

UberCMS

A Unique Solution for Software Engineers

Version 0.5

UberSoft

February 2006



Table of Contents

Executive Summary.....	page 3
Authors.....	page 4
1. Project Drivers.....	page 5
1.1 Purpose.....	page 5
1.2 Vision.....	page 5
2. Functional Requirements.....	page 5
2.1 Atomic Requirements.....	page 5
2.2 Use Cases.....	page 9
2.3 Use Case Diagrams (Scope of Project).....	page 27
3. Non-Functional Requirements.....	page 31
3.1 Prototype Documentation.....	page 31
3.2 Look and Feel.....	page 33
4. Project Issues.....	page 33
4.1 Risks.....	page 33
5. Major Revision History.....	page 34
6. Marketing Assessment and Competitive Review	page 35
6.1 Reviews.....	page 36
6.2 Conclusion.....	page 40
6.3 Works Cited.....	page 40
7. Glossary.....	page 41

Executive Summary

UberSoft is proud to present a new up and coming Content Management System called UberCMS. UberCMS is an innovative, feature packed online group project management system for working with software engineering documents. This system will allow for the management of multiple projects through an efficient and user friendly interface. Each project space will contain its own directory for file storage, announcements, a workspace for software engineering documents, calendar, forum, and contact information. The system can store any file type in a customizable directory. It allows for the editing of rtf documents with templates for requirements and use case formats.

The system will also allow easy communication between members of a project. Project leads and administrators may make announcements for all project members to see. Users may use their private project forum to discuss various aspects of their project or they can send emails using a built in contact list. The contact list will include email addresses and personal information for all the members of a project.

The system allows for strong project management. It has a calendar for each project which has events associated with it. It allows project leads to add events to the calendar to manage milestones and important dates.

Project leads can add announcements that are seen as the users enter a project. This allows for easy and effective communication between project lead and workers.

Additionally, in each project there is a list of all of the members of the project with all of their contact information. This includes everyone's email address with a button to e-mail individual users or all of the users in the project

Full documentation for all file creations, modifications, and version numbers, makes it easy for management to control the quality of your source code with confidence. The ability to roll back versions of files will keep your project secure.

Authors

The members of this company include:

Devon Austen

- Executive Summary
- Prototype
- Prototype Documentation
- Use Cases
- Major Revision History

Daniel Clark

- Executive Summary
- Use Cases
- Marketing Reviews
- Marketing Conclusion

Amy Gill

- Executive Summary
- Requirements
- Use Cases
- Compilation of this Document

Jason Mansfield

- Executive Summary
- Marketing Reviews
- Marketing Conclusion
- Marketing Works Cited

Andrew McNally

- Vision
- Executive Summary
- Requirements
- Use Cases
- Glossary

Jessica Reyes

- Executive Summary
- Requirements
- Use Case Diagram
- Glossary

1. Project Drivers

1.1 Purpose

The purpose of UberCMS is to facilitate the easy manipulation of Software Engineering documents and projects.

1.2 Vision

UberSoft's Content Management System will be used for the organization and management of Software Engineering documents. It will have an appealing user friendly interface and will be easy to use & maintain. The system will support archiving, correspondence among members of a project and organization by project.

2. Functional Requirements

2.1 Atomic Requirements

1 File Requirements

1.1 Store old versions of files.

Use Case: UC1, UC2, UC4, UC5

1.2 Display all files that a user has access to.

Use Case: UC1, UC3, UC5

1.3 Allow users to download accessible files.

Use Case: UC3

1.4 Allow users to edit accessible files.

Use Case: UC1

1.5 Only allow one user to edit a particular file at a time.

Use Case: UC1

1.6 When a user chooses to edit a file, it will be opened in an rtf editor.

Use Case: UC1

1.7 Administrator may unlock a file.

Use Case: UC23

1.8 Administrator may lock a file.

Use Case: UC24

1.9 Allow users to upload accessible files that are not on lockdown.

Use Case: UC2

1.10 Allow users to save files.

Use Case: UC4

1.11 Users may create new files.

Use Case: UC7

1.12 Allow users to open accessible files.

Use Case: UC6

1.13 There are templates for the user to use.

Use Case: UC8, UC9

1.14 Time stamp files upon creation.

Use Case: UC7

1.15 Time stamp files when edited

Use Case: UC1

1.16 Maintain directory structure for file browsing

Use Case: UC11, UC12, UC13

2 Project Leader, Administrator and User Access Rights:

2.1 Allow directory creation.

Use Case: UC11

2.2 Allow directory deletion.

Use Case: UC12

2.3 Allow renaming of directories.

Use Case: UC13

2.4 Allow moving of files from one directory to another.

Use Case: UC10

2.5 System must allow user to login/logout

Use Case: UC33, UC36

2.6 System must have different levels of user access rights: worker, project leader, administration

Use Case:

2.7 Must allow only certain users access to each project

Use Case: UC25

2.8 Must allow Worker write access to all files within their designated projects

Use Case: UC1, UC2, UC4

2.9 Must allow Project leader to control who has access to their project.

Use Case:

2.10 Must allow Project leader and Administrator who can set or change deadlines.

Use Case:

2.11 Must allow Administrator to be the only person who can create or remove projects, and create or remove users.

Use Case: UC16, UC17, UC19, UC20,

3 Calendar

3.1 All important deadlines must be able to be seen in a user's calendar

Use Case: UC28

3.2 Daily, monthly and total project timelines of the project may be viewed in a user's calendar

Use Case: UC28

4 Forum

4.1 Allow user to post message on forum.

Use Case: UC34

4.2 Allow user to reply to posts on forum.

Use Case: UC34

4.3 Allow user to edit own posts.

Use Case: UC30

4.4 Administrator can delete posts.

Use Case: UC18

4.5 Allow user to create new thread.

Use Case: UC29

4.6 Administrator or Project Leader can create announcements.

Use Case: UC14

4.7 Recent announcements displayed on announcements page.

Use Case: UC27

4.8 Announcements older than 14 days are in announcements archive.

Use Case:

4.9 Users can view announcements on announcement page.

Use Case: UC25, UC27

4.10 Users can view archived announcements.

Use Case: UC26

5 Contact List

5.1 Each project has a contact list.

Use Case:

5.2 Contact lists relevant contact information for members of project.

Use Case:

5.3 Users can email people on contact list.

Use Case: UC32

5.4 Users may email everyone on contact list.

Use Case: UC31

2.2 Use Cases

File Use Cases

UC1: Selecting a file to edit

Primary Actor: Administrator, user or project leader

Goal: To edit the selected file.

Stakeholder Lists: All members of the project that the file belongs to.

Initiating Event: A user's need to edit a particular file.

Main Success Scenario: The file is edited, saved and time stamped. The old version is archived.

Alternate Flows or Exceptions:

1. The file is being edited by another user.

Use Cases Utilized: None.

UC2: Selecting a file to upload.

Primary Actor: Administrator, user or project leader

Goal: To upload the selected file to the system.

Stakeholder Lists: All members of the project that the file belongs to.

Initiating Event: User has completed editing a file and now wants to upload it.

Main Success Scenario: The file is uploaded (and is time stamped) onto the system and any old versions are archived.

Alternate Flows or Exceptions:

1. The file is currently under lockdown.

Use Cases Utilized: None

UC3: Selecting a file to download.

Primary Actor: Administrator, user or project leader

Goal: To download the selected file to the user's computer.

Stakeholder Lists: The current user.

Initiating Event: The user wishes to download a particular file.

Main Success Scenario: The file is downloaded to the location on the user's computer specified by the user.

Alternate Flows or Exceptions: None

Use Cases Utilized: None

UC4: Saving a file

Primary Actor: Administrator, user or project leader

Goal: To replace the current version of the file with the one being saved and archive the old version.

Stakeholder Lists: All members of the project.

Initiating Event: The user has finished editing a file and now wants to save it.

Main Success Scenario: The file is saved and time stamped. The old version of the file is archived.

Alternate Flows or Exceptions: None

Use Cases Utilized: None

UC5: Selecting an archived file.

Primary Actor: Administrator, user or project leader

Goal: To view the selected file.

Stakeholder Lists: The user that is selecting the file.

Initiating Event: The user wants to view an archived file.

Main Success Scenario: The archived file is opened so that the user can view it.

Alternate Flows or Exceptions: None

Use Cases Utilized: None

UC6: Opening a file in the editor.

Primary Actor: Administrator, user or project leader

Goal: To open the specified file.

Stakeholder Lists: The user that is choosing to open a file.

Initiating Event: The user wants to open a file for editing.

Main Success Scenario: The file is opened in an rtf editor.

Alternate Flows or Exceptions:

1. The file is being edited by another user.

Use Cases Utilized: None

UC7: Creating a new file.

Primary Actor: Administrator, user or project leader

Goal: To create and save a file.

Stakeholder Lists: All the members of the project.

Initiating Event: The need for a new file in a project.

Main Success Scenario: The user creates a new file in the editor and saves it. The file is time stamped upon saving.

Alternate Flows or Exceptions:

1. A file with the same name already exists in the project.

Use Cases Utilized: UC4

UC8: Selecting a Requirements template.

Primary Actor: Administrator, user or project leader

Goal: To insert a Requirement template into a file.

Stakeholder Lists: All members of the current project.

Initiating Event: The user wants to add a template to the file that they are working on.

Main Success Scenario: The template is inserted into the file that the user is working on.

Alternate Flows or Exceptions: None

Use Cases Utilized: None

UC9: Selecting a Use Case template.

Primary Actor: Administrator, user or project leader

Goal: To insert a Use Case template into a file.

Stakeholder Lists: All members of the current project.

Initiating Event: The user wants to add a template to the file that they are working on.

Main Success Scenario: The template is inserted into the file that the user is working on.

Alternate Flows or Exceptions: None

Use Cases Utilized: None

UC10: Moving a file from one directory to another.

Primary Actor: Administrator or Project Leader

Goal: To change the location of a file.

Stakeholder Lists: All the members of the project.

Initiating Event: The user wants to change the location of a particular file.

Main Success Scenario: The user deletes the copy of the file in the old directory.

Alternate Flows or Exceptions:

- 1) File with the same name exists in the directory that it is being moved to.

Use Cases Utilized: None

Project Lead and Administrator Use Cases

UC11: Creating a Directory

Primary Actor: Project Leader or Administrator

Goal: To create a new directory for storing files.

Stakeholder Lists: All members of the project.

Initiating Event: The user wants to create a new directory.

Main Success Scenario: The directory with the specified name is created and displayed in the directory structure.

Alternate Flows or Exceptions:

1. The name chosen for the directory already exists.

Use Cases Utilized:

UC12: Deleting a Directory

Primary Actor: Project Leader or Administrator

Goal: To delete an existing directory

Stakeholder Lists: All members of the project.

Initiating Event: The user wants to delete a directory.

Main Success Scenario: The directory is deleted and removed from the directory structure.

Alternate Flows or Exceptions: None

Use Cases Utilized: None

UC13: Renaming a Directory

Primary Actor: Project Leader or Administrator

Goal: To rename an existing directory

Stakeholder Lists: All members of the project.

Initiating Event: The user wants to change the name of a directory.

Main Success Scenario: The name of the directory is changed and its new name appears in the directory structure.

Alternate Flows or Exceptions:

1. The name already exists.

Use Cases Utilized: None

UC14: Adding an announcement

Primary Actor: Administrator or project leader

Goal: To add a new announcement to a project's announcement section

Stakeholder Lists: All members of the project.

Initiating Event: The project leader of a project needs an announcement added to the project

Main Success Scenario: The new announcement gets successfully added to the project

Alternate Flows or Exceptions:

1. The system goes down

Use Cases Utilized: None

UC15: Adding an event or due date

Primary Actor: Administrator or project leader

Goal: To add a new event or due date to the calendar of a project

Stakeholder Lists: All members of the project

Initiating Event: The project leader of a project needs an event or due date added to the calendar of a project

Main Success Scenario: The new event or due date gets successfully added to the project

Alternate Flows or Exceptions:

- 1 The project does not exist
 - a. The administrator must add the project to the system

Use Cases Utilized: None

UC16: Adding a project

Primary Actor: Administrator

Goal: To add a new project to the system.

Stakeholder Lists: Administrator

Initiating Event: The administrator receives a new project request.

Main Success Scenario: The project is successfully added to the system with all the requested access rights.

Alternate Flows or Exceptions:

1. The project already exists
2. the project leader does not have an account in the system
 - a. the administrator adds the project leader to the system

Use Cases Utilized: None

UC17: Adding a user

Primary Actor: Administrator

Goal: To add a new user to the system.

Stakeholder Lists: Administrator, user to be added

Initiating Event: The administrator receives a new user request from a project leader.

Main Success Scenario: The user is successfully added to the system with all the the requested access rights.

Alternate Flows or Exceptions:

1. The user already exists

Use Cases Utilized: None

UC18: Delete posts

Primary Actor: Administrator or project leader

Goal: To delete posts from the forum

Stakeholder Lists: User and project leader

Initiating Event: Administrator or project leader chooses a post from the forum to delete

Main Success Scenario: The post gets successfully deleted from the forum

Alternate Flows or Exceptions:

1. The system goes down

Use Cases Utilized: None

UC19: Deleting a project

Primary Actor: Administrator

Goal: To delete a finished/old project from the system.

Stakeholder Lists: Administrator

Initiating Event: the administrator receives a project removal request.

Main Success Scenario: The project is successfully removed from the system

Alternate Flows or Exceptions:

1. The project does not exist in the system

Use Cases Utilized: None

UC20: Deleting a user

Primary Actor: Administrator

Goal: To delete a current user from the system.

Stakeholder Lists: Administrator

Initiating Event: The administrator receives a user removal request.

Main Success Scenario: The user is successfully removed from the system

Alternate Flows or Exceptions:

1. The user does not exist in the system

Use Cases Utilized: None

UC21: Assigning a project leader

Primary Actor: Administrator

Goal: To change the project leader of a project

Stakeholder Lists: Administrator, project leader

Initiating Event: The project leader of a project needs to be changed/assigned

Main Success Scenario: The a new project leader is successfully assigned to the project(s)

Alternate Flows or Exceptions:

1. The user does not exist in the system
 - a. The administrator must add the user to the system
2. The user already is the project leader
3. The project does not exist
 - a. The administrator must add the project to the system

Use Cases Utilized: None

UC22: Modifying user's access rights

Primary Actor: Administrator or project leader

Goal: To change the access rights a user has to projects

Stakeholder Lists: Administrator, project leader, the user

Initiating Event: A user needs their access rights changed.

Main Success Scenario: The user successfully has their access rights changed

Alternate Flows or Exceptions:

1. The user does not exist in the system
 - a. The administrator must add the user to the system
2. The project does not exist
 - a. The administrator must add the project to the system

UC23: Unlocking a file

Primary Actor: Administrator, Project leader

Goal: To unlock a locked file in a project

Stakeholder Lists: All members of the project

Initiating Event: The project leader or administrator receives an unlock file request from a user

Main Success Scenario: The file is successfully unlocked

Alternate Flows or Exceptions:

1. The file is corrupt
 - a. The administrator restores the most recent version of the file
2. The file is already unlocked

Use Cases Utilized: None

UC24: Locking a file

Primary Actor: Administrator, Project leader

Goal: To lock file in a project

Stakeholder Lists: All members of the project

Initiating Event: The project leader or administrator wants to lock a file.

Main Success Scenario: The file is successfully locked

Alternate Flows or Exceptions: None

Use Cases Utilized: None

Miscellaneous

UC25: Selecting a Project

Primary Actor: Users, Project leaders, and Administrators

Goal: To login to a specific project.

Stakeholder Lists: Users, Project leaders, and Administrators

Initiating Event: Logging in

Main Success Scenario: The announcement page for the selected project is displayed.

Alternate Flows or Exceptions:

1. System is unable to load project.
2. System unable to display announcement page

Use Cases Utilized: None

UC26: Looking at Archived Announcements

Primary Actor: All Users, Project leaders, and Administrators

Goal: To access all the older announcements.

Stakeholder Lists: Users, Project leaders, and Administrators

Initiating Event: Viewing the announcements page

Main Success Scenario: The archived announcements are displayed in a new page.

Alternate Flows or Exceptions: None

Use Cases Utilized: None

UC27: Selecting Menu Options

Primary Actor: Users, Project leaders, and Administrators

Goal: To allow users to easily navigate between all parts of the project from any part of the project.

Stakeholder Lists: Users, Project leaders, and Administrators

Initiating Event: Selecting a project

Main Success Scenario: Displays selected project page

Alternate Flows or Exceptions: None

Use Cases Utilized: None

UC28: Selecting a Event from the Calendar

Primary Actor: Users, Project leaders, and Administrators

Goal: Allow users to get detailed information about events and due dates.

Stakeholder Lists: Users, Project leaders, and Administrators

Initiating Event: Viewing the project calendar

Main Success Scenario: Displays selected event info

Alternate Flows or Exceptions: None

Use Cases Utilized: None

UC29: Create a new thread

Primary Actor: Administrator, User or Project leader

Goal: To add a new thread to the thread list

Stakeholder Lists: All members of the project

Initiating Event: Administrator, user or project leader chooses create a new thread from the forum's options

Main Success Scenario: The new thread gets successfully added to the thread list

Alternate Flows or Exceptions:

1. The system goes down

Use Cases Utilized: None

UC30: Edit post

Primary Actor: Administrator, User, Project leader

Goal: To be able to edit a post in the forum

Stakeholder Lists: Administrator, User, Project leader

Initiating Event: An administrator, user or project lead chooses a post to edit

Main Success Scenario: The administrator, user or project lead makes the appropriate changes to the post

Alternate Flows or Exceptions:

1. The system goes down

Use Cases Utilized: None

UC31: Email all users

Primary Actor: Administrator, user or project leader

Goal: To send an email to all users

Stakeholder Lists: Administrator, user or project leader

Initiating Event: Administrator, user or project leader chooses select all from the options in the user list

Main Success Scenario: The email gets sent to the all users

Alternate Flows or Exceptions:

1. The system goes down

Use Cases Utilized: None

UC32: Email selected users

Primary Actor: Administrator, user or project leader

Goal: To send an email to selected users

Stakeholder Lists: Administrator, user or project leader

Initiating Event: User chooses users from a list to email

Main Success Scenario: The email gets sent to the selected users

Alternate Flows or Exceptions:

1. The system goes down

Use Cases Utilized: None

UC33: Logging in

Primary Actor: The user trying to login

Goal: To gain access to the system

Stakeholder Lists: The user trying to login

Initiating Event: The user trying to login goes to the systems login page

Main Success Scenario: The user is successfully logged into the system

Alternate Flows or Exceptions:

1. The system is unavailable
 - a. an error message will be displayed
2. The user does not exist
 - a. an error message will be displayed
3. Invalid password provided
 - a. an error message will be displayed

Use Cases Utilized: None

UC34: Reply to a previously written message

Primary Actor: Administrator, user or project leader

Goal: Reply to a previously written message within a thread

Stakeholder Lists: Administrator, user or project leader

Initiating Event: Administrator, user or project leader chooses to reply to a message

Main Success Scenario: The new message gets successfully added to the thread

Alternate Flows or Exceptions:

1. The system goes down

Use Cases Utilized: None

UC35: Select a thread

Primary Actor: Administrator, User, Project leader

Goal: To be able to view a forum thread and the messages contained therein

Stakeholder Lists: Administrator, User, Project leader

Initiating Event: An administrator, user or project leader chooses a forum thread from the thread list

Main Success Scenario: The administrator, user and project leader can view the threads messages

Alternate Flows or Exceptions:

1. The system goes down

Use Cases Utilized: None

UC36: Logging out

Primary Actor: The user trying to logout

Goal: To close out of the system

Stakeholder Lists: The user trying to logout

Initiating Event: The user trying to logout presses the logout button

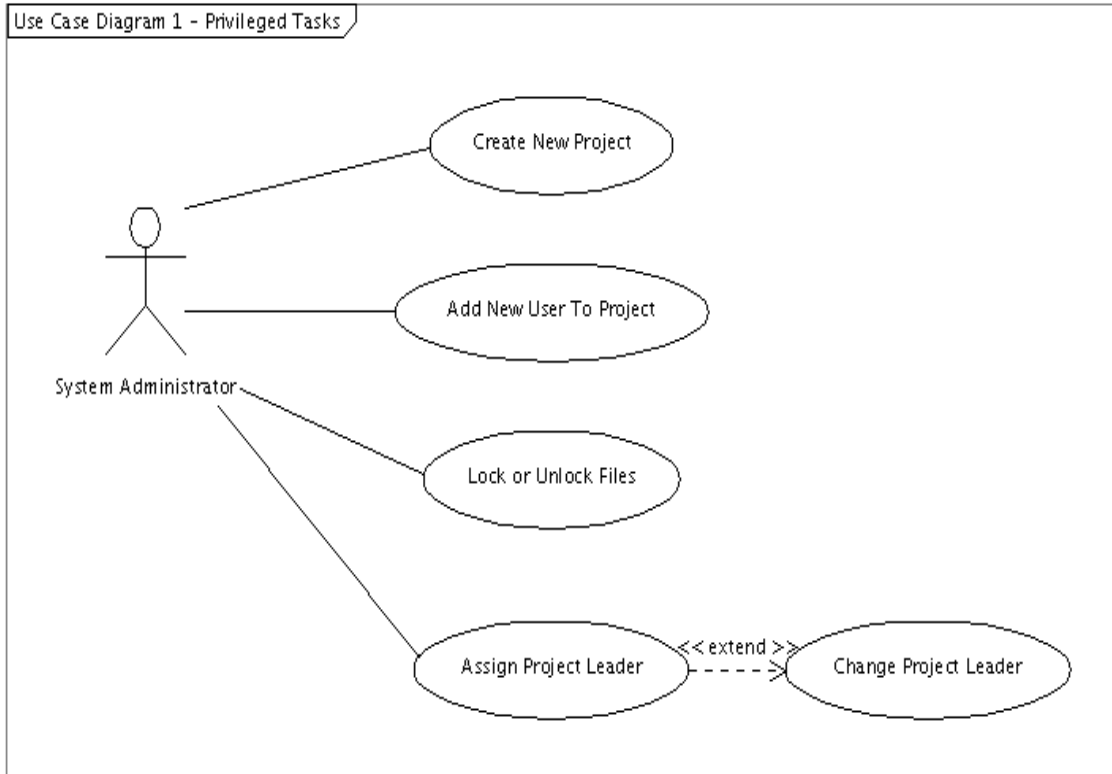
Main Success Scenario: The user is successfully logged out of the system

Alternate Flows or Exceptions:

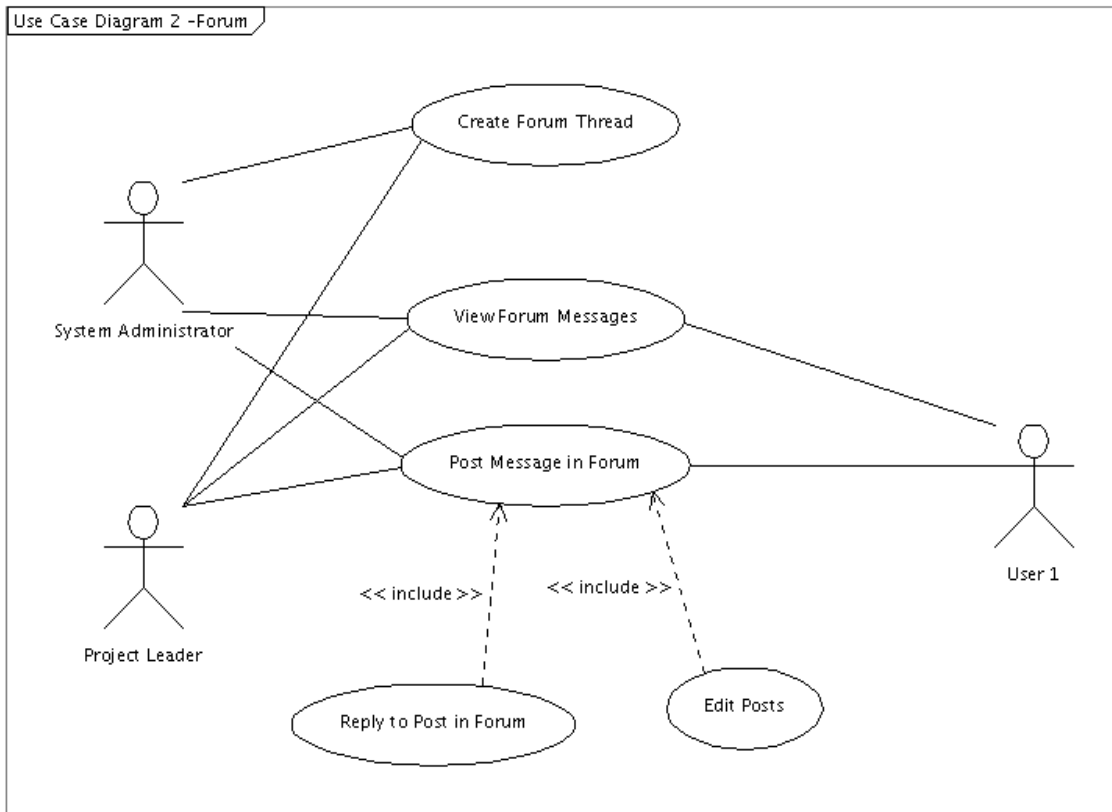
Use Cases Utilized: None

2.3 Use Case Diagrams

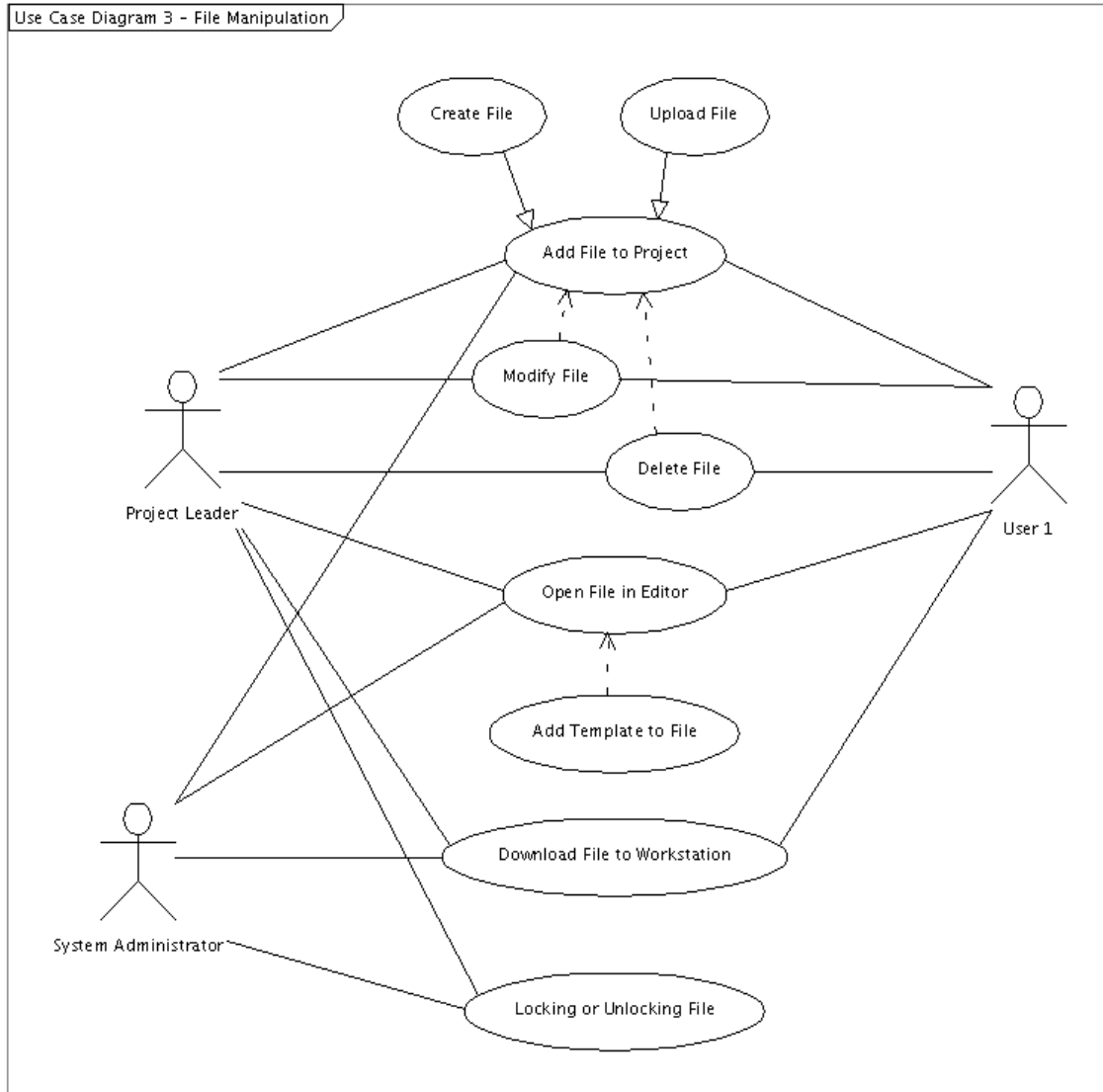
This diagram illustrates the tasks that are assigned to the administrator but none of the workers.



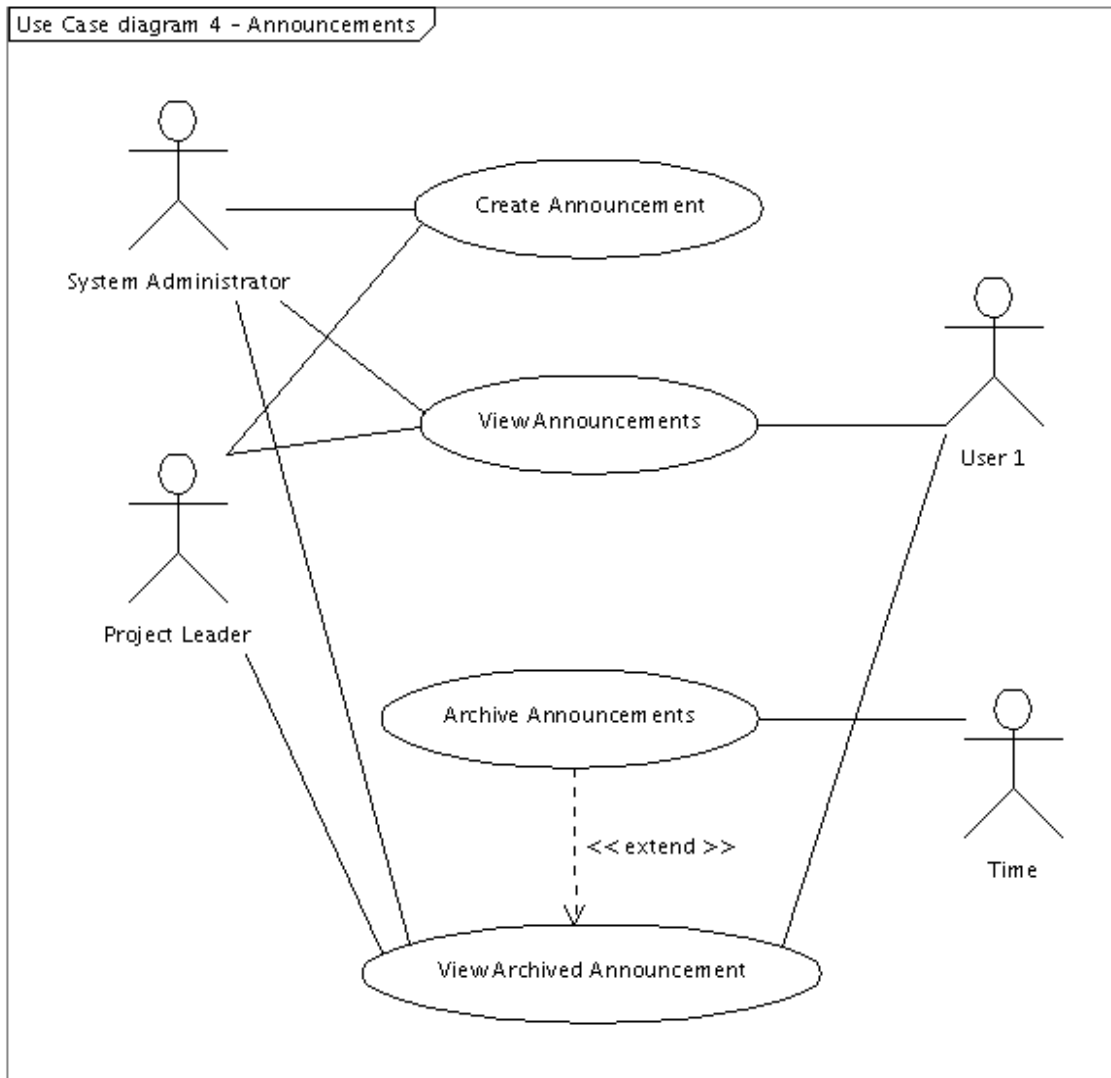
This diagram illustrates the interaction of all types of users with the forum.



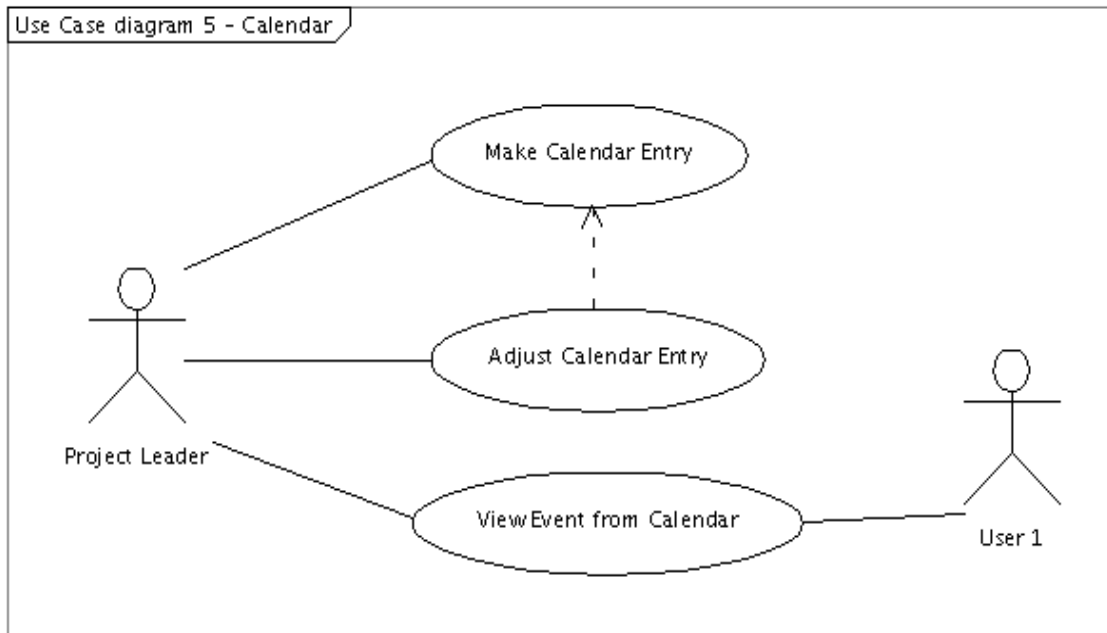
This diagram illustrates file manipulation that can be performed by the three types of users.



This diagram illustrates actions of the three types of users as related to announcements.



This diagram illustrates user interaction with the calendar.



3. Non-Functional Requirements

3.1 Prototype

Our prototype is a fully dynamic web page. It is accessible at the following web address.

http://www.cs-club.org/~chigau/Login/login_main.htm

There is no need for extensive documentation for our prototype it is self documenting since it is a fully dynamic webpage.

The following is a basic walkthrough outlining the basic functionality shown through the prototype.

Login Page

- Enter username into username text field
- Enter password into password text field
- Click on login button

Project Selection Page

- Click on corresponding picture for desired project

Announcements Page

- Click on archive link
- Click on back button
- Choose file menu item

Files

- Choose drop down menu of a file and pick a archived version
- Click the edit button

Workspace

- Edit the document in the text area
- Click on the save button
- Click on the Requirements template button
- Edit the document in the text area
- Click on the save button
- Click on calendar menu button

Calendar

- Click on the 15th
- Click on the back button
- Click on forum menu button

Forum

- Click on the first post
- Click on the reply button
- Type in text area
- Click on reply button
- Click on Contacts menu button

Contacts

- Click on email all button
- Click on logout button at bottom right hand side of screen

3.2 Look and Feel

3.2a. The Interface

- The product will have the company logo in the top left hand corner of every part of project.
- The product will have the product name directly underneath the company logo.
- The product will use simple interactive buttons to give users feedback.
- The product will have a neat, tidy, and clean interface free of clutter and easy to use.
- The navigation system of the product shall be very straightforward the menu and logout features will always be accessible from every part of the project.
- The features associated with the administrator and project lead will be located throughout the system where those feature would be expected.

3.2b. The Style of the Product

- The product will have a professional appearance.
- The product will appear to be simple to use while still appearing effective.
- The product will be visually appealing to all; by tasteful use of graphics without cluttering the interface.

4. Project Issues

4.1 Risks

1. Spending too much time learning instead of using technology like PHP
2. Not having enough time to work on project due to other classes and commitments.
3. Using libraries or source code that is too complex or hard to adapt.
4. Spend too much time discussing non-relevant topics
5. Trying to implement and design too many features and too much functionality in the time we have to complete the project.
6. Not coming to a consensus on certain issues (ie features that should be implemented)
7. Designating inappropriate positions to each group member
8. A team member not completing assigned work on time.
9. Underestimation of work required to complete a task
10. Poor prioritization

5. Major Revision History

Version	Date	Description	Author
0.1	February 1 st 2006	-High level outline of the requirements -This was the rough drawings of what we thought the prototype would look like.	UberSoft
0.15	February 8 th 2006	-Built basic atomic requirements. -The basic high fidelity prototype was built and included all main pages without content, menu buttons, and logout. Button	UberSoft
0.2	February 9 th 2006	-Created basic use cases from requirements. -The announcement page content, file page content, and calendar page content were added to the prototype.	UberSoft
0.25	February 11 th 2006	-Updated requirements based on knowledge gained form prototype and use cases. -The workspace page content, the forum page content, and the contact page content were all added.	UberSoft
0.3	February 13 th 2006	-Refined use cases based on new requirements. -Revised a few pages based on requirements and use cases.	UberSoft
0.4	February 15 th 2006	-Compiled requirements and use cases into a SRS document. -Included executive summary, use case diagram, and glossary etc.	UberSoft
0.5	February 16 th 2006	-Refined SRS document by reviewing and editing fixing grammar, spelling, formatting issues -Made a few changes to format of requirements and use cases in SRS document	UberSoft

6. Market Assessment and Competitive Review

Before creating our content management system, we looked at other Content Management Systems (CMS) that had already been created to see where we could improve upon their systems. We designed a ranking system based upon 5 different criteria so we could see where we should focus our design efforts to improve on the CMS's currently in the market.

Our ranking system's criteria were: ease of use, flexibility, appearance, functionality and documentation. We judged ease of use exactly how it sounds; if we couldn't figure out how to use something easily or if some aspect of the CMS didn't make sense, then the ranking was affected appropriately. The flexibility was marked on how we could change and modify the environment to suit our needs. We judged the appearance on how appealing the CMS was on first glance, on how the menus were set up from one screen to the next and how easy it was to navigate through the site. The functionality aspect was ranked on how well the CMS performed. Finally, the documentation was judged on whether or not we could find help documents and tutorials to walk us through the CMS.

Each CMS was ranked by a set of stars for each criteria. If a CMS was given one star, it performed poorly and had little to recommend it, 2 stars meant that the CMS needs much improvement in that area, 3 stars meant that it was acceptable but could still use more work, 4 stars was well done and is something we'd show to our friends, and 5 stars means that the CMS had everything we were expecting. We looked at 4 different CMS' and we tried to find 4 that had different qualities associated with them. The CMS' that we looked at were: WebCollab, MyACollab, PhpGroupWare and NetOffice.

6.1 Reviews

WebCollab

Website: <http://webcollab.sourceforge.net/>

Current Version: 1.81

Language: PHP/MySQL

Cost: Free






From the WebCollab Developers:

A collaborative web-based system for projects and project management; WebCollab is easy to use, and encourages users to work together. The software is functionally elegant and secure without being cumbersome for users, or graphically intensive.

The software is ideally suited to tracking multiple projects and innumerable small tasks across an organization of any size.

Why we are reviewing it:

We have found many positive WebCollab reviews on the Internet and determined it to be a very popular CMS for project management. By reviewing it we hope to find problems in it which we will be able to have solved in our own CMS.

<i>Ease Of Use</i>	<i>Flexibility</i>	<i>Appearance</i>	<i>Functionality</i>	<i>Documentation</i>
 The CMS would be a lot easier to use if it included a explanation for some of the things needed in order to set up projects. It leaves the user in the dark and makes them guess how certain things work.	 WebCollab has very little that it allows the user to customize. It doesn't allow for enabling and disabling of unwanted features, changing permissions, etc.	 The interface is very pleasant on the eyes and it is very easy to make your way through it. The navigation is to the point and allows the user to quickly find what he is looking for.	 Most of the features included are very good. It includes multiple projects, task listing, calendars, to-do lists but seems to be missing some functionality in the area of user groups and user permissions.	 The documentation available is very limited. The website includes basic troubleshooting techniques but it doesn't explain how to use the CMS at all.

netOffice

Website: <http://netoffice.sourceforge.net>

Current Version: 2.60b2

Language: PHP/MySQL






Cost: Free

From the netOffice Developers:

netOffice is a free web based project-management environment written in php/mysql. netOffice allows managing and sharing information about teams, projects, tasks, deadlines and much more. It's intuitive interface makes it a fast and optimal solution for medium size companies.

Why we are reviewing it:

netOffice presents itself with a very professional image and by the way some of the things are written on the website it believes it's one of the best in the industry. We feel that if they are that confident about their CMS that we should include it in the review and find out what makes it so good.

<i>Ease Of Use</i>	<i>Flexibility</i>	<i>Appearance</i>	<i>Functionality</i>	<i>Documentation</i>
 A lot of areas in the CMS need more explanation on how they work. It has some advanced configurations which a new user wouldn't even be able to figure out without being taught how to use them.	 netOffice allows you to totally customize the main page you see after you login. You can open and close various tabs to have different data displayed. This is a great feature to have.	 While the overall interface of the software is good, the navigation could be improved upon a lot. Some of the pages are organized in an overall bad way.	 While the overall interface of the software is good, the navigation could be improved upon a lot. Some of the pages are organized in an overall bad way.	 Little to no documentation is available for users; however, netOffice seems to have a message forum which is very active and could help solve problems.

MyACollab

Website: <http://www.atutor.ca/acollab/>

Current Version: 1.2

Language: PHP/MySQL

Cost: Free

From the MyACollab Developers:

ACollab is a fully accessible, open source, multi-group, Web-based collaborative work environment. ACollab is available as a work groups addon for ATutor 1.4+ , and as a standalone collaborative work environment that will run on its own. ACollab is ideal for groups working at a distance developing documentation, collaborating on research, or writing joint papers, and ideal for online educators who wish to add group learning activities to their ATutor courses.

Why we are reviewing it:

We wanted to be able to compare a CMS that had a very good appearance and ease of use to be able to see how that influenced our opinions of it. By reviewing MyACollab we are hoping to be able to know how much time we should allocate in designing the website.

<i>Ease Of Use</i>	<i>Flexibility</i>	<i>Appearance</i>	<i>Functionality</i>	<i>Documentation</i>
☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆
This CMS is very easy to use. Its boundaries are clear and you always know where you are. The only problem with it is that it is a little difficult to figure out how to create and upload files	Admin can set permissions to each group, however there is very little flexibility in the overall use of the CMS.	MyACollab has a very dynamic appearance that is very appealing and easy to understand. It is not cluttered at all and yet conveys a sophistication that allows easy use.	This CMS seemed to do everything that it set out to do in a clear and precise manner. Admin could create and remove groups, and set permissions as well.	Many types of documentation available online. The documentation includes very helpful tutorials on all basic operations.

PhpGroupWare

Website: <http://www.phpgroupware.org>

Current Version: 1.2

Language: PHP/MySQL






Cost: Free

From the PhpGroupWare Developers:

phpGroupWare - formerly known as webdistro - is a multi-user groupware suite written in PHP. It provides about 50 web-based applications, as there are the Calendar, Addressbook, an advanced Projects manager, Todo List, Notes, Email, Newsgroup- and Headlines Reader, a Filemanager and many more Applications. The calendar supports repeating events and includes alarm functions. The email system supports inline graphics and file attachments.

Why we are reviewing it:

We wanted to review a CMS that was close in functionality to what we had in mind for our own CMS. According to the phpGroupWare website, this CMS seemed to do everything that we had in mind plus a couple extra frills.

<i>Ease Of Use</i>	<i>Flexibility</i>	<i>Appearance</i>	<i>Functionality</i>	<i>Documentation</i>
 This CMS was not very easy to use. There was no clear way to add users or create new groups. Also, the menu system was not easy to use and didn't stay consistent throughout the website.	 Every aspect of this CMS has its own preference section, i.e you could change the preferences for e-mail, calendar etc. Most of the preferences, however, had to do with font changes or text based changes. So overall, the system was not very flexible.	 The colours used in this CMS are pleasing, however the icons used to represent the menu are confusing and not clear. There is no text by the icons to explain what they are, you have to put the mouse over the icon to figure out what it means.	 It was confusing trying to get the functions to work properly. To get anything to work, you were given a list of options in an unattractive format that made it difficult to figure how things worked.	 There is no documentation available for this CMS. If you have problems, there is a forum that you can write to.

6.2 Conclusion

This market assessment has allowed UberSoft to determine exactly how to build UberCMS so that it does not contain the flaws of the currently distributed content management systems. All of the CMS' analyzed were feature packed but contain interfaces that are designed in a way so that they are very hard to use and are really just visually unpleasant to work with. We plan to make an interface that is clean and straight to the point, gives hints and tips on how to work with the different features and to make it as easy as possible for the user. Having good documentation is critical with any piece of software and was found to be consistently lacking in the CMS' that we reviewed. Without documentation it can be extremely difficult and unpleasant for users to navigate through the software simply because not all the users will have the same knowledge level in computers.

In order to make it the leading content management system we here at UberSoft are taking extra measures to ensure that all the aforementioned issues are not present in the initial release of UberCMS.

6.3 Works Cited

WebCollab [Internet]

Available from <<http://webcollab.sourceforge.net/>>

NetOffice [Internet]

Available from <<http://netoffice.sourceforge.net>>

MyACollab [Internet]

Available from <<http://www.atutor.ca/acollab/>>

phpGroupWare [Internet]

Available from <<http://www.phpgroupware.org>>

7. Glossary

Administrator – the person in charge of the system control and maintenance with no access restrictions

Announcement – important information posted (by the project lead or administrator) on the main page of a project

Announcement Archive – location where all of the announcements for a given project older than 14 days are stored

Calendar – list of the important events or due dates for a given project

CMS – Content Management System

Download a file – save a copy of a file from a given project to your own computer for reference.

Editing a file - save a copy of a file from a given project to you own computer so you can edit it. The copy of the file on the server gets locked.

File Archive – location where all of the old versions of files for a project are stored until the project is completed

Locked file – a file which is being edited by someone and cannot be accessed by anyone else until the person editing it uploads it back to the system

Project leader – the person in control of each project. Has similar access to the administrator but can't create/delete users or projects, and can only access the projects the are leading.

Unlock a file – setting the file back to a state that all users can access it.

Upload a file – transfer a file from your computer to the system. If the file was locked, the old copy of the file is unlocked and placed in the file archive, and the most current version of the file is updated to be the file you just uploaded to the system.

User/Worker – the front line workers of a given project. Can only access items contained in the projects they are given permission to access, can't delete anything and can only edit their own forum posts.