Software Requirements Specification (SRS) 2005

Version 1.0
April 18, 2005

Produced by:
Team Homer
(s340gh)

Rohin Aggarwal, Andrew Homer, Tushar Kuchhal
Yi-Jie Lee, Sreedhar Valicherla

Abstract

This document describes the software requirements of the Student Project Activity Module for Moodle. It provides a description of all functional and non-functional requirements for the system. It also acts as a contract between the client and team H for what will entail a high-quality end product for the system.

Contents

List of Figures

List of Tables

1 Introduction

The Introduction contains the purpose of the Software Requirements Specification (SRS) in Section 1.1 and its connection with the entire software production process, as well as how this document will change throughout this process as part of the evolution of this document in Section 1.2.

The scope of this document is mentioned in Section 1.3, and details of the personnel involved with this project are contained in Section 1.4.

Any acronyms, abbreviations or definitions used throughout this document are contained in sections 1.5 and 1.6 respectively, and the references we utilised in the construction of this document are included in Section 1.7.

Finally, a complete overview of the remainder of this document can be found in Section 1.8.

1.1 Purpose

This document is intended to present the requirements of the system to be produced, both functional and non-functional. The specifications contained in this document will be used to support the production of the system in later stages, in an attempt to reduce the development effort involved.

The SRS explains the current system being used by the customer, the reason for the creation of the new system and a broad overview of the proposed system to be developed.

It also serves as a contract between the customer and the supplier to approve the requirements contained here, as specified in the acceptance criteria.

The audience of this document (Software Requirements Specification) primarily includes the Project Team, the Client and the Project Supervisor.

1.2 Evolution of this Document

Our Team will endeavour to elicit all of the requirements for the proposed system in the early stages of the project life-cycle. However, it may be impracticable for some requirements to be specified during the initial stages.

As such, this document has been produced as thoroughly as is possible at this time. Any changes deemed necessary by the client in conjunction with our Team will be formally documented by revising this document in a traceable manner.

The procedure to be followed in the event of a requirements change is outlined in Section 6 of this document.

In order to prevent alterations to the requirements at a later phase in the production process, our Team will provide the client with a prototype of the system, showing all interfaces, and allowing the client to suggest any changes.

1.3 Personnel

The personnel involved with this project are outlined in this section. This includes the Clients, the Project Team and the Project Supervisor.

1.3.1 Clients

The Clients for this project are:

Don Hinkelman, PhD(Student)
Horwood Language Centre, School of Languages,
University of Melbourne VIC 3010 Australia
Email: hinkel AT sgu DOT ac DOT jp
Phone: 8344-3481

Paul Gruba, PhD
Deputy Director
Horwood Language Centre, School of Languages
University of Melbourne VIC 3010 Australia
Phone: 8344-5147

1.3.2 Our Team

Our development Team consists of (the item in brackets is each person's login):

- **Rohin Aggarwal** (rohina)
- **Andrew Homer** (ajhomer)
- **Tushar Kuchhal** (tkuchhal)
- **Yi-Jie Lee** (leeyj)
- **Sreedhar Valicherla** (sreedhar)

The email address of each team member is created by appending the following each login:
AT students DOT cs DOT mu DOT oz DOT au

1.3.3 Supervisor

The supervisor of our team is:

Joseph Lee (josephml)
Department of Computer Science and Software Engineering
The University of Melbourne
Email: josephml AT cs DOT mu DOT oz DOT au

1.4 Acronyms & Abbreviations

- **CLO**: Client Liaison Officer
- **CMS**: Course Management System
- **SRS**: Software Requirements Specification

1.5 Definitions

- **Course Management System:** A software package designed to assist teachers with the creation of online courses for their students.
- **Customer:**The client of the system.
- **Module:** Each separate component developed within Moodle.
- **Moodle:** An open-source course management system used by teachers worldwide.
- **Our Contract:** The document used to verify that the SRS requirements meet the clients' needs and expectations.
- **project:** The name used for assignments teachers give to their students (Contrast with 'Project'). This term is also used in this document to refer to the system that we are developing.
- **'Project':** The name of the sub-module we are developing within Moodle.
- **Supplier:** The development team of this project.
- **Task:** A function to be performed by students. Tasks are created by the teacher, and comprise one of the following sub-modules: Brainstorm, Signup, Schedule, Submit or Assessment.
- **Topic:** The subject of the project selected by students.
- **User:** Those people who will be using the 'Project' module. This may be, but is not limited to, our client.

1.6 References

- IEEE-SA Standards Board, IEEE Recommended Practice for Software Requirements Specifications
- Gavens, S. et. al., Software Requirements Specification, v1.0, University of Melbourne, April 17, 2004
- 433-340 Software Requirements Specification Template, Revision 1.0, March 27, 2005

1.7 Overview

The remainder of this document is structured in the following way:

- Section 2: Project Overview - Contains details of the current system, why the system is to be modified, and the proposed system to be developed.
- Section 3: Functional Requirements - All of the functional requirements of the proposed system are detailed here.
- Section 4: Non-functional Requirements - Details the non-functional targets of our system.
- Section 5: Use-cases - Use case diagrams are presented to identify how the system will be used.
- Section 6: SRS Modification Process - This section details the process involved with making any modifications to the SRS.
- Section 7: Acceptability Criteria - Detailed list of criteria is presented here for the client to assess.
- Section 8: Client Sign-Off - The contract between the development Team and the Client is contained here.

2 Project Overview

2.1 Existing System(s)

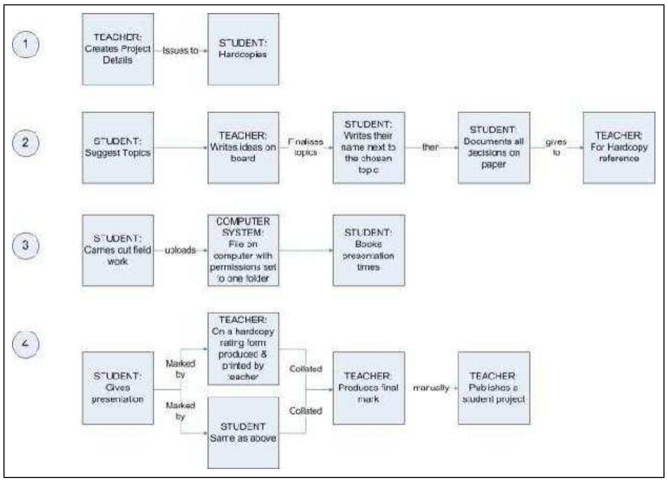

1
TEACHER: Creates Project Details
Issues to
STUDENT: Hardcopies
2
STUDENT: Suggest Topics
TEACHER: Writes ideas on board
Finalises topics
STUDENT: Writes their name next to the chosen topic
then
STUDENT: Documents all decisions on paper
gives to
TEACHER: For Hardcopy reference
3
STUDENT: Carries out field work
uploads
COMPUTER SYSTEM: File on computer with permissions set to one folder
STUDENT: Books presentation times
4
STUDENT: Gives presentation
Marked by
Marked by
TEACHER: On a hardcopy rating form produced & printed by teacher
STUDENT Same as above
Collated
Collated
TEACHER: Produces final mark
manually
TEACHER Publishes a student project

Figure 1: Existing System Being Used By Client

2.1.1 Context

2.2 Proposed System

Moodle is a course management system (CMS). The Moodle software was designed to help educators create online courses. Many online courses can be run through Moodle, but each course has access to the activity modules. These modules help to facilitate e-learning in the course by allowing educators to set an activity for the students.

The Moodle system has a variety of standard activity modules (e.g. Assignment, Journal, Quiz, etc.). Each module is a different activity and can be selected by the teacher at any time. The proposed system is an addition to the set of activity modules within the Moodle system (refer to Figure 2.2.1). Within the Project Module, there are eight tasks (Refer to Figure 2.2.3):

- Coordination
- Brainstorm
- Group Selection
- Topic Selection
- Submit

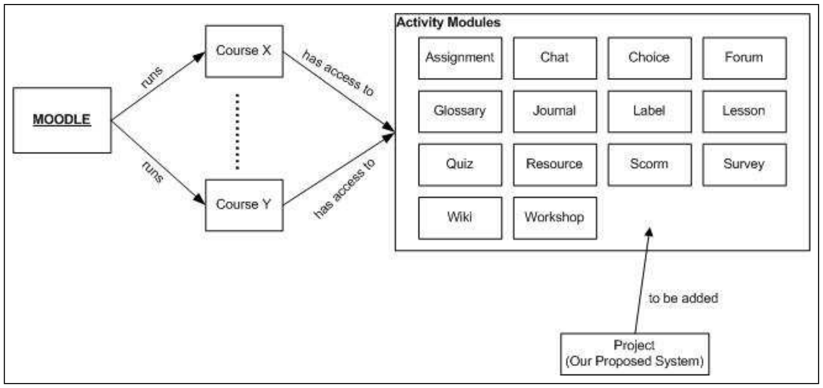


Figure 2: Model of the Moodle System

- Schedule
- Assessment
- Archive

Each task within the project has it's own interface and functionality. The functionality (refer to Section 3) will also depend on the user type (Teacher or Student).
When a Teacher adds Project as an activity, it will always go to the Coordination task first. From the Coordination task, the teacher can create the remaining tasks of the Project.

The general flow then involves the Teacher choosing the Brainstorm, Group Selection and Topic Selection tasks. Then either the Submit or Schedule tasks will be chosen, followed by the Assessment tasks. This process can be iterated starting from certain points as frequently as the Teacher desires (Refer to Figure 2.2.2). Finally, the Teacher can choose the Archive task.

The process is much the same if a Teacher decides to edit a Project. The Coordination task will appear first and then the list of tasks set. The Teacher can then add more tasks, remove tasks, or change the settings for tasks.

When a Student selects a Project given to them. They will also go to the Coordination task, but nothing that can be edited will be seen. Essentially, it will be an overview of the project and the timeline of tasks. The Student can then select available tasks, each view dependent on the type of task and also the settings chosen by the Teacher.

The Brainstorming tasks, allows ideas for topics to be publicised. Both the Teacher and/or Student can do this. Only the Teacher can remove topics.

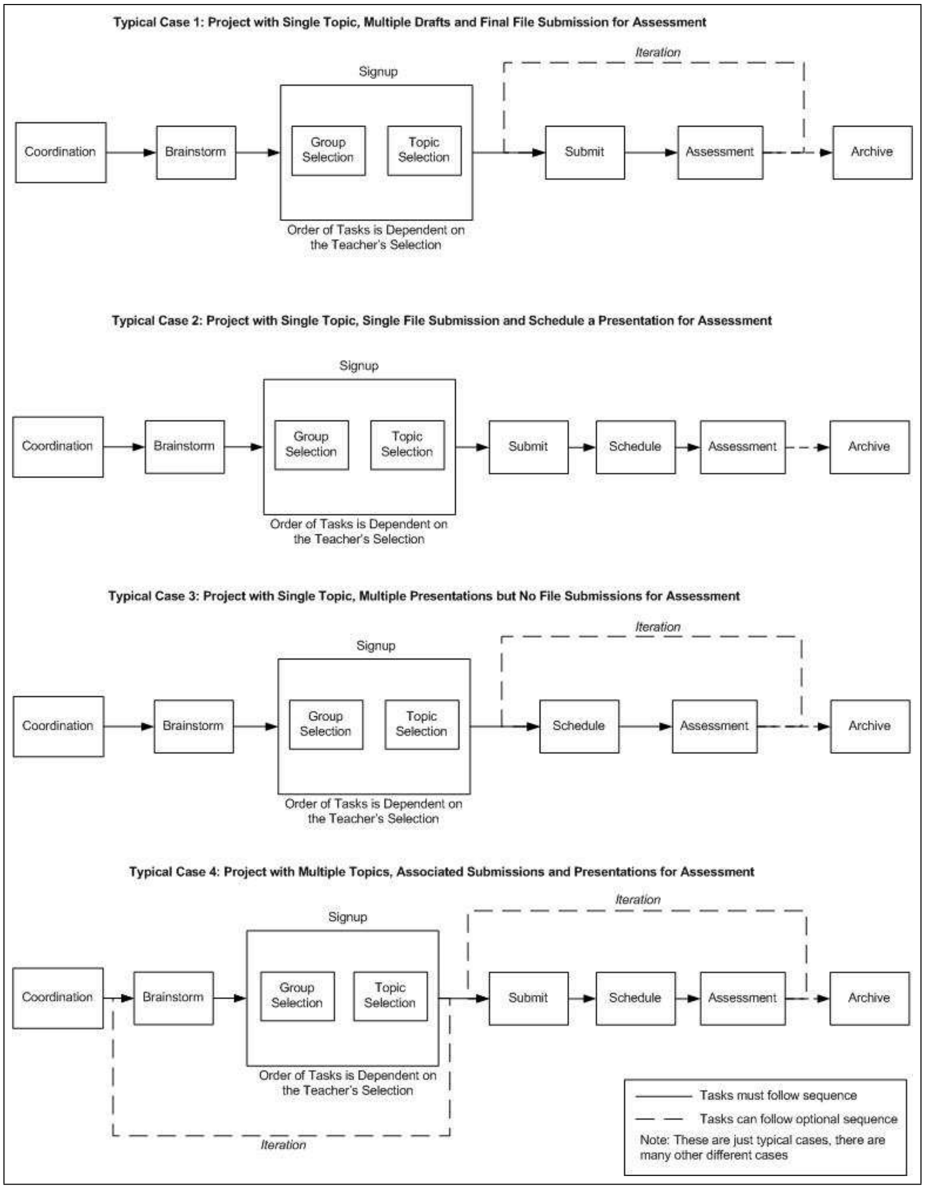

Typical Case 1: Project with Single Topic, Multiple Drafts and Final File Submission for Assessment
Iteration
Signup
Coordination
Brainstorm
Group Selection
Topic Selection
Submit
Assessment
Archive
Order of Tasks is Dependent on the Teacher's Selection
Typical Case 2: Project with Single Topic, Single File Submission and Schedule a Presentation for Assessment
Signup
Coordination
Brainstorm
Group Selection
Topic Selection
Submit
Schedule
Assessment
Archive
Order of Tasks is Dependent on the Teacher's Selection
Typical Case 3: Project with Single Topic, Multiple Presentations but No File Submissions for Assessment
Iteration
Signup
Coordination
Brainstorm
Group Selection
Topic Selection
Schedule
Assessment
Archive
Order of Tasks is Dependent on the Teacher's Selection
Typical Case 4: Project with Multiple Topics, Associated Submissions and Presentations for Assessment
Iteration
Signup
Coordination
Brainstorm
Group Selection
Topic Selection
Submit
Schedule
Assessment
Archive
Order of Tasks is Dependent on the Teacher's Selection
Iteration
Tasks must follow sequence
Tasks can follow optional sequence
Note: These are just typical cases, there are many other different cases

Figure 3: General Flow of Creating a Project

The Signup stage involves 2 different tasks. It includes group selection and topic selection. The Teacher has a few choices in which these 2 tasks are done.

Group selection can be done by Teachers, the Students, or by Students selecting the same Topic. The Teacher also is capable of removing individuals and groups.

Topic selection can be done by Teachers or by Students. If Student choose Topics, it is done by a voting system or by a first-come-first-serve system. The Teacher is able to assign topics to any unassigned Students. This is provided, a list of topics are available from the Brainstorming task.

The Submit task allows the Student submit a file. The Teacher will specify the type of file. Only 1 file can be submitted, multiple files will be submitted in 1 zipped file. Any subsequent submission after the first will replace the previous submission.

The Schedule task allows the Teacher to specify available times for Students to present in class. Students have to reserve a time. They can only reserve 1 time.

The Assessment task allows the Teacher to define the assessment criteria and also the rating scales for each criteria. For the Students, they will see all the submitted files. They can view, rate and comment on each one. They can also see the average rating given and previous comments for each submitted file.

The Archive task is a Teacher only task. It allows the Teacher to publish Student's work publicly. This task will be positioned in the Coordination interface, but not with the other tasks.

The following Figure shows a summary of the functions in each task.

2.3 Scope

The objective of our Team is to provide the 'Project' sub-module to be integrated into the pre-existing Moodle system. This sub-module will be able to interact reliably and efficiently with the Moodle system, carrying out all of the functional requirements, and adhering to the non-functional requirements.

As part of the new Project module in Moodle, we will be creating five sub-sections, entitled 'Coordination', 'Brainstorm‘, 'Signup', 'Submit', 'Schedule‘, 'Assessment‘, and 'Archive'. These sections will all be used in different ways to administer projects set by teachers using the Moodle system. Each section will be described later within the functional requirements in Section 3 of this document.

2.4 Product Integration

The proposed system is actually an activity module to be embedded within Moodle system. This section aims to describe how the Project Module will be integrated with the Moodle system.

2.4.1 System Interface

The Moodle system already has a process in identifying all the activity modules. Documentation on the Moodle website needs to be followed for the proposed activity module to be recognised.

Once recognised, the Project Module will not be in use until a Teacher has added it as an activity. Once the Project activity is added, the functionality of the module will be dependent on the type of user (Teacher or Student).

2.4.2 User Interface

Moodle, being a GUI dependent system, all the user interfaces have already been set. The Project Module layout of user interfaces (for Teachers and Students) should be consistent with the standard activity modules. The content of each user interface will be dependent on the type of user and the requirements specified for the user(refer to Section 3 Functional Requirements). All user interfaces should be intuitive and easy to use with help documentation.

2.4.3 Software Interface

The Project Module, being embedded into the Moodle System, will interface with the same software that Moodle itself interfaces with. Moodle runs on Unix, Linux, Windows, Mac OS X, Netware and any other system supporting PHP. This includes most webhost providers. Data is stored in a single database, MySQL and PostgreSQL are best supported, but it can also be used with Oracle, Access, Interbase, ODBC and others.

2.5 User Characteristics

Moodle is a Course Management System used by many people worldwide. The main users of the Project system under development will be teachers and students.

Teachers will have a higher level of access to the Project module than students as they will have certain administrative functions available to them in order to set up and manage each project.

The term ‘teacher’ will collectively be used to describe all teaching users, such as lecturers and tutors. There will be differing levels of access available to each type of teacher, and this will be specified upon creation of each user.

Students will have a standard level of access that will not differ between individuals.

2.6 Constraints

This section outlines the constraints imposed on our system due to various factors. Some of these constraints include:

- Hardware limitations: Some of the users of this system will not have high-end computers with which to access this module. Therefore, we must take into account processing and internet speed limitations when designing the system.
- Interface with Moodle: Our system will have to utilise other functions within the existing Moodle system, and thus we will be constrained by any existing limits of this system.

2.7 Assumptions & Dependencies

This section lists all of the assumptions and dependencies that affect the requirements specified in this document.

- a’s & d’s

3 Functional Requirements

3.1 Coordination

The entire project will be coordinated by the teacher through this stage. High-level project details and tasks will be created. This will be the first stage in creating a project when the ‘Project’ activity within Moodle is chosen.

3.1.1 Enter Project Details

User: Teacher
Description: Teacher will be able to define specifications of the project.
The following details must be entered:

- Project Title
- Project Description
- Project Commencement Date
- Project Completion Date
- Project Late Submission Date

Dependencies/Constraints:

- Commencement date must occur prior to completion date
- Completion date must occur prior to late submission date

Ranking: Essential

3.1.2 Upload Project Files

User: Teacher
Description: Teacher will be able to upload file(s) containing additional project details.
Dependencies/Constraints: Uploaded files must be one of the following types:

- doc, pdf, txt, jpg, bmp, gif, html, xls, mdb, ppt

Ranking: Essential

3.1.3 Create Task

User: Teacher
Description: Numerous tasks may be created by entering the following:

- Task Type - May be one of Brainstorm, Signup, Schedule, Submit or Assessment
- Task Title
- Task Description
- Task Commencement Date
- Task Completion Date
- Task Late Submission Date
- Details specific to individual tasks (Brainstorm, Assessment)

Dependencies/Constraints:

- Commencement date must occur prior to completion date
- Completion date must occur prior to late submission date
- Task commencement date cannot occur prior to Project commencement date
- Task completion date cannot exceed Project completion date
- Task late submission date cannot exceed Project late submission date

Ranking: Essential

3.1.4 Delete Task

User: Teacher
Description: Created tasks may be deleted from within the project
Dependencies/Constraints: Task must exist
Ranking: Essential

3.1.5 View Project Details

User: Student, Teacher
Description: The following project details will be displayed:

- All items listed in Enter Project Details
- List of scheduled tasks and associated dates (ordered ascending by date)
- Status of tasks (completed/not completed)

Dependencies/Constraints: None
Ranking: Essential

3.1.6 Top Ten List

User: Student, Teacher
Description: View the top ten individuals/groups ordered by overall rating for the most recent Assessment Task.
Dependencies/Constraints:

- An Assessment Task must have been completed
- Rating must have been allowed at the Assessment Task creation.

Ranking: Essential

3.1.7 Download Project File

User: Student, Teacher
Description: The user can download uploaded project file(s)
Dependencies/Constraints: Files must previously have been uploaded
Ranking: Essential

3.2 Brainstorm

This must be the first task students will be required to perform as part of a project. This task entails brainstorming topics by the teacher in conjunction with their students. Topics will be added to a brainstorm list. These topics will be fully finalised at the task completion date. No late submissions will be accepted for this task, and any late submission date given will be disregarded.

3.2.1 Brainstorm Details

User: Teacher
Description: The following high-level changes can be made:

- Student ability to add topics may be toggled

Dependencies/Constraints: None
Ranking: Essential

3.2.2 Add Topic

User: Student, Teacher
Description: Topics may be added to a brainstorm list.
Dependencies/Constraints:

- Duplicate topic names will be rejected
- Students can only add topics if allowed during task creation

Ranking: Essential

3.2.3 Edit Topic

User: Teacher
Description: Topic names may be reworded.
Dependencies/Constraints: Renamed topic must have a unique title
Ranking: Essential

3.2.4 Delete Topic

User: Teacher
Description: Selected topics may be removed from the brainstorm list.
Dependencies/Constraints: A topic must have been selected
Ranking: Essential

3.2.5 View Topics

User: Student, Teacher
Description: Brainstorm list will be displayed. The following details will appear:

- Topic Name
- Topic Number

Dependencies/Constraints: None
Ranking: Essential

3.3 Signup

This must be the next stage performed after Brainstorm has occurred. It entails two interdependent tasks, Topic Selection and Group Formation; the order of which will change depending on the choices made by the teacher. On a highlevel, students will be able to assign themselves to topics, finalised from the brainstorm task, and/or assign themselves to groups. Here again there is no late submission dates for this stage. Any changes required after the completion date must be carried out manually with the teacher.

3.3.1 Select Topic Stream

User: Teacher
Description: Links current task to a particular topic stream within a project.
Dependencies/Constraints: Only one topic stream can exist for each signup task.
Ranking: Essential

3.3.2 Define Groups & Topics

User: Teacher
Description: Method for group and topic assignment must be declared.
Method used must be one of:

- Individual work
- Group nomination before topic selection (teacher nominated groups, groups select topics)
- Group selection then topic selection (groups are chosen by students, then group selects topic)
- Topic selection then group view (groups assigned based on topic of interest)

Maximum and minimum group sizes shall also be defined **Dependencies/Constraints**: None
Ranking: Essential

3.3.3 Create Groups

User: Student, Teachers
Description: Several students currently unassigned to a group may be selected and allocated as a group.
Dependencies/Constraints:

- Number of students selected must be less than maximum possible group size
- Students may not create groups where they are not one of the members

Ranking: Essential

3.3.4 Select Topic

User: Student
Description: Students may select topic from a list
Dependencies/Constraints: Voting method of topic selection must be disabled at signup task creation
Ranking: Essential

3.3.5 Vote Topic

User: Student
Description: Students may enter three ordered preferences for topics. Any previous votes cast by student shall be overridden.
Dependencies/Constraints:

- A topic may not be voted for more than once in the first three preferences
- Voting method of topic selection must be enabled at signup task creation
- A total of three preferences must be listed

Ranking: Essential

3.3.6 Allocate Topics (algorithm)

User:
Description:
Dependencies/Constraints:
Ranking:

3.3.7 Assign Topics

User: Teacher
Description: Topics may be assigned to students or groups.
Dependencies/Constraints:

- Topic being assigned must not be currently assigned to another group.
- Group being assigned a topic must not have an existing topic allocation

Ranking: Essential

3.3.8 Unassign Topic

User: Teacher
Description: Students may be separated from their assigned topic.
Dependencies/Constraints: Students may not be unassigned from topics to which they were not allocated.
Ranking: Essential

3.3.9 Add to Groups

User: Teacher
Description: Several students may be selected and assigned an existing group.
Dependencies/Constraints: Transfer of students to a group must not result in groups with more than the maximum number of allowable members.
Ranking: Essential

3.3.10 Disband Groups

User: Teacher
Description: An existing group may be disbanded, moving all students in that group to the unassigned list.
Dependencies/Constraints: None
Ranking: Essential

3.3.11 Edit Voted Topics

User: Student
Description: Previously cast votes on topics may be resubmitted. Previous votes shall be displayed and may be changed.
Dependencies/Constraints: Topics must previously have been voted for
Ranking: Essential

3.3.12 View Topics

User: Student, Teacher
Description: A list of all possible topics must be viewable
Dependencies/Constraints: None
Ranking: Essential

3.3.13 View Topic Allocations

User: Students, Teacher
Description: A list of all students or groups and topics assigned to each must be displayed.
Dependencies/Constraints: None
Ranking: Essential

3.3.14 View Group Members

User: Student
Description: Students may view a list of all students in their group
Dependencies/Constraints: Student must be assigned to a group
Ranking: Essential

3.3.15 View Unallocated Topics

User: Student, Teacher
Description: A list of all topics currently not assigned to a student of group must be shown
Dependencies/Constraints: None
Ranking: Essential

3.3.16 View Unassigned Students

User: Student, Teacher
Description: Students not currently assigned to a group are listed
Dependencies/Constraints: None
Ranking: Essential

3.3.17 View Assigned Students

User: Student, Teacher
Description: Students currently assigned to a group are listed along with their group members
Dependencies/Constraints: None
Ranking: Essential

3.4 Submit

This task must be performed only after Signup has occurred. Submit will enable students to upload their project files. These files can later be assessed by teacher and/or other students through the Assessment Task.

3.4.1 Select Topic Stream

Refer to Select Topic Stream. However, there are no dependencies relevant to this task.

3.4.2 Define Submission File Types

User: Teacher
Description: File types allowed for submission can be selected from the following list: doc, pdf, zip, ppt, xls, txt, mdb
Dependencies/Constraints: None
Ranking: Essential

3.4.3 Upload File

User: Student
Description: Student files are uploaded for perusal. The following must be satisfied:

- Check file type against those allowed by the teacher. Refer to Define Submission File Type.
- Rename file as *tasknamegroupname* for groups
- Rename file as *tasknamelogin* for individuals
- Multiple uploads will overwrite existing submissions

Dependencies/Constraints: None
Ranking: Essential

3.4.4 Delete Submitted File

User: Student
Description: Previously submitted file may be deleted.
Dependencies/Constraints: File must have been submitted previously.
Ranking: Essential

3.4.5 View Submitted File

User: Student
Description: Submitted file may be viewed.
Dependencies/Constraints: File must have been submitted previously.
Ranking: Essential

3.4.6 View All Submissions

User: Teacher
Description: All uploaded files by students may be viewed. The following details will appear:

- Show file name
- Show author

Dependencies/Constraints: None
Ranking: Essential

3.5 Schedule

Signup must occur prior to Schedule but does not have to be directly before. There may be more than one Schedule task per topic stream. Each Schedule task is only associated with a maximum of one Assessment task.
This task will enable students to schedule a presentation time with their teacher.

3.5.1 Select Topic Stream

Refer to Select Topic Stream. However, there are no dependencies relevant to this task.

3.5.2 Define Appointment Duration

User: Teacher
Description: Must define the duration of each presentation.
Dependencies/Constraints: None
Ranking: Essential

3.5.3 Add Session

User: Teacher **Description**: Additional teacher availability sessions may be defined. The following must be satisfied:

- Date of availability
- Start time (24 hour)
- End time (24 hour)

Dependencies/Constraints:

- The start time must be prior to the end time.
- The availability must be scheduled for a future date.
- New sessions cannot overlap with prior session times.
- Excess time remaining that does not divide equally by the appointment duration will be discarded.

Ranking: Essential

3.5.4 Schedule Booking

User: Student
Description: Single available time is scheduled for a presentation. New booking will overwrite prior booking
Dependencies/Constraints: Must be an available time remaining
Ranking: Essential

3.5.5 Remove Available Session

User: Teacher
Description: Previously available session may be removed
Dependencies/Constraints: The relevant session must already be created
Ranking: Essential

3.5.6 Remove Booking

User: Student
Description: Presentation booking can be cancelled.
Dependencies/Constraints: A booking must have already been made.
Ranking: Essential

3.5.7 View Available Sessions

User: Student
Description: The currently available sessions for the selected date may be viewed. The following details will appear:

- Start and End Time of session
- Booking Availability

Dependencies/Constraints: None
Ranking: Essential

3.5.8 View Booking

User: Student
Description: The current session that has been booked may be viewed. The following details will appear:

- Start and End Time of session
- Date

Dependencies/Constraints: None
Ranking: Essential

3.5.9 View Unscheduled Individuals/Groups

User: Teacher
Description: All unscheduled individuals/groups may be viewed. The following details will appear

- Group Name/Number (if group)
- Show Group Members (if group)
- Individual Name and Login (if individual)

Dependencies/Constraints: None
Ranking: Essential

3.5.10 View Scheduled Individual/Groups

User: Teacher
Description: All scheduled individuals/groups may be viewed. The following details will appear

- Group Name/Number (if group)
- Show Group Members (if group)
- Individual Name and Login (if individual)
- Booking Time and Date

Dependencies/Constraints: None
Ranking: Essential

3.6 Assessment

3.6.1 Select Topic Stream

Refer to Select Topic Stream. However, there are no dependencies relevant to this task.

3.6.2 Assessment Details

User: Teacher
Description: The following high-level changes can be made:

- Allow Commenting
- Allow Rating
- Allow students to assess their peers
- Guidelines on commenting

Dependencies/Constraints: None
Ranking: Essential

3.6.3 Create Rating Form

User: Teacher
Description: Ratings forms may be created. The following details will be required:

- Size of scale for ratings

Dependencies/Constraints: None
Ranking: Essential

3.6.4 Add Rating Criteria

User: Teacher
Description: Additional rating criteria may be added with these details:

- Criteria Name
- Enter labels for each rating scale
- First scale entry should relate to a 0 for that criteria

Dependencies/Constraints: None
Ranking: Essential

3.6.5 Comment File/Presentation

User: Student, Teacher
Description:

- Comments may be added for assessments.
- Comments entered shall be in text format. A guideline for commenting will be shown in order to aid the assessment.

Dependencies/Constraints:

- Commenting is allowed. Refer to Assessment Details
- Students are allowed to comment. Refer to Assessment Details

Ranking: Essential

3.6.6 Rate File/Presentation

User: Student, Teacher
Description:

- Ratings may be given for assessments.
- Ratings will be entered and an overall score shall be automatically generated

Dependencies/Constraints: Ratings must be entered for all provided criteria
Ranking: Essential

3.6.7 Edit Rating Form

User: Teacher
Description: Size of ratings scales may be changed.
Dependencies/Constraints:

- Rating scales may not be of zero length
- If rating scale is changed, all ratings made using the old scale will be removed

Ranking: Essential

3.6.8 Remove Rating Criteria

User: Teacher
Description: Rating criterias may be removed
Dependencies/Constraints: None
Ranking: Essential

3.6.9 View Students & Topics

User: Student, Teacher
Description: View list of individual/groups. The following details will appear:

- Individual Name & Login (if individual)
- Group Name/Number (if group)
- Topic Name
- File name (if for Submit Task)
- Overall Rating for each criteria
- List of Comments

Dependencies/Constraints: None
Ranking: Essential

3.6.10 View File for Assessment

User: Student, Teacher
Description: Uploaded files may be viewed for assessment
Dependencies/Constraints: This requires that a file was previously submitted under this topic stream
Ranking: Essential

3.6.11 View Ratings

User: Student, Teacher
Description: View overall ratings for each criteria given to a specific individual/group may be viewed
Dependencies/Constraints: None
Ranking: Essential

3.6.12 View Comments

User: Student, Teacher
Description: A list of all comments given to specific students/groups may be viewed
Dependencies/Constraints: None
Ranking: Essential

3.7 Archive

3.7.1 View List of Uploaded Files

User: Teacher
Description: A list of all files uploaded for the project will be visible here. The following details will appear:

- Individual Name and Login (if individual)
- Group Name/Number (if group)
- Topic Name
- File Name

Dependencies/Constraints: Files have been uploaded during Submit Task
Ranking: Essential

3.7.2 Publish File(s)

User: Teacher
Description:

- All selected files will be published for public viewing
- A page will be created with links to all published files
- All files unchecked will be removed from the public folder (if present)

Dependencies/Constraints:

- Files have been uploaded during Submit Task
- Files have been selected for upload

Ranking: Essential

4 Non-Functional Requirements

This section describes the non-functional requirements of the Product Module. This includes the performance requirements, database requirements, other requirements, hardware constraints and software constraints.

4.1 Performance Requirements

The performance requirements refers to static numerical requirements placed on the interaction between the users and the software.

4.1.1 Interface Loading Time

Description The loading time of each page/interface from the Product Module should be consistent with the other pages already in Moodle. Each page/interface should load in less than 30 seconds.

Constraints The static figure is based on experimentation with the existing Moodle system. This requirement can be dependent on the user's computer or the server itself.

Ranking Desirable

4.1.2 Multiple Access

Description The module shall allow a multiple users to access it. The maximum number of users shall be 100 users at one time.

Constraints The static figure is on an assumption that a very large class will be around 100 students. This requirement could be dependent on the load capacity of the Moodle system and the server it is located on.

Ranking Desirable

4.2 Database Requirements

The database requirements refers to what the module has to create or store in a database.

4.2.1 Create Tables

Description The Project Module when recognised by Moodle, shall create all tables required for the Project Module to store information. The tables may contain information about each Project set, any Project submissions, users, task settings.

Constraints

Ranking Essential

4.3 Other Requirements

The following non-functional requirements are related to some of the software system attributes.

4.3.1 Maintainability

Description The module shall be designed with the view that bugs may need fixing, for future optimisations and for adding extra functionalities for other developers in the Moodle society.

Constraints Inexperience of Team Members in design.

Ranking Essential

4.3.2 Extendability

Description The module shall be designed with the view that thes ”optional” functional requirements will be implemented at a later date. Also the module shall be designed with a view that other tasks may be added to the module.

Constraints The current set of tasks are not independent of each other and the inexperieience of Team Members in design.

Ranking Desirable

4.3.3 Portability

Description The module shall be designed such that it can be added to any Moodle system. It is also necessary for the module to be implemented in PHP and html like the rest of Moodle to make it platform independent.

Constraints None

Ranking Essential

4.3.4 Security

Description The module shall make sure that the correct interfaces and functionalities are available according to the user type (Teacher or Student) that is using it.

Constraints The user type is recognised by the Moodle system.

Ranking Essential

4.4 Hardware Constraints

This hardware Constraints refers to the server-side and user-side required hardware specifications so that the module can be run with Moodle.

4.4.1 Server System Specification

Description Any server system can host Moodle provided they have the software required (Refer to Software Constraints).

Constraints The server system must have the minimum hardware which the required software specifies.

Ranking Essential

4.4.2 User System Specification

Description Any computer system can run Moodle provided they have the software required. (Refer to Software Requirements).

Constraints The user's computer system must have the minimum hardware which the required software specifies.

Ranking Essential

4.5 Software Requirements

This software requirements refers to the server-side and user-side requiredhardware specifications so that the module can be run with Moodle.

4.5.1 Server System Software

Description The server system must have the minimum software to host Moodle.

Constraints Unix, Linux, Windows, Mac OS X, Netware and any other system that supports PHP. Data is stored in a single database: MySQL and PostgreSQL are best supported, but it can also be used with Oracle, Access, Interbase and ODB.

Ranking Essential

4.5.2 User System Software

Description The user's computer system must have the minimum software to run Moodle.

Constraints Any web browser more current than Internet Explorer 5, Firefox 1, Mozilla 1 or Netscape 6 on any version of Mac/Linux/Windows.

Ranking Essential

5 Use-cases

6 SRS Modification Process

This contains the procedure to be followed if the needs arises to modify the SRS once the requirements have been signed-off by the client.

6.1 Team Modification Process

The procedure for the Team to modify the SRS is listed below:

1. A meeting will be called by the initiating Team Member to inform the Team of the proposed modifications.
2. If all Team members agree to the proposition, the Team Supervisor will be informed.
3. If the Team Supervisor agrees to the modifications, the Client will be informed.
4. The Client will have a meeting with the Team Members and the Team Supervisor (where possible).
5. If the proposed modifications are agreed upon, the changes will be made to the SRS, and a sign-off will take place on the revised requirements. This will effectively create a new agreement between the Project Team, annulling the original SRS Acceptance.
6. If no agreement is made, the Team and the Client will abide by the original SRS Acceptance agreement.

6.2 Client Modification Process

The procedure for the Client to modify the SRS is listed below:

1. The Client will modify the CLO of any proposed changes (via email or a hard copy).
2. The CLO will schedule a Team meeting to inform all other Team Members of the proposed changes and make a decision on their feasibility.
3. If the Team agrees to the proposed changes, the Team Supervisor will be informed.
4. If the Supervisor advises against the changes or the Team does not believe the changes are feasible, the client will be informed. The Client will be given the option of revising the proposed changes or maintaining the original SRS.
5. If the Supervisor advises the Team to proceed with the changes, the SRS will be modified to incorporate the additions and a sign-off will take place.

7 Acceptability Criteria

This section outlines the criteria that the Project Team must meet to ensure that the Client accepts the SRS and the final software product.

7.1 SRS Acceptance

To ensure that this document is acceptable, the Client must review all requirements and verify that they are consistent with those specified.

The Client must inform the Development Team of any ambiguities or inconsistencies with this document and all requirements therein prior to sign-off.

7.2 Product Acceptance

For the final product to be accepted by the client the following criteria must be met:

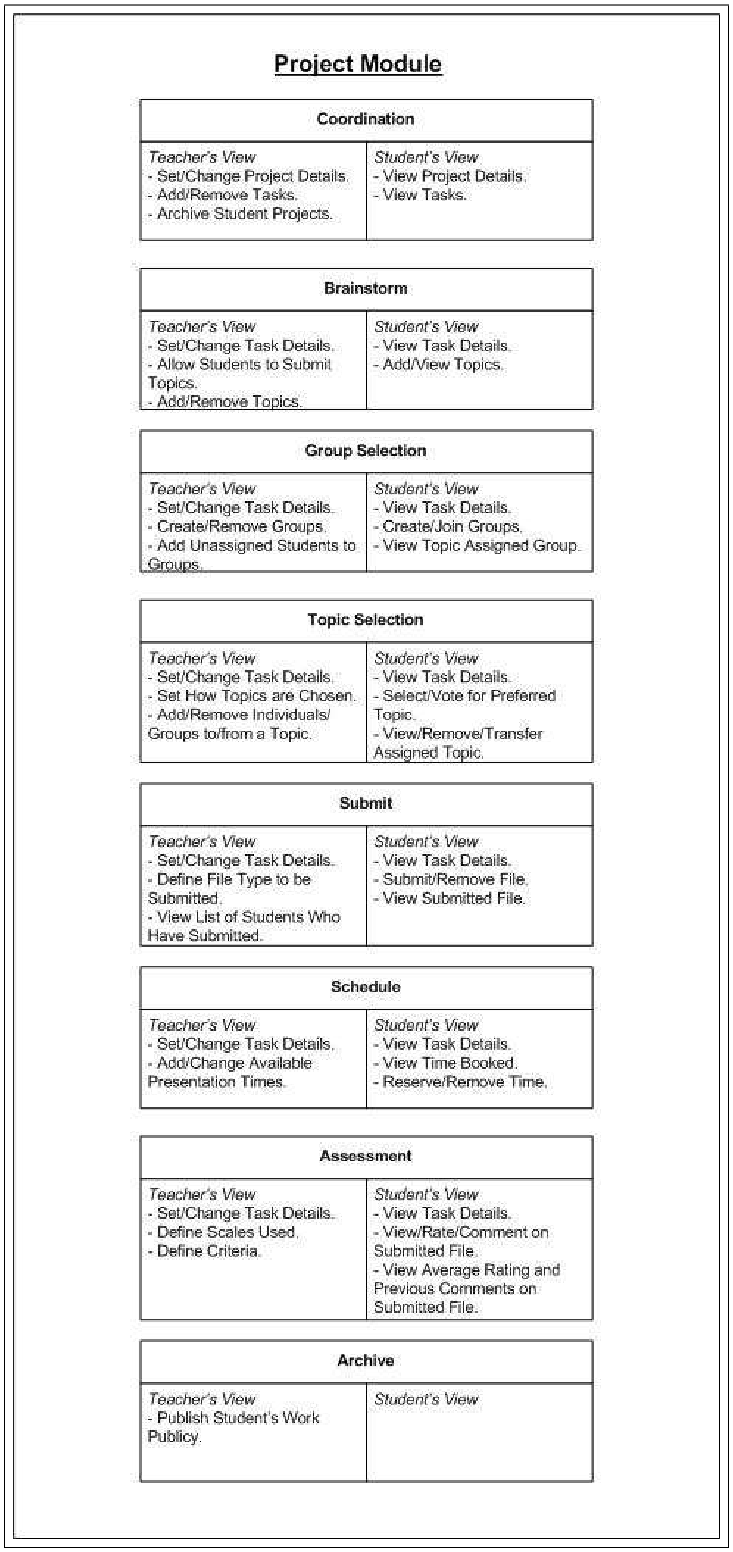
- The product must include all of the essential functional requirements as specified in Section 3 of this document for the product to be considered useable.
- The product does not have to include any of the optional functional requirements as specified in Section 3. The Team will attempt to incorporate these requirements, time permitting.
- The system must be installed, free of bugs, onto the Moodle CMS.
- The code must be documented (with comments) according to the Moodle coding standards adopted by Moodle engineers worldwide.
- A hard-copy of the User Documentation must be provided to the Client.

8 Client Sign-Off

8.1 SRS Sign-Off

I (the Client) hereby agree that I have read and understood this document. The requirements specified in this document accurately represent the requirements for the proposed Project Module to be incorporated into the Moodle CMS.

Any proposed changes to the requirements as specified in this document are to be made in writing to the Team as outlined in 6 SRS Modification Process.



Project Module

Coordination	
Teacher's View - Set/Change Project Details. - Add/Remove Tasks. - Archive Student Projects.	*Student's View* - View Project Details. - View Tasks.

Brainstorm	
Teacher's View - Set/Change Task Details. - Allow Students to Submit Topics. - Add/Remove Topics.	*Student's View* - View Task Details. - Add/View Topics.

Group Selection	
Teacher's View - Set/Change Task Details. - Create/Remove Groups. - Add Unassigned Students to Groups.	*Student's View* - View Task Details. - Create/Join Groups. - View Topic Assigned Group.

Topic Selection	
Teacher's View - Set/Change Task Details. - Set How Topics are Chosen. - Add/Remove Individuals/ Groups to/from a Topic.	*Student's View* - View Task Details. - Select/Vote for Preferred Topic. - View/Remove/Transfer Assigned Topic.

Submit	
Teacher's View - Set/Change Task Details. - Define File Type to be Submitted. - View List of Students Who Have Submitted.	*Student's View* - View Task Details. - Submit/Remove File. - View Submitted File.

Schedule	
Teacher's View - Set/Change Task Details. - Add/Change Available Presentation Times.	*Student's View* - View Task Details. - View Time Booked. - Reserve/Remove Time.

Assessment	
Teacher's View - Set/Change Task Details. - Define Scales Used. - Define Criteria.	*Student's View* - View Task Details. - View/Rate/Comment on Submitted File. - View Average Rating and Previous Comments on Submitted File.

Archive	
Teacher's View - Publish Student's Work Publicy.	*Student's View*

Figure 4: Tasks within Project Module