groups
  - Add and remove students to existing groups

- **Availability**

  Since WebGMS is a module that is integrated with the current WebCMS2 system; it is available online to all CSE students.

Conclusion: WebGMS provides some group management procedures that can be used with the current WebCMS2. Student roles can be added to this module.

2.4.4 WebTMT

(Information from Chan Xuong Vien 2008.)

Overview:

WebTMT is a module developed by a thesis student at UNSW that integrates with the present WebCMS system. WebTMT is a task management system that relies on the groups system created in WebGMS. It includes functionality to create, assign, track and update progress of tasks as well as providing features to assign a quality assurer to each task.

Evaluation:

- **Usability**

  WebTMT is an extra module that integrates well with WebCMS, so it is simple to get up and running. It is based around WebCMS2's GUI and is generally user-friendly, with most available options being self-explanatory.

- **Functionality**

  WebTMT provides a significant amount of the project management functionality we require, primarily concerned with task management.

  WebTMT provides the following functions:

  - Create, edit, delete and view requirements
  - Assign a group leader who is granted additional privileges in the system
  - Create, edit, delete and view tasks including setting deadlines and estimated times
  - Link tasks to requirements
  - Assign group members to tasks
  - Assign quality assurers to tasks
  - Assign input and output files to a task (note that this can be linked with DoCo's file repository)
  - Start, end and submit tasks
  - Approve tasks (via group leader or quality assurer for that task)

- **Availability**

  Since WebTMT is a module that can be integrated with the current WebCMS2 system; it is available online to all CSE students.

Conclusion: a very useful tool that can satisfy a number of our project management related requirements; easy to add as a module to WebCMS.

2.4.5 DoCo

(Information from Daniel Joseph Troy 2008.)

Overview:

DoCo is a version control system that can be used as a module within WebCMS and also has links with WebTMT. Tasks created in WebTMT can have input and output files linked to a DoCo file repository.

DoCo provides the functionality of a basic version control system with a few more advanced features.

Evaluation:

- **Usability**

  DoCo uses a simple GUI primarily consisting of buttons and text boxes in order to perform most of its functions and in particular, the basic features such as file creation and uploading are self-explanatory. It's easy enough to learn how to use DoCo on the fly so there are no issues here.

- **Functionality**

  DoCo provides the following functions:

  - Add and remove files from a repository
  - User-group attachments (i.e. only group members can access their repository)
  - Upload new versions of files, while storing previous versions under their version number.
  - File locking whilst one group member is working on it
  - Tracking of files to indicate which group member is currently working on it
  - Peer ratings of files
  - Submission of files (indicating that they are complete)
  - Integration with WebTMT's task input and output files

- **Availability**

  Since DoCo is a module that can be integrated with the current WebCMS2 system; it is available online to all CSE students.

Conclusion: a useful tool that can satisfy our requirements for a shared group file repository. Due to its compatibility with WebCMS and WebTMT, it would be advantageous and significantly less effort to use DoCo for our file repository needs, rather than developing our own interface for another available file repository system.

2.5 Background and Existing Systems Summary

From our study of the tools presently available, it was clear to us that our needs necessitated a tailor-made solution, as opposed to merely inserting or adapting an existing product. This primarily came about due to the desired system's unique nature as a system to be used by university staff and students, where we have issues of access levels and data-hiding. We require that groups can only see their own work, and that lecturers are able to see all of the groups' work, whereas standard project management tools desired for industry use have no such restrictions.

In order to provide a solution available to all software engineering students and staff, it is evident that it would be best for us to have our solution integrated with WebCMS2. As such, our solution will be presented as a single selectable module for courses using WebCMS2, which will include the functionality of the existing WebGMS and WebTMT software, plus our own extensions. DoCo connectivity will be dealt with in a future project, though that is essentially just a matter of switching it on within WebTMT and configuring the two modules to interact with each other.

3 Programmer's Manual

3.1 Architecture

As is the standard for WebCMS and modules integrating with WebCMS, our system is split into three main components as follows:

- **Web Browser Interpreting HTML**

  Users will access and interact with the system via webpages within the WebCMS site for their particular course (e.g. Software Engineering Workshop 2A). The webpages will be PHP files generating HTML code for the user's web browser to interpret.

- **Apache Web Server and PHP Engine**

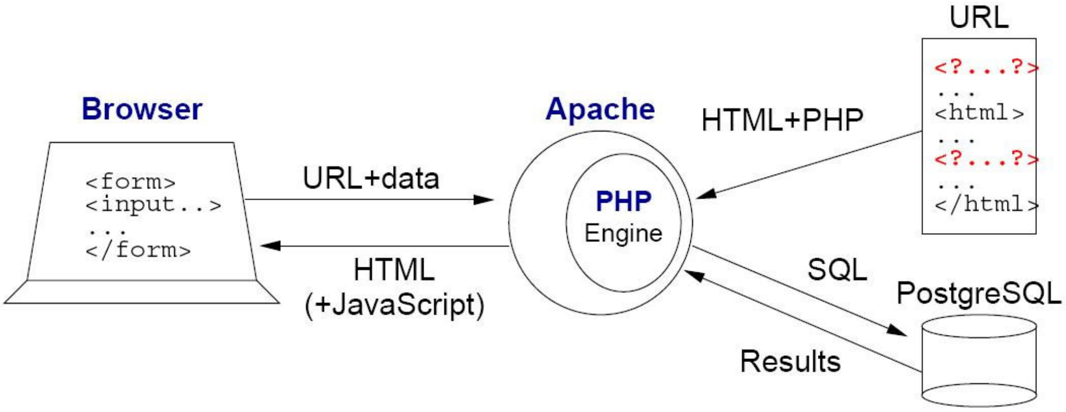
  As mentioned above, the HTML webpages that the users actually interact with will be generated from PHP files that will be stored on the server within CSE. An Apache web server will be used to host the WebCMS website and for courses with the WebPMS module selected, this server will host the WebPMS files too.

  The PHP files are where most of our source code will be stored and where most of the system's functionality and user interface come from. The PHP code is used to perform user-level functions, query and update the database (mentioned below), manipulate and process data coming from or going to the database, and generate HTML code to display the data with an interface consistent with the rest of the WebCMS modules. This is also where functionality such as viewing rights, task schedule generation, and assessment of tasks/schedules/contributions (with results stored in the database) will be implemented.

- **PostgreSQL Database**

  The database component of the system will be used to store all system information and data, including (but not limited to) student roles within groups, task information and status, task assignees, output files from tasks, and task schedule data/files. There will also be some data manipulation functions (i.e. PL/pgSQL functions) stored in the database, for efficient gathering of certain information required at the top-level, as well as various triggers to function as assertions to ensure basic correctness of data received and stored.

The below diagram illustrates the system's architecture (from John Shepherd 2009, slide 20):



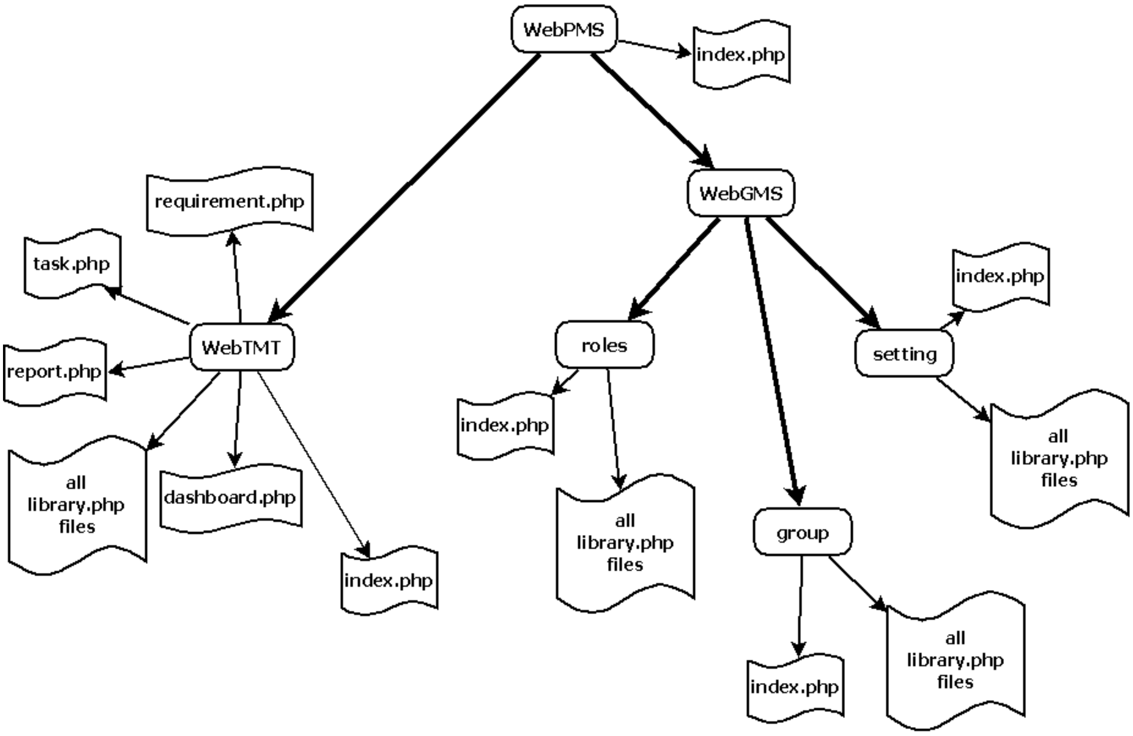
3.2 Implementation Details

- **Database**

  For the WebPMS module to function correctly the database had to have changes made. These changes were made to the current "'webcms"' tables, "'webtmt"' tables and new tables were added. These changes included:

  - WebCMS Tables
    - In table "'groupmember"' the column "'role"' was added.
  - WebTMT Tables
    - In table "'tmt_task"' the column "'type"' was added.
  - Original Tables
    - Table "'pms_setting"' was added with columns "'type"', "'have_roles"' and "'courseid"'.

- **Structure** The WebPMS module structure is made up of a combination of WebTMT and WebGMS with extra functionality. The figure below shows the file structure of the WebPMS Module.



4 User Manual

4.1 User Manual Introduction

This section, the user manual, highlights the functionality of WebPMS both from the perspective of academic staff (lecturer-in-charge, mentors/tutors) and students taking the relevant courses.

Details on what can be done (with some details on "how") are included along with screenshots of the relevant aspects of the interface.

4.2 Functionality - Staff

4.2.1 Admin For The WebPMS Module

The Web Project Management System (WebPMS) set of tools is implemented as a tool/module that can be added to the existing WebCMS2 system, on a course-by-course basis. Course administrators are provided the ability to add these tools via the same WebCMS tool system that is used presently to include functionality such as course outlines, notices, messageboard, and calendar.

Once WebPMS is added to a course in WebCMS2 it can be configured to provide different levels of functionality, with the following options available:

- Light

  Functionality provided:

  - Assign students to named groups (staff)

  Motivation:

  For courses that merely want to be able to refer to collections of students as groups, but do not need the functionality associated with ongoing task management, the light version of WebPMS will suit. Essentially it replaces WebGMS, sharing most of that functionality. This functionality will be extended in a following thesis project where marks will be able to be assigned to "groups" rather than just individual students.

- Full

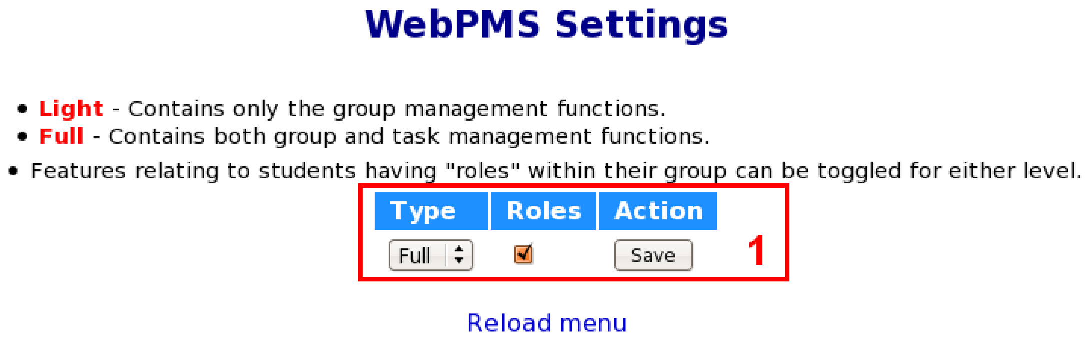
  Functionality provided:

- Assign students to named groups (staff)
- Add/remove/modify requirements (student - team leader)
- Add/remove/modify/update tasks and their status, linked to requirements (student - team leader)
- Dashboard containing the current outstanding tasks for you (individual students)
- Automatic task schedule generation (for viewing by students and staff)
- Automatic student statistics generation (for viewing by students and staff)

Motivation:

The bulk of WebPMS's functionality is designed to support the students working on projects in CSE's software engineering workshops. The above functionality has all been deemed useful tools for this specific purpose, though they may also come in handy for managing project work in other courses within CSE.

In addition to this, the group roles functionality – assigning each student in a group a ”role” – can be toggled on and off and applied to either of the above levels of WebPMS. In the full level, this determines which students can be assigned which tasks. These roles can be changed dynamically throughout the session.

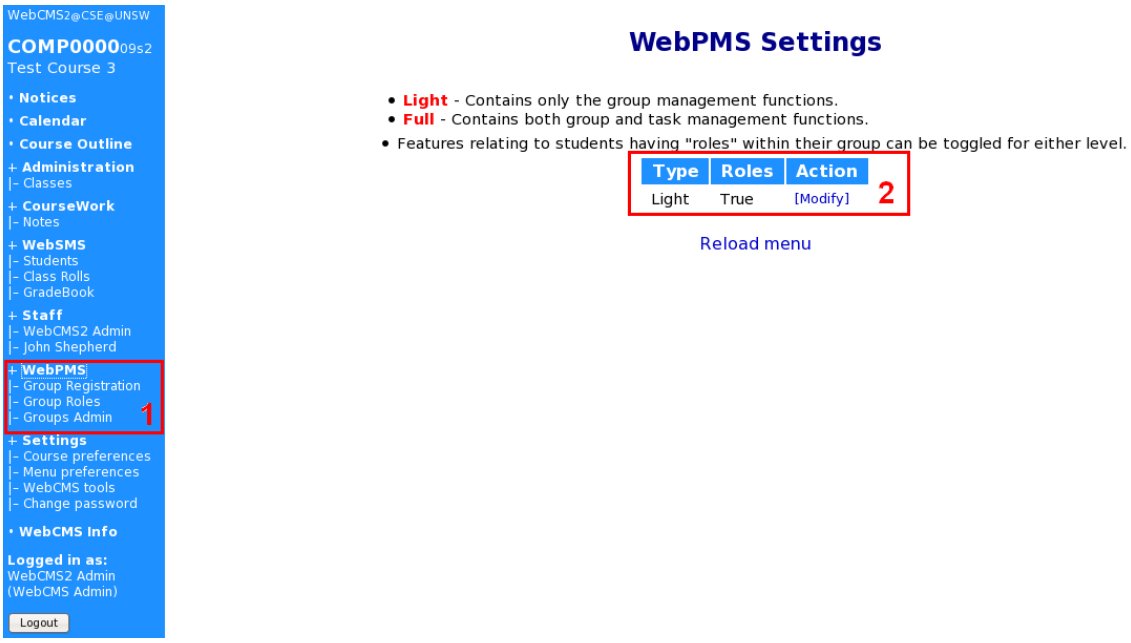


WebPMS Settings

- Light - Contains only the group management functions.
- Full - Contains both group and task management functions.
- Features relating to students having "roles" within their group can be toggled for either level.

Type	Roles	Action
Full	☑	Save

To configure the WebPMS settings the user will either select the ”’Full”’ or ”’Light”’ options from the drop down box and have roles turned on or off by the toggling the checkbox as indicated in marker 1 in the above figure.



WebPMS Settings

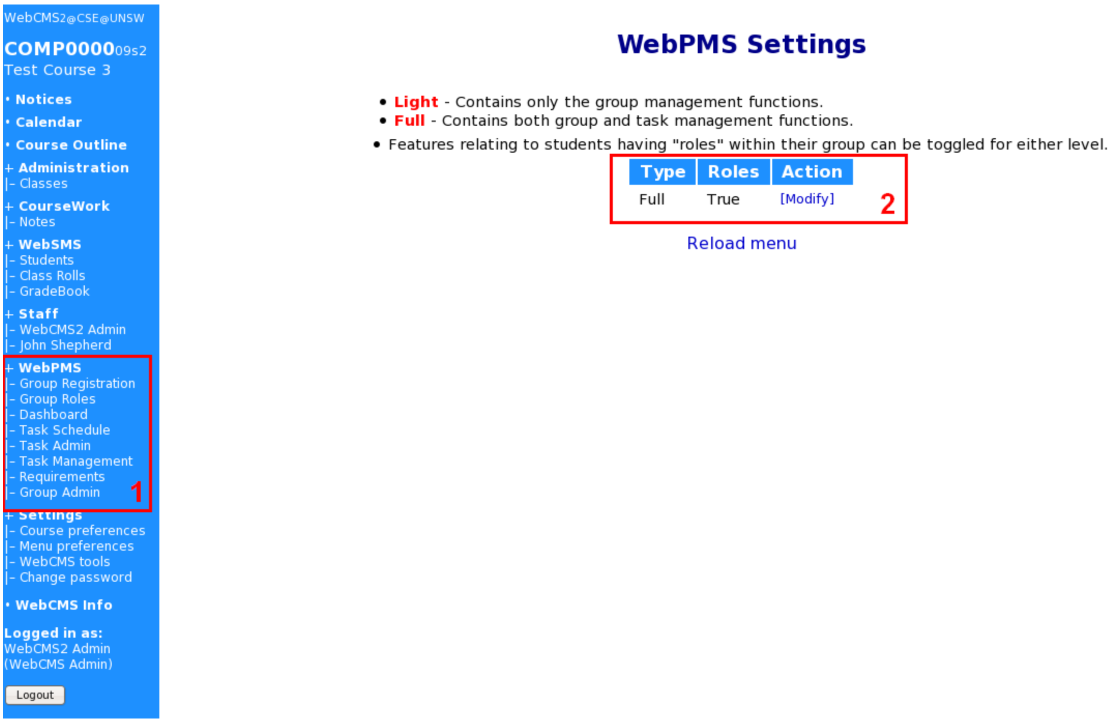
- **Light** - Contains only the group management functions.
- **Full** - Contains both group and task management functions.
- Features relating to students having "roles" within their group can be toggled for either level.

Type	Roles	Action
Light	True	[Modify]

Marker 2 on the above figure shows the light functionality with roles turned on settings. Marker 1 on the above figure shows links for user that are available.

These links provide the functionality to:

- Assign/edit/remove roles.
- Create/edit/remove groups.
- Add students to groups.
- Edit overall group settings.



WebPMS Settings

- **Light** - Contains only the group management functions.
- **Full** - Contains both group and task management functions.
- Features relating to students having "roles" within their group can be toggled for either level.

Type	Roles	Action
Full	True	[Modify]

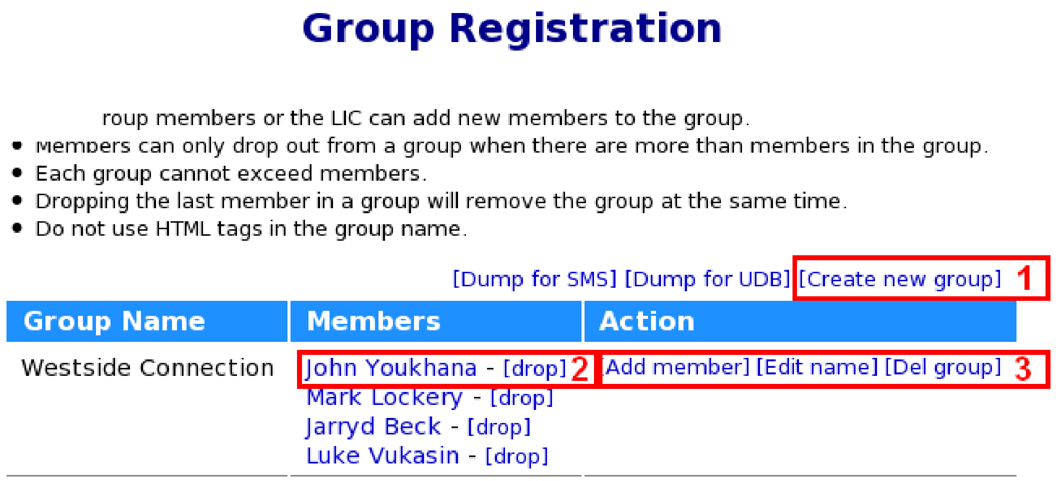
Marker 2 on the above figure shows the full functionality with roles turned on settings. Marker 1 on the above figure shows links for user that are available.

These links provide the functionality to:

- Same as light functionality.
- Add/edit/remove tasks.
- Add/edit/remove requirements.
- Assign students to tasks.
- Edit overall task management settings.

4.2.2 Group Registration

Staff have the ability to create (named) groups for students, assign enrolled students to these groups and manage the groups (add/remove members, delete groups, etc). Whilst this functionality will primarily be used by the students themselves, the lecturer in charge in particular (and other staff when necessary) need the ability to assign students to groups when they cannot form groups themselves.



Group Registration

roup members or the LIC can add new members to the group.
- Members can only drop out from a group when there are more than members in the group.
- Each group cannot exceed members.
- Dropping the last member in a group will remove the group at the same time.
- Do not use HTML tags in the group name.

[Dump for SMS] [Dump for UDB] [Create new group]

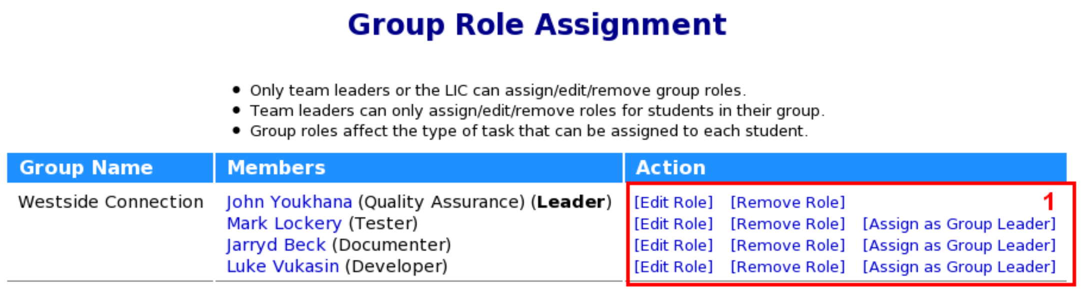
Group Name	Members	Action
Westside Connection	John Youkhana - [drop] Mark Lockery - [drop] Jarryd Beck - [drop] Luke Vukasin - [drop]	Add member] [Edit name] [Del group]

To create a group the staff will click on the link show on the above figure as marker 1. Marker 2 shows the student's name and a link to drop the user from the group. Groups can be managed by clicking on the links indicated as marker 3 in the above figure.

4.2.3 Group Member Roles and Group Leader Assignment

Staff have the ability to assign/overwrite the roles of students within their groups in case that is ever required. The group roles functionality is described in further detail in the next section (student functionality).

The staff may be required to change a group's leader if their current leader is unreachable. This is due to the fact that the leader is given extra power that other group members do not have, including the ability to create requirements, create and assign tasks, and assign group members roles within the group.



Group Role Assignment

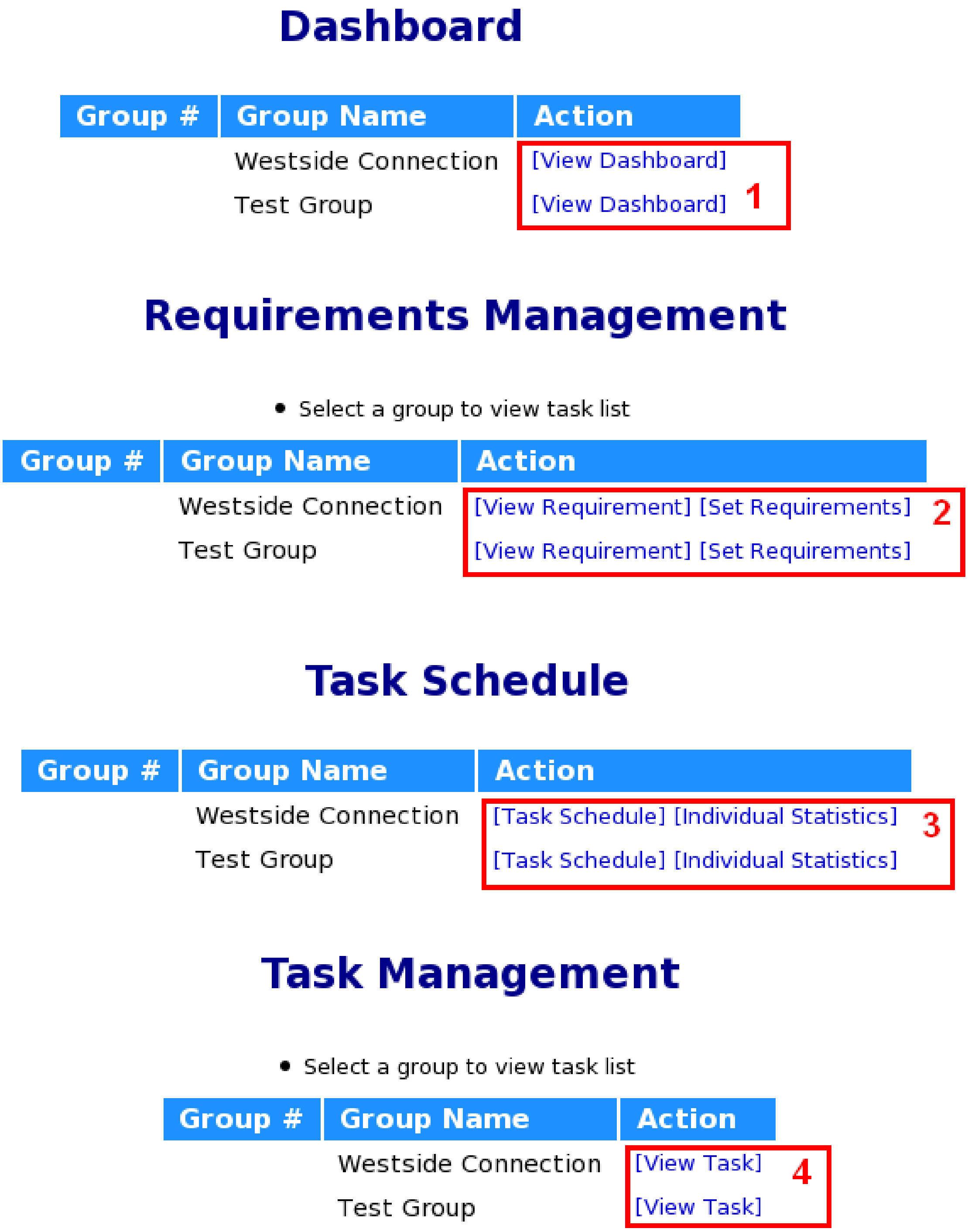
- Only team leaders or the LIC can assign/edit/remove group roles.
- Team leaders can only assign/edit/remove roles for students in their group.
- Group roles affect the type of task that can be assigned to each student.

Group Name	Members	Action
Westside Connection	John Youkhana (Quality Assurance) (**Leader**) Mark Lockery (Tester) Jarryd Beck (Documenter) Luke Vukasin (Developer)	[Edit Role] [Remove Role] [Edit Role] [Remove Role] [Assign as Group Leader] [Edit Role] [Remove Role] [Assign as Group Leader] [Edit Role] [Remove Role] [Assign as Group Leader]

The staff can assign/edit/remove the corresponding student's role. The staff also has the power to assign/change the leader. This is in case the current leader cannot be reached, as stated previously. These functions are performed by clicking on the appropriate links as indicated as marker 1 on the above figure.

4.2.4 Viewing of Functions for Student use

In order to keep track of the progress of the groups in their course, staff are given read-only access to all of the pages that the students use, i.e. requirements, task management, and dashboard (these are all expanded upon in the next sub-section: Functionality - Students). Unlike the student views for each of the sections, staff are first presented with a page asking which group's requirements/tasks/dashboard they would like to view before they are taken to the page that shows up from the perspective of the selected group.



Dashboard

Group #	Group Name	Action
	Westside Connection	[View Dashboard]
	Test Group	[View Dashboard]

Requirements Management

- Select a group to view task list

Group #	Group Name	Action
	Westside Connection	[View Requirement] [Set Requirements]
	Test Group	[View Requirement] [Set Requirements]

Task Schedule

Group #	Group Name	Action
	Westside Connection	[Task Schedule] [Individual Statistics]
	Test Group	[Task Schedule] [Individual Statistics]

Task Management

- Select a group to view task list

Group #	Group Name	Action
	Westside Connection	[View Task]
	Test Group	[View Task]

The staff's view on the student functions is shown in the above figure. The staff will see a list of groups and a link to view the actual data as indicated by markers 1,2,3,4 on the above figure.

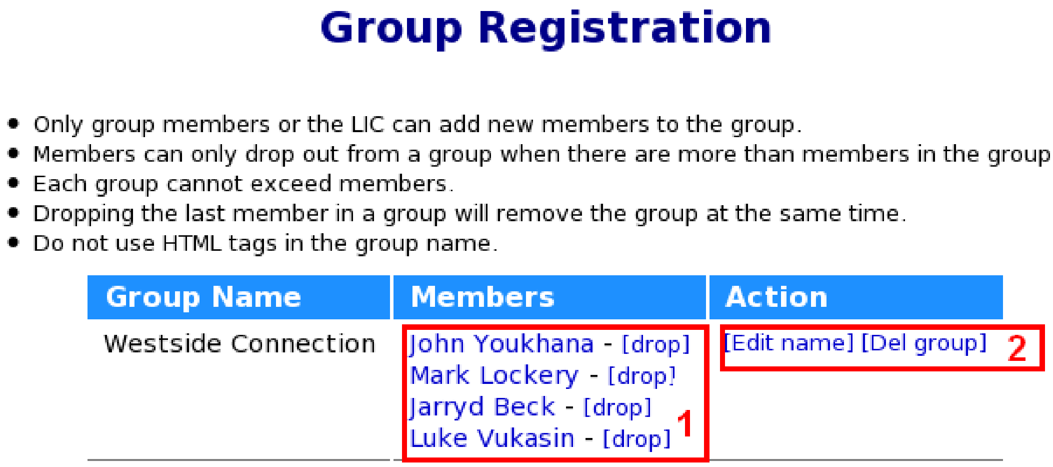
4.2.5 Task Schedule and Statistics

Of particular importance to the staff are the groups' task schedules and accompanying statistics. Whilst the students have access to the task schedule and statistics also (read-only also, as this is all automatically generated based off of the tasks and their statuses), they will find the dashboard more useful for instructing them on what tasks they need to be working on at the present time and in the near future. The task schedule on the other hand is a convenient way for lecturers and other staff to get a feel for how much work each group has done thus far, what they are doing at present, and what they currently have planned for the near-future and remainder of the semester. The statistics associated with the task schedules (both individual and group-oriented) are a useful tool to aid the lecturer in determining the effort levels of groups and their individual members, including a comparison of each group member against the others in their group to see if the workload was split fairly – this information can be used as an objective basis for assessment.

4.3 Functionality - Students

4.3.1 Group Registration

As staff do, students have the ability to create (named) student groups, assign themselves to these groups and manage the groups (add/remove members, delete groups, etc).



Group Registration

- Only group members or the LIC can add new members to the group.
- Members can only drop out from a group when there are more than members in the group.
- Each group cannot exceed members.
- Dropping the last member in a group will remove the group at the same time.
- Do not use HTML tags in the group name.

Group Name	Members	Action
Westside Connection	John Youkhana - [drop] Mark Lockery - [drop] Jarryd Beck - [drop] Luke Vukasin - [drop]	[Edit name] [Del group]

Students who are a members of the group can drop group members as indicated by marker 1 and also have the option edit/remove the group by clicking on the appropriate links as shown in marker 2 on the above figure.

4.3.2 Group Member Roles

WebPMS provides functionality for team leaders to assign, edit, and remove roles to/from the members of their group including themselves. The group leader assignment functionality is handled by this page.

These roles are there to provide an idea of what types of tasks each student should be assigned. The roles that can be assigned are:

- Developer
- Documenter
- Tester
- Quality Assurance

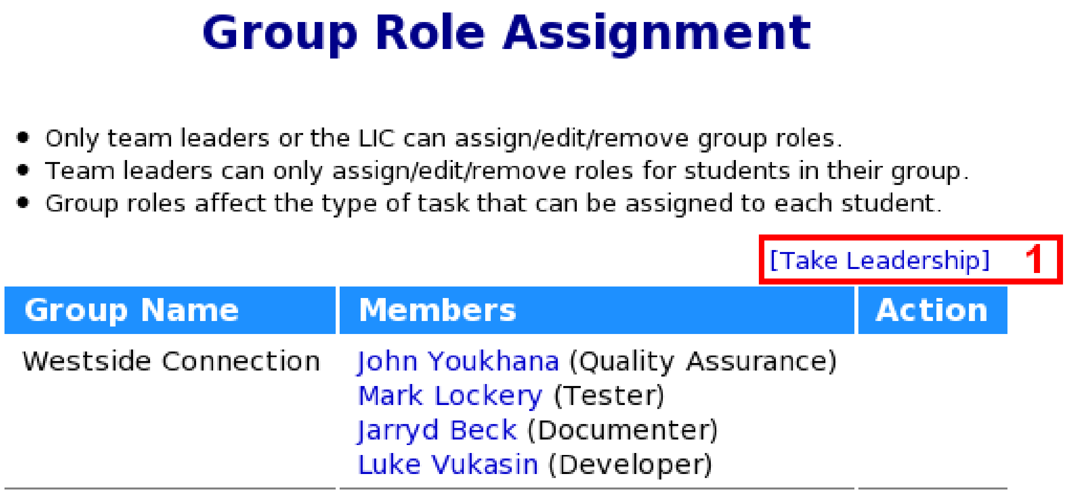
The roles functionality restricts quality assurance tasks to those students designated as quality assurers, i.e. when the team leader creates a new task, they will only be able to assign members with the ”Quality Assurance” role to be the task’s quality assurer. As well as this, all tasks created are given a task type that will be used to determine which group members are acceptable assignees, based on the role they have.

Note that roles can be updated when necessary throughout the session by the team leader.

Group leader assignment works as follows:

1. If no group leader is presently assigned, any student may simply take the leadership role by logging in to the course webpage and clicking on the ”Take Leadership” link on the group roles page.

2. If a group leader is already assigned, that student is able to assign the leadership role to another student by logging in to the course webpage, clicking on the ”Assign Leadership” link on the group roles page, and selecting another group member from a drop-down menu.



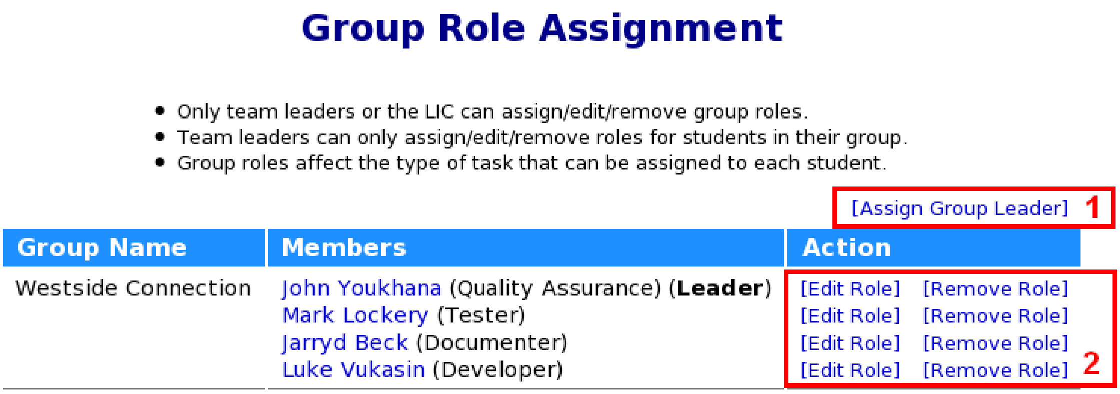
Group Role Assignment

- Only team leaders or the LIC can assign/edit/remove group roles.
- Team leaders can only assign/edit/remove roles for students in their group.
- Group roles affect the type of task that can be assigned to each student.

[Take Leadership]

Group Name	Members	Action
Westside Connection	John Youkhana (Quality Assurance) Mark Lockery (Tester) Jarryd Beck (Documenter) Luke Vukasin (Developer)	

Students can take leadership by clicking on the link indiciated by marker 1 on the above figure.



Group Role Assignment

- Only team leaders or the LIC can assign/edit/remove group roles.
- Team leaders can only assign/edit/remove roles for students in their group.
- Group roles affect the type of task that can be assigned to each student.

[Assign Group Leader]

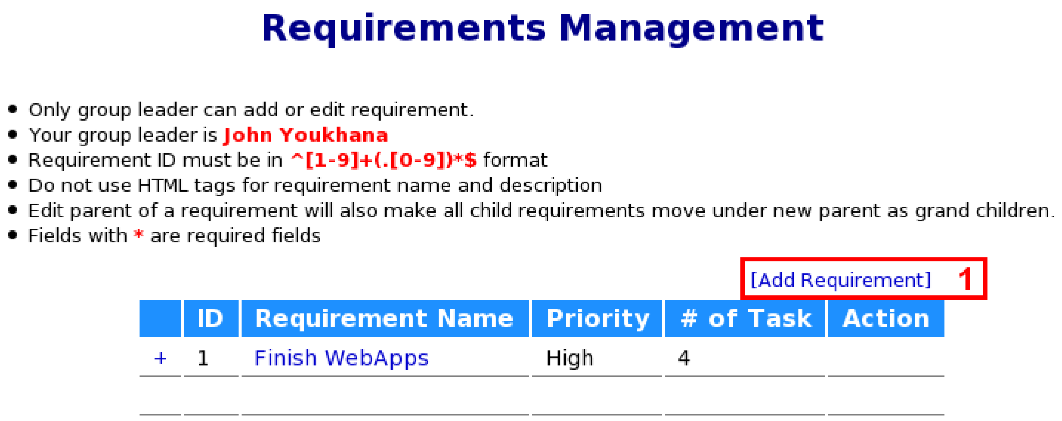
Group Name	Members	Action
Westside Connection	John Youkhana (Quality Assurance) (**Leader**) Mark Lockery (Tester) Jarryd Beck (Documenter) Luke Vukasin (Developer)	[Edit Role] [Remove Role] [Edit Role] [Remove Role] [Edit Role] [Remove Role] [Edit Role] [Remove Role]

Once the leader has been set, the leader has the options to assign/edit/remove roles of fellow group members by clicking on the appropriate links shown on marker 2 on the above figure. The current group leader can also assign a new leader by clicking on the link indicated by marker 1 on the above figure.

4.3.3 Requirements

Group leaders are (exclusively) given the ability to create, edit, and remove requirements (so long as they do not have tasks attached to them). Requirements are viewable by all group members and are assigned the following pieces of information:

- ID
- Name
- Priority
- Description



Requirements Management

- Only group leader can add or edit requirement.
- Your group leader is **John Youkhana**
- Requirement ID must be in **^[1-9]+(.[0-9])*$** format
- Do not use HTML tags for requirement name and description
- Edit parent of a requirement will also make all child requirements move under new parent as grand children.
- Fields with * are required fields

[Add Requirement] 1

	ID	Requirement Name	Priority	# of Task	Action
+	1	Finish WebApps	High	4	

The requirements page also lists the created requirements for the group. New requirements can be created by the group leader by clicking on the link indicated by marker 1 on the above figure.

4.3.4 Tasks

Group leaders are (exclusively) given the ability to create new tasks and cancel tasks no longer required. Tasks are created with the following information:

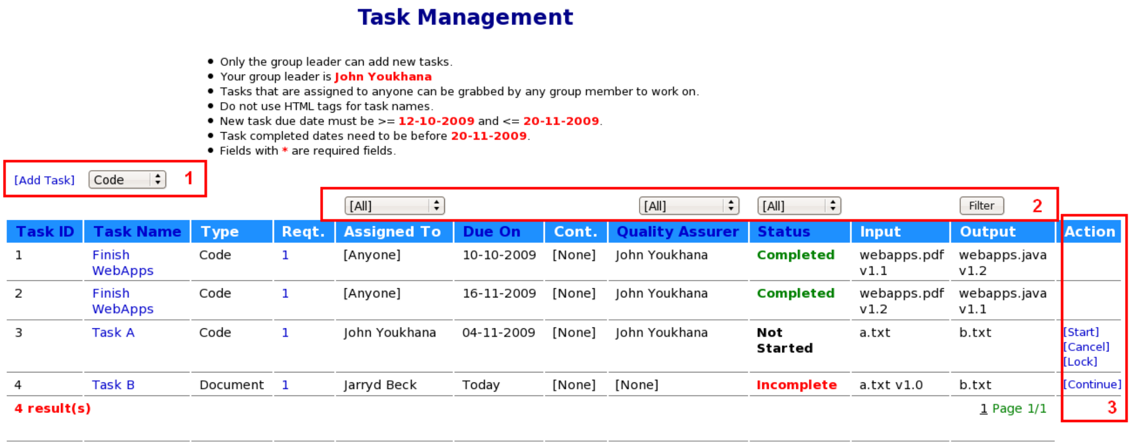
- ID
- Name
- Type (code, document, testing task)
- Related requirement
- Due date
- Assignee
- Quality assurer

The assignees available depend on the roles that the group members have been assigned. Code tasks may only be assigned to those with the developer role, and similarly with testing tasks and documenting tasks with the tester and documenter roles, respectively. Assigning a quality assurer functions the same way in that only those with the quality assurer role within the group can be assigned as QA to a task.

The due date is restricted to dates between the current date and the end of the course's session.

Once a task is created and assigned, the current student (i.e. the one currently assigned an incomplete task, or the QA for a completed but unapproved task) is able to update the task's status with a context-sensitive link that will appear in the task list, e.g. start task, submit task, approve task, cancel task. Note that the group leader is also permitted to perform QA on tasks.

Tasks are also attached to an input and output file name, which are entered as links to files in a file repository when the DoCo extension is included, though DoCo features are out of scope for our WebPMS contribution.



Task Management

- Only the group leader can add new tasks.
- Your group leader is John Youkhana
- Tasks that are assigned to anyone can be grabbed by any group member to work on.
- Do not use HTML tags for task names.
- New task due date must be >= 12-10-2009 and <= 20-11-2009.
- Task completed dates need to be before 20-11-2009.
- Fields with * are required fields.

Task ID	Task Name	Type	Reqt.	Assigned To	Due On	Cont.	Quality Assurer	Status	Input	Output	Action
1	Finish WebApps	Code	1	[Anyone]	10-10-2009	[None]	John Youkhana	Completed	webapps.pdf v1.1	webapps.java v1.2	
2	Finish WebApps	Code	1	[Anyone]	16-11-2009	[None]	John Youkhana	Completed	webapps.pdf v1.2	webapps.java v1.1	
3	Task A	Code	1	John Youkhana	04-11-2009	[None]	John Youkhana	Not Started	a.txt	b.txt	[Start] [Cancel] [Lock]
4	Task B	Document	1	Jarryd Beck	Today	[None]	[None]	Incomplete	a.txt v1.0	b.txt	[Continue]
4 result(s)										1 Page 1/1	

In the task management page the group leader can add a task by selecting a task type and clicking on the link as indicated by marker 1 on the above figure. Any member of the group can filter the tasks being displayed by selecting the filter and pressing the "'filter"' button as indicated by marker 2:

Searches can be filtered by:

- The student that is assigned to the task.
- The student that is assigned as the quality assurer.
- The status of the task.

The group leader can perform actions on in progress tasks as indicated by marker 3 on the above figure. These actions include:

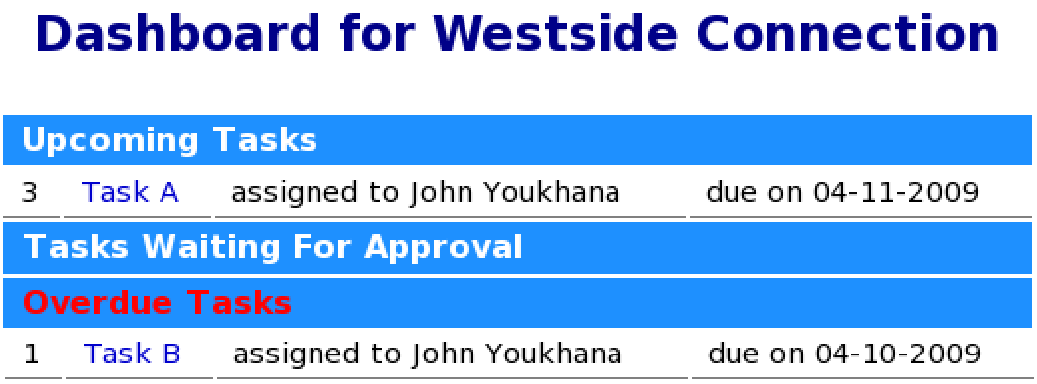
- Start: Start progress on the task.

- Cancel: Cancel the task.
- Approve: The quality assurer can approve a task once it has been submitted.
- Lock: After a while a task can be locked and no work can continue on the task.
- Continue: When a task is incomplete it can be continued with a new task.
- Grab: When a task is assigned to ”’Anyone”’, any student can grab the task and work on the task.
- Complete: Once a task is submitted and approved if required, the task can be completed.
- Update Cancelled Reason: Update the reason why the task was cancelled.
- Submit: Submit a task that is completed by the assigned student.

4.3.5 Dashboard

All students have access to the ”dashboard”. This page shows a quick view of all of the current tasks within a group, merely showing the task name and due date, with links to the tasks. The tasks are separated into three different sections:

- Upcoming tasks (tasks due in the near future)
- Tasks waiting for approval (submitted but not yet quality assured)
- Overdue tasks (incomplete tasks that have passed their due date)



Dashboard for Westside Connection

Upcoming Tasks			
3	Task A	assigned to John Youkhana	due on 04-11-2009
Tasks Waiting For Approval			
Overdue Tasks			
1	Task B	assigned to John Youkhana	due on 04-10-2009

The dashboard will display tasks that are:

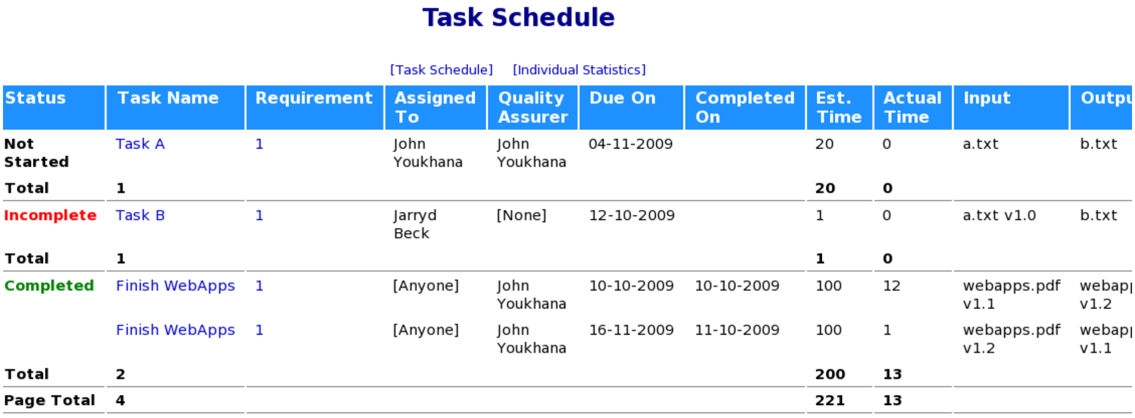
- Upcoming that is that haven’t yet been started.

- Tasks that require approval before being completed.
- Tasks that are overdue.

Once tasks have the status of complete they will not be displayed on the dashboard.

4.3.6 Task Schedule and Statistics

As mentioned in the previous staff functionality section, students are also provided access to the complete task schedule and accompanying statistics for their group.



Task Schedule

[Task Schedule] [Individual Statistics]

Status	Task Name	Requirement	Assigned To	Quality Assurer	Due On	Completed On	Est. Time	Actual Time	Input	Outpu
Not Started	Task A	1	John Youkhana	John Youkhana	04-11-2009		20	0	a.txt	b.txt
Total	**1**						**20**	**0**		
Incomplete	Task B	1	Jarryd Beck	[None]	12-10-2009		1	0	a.txt v1.0	b.txt
Total	**1**						**1**	**0**		
Completed	Finish WebApps	1	[Anyone]	John Youkhana	10-10-2009	10-10-2009	100	12	webapps.pdf v1.1	webap v1.2
	Finish WebApps	1	[Anyone]	John Youkhana	16-11-2009	11-10-2009	100	1	webapps.pdf v1.2	webap v1.1
Total	**2**						**200**	**13**		
Page Total	**4**						**221**	**13**		

5 System Evaluation And Analysis

5.1 System Evaluation and Analysis Introduction

In this section we evaluate the system we have produced in three ways: functionality provided vs. functionality we desired, system reliability, and system usefulness.

The functionality evaluation is done by mapping our requirements (in Appendix A: WebPMS Requirements) to comments on their status within our system (satisfied, partially satisfied, moved to a future enhancement, etc).

System reliability is demonstrated via the significant scenario-based testing we have performed, and the results are contained in the second sub-section of the evaluation.

The system's usefulness is evaluated along with the conclusions drawn from our evaluation work on WebPMS in this section's conclusion.

5.2 Requirement Satisfaction

The below table indicates the current (after our work) and prior (before our work) satisfiability status of each functional requirement WebPMS had, i.e. if the requirement has been satisfied by our solution, partially satisfied and needs to be improved upon later (in another thesis project), deemed unnecessary, etc. The "Satisfied by" column refers to the functionality section in the user manual of this report.

Req ID	**Prior Satisfiability Status**	**Current Status**	**Satisfied by**
F1.1	Satisfied	Satisfied	4.2.3 Group Member Roles and Group Leader Assignment (staff), 4.3.2 Group Member Roles (student)

F1.2	Partially satisfied; staff assignment system needed a much cleaner interface	Satisfied	4.2.3 Group Member Roles and Group Leader Assignment (staff), 4.3.2 Group Member Roles (student)
F1.3	Largely unsatisfied	Satisfied	4.3.2 Group Member Roles (student), 4.2.3 Group Member Roles (staff)
F1.3.1	Partially satisfied; not a proper ”role” that a student could have (leader did QA)	Satisfied	4.3.2 Group Member Roles
F1.3.2	Unsatisfied	Satisfied	4.2.3 Group Member Roles and Group Leader Assignment (staff), 4.3.2 Group Member Roles (student)
F1.3.3	Unsatisfied	Satisfied	4.2.3 Group Member Roles and Group Leader Assignment (staff), 4.3.2 Group Member Roles (student)
F1.3.4	Unsatisfied	Satisfied	4.2.3 Group Member Roles and Group Leader Assignment (staff), 4.3.2 Group Member Roles (student)

F1.3.5	Unsatisfied	Satisfied	4.2.3 Group Member Roles and Group Leader Assignment (staff), 4.3.2 Group Member Roles (student)
F1.3.6	Unsatisfied	Satisfied	4.2.3 Group Member Roles and Group Leader Assignment (staff), 4.3.2 Group Member Roles (student)
F1.4	Partially satisfied	Partially satisfied; DoCo related features have been descoped from our project	4.3.1 Group Registration, 4.3.2 Group Member Roles, 4.3.4 Tasks
F1.4.1	Partially satisfied; no roles	Satisfied	4.3.1 Group Registration, 4.3.2 Group Member Roles
F1.4.2	Partially satisfied; what was done had this constraint	Satisfied	4.3.1 Group Registration, 4.3.2 Group Member Roles, 4.3.4 Tasks
F1.4.3	Partially satisfied; does not interface with WebCMS2	DoCo related features have been descoped from our project	
F2.1	Partially satisfied	Satisfied	4.3.2 Group Member Roles, 4.3.4 Tasks
F2.1.1	Satisfied	Satisfied	4.3.4 Tasks
F2.1.2	Partially satisfied; could assign to anyone (should be restricted by roles)	Satisfied	4.3.2 Group Member Roles, 4.3.4 Tasks
F2.1.3	Partially satisfied; anyone could have been QA	Satisfied	4.3.2 Group Member Roles, 4.3.4 Tasks
F2.1.4	Satisfied	Satisfied	4.3.4 Tasks
F2.1.5	Satisfied	Satisfied	4.3.4 Tasks

F2.2	Partially satisfied	Partially satisfied; DoCo related features have been descoped from our project	4.3.2 Group Member Roles, 4.3.4 Tasks, 4.3.6 Task Schedule and Statistics
F2.2.1	Partially satisfied (no QA and task type data)	Satisfied	4.3.2 Group Member Roles, 4.3.4 Tasks, 4.3.6 Task Schedule and Statistics
F2.2.2	Partially satisfied	DoCo related features have been descoped from our project	
F2.2.3	Partially satisfied; unproven	Partially satisfied	
F2.2.3.1	Partially satisfied; unproven	Satisfied through scenario testing	Scenario testing, 4.3.6 Task Schedule and Statistics
F2.2.3.2	Partially satisfied; unproven	Satisfied through scenario testing	Scenario testing, 4.3.6 Task Schedule and Statistics
F2.2.3.3	Partially satisfied; unproven	Satisfied through scenario testing	Scenario testing, 4.3.6 Task Schedule and Statistics
F2.2.3.4	Partially satisfied; unproven	DoCo related features have been de-scoped from our project	
F2.2.3.5	Unsatisfied	DoCo related features have been descoped from our project	
F2.2.4	Unsatisfied	Unsatisfied; deemed unnecessary / future enhancement	
F2.3		Partially satisfied	
F2.3.1	Partially satisfied; QA was not assigned to individual students	Satisfied; alert is simply by it being available for the QA'er and on their dashboard	4.3.2 Group Member Roles, 4.3.4 Tasks, 4.3.5 Dashboard

F2.3.2	Partially satisfied; no notification	Mostly satisfied; can only approve (must inform task assignee manually – denying was deemed unnecessary)	4.3.2 Group Member Roles, 4.3.4 Tasks
F2.3.3	Satisfied	Satisfied	4.3.4 Tasks
F2.3.4	Partially satisfied; QA was not assigned to individual students	Satisfied	4.3.2 Group Member Roles, 4.3.4 Tasks
F2.3.5	Satisfied	DoCo related features have been de-scoped from our project	
F2.3.6	Partially satisfied; some cases	Satisfied	4.3.4 Tasks
F2.4	Satisfied	Satisfied	4.3.4 Tasks, 4.3.5 Dashboard, 4.3.6 Task Schedule and Statistics
F2.5	Partially satisfied	Mostly satisfied	4.3.4 Tasks, 4.3.6 Task Schedule and Statistics
F2.5.1	Satisfied	Satisfied	4.3.4 Tasks, 4.3.6 Task Schedule and Statistics
F2.5.2	Partially satisfied; no QA data and the data here was wrong (totalled all courses, not just the current one)	Satisfied; QA data added and bug fixed so that this data is course-dependent	4.3.4 Tasks, 4.3.6 Task Schedule and Statistics
F2.5.3	Unsatisfied	Satisfied	4.3.4 Tasks, 4.3.6 Task Schedule and Statistics
F3.1	Satisfied	DoCo related features have been de-scoped from our project	
F3.2	Satisfied	”	
F3.3	Satisfied	”	
F3.4	Unsatisfied	”	
F3.4.1	Unsatisfied	”	

F4.1	Unsatisfied	Assessment has been descoped; to be done in Alison Kao’s thesis project	
F4.1.1	Unsatisfied	”	
F4.1.2	Unsatisfied	”	
F4.1.3	Unsatisfied	”	
F4.2	Unsatisfied	”	
F4.2.1	Unsatisfied	”	
F4.2.2	Unsatisfied	”	
F4.2.3	Unsatisfied	”	
F4.2.4	Unsatisfied	”	
F4.2.5	Unsatisfied	”	
F5.1	Unsatisfied	Satisfied	4.2.1 Admin For The WebPMS Module
F5.2	Unsatisfied	Satisfied	4.2.3 Group Member Roles (staff), 4.3.2 Group Member Roles (student)

5.3 Scenario-based Testing

With two weeks until the due date, we started undertaking scenario-based tests on all areas of WebPMS. The results follow (Tests section), along with the defects found and how they were resolved (Defect resolution section) within that two week period. Note that we had done informal testing along the way as each new feature was implemented, so the majority of functions were expected to work at this point in time. Note that since these tests were not all done at the same time, the results of some tests are passes despite previous tests that are prerequisites failing.

5.3.1 Tests

1. Navigation Menu Tests

Scenario	Context	Input data	Expected Result	Pass/Fail + Reason For Failure

1.1 Log in as staff	Course = COMP9999
	session = 09s2
		u: webcms
		p/w:abc123
			WebPMS and relevant sub-links including admin-only ones are displayed	Pass

1.2 Log in as student	Course = COMP9999
	session = 09s2
		u: jyou147
		p/w:xyz123
			WebPMS and relevant sub-links not including admin-only ones are displayed	Pass

2. WebPMS Settings Tests

Scenario	Context	Input data	Expected Result	Pass/Fail + Reason for failure

2.1 View/edit settings with course staff	Course = COMP9999
	session = 09s2
		-	WebPMS link allows you to edit WebPMS settings	Pass

2.2 View/edit settings with student	Course = COMP9999
	session = 09s2
		-	WebPMS link is not available to students	Pass

2.3 Set to light mode with no roles	Course = COMP9999
	session = 09s2
		Type = Light
		Roles = false
			Group registration and groups admin are the only links under WebPMS	Fail - missing groups admin link

2.4 Set to light mode with roles	Course = COMP9999
	session = 09s2
		Type = Light
		Roles = true
			Group registration, groups admin and group roles all appear as links under WebPMS	Fail - missing groups admin link

2.5 Set to full mode with no roles	Course = COMP9999
	session = 09s2
		Type = Full
		Roles = false
			All WebPMS links except group roles appear under WebPMS
			Tasks page and schedule do not have QA
				Pass

2.6 Set to full mode with roles	Course = COMP9999
	session = 09s2
		Type = Full
		Roles = true
			All WebPMS links including group roles appear under WebPMS
			Tasks page and schedule have QA options/columns
			Tasks page has a drop-down to select the type of task to create
				Pass

2.7 Change WebPMS settings in one course and check they do not affect other courses	Course = COMP9999 (full, roles on)
	check course = COMP0000 (full, roles on)
		Type = Light
		Roles = false
			COMP0000 stays at full level with roles on	**Fail** - change turns out to be global

3. Group Registration Tests

Scenario	**Context**	**Input data**	**Expected Result**	**Pass/Fail + Reason For Failure**
3.1 Create group	Four students enrolled in course COMP9999	Assign 3 students to group name = ”Westside Connection”	New group created; 3 students added	Pass

3.2 Create second group	Four students enrolled in course COMP9999 3 in other group	Assign 1 student to group name = ”Group 2”	Only the unassigned student is available to assign to the new group	Pass
3.3 Attempt to create group with no remaining students	All students enrolled in COMP9999 already assigned groups	Group name = ”Group 3”	Group cannot be created with no members	Pass
3.4 Drop lone student from group	Single student is enrolled in the group	-	Group is deleted when student is removed	Pass
3.5 Add a student from a dropped group into another group	Single student has been removed from group	Add student to ”Westside Connection”	Student is available to be added to a group again	Pass
3.6 Edit group name	Group ”Westside Connection” exists	group name = ”Westside Connection Edit”	Group name is successfully changed instantly	Pass

4. Group Roles Tests (including leadership assignment)

Scenario	**Context**	**Input data**	**Expected Result**	**Pass/Fail + Reason For Failure**
4.1 Check default roles	Two groups have just been set up	-	All students have default role ”No Role”	Pass

4.3 Take leadership	Group is set up that has no leader Logged in as jyou147	-	Group roles page should display John Youkhana as leader	Pass
4.4 Assign leadership as leader	Group is set up with John Youkhana as leader Logged in as jyou147	New leader = Mark Lockery	Group roles page should display Mark Lockery as leader	Pass
4.5 Assign leadership as staff	Group is set up with Mark Lockery as leader Logged in as webcms	New leader = John Youkhana	Group roles page should display John Youkhana as leader	**Fail** - No students appear in the drop-down box for staff
4.6 Assign roles as staff	Two groups have just been set up Logged in as webcms2	Role for John Youkhana = Developer Role for Mark Lockery = Documenter	Respective roles should be assigned to John and Mark	**Fail** - Assign/edit/remove role links did not show up for staff
4.7 Assign roles as student (leader)	Group is set up with John Youkhana as leader Logged in as jyou147	Role for Jarryd Beck = Quality Assurer	Role should be assigned to Jarryd	Pass
4.8 Edit/remove roles as staff	Groups as after 4.7	Remove role for John Youkhana Role for Mark Lockery = Developer	Role should be assigned to Mark; John = ”No Role”	**Fail** - Assign/edit/remove role links did not show up for staff

4.9 Edit/remove roles as student (leader)	Groups as after 4.7	Remove role for John Youkhana Role for Mark Lockery = Developer	Role should be assigned to Mark; John = ”No Role”	Pass

5. Group Leader’s Rights Tests

Scenario	**Context**	**Input data**	**Expected Result**	**Pass/Fail + Reason For Failure**
5.1 Group leader can change roles	jyou147 = leader mloc835 is not	-	jyou147 Can see assign/edit/remove roles links; mloc835 can not	Pass
5.2 Group leader can add requirements	jyou147 = leader mloc835 is not	-	jyou147 Can see add requirement link; mloc835 can not	Pass
5.3 Group leader can add tasks	jyou147 = leader mloc835 is not	-	jyou147 Can see add task link; mloc835 can not	Pass

5.4 Group leader can approve all tasks (do QA)	Task created and completed and given QA to another student Logged in as jyou147 (leader)	-	Link for approving task appears for leader	Pass

6. Requirements Tests

Scenario	**Context**	**Input data**	**Expected Result**	**Pass/Fail + Reason For Failure**
6.1 Create new requirement	-	1, TestReq1, High priority	Requirement should be created with details mentioned	Pass
6.2 Create new sub-requirement	TestReq1 as above	2, TestReq1.1, Medium priority	Requirement should be created with details mentioned and attached to TestReq1, thus only visible after expanding TestReq1	Pass
6.3 Remove requirement not linked to task(s)	Test Req2 has no tasks attached to it	-	Remove link should delete this requirement and it should be gone from the page	Pass

6.4 Attempt to edit/remove requirement linked to task(s)	Test Req1 has 8 tasks attached to it	-	No edit or remove links are present for this requirement	Pass
6.5 Edit requirement	Have Req2, Req2.1 which has no parent	Set Req2 to be Req2.1's parent	Req2 should now have Req2.1 attached as a child and Req2.1 should not be visible on the page until you expand Req2	Pass
6.6 Check requirement task counter	Test Req1 has 8 tasks attached to it	-	Number of attached tasks value should display: 8	Pass
6.7 View requirement	Test Req1 has data from 6.1	-	View link should show all information for this requirement including created and modified dates	Pass

7. Tasks Tests

Scenario	**Context**	**Input data**	**Expected Result**	**Pass/Fail + Reason For Failure**
7.1 Create task without type (roles off)	Roles turned off	Create task "test task 1" due in a week	Task created and displayed in the table with relevant info	**Fail** - Could not create task due to invalid due date

7.2 Create task with type (roles on)	Roles turned on	Create code-type task ”test code task 1” due in a week	Task created and displayed in the table with relevant info	**Fail** - Could not create task due to invalid due date
Note: testing was halted here and proceeded after the above issues were dealt with				
7.3 Cancel task	Task created ”Testeriffic”	Set status = ”Cancelled”	Task’s status turns to green text displaying ”Cancelled” Leader/assignee can update cancelled reason	Pass
7.4 Start task	Task created ”a”; not started but is assigned to John Youkhana	Set status = ”Start”	Task’s status changes to ”In progress”	Pass
7.5 Continue task	Task created ”2” and locked whilst incomplete	Set status = ”Continue”	New task can be created to continue this one New task has a link to the previous task in the ”Cont.” column	Pass

7.6 Task becomes overdue	Task created with due date 27/09/2009 ”new one”	Viewed after 27/09/2009 with task not yet started	Task is marked as overdue	Pass
7.7 Task needs continueing	Task created with due date 15/09/2009 ”test code”	Viewed after 15/09/2009 with task started but not submitted	Task now requires you to ”continue” it through a new task	Pass
7.8 Submit task for approval	Task created with assignee = John Youkhana ”a”	Status = ”Submit”	Task status goes to submitted; QA and group leader can now approve the task	Pass
7.9 Approve task with QA	Task created with QA = John Youkhana	Status = ”Approve” Logged in as jyou147	Task status should change to ”Completed”	Pass
7.10 Approve task with leader	Task created with QA = Jarryd Beck	Status = ”Approve” Logged in as jyou147 (leader)	Task status should change to ”Completed”	Pass
7.11 Filter task list	List of 8 tasks – 4 of which are assigned to John Youkhana, 1 has him as QA, 5 total overdue tasks	Filter by assigned to = John Youkhana Repeat for QA and status column tests	Only relevant tasks should be displayed	Pass

7.12 Created task must be within date range of current session	Session is marked as 1 October - 30 December 2009	Due date = 1 Jan 2010 (and try a few others outside of the date range)	Task should not be able to be created	Pass

8. Dashboard Tests

Scenario	**Context**	**Input data**	**Expected Result**	**Pass/Fail + Reason For Failure**
8.1 Dashboard correctly displays ”Upcoming Tasks”	There are incomplete tasks for group ”Westside Connection” with their due date still in the future	-	The incomplete tasks that have not reached their due date are displayed on the dashboard page under ”Upcoming Tasks”	Pass
8.2 Dashboard correctly displays ”Tasks Waiting For Approval”	There are completed tasks for group ”Westside Connection” that have not been approved	-	The unapproved tasks are displayed on the dashboard page under ”Tasks Waiting For Approval”	Pass
8.3 Dashboard correctly displays ”Upcoming Tasks”	There are incomplete tasks for group ”Westside Connection” that have reached their due date	-	The incomplete tasks that have reached their due date are displayed on the dashboard page under ”Overdue Tasks”	Pass

8.4 Dashboard only displays relevant tasks	There are completed and approved tasks for group ”Westside Connection”	-	The completed and approved tasks do not appear anywhere on the dashboard	Pass

9. Task Schedule / Statistics Tests

Scenario	**Context**	**Input data**	**Expected Result**	**Pass/Fail + Reason For Failure**
9.1 Check that all tasks in task management are present in task schedule	8 tasks exist for ”Westside Connection”, including some of each type (incomplete, overdue, complete, etc)	-	All 8 tasks should be displayed on the task schedule page	Pass
9.2 Check that task links work within the task schedule	As in 9.1	-	The relevant task information should be displayed for any task link clicked on	Pass

9.3 Check that the tasks are separated into groups based on their status	As in 9.1	-	5 Tasks should be marked as overdue, 1 complete, 1 incomplete green (continued in a later task), 1 incomplete red as per task management page	Pass
9.4 Check that sub-totals (by status) and page totals are correct	As in 9.1	-	Sub-totals and totals should be equal to the sum of the number of tasks, estimated times, and actual times for both status groups and the whole page	Pass
9.5 Check that all relevant students (and only those) are present in individual statistics page	”Westside Connection” group has 3 members: Jarryd Beck, John Youkhana, Mark Lockery Logged in as jyou147	-	Only the previously mentioned three students should appear	Pass
9.6 Check that data in individual statistics is consistent with the tasks page’s data	8 tasks exist for ”Westside Connection”, including some of each type (incomplete, overdue, complete, etc)	-	Task counts under each heading (in progress, completed, overdue, etc) should match data on the tasks page	**Fail** - There is more data than the tasks page has

10. Staff Access Tests

Scenario	Context	Input data	Expected Result	Pass/Fail + Reason For Failure
10.1 Staff can access group registration page with write-access	pms_setting = full, roles	-	Can access page and make changes	Pass
10.2 Staff can access group roles page with write-access	pms_setting = full, roles	-	Can access page and make changes	Pass
10.3 Staff can access tasks admin and groups admin pages and change settings	pms_setting = full, roles	-	Can access page and make changes	Pass
10.4 Staff given read-only access to requirements pages for each group	pms_setting = full, roles	-	Can view requirements for each group but not make changes	Pass
10.5 Staff given read-only access to tasks pages for each group	pms_setting = full, roles	-	Can view tasks for each group but not make changes	Pass

10.6 Staff can access task schedule pages for each group	pms_setting = full, roles	-	Can view task schedules for all groups	Pass
10.7 Staff can access dashboard pages for each group	pms_setting = full, roles	-	Can view dashboards for all groups	Pass

5.3.2 Defect Resolution

Scenario	Defect	Cause of defect	Fix Required	Status
2.3 Set to light mode with no roles	Groups admin link does not show up	Menu PHP code was checking if the link was named ”Group Management”... once we changed its name it stopped appearing and we hadn’t noticed	Change PHP to check for ”Groups Admin”	Fixed
2.4 Set to light mode with roles	Groups admin link does not show up	Menu PHP code was checking if the link was named ”Group Management”... once we changed its name it stopped appearing and we hadn’t noticed	Change PHP to check for ”Groups Admin”	Fixed
2.7 Change WebPMS settings in one course and check they do not affect other courses	Changing settings (type = full/light, roles on/off) in one course affects all courses	DB table pms_setting has no link to course	Add foreign key ”courseid” to pms_setting table and reference it accordingly in php code	Fixed

4.5 Assign leadership as staff	No students appear in the drop-down box for staff	Code was using the group ID of the user logged in to get the students list... this does not work for staff	Separate interface needs to be created - put ”assign as group leader” link next to each student on the group roles page when logged in as staff	Fixed
4.6 Assign roles as staff	Assign/edit/remove role links did not show up for staff	Links for roles were only set up in the PHP code to display for group leaders	Alter check to check if the current user is either group leader or staff	Fixed
4.8 Edit/remove roles as staff	Assign/edit/remove role links did not show up for staff	Links for roles were only set up in the PHP code to display for group leaders	Alter check to check if the current user is either group leader or staff	Fixed
7.1 Create task without type (roles off)	Could not create task due to invalid due date	The WebCMS course session we had created expired on September 30 – a mistake on our part but not an issue with WebPMS	Create a new session and put a new test course under that	Not a defect

7.2 Create task with type (roles on)	Could not create task due to invalid due date	The WebCMS course session we had created expired on September 30 – a mistake on our part but not an issue with WebPMS	Create a new session and put a new test course under that	Not a defect
9.6 Check that data in individual statistics is consistent with the tasks page's data	There is more data than the tasks page has	The DB queries are not specifying a course or group (only a student), so it is getting the total of all of their tasks across all courses	Make DB lookup course-dependent or group-dependent. Group is already listed in the task tables so we'll use that.	Fixed

As shown above, no discovered defects are presently outstanding.

5.4 Conclusions From System Evaluation and Analysis

By meeting a majority of the system requirements we gathered during Thesis A, with the remainder of the requirements (an isolated subset) planned for a future thesis project, we believe that our system provides *useful* functionality to the staff and students working on the software engineering workshops run at UNSW (with potential for other courses to use it too). Requirements were gathered from and discussed with university staff that have run the software engineering workshops (namely Albert Nymeyer and Peter Ho), as well as students that have been through the workshops (including the two of us). As such, the list of requirements produced came out as a practical list of desired functions that we have seen uses or needs for in the past. In other words, the requirements are based on experience rather than just guesswork or a rough idea of an ideal situation.

Our scenario-based testing work aided us in identifying potential code and database issues and correct our implementation in cases where code/database defects were found. An exhaustive set of scenario-based tests has allowed us to provide a greater guarantee that our system will not only do what is specified, but is reliable and robust.

6 Conclusions and Future Work

6.1 Conclusions

The aim of this project as stated in the introduction was essentially to produce an immediately usable, integrated set of project management tools to aid staff and students in performing and keeping track of project management tasks within the software engineering workshops run by CSE (with potential for the tools to be used by any course within CSE). This was to be achieved by building on the tools provided by the WebGMS (group management) and WebTMT (task management) modules and integrating the set of tools with the WebCMS2 course management system used already by CSE.

We have successfully implemented a multi-level system (customizable to provide full-functionality or only a subset of options if you do not need the full feature set) that meets the majority of our requirements including features for group management, group roles assignment/management (including assignment of a group leader), creation and management of tasks including quality assurance processes, automatic generation of task schedules and statistics, and the separation of functionality (editable / read-only) based on group roles and the staff/student viewer type.

Via the "System Evaluation and Analysis" section in this report, we have demonstrated that our system has met the majority of our requirements and that the functionality provided has been thoroughly tested with scenario-based testing (most appropriate for highly-interactive systems like this) with no outstanding discovered defects at this point. We also feel that as our requirements were gathered from staff (Albert Nymeyer and Peter Ho) and students (ourselves) that have been through the software engineering workshops, we were able to produce an appropriate set of requirements and as such a system satisfying these requirements would be *useful* to the intended set of users.

As a result of time constraints we found ourselves having to de-scope functionality relating to staff assessment of students and groups, as this would require significant research as well as development time. Also, the DoCo-related file repository functionality we had access to was not integrated with our system. The failure to satisfy these requirements in our thesis project is noted in the "System Evaluation and Analysis" section of this report, and they are planned as future additions to WebPMS, as noted in the next sub-section.

6.2 Future Work and Additions to WebPMS

Several enhancements are planned (as thesis projects) to build on the current functionality of the system we have assembled, in order to provide a more complete set of tools for CSE staff and students. These enhancements come from our requirements (see Appendix A: WebPMS Requirements) that were not satisfied in this thesis project, primarily due to time constraints. Major additions include:

- Reintroduce DoCo functionality for file repository

  DoCo, a 2008 thesis project, provided basic file repository functions including adding versioned files to a repository, checking out files, locking files, etc. It was also designed to interact with the tasks created in WebTMT, so that you could provided links to files hosted in DoCo as the input and output files related to tasks created in WebTMT.

  Whilst DoCo functionality was already provided and it was able to interact with WebTMT prior to our project, this connectivity was not deemed an important part of our project as none of our additions would affect the interface between DoCo and WebTMT. Time constraints meant that we felt we were better served devoting our time to new additions and restructuring the WebTMT and WebGMS systems into WebPMS than concerning ourselves with configuring the DoCo file repository to function with the task management system and testing the subsequent interactions.

  The DoCo functionality will likely be added in and tested with the new WebPMS system in a thesis project in the near future.

- Staff assessment of both group and individual work built in to the system

  A section of our requirements based on staff assessment of the groups and individuals within those groups has to be de-scoped from our thesis project due to time constraints. Ideally, WebPMS will provide functions for staff to assign marks to students based for the following:

- Task Schedules

  This would be a mark assigned to the entire group for the production of their task schedule on a weekly or other basis.

- Individual Contribution

  This would be a mark assigned to each individual student based on how much they are shown to have contributed to their group's efforts. This could possibly even be automatically generated based on the statistics that can be found in WebPMS's "Individual Statistics" page under "Task Schedule" (and possibly other data).

- Output Files / Deliverables

  With DoCo functionality, WebPMS can accept links to output files for the tasks entered into the system. As such, it would be possible for staff to view these and thus assign marks to and annotate these submitted files, i.e. with some minor additions WebPMS could be used as an assignment submission system as well as a project management system.

This functionality is currently slated to be added in a thesis project by Alison Kao, to be concluded in session 1 of 2010.

7 References

1. Microsoft 2009, *Microsoft Office Project Standard 2007 top 10 benefits*, Microsoft, accessed 29 March 2009,
<http://office.microsoft.com/en-us/project/HA101650291033.aspx>

2. Chan Xuong Vien 2008, 'Task Management Tool', Honours thesis, University of New South Wales.

3. Daniel Joseph Troy 2008, 'Document Collaboration Online System', Honours thesis, University of New South Wales.

4. John Shepherd 2009, 'Programming with Databases', COMP3311 course notes, University of New South Wales,
<http://www.cse.unsw.edu.au/ cs3311/09s1/notes/J/notes-6up.pdf>

A WebPMS Requirements

The priorities used below are as follows:

1 - Essential. Must be implemented.

2 - Desired. Implement as many of these as possible, based on time constraints.

A.1 Functional Requirements

Group Management

F1.1: Add and remove students from project groups.
Priority: Essential

F1.2: Allow students to take/assign leadership and staff to elect a student in project groups as team leader.
Priority: Essential

F1.3: Allow group leader to assign roles to students in project groups.
Priority: Essential

> F1.3.1: Allow team leader to assign a project group member as the quality assurance role for each task.
> Priority: Essential
>
> F1.3.2: Allow team leader and staff to assign a project group member as the testing role.
> Priority: Essential
>
> F1.3.3: Allow team leader and staff to assign a project group member as the documentation role.
> Priority: Essential
>
> F1.3.4: Allow team leader and staff to assign a project group member as the developer role.
> Priority: Essential
>
> F1.3.5: Allow team leader and staff to change already assigned roles.
> Priority: Desired

F1.3.6: Allow team leader and staff to remove already assigned roles.

Priority: Desired

F1.4: Group system must be integrated with the rest of the system to allow group-specific content (tasks, repository).

Priority: Essential

F1.4.1: Students can only see their own group members and their roles.

Priority: Essential

F1.4.2: Students can only create/view/update the tasks and task schedules belonging to their group.

Priority: Essential

F1.4.3: Students can only interact with their group's file repository.

Priority: Essential

Project Management

F2.1: Allow project group's team leader to create, edit and cancel "tasks".

Priority: Essential

F2.1.1: Give tasks a name and description, as well as an input and output file.

Priority: Essential

F2.1.2: Allow team leader to assign tasks to relevant group members based on their roles.

Priority: Essential

F2.1.3: Assign quality assurers to tasks (Quality assurers are just students that have been assigned the role of quality assurance).

Priority: Essential

F2.1.4: Assign tasks desired completion times (deadlines).
Priority: Essential

F2.1.5: Cancel tasks deemed unnecessary.
Priority: Essential

F2.2: Generate a ”task schedule” document.
Priority: Essential

F2.2.1: Generate a document with relevant information from all created tasks (task description, assigned to, QA, started on, status, ETA, etc).
Priority: Essential

F2.2.2: Have links from the task output file columns in the task schedules to the corresponding file in the repository.
Priority: Essential

F2.2.3: Perform basic sanity checks on the task schedule.
Priority: Desired

F2.2.3.1: Ensure correct format of the task schedule with appropriate columns... task ID, name, assigned to, QA, status, input/output files, etc.
Priority: Desired

F2.2.3.2: Ensure that the total hours match the sum of hours spent on all tasks.
Priority: Desired

F2.2.3.3: Ensure that all tasks have values in mandatory fields (assigned to, QA, time spent, deadline).
Priority: Desired

F2.2.3.4: Ensure that all completed tasks have an output file attached to them.
Priority: Desired

F2.2.3.5: Ensure that files uploaded/submitted to the repository are non-empty once task schedules are submitted.
Priority: Desired

F2.2.4: "Submit" a task schedule for the current week.
Priority: Desired

F2.3: Provide facilities to update a task's progress (to be done by the group leader, the member assigned the task, and the quality assurer for task).
Priority: Essential

F2.3.1: Mark as ready for QA; alert quality assurer
Priority: Essential

F2.3.2: Mark as passed/failed QA; notify member assigned to task if the task has failed QA.
Priority: Essential

F2.3.3: Mark as submitted and "submit" the output.
Priority: Essential

F2.3.4: Ensure that tasks cannot be marked complete and code cannot be submitted until it has been signed off by its assigned quality assurer.
Priority: Essential

F2.3.5: Ensure that tasks' output cannot be edited anymore once a task is marked 100% complete and submitted.
Priority: Desired

F2.3.6: Colour-code tasks based on status - green for complete/cancelled/incomplete but continued via a later task, red for incomplete and past the deadline, blue for in progress and the deadline has not passed.
Priority: Desired

F2.4: Provide ability for students to view tasks related to them (assigned to them or they are assigned the QA role)
Priority: Desired

F2.5: Provide statistics on each group's task schedules.
Priority: Desired

> F2.5.1: Provide summation statistics of estimated time and actual time spent on tasks, both for separated by status and for page totals.
> Priority: Desired
>
> F2.5.2: Provide individual statistics showing the number of tasks each member of a group has that are not started, in progress, incomplete, overdue, completed, quality assured, etc.
> Priority: Desired
>
> F2.5.3 Calculate total hours contributed by each student; display along with assigned hours.
> Priority: Desired

File Repository and Management

F3.1: Add/remove files to/from a repository.
Priority: Essential

F3.2: Upload new versions of files with version number; store previous versions.
Priority: Essential

F3.3: Lock files currently being edited (and track who is editing it).
Priority: Desired

F3.4: Perform basic sanity checks on files submitted or uploaded as output files.
Priority: Desired

> F3.4.1: Ensure that files uploaded/submitted are of the correct file type and name (from the output file linked to the task).

Priority: Desired

Staff Assessment and Task Schedule Tracking

F4.1: Provide staff with the ability to view task schedules for each group.

Priority: Essential

F4.1.1: Staff can select a group and then view their current task schedule.

Priority: Desired

F4.1.2: Staff can select a group and then choose to view their previous task schedules.

Priority: Desired

F4.1.3: Staff can select a group and then choose to view their overall task schedule.

Priority: Essential

F4.2: Provide staff with the ability to assess submitted tasks (marks, comments).

Priority: Desired

F4.2.1: Staff can assign a mark to the output file of submitted tasks (100% complete tasks).

Priority: Desired

F4.2.2: Staff can write comments relating to the output file of submitted tasks (100% complete tasks).

Priority: Desired

F4.2.3: Staff can assign contribution marks to students (i.e. based on how much work appears to have been done by them, judging by the task schedules).

Priority: Desired

F4.2.4: Staff can assign marks to each week's submitted task schedule.

Priority: Desired

F4.2.5: Students can view marks and comments on their tasks and task schedules via the task schedule page.
Priority: Desired

WebPMS Settings

F5.1: Staff can configure the level of WebPMS module functionality.
Priority: Essential

F5.2: Staff can toggle the roles functionality on and off.
Priority: Essential

A.2 Non-Functional Requirements

NF1: Accessibility and availability

NF1.1: Accessible on most computers, regardless of hardware, operating system, or other specifications.

NF1.2: Provided free of charge with no license restrictions.

NF2: Usability

NF2.1: Simple, intuitive interface

NF3: Reliability

NF3.1: Accessible at all times of the day and night with little to no downtime during semesters.

NF4: Documentation

NF4.1: Sufficient documentation on each feature of the system.