PPMCalendar

Practical Project Management - Group Jii

Barry O'Connor (Project Manager) - N0813926

Nick Murray-Hill (Deputy Project Manager) - N0821510

Kieran Olney (Software Architect) - N0813911

Soham Jaiswal (Software Developer) - N0820198

Will Peters (Software Tester) - N0814585

Please note: when accessing the website, it is one of the free tier servers on Microsoft Azure and can be prone to delay when the server is busy. When performing an action, it can take several seconds for a response to happen but this is beyond the control of the group. A full-cost hosting platform would not suffer from this issue.

Abstract

It is a legal requirement that employees are able to take time off. To do this, a process of approval or rejection happens where each employee requests a holiday and their respective managers make a decision. How this is achieved varies wildly from company to company, with some having an informal approach and others having a more formal set of documents which need to be filled in to request a holiday. This can end up a time-consuming task since such documents need to be retained as a record and take time to complete in the first instance.

The purpose of this project is to tie this functionality into a calendar, allowing companies to have a quick and user-friendly way of handling employee holidays and company events. Using a calendar format gives users the familiarity of a type of software they will already have experience using, helping to reduce any training time required. This format also allows employees to see existing holidays and allows them to make a judgement call on whether to request holidays that coincide with busy periods.

Managers are notified of any holiday requests and can quickly respond, using the software. This helps to keep the time taken up by this activity and also doubles as a way to keep track of holidays without the need of paper forms.Managers have full control over Employee accounts and can update these as needed when staff leave or join the company, allowing the transfer of existing accounts and events to staff replacing an existing role within the company.

To achieve this, the team conducted research into the market and identified the features of key competitor products. Once these were identified, common features were discussed in an attempt to discover any areas in which improvements could be made or additional features included to add value to the application. The team used this information to create a list of requirements for the project which would outline the key aims and features of the software. The team then identified the key elements of the final application: a database to store the information, a server side API which would act as an interface between the database and the front end website and the front end website itself which would act as the main user interface. It was decided to use existing experience within the group to create a html/css/javascript frontend, a .Net Core C# api and a SQLite database.

The completed software was tested rigorously against the list of requirements and adheres to current laws regarding GDPR and accessibility guidelines, offering a web based solution which is responsive and works well on desktop, tablet and mobile. Employees and Managers can handle their events and holidays

Table of Contents

1 Introduction

During the everyday operations of a modern company, it is of growing importance that employees' schedules and bookings are organised and easily managed by their fellow colleagues or administrators. Unfortunately, this requires dedicated resources to maintain in an orderly and effective manner. Often employees' holidays or meetings will be logged onto an online calendar, spreadsheet or even paper forms. None of which offer the diverse functionality that some workplaces require, meaning some companies have to rely on several separate pieces of software instead of one central system.

Although similar tools are readily available, such as Google Calendar or Meetingbird, they often come with very limited functionality that will not fulfill all business needs or aren't available on both desktop and mobile devices. Some of the more fully featured applications can also come with a hefty price tag, meaning that companies with many employees can be left facing a large bill. Paper based solutions to this problem also exist but the team discounted these for their lack of environmental friendliness and the cumbersome nature of having to physically store sheets of paper.

This application has been conceived with this need in mind and provides its end users with all of the helpful tools they require to manage their busy diaries. PPMCalendar will offer all of the important features and more in a single solution that is easily accessible from all devices via web page, allowing employees to easily schedule their events from multiple devices using an extensive selection of categories keeping all relevant parties updated every step of the way.

1.1 Aims

The project aims to develop a calendar website which businesses can use internally to handle both employee holidays and company events and meetings. This website will adhere to modern web standards including accessibility and be accessible via desktop, tablet and mobile devices. The resultant software will allow users to schedule events or tasks on particular days across the year, which can be shared and conveniently managed within their work group. Managers will have the additional abilities of administering users and managing holiday requests.

1.2 Objectives

The above objectives for the application were discussed and 3 main areas of discussion occurred.

Firstly, the need for the final website to be accessible to all through the use of responsive and accessible coding standards. By incorporating these standards into the website, the user base can be expanded to include more people and the companies who use the software can save money on ensuring standards compliance themselves.

Secondly, the final software should have the features of a mainstream calendar application. This would ensure that users would find the application familiar and easy to use and would give them the tools needed to manage meetings, events and holiday requests in a familiar interface. This would also save on training costs for the companies who use the end software.

Finally the software would include administrator functionality which would allow managers additional privileges to administer user accounts and accept or reject holidays as needed. This would save time

during staff turnover as it would allow managers to reassign existing accounts, keeping all events and meeting information intact.

These key areas were broken down into the following objectives:

- The design of the website must be user friendly.
- The website must be responsive and display well on all screen sizes.
- The website must conform to accessibility standards.
- The website must consist of a secure server which stores user data in a database and secure communication to the frontend.
- The UI design must be intuitive, professional looking and similar to other calendar applications.
- The user should be able to create, edit and delete events from the calendar.
- Events should be edited and created by opening a form with the required fields.
- The form should be validated in a user friendly way with descriptive messages.
- The website should provide notifications to all invitees of an event when the event is created, edited or deleted.
- There must be the ability to set an event as recurring.
- The site should provide users with notifications for upcoming events.
- Events should contain information about date and time, invitees, description of the event and perhaps a location.
- The site should allow businesses to assign administrative rights to managers within the company.
- The site should allow users to request holidays from their manager, these requests will be forwarded to Administrators for approval.
- The site should allow Administrators to approve holidays from their staff, who will then be notified.
- The site should allow Administrators to manage user account details.
- The site should allow Administrators to create permanent/priority events.

2 Survey of Existing Solutions

2.1 Existing Products

Once the team had decided upon an idea, market research was carried out to help understand what products were currently available and what features they had to offer. This research was used to create the following list of strengths and weaknesses for the main competitors in the market. By spending time doing this, the team was able to refine the ideas and features that would go into the PPMCalendar project.

Many of the solutions researched relied heavily on the use of Excel or required the use of apps, sometimes vendor specific to Apple or Android, which the team felt was limiting the reach of the software.

Product	Key Features	Strengths	Weaknesses
Demand Metric - Employee Holiday Calendar	• Uses Microsoft Excel. • Customisable. • Portable.	• Compact. • Free. • Quickly get an overview of staff holidays.	• Not multiuser, so the file would have to be passed around and edited manually. • Only displays employee holidays.
Timetastic	• Individual Calendar. • Notifications • Various leave types • Google / Outlook integration • Slack Integration	• Slack Integration • Developer API • Export to Excel • Paperless • Multi-user • Support included • Annual Leave types	• Subscription based • Cannot repeat email address • Have to completely delete and resubmit events just to change duration by a day
Breath HR Holiday Management Software	• Simple HR management • All-in-one solution • For Small to Medium sized Enterprises (SMEs)	• Low subscription fee • No extra setup costs or hardware • Records leave / expense requests	• Costs money • Can't fully • Customise for specific needs • Dated interface • Not very responsive experience
HRLocker	• HR management with features beyond a holiday calendar • Management of Time on / off, digital signatures, compensation & benefits	• Sub fee per employee, good for small enterprises • Simple interface	• Lots of features hidden behind paywalls • Different prices across different currencies, not consistent

Zenefits	• All-in-one solution • Provide Advisory Services	• Automatic sync with payroll, benefits and HR • Build schedules around roles • Records location of employee clock in	• Expensive monthly payment per employee (minimum $8) • Poor support team • Not as fully automated as it states • Lack of customising
brightHR	• All-in-one solution • Employee Support • Document Storage • Health and Safety advice	• Staff Holiday planner as outstanding feature • Law advice • Notifications	• Costs money • Lacks certain automotive aspects • Poor Support
TimePage	• Heat Map shows how busy the month is • Integrated gps service, shows travel time to event destination	• Good UI design • Low Monthly Cost	• Only supports IOS • Not Free, costs $2/month or $12/year • Lacks Android & Window Support
Business Calendar 2	• Voice Commands • Learn about User Preferences and suggest common location and attendees	• Customizable UI • Easy to use	• Free but with ads • Cost $7 for pro version • Functionality like file attachment and templates locked behind paywall
Fantastical 2	• Multilingual support • Location Based Reminder tool	• Good GUI • All basic scheduling capabilities • Integrates with google calendar & outlook	• IOS & Mac only application • Expensive

2.2 Demand for Product

Information released by Gartner (Gartner, Inc., 2019) shows a trend towards cloud-centric human resources solutions which alleviate some of the administrative workload placed on managers. The analyst firm believes that by 2022 approximately 50% of large firms will have invested into cloud based solutions, showing a great demand for products in this area.

In a Human Resources Capital forecast for the years leading up to 2024, Research and Markets (2019) predicted strong growth in the sector, especially in software. This sector includes household names such as Microsoft, IBM and Oracle and the company expects the market to reach a peak value of $26.5 Billion, with a growth rate of 9.7%. The forecast goes on to explain that increasing legal requirements on information stored about employees is creating more work for companies and that, in turn, they are seeking more automated ways of dealing with this pressure.

All signs point toward the market being a healthy one with continual growth, making the decision to create software for this sector a potentially profitable one.

2.3 Company Information

The team had no particular company in mind and the project is not being created in response to a request from any particular company. The software is primarily aimed at small to medium enterprises looking for an off the shelf product to use as is, however the software could be customised or expanded upon to meet the needs of a particular company with ease.

2.4 Software Required

Due to the highly graphical nature of a calendar and the limited time available, it was decided early on that a more traditional approach involving a language like C++ would prove to be problematic. The findings of our research coupled with the preexisting experience within the group led to the decision to create a web based solution. This solution would involve a HTML / JavaScript / CSS front end and a C# .Net Core API. Many database solutions were available however, the team decided that the portability of SQLite would make the job of development easier as the database file could be included within the GitHub repository and simply downloaded by each member.

Creating an application in this manner also helps to keep the application current by cutting down the costs of future development, limiting the need for expertise in mobile development and by using an API any company adopting the software could use that interface to integrate this software into their own systems.

3 New Ideas

While the market is certainly a busy one, with many different solutions on offer, most solutions focus heavily on the sole functionality of having a quick and easy way of keeping track of employee holidays, reducing their functionality to displaying a very compact interface which only displays employee holidays. While this is useful in itself, the team believes that expanding upon this to include company holidays, meetings and events would offer a more rounded solution. The functionality for all of these events is, essentially, possible using a calendar.

Following this route would offer the business several benefits. Several key needs could be met in one single application, with holidays, meeting and company events being available to employees in one location rather than being on several applications. Companies using paper only solutions could cut down on the costs in terms of paper, ink and ultimately disposal of forms that were no longer needed. Finally, by focusing on a web based application, the team could keep the user base as wide as possible, ensuring that companies and individuals would not have to face compatibility or accessibility issues that could potentially arise from using spreadsheets or mobile phone apps.

3.1 functional requirements

Requirement ID	Requirement	Description	Implications
1	Calendar frontend	The application must have a frontend which looks like a calendar.	This is a standard interface users will know so it's vital the project uses standard layout and navigation.
2	Responsive	The application must adapt to be viewable on various devices (mobile, tablet, desktop)	A user who cannot use the site will go elsewhere, it is important to ensure the site works on different devices.
3	Accessible	The application must adhere to W3C/WAI AA standards	Accessible sites are required under law. A user who cannot use the site will go elsewhere, it is important to ensure the site is accessible for everyone.
4	Multiple control methods	The application must be controllable by several input devices (keyboard, mouse, touch)	Similar to accessible, we must allow users of keyboard and other devices to be able to use the site.
5	Security	The website must be housed on a secure server	Under law we must keep users data secure. Using a secure server and secure communication will allow this.
6	Personalised Accounts	The user can view and manage events which	Under law we must keep users data secure and available only to the user

		are relevant to their schedule	it belongs to or those allowed access to their data.
7	Meeting types	Differentiate between events and holidays and make either visible/hide.	This makes it easy for users to view different types of events and helps with clarity.
8	Upcoming Event Notifications	This must be via email but should be extended to push messages on mobile if possible.	This allows for users to be notified before an event and helps ensure they do not miss events.
9	Event Management Notifications	This must be via email but should be extended to push messages on mobile if possible.	This allows for invitees of an event to be notified about the creation of the event and any changes made to the event. This way they can be aware of any changes to date or time.
10	Event management through popup forms	The main calendar screen should enable the user to add, edit and delete events.	This allows the user to quickly manage their calendar from a single screen.
11	Add Event	Select a date to add a desired event type (e.g. Holiday, sick day, meeting). Selected parties should be notified of the added event	This should employ the correct input types to ensure the user has the most efficient way of inputting on multiple devices.
12	Delete Event	Deleting an event must remove the event and any associated recurrences of the event.	This should be a one click event for the user and should seamlessly remove all instances of the event and notify any invitees.
13	Update event	Once an event has been created, it should then be possible to modify the following ● Date ● Time ● Parties involved ● Description ● location ● Event type ● Recurring event instances	This should be as intuitive as possible for the user and should seamlessly update instances of the event and notify invitees.
14	Navigate month / year	Ability to scroll select the month and or year whether that be before or after the current date.	The user needs to be able to move backwards and forwards through the year and view

15	RSVP event	Confirm attendance at a meeting or reject	The user should be able to respond to an invite by accepting or refusing and theis should notify the organiser.
16	Repeat events	Events can be scheduled to repeat at an interval (e.g. 1st of every month). This will be automatically accounted for on the calendar and notify all parties with required notice	Users or Administrators may want recurring events every year - Christmas Holidays for example. This should be simple and intuitive to accomplish.
17	Login	Users must be able to login using their credentials to access their calendar and personal information.	This feeds back into the data protection and privacy laws, allowing privacy and security to surround a user's account.
18	Logout	The user should be able to fully logout of their account.	This will totally remove access to their personal data and content until they log back in again.
ADMIN			
19	Approve holidays	When an event has been created by a user (not admin), it is required that this event is approved in cases of holidays being booked etc.	Management must be able to approve holidays after consultation and notify staff of the decision.
20	Account management	Managing accounts of subordinates.	Management must be able to manage accounts of staff to allow for staff turnover.

3.2 Non-Functional Requirements

The following non-functional requirements were identified by the team:

- The application should be scalable and allow for many users.
- The system should load within a reasonable timeframe.
- The system should be user friendly and intuitive to use.
- The code for the application should be easy to maintain.

4 Design & Development

4.1 Gantt Chart

In order to plan timescales for the project and to track the progress of the group, a Gantt chart was created. The initial few weeks of the project including the development of the definition document are highlighted here and the full version of the chart is available in Appendix A.

Calendar Software Project	117 days	Fri 22/11/19	Mon 04/05/20		
Draft Definition Document	11 days	Fri 22/11/19	Fri 06/12/19		
Slack / Google Docs Setup	1 day	Fri 22/11/19	Fri 22/11/19		Barry
Project Idea Discussion	5 days	Mon 25/11/19	Fri 29/11/19		Barry,Kieran,Nick,Soham,Will
Competitor Research / Writeup	2 days	Mon 02/12/19	Tue 03/12/19	5	Soham
Ethics Research / Writeup	2 days	Mon 02/12/19	Tue 03/12/19	5	Will
Features Research	2 days	Mon 02/12/19	Tue 03/12/19	5	Kieran
Introduction/Objectives	2 days	Mon 02/12/19	Tue 03/12/19	5	Nick
Risk Analysis / Project Management Writeup	2 days	Mon 02/12/19	Tue 03/12/19	5	Barry
Documentation Draft	2 days	Wed 04/12/19	Thu 05/12/19	6,7,8,9,10	Barry,Kieran,Nick,Soham,Will
Deliverable Draft Definition Document	0 days	Fri 06/12/19	Fri 06/12/19	11	
Final Project Definition	41 days	Fri 22/11/19	Fri 17/01/20	3	
Methodology selection	1 day	Fri 10/01/20	Fri 10/01/20		Barry,Kieran,Nick,Soham,Will
Initial Functional Reqirements	1 day	Mon 13/01/20	Mon 13/01/20		Nick
Functional Requirements discussion	1 day	Tue 14/01/20	Tue 14/01/20	16	Barry[25%],Kieran,Soham,Will
Gantt Chart	2 days	Tue 14/01/20	Wed 15/01/20	15,16,17SS	Barry[75%]
Deliverables / Milestones	1 day	Thu 16/01/20	Thu 16/01/20	18	Barry,Nick
Finalise Documentation	1 day	Thu 16/01/20	Thu 16/01/20		Kieran,Soham,Will
Deliverable Final Project Definition	0 days	Fri 17/01/20	Fri 17/01/20		

Figure 1: Partial screenshot of the Gantt chart

4.2 Concept Map

The first task undertaken during the design phase of the project was the creation of a concept map. A concept map is a diagram that links all the ideas of the project together. By creating one allowed the team to visualise and consider the task required to fit the project together. The map was broken into 4 major categories [Planning, Analysis, Implementation, Design] each having their own subcategories to flesh out the tasks.

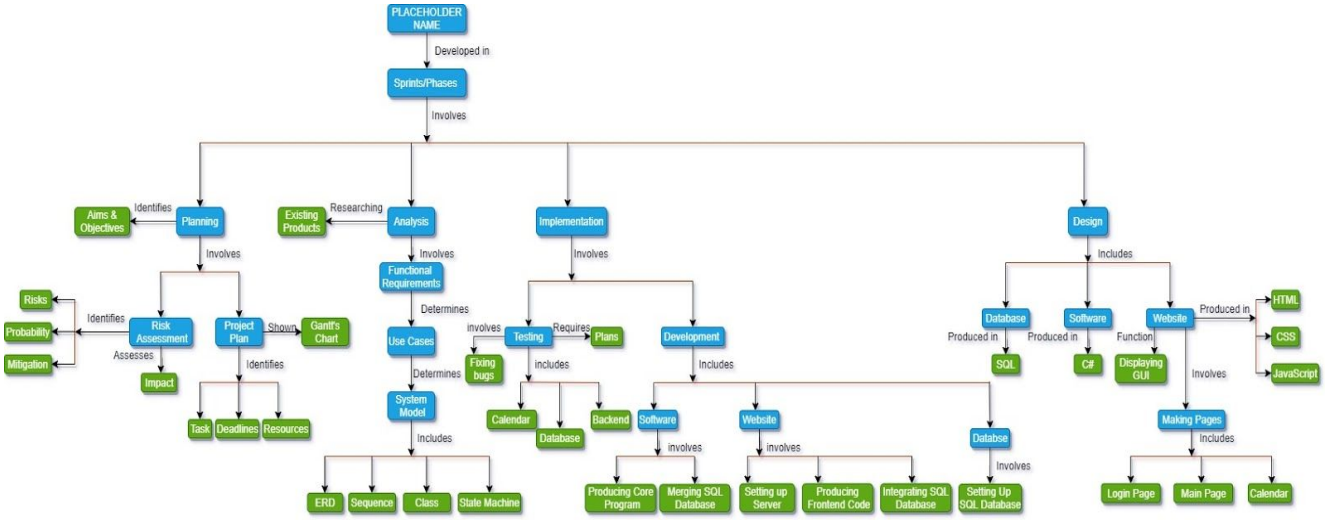


Figure 2: Concept map depicting the various concepts within the project

4.3 Use Case Analysis

4.3.1 Use Case Diagram

A use case diagram and several use cases were decided upon initially. This diagram shows that the Administrator role can manage users and holiday requests, while the User role and the Administrator role can manage events and request holidays. An additional, automatic event occurs allowing for the sending of reminder emails.

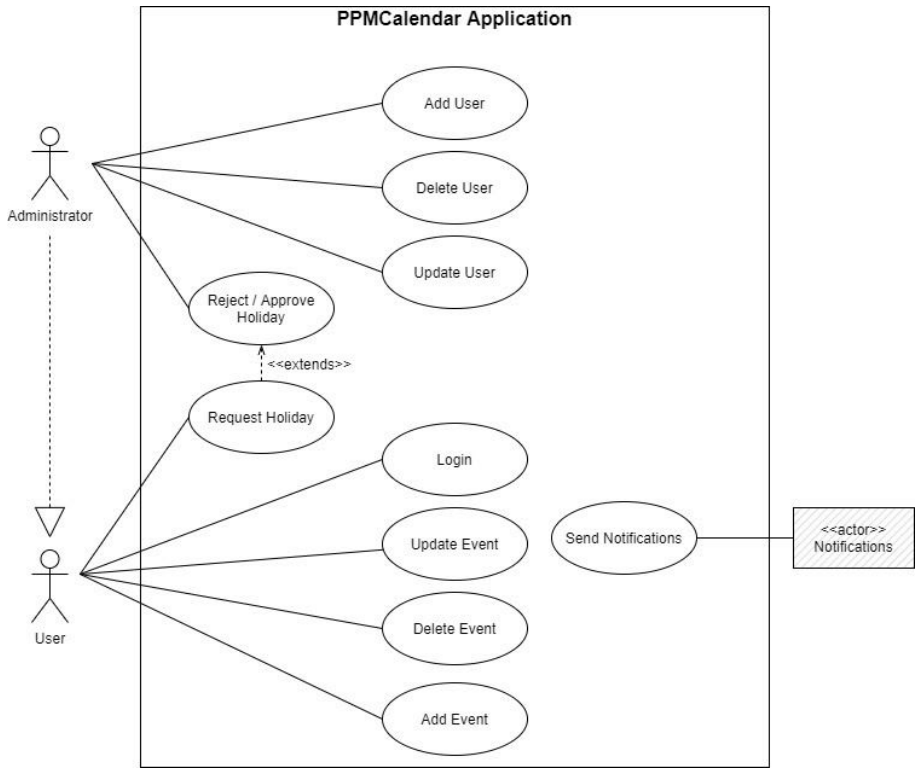


Figure 3: Use case diagram for the software

4.3.2 Use Cases

The following use case details the process of logging a user into the system, this use case was used to create the sequence diagram in the following section. The full set of use cases are available in Appendix B.

<table>
<tr><td colspan="3">Use Case Name: Login</td><td>ID Number:1</td></tr>
<tr><td colspan="4">Short Description: The process of logging into the calendar as completed by a user or an administrator. The login information is sent to the UserController which creates an instance of DBManager and queries the database, returning information regarding the success of the operation to the user.</td></tr>
<tr><td>Trigger:</td><td colspan="3">A user loads the site</td></tr>
<tr><td>Type:</td><td colspan="3">Internal</td></tr>
<tr><td>Major Inputs:
Username and password</td><td>Source:
Login Form</td><td>Major Outputs:
The success or failure of the operation is displayed and the user redirected to the main calendar on success</td><td>Destination:
Login Form or Calendar</td></tr>
<tr><td colspan="3">Major Steps Performed: The user loads the website, which displays the login page. The user fills in their username and password and submits the site. The data is posted to the UserCOntroller part of the API which creates an object of type DBManager to handle the query. The results of this are returned via the event manager to the user.</td><td>Information Required:
Username and password of the user</td></tr>
</table>

4.4 Sequence Diagrams

A set of sequence diagrams was created to show the interaction of objects in the proposed software and the messages exchanged between them during this interaction. Several diagrams were created and for brevity a single diagram demonstrating the login process, the full set of diagrams can be viewed in Appendix C.

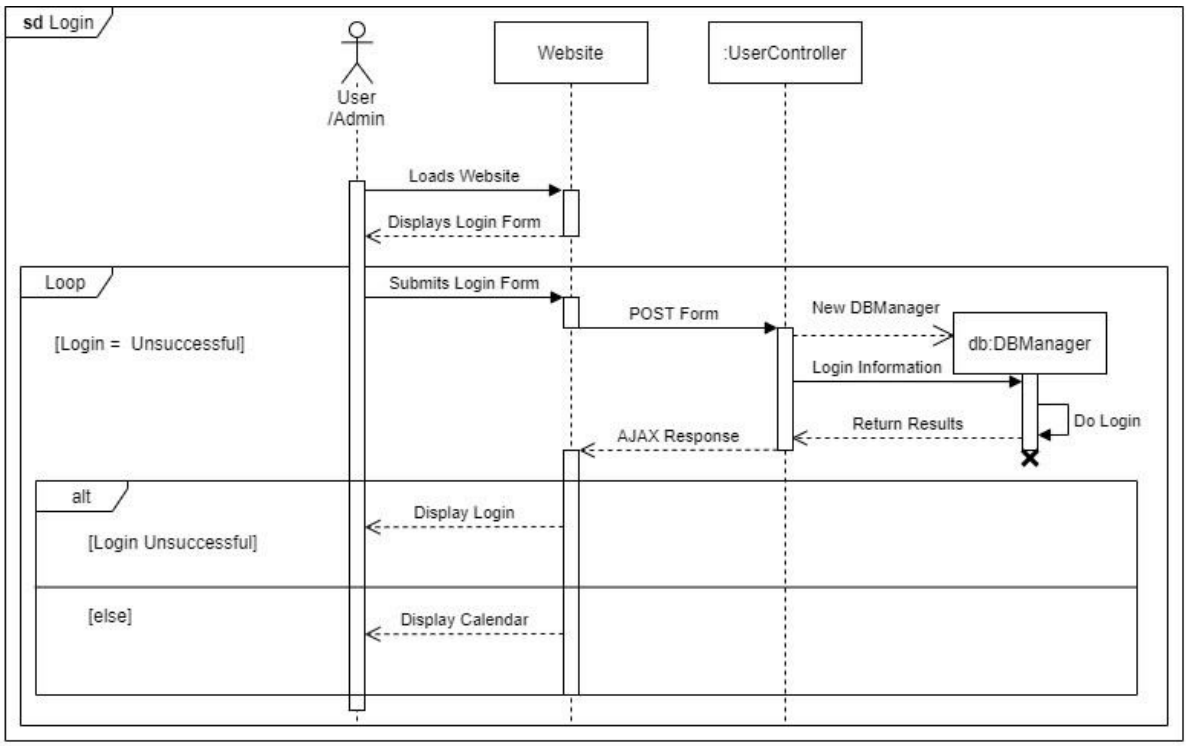


Figure 4: sequence diagram representing the login use case.

The above diagram shows the sequence diagram for the use case of a user or administrator logging into the system.

4.5 Class Diagram

The team created a class diagram to help document the classes that would be required by the software and the relationships between the classes. Using classes in the software allows for a modular approach to the software and helps to keep distinct areas of the software and functionality separate.

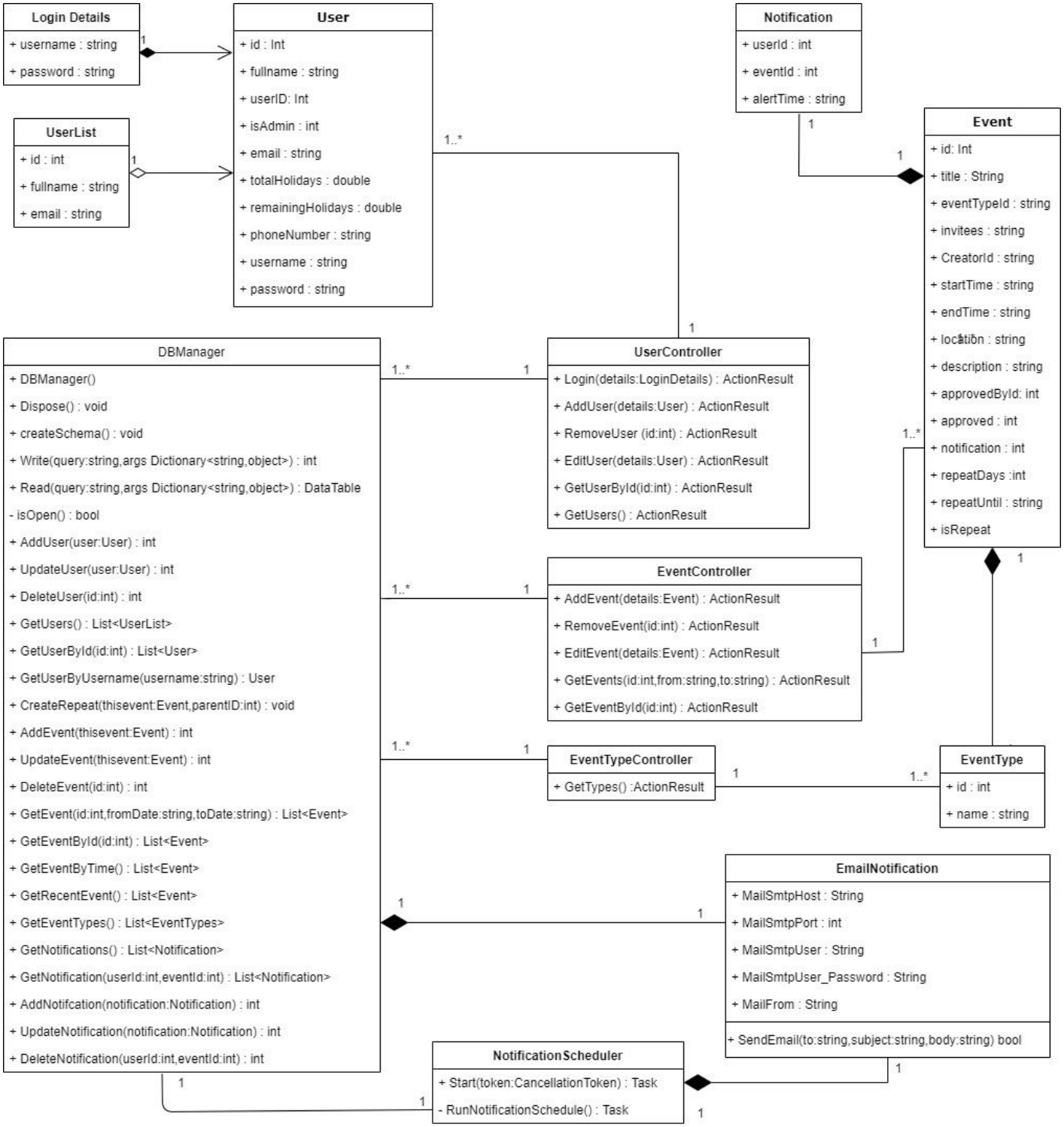


Figure 5: class diagram of all the classes in the software

4.6 Entity Relationship Diagram

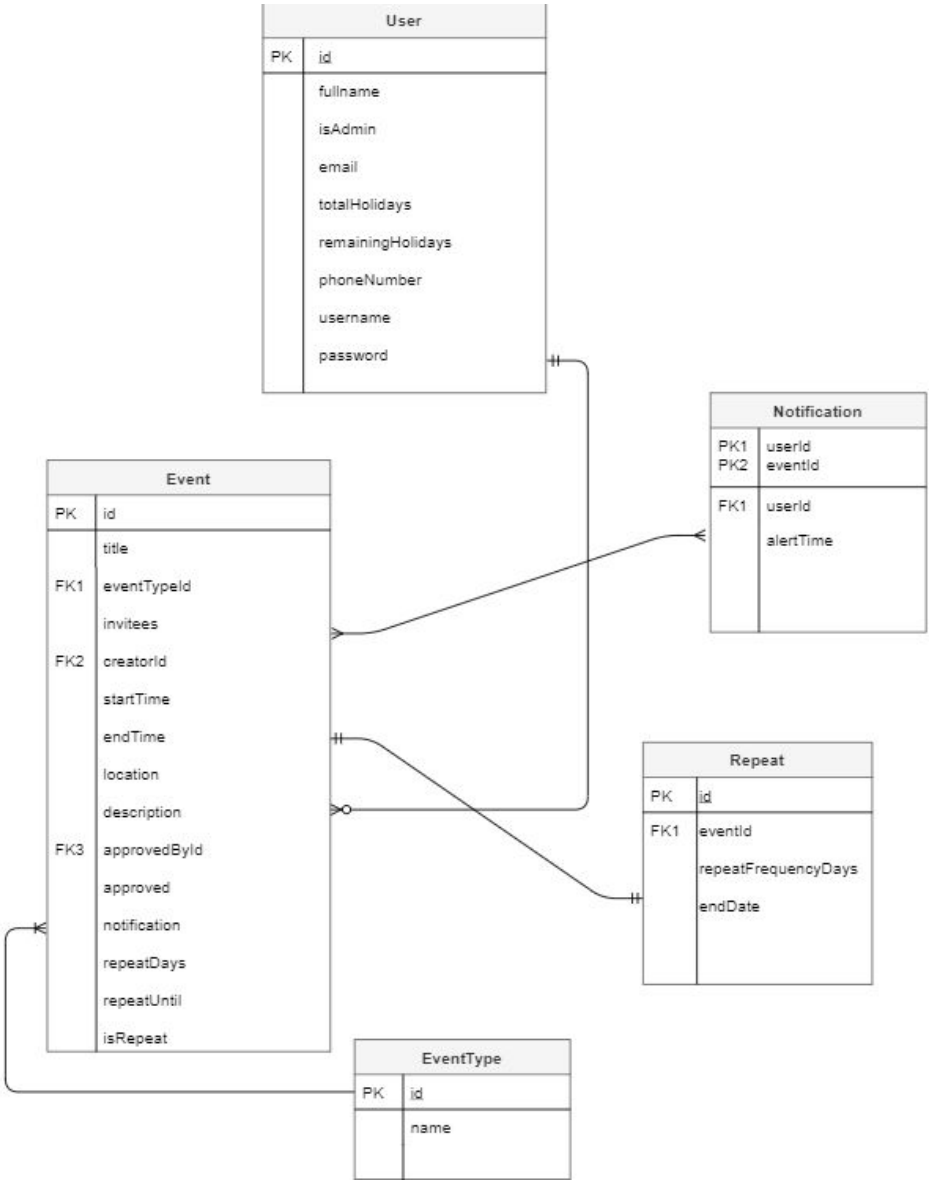


Figure 6: Entity Relationship Diagram for the project

The entity relationship diagram above describes the database and how all of the tables relate with one another. It was important that a strict design and implementation plan was constructed and stuck to, as the database is a core component to the functioning of our website. The Event table holds all events which have been created by users, sharing a foreign key with EventType (Many-to-one) to specify whether the Event is a holiday, meeting, sick day etc. Another foreign key Event has, is with Repeat (One-to-one). Whenever an event is created, it can be repeated on a set interval and up until a specific date. Notification has a foreign key to the Event table (Many-to-many) so that the event details can be retrieved, along with the Users (Many-to-many) participating in the event for emailing/reminder purposes.

4.7 Team Meetings

Team meetings were organised and attended on a weekly basis during this project and details of these can be found in Appendix F. The team engaged well with these throughout the project and all cases of absence were communicated to the Project Manager in a timely fashion. After the Covid-19 outbreak, the team switched to performing online meetings and using Slack or Microsoft Teams to discuss any items or questions. This worked well as a method of communicating during the final weeks of the project.

5 Evaluation of Project

The software produced has met with the functional requirements defined at the start of the project and meets the non functional requirements too This can be demonstrated by the test results produced in Appendix E.

In terms of objectives, the software performs well and meets all requirements.

- The design of the website must be user friendly.
- The UI design must be intuitive, professional looking and similar to other calendar applications.

The final design for the project is comparable to several mainstream calendar applications and has similar methods of interaction with the user, giving a user the ability to intuitively understand how to use the core features of the calendar.

- The website must be responsive and display well on all screen sizes.

As seen in the tests performed, the website displays well on many screen sizes and devices.

- The website must conform to accessibility standards.

The use of high contrast colour schemes, links to skip menu content, ARIA definitions of all dates and events on the calendar and use of keyboard controls have allowed the calendar to remain accessible. It should be mentioned that while the calendar is accessible it may be tedious for keyboard users to move through the calendar using the tab key, especially in the Monthly view. Unfortunately this is the nature of calendars and such users may find the Daily views more keyboard friendly due to the reduced amount of content.

- The website must consist of a secure server which stores user data in a database and secure communication to the frontend.

The database and API are hosted on a Microsoft Azure server, securing the data in a safe environment.

- The user should be able to create, edit and delete events from the calendar.

The user is able to complete these tasks with only a few clicks or keystrokes and suitable notifications and messages are displayed to the user during the process to keep them aware of progress.

- Users should be able to log into their account and have an up-to-date visualisation of events and/or booking they are directly or indirectly involved with

Each user has the ability to log into their account using a password-protected login. Once logged in they will have access to an up-to-date visualisation and summary of all events they are associated with

- Events should be edited and created by opening a form with the required fields.

This objective has been met and a popup form containing the event fields is employed.

- The form should be validated in a user friendly way with descriptive messages.

The form has custom javascript validation routines and each control has the ability to display an error message individually, allowing for a very clear method of indicating issues with the information supplied.

- The website should provide notifications to all invitees of an event when the event is created, edited or deleted.

As discussed in section 6, desktop notifications delivered via the website itself proved to be unreliable due to needing user permission to run. However the accompanying email notifications were successfully completed and can be seen in action in the final software.

- There must be the ability to set an event as recurring.

This functionality was successfully added and a user can choose from various lengths of time to customise when they need the duration between the recurring events to be. This is possible while creating the event and is in line with similar calendar features in competitor products.

- The site should provide users with notifications for upcoming events.

A Notification Scheduler was developed to handle this aspect and this successfully automates all reminder emails and sends them to all users involved with an event at the indicated time before an event.

- Events should contain information about date and time, invitees, description of the event and perhaps a location.

This feature exists as a text version, allowing the user to make notes or add an address. As discussed in section 6, this could be expanded to include Google Maps integration however there was not enough time to consider this as part of the project.

- The site should allow businesses to assign administrative rights to managers within the company.

Administrators can add or even elevate users to Administrator roles simply by updating the relevant fields in the user data. Once this is completed the new Administrator will have administrator privileges however this can also be revoked as needed.

- The site should allow users to request holidays from their manager, these requests will be forwarded to Administrators for approval.
- The site should allow Administrators to approve holidays from their staff, who will then be notified.

This was one of the key features of the website and was part of the core functionality for the website. Both of these objectives were met and it is a simple process to submit or approve/reject these requests. Additional functionality was added to adjust the amount of holidays based upon this.

- The site should allow Administrators to manage user account details.

Administrators have the ability to access user administration functionality via a menu specific to this purpose which is only available to users with Administrator privileges. This menu offers options for adding, deleting and updating a user, making the process straightforward.

- The site should allow Administrators to create permanent/priority events.

Administrators gain all the functionality of a basic user so they are able to easily create events as needed.

5.1 Testing

The team produced a suite of tests as part of a testing strategy to rigorously test the final software. A test plan was created initially to oversee the process, allowing each member to add tests as functionality was added to the project. When the software was finalised, this existing plan was used to allow team members to perform tests on the software and the results of these tests can be seen in Appendix E. Appendix E also contains evidence supporting the tests in the form of images and video, these are arranged with the relevant test table to give a clear overview of the process and results.

In some cases the tests did not complete successfully initially and this allowed the team to identify several minor bugs in the system which were addressed as part of the process. Once these issues were addressed, these tests passed and the testing phase was able to be completed successfully.

6 Discussion / Conclusion

This project has offered the members of the team a great deal of experience in new languages and approaches to software that do not follow the C++ paradigm. Most members within the team have only used HTML as part of basic website building, so this project helped in understanding the possibilities of using the web medium alongside a language like C# to create a more dynamic application. Alongside this, the team have learned about elements of accessibility and responsiveness and the importance of those in a legal and ethical sense in software development.

The team had learned a new language, C#, which only one member of the team previously had experience in. This is a language which is extremely popular among software development companies and, having used it for 5 months of development, the team can use that to their advantage in future interviews.

The experience gained from working with colleagues with industry experience was valuable. Seeing how professional developers work and learning from them during the process helped the team get a feel for what professional projects might be like. This also allowed the team to have experienced people in management positions, allowing for a strong, solid leadership approach and the ability to plan ahead and solve problems ensured that the project progressed steadily and remained on time.

It should be pointed out that the team experienced issues with Zach Russel not engaging with the project at all. His presence on Slack, Email and in project meetings was non existent, as was his contribution to the production of this report or the underlying software produced. However the team handled this well and adjusted workload and roles to cover for this eventuality.

During the process of researching, planning and developing this software, the team have identified several areas that could potentially be used to further develop the software. For example, the integration of the software with other communication applications like Slack were present in competitor solutions and with the extensive use of Slack in companies, the team felt this would be a great selling point.

The ability to import/export and integrate with existing calendars like Google Calendar, Outlook or iCal was something the team wished to include from the beginning. Unfortunately this was not possible due to time constraints but it remains a strong possibility for future development.

Several other non essential features like the ability to search for an event using a search bar or the ability to customise the interface to the user's liking were discussed from an early stage although these didn't make the final software, these may be interesting additions.

The team discussed the ability to link the location of an event to Google Maps, allowing for additional features like a route planner, distance calculator and GPS directions to your meeting. Unfortunately this functionality was too complex to include due to the time constraints. However this would open up many possibilities for expansion and should be considered for further development.

Moving outside the sphere of web entirely, it would be possible to extend the use of this application and provide mobile apps which would have the benefit of being able to reliably use notifications. While it is possible to do this with current browser technology, this involves several stages of user approval and the team felt that although this functionality was really useful, the added complexity of this just made it likely that many users would not be able to access the features.

7 Professional, Social, Ethical and Legal issues

There are many professional, social, ethical and legal issues related to designing and developing a piece of software which the team must take into consideration during this process. The following sections discuss each in detail, however there is some overlap between these sections as some areas such as accessibility fall into more than one category. Where possible, an attempt to discuss only the relationship between the issue and the current section is made.

7.1 Professional

The British Computer Society (BCS) Code of Conduct establishes a list of guidelines that the British computer industry and members of the BCS need to follow. The code states that members should: "develop your professional knowledge, skills and competence on a continuing basis, maintaining awareness of technological developments, procedures, and standards that are relevant to your field." (British Computer Society, 2019). The team will be learning new skills and languages during the project including (C#, .NET Core and SQLite) and will therefore be fulfilling this guideline.

It goes on to say that members should "NOT claim any level of competence that you do not possess." (British Computer Society, 2019) and that members should "have due regard for the legitimate rights of Third Parties" (British Computer Society, 2019). The team will create, where possible, all code used on the project and where any third-party code or library is used, relevant attribution and reference to the authors within the code and documentation will exist.

The code also states that members must "respect and value alternative viewpoints and, seek, accept and offer honest criticisms of work."(British Computer Society, 2019). The team will employ this at all levels throughout the project, listening to all viewpoints and coming to a compromise that is acceptable should a difference of opinion occur. The feedback and criticisms offered in meetings with the Tutor will be taken on board and, wherever possible, acted upon.

Finally, the code suggests that we should "accept professional responsibility for your work and for the work of colleagues who are defined in a given context as working under your supervision." (British Computer Society, 2019). All team members will have responsibility for themselves and to the team as a whole to ensure that the project stays on track and deadlines can be reached. This might mean helping others after an absence or sharing skills with members who need some training for their part in the project. The team management has been chosen in a way that allows for those with the greatest experience to be in a position to oversee and teach the team.

7.2 Social

The code states that "conduct your professional activities without discrimination on the grounds of sex, sexual orientation, marital status, nationality, colour, race, ethnic origin, religion, age or disability, or of any other condition or requirement" (British Computer Society, 2019). This can be achieved by ensuring we have as much of a diverse team as possible and by respecting each member of the team. This can also be applied to the final product by ensuring that any language and imagery used is neutral since different cultures have different standards and failure to do so may limit the success of the project among certain demographics.

It also states that members should “promote equal access to the benefits of IT and seek to promote the inclusion of all sectors in society wherever opportunities arise” (British Computer Society, 2019). This can be satisfied by using best practices to ensure as diverse an audience as possible, making sure it isn’t exclusive to any specific category of individuals. Adhering to the W3C guidelines on accessibility (W3C, 2019) will allow for users with vision or motor control issues to be able to use the website, increasing the inclusivity of the software.

7.3 Ethical

The ethical implications of the project are strongly echoed in the other three sections. These include adhering to legal requirements like the Data Protection Act 2018 (Department for Digital, Culture, Media & Sport, 2020) and GDPR (GDPR, 2009) with regard to user data and privacy and legislation regarding discrimination. Further to that, not claiming ownership of third party work and honestly declaring the limit of our technical expertise is a good ethical standpoint.

Again, it would be unethical to restrict access to the website to only those with the full range of abilities. Making the website accessible to all, including those who cannot use a mouse, have impaired vision or use a screen reader is of utmost importance.

In terms of the team, it is ethical for us to be responsible for our own actions and collectively for those of the tem, ensuring that we help other members when appropriate and take on any responsibilities needed by the team to keep the project progressing. Failure to engage with the project in this way would result in a lower mark, undermining the work of other members of the team. All members of the team and the team itself should aim to be open, honest and transparent in their approach to the project, the team and client/user feedback. This would help get the most out of the project and the transparency would help gain trust from any potential user base.

The use of the final product could also potentially help with environmental issues. It is hoped that the final software will make it easier to book and keep track of meetings, possibly saving on printing to paper and reducing the amount of email and telephone communication required to arrange a holiday or meeting.

7.4 Legal

The Code of Conduct states that members must “ensure that you have the knowledge and understanding of Legislation and that you comply with such Legislation, in carrying out your professional responsibilities.” (British Computer Society, 2019). The team must ensure that we meet all the legal requirements pertaining to this project.

Following the Data Protect Act 2018 (Department for Digital, Culture, Media & Sport, 2019), GDPR legislation (GDPR, 2009), and advice on what information a company can legally hold about an employee (Department for Work and Pensions, 2020) we will make sure the data is processed fairly and lawfully, only obtained through lawful purposes, relevant, not excessive to the purpose, kept accurate and up to date, and not kept for longer than is necessary. As well as this, the data will only be kept within the business software utilised it. Technical and organisation measures will be followed to ensure the data won’t be accidentally lost, destroyed, or damaged.

Accessibility requirements for public sector websites (Government Digital Service, 2020), provides advice for public websites, allowing them to conform to laws within the Equality Act 2010 (or the

Disability Discrimination Act 1995 in Northern Ireland) regarding the legal requirement that “reasonable adjustments” are made to accommodate the needs of disabled users. The team will make the calendar adhere to these guidelines and will include the Accessible Rich Internet Applications suite (ARIA) (Web Accessibility Initiative, 2016) in achieving this.

Bibliography

British Computer Society, 2019. *BCS Code of Conduct* [online]. Available at: https://www.bcs.org/membership/become-a-member/bcs-code-of-conduct/ [Accessed 30 April, 2020].

Department For Digital, Culture, Media & Sport, 2019. *Data protection - GOV.UK* [online]. Available at: https://www.gov.uk/data-protection [Accessed Apr 30, 2020].

Department For Work And Pensions, 2020. *Personal data an employer can keep about an employee* [online]. Available at: https://www.gov.uk/personal-data-my-employer-can-keep-about-me [Accessed Apr 30, 2020].

Gartner, Inc., 2019. *4 Key Trends in the Gartner Hype Cycle for Human Capital Management Technology, 2019* [online]. Available at: http://www.gartner.com/smarterwithgartner/4-key-trends-gartner-hype-cycle-human-capital-management-technology-2019/ [Accessed May 2, 2020].

GDPR, 2009. *Cookies, the GDPR and the ePrivacy Directive* [online]. Available at: https://gdpr.eu/cookies/ [Accessed 30 April, 2020].

Government Digital Service, 2020. *Understanding accessibility requirements for public sector bodies* [online]. Available at: https://www.gov.uk/guidance/accessibility-requirements-for-public-sector-websites-and-apps [Accessed Apr 30, 2020].

Research and Markets, 2019. *Global Human Capital Management Market by Software, Service, Deployment Type, Organization Size & Region - Forecast to 2024* [online]. Available at: https://www.researchandmarkets.com/r/p3let1 [Accessed 30th April, 2020].

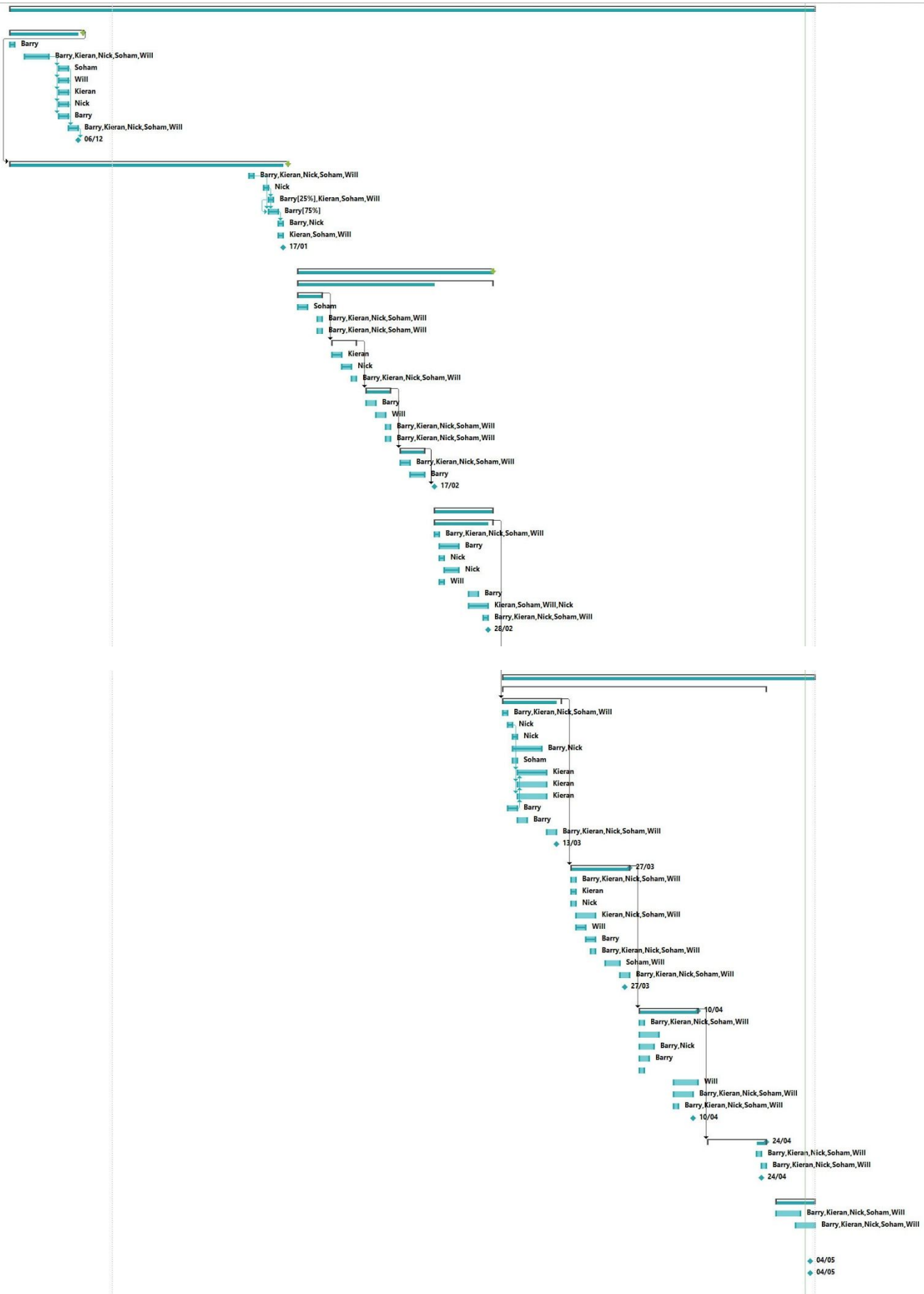
W3C, 2019. *Accessibility Principles* [online]. Available at: https://www.w3.org/WAI/fundamentals/accessibility-principles/ [Accessed Apr 30, 2020].

Web Accessibility Initiative, 2016. *WAI-ARIA Overview* [online]. Available at: https://www.w3.org/WAI/standards-guidelines/aria/ [Accessed May 3, 2020].

Appendix A - Gantt Chart

Calendar Software Project	117 days	Fri 22/11/19	Mon 04/05/20		
Draft Definition Document	11 days	Fri 22/11/19	Fri 06/12/19		
Slack / Google Docs Setup	1 day	Fri 22/11/19	Fri 22/11/19		Barry
Project Idea Discussion	5 days	Mon 25/11/19	Fri 29/11/19		Barry,Kieran,Nick,Soham,Will
Competitor Research / Writeup	2 days	Mon 02/12/19	Tue 03/12/19	5	Soham
Ethics Research / Writeup	2 days	Mon 02/12/19	Tue 03/12/19	5	Will
Features Research	2 days	Mon 02/12/19	Tue 03/12/19	5	Kieran
Introduction/Objectives	2 days	Mon 02/12/19	Tue 03/12/19	5	Nick
Risk Analysis / Project Management Writeup	2 days	Mon 02/12/19	Tue 03/12/19	5	Barry
Documentation Draft	2 days	Wed 04/12/19	Thu 05/12/19	6,7,8,9,10	Barry,Kieran,Nick,Soham,Will
Deliverable Draft Definition Document	0 days	Fri 06/12/19	Fri 06/12/19	11	
Final Project Definition	41 days	Fri 22/11/19	Fri 17/01/20	3	
Methodology selection	1 day	Fri 10/01/20	Fri 10/01/20		Barry,Kieran,Nick,Soham,Will
Initial Functional Reqirements	1 day	Mon 13/01/20	Mon 13/01/20		Nick
Functional Requirements discussion	1 day	Tue 14/01/20	Tue 14/01/20	16	Barry[25%],Kieran,Soham,Will
Gantt Chart	2 days	Tue 14/01/20	Wed 15/01/20	15,16,17SS	Barry[75%]
Deliverables / Milestones	1 day	Thu 16/01/20	Thu 16/01/20	18	Barry,Nick
Finalise Documentation	1 day	Thu 16/01/20	Thu 16/01/20		Kieran,Soham,Will
Deliverable Final Project Definition	0 days	Fri 17/01/20	Fri 17/01/20		
Interim Review - Software Demo	30 days	Mon 20/01/20	Fri 28/02/20		
Design	30 days	Mon 20/01/20	Fri 28/02/20		
Week 1	5 days	Mon 20/01/20	Fri 24/01/20		
Concept Map	2 days	Mon 20/01/20	Tue 21/01/20		Soham
Concept Map	1 day	Fri 24/01/20	Fri 24/01/20		Barry,Kieran,Nick,Soham,Will
Deliverable Review of Previous Work	1 day	Fri 24/01/20	Fri 24/01/20		Barry,Kieran,Nick,Soham,Will
Week 2	5 days	Mon 27/01/20	Fri 31/01/20	25	
Class Diagram	2 days	Mon 27/01/20	Tue 28/01/20		Kieran
ERD	2 days	Wed 29/01/20	Thu 30/01/20		Nick
Class/ERD Finalising	1 day	Fri 31/01/20	Fri 31/01/20		Barry,Kieran,Nick,Soham,Will
Week 3	5 days	Mon 03/02/20	Fri 07/02/20	29	
Sequence Diagram	2 days	Mon 03/02/20	Tue 04/02/20		Barry
Use Cases / Use Case DIagram	2 days	Wed 05/02/20	Thu 06/02/20		Will
Sequence/Use Cases Finalising	1 day	Fri 07/02/20	Fri 07/02/20		Barry,Kieran,Nick,Soham,Will
Deliverable Progress Review / Software Demo	1 day	Fri 07/02/20	Fri 07/02/20		Barry,Kieran,Nick,Soham,Will
Week 4	5 days	Mon 10/02/20	Fri 14/02/20	33	
Diagram Overrun/Finalising	2 days	Mon 10/02/20	Tue 11/02/20		Barry,Kieran,Nick,Soham,Will
Visual Design	3 days	Wed 12/02/20	Fri 14/02/20		Barry
Design Phase Complete	0 days	Mon 17/02/20	Mon 17/02/20	38	
Software Development	10 days	Mon 17/02/20	Fri 28/02/20		
Sprint 1	10 days	Mon 17/02/20	Fri 28/02/20		
Scrum Meeting	1 day	Mon 17/02/20	Mon 17/02/20		Barry,Kieran,Nick,Soham,Will
Core Calendar HTML / CSS	4 days	Tue 18/02/20	Fri 21/02/20		Barry
Database Creation	1 day	Tue 18/02/20	Tue 18/02/20		Nick
C# training preparation	3 days	Wed 19/02/20	Fri 21/02/20		Nick
Login Screen	1 day	Tue 18/02/20	Tue 18/02/20		Will
Responsive / Accessible HTML	2 days	Mon 24/02/20	Tue 25/02/20		Barry
C# training	4 days	Mon 24/02/20	Thu 27/02/20		Kieran,Soham,Will,Nick
Sprint Code Testing	1 day	Thu 27/02/20	Thu 27/02/20		Barry,Kieran,Nick,Soham,Will
Deliverable Interim Software Review/Demo	0 days	Fri 28/02/20	Fri 28/02/20		

Final Report& Software Submission	**46 days**	**Mon 02/03/20**	**Mon 04/05/20**		
Software Development	**40 days**	**Mon 02/03/20**	**Fri 24/04/20**		
Sprint 2	**10 days**	**Mon 02/03/20**	**Fri 13/03/20**	**44**	
Scrum Meeting	1 day	Mon 02/03/20	Mon 02/03/20		Barry,Kieran,Nick,Soham,Will
Set up Server side project and repo	1 day	Tue 03/03/20	Tue 03/03/20		Nick
Database connections	1 day	Wed 04/03/20	Wed 04/03/20		Nick
Login/Logout functionality	4 days	Wed 04/03/20	Mon 09/03/20		Barry,Nick
SQL Data Generation	1 day	Wed 04/03/20	Wed 04/03/20		Soham
Add an Event	4 days	Thu 05/03/20	Tue 10/03/20	59,66	Kieran
Edit an event	4 days	Thu 05/03/20	Tue 10/03/20	59,66	Kieran
Delete an event	4 days	Thu 05/03/20	Tue 10/03/20	59,66	Kieran
Event HTML Forms	2 days	Tue 03/03/20	Wed 04/03/20		Barry
Event Types display	2 days	Thu 05/03/20	Fri 06/03/20		Barry
Sprint Code Testing	2 days	Wed 11/03/20	Thu 12/03/20		Barry,Kieran,Nick,Soham,Will
<Milestone> Team Software Review	0 days	Fri 13/03/20	Fri 13/03/20		Barry,Kieran,Nick,Soham,Will
Sprint 3	**10 days**	**Mon 16/03/20**	**Fri 27/03/20**	**57**	
Scrum Meeting	1 day	Mon 16/03/20	Mon 16/03/20		Barry,Kieran,Nick,Soham,Will
Event Add/Edit/Delete	1 day	Mon 16/03/20	Mon 16/03/20		Kieran
User Add/Edit/Delete	1 day	Mon 16/03/20	Mon 16/03/20		Nick
User / Event / Settings classes	4 days	Tue 17/03/20	Fri 20/03/20		Kieran,Nick,Soham,Will
HTML user pages and forms	2 days	Tue 17/03/20	Wed 18/03/20		Will
Calendar Navigation	2 days	Thu 19/03/20	Fri 20/03/20		Barry
****Deliverable** Progress Review/Software Demo**	**1 day**	**Fri 20/03/20**	**Fri 20/03/20**		Barry,Kieran,Nick,Soham,Will
Email/Notification functionality	3 days	Mon 23/03/20	Wed 25/03/20		Soham,Will
Sprint Code Testing	2 days	Thu 26/03/20	Fri 27/03/20		Barry,Kieran,Nick,Soham,Will
<Milestone> Team Software Review	0 days	Fri 27/03/20	Fri 27/03/20		Barry,Kieran,Nick,Soham,Will
Sprint 4	**10 days**	**Mon 30/03/20**	**Fri 10/04/20**	**71**	
Scrum Meeting	1 day	Mon 30/03/20	Mon 30/03/20		Barry,Kieran,Nick,Soham,Will
Import/Export functionality (abandoned)	4 days	Mon 30/03/20	Thu 02/04/20		
Recurring Events	3 days	Mon 30/03/20	Wed 01/04/20		Barry,Nick
Admin holiday approval	2 days	Mon 30/03/20	Tue 31/03/20		Barry
Language filter (abandoned)	1 day	Mon 30/03/20	Mon 30/03/20		
Test plan & Tables	5 days	Mon 06/04/20	Fri 10/04/20		Will
Coding overrun time	4 days	Mon 06/04/20	Thu 09/04/20		Barry,Kieran,Nick,Soham,Will
Sprint Testing	1 day	Mon 06/04/20	Mon 06/04/20		Barry,Kieran,Nick,Soham,Will
<Milestone> Team Software Review	0 days	Fri 10/04/20	Fri 10/04/20		Barry,Kieran,Nick,Soham,Will
Sprint 5	**10 days**	**Mon 13/04/20**	**Fri 24/04/20**	**83**	
Sprint Testing	1 day	Thu 23/04/20	Thu 23/04/20		Barry,Kieran,Nick,Soham,Will
<Milestone> Team Software Review	1 day	Fri 24/04/20	Fri 24/04/20		Barry,Kieran,Nick,Soham,Will
****Deliverable** Software Development Complete**	0 days	Fri 24/04/20	Fri 24/04/20		
Documentation	**6 days**	**Mon 27/04/20**	**Mon 04/05/20**		
Finalisation of documentation	5 days	Mon 27/04/20	Fri 01/05/20		Barry,Kieran,Nick,Soham,Will
****Deliverable** Documentation Complete**	2 days	Fri 01/05/20	Mon 04/05/20		Barry,Kieran,Nick,Soham,Will
Project Presentation	0 days	Mon 04/05/20	Mon 04/05/20		Barry,Kieran,Nick,Soham,Will
Software Demo	0 days	**Mon 04/05/20**	Mon 04/05/20		Barry,Kieran,Nick,Soham,Will


Barry
Barry,Kieran,Nick,Soham,Will
Soham
Will
Kieran
Nick
Barry
Barry,Kieran,Nick,Soham,Will
06/12
Barry,Kieran,Nick,Soham,Will
Nick
Barry[25%],Kieran,Soham,Will
Barry[75%]
Barry,Nick
Kieran,Soham,Will
17/01
Soham
Barry,Kieran,Nick,Soham,Will
Barry,Kieran,Nick,Soham,Will
Kieran
Nick
Barry,Kieran,Nick,Soham,Will
Barry
Will
Barry,Kieran,Nick,Soham,Will
Barry,Kieran,Nick,Soham,Will
Barry,Kieran,Nick,Soham,Will
Barry
17/02
Barry,Kieran,Nick,Soham,Will
Barry
Nick
Nick
Will
Barry
Kieran,Soham,Will,Nick
Barry,Kieran,Nick,Soham,Will
28/02
Barry,Kieran,Nick,Soham,Will
Nick
Nick
Barry,Nick
Soham
Kieran
Kieran
Kieran
Barry
Barry
Barry,Kieran,Nick,Soham,Will
13/03
27/03
Barry,Kieran,Nick,Soham,Will
Kieran
Nick
Kieran,Nick,Soham,Will
Will
Barry
Barry,Kieran,Nick,Soham,Will
Soham,Will
Barry,Kieran,Nick,Soham,Will
27/03
10/04
Barry,Kieran,Nick,Soham,Will
Barry,Nick
Barry
Will
Barry,Kieran,Nick,Soham,Will
Barry,Kieran,Nick,Soham,Will
10/04
24/04
Barry,Kieran,Nick,Soham,Will
Barry,Kieran,Nick,Soham,Will
24/04
Barry,Kieran,Nick,Soham,Will
Barry,Kieran,Nick,Soham,Will
04/05
04/05

Appendix B - Use Cases

<table>
<tr><td colspan="3">Use Case Name: Login</td><td>ID Number:1</td></tr>
<tr><td colspan="4">Short Description: The process of logging into the calendar as completed by a user or an administrator. The login information is sent to the UserController which creates an instance of DBManager and queries the database, returning information regarding the success of the operation to the user.</td></tr>
<tr><td>Trigger:</td><td colspan="3">A user loads the site</td></tr>
<tr><td>Type:</td><td colspan="3">Internal</td></tr>
<tr><td>Major Inputs:
Username and password</td><td>Source:
Login Form</td><td>Major Outputs:
The success or failure of the operation is displayed and the user redirected to the main calendar on success</td><td>Destination:
Login Form or Calendar</td></tr>
<tr><td colspan="3">Major Steps Performed: The user loads the website, which displays the login page. The user fills in their username and password and submits the site. The data is posted to the UserController part of the API which creates an object of type DBManager to handle the query. The results of this are returned via the event manager to the user.</td><td>Information Required:
Username and password of the user</td></tr>
</table>

<table>
<tr><td colspan="3">Use Case Name: Add an Event</td><td>ID Number:2</td></tr>
<tr><td colspan="4">Short Description: The process of adding a new event to the calendar.</td></tr>
<tr><td>Trigger:</td><td colspan="3">A user clicks a blank area of the calendar.</td></tr>
<tr><td>Type:</td><td colspan="3">Internal</td></tr>
<tr><td>Major Inputs:
Event information from the user</td><td>Source:
Calendar web page</td><td>Major Outputs:
This process creates a new entry in the events table of the database and notifies the user of success or failure.</td><td>Destination:
Calendar web page</td></tr>
<tr><td colspan="3">Major Steps Performed: The user clicks a blank area of the calendar, opening an event form with default values. The user completes the required fields and clicks on the Create button. This POSTs the information to the EventController which creates an object of type DBManager to insert the information into the database. An object of type EmailNotification is created and used to send an email notification to invitees of the event. The results of this are returned via the event manager to the user.</td><td>Information Required:
Event information</td></tr>
</table>

Use Case Name: Edit an Event			**ID Number**:3
Short Description: The process of updating an existing event within the calendar.			
Trigger:	A user clicks an existing event and edits and submits the data.		
Type:	Internal		
Major Inputs: Event information from the user	**Source**: Calendar web page	**Major Outputs**: This process updates an existing entry in the events table of the database and notifies the user of success or failure.	**Destination**: Calendar web page
Major Steps Performed: The user clicks an existing event on the calendar, opening an event form with existing values. The user updates the required fields and clicks on the Update button. This POSTs the information to the EventController which creates an object of type DBManager to update the information into the database. An object of type EmailNotification is created and used to send an email notification to invitees of the event. The results of this are returned via the event manager to the user.			**Information Required**: Event information

Use Case Name: Delete an Event			**ID Number**:4
Short Description: The process of deleting an existing event from the calendar.			
Trigger:	A user clicks an existing event and deletes and submits the data.		
Type:	Internal		
Major Inputs: Selection of event to delete	**Source**: Calendar web page	**Major Outputs**: This process deletes an existing entry from the events table of the database and notifies the user of success or failure.	**Destination**: Calendar web page
Major Steps Performed: The user clicks an existing event on the calendar, opening an event form with existing values. The user clicks on the Delete button. This POSTs the information to the EventController which creates an object of type DBManager to delete the information into the database. An object of type EmailNotification is created and used to send an email notification to invitees of the event. The results of this are returned via the event manager to the user.			**Information Required**: Event ID

Use Case Name: Add a User	**ID Number**:5
Short Description: The process of adding a new user to the calendar.	

Trigger:	An administrator chooses the option to add a user from the administrator menu.		
Type:	Internal		
Major Inputs: New User Information	**Source**: Calendar web page	**Major Outputs**: This process creates a new entry in the users table of the database and notifies the administrator of success or failure.	**Destination**: Calendar web page
Major Steps Performed: The administrator clicks the option to add a user in the administrator menu, opening a user form with default values. The administrator completes the required fields and clicks on the Create button. This POSTs the information to the UserController which creates an object of type DBManager to insert the information into the database. The results of this are returned via the event manager to the administrator .			**Information Required**: User information

Use Case Name: Update a User			**ID Number**:6
Short Description: The process of updating an existing user within the calendar.			
Trigger:	An administrator chooses the option to edit a user from the administrator menu.		
Type:	Internal		
Major Inputs: Updated User Information	**Source**: Calendar web page	**Major Outputs**: This process updates an existing entry in the users table of the database and notifies the administrator of success or failure.	**Destination**: Calendar web page
Major Steps Performed: The administrator clicks the option to edit a user in the administrator menu, opening a user select screen with existing users. Once a user is selected, a user form with the details of the selected user is opened. The administrator updates the required fields and clicks on the Update button. This POSTs the information to the UserController which creates an object of type DBManager to update the information into the database. The results of this are returned via the event manager to the administrator .			**Information Required**: User information

Use Case Name:Delete a User	**ID Number**:7
Short Description: The process of deleting an existing user within the calendar.	

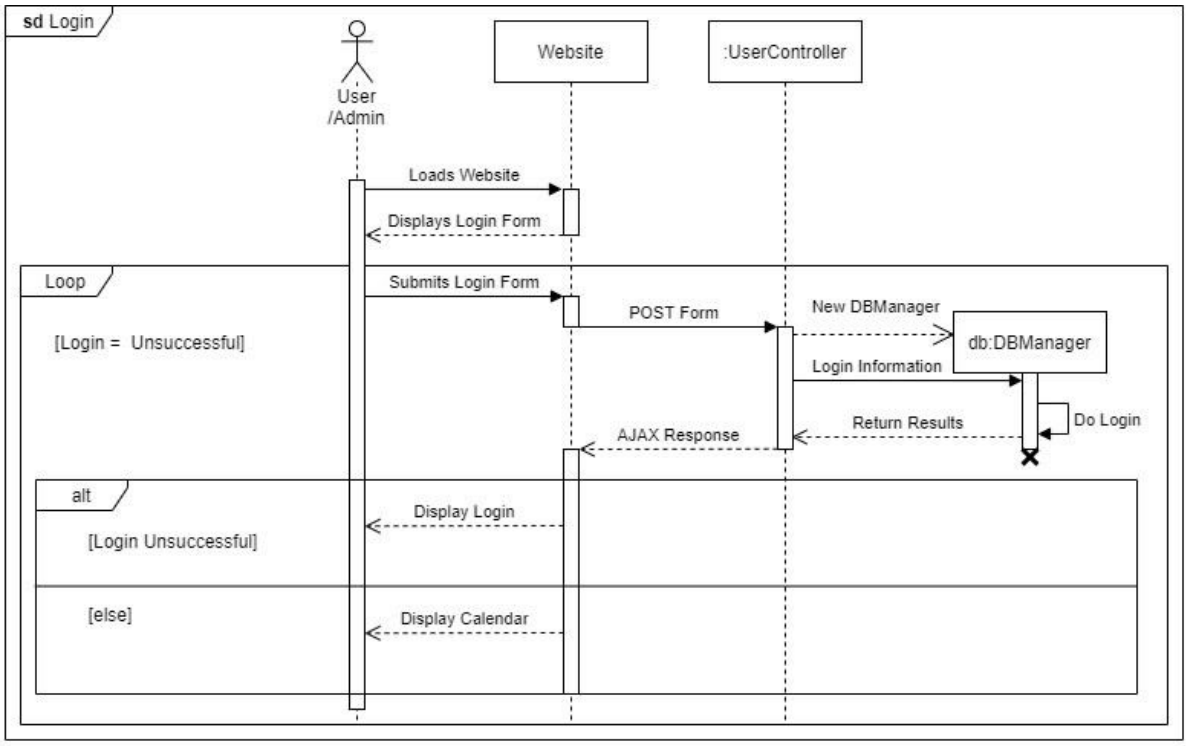
Trigger:	An administrator chooses the option to delete a user from the administrator menu.		
Type:	Internal		
Major Inputs: Selection of user to delete	**Source**: Calendar web page	**Major Outputs**: This process deletes an existing entry in the users table of the database and notifies the administrator of success or failure.	**Destination**: Calendar web page
Major Steps Performed: The administrator clicks the option to delete a user in the administrator menu, opening a user select screen with existing users. Once a user is selected, a user form with the details of the selected user is opened. The administrator clicks on the Delete button. This POSTs the information to the UserController which creates an object of type DBManager to delete the information from the database. The results of this are returned via the event manager to the administrator .			**Information Required**: User Id

Use Case Name: Request a holiday			**ID Number**:8
Short Description: The process of a user requesting a holiday.			
Trigger:	A user clicks a blank area of the calendar to create a new event and chooses an event type of Holiday Request		
Type:	Internal		
Major Inputs: Holiday information from the user	**Source**: Calendar web page	**Major Outputs**: This process creates a new holiday request in the events table of the database and notifies the user of success or failure. A notification is sent to the selected administrator showing that a request has been made.	**Destination**: Calendar web page
Major Steps Performed: The user clicks a blank area of the calendar, opening an event form with default values. The user completes the required fields, selecting Holiday Request as the type, inputs an administrator into the list of invitees and clicks on the Create button. This POSTs the information to the EventController which creates an object of type DBManager to insert the information into the database. An object of type EmailNotification is created and used to send an email notification to the administrator.. The results of this are returned via the event manager to the user.			**Information Required**: Holiday information

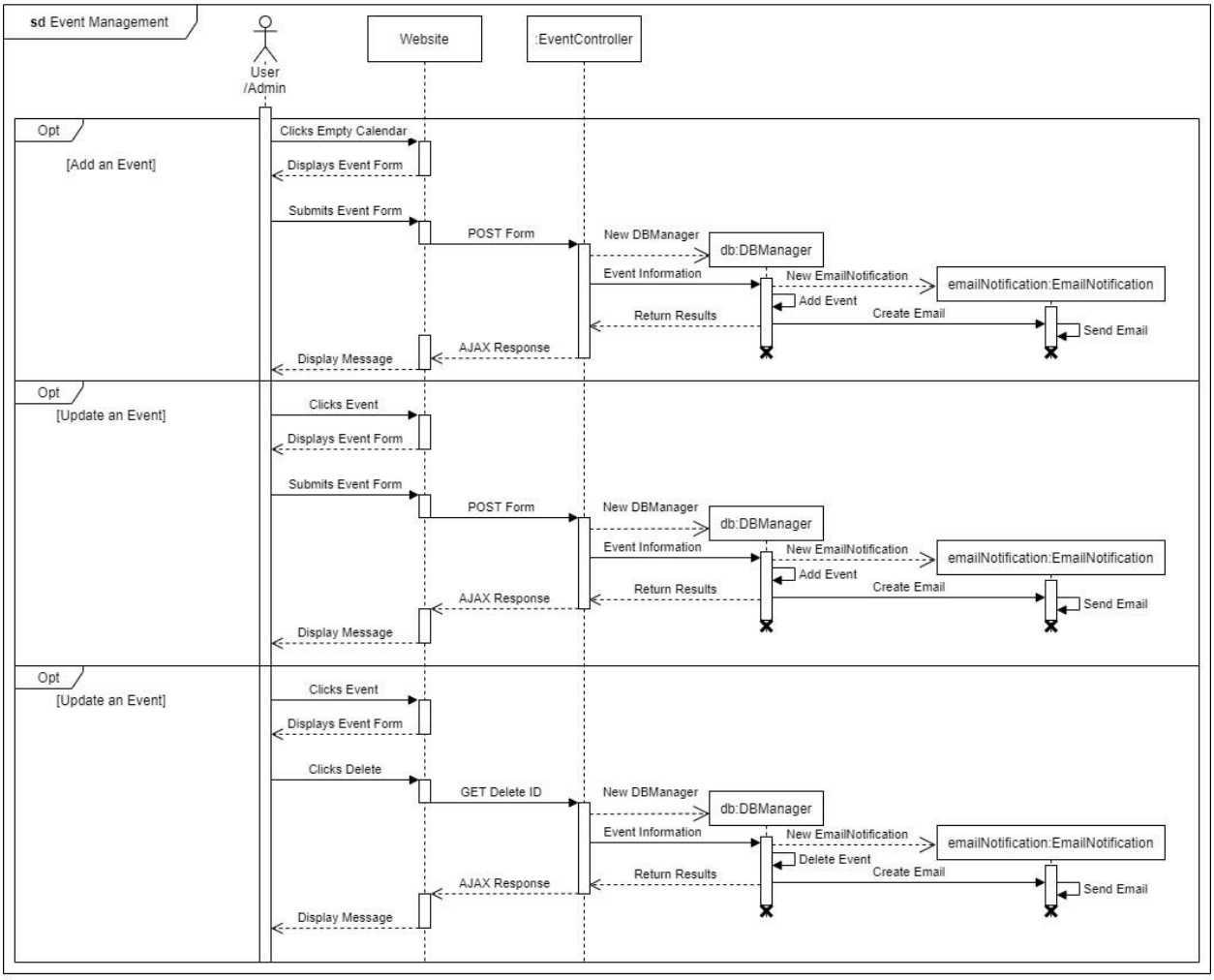
Use Case Name: Approve/Reject a holiday request			**ID Number**:8
Short Description: The process of an administrator approving or rejecting a holiday request.			
Trigger:	An administrator clicks the specific Holiday Request event in the calendar.		
Type:	Internal		
Major Inputs: Holiday approval or rejection status	**Source**: Calendar web page	**Major Outputs**: This process updates an event within the database to approve or reject the event as needed. This changes the type of the event within the database and adjusts the total number of holidays the user has.	**Destination**: Calendar web page
Major Steps Performed: The administrator clicks an existing event which is of type Holiday Request, opening an event form with existing values. The administrator updates the Holiday Approval section of the form, selecting whether to approve or reject the holiday and clicks on the Update button. This POSTs the information to the EventController which creates an object of type DBManager to update the information in the database. An object of type EmailNotification is created and used to send an email notification to the user. The results of this are returned via the event manager to the administrator .			**Information Required**: Holiday approval or rejection status

Use Case Name: Reminder Notifications			**ID Number**:9
Short Description: The process of sending reminder notifications to users automatically.			
Trigger:	This is a timed event which is triggered by a timer.		
Type:	Internal		
Major Inputs: none	**Source**: NotificationScheduler	**Major Outputs**: A reminder email is sent to the user at a chosen time, reminding them that they have an upcoming event	**Destination**: none
Major Steps Performed: The NotificationScheduler has an inbuilt timer which runs every 5 minutes. When this process is triggered, the database is searched for any notifications which have happened in the past 6 minutes (to allow for edge cases). These notifications are then used to generate reminder emails which are sent to the relevant parties and the notifications are removed from the database once they have been dealt with.			**Information Required**: none

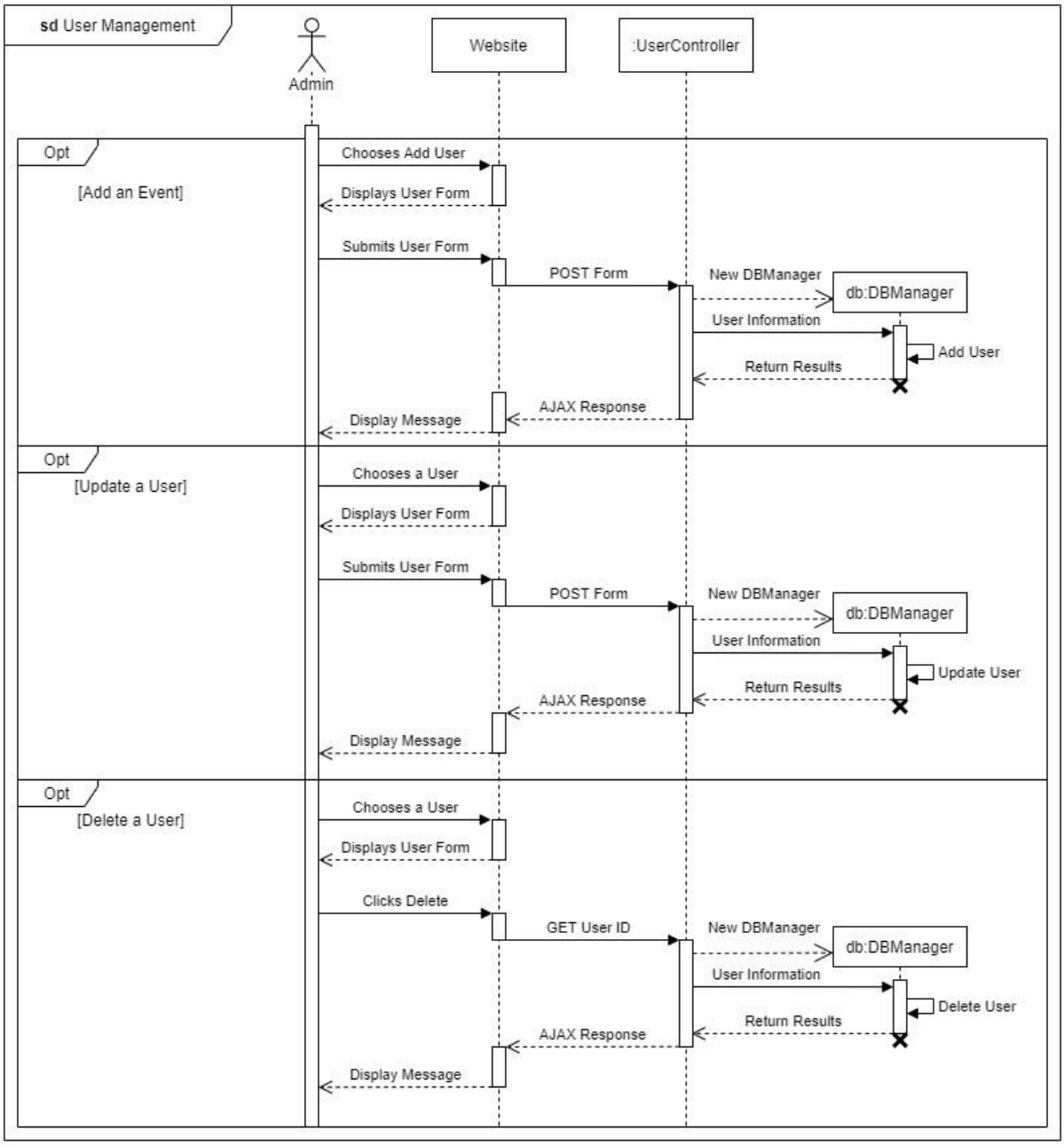
Appendix C - Sequence Diagrams



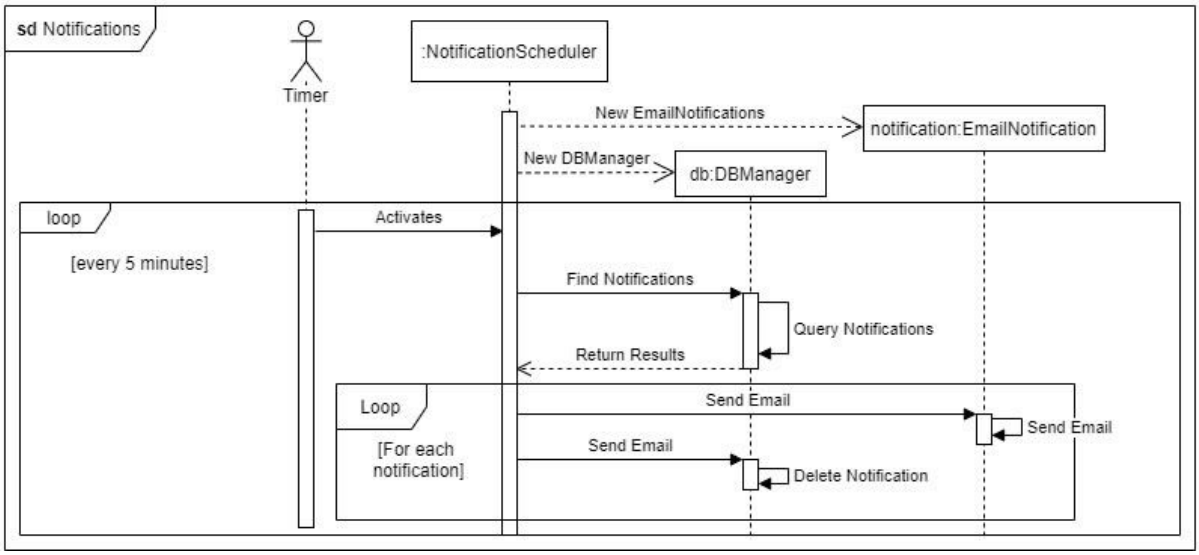
The above diagram shows the sequence diagram for the use case of a user or administrator logging into the system.



The above diagram shows the use sequences surrounding event management in the software. Options for adding, updating and deleting an Event are demonstrated.



The above diagram shows the use sequences surrounding user management by an administrator in the software. Options for adding, updating and deleting a User are demonstrated.



The above diagram demonstrates the sequences required to handle time based notifications, which alert a user to upcoming events at a predetermined time. These events require the use of a timer which repeatedly checks for new events, sends out emails to those users and removes the notification when it has been dealt with.

Appendix E - Testing

Test Plan

Test ID	Test Description	Test Steps	Expected Results
1	Test Login failure	Try and log in with garbage data Username:dfddfffd passwordh:8271h8231	Login should be refused
2	Test Login Success	Log in with a valid account Username:barryoconnor password:password	Login successful
3	Test Logout	Hover over user icon and click “Logout”	Logout successful
4	Test Darkmode Toggle	Hover over settings cog and click “Dark mode” toggle	Successfully toggled between Dark and Light mode
5	Test Open Event Viewer/Modifier	Click on event square within the calendar while logged in	Box with the event details should appear in dialog box
6	Test Create Event	Click on an empty part of a day within the calendar while logged in	Box with the title “New Event” and blank details should appear in dialog box
7	Test Closing Event box (View/Modify/ Create)	Click the cross in the top right corner of the pop-up box	Should close the box without any changes or amendments to the event
7	Test “Reset” button when creating/modifying Event	Click on the “Reset” button at the bottom of the “New Event” popup box	Should reset each of the input to default values
8	Test “Create” button when creating Event	Click on the “Create” button at the bottom of the “New Event” popup box	Should create a new event based around the user entered details into each of the inputs
9	Test “Delete” button when modifying Event	Click on the “Delete” button at the bottom of the modify event box under the event name	Should remove the specific instance of the event that the user selected
10	Test “Update” button when modifying Event	Click on the “Update” button at the bottom of the modify event box under the event name	Should update the event with the new/updated data the user entered

11	Test selecting calendar view type “Monthly”	Click on the dropdown box with title being “Weekly” or “Daily” and click “Monthly”	Should update the user interface with a monthly view of the calendar
12	Test selecting calendar view type “Weekly”	Click on the dropdown box with title being “Daily” or “Monthly” and click “Weekly”	Should update the user interface with a weekly view of the calendar
13	Test selecting calendar view type “Daily”	Click on the dropdown box with title being “Weekly” or “Monthly” and click “Daily”	Should update the user interface with a daily view of the calendar
14	Test changing mini calendar view	Click on either the left or right pointing arrows either side of the current month name	Should change the calendar view to the corresponding month shown on the mini calendar
15	Test open “Add a User” dialog box	Hover over icon with 2 users and settings cog and click “Add a User” button	Should open “New User” box with empty inputs for the user to enter
16	Test open “Edit a User” dialog box	Hover over icon with 2 users and settings cog and click “Edit a User” button	Should open “Select a user” box with a list of users to choose from to edit
17	Test open “Delete a User” dialog box	Hover over icon with 2 users and settings cog and click “Delete a User” button	Should open “Select a user” box with a list of users to choose from to delete
18	Test adding user in “New User” dialog box	After entering the new users information, click “Create” button	Should create a new user that can be used via login
19	Test deleting user in user dialog box	After selecting a user from the dropdown box, click the “Delete” button	Should remove the user and all their information from the database
20	Test Calendar App responsiveness	Resize browser window running Calendar app to be narrow	Should hide the mini calendar on the left and integrate monthly changing controls to above the main calendar
21	Test making a repeating Event	Set the repeat every1 or more days until a later date	Should repeat the same event every stated day until the stated date
22	Testing Email Notification when an event is created	Creating an event and setting the “notify me” to an appropriate option	The attendees should receive an email about the event, depending on the notification option chosen
23	Testing Notify option when it set to “Never”	Creating an event and setting the “Notify Me” to “never”	The attendees should not receive an email about the event

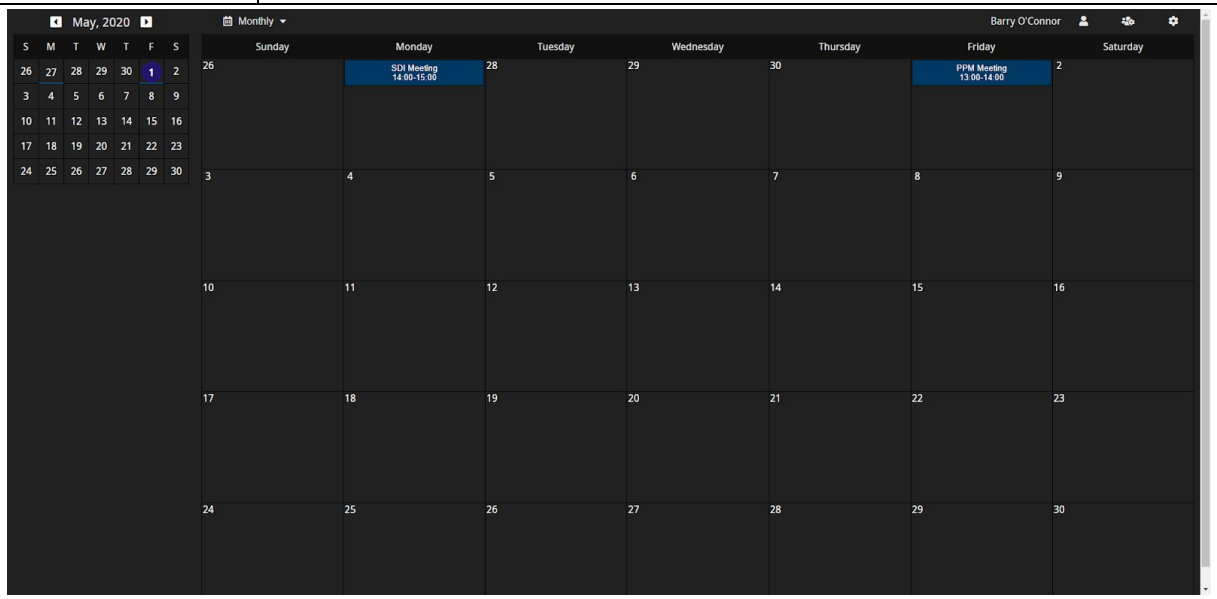
24	Test responsive nature of the site	Open the site in various browsers and check how it looks at different sizes	The site should look good at all sizes on all browsers.

Test Tables

Test ID	1	**Test Name**	Login Failure
Module Name	Login	**No Of Attempts**	1
Requirement Id	17	**Status of Test**	SUCCESS
Test Description	Attempt to login with invalid credentials		
Setup Instructions	1. Open the website and go to the login screen 2. Login with the following credentials: Username: nobody Password: password		
Success Criteria	Login attempt will be rejected		
Test Results	Login attempt was rejected with the message “Invalid credentials”		
Tester Name	Barry O’Connor		

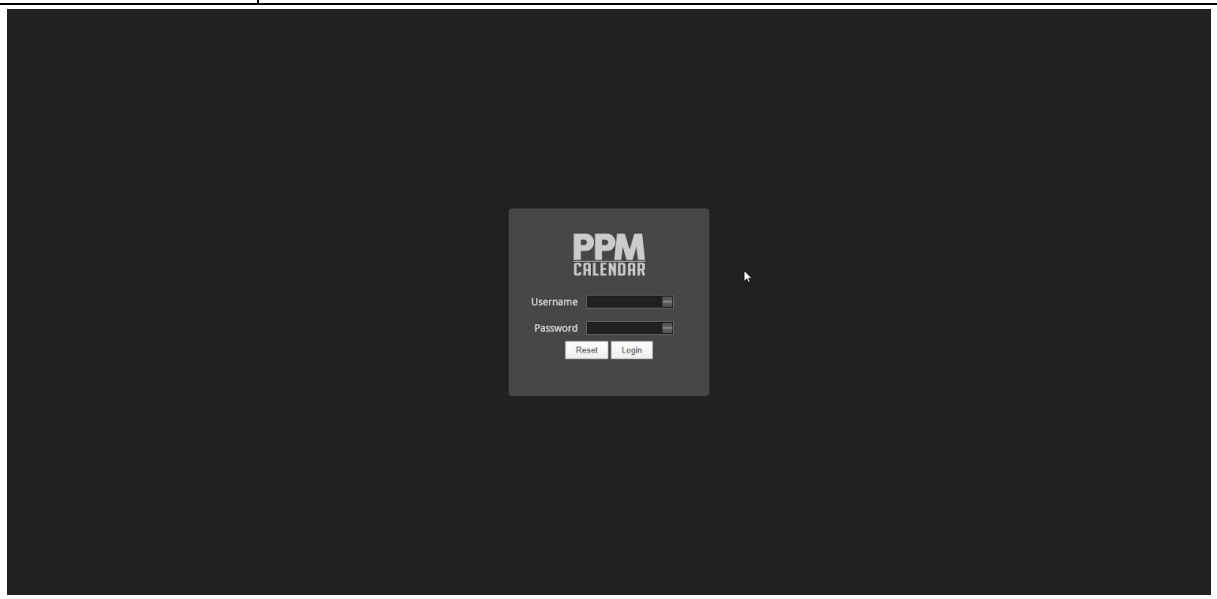
Related Evidence - Test1.png image

Test ID	2	**Test Name**	Login Success
Module Name	Login	**No Of Attempts**	1
Requirement Id	17	**Status of Test**	SUCCESS
Test Description	Attempt to login with valid credentials		
Setup Instructions	3. Open the website and go to the login screen 4. Login with the following credentials: Username: barryoconnor Password: password		
Success Criteria	Login attempt will be successful and redirect to the main calendar page		
Test Results	Login attempt was successful and I was redirected to the main calendar page successfully		
Tester Name	Barry O'Connor		



Related Evidence - Test2.png image

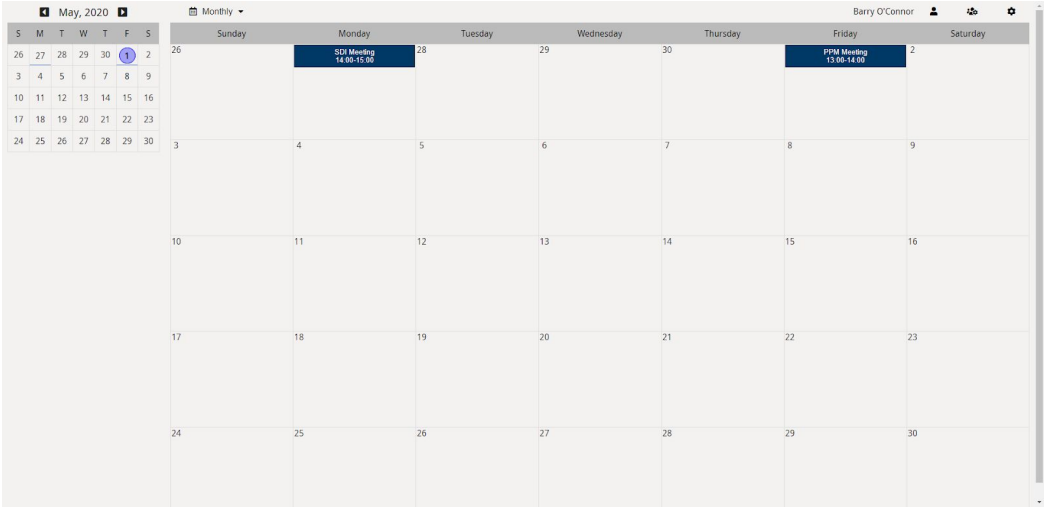
Test ID	3	**Test Name**	Test Logout
Module Name	Logout	**No Of Attempts**	1
Requirement Id	18	**Status of Test**	SUCCESS
Test Description	Attempt to logout of calendar app		
Setup Instructions	1. Hover mouse over user icon 2. Click “Logout”		
Success Criteria	Logout attempt will be successful and redirect the user to the calendar app login page		
Test Results	Logout attempt was successful and I was redirected to the calendar app login page		
Tester Name	William Peters		



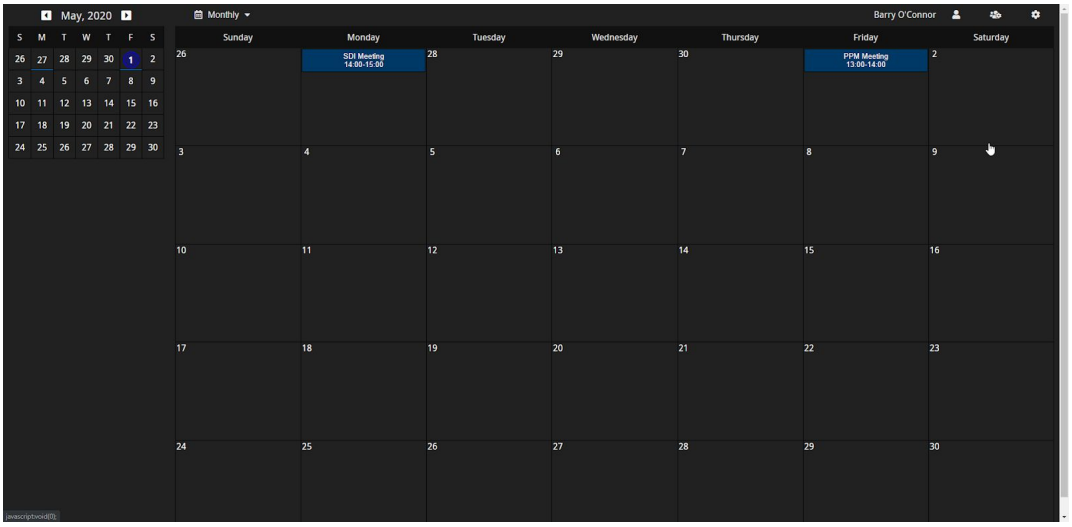
Related Evidence - Test3.png image

Test ID	4	**Test Name**	Test Darkmode Toggle
Module Name	Personalisation	**No Of Attempts**	1

Requirement Id	4	**Status of Test**	SUCCESS
Test Description	Attempt to toggle between dark and light colours of calendar application		
Setup Instructions	1. Hover mouse over settings cog in top right corner 2. Click “Dark mode” toggle button		
Success Criteria	Toggle attempt will be successful from the colours inverting from dark to light / light to dark		
Test Results	Toggle attempt was successful because the colours inverted from dark to light / light to dark		
Tester Name	William Peters		



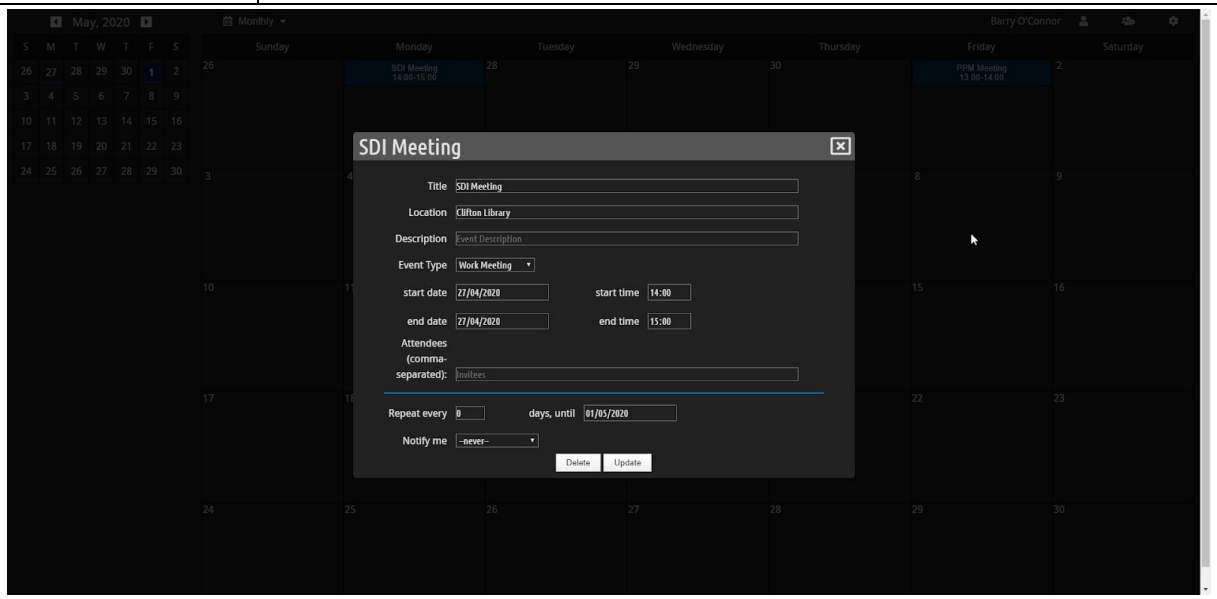
Related Evidence - Test4.1.png image



Related Evidence - Test4.2.png image

Test ID	5	**Test Name**	Test Open Event Viewer/Modifier
Module Name	Event Dialog Box	**No Of Attempts**	1
Requirement Id	6,10	**Status of Test**	SUCCESS

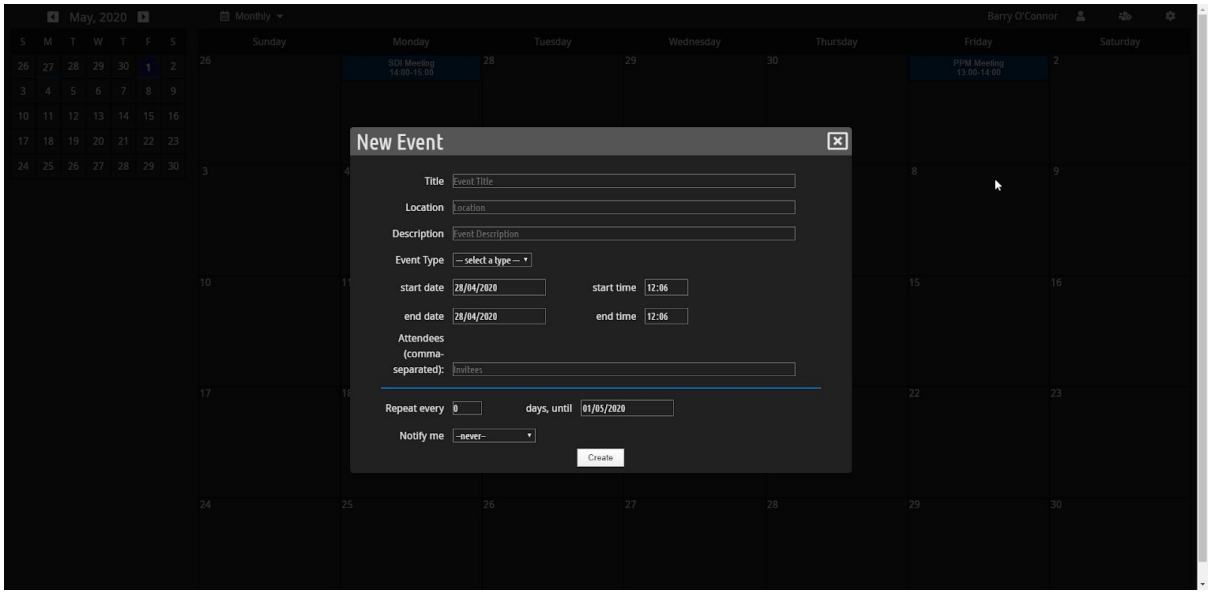
Test Description	Attempt to open the event viewer/modifier dialog box
Setup Instructions	1. Ensure user is logged in and has events on calendar 2. Click on event square within calendar
Success Criteria	Box with the event details should appear in a dialog box
Test Results	Box with the event details does appear in a dialog box
Tester Name	William Peters



Related Evidence - Test5.png

Test ID	6	**Test Name**	Test Open Event Creator
Module Name	Event Dialog Box	**No Of Attempts**	1
Requirement Id	10,11	**Status of Test**	SUCCESS
Test Description	Attempt to open the event creator dialog box		
Setup Instructions	1. Ensure the user is logged in 2. Click on a empty space on a day within calendar		

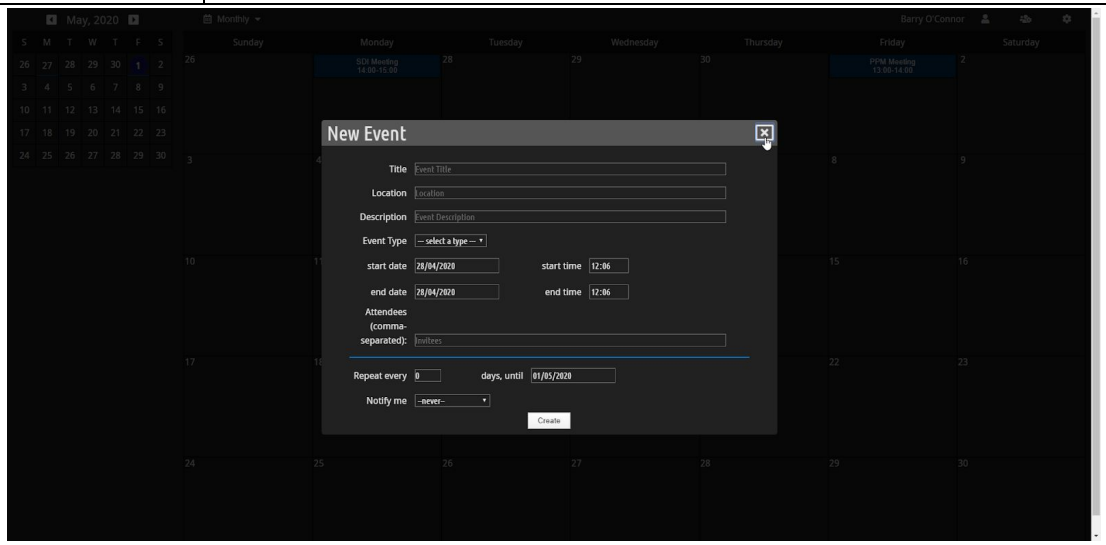
Success Criteria	Box with empty event details should appear in dialog box
Test Results	Box with empty event details does appear in dialog box
Tester Name	William Peters



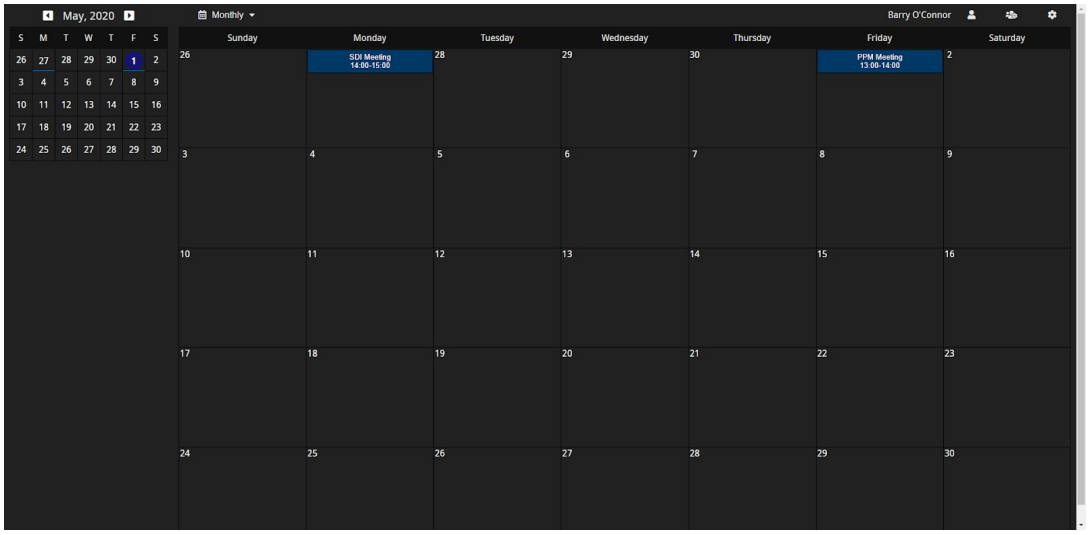
Related Evidence - Test6.png

Test ID	7	**Test Name**	Test Closing Event Box (View/Modify/ Create)
Module Name	Event Dialog Box	**No Of Attempts**	1
Requirement Id	10	**Status of Test**	SUCCESS
Test Description	Attempt to close event dialog box		
Setup Instructions	1. Have some form of event dialog box open (being to view, modify, or create an event) 2. Click the “X” in the top right corner		
Success Criteria	Box should close displaying the calendar application fully		

Test Results	Box does close displaying the calendar application fully
Tester Name	William Peters



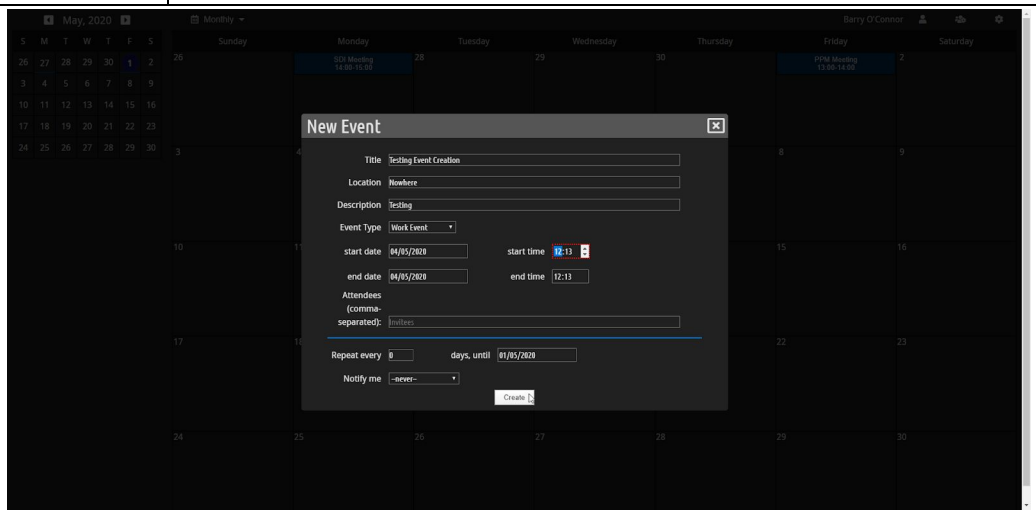
Related Evidence - Test7.1.png



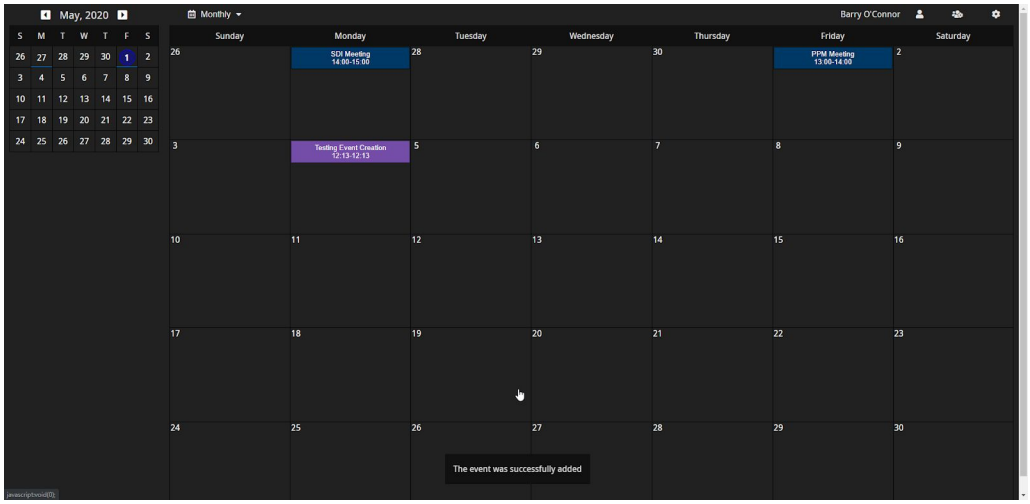
Related Evidence - Test7.2.png

Test ID	8	**Test Name**	Test “Create” button when creating Event
Module Name	Event Creation	**No Of Attempts**	1
Requirement Id	11	**Status of Test**	SUCCESS
Test Description	Attempt to confirm and create new event		
Setup Instructions	1. Ensure there is information in the new event dialog box 2. Click the “Create” button		
Success Criteria	New Event dialog box should close, showing the updated calendar with the new event and a message at the bottom of the page saying “The event was successfully added”		
Test Results	New Event dialog box does close, showing the updated calendar with the		

	new event and a message at the bottom of the page saying “The event was successfully added”
Tester Name	William Peters



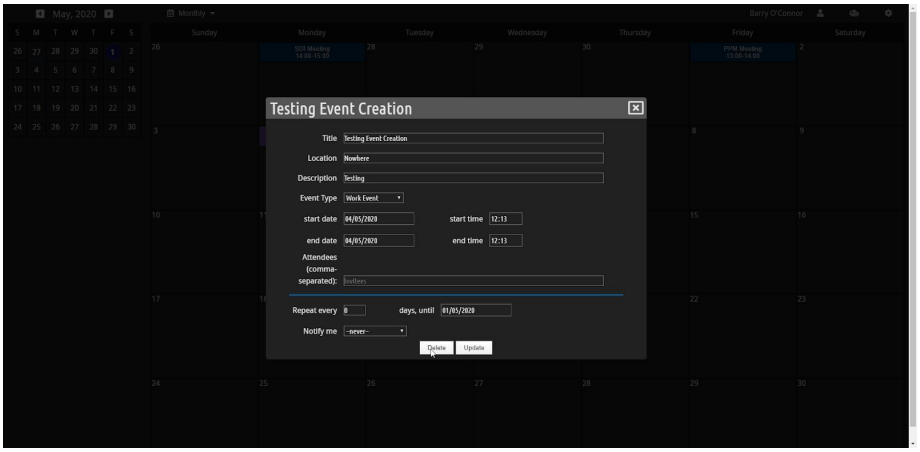
Related Evidence - Test 8.1.png



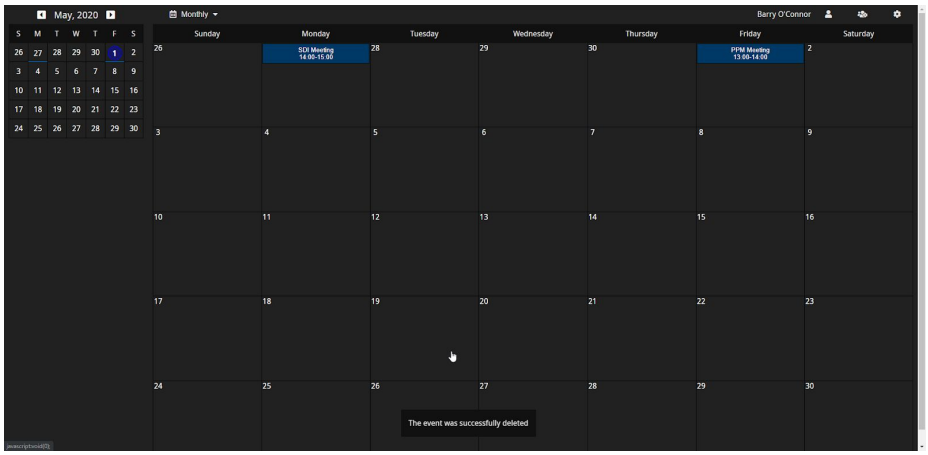
Related Evidence - Test 8.2.png

Test ID	9	**Test Name**	Test “Delete” button when modifying Event
Module Name	Event Delete	**No Of Attempts**	1
Requirement Id	12	**Status of Test**	SUCCESS
Test Description	Attempt to click “Delete” button when modifying Event to delete it		
Setup Instructions	1. Ensure modify event dialog box is open by clicking on an event on the calendar 2. Click “Delete” button		
Success Criteria	Modify Event dialog box should close, showing the updated calendar with the event now removed and a message at the bottom of the page saying “The event was successfully deleted”		

Test Results	Modify Event dialog box did close, showing the updated calendar with the event now removed and a message at the bottom of the page saying “The event was successfully deleted”
Tester Name	William Peters



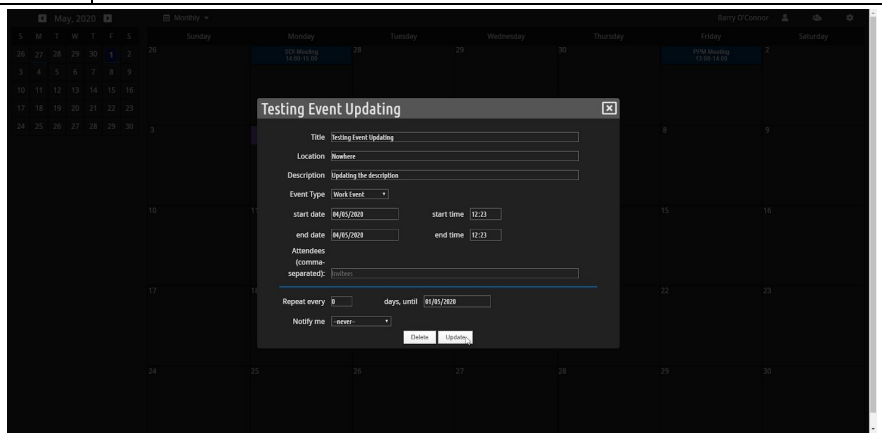
Related Evidence - Test 9.1.png



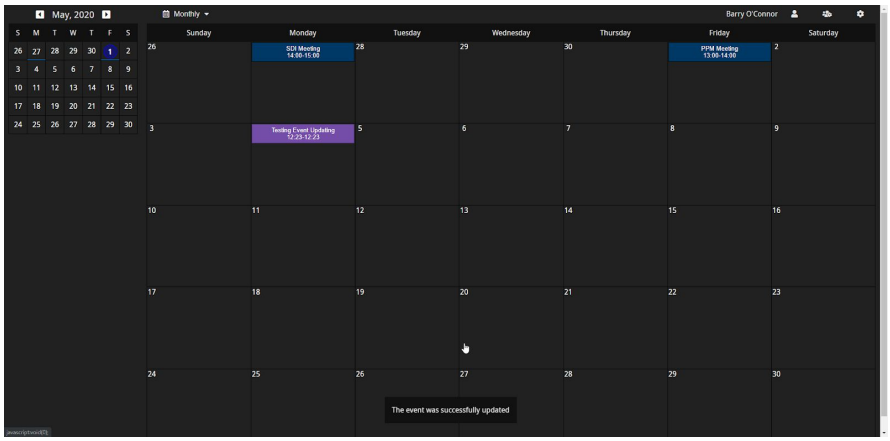
Related Evidence - Test 9.2.png

Test ID	10	**Test Name**	Test “Update” button when modifying Event
Module Name	Event Update	**No Of Attempts**	1
Requirement Id	13	**Status of Test**	SUCCESS
Test Description	Attempt to click “Update” to update the details of a event		
Setup Instructions	1. Ensure modify event dialog box is open by clicking on an event on the calendar 2. Click “Update” button		
Success Criteria	Modify event dialog box should close, showing the updated calendar with the event with its new updated information and a message at the bottom of the page saying “The event was successfully updated”		
Test Results	Modify event dialog box does close, showing the updated calendar with the event with its new updated information and a message at the bottom of the page saying “The event was successfully updated”		

Tester Name	William Peters

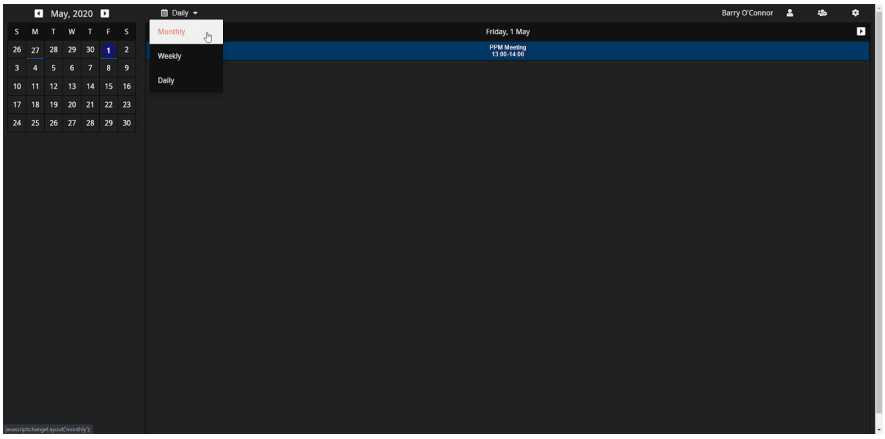


Related Evidence - 10.1.png

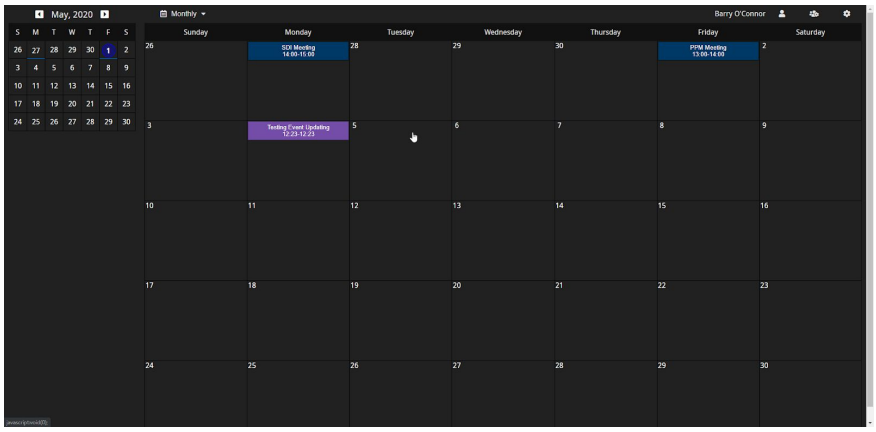


Related Evidence - 10.2.png

Test ID	11	**Test Name**	Test selecting calendar view type “Monthly”
Module Name	Calendar View	**No Of Attempts**	1
Requirement Id	11	**Status of Test**	SUCCESS
Test Description	Attempt to change the calendar view from “Weekly” or “Daily” to “Monthly”		
Setup Instructions	1. Have the calendar application open on the main page 2. Click the dropdown box above the calendar that says either “Weekly” or “Daily” 3. Select “Monthly”		
Success Criteria	Current calendar view should change from “Weekly” or “Daily” to “Monthly”, showing all of the days of the month and name of the day of the week		
Test Results	Current calendar view does change from “Weekly” or “Daily” to “Monthly”, showing all of the dates of the month and names of the days of the week		
Tester Name	William Peters		

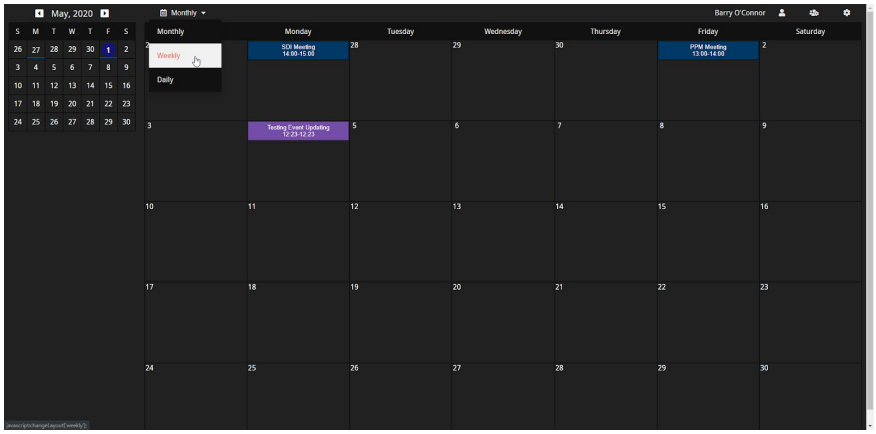


Related Evidence - 11.1.png

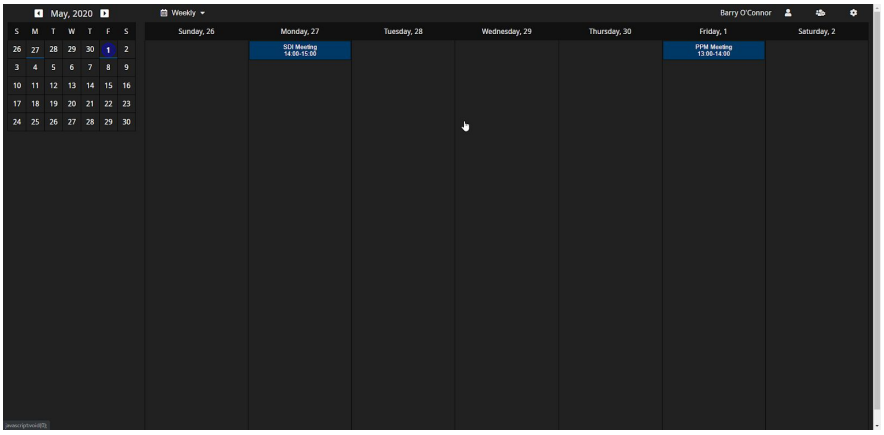


Related Evidence - 11.2.png

Test ID	12	**Test Name**	Test selecting calendar view type “Weekly”
Module Name	Calendar View	**No Of Attempts**	1
Requirement Id	12	**Status of Test**	SUCCESS
Test Description	Attempt to change the calendar view from “Daily” or “Monthly” to “Weekly”		
Setup Instructions	1. Have the calendar application open on the main page 2. Click the dropdown box above the calendar that says either “Monthly” or “Daily” 3. Select “Weekly”		
Success Criteria	Current calendar view should change from “Daily” or “Monthly” to “Weekly”, showing all of the dates of the week and names of the days of the week		
Test Results	Current calendar view does change from “Daily” or “Monthly” to “Weekly”, showing all of the dates of the week and names of the days of the week		
Tester Name	William Peters		

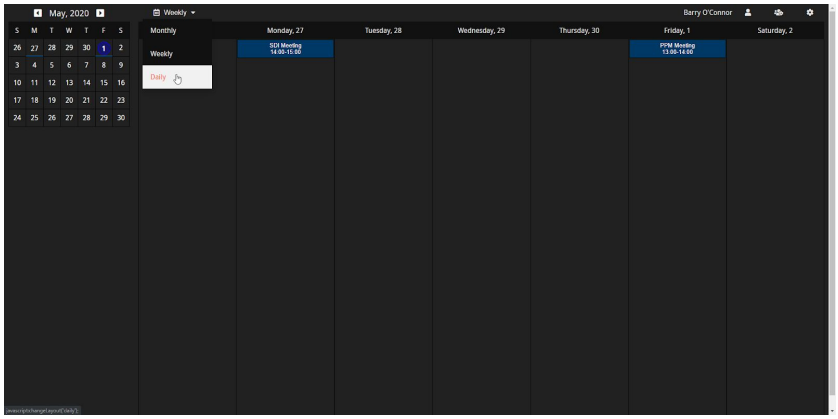


Related Evidence - 12.1.png

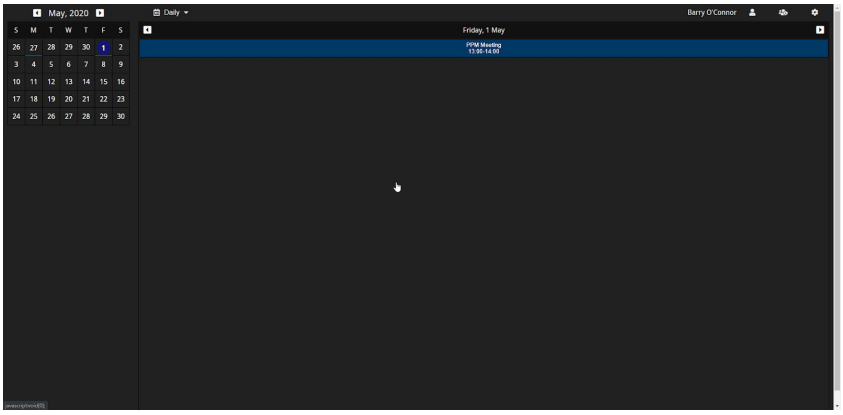


Related Evidence - 12.2.png

Test ID	13	**Test Name**	Test selecting calendar view type “Daily”
Module Name	Calendar View	**No Of Attempts**	1
Requirement Id	13	**Status of Test**	SUCCESS
Test Description	Attempt to change the calendar view from “Weekly” or “Monthly” to “Daily”		
Setup Instructions	1. Have the calendar application open on the main page 2. Click the dropdown box above the calendar that says either “Weekly” or “Monthly” 3. Select “Daily”		
Success Criteria	Current calendar view should change from “Weekly” or “Monthly” to “Daily”, showing just the selected day’s date and the name of the selected day		
Test Results	Current calendar view does change from “Weekly” or “Monthly” to “Daily”, showing just the selected day’s date and the name of the selected day		
Tester Name	William Peters		

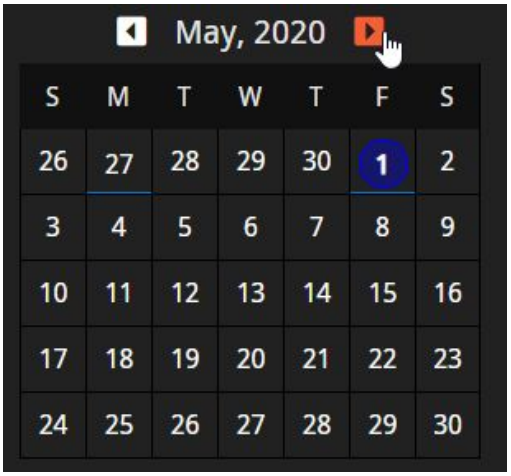


Related Evidence - 13.1.png



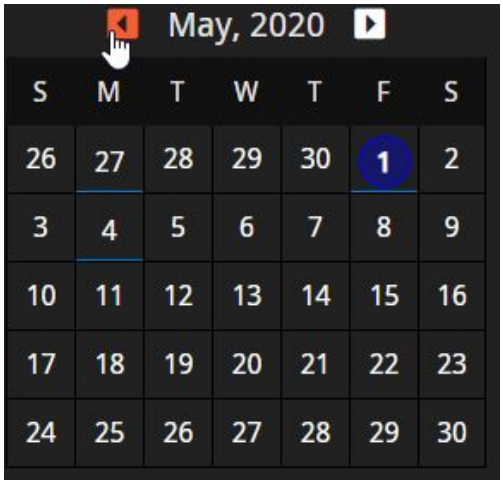
Related Evidence - 13.2.png

Test ID	14	**Test Name**	Test changing mini calendar month
Module Name	Mini Calendar	**No Of Attempts**	1
Requirement Id	14	**Status of Test**	SUCCESS
Test Description	Attempt to change mini calendar month view		
Setup Instructions	1. Be on the main calendar application page 2. Click on either the left or right arrows by the Month+Year in the top left corner		
Success Criteria	The month being shown should go along or back to future or previous months with the correct number of days, with the correct day of the week for each day		
Test Results	The month being shown does go along or back to future or previous months with the correct number of days, with the correct day of the week for each day		
Tester Name	William Peters		

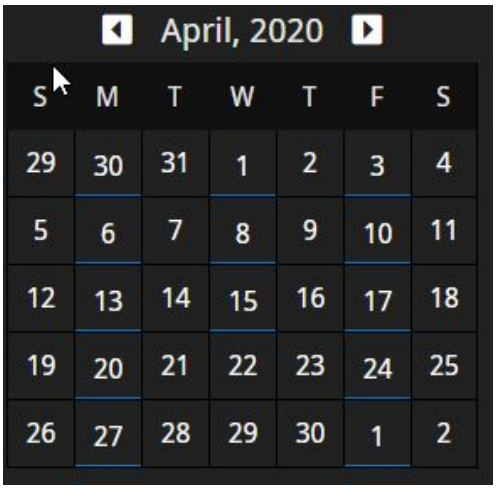


Related Evidence - 14.1.png

Related Evidence - 14.2.png

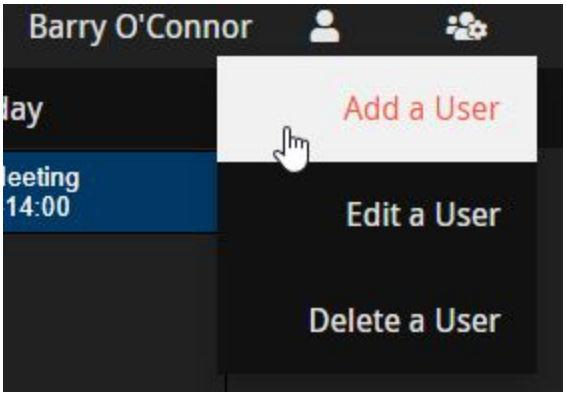


Related Evidence - 14.3.png

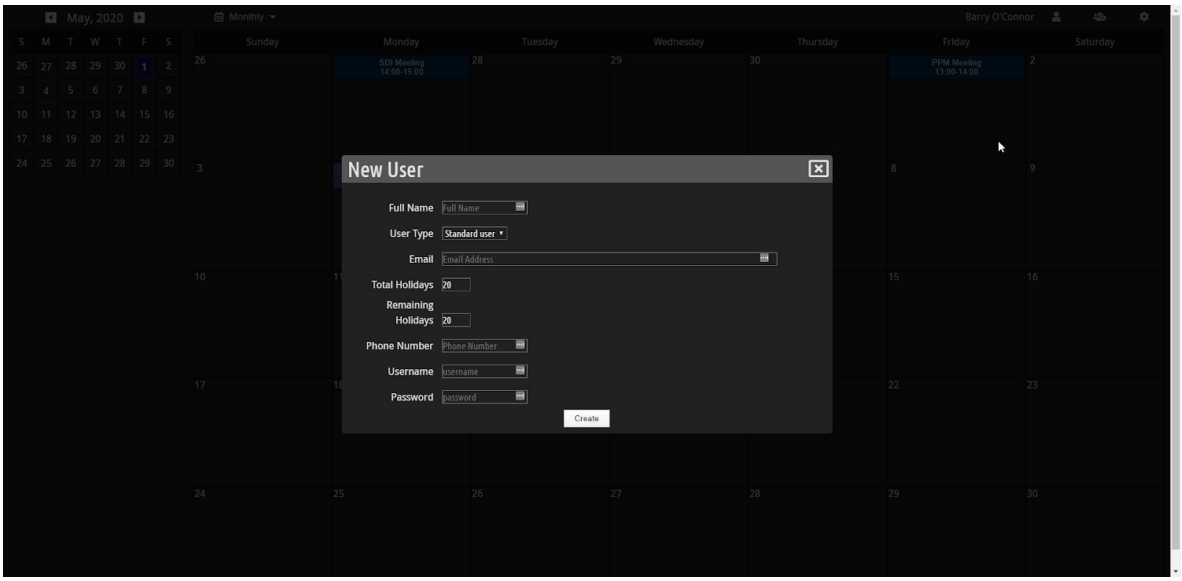


Related Evidence - 14.4.png

Test ID	15	**Test Name**	Test click “Add a User” button
Module Name	User Addition Dialog Box	**No Of Attempts**	1
Requirement Id	20	**Status of Test**	SUCCESS
Test Description	Attempt to click “Add a User” button, to open the “Add a User” dialog box		
Setup Instructions	1. Ensure user that is logged in has admin privileges 2. Hover over User/Cog icon 3. Click “Add a User”		
Success Criteria	Upon clicking “Add a User” button, a “New User” dialog box should open with empty boxes for the admin user to fill in		
Test Results	Upon clicking “Add a User” button, a “New User” dialog box does open with empty boxes for the admin user to fill in		
Tester Name	William Peters		

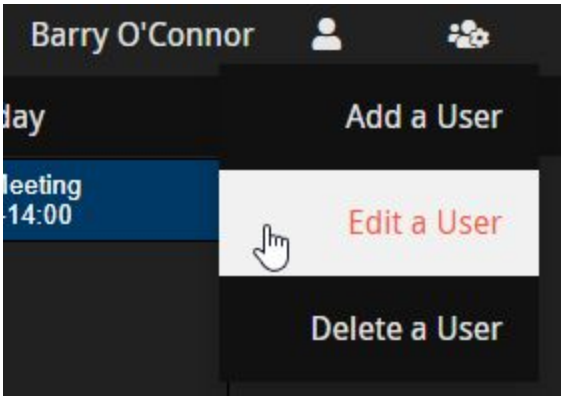


Related Evidence - 15.1.png

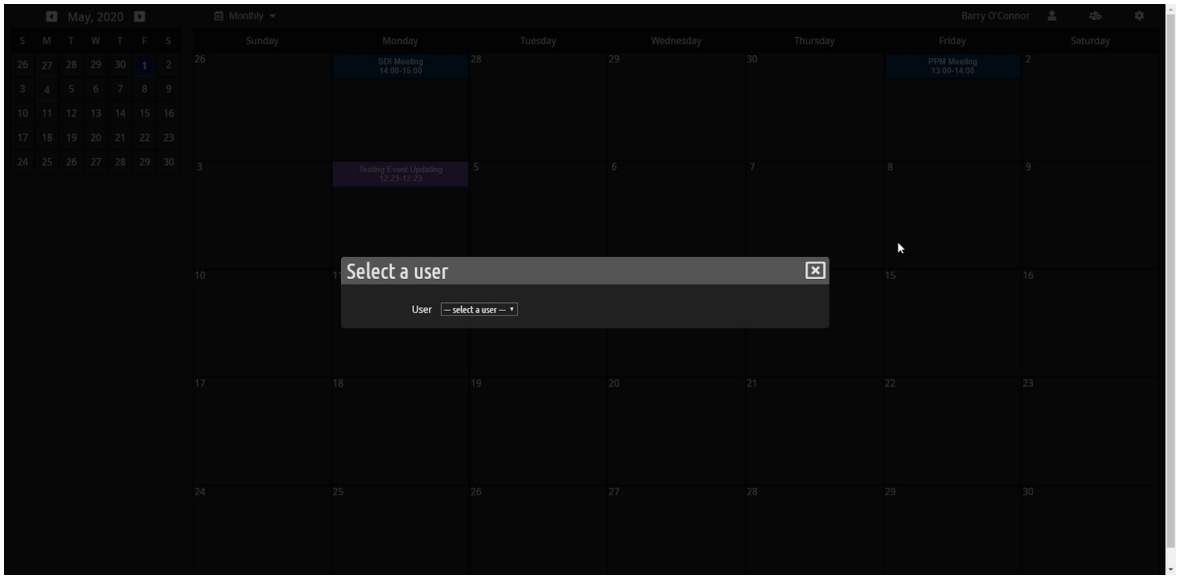


Related Evidence - 15.2.png

Test ID	16	**Test Name**	Test click “Edit a User” button
Module Name	Select User Dialog Box	**No Of Attempts**	1
Requirement Id	20	**Status of Test**	SUCCESS
Test Description	Attempt to click “Edit a User” button, to open the “Select a User” dialog box		
Setup Instructions	1. Ensure user that is logged in has admin privileges 2. Hover over User/Cog icon 3. Click “Edit a User”		
Success Criteria	Upon clicking “Edit a User” button, a “Select a User” dialog box should open with a dropdown list of users to select		
Test Results	Upon clicking “Edit a User” button, a “Select a User” dialog box does open with a dropdown list of users to select		
Tester Name	William Peters		

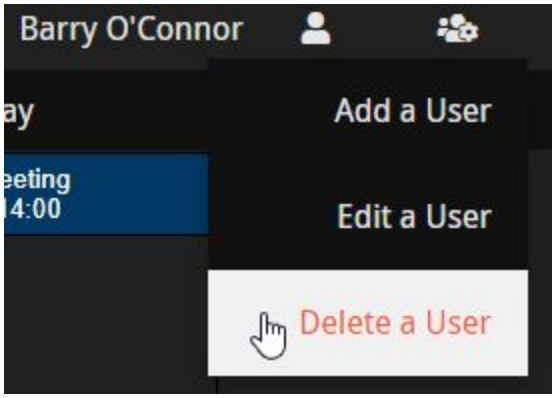


Related Evidence - 16.1.png

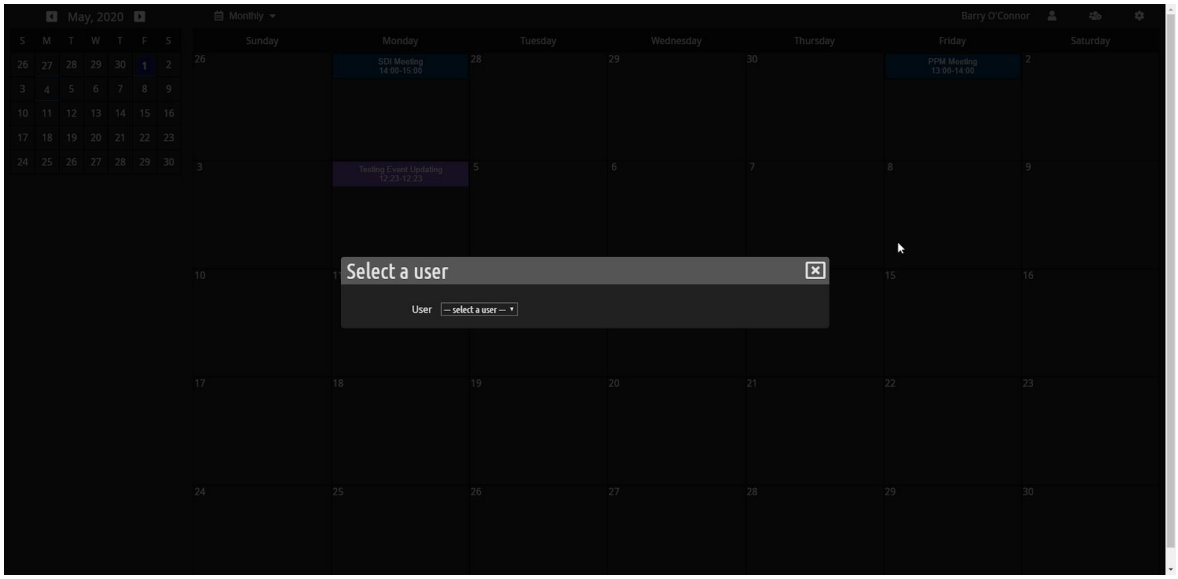


Related Evidence - 16.2.png

Test ID	17	**Test Name**	Test click “Delete a User” button
Module Name	Select User Dialog Box	**No Of Attempts**	1
Requirement Id	20	**Status of Test**	SUCCESS
Test Description	Attempt to click “Delete a User” button, to open the “Select a User” dialog box		
Setup Instructions	1. Ensure user that is logged in has admin privileges 2. Hover over User/Cog icon 3. Click “Delete a User”		
Success Criteria	Upon clicking “Delete a User” button, a “Select a User” dialog box should open with a dropdown list of users to select		
Test Results	Upon clicking “Delete a User” button, a “Select a User” dialog box does open with a dropdown list of users to select		
Tester Name	William Peters		



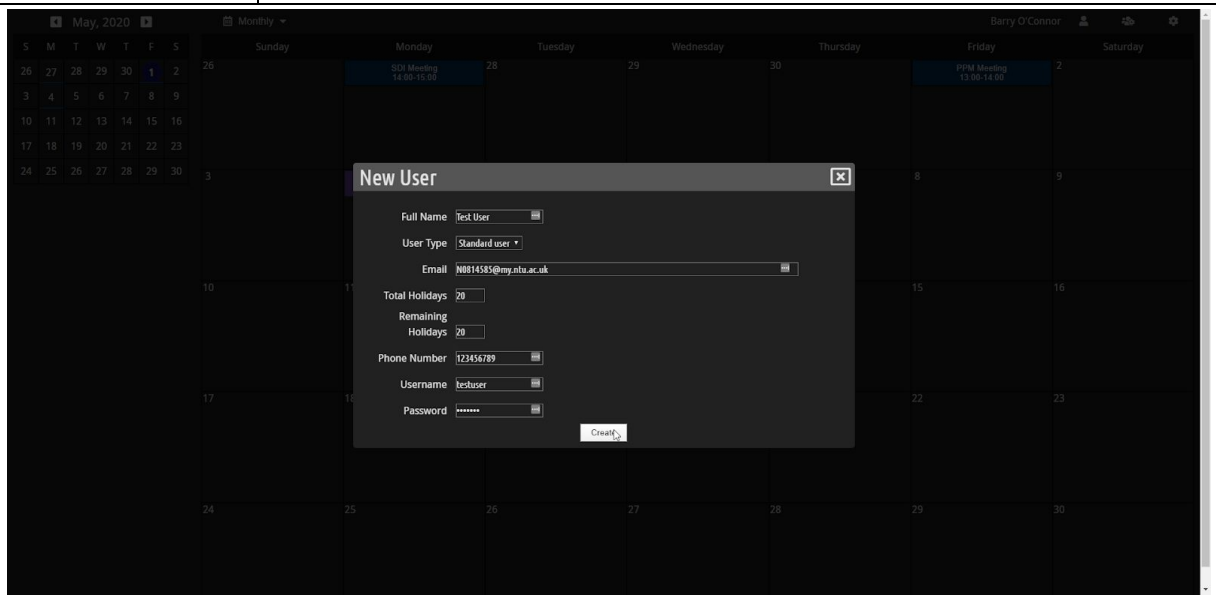
Related Evidence - 17.1.png



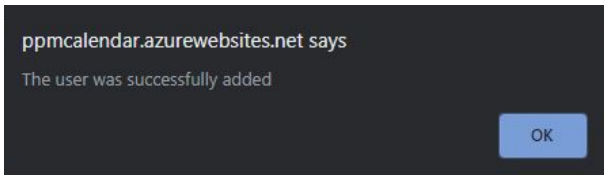
Related Evidence - 17.2.png

Test ID	18	**Test Name**	Test adding user in “New User” dialog box
Module Name	User Addition	**No Of Attempts**	1
Requirement Id	20	**Status of Test**	SUCCESS
Test Description	Attempt to add a new user to the calendar database		
Setup Instructions	1. Ensure the user logged in has admin privileges and has opened the “New User” dialog box 2. Enter the new users’ details 3. Click the “Create” button		
Success Criteria	When the admin user clicks “Create”, the dialog box should close and a browser popup box should appear saying “The user was successfully added” and the user should be available from the list of users for editing or deleting		
Test Results	When the admin user clicks “Create”, the dialog box does close and a browser popup box should appear saying “The user was successfully added” and the user is available from the list of users for editing or deleting		

Tester Name	William Peters



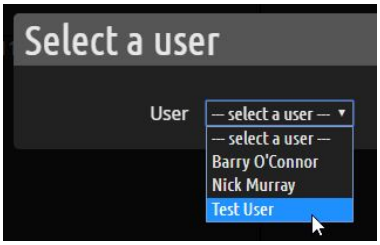
Related Evidence - 18.1.png



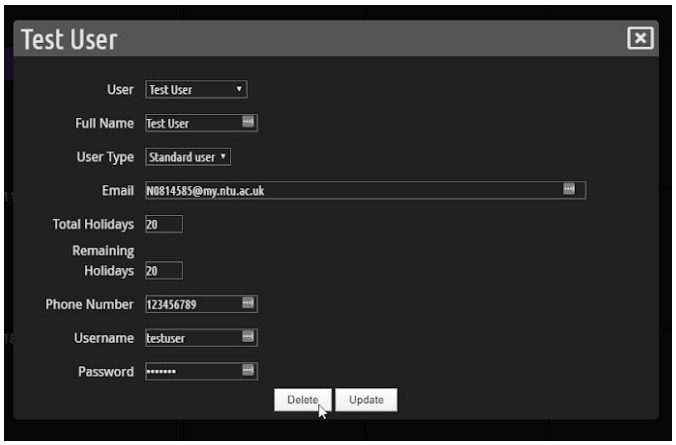
Related Evidence - 18.2.png

Related Evidence - 18.3.png

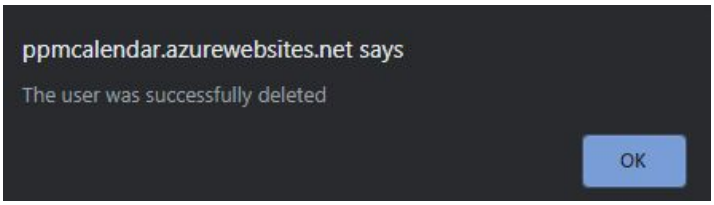
Test ID	19	**Test Name**	Test Deleting user in user dialog box
Module Name	User Modification	**No Of Attempts**	1
Requirement Id	20	**Status of Test**	SUCCESS
Test Description	Attempt to delete user from database		
Setup Instructions	1. Ensure user is logged in with admin privileges 2. Select user by either clicking “Edit a User” or “Delete a User” 3. Click “Delete” button		
Success Criteria	When the “Delete” button is clicked, a popup should appear saying “The user was successfully deleted” and the user deleted shouldn’t appear in the list of users to select from		
Test Results	When the “Delete” button is clicked, a popup does appear saying “The user was successfully deleted” and the user deleted does not appear in the list of users to select from		
Tester Name	William Peters		



Related Evidence - 19.1.png



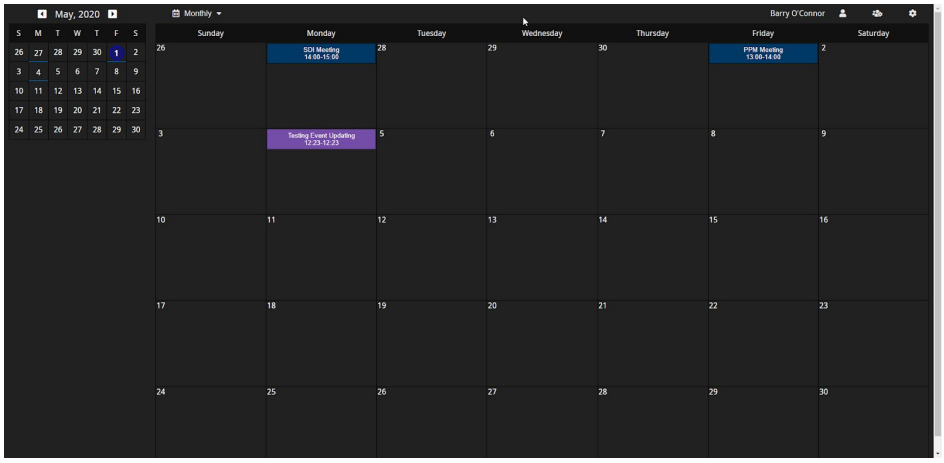
Related Evidence - 19.2.png



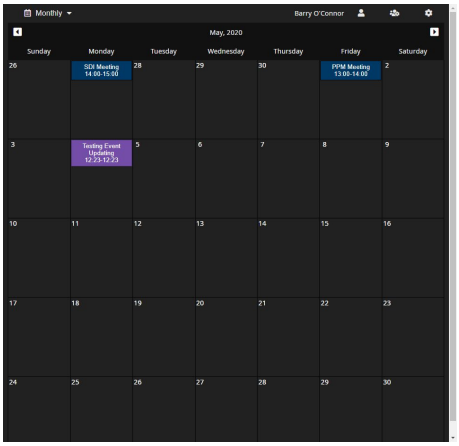
Related Evidence - 19.3.png

Related Evidence - 19.4.png

Test ID	20	**Test Name**	Test Calendar App Responsiveness
Module Name	Compatibility	**No Of Attempts**	1
Requirement Id	2	**Status of Test**	SUCCESS
Test Description	Attempt to resize browser window containing Calendar Application to see responsiveness		
Setup Instructions	1. Ensure the calendar application is open 2. Resize window width to replicate mobile / portrait devices		
Success Criteria	When the user resizes the window from fullscreen/default to a narrower size, the mini calendar on the left should be hidden with monthly changing controls now integrated into the top of the calendar application		
Test Results	When the user resizes the window from fullscreen/default to a narrower size, the mini calendar on the left does hide with monthly changing controls now integrated into the top of the calendar application		
Tester Name	William Peters		



Related Evidence - 20.1.png



Related Evidence - 20.2.png

Test ID	22	**Test Name**	Email Notification
Module Name	Email	**No Of Attempts**	1
Requirement Id	8,9	**Status of Test**	SUCCESS
Test Description	Testing Email Notification when an event is created		
Setup Instructions	1. Create a new event 2. Fill in the required fields 3. Set “Notify Me” to an appropriate option		
Success Criteria	Email will be sent to attendees in the selected time frame		
Test Results	Email was received in the required time		
Tester Name	Soham Jaiswal		

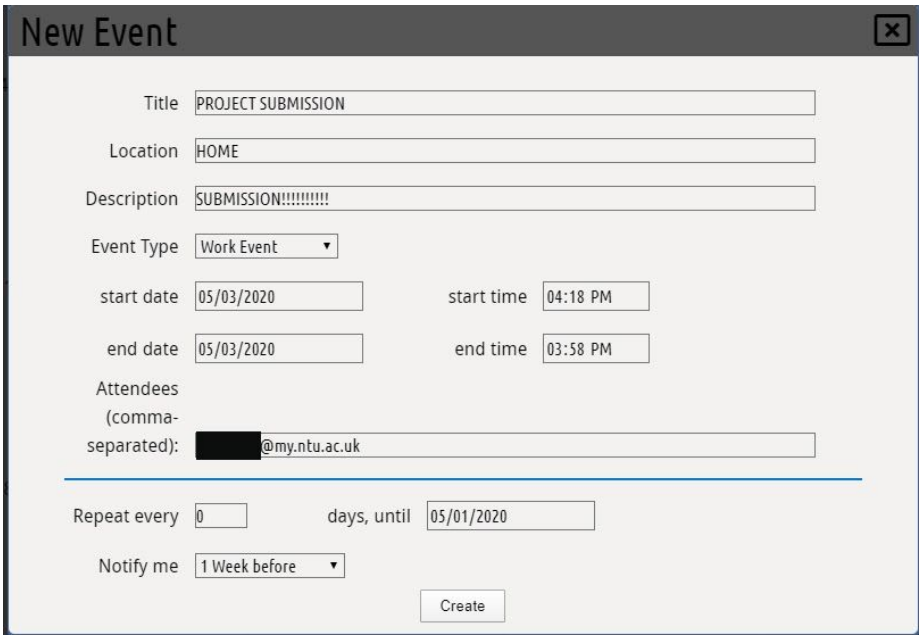


Fig: 22 – 1 Showing Creation of event

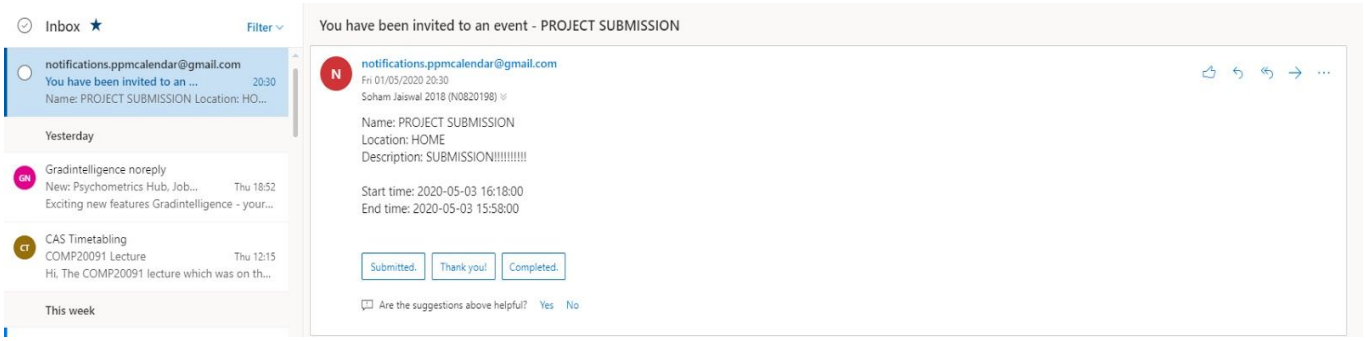


Fig: 22 – 2 Showing Email was received by attendees

Test ID	23	**Test Name**	Notify Me option
Module Name	Email	**No Of Attempts**	1
Requirement Id	8,9	**Status of Test**	SUCCESS
Test Description	Testing Notify option when it set to “Never”		

Setup Instructions	1. Create a new event 2. Fill in the required fields 3. Set “Notify Me” to an “never”
Success Criteria	Email will not be sent to the creator
Test Results	Email notification not sent
Tester Name	Soham Jaiswal

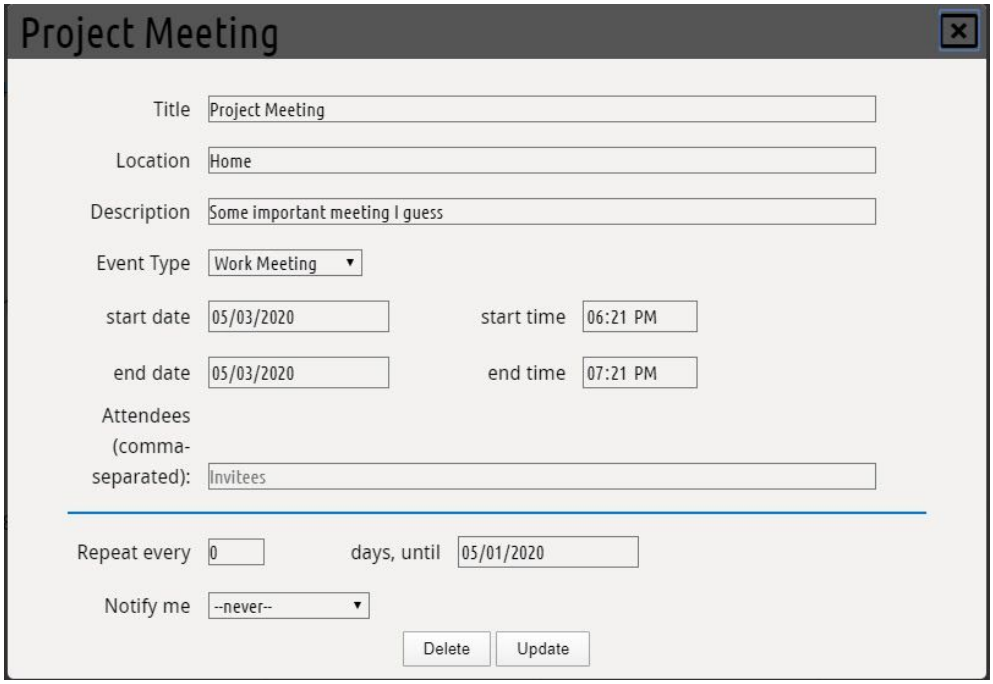


Fig 23 – 1 Demonstrating creation of event with “Notify Me” set to “never”

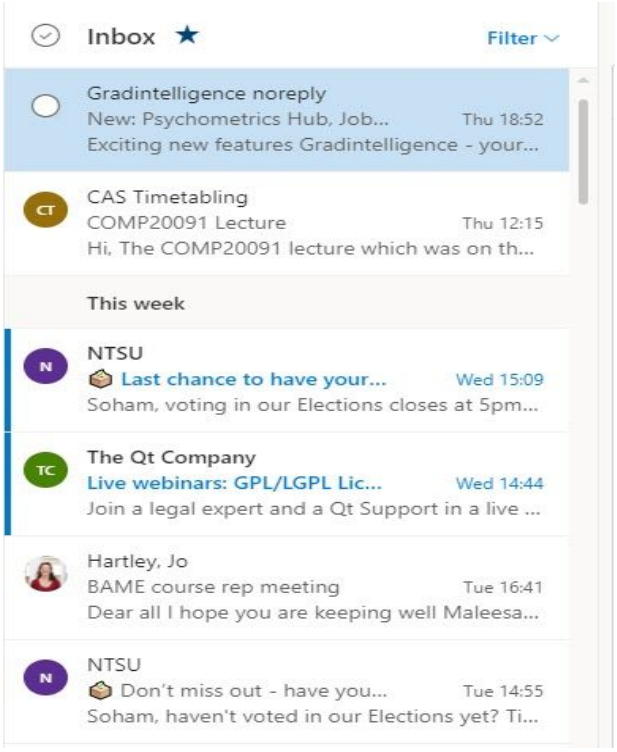


Fig 23 – 2 showing no email notification was received

Test ID	24	**Test Name**	Responsive Testing
Module Name	Responsive HTML	**No Of Attempts**	1
Requirement Id	1,2,3	**Status of Test**	Success
Test Description	Open the website in several browsers and test the look of the site at various screen sizes.		
Setup Instructions	Simply resize the browser while using the site and see how it responds.		
Success Criteria	Site should look good and remain usable at different screen sizes		
Test Results	https://youtu.be/IMp7MyL3EAU SHows a video of the test of responsivity across several browsers		
Tester Name	Barry O'Connor		

Appendix F - Meeting Minutes

Meeting Type: Group Meeting Date: 04/12/2019

Attendees:

Soham Jaiswal
Nick Murray-Hill
Barry O'Connor
Kieran Olney
Will Peters

Absent:

Zach Russell

Topics Discussed:

- To go forward with a scheduling web page
- Initial 3 sections spread across all team members - fill in your role below
    - Introduction/ Aims & Objectives - Nick
    - Project Management / Risk Assessment - Barry
    - Professional/Social/Ethical/Legal - Will
    - Market Research - Soham
    - Functionality - Kieran

- Get feedback in Fridays tutorial session
    - If idea is suitable
    - What are the expectations of final product

Meeting Type: Tutor Meeting Date: 06/12/2019

Attendees:

Soham Jaiswal
Barry O'Connor
Kieran Olney
Will Peters

Apologies:

Nick Murray-Hill (Notified group and tutor of absence)

Absent:

Zach Russell

Topics Discussed:

- Discussed draft document - suggested we follow a layout that was different to that on NOW. I have emailed Andreas to ask for clarification on what to follow regarding this.
- Didn't read through the content, so most comments were telling us to do things we already had done.
- Suggested we justify the entire document (we're not) so it looks neater. I argued that random spacing between words is far more distracting than a jagged right-hand margin.
- Discussed Zach's continued absence, agreed to keep trying to contact him for a period of time (just in case there's a genuine reason he's been uncontactable) and continue to note down each person's contribution. Tutor said that if he continues to be uncontactable, we can continue as if the group is 5 people.
- Discussed the size of the project. Basically the guidance is that we need to sit down and decide how much we can deliver within the timeframe. We should not try and overdeliver and add loads of functionality if we feel we cannot finish what we have started. I think this is something we will have to discuss over the Xmas break a little so we don't submit this doc with a list of functionality we're unsure we can handle.

Meeting Type: Group Meeting Date: 12/12/2019

Attendees:

Soham Jaiswal
Barry O'Connor
Kieran Olney
Will Peters
Nick Murray-Hill

Absent:

Zach Russell

Topics Discussed:

- Discussed team roles and decided upon:
    - Barry - Project manager
    - Nick - Deputy Project Manager
    - Will - Software Developer
    - Kieran - Software Architect
    - Soham - Software Developer
    - Zach - Software Tester
- Discussed project storage and decided to have multiple storage locations. Requested everyone check whether they had access to the 'Olympuss' server.
- Discussed the pass/fail sessions so that everyone was aware of them and can notify the tutor and team if they cannot attend.
- Discussed the project and technologies used. We came to a decision on languages, so the backend will be in c#.
- Discussed the experience within the team - Nick has c# experience, Barry has HTML/CSS experience so we decided to allow them to oversee those parts to make sure everyone follows the same guidelines.

Actions:

- Everyone should read through the document and modify it as they see fit. Especially the sections on features, risk assessment and so on.
- Everyone should check the availability of the Olympuss server off campus to decide if we can use it
- Need to begin work on the Gantt chart and find software that creates them.

Meeting Type: Tutor Meeting Date: 17/01/2020

Attendees:

Soham Jaiswal
Barry O'Connor
Kieran Olney
Will Peters
Nick Murray-Hill

Absent:

Zach Russell

Topics Discussed:

- Discussed the final definition document

Meeting Type: Group Meeting Date: 24/01/2020

Attendees:

Soham Jaiswal
Barry O'Connor
Kieran Olney
Will Peters

Apologies

Nick Murray-Hill (work commitments)

Absent:

Zach Russell

Topics Discussed:

- Concept Map

Actions:

- Everyone adding to the Concept Map so that it can be finalised
- Nick to work on Database Diagram
- Kieran to work on Class Diagram

Meeting Type: Group Meeting Date: 31/01/2020

Attendees:

Barry O'Connor
Kieran Olney
Nick Murray-Hill

Apologies

Soham Jaiswal (multiple submission deadlines)
Will Peters (visiting family)

Absent:

Zach Russell

Topics Discussed:

- Class and Database Diagram - we worked through the classes together during the meeting

Actions:

- Nick to finalise Database Diagram
- Kieran finalise on Class Diagram

Meeting Type: Group Meeting Date: 31/01/2020

Attendees:

Barry O'Connor
Kieran Olney
Nick Murray-Hill
Soham Jaiswal
Will Peters

Absent:

Zach Russell

Topics Discussed:

- Discussed Database, Class and Mockups as well as plans for the next 3 weeks before the next deliverables are needed

Actions:

- Nick to finalise Database Diagram and training plan for C#
- Will to work on Use Cases
- Soham to work on Concept Map
- Barry to work on Sequence Diagram and Front End HTML

- ALL to take part in training before 28th Feb

Meeting Type: Group Meeting Date: 07/02/2020

Attendees:

Barry O'Connor
Kieran Olney
Nick Murray-Hill
Soham Jaiswal
Will Peters

Absent:

Zach Russell

Topics Discussed:

- Discussed the Progress review.
- Discussed the first draft of the Sequence Diagram, Use Case Diagram use Use Cases.

Actions:

- All members to update diagrams as per discussions.
- All members to create feedback on the front end of the application.

Meeting Type: Group Meeting Date: 14/02/2020

Attendees:

Barry O'Connor
Kieran Olney
Nick Murray-Hill
Soham Jaiswal
Will Peters

Absent:

Zach Russell

Topics Discussed:

- Discussed what needed to be done for training team members in C# and what format that should take.
- Discussed the feedback to the front end and came up with a list of amends that needed to be done.

Actions:

- Barry to work on creating front end core HTML, the team felt that having a dark mode would be a good addition.
- Nick is to spend time preparing training materials so that training can begin the following week.

Meeting Type: Group Meeting Date: 21/02/2020

Attendees:

Barry O'Connor
Kieran Olney
Nick Murray-Hill
Soham Jaiswal
Will Peters

Absent:

Zach Russell

Topics Discussed:

- The team had a feedback session on the final HTML with both themes present.
- Nick presented feedback materials and talked the team through those briefly.

Actions:

- Barry and Nick are to work on creating a login screen and database to understand the process of front end / api communication and to create a coding standard.
- All other members to enter into C# training and to work on the example task set by Nick.

Meeting Type: Group Meeting Date: 28/02/2020

Attendees:

Barry O'Connor
Kieran Olney
Nick Murray-Hill
Soham Jaiswal
Will Peters

Absent:

Zach Russell

Topics Discussed:

- The team discussed the Interim Software Review.
- The team discussed options for server hosting and which databases would be best to use.
- The team have completed training and any final questions regarding this were dealt with.

Actions:

- Barry to look at server hosting options and create Event forms.
- Nick to look at database and server side setup options .
- Soham to create some dummy events for testing purposes.
- Kieran to research how to implement Event functionality in C#.

Meeting Type: Group Meeting Date: 06/03/2020

Attendees:

Barry O'Connor
Kieran Olney
Nick Murray-Hill
Soham Jaiswal
Will Peters

Absent:

Zach Russell

Topics Discussed:

- Further training related issues or difficulties in adjusting to C#.
- Discussed HTML RESTful API's, html methods (GET, PUT, POST) as most of the team were new to these.

Actions:

- Barry to keep one step ahead of the group and ensure the front end forms were available when the back end functionality required them.
- Will to help Barry with HTML forms
- Kieran to implement Event functionality.
- Nick to implement User functionality

Meeting Type: Group Meeting Date: 13/03/2020

Attendees:

Barry O'Connor
Kieran Olney
Nick Murray-Hill
Soham Jaiswal
Will Peters

Absent:

Zach Russell

Topics Discussed:

- Discussed the following week's Software Review and what functionality we could implement for that.
- Discussed the research from the previous week.
- Discussed how to implement Notifications that happen at set intervals.

Actions:

- Will to help Barry with HTML forms.
- Barry to research Web Events and finalise calendar navigation.
- Nick to research thread/timer option for Notifications.
- Kieran to implement Event functionality.
- Nick, Kieran, Soham, Barry, Will to work on classes for the various functionality they have implemented.

Meeting Type: Group Meeting Date: 20/03/2020

Attendees:

Barry O'Connor
Kieran Olney
Nick Murray-Hill
Soham Jaiswal
Will Peters

Absent:

Zach Russell

Topics Discussed:

- Discussed the Software Review.
- Most of the team had moved home because of COVID-19 so discussed options for meetings online.

Actions:

- Barry to set up MS Teams group for meetings.
- Soham & Will to implement Email and Notification functionality.

Meeting Type: Group Meeting Date: 27/03/2020

Attendees:

Barry O'Connor
Kieran Olney
Nick Murray-Hill
Soham Jaiswal
Will Peters

Absent:

Zach Russell

Topics Discussed:

- Discussed Import/Export functionality and decided that it was too complex to implement at that point.
- Discussed the inherent issues with a Language Filter and decided not to continue with that idea.

Actions:

- Barry, Will, Kieran, Nick, Soham to finalise any outstanding work and begin preparation for documentation and testing