Comp3006 Report – This Is It

Video Demonstration: https://youtu.be/V5Obf46W_Gc

Requirements

"This Is It" is an interactive newsletter service. The website allows users to register for an account, view weekly newsletters relating to a topic of their interest, and message others within a global webchat.

This application is aimed at people who enjoy reading the news or reading about new topics of interest. This application is also aimed at users who have an interest in different topics who wish to keep up to date with the recent information/news regarding a topic. The application is also aimed at users who wish to meet new people with similar interests in a topic. This can be done by a global webchat.

One of the main features of the website is the ability to read newsletters relating to a chosen topic. The user can select a topic to view from the list of many topics and view all newsletters related to that topic. The user can then select a newsletter to read or download. This is the main feature of the websites purpose and so was essential to implement it right.

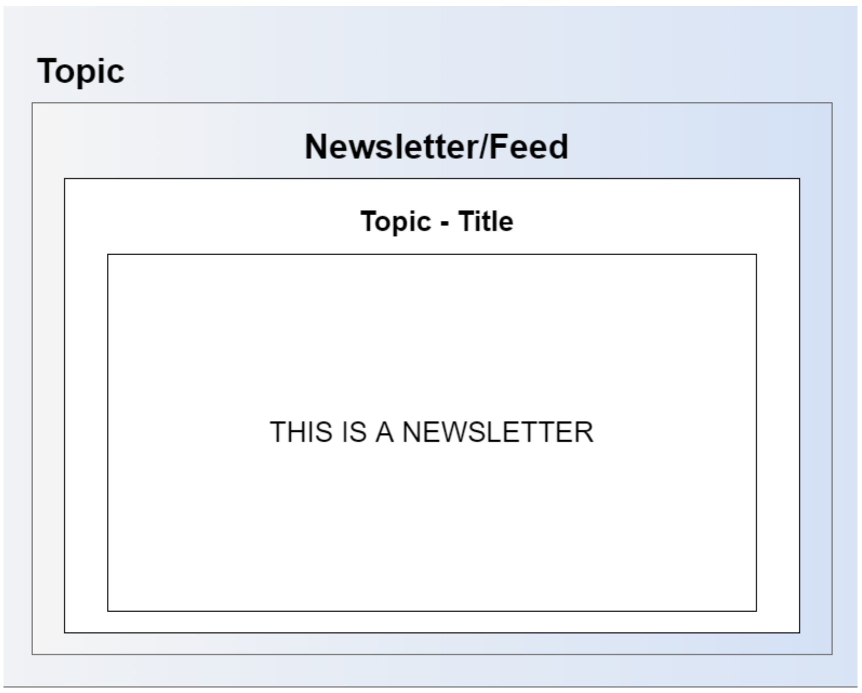


Figure 1 Newsletter Storyboard

Another big feature of the website is the webchat. The chat allows user to message eachother in a global chat, meeting new people with similar interests and ideas. It's also a way for users to talk about their thoughts of the latest news, information and the future of any of the topics. This feature is an essential part of the website. It allows users to create their own community of people who all have somehting in common and so it was important to get it right.

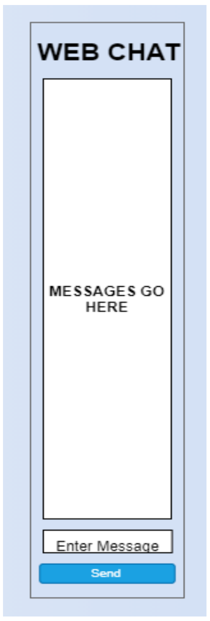


Figure 2 Webchat Storyboard

Another feature of the website is the ability for users to register for an account and then login using this account. This allows the user to create their own custom username which will then be displayed when messaging others in the webchat. It is important that the user has complete control of their data. To allow this, they have the option to change their email and password and delete their account along with all their data if they wish to do so. This was an important feature to include because without it, there is no way of identifying a user within the chat, and it is important to enusre GDPR rules are met.

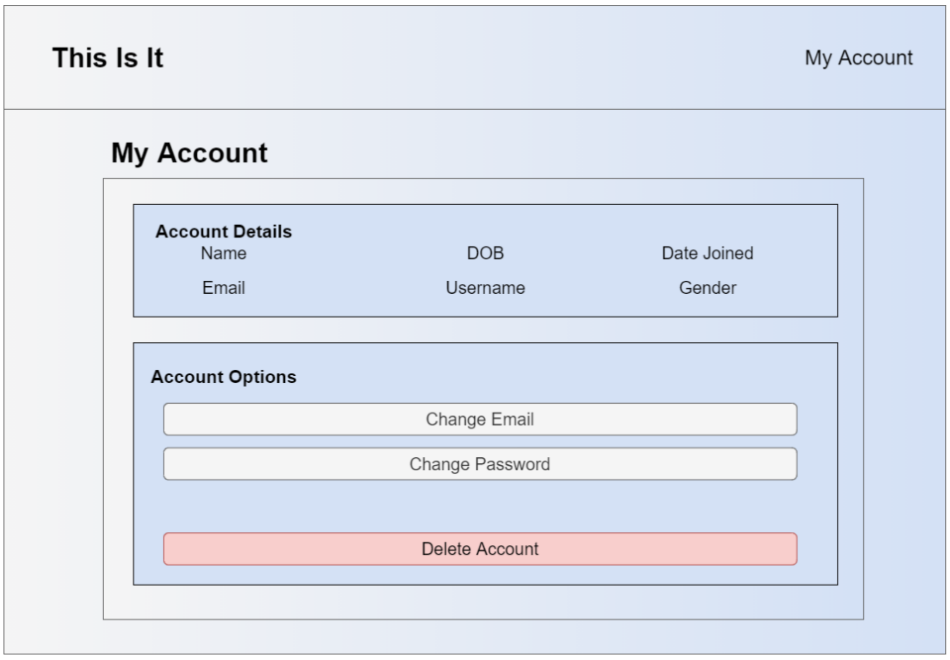


Figure 3 Account Options Storyboard

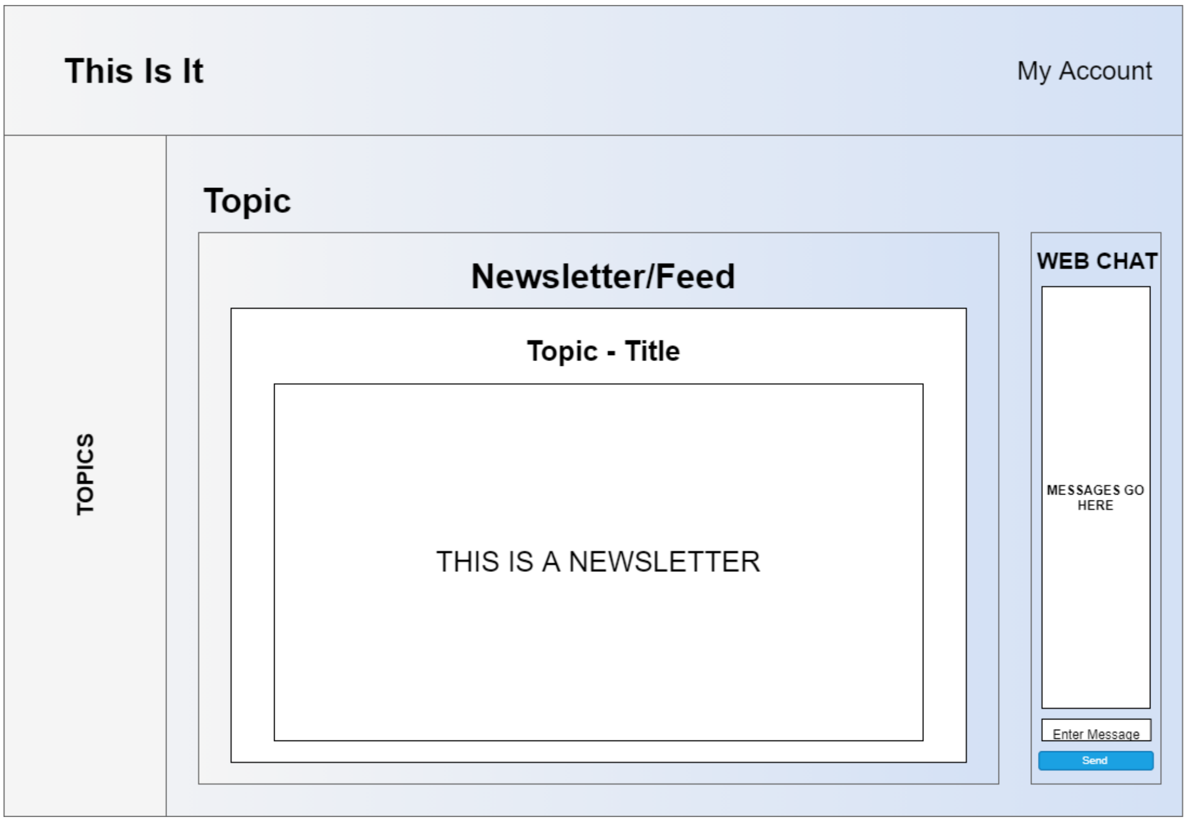


Figure 4 Main Page Storyboard

Design

This website uses a Client-Server architecture. Users will interact with clients through the webpages. Clients will then communicate with the Server to request or send data based on the user's inputs and requests. The Server will then send requests to the MongoDB Database to retrieve or send data. This data is then returned to the Client by the Server to be viewed by the User. In this case, Users can select a topic via the View, the client will send a request to the server to get all newsletters relating to that topic, the Server will then fetch the relevant newsletters from the database and return them to the client to be processed.

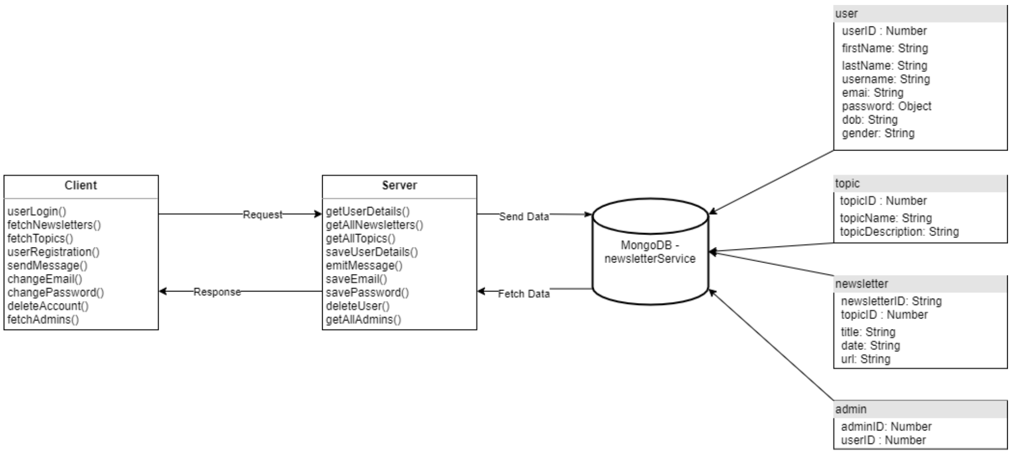


Figure 5 Application Architecture

In order for the client to communicate with the server, APIs are used. Clients can send or request data from the server by sending an AJAX request. The server can recognise this request, process it, and then do what every needs to be done. In this case, clients can send an AJAX POST request, with the 'userID' embedded in the request, to the server to fetch a user's personal details from the MongoDB Database being used. The server will intercept this request, extract the 'userID' through the request's body, and execute a function that will get a user's information from the database, using the 'userID', in JSON format. This JSON data is then returned to the client to be processed.

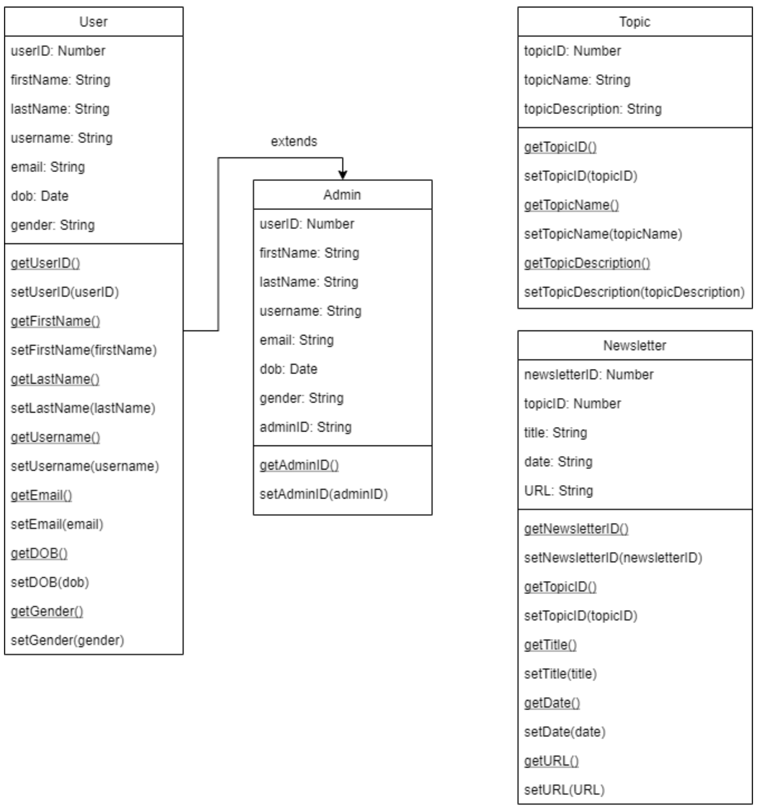


Figure 6 UML Class Diagrams

This website uses the MVC Architecture. The Model contains all the classes that are used to structure data being fetched from the database. In this case, the Model contains classes for the User, Topic and Newsletter. The View contains all the webpages that the user will view and interact with. In this case, the View contains pages for the 'main'/'index' page, the 'account' page, 'login' page and 'registration' page. The controller contains the APIs that will communicate with the database and send the information back to the Views to be displayed to the user. In this case, the Controller contains all the JavaScript code that will send AJAX requests to the Model/Server to be return data to relevant page to be viewed.

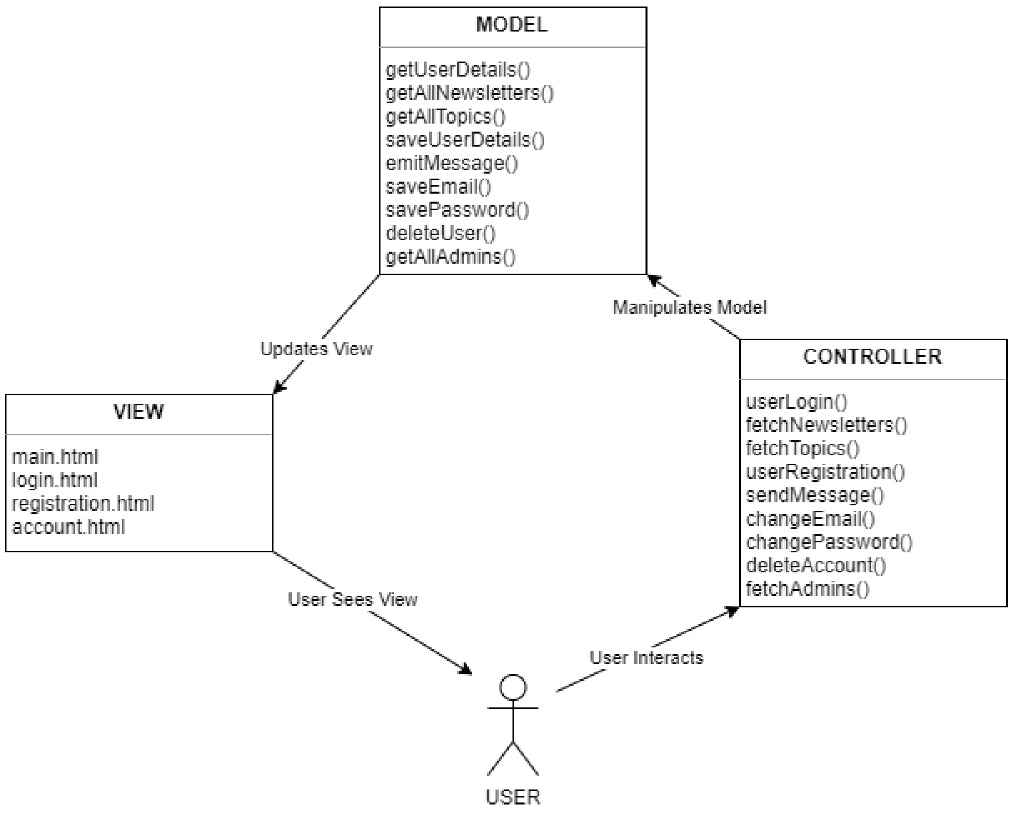


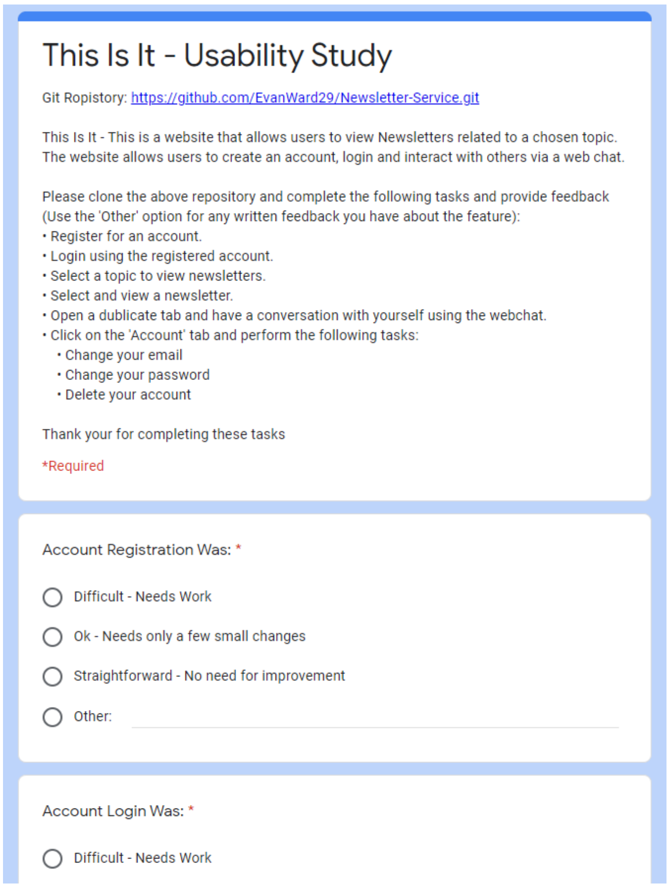
Figure 7 MVC Pattern

The Client-Server and MVC structures used for this website are widely known architectures. They are industry standard and so are widely accepted. The Client-Server architecture was appropriate in this case because clients/users need a way to store data as well as request data as securely as possible. Clients cannot do this on their own safely and so communication between the client to server and server to database and vice versa was important to have. MVC was appropriate in this case because it provides a clear and understanding structer to the application. Each part of MVC has it's own unique and important part to play in the application. In this case, the view allows the website to display the webpages to the user, the controller allows the client to make AJAX requests to the server to fetch or send data, and the model is able to process these requests.

Testing

Performing unit tests on this application was of significant importance. It made identifiying and fixing bugs early on in the development process much easier to do. The two main functions of the application that was tested was the Client-Side coding and Server-Side coding. Client-Side testing is performed on a client's browser. In this case, testing was used to ensure the topics were displayed correctly on the client's browsers. The different topics are stored within the database. When the page loads, the server sends a list of topics to the client to be shown. The unit test will then check that the html for the topics is present on the page by checking if the ‘id’ exists. Server-Side testing is performerd on the main server. In this case, server-side testing was used to test the APIs used to ensure the correct data was being retrieved from the database. As an example for this website, a test was performed on the API that registers a new user. A POST request is sent to the server containing test data for a new user. If the user is added to the database, the test was a success.

Another form of testing that was used is a usability study. Usuability studies are vital for developers as it allows for valuable feedback from real users who are likely to use the application. In this case, a usuability study allowed me to feedback on my website. For this, I used Google Forms to create a questionnaire with instructions to perform specific tasks on the website and then provide feedback.


This Is It - Usability Study

Git Ropistory: https://github.com/EvanWard29/Newsletter-Service.git

This Is It - This is a website that allows users to view Newsletters related to a chosen topic. The website allows users to create an account, login and interact with others via a web chat.

Please clone the above repository and complete the following tasks and provide feedback (Use the 'Other' option for any written feedback you have about the feature):
- Register for an account.
- Login using the registered account.
- Select a topic to view newsletters.
- Select and view a newsletter.
- Open a dublicate tab and have a conversation with yourself using the webchat.
- Click on the 'Account' tab and perform the following tasks:
  - Change your email
  - Change your password
  - Delete your account

Thank your for completing these tasks

*Required

Account Registration Was: *

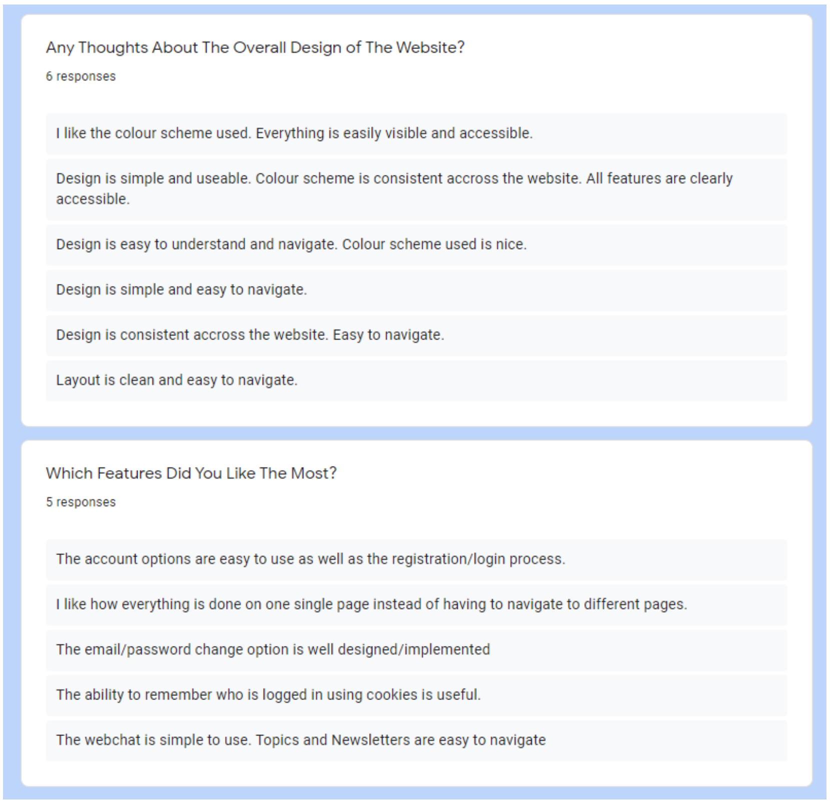
- ○ Difficult - Needs Work
- ○ Ok - Needs only a few small changes
- ○ Straightforward - No need for improvement
- ○ Other:

Account Login Was: *

- ○ Difficult - Needs Work

Figure 8 Usability Questionnaire

From this I was able to see what was good about the website, and what wasn't so good that needed improving. For example, from the feedback I received I was able to improve the webchat functionality of the website by making it bigger and more clearer as the feedback said the message box was too small.



Any Thoughts About The Overall Design of The Website?

6 responses

I like the colour scheme used. Everything is easily visible and accessible.

Design is simple and useable. Colour scheme is consistent accross the website. All features are clearly accessible.

Design is easy to understand and navigate. Colour scheme used is nice.

Design is simple and easy to navigate.

Design is consistent accross the website. Easy to navigate.

Layout is clean and easy to navigate.

Which Features Did You Like The Most?

5 responses

The account options are easy to use as well as the registration/login process.

I like how everything is done on one single page instead of having to navigate to different pages.

The email/password change option is well designed/implemented

The ability to remember who is logged in using cookies is useful.

The webchat is simple to use. Topics and Newsletters are easy to navigate

Figure 9 Usability Study Feedback

Overall, this testing strategy worked very well. Using unit tests I was able to ensure that important data is presented on the browsers client correctly. More importantly, it allowed me to test the APIs being used for requesting and returning data between the client, server and database. This was critical because if the APIs did not work as intended, the website would not function correctly. It also ensured their was always a connection between the server and database.

<u>DevOps Pipeline</u>

The development environment is a very important aspect of application development. My development environment consisted of Visual Studio for writing code, the Chrome Development Tools for debugging the website, GitHub for version control and sprint planning, and Travis-Ci for continuous integration.

Visual Studio was my choice of IDE because of it's variety of libraries and its smart intellisence. The intellisence helped me to complete any JavaScript functions that I was not familiar with or struggled to remember what the syntax was. This was a very useful feature and helped immensely.

Google Chrome's Development Tools was my main way of debugging the javascript I was writing. Visual Studio allowed me to identify issues with the code's structure such as syntax errors, whereas Chromes's Debugger allowed me to debug the outputs of the code that I produced. This was done use 'console.log()'. Using this, I was able to check that data reqeusted from the database was correct, user inputs were fetched correctly and jQuery event listeners worked as intended. The function 'console.log()' was also useful for testing Server-Side code as it allowed me to check if data was being sent to the server through APIs correctly.

GitHub was used for version control. I chose to use GitHub because I'm familiar with how to use it and I was able to make use of GitHub's project board for planning Sprints. This allowed me to easily plan my time on developing a feature.

For continuous integration, I used 'travis-ci.com'. The alternative option was GitHub Actions but I chose to use Travis-Ci because I understood how it worked more than I did GitHub Actions. Using Travis-ci allowed me to perform integration tests on the server. Whenever a new feature is added and the branch of that new feature is merged into the main branch of the application, an integration test is automatically carried out on 'travis-ci.com'. This allowed me to test the new feature was working correctly on the main version of the application. Using this I was able to identify any bugs or issues with the application and fix them appropriately so that the two version can be integrated.

Personal Reflection

While developing this application, things didn't always go to plan. Some features were time consuming to implement due to them being complex or me not having all the knowlegde to complete to task. This lead to some bugs that took time to fix fully. For example, getting the topic list to display correctly on the page and having the user be able to interact with the list took some time complete due to JavaScript being synchronous at it's core. This meant I had to get my head around asynchronous functions and how they worked and were used in JavaScript. This is something I didn't take into account when planning my time and so some sprints were not completed when I planned them to be.

Although there were some issues while developing the application, there are some things that worked well. My understanding of how APIs are used by Client's to interact with the Server has improved. I also learnt how to perform Integration Testing which is widely used in industry and so is very useful to know. Making better use of Chrome's Debugging Tools was also a good takeaway from this as it means I can debug code much easier in the furture.

There are some things from this project that I will definitley be taking away to my next project. One of these things is a much better understanding of JavaScript and jQuery and how they are used in industry. Another important takeaway from this project would be time management. Next time I will definitley be planning my time better, allowing for extra time for debugging and research of code. The use of Unit and Integration testing will also play a big part in how I test and develop future projects. My undersatnding of APIs has also improved immensily and will deffinitley help with any future projects.

Words: 1981

<u>Appendix</u>

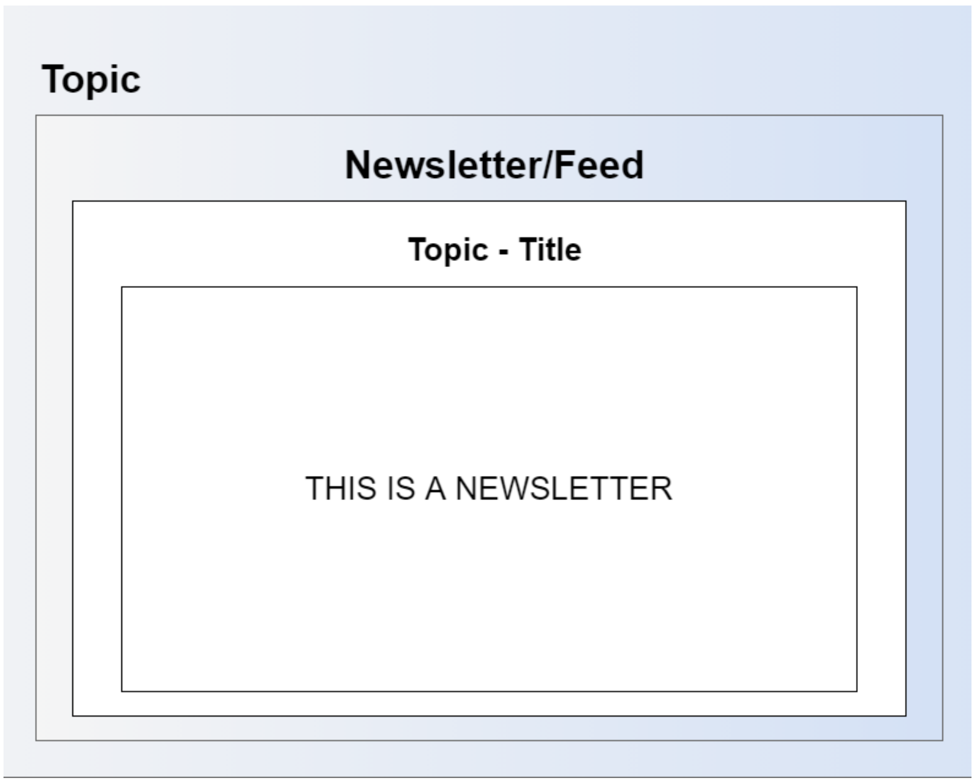


Figure 10 Newsletter Storyboard

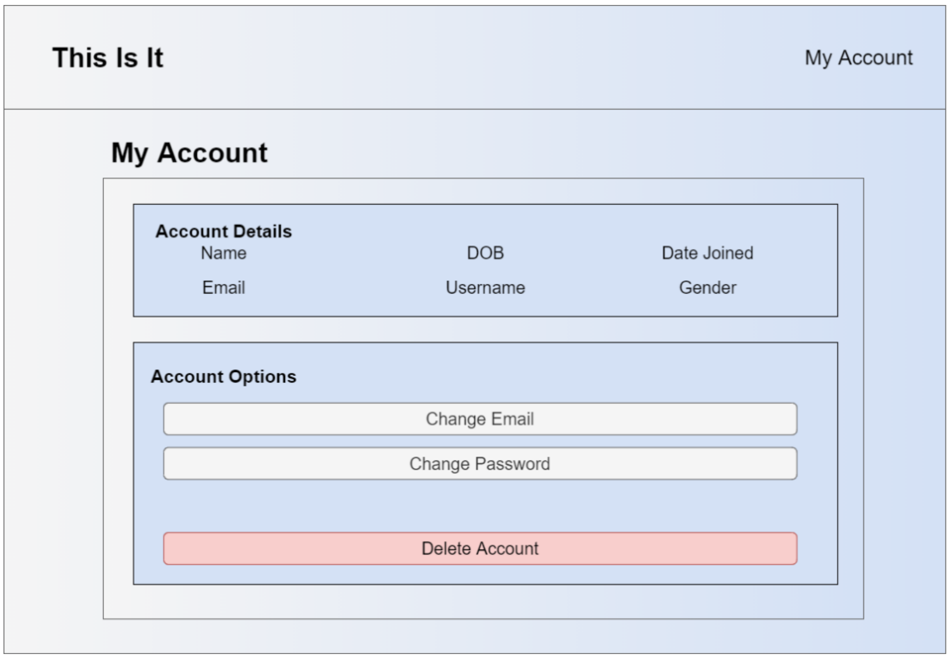


Figure 3 Account Options Storyboard

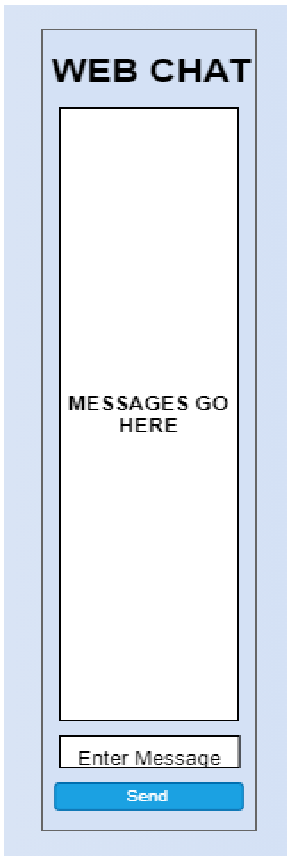


Figure 11 Webchat Storyboard

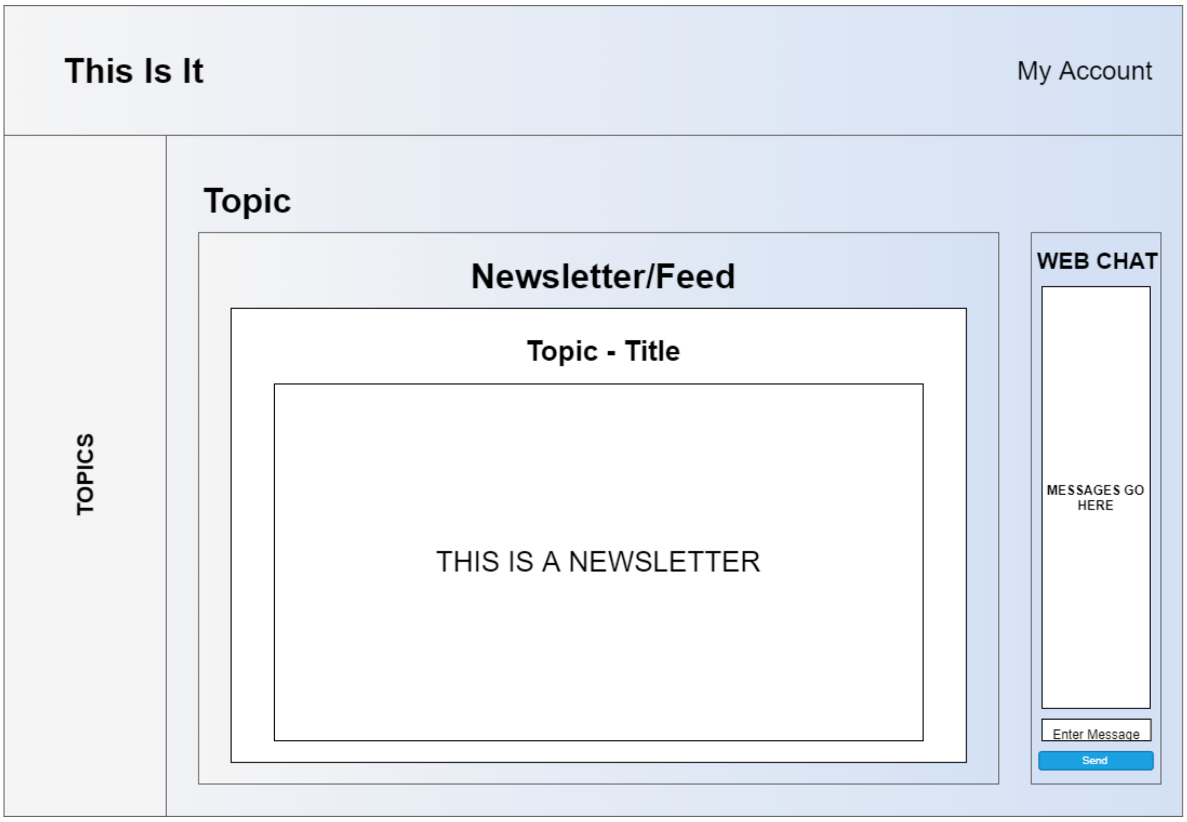


Figure 4 Main Page Storyboard

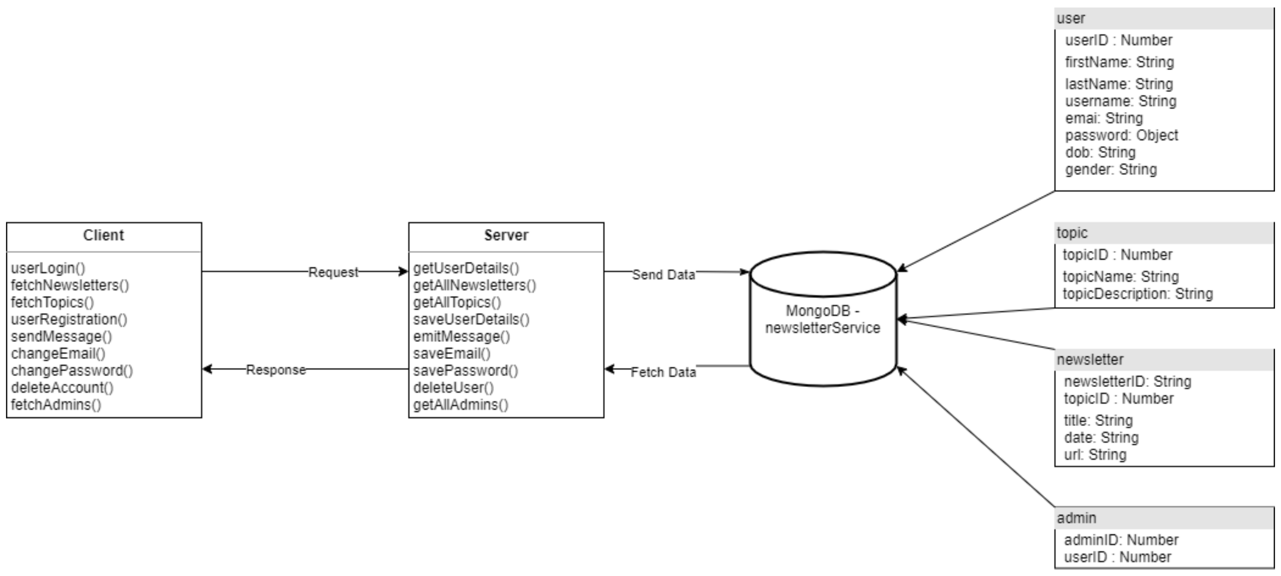


Figure 5 Application Architecture

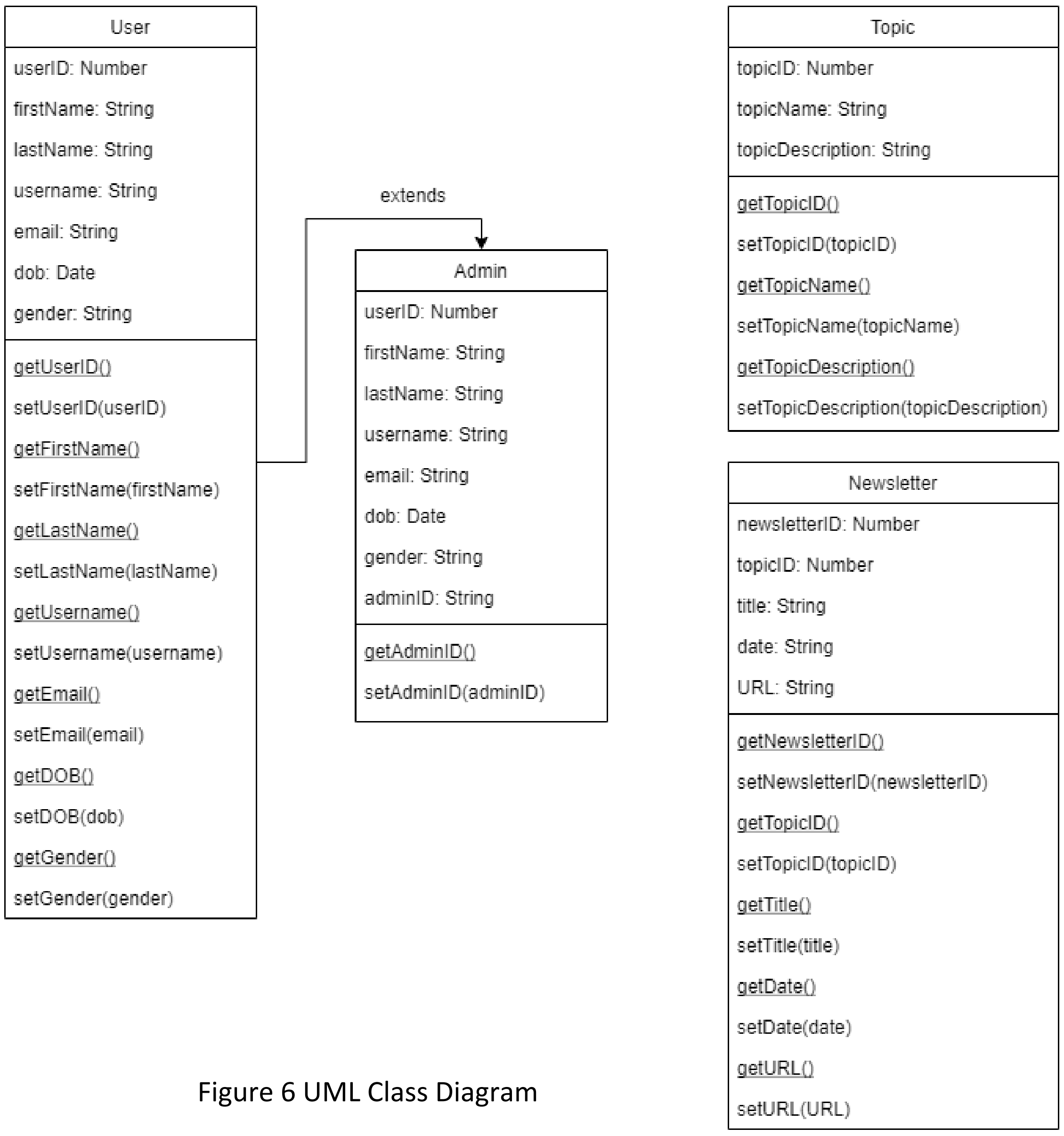


Figure 6 UML Class Diagram

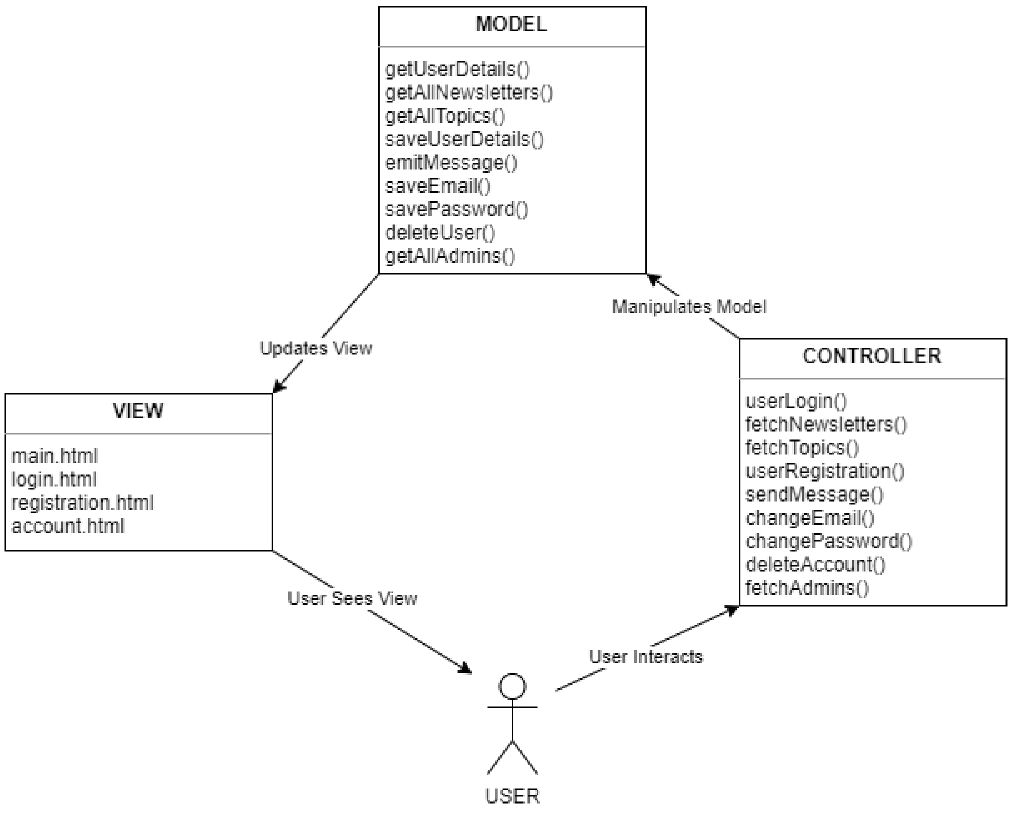
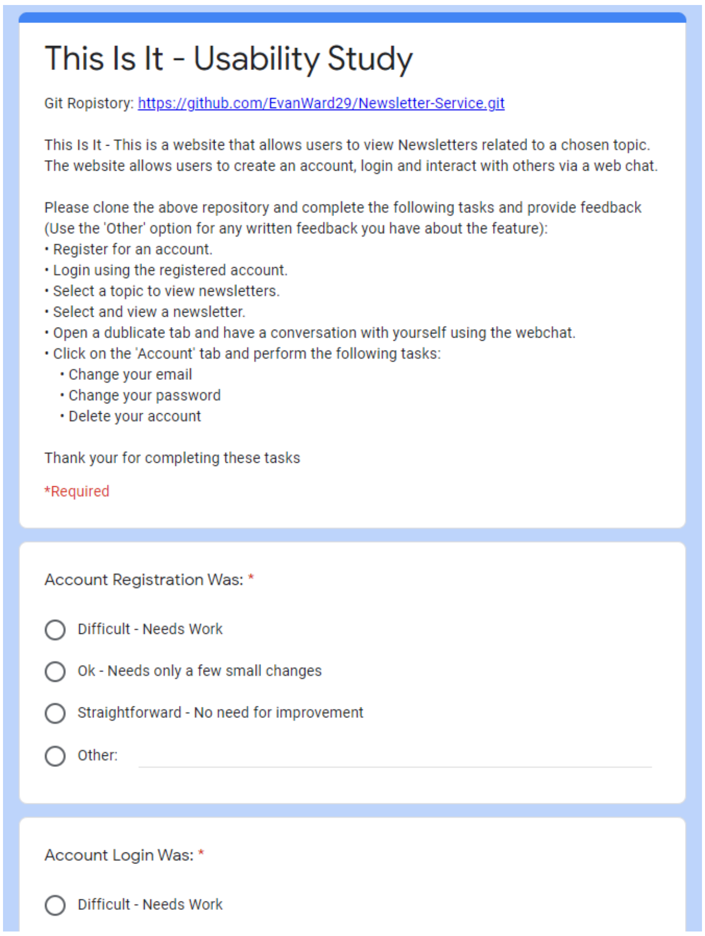


Figure 7 MVC Pattern

Usability Study Questionnaire: https://forms.gle/oYBSWUuz6nEN4Dey9

Usability Study Responses:
https://docs.google.com/spreadsheets/d/1_PV4d18XX_qpKQcsK7b0xPAfc3Y9c18iKojSZopOr6A/edit?usp=sharing



This Is It - Usability Study

Git Ropistory: https://github.com/EvanWard29/Newsletter-Service.git

This Is It - This is a website that allows users to view Newsletters related to a chosen topic. The website allows users to create an account, login and interact with others via a web chat.

Please clone the above repository and complete the following tasks and provide feedback (Use the 'Other' option for any written feedback you have about the feature):
• Register for an account.
• Login using the registered account.
• Select a topic to view newsletters.
• Select and view a newsletter.
• Open a dublicate tab and have a conversation with yourself using the webchat.
• Click on the 'Account' tab and perform the following tasks:
  • Change your email
  • Change your password
  • Delete your account

Thank your for completing these tasks

*Required

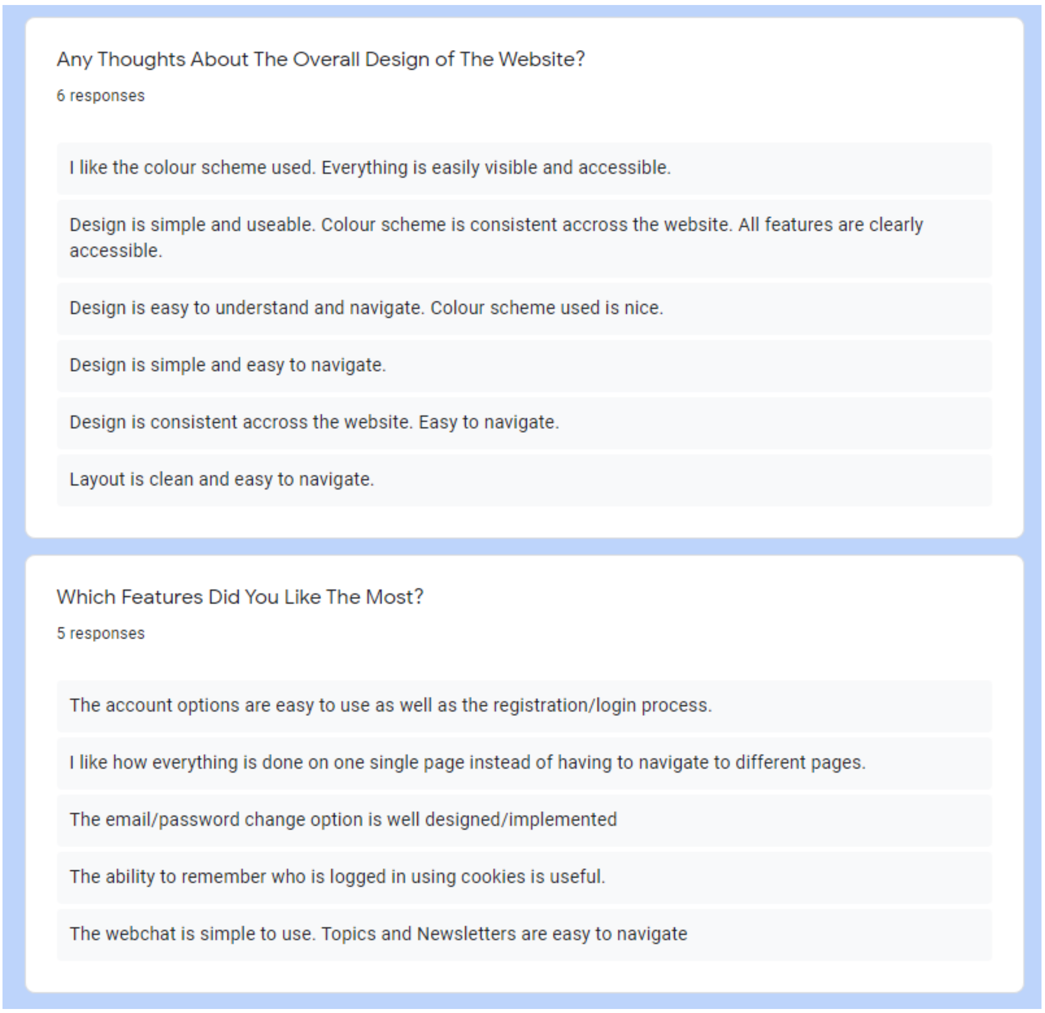
Account Registration Was: *

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

Account Login Was: *

○ Difficult - Needs Work

Figure 8 Usability Questionnaire


Any Thoughts About The Overall Design of The Website?

6 responses

I like the colour scheme used. Everything is easily visible and accessible.

Design is simple and useable. Colour scheme is consistent accross the website. All features are clearly accessible.

Design is easy to understand and navigate. Colour scheme used is nice.

Design is simple and easy to navigate.

Design is consistent accross the website. Easy to navigate.

Layout is clean and easy to navigate.

Which Features Did You Like The Most?

5 responses

The account options are easy to use as well as the registration/login process.

I like how everything is done on one single page instead of having to navigate to different pages.

The email/password change option is well designed/implemented

The ability to remember who is logged in using cookies is useful.

The webchat is simple to use. Topics and Newsletters are easy to navigate

Figure 9 Usability Study Feedback