



ABERYSTWYTH UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

PROJECT AWESOME

Project Plan

Author:
Keiron O'SHEA (KEO7@ABER.AC.UK)

June 2014

version: 0.5
status: draft

Contents

1	Introduction	1
1.1	Purpose of this document	1
1.2	Scope	1
1.3	Objectives	1
2	Overview	2
2.1	Platforms and High Level Architecture	2
2.1.1	Hypertext Markup Language (HTML)	2
2.1.2	Cascading Style Sheets (CSS)	2
2.1.3	PHP: Hypertext Preprocessor (PHP)	2
2.1.4	PostgreSQL	2
2.1.5	Analytics	2
2.1.6	Internet Connectivity	2
2.1.7	Staff Access to SAMS/AStRA	2
2.1.8	Generate student questionnaires	3
2.1.9	Collate feedback	3
2.1.10	Automated targeted reminder emails	3
2.2	Description of Target Users	3
3	Use Case	4
3.1	Use Case Diagram	4
3.2	Use Case Descriptions	4
4	User interface design	5
4.1	User Navigation	5
4.2	GUI Design	6
4.2.1	Home Page	6
4.2.2	Questionnaire Home	7
4.2.3	Generating a Questionnaire	8
4.2.4	Vetting a Questionnaire	9
4.2.5	Student Questionnaire	10
4.2.6	Reports Home	11
4.2.7	Report of a Module	12
4.2.8	Help	13
5	Project Management	14
5.1	Gantt Chart	14
5.2	Risk Analysis	14
5.3	Technical Risks	14

5.4	Project Risks	14
-----	-------------------------	----

1 Introduction

1.1 Purpose of this document

The purpose of this document is to present how I've translated the project brief, the Aberystwyth Web Evaluation Surveys Of Module Experiences (AWESOME), a web-based module evaluation questionnaire generator for the monitoring and evaluation of teaching, into subsystems of objectives and milestones. Moreover this document will describe the various interactions between separate components of the system and outline initial ideas surrounding the visual appearance of the user interface.

This document will also discuss project management with the use of a Gantt Chart, providing an overview of the project's major tasks and related milestones. This will coexist with a probabilistic analysis of risks, of which will cover some main issues that may occur throughout the development of the project.

1.2 Scope

The project plan should intend to give an overview of the proposed system, including a brief description of platforms and HLA - as well as provide a description of target users. It will also contain a user case diagram that aims to give an overview of how the users might interact with each-other. An idea of the user-interface design will also be provided with descriptions of how it will interact with the end user. In terms of project management, a Gantt chart will be used to display the start and end dates for the main tasks of the project and a risk analysis that will highlight issues the project may encounter.

1.3 Objectives

The objectives of this particular document are:

- to describe the appearance of the user interface.
- to indicate an understanding of user behaviour.
- give further indication of the proposed time-line surrounding the development of the project.
- to indicate an understanding of the risks and other factors of the project.

2 Overview

The "Aberystwyth Web Evaluation Surveys Of Module Experiences", or AWESOME for short, is a web-based module evaluation questionnaire generator for the monitoring and evaluation of teaching.

2.1 Platforms and High Level Architecture

I intend to use the following platforms for the system.

2.1.1 Hypertext Markup Language (HTML)

The project brief stated specifically that the system is to be deployed completely via the web. We will be using Hypertext Markup Language (HTML) as the markup language used to create the application.

2.1.2 Cascading Style Sheets (CSS)

In order to produce an aesthetically-pleasing web application I will utilise CSS to specify the styles of the visual elements of the website.

2.1.3 PHP: Hypertext Preprocessor (PHP)

On the server side PHP will be used to handle communication between the client and the database and handle the creation of the questionnaires as well as providing dynamic web pages. PHP is widely available on the majority (if not all) web servers and is relatively easy to deploy compared to similar systems.

2.1.4 PostgreSQL

PostgreSQL is one of the most widely used database platforms and will be used to store application information. The platform is renowned for its simplistic commands and is easily deployed and managed.

The high level architecture consists of the following elements and application mechanics;

2.1.5 Analytics

For staff to analyse feedback analytics consisting of visually-appealing graphs and textual responses will be rendered.

2.1.6 Internet Connectivity

Both staff and students will require connection to the server to allow them to use the application. Google Charts has been pinpointed as a possible way of doing this as it comes with a readily available API, and still keeps in line with the approach of using free software to build the entire system.

2.1.7 Staff Access to SAMS/AStRA

In order to retrieve module information both both staff and students we would like to utilise either SAMS (Student Attendance Monitoring System) or AStRA (Aberystwyth Student Records and Admissions). Both of these systems require staff authentication.

2.1.8 Generate student questionnaires

We need to ensure that the questions asked are relevant to the module. An example could be that in the Department of Computer Science we could have department-level questions (about certain rooms, or advisory). Although not a necessity, AWESOME needs to be robust and portable enough to extend so that other departments can also ask specific questions.

Most, if not all questionnaires will contain questions such as:

- "Rate this on a scale of 1-5"
- Multiple choice questions.
- Additional feedback about department. (Textual)

2.1.9 Collate feedback

This component is activated when a student submits a completed questionnaire. In order to ensure that the responses are completely anonymous it will store the answers to the database with no record of the user ID of the submitter. Ideally it would be easy to export the data to a format for processing in Microsoft Excel/LibreOffice Calc (CSV)- we also aim to utilise the stored data to implement some visualisation on the AWESOME website.

2.1.10 Automated targeted reminder emails

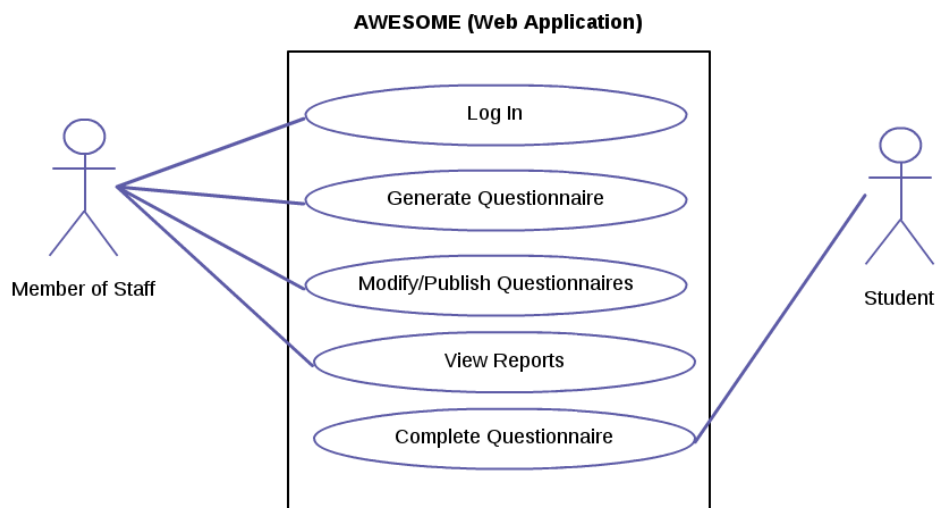
Students that are vet to complete a form will be targeted and 'pestered' about completing a questionnaire. This should be automatic and persuasive - however it's important not to fill the email inboxes of those students, there's nothing more scary than a disgruntled Computer Science student.

2.2 Description of Target Users

All design must begin with the understanding of who the user is, but due to the relatively large range of age and expertise of the target-base this will be difficult to achieve. The proposed system will be targeted to both staff and students who are currently working or studying at a higher education institute. Because of this inability to pinpoint a specific user base, it would be best to ensure that when designing the user interface we need to ensure that the application is simple to use (in order to avoid frustration and confusing) it must also be informative.

3 Use Case

3.1 Use Case Diagram



3.2 Use Case Descriptions

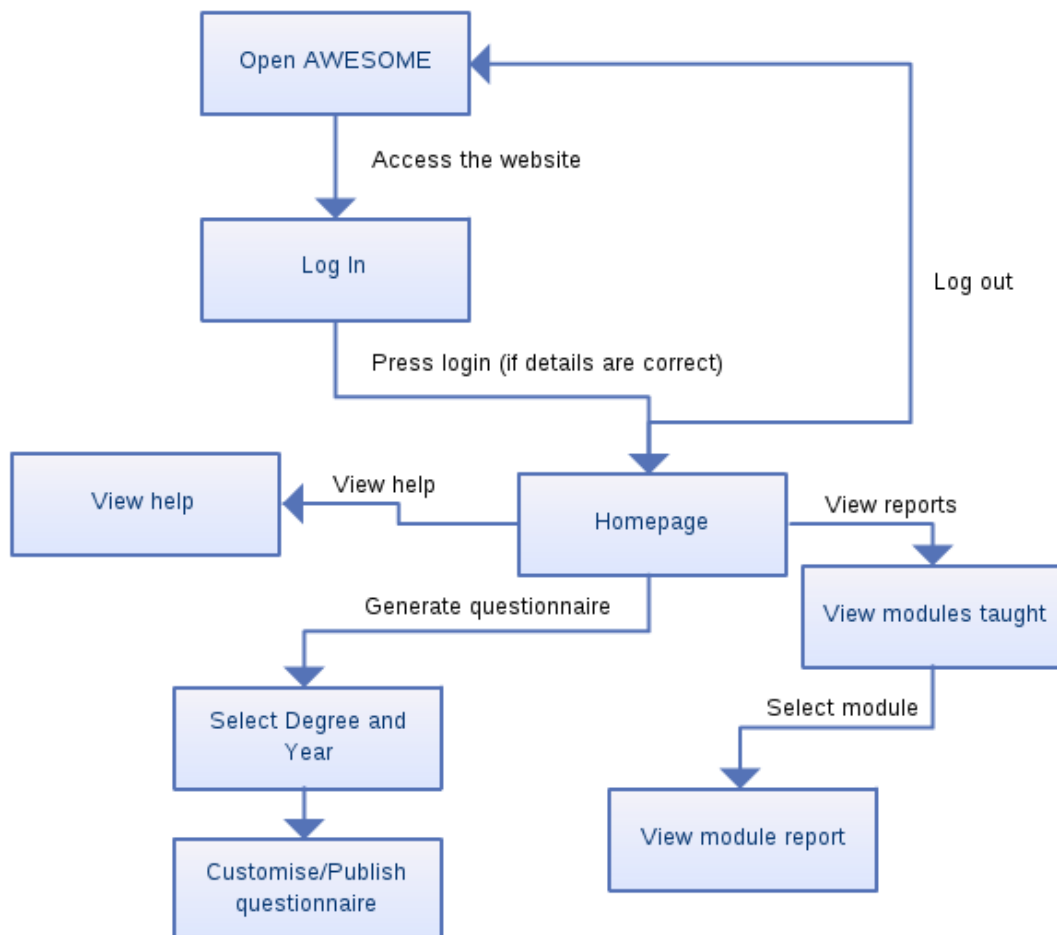
Member of staff	
Log in	A registered member of staff should be able to log in to AWESOME providing valid AU credentials.
Generate Questionnaire	Must be able to select what year/degree to generate a questionnaire for.
Modify/Publish Questionnaire	Must be able to add additional questions and publish questionnaires to students.
View Reports	Must be able to view statistics of past/ongoing questionnaires and compare them within the department or university.

Student	
Complete Questionnaire	When the student has got a individual link to complete the questionnaire, then they must be able to complete a questionnaire and submit it anonymously.

4 User interface design

4.1 User Navigation

Below is a flow diagram which describe how the application is navigated.



4.2 GUI Design

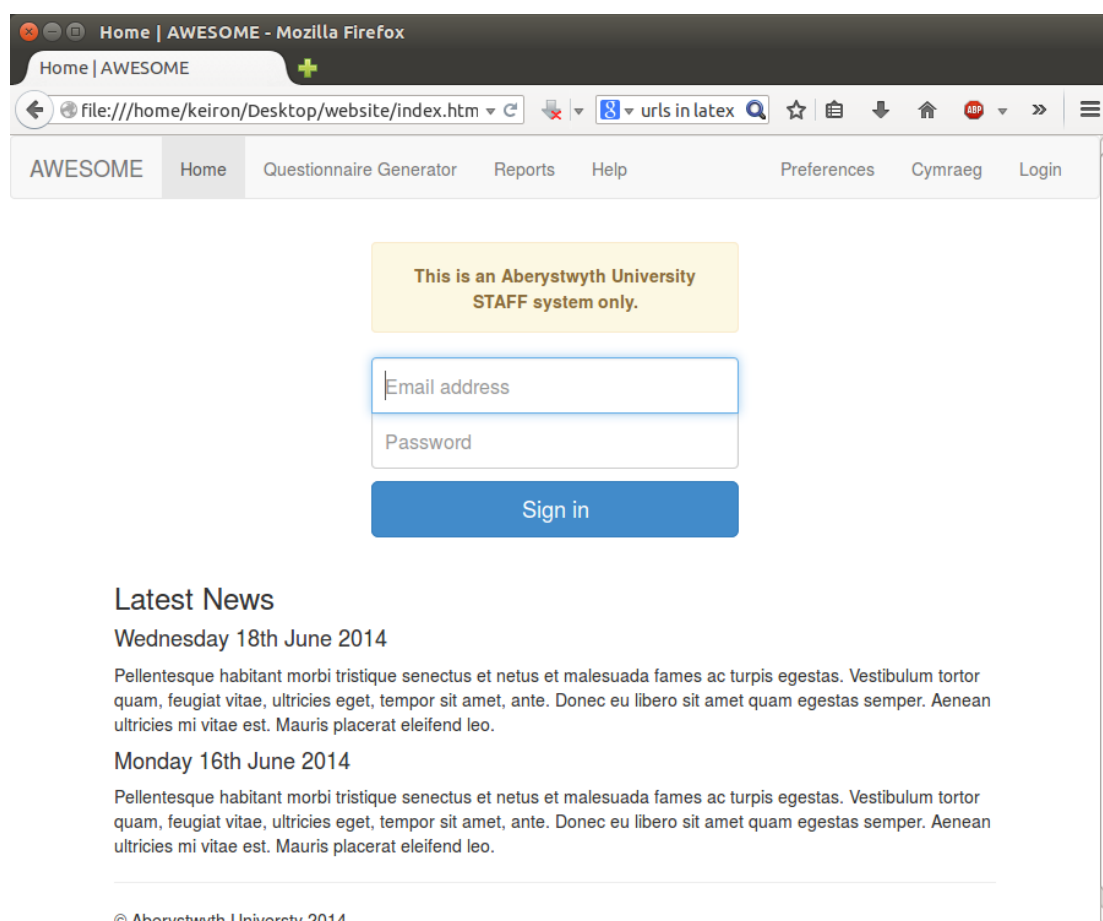
The following images cover the initial concept of the application. These designs were made using bootstrap and can be accessed at the following url:

<http://htmlpreview.github.io/?https://github.com/Keiron0/AWESOME/blob/master/Prototyping/UIDesign/index.html>

4.2.1 Home Page

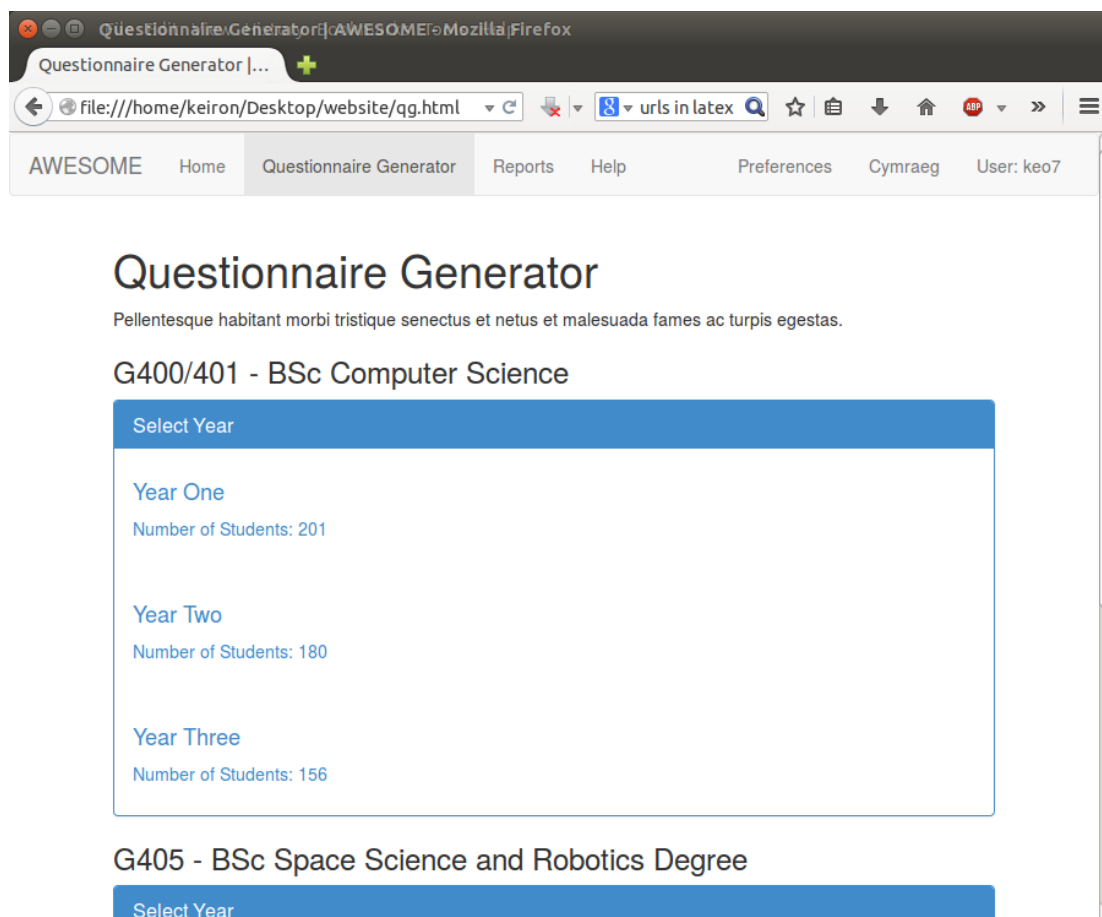
The homepage is the first view to be presented to the applications user, inviting them to log onto the AWESOME system by entering their staff credentials for the e-mail address and password fields that are to be validated by the server upon the push of the login button. Inline with other BIS systems, a warning will be placed announcing that you need to be a member of staff of the University to have access to this system.

The page will also consist of a section consisting of news surrounding the AWESOME system. It's important to keep staff informed of expected downtime due to maintenance.



4.2.2 Questionnaire Home

Once the user has successfully logged in using the correct credentials they can then have full access to the AWESOME system. On the homepage of the questionnaire generator the user will be greeted by options of what degree scheme and year they would like to send a questionnaire to. Underneath each year the number of students enrolled for that year will be listed.



4.2.3 Generating a Questionnaire

Once the user has selected what degree and year they would like to generate a questionnaire for, they will be redirected to the questionnaire generator which will display all modules on offer for that particular degree/year and give the option to add additional (*special/departmental*) questions.

Although not in the figure below, it may be a good idea to also show the members of staff that teach that module as well as the number students enrolled on each module. I also plan on giving a checkbox to allow for 'quick' questionnaires. These will only consist of the "one-question-per-module" schema.

The screenshot shows a web browser window titled "G400 - Questionnaire Generator" with the address bar displaying "file:///home/keiron/Desktop/website/qg2.html". The main content area lists eight modules:

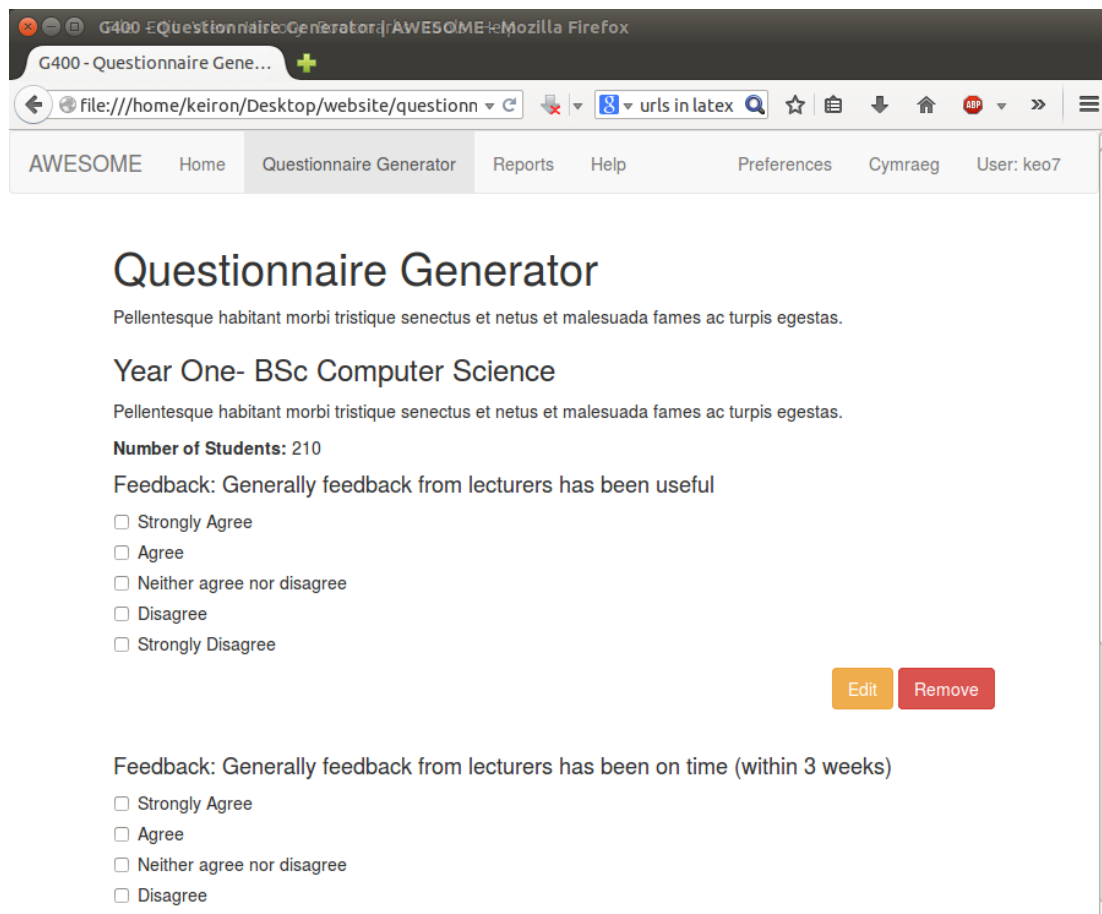
- CS10110 - Introduction to Computer Hardware, Operating Systems and Unix Tools
- CS10410 - The Mathematics Driving License for Computer Science
- CS12020 - Introduction to Programming
- CS12510 - Functional Programming
- CS15020 - Web Development Tools
- CS10720 - Problems and Solutions
- CS12320 - Programming Using and Object-Oriented Language
- CS15210 - An Introduction to Communications and Telematics
- CS18010 - Professional and Personal Development

Below the module list is a section titled "Additional Questions" containing three input fields labeled "Additional Question One", "Additional Question Two", and "Additional Question Three". Each field has a placeholder text "Additional Question". To the right of these fields is a button labeled "Add Questions". At the bottom right of the page is a blue button labeled "Generate Questionnaire". The footer on the bottom left reads "© Aberystwyth Universty 2014".

4.2.4 Vetting a Questionnaire

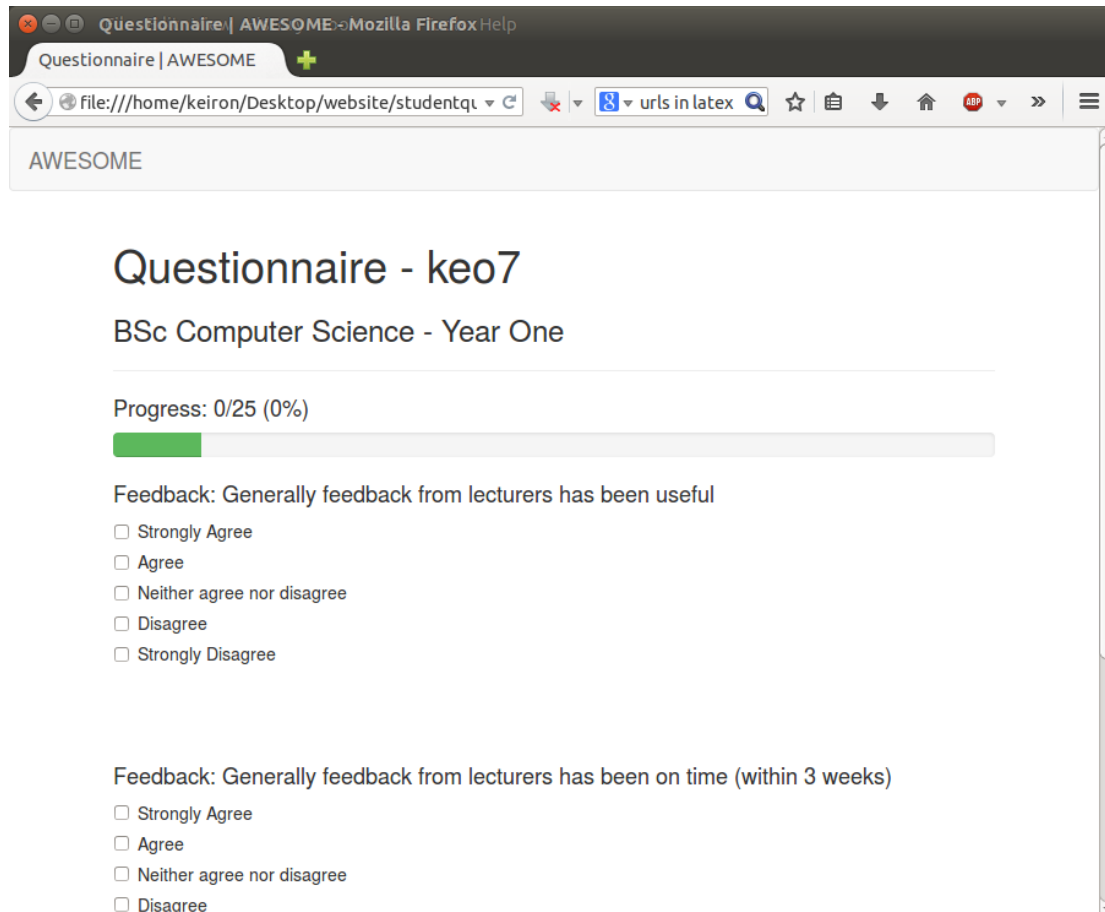
After additional questions have been added and the user has generated a questionnaire, they will then be able view the questionnaire and make any desired changes.

Once finished, a button will be provided in order to email the survey out to the students.



4.2.5 Student Questionnaire

This will be the webpage sent out to the students, it will contain simple instructions as well as a progress bar to ensure that the student knows how long until they have successfully completed the questionnaire. Once they have completed the questionnaire, a dialogue box will appear congratulating the user upon his/hers completion.



Questionnaire | AWESOME

Questionnaire | AWESOME

File:///home/keiron/Desktop/website/studentq

AWESOME

Questionnaire - keo7

BSc Computer Science - Year One

Progress: 0/25 (0%)

Feedback: Generally feedback from lecturers has been useful

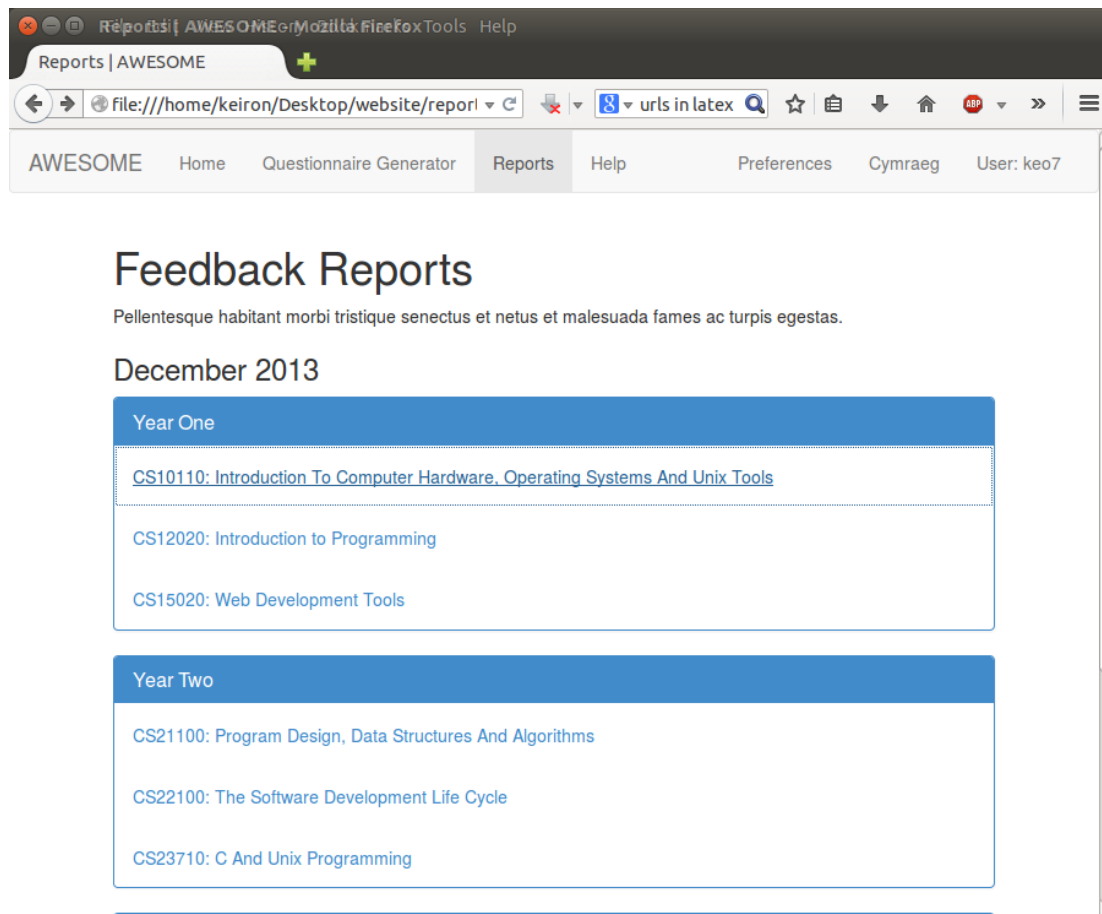
- ☐ Strongly Agree
- ☐ Agree
- ☐ Neither agree nor disagree
- ☐ Disagree
- ☐ Strongly Disagree

Feedback: Generally feedback from lecturers has been on time (within 3 weeks)

- ☐ Strongly Agree
- ☐ Agree
- ☐ Neither agree nor disagree
- ☐ Disagree

4.2.6 Reports Home

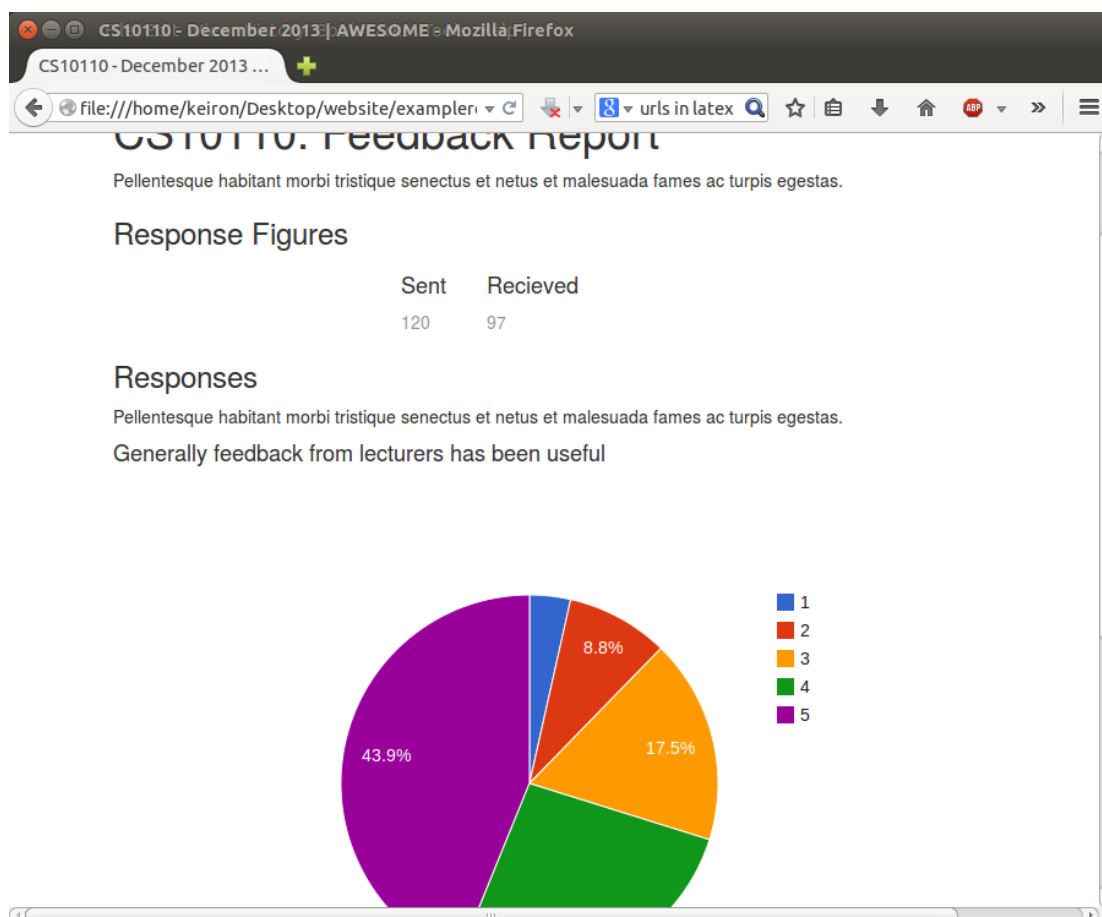
The reports page will consist of all the feedback available for that member of staff to look at, consisting of both modules taught and their department. It may also be a good idea to allow the comparison of departmental performance, but this does not feature in the prototype.



4.2.7 Report of a Module

Once a user has selected a module/department to analyse they will be greeted by an in-depth report of how that module has done in previous surveys. It will consist of response figures outlining how many questionnaires were sent out and how many the department received.

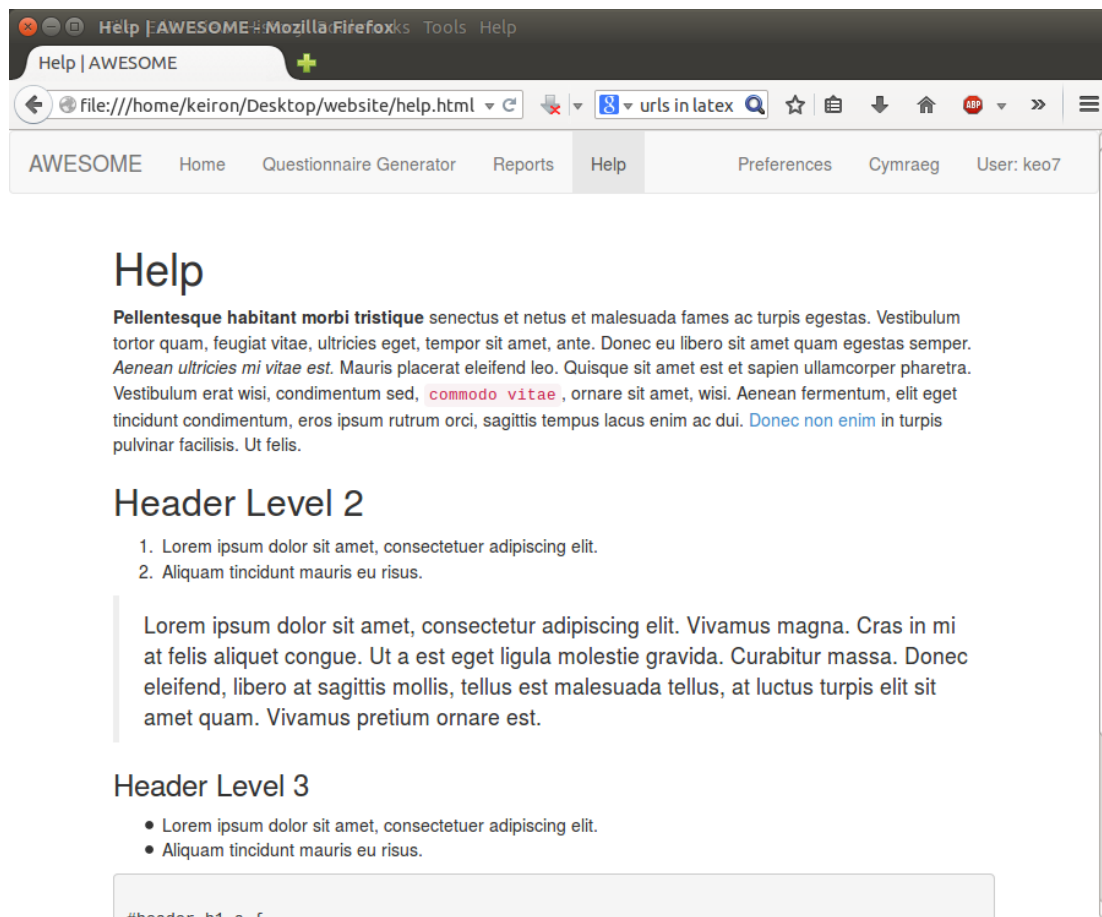
Below that carefully formatted results of each question will be provided with HTML5 driven charts driven by Google Charts. If the user wants to analyse the results more closely then they can by seeing the direct results. Each chart/result will also contain a brief textual explanation.



4.2.8 Help

This page will mainly be text-based, but it may be a good idea to include video walking users through some of the more common issues they may encounter.

Not that there will be, my software is flawless.¹

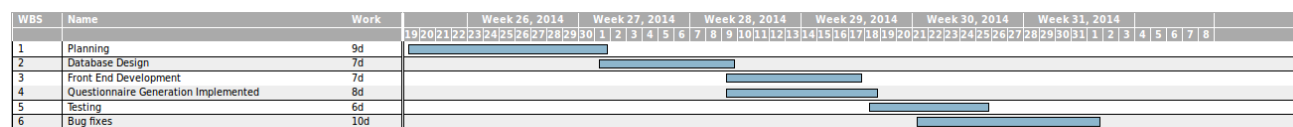


¹I wish

5 Project Management

5.1 Gantt Chart

The below figure is a Gantt chart, a visual representation of the project schedule. It describes the project milestones and task dependencies.



5.2 Risk Analysis

5.3 Technical Risks

Potential Risk	Mitigation	Concern
Bugs Prevent Correct Usage	Produce a comprehensive test plan, assign multiple testers	High
Push broken code into repository, rendering software unusable	Use of branches mandated for code, pull requests function as a code review	High
Server is temporarily unavailable	Wait. Nothing we can do.	Medium
Error with git repository causes loss of data / unavailability	Use of GitHub provides a reliable git service, git is decentralised	Low

5.4 Project Risks

Potential Risk	Mitigation	Concern
Some features take too long to implement	Prioritise features and postpone lower priority ones	High
Application not working as described in requirements	Leave enough time for testing	High
One part of the project falls behind	Modular system allows work to continue in other areas	Medium
Incomplete areas of the documentation	Regular feedback to be provided and more people to assist with tasks if required	Medium
Codebase becomes fragmented	All code to be pushed to git repository, coordinate work	Medium
The client redefines the project requirements	Unlikely to happen, meetings will be held and requirements can be re-drafted	Low
Accidental deletion of file from repository	GitHub provides a robust mechanism for retrieving backups and versioning	Low

Document History

Version	Date	Changes made to Document	Changed by
0.5	19-06-2014	Completed UI / Risk Analysis	KeironO
0.2	17-06-2014	Completed overview	KeironO
0.1	16-06-2014	Initial creation	KeironO