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 6, 2015
Revision	0.1
Status	Draft

1 Project description

The Aberystwyth Web Evaluation Surveys Of Module Experiences (AWESOME), is primarily a tool 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 answering.

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.

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 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. Additionally, the system must be multilingual and accessible to adhere to the university's policies.

2 Proposed tasks

The project is currently written in procedural PHP, using Twig [2] 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) [5] architectural pattern.

The number one priority on this project is to do a security audit on the software currently as it stands and a sanity check on the logic behind it. After this, the design of the OOP MVC version of the software can begin as well as the other tasks listed.

2.1 Short Term Tasks

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

Unit Testing – Use Travis CI [1] to do automated unit testing and vulnerability scanning.

Protect admin dashboard – Using 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.

2.2 Long Term Tasks

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

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

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

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

2.3 Future Considerations

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

3 Project deliverables

TODO: Temperature Test – 2015-03 – Internal/closed functional testing.

TODO: Temperature Questionnaire – 2015-03 – Questionnaire sent out to two departments.

–

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 software.

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

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

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

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

Annotated Bibliography

[1] "Travis CI: Building a PHP project," http://docs.travis-ci.com/user/languages/php/, Feb. 2015, accessed Feb 2015.

> Travis CI is a hosted continous integration system that connects to GitHub to help with automated unit testing and vulnerability scanning.

[2] "Twig - The flexible, fast, and secure template engine for PHP," http://twig.sensiolabs.org, Feb. 2015, accessed Feb 2015.

> Twig is the templating engine using by Keiron

[3] K. T. Brinko, "The Practice of Giving Feedback to Improve Teaching: What Is Effective?" *The Journal of Higher Education*, vol. 64, no. 5, 1993. [Online]. Available: http://www.jstor.org/stable/2959994

[4] C. Hopkins, "PHP Master — The MVC Pattern and PHP," http://www.sitepoint.com/the-mvc-pattern-and-php-1, Feb. 2015.

[5] J. Stump, "Understanding MVC in PHP," http://archive.oreilly.com/pub/a/php/archive/mvc-intro.html, Feb. 2015.

[6] H. K. Wachtel, "Student Evaluation of College Teaching Effectiveness: a brief review," *Assessment & Evaluation in Higher Education*, vol. 23, no. 2, pp. 191–212, Jan. 1998. [Online]. Available: http://dx.doi.org/10.1080/0260293980230207