Ewan Beaton (S1514665), Carlos Leal (S1828057), Scott Gallacher (S1918154) and Daniel Russel (S1707149)

  Team: Y

Web Platform Development 2 – Team Report

Contents

[Introduction 2](#_Toc39649489)

[Link Design 2](#_Toc39649490)

[Persistence Mechanisms 2](#_Toc39649491)

[Testing 4](#_Toc39649492)

[Summary of Testing 10](#_Toc39649493)

[Security Features 10](#_Toc39649494)

[Conclusion 11](#_Toc39649495)

# Introduction

The successful completion of the Web Platform Development 2 module required the production of an, interactive, web-based application. The requirements specification stated that a development team, of four members, must produce a web application that students can use to manage coursework assignments. The main features of the system will allow for the user to Create, Modify and Remove coursework assignments. A project entity will consist of several items of data, including; assignment title, relevant module, intended due date and a completion date. Additionally, the application will allow for the creation of milestones for each assignment. The milestones are intended to allow for the user to keep track of their progress with a project. The user will be able to view all incomplete projects along with their respective milestones as well as allowing for them to share projects with each other. There were several important technical aspects that were taken into consideration when undertaking this project, such as; Link Design, Persistence Mechanisms, Testing and Security.

# Link Design

When developing a web-based application, it is important to consider how the URLs should be structured within the it. Our team opted for a user centric approach to link design since the user is more likely to use links that they can understand. Using easily understandable terminology within a URL will also increase its trustworthiness to the user. The implementation of user-centric link designs in our application were achieved by; avoiding the use of underscores, avoiding the use of special characters, keeping them short and making them descriptive.

Namespaces are useful tools for expanding the existing URL structure within the application because they denote distinct content. This technique is particularly useful for websites that host larger quantities of content. In terms of the coursework scheduling application, our team only used one Namespace. The coursework scheduling application only requires for the use of one Namespace because of the limited scale of the system, it would be inefficient to introduce more. Namespacing is used within our development of this application so that the user can be delivered content that is specific to their account.

A powerful feature of any application is its ability to be easily bookmarked and shared among internet users, increasing its overall popularity. Taking this into consideration, it is important for developers to avoid implementing methods or using protocols that would impede the shareability of the application. When using HTTP requests, it is beneficial to use GET rather than POST. The main reason for this is that GET requests can be cached and bookmarked whereas POST requests cannot. However, POST requests are beneficial in certain situations, for example: when dealing with sensitive information. This is because POST requests are not persisted in any way after use and don’t have any restrictions on data length.

# Persistence Mechanisms

Nedb was used as means of supplying the application with a means of persisting user information. Nedb is a lightweight database management system that supports the storage of limited amounts of data. The main reason for using this system to store user information for our application is that it is easy to set up and provides the system with quick and easy access to database functionality. Nedb is useful because it uses the same API as MongoDB which means that it is easy to switch to MongoDB when the data to be stored is exceeding the practical limits of Nedb.

Data persistence was achieved by creating the below setup method in both the User and Project Models that defines a path for the program to create and access a flat file database for both Users and Projects, it then creates a new instance of the Nedb Database using the new defined file path and finally returns the database outside of the function so that other files such as the Controller “projectroutes.js” can access it by importing the class:

“const projectDao = require('../Model/project.js');”

function setup() {,

var path = require('path');

var appDir = path.dirname(require.main.filename) + "/app.project.db";

let dao = new ProjectDAO(appDir);

dao.init();

return dao;

}

Unless deleted, the flat file database and its contents will persist between runtime sessions so that users can access their accounts and projects without needing to Register every time they wish to use the system and even if the system is restarted.

As we are using Nedb are our database storage system we do not have a database schema because Nedb is not a relational database and as such a schema doesn’t seem necessary and hasn’t been created for the User and Project.

The Controllers act as the Data Access Layer of the application and work by importing their Model equivalents are receiving the results of their functions to then be used in the Controller’s GET and POST methods, below is an example of the projectRoutes.js Controller using the project.js Model’s “create” function in its POST method for adding projects:

projectController.post('/AddProject', ensureLoggedIn('/Login'), function (request, response) {

/\*// Check if the user is logged in (not working)

if (request.user == null) { response.redirect('/'); return; } \*/

if (!request.body.projectTitle || !request.body.modulename || !request.body.description ||

!request.body.dueDate || !request.body.completionDate) {

response.status(400).send("Please fill in the empty fields.");

return;

}

projectDao().create(request.body.projectTitle, request.body.modulename, request.body.description, request.body.isPrivate !== undefined, request.body.dueDate, request.body.completionDate);

response.redirect("/HomePage");

});

# Testing

Testing is an important phase of the development process, the main objective at this stage of the project is to identify and correct bugs in the code which could prevent the application from being able to carry out its intended function. It was necessary for the development team to employ several testing methods to ensure that the application functions as required. The testing methods used targeted various aspects of the application, for example; Logic, Performance and Security.

The first stage of the testing process is to carry out individual unit tests. These unit tests are designed to test the functionality of specific sections of code so that errors can be identified and evaluated. The unit tests for the coursework scheduling application were written and carried out in a white box environment so that the specific function operates as it is expected. This type of testing is carried out at every stage of the development cycle. By carrying out unit tests early in the development of the application we eliminated basic errors that could potentially cause major problems later in the project, while the overall body of code was small.

Integration testing was also used to assess the overall quality of the application. This testing phase involved combining individual modules of the application and testing them as a group. A top-down approach was adopted during this stage. This approach to integration testing involves the testing of the high-level components of the system and working down to the lower level components. The aim of the integration testing stage is to identify any potential inconsistencies between the modules of the system and take steps to rectify them.

When all modules had passed the integration test, they were subjected to the System Test. The aim of this test was to identify any further irregularities between the integrated modules that would affect the deployment of the final product. The system test is carried out on a fully integrated and potentially deployable artefact. Our team conducted this test by running the application in a controlled environment and carrying out all procedures that a typical user should be able to perform.

The Acceptance phase of the testing process involves both making sure that the application meets the criteria specified in the requirements as well as observing how users respond to the application. Due to the nature of this application, it would be deployed, and the public would be permitted to use it as they pleased. It is likely that some errors will appear at this stage that may have been overlooked during the development of the application.

In order to fully evaluate the quality of the software application that our team had produced, it was necessary to carry out Functional Testing. This kind of software testing will evaluate the finished, potentially deployable, product by comparing it with the requirements specification. This process is usually carried out in a black box style environment with the tester being unaware of the internal workings of the system, but a member of our group had to fill this role due to us having limited access to resources. The team member was required to input pre-determined values into the application, depending on the function that was being tested. This data was drawn from the categories; Normal, Extreme and Exceptional. The actual outputs were then compared with the expected outputs.

##### Functional Testing

**Login Account**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | Name – tester  Password – 12345678  Email – tester@gmail.com | Login successful  Home page displayed | Login successful  Home page displayed |
| Exceptional | Name – tester  Password – 1  Email – tester@gmail.com | Login failed | Login failed |

**Register Account**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | Name – tester  Password – 12345678  Email – tester@gmail.com | Account created  Login page displayed | Account created  Login page displayed |
| Exceptional | Name – test  Password – 1  Email – test2@gmail.com | Register failed  Password not valid | Register failed  Password not valid |

**Update Account**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | Name – tester  Email – [tester@gmail.com](mailto:tester@gmail.com)  Old Password – 12345678  New Password - 87654321 | Password changed successfully | Password changed successfully |
| Exceptional | Name – noteregistered  Email – [notregistered@gmail.com](mailto:tester@gmail.com)  Old Password – 12345678  New Password - 87654321 | Invalid | Invalid |
| Exceptional | Name – tester  Email – [tester@gmail.com](mailto:tester@gmail.com)  Old Password – notthepassword  New Password - 11111111 | Invalid | Invalid |

**Logout Account**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | Logout button clicked | User logged out | User logged out |

**Delete Account**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | Delete account clicked  Username - tester | User signed out and cannot sign in again | User is sent back to Landing Page, where they must log in or register. |

**Create Project**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | Project title – Test  Module Name – Test Module  Is Private – false  Start date – 05/05/2020  Completion deadline – 20/05/2020 | Project made with correct details | Project made with correct details |
| Extreme | Project created with due date the same day as start date | Project created successfully | Project created successfully |
| Exceptional | Project created with due date before start date | Error message displayed | Error message displayed |

**Update project**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | Update project clicked  Title, module and description changed | Title, module and description updated | Title, module and description updated |
| Normal | Update project clicked  Start and end date changed | Start and end date changed | Start and end date changed |
| Exceptional | Title, module and description removed | Asked to fill out fields | Asked to fill out fields |

**View Project**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | View project clicked | View project page displayed | View project page displayed |

**Create milestone**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | View project clicked  Create milestone clicked  Milestone name – Test Milestone  Description – This is a test  Start date – 10/05/2020  Due date 18/05/2020 | Milestone created | Milestone created |
| Exceptional | Due date set before start date | Error message | Error message |

**Create task**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | View project clicked  View milestone clicked  Create task clicked  Task name – Test task  Description – This is a test  Start date – 11/05/2020  Due date 17/05/2020 | Task displayed for milestone | Task displayed for milestone |
| Exceptional | Due date set before start date | Error message | Error message |

**Update task**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | Task name and description changed | Name and description updated | Name and description updated |
| Normal | Start and end date changed | Start and end date updated | Start and end date updated |
| Exceptional | Start and end date changed to invalid dates | Error message | Error message |

**Remove task**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | Remove task clicked  Confirmed | Task removed | Task removed |

**Update milestone**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | Name and description changed | Name and description updated | Name and description updated |
| Normal | Start and end date changed | Start and end date updated | Start and end date updated |
| Exceptional | End date set before start date | Error message | Error message |

**Share project**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | Copy share link  Log in as other account  Click view shared project  Paste share link | View project | All projects were viewable as soon as logging in to another account |

**Remove milestone**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | Remove milestone clicked | Milestone removed | Milestone removed |

**Remove project**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Input | Expected | Actual |
| Normal | Remove project clicked | Project removed | Project removed |

# Summary of Testing

The tests carried out on the application show that the main functionality works as intended, but some issues were identified. The main functionality related issue that affects the system is the failure of the ‘share project’ function to work. By triggering this function, all users of the application can view all projects that are stored in the database. This would suggest that there are issues with the assignment of privileges within the system, this issue should be investigated and rectified during future maintenance since it poses a significant security risk. Another issue with the application is that it crashes when the ‘delete project’ function is triggered by users who are running the application on certain computer systems. This is another issue that requires investigation and should be rectified during future maintenance of the system as it could impact the popularity of the application when it is deployed. It would also be recommended to implement some input validation during future developments of the system since the ‘create project’ function allows for a user to create a project with an earlier end date than the start date.

# Security Features

When developing any web-based application it is important to consider the security threats that it may encounter and devise a strategy for coping with them. A risk assessment of the potential security issues faced by the project scheduling application found that the most likely motive for an attack would be the acquisition of any user information that is stored on the system. There are a few ways that a knowledgeable and motivated individual could acquire this information, such as; through a session hijacking attack, cross site scripting attack, attacks and SQL Injection attacks. An attacker may also wish to prevent typical users of the application from being able to access it in order to impact the business, in this case they may use a DDoS attack.

A session hijacking attack involves the attacker exploiting a valid session key in order to gain access to restricted information. This kind of attack is carried out using packet sniffing software with the intent of retrieving the session ID of the victim and gaining access to their information as a result. An effective way of countering this style of attack is to ensure that packets of information are not transmitted in clear text format, they should be obfuscated using a hash function. Complex algorithms should be used to create session keys that cannot be easily guessed.

Cross-site scripting attacks are another style of attack that are commonly used to target web-based applications. This involves the attacker sending a specially designed link to the victim that contains malicious JavaScript code. When the victim clicks the link, the malicious code is executed in their browser. This could result in a session token being sent to the attacker. This type of attack will attempt to exploit vulnerabilities in the following; the application, server or plugin-systems that they the application relies on.

DDoS attacks pose a common threat that is faced by all applications that are hosted on a server and can be accessed via the internet. This type of cyber-attack involves the perpetrator attempting to limit the ability for the system to provide service to its users. This is the most effective kind of denial of service attack since the attack is carried out by many machines in different locations, making it difficult to simply block the source of the attack. Mitigation is the best strategy to deal with a DDoS attack. A DNS sinkhole could be implemented which would analyse incoming packets and filter out the bad ones, only allowing verified low risk packets to connect to the server.

Another common way in which hackers target web-based applications is through using SQL injection attacks. In order to carry out an SQL injection attack, the hacker will exploit user input fields in order to write and execute malicious SQL code. This can be prevented by ensuring that all user input fields require for strongly typed responses and by ensuring that all string literal escape characters are correctly filtered. Even though SQL attacks are extremely common, our application is protected against it because we are not using a relational database. Having said this, there is now the emerging threat of the ‘NoSQL’ attack which involves injecting malicious code into a system via the user input fields. The difference is that a NoSQL attack does not rely on SQL statements.

# Conclusion

In retrospect, the development of the coursework scheduling application was a success. Our team has faced many challenges during this project, all of which we have been able to weather. Our group had adopted an autonomous approach to carrying out tasks, this has been achieved using teamworking platforms like; Github, Trello and OneDrive. This approach allows for each member of the team to work on separate tasks simultaneously which reduces the workload and overall development time for the system. In addition to this, we used a combination of social media and VoIP services to maintain a high level of communication throughout all stages of the project. Recognising the importance of communication is a key factor in the success of any project and helps for the group to react quickly to unexpected changes. However, when undertaking a project of this nature in future it may be beneficial to conduct more research and implement some features of secure programming in order to add greater security for the users of the application.