F21DG - Design and Code Project

MACS Department Workload Management System

Deliverable One

February 7th 2020

Authors

Tommy Lamb - H00217505
Dylan Forrest - H00216546

Contents

1 Management Plan

Due to the small size of the team a flat informal management system will be applied — the overheads of any formal system would not yield any benefit to the project. Meetings between members and stakeholder(s) will be held regularly to ensure project success.

2 Evaluation of Existing System

We've evaluated the existing system and decided that we want to reuse, as much as possible, the existing application code. In particular we wish to reuse most of the GUI code to minimise changes to the end users. We do anticipate needing to refactor the existing code however, to improve maintainability and clarity. Similarly we have decided to redesign and implement the database component with only vague reference to the existing design, due to inadequacies in it.

3 Requirements Specification

3.1 Purpose

At a high level the primary purpose of the Workload Management System (WMS) is to facilitate the fair distribution of work within the department across staff members. Additionally it is to provide a transparent view of all work currently being carried out (and by whom) to all members of the department. To achieve this goal the WMS will enable staff members to record their existing and upcoming tasks, and provide an overview of the data. The system may also draw attention to situations when certain conditions are met, to assist administration of the department.

3.2 User Characteristics

There are three conceptual groups of users which will interact with the system. Note that any permissions (or lack thereof) detailed below are conditional on implementation of item F-UR-3.2.

1. **Superuser/root**
   This group of users are responsible for maintaining and running the WMS software. They will have the ability to override or change any aspect of the system, and will be the only user (in principle) with direct database and server access. They will, whether directly or by proxy, have all of the permissions granted to the Administrator group in addition to their own.

2. **Administrator**
   This group are those members of staff responsible for the assignment and management of tasks at a departmental (or similar) level. For example, the Head of School. These users will have permission to create, view, and edit any task within the system, however they will be constrained to use the WMS interface. They will also be the only users (alongside the Superuser by necessity) who will be able to view sensitive information.

3. **Other Staff**
   Any member of staff within the MACS department who does not fall within one of the other 2 groups belong to this group. For example this may include, but is not limited to, Research Assistants, PhD Students, Lecturers, and Administrative Staff. This group will have the ability to view (but not edit) all tasks within the system. They will have the ability to create tasks to which they are assigned (in other words, to record in the system agreements made by external processes) and edit these – and only these – tasks.

   Note that as a consequence of this definition, all staff members of the MACS department are considered users of the system (either in this group, or another).

3.3 Functional User Requirements

F-UR-1 **Record individual tasks**
Priority: Must Have

F-UR-1.1 **Recording of generic task details**
Priority: Must Have
For every new task created the appropriate data must be recorded, and no task will be recorded which fails to meet the minimum requirements. These requirements as follows:

1. Name
2. General Text Description
3. Task Classification (See item F-UR-1.2)
4. One or more Assigned Staff Members (See item F-UR-1.5)
5. Estimated Measure of Workload
6. Timespan over which the task will be carried out (See item F-UR-1.6)

Specific task classifications may require the recording of additional data.

F-UR-1.2 **Classification of Tasks**

Priority: Must Have

Tasks recorded in the system must be assigned one and only one of the following classifications:

1. Teaching
2. Research
3. Administrative
4. External Activity
5. Other Authorised Engagement

No classifications outside of those listed here should be used.

F-UR-1.3 **Recording of teaching task details**

Priority: Must Have

In addition to the details required in item F-UR-1.1, teaching tasks require and must not be stored without the following details:

1. Course Code (For example, F29LP)
2. Course Co-ordinator. The single member of staff principally responsible for delivering the course.
3. Number of students participating in the course

F-UR-1.4 **Recording of research task details**

Priority: Must Have

In addition to the details required in item F-UR-1.1, research tasks require and must not be stored without the following details:

1. State of funding – either Funded or Unfunded (and not both)
2. Principal Investigator. The single member of staff principally responsible for carrying out the research.

F-UR-1.5 **Assignment of staff to tasks**

Priority: Must Have

For any given task it must be possible to assign it to one or more staff members. It must also be possible to remove staff from any given task, except where that member of staff is the only one assigned to it. Where only one member of staff is assigned to a task, removal of that member of staff is as yet undefined behaviour and must be considered unsupported.

For every assignment of a staff member to a task, there must be an associated percentage denoting to what degree the staff member contributes to the overall work of the task. The sum of staff member contributions for any given task must not exceed 100%. The sum of contributions is permitted to be less than 100%, however this should be considered a degenerate case (see item F-UR-6.4). No staff member is permitted to have a contribution of 0% to any task.

F-UR-1.6 **Scheduling of tasks**

Priority: Must Have

Every task must have a start date specified. If applicable, a task may also have an end date. Where a task does not have an end date specified it must be considered to continue in perpetuity. Where an end date is specified, the task may optionally repeat after some time interval – for example, annually. The start and end dates, and any repetition must be able to be changed at any time.

F-UR-2 **Editing and Deletion of tasks**

Priority: Must Have

It must be possible to edit the details of any given task at any time. It must also be possible to delete tasks from the system at any time. Editing, and particularly deleting, tasks may be subject to restrictions to maintain data or organisational integrity where necessary.

F-UR-3 **Access control**

Priority: Variable

Highest Priority: Must Have

F-UR-3.1 **Restrict access to authorised users only**

Priority: Must Have

Access to the application must be controlled, such that only authorised users can access it. In general, any staff member of the MACS department (and only of the MACS department) is considered an authorised user. This does not preclude other persons outwith the department being granted access, nor persons within the department being denied access. The mechanics of granting or denying specific persons access is an organisational and technical implementation detail outwith the scope of this document.

F-UR-3.2 **Variable permissions levels for users**

Priority: Should Have

The ability of any given user to perform operations within the system should be limited inline with their responsibilities within the department. More specifically, users and their permissions should be limited as defined in Subsection 3.2.

F-UR-3.3 **Use of existing authentication mechanisms**

Priority: Could Have

If possible, the system could have the ability to utilise an existing authentication mechanism used within the university – for example, the MACS departmental login. Depending on the func-

tionality afforded by any such mechanism, that mechanism may become the only used within the system.

F-UR-4 **Recording of staff details**
Priority: Must Have
In order that they may usefully be assigned to tasks (as per item F-UR-1.5), the following details should be recorded for all users:

1. Name
2. Office Number
3. Email address
4. Telephone Extension Number
5. Campus

This data must be able to be edited at any time.

F-UR-5 **Grouping of staff members**
Priority: Could Have
The system could support the grouping of staff members to mirror the organisation of the department. For example, by research interest or by department within a school.

F-UR-6 **Notification of degenerate or noteworthy conditions**
Priority: Variable
Highest Priority: Should Have

F-UR-6.1 **Reporting tasks nearing completion**
Priority: Should Have
Users should be notified when tasks they are assigned to are nearing completion. Such notifications could also be shown to Admin users.

F-UR-6.2 **Reporting exceptional individual loads**
Priority: Should Have
Admin users should be notified when members of staff have workloads above or below the departmental average, where the workload exits variance. This notification may also include fixed limits which are not defined relative to the departmental average.

F-UR-6.3 **Reporting imbalances in an individual's task classifications**
Priority: Should Have
Admin users should be notified when the distribution of a user's

time between Teaching, Research, and Administrative tasks exceeds acceptable limits. For example, if a user spends a disproportionate amount of time on Research tasks compared to teaching and administration.

F-UR-6.4 **Reporting under-assigned tasks**
Priority: Should Have
Admin users should be notified when the total percentage contribution of users to a given task does not equal 100%.

F-UR-6.5 **Suppression of notifications**
Priority: Could Have
Users could have the ability to suppress notifications such that ongoing situations, or exceptions to the established rules, do not dilute the value of the notifications overall. An excessive number of notifications could cause users to disregard all notifications. Conversely users may forget and fail to deal with situations after suppressing notifications(s). As such, the extent to which notifications can be suppressed, if at all, must be decided through experimentation and consequently is out of scope for this document.

F-UR-7 **Planning view**
Priority: Should Have
The system should provide to admin users a 'planning view', allowing them to carry out item F-UR-1 and item F-UR-2 without the changes being visible to other users in order to plan for the future. Tasks created in this view will be subject to all of the same restrictions and notifications. There should also be the ability to 'import' all tasks from the main system view in the planning view, such that they can be edited in isolation without affecting the main view of the system or other non-admin users.

F-UR-8 **Overview of assigned tasks**
Priority: Must Have
The system must provide to all users a general overview of all assigned tasks. The exact data to be included is not specified, however the following are suggested and given as examples:

1. A bar chart of the workloads of individual staff members to allow easy visual identification of outliers.
2. A pie chart of the distribution of work between different task classifications for a given user. This may also be aggregated over a group of users.
3. A line chart showing the anticipated future workload for an individual. This may also show multiple users, either as multiple

lines on the chart or in aggregation.

F-UR-9 **Support of multiple languages**
Priority: Will Not
The system will not support multiple languages. The system should support Unicode, however the interface will be in English only.

3.4 Non-Functional User Requirements

NF-UR-1 **Academic date system**
Priority: Should Have
The date system used within WMS should be based upon the academic year, rather than the calendar (or any other) year system.

NF-UR-2 **Supported browsers**
Priority: Must Have
The system must be accessible and functional for end users using either Google Chrome or Mozilla Firefox. Other browsers may work, but are not explicitly supported.

4 Risk Analysis

The risks identified in this section should be quantified with respect to the following impact matrix:

	Consequence				
Likelihood	Trivial	Minor	Moderate	Major	Severe
Almost Certain	Low	High	High	Extreme	Extreme
Likely	Low	Moderate	High	High	Extreme
Possible	Low	Moderate	Moderate	High	Extreme
Unlikely	Low	Moderate	Moderate	High	High
Rare	Low	Low	Moderate	Moderate	High

Extreme risks are those requiring immediate corrective action. High risk issues require some immediate mitigation, and may require investigation leading to additional corrective actions. Moderate risks will benefit from adaptations or adjustments to mitigate consequence or likelihood, but do not require immediate action. Low risk items are either considered acceptable occurrences, or can be adequately dealt with as and when they arise.

The following risks have been identified (see next page):

Risk	Mitigation	Likelihood	Consequences
A team member falls ill	Keep group members up to date on current work in progress, and keep this work accessible	Unlikely	Moderate
Loss of data	Back up work regularly to an external version control server/service. Ensure that there is always at least 2 copies of any data and those copies are not co-located either physically or virtually.	Rare	Major
Failure to implement main features in time available	Weekly progress meetings with stakeholder(s) to ensure the project is on track. Triaging features if necessary to ensure a minimum-viable product is delivered	Possible	Moderate
Change of requirements	Regular meetings with stakeholder(s) to ensure changes are communicated as early as possible	Likely	Moderate
Insufficient time for thorough testing	Test software continuously during development as new features are implemented (unit and integration testing)	Possible	Moderate
Major issue arising after delivery	Extensive acceptance testing by the end users prior to handover, and continuous testing during development	Possible	Major
UCU members strike	Ensure that all required data is accessible without reliance on members of the UCU. University facilities have been maintained during previous strike action, and with no related teaching being delivered, impact is trivial	Almost Certain	Trivial
Severe weather preventing travel	Ensure mechanisms are in place for communication and data sharing across the internet	Rare	Minor
Expected hosting environment is insufficient	Develop and continually test the software in the intended hosting environment. Identify early any inadequacies and discuss resolutions with stakeholder(s)	Rare	Major

5 Test Plan

5.1 Scope

5.1.1 In Scope

- Unit tests for each new feature that is added.
- Systems and integration test for each new feature.
- Test that the system can function on a range of devices with different resolutions.
- Test the system on various browsers, focusing on those listed in item NF-UR-2.
- Test the system for basic security flaws such as cross site scripting and SQL injection.

5.1.2 Out of Scope

These features will not be tested due to be out with the requirements of the project.

- Stress test. Not necessary as only the MACS department will be using this.
- Advanced security testing, e.g denial of service attacks.
- Performance of the system. This is something that we will be monitoring throughout the implementation but not explicitly testing it.

5.2 Test Methodology

The test methodology that we will be adopting for this project is the agile methodology. We are using agile so that the system can be thoroughly tested throughout the implementation process. Due to only having two people as part of this team we will be carrying out a lot of these tests as we go. Before a new feature is added to the system we will each test it to make sure it doesn't disrupt the flow of the current system. Once we have implemented the entire system we will rerun our test plan over the entire system. During the implementation of the system we will also be conducting code reviews to make sure that the code used in the final system is properly written and commented.

5.2.1 Test Levels

- Unit testing: Testing the individual components (e.g. functions, database queries) that comprise the new feature.
- Integration testing: Conducted when a new feature is added to the system to check that the new feature doesn't disrupt the current flow of the system.
- System testing: Conducted on the completed system to check if all of the requirements have been met and there are no major issues with the completed system.

5.2.2 Bug Triage

When there is an issue detected during testing we will add this new issue as a new ticket in GitHub. This ticket should have a sensible title and a complete description of what the problem was and how to recreate it. Tickets will only be raised for issues found in the master branch.

5.3 Test Deliverables

Below is a list of each deliverable that will be provided during each stage of development.

Before Testing

- Test plan
- Evaluation strategy

During Testing

- Test data
- Test scenarios

After Testing

- Final test results
- Release notes

6 Evaluation Strategy

6.1 Introduction

The point of the evaluation is to get user feedback on the design and the feel of the system. The evaluation will be conducted once near the end of the development process to check with the stakeholder(s) if it is what they envisioned, and that it meets acceptable usability levels.

6.2 Evaluation Methodology

6.2.1 Aims

- Brief each participant
- Gather data on the usability of the product
- Record any critical flaws about the design and functionality of the product that arise
- Debrief each participant

6.2.2 Hypothesis

The participant can complete a given scenario with ease and with no help from a third party.

6.2.3 Participants

The desired participants are those who will, upon completion, be using the system as part of their regular work activities. We aim to have a representative sample of all anticipated users in the MACS department participating.