DECO1800 Zone 1 Work In Progress Report

Elliot Randall

Henry Chladil

Angus Payne

Gary Myles

Alek Thompson

The University of Queensland

School of ITEE

# Project Overview

The project’s files are separately logically across a number of subdirectories to provide a clear and systematic access method. The directory of the site is in “/var/www/htdocs/”, this directory is herein referred to as “/”.

* /
  + Core front end site (both mobile and desktop)
    - All \*.php files are part of the application’s View
* /resources
  + All site resources such as images, style sheets, JavaScript are within this folder
  + /css
    - Style sheets
  + /fonts
    - Custom fonts
  + /images
    - All design related images (not for user uploaded data)
  + /plugins
    - Vendor software (third party libraries)
* /app
  + The site’s core application and programming logic, contained neatly within a Model-View-Controller pattern.
  + /configs/Global\_Config.php
    - Global site configurations (reused code)
  + /controllers/Controller.class.php
    - The controller, responsible for user input and calling View
  + /models/Model.class.php
    - The model, responsible for application logic
  + /models/Mysql\_Connection.class.php
    - A neat, abstracted wrapper for the MySQL connection
  + /tests
    - Test scripts and staging area (not part of application)
  + /vendor
    - Third party plugins and libraries **for testing only** (not part of application)
  + /views/View.class.php
    - The view, extra logic responsible for presenting application output

# Progress Report

The team has been working tirelessly over the past few weeks to get as much of the project completed as possible. Following the steps of iterative design we have endlessly sifted through research, design, prototype and implementation. This document deliberates on what the team has completed to date, a collection of additional research that we have completed in order to increase the overall appeal of the application to a target audience and our future scope for the implementation of the application. Responses and reflections that affect our design and additional design changes will be dissected and discussed throughout this document, giving reason and motivation behind the way our team is developing the application. Our plan for finishing implementation will also be created, paving the footholds for our next actions in the process of design.

The group has discussed the major issues in the design and application as well as its strong points and have decided to focus the remaining implementation to fix the following issues. User flow has been a major part of our recent research; we have spent much time finding solutions to how we can include an improved user involvement in the application. We believe that by including direct goals and a user-friendly navigation, we will be able to allow the user to maximise their time on the site.

After the paper-prototyping phase, it is believed that reductionism and simplification of the user site will contribute positively to the way the user understands the different functions and has the ability to move around the site, ultimately contributing to the user-flow.

Accessibility, and the way we layout and present the content to different users with different abilities and devices. By implementing responsive web design and a mobile version of the site, we believe that we can create an experience that allows any user, anywhere to make the most of our concept.

# Additional Research

Over the past few weeks the team has made a conscious effort to create and bring to life the concept that was pitched in the design document. By moving through the many different stages of design and interaction design, we have been able to prototype, test and further define how we want our concept to function. Congregating and progressing through the design process, we stumbled upon many hidden strengths and faults of our project. Upon discovering these we shaped and formulated ways to mould our concept around the mishaps and fortes of the concept, helping to iteratively move further to the final stages of design. From the first insight and our conscious efforts to formulate the design, e.g. preparation, illumination and incubation, we were able to verify what we needed in the concept through analysis, synthesis, examination and prototyping. Repeating this process allowed us to filter these faults and deliberate a design in which ultimately allows the end user or target audience to seamlessly interact with the application.

Conclusively, we wanted to shape an experience for the user, guide them to information that they want to retrieve and manipulate and provide for ways in which these users will be able to effortlessly interpret the abundance of data located on Trove. We needed to further research into what the user wanted to bring out of the application and how they intend to navigate and interact with the application. After completing most of the design aspects of our project (HTML/CSS), it was found appropriate to look into different criterial matters involving design and the users experience with the application. These were structure, mapping, affordance, orientation, visibility, feedback, error prevention, language and consistency.

Reflecting on the structure of our design, we wanted to find a way to organise content meaningfully, allowing the user to keep flowing through the task that they are performing. With our application that involved keeping the application self-contained. In our design document, it was specified that the app be split over different pages, that being, the search function on one page, and the creation of a History Chain on another. To keep the user integrated and involved with the application, it was found suitable to put the whole application in a page. Much criticism about our structure came from the paper prototype and users finding it hard to know where they are, what they are doing and what they need to do next. The team was also inclined to include a splash page that would allow the user to login/sign up, facilitating the loading and saving of the History Chains. This in itself portrays changes in the mapping and orientation of the application. The main idea behind it being when a user enters the web-application, they will be presented with a splash page and the choice to either login or sign up. However, much like social media sites Tumblr (Yahoo!, 2014) and Twitter (Twitter, 2014), a user will be able to view limited parts of the site with limited functionality. These provide alternate entry points into the site, allowing the user to stay within the site when they have logged in. By doing this, we provide an interactive environment greater than that of what we had planned.



Research from our self-contained app came from sources such as Facebook, and other social media sites, where the main page was an updateable ‘feed’. This itself keeps the user emerged in the surrounding application and creates a sphere of interaction, creating an immersive user experience. In the originally planned design, it was found that after prototyping, that it would be difficult to find a level of interaction with the user navigating between many different pages.

Reflections from the prototyping stage of our concept allowed us to clearly highlight the affordances that the application provides to the user. The greatest comment being that “It was difficult to know what to do and where to do it”. Again this leads us to believe that the application would be suited more appropriately in a self-contained page. By doing this, we will let the user know where they are, where they can go and what they can do. On the first design we had a create chain page, where the user would initially create a ‘Chain’ and then edit it later throughout the site. Many users said that this was misconceiving and were confused as to where to go next. Another solution to this problem was to include helping information at the side of the page as the user moves through the various stages of setting up a chain, though this will most likely be implemented after we have finished implementing the actual chain.

Feedback was another problem that the user faced, when searching, how did they know it was successful or how did they know that they had saved/loaded their chain? These were some underlying functions, though small, but very important, that allow us to communicate with the user and respond to their workings and manipulation of the application. Additionally, this links to error prevention and language that the application provides in response to the users interacting with the multiple functions available. Seeing as the main focus of this concept is human-computer interaction, it is pivotal that the website responds the various tasks and functions the user provides.

Looking through other educational platforms, it seemed that most social-media educational tools were aimed towards that of teenagers, aged 13+. Seeing as we wanted to include a social aspect, it seemed appropriate to change the audience age from primary school children up to teenagers in high school. Another option was to allow ‘sessions’ for the students to collaborate to.

We are also looking to increase the flow of the application, allowing users to subconsciously know how to navigate through the site and use the application with ease; making it not too challenging to use with clear and concise goals that the user will be able to complete. To achieve this single-minded immersion in the task, we need to allow the application to provide clear goals and direct feedback whilst giving the user a sense of control. To drive this influence to the user, we want to make sure that we allow the application to allow discovery. A feature that we have included that induces this is the use of a tagging system, as well as a search for not only Trove articles, but other history chains. We hope to also set up our own ‘bank’ of search results that are ordered by tags, allowing the user to find articles through user/semantic defined searches; articles that the average user wouldn’t be able to find on the basic search portal. We are also striving to create a clear and concise navigation system, one that encourages the user to explore, but not make their immersive experience interrupted by loss of orientation. Our solution being to keep the design simple. This is another point to as of why we decided to change the layout of the website and make the application self-contained in one page. By doing this, the user does not have to worry about navigational issues in regard to the application, allowing the user to *flow* through the usage of the application (Ghani, 1994).

We aim to create the application so that it induces search *serendipity,* regardless of how complex the task at hand is. If a user finds it hard to find sources on Trove, we *want* them to be able to painlessly seek through our database of different user-defined search results that are engraved in the form of a chain. To accompany this ease-of-use, we realised that we needed to implement a sense of reductionism in our application and the way that our content, code and design is organised. We have decided to work through content and structure by reducing the amount of unnecessary text through the implementation of different JQuery functions such as pop-up help buttons. Although subtle, these will help the user to navigate through the site without being confronted by blocks of different text.

Facebook has a very nifty feature that prompts reductionism in design. That being the newsfeed side bar (Facebook, 2014). Normally, that plethora of information would be located on the newsfeed, and would have to have been accessed by the user by scrolling through countless articles or posts that they may have not wanted to see. By reducing the amount of text on screen and compressing each post, it makes it much easier for the user to scroll through different ‘stories’. Our group is looking to implement something of that sort, though not necessarily in the form of a side-scroller. The JQuery accordion or even a simple toggle function will prove to enough to allow the user to navigate effortlessly through the different content on the website. UXMyths has been a helpful tool in the form of a checklist for our website and how we can apply good design principles to maximise a user-computer interaction (UXMyths, 2014).

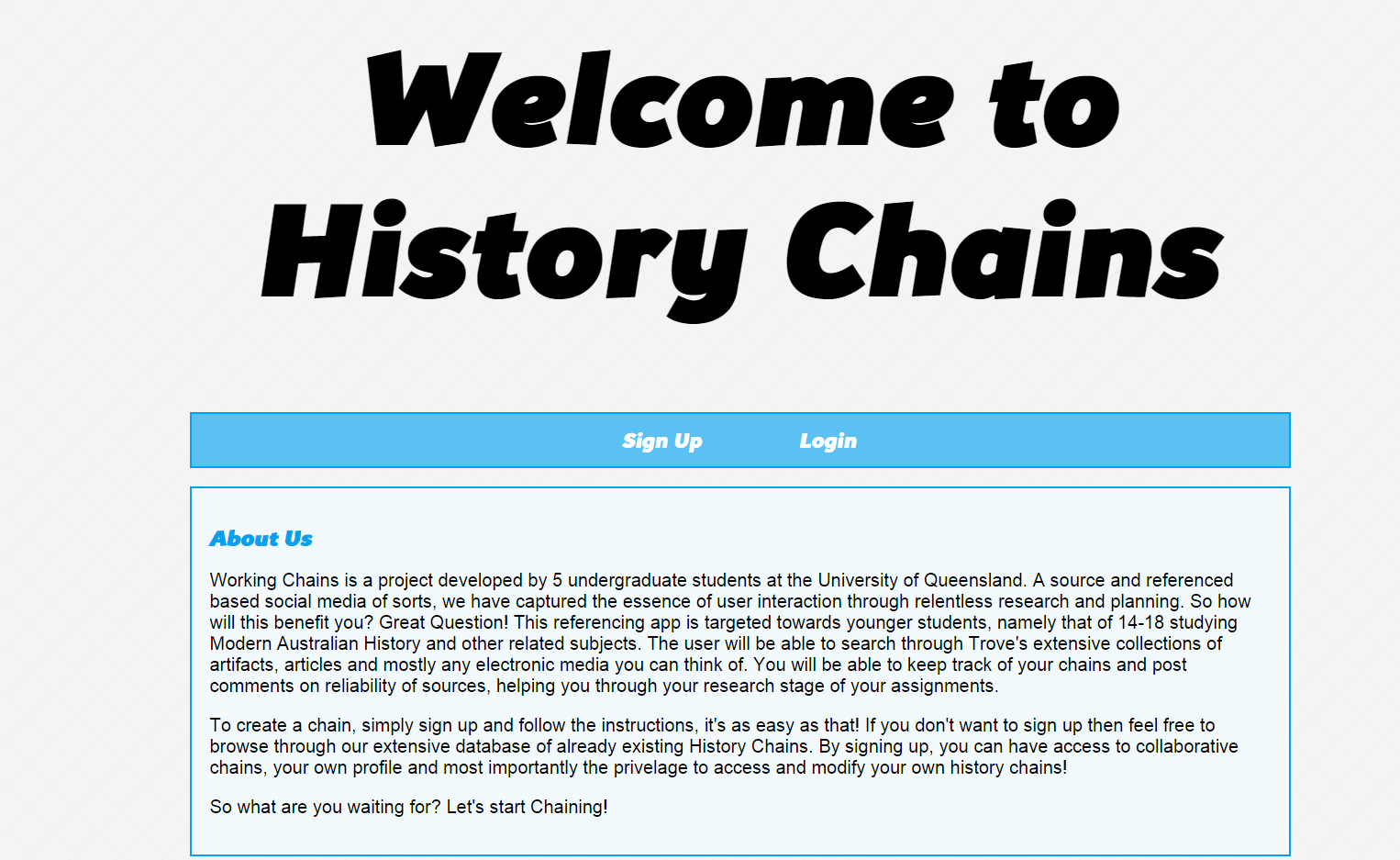
# Work Completed

The History Chains project was originally difficult as so many team members were making changes on the same files, which created numerous conflicts on certain heavily in use files (such as the home page). In a response to this conflict, we decided to use a centralized revision control system. We chose the Git protocol and created a remote repository on the website GitHub ([www.github.com](http://www.github.com)) where our project is now[[1]](#footnote-1). As a result of implementing version control, we are able to provide an accurate “history” of all changes made to the project from the first line written to the project’s current state[[2]](#footnote-2).

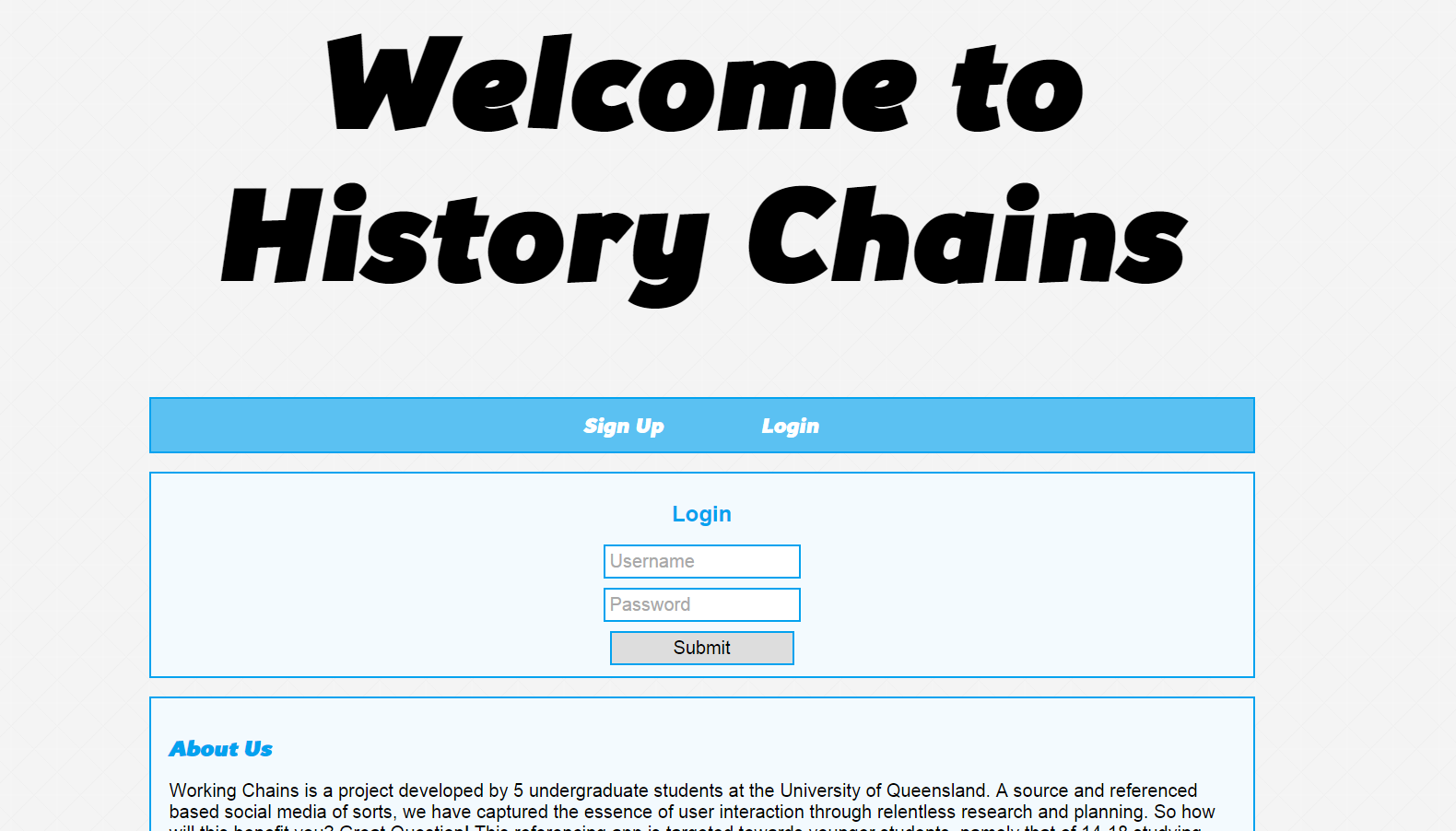
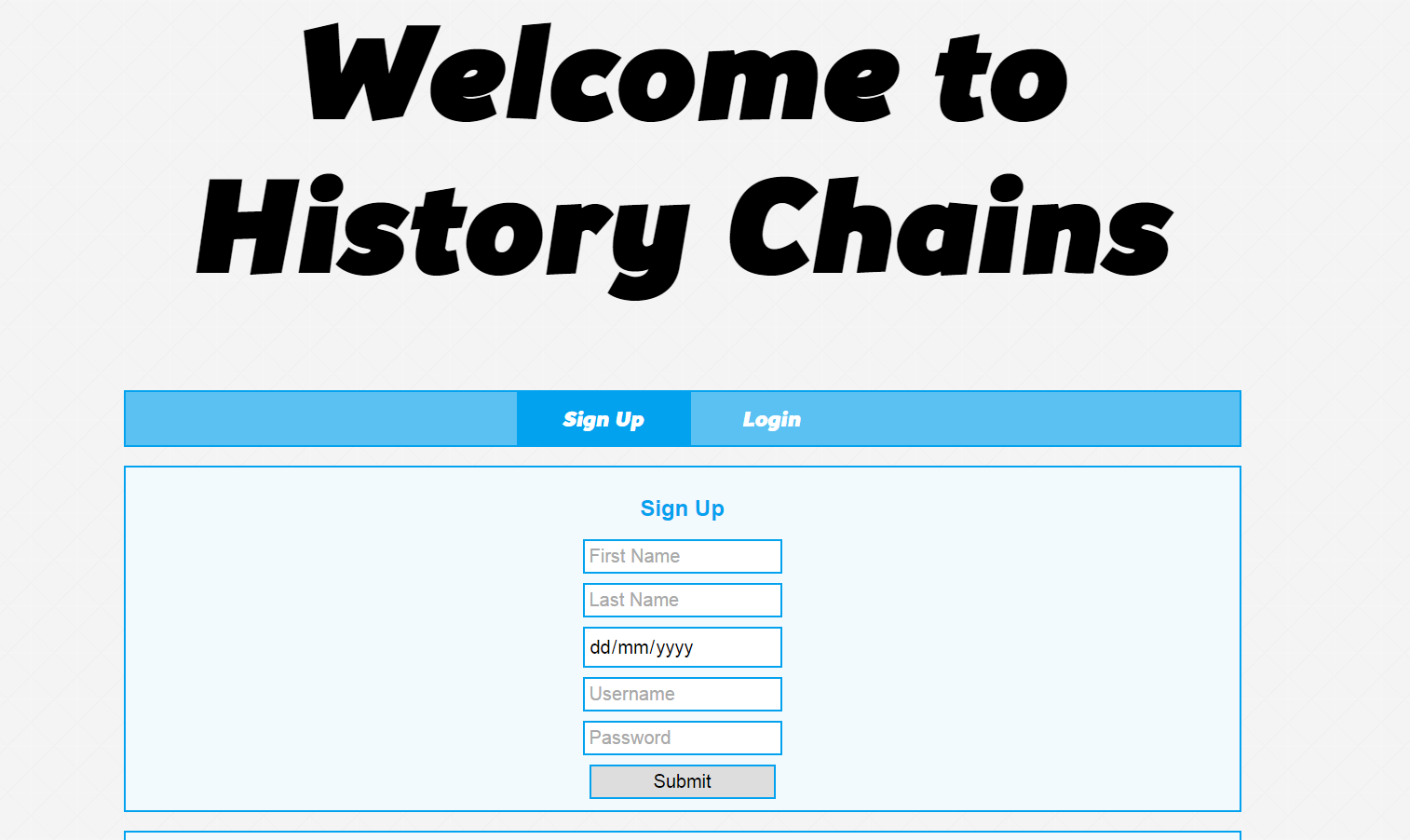
Our elected method of deployment is by manually executing a remote Bash script on the “Zone” server via Secure Shell. This script performs a “git pull” on the remote repository, fetching all changes the group has made and then copying them into our live site.

All of the HTML/CSS has been completed, with all the containers being filled with content ready for use by the user. It must be noted that the design is subject to change to enforce flow and reductionism in the design, allowing the user to be able to make the most of the application. As discussed in the scope for implementation below, some little aspects of the HTML/CSS have to be changed. The website includes 4 main pages and a splash page. This has changed from the original 3 pages, which were not well defined at all. As a team, we have decided to re-vamp the design and include an about page, home page, profile page and create chain page, that being the self-contained application page.

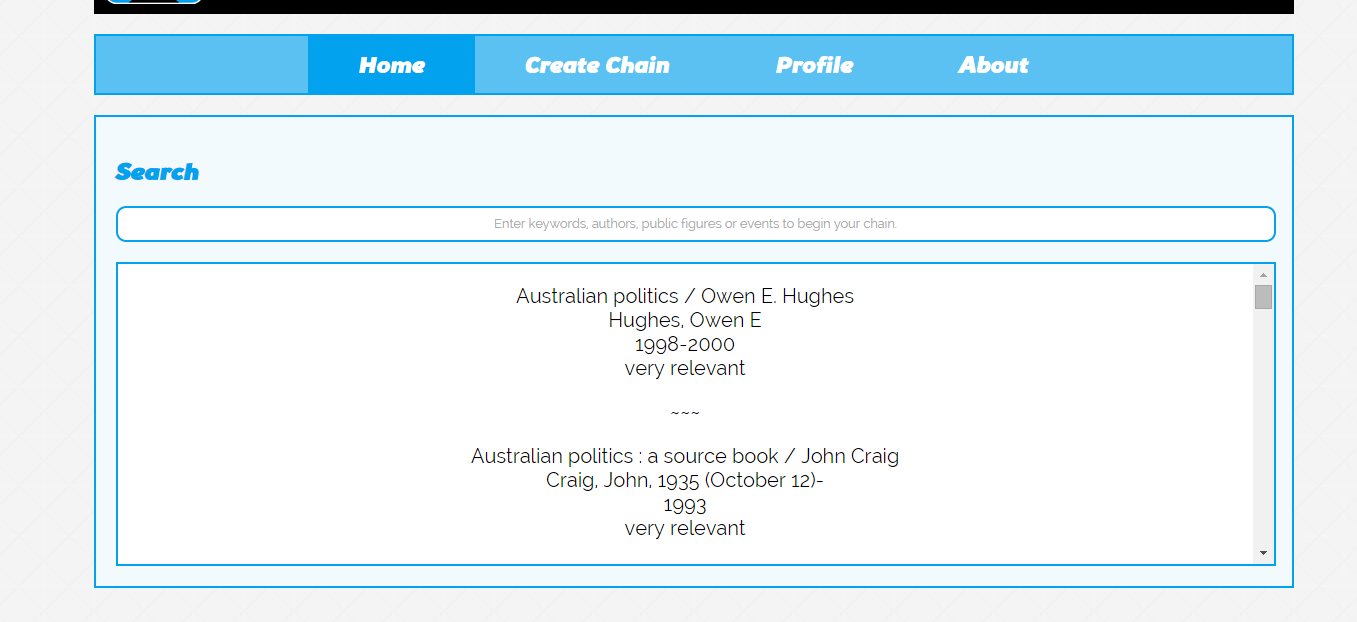
The home and about page are quite static in relation to the rest of the website, with information displayed as a place for the users to be able to orientate themselves with the site and get an idea of how the application works. It must be noted though, that when the user initially enters the website, it they are not logged in, they will be directed to a splash screen, where they have the option to sign up or login.



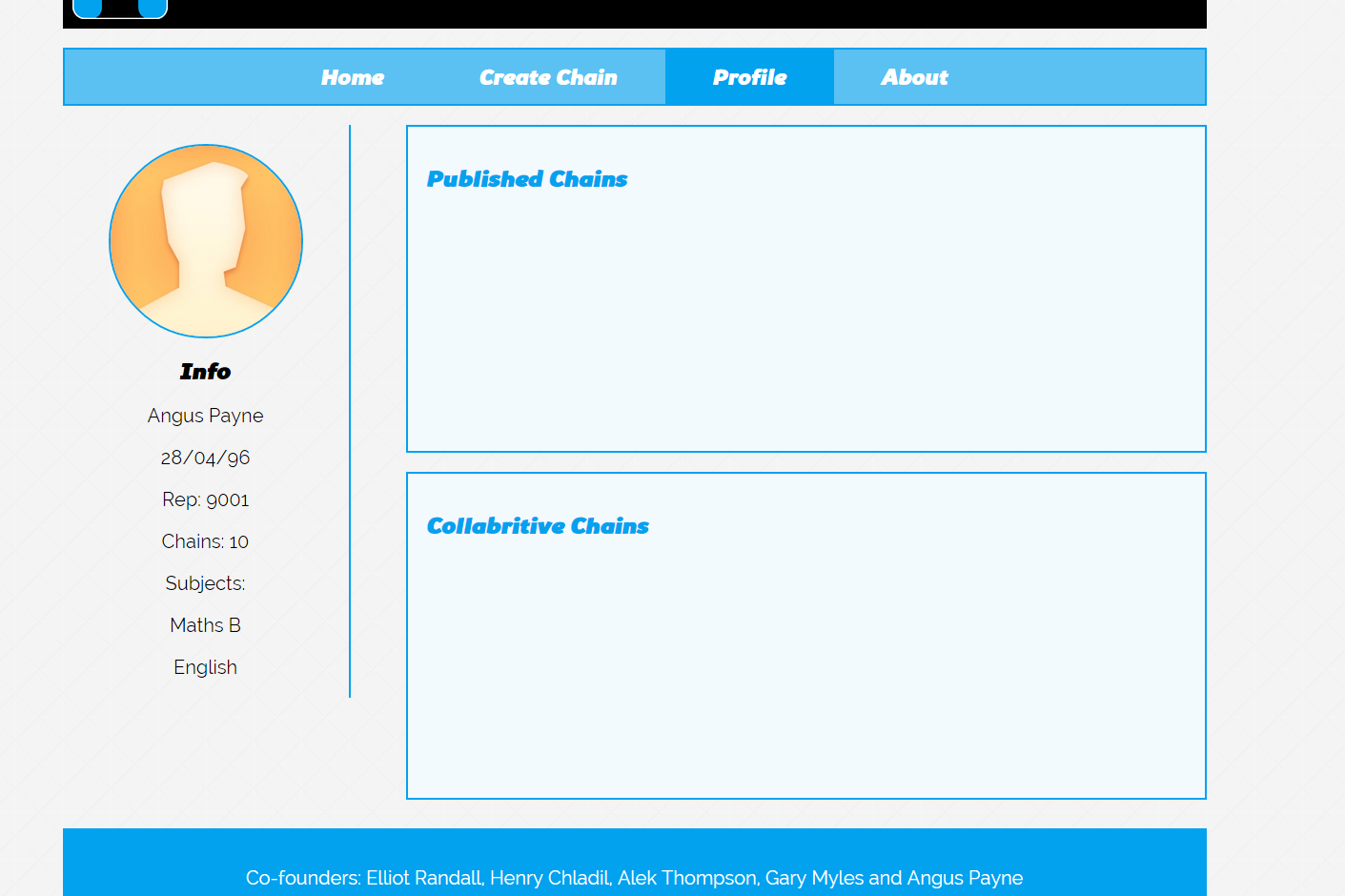
Through JQuery, the user will have the ability to switch between sign up and login. Below are two screenshots of the applications splash screen.



Once the user has logged in/signed up, they will be able to access the site content. That being the home page, the profile page and the create chains page. The user will be able to freely browse through the site content. A search bar on the main home page will allow the user to search the database of different History Chains made by different users. As well as this, the home page includes a default place holder that holds the most-top ranked chain, that being the one with the greatest reputation points. Below is a figure which displays the search results for different chains.

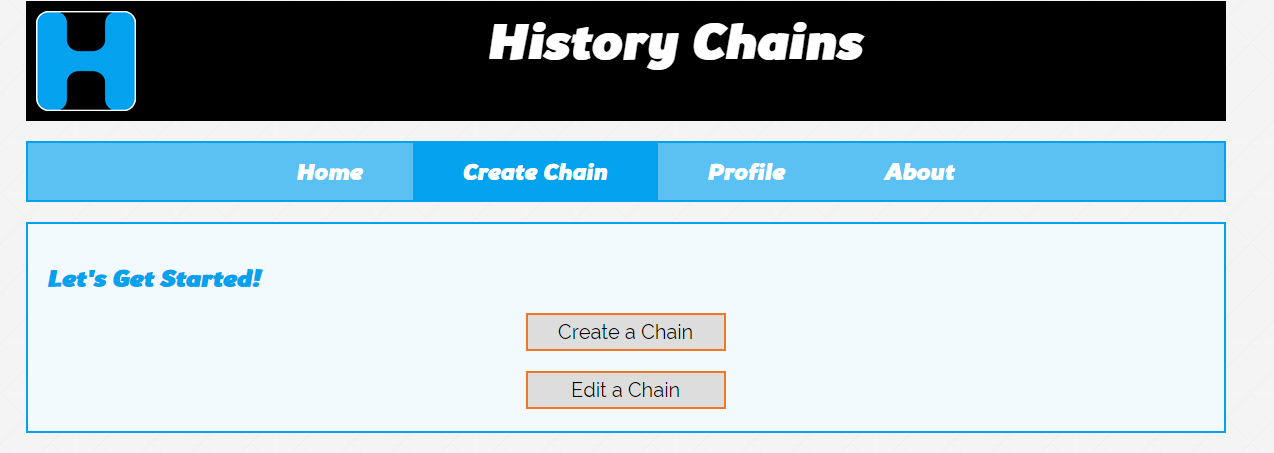


Moving on through the website, the user is also presented with a profile page of sorts, which allows them to show off their different chains and information about themselves. The development of the homepage was done through the research of various design patterns included in the makeup of different social-media profile pages. We decided to use the standard side-bar information profiling system. At the moment, we have not implemented the chain system on the profile, and will be implemented after we complete the minimum requirements of the concept. The profile page stands as seen below.

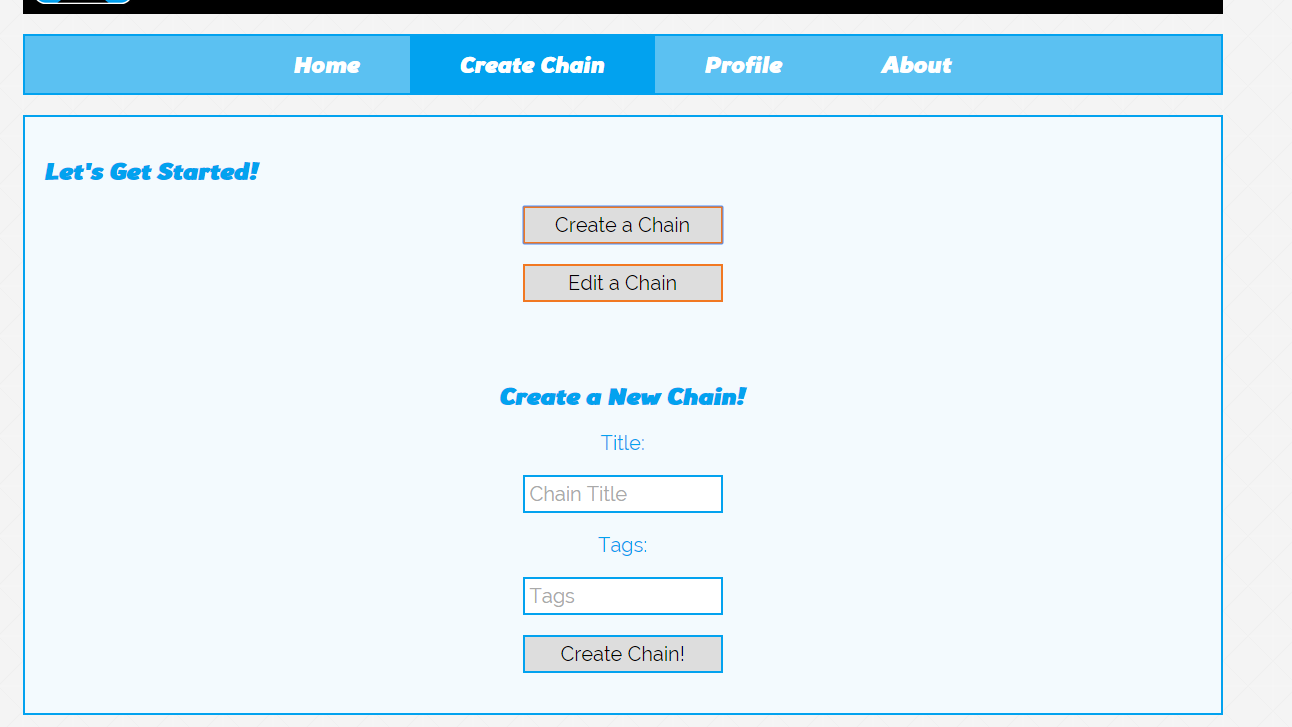


Being quite static, the profile page doesn’t serve much purpose in this stage of the implementation. The plans for future implementation will allow the user to store their various chains, rating and personal information. It is to be used as somewhat as a hub of the application where users can interact and socialize through the object of a chain. Much like other social media outlets such as Flickr where the main social ‘object’ is photos, we will allow users to connect and create social ‘networks’ through the object of a History Chain.

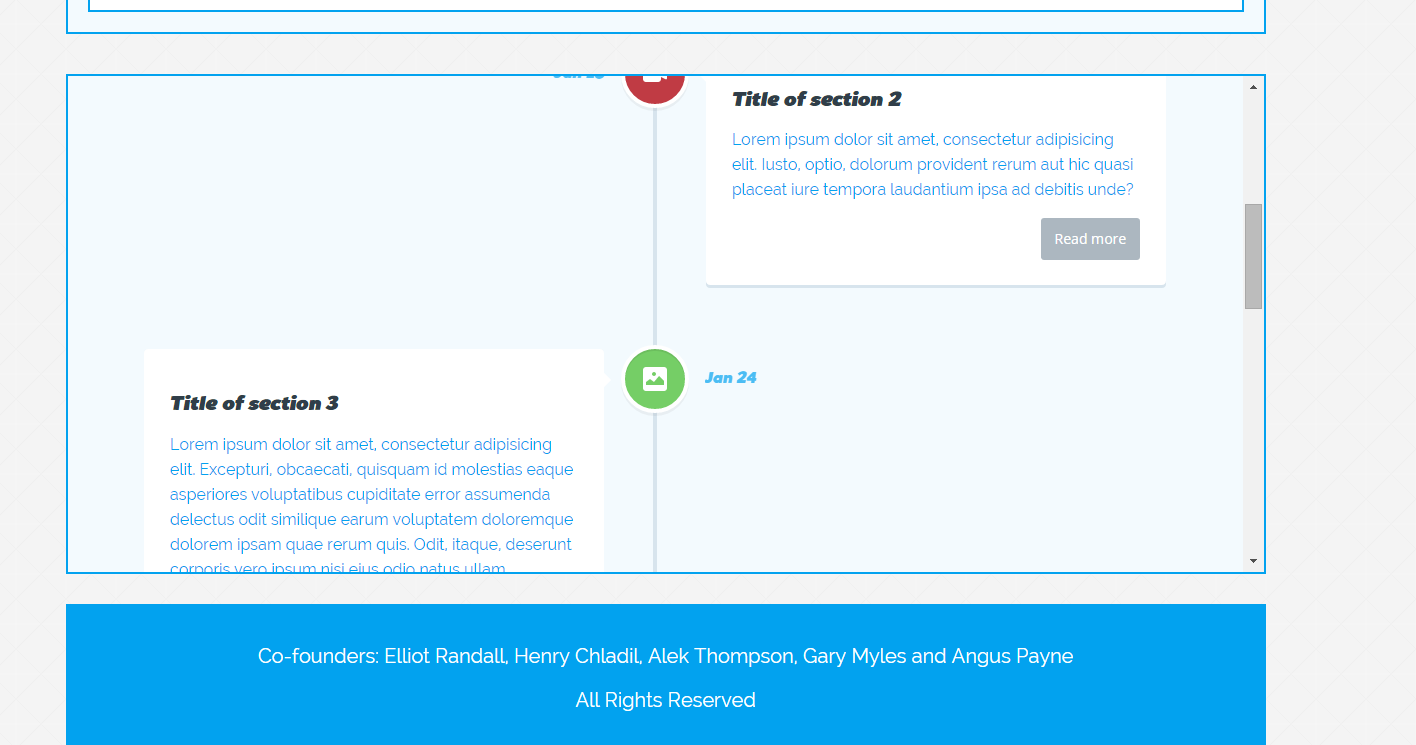
Once the user has set up their profile, they have the ability to start creating and sharing their chains. The first action they will be able to use is to choose between creating a new chain and editing an already made one of theirs.



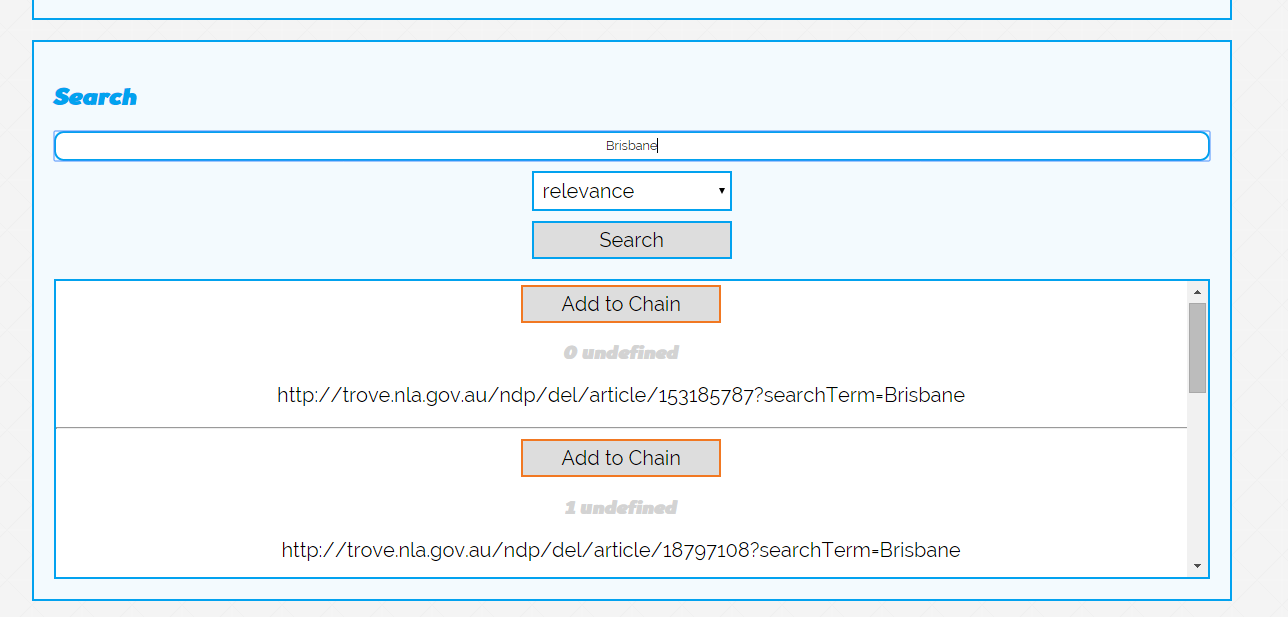
The user will proceed to either create or edit a chain accordingly. When the create chain is clicked, they will have the opportunity to enter information about the chain, for instance, Title, date created etc.



Once having created a chain, or chosen a chain to edit, it will pop up in a div below, and users will be able to enable search and add articles to the chain. The chain itself has been changed from the original horizontal view to a vertical view as depicted below. The timeline itself is a plugin from CodyHouse.



At the moment, placeholder content is being held in each of the source holders, though the general idea is made clear. The read more article is expected to link the user to the source in a pop-up window. This link will not take the user to a different page, but keep them on the same page as to keep the user immersed in the application and keeping them engaged within the flow of the application. Progressing through the page, the user finds that they will be able to search through Trove, this is done in a simple search portal.



Each ‘Add to chain’ button has a unique ‘auto-incrementing’ id that allows us to save the data to the database and access each source that is being used. Even though the implementation of transferring the data to the database hasn’t been completed, we hope to use CQRS and PRG design patterns to make sure that the data that the database receives is of high quality. Seeing as this application is quite complicated with data send and receive, it is important that the database keeps its integrity in every state.

# Mobile Implementation

To date, all the desktop pages have a mobile version. On starting the implementation of the mobile site there were two possible approaches to take. The first being that we make the desktop version of the site a responsive site using CSS media queries. The second was two use JQuery/JavaScript to check whether the page is being accessed by a mobile device or by a desktop device and then directing the user to the appropriate version of the site.

The difference in these two approaches is that with the first one, you are limited to merely customising the content that is on the desktop site for the mobile version, while the second allows you to dictate what content you carry over from the desktop site and what content you make specifically for the mobile site. This is the reason we chose to go with this method as we decided that because of the limitations of a mobile device (specifically screen size) we would rather have complete control over the content rather than playing hide and seek with content from the desktop version.

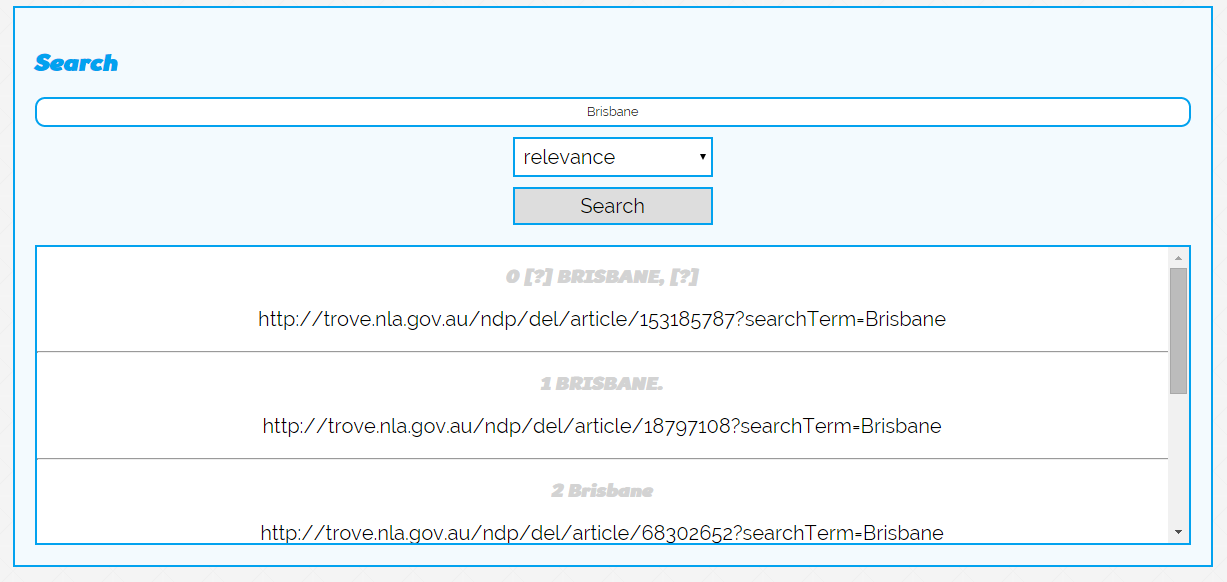
The trade off with this approach though, is that when updating shared content, you have to maintain two versions of the site. However we felt that complete control of the content outweighed having to update the site in two different places.

With this decision we began to plan out the mobile site, stacking the elements vertically rather than horizontally. This is because, mobile applications and mobile web surfing tends to lend itself to scrolling down rather than across. Because of the constraints of the screen size we decided that allowing the user to selectively view the content would make it easier for the user. To do this we have heavily used the JQuery ‘.hide()’, ‘.show()’ and ‘.toggle()’ features.

# Scope for Implementation

The group is now well on the way to finishing the basic implementation of the concept with all the HTML/CSS being completed and the basic search function implemented into the application. The pages that the web-app features is the splash page, the home page, the about page, a profile page and the create chain page. The create chain page is where most of the development has been happening, with the search function and the main object, that being the chain, being displayed in the browser. The database has also been developed, allowing us to get a fast start on the PHP, which is currently still in development. Below are the screenshots of the application as it stands.

Below, figure 1 is a snippet of the search function of the website. At the moment, the search has not been filtered as we are in the development stages trying to include each search in the database. The database design has also been completed and as a result, we are also working on the login function, which will ultimately allow users to access different sources and chains that they have saved in the past. In the progress of the implementation of the database and at this early stage, we have left out tags, though in future we hope to include these. From the perspective of websites such as Flickr, tags can prove to be a very useful for clustering content and allowing user-influence search semantics to create meaningful Meta data, instead of plain search results that are being pulled from Trove. Certainly in the future, the team is willing to implement the feature of tagging through a new table in the database. Seeing as a chain can have many tags, it will be a multivalued attribute to the already existing source table. This will derive into a new table, linking to the source and a tag that has been placed on it. In future we hope to allow the user to search on this database and retrieve items that other users have tagged, and may not be necessarily easy to find on Trove because of their classification system.

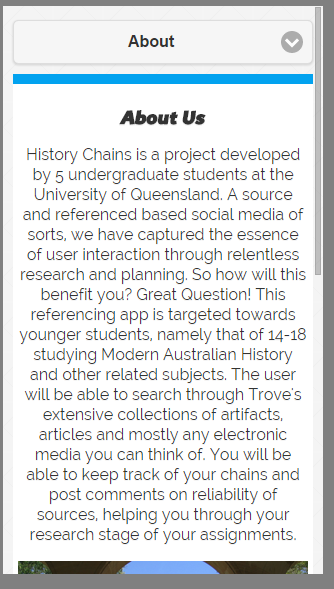


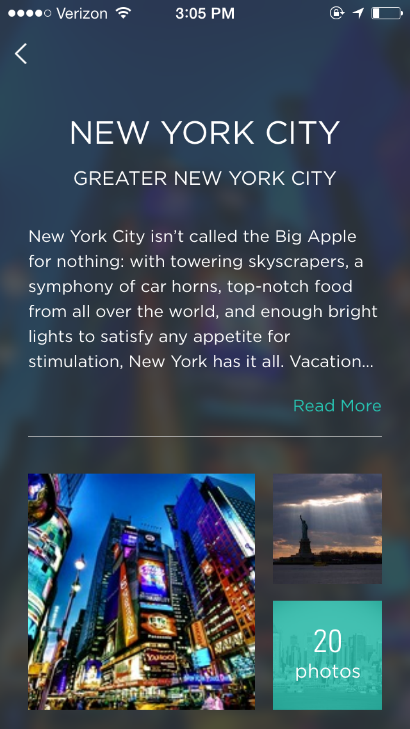
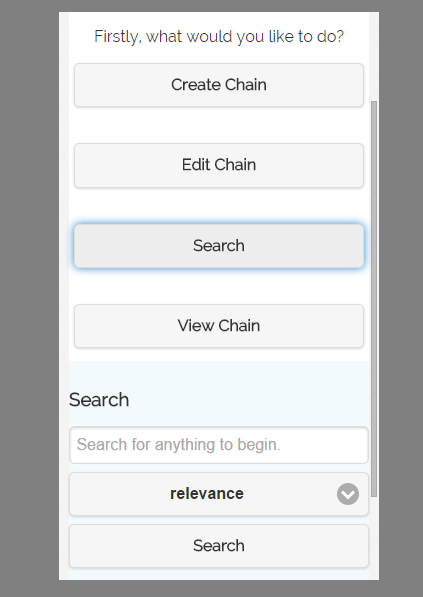
Further discussions on the filtering system for the search we receive from Trove have been talked about, with our search at the moment, retrieving all *types* of results (types meaning newspaper, article etc.) For the moment we are looking to limit the results to articles and books, allowing us to easily create a workable implementation to receive feedback on.

# A Plan for Implementation

Moving forward, the team has set tasks for one another to help speed up the implementation and make the final product as user friendly and workable as possible. The team depends on each member working hard towards completing their task before the given dates and allowing extra time for criticism from other members and the ability to edit their own work, making it as optimal as possible. The main object of the project, being able to create a chain, holds the utmost priority for finishing. The tasks that have been set are as follows:

|  |  |  |
| --- | --- | --- |
| **Task** | **Priority** | **Owner** |
| Home page splash screen | Medium |  |
| Redirection for login | Low |  |
| Home page top chains | Low |  |
| “Create Chain” functionality | Very High |  |
| Adding source to chain | Very High |  |
| Drawing the chain | High |  |
| Saving state of chain | High |  |
| Load an existing chain | Medium |  |
| Edit an existing chain | Medium |  |
| Visually pleasing chain workflow | Low |  |
| Visually appealing search results | Low |  |
| User System | Medium |  |
| Profile page | Low |  |
| Logging In/Out (cookies) | Low |  |
| Rating/Rep System | Low |  |
| Simple tutorial / infographic | Low |  |

We have also been working on a mobile version of the site, which is parallel to the desktop version. Further advancement in this side of the project means that we will be able to make user and device friendly versions for our target audience, who are students. The work on the mobile version of the site is adjacent to that of the desktop version, and we hope to develop both of the mediums at the same time. The work completed on that side of the project to date is the same as the desktop version, with the search working and the HTML/CSS set up correctly. Angus is in charge of the mobile site and coordinates what has to be done in the developmental aspect. So far, most of the mobile application has been implemented and is on par with the desktop version.

PTTRNS.com (PTTRNS, 2014) proves to be a useful source towards the development of our mobile application. With a bunch of different design patterns, including features such as navigation, search, home about pages and much more, we will be able to shape and form the mobile website to the best of standards. Below is a screen cap of a design that seemed to be suitable for our about page on the mobile device. Not only does it reference reductionism in its design, but also incorporates a sense of simplicity to allow the user to generate flow between themselves and the application. Many more designs have been references from PTTRNS.com to help use abide satisfactory web standards.

To the left is the navigational menu implementation of the mobile website. As can be seen, it follows a strict design patter that allows the user to prompt a sense of flow when using the interface. By removing the large walls of text, and applying a step-by-step or goal by goal layout, goal-orientated users will be able to find a sense of achievement when navigating through the different pages. For experiential users, this layout confines to strict navigational design patterns, allowing them to explore the many different features of History Chains with ease, as many other different mobile applications use the same layout.

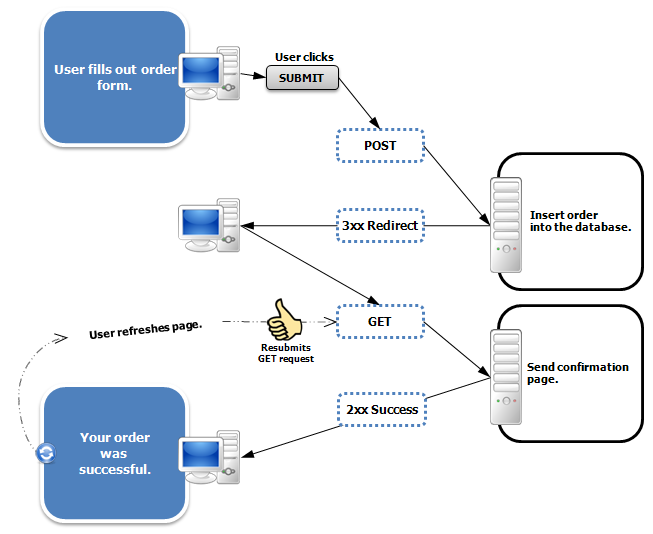
Another source that proved to be useful in the creation of our web-layout was a ‘google developers’ (Google, 2014) article about responsive mobile design and the design patterns used to increase user flow and overall ease of use. The most useful part deliberating about forms and inputs. From this, we found it important to make sure we used existing data or placeholders, included clearly labelled progress bars and provide visual aids for accessibility. This can be copied over to the desktop implementation; again, by using these simple features we will be able to increase the flow of the design and enable flow and a greater user-computer interaction.

# What Issues Will You Confront?

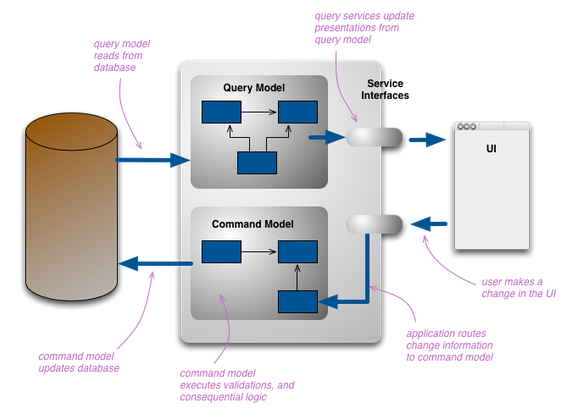
There are a few confronting issues that may prove to be a problem in the ongoing development of our application. The most significant and confronting concern is the idea of transferring the data straight from a search result to an entry on the History Chain. Our first idea was to move the content directly to the Chain through an asynchronous JavaScript call and then move it to the database, instead of using a server-side approach where sending the data is sent to the database first, and then back to the application. Our other solution was a simple logical condition statement to check whether the data has been submitted to the database before every displaying the data to the user.

For improved efficiency, the first option seems a little more viable and easier to implement. By dynamically displaying the data, a quicker response time can be implemented increasing the flow of the user. Other problems that result from this is when two users submit data at the same time which could result in data redundancy in the user table, or the user being impatient and pressing enter more than once, entering the data twice or more in the database. Because there is an auto-increment for the source id and user id, it is possible that the user will sign up twice, seeing as the user ID will always be unique. A way to stop this is too make sure that there is a primary index on the ‘username’ attribute, as well as making sure that it is a unique field.

Slow loading data also proves to be a problem towards our design as we are looking to make sure that we increase the users flow as much as possible. A way to optimize search results is to maximize the search functions and optimize querying. A solution to this is to implement an index on each of the most commonly queried fields. This will in turn will allow binary search over the files and increase read/write time dramatically. Adding a loading gif will also prove to be useful in maintaining user maintenance and increasing direct feedback to the user in order to allow them to move seamlessly throughout the application’s interface.

A simple solution to the problem of a ‘double submission’ is by using the Post/Redirect/Get design pattern, a well-established web development design that stops the double submission or duplicate submissions when a user presses refresh numerous times. In the event of a form submission (HTTP POST request) to the server, a web user that refreshes the webpage can cause the HTTP POST request to resubmit, leading to the problem of a duplicated submission. P/R/G can be used to solve this by adding an additional step to the form submission process whereby after a successful POST request, the user is redirected to another page, where it checks via the HTTP GET method if the query was successful or not. By using the Redirect/Get pattern, if the user refreshes, they will just reload the static page of “success” or “failure” based on the HTTP GET response (Java Code Book, 2014).

The ability to filter results also comes as a challenge for us. In the initial design document of the concept, it became apparent that we wanted to only include articles on Trove that had relevance to the topic of Australian Modern History. Looking through the Trove API documentation, it is seen that there are is an abundance of useful information on how to limit searches, though these mostly deal with limiting dates, search character size etc., the problem still stands in trying to limit these results to a certain topic. Tagged articles can be used in order to solve this, though the relevance and trustworthiness of user-defined tags may be questionable.

The last thing that looks to be a challenge is to retrieve already stored chains, seeing as they use different sources. This is a little more complicated than storing data into a chain because you are only dealing with one search result. The problem arises when you are trying to retrieve many search results to put into a chain. The major issue with this is the speed of retrieval, with another design pattern being of great assistance to its solution. CQRS stands for Command Query Responsibility Segregation. (Fowler, 2014) The approach that our team took to implementing an information system was to use CRUD, though our application needs to be dynamic with fast loading data. The idea behind CQRS is to ‘split’ the conceptual model to suit updateability. The benefit of CQRS is to handle high performance applications, and allows the user to separate the load from read and write.

By breaking the conceptual model into different parts, the database can function independently on reading and writing, ultimately increasing the efficiency of storage and data retrieval. These challenges that we mention can be foreseen and solved through the use of many design patterns. Though it is understood that many other problems are going to arise throughout the rest of the implementation. By implementing a solid database structure with integrity, we will be able to make sure that the data the user is saving and reading is the data that they want to manipulate.

Feedback

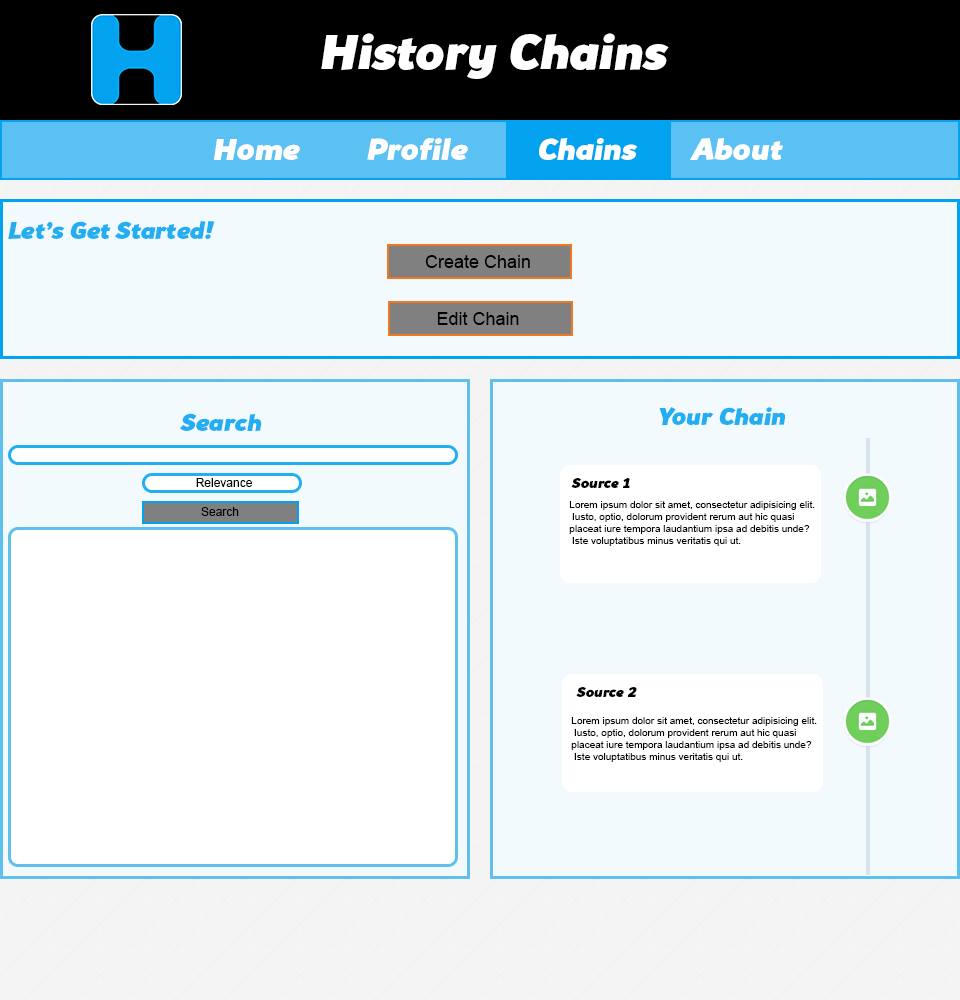
To gain some insight in the direction of our project, we thought it was suitable to ask three questions that would allow us to clear up ambiguities in the development of our project, allowing us to move further forward in the construction of our application. The three questions are as follows;

What information should we insert into the search results for the user so they will be able to retrieve and share useful data?

It was suggested that we include the basic information, that being … as well as an option for the user to view the article in a self-contained window. This way they can see what they are appending to their chain.

Do we have a clear, concise and logical layout for the application so that the user has the ability to move through each page, induced by flow?

This is the question that the group had been asking for a long time; how well has out layout been portrayed. Lorna and the tutors suggested to us that we change the design of the application screen. The suggestion being that we locate both the chain editor div next to the search div. This also suits the vertical timeline and allows the user to see their chain update in real time. When re-designing the new layout of the page, it was clearly evident that this would simplify how the user view and interact with the timeline. The functionality of it hasn’t changed, though the layout has. Below is a mock-up of the new layout.



Do you suggest we implement the different patterns we mentioned, e.g. PRG and CQRS to combat data duplication problems?

The group has been researching and sifting through different design patterns and implementing a few, the main one being MVC (Model-View-Controller). We are also able to foresee issues with data retrieval and writing and look to combat data duplication in order to keep the data of a high integrity. We also found that a lot of the data retrieval from the database was slow, and after a discussion with a tutor, we found it appropriate to implement indexes for the faster retrieval of data.

Conclusion

As a group, we feel that we have achieved a great step forward in the production and completion of our concept. A large proportion of the application is already working, so we look to scope our future implementation down to useability, design and extra nifty little add-ons. With the implementation laid out, we are able to organise the procedure in which we look to implement the concept and finish it in an organised manner.

# References

Facebook. (2014). *Facebook*. Retrieved from www.facebook.com

Fowler, M. (2014). *CQRS*. Retrieved from CQRS Pattern: http://martinfowler.com/bliki/CQRS.html

Ghani, J. (1994). *Flow in Computer Human Interaction: Test Model.* New Jersey : Ablex Publishing Corps.

Google. (2014). *Google Developers*. Retrieved from Design Efficient Forms: https://developers.google.com/web/fundamentals/input/form/design-efficient-forms

Java Code Book. (2014). *Post Redirect Get*. Retrieved from Java Code Block: http://www.javacodebook.com/2013/08/20/post-redirect-get-pattern-in-spring-mvc/

PTTRNS. (2014). *Mobile Patterns*. Retrieved from www.PTTRNS.com

Twitter. (2014). *Twitter*. Retrieved from www.twitter.com

UXMyths. (2014). *UXMyths*. Retrieved from http://uxmyths.com/

Yahoo! (2014). *Tumblr.com*. Retrieved from Tumblr: www.tumblr.com

1. [github.com/henrycjc/historychains/](https://github.com/henrycjc/historychains/) [↑](#footnote-ref-1)
2. [github.com/henrycjc/historychains/commits/master](https://github.com/henrycjc/historychains/commits/master) [↑](#footnote-ref-2)