

Aberystwyth Web Evaluation Surveys Of Module Experiences (*AWESOME*)

Report Name	Outline Project Specification
Author (User Id)	Benjamin Brooks (beb12)
Supervisor (User Id)	Hannah Dee (hmd1)

Module	CS39440
Degree Scheme	G401 (Computer Science)

Date	February 12, 2015
Revision	1.0
Status	Release

1 Project description

The Aberystwyth Web Evaluation Surveys Of Module Experiences (AWESOME), is a prototype that enables departments to gather feedback by students about modules, lecturers, and other departmental issues. It is intended to replace and improve upon the current method of collecting feedback via Google Forms. This can be achieved by providing a personalised survey for each student to make questions personalised whilst also keeping results anonymous and confidential, and being able to chase up students for not completing their questionnaire.

AWESOME is a PHP web application developed by Keiron O'Shea during the summer of 2014 under the supervision of Dr. Hannah Dee. This project's goal is to bring the current prototype up to a well written, functioning, implementable, and extensible standard. Security of the system is critical, and so implementing a continuous integration system with unit tests and vulnerability scanning is vital to get up and running early on in the project. The system must be multilingual and accessible to adhere to the university's policies.

Advanced analytics and reports is also a feature which is highly requested by university management. For example, the system needs to be able to extract textual comments from the worst performing modules containing the word 'Feedback'. This can help upper management identify problematic areas in the university and look into the issue further.

The prototype has been demoed at a Learning and Teaching Enhancement Committee meeting to gather feedback about the current status of the project. The consensus from the committee is that the prototype is very impressive and would solve a need that is longed for by the university management. This confirmation by the committee further proves a need for this software to follow best software development practices to ensure that it can be used and extended in the future.

2 Proposed tasks

The prototype is currently written in procedural PHP, using the Twig [?] framework as a templating engine. In order to make the program more extensible and easier to maintain, it would make sense to refactor the current codebase to follow an object oriented (OOP) model-view-controller (MVC) [?] [?] architectural pattern.

The first task that needs to happen is a full security audit of the current version of the prototype. From there, the project can be refactored using best software practices, such as OOP, unit testing and MVC to separate the view and logic.

After the refactoring is done and the software is up to the same functional specification, additional functionality can then be introduced. The following task lists are what can be expected of each aspect of the project.

2.1 Refactoring tasks

Change procedural design – Design a new software architecture using MVC with OOP.

Unit Testing – Use Travis CI [?] to do automated unit testing and vulnerability scanning.

Secure admin dashboard – Use LDAP HTTP authentication via *.htaccess*.

PHP Data Objects (PDO) – Change the current *mysqli* and *tidy_sql* implementation to use PDO for greater security, flexibility and features when interacting with databases.

Accessibility (a11y) audit – Ensure all student-facing pages are accessible for disabled users.

2.2 Additional Functionality

Internationalisation (i18n) – Reimplement i18n system to support additional languages.

Relational Database – Modify the database schema to be object oriented and relational.

Advanced Analytics – Create a system which can narrow down responses to criteria (e.g. Find modules with a low satisfaction score which mention ‘feedback’ in comments)

Traffic Light Dashboard – Have a dashboard showing traffic lights for all modules and departments to detect current issues.

3 Project deliverables

As this project has a fairly tight and a fixed deadline on the temperature questionnaire testing, a lot of the refactoring work will be done as soon and as quickly as possible in order to get the project to a state that could be used.

After this initial sprint, the project can then be extended with the additional features described above. A rough outline of timeframes for deliverables is below.

Outline Project Specification – 2015-02-06 – This document.

OOP MVC Class Diagram – 2015-02-09 – UML Class diagram to describe MVC design.

OOP MVC Release – 2015-02-20 – Functional OOP MVC version of the prototype.

i18n and a11y – 2015-02-24 – Add internationalisation support and audit accessibility.

Temperature Test – Week 4-5 – Internal/closed functional testing.

Temperature Questionnaire – Week 6 – Questionnaire sent out to two departments.

Mid-Project Demonstration – 2015-03-09 – Start date of Mid-Project Demonstrations.

Analytics/Reports feature – 2015-04-17 – Have the analytics feature finished.

Final Report – 2015-05-07 – Final report hand-in.

Final Demonstrations – 2015-05-11 – Final project demonstrations.