

University of Plymouth

School of Engineering, Computing, and Mathematics

COMP3000
Computing Project
2020/2021

Back to Work

Evan Ward

10616392

BSc (Hons) Computer Science

Acknowledgements

I would like to thank my family for the continued support throughout this project, helping me to stay focussed and motivated.

Abstract

Households across the world, whether that be families or shared homes, tend to struggle to complete chores throughout the home. This is due to a lack of motivation (especially with children), lack of time or lack of organisation. This project aims to solve this problem by providing a universal interactive website accessible by anyone with any device that has access to a web browser.

The main body of this report discusses the project in full, outlining the project's background, aims and objectives and a discussion about the method of approach. As well as this, this report covers the legal, social and ethical issues that encompass the project, an outline of the project management, the technologies used and system architecture and design. This report also documents the development process of the project and the solutions to some hurdles that entailed.

The final three sections of this report evaluate the final outcome of the project as a whole, outlining the successes and issues that were encountered. Here, the initial aims and objectives are evaluated as well as the developer's performance throughout the project.

At the end of this report, the references used within the main body can be found, along with an appendix containing all relevant documents relating to the project.

Table of Contents

Acknowledgements	2
Abstract	2
1 Introduction	6
1.1 Background & Existing Solutions	6
1.1.1 Background	6
1.1.2 Existing Solutions	6
1.2 Aims, Objectives and Deliverables	6
1.2.1 Project Aims	7
1.2.2 Project Objectives	7
1.2.3 Project Deliverables	7
2 Methods of Approach	8
2.1 Project Structure	8
2.2 Agile Project Management	8
2.3 Risk Assessment	9
2.4 Minimum Viable Product, Optional Requirements & Final Solution Limitations	10
2.4.1 Minimum Viable Product	10
2.4.2 Optional Requirements	11
2.4.3 Final Solution Limitations	11
2.4 Tools & Technologies	11
2.4.1 HTML, CSS & PHP	11
2.4.2 JavaScript	11
2.4.3 MySQL	11
2.4.4 WAMP	12
2.4.5 Amazon Web Services	12
3 Legal, Social & Ethical	13
3.1 Legal	13
3.1.1 Laws	13
3.1.2 Licenses & Intellectual Property	13
3.2 Social & Ethical	14
4 Project Management	15
4.1 Microsoft Office 365 Planner	15
4.2 Agile Methodology	15

4.2.1 Product Backlog	16
4.2.2 Sprint Backlog	16
4.2.3 Increments	16
4.3 GitHub	16
5 Architecture & Design.....	18
5.1 Architecture.....	18
5.2 Planning.....	19
5.2.1 Functional Requirements	19
5.2.2 Non-Functional Requirements.....	19
5.2.3 User Stories	19
5.2.4 Risk Assessment.....	19
5.3 Design	20
5.3.1 Entity Relationship Diagrams	20
5.3.2 Storyboards.....	22
5.3.3 UML Diagrams	24
5.3.4 Use Case Diagrams	25
5.4 Design Principles	25
5.4.1 Responsiveness.....	25
5.4.2 User Focused Design.....	27
5.4.3 Don't Repeat Yourself	27
5.4.4 Security	27
6 Project Development.....	28
6.1 Sprint Zero: Project Initialisation	28
6.2 Sprint One: Website & Database Design.....	28
6.3 Sprint Two: User Accounts	29
6.4 Sprint Three: Chores	31
6.5 Sprint Four: Rewards System.....	33
6.6 Sprint Five: Calendar	35
6.7 Sprint Six: Clean Up & Testing	37
7 Usability Study	39
7.1 Results.....	39
8 End-Project Report.....	42
8.1 End-Project Summary.....	42

8.2 Project Changes	42
9 Project Post-Mortem.....	43
9.1 Project Management Approach Evaluation.....	43
9.2 Technologies Evaluation.....	43
9.3 Developer Performance Evaluation	43
9.4 Future Work.....	44
10 Conclusion	44
References.....	45
Appendix	47
Appendix 1: User Guide.....	47
Appendix 2: Project Management.....	61
2.1 Product Vision	61
2.2 Functional Requirements	61
2.3 Non-Functional Requirements	63
2.4 User Stories	64
2.5 Risk Assessment.....	65
2.6 Storyboards.....	66
2.7 UI-Flow Charts	73
2.8 Use Case Diagram.....	80
2.9 Sprint Plans.....	81
2.10 Usability Study	95

Word Count: 9996

GitHub: <https://github.com/EvanWard29/COMP3000-BackToWork.git>

Website: <https://ec2-52-56-217-123.eu-west-2.compute.amazonaws.com>

Video Demonstration: <https://youtu.be/fJ-Kdh1ITcc>

Backlog:

<https://tasks.office.com/live.plymouth.ac.uk/Home/PlanViews/pqzmBFL2ZEK9Pic2CDigSZYAEeat?Type=PlanLink&Channel=Link&CreatedTime=637566766037770000>

1 Introduction

1.1 Background & Existing Solutions

1.1.1 Background

This project is designed to support households who struggle to keep their houses clean and tidy on a day-to-day basis and want to be a bit more organised around the home. This can apply to families or student/shared homes. As any person knows, doing chores is not fun and can be a daunting task at times. Children especially find it difficult to complete chores because they may feel it is a waste of their time which could be spent doing more exciting things that interest them. The same can be applied to some adults, however most adults see the importance of doing chores and just get on with it. From personal experience growing up with siblings, chores were one thing that was always disliked, and to this day they can still be tedious.

Some families may also struggle to stay organised with assigning tasks. Methods such as displaying paper-based or whiteboard charts in the kitchen may not always work due to them being too tedious to physically write up a plan for the week or because they are easily destroyed. In an increasingly digital world, outdated methods of organising chores through paper-based and whiteboard charts have become obsolete. Every family nowadays has some access to the internet through various devices such as mobile phones, tablets or PCs. This project aims to take advantage of this by keeping families organised and digitally connected through a single website accessible through any device with an internet browser.

1.1.2 Existing Solutions

There are numerous mobile applications that offer some form of chore rosters that can be downloaded and installed from app stores. Each have similar features, such as the ability to assign chores to members of a group or offer rewards as an incentive for completing chores. However, these applications are restricted to mobile phones only, there are very few websites available on the internet that offer this kind of service. This project aims to fill that gap by offering a unique website that functions like mobile versions but with additional features and free of charge. By creating a full-stack website instead of a mobile application, this service can be accessed at any time through any device with an internet browser, making it portable and easy to use.

1.2 Aims, Objectives and Deliverables

1.2.1 Project Aims

The aim of this project is to produce a fully functioning full-stack website that offers the ability for users to add members to a household group where they can then assign chores to each member to complete within a given time.

This project is mainly aimed at families who struggle to organise and complete chores around the house. This project is also targeted at shared student homes who wish to have some form of chore roster to split chores amongst housemates.

Users will be rewarded with points for completing chores and can be redeemed for rewards of their choice when enough points have accumulated. If chores are not completed on time, points will be deducted from the user.

Users will also be able to set up a calendar that will be synced with the group. On this calendar, users can plan their week ahead by creating new events on the calendar. Users will also be able to view when their assigned chores need to be completed by on this calendar.

1.2.2 Project Objectives

The following project objectives were established at the beginning of the project:

- Design a full stack interactive website that is user-friendly using storyboards.
- Set out the structure of a relational database for storing user data using entity relationship diagrams.
- Use HTML and CSS to implement the designed website.
- Setup a relational database using MySQL.
- Implement the functionality of the website using PHP as the server-side language and JavaScript as the client-side language.
- Deploy the website and database to Amazon Web Services for hosting.

1.2.3 Project Deliverables

The final deliverables of the project will consist of a fully functioning full-stack website and a project report outlining the process of designing, implementing and deploying of this project.

2 Methods of Approach

2.1 Project Structure

The structure of this project can be broken down into three parts:

- Planning
- Development
- Analysis

The planning phase at the start of the project was used to identify the problem and outline how a solution to this problem would be approached. From this, an appropriate project management approach was selected that would be used to structure the development of the solution and which technologies would be used to develop the solution. This provided a clear guide as to how the solution would be designed and implemented in the development phase.

The development phase was the execution of the planned approach to the problem identified in the planning phase. The first step of this phase was to design the solution using storyboards, entity relationship diagrams and UML diagrams. The second step was to implement the designed solution. The third step was to test the implemented solution through usability studies to get valuable feedback that would be used to perfect the solution.

The analysis phase is the last phase. This phase would evaluate the project as a whole in the form of this written report.

2.2 Agile Project Management

Agile project management was chosen as the approach to take for structuring this project. Agile project management is an iterative approach to software development where work is delivered in small increments throughout the development process instead of one big launch. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly ((i) Atlassian, 2021).

For this project, scrum was chosen as the agile methodology to be followed. The scrum methodology splits a project's development into three parts:

- The Product Backlog

- The Sprint Backlog
- Increments

The product backlog outlines the master list of work that needs to be done. This is a dynamic list of features, requirements, enhancements, and fixes that acts as the input for sprints ((i) Atlassian, 2021). For this project, my product backlog was outlined using Microsoft Planner.

The sprint backlog is the list of items, user stories, or bug fixes, selected by the development team for implementation in the current sprint cycle ((i) Atlassian, 2021). Scrum breaks the development of a product into short sprints that last 2-4 weeks. Each sprint implements a select number of features chosen at the beginning of each sprint.

Increments are the usable end-product of a sprint ((i) Atlassian, 2021). At the end of each sprint, a review is carried out that goes over what has been completed during the current sprint, any bugs that have been identified and need fixing, and planning for the next sprint cycle.

This methodology has allowed the project to be broken down into five sprints to be worked on over a period of 3-5 weeks each. At the end of each sprint, a sprint review is carried out and planning for next sprint is carried out. Doing this provides a clear guide as which stage the project's development is in and how to go about developing the project.

2.3 Risk Assessment

At the start of the project during the planning phase, a risk assessment was carried out to identify any risks that may affect the project and how to overcome them.

Potential Risk	How to Address Risk
Sprints not being completed on time/unrealistic time management.	Planning sprints ahead of time will allow me to appropriately organise my time to maximise efficiency and complete sprints within a reasonable timeframe.
Developing wrong software functions.	Identifying the most important features being implemented into the application and organising sprints will allow me to prioritise my time in developing different features. Outlining the Minimal Viable Product and focusing on features that meet this requirement will prevent any unnecessary features being developed.
Late changes to requirements.	Identifying the Minimal Viable Product at the start of development will allow me to easily outline what it is I need to do. To avoid any late changes being made, reviewing each sprint at the end of development and making any necessary changes then will prevent any late and unnecessary changes to the requirements being made.
Development too technically difficult.	Ensuring that all requirements are viable, and I am confident that they can be implemented using the available software, as well as my understanding and experience with the software, will prevent any issues where I may become stuck when implementing a feature.

Figure 1 Risk Assessment

2.4 Minimum Viable Product, Optional Requirements & Final Solution Limitations

2.4.1 Minimum Viable Product

The following requirements are the minimum requirements of the solution and are considered to be the Minimal Viable Product:

- Allow users to create groups to add members of their family/household.
- Allow users to create new chores to be completed around the house.
- Allow users to assign chores to a member of their group to complete.
- Allow users to plan ahead by creating a new event on a calendar synced between the group.
- Allow users to view their assigned chores in a table and on the calendar.
- Allow users to earn reward points when completing a chore or deduct rewards when failing to complete a chore.

2.4.2 Optional Requirements

The following requirements are considered to be optional and are only to be attempted if time permits it:

- Allow group members to message each other in a group chat.
- Allow group members to private message others within the same household group and outside their household group.
- Allow users to receive notification reminders for completing chores.

2.4.3 Final Solution Limitations

The following is considered to be the limitations of the final solution:

- The final solution will be a website and so is only accessible through an internet web browser.

2.4 Tools & Technologies

2.4.1 HTML, CSS & PHP

PHP was used as the main backend language as well as HTML and CSS for the website design. This language was chosen because PHP can be easily embedded in HTML (PHP, 2021). PHP was also chosen as the server language due to the developer's experience of working with PHP, as opposed to NodeJS for example where experience is limited.

2.4.2 JavaScript

JavaScript was used as the main frontend language. JavaScript is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else (MDN Web Docs, 2021). JavaScript was chosen as the main front-end language because it is widely accepted by most web browsers and the developers experience and knowledge of using the language was much more developed.

2.4.3 MySQL

A MySQL server was used for storing user data instead of an alternative database such as MongoDB because MySQL allows for relational databases. This allowed the developer to easily implement the entity relationship diagrams that were designed for

each table of the database. MySQL is also an industry standard language that is widely accepted.

2.4.4 WAMP

Windows, Apache, MySQL, PHP (WAMP), is a software stack that acts like a virtual server on your computer (Hostinger Tutorials, 2021). For this reason, as well as the developers extensive experience using this stack, WAMP was chosen as the software stack for this project. WAMP is easy to use as it is a secure locally hosted server that can be easily accessed and does not require files to be uploaded to an external service, saving time and ensuring files are safely stored locally.

2.4.5 Amazon Web Services

Amazon Web Services (AWS) was used for the hosting of the website and MySQL database. AWS has a variety of services that users can use to build and deploy applications. AWS was used to create an EC2 Windows Server 2019 instance running Apache for hosting the website. AWS was also used to create and host a RDS MySQL database server for storing user data securely.

3 Legal, Social & Ethical

3.1 Legal

There are various legal requirements that must be considered and adhered to when designing any application. For this project, there are two important laws that must be considered when building the final solution. These are GDPR (General Data Protection Regulation, 2018) and the Data Protection Act (Data Protection Act, 2018).

3.1.1 Laws

3.1.1.1 General Data Protection Regulation (GDPR)

The final solution of this project will be GDPR compliant. Cookies will be used by the website. In order to comply with GDBR, the website will ask the user for consent for the use of cookies via a popup. Any data that is collected can be fully controlled by the user. Data collected will be stored in a remote secure database. Users will also be able to delete their accounts which will also delete all user data collected.

3.1.1.2 Data Protection Act

The final solution of this project will also comply with the Data Protection Act. The website will only collect data that is required by the website to function correctly. Users will be able to view and modify this data from the 'My Account' page. Users are able to change their email and password but cannot update any other data. All data collected is first validated before being processed in order to ensure the data is correct. Data such as passwords are also encrypted for security reasons. The website is also hosted on a secure encrypted server via Amazon Web Services.

In order to comply fully with the Data Protection Act users must be able to update their information. This is a feature that would need to be implemented if the project were to become commercially viable. However, this is out of the scope of the project and is intended for educational purposes only.

3.1.2 Licenses & Intellectual Property

The final solution will make use of a number of third-party features. All licenses have been listed in the project's 'README.md' file where appropriate. The final solution has made use of three third-party libraries/assets:

- jQuery – licenced under the MIT Licence. jQuery is a plugin for JavaScript and has been used in this solution for the client-side coding of the website. jQuery is free to use.
- Bootstrap – licenced under the MIT Licence. Bootstrap was used for easily styling HTML elements via CSS classes. Bootstrap is free to use.
- Merriweather Font Family – licenced under the Open Fonts Licence (OFL). This font has been used for styling the website. This font is free to use, as long as it is not sold by itself.
- CryptoJS – licenced under the MIT License. CryptoJS is a plugin for JavaScript. This plugin provides a secure way of encrypting data and was used in this solution for encrypting and decrypting user data. CryptoJS is free to use.

3.2 Social & Ethical

One of the biggest social and ethical issues websites face is the collection of user data. User data is collected by ‘Back to Work’ in order to fully function correctly. Data that is collected for the user is the minimum amount needed for website to function. This avoids any issues with collecting any unnecessary data and keeps the social and ethical use of data under control.

Another social and ethical issue is the sharing of user data with third-party businesses. The solution does not make use of any third-party APIs and so all data collected by the solution is not shared with anyone.

Back to Work allows users to create admin accounts. These accounts can only be made by people over the age of 14. This is due to the fact that sensitive user data is collected, and admins of a group have full control of accounts under the age of 14 and so the user needs to be sensible with the data they are handling. If a user is under the age of 14, an account can be made for them via an admin, but the account will be restricted to the group of the admin making the account. Accounts can also be deleted by the admin if need be. This ensures parents have control of their child’s accounts at all times if a family is using the solution.

Test data collected via a usability study must also be acceptable in terms of social and ethical issues. In accordance with the ‘University of Plymouth Ethics Policy’, all data is anonymously collected deleted afterwards. All test data was gathered from students of ‘University of Plymouth’ in line with the ‘University of Plymouth Ethics Policy’.

4 Project Management

4.1 Microsoft Office 365 Planner

Microsoft Planner was the main online project management tool used for this project. Office 365 Planner provides a hub for team members to create plans, organise and assign tasks to different users and to check updates on progress through dashboards (Curry, 2019). Microsoft Planner allows users to create buckets containing tasks that need to be completed. These tasks can be assigned to specific users within a group along with a start date and deadline for completion. Microsoft Planner and other similar project management tools are widely used for Scrum style workflows across the industry.

Within the developer's Planner, six buckets were created, one for the Product Backlog and five for sprints. Each bucket contained tasks that needed to be completed before a specified deadline. This allowed the developer to structure their time wisely and ensure there was enough time to focus on fully implementing a feature within the backlog correctly and efficiently.

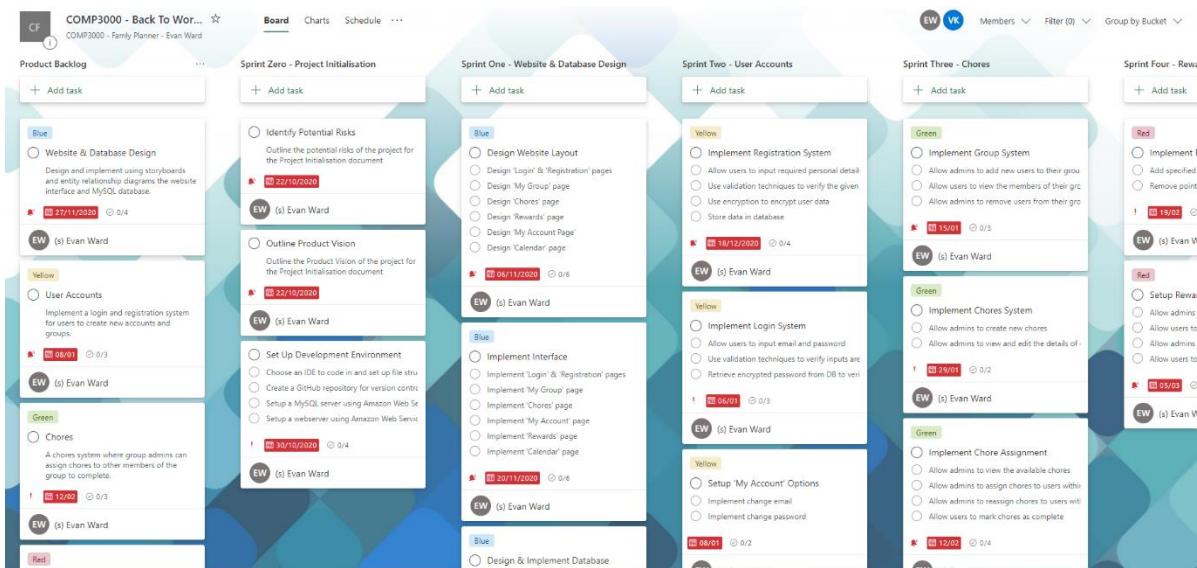


Figure 2 Microsoft Office 365 Planner

4.2 Agile Methodology

Scrum was the chosen Agile Methodology for this project. As such, the project was split into three parts: Product Backlog, Sprint Backlog and Increments.

4.2.1 Product Backlog

Within the developer's Planner, one bucket was created specifically for the product backlog. This bucket contained tasks that represented a feature of Back to Work that needed to be implemented. Each feature was given a colour label, a priority and set a start date and deadline for completion. This allowed the developer to keep track of which features had been implemented already and which features still needed completing.

4.2.2 Sprint Backlog

In order to tackle the features within the product backlog, five sprints were set up that lasted 3-5 Weeks. Each sprint contained multiple tasks that focused on completing one or two features within the backlog, depending on the difficulty of implementing each feature. Each task within a sprint was colour labelled which linked to a feature within the product backlog. This allowed the developer to easily check when a feature could be marked as complete and fully implemented within the product backlog. Creating sprints allowed the developer to structure their time appropriately and efficiently, ensuring there was enough time to fully implement a feature correctly.

4.2.3 Increments

Increments were the final products of sprints. Once a sprint had been completed and merged with the master product, a sprint review was carried out. This review focused on evaluating what went well during the sprint cycle, what could have gone better, and things to consider when moving onto the next sprint cycle. These reviews allowed the developer to reflect on the work that had been done so far and decide whether extra time would be needed for completing other sprints or whether a features priority needed to be changed depending on time constraints.

4.3 GitHub

Version Control is an important aspect of project management as it allows developers to easily structure the development of a product through branches. It also allows developers to revert changes to previous versions of the product if the changes made on the current branch need to be scrapped or in the case of irreversible data loss. GitHub is a widely used version control system across the industry. GitHub allows developers store data remotely, ensuring there is always a backup to the current project in the case of data loss. It is also easy to use through command line or with the desktop app. As such, GitHub was chosen as the version control system for this project.

GitHub allows users to create branches off of a master branch. This allows developers to work on a feature that needs implementing on a separate branch that will not affect the master branch. When a change is made to the product, developers can commit these changes to the current branch. This allows developers to easily revert changes to an older commit if need be.

For this project, the developer created a new branch of the master product whenever work was started on a new sprint from the sprint backlog. Once a sprint had been completed, the current branch would be merged with the master branch and that would become the master product. Overall, there were five branches made, one for each sprint and an additional sixth branch at the end for testing and project clean-up.

5 Architecture & Design

Before starting the development of the project, it was essential to plan the architecture and design of the product to allow for a smooth development process which could easily follow the design plans. This was vital in order to ensure the success of the project. Each aspect of the solution was carefully planned ahead in order to outline the key features required to be able to satisfy the minimum viable product, as well as any other features that were optional and could be introduced if there was enough time or at a later date in the future. During planning, the solution's architecture was chosen, as well as the technologies that would be used to produce the final solution.

5.1 Architecture

WAMP was the chosen software stack for this project. This is due to the developer's experience using the stack and the technologies associated with it. WAMP was also chosen as the project's software stack because it allows developers to host websites locally which made developing and debugging the website much easier. Apache was chosen as the web server to use because it is the most widely used and accepted web server available (wpbeginner, 2021). MySQL was chosen as the database to use because the developer has the most experience using MySQL than they do any other database, such as MongoDB. MySQL is a relational database management system which is widely used alongside web development for storing relational data (123 Reg, 2021). This would make building the database much easier as the developer planned for it to be mainly relational. PHP was the chosen backend server language to use because it is a widely used and open source which allows developers to easily embed code within HTML (PHP, 2021). This backend language was chosen because the developer has the most experience using PHP than any other backend language such as NodeJS. JavaScript was used for the frontend development of the website because it is easy to use and has very useful plugins, such as jQuery, that allow you to easily manipulate elements on the webpage. The below image is the full system architecture of the website.

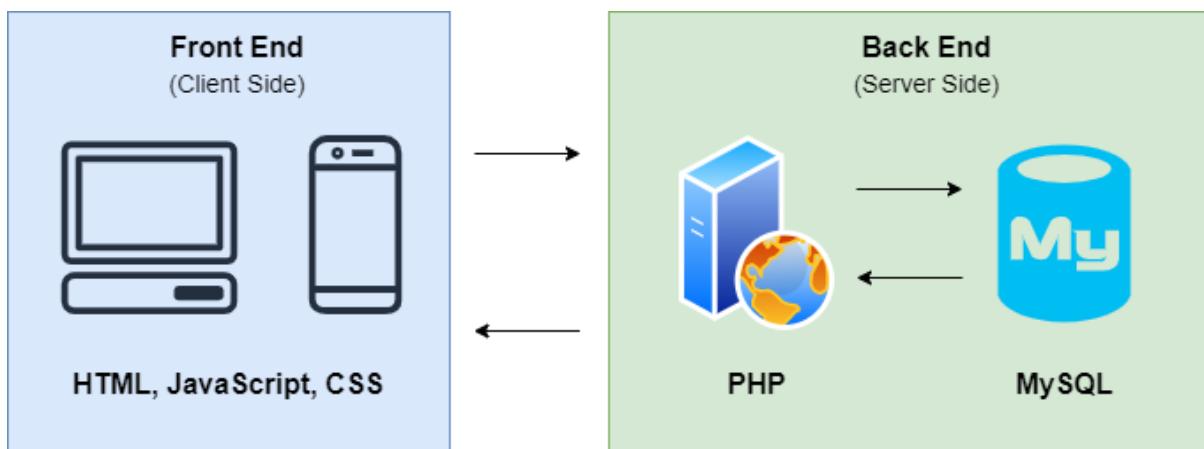


Figure 3 System Architecture of Back to Work

5.2 Planning

5.2.1 Functional Requirements

When planning the project, clear functional requirements, that are available within the appendix of this report, of the final solution were carefully chosen and outlined (2.2 Functional Requirements). This ensured that a working and finished solution that satisfied the minimum viable product could be produced before the deadline. This made planning sprints easier as the developer was able to split the functional requirements amongst the five sprints in order of importance and difficulty, allowing for a smooth implementation of each feature during the development process.

5.2.2 Non-Functional Requirements

Some additional non-requirements were also planned at the beginning of the project. These requirements were not necessary in order to satisfy the minimum viable product but were optional features that would further extend the functionality of the final solution (Non-Functional Requirements). These requirements were only to be implemented if time permitted it or at some point in the future.

5.2.3 User Stories

In order to plan the functional requirements and outline the minimum viable product, the developer generated some user stories that envisioned what a user would want to do with the final product (2.4 User Stories). The definition of user stories is as follows: “A user story is an informal, general explanation of a software feature written from the perspective of the end user. Its purpose is to articulate how a software feature will provide value to the customer” ((ii) Atlassian, 2021). These user stories helped the developer to prioritise each requirement and plan the five sprints appropriately based off of each requirement’s priority.

5.2.4 Risk Assessment

At the beginning of the project, the developer performed a risk assessment (2.5 Risk Assessment). Risk assessments are usually performed during a project’s start-up and are essential within the agile framework and when working with a team within an agile project. This is because teams need to know how to avoid risks or overcome a risk if one is encountered during a product’s development. During the initialisation phase of the project, the developer performed a risk assessment which consisted of the following steps:

- Risk Identification
- Risk Assessment
- Risk Response

During the risk identification step, the developer identified any risks that could occur during the product's development. These risks could have been anything from encountering risks of software bugs or risks of completing sprints past the deadline of the sprint. The next step in the risk assessment was to assess what level of threat the potential risk would be to the project. Some risks were low and would have little impact on the project's development. Other risks were more extreme, such as data getting lost, and would have a much bigger impact on the project's development. The next step was used to identify solutions to the risks and how the developer could avoid encountering the risks in the first place. This process allowed the developer to easily continue development of the project without worrying about unexpected risks that could potentially hinder the development of the product.

5.3 Design

Before starting development of a project, development teams need to plan the architecture of the software to be developed. This means designing the structure of the code, databases and interfaces, etc. This allows development teams to have a clear vision as to what they need to be building and how they should go about doing it.

5.3.1 Entity Relationship Diagrams

Various diagrams are used to design the structure of code, database and interfaces. One of these diagrams is Entity Relationship Diagrams (ERDs). These diagrams are used for designing the structure of databases as well as the relationships of the tables within the database if using a relational database such as MySQL.

During the planning phase of this project, the developer used entity relationship diagrams to define the structure of the relational database that would be used to store user data for Back to Work. The developer had a clear idea of how the data within the tables would relate with one another. Visually representing this idea with an entity relationship diagram made it easier to build the actual database. The below figure is the initial entity relationship diagram when planning the project.

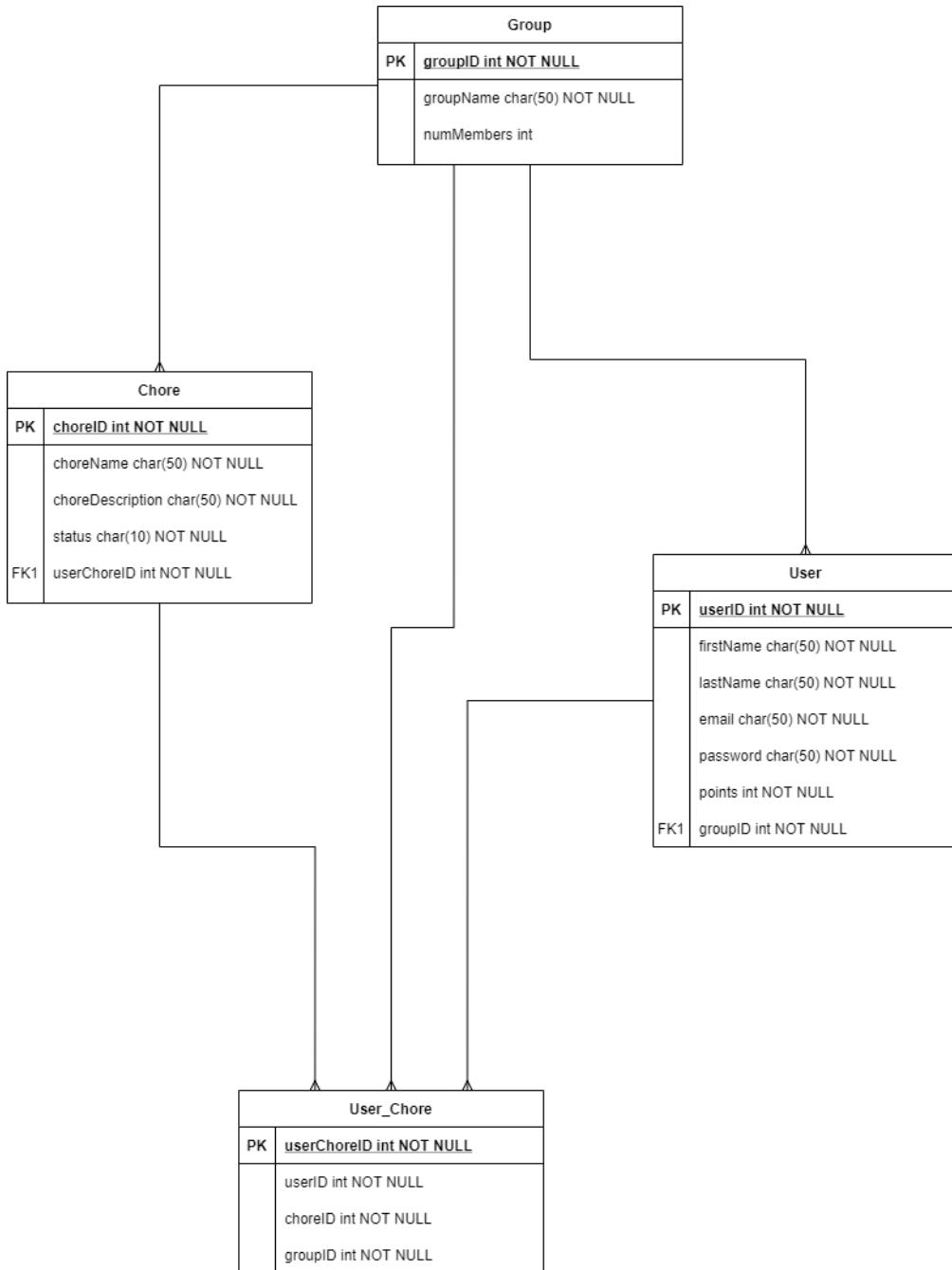


Figure 4 Initial Entity Relationship Diagram

Throughout the development of the project, more tables were added to the database due to having new features and ideas to add to the website. One of these ideas was the calendar. Originally the calendar was to only be implemented if there was enough time for the developer to do so. However, after carefully considering it, the calendar functionality was added earlier due to new ideas that would improve the user experience. As a result, the following figure is the final entity relationship of the system.

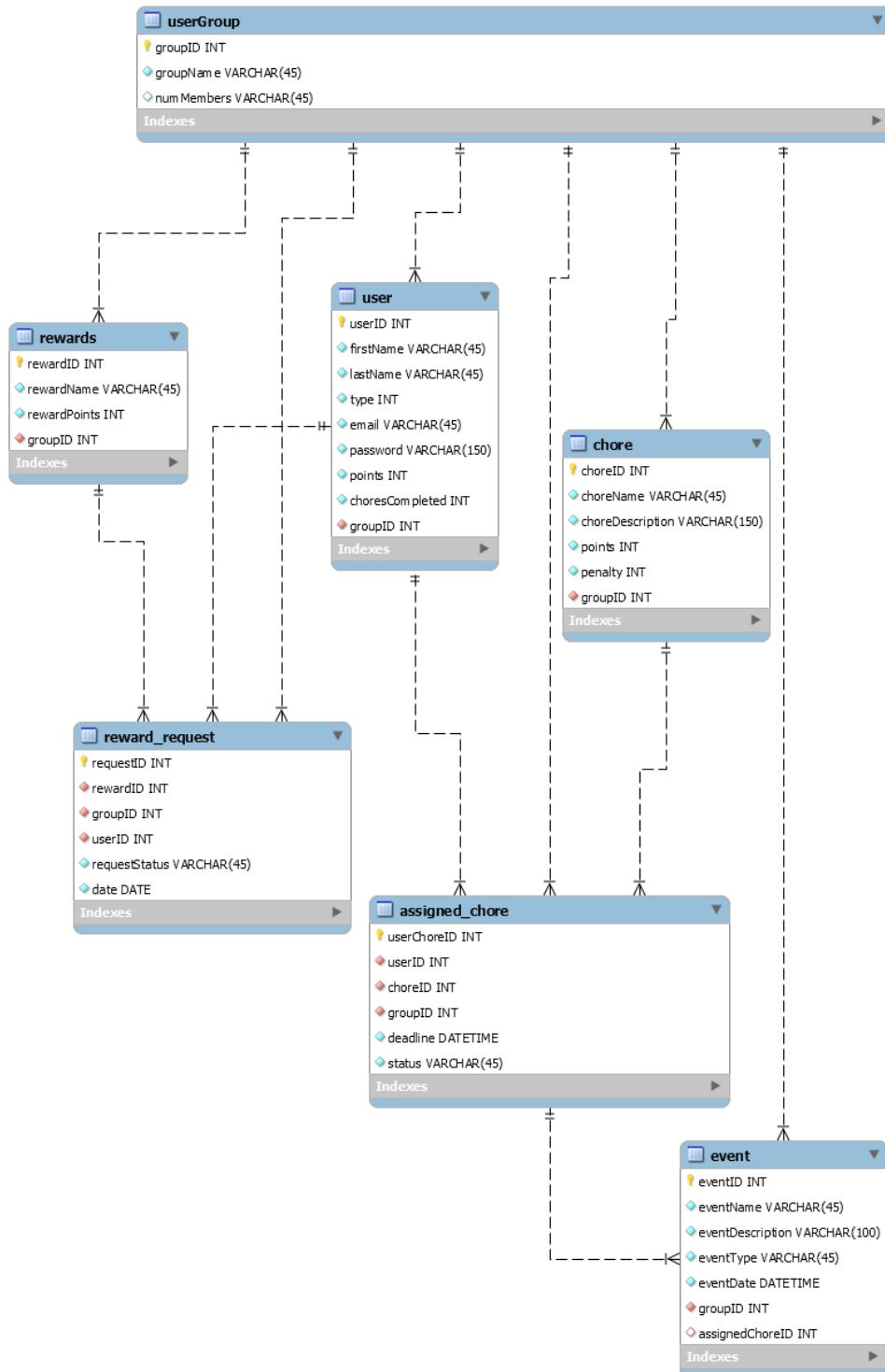


Figure 5 Final Entity Relationship Diagram

5.3.2 Storyboards

Storyboards are another useful diagram that developers use. Storyboards allow developers to design user interfaces of software, whether that be for mobile applications or websites. This makes coding the interfaces easier for developers as they already have a rough design for the interfaces that they need to create, instead

of having to come up with the design on the go whilst coding. Storyboards also allow developers to represent how pages on a website or application will interact with each other through buttons, menu items, etc. They are also used for designing the general theme of the overall application or webpages, making it easier for developers to implement because they have a relatively clear vision of what the colour scheme of a webpage should look like.

During the planning phase, the developer used storyboards to design the interface and general theme of the website. Doing this provided a clear idea of how the developer would implement the HTML elements, such as the navbar, buttons and page structures. Creating storyboards also made styling the HTML using CSS much easier because they knew what the website's theme would look like. A full list of storyboards that was made for Back to Work can be found in the appendix (**Error! Reference source not found.**). Below is one of the storyboards that was created for the website.



Figure 6 'My Group' Page Storyboard

5.3.3 UML Diagrams

Probably the most commonly used diagram by developers is UML diagrams. UML (Unified Modelling Language) diagrams are used to visually represent the different components within a system. This helps developers when coding the application and how the code links together through classes, methods, etc. It also means they can visually see how the system should be structured which makes it clearer to implement within the code.

During the planning phase, the developer used UML diagrams and more specifically UI-Flow diagrams to plan how a user would navigate the website and what each interaction the user had with the website would do. UI-Flow charts are a great way of visually displaying the actions a user can take and what the system will do in response to those actions. Flowcharts and UI-Flow charts are also a great way to visually represent the logic of how the system works. These diagrams are also useful for visualising user stories and so after coming up with some suitable user stories, the developer designed some UI-Flow charts to support them. Below is one of the UI-Flow charts I designed to visually display how a user would interact with the 'Chores' page of the website, a full list of UI-Flow charts made for Back to Work can be found in the appendix (2.7 UI-Flow Charts).

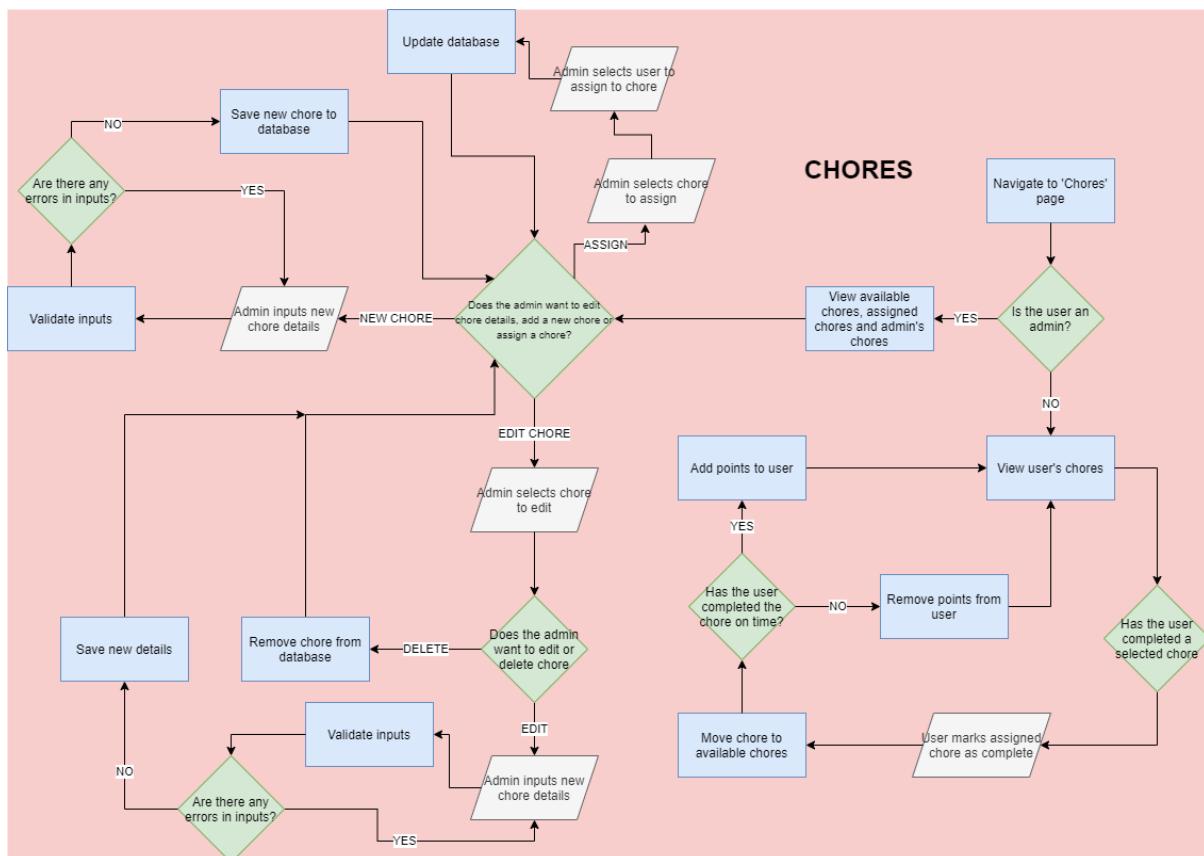


Figure 7 UI-Flow Chart of 'Chores' Page

5.3.4 Use Case Diagrams

Another way for developers to visualise user stories is to design Use Case Diagrams. These diagrams are another great way of visualising how a user would interact with the interface of the system and how the system functions would react. Each function or webpage of the system is usually designed around a storyboard or functional requirement. This makes it easier for developers to implement some of the logic behind the system within the code. During the planning phase, the developer designed a use case diagram alongside the user stories and storyboards so that they had a clear idea of how they could go about implementing the different functions of the website. This diagram can be found in the appendix (2.8 Use Case Diagram).

5.4 Design Principles

When development teams plan projects, it is important for them to consider certain design principles to ensure the product being developed is of high standards. Having universal standards for developers within a team to follow makes working in teams a much easier task as everyone should be contributing work of the same standards. This avoids any issues with others being unable to understand the code or code being structured wrong. Sticking to these design principles will also ensure the final product is suitable for users to use and is user friendly.

5.4.1 Responsiveness

Back to Work is an interactive website that can be accessed by any user on any device that has access to the internet and a web browser. As a result of this, the website needed to be designed to work on not just desktop browsers but also mobile browsers. To do this, the website was designed with responsiveness in mind. When a website is responsive, it allows elements of the webpage to resize, reorder and structure itself based on the device being used, changing the appearance of the website to be more device friendly. Having the website responsive meant the developer did not need to focus on building two versions of the website but instead allowed them to focus on just one responsive site. Below is an example of how the website has been made to be responsive.

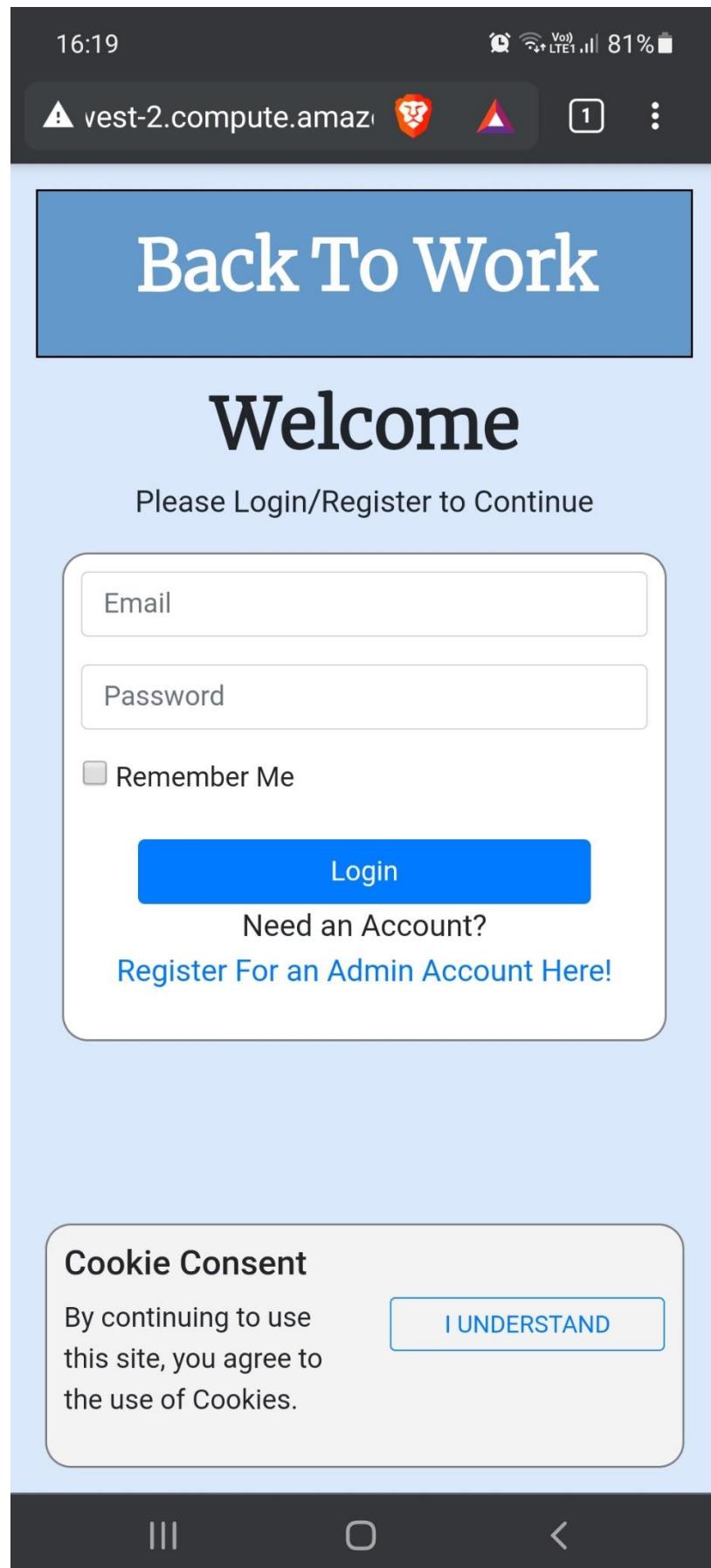


Figure 8 Responsive Login Page

5.4.2 User Focused Design

Throughout the development of the website, the user and their experience of using the website is always kept in mind. This was especially important when it came to the user interface of the webpages. If the user struggles to navigate the website, the user would be unlikely to return and so it was vital that the implementation of the webpages and their interfaces was easy to navigate, visually appealing and user interaction, such as inputting data and validating that data, was as simple as possible in order for the user to get the best experience possible.

5.4.3 Don't Repeat Yourself

The Don't Repeat Yourself (DRY) principle states that duplicate code should be eliminated via abstraction and automation to avoid waste code. Adding wasteful code adds to the maintenance of software and could lead to avoidable bugs (DeVIQ, 2016). During development of Back to Work, this principle was followed to the best of the developer's ability. To do this, code was structured appropriately to avoid messy code and reduce the risk of bugs appearing.

5.4.4 Security

The user's security is the most important aspect of a website. To ensure user data was safe and secure throughout the website, only data that was necessary was retrieved from the database when needed and all sensitive data was encrypted appropriately. By encrypting sensitive data, such as passwords, the user's data is protected when being transmitted from the database to the server and then the client and vice versa. The website also always checks if a user is logged in and if they have the correct permissions to access certain aspects of a page. If a user is not logged in, they are always redirected to the login page and all cookies are erased. All fields that require data input by the user are also validated extensively to ensure the correct data formats are being entered and no malicious data can enter the database and potentially cause serious harm.

6 Project Development

During the development of the project, each functional requirement that was set out during the project initialisation phase was split into five sprints that would last 3-5 weeks. This allowed the developer to prioritise some of the more important features that were necessary to satisfy the minimal viable product. Each sprint was split into smaller tasks to complete along with a checklist of smaller tasks that would need to be completed to finish the main task. This allowed for a clearly structured development process to take in order to complete the current sprint. Each full sprint along with its review can be found within the appendix of this report (2.9 Sprint Plans). Below is an outline of the development process that was taken for each sprint.

6.1 Sprint Zero: Project Initialisation

Before starting work on the project, the project initialisation phase was used to come up with a project idea. From this, the idea of ‘Back to Work’ was born. The project initialisation phase was used to outline the product vision for this project (2.1 Product Vision), set up the development environment, perform a risk assessment (2.5 Risk Assessment) and generate some user stories (2.4 User Stories). Doing this meant the developer was prepared and ready to start planning how they would structure the development of the project in sprints of 3-5 weeks. This gave the developer a clear path to building a final working solution.

6.2 Sprint One: Website & Database Design

Sprint one was the start of the development of Back to Work and was completed during the first month of product development.

Objective

The objective of sprint one was to design and implement the different webpages that would be used for the website as well as the database structure. By the end of this sprint, a complete first draft of the website and database was expected.

Development

During sprint one, the developer focussed on designing the website interface using storyboards and the user stories as a guide. Each functional requirement was broken down into webpages, such as a chores page for assigning chores and a rewards page for redeeming rewards. They then designed a rough layout for each webpage that they thought was appropriate. By doing this, they were able to establish a general colour theme and layout for the website.

During sprint one, the developer also focussed on designing the structure and data for the relational database that would be used by creating an entity relationship diagram. These diagrams allowed them to get a rough idea of how the data within the different

tables would be stored and how it would relate to other tables. During this time, the developer also made the decision to use a remote MySQL server hosted by Amazon Web Services to host the database securely.

Once the website and database's structure had been designed, the developer started work on implementing them. Using HTML & PHP, they implemented first draft webpages that would be used. Then then setup the remote MySQL server and implemented the different tables that were planned to be used. During this stage, the developer made sure there was enough room for them to add new features easily to the webpages and database, in case during the development of the project new ideas for features developed or existing ideas developed further.

6.3 Sprint Two: User Accounts

Sprint two was started right after the completion of sprint one and lasted 6 weeks. This was due to taking a break during the Christmas period.

Objective

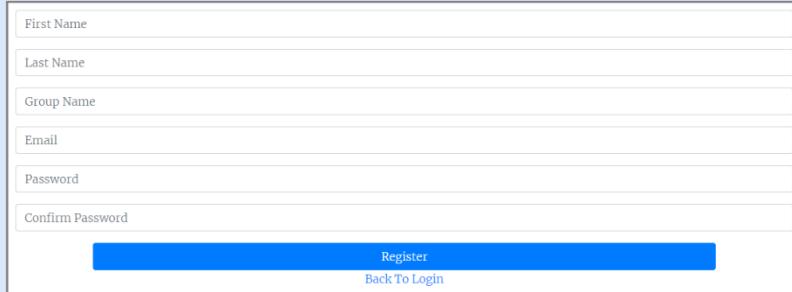
The objective of sprint two was to have a fully integrated registration and login system that supported the creation of admin and user accounts. This feature was of high priority due to the website being user focussed and needing a user to be logged in for it to operate properly.

Development

The first step of sprint two was to set up the registration system. To do this, the developer used a classic registration form containing all relevant fields for the user to input the required user details, such as an email, password and name. They then used JavaScript to validate this data client-side. By validating the data client-side it removed any possible risks of malicious or incorrect data reaching the server. Before sending the data to be processed by the server and added to the database, the developer ensured all sensitive data, such as passwords, was encrypted using a plugin for JavaScript called 'CryptoJS'. They then used jQuery's POST function to send the data to an API that would add the data to the database.

Back To Work

Registration



The registration page features a light blue header with the text "Back To Work". Below it is a white content area with a title "Registration". On the left side of the content area, there are six input fields labeled "First Name", "Last Name", "Group Name", "Email", "Password", and "Confirm Password". Each field has a placeholder text inside. On the right side, there is a blue rectangular button with the word "Register" in white. Below the button is a link "Back To Login".

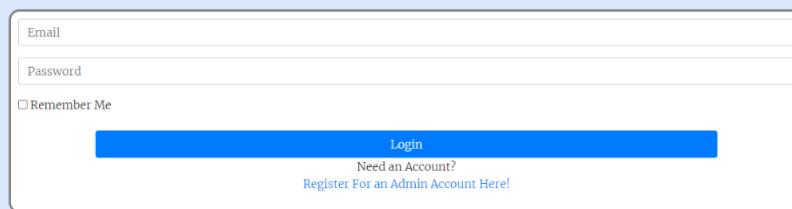
Figure 9 Registration Page

The second step of sprint two was to implement the login functionality. A simple login form was made for users to input their email and password. JavaScript and jQuery were then used to validate the input data. To validate the user's inputted credentials, an API was used to query the database to find the user's email. If the email is found, the encrypted password stored in the database is returned and compared with the user inputted password. If they match, the user is logged in.

Back To Work

Welcome

Please Login/Register to Continue



The login page features a light blue header with the text "Back To Work". Below it is a white content area with a title "Welcome". A message "Please Login/Register to Continue" is displayed. There are two input fields: "Email" and "Password", each with a placeholder text. Below the fields is a checkbox labeled "Remember Me". At the bottom is a large blue rectangular button with the word "Login" in white. Below the button, there is a link "Need an Account?" and another link "Register For an Admin Account Here!".

Figure 10 Login Page

The final step of sprint two was to set up the options for user accounts. These options included the ability for users to change their email and password or delete their

account and group. This feature was of highest importance due to ensuring the user has full control of their data and following GDPR rules.

The screenshot shows the 'My Account' page of a web application. At the top, there is a navigation bar with links: 'My Group', 'Chores', 'Calendar', 'Rewards', 'My Account', and 'Logout'. Below the navigation bar, the title 'Back To Work' is displayed in a blue header. The main content area is titled 'My Account'. It contains two sections: 'My Details' and 'Options'. The 'My Details' section displays the user's name ('Evan Ward'), email ('Evan29Ward@gmail.com'), and points ('12'). The 'Options' section contains four buttons: 'Change Email' (blue), 'Change Password' (blue), 'View Past Reward Claims' (blue), and 'Disband Group' (red). The 'Disband Group' button is highlighted with a red background.

Figure 11 My Account Page

6.4 Sprint Three: Chores

Sprint three was started after the completion of sprint two and lasted 5 weeks.

Objective

The objective of sprint three was to have a fully functional chores system where users could be added to a group and assigned a chore for them to complete within a given timeframe.

Development

The first step of sprint three was to implement a group system that allowed users who registered for an account to create a new group and be able to add new users to that group. When a new user registers for an account, they automatically become an admin. This is due to the age restriction of 14+ that was put in place to avoid any issues with younger users having too much control of the system. Admins are able to create new user accounts through the 'My Group' page which are then automatically added to the admins group. They also have the option to assign an admin role to these new users if they wish to do so.

The screenshot shows a web application interface for managing a group. At the top, there is a navigation bar with links: 'My Group', 'Chores', 'Calendar', 'Rewards', 'My Account', and 'Logout'. Below the navigation bar, a blue header bar contains the text 'Back To Work'. The main content area has a light blue background and features a title 'My Group' above a table. The table has three columns: 'Name', 'Chores Completed', and 'Points'. The data in the table is as follows:

Name	Chores Completed	Points
Annabel	6	18
Evan	18	12
Alison	4	10
Richard	3	9
Rosie	3	8
Poppy	4	7
David	1	5
Emilie	9	4
Zachary	2	4

At the bottom of the table, there is a blue button labeled 'Register New Group Member'.

Figure 12 My Group Page

The second step of sprint three was to implement the chores system. Admins have the ability to create new chores for the group which they can then assign to users within their group. To do this, a simple form was made where the admin can input the name and description of the chore with a points and penalty value. This data is then validated using JavaScript and jQuery then sent to an API to be added to the database. Admins can then view the available chores that have not been assigned to a user yet. To assign a chore, the admin chooses a chore and selects a user within the group to assign the chore to. Admins have the ability to view all assigned chores within the group as well as their own. Normal users can only view the chores that they have been assigned to. Admins also have the ability to edit a chore's details and reassign a chore to another user within the group.

Available Chores	Assigned Chores	My Chores
Vacuum	Put Away Shopping Assigned: David	Dishes
Dusting		Tidy Kitchen
Drying	Mop Floor Assigned: Annabel	
Tidy Livingroom		
Clean Fish Tank		
Feed Animals		
Rubbish Night		
Change Beds		

New Chore

Figure 13 Chores Page

6.5 Sprint Four: Rewards System

Sprint was started after the completion of sprint three and lasted 3 weeks.

Objective

The objective of sprint four was to have a fully implemented rewards system where users could earn points for completing chores on time and lose points for completing chores past their deadline. User would then be able to redeem those points for a physical reward.

Development

The first step of sprint four was focussed on implementing a way for admins to add new rewards for users to redeem. This was done using a form, JavaScript and jQuery for validation and an API to add the new reward to the system. These rewards were group specific and could only be redeemed if a user had enough points.

Rewards

Reward	Points
£10	100
Sweets	10
New Reward	
Reward Name:	
Points Cost:	
	10 30 15 5 50
	Add Cancel

My Points: 12

Selected Reward:

Select A Reward

Redeem

Reward Requests

New Reward

Figure 14 Form to Add New Reward

The second step of sprint four was to implement the points themselves. Each new user starts with zero points. When a user completes an assigned chore, an API is used to check if the chore being completed was completed on time or late. If the chore is completed on time, the points value of the chore is added to the user's points in the database and the assigned chore is removed from the database and moved to the available chores. If the chore is completed late, the same process happens but instead of adding the points, the penalty value is subtracted from the user's points instead.

Rewards

Reward	Points
£10	100
Sweets	10
Takeaway Of Your Choice	30
Ice Cream	15
Stickers	5
Small Toy	50

My Points: 12

Selected Reward: You Do Not Have Enough Points For That! Select A Different Reward.

Takeaway Of Your Choice

Redeem

Reward Requests

New Reward

Figure 15 Select Reward.

The last step of sprint four was to implement the rewards redemption process. Users can redeem their points for available rewards set up by an admin. A reward request is then added to the database for admins to either approve or decline the request. If the

request is approved, the request status is set to 'PROCESSED'. If the request is declined, the user's points are refunded, and the status is set to 'PROCESSED'.

The screenshot shows a web application interface for managing rewards. At the top, there is a navigation bar with links: 'My Group', 'Chores', 'Calendar', 'Rewards', 'My Account', and 'Logout'. Below the navigation bar, the title 'Back To Work' is displayed in a blue header. The main content area is titled 'Rewards' and contains a table listing various rewards and their corresponding points:

Reward	Points
£10	100
Sweets	10
Takeaway Of Your Choice	30
Ice Cream	15
Stickers	5
Small Toy	50

Below the table, a message says 'My Points: 12'. A section titled 'Selected Reward:' contains a dropdown menu labeled 'Select A Reward' and a blue button labeled 'Redeem'.

Figure 16 Rewards Page

6.6 Sprint Five: Calendar

The final sprint was started after the completion of sprint four and lasted 5 weeks.

Objective

The objective of sprint five was to implement the functionality of the calendar by integrating assigned chore deadlines with the calendar and allowing admins to add new events to the calendar.

Development

The first step of sprint five was to build the calendar itself. To do this, the developer designed a calendar class that would generate a calendar for the current date. When the calendar page is loaded, the calendar class builds the calendar on the page for the user to view.

Back To Work

Group Calendar

Prev		2021 Mar					Next	
Mon	Tues	Wed	Thur	Fri	Sat	Sun		
1	2	3	4	5	6	7		
8	9	10	11	12	13	14		
15	16	17	18	19	20	21		
22	23	24	25	26	27	28		
29	30	31						

Figure 17 Calendar Page

The second step for sprint five was to implement the events functionality. jQuery was used to detect when a user clicks on a date within the calendar. A Bootstrap modal popup then appears for the user to either view events for that date or add a new event to that date. User can view events but only admins can add events to the calendar. To do this, JavaScript and jQuery are used to fetch and validate the inputted data and then send the data to an API to save the event to the database. To show the events on the calendar, an API is called by the server to fetch the events for the group. The dates with events are then coloured to stand out and represent an event. When a user clicks on a date with an event, the event is displayed within a table.

Back To Work

Group Calendar

Prev		2021 Apr					Next	
Mon	Tues	2021-04-15				Sat	Sun	
		Events				3	4	
5	6	Event Name	Event Description			10	11	
12	13	Gardening Club	Poppy and Rosie afterschool gardening club			17	18	
19	20	Chores	No Chores Added			24	25	
26	27							

[New Event](#)
[Close](#)

Figure 18 Calendar Event

The final step of sprint five was to integrate the deadlines of assigned chores with the calendar. To do this, a new calendar event was made whenever a chore is assigned. This event is then saved to the database along with the assigned chore's id. Chore events are then displayed as an event on the calendar but within a separate table when viewed. Users of a group are able to see all chores assigned to other users within the group. Each chore event shows the details of the chore, including the name of the chore and who it is assigned to. When a chore is completed, the chore event is removed from the database.

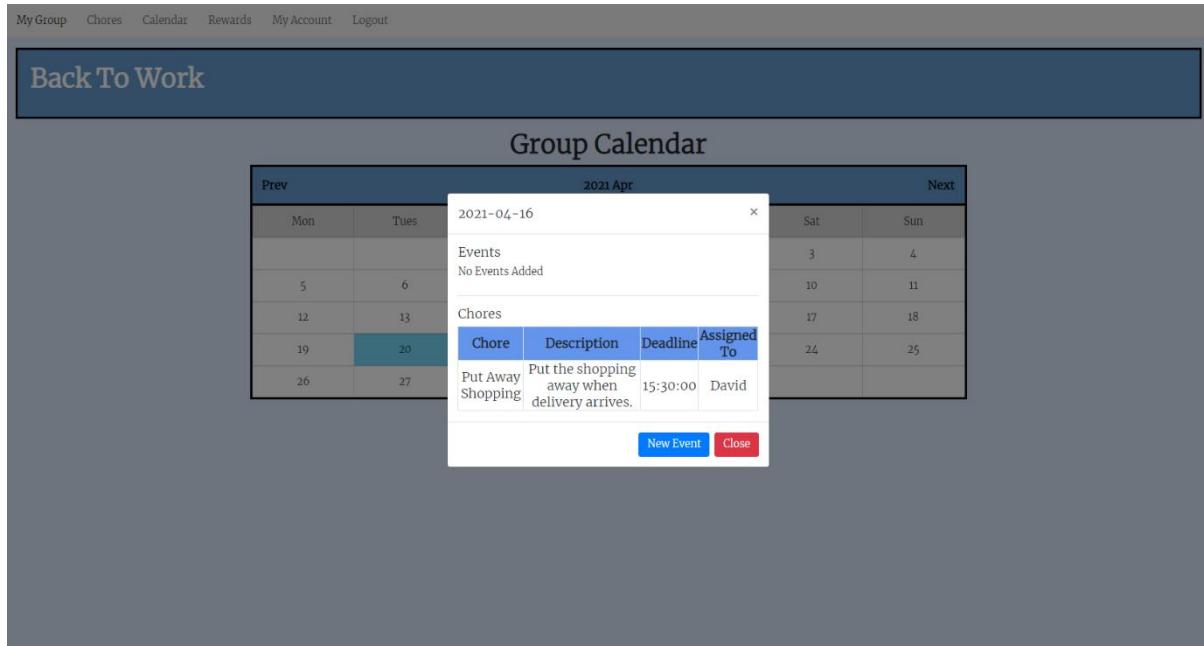


Figure 19 Chore Event

6.7 Sprint Six: Clean Up & Testing

Sprint six was started after the completion of sprint five and lasted until the end of the project deadline. Sprint six was used only for tidying up the code and performing a usability study to test the website.

Objective

The objective of sprint six was to ensure the file and code structure was organised appropriately. Sprint six was also used for gathering feedback on the website through a usability study.

Development

Throughout the development of Back to Work, the developer made sure to keep the file structure and code as clean and tidy as possible, ensuring they used comments to explain the code as well as possible. The aim of sprint six was to just go through the code, adding additional comments and ensuring code is indented correctly and

variables are named appropriately. Doing this also gave the developer the opportunity to find any bugs that may have been missed and remove any duplicated code.

The usability study gave the developer the opportunity to receive valuable feedback that could be used to improve the user experience of using Back to Work.

7 Usability Study

Usability studies are used by developers to gain valuable feedback from test users. The main goal of a usability study is to identify problems within an application, uncover opportunities for improvement and learn about the target user's behaviour and preferences whilst using the application (Moran, 2019).

During sprint six, the developer performed a usability study to receive feedback on Back to Work. The study was a simple questionnaire that asked the user to perform a number of tasks that tested a feature of the website and provide feedback on this task. The user then had the opportunity rate the difficulty of the task performed, provide more written feedback on the task if they wished to do so, provide feedback on the design and layout of the website and provide feedback about the overall experience of using Back to Work and any suggestions they may have for improving the experience. All responses received were anonymous in accordance with the University of Plymouth's Ethics policy.

7.1 Results

Below are some of the results of the usability study. The full questionnaire (2.10.1 Questionnaire) and all responses are available in the appendix (2.10.2 Results).

Editing a chore's details was:

6 responses

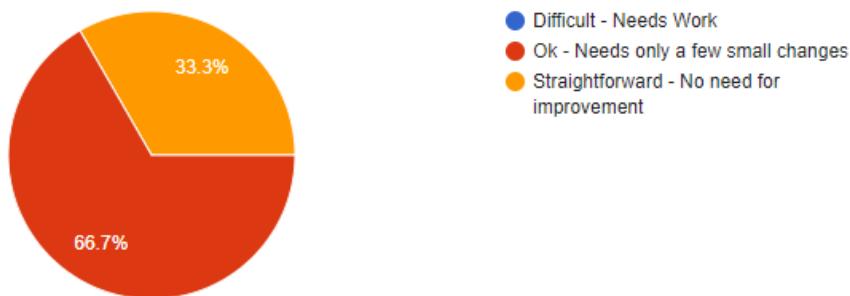


Figure 20 Editing Chore Details – Responses

Some of the feedback that was received for editing chore details said that the popup that appears when selected a chore should be made cleaner when editing the chore. For example, the below response said that I should hide the option for assigning a chore when editing it instead of simply disabling it.

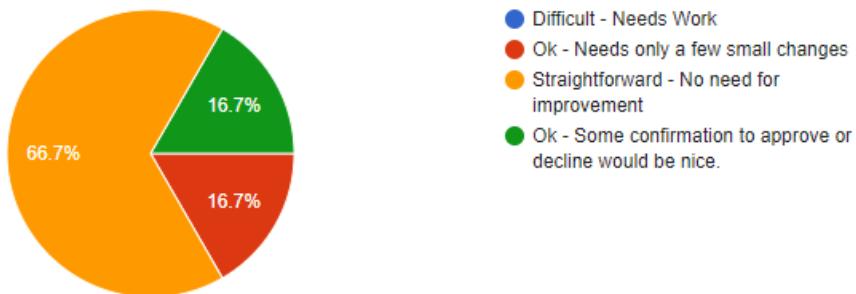
Easy to use, maybe make editing chore details a bit more clean, eg. hide the chore assign options.

Figure 21 Suggestion for Changing the Way Chores Are Edited

After receiving the above response, the developer changed the way chores were edited to make the UI cleaner and tidier. Some similar responses are shown below.

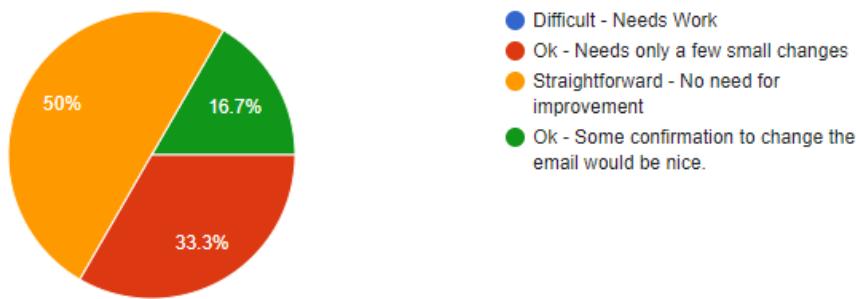
Approving/Declining a reward was:

6 responses



Changing your email was:

6 responses



Changing your password was:

6 responses

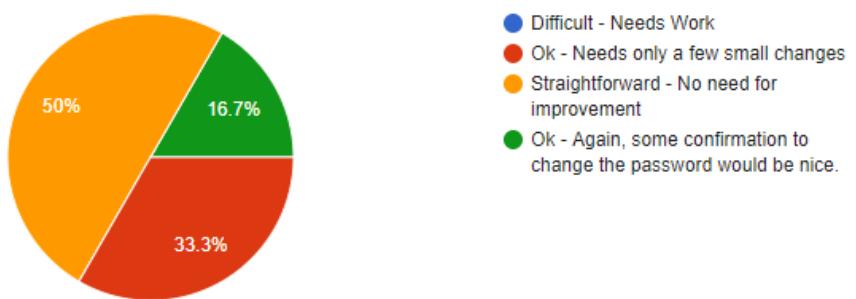


Figure 22 Responses Relating to Confirmation Popups

The above responses were very clear that some of the user interactions with changing any details needed a confirmation popup to ensure the user is making the right decision. As a result of this valuable feedback, the developer added a confirmation

popup, similar to the way users are asked to confirm whether they wish to delete their group or not.

Overall, the feedback that was received from performing this usability study was extremely useful for making changes to the website to be even more user friendly. This usability study also gave the developer a clear idea as to how they should expect potential users to interact with the website.

8 End-Project Report

8.1 End-Project Summary

Overall, the project the developer set out to undertake was a success. The main goal they had from the very beginning of the project was to build some form of application that would allow users to create groups and assign chores to members of their group to complete. This goal was achieved. All features that were intended to be added to the system were implemented successfully with only small alterations to what was initially set out during the planning phase.

8.2 Project Changes

During the development of the project, there were only a few changes that were made to the idea that was in mind. Probably the biggest change was the fact that the developer originally planned to build a mobile application instead of a website. However, after starting to attempt to build the application to work on mobile, they decided to pursue an alternative method and instead chose to build a website instead. They made this decision due to two reasons. One, their experience in building mobile applications was very limited and next to none. This meant they would have to learn how to code mobile applications whilst trying to build the application. This was not ideal and so they decided to build a website instead because they had much more experience doing this and working with the technologies involved. The second reason the developer decided to build a website instead of a mobile application was because a mobile application would mean it's restricted to mobile users whereas a website is accessible by any device with a web browser. By building a responsive website, users with any device could access the application from anywhere.

The initial idea for the application centered around families assigning chores to members of their family. During development, the developer decided to change this to make the website more accessible by a variety of users, such as families and student house groups. This is why the project name was changed from 'Mobile Family Planner' to 'Back to Work', to target all users who wished to use the application.

Some additional features that the developer wanted to implement were never implemented due to time constraints. One of these features was the ability for users to receive notifications to remind them that they have chores to complete. This would have been a useful feature to have but was not possible to implement within the time they had.

9 Project Post-Mortem

9.1 Project Management Approach Evaluation

The agile methodology was chosen as the approach to this project. This allowed the developer to organise the development of the project more efficiently, allowing for a smooth development process between implementing features. The use of Microsoft Office 365 Planner meant that preparing sprints in detail was a simple task. However, due to the developer's lack of experience using this service, sprints were sometimes hard to follow due to the way buckets and tasks are presented on the page. An alternative service for planning sprints and organising the product backlog would have been the GitHub Project Board due to the developer's experience using the service and the easy-to-follow structure of sprints being organised as 'To Do', 'In Progress' and 'Done', along with its simple drag and drop system.

Overall, the project was managed well. Throughout the project development, time was managed efficiently and only a few deadlines that were personally set were missed due to unavoidable circumstances. Designing appropriate planning documents, such as storyboards and user stories, made developing the application much easier and clearer to follow along. An effective version control system using GitHub also made implementing features hassle free and provided a smooth development process.

9.2 Technologies Evaluation

The decision to use PHP, HTML, MySQL and JavaScript for the backend, frontend and database languages, was the right choice. This was due to the developer's extensive experience and understanding of working with these technologies in the past. This avoided any issues with having to learn new languages or using technologies, such as NodeJS and MongoDB, where the developer's experience is very limited, providing a clear path as to how to develop the website.

The decision to switch from working on a mobile application to a website was the right choice. This avoided wasting time learning how to develop mobile applications and instead allowed the developer to focus on developing a website where they have a much more extensive understanding of using the technologies used to develop websites.

9.3 Developer Performance Evaluation

Throughout the development of the project, most sprints were completed before the planned deadlines of 3-5 weeks. Only one sprint went past the deadline due to changes in the developer's accommodation because of the COVID 19 Pandemic.

However, this did not affect the development process as the developer ensured there was enough leeway between working on sprints in case a sprint had to be pushed back. An appropriate 15 hours of development each week was also achieved for the majority of the time. There were some cases where the developer was unable to work during some weeks due to other commitments or unavoidable circumstances, however, this did not affect the development of the project greatly.

Overall, the developer's performance was acceptable considering the circumstances. There were very few mishaps during the development of the project and the developer was always focussed on producing the highest of quality work. The developer never lost interest in the project due to the personal connection to the project and the rewarding experience that would be gained during the process.

9.4 Future Work

The final solution developed for this project is working as intended and is fully functioning. The functional requirements set out at the beginning of the project were all implemented successfully. This however does not mean there is room for improvement and further expansion. For example, with more time, the developer would have liked to implement a notification system within the website but due to time constraints, this was not possible. This feature would be a potential feature that could be implemented sometime in the future if the project were to continue.

To further develop this project in the future, a mobile application could be made to be more mobile friendly. This app would work in parallel with the website but would be its own dedicated application. Although this is not necessary to do due to the website being responsive, a dedicated mobile application would allow mobile users to use the device offline if connection to the internet is lost. It would also make using push notifications easier and more accessible.

A feature the developer would like to implement for Back to Work in the future would be the ability for admins to have more control over accounts under the age of 10 within their group. Giving the admins the option to change the email and passwords of others in their group who are too young to be trusted or lack experience would mean younger users won't have to struggle when changing their email and password and if an email or password is forgotten, it can be easily changed by the admin.

10 Conclusion

The main objective of this application was to build an interactive website that would allow users to create household groups where they could assign chores to members of the household to complete. This main objective has been achieved as well as all

other objectives set out at the beginning of the project, such as the rewards system and calendar.

Overall, I am pleased with the outcome of the project. It has allowed me to further develop my understanding of Full Stack Development, providing me with invaluable experience that will be useful to me when working on future projects.

References

- (i) Atlassian, 2021. *What is Agile?*. [Online]
Available at: <https://www.atlassian.com/agile>
- (ii) Atlassian, 2021. *User Stories with Examples and Template*. [Online]
Available at: <https://www.atlassian.com/agile/project-management/user-stories>
- 123 Reg, 2021. *What Is MySQL And Why Do I Need It?*. [Online]
Available at: <https://www.123-reg.co.uk/support/servers/what-is-mysql-and-why-do-i-need-it/>
- Curry, L., 2019. *What is Office 365 Planner? A Beginner's Guide*. [Online]
Available at: <https://www.chorus.co/resources/news/what-is-office-365-planner-a-beginners-guide>
- Data Protection Act, 2018. Data Protection Act. *Data Protection Act 2018*.
- DevIQ, 2016. *Don't Repeat Yourself*. [Online]
Available at: <https://deviq.com/principles/dont-repeat-yourself>
- General Data Protection Regulation, 2018. GDPR. *Official Journal of the European Union*.
- Hostinger Tutorials, 2021. *What is WAMP?*. [Online]
Available at: <https://www.hostinger.co.uk/tutorials/what-is-wamp>
- MDN Web Docs, 2021. *What is JavaScript*. [Online]
Available at: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- Moran, K., 2019. *Usability Testing 101*. [Online]
Available at: <https://www.nngroup.com/articles/usability-testing-101/>
[Accessed 29 April 2021].
- PHP, 2021. *PHP*. [Online]
Available at: <https://www.php.net/manual/en/intro-whatis.php>
- wpbeginner, 2021. *What is: Apache*. [Online]
Available at: <https://www.wpbeginner.com/glossary/apache/>

Appendix

Appendix 1: User Guide

The following is a general user guide for users to follow along. Back to Work is hosted on a remote AWS server. To access Back to Work, follow this link:

<https://ec2-52-56-217-123.eu-west-2.compute.amazonaws.com>

1.1 Registration & Login

Users are required to create an account before accessing Back to Work. To create an account, select the “Register for an Admin Account Here!” button on the “Login” page (Figure 24 User Login). You will be redirected to the “Registration” page (Figure 23 User Registration) where you can enter your details. Once completed, click register. This will return you to the “Login” page where you can now enter your login details.

The screenshot shows the 'Registration' page of the 'Back To Work' application. At the top, there is a header bar with the text 'Back To Work'. Below this is a form titled 'Registration' containing several input fields: 'First Name' (Joe), 'Last Name' (Bloggs), 'Address' (Joe's House), and 'Email' (JoeBloggs@gmail.com). There are also two password fields, both marked with '.....'. At the bottom of the form is a blue 'Register' button and a 'Back To Login' link. At the very bottom of the page, there is a 'Cookie Consent' banner with the text 'By continuing to use this site, you agree to the use of Cookies.' and a 'I UNDERSTAND' button.

Figure 23 User Registration

Back To Work

The screenshot shows a login form titled "Welcome" with a sub-instruction "Please Login/Register to Continue". It includes fields for "Email" (containing "JoeBloggs@gmail.com") and "Password" (containing "....."). A "Remember Me" checkbox is checked. Below the form is a blue "Login" button, a link "Need an Account?", and a link "Register For an Admin Account Here!". At the bottom left is a "Cookie Consent" message with "I UNDERSTAND" at the bottom right.

Cookie Consent
By continuing to use this site, you agree to the use of Cookies.

I UNDERSTAND

Figure 24 User Login

1.2 Adding New Members to Your Group

Once logged in, all users are redirected to the “My Group” page. From here, you can view all the members of your group. To add more users to your group you must be an admin. Registering from the login page automatically makes you an admin and so to add new users to your group, select the “Register New Group Member” button (Figure 25 View Members). A popup box will appear where you can enter the user’s details (Figure 26 Add New Member). If a user is less than 14 years old, their account is restricted to a normal user account. If the user is 14 or older, you have the option to give them admin powers by select the account type.

Back To Work

Joe's House

Name	Chores Completed	Points
Joe	0	0
Bill	0	0

[Register New Group Member](#)

Figure 25 View Members

The screenshot shows a modal window titled "Add Family Member" overlaid on a dark background. The modal contains fields for First Name (Bill), Last Name (Gates), Date of Birth (12/05/2020), Email (BillGates@gmail.com), Password (hidden), Confirm Password (hidden), and a dropdown for Child status. At the bottom are "Save" and "Close" buttons.

Figure 26 Add New Member

1.3 Editing a member's Points/Chores Complete

Once a new user has been added, they will appear in the member's list. From here, if you are an admin, you can select a member and edit their points and chores completed (Figure 27 Edit User Points/Chores Completed).

The screenshot shows a modal dialog box titled "Family Member" with the identifier "Joe's House". The dialog contains fields for "Name" (Bill), "Chores Completed" (3), and "Points" (12). It includes a "Delete Account" link and "Save" and "Close" buttons. The background shows a table with rows for Joe and Bill.

Name	Chores Completed	Points
Joe	0	0
Bill	3	12

Register New Group Member

Figure 27 Edit User Points/Chores Completed

If you are and admin, you also have the option to delete the selected user's account. Doing this will remove all data associated with their account (Figure 28 Delete User Account).

The screenshot shows a modal dialog box asking "Are You Sure You Wish To Delete Bill's Account?" with "Confirm" and "Cancel" buttons. The background shows a table with rows for Joe and Bill.

Name	Chores Completed	Points
Joe	0	0
Bill	0	0

Register New Group Member

Figure 28 Delete User Account

1.4 Adding New Chores

Navigating to the “Chores” page will allow you to view your assigned chores. If you are an admin, you will see the available chores that can be assigned on the left, the chores that have already been assigned to members of your group and your own personal assigned chores (Figure 29 Empty Chores Page). If you are a normal user, you will only be able to see your own assigned chores.

If you are an admin, clicking on the “New Chore” button will open a popup that will allow you to enter the details of a new chore (Figure 30 Adding New Chore).

The screenshot shows a web application interface titled "Back To Work". At the top, there is a navigation bar with links: "My Group", "Chores", "Calendar", "Rewards", "My Account", and "Logout". Below the navigation bar, the main content area is titled "Chores". It contains three sections: "Available Chores" (with the message "There Are No Chores Available!"), "Assigned Chores" (with the message "No Chores Have Been Assigned!"), and "My Chores" (with the message "You Have No Chores To Complete!"). A blue button labeled "New Chore" is located at the bottom of the "Chores" section.

Figure 29 Empty Chores Page

The screenshot shows a modal dialog box titled "New Chore" over a dark background. The dialog box has fields for "Name" (containing "Dishes"), "Description" (containing "Wash the dishes."), "Points Value" (containing "5"), and "Penalty Value" (containing "3"). At the bottom of the dialog box are two buttons: "Add Chore" (blue) and "Close" (red). In the background, the "Chores" section from Figure 29 is visible, showing the "Available Chores" section with the message "There Are No Chores Available!".

Figure 30 Adding New Chore

1.5 Assigning Chores to Users

Once chores have been added, if you are an admin, selecting a chore will allow you to assign it to a user within your group (Figure 31 Assigning Chores).

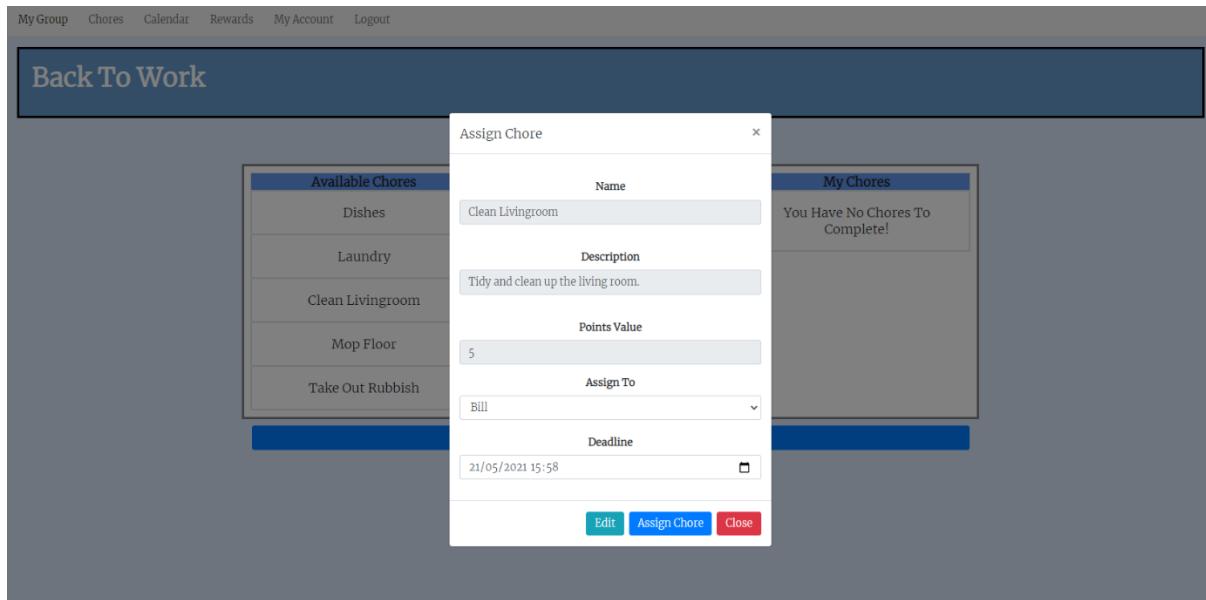


Figure 31 Assigning Chores

Available Chores	Assigned Chores	My Chores
Laundry	Clean Livingroom Assigned: Bill	Take Out Rubbish
Mop Floor	Dishes Assigned: Bill	

Figure 32 Assigned User Chores

Selecting an assigned chore will allow all users to view the details of that assigned chore. If you are an admin, you will have the option to reassign that assigned chore to another user (Figure 33 Reassign Chore). From here, users can also complete chores. Depending on the deadline of the assigned chore, the user will either gain or lose points if completed before or after the deadline.

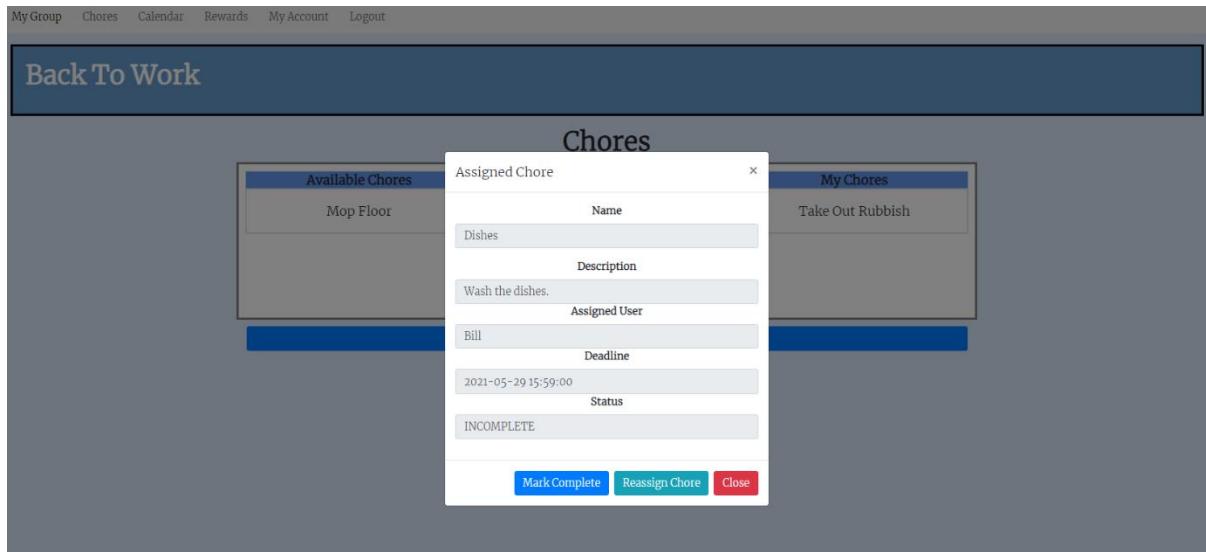


Figure 33 Reassign Chore

1.6 Editing Chores

If you are an admin, selecting a chore will also allow you to edit the details of that chore (Figure 34 Edit Chore).

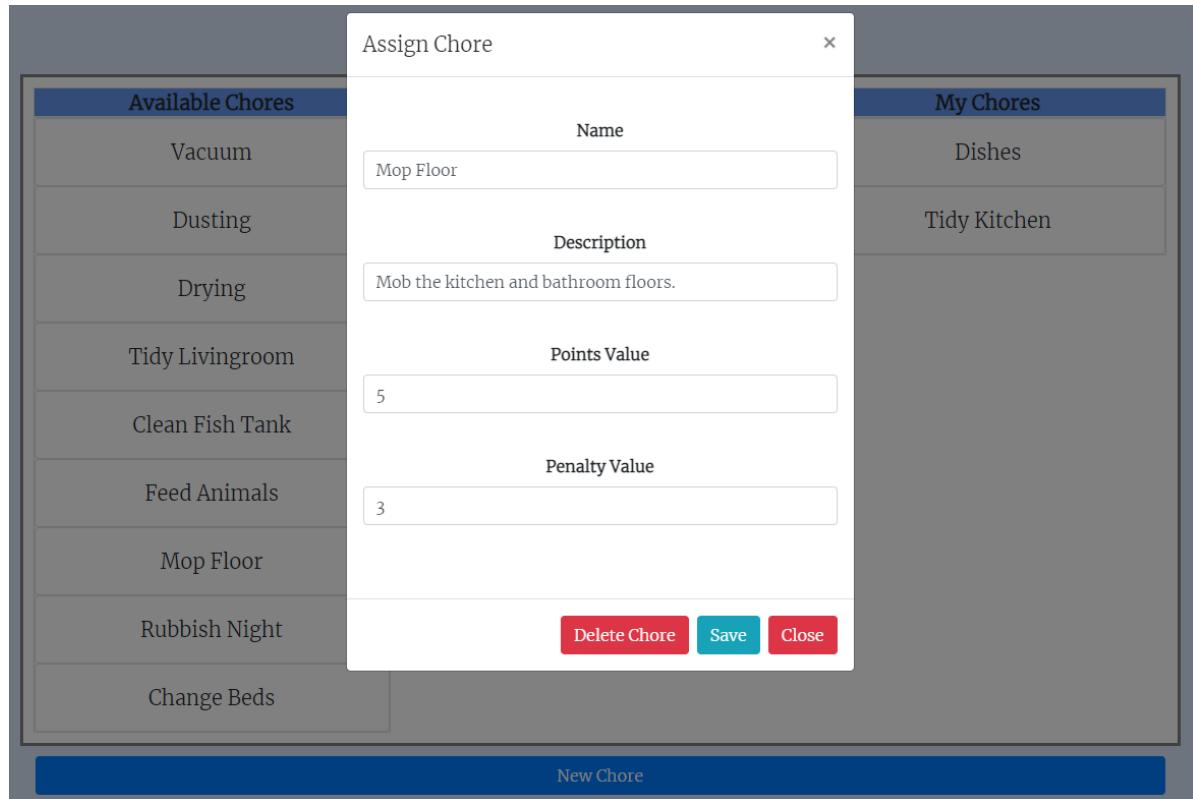


Figure 34 Edit Chore

From here, you can also delete a chore (Figure 35 Delete Chore).

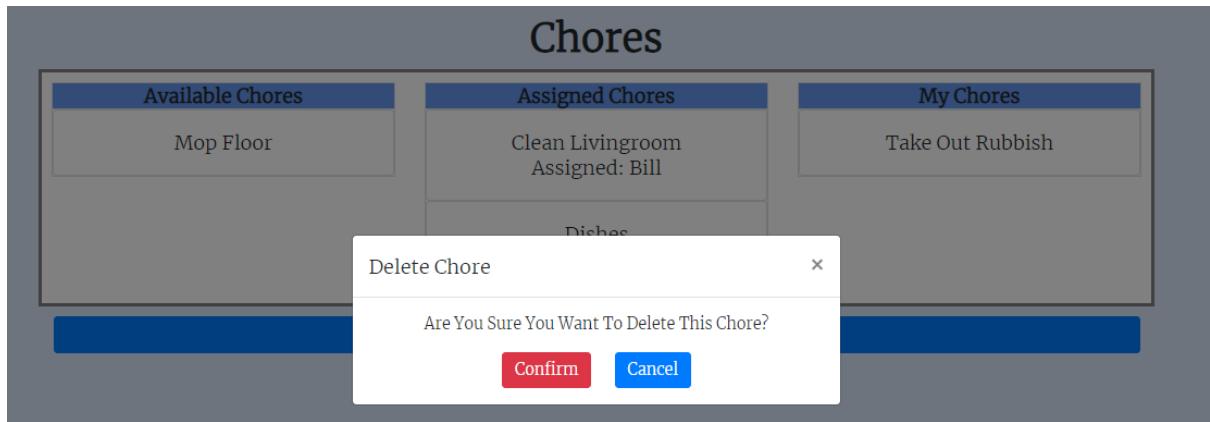


Figure 35 Delete Chore

1.7 View Calendar Events

Navigating to the calendar page will display a calendar. Some dates will be blue because there is an event for that date (Figure 36 Calendar). Selecting one of these dates will display the details of those events (Figure 37 Calendar Event).

The screenshot shows a 'Group Calendar' for May 2021. The days of the week are labeled from Mon to Sun. The dates are arranged in a grid. Several dates are highlighted in light blue: 19, 21, 25, 26, 28, 29, and 30. These likely represent days with scheduled events. The header 'Group Calendar' is centered above the calendar grid. Navigation buttons 'Prev' and 'Next' are at the top right of the grid.

2021 May						
Mon	Tues	Wed	Thur	Fri	Sat	Sun
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Figure 36 Calendar

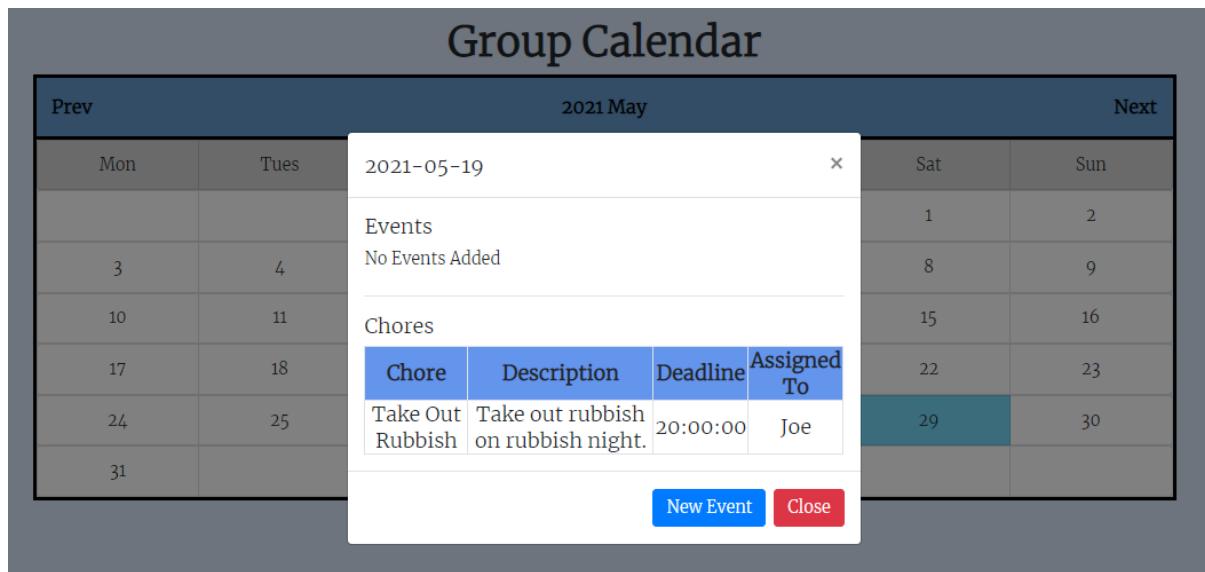


Figure 37 Calendar Event

1.8 New Calendar Event

If you are an admin, selecting a date will give you the option to add new custom event. A popup will appear allowing you to enter the event's details (Figure 38 New Calendar Event). These events will be displayed to all users of the group.

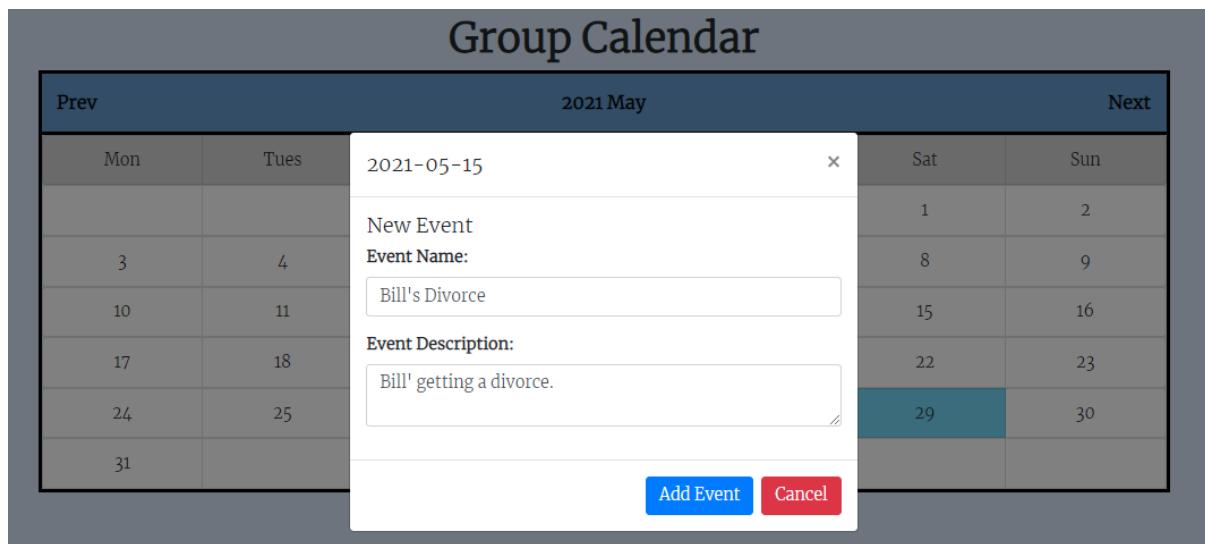


Figure 38 New Calendar Event

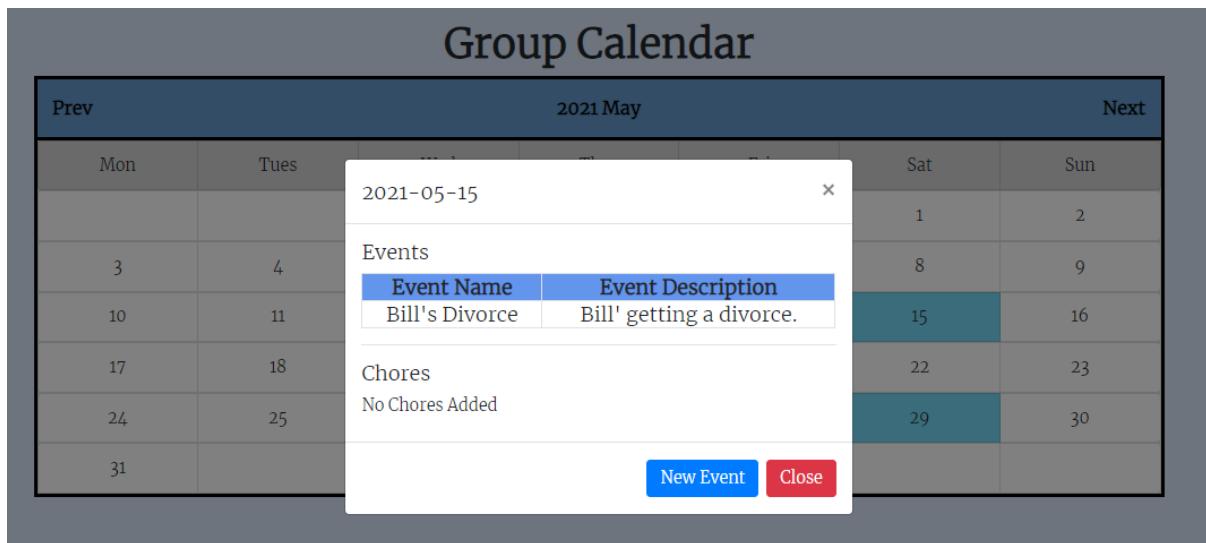


Figure 39 Viewing New Calendar Event

1.9 Adding New Rewards

If you are an admin, navigating to the rewards page will allow you to add new rewards that users within the group can redeem their points for. Clicking the “New Reward” button will create a popup, allowing you to enter the new reward’s details (Figure 40 New Reward).

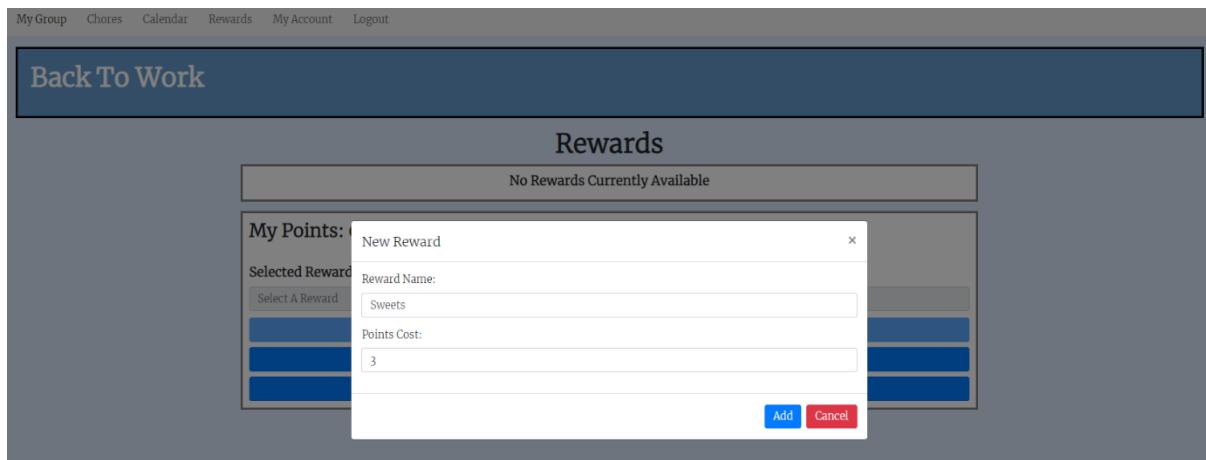


Figure 40 New Reward

1.10 Redeeming Rewards

To redeem a reward, you can select a reward from the list of available rewards (Figure 41 Redeeming A Reward). If you have enough points, a reward request will be created.

The screenshot shows the 'Rewards' section of the 'Back To Work' application. At the top, there is a navigation bar with links: 'My Group', 'Chores', 'Calendar', 'Rewards', 'My Account', and 'Logout'. Below the navigation bar, the title 'Back To Work' is displayed. The main content area is titled 'Rewards' and contains a table showing three reward options:

Reward	Points
Sweets	3
£10	100
Choice of Takeaway	45

Below the table, a message says 'My Points: 6'. A dropdown menu is open, showing 'Selected Reward: Sweets' and three options: 'Redeem', 'Reward Requests', and 'New Reward'.

Figure 41 Redeeming A Reward

If you are an admin, you have the ability to either approve or decline user's reward requests (Figure 42 Reward Requests).

The screenshot shows a 'Reward Requests' modal window overlaid on the 'Rewards' section of the application. The modal has a header 'Reward Requests' and a close button 'x'. It displays a table with one row of data:

User Name	User Points	Reward Name	Reward Cost	Action
Joe	6	Sweets	3	APPROVE DECLINE

At the bottom of the modal, there is a red 'Cancel' button. The background of the application shows the same rewards table and points information as Figure 41.

Figure 42 Reward Requests

1.11 View Account Details

Navigating to the My Account page allows users to view their account details (Figure 43 View Account Details).

The screenshot shows the 'My Account' section of a web application. At the top, there's a header bar with links: 'My Group', 'Chores', 'Calendar', 'Rewards', 'My Account' (which is highlighted in blue), and 'Logout'. Below the header, a title 'Back To Work' is displayed in a dark blue box. The main content area is titled 'My Account'. It contains two main sections: 'My Details' and 'Options'. The 'My Details' section shows the user's name as 'Joe Bloggs', DOB as '2000-02-27', and points as '3'. It also lists the email as 'JoeBloggs@gmail.com' and the group name as 'Joe's House'. The 'Options' section includes buttons for 'Change Email' (highlighted in blue), 'Change Password', 'View Past Reward Claims', and 'Disband Group'.

Figure 43 View Account Details

1.12 Change Email

To change your email, click the “Change Email” Button to display a popup that allows you to change your email (Figure 44 Change Email).

The screenshot shows a 'Change Email' modal dialog box overlaid on the 'My Account' page. The modal has a light gray background and contains three input fields: 'Current Email' (containing 'JoeBloggs@gmail.com'), 'New Email' (containing 'JoeBloggs@outlook.com'), and 'Confirm Password' (containing '*****'). At the bottom of the modal are two buttons: 'Save' (in blue) and 'Cancel' (in red). The background of the main page is dimmed to indicate the modal is active.

Figure 44 Change Email

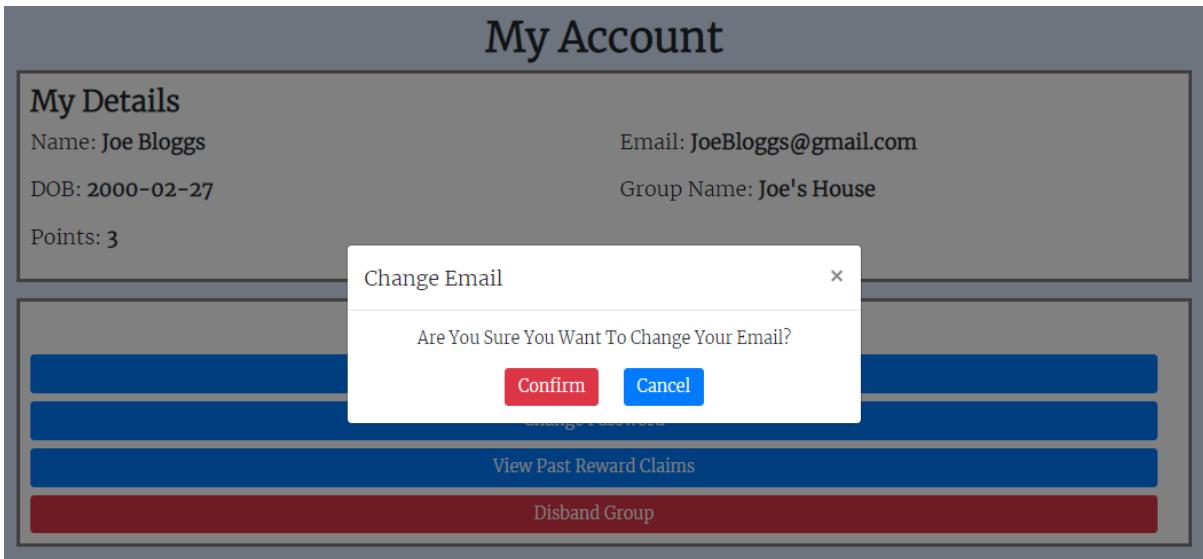


Figure 45 Confirm Email Change

1.13 Change Password

To change your password, select the “Change Password” button. A popup will appear which will allow you to confirm your current password and enter a new password (Figure 46 Change User Password).

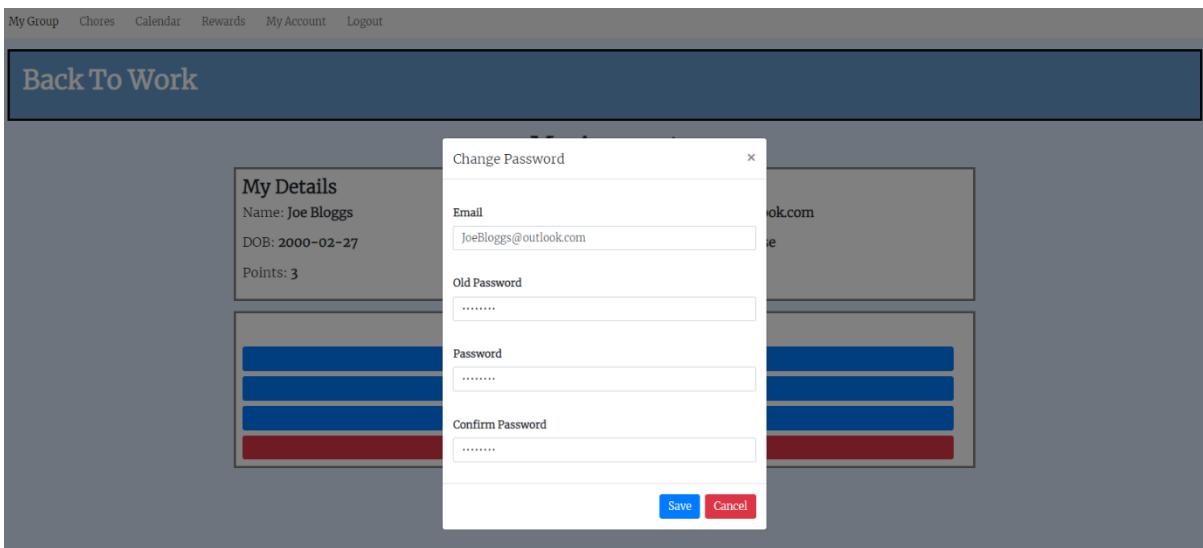


Figure 46 Change User Password

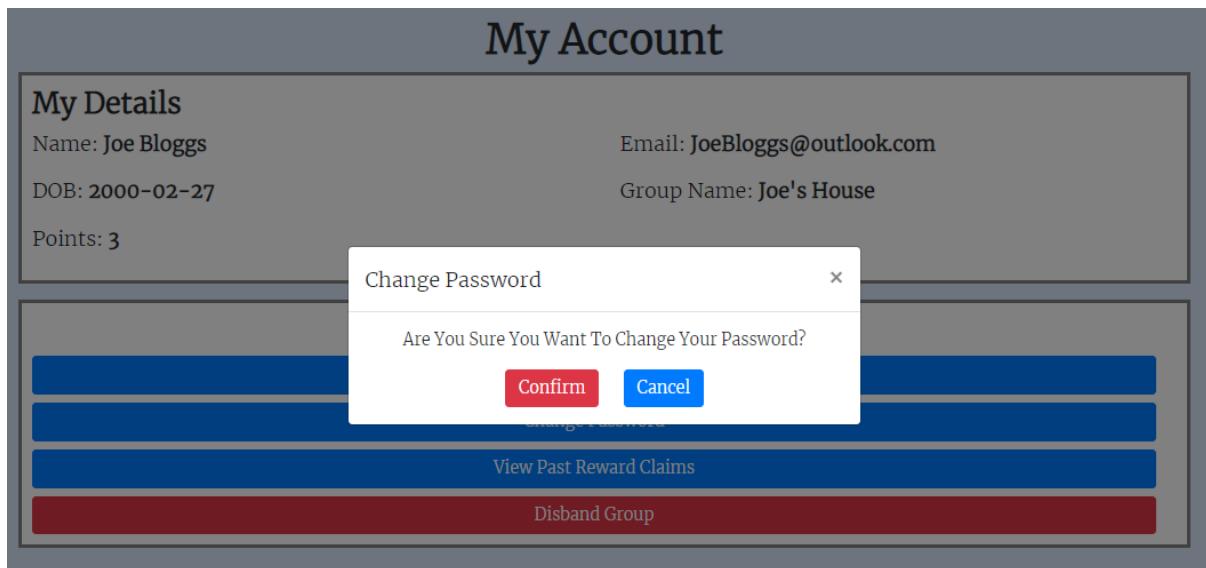


Figure 47 Confirm Password Change

1.14 View Past Rewards

To view your past reward claims, select the “View Past Reward Claims” button. A popup will appear and display your past reward claims.

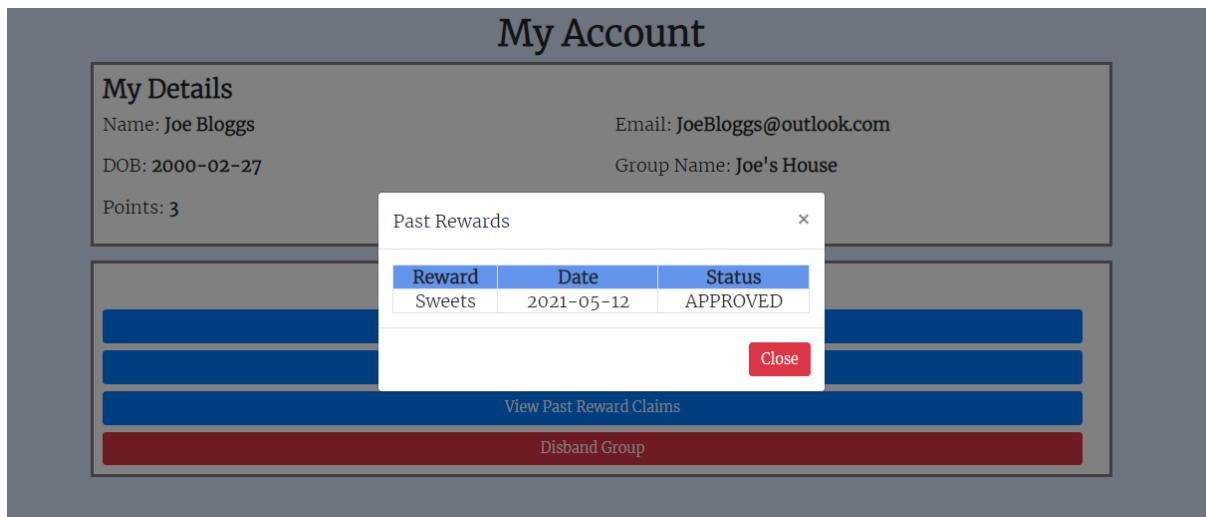


Figure 48 Past Reward Claims

1.15 Disband Group

As an admin, you can disband your current group by clicking the “Disband Group” button and confirming (Figure 49 Disband Group). Doing this will delete all accounts linked to your group and all associated data.

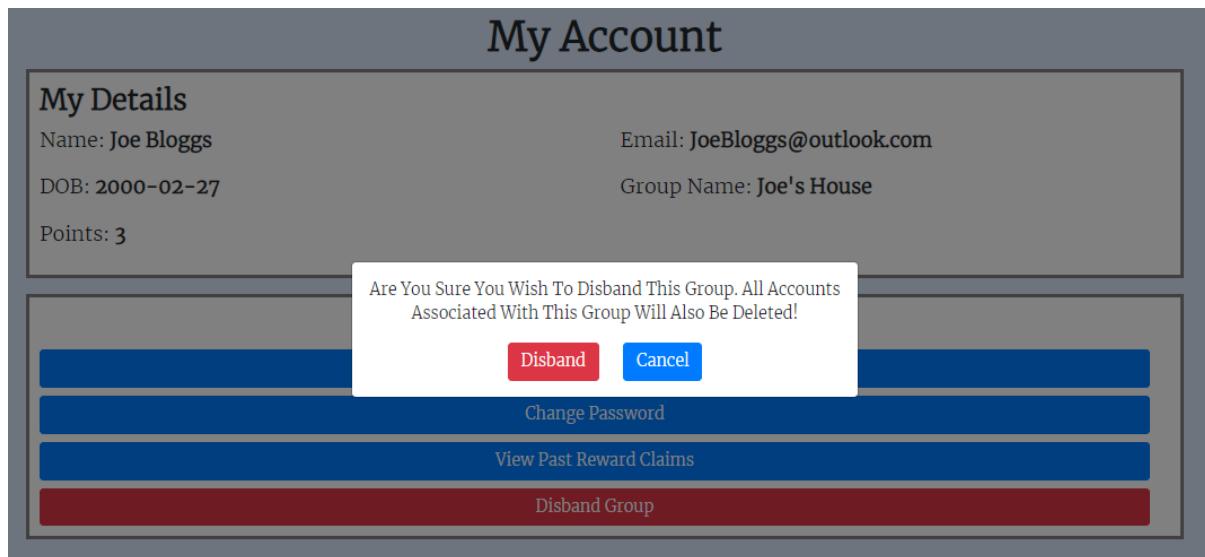


Figure 49 Disband Group

Appendix 2: Project Management

2.1 Product Vision

Back to Work is an interactive website that allows users to assign chores for completion to a member of a group. This system is aimed at families who struggle to get their children to their chores or students who are sharing a house and wish to have an organised chore roster. Back to Work uses a reward system where points can be earned for completing a chore and then redeemed for a reward of the user's choice. This acts as an incentive for kids or housemates to complete their chores. Users also have access to a calendar for being more organised by planning their week ahead and keeping track of when chores need to be completed.

2.2 Functional Requirements

Below are the functional requirements that were outlined at the start of the project during the planning phase.

Main Features

User Features

- Registration Page
 - User enters: First & Last Name, Email, Password, DOB.
 - Validate details using JavaScript & send to server using jQuery.
 - Server sends details to database to be stored.
- Login Page
 - User enters Email and Password.
 - Validate inputs using JavaScript and call an API to check if the user exists in database.
 - Check inputted password with saved password and login if they match.
- My Group Page
 - Users can view the other members in their group.
 - User can view the points and number of chores completed of themselves and others in their group.
- Chores Page
 - Users can view their assigned chores.
 - Users can view the details of their assigned chores.
 - Users can mark their assigned chores as complete and earn points if completed before the deadline and lose points if completed after the deadline.
- Rewards Page
 - Users can view the rewards available to redeem.
 - Users can select a reward to redeem if they have enough points.
- Calendar Page
 - Users can select a date to view the events of that date.
 - Users can view the details of an event or chore.
- My Account Page
 - User can view their account details.
 - User can change their email.
 - User can change their password.
 - User can view their past reward claims.

Admin Features

- My Group Page
 - Admins can edit the points and chores completed of a user in their group.
 - Admins can remove a user from their group.
 - Users can add a new user to their group.
- Chores Page
 - Admins can add new chores for their group.
 - Admins can edit the details of chores.
 - Admins can assign chores to members of their group.
 - Admins can reassign chores to another member of their group.
 - Admins can remove a chore for their group.
- Rewards Page
 - Admins can add new rewards for users to redeem.
 - Admins can view user reward requests.
 - Admins can approve or decline user reward requests.
 - Admins can remove rewards for their group.
- Calendar Page
 - Admins can add new events to a selected date.
- My Account Page
 - Admins can disband their group, removing all associated data.

Figure 50 Functional Requirements

2.3 Non-Functional Requirements

Additional Features

- My Group Page
 - Users can send messages to each other.
 - Admins can send group messages/announcements.
- Chores Page
 - Users can receive push notifications to remind them of chore deadlines.
 - Users can view their past chore assignments.
- Rewards
 - Users can propose a reward to be added.
 - Users can cancel their reward requests.
- Calendar
 - Users can add events to a personal calendar.
- My Account
 - Users can edit their personal data (First & Last Name, DOB)

Figure 51 Non-Functional Requirements

2.4 User Stories

Back to Work – User Stories

Admin

- As an admin, I wish to be able to create a new account and chore group.
- As an admin, I wish to be able to add new users to my chore group.
- As an admin, I wish to be able to add new chores for my chore group.
- As an admin, I wish to be able to assign users within my chore group to a chore to complete.
- As an admin, I wish to be able to create a new calendar event for my chore group to view.
- As an admin, I wish to be able to edit the details of chores, users in my group and events that I add.
- As an admin, I wish to be able to edit the details of the users in my chore group, delete/remove accounts from my group and even delete the whole group and all associated data.
- As an admin, I wish to be able to add new rewards for users to redeem.
- As an admin, I wish to be able to approve or decline user's requests to redeem a reward.

User

- As a user, I wish to be able to view the members of my chore group and their details.
- As a user, I wish to be able to view the chores that have been assigned to me and mark them as complete, either adding or subtracting my points if chore is completed on time or late.
- As a user, I wish to be able to view the details of my assigned chores.
- As a user, I wish to be able to view a calendar showing the deadlines of the chores assigned to me and the details of new events added by an admin.
- As a user, I wish to be able to change my account details.
- As a user, I wish to be able to view the amount of points I have and exchange those points for physical rewards.

Figure 52 User Stories

2.5 Risk Assessment

Potential Risk	How to Address Risk
Sprints not being completed on time/unrealistic time management.	Planning sprints ahead of time will allow me to appropriately organise my time to maximise efficiency and complete sprints within a reasonable timeframe.
Developing wrong software functions.	Identifying the most important features being implemented into the application and organising sprints will allow me to prioritise my time in developing different features. Outlining the Minimal Viable Product and focusing on features that meet this requirement will prevent any unnecessary features being developed.
Late changes to requirements.	Identifying the Minimal Viable Product at the start of development will allow me to easily outline what it is I need to do. To avoid any late changes being made, reviewing each sprint at the end of development and making any necessary changes then will prevent any late and unnecessary changes to the requirements being made.
Development too technically difficult.	Ensuring that all requirements are viable, and I am confident that they can be implemented using the available software, as well as my understanding and experience with the software, will prevent any issues where I may become stuck when implementing a feature.

Figure 53 Risk Assessment

2.6 Storyboards

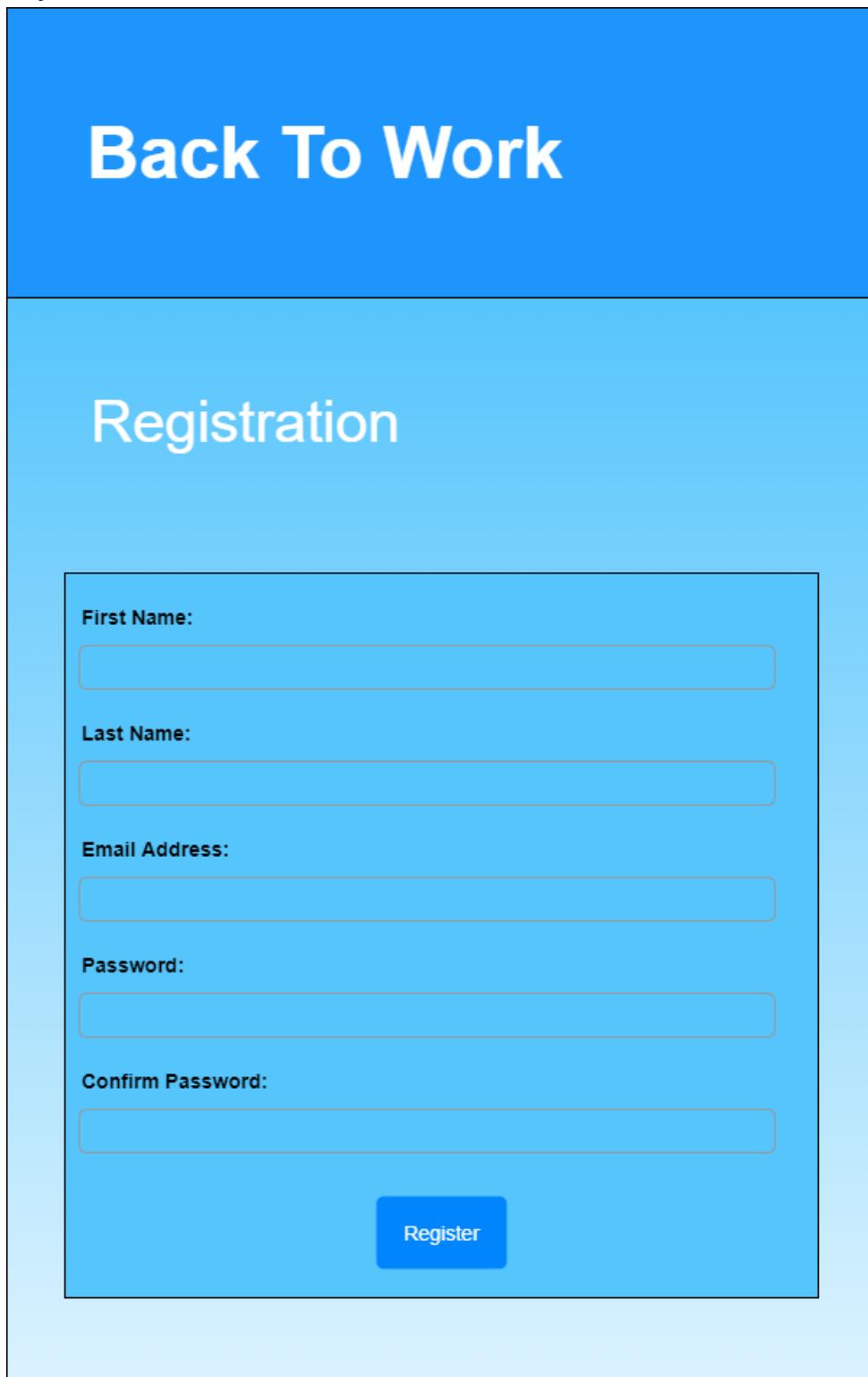


Figure 54 Registration Page Storyboard

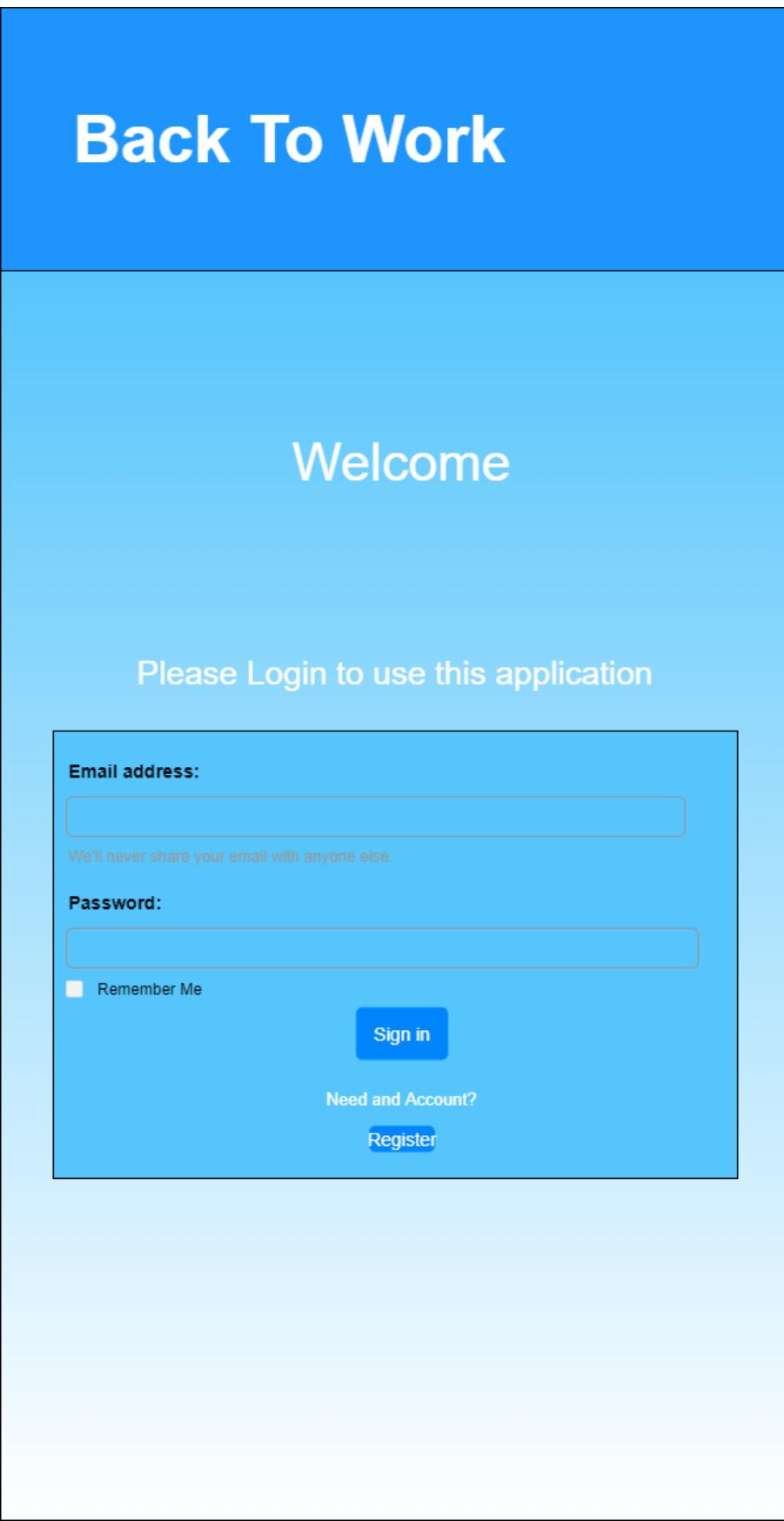


Figure 55 Login Page Storyboard

Back To Work

My Group

Name	Chores Completed	Points

Admin Button
Only

Setup/Manage
Family

MY GROUP CHORES CALENDAR REWARDS MY ACCOUNT

Figure 56 My Group Storyboard

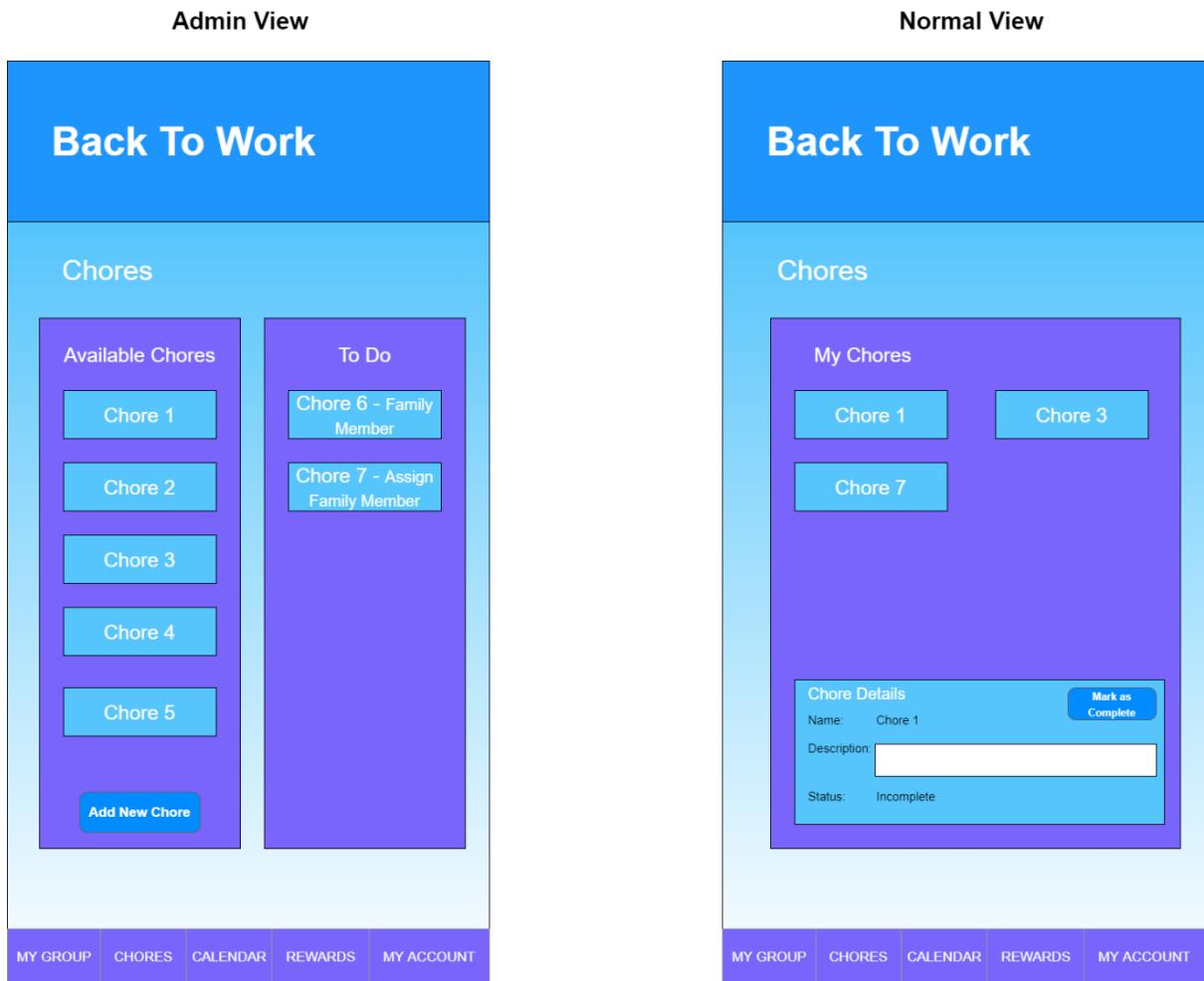


Figure 57 Chores Storyboard

Back To Work

Rewards

Reward Name	Points Cost

Selected Reward:

Redeem

Reward Requests

New Reward

Admin
Only

MY GROUP

CHORES

CALENDAR

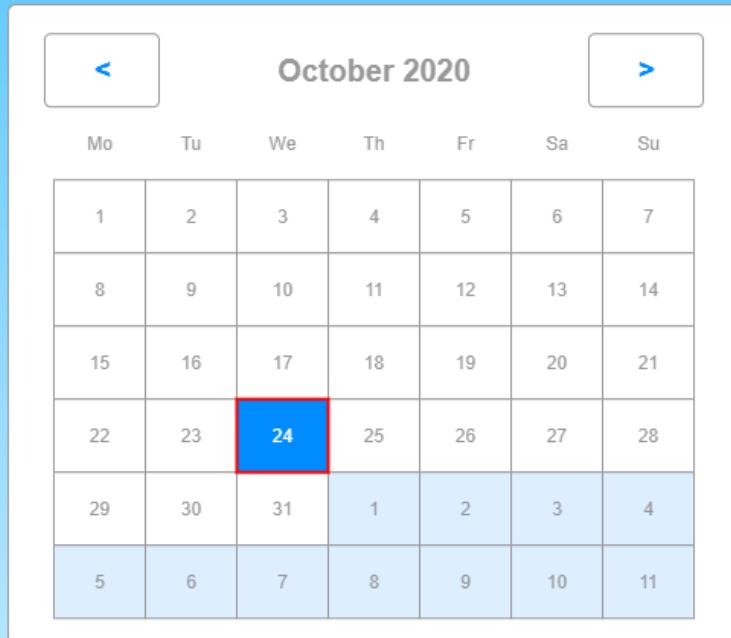
REWARDS

MY ACCOUNT

Figure 58 Rewards Storyboard

Back To Work

Calendar



Date: 24/10/2020

Event: EVENT NAME

Event Details:

New Event

MY GROUP

CHORES

CALENDAR

Rewards

MY ACCOUNT

Figure 59 Calendar Storyboard

Back To Work

My Account

Name: Full Name
Email: Email Address
Family Group: Family Name
Chore Points: Points

Options

Change Email
Change Password
Past Reward Requests
Logout

Delete Account

MY GROUP CHORES CALENDAR REWARDS MY ACCOUNT

Figure 60 My Account Storyboard

2.7 UI-Flow Charts

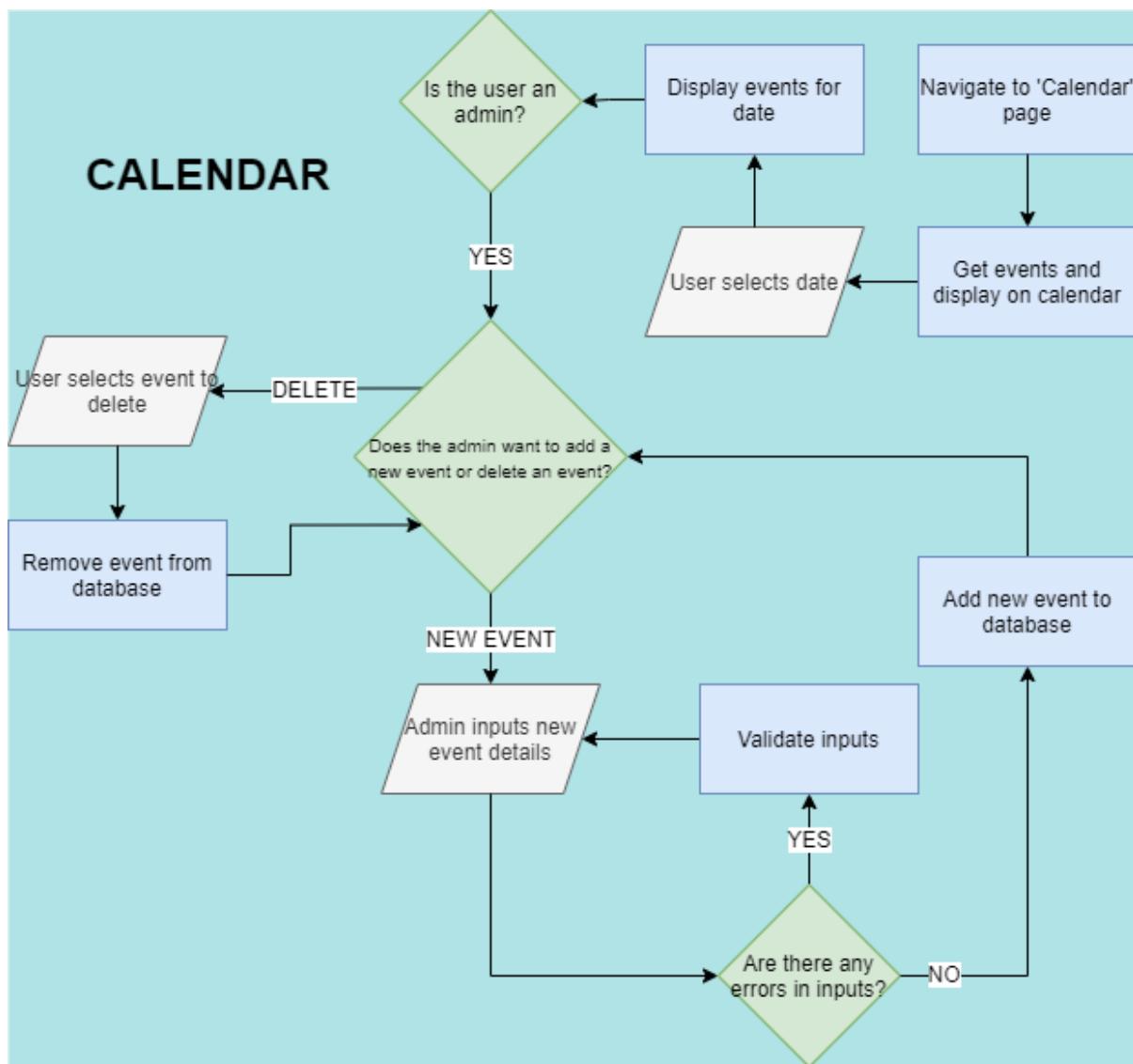


Figure 61 Calendar UI-Flow Chart

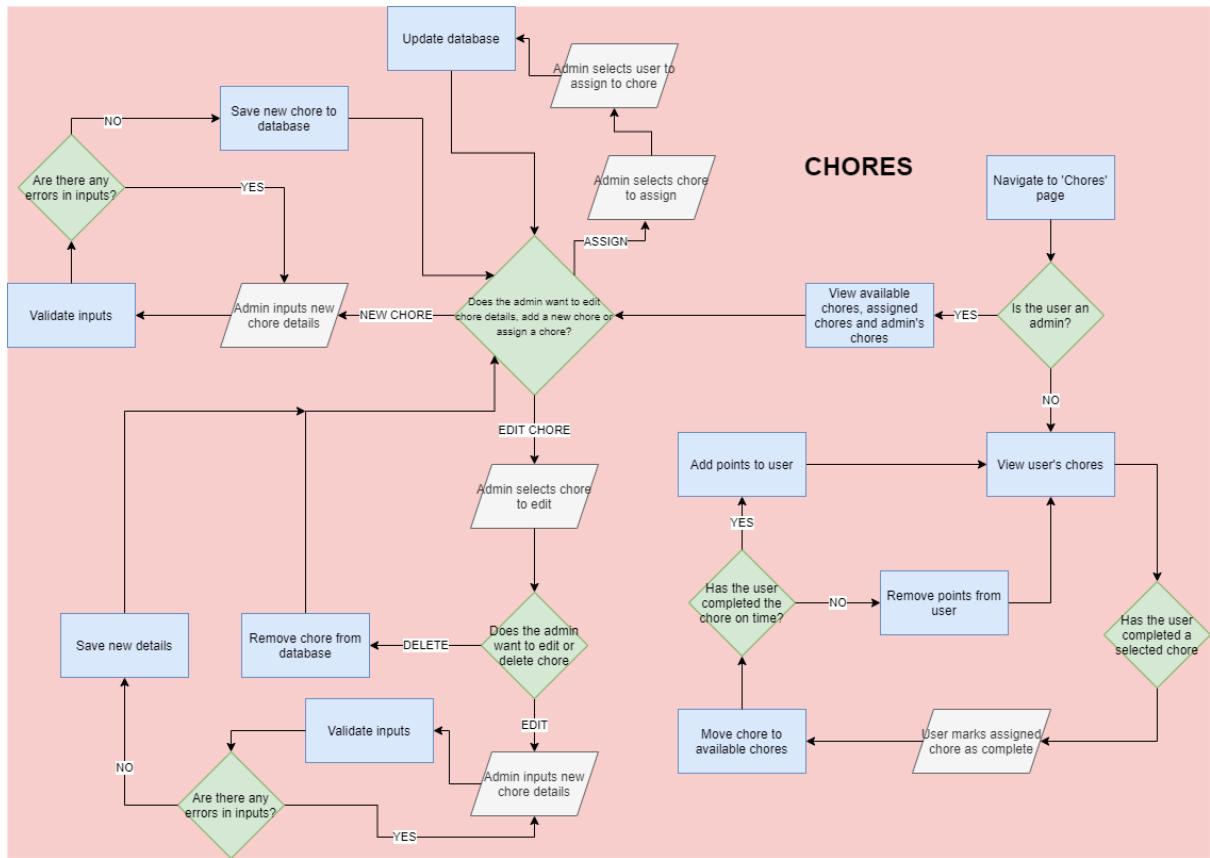


Figure 62 Chores UI-Flow Chart

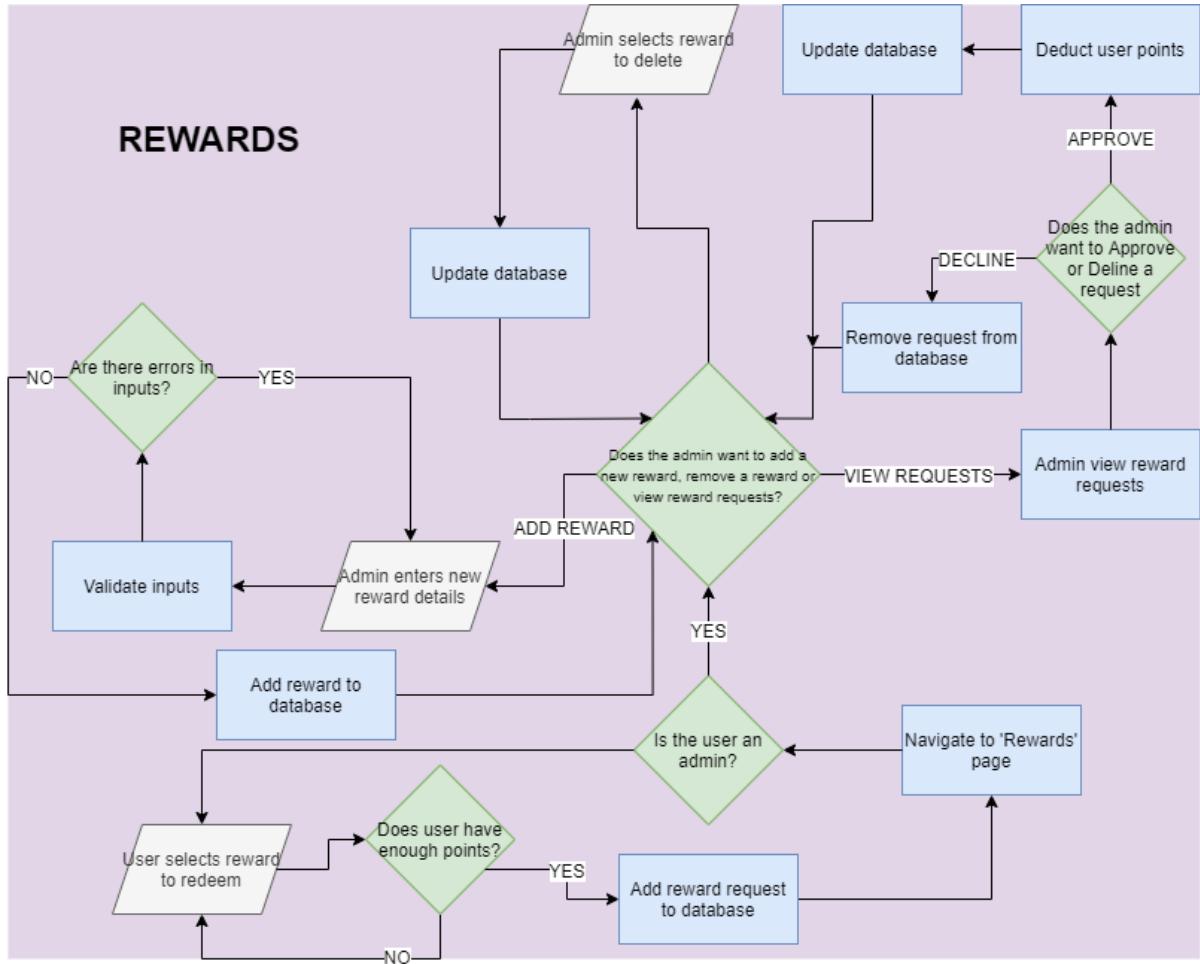


Figure 63 Rewards UI-Flow Chart

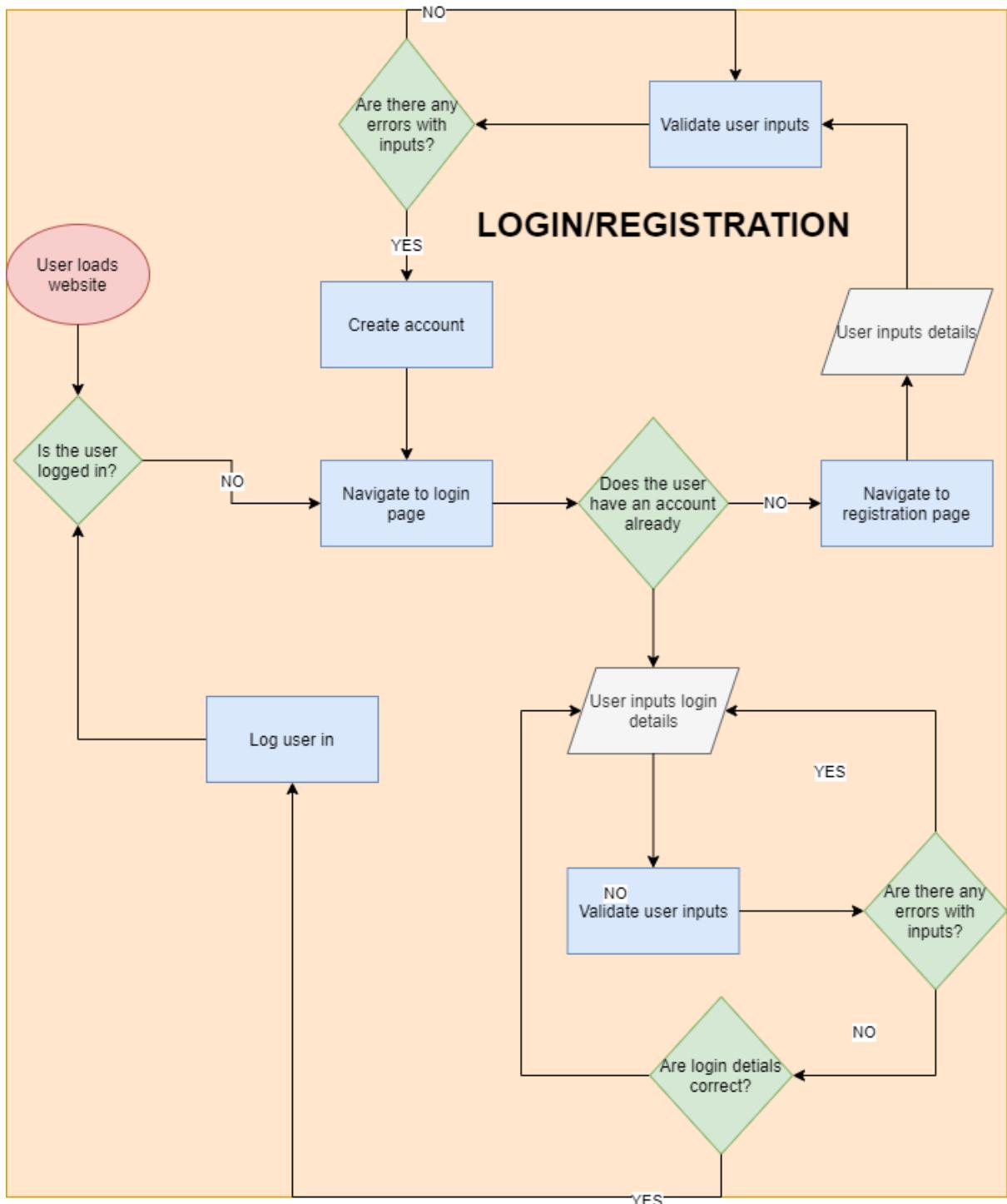


Figure 64 Login & Registration UI-Flow Chart

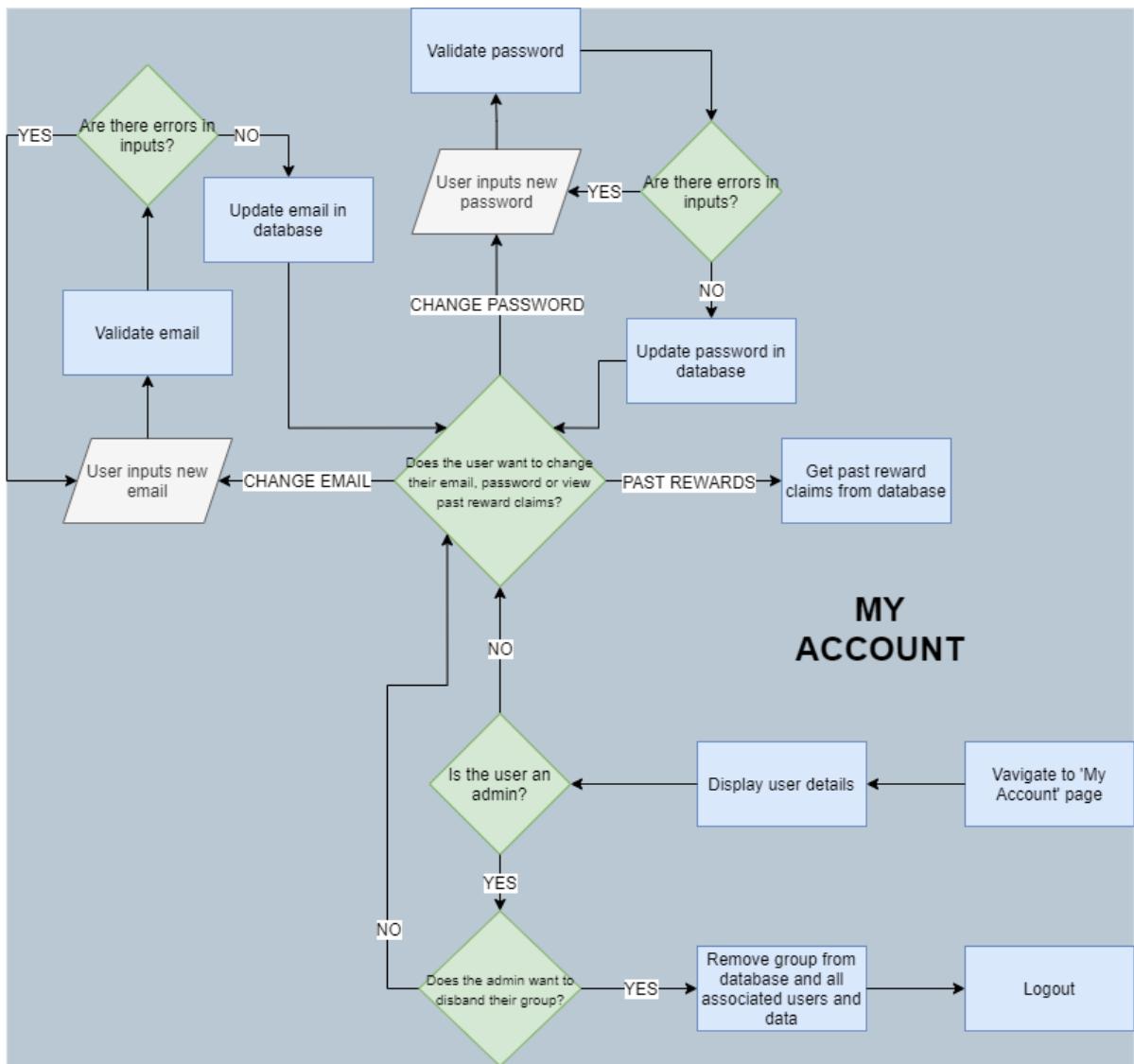


Figure 65 My Account UI-Flow Chart

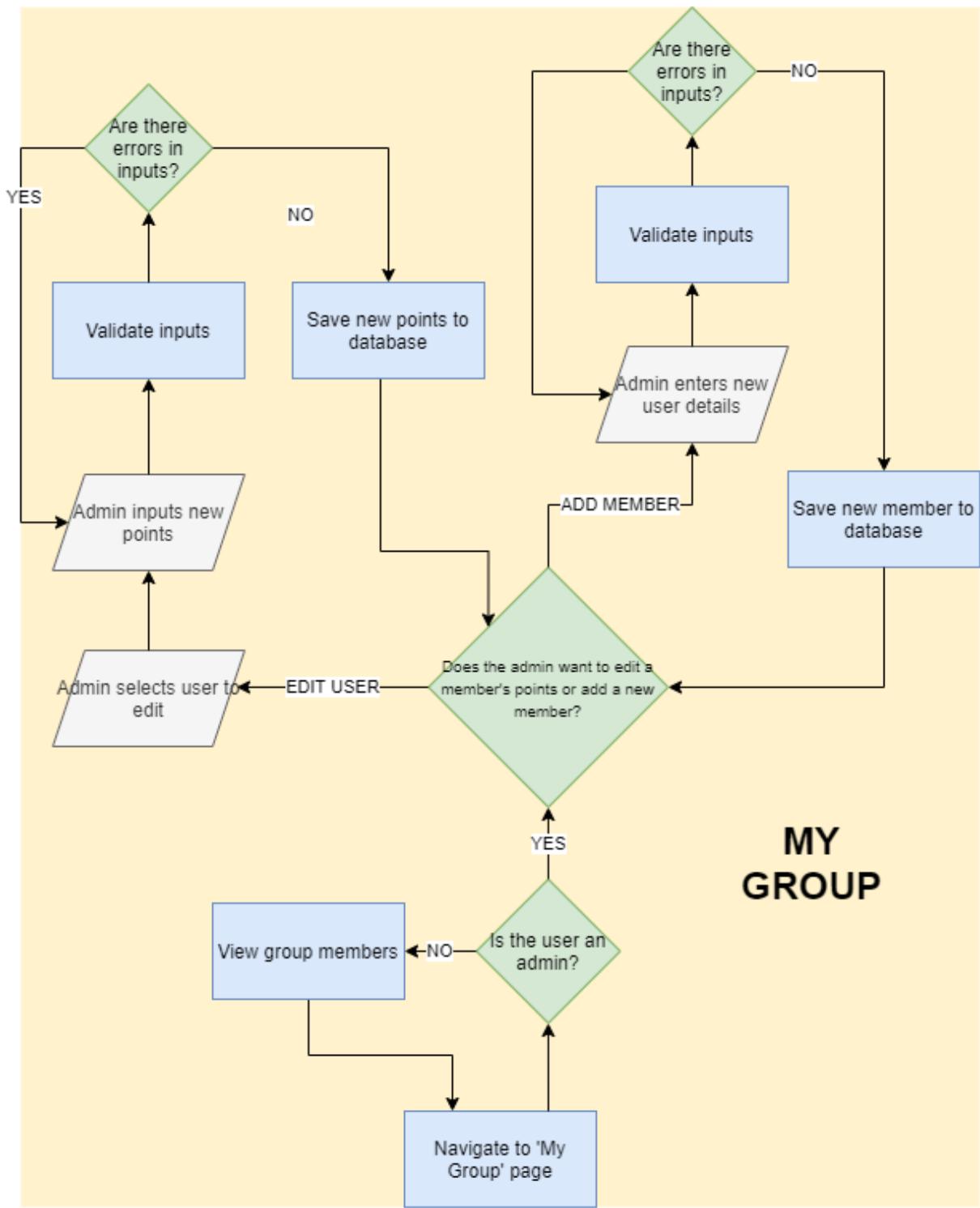


Figure 66 My Group UI-Flow Chart

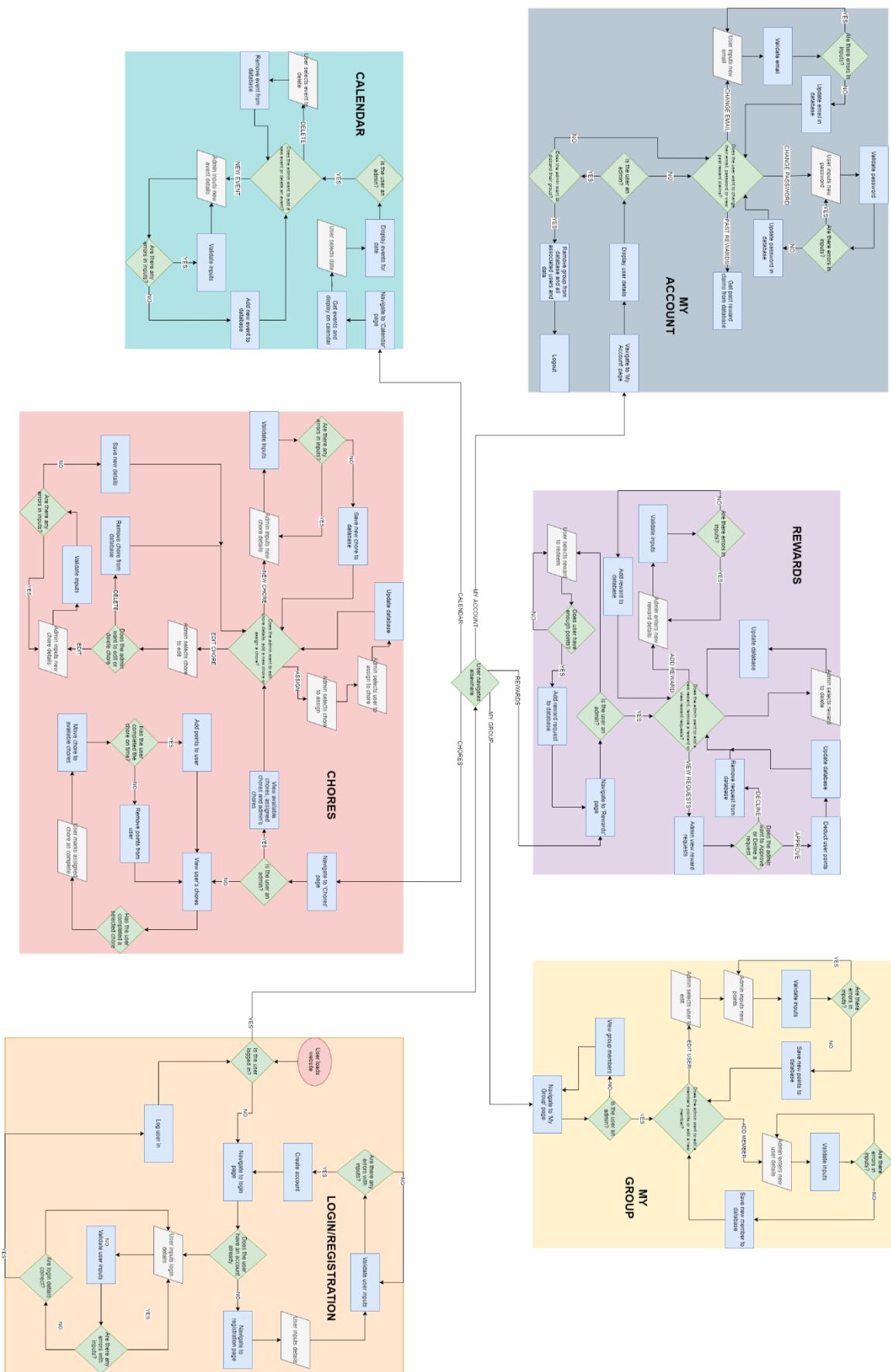


Figure 67 UI-Flow Chart in Full

2.8 Use Case Diagram

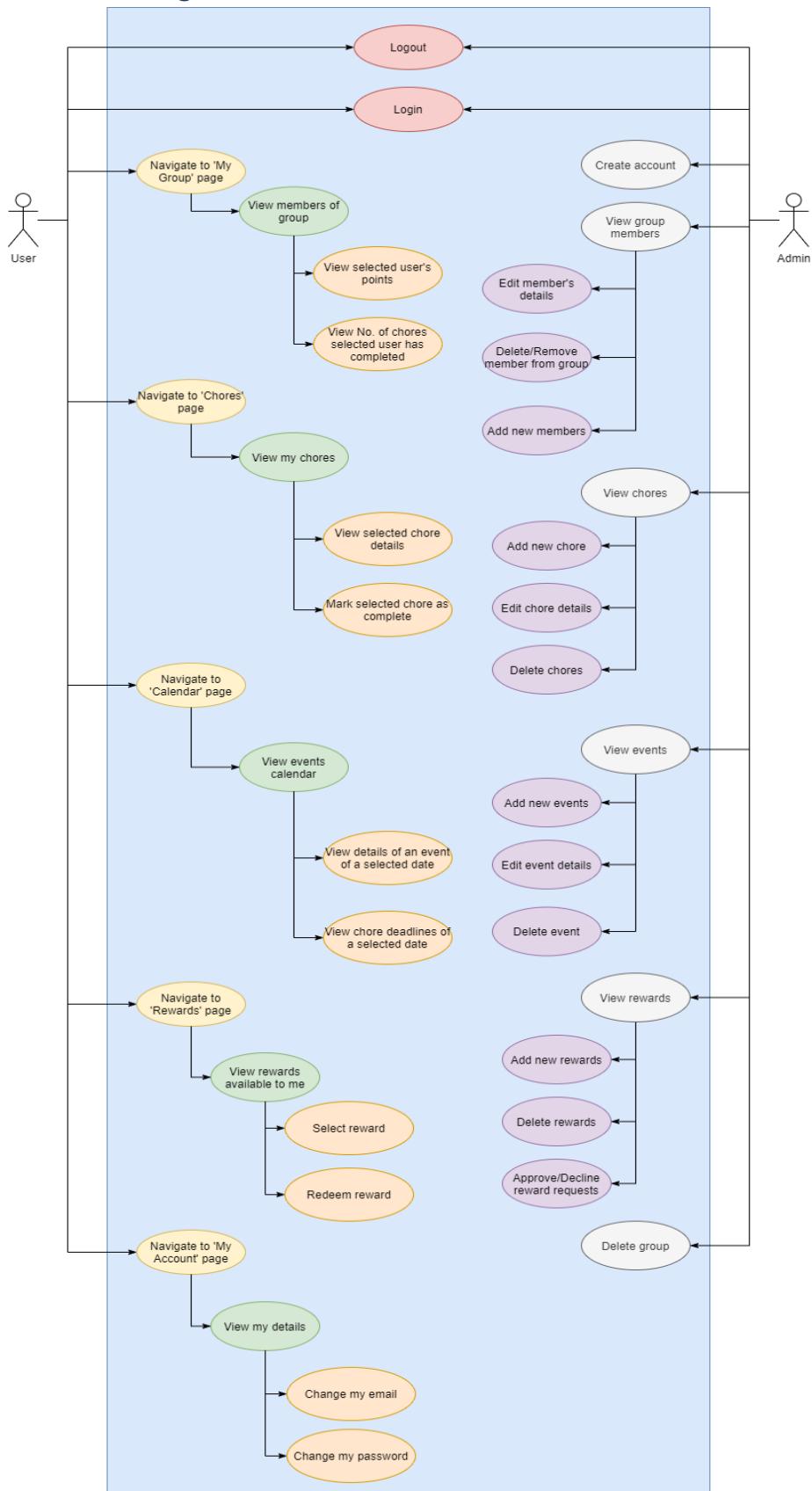


Figure 68 Use Case Diagram

2.9 Sprint Plans

The following are the plans and reviews for each of the seven sprints (0-6).

2.9.1 Sprint Zero: Project Initialisation

Sprint Zero

Start: 05/10/2020

End: 30/10/2020

Task: Project Initialisation

Objectives

- Outline product vision.
- Setup development environment (GitHub, IDE, WAMP, MySQL, AWS)
- Perform risk assessment.
- Generate user stories.

Details

The project initialization documents are due by 22/10/2020 and so writing up the project vision and risk assessment takes priority. Once these two tasks have been completed and I have a clear idea of what I need to do, generate some user stories to aid with the designing of storyboards and functional requirements. When this is complete, the development environment can be set up ready to start sprint one.

Review

What Went Well

The initialisation phase was completed on time before the deadline.

- The product vision was outlined, and an extensive risk assessment was carried out thoroughly.
- Some initial user stories were generated to start website design and development.
- The development environment was partially set up.

Problems Encountered

Overall, sprint zero was completed on time with only one problem:

- Setting up Amazon Web Services for database and webserver hosting took longer than expected due to a lack of experience using the service.

Possible Solutions to These Problems

- Follow tutorials on building an AWS hosted database using RDS.
- Follow tutorials on building an AWS hosted webserver using EC2.

Project Progress

So far, the project's progress has been good. The project has started smoothly, and I'm prepared to start the next sprint.

Figure 69 Sprint Zero - Plan & Review

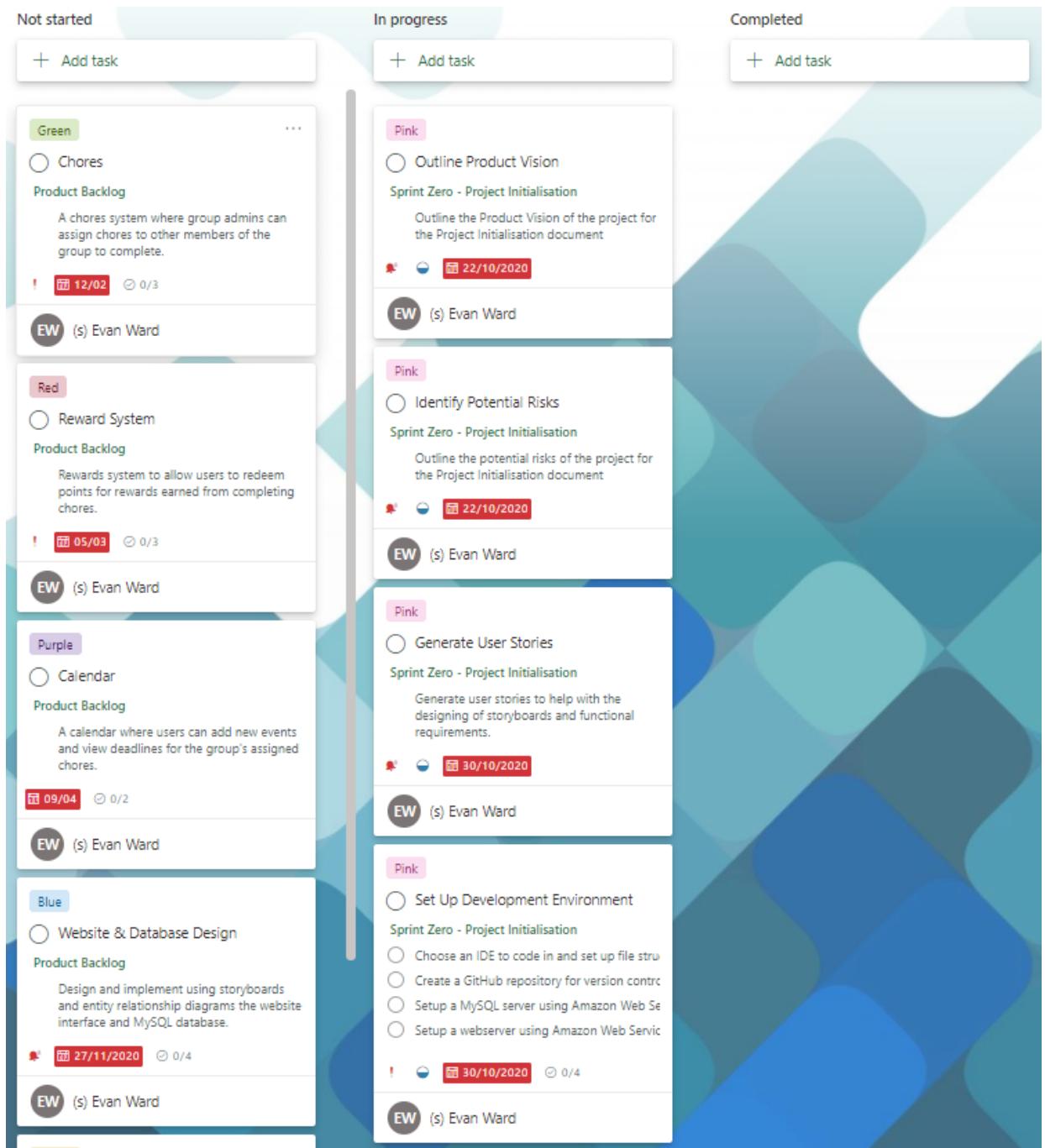


Figure 70 Sprint Zero - Planner Board

2.9.2 Sprint One: Website & Database Design

Sprint One

Start: 02/11/2020

End: 27/11/2020

Task: Website & Database Design

Objectives

- Establish what pages will be needed for the website.
- Design the layout of each page using storyboards.
- Establish what tables will be needed for the database.
- Design the structure of the database using entity relationship diagrams.
- Implement the webpages using HTML & CSS.
- Implement the database using MySQL.

Details

Using the user stories generated in the previous sprint, design using storyboards the layout of the different pages to be used on the website. By doing this, a general colour and layout theme will be established. Attempt to implement these layouts using HTML for the structure of the webpages and CSS for the styling of the webpages. Use the WAMP local webserver for development then deploy the website to a Windows 10 Server hosting Apache hosted by Amazon Web Services.

Using the user stories, design using entity relationship diagrams the initial structure for the database. Attempt to implement the designed database using MySQL hosted on a RDS database server hosted by Amazon Web Services.

Review

What Went Well

Sprint one was completed on time before the deadline.

- Using tutorials, I was able to setup a webserver and MySQL database using AWS.
- Some initial designs for the layout of the different webpages were designed and implemented successfully.
- The initial structure of the database was designed and implemented successfully.

Problems Encountered

Overall sprint one was a smooth process with no major issues except:

- Setting up the remote database and webserver took longer to do than expected.

Possible Solutions to These Problems

- Extended development time for sprint one by an extra week.

Project Progress

So far, I am happy with the progress being made, even with the small delay to started sprint two.

Figure 71 Sprint One - Plan & Review

Not started

- Green**
 - Chores**
 - Product Backlog**
 - A chores system where group admins can assign chores to other members of the group to complete.
 - EW (s) Evan Ward**
 - Red**
 - Reward System**
 - Product Backlog**
 - Rewards system to allow users to redeem points for rewards earned from completing chores.
 - EW (s) Evan Ward**
 - Purple**
 - Calendar**
 - Product Backlog**
 - A calendar where users can add new events and view deadlines for the group's assigned chores.
 - EW (s) Evan Ward**
 - Yellow**
 - User Accounts**
 - Product Backlog**
 - Implement a login and registration system for users to create new accounts and groups.
 - EW (s) Evan Ward**

In progress

 - Blue**
 - EW (s) Evan Ward**
 - Sprint One - Website & Database Design**
 - Design Website Layout**
 - Design 'Login' & 'Registration' pages**
 - Design 'My Group' page**
 - Design 'Chores' page**
 - Design 'Rewards' page**
 - Design 'My Account Page'**
 - Design 'Calendar' page**
 - EW (s) Evan Ward**
 - Blue**
 - Implement Interface**
 - Sprint One - Website & Database Design**
 - Implement 'Login' & 'Registration' pages**
 - Implement 'My Group' page**
 - Implement 'Chores' page**
 - Implement 'My Account' page**
 - Implement 'Rewards' page**
 - Implement 'Calendar' page**
 - EW (s) Evan Ward**
 - Blue**
 - Design & Implement Database**
 - Sprint One - Website & Database Design**
 - Design database structure using entity relational**
 - Use MySQL to build designed database**
 - EW (s) Evan Ward**

Completed

 - Pink**
 - Outline Product Vision**
 - Sprint Zero - Project Initialisation**
 - Outline the Product Vision of the project for the Project Initialisation document
 - EW 22/10/2020**
 - EW Completed by (s) Evan Ward on ...**
 - Pink**
 - Identify Potential Risks**
 - Sprint Zero - Project Initialisation**
 - Outline the potential risks of the project for the Project Initialisation document
 - EW 22/10/2020**
 - EW Completed by (s) Evan Ward on ...**
 - Pink**
 - Generate User Stories**
 - Sprint Zero - Project Initialisation**
 - Generate user stories to help with the designing of storyboards and functional requirements.
 - EW 30/10/2020**
 - EW Completed by (s) Evan Ward on ...**
 - Pink**
 - Set Up Development Environment**
 - Sprint Zero - Project Initialisation**
 - EW 30/10/2020 0/4**
 - EW Completed by (s) Evan Ward on ...**

Figure 72 Sprint One - Planner Board

2.9.3 Sprint Two: User Accounts

Sprint Two

Start: 30/11/2020

End: 08/01/2021

Task: User Accounts

Objectives

- Design and implement a registration system.
- Design and implement a login system.
- Implement the functionality of the My Account page.

Details

Using HTML, build a registration form for the user to input the required details (First & Last Name, Email, Password, DOB). Use JavaScript & jQuery to validate the user's inputs. If all checks pass, encrypt the user's password and safely send the data using jQuery POST to an API that will save the details in the database.

Use HTML to build a basic login form for user to be greeted with when first entering the site. Use JavaScript and jQuery to validate the inputted details. Call an API that will search the database for the user's login email. If the email exists, return the user's encrypted password to the client and check if the two passwords match. If inputted password matches, log user in.

Use PHP to display the details of the logged in user on the My Account Page. Build some forms that allow the user to change their email and password. Use JavaScript and jQuery to validate the inputted data and POST data to an API to update the database if inputted data passes validation checks.

Review

What Went Well

Sprint two was completed on time before the deadline.

- Registration system was successfully implemented as intended with only one small issue.
- Login system was implemented correctly.
- My Account page was set up as intended and correctly displays the user's details and account options.

Problems Encountered

Overall sprint two was a smooth development process with only one issue:

- Encrypting the user's password proved to be a little difficult using methods I had previously used.

Possible Solutions to These Problems

- CryptoJS, an encryption plugin for JavaScript, was used to encrypt the passwords. This solved the issue and did not affect the deadline.

Project Progress

The project's progress made so far has been good. I am happy to continue onto sprint three.

Figure 73 Sprint Two - Plan & Review

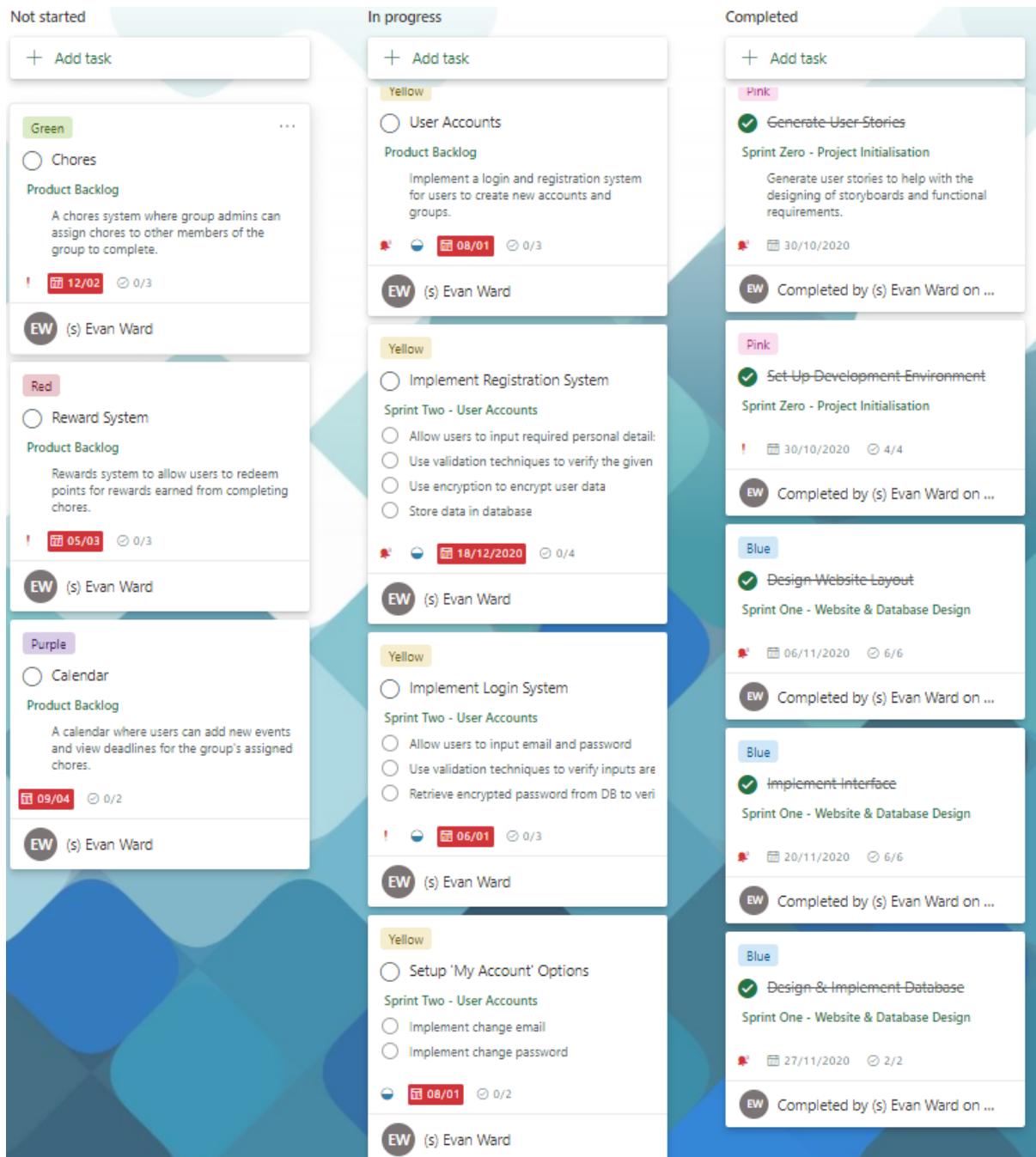


Figure 74 Sprint Two - Planner Board

2.9.4 Sprint Three: Chores

Sprint Three

Start: 11/01/2021

End: 12/02/2021

Task: Chores

Objectives

- Setup household group system.
- Setup chores system.
- Setup chores assignment system.

Details

Implement the functionality of the My Group page by setting up the group system. Add cookies to identify what kind of user is logged in (user/admin). Allow admins to register new users to their group using a HTML form. Use JavaScript and jQuery to validate the user's inputs and POST the data to an API to save the data to the database if all validation checks pass.

Allow admins to add new chores for their group that can be assigned to members of the group. Use a HTML form for the user to input the chore details, validate with JavaScript and jQuery and POST to an API to be saved in the database. Use PHP to get the chores from the database and display on the Chores page when the user loads the page. Allow the admin to edit the details of the displayed chores.

Allow the admin to select a chore and assign the chore to a user within their group. Use jQuery to call an API and add a new chore assignment to the database.

When a user loads the chore page, call an API that gets the user's assigned chores from the database and display them on the page.

Review

What Went Well

Sprint three was completed on time before the deadline.

- The user group system was successfully implemented. Admins can easily add new members to their group.
- The chores system has been successfully implemented. Admins can add new chores that they can then assign to members of their group.
- The admin can reassign chores if needs be. The admin can view the available chores (unassigned chores), the assigned chores and their own chores.
- Users can only view their own assigned chores.

Problems Encountered

Sprint three didn't encounter many issues.

- Organising the admin's chores, available chores and assigned chores on the chore page was hard to layout correctly.

Possible Solutions to These Problems

- Using Bootstrap, I was able to organise the three options into three columns without any issues.

Project Progress

Progress has been steady since the beginning of the project and I am happy with how things are going.

Figure 75 Sprint Three - Plan & Review

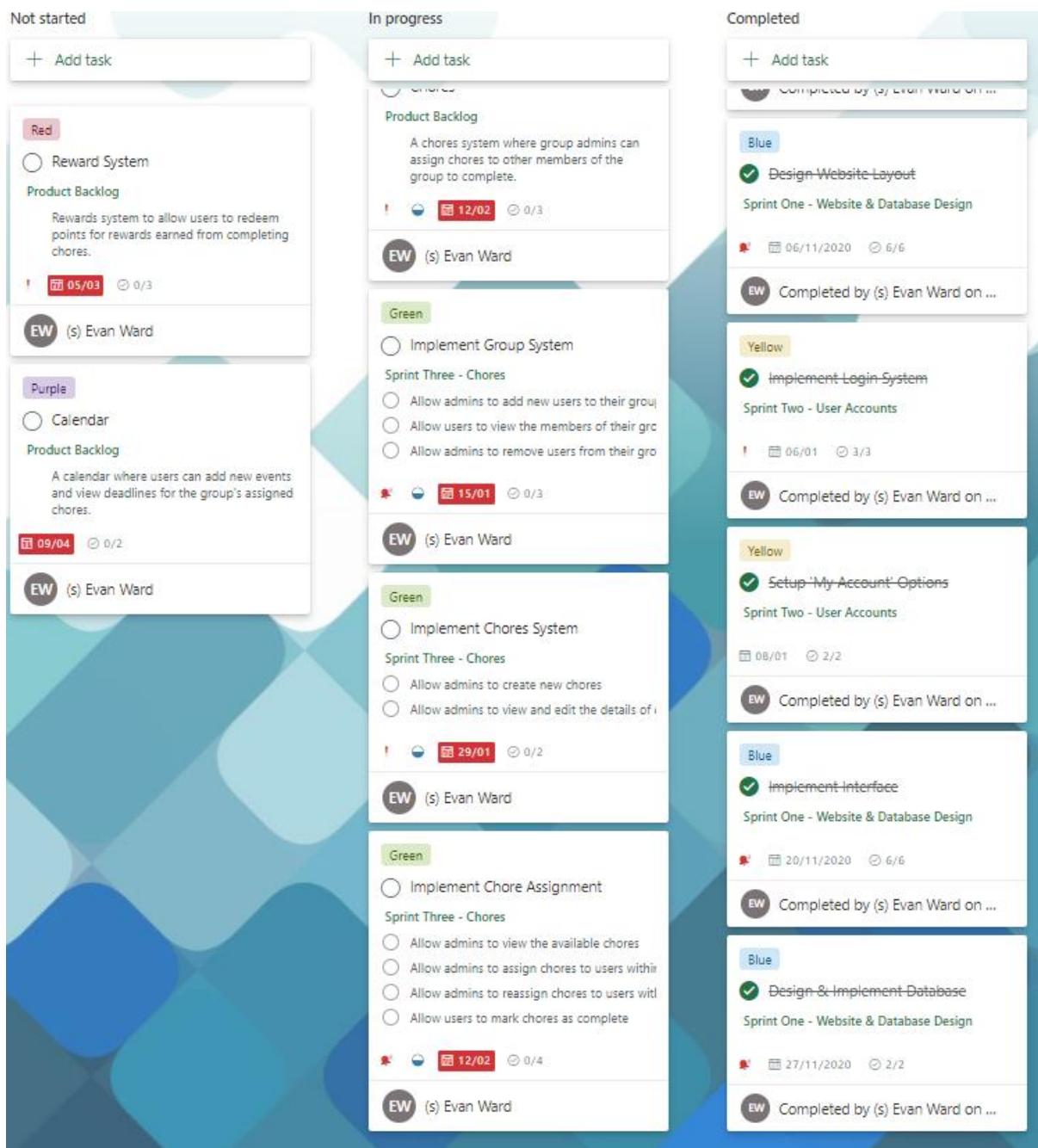


Figure 76 Sprint Three - Planner Board

2.9.5 Sprint Four: Rewards System

Sprint Four

Start: 15/02/2021

End: 05/03/2021

Task: Reward System

Objectives

- Setup a points system.
- Setup a rewards system.

Details

Implement a points system that allows users to earn points if they complete a chore before the deadline and lose points if they complete the chore after the deadline.

Allow the admins to add new rewards that users can trade points for. Use JavaScript and jQuery to validate whether the user has enough points by using an API to fetch the user's points from the database. When a user redeems a reward, deduct their points and save to database.

When a user redeems a reward, create a reward request that admins can either approve or decline. If the request is declined, refund the user's points.

Review

What Went Well

Sprint four was completed on time before the deadline.

- Points system was easy to implement, allowing more time to work on reward system.
- Reward system works as intended and was completed one week earlier before the deadline. This allowed for an additional week to work on the next sprint.

Problems Encountered

Sprint four only encountered one small issue.

- There were a couple of issues with trying to implement the functionality of the points system, such as knowing which chore the user had completed and how many points should be added to the user.

Possible Solutions to These Problems

- A function within the database allowed me to get the points value of the chore being completed using the choreID, allowing me to correctly add points to the user who completed the chore using the userID.

Project Progress

I am really happy with the progress that has been made. Only a few small issues have arisen but none that were too serious. The project is on track to be completed on time.

Figure 77 Sprint Four - Plan & Review

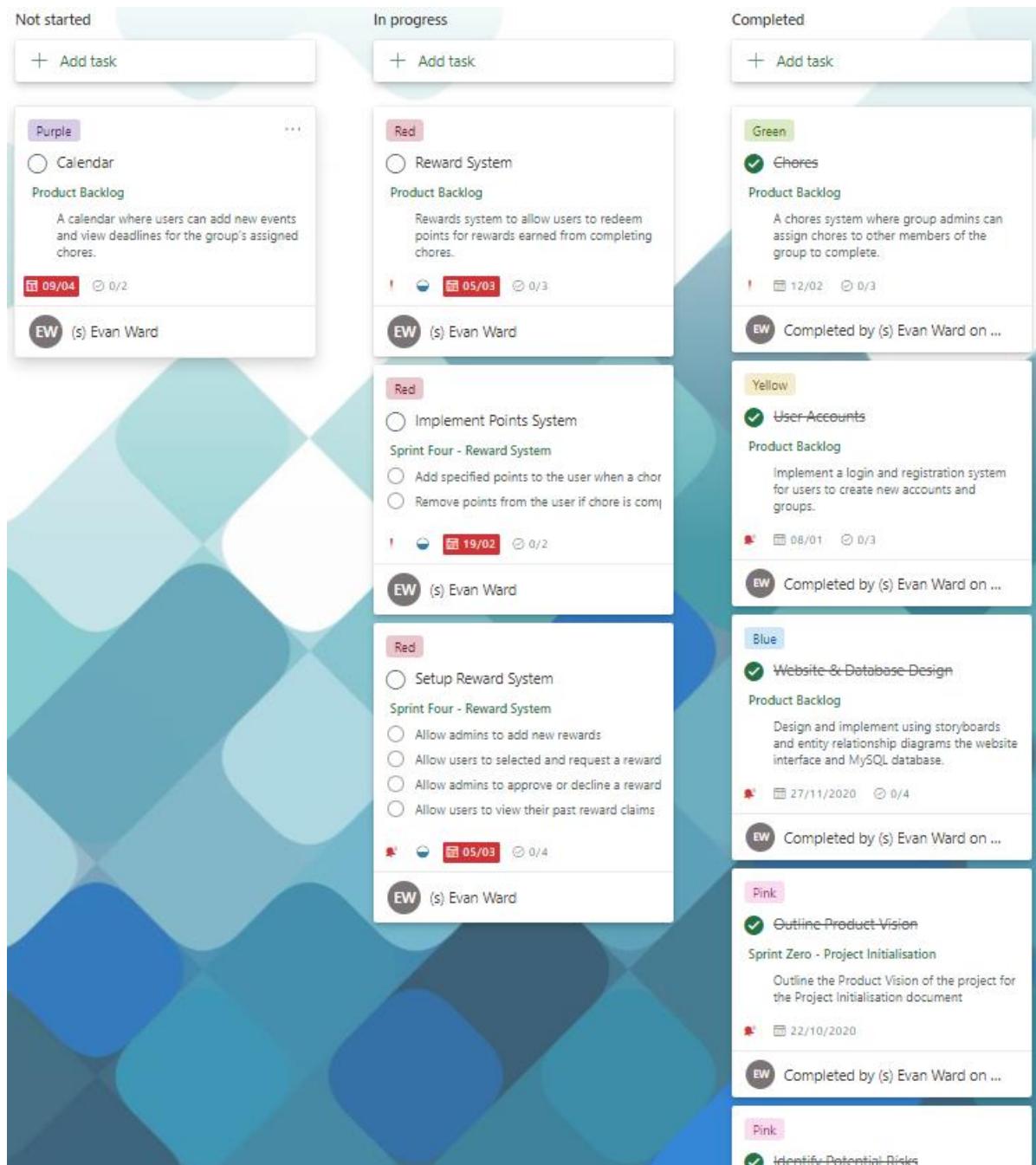


Figure 78 Sprint Four - Planner Board

2.9.6 Sprint Five: Calendar

Sprint Five

Start: 08/03/2021

End: 09/04/2021

Task: Calendar

Objectives

- Setup a calendar for users to view.
- Allow admins to add new events to the calendar.
- Integrate calendar with chores system for displaying assigned chores.

Details

Build a calendar using PHP that displays the correct dates to the user when the user loads the page. Using bootstrap modals, JavaScript and jQuery, build a system that allows users to select a date and view the events for that date.

Build a system that allows admins to select a date and add new events to that date.

Integrate the chores system with the calendar. When a new chore is assigned to a member of the group, create a new event that will be displayed on the group calendar for users to see the deadline of that assigned chore.

Review

What Went Well

Sprint five was possibly the most challenging sprint of all.

- Calendar was successfully implemented after some trial and error.
- Calendar correctly displays the deadline of an assigned chore.
- Admins can easily add new events to the calendar to be viewed by all members of the group.

Problems Encountered

Sprint five took some time to complete due to the challenge of building a custom calendar.

- Initially, I tried to build the calendar using JavaScript. I was unsuccessful doing this but eventually the calendar was setup correctly.

Possible Solutions to These Problems

- Instead of using JavaScript to build the calendar client-side, I decided to use PHP to build the calendar server-side. This meant the calendar was being built and displayed correctly when the page loads.
- PHP made using classes to build the calendar much easier to do.

Project Progress

The calendar was the final main feature that needed to be implemented. I am pleased with how the project has progressed. However, with some more time, I would have liked to implement some additional features such as a chat functionality but these were non-functional requirements so can be implemented at some point in the future..

Figure 79 Sprint Five - Plan & Review

Not started

+ Add task

In progress

+ Add task

Completed

+ Add task

Not started	In progress	Completed
+ Add task	+ Add task	+ Add task
	Purple <input type="radio"/> Calendar Product Backlog A calendar where users can add new events and view deadlines for the group's assigned chores. 09/04 0/2 (s) Evan Ward	Red Reward System Product Backlog Rewards system to allow users to redeem points for rewards earned from completing chores. 05/03 0/3 Completed by (s) Evan Ward on ...
	Purple <input type="radio"/> Implement Events Sprint Five - Calendar <input type="radio"/> Allow user to add and view new events on the website <input type="radio"/> Allow users to view the deadline of the group 09/04 0/2 (s) Evan Ward	Green Chores Product Backlog A chores system where group admins can assign chores to other members of the group to complete. 12/02 0/3 Completed by (s) Evan Ward on ...
	Purple <input type="radio"/> Implement Calendar Sprint Five - Calendar <input type="radio"/> Add a Calendar class to the PHP source code <input type="radio"/> Build the calendar on the website and customize it 26/03 0/2 (s) Evan Ward	Yellow User Accounts Product Backlog Implement a login and registration system for users to create new accounts and groups. 08/01 0/3 Completed by (s) Evan Ward on ...
		Blue Website & Database Design Product Backlog Design and implement using storyboards and entity relationship diagrams the website interface and MySQL database. 27/11/2020 0/4 Completed by (s) Evan Ward on ...
		Pink

2.9.7 Sprint Six: Clean Up & Testing

Sprint Six

Start: 12/04/2021

End: 30/04/2021

Task: Clean Up & Testing

Objectives

- Perform usability study.
- Clean up code a file structure.

Details

Create a questionnaire that gives test users some tasks to complete and provide some feedback on their experience of doing these tasks. Analyse the feedback received and act on some of the feedback I receive if time permits it.

Ensure code is structured properly and commented appropriately. Reorganise the project's file structure to be more appropriate and logical.

Review

What Went Well

Sprint six was a success.

- The usability study was a success. I received plenty of responses to the questionnaire.
- Not much code needed to be cleaned up. Only a few pieces of redundant and duplicate code needed to be removed. The file structure is now appropriate and logical.

Problems Encountered

- Although the usability was a success, I wasn't able to act on some of the feedback received due to time constraints.

Possible Solutions to These Problems

- The feedback can be used to improve the solution sometime in the future.

Project Progress

Sprint six marked was the final sprint and marked the end of the project. I am very pleased with how this project has turned out. All that's left to do is continue work on the final report.]

Figure 80 Sprint Six - Plan & Review

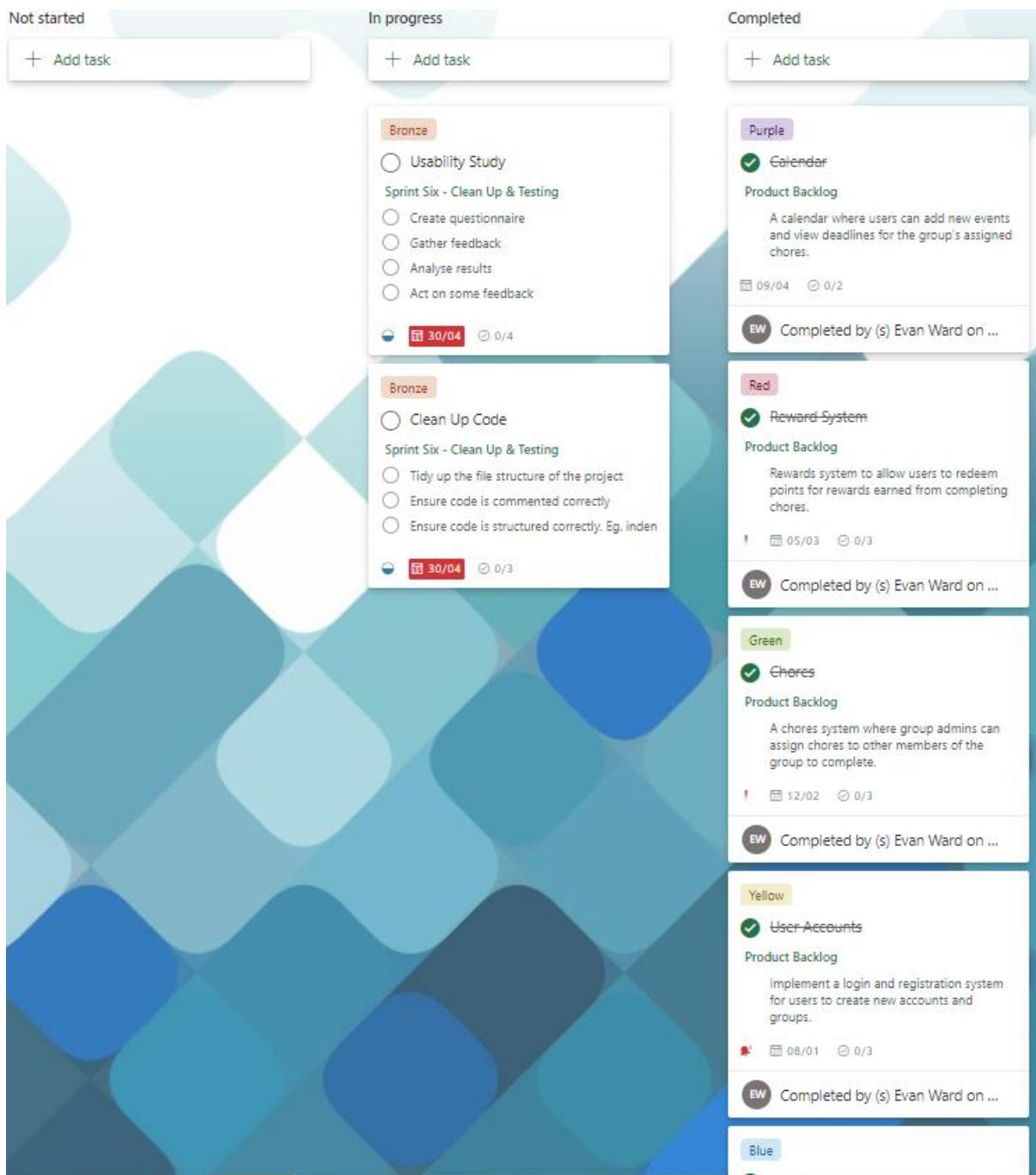


Figure 81 Sprint Six - Planner Board

2.10 Usability Study

2.10.1 Questionnaire

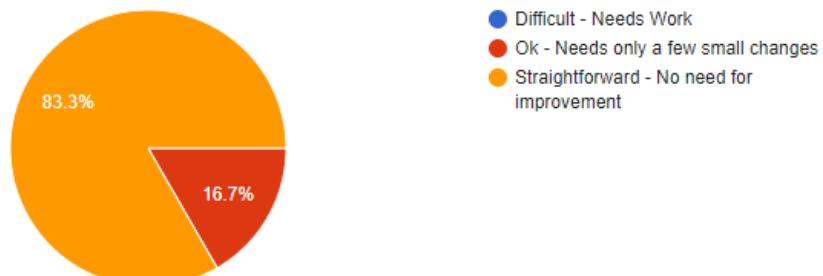
The full questionnaire can be found using the following link:

<https://forms.gle/dsqJGNTwyiQFgGSaA>

2.10.2 Results

Registration & Login was:

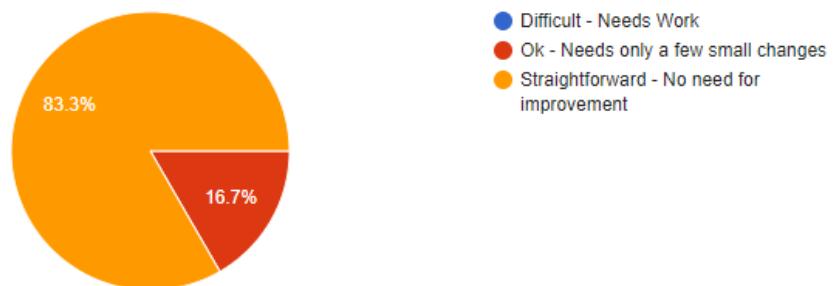
6 responses



My Group

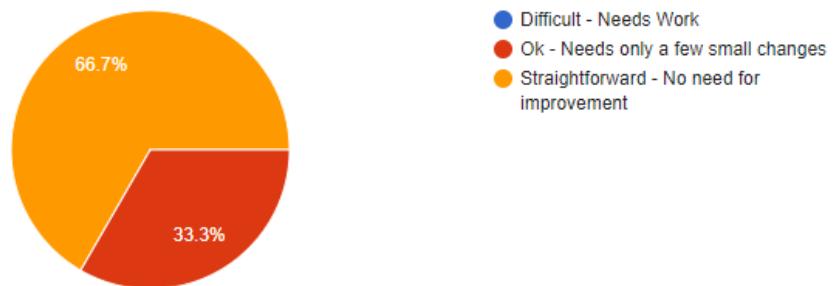
Adding three new members was:

6 responses



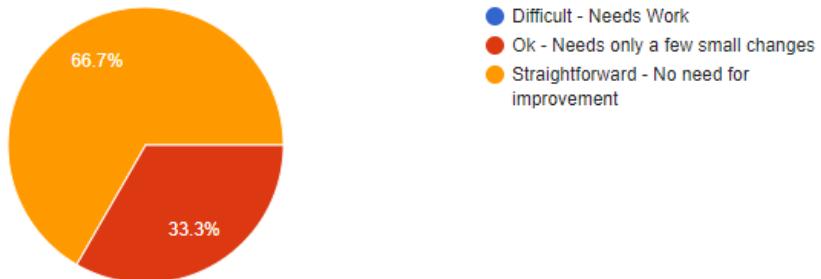
Editing user points was:

6 responses



Removing group member was:

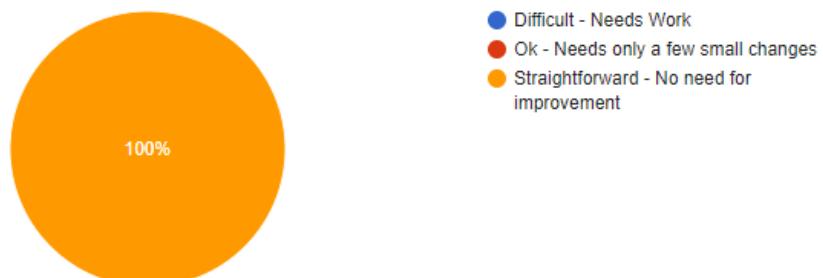
6 responses



Chores

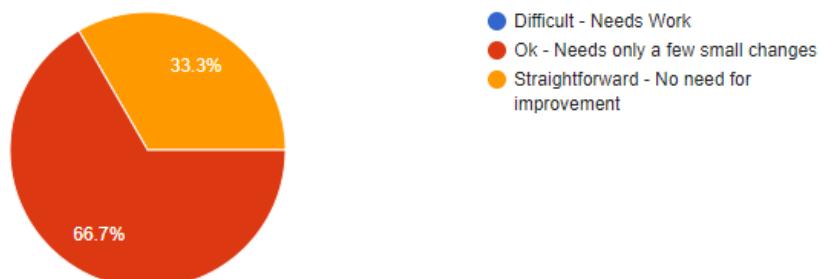
Adding four new chores was:

6 responses



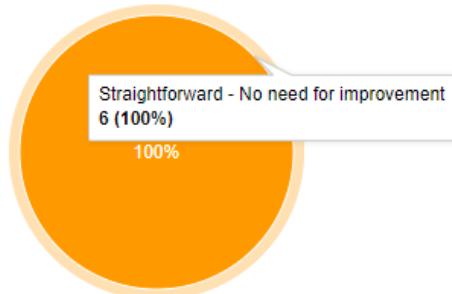
Editing a chore's details was:

6 responses



Assigning chores was:

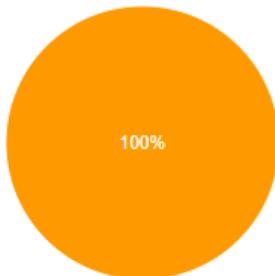
6 responses



- Difficult - Needs Work
- Ok - Needs only a few small changes
- Straightforward - No need for improvement

Completing a chore was:

6 responses



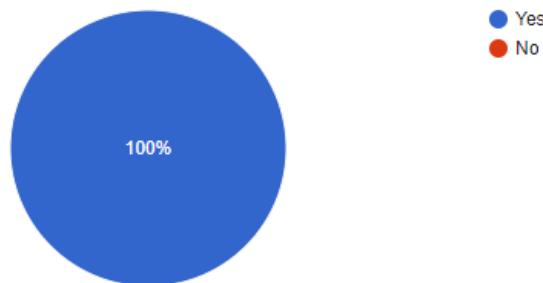
- Difficult - Needs Work
- Ok - Needs only a few small changes
- Straightforward - No need for improvement

Calendar

Can you easily see your assigned chore on the calendar?

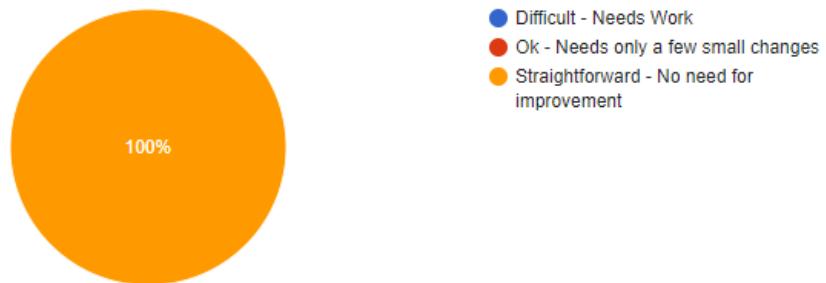


6 responses



Adding a new event was:

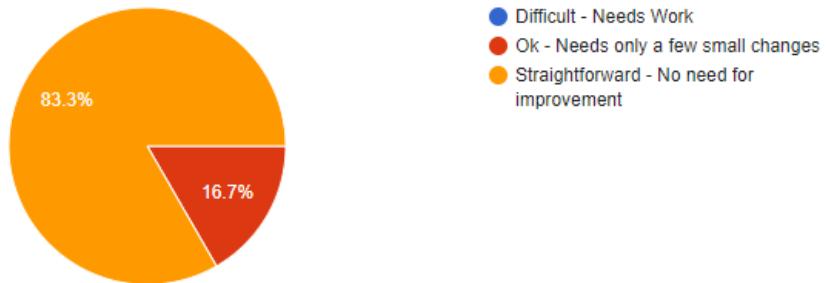
6 responses



Rewards

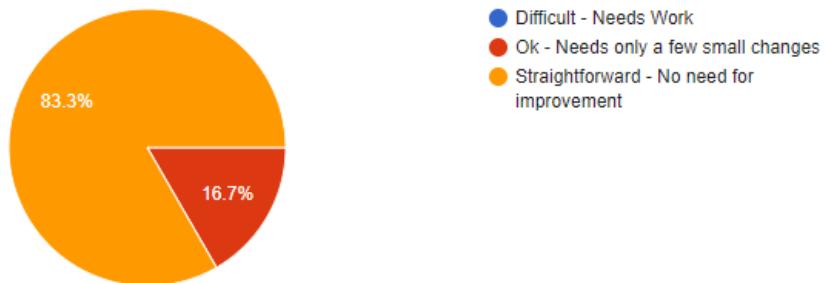
Adding a new reward was:

6 responses



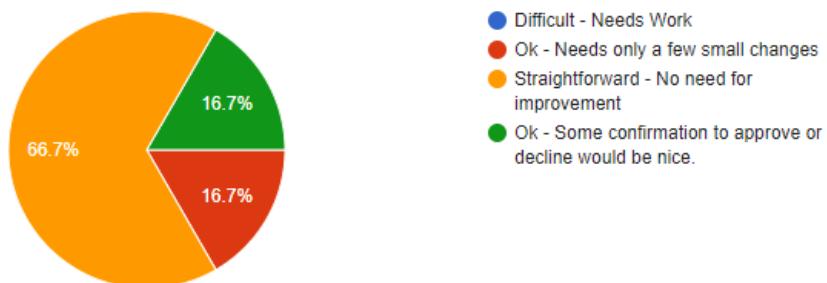
Redeeming a reward was:

6 responses



Approving/Declining a reward was:

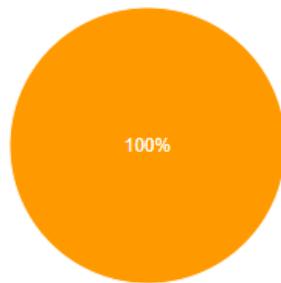
6 responses



My Account

View past rewards was:

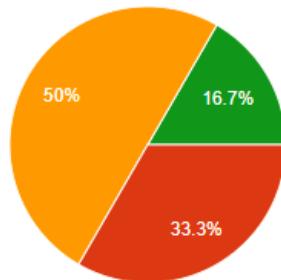
6 responses



- Difficult - Needs Work
- Ok - Needs only a few small changes
- Straightforward - No need for improvement

Changing your email was:

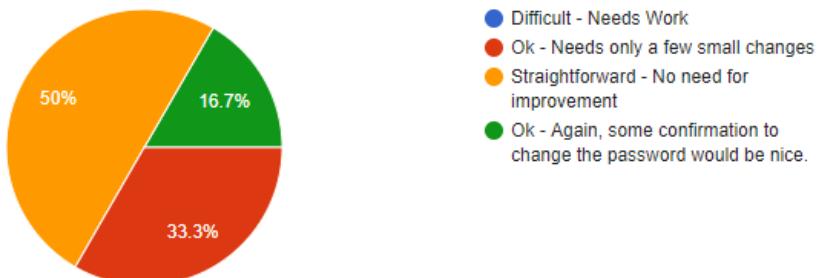
6 responses



- Difficult - Needs Work
- Ok - Needs only a few small changes
- Straightforward - No need for improvement
- Ok - Some confirmation to change the email would be nice.

Changing your password was:

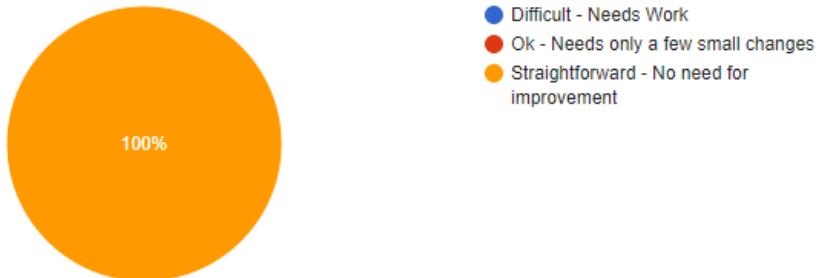
6 responses



- Difficult - Needs Work
- Ok - Needs only a few small changes
- Straightforward - No need for improvement
- Ok - Again, some confirmation to change the password would be nice.

Disbanding your group was:

6 responses



- Difficult - Needs Work
- Ok - Needs only a few small changes
- Straightforward - No need for improvement

General Feedback

Do you have any comments about the theme of the website?

5 responses

Very user friendly colour scheme. Easy to read words.

General theme is nice. Good colour scheme.

Colour scheme is simple and clean. Words stand out well and are easy to read.

Nice colours used.

Consistent theme across the whole website

Do you have any comments about the layout of the website?

5 responses

Layout is simple and similar across the whole website.

Some buttons should be reorganised to a more relevant location, such as moving the new reward button to underneath the rewards table.

Simple consistent layout across the website.

Layout is simple to navigate.

Simple layout, easy to navigate.

Do you have any comments about your overall experience using the website?

6 responses

Is website to use overall, tasks were clear to do.

Disbanding a group has plenty of confirmations to ensure the user is not making a mistake - very good.

Relatively simple to use. Not many issues with the site.

Overall good experience. Some confirmation popups would be useful for changing some details of things.

Easy to use, maybe make editing chore details a bit more clean, eg. hide the chore assign options.

Simple to use. Some confirmation popup for changing your details would be good.