# CMM007 – Intranet Systems Development Coursework Overview

|  |  |
| --- | --- |
| Date of Assessment Release | Coursework is released during the second teaching week of the semester. |
| Word Count | There is no word count limit to this coursework |
| Submission Date | Part 1: 4pm Friday 28th February  Part 2: 4pm Friday 17th April |
| Submission Method | All coursework submissions are to be submitted via CampusMoodle |
| Number of Copies to be Submitted | Only 1 digital copy of each submission should be made.  No physical copies are required |
| Date for Release of Marks | Marks will be released 2 weeks after the final submission. |
| Date for Release of Feedback | A provisional grade with individual feedback will be made available on Moodle within 20 working days of submission. |
| How Feedback Will be Returned | Feedback will be returned via CampusMoodle |
| Learning Outcomes | This coursework covers the following learning outcomes:   1. Design and implement dynamic WWW pages appropriate to a given objective.   3. Develop server−side applications. |
| How Grades are Calculated | Grades are combined from each coursework element and then a weighted average is used to calculate the final grade. |

## Introduction

In this coursework you are required, as an individual, to design, build and test a web application of your choice that employs some of the technologies you will learn throughout this module. This is likely to include HTML, CSS, PHP and MySQL.

## Choosing a web application

The decision of what web application you wish to create is up to you. You can either come up with your own idea or use one of the ideas shown as a guide. Below are five sample web applications that you may wish to consider creating. Note these are only ideas and, if you choose one, you will still need to precisely specify the functionality you plan to deliver. However, you are encouraged to create an application of your own choice.

Please note that you only need to create **one** application.

### A system for monitoring project ethical approval

Every experiment that takes place at Robert Gordon University by a student must receive ethical approval before commencing. This is currently a manual process achieved with a form that is filled in and then stored online. This project will digitize the process by creating a web application so that the ethical approval of an experiment can take place in a more efficient manner. In this system, there are the following types of users:

* Administrators
* Experiment Approval Officers (EAO)
* Students

Accompanying this, the following basic functionality should be implemented

* Students should be able to sign up to the system, create an experiment, and then fill in the ethics form attached to this. They should also be able to edit the ethics form of a project.
  + Students should be able to upload additional files that supplement their ethical approval form (e.g. consent forms)
* Experiment Approval Officers should be able to view experiments that have been assigned to them, give comments on an experiment, and finally decide whether to grant ethical approval for a project.
* Administrators should be able to assign experiments to EAO’s.
* All experiments should be assigned to 2 different EAO’s for feedback.

### Peer assesement in group projects

In some modules, you are required to take part in a group project. In these instances, it is necessary for all group members to work together to complete a project. It is expected that all members will work with the same level of commitment and enthusiasm. However, this does not always occur and as such, peer assessment of individuals within group tasks is an important aspect. There is an opportunity for a web application to be created that could support this process. In this system, there are the following types of users:

* Administrators
* Students

Accompanying this, the following basic functionality should be implemented.

* Administrators should login to the system to gain access
* Administrators should be able to set up a coursework item and then should be able to split students into groups for the coursework.
* Students should enter the system through a personalized link and can then give feedback on how they worked in a group.
  + Students should be able to upload files to supplement their feedback
  + Students should not see the feedback from other students
* Administrators should be able to see a detailed report for each group showing how students worked together

### Group Project Team Manager

When creating groups for projects, group selection can either take place by members of academic staff, or by students selecting their own groups. There are advantages and disadvantages to both methods. A project exists where a web application can be made to aid in the creation of these groups. This system will require the following types of users:

* Administrator
* Students

Accompanying this, the following basic functionality should be implemented:

* Administrators should be able to enter the names / emails of students in a class (either manually or through upload)
* The system should be able to randomly create groups from a class list
  + This may involve constraints such as a minimum/maximum number of students in a group or splitting degree cohorts.
* The list of groups should be viewable by all students in the class (through a login system)
* Students should be able to request to swap groups with other students
* Group members should be able to leave comments for each other in a group area
  + Group members should also be able to share files in this area.

### Project Research Paper Sharing App

In some modules, you are required to take part in a group project in which the project group members read and review research papers as part of the project. It can be difficult for group members to keep up-to-date with the research papers that have been identified and which ones they should be reviewing. There is an opportunity for a web application to be created that could support this process allowing project teams to be created, relevant research papers to be stored, and reviews to be allocated and uploaded. In this system, there are the following types of users:

* Administrator
* Student Team Leader
* Students

Accompanying this, the following basic functionality should be implemented.

* Administrators should login to the system to gain access
* Administrators should be able to sets up a project and allocate students to the project, giving one the role of team leader.
* Students should enter the system through a personalized link and can then
  + be able to upload research papers
  + see any papers that have been allocated to them to review and upload their reviews
  + be able to view all papers and reviews
* Team Leaders in addition to the student functionality should be able to allocate papers to students for review

### Sports Team Manager

The Student Union sports section wants an application to help sports team managers organize team selection for sports fixtures in sports such as five-a-side football or badminton. It is often difficult knowing who is available for particular fixtures and then informing the players they have been selected. An application would allow a squad to be allocated to a team, squad members to identify their availability for particular fixtures and the team manager to select a team from the available squad members. A project exists where a web application can be made to aid in the managing of the team selection process. This system will require the following types of users:

* Student Team Manager
* Students Squad Members

Accompanying this, the following basic functionality should be implemented:

* A team manager should be able to log in and enter a fixture list
* The system should be able to allow students to sign up as squad members
* Student squad members should be able to identify their availability for each fixture
* The team manager should be able to select a team for each fixture.
* Students should be able to request to swap groups with other students
* All users should be able to view teams selected for each fixture.

## Project Constraints

The following constraints must be followed for any project that is created:

* Your completed web application must be hosted on an server (accessible to the marker) with up-to-date code stored on GitHub (All code **must** be available on GitHub);
* Your web application must contain both front end (client side) and back end (server side) code;
* The created web application must contain the following features:
  + A login system;
  + More than one user role, e.g. administrator and general user;
  + Some type of file upload system; and
  + A system for users to input data that is stored and then recalled from a database.

Submissions that do not follow the above constraints do not meet the coursework requirements.