

RAD Master Document

RAD DEVELOPMENT

The Imposters | Rapid App Development | 05/11/2020 - 10/12/2020

tABLE OF cONTENTS

[Sprint 1 1](#_Toc57885866)

[Meeting Minutes 2](#_Toc57885867)

[Meeting Minutes 3](#_Toc57885868)

[Agenda Items 3](#_Toc57885869)

[Source Control Snapshot and History 4](#_Toc57885870)

[Source Control Snapshot 5](#_Toc57885871)

[Source Control History 6](#_Toc57885872)

[Project Management Plan 10](#_Toc57885873)

[Project Management Plan Snapshot 11](#_Toc57885874)

[Software Development Testing Plan 12](#_Toc57885875)

[Introduction 13](#_Toc57885876)

[Scope 13](#_Toc57885877)

[In Scope 13](#_Toc57885878)

[Out of Scope 13](#_Toc57885879)

[Quality Objective 13](#_Toc57885880)

[Roles and Responsibilities 14](#_Toc57885881)

[Test Methodology 15](#_Toc57885882)

[Overview 15](#_Toc57885883)

[Test Levels 15](#_Toc57885884)

[Bug Triage 15](#_Toc57885885)

[Suspension Criteria & Resumption Requirements 16](#_Toc57885886)

[Test Completeness 16](#_Toc57885887)

[Test Deliverables 17](#_Toc57885888)

[Resource & Environment Needs 17](#_Toc57885889)

[Testing Tools 17](#_Toc57885890)

[Test Environment 17](#_Toc57885891)

[Hardware Environment 17](#_Toc57885892)

[Required Software 17](#_Toc57885893)

[Terms / Acronyms 18](#_Toc57885894)

[Analysis Report 19](#_Toc57885895)

[CITE Business Rules 20](#_Toc57885896)

[CITE Managed Services QA 21](#_Toc57885897)

[A.E. Development Requirements 21](#_Toc57885898)

[Multi-Platform Report 22](#_Toc57885899)

[Responsive vs Adaptive 23](#_Toc57885900)

[Our Decision 24](#_Toc57885901)

[Code Testing 25](#_Toc57885902)

[PHP\_CodeSniffer Snapshots 26](#_Toc57885903)

[Sprint 2 28](#_Toc57885904)

[Meeting Minutes 29](#_Toc57885905)

[Meeting Minutes 30](#_Toc57885906)

[Agenda Items 30](#_Toc57885907)

[Source Control Snapshot and History 31](#_Toc57885908)

[Source Control Snapshot 32](#_Toc57885909)

[Source Control History 33](#_Toc57885910)

[Project Management Plan 37](#_Toc57885911)

[Project Management Plan Snapshot 38](#_Toc57885912)

[Software Review Plan 39](#_Toc57885913)

[Review Checklist 40](#_Toc57885914)

[Performance Report 41](#_Toc57885915)

[Website Performance 42](#_Toc57885916)

[PHP CodeSniffer 47](#_Toc57885917)

[Software Development Testing Plan 50](#_Toc57885918)

[Test Table 51](#_Toc57885919)

[Sprint 3 54](#_Toc57885920)

[Meeting Minutes 55](#_Toc57885921)

[Meeting Minutes 56](#_Toc57885922)

[Agenda Items 56](#_Toc57885923)

[Source Control Snapshot and History 57](#_Toc57885924)

[Source Control Snapshot 58](#_Toc57885925)

[Source Control History 59](#_Toc57885926)

[Project Management Plan 63](#_Toc57885927)

[Project Management Plan Snapshot 64](#_Toc57885928)

[Optimisation Report 65](#_Toc57885929)

[Performance Optimisation 66](#_Toc57885930)

[Client-side and Server-side Optimisation 66](#_Toc57885931)

[Client-side Performance Optimisation 67](#_Toc57885932)

[Software Development Testing Plan 68](#_Toc57885933)

[Updated Test Table 69](#_Toc57885934)

[Exit Criteria 71](#_Toc57885935)

[Conclusion 71](#_Toc57885936)

[Project Handover 72](#_Toc57885937)

[Meeting Minutes 73](#_Toc57885938)

[Meeting Minutes 74](#_Toc57885939)

[Agenda Items 74](#_Toc57885940)

[Source Control Snapshot and History 75](#_Toc57885941)

[Source Control Snapshot 76](#_Toc57885942)

[Source Control History 77](#_Toc57885943)

[Project Management Plan 81](#_Toc57885944)

[Project Management Plan Snapshot 82](#_Toc57885945)

[Software Review Report 83](#_Toc57885946)

[Development Quality 84](#_Toc57885947)

[Code Testing 84](#_Toc57885948)

[Future Modification and Refinements 85](#_Toc57885949)

[Mapping 86](#_Toc57885950)

[Software Development Testing Plan 88](#_Toc57885951)

[Introduction 89](#_Toc57885952)

[Scope 89](#_Toc57885953)

[In Scope 89](#_Toc57885954)

[Out of Scope 89](#_Toc57885955)

[Quality Objective 89](#_Toc57885956)

[Roles and Responsibilities 90](#_Toc57885957)

[Test Methodology 91](#_Toc57885958)

[Overview 91](#_Toc57885959)

[Test Levels 91](#_Toc57885960)

[Bug Triage 91](#_Toc57885961)

[Suspension Criteria & Resumption Requirements 92](#_Toc57885962)

[Test Completeness 92](#_Toc57885963)

[Test Deliverables 93](#_Toc57885964)

[Resource & Environment Needs 94](#_Toc57885965)

[Testing Tools 94](#_Toc57885966)

[Test Environment 94](#_Toc57885967)

[Hardware Environment 94](#_Toc57885968)

[Required Software 94](#_Toc57885969)

[Test Table 95](#_Toc57885970)

[Exit Criteria 100](#_Toc57885971)

[Exit Criteria 100](#_Toc57885972)

[Conclusion 100](#_Toc57885973)

[Terms / Acronyms 101](#_Toc57885974)

Sprint 1

Team Name : The Imposters

Scrum Master (#1) : Lisa Mckenna

Team Member (#2) : Swen Lee

Team Member (#3) : Daniel Ewen

Meeting Minutes

Meeting Minutes for Sprint One

Location : Murdoch T101 Office

Date : 5th November 2020

Time : 9:00 a.m.

# Meeting Minutes

Items discussed during team meeting in sprint one

## Agenda Items

1. Discuss design options (adaptive or responsive)
2. Discuss client requirements
3. Allocate tasks:
   * Develop master document
   * Set up GitHub repository
   * Develop analysis report
   * Develop project management plan
   * Develop multi-platform report
   * Develop software testing plan
   * Update website according to client requirements
   * Finalise documentation (update master document)

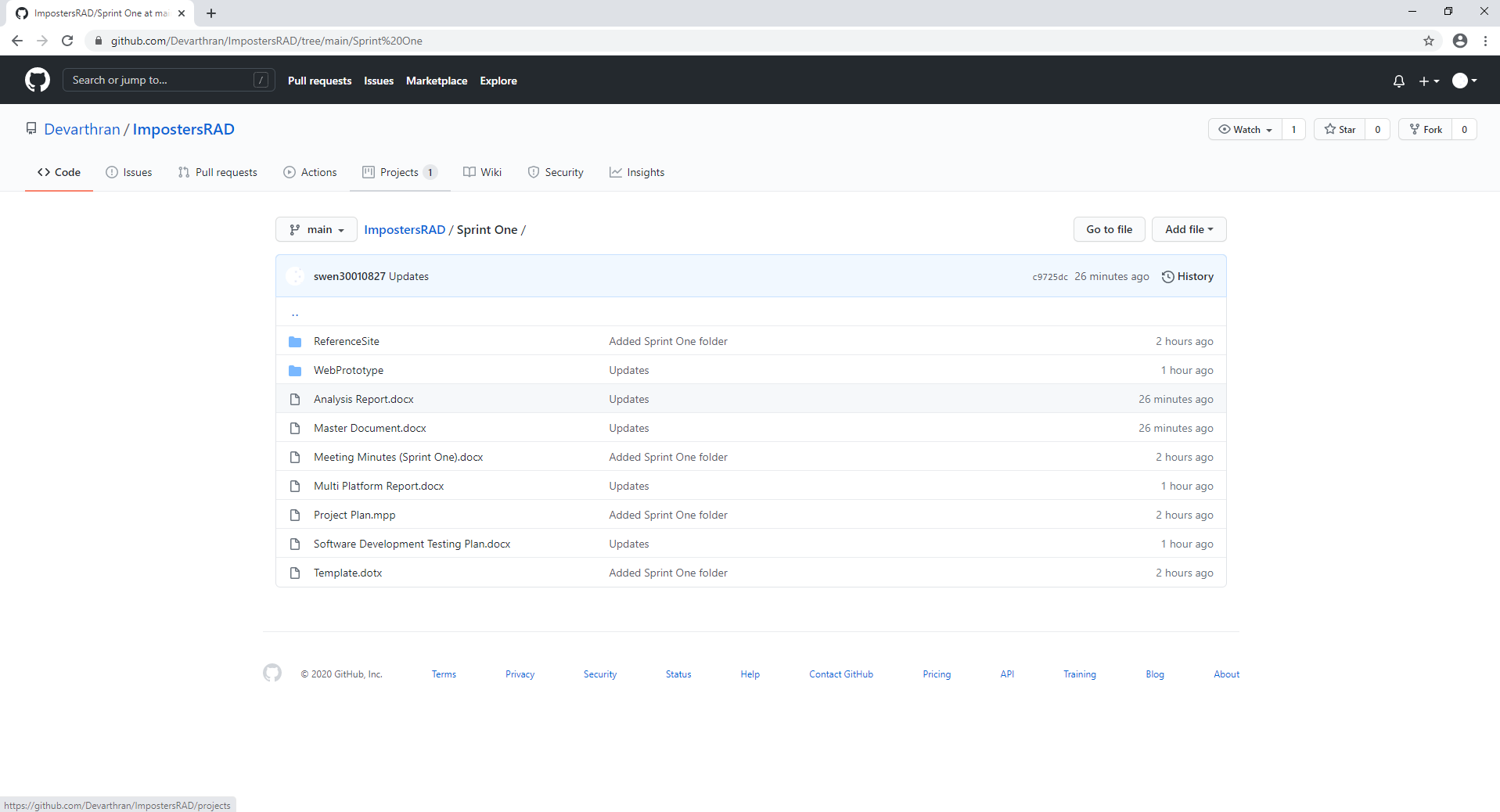
|  |  |  |  |
| --- | --- | --- | --- |
| Action Items | Owner(s) | Deadline | Status |
| Develop Master Document | Swen | 11/11/2020 | In Progress |
| Set up GitHub Repository | Daniel | 05/11/2020 | Complete |
| Develop Analysis Report | Swen | 11/11/2020 | In Progress |
| Develop Project Management Plan | Team | 05/11/2020 | Complete |
| Develop Multi-platform Report | Lisa | 11/11/2020 | In Progress |
| Develop Software Testing Plan | Daniel | 11/11/2020 | In Progress |
| Update website | Daniel | 11/11/2020 | In Progress |
| Finalise Documentation | Swen | 11/11/2020 | Pending |

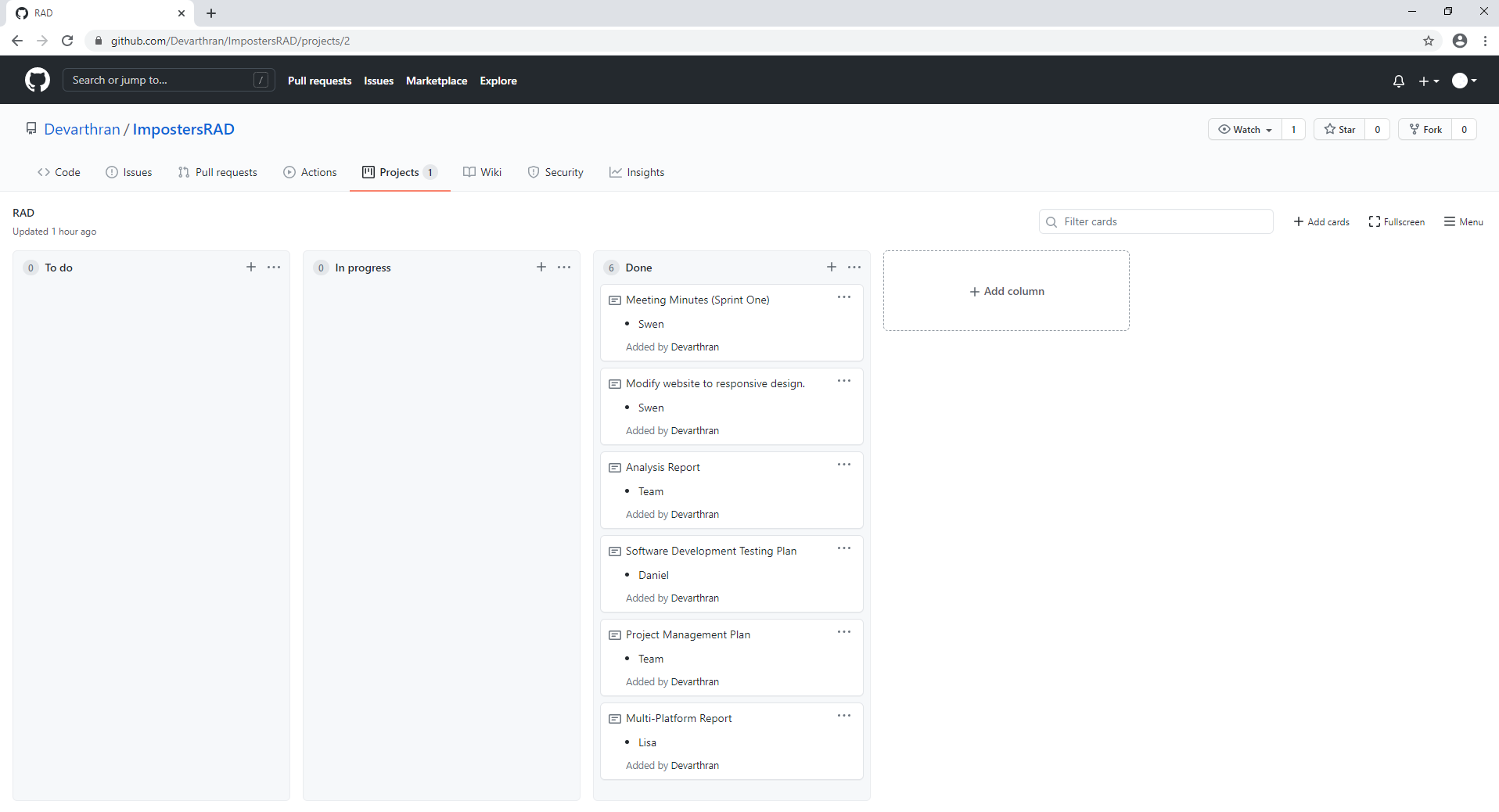
Source Control Snapshot and History

Contains snapshot for github as well as the progress of the work

# Source Control Snapshot

Below is the snapshot for of our GitHub repository. Click [here](https://github.com/Devarthran/ImpostersRAD) to access it.

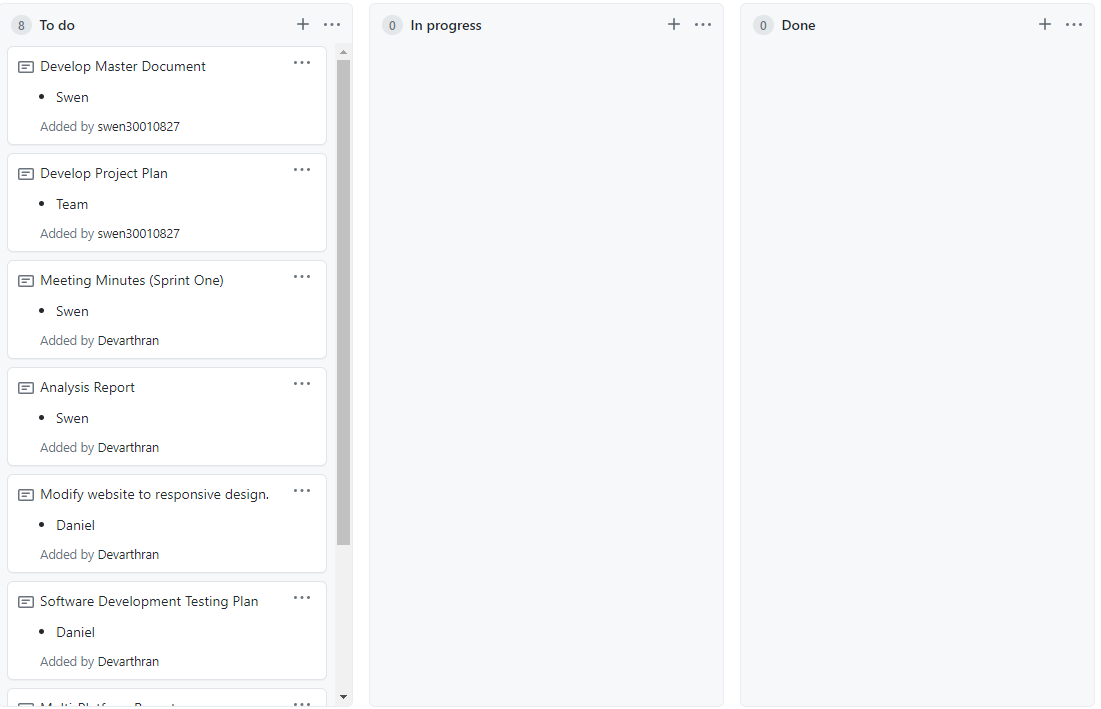
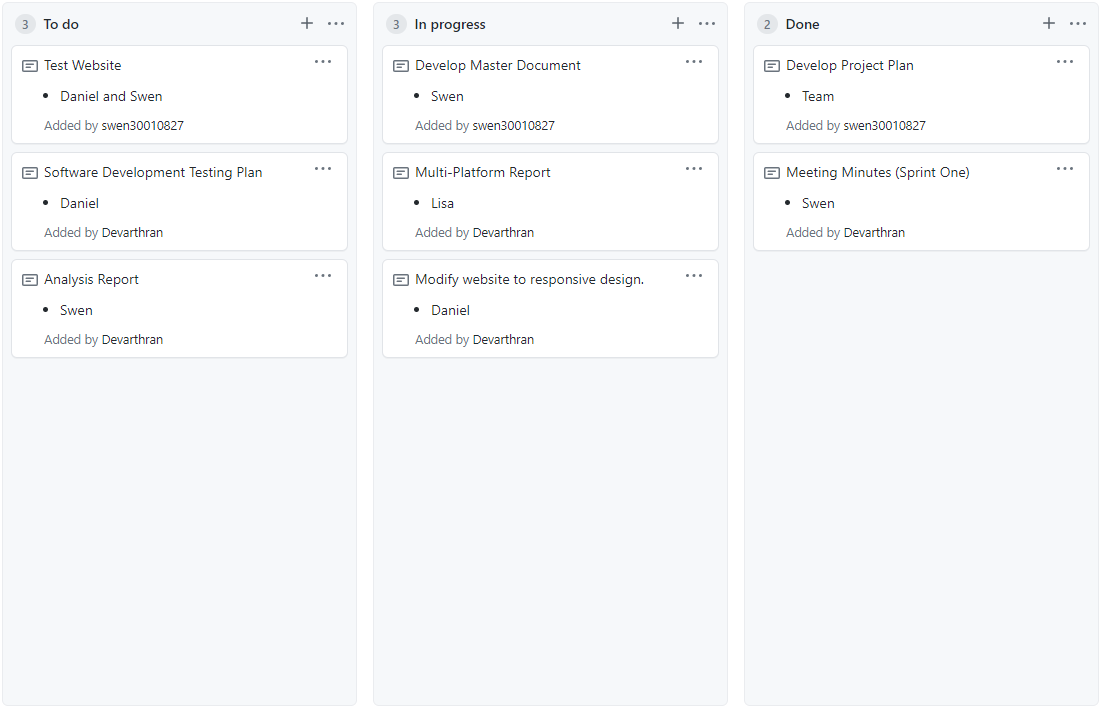




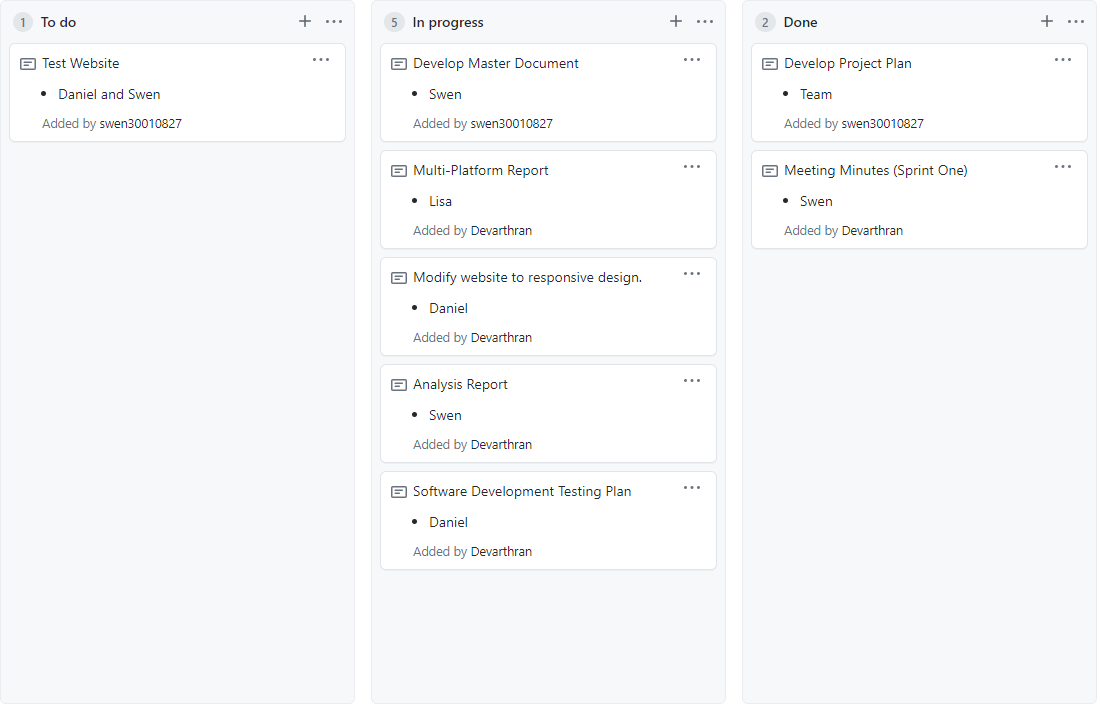
# Source Control History

|  |  |
| --- | --- |
| Date | Evidence |
| 05/11/2020 (Thursday) | Ref 1 |
| 06/11/2020 (Friday) | Ref 2 |
| 09/11/2020 (Monday) | Ref 3 |
| 10/11/2020 (Tuesday) | Ref 4 |
| 11/11/2020 (Wednesday) | Ref 5 |

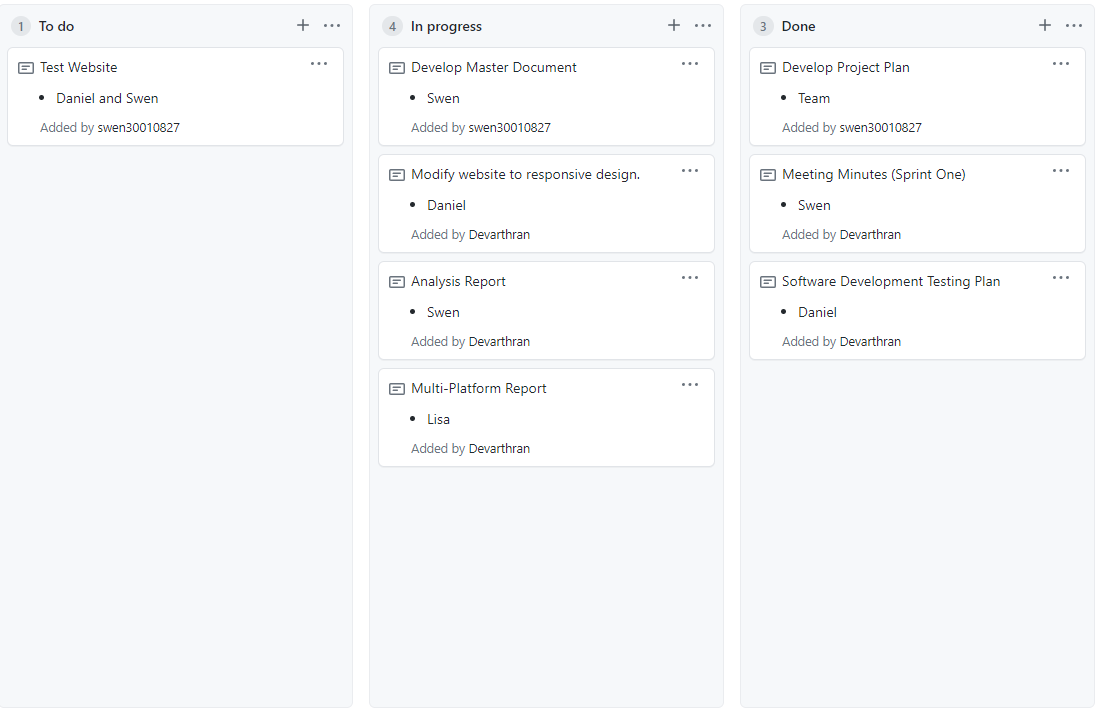
Ref 1

1. 
2. 

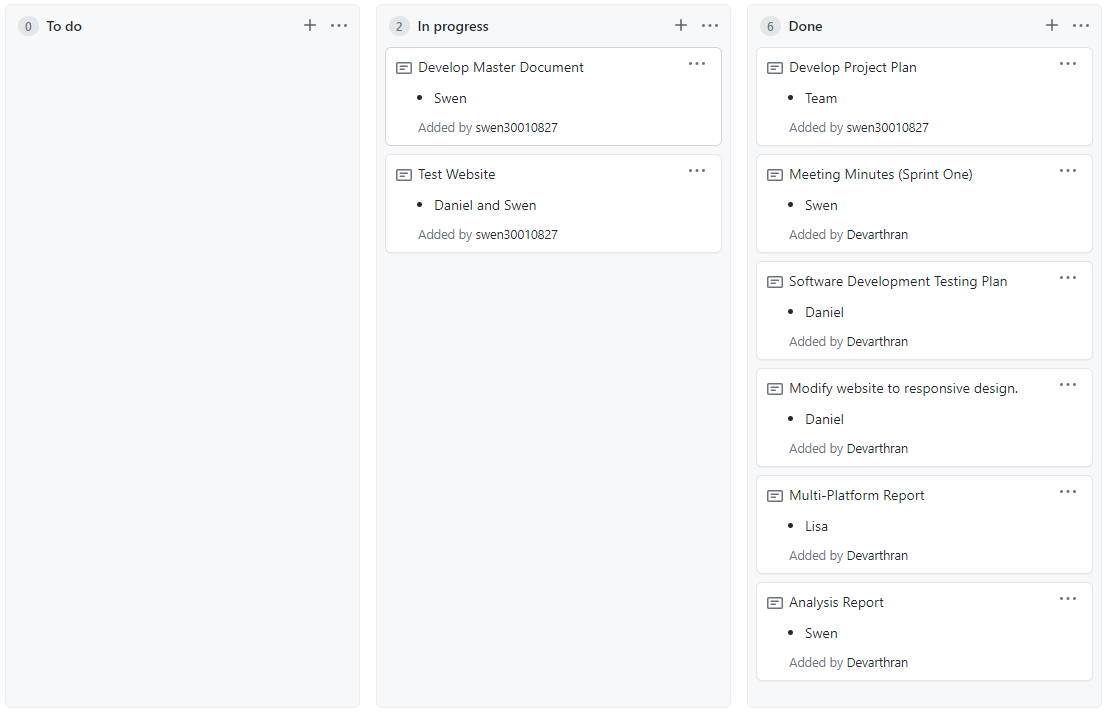
Ref 2

1. 

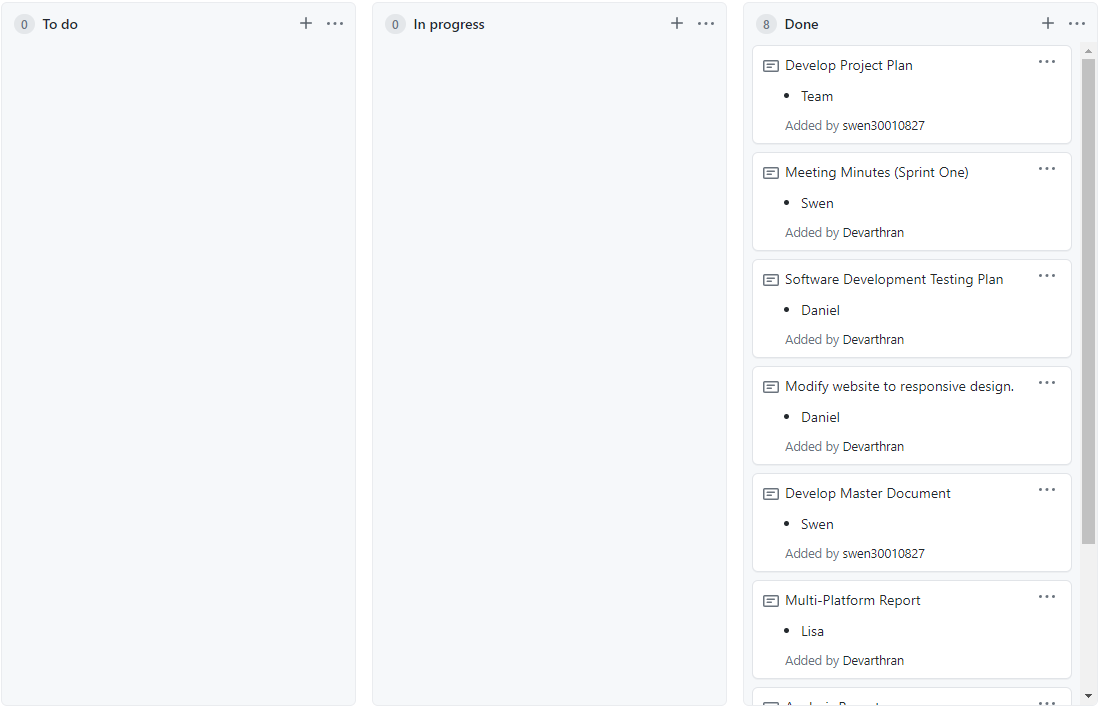
Ref 3

1. 

Ref 4

1. 

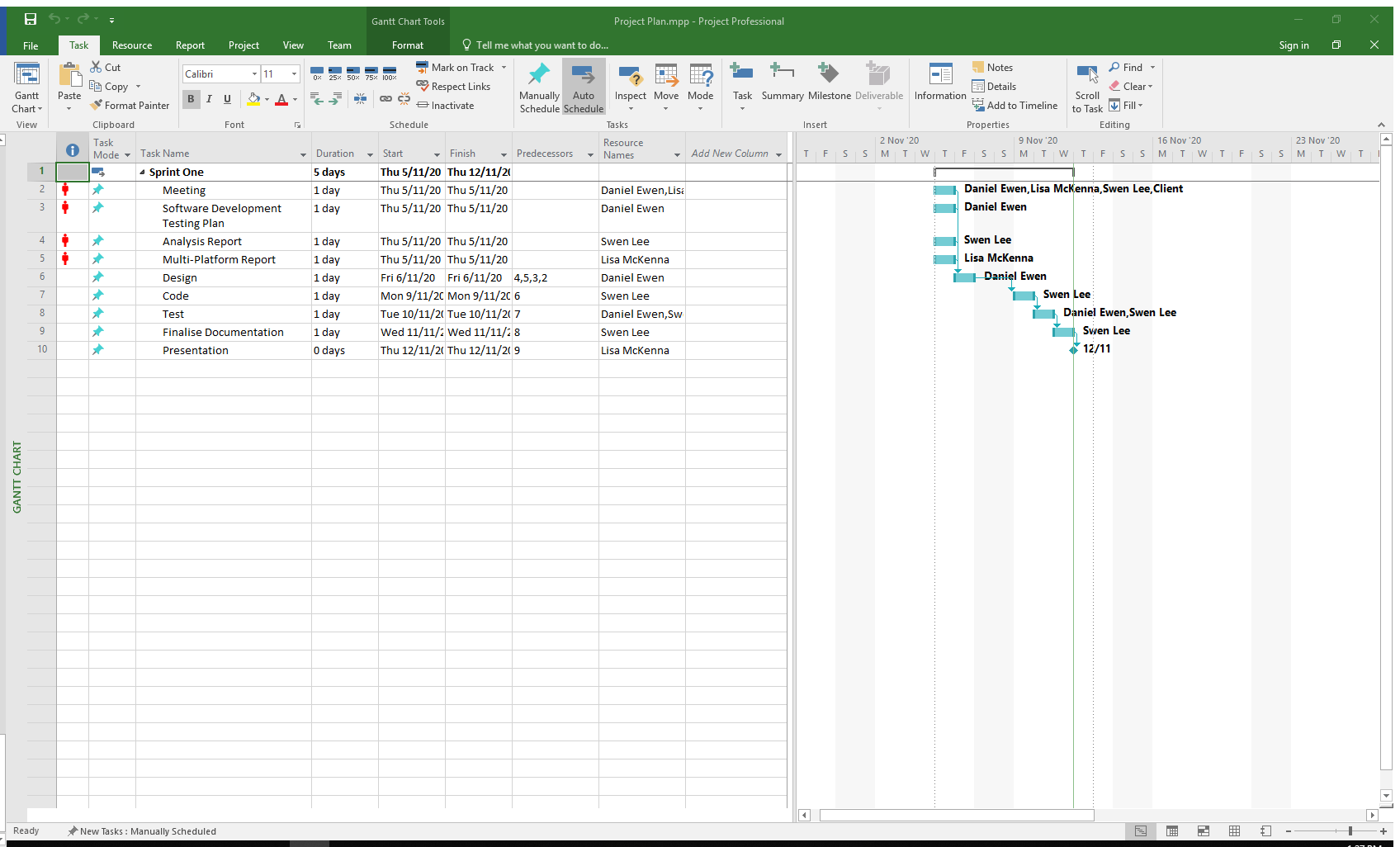
Ref 5

1. 

Project Management Plan

Contains the project management plan for sprint one

# Project Management Plan Snapshot



Software Development Testing Plan

Develop a test plan for the project

CHANGELOG

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Date of change | Change by | Outline |
| 1.0 | 5/11/2020 | Daniel Ewen | Test Plan Creation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Scope

### In Scope

Features to be tested:

* Application design is responsive
* User Interface
* Interface responds in real-time
* Database migrates correctly to the new design.

### Out of Scope

Features that won’t be tested:

* Movie database

## Quality Objective

Our team’s objective is to deliver a working prototype to the client with the following qualities:

* AUT[\*](#_Terms_/_Acronyms) must conform to the client’s requirements for each sprint.
* AUT must meet the client’s quality standards.
* Bugs/defects are tested and fixed before the application is deployed for the client.
* The AUT meets coding and commenting standards defined by each coding language standard.

## Roles and Responsibilities

|  |  |
| --- | --- |
| Role | Description |
| Developers | Research, design, implement and manage software programs |
| Test Manager | Organise and control the testing process to deliver a high-quality software |
| QA analyst | Test software on other computers to ensure they are functioning accurately |
| Business Analyst | Connect IT and business using data analytics and determine client requirements |
| Bug Triage | Evaluate, prioritise and assign resolution defects |

# Test Methodology

## Overview

Our team will utilize the RAD[\*](#_Terms_/_Acronyms) test methodology throughout this project. Each sprint will seek to develop, test, and deliver a fully functioning prototype to the specification of the client.

## Test Levels

Our testing team will implement the following test types for this project:

* Exploratory Testing
* Functional Testing
* Accessibility Testing
* Compatibility Testing
* Integration Testing
* System testing
* User Acceptance Testing (UAT)

## Bug Triage

Our process for bug/defect triage is as follows:

* **Bug discovered:** Bug report is added to the bug list. For this our team will use the issue tracker built into GitHub, detailing the bug.
* **Investigate:** Developers read the report and try to replicate the bug. If replication occurs, the bug can then be prioritized.
* **Bug added to backlog:** Bug is awaiting resolution during sprint.
* **Resolved:** Bug is fixed during development and the issue is closed.

## Suspension Criteria & Resumption Requirements

During testing, should any test return more than a 30% fail the following should occur:

* Testing ceases
* Test case / module being tested investigated for design flaws
* If bugs are discovered, add them to the triage queue.
* Once the bug has resolved in the triage queue, run the test again.

## Test Completeness

Testing will be complete when:

* The AUT has been deemed to have 100% test coverage,
* All designed test cases return with a pass,
* All current bugs/issues have been resolved,
* The client is satisfied with the current condition of the AUT and has signed off on its deployment.

# Test Deliverables

|  |
| --- |
| Deliverables |
| Test Plan |
| Test Cases with validation |
| Requirements |
| Analysis Report |
| Bug Reports |
| Client Sign-off |

# Resource & Environment Needs

## Testing Tools

* **Chromium developer tools:** Exploratory testing and during development.
* **PHP\_CodeSniffer:** Checks the PHP code for syntax and standards errors.

## Test Environment

### Hardware Environment

A modern development workstation with the following minimum specifications:

|  |
| --- |
| Part |
| Intel CPU 4-cores or more with hyperthreading |
| 16GB DDR4 RAM 2600MHz |
|  |

### Required Software

|  |  |
| --- | --- |
| Software | |
| Xampp | Local webserver and database hosting |
| Visual Studio Code | Code and application development |
| GitHub Desktop | Source Control |

# Terms / Acronyms

|  |  |
| --- | --- |
| Term / Acronym | Definition |
| RAD | Rapid Application Development |
| AUT | Application Under Test |

Analysis Report

describes CITEMS Rules, Quality Assurance Practices and Acme’s Development Requirements

# CITE Business Rules

Describes the CITE Business Rules for Software Development

At CITEMS we adhere to our business rules which consist of:

* Prioritising client interests
* Operate in an ethical manner in regards to the regulations
* Uphold accountability during software development
* Always strive to achieve a better solution to a client’s issues
* Prioritise firm and client interests
* Be fair competitors and not degrade other firms
* Maintain high ethical standards
* Abide by coding policies and standards

# CITE Managed Services QA

CITE Managed Services (CITEMS) Quality Assurance Practices

According to the [CITEMS](http://www.citems.com.au/?page_id=84) website, our organisation has launched processes which evaluate the project’s performance as well as ensure that quality standards are being followed and that the deliverables meet the client’s requirements.

Some of our quality assurance practices include:

* Conduct full-cycle quality assurance testing
* Document and code reviews
* Defect tracking
* Configuration management
* Process monitoring
* Risk management

CITEMS conducts quality assurance throughout the development lifecycle with our QA team members engaged in every stage. A standard QA lifecycle at CITEMS will involve the following stages:

1. Initiation and Planning – Project specification analysis, test plan explanation and team assignment
2. First review – Early testing of first development deliverables, refining test plan as well as test items (if required)
3. Iteration audits – Ongoing test of intermediate iteration builds
4. Final verification and validation – Final product testing to guarantee bespoke quality and readiness for deployment

# A.E. Development Requirements

Acme Entertainment Pty Ltd Development Requirements

* Acme Entertainment have commissioned a prototype movie database, however, this application is required to be updated to meet the following requirements:
* Able to use across all major digital platforms

Multi-Platform Report

Describes the difference between adaptive and responsive design

# Responsive vs Adaptive

Responsive sites and adaptive sites are the same in that they both change appearance based on the browser environment they are being viewed on.

Responsive websites respond to the bowser no matter what the browser width may be, the site adjusts its layout (and perhaps functionality) in a way that is optimised to the screen.

Adaptive websites are only concerned about the browser being a specific width, at which point it adapts the layout.

The simplest way to view this is the difference between smooth and snap design. Responsive design is smooth because the layout fluidly adjusts regardless of what device it is viewed on. Adaptive design, on the other hand, snaps into place because the page is serving something different because of the browser or device it is viewed on.

Here’s an image comparing responsive and adaptive designs from [Medium](https://popart-studio.medium.com/fluid-vs-adaptive-vs-responsive-design-62de51e036bd).



In the example above the responsive picture flows with the environment whereas the adaptive snaps into place at a defined environment.

Responsive design is both fluid and flexible which is a means of becoming device agnostic in the sense that it seeks to create an optimised experience for any screen.

Essentially, we challenge ourselves to create sites that shift context according to how a site is being consumed on any given occasion.

Compare that with an adaptive way of thinking, which is neither fluid nor flexible, but looks for specific points at which to adapt. While it might be difficult to adapt a website on all the various devices being used today.

Both responsive and adaptive designs are similar in the way that they are methods for dealing with the reality that websites are often viewed on different devices in different contexts. They just go about this in different ways.

(Graham, 2015)

# Our Decision

Our team’s recommendation is the responsive design, where we have produced a prototype application. As technology is ever changing and moving at a rapid pace, the responsive prototype is delivering a design that is both fluid, flexible and able to adjust its layout and perhaps its functionality in a way that it is optimised to the screen, there for, it can be view on current and future devices. In turn this will future proof the business where it is able grow exponentially

Code Testing

use php\_codesniffer to test and fix errors on the website

# PHP\_CodeSniffer Snapshots

index.php

1. Text

   Description automatically generated
2. Graphical user interface

   Description automatically generated

search.php

1. Text

   Description automatically generated
2. A picture containing timeline

   Description automatically generated

genre\_scr.php

1. Text

   Description automatically generated
2. A picture containing text

   Description automatically generated

rating\_scr.php

1. Text

   Description automatically generated
2. Graphical user interface, text

   Description automatically generated

top10.php

1. Text

   Description automatically generated
2. Graphical user interface, text

   Description automatically generated

Sprint 2

Team Name : The Imposters

Scrum Master (#1) : Swen Lee

Team Member (#2) : Daniel Ewen

Team Member (#3) : Lisa Mckenna

Meeting Minutes

Meeting Minutes for Sprint Two

Location : Murdoch T101 Office

Date : 12th November 2020

Time : 12:00 p.m.

# Meeting Minutes

Items discussed during team meeting in sprint two

## Agenda Items

1. Discuss software review plan
2. Discuss client requirements
3. Allocate tasks:
   * Update GitHub repository
   * Update project management plan
   * Develop performance report
   * Develop software review plan
   * Update software testing plan
   * Update website according to client requirements
   * Finalise documentation (update master document)

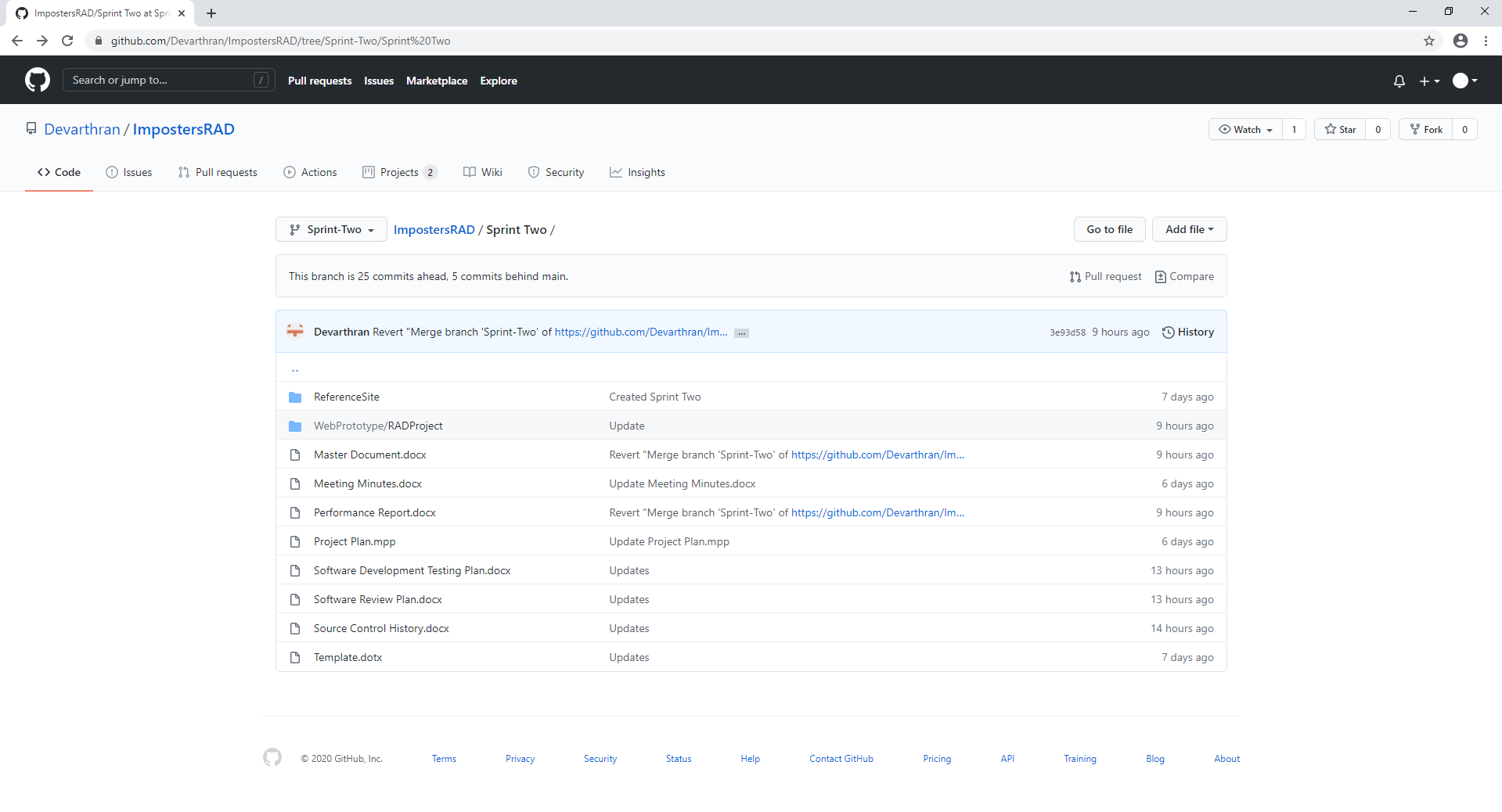
|  |  |  |  |
| --- | --- | --- | --- |
| Action Items | Owner(s) | Deadline | Status |
| Update GitHub repository | Daniel | 12/11/2020 | Complete |
| Update Project Plan | Team | 12/11/2020 | Complete |
| Performance Report | Swen | 18/11/2020 | In Progress |
| Software Review Plan | Lisa | 18/11/2020 | In Progress |
| Update Software Testing Plan | Daniel | 18/11/2020 | In Progress |
| Update Website | Daniel | 18/11/2020 | In Progress |
| Finalise Documentation | Swen | 18/11/2020 | Pending |

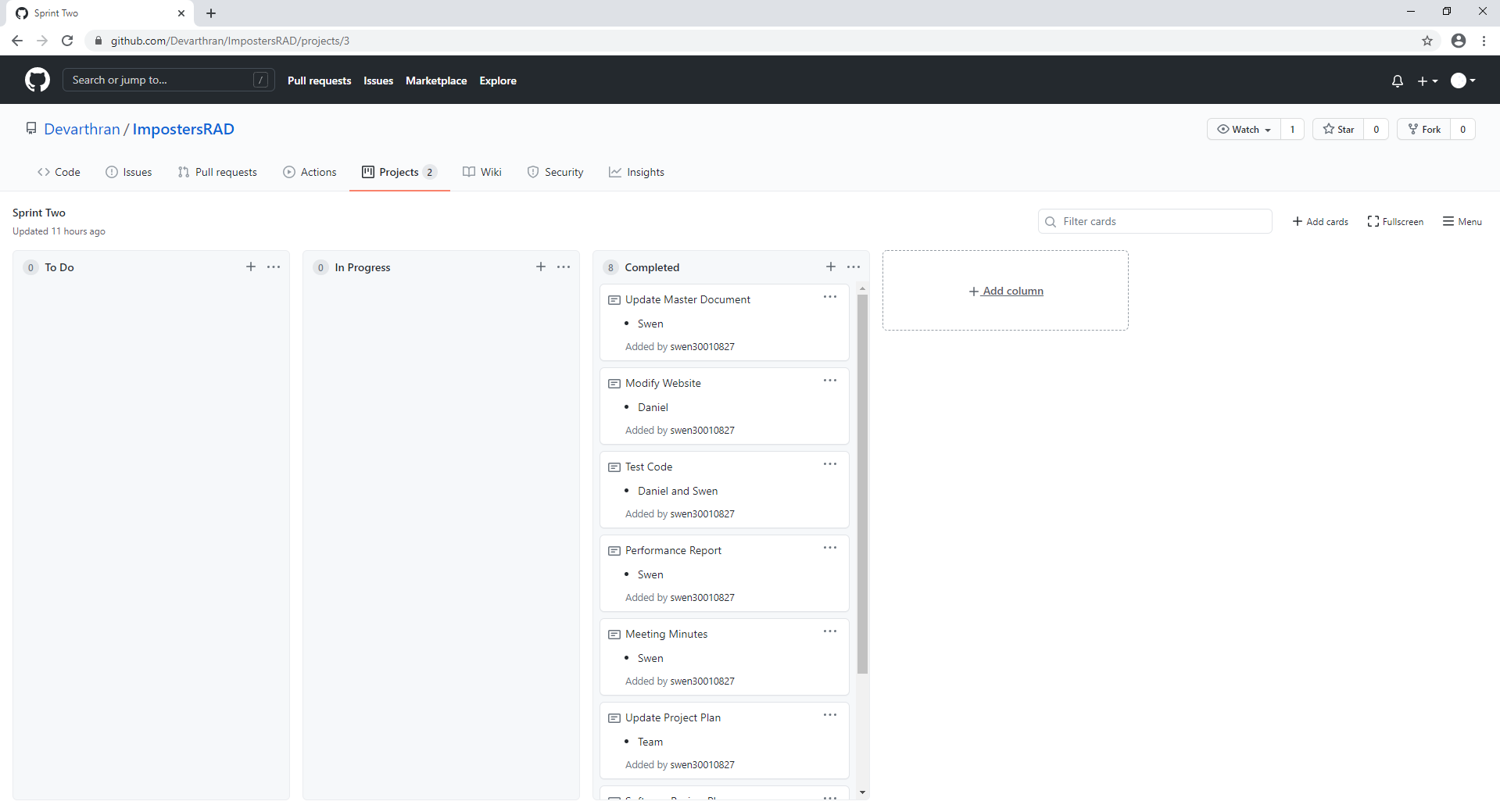
Source Control Snapshot and History

Contains snapshot for GitHub as well as the progress of the work

# Source Control Snapshot

Below is the snapshot for out GitHub repository. Click [here](https://github.com/Devarthran/ImpostersRAD/tree/main/Sprint%20Two) to access it.





# Source Control History

|  |  |
| --- | --- |
| Date | Evidence |
| 12/11/2020 (Thursday) | Ref 1 |
| 13/11/2020 (Friday) | Ref 2 |
| 16/11/2020 (Monday) | Ref 3 |
| 17/11/2020 (Tuesday) | Ref 4 |
| 18/11/2020 (Wednesday) | Ref 5 |

Ref 1

1. Graphical user interface, application, Teams

   Description automatically generated
2. Graphical user interface, application

   Description automatically generated

Ref 2

1. Graphical user interface, application

   Description automatically generated

Ref 3

1. Graphical user interface, application

   Description automatically generated

Ref 4

1. Graphical user interface, application

   Description automatically generated

Ref 5

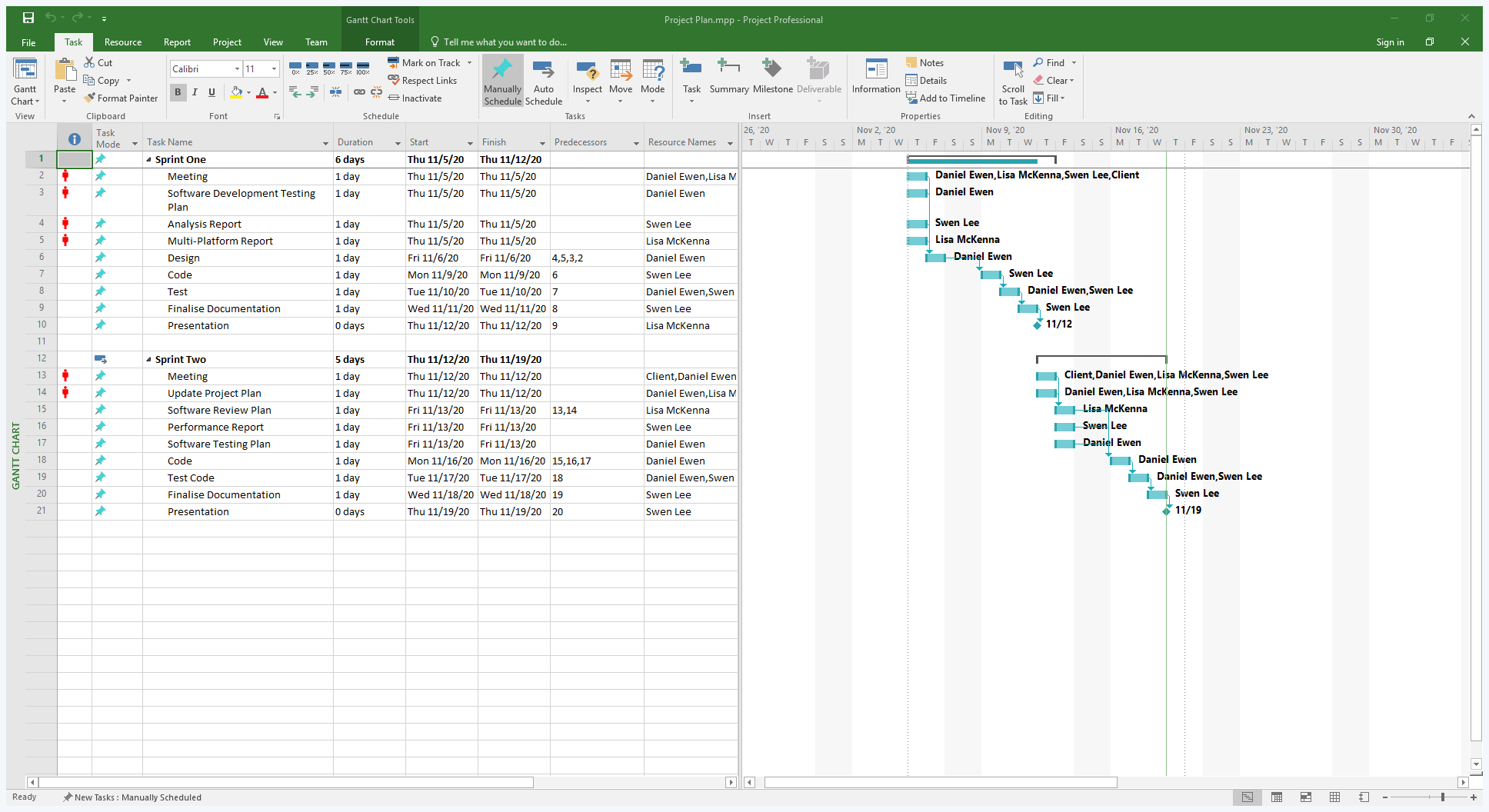
1. Graphical user interface, application, Word

   Description automatically generated

Project Management Plan

Contains the project management plan for sprint Two

# Project Management Plan Snapshot



Software Review Plan

Development and Design

# Review Checklist

|  |  |  |
| --- | --- | --- |
| ID | Description | Status |
| 1 | Website has a front end |  |
| 2 | Website has a search page |  |
| 3 | Website searches according to conditions selected |  |
| 4 | User is able to sign up to subscribe for newsletter or notifications |  |
| 5 | Website has a profile page for users to sign in and change their subscription |  |
| 6 | User is able to unsubscribe as they please |  |
| 7 | User is able to rate the movies |  |
| 8 | Website displays a graph with the top 10 rated movies |  |
| 9 | Graph refreshes automatically |  |
| 10 | Website has a login page for administrators |  |
| 11 | Administrators are able to add, edit and delete users |  |
| 12 | Administrator passwords are secured |  |
| 13 | Administrator passwords pass a minimum complexity test |  |

Performance Report

Demonstrates the performance of the website has been measured

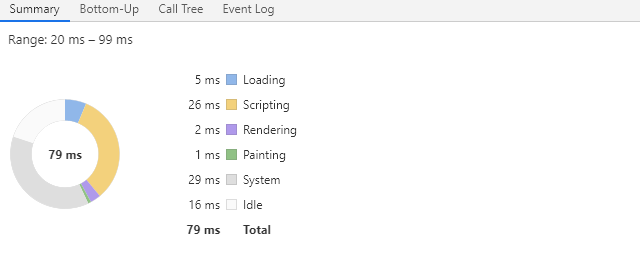
# Website Performance

The website’s performance was tested using the performance tool in Google Chrome’s Developer tools. The Chrome performance monitor records the website’s performance metrics while the page is running or reloading. The user is able to review and interact with snapshots of recordings. After a recording is taken, a graph is displayed with the summary of the website’s performance.

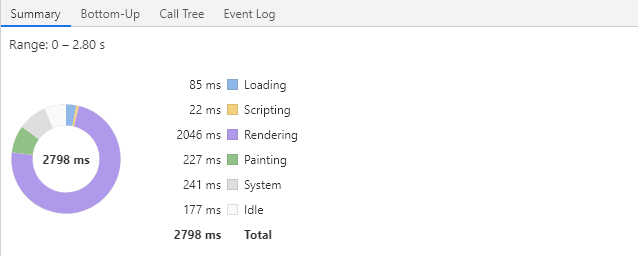
(BitDegree, 2019)

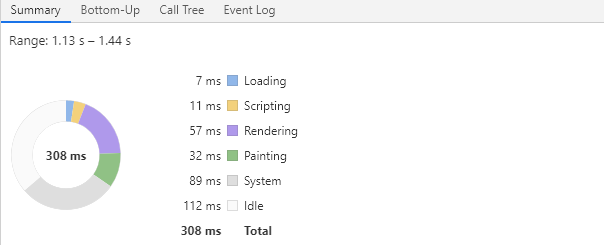
Home page (index.php)

Running the home page takes a relatively shorter period of time as it does not contain much code, it just contains the design and a few words for now.

1. 

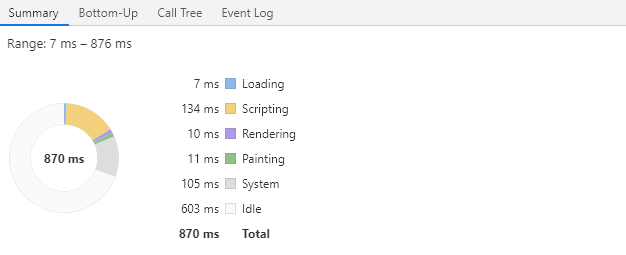
Search page (search.php)

1. According to the graph, rendering the search page takes up most of the time as the website is grabbing all the data from the database in one go and rendering it on the page.
2. When searching, system uses most of the time as it is searching in the database for the information requested



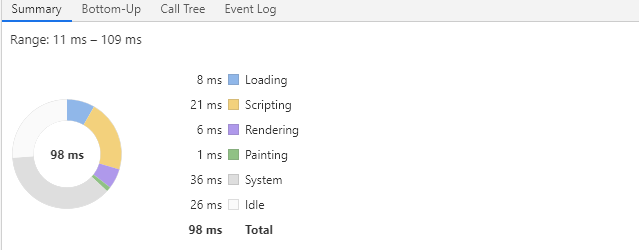
Top 10 Movies page (top10.php)

Scripting takes up most of the time on the top 10 movies page as jQuery and JavaScript is used to display the graph and execution of them takes up time.

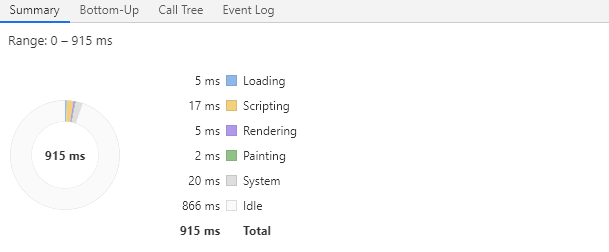
1. 

Sign Up page (signup.php)

1. Signing up to the page requires the system to load the data to the database, hence system takes up most of the time here.

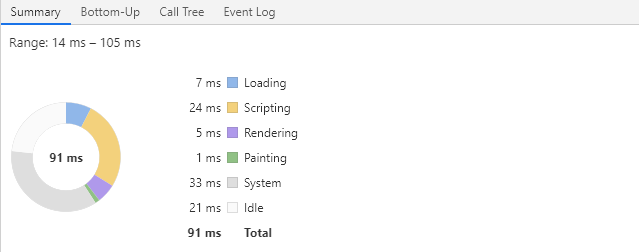


1. Signing up to the page requires the system to save the information to the database, hence, system takes up most of the time

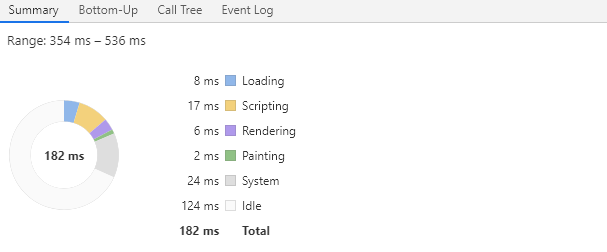


Login page (login.php)

1. Loading the login page doesn’t take up much time as it just contains a few text fields and buttons

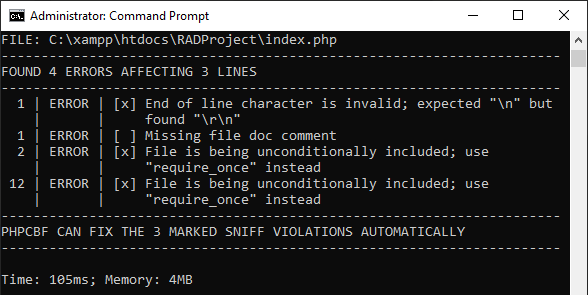
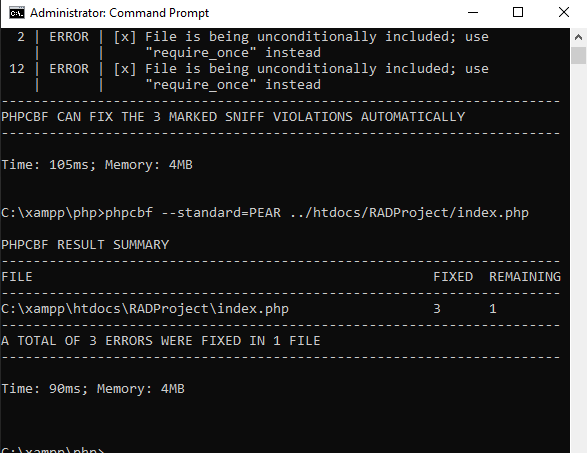


1. When logging in, the system takes up most of the time as it is required to load the database to ensure the credentials inputted are accurate

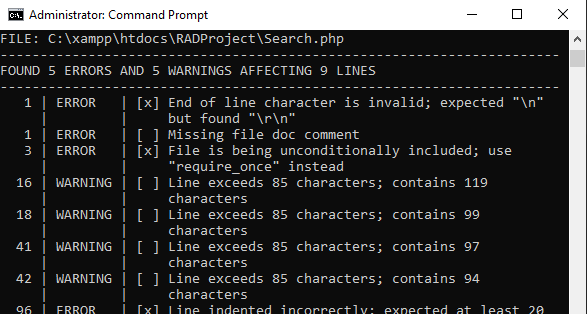
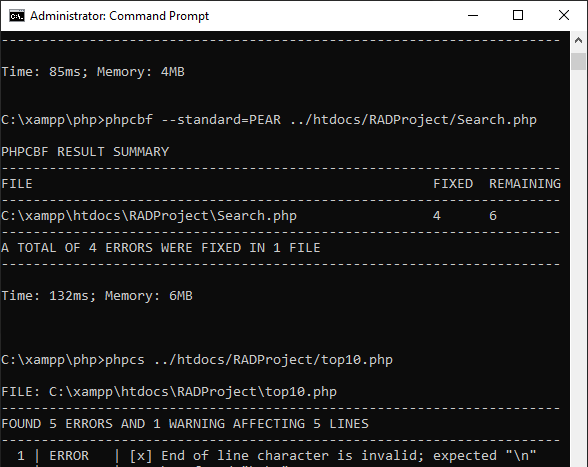


# PHP CodeSniffer

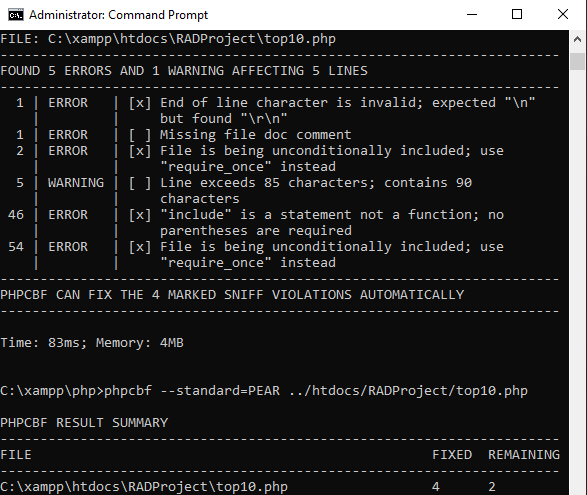
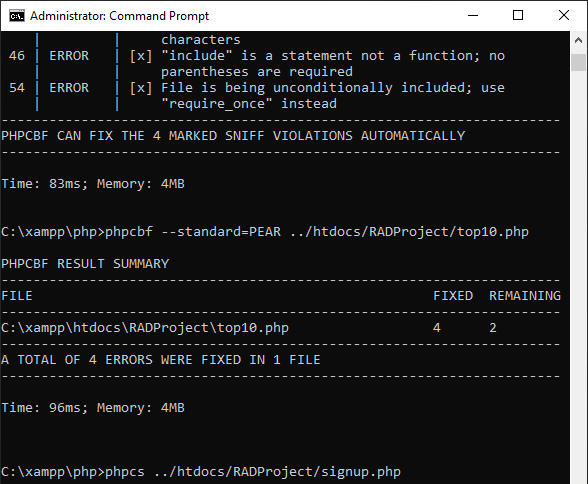
Home page (index.php)

1. 
2. 

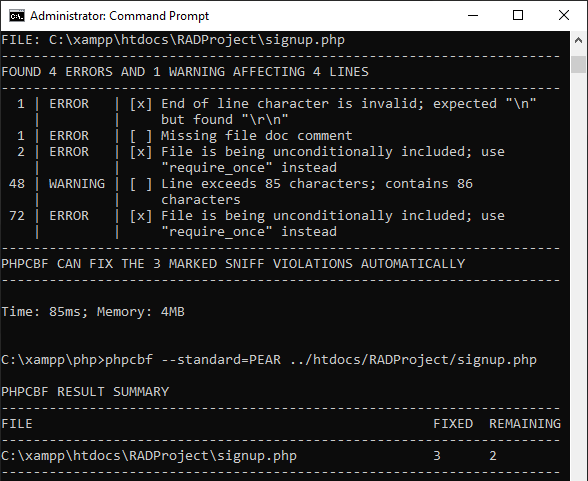
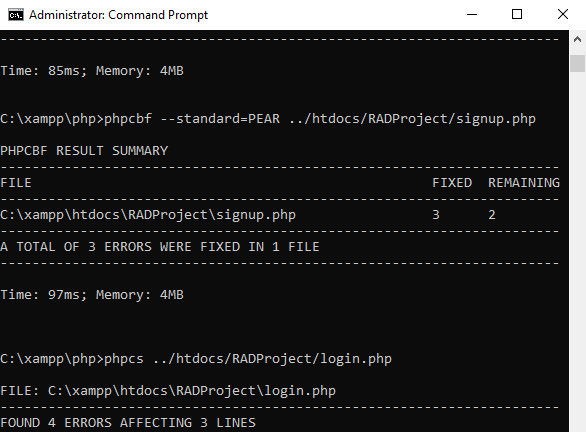
Search page (search.php)

1. 
2. 

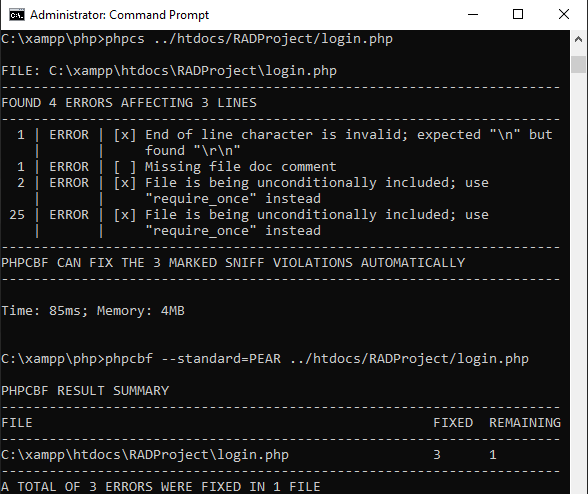
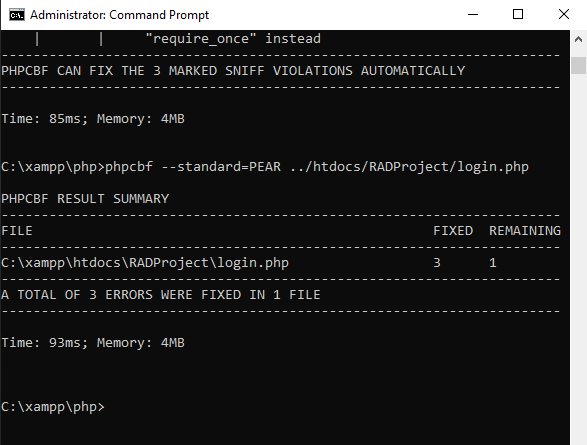
Top 10 Movies page (top10.php)

1. 
2. 

Sign Up page (signup.php)

1. 
2. 

Login page (login.php)

1. 
2. 

Software Development Testing Plan

UPDATE a test plan for the project

CHANGELOG

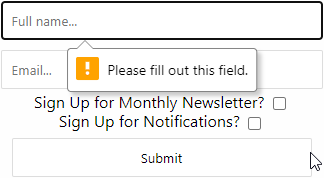
|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Date of change | Change by | Outline |
| 1.0 | 5/11/2020 | Daniel Ewen | Test Plan Creation |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Test Table

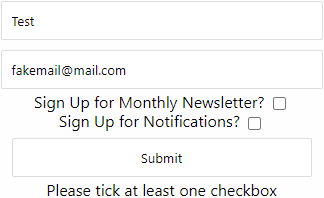
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Expected Results | Actual Results | Pass/Fail |
| Sign-Up Form | | | | |
| Precondition: Enter full name | | | | |
| 1 | User enters Full name with any combination of letters, symbols or numbers | Name validates on form submission as the input is not empty | Name validates on form submission as the input is not empty | Pass |
| 2 | User does not enter anything into full name field. | Name fails to validate and form submission is rejected | Name fails to validate and form submission is rejected | Pass |
| Precondition: Enter email address | | | | |
| 3 | User enters email with correct format | Email validates during form submission | Email validates during form submission | Pass |
| 4 | User enters email with incorrect formatting | Email does not validate at form submission and prints error. | Email does not validate at form submission and prints error. | Pass |
| Precondition: Communication option selection (monthly newsletter and/or breaking news flash notification as they occur) | | | | |
| 5 | One or both checkboxes are checked | If rest of form validates, prints success and preferences will be added to database | If rest of form validates, prints success and preferences will be added to database | Pass |
| 6 | Neither box is checked | Form fails to validate and prints error | Form fails to validate and prints error | Pass |

**Ref.**

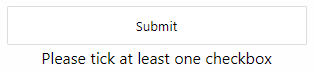
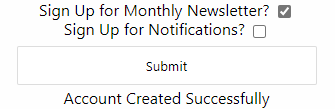
1&2.

3&4.

5&6.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Update Form  Functions identically to the signup form but if it validates and the account exists it will change their preferences. | | | | |
| Precondition: User enters details with new preferences | | | | |
| 7 | User enters valid account details with different preferences to current ones. | Success message printed.  User’s details are updated | Success message printed.  User’s details are updated | Pass |
| 8 | User enters invalid account | Failure message printed | Failure message printed | Pass |

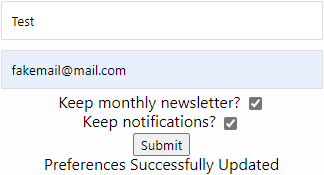
Ref:

7.

Original preferences:

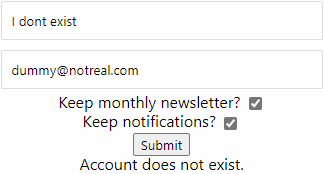


New Preferences:





8.



Sprint 3

Team Name : The Imposters

Scrum Master (#1) : Daniel Ewen

Team Member (#2) : Lisa Mckenna

Team Member (#3) : Swen Lee

Meeting Minutes

Meeting Minutes for Sprint THREE

Location : Murdoch T101 Office

Date : 19th November 2020

Time : 12:00 p.m.

# Meeting Minutes

Items discussed during team meeting in sprint three

## Agenda Items

1. Discuss client requirements
2. Confirm requirements with client
3. Allocate tasks:
   * Update GitHub repository
   * Update project plan
   * Develop software testing plan
   * Develop optimisation report
   * Update website according to client requirements
   * Finalise documentation (update master document)

|  |  |  |  |
| --- | --- | --- | --- |
| Action Items | Owner(s) | Deadline | Status |
| Update GitHub repository | Daniel | 19/11/2020 | Complete |
| Update Project Plan | Team | 19/11/2020 | Complete |
| Optimisation Report | Lisa | 25/11/2020 | In Progress |
| Update Software Testing Plan | Swen | 25/11/2020 | In Progress |
| Update Website | Daniel | 25/11/2020 | In Progress |
| Finalise Documentation | Swen | 25/11/2020 | Pending |

Source Control Snapshot and History

Contains snapshot for GitHub as well as the progress of the work

# Source Control Snapshot

Below is the snapshot for out GitHub repository. Click [here](https://github.com/Devarthran/ImpostersRAD/tree/main/Sprint%20Three) to access it.

A picture containing text, screenshot, computer

Description automatically generated

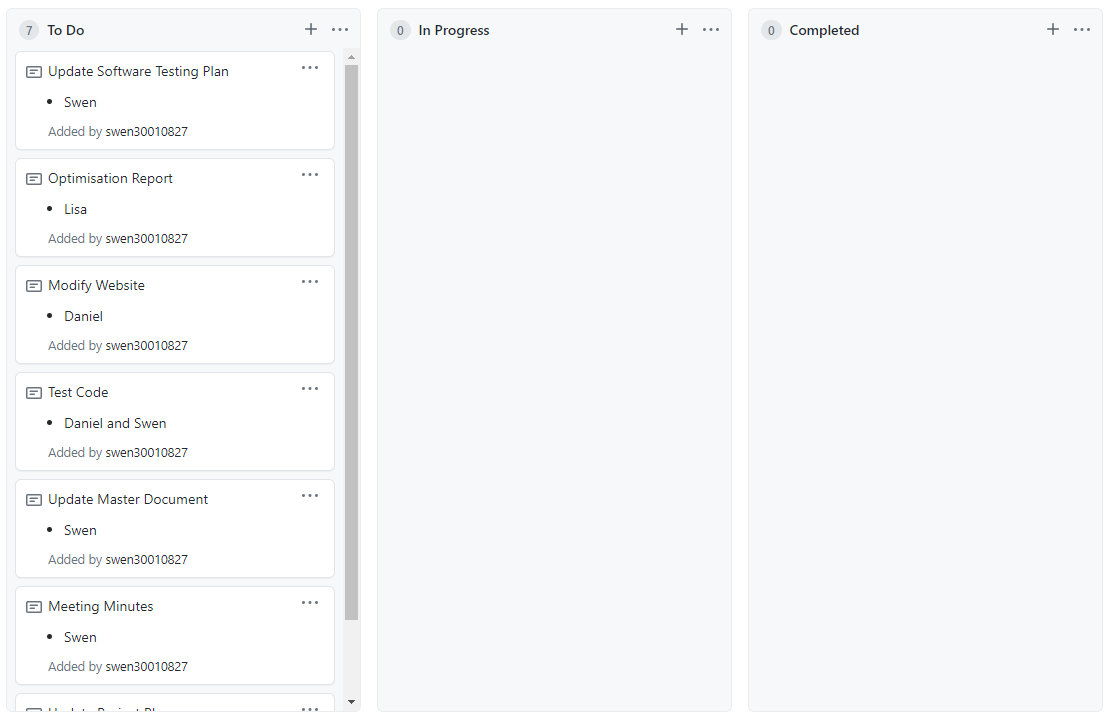
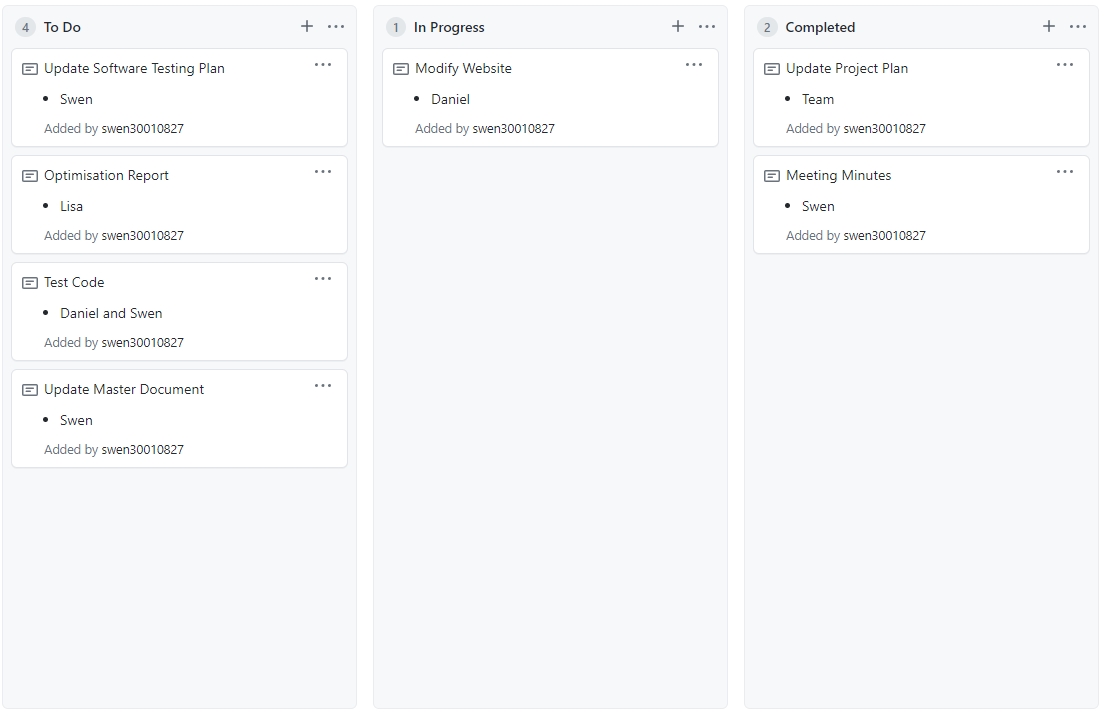
Graphical user interface, text, application

Description automatically generated

# Source Control History

|  |  |
| --- | --- |
| Date | Evidence |
| 19/11/2020 (Thursday) | Ref 1 |
| 20/11/2020 (Friday) | Ref 2 |
| 23/11/2020 (Monday) | Ref 3 |
| 24/11/2020 (Tuesday) | Ref 4 |
| 25/11/2020 (Wednesday) | Ref 5 |

Ref 1

1. 
2. 

Ref 2

1. Graphical user interface, application

   Description automatically generated

Ref 3

1. Graphical user interface, application

   Description automatically generated

Ref 4

1. Graphical user interface, application, Teams

   Description automatically generated

Ref 5

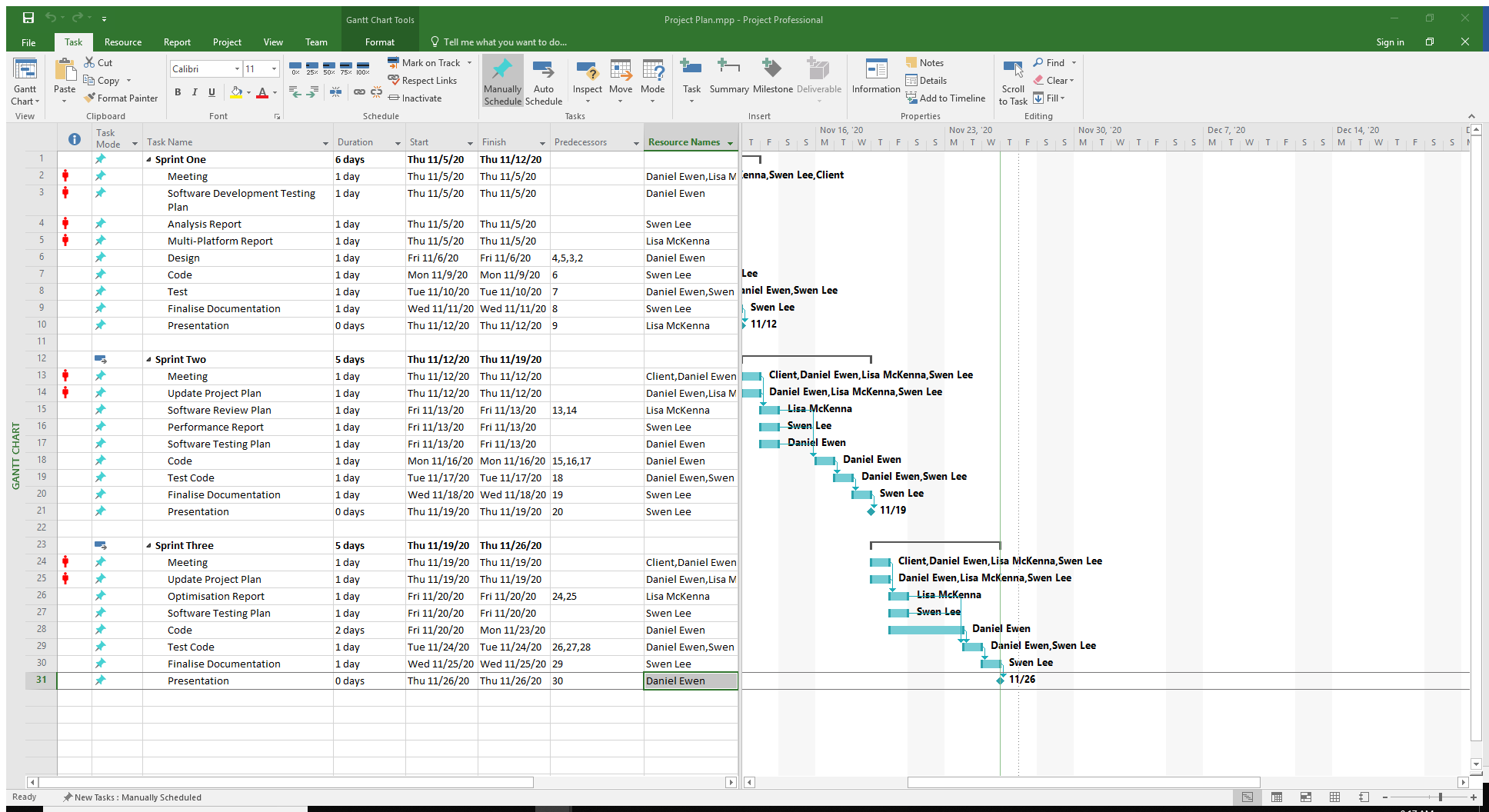
1. Graphical user interface, application

   Description automatically generated

Project Management Plan

Contains the project management plan for sprint THREE

# Project Management Plan Snapshot



Optimisation Report

# Performance Optimisation

Performance optimisation is key in having an efficiently functional application and is executed by monitoring and analysing the performance of an application and identifying ways to improve it to work more efficiently and execute more rapidly.

At CITE we have identified the levels of optimisation to be focused on:

1. Design Level
   * + - The design of our system is making the best use of the available resources, given goals, and expected load.
       - The architectural design of our system plays an important role on affecting system performance.
       - Optimise the system to minimize network requests, ideally making a single request rather than multiple requests.
2. Algorithms and Data Structures

* Algorithm and data structures are key players, being crucial to the systems performance.
* To ensure our system is optimised, we are making sure the algorithms are constant, logarithmic, linear, or log linear.
* We are implementing abstract data types as they are more efficient for system optimisation.

1. Source Code Level

* On our implementation of algorithms, along with our source code choices are quite crucial on system optimisation.

## Client-side and Server-side Optimisation

Our client-side relates to how the performance is seen on the web browser or the user interface. This includes page load time, downloading of all resources, image load time etc.

Our server-side relates to how long it takes to run on the server to execute requests. Optimising our performance on the server generally involves Optimising the database queries and other application dependencies.

## Client-side Performance Optimisation

Below are some of the ways we are optimising performance on the client side:

1. Content Delivery Network

* The content delivery networks are an intelligent way that we are handling our static files like JavaScript, CSS and image files which do not change.

1. Bundle and Minification

* Bundling our files together and producing fewer files improves performance.
* Minifying our files and removing all unnecessary characters e.g. white spaces also improves our performance

1. Optimising Image Usage

* Most of our images can be optimised and made smaller

1. Removing duplicate code JavaScript and CSS

* Removing our duplicate code reduces the size of the files hence better performance.

1. Using a Minimalistic Styling Framework

* This will help with our styling aspect and as the styling framework has already been optimised and minified for better performance it should be effective in keeping the performance of our system optimal.

Software Development Testing Plan

UPDATE test plan for the project

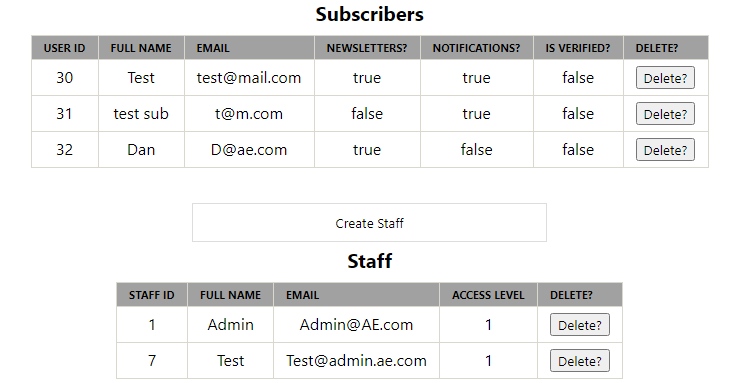
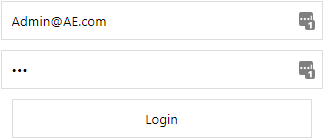
|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Date of change | Change by | Outline |
| 1.0 | 5/11/2020 | Daniel Ewen | Test Plan Creation |
| 1.1 | 16/11/2020 | Lisa McKenna | Update Test Plan for Sprint 2 |
| 1.2 | 23/11/2020 | Swen Lee | Update Test Plan for Sprint 3 |
|  |  |  |  |

# Updated Test Table

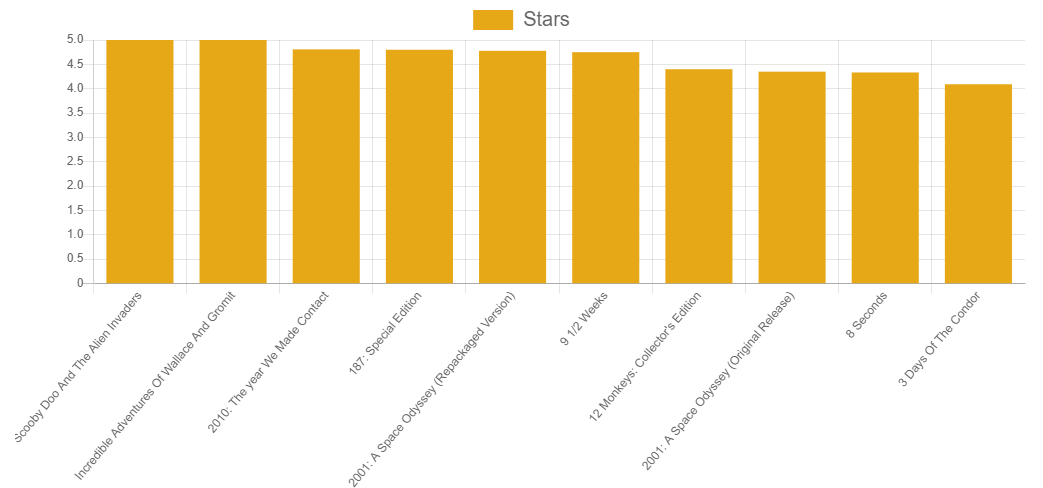
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Expected Results | Actual Results | Pass/Fail |
| Precondition: Administrator enters username and password | | | | |
| 11 | Login as administrator with correct credentials | Able to access and edit database | Able to access and edit database | Pass |
| Precondition: N/A | | | | |
| 12 | Click on ratings page | Display the top 10 rated movies | Display the top 10 rated movies | Pass |
| 13 | Remain at the top 10 ratings page | Refreshes the page automatically every few minutes | Refreshes the page automatically every few minutes | Pass |
| 14 | Click star to rate movie on movie page. | Rating increases / decreases based on rating direction. | Rating number changes expectedly | Pass |

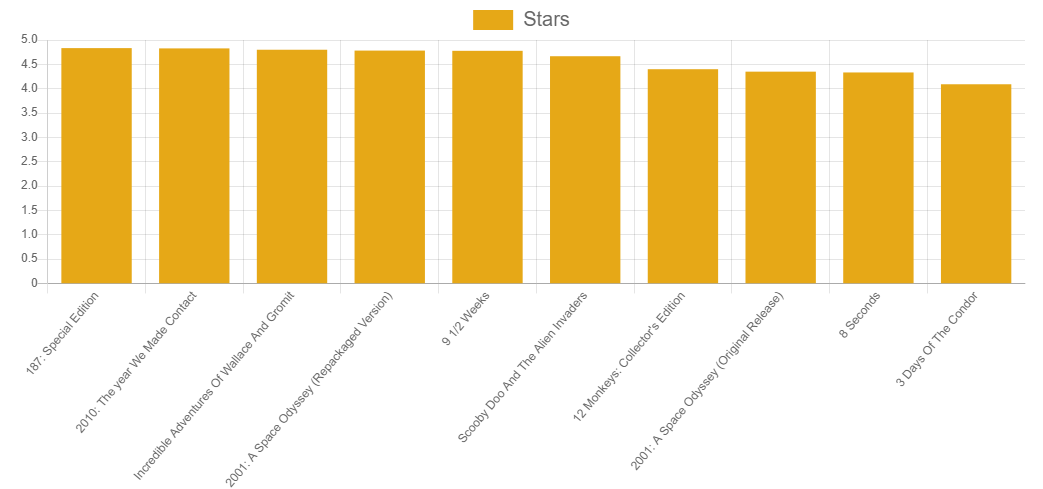
Reference:

11.a/b



12/13.





14.a/b



# Exit Criteria

Exit criteria defines when to stop the testing process, for example, when a set of tests has achieved its objective.

Exit criteria may involve:

* Estimates of defect density or reliability measures
* Cost
* Residual risks (e.g. unfixed bugs)
* Thoroughness measures (e.g. code functionality)

(Mostafa, 2018)

# Conclusion

With Agile development, project development is split into sprints or iterations. At the end of every sprint, the product is tested and client input is obtained to ensure the product meets the requirements. If there are errors or bugs detected in the sprint, that sprint is repeated until it meets the specified requirements. With Agile, client satisfaction is prioritised and therefore the quality of the product is ensured.

Project Handover

Team Name : The Imposters

Team Member (#1) : Lisa Mckenna

Team Member (#2) : Swen Lee

Team Member (#3) : Daniel Ewen

Meeting Minutes

Meeting Minutes for Project Handover

Location : Murdoch T101 Office

Date : 26th November 2020

Time : 12:00 p.m.

# Meeting Minutes

Items discussed during team meeting for project handover

## Agenda Items

1. Discuss handover requirements
2. Allocate tasks:
   * Update GitHub repository
   * Update project plan
   * Update software review plan
   * Update software testing plan
   * Update website according to client requirements
   * Finalise documentation (update master document)

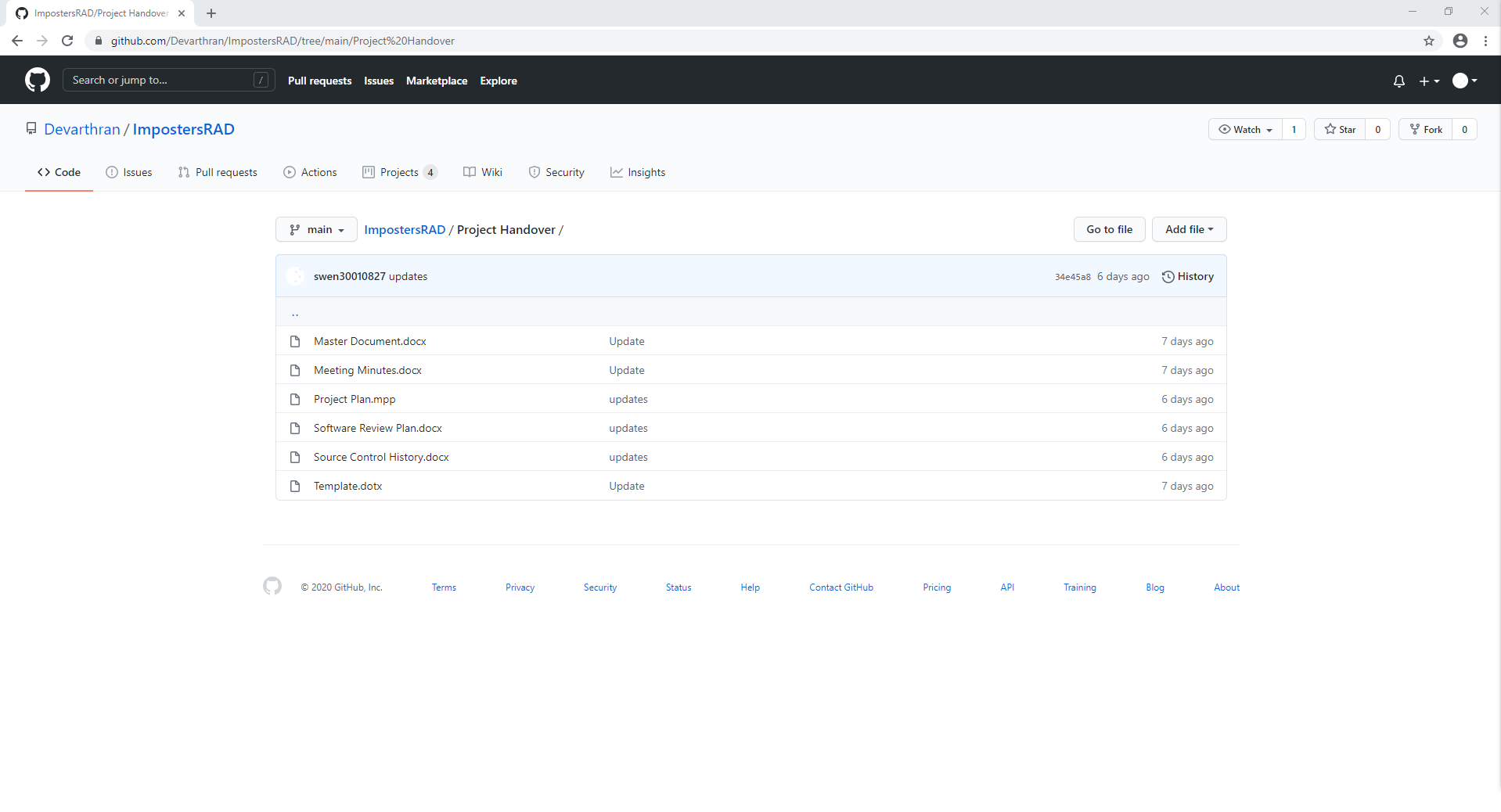
|  |  |  |  |
| --- | --- | --- | --- |
| Action Items | Owner(s) | Deadline | Status |
| Update GitHub repository | Daniel | 26/11/2020 | Complete |
| Update Project Plan | Team | 26/11/2020 | Complete |
| Update Software Review Plan | Swen | 02/11/2020 | In Progress |
| Update Software Testing Plan | Daniel | 02/12/2020 | Pending |
| Update Website | Lisa | 02/11/2020 | In Progress |
| Finalise Documentation | Swen | 02/12/2020 | Pending |

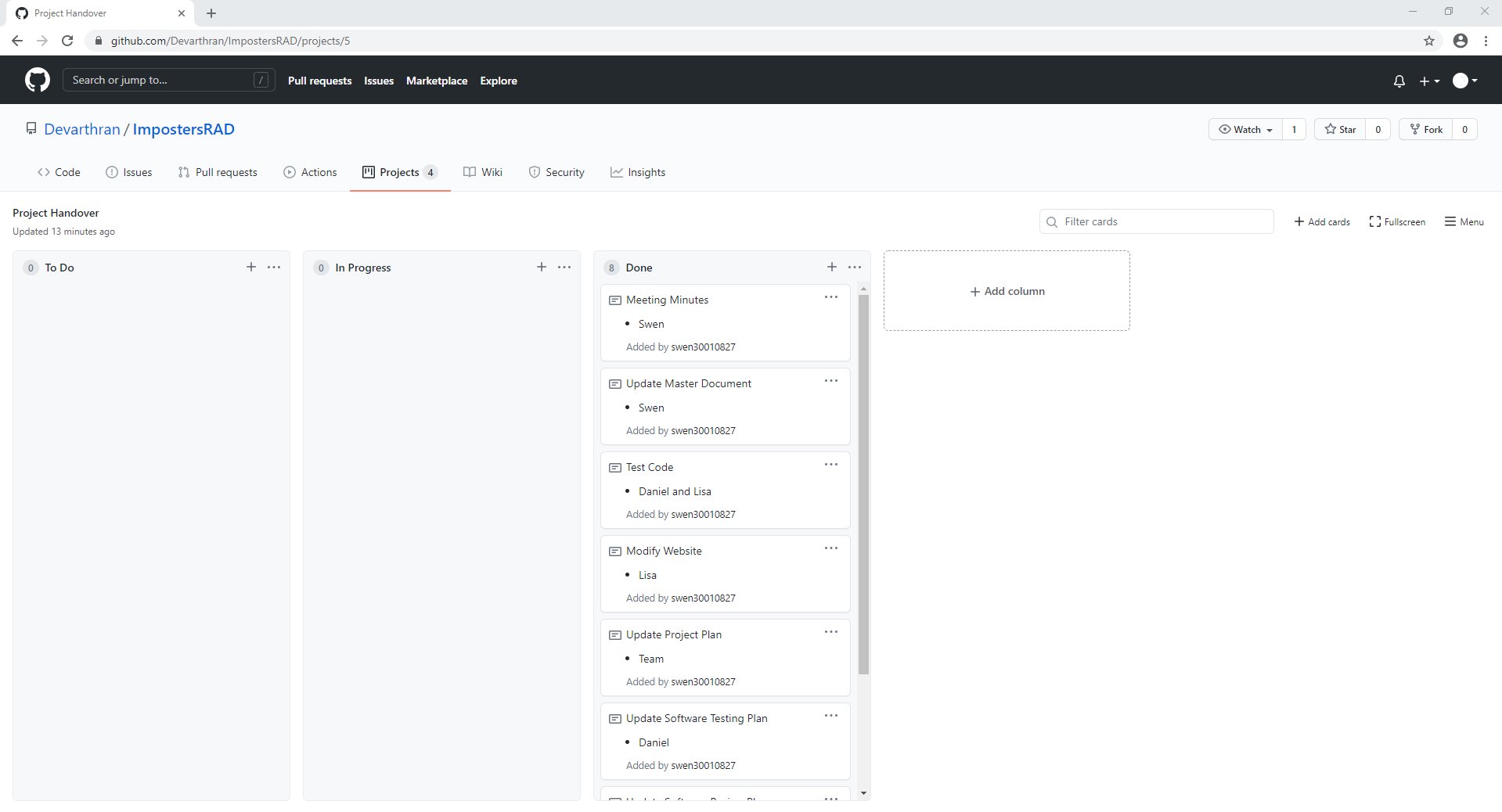
Source Control Snapshot and History

Contains snapshot for github as well as the progress of the work

# Source Control Snapshot

Below is the snapshot for of our GitHub repository. Click [here](https://github.com/Devarthran/ImpostersRAD/tree/main/Project%20Handover) to access it.

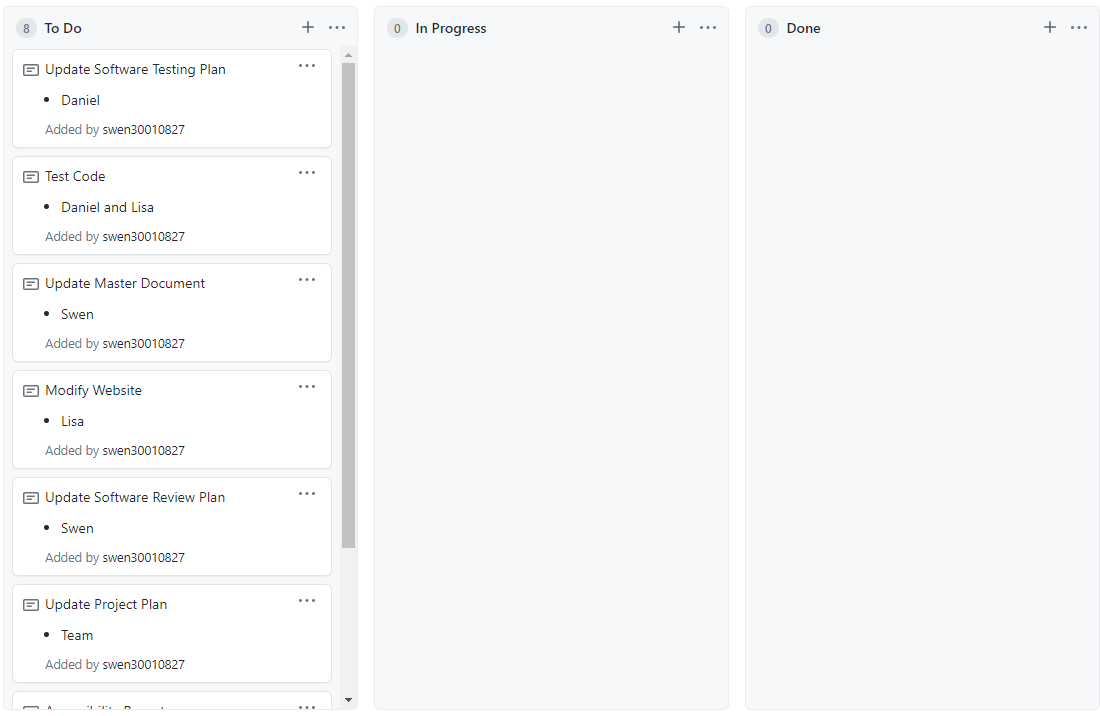
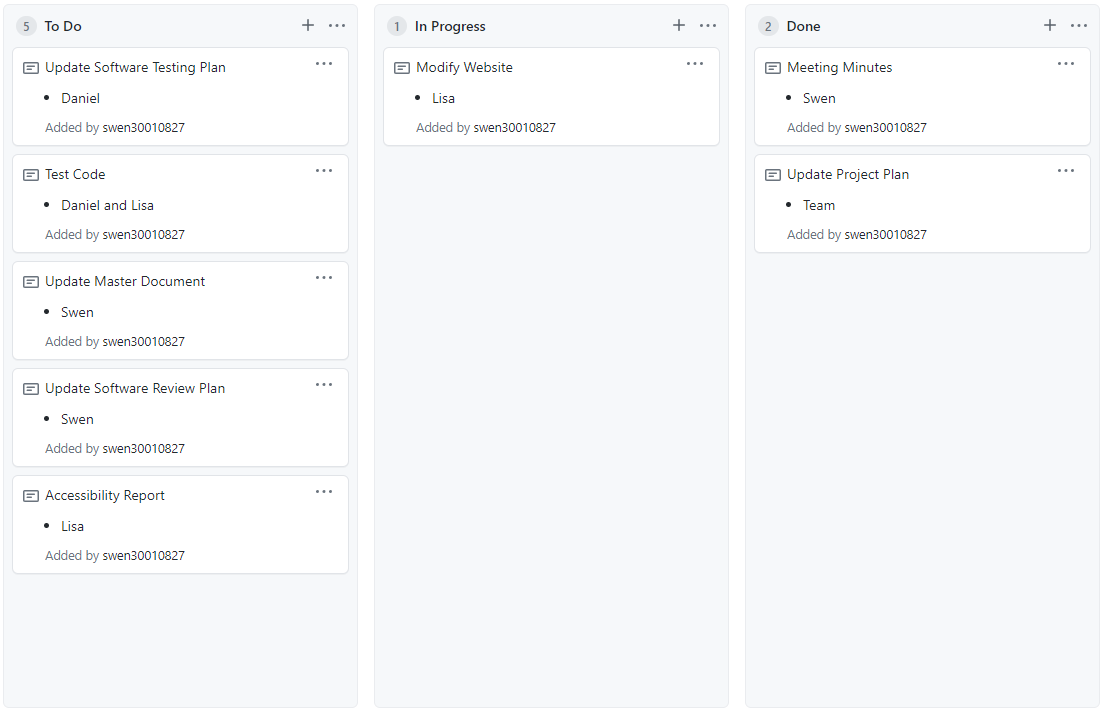




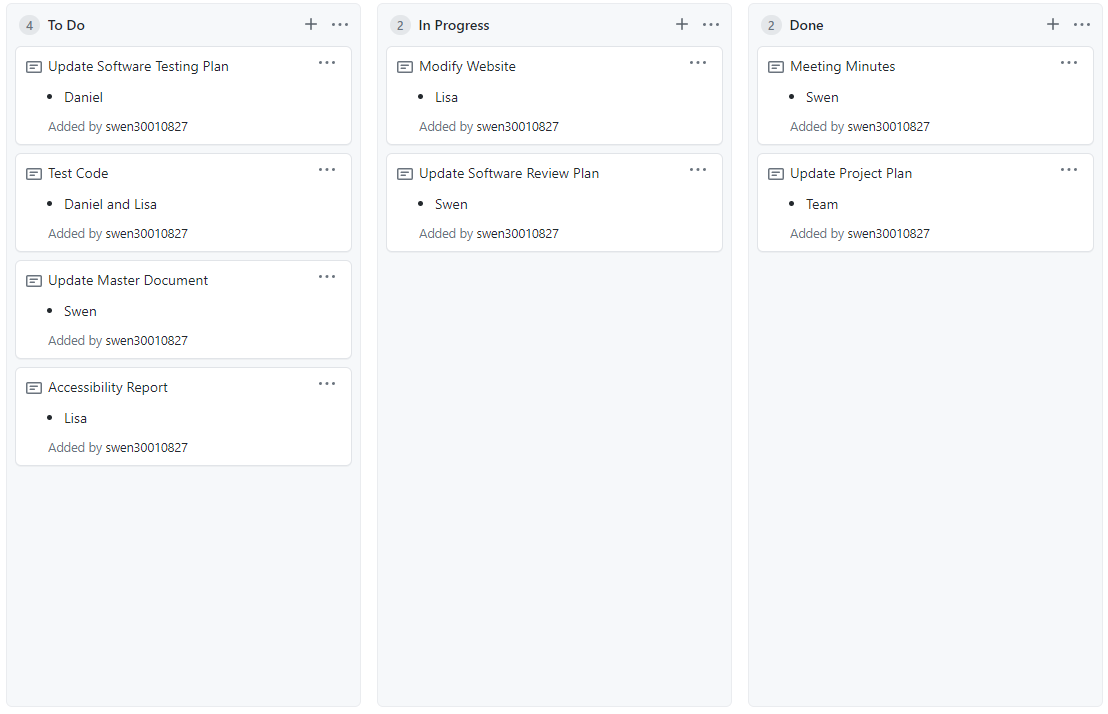
# Source Control History

|  |  |
| --- | --- |
| Date | Evidence |
| 26/11/2020 (Thursday) | Ref 1 |
| 27/11/2020 (Friday) | Ref 2 |
| 30/11/2020 (Monday) | Ref 3 |
| 01/12/2020 (Tuesday) | Ref 4 |
| 02/12/2020 (Wednesday) | Ref 5 |

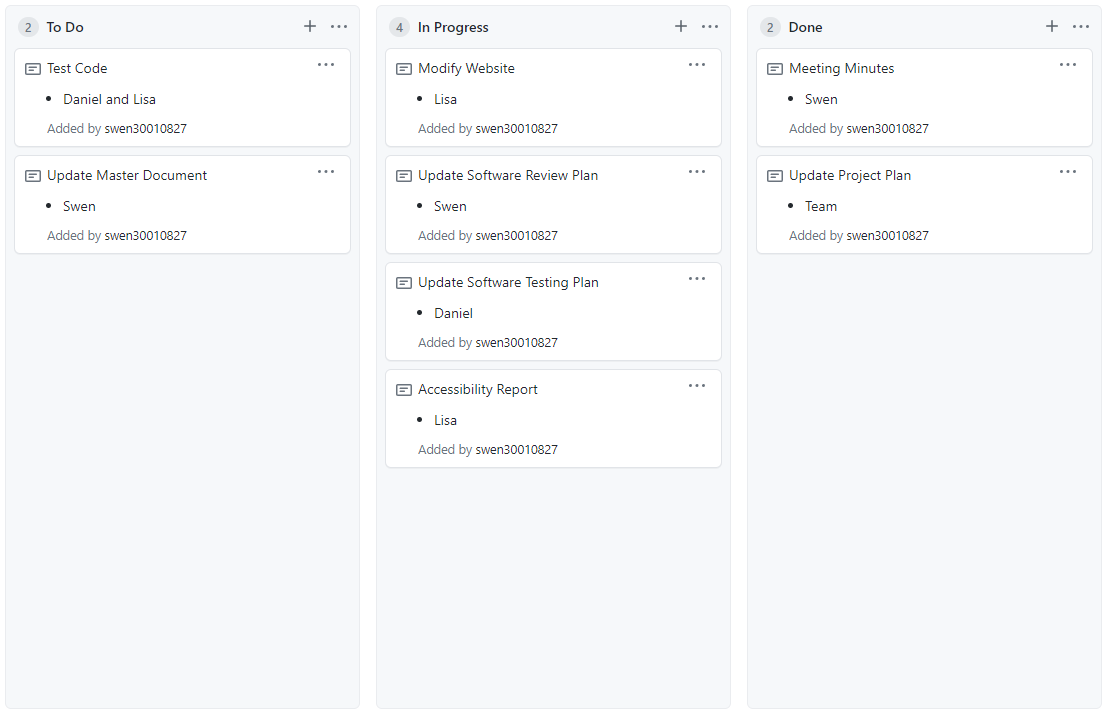
Ref 1

1. 
2. 

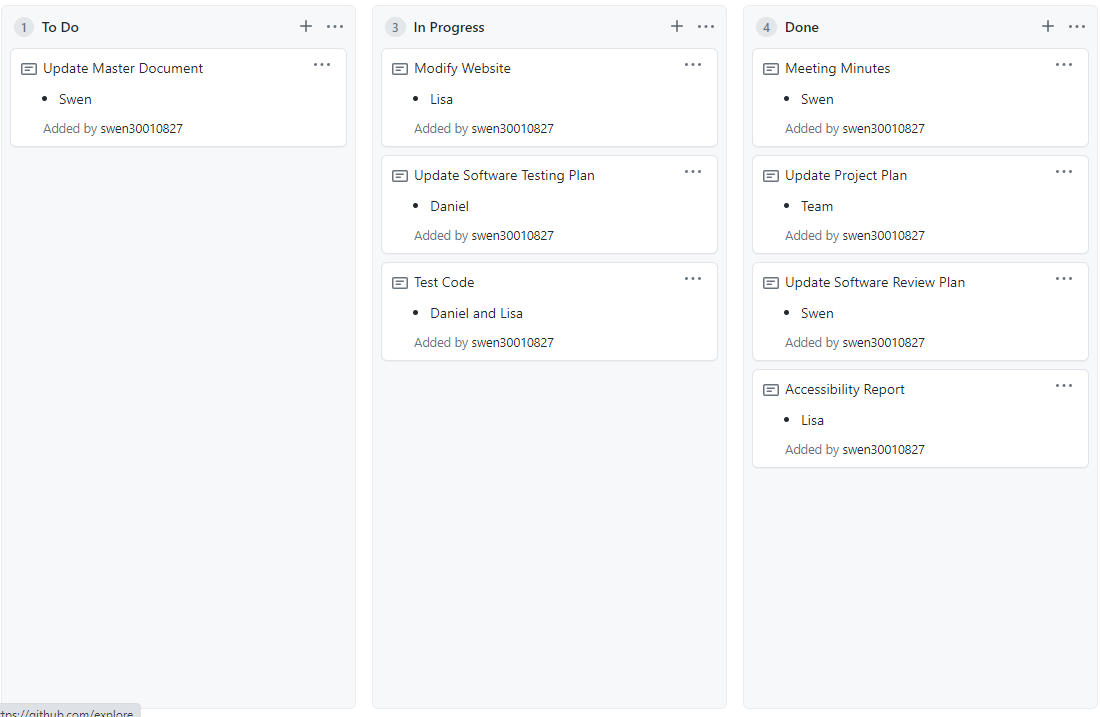
Ref 2

1. 

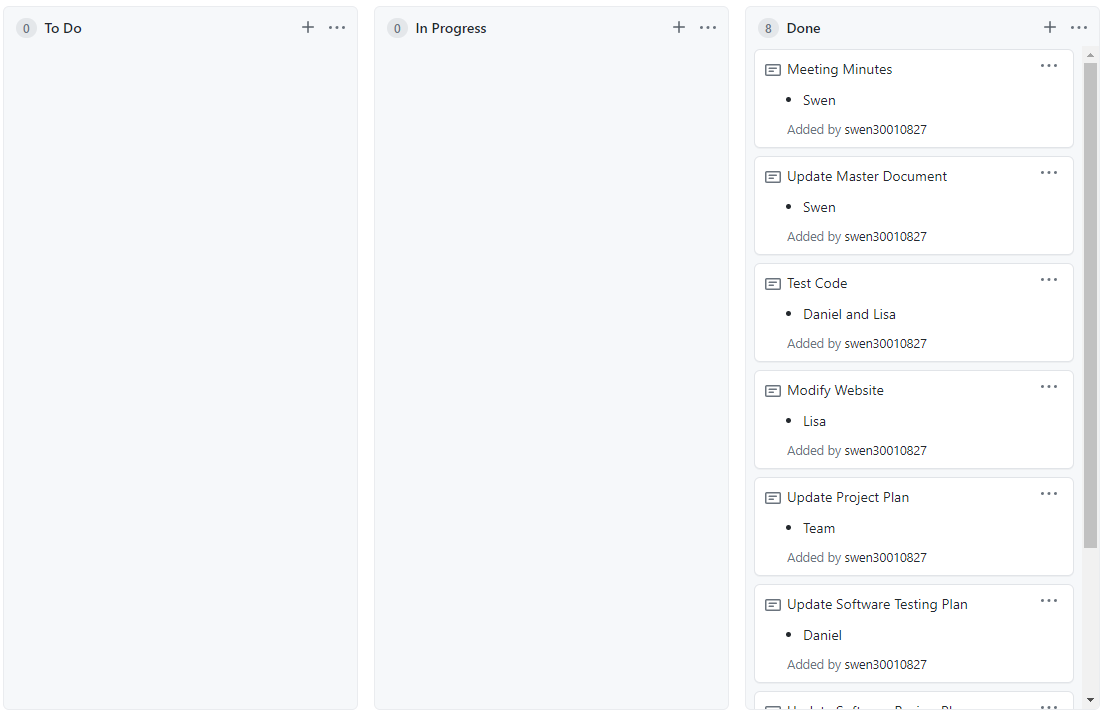
Ref 3

1. 

Ref 4

1. 

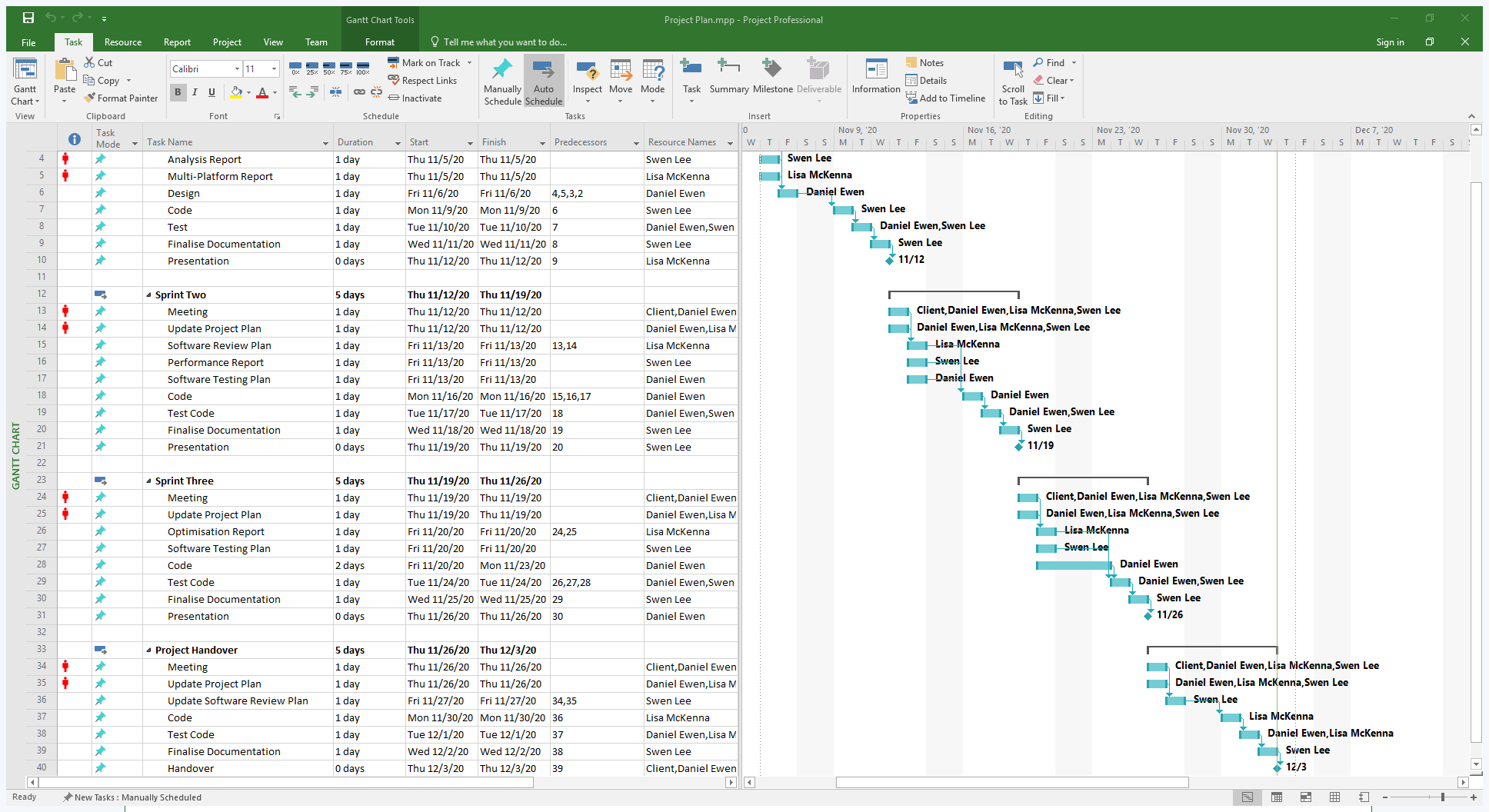
Ref 5

1. 

Project Management Plan

Contains the project management plan for Project Handover

# Project Management Plan Snapshot



Software Review Report

# Development Quality

A few appropriate measures have been taken to ensure the product developed meets the client requirements. For example:

* A test plan has been developed (Software Development Testing Plan.docx) which includes:
  + Scope
  + Roles and Responsibilities
  + Test Methodology
  + Test Deliverables
  + Testing Tools and Environment
  + Test Table
  + Exit Criteria
* A software review plan has been developed (Software Review Plan.docx) which includes a checklist of how the website should function

# Code Testing

A few tests have been conducted on the website, such as:

* PHP\_CodeSniffer
  + Checks the website for errors and warnings and fixes those errors.
  + Screenshot evidence is located in the master document (Master Document.docx)
* Performance Test
  + A performance test is conducted using Google Chrome’s Developer Tools. It checks the performance metrics of the website while running or reloading
  + Screenshot evidence is located in the performance report (Performance Report.docx)
* Unit Test
  + Unit tests has been conducted on the website according to the test table developed.
  + Screenshot evidence is located in the Software Development Testing Plan.

# Future Modification and Refinements

To ensure future modification and refinements are supported:

* The code is commented appropriately
* The documentation for the code is properly stored
* The development plans are properly documented and put into a master document
* The code is uploaded to GitHub to ensure all the developers receive the latest updates

# Mapping

|  |  |  |
| --- | --- | --- |
| User Requirements | Implementation |  |
| Website has a front end | A front end is implemented where client can fill articles and announcements |  |
| Website has a search page | A search page is implemented with the option to search according to title, genre, rating or year |  |
| Website searches according to conditions selected | When the search button is clicked, the table displays the output according to the condition inputted |  |
| User is able to sign up to subscribe for newsletter or notifications | There is a newsletter page for users to sign up and subscribe to newsletter or notifications |  |
| Website has a profile page for users to sign in and change their subscription | In the newsletter page, there is an option for users to update their subscription details |  |
| User is able to unsubscribe as they please | In the newsletter page, there is an option for users to unsubscribe |  |
| User is able to rate the movies | In the search page, there is a column called ratings allowing users to click the stars (rating) |  |
| Website displays a graph with the top 10 rated movies | There is a page called Top 10 Movies displaying the top 10 rated movies |  |
| Graph refreshes automatically | The graph refreshes automatically every 1.5 seconds |  |
| Website has a login page for administrators | There is a page called Login for administrators to access the database |  |
| Administrators are able to add, edit and delete users | After the administrator is logged in, they can add, edit and delete the users |  |
| Administrator passwords are secured | The administrator passwords are hashed and stored in the database |  |
| Administrator passwords pass a minimum complexity test | The administrator password has to contain 8 characters including at least 1 capital and 1 number. |  |
| Website is accessible | Semantic elements are implemented for the tables and forms, however not for the ratings |  |

Software Development Testing Plan

Develop a test plan for the project

CHANGELOG

|  |  |  |  |
| --- | --- | --- | --- |
| Version # | Date of change | Change by | Outline |
| 1.0 | 5/11/2020 | Daniel Ewen | Test Plan Creation |
| 1.1 | 16/11/2020 | Lisa McKenna | Update Test Plan for Sprint 2 |
| 1.2 | 23/11/2020 | Swen Lee | Update Test Plan for Sprint 3 |
| 1.3 | 3/12/2020 | Daniel Ewen | Updated plan and added screenshots to testing plans. |

# Introduction

## Scope

### In Scope

Features to be tested:

* Application design is responsive
* User Interface
* Interface responds in real-time
* Database migrates correctly to the new design.

## Out of Scope

Features that won’t be tested:

* Movie database

## Quality Objective

Our team’s objective is to deliver a working prototype to the client with the following qualities:

* AUT[\*](#_Terms_/_Acronyms) must conform to the client’s requirements for each sprint.
* AUT must meet the client’s quality standards.
* Bugs/defects are tested and fixed before the application is deployed for the client.
* The AUT meets coding and commenting standards defined by each coding language standard.

## Roles and Responsibilities

|  |  |
| --- | --- |
| Role | Team Member/s |
| Developers | Research, design, implement and manage software programs |
| Test Manager | Organise and control the testing process to deliver a high-quality software |
| QA analyst | Test software on other computers to ensure they are functioning accurately |
| Business Analyst | Connect IT and business using data analytics and determine client requirements |
| Bug Triage | Evaluate, prioritise and assign resolution defects |

# Test Methodology

## Overview

Our team will utilize the RAD[\*](#_Terms_/_Acronyms) test methodology throughout this project. Each sprint will seek to develop, test, and deliver a fully functioning prototype to the specification of the client.

## Test Levels

Our testing team will implement the following test types for this project:

* Exploratory Testing
* Functional Testing
* Accessibility Testing
* Compatibility Testing
* Integration Testing
* System testing
* User Acceptance Testing (UAT)

## Bug Triage

Our process for bug/defect triage is as follows:

* **Bug discovered:** Bug report is added to the bug list. For this our team will use the issue tracker built into GitHub, detailing the bug.
* **Investigate:** Developers read the report and try to replicate the bug. If replication occurs, the bug can then be prioritized.
* **Bug added to backlog:** Bug is awaiting resolution during sprint.
* **Resolved:** Bug is fixed during development and the issue is closed.

## Suspension Criteria & Resumption Requirements

During testing, should any test return more than a 30% fail the following should occur:

* Testing ceases
* Test case / module being tested investigated for design flaws
* If bugs are discovered, add them to the triage queue.
* Once the bug has resolved in the triage queue, run the test again.

## Test Completeness

Testing will be complete when:

* The AUT has been deemed to have 100% test coverage,
* All designed test cases return with a pass,
* All current bugs/issues have been resolved,
* The client is satisfied with the current condition of the AUT and has signed off on its deployment.

# Test Deliverables

|  |
| --- |
| Deliverables |
| Test Plan |
| Test Cases with validation |
| Requirements |
| Analysis Report |
| Bug Reports |
| Client Sign-off |

# Resource & Environment Needs

## Testing Tools

* **Chromium developer tools:** Exploratory testing and during development.
* **PHP CodeSniffer:** Checks the PHP code for syntax and standards errors.

## Test Environment

### Hardware Environment

A modern development workstation with the following minimum specifications:

|  |
| --- |
| Part |
| Intel CPU 4-cores or more with hyperthreading |
| 16GB DDR4 RAM 2600MHz |
|  |

### Required Software

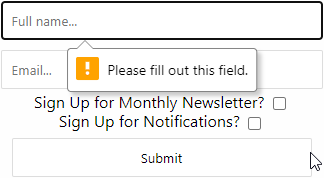
|  |  |
| --- | --- |
| Software | |
| Xampp | Local webserver and database hosting |
| Visual Studio Code | Code and application development |
| GitHub Desktop | Source Control |

# Test Table

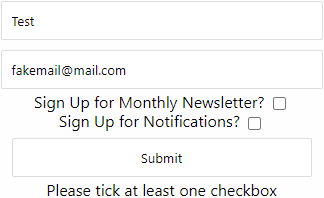
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Expected Results | Actual Results | Pass/Fail |
| Sign-Up Form | | | | |
| Precondition: Enter full name | | | | |
| 1 | User enters Full name with any combination of letters, symbols or numbers | Name validates on form submission as the input is not empty | Name validates on form submission as the input is not empty | Pass |
| 2 | User does not enter anything into full name field. | Name fails to validate and form submission is rejected | Name fails to validate and form submission is rejected | Pass |
| Precondition: Enter email address | | | | |
| 3 | User enters email with correct format | Email validates during form submission | Email validates during form submission | Pass |
| 4 | User enters email with incorrect formatting | Email does not validate at form submission and prints error. | Email does not validate at form submission and prints error. | Pass |
| Precondition: Communication option selection (monthly newsletter and/or breaking news flash notification as they occur) | | | | |
| 5 | One or both checkboxes are checked | If rest of form validates, prints success and preferences will be added to database | If rest of form validates, prints success and preferences will be added to database | Pass |
| 6 | Neither box is checked | Form fails to validate and prints error | Form fails to validate and prints error | Pass |

**Ref.**

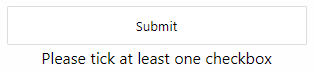
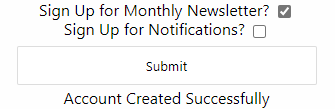
1&2.

3&4.

5&6.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Update Form  Functions identically to the signup form but if it validates and the account exists it will change their preferences. | | | | |
| Precondition: User enters details with new preferences | | | | |
| 7 | User enters valid account details with different preferences to current ones. | Success message printed.  User’s details are updated | Success message printed.  User’s details are updated | Pass |
| 8 | User enters invalid account | Failure message printed | Failure message printed | Pass |

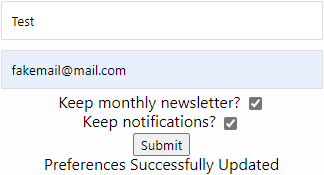
Ref:

7.

Original preferences:

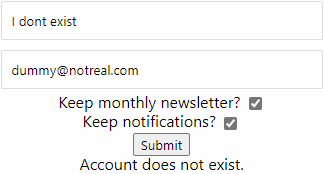


New Preferences:





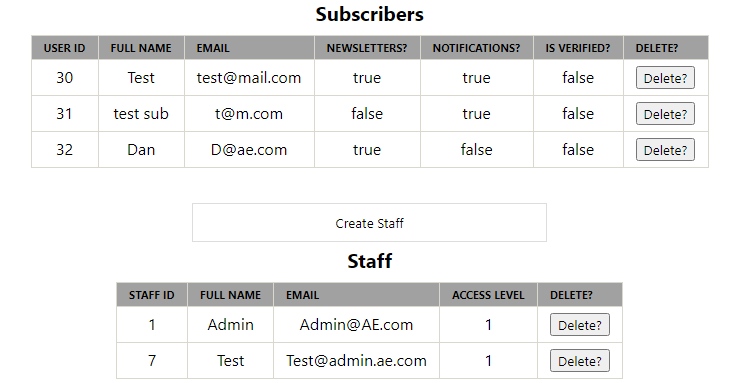
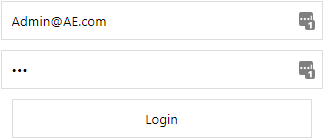
8.



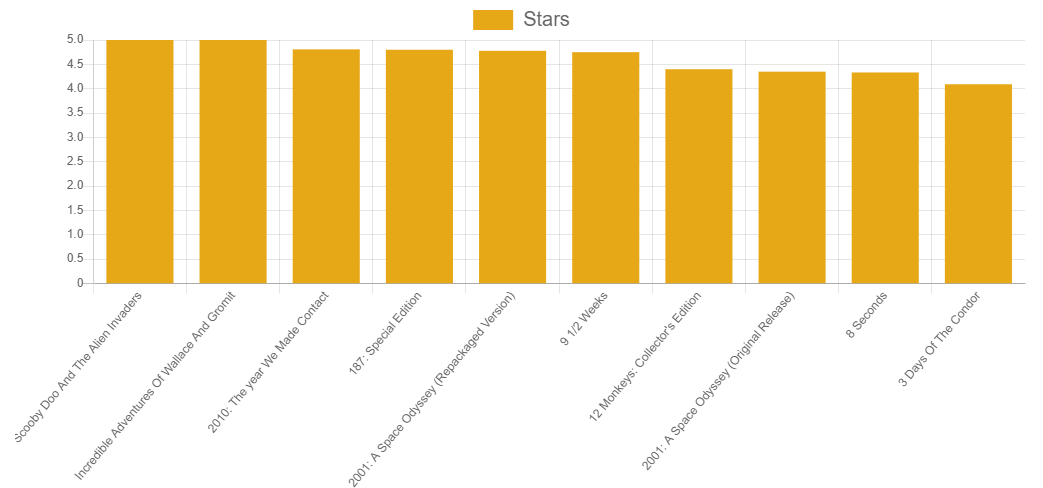
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test ID | Description | Expected Results | Actual Results | Pass/Fail |
| Sprint 3 Update | | | | |
| Precondition: Administrator enters username and password | | | | |
| 9 | Login as administrator with correct credentials | Able to access and edit database | Able to access and edit database | Pass |
| Precondition: N/A | | | | |
| 10 | Click on ratings page | Display the top 10 rated movies | Display the top 10 rated movies | Pass |
| 11 | Remain at the top 10 ratings page | Refreshes the page automatically every few minutes | Refreshes the page automatically every few minutes | Pass |
| 12 | Click star to rate movie on movie page. | Rating increases / decreases based on rating direction. | Rating number changes expectedly | Pass |

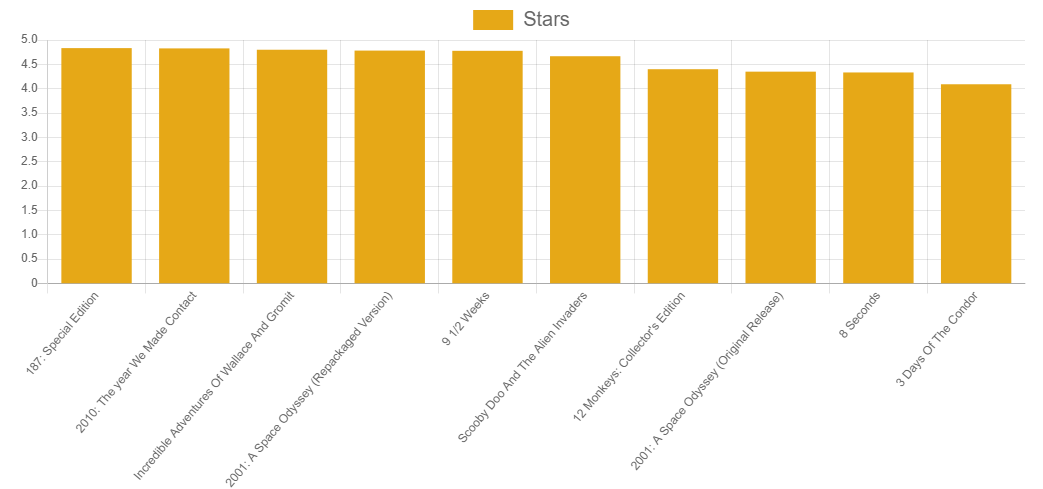
Reference:

9.a/b



10/11.





12.a/b



# Exit Criteria

## Exit Criteria

Exit criteria defines when to stop the testing process, for example, when a set of tests has achieved its objective.

Exit criteria may involve:

* Estimates of defect density or reliability measures
* Cost
* Residual risks (e.g. unfixed bugs)
* Thoroughness measures (e.g. code functionality)

(Mostafa, 2018)

# Conclusion

With Agile development, project development is split into sprints or iterations. At the end of every sprint, the product is tested and client input is obtained to ensure the product meets the requirements. If there are errors or bugs detected in the sprint, that sprint is repeated until it meets the specified requirements. With Agile, client satisfaction is prioritised and therefore the quality of the product is ensured.

# Terms / Acronyms

|  |  |
| --- | --- |
| Term / Acronym | Definition |
| RAD | Rapid Application Development |
| AUT | Application Under Test |