Contents

[1. Introduction 2](#_Toc508700443)

[2. Agile Scrum Environmental Framework 3](#_Toc508700444)

[2.1 Delegate roles 3](#_Toc508700445)

[2.2 Tools and technology 3](#_Toc508700446)

[2.3 Development team meetings. 3](#_Toc508700447)

[2.3.1 Sprint Zero Planning - first meeting 3](#_Toc508700448)

[2.3.2 3](#_Toc508700449)

[2.3.3 4](#_Toc508700450)

[2.3.4 4](#_Toc508700451)

[2.4 Prioritisation Product Backlog 4](#_Toc508700452)

[2.5 Sprints planning 4](#_Toc508700453)

[2.6 Sprints burndown charts. 4](#_Toc508700454)

[2.6.1 Sprint one 4](#_Toc508700455)

[2.6.2 Sprint two 4](#_Toc508700456)

[2.6.3 Sprint three 4](#_Toc508700457)

[3. Product Design and documentation 4](#_Toc508700458)

[3.1 UML Diagram 4](#_Toc508700459)

[3.2 Database design and documentation 5](#_Toc508700460)

[3.2.1 Third normal form 5](#_Toc508700461)

[3.2.2 ERD diagram 5](#_Toc508700462)

[3.3 User Interface Design 5](#_Toc508700463)

[3.3.1 Typeface 5](#_Toc508700464)

[3.3.2 Colour schema 5](#_Toc508700465)

[3.3.3 logo 5](#_Toc508700466)

[3.3.4 Wireframe design 5](#_Toc508700467)

[3.3.5 Site map 5](#_Toc508700468)

[4. Product Development 5](#_Toc508700469)

[4.1 Application File structure 5](#_Toc508700470)

[4.2 Code snippets 5](#_Toc508700471)

[4.3 Screen shots of the application in operation 5](#_Toc508700472)

[5. Testing 5](#_Toc508700473)

[5.1 Testing Plan 5](#_Toc508700474)

[5.1.1 Name of the item for test 5](#_Toc508700475)

[5.1.2 6](#_Toc508700476)

[5.2 Test log 6](#_Toc508700477)

[6. Evaluation 6](#_Toc508700478)

[6.1 Initial assumptions 6](#_Toc508700479)

[6.2 Product Strength 6](#_Toc508700480)

[6.3 Product Weaknesses 6](#_Toc508700481)

[6.4 Conclusion 6](#_Toc508700482)

# Introduction

The role of this group report is to outline the development process in Agile environment of a web-base software application designated to serve as an important tool for the QA manager of a large university and to encourage academics and support staff, not only to post ideas for improvements but also to comment and vote on each other’s thoughts. The application helps the university’s QA Manager to oversee this process and provide real support for administration decisions making, and all through the power of processing data used to crate reports and graphic representation. The application’s name is Inspiration and is the same as its development team.

The following list is the high-level requirements specification extracted from the coursework scenario.

* Secure role base system
* Unregistered user can navigate through the website and view ideas and comments.
* Logged in user can view ideas and comments but also can post new ideas (optional can attach a files images, pdf and Microsoft Word documents) or comment on other users’ ideas.
* Logged in user must agree T&C before posting an idea or a comment.
* Logged in user can see the date/time of last logged in.
* The user must receive email notification when new comment is made to his idea.
* The users can post ideas or comments anonymously, but its details will be recorded in the database.
* The logged in user must be able to report any inappropriate comments/idea.
* The QA Manager can view reports about users’ activity, number of ideas per university’s departments, anonymous posted ideas.
* The QA Manager must be able to crate new category for tagging ideas but also to delete those ones that hasn’t got any ideas.
* The QA Manager must be able to block/unblock users.
* The QA Manager must be able to hide/show ideas form/to other users.
* The QA Manager must be able to view Google Analytics.
* The Departments QA Coordinators can see the activities for their department and receives email notification when new idea has been posted by one of their department staff.

# Agile Scrum Environmental Framework

## 2.1 Delegate roles

## 2.2 Tools and technology

## 2.3 Development team meetings.

### 2.3.1 Sprint Zero Planning - first meeting

### 2.3.2

### 2.3.3

### 2.3.4

## 2.4 Prioritisation Product Backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Priority** |  | **Scoring points** | **Assigned to** |
|  | High |  |  |  |
| 1 |  |  |  |  |
|  | Medium |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
|  | Low |  |  |  |
| 4 |  |  |  |  |

## 2.5 Sprints planning

## 2.6 Sprints burndown charts.

### 2.6.1 Sprint one

### 2.6.2 Sprint two

### 2.6.3 Sprint three

# Product Design and documentation

## 3.1 UML Diagram

## 3.2 Database design and documentation

### 3.2.1 Third normal form

### 3.2.2 ERD diagram

## 3.3 User Interface Design

### 3.3.1 Typeface

### 3.3.2 Colour schema

### 3.3.3 logo

### 3.3.4 Wireframe design

### 3.3.5 Site map

# Product Development and functionality

Following its design and requirements specification, using a combination of tools and techniques the application has been developed to align with current technology standards. PHP Object Oriented programming is at the core of its development along with MySQL HTML5, CSS3 and JavaScript. The decision of utilizing those technologies along with other tools (NetBeans IDE, Highcharts) has been taken by the group at its initial meeting (Minutes, p1).



Figure 1. Responsive design.

The application’s responsive design feature is detected by the browser through the “viewport” meta tag that triggers the CSS media queries “@media screen and (sizes)”.

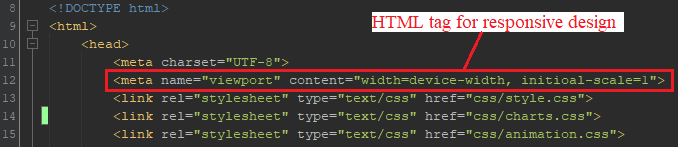


Figure 2. HTML code for responsive design

For crating CSS files, the “mobile first” approach was considered. Below is an example of CSS code for responsive design.

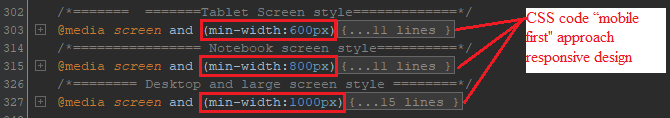


Figure 3 CSS code for responsive design

The security of the system is addressing four main issues: password encryption, Cross-Site Request Forgery (CSRF), SQL injection and Role-base access.

To apply the technology standards, the user’s login password credential is saved in the database with 64 characters encrypted format using the PHP “password\_hash” function. This not only prevents from hacking the account but also provides user details privacy.

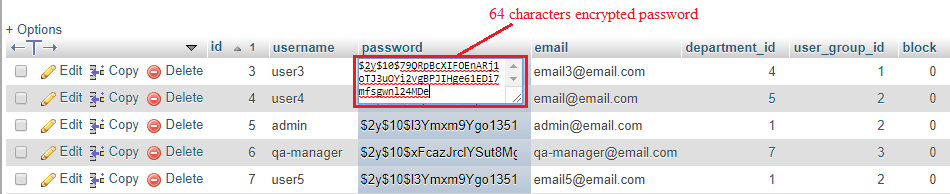


Figure 4. Database user password encryption format.

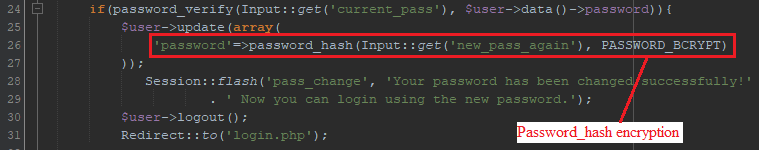


Figure 5 PHP password\_hash encryption function

The Cross-Site Request Forgery (CSRF) security issue is addressed by calling a token.

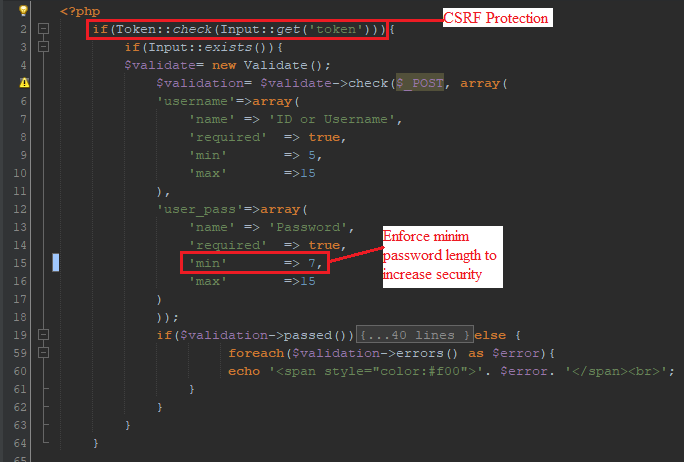


Figure 6. Use of token for CSRF and minimum password length enforcement code example

To protect the system from malicious SQL injection, the “htmlentities()” function has been used either ways to sanitize data for the input and escape data for the output.

This function not only escapes the string that is passed, but also allows the single and double quotes and defines the type of character encoding (in this case the UTF-8).



Figure 7. Escape function that prevents from SQL injection

Role base security

The application has been developed with a role base security system that is capable to restrict user access to different pages. The third sprint of this project development process has included this functionality and is relevant from the user story number IN-34.

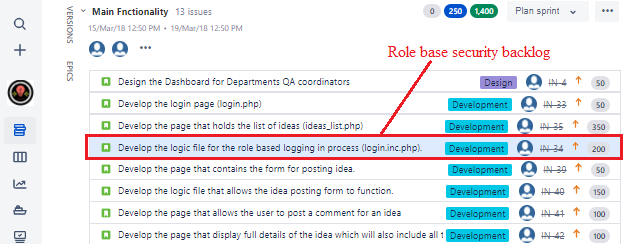


Figure 8. JIRA Sprint Three Role base security backlog

This functionality has been created with a combination of a database table and a method (function) from “User” class. The database table contains all the roles that a user of the system can be registered with. The user role can be changed at any time but only by the system administrator (by default all users are registered as standard user).

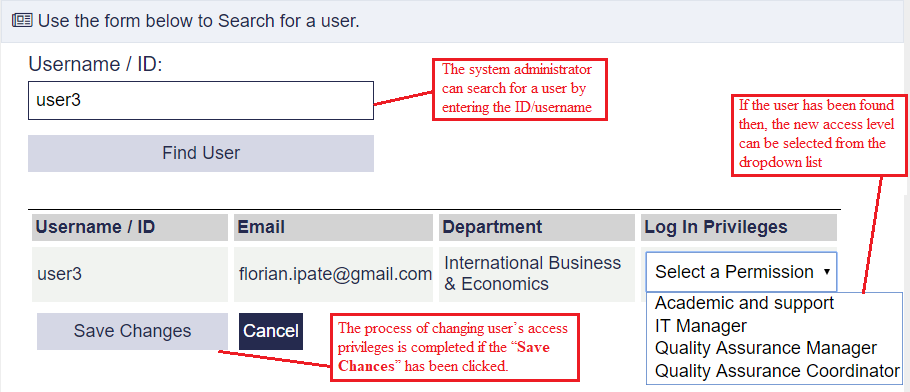


Figure 9. System admin changing the user3 access level.

Depending on the role, the user, after logs in, is redirected to specific pages. For example, the academics and support staff will view the page that lists all the posted ideas in a chronological order.



Figure 10 The landing page for the logged in Standard user

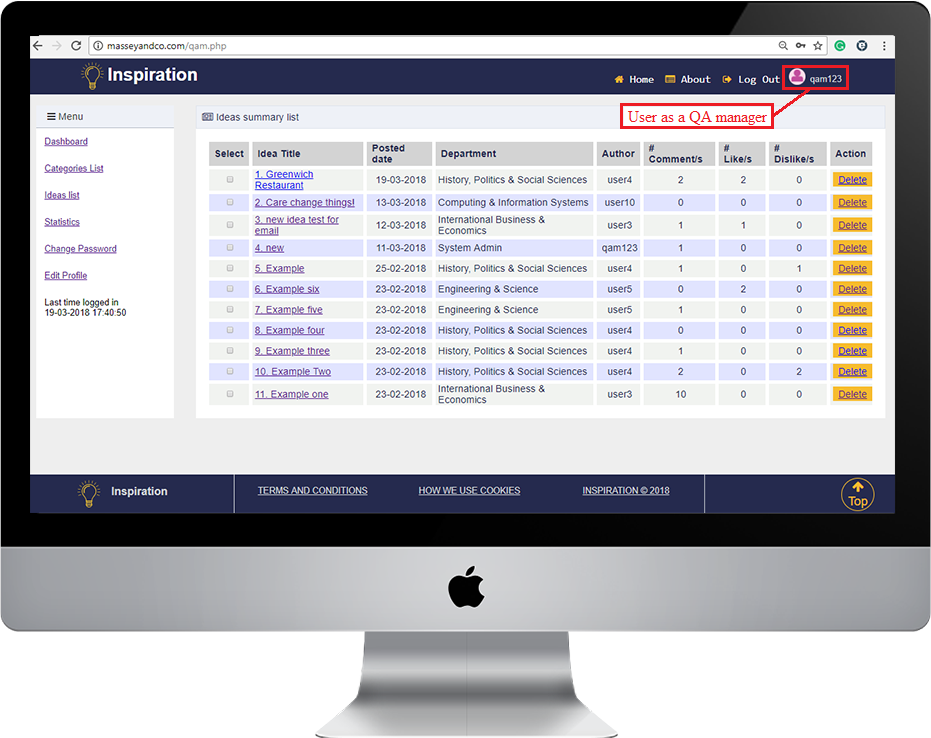


Figure 11 The landing page for the User as QA Manager (Dashboard for the QA Manager)

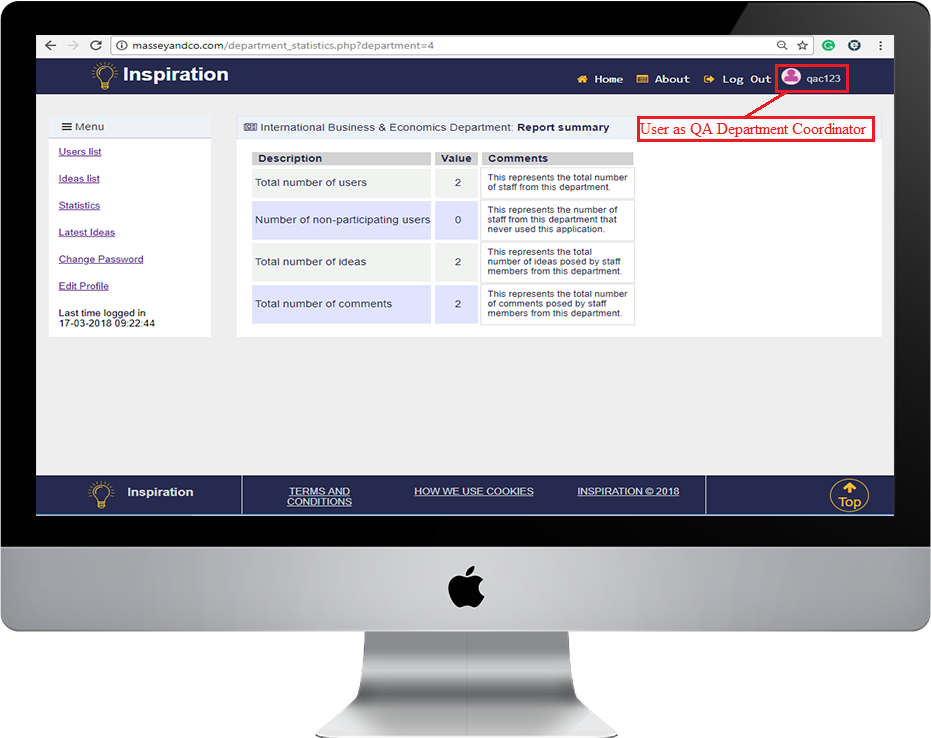


Figure 12 The landing page for the user logged in as a QA Department Coordinator

## 4.1 Application File structure

## 4.2 Code snippets

## 4.3 Screen shots of the application in operation

# Testing

Short description why testing

## 5.1 Testing Plan

### 5.1.1 Name of the item for test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | T-0-0-1 | | | | | |
| Item Name |  | | | | | |
| Purpose of testing |  | | | | | |
| Requested | Yes/No | | | | | |
| Test procedures |  | | | | | |
| Expectations |  | | | | | |
| Comments |  | | | | | |
| Test results | Pass | Yes/No | Fail | Yes/No | Not Tested | Yes/No |

### 5.1.2

## 5.2 Test log

# Evaluation

### 6.1 Initial assumptions

### 6.2 Product Strength

### 6.3 Product Weaknesses

### 6.4 Conclusion