YARN SWAP

Yarn Swap
Progressive Web Application
Final Report

Name:	Fiona Waters
Student No:	20095357
Programme:	Higher Diploma in Computer Science
Module:	Project
Project Supervisor:	Richard Lacey
Submission Date:	2nd April 2023

Declaration

I declare that the work which follows is my own, and that any quotations from any sources (e.g. books, journals, the internet) are clearly identified as such by the use of ‘single quotation marks,’ for shorter excerpt and identified italics for longer quotations. All quotations and paraphrases are accompanied by (date, author) in the text and a fuller citation is the bibliography. I have not submitted the work represented in this report in any other course of study leading to an academic award.

Signature of Student:

Fiona Waters

Date 02/04/2023

Acknowledgments

The Higher Diploma in Computer Science is likely one of the most difficult things that I have undertaken. There are many people that have made it possible for me to get to this point and for this I am extremely thankful. It takes a village.

First and foremost, I would like to thank my friends and family, my daughters Juno and Willow and most especially my husband Ciarán; I would not have been able to complete this program or project if it weren't for their unwavering support, enthusiasm and understanding.

To my colleagues at Red Hat for their understanding and support while I completed work placement and project simultaneously, particularly Brendan O'Farrell, Paul McCarthy, and Carl Kyrillos.

To my project supervisor Richard Lacey for sharing his valuable experience and helpful advice.

To the lecturers and team that run the Higher Diploma in Computer Science at SETU, as well as my fellow students; this course is a difficult one, but the outcome has already been rewarding, of which I am very appreciative.

Titles

Commercial Title

Yarn Swap

Academic Title

Progressive Web App Community Swap Platform (Yarn Swap)

Preface

This report should be read together with the following resources:

Project Planning - Jira Board:

https://yarn-swap-project.atlassian.net/jira/software/projects/YS/boards/1

Github Project Repositories: YarnSwap (Back End) and YarnSwapFE (Front End), as submodules of yarnswap-project:

https://github.com/Fiona-Waters/yarnswap-project

*Deployed Yarn Swap Backend:

http://yarnswap-yarn-swap.apps.fwaters.uw4y.s1.devshift.org/

*Deployed Yarn Swap Frontend:

http://yarnswap-fe-yarn-swap.apps.fwaters.uw4y.s1.devshift.org/

* The OpenShift Cluster where the project is deployed will not be running constantly but will be shown in the project video and presentation.

Abstract

The aim of this project is to create a place where crafters can share yarn with others that may otherwise have been left unused. This serves to reduce waste while also fostering a sharing community that is focused on using what has already been produced and contributing to a circular economy. This idea can be transferred to many other areas.

Yarn Swap is a progressive web application with a React/Vite/ChakraUI frontend, a Golang/Gin backend which employs Firebase Realtime Database for the storing of data. The project utilises elements of continuous integration and continuous deployment and has been deployed on Red Hat OpenShift.

Table of Contents

Table of Contents

Table of Figures

1 Introduction

1.1 Background

Not long after my first child was born, I started to crochet, then knit, and soon became obsessed with yarn and colour. Because of this I have a lot of yarn, I wish I could use it all but unfortunately, I have acquired too much, this is sometimes known as **S**tash **A**cquired **B**eyond **L**ife **E**xpectancy (SABLE).

As a crafter, I have spent a lot on yarn over the past number of years. At the time of purchase, I almost always have a plan, an idea of what I intend to use the yarn for. Just recently I crocheted a shawl with yarn purchased for that purpose in 2015, 8 years ago. But often it doesn't work out this way, and yarn is left untouched in my stash and may never be used. As well as this, it is almost certain that I will buy more yarn with the promise of new projects, and this goes on and on. The thrill of a new project and some lovely new yarn outweighs the short sadness felt for the yarn left unused. You can see how this could be considered a problem.

1.2 Problem Definition

Anyone who enjoys crafting, be it knitting, sewing, painting etc., will spend a lot on supplies. Sometimes these supplies are used immediately but other times they will languish in the craft drawer, sometimes never being used. For this project, yarn will be the subject we are most concerned with. Yarn is used for knitting, crochet, macrame, weaving and many other crafts. The following idea could be applied to any craft supplies, fabric, paint, beads etc. The solution is versatile and will even lend itself to other industries. For now, the focus will be on yarn.

There are a lot of knitters and crocheters and there are lots of businesses that sell different types of yarns. As a knitter, is it very difficult to be near a yarn shop and not buy yarn. As well as that, there is a large online community of pattern creators and online yarn shops. These patterns are so beautifully photographed and marketed that you can't help but want to make the items immediately, this causes you to purchase the pattern and the yarn quite soon after you see the first glimpse of the pattern. The pattern and yarn arrive the following week, but you don't have time to start working on it straight away because you already have 4 works in progress (WIP), a full-time job, a family and other priorities. So, the yarn gets safely stored away with all your other yarn (in your stash) to be taken out when you have time.

It will likely take some time for you to finish your other WIP's and by the time you do (if you do), you will probably have seen at least several other patterns that you want to make, and you may have bought some yarn for some of them also. You will also sometimes buy yarn with no project in mind because you just must have it and why not! Eventually you end up with an amount of yarn larger than you will ever have time to knit, and maybe your taste will change, and you will have yarn that you no longer like or want to use.

1.3 Solution

What has been described above is not unusual, most crafters, will have yarn that they know they will never use. The aim is to create a place where crafters can list and view yarns and can swap or procure yarns from others that could otherwise be forgotten or wasted.

This is akin to a clothing swap, or a book swap, allowing people to reuse and make use of items already available rather than providing the need for more to be created or produced. This could play a small part in a circular economy.

The proposed solution is a Progressive Web App – an application that is available to use in the browser but can also be installed on a mobile phone or computer.

1.4 Project Goal

The goal of this project was to create a progressive web application and CI/CD pipeline.

The web application provides a place for crafters to create an inventory of their yarn, with descriptive listings and images. This, in the first instance, allows them to realise or remember what yarn is in their possession. They can then mark items as available to swap, and these yarns will be available for others to view. Swapping is the main function of the proposed web application. A user can earn tokens for yarn they have uploaded and can use these tokens to acquire yarn from other users.

2 Research and Analysis

2.1 Requirements analysis

When deciding what tools and technologies to use for this project, it made sense to use those which I am learning at my work placement where possible. This would allow further upskilling and progression. Therefore, the technologies chosen include a mix of those used on work placement and those learned in other modules on this course.

Figure 1 – Planned Technologies

Front End	**Back End**	**Pipeline + Deployment**
React	Golang	Jenkins
ChakraUI	Gin Web Framework	Docker
	Firebase (Auth + Database)	Kubernetes
		Red Hat OpenShift

Figure 2 – Technologies Used

Front End	**Back End**	**CI/CD + Deployment**
React	Golang	Quay.io Github Actions
ChakraUI	Gin Web Framework	Docker
	Firebase (Auth + Database + Storage)	Red Hat OpenShift

2.1.1 Front End

React

React is a JavaScript library for creating user interfaces. It is one of the most widely used technologies for front end development. According to Javatpoint (2011), it is easy to learn and use, and makes creating dynamic web applications easier. One of its main advantages is the use of components, which are reusable throughout an application.

Other front-end technologies include Angular, Vue and Svelte. There are pros and cons for each of these. React is widely used and therefore there is a wealth of useful documentation and information available. This makes it a good choice for front end development and for Yarn Swap.

Chakra UI

There are many component libraries that can be used with React, and ChakraUI is one of these. Many are straightforward and quite accessible. Having used other libraries including Material UI, and Semantic and Fomantic UI during the course, I felt confident to research another option and decided on Chakra UI. Chakra UI has several components that will work well for the application, and they are all well documented.

2.1.2 Back End

Golang

Golang is an open-source programming language which was designed at Google.

At my work placement, my team uses Golang. It is an interesting language and continuing to learn and gain more experience in it, by using it for this project, has allowed me to upskill in the language in new ways, while also learning more about its basic syntax.

Gin Web Framework

While researching the use of Golang to create backend API's, the Gin Web Framework came to my attention, and a related tutorial that would help with learning in this area. This framework allows you to create a restful API using Golang.

Firebase

Firebase offers many products that are useful to software developers and engineers. Firebase Authentication, Firebase Realtime Database and Firebase Storage have been used for this project. Firebase authentication allows users to be authenticated when they sign up and log in to the application. Firebase Realtime Database allows the relevant application data to be stored and linked to its user, and Firebase Storage is used to store listing images.

Firebase documentation is quite comprehensive and easy to follow, and the resulting actions and processes work smoothly and efficiently.

Project Data

All data used in, and stored for, this project is test/dummy data created by me. No real data has been used.

2.1.3 CI/CD

“A continuous delivery pipeline is an automated expression of your process for getting software from version control right through to your users and customers.” (Jenkins.io, 2023)

There are many technologies available that assist with the continuous integration and deployment of an application. The intention here, was for the project to be deployed via a CI/CD (Continuous Integration/Continuous Deployment) Pipeline using Jenkins but instead another route was taken using technologies such as Docker, Quay.io, and GitHub Webhooks.

2.2 Market analysis

Following research, there were no apparent web applications or sites that offer the swapping or exchange of yarn. The closest was a site called Woolswap (https://woolswap.com.au/) which arranges swapping of yarn parcels from yarn lovers around the world. In this case, those that sign up do not choose the yarn, but instead get a surprise parcel from their swap partner.

Similar swap related sites were found but in other areas, including fashion, and literature. Two of these were:

Nuw

https://www.thenuwardrobe.com/

Nuw is a mobile application that allows users to swap items of clothing and other fashion items. It is inspiring and has an extremely attractive user interface and design and an appealing ethos. Its use seems clear, simple, and quite practical, attributes that are very important.

The app allows users to sign up and earn a coin for each item they upload; they then use these coins to request items from other users and pay a small fee to the site on top of this for each transaction.

Bookswap

www.bookswap.co.uk

Bookswap is a web application that allows users to swap books. The users earn points by offering books and can use these to acquire books that others are offering. If you do not have enough points, you can purchase some for a small fee.

Github

A search was conducted on Github for open-source projects that may be offering the same functionality, none relating to yarn were found. There were some book swap repositories written in several different languages including python and PHP, as well as a swap for college textbooks written in JavaScript. There were also some clothing swap repositories which mostly focused on connecting people living in the same area to swap clothes with each other.

The swap model is quite versatile and could be utilised for any item. While this project is focusing on yarn, the application could be altered for any item that can be reused or swapped.

2.3 Feature analysis

Figure 3 below shows the comparison of features on the applications discussed in the market analysis section 3.2 above and the features proposed for Yarn Swap.

Figure 3 - Feature Comparison – Other Swap Web Applications

	Upload Items	Browse Listings	Earn points	User ratings	Wishlist	Member chat
Bookswap	✓	✓	✓	✓	✓	X
Nuw	✓	✓	✓	X	✓	✓
YarnSwap	✓	✓	✓	X	✓	Forum

As you can see, the applications reviewed have a lot of the same features including the uploading of items, the ability to browse listings, and to earn points/coins/tokens. Some allowed for users to be rated and some allowed for members to chat. The following are proposed features for the Yarn Swap - Progressive Web Application:

- Upload items - a user will upload yarn listings.
- Browse listings - a user can browse listings.
- Earn points - a user will earn points/tokens when they add a listing and mark it as swappable, they will then use these to request yarn from other users.
- Wishlist - a user will be able to create a Wishlist, which will contain items they hope to obtain.
- Forum - a user will be able to start a topic and ask a question or for advice on the forum. A user will also be able to contribute to other users' topics.

3 Modelling

3.1 Back End and Database

3.1.1 Data Modelling

Data for Yarn Swap is stored in Firebase Realtime Database. Firebase Realtime Database is a NoSQL database.

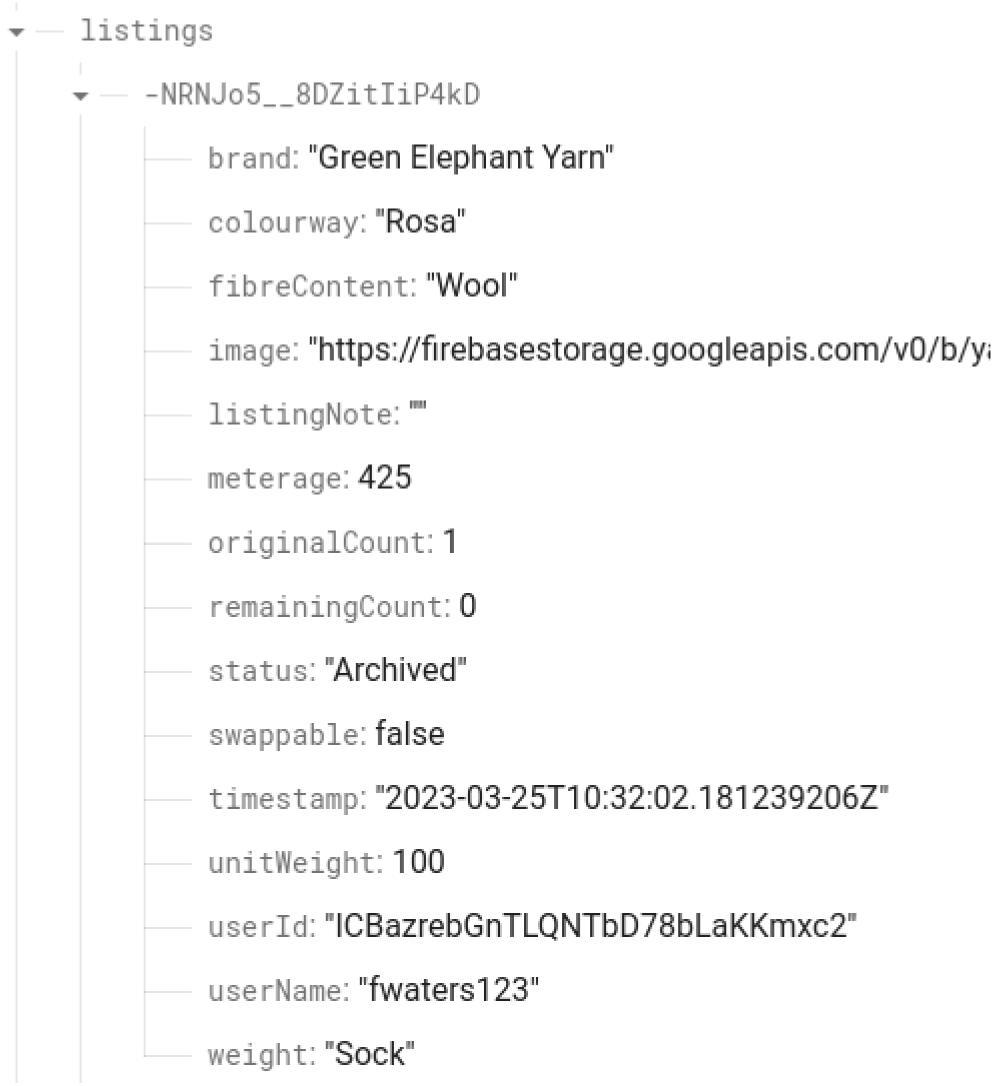
“NoSQL databases are non-tabular databases and store data differently than relational tables. They come in a variety of types based on their data model. The main types are document, key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of data and high user loads.” (MongoDB, 2019).

“NoSQL databases are used in real-time web applications and big data, because their main advantages are high scalability and high availability. NoSQL databases are also the preferred choice of developers, as they naturally lend themselves to an agile development paradigm by rapidly adapting to changing requirements”. (Oracle.com, 2022)

Yarn Swap requires a database, which allows for uncomplicated and unchallenging processes to allow for change wherever necessary. The documentation for using Firebase Realtime Database is quite extensive and covers many use cases quite effectively - this has been of benefit in terms of time constraints for this project. Firebase Realtime Database has been used for assignments in other modules and so can attest to its ease of use, making the decision to use it easier.

When storing data in Firebase Realtime Database, “The data is stored as JSON objects, there are no tables or records - when you add data to the JSON tree, it becomes a node in the existing JSON structure with an associated key.” (Firebase 2019).

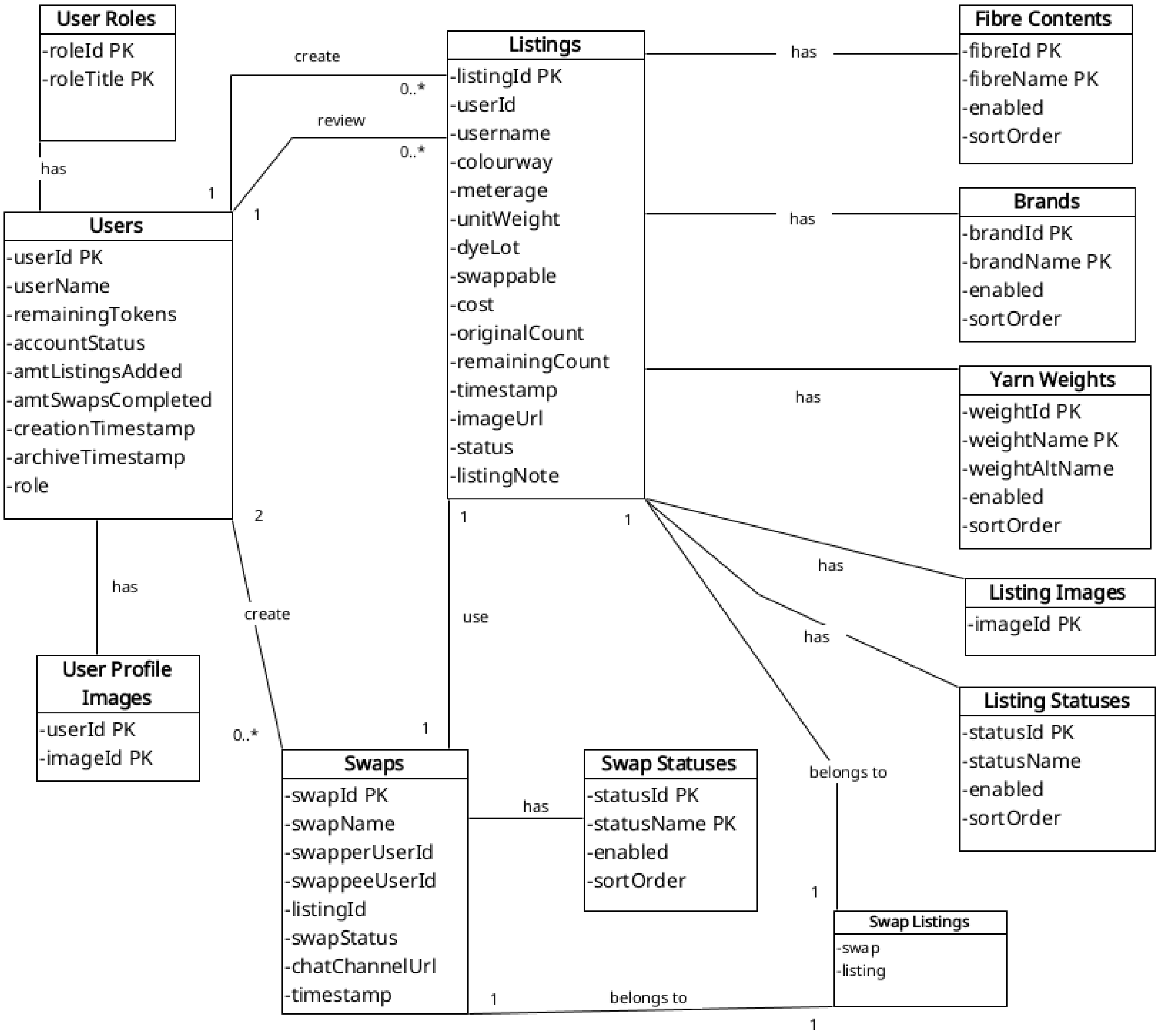
Figure 4 - JSON Structure Example - Yarn Swap Listing



```
▾ — listings
    ▾ — -NRNJo5__8DZitIiP4kD
          — brand: "Green Elephant Yarn"
          — colourway: "Rosa"
          — fibreContent: "Wool"
          — image: "https://firebasestorage.googleapis.com/v0/b/y
          — listingNote: ""
          — meterage: 425
          — originalCount: 1
          — remainingCount: 0
          — status: "Archived"
          — swappable: false
          — timestamp: "2023-03-25T10:32:02.181239206Z"
          — unitWeight: 100
          — userId: "ICBazrebGnTLQNTbD78bLaKKmxc2"
          — userName: "fwaters123"
          — weight: "Sock"
```

Although this Database does not employ the relational database paradigm, illustrating the data model using a traditional Entity Relationship (ER) diagram and mapping these details can be useful to allow for scope of modelling the data.

Figure 5 - ER Diagram showing attributes and relationships.



4 Planning

4.1 Agile

An agile methodology is one where a project is broken down into smaller chunks and worked on incrementally rather than all in one go. This allows things to be changed and updated as needed along the way to end up with a more sculpted result that meets the project requirements more precisely. This result may not be exactly what was planned at the beginning as requirements may have changed over the course of the project. Re-evaluation and reflection on the project requirements and implementation are needed at the end of each sprint - this can be done by way of a retrospective look back on how the sprint went, what worked well and what could have been improved upon. This methodology is well suited to any software development project including the one at hand.

4.2 Tools

There are many tools that aid with planning a project using an agile methodology. Having used two of these such tools previously - Trello and Jira, the decision was made to use Jira. Its organisation of tickets and ability to create a backlog and move these into the relevant sprint when ready to work on them, is useful to this type of project. The Kanban style of moving tickets across the board as they progress allows you to easily see the status of each task, which helps with time management and organisation for the following sprint and tasks.

There are many new things to learn when working on a project on an individual level, which will help to improve my use and understanding of it for any future use on personal or work-based projects.

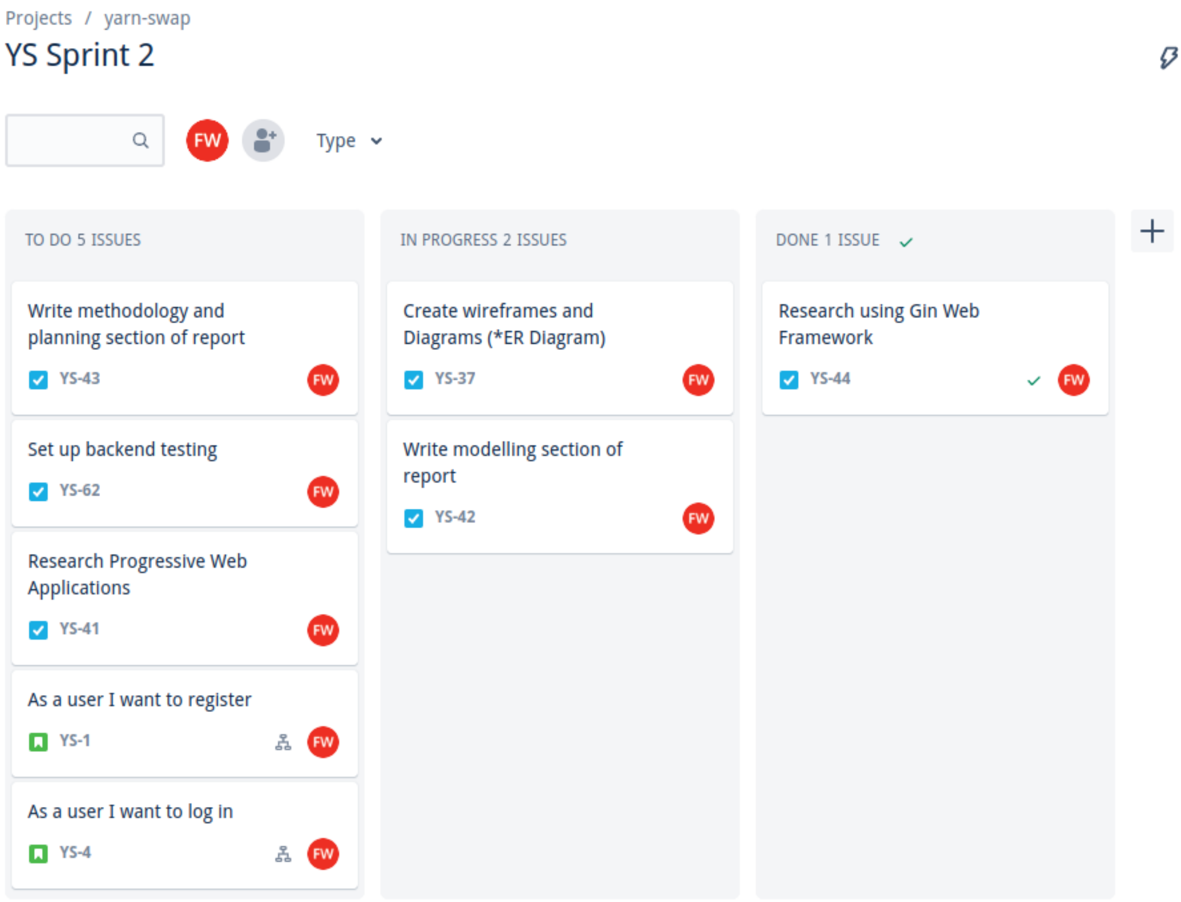
4.3 Getting Started

To create and populate tickets on the Jira board, the work had to be broken down into smaller parts. This was completed by creating user stories.

“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.” (Rehkoph, 2019)

Each user story was added to a ticket in the Jira project, with 2-week sprints planned. The first sprint was populated with tickets; This allows re-evaluation at the end of each sprint before populating items in the next. Below are some images from the Jira project. More information has been added at Appendix C.

Figure 6 - Jira Board, Yarn Swap Sprint 2



Projects / yarn-swap

YS Sprint 2

FW Type

TO DO 5 ISSUES

Write methodology and planning section of report
☑ YS-43 FW

Set up backend testing
☑ YS-62 FW

Research Progressive Web Applications
☑ YS-41 FW

As a user I want to register
YS-1 FW

As a user I want to log in
YS-4 FW

IN PROGRESS 2 ISSUES

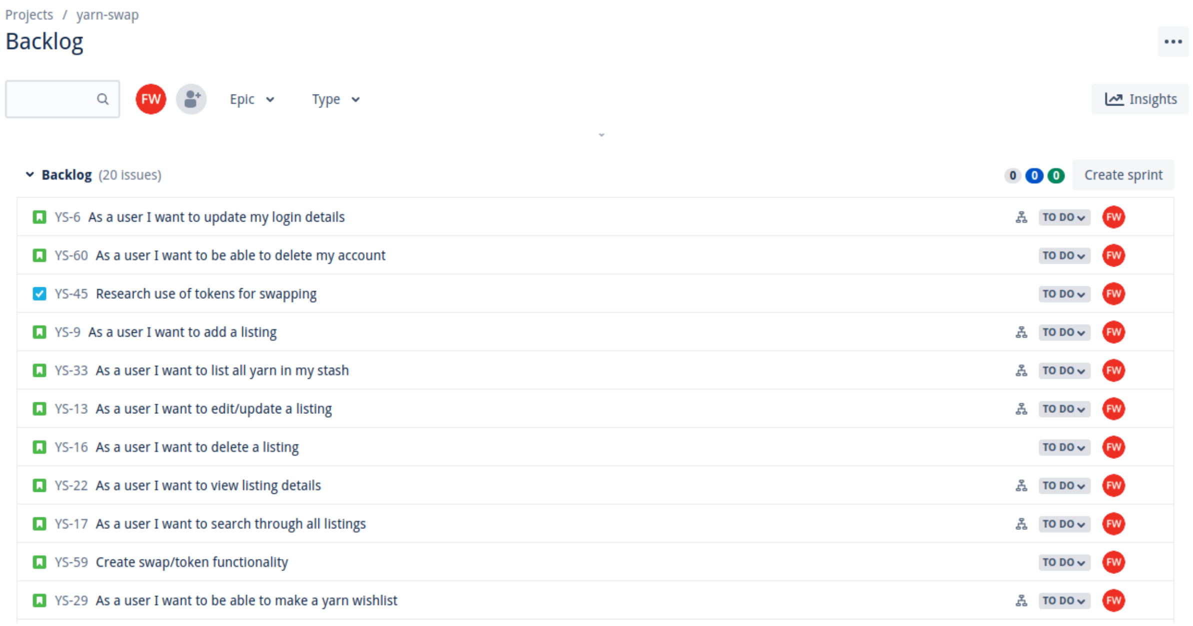
Create wireframes and Diagrams (*ER Diagram)
☑ YS-37 FW

Write modelling section of report
☑ YS-42 FW

DONE 1 ISSUE ✓

Research using Gin Web Framework
☑ YS-44 ✓ FW

Figure 7 - Jira Backlog, Yarn Swap Project



Projects / yarn-swap

Backlog

FW Epic Type Insights

Backlog (20 issues) 0 0 0 Create sprint

YS-6	As a user I want to update my login details	TO DO	FW
YS-60	As a user I want to be able to delete my account	TO DO	FW
☑ YS-45	Research use of tokens for swapping	TO DO	FW
YS-9	As a user I want to add a listing	TO DO	FW
YS-33	As a user I want to list all yarn in my stash	TO DO	FW
YS-13	As a user I want to edit/update a listing	TO DO	FW
YS-16	As a user I want to delete a listing	TO DO	FW
YS-22	As a user I want to view listing details	TO DO	FW
YS-17	As a user I want to search through all listings	TO DO	FW
YS-59	Create swap/token functionality	TO DO	FW
YS-29	As a user I want to be able to make a yarn wishlist	TO DO	FW

5 Implementation

5.1 System Model

Figure 8 - System Architecture Diagram

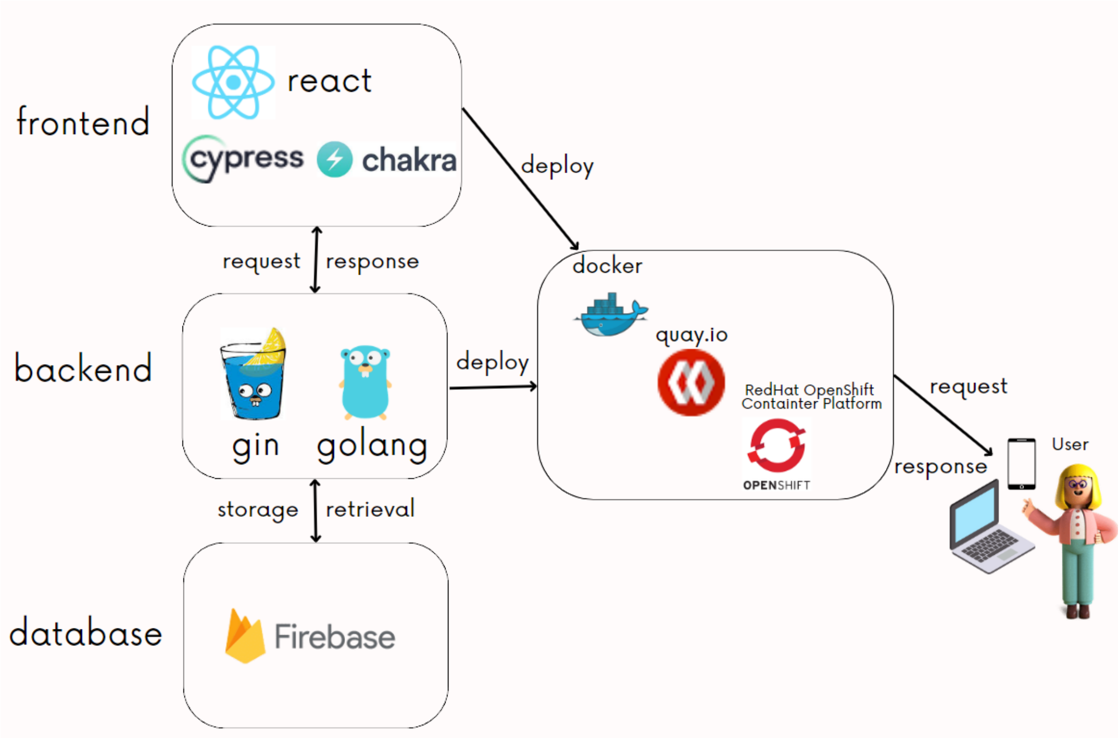


Figure 8 shows the architecture of the system. The user will interact via their browser, through Red Hat OpenShift which will expose the application which has been deployed via docker images hosted on Quay.io.

The front end is a single page app created using React with Vite.js and ChakraUI. Some end-to-end tests have been created using Cypress. This front end will interact with a back-end API created using Golang with the Gin Web Framework. This in turn will communicate with Firebase Realtime Database, and Firebase Storage for storage and retrieval of data and images, as well as Firebase Authentication.

5.2 Feature Summary

The following features have been implemented creating a Minimum Viable Product (MVP):

The core feature of this application is the exchange of yarn. Yarn can change hands so that another person can make use of it.

1. Register and Login.

    - A member of the public will arrive at the site, they will be able to view some of the most recent listings added to the site by users. This is the only feature available to a non-registered user.
    - A member of the public can choose to register. They then can access all of features of the application.
2. Create/Read/Update/Delete listings.
    - A logged in user can create a listing. This listing contains details of a skein/ball of yarn that they own. This listing can serve as a reminder to the user that this yarn exists in their collection but can also be marked as ‘swappable’ which will allow another user to obtain it.
    - The user can also update or archive this listing.
3. Use of Tokens. When a user adds a listing, it is reviewed by an administrator. If the listing is accepted the user will be awarded a token. This token is a virtual coin which will allow the user to request a swap from another user. Without uploading a listing, a user cannot obtain a token and therefore cannot request a swap.
4. View other users’ listings.
    - Each logged in user can view all other users’ listings which are marked as ‘swappable.’
5. Requesting a swap.
   The user can request a swap by clicking a button on the listing card. If the user is in possession of the required token the swap process can commence. This will notify the other user who can accept or deny the swap transaction. This notification will be available in the swap section of the application. If the swap is accepted this will create a chat between the users so that they can arrange the swap. This arrangement will include payment of postage to be made by the user requesting the yarn via third party payment providers such as Revolut or Paypal. Both parties will participate in confirming that the swap has been completed – at this point the Token will be removed from the user’s account.
    - If the swap is denied, the user requesting the swap will be notified.
6. Search and Filter functionality is present on Dashboard, and Listings pages.
7. The user will have access to their profile and will be able to update information or delete their account.
8. Clear guidelines and terms & conditions will be provided on the application About page.

The following features were proposed but not completed and can be added to a future feature list:

1. A user can add other user’s listings to their Wishlist. This allows a user to reflect before requesting a swap. The Wishlist can be organised into projects so that the user can attribute these items to a particular item that they would like to make.
2. A Forum would provide the ability for users to form a community to be able to ask for advice on projects and yarn selection.

Other potential future features:

- The ability to purchase yarn from another user.
- Increased token amount for higher quality yarn.
- Ability to obtain more tokens by purchasing them or by gaining bonus loyalty tokens.

- User ratings.
- Project To Do list.
- Request another user to knit/make an item for you.

5.3 Frontend and UI:

Technologies used:

- React
- Vite.js
- Chakra UI
- Storybook
- Cypress
- Github Action
- SendBird
- Firebase Authentication

React with Vite.js and Chakra UI were used to create a front facing single page web application.

When creating React components, the Atomic Web Design strategy was followed. This teaches the breaking down of features into their smallest components and building them into "thoughtful design systems" (Frost, 2013).

Some Storybook stories were created to trial components, but this is not extensive.
Some Cypress End to End tests have also been added and are run via a Github Action on all pushes to the Yarn Swap frontend Github Repo.

SendBird is a third-party chat app that allows you to build in-app chat, calls and live streaming. The SendBird UIKit was employed to add chat functionality to Yarn Swap.

PWA Vite Plugin was utilised in making this app progressive and available to install as an app on a mobile phone or computer desktop.

'Progressive Web Apps (PWAs) are web application built and enhanced with modern APIs to deliver enhanced capabilities, reliability, and installability while reaching anyone, anywhere, on any device, all with a single codebase.' (A. Fu, 2023)

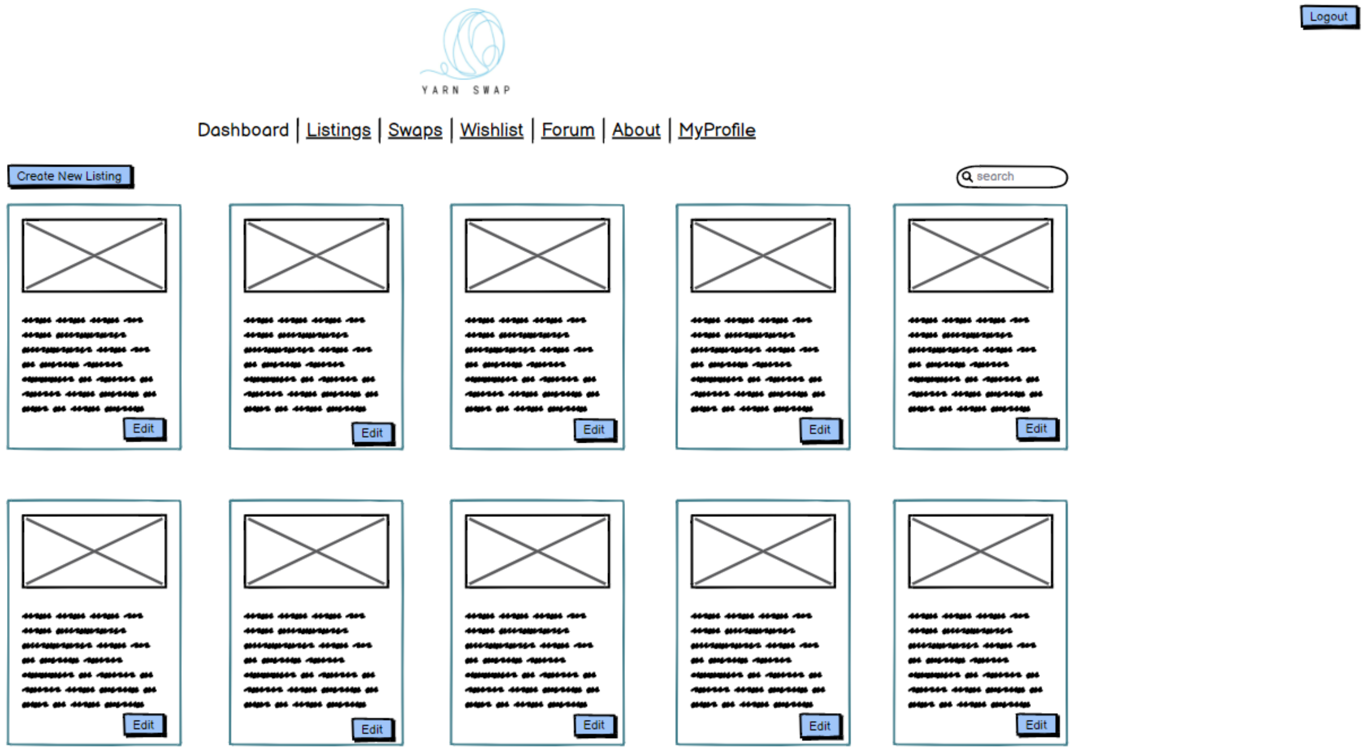
A progressive web app can be used online and can be configured to cache certain data and allow limited capabilities when a user is offline. For the Yarn Swap application it has been configured to cache the data that is loaded by the application on initial login, and to inform the user of the limited capabilities when the app goes offline. The user is asked on login If they would like to install the app, once they have installed it they will not be asked repeatedly.

5.3.1 UI/System Walkthrough

At the beginning of the project wireframes were created using Balsamiq, a web-based mock-up tool. These were then used when creating the frontend components.

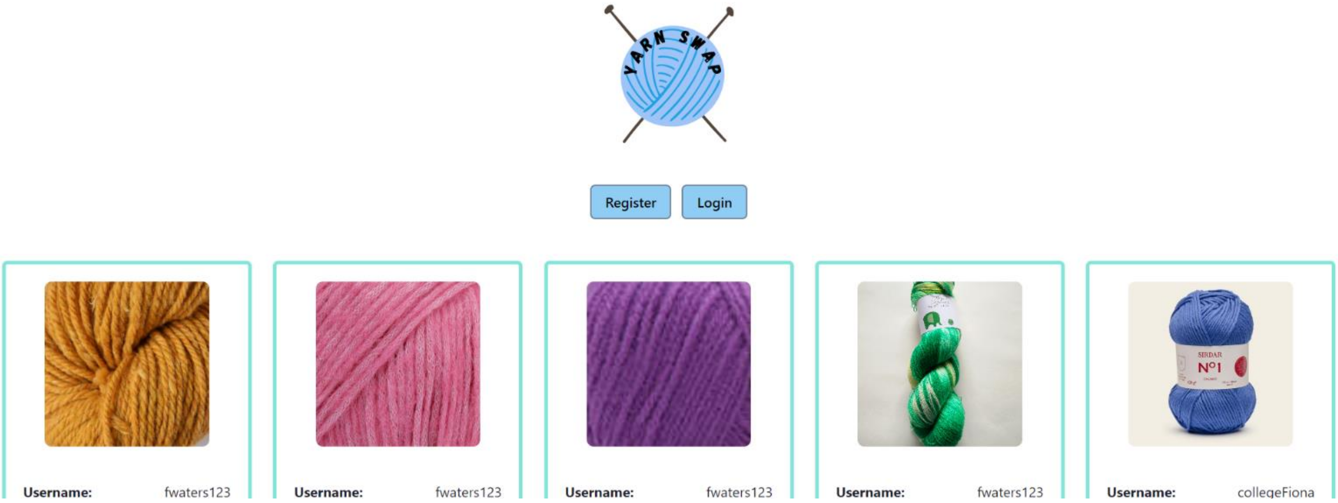
Below is the wireframe created for the Dashboard page. The resulting UI is very close to what was planned when creating the wireframes. All wireframes and details can be seen at appendix B.

Figure 9 - Dashboard Wireframe


Logout
YARN SWAP
Dashboard | Listings | Swaps | Wishlist | Forum | About | MyProfile
Create New Listing
search
Edit

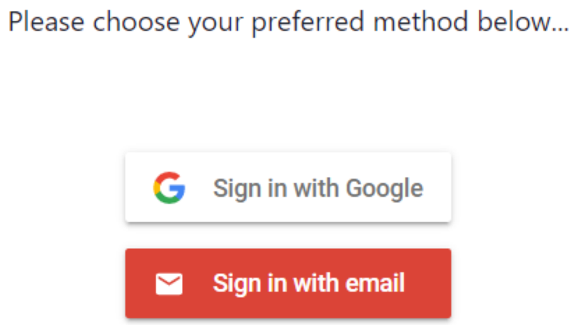
On first encounter with the system, the user or a member of the public will be offered the option to register or login. As a non-registered user, you can view a limited number of listings on the home page. The listings will include an image and text describing the item listed.

Figure 10 - Home Page



Once a user has clicked on the Register or Login button, they will be brought to a screen which will allow them to authenticate via Google or Email. The firebaseui-web library was paired with Firebase Authentication to create this auth system.

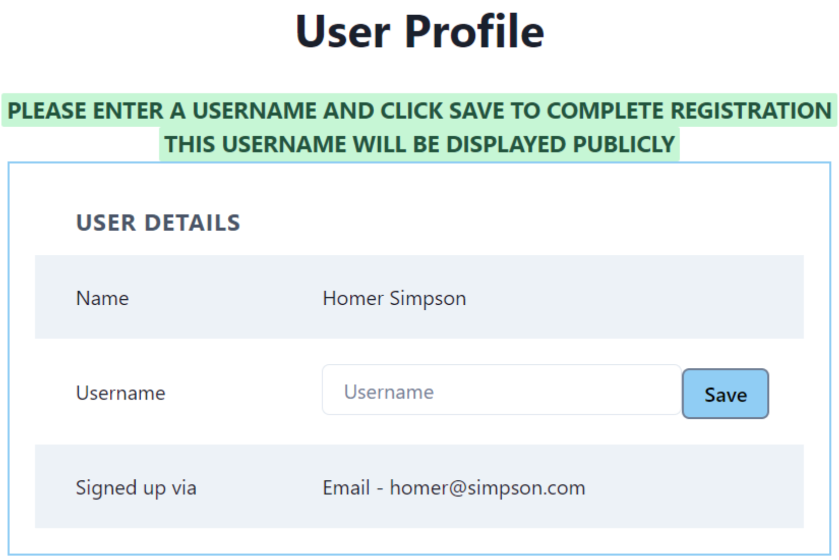
Figure 11 - User choice of signup/login route



Please choose your preferred method below...

When a user logs in for the first time, they must create a username; They are presented with this screen:

Figure 12 - Add username on first login



User Profile

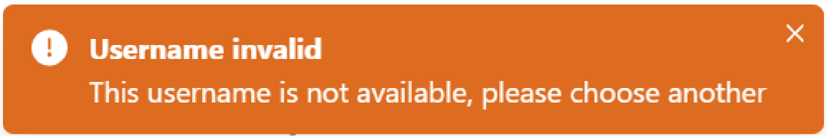
PLEASE ENTER A USERNAME AND CLICK SAVE TO COMPLETE REGISTRATION
THIS USERNAME WILL BE DISPLAYED PUBLICLY

USER DETAILS

Name	Homer Simpson
Username	Username
Signed up via	Email - homer@simpson.com

The username must be unique – this is checked via a call to the backend API and the database. If it is not unique, the user will receive a warning that they must enter another username.

Figure 13 - Warning that username is not unique



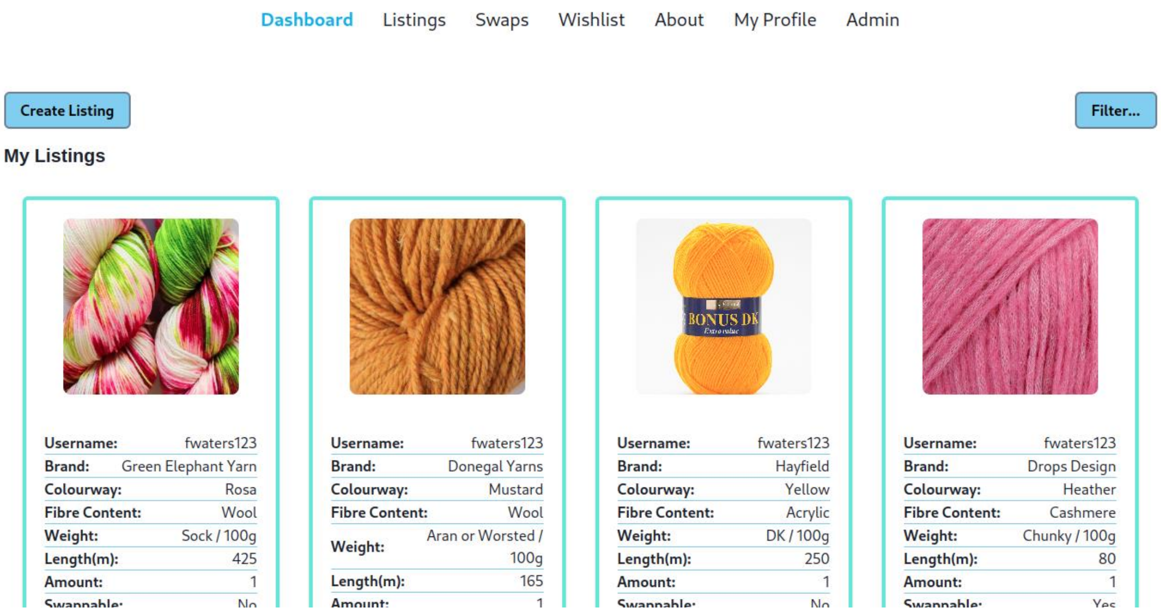
Username invalid

This username is not available, please choose another

Once a unique username is entered the user will be directed to their Dashboard page.

From here they can create a new listing, and view and filter their own listings (once added).

Figure 14 - Dashboard



My Listings

Username:	fwaters123
Brand:	Green Elephant Yarn
Colourway:	Rosa
Fibre Content:	Wool
Weight:	Sock / 100g
Length(m):	425
Amount:	1
Swappable:	No

Username:	fwaters123
Brand:	Donegal Yarns
Colourway:	Mustard
Fibre Content:	Wool
Weight:	Aran or Worsted / 100g
Length(m):	165
Amount:	1

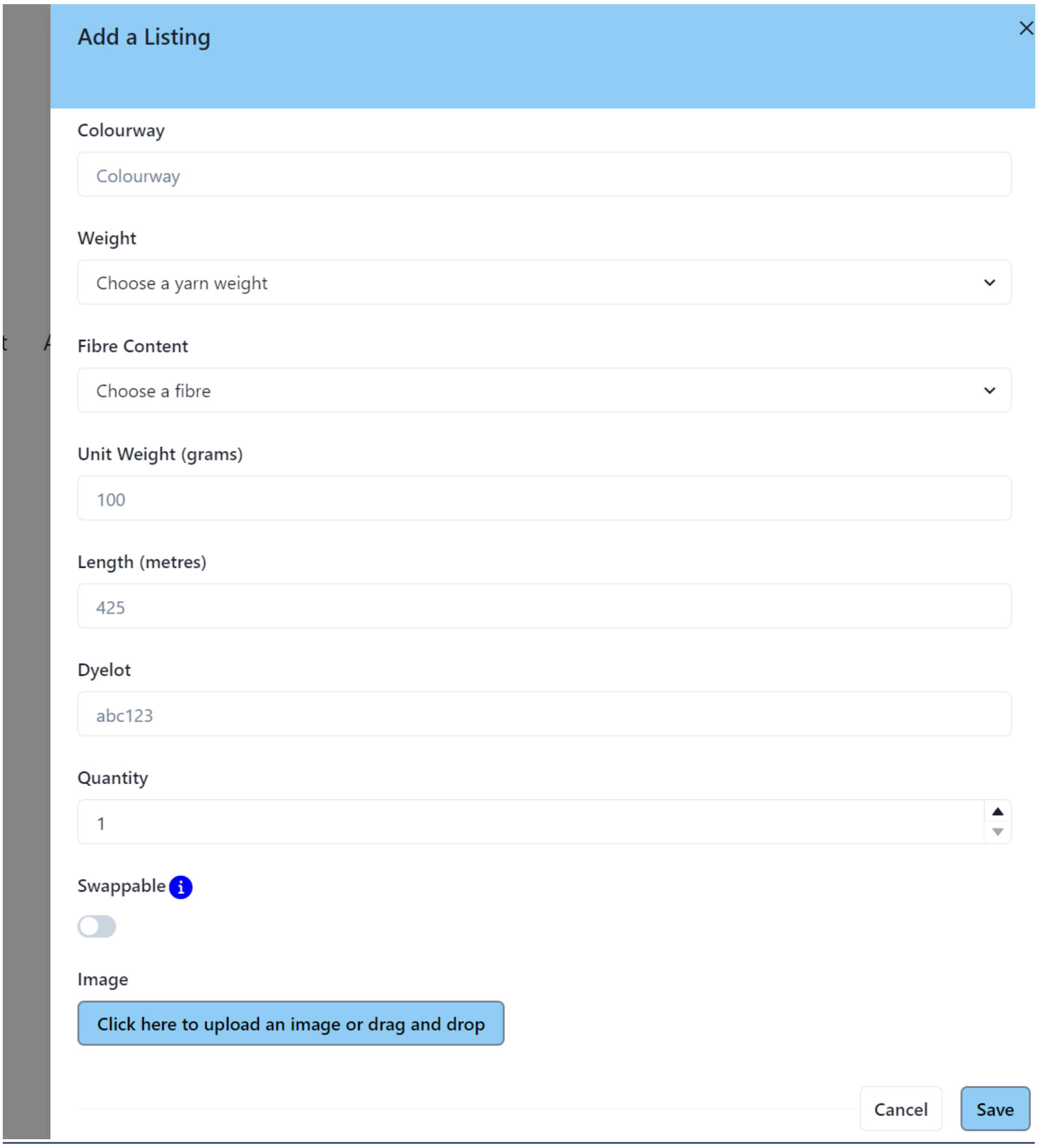
Username:	fwaters123
Brand:	Hayfield
Colourway:	Yellow
Fibre Content:	Acrylic
Weight:	DK / 100g
Length(m):	250
Amount:	1
Swappable:	No

Username:	fwaters123
Brand:	Drops Design
Colourway:	Heather
Fibre Content:	Cashmere
Weight:	Chunky / 100g
Length(m):	80
Amount:	1
Swappable:	Yes

Each page a logged-in user navigates to will include a navigation bar which will allow them to navigate to Dashboard, Listing, Swaps, Wishlist, About and My Profile. An admin user will also have an Admin option.

If a user decides to create a new listing, they are presented with a simple form presented in a drawer to the right of the window. They will be prompted to fill in the yarn details. There are several different fields ranging from number input and dropdown menus to a toggle button and drag and drop image section. Clicking add listing will return the user to their Dashboard, where they can now view this new listing alongside others already added. If the listing is marked as swappable, it must be approved by an admin user before it is available to swap. When it is marked as approved the owner receives a token. Number of tokens can be viewed in the users My Profile page which will be discussed shortly.

Figure 15 - Add Listing Form



Add a Listing

Colourway

Colourway

Weight

Choose a yarn weight

Fibre Content

Choose a fibre

Unit Weight (grams)

100

Length (metres)

425

Dyelot

abc123

Quantity

1

Swappable

Image

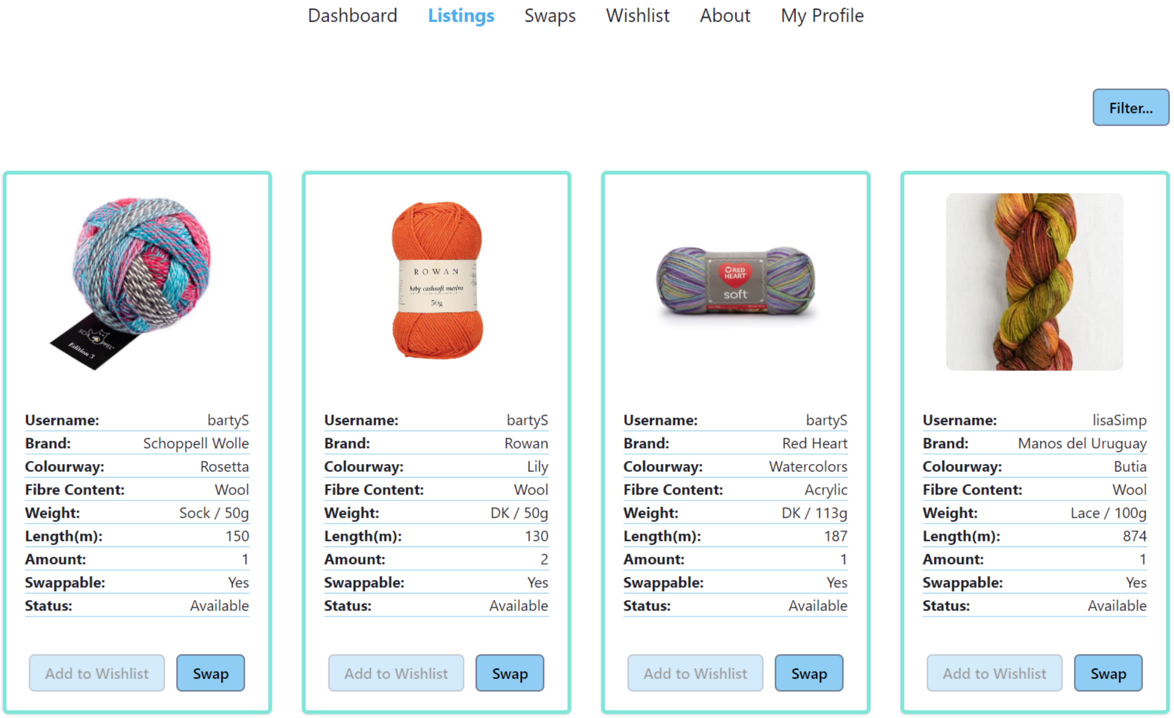
Once a listing is added, the card will include an Edit button, clicking this will open the same drawer. The data will be prepopulated, and the user will be able to update it as appropriate and save it. Clicking the save button will bring the user back to their Dashboard. The user also has the option to archive the listing at this stage. This will not delete the listing from the database but will label it as archived. An archived listing remains on the users own Dashboard but will not be available for other users to view/request to swap.

On the Listings page, a user will see a page like their own dashboard but without functionality to add a new listing or to edit the listing. These are listings belonged to other users that have been marked as swappable. The user will see a Swap button and an Add to Wishlist button on each card, as well as an image and text containing listing details.

The Add to Wishlist button is currently disabled as the Wishlist functionality has not been completed.

If the user does not have any tokens, the Swap button will not be present.

Figure 16 - Listings Page



Username:	bartyS
Brand:	Schoppell Wolle
Colourway:	Rosetta
Fibre Content:	Wool
Weight:	Sock / 50g
Length(m):	150
Amount:	1
Swappable:	Yes
Status:	Available

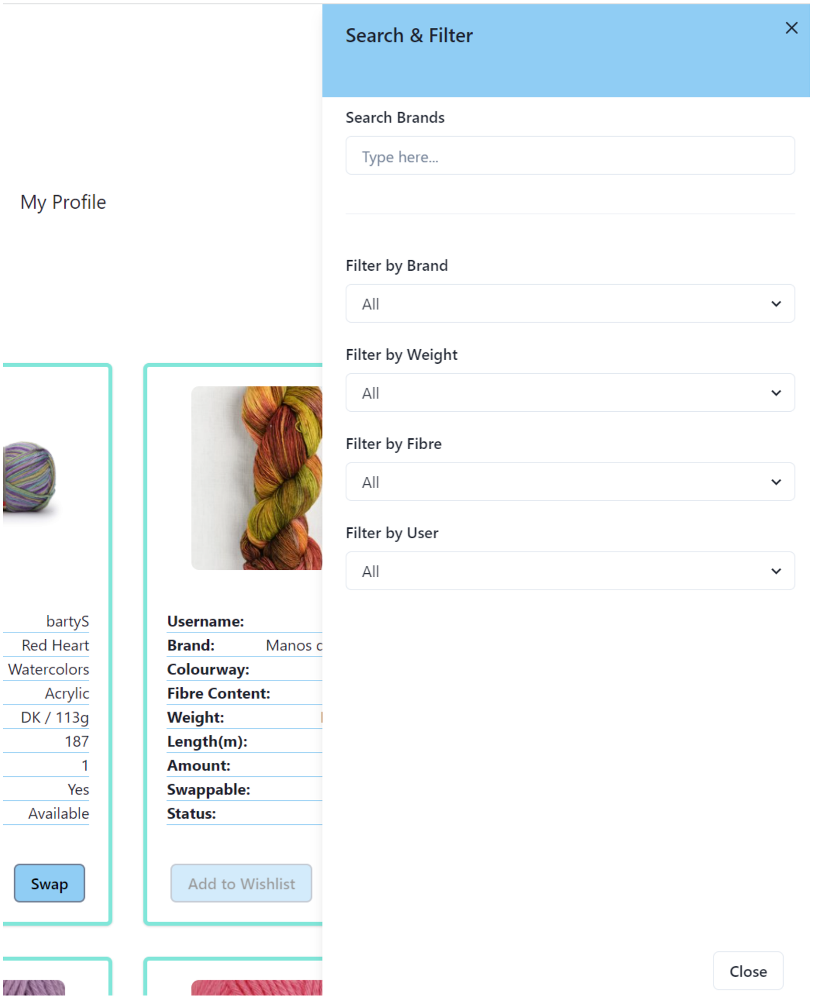
Username:	bartyS
Brand:	Rowan
Colourway:	Lily
Fibre Content:	Wool
Weight:	DK / 50g
Length(m):	130
Amount:	2
Swappable:	Yes
Status:	Available

Username:	bartyS
Brand:	Red Heart
Colourway:	Watercolors
Fibre Content:	Acrylic
Weight:	DK / 113g
Length(m):	187
Amount:	1
Swappable:	Yes
Status:	Available

Username:	lisaSimp
Brand:	Manos del Uruguay
Colourway:	Butia
Fibre Content:	Wool
Weight:	Lace / 100g
Length(m):	874
Amount:	1
Swappable:	Yes
Status:	Available

On both the Dashboard and Listings pages, the user can search/filter listings. Clicking on the Filter button will open a drawer from the right, as below:

Figure 17 - Search & Filter Drawer



Search & Filter

Search Brands: Type here...

Filter by Brand: All

Filter by Weight: All

Filter by Fibre: All

Filter by User: All

bartyS
Red Heart
Watercolors
Acrylic
DK / 113g
187
1
Yes
Available

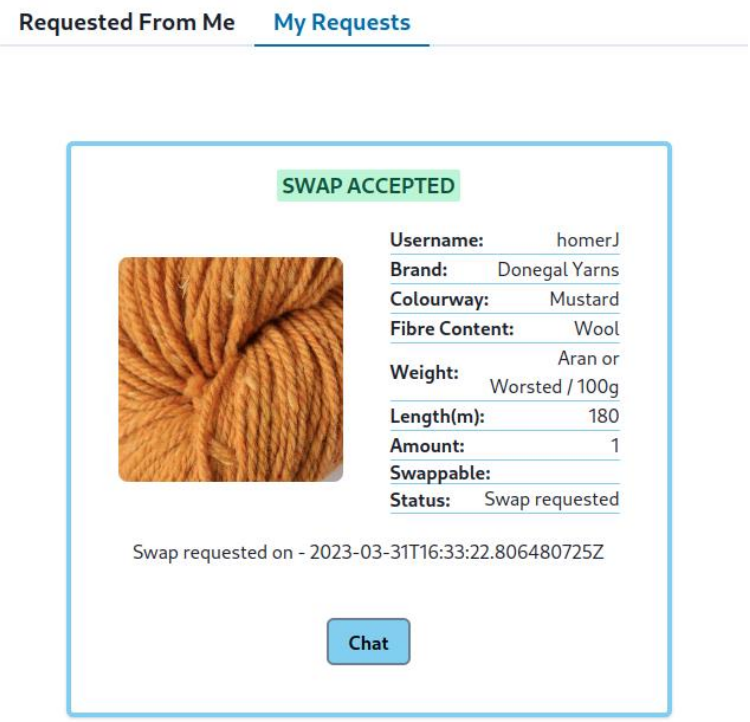
Username:
Brand: Manos c
Colourway:
Fibre Content:
Weight:
Length(m):
Amount:
Swappable:
Status:

On the Listings page there is an option to filter by User, this is replaced with an option to filter by Listing Status on the Dashboard page, as filtering by user does not make sense there. All other filters are the same for both pages.

If the user decides to request a swap they will click on the Swap button, on the listing card. This will trigger the beginning of the swap process. Swaps can be view on the Swaps page.

On the Swaps page there are two tabs, one containing swaps that were requested from the user, and one containing their requests to other users.

Figure 18 - Swaps



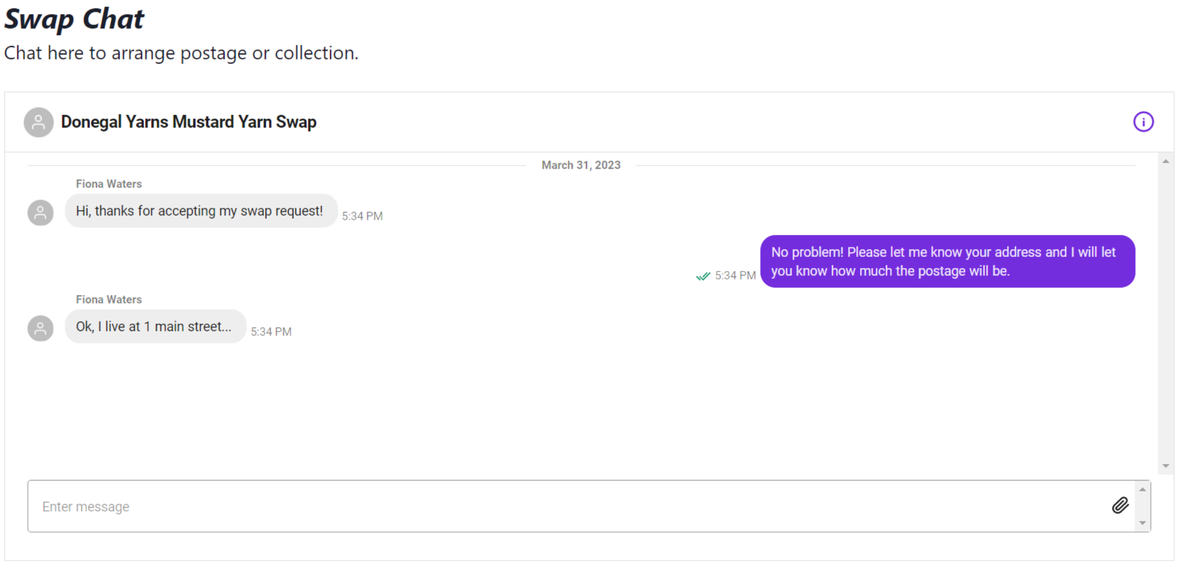
Username:	homerJ
Brand:	Donegal Yarns
Colourway:	Mustard
Fibre Content:	Wool
Weight:	Aran or Worsted / 100g
Length(m):	180
Amount:	1
Swappable:	
Status:	Swap requested

Swap requested on - 2023-03-31T16:33:22.806480725Z

The swap card shows a flag at the top with the colour coded swap status, the listing details along with an image and certain buttons depending on the stage of the swap. If the swap is declined, there is an option to remove the swap, along with the reason it was declined; if the swap is accepted, a chat button will be present.

Clicking on the Chat button will bring the user here:

Figure 19 - Swap Chat



Swap Chat

Chat here to arrange postage or collection.

Donegal Yarns Mustard Yarn Swap

March 31, 2023

Fiona Waters

Hi, thanks for accepting my swap request! 5:34 PM

5:34 PM No problem! Please let me know your address and I will let you know how much the postage will be.

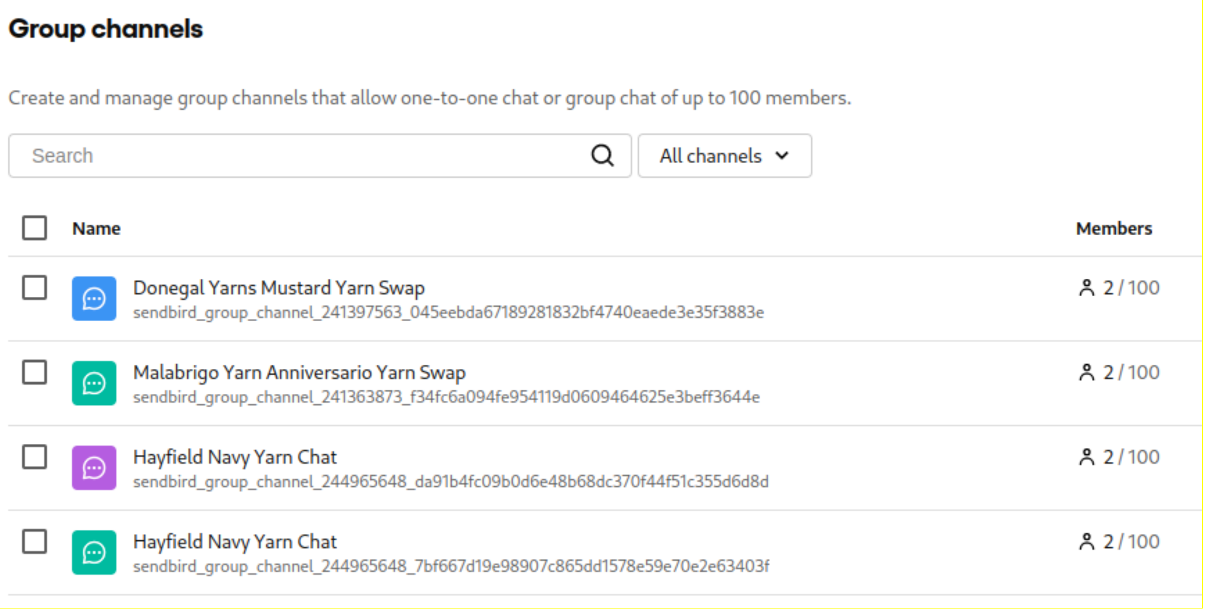
Fiona Waters

Ok, I live at 1 main street... 5:34 PM

This chat is only between these two users and can only be accessed via the chat button on the swap card. The users discuss and arrange postage of the yarn. This chat feature has been created using SendBird.

SendBird has a helpful admin UI, which allows administrators to carry out administrative tasks, for example intercepting the chat if necessary.

Figure 20 - Swap Chat



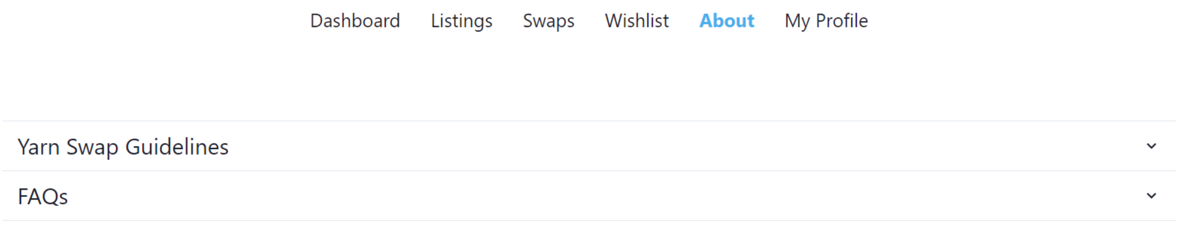
Group channels

Create and manage group channels that allow one-to-one chat or group chat of up to 100 members.

□	Name	Members
□	Donegal Yarns Mustard Yarn Swap sendbird_group_channel_241397563_045eebda67189281832bf4740eaede3e35f3883e	2 / 100
□	Malabrigo Yarn Anniversario Yarn Swap sendbird_group_channel_241363873_f34fc6a094fe954119d0609464625e3beff3644e	2 / 100
□	Hayfield Navy Yarn Chat sendbird_group_channel_244965648_da91b4fc09b0d6e48b68dc370f44f51c355d6d8d	2 / 100
□	Hayfield Navy Yarn Chat sendbird_group_channel_244965648_7bf667d19e98907c865dd1578e59e70e2e63403f	2 / 100

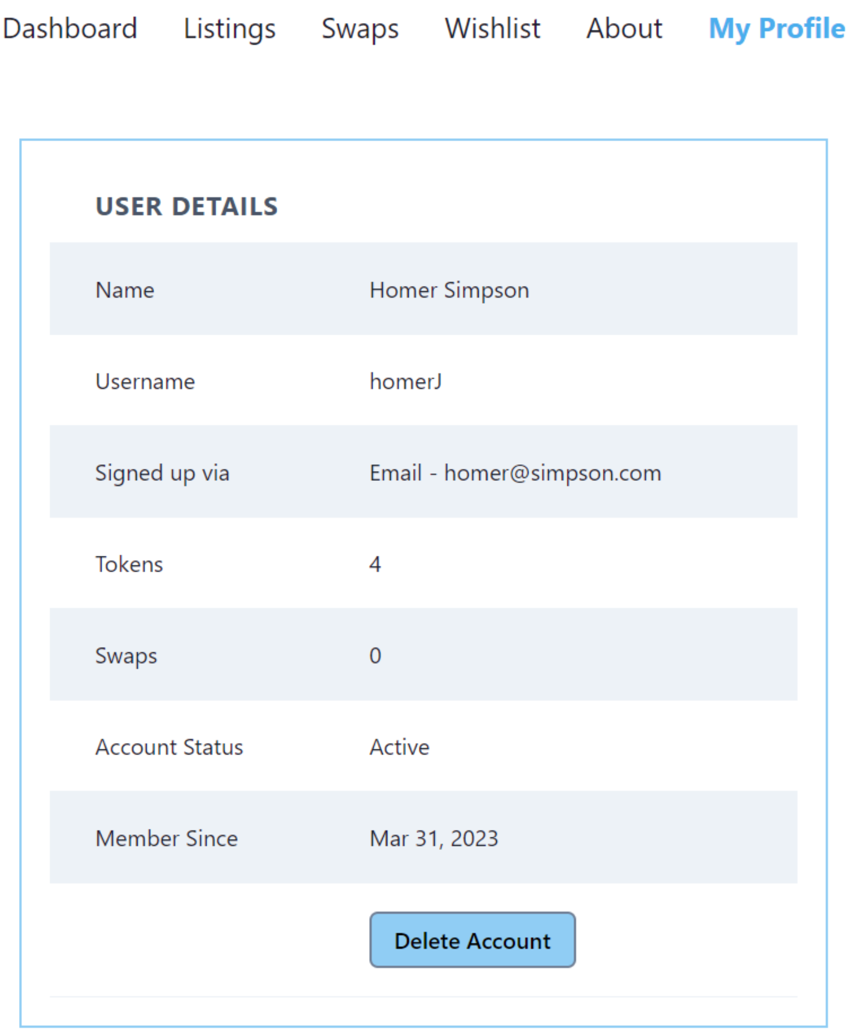
Other screens:

Figure 21 - About Page



The about Page contains Yarn Swap Guidelines and FAQ in an accordion format.

Figure 22 - My Profile Page



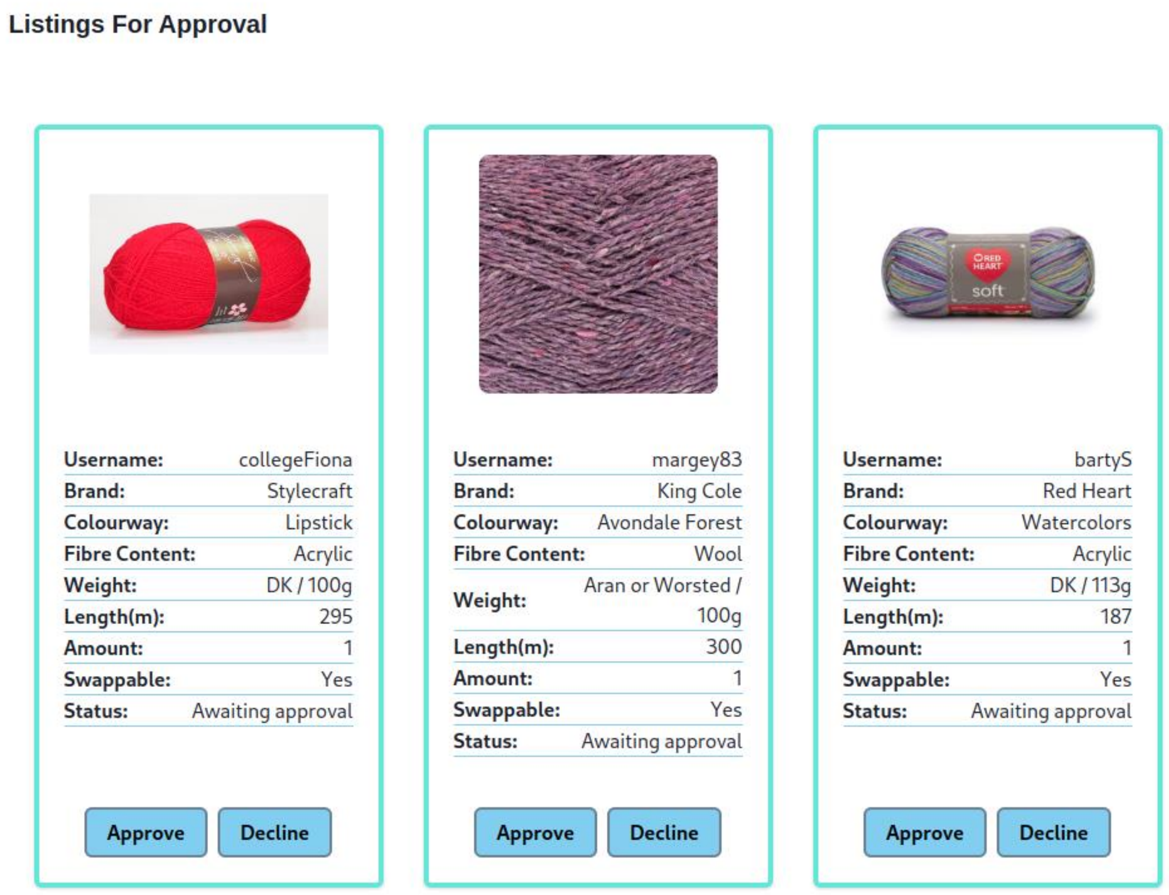
USER DETAILS

Name	Homer Simpson
Username	homerJ
Signed up via	Email - homer@simpson.com
Tokens	4
Swaps	0
Account Status	Active
Member Since	Mar 31, 2023

The My Profile page contains details regarding the user’s profile. The user can Delete their account from here. When the delete button is clicked, and the subsequent pop-up warning accepted, the users account, along with their listings, are archived for 30 days. Within this time the user can reactivate their account. If they do not, it is deleted.

Functionality to complete this deletion has been added to the backend of the Yarn Swap application. However, it is not completely implemented. The proposed implementation involves creating a cron-job that checks the timestamp difference at intervals and deletes accounts that have passed the 30-day limit, due to time constraints this has not yet been implemented.

Figure 23 - Admin Page



Listings For Approval

Username:	collegeFiona
Brand:	Stylecraft
Colourway:	Lipstick
Fibre Content:	Acrylic
Weight:	DK / 100g
Length(m):	295
Amount:	1
Swappable:	Yes
Status:	Awaiting approval

Username:	margey83
Brand:	King Cole
Colourway:	Avondale Forest
Fibre Content:	Wool
Weight:	Aran or Worsted / 100g
Length(m):	300
Amount:	1
Swappable:	Yes
Status:	Awaiting approval

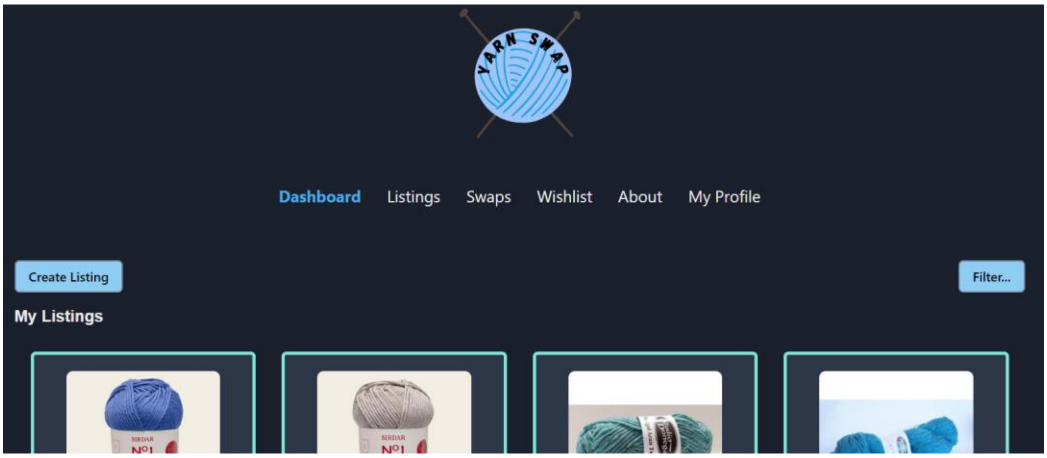
Username:	bartyS
Brand:	Red Heart
Colourway:	Watercolors
Fibre Content:	Acrylic
Weight:	DK / 113g
Length(m):	187
Amount:	1
Swappable:	Yes
Status:	Awaiting approval

An admin user will have an extra navigation item called Admin. This is where they can view the listings that users have requested to add as swappable. The admin user can approve or decline these listings, via the Approve and Decline buttons on the listing card. If they choose to approve it, the listing status will update to ‘Available,’ and the user will be allocated 1 token. This token can then be used to request a swap.

If they choose to decline the listing, a pop-up will request a reason which will be communicated to the user in question via the listing on their Dashboard.

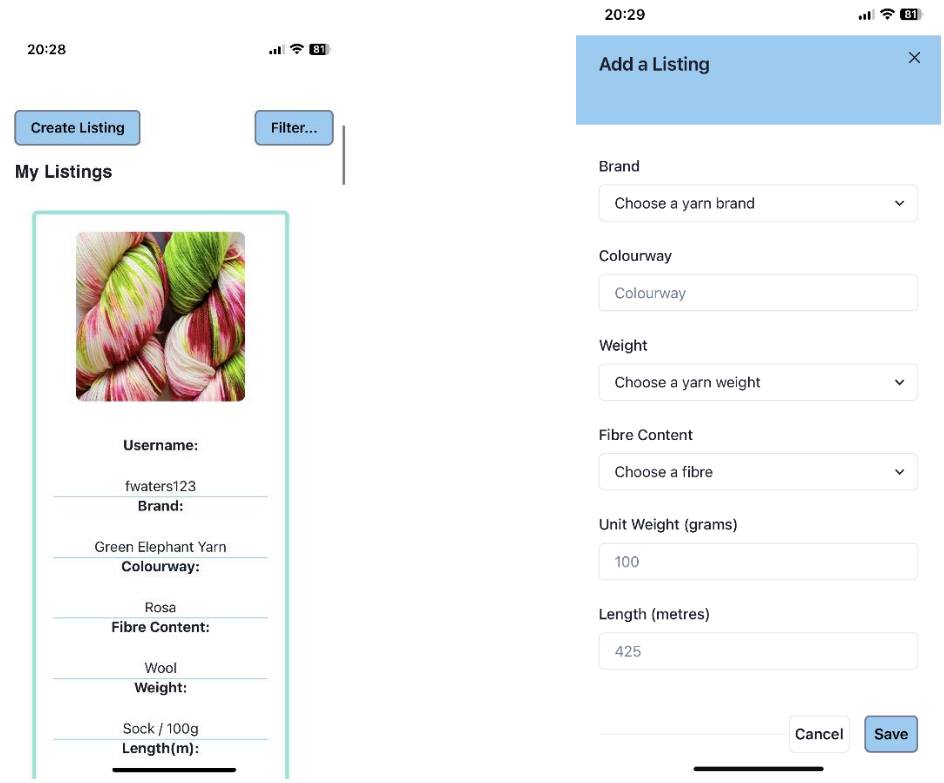
A dark/night mode option has also been implemented.

Figure 24 - Dark Mode



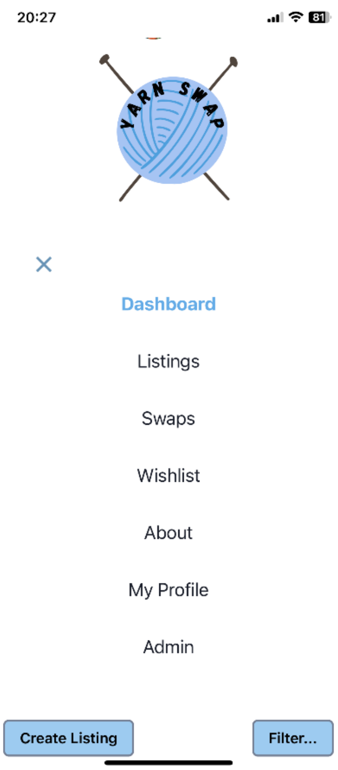
As Yarn Swap is a progressive web application it can also be installed as an app on a mobile device. Therefore, it has been created with a responsive design. Samples of this can be seen below.

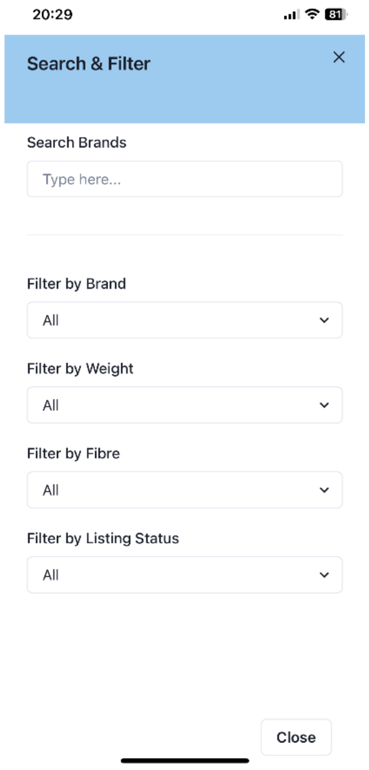
Figure 25 - Mobile Responsive Design Examples (4)



Username:
fwaters123
Brand:
Green Elephant Yarn
Colourway:
Rosa
Fibre Content:
Wool
Weight:
Sock / 100g
Length(m):

Brand
Choose a yarn brand
Colourway
Colourway
Weight
Choose a yarn weight
Fibre Content
Choose a fibre
Unit Weight (grams)
100
Length (metres)
425





Search Brands

Type here...

Filter by Brand: All

Filter by Weight: All

Filter by Fibre: All

Filter by Listing Status: All

This concludes the UI walkthrough, any remaining detail not shown here will be covered in the project video and presentation.

5.4 Backend API

Technologies used:

- Golang
- Gin
- Go Testing Package
- Github Actions
- Firebase Realtime Database, Firebase Admin SDK

The backend web application has been written in Golang with use of the Gin Web Framework. The Gin Web Framework allows you to route requests, retrieve request details, and marshal JSON for responses, as specified in the Go official documentation. In conjunction with these, testing of the API endpoints was completed using the Go testing package. These tests are run each time a commit is pushed to the backend GitHub repo, this has been set up via a Github Actions action.

The backend API connects with the application database – Firebase Realtime Database. To communicate with the database, the Firebase Admin SDK package has been used, this allows us to read from and write to the database.

The Yarn Swap backend API has eleven endpoints in total which can be seen below in func main(), the function which is run immediately after the application starts.

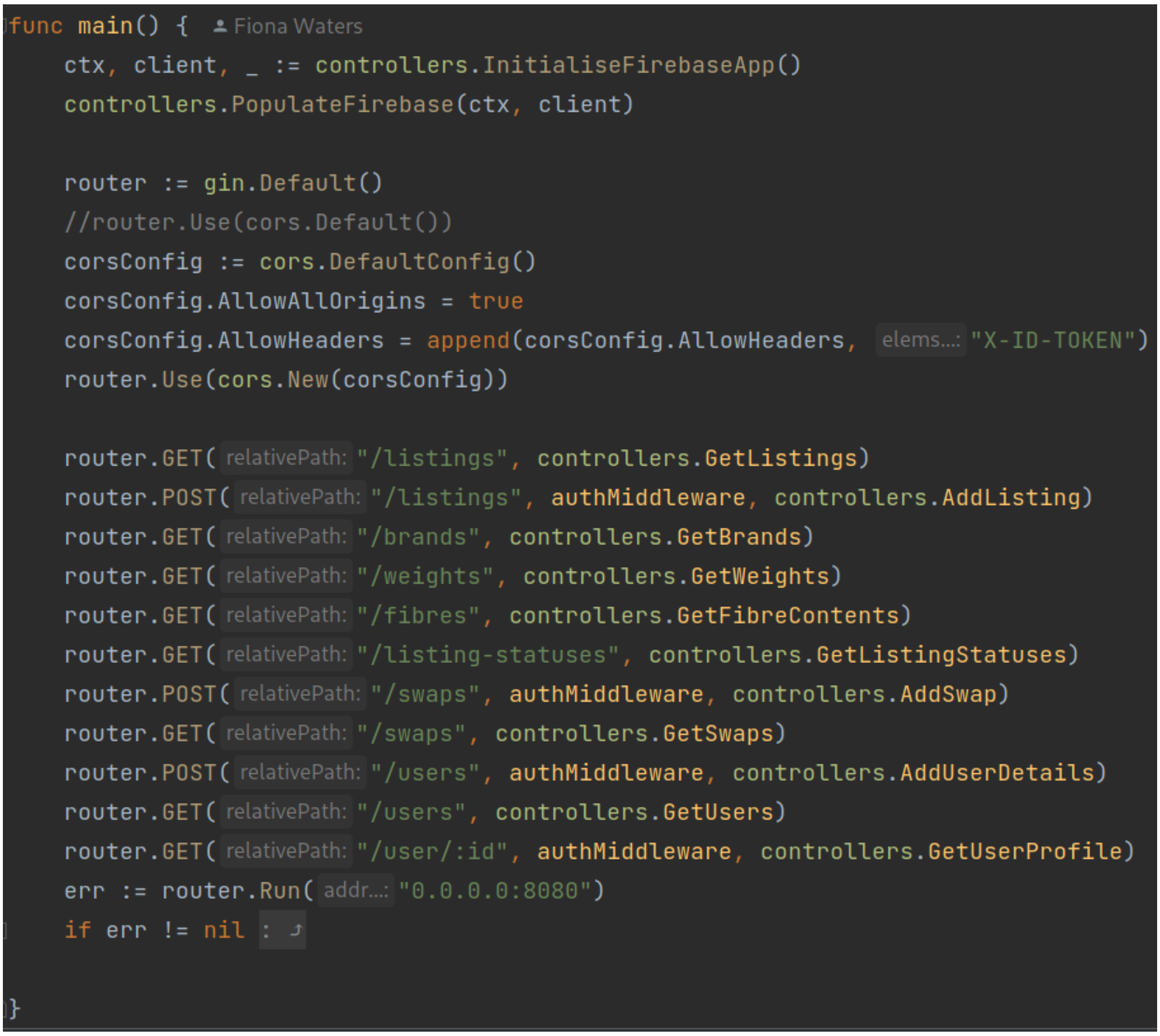
The Firebase App is initialised and populated with data required for dropdown menus etc (only if the data has changed since the last time the application has restarted). Then the router is set up, this is where the Gin Framework comes in. There is some CORS* settings to allow required requests, and then our routes. Each of these is connected to a function allowing access to the data saved in the Firebase Realtime Database.

*According to developer.mozilla.org (2023), 'Cross-Origin Resource Sharing (CORS) is an HTTP-header based mechanism that allows a server to indicate any origins (domain, scheme, or port) other than its own from which a browser should permit loading resources'.

The authMiddleware function, which prepends some of the endpoints allows an ID token to be checked before allowing access to write or read from the database. This has been added to endpoints that add data to the database or retrieve a specific user's profile. This ID token, if available, is sent via the request from the frontend. Only an authorised user has access.

All other API functions can be viewed in the Yarn Swap backend GitHub repo. In the main_test.go file, the functions that test these API endpoints can be seen. The command 'go test' will run these.

Figure 26 - Yarn Swap Backend main.go main()



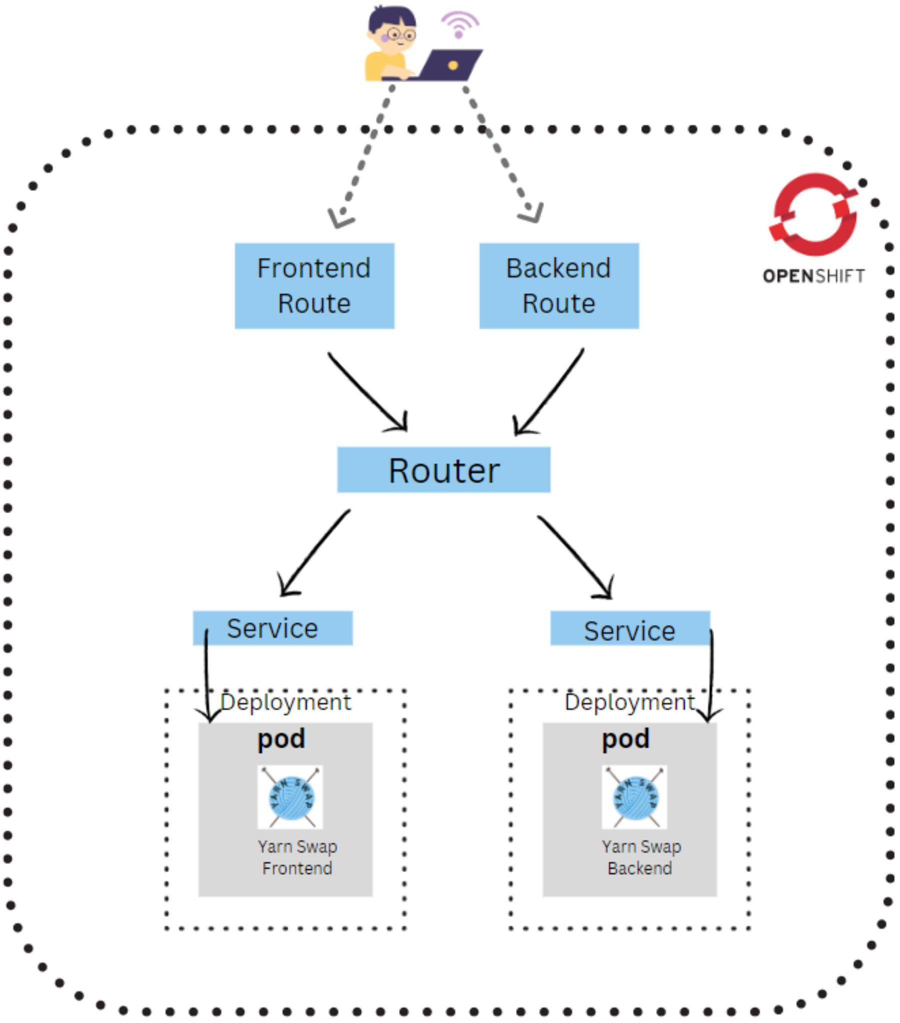
```
func main() {  Fiona Waters
    ctx, client, _ := controllers.InitialiseFirebaseApp()
    controllers.PopulateFirebase(ctx, client)

    router := gin.Default()
    //router.Use(cors.Default())
    corsConfig := cors.DefaultConfig()
    corsConfig.AllowAllOrigins = true
    corsConfig.AllowHeaders = append(corsConfig.AllowHeaders,  elems...: "X-ID-TOKEN")
    router.Use(cors.New(corsConfig))

    router.GET( relativePath: "/listings", controllers.GetListings)
    router.POST( relativePath: "/listings", authMiddleware, controllers.AddListing)
    router.GET( relativePath: "/brands", controllers.GetBrands)
    router.GET( relativePath: "/weights", controllers.GetWeights)
    router.GET( relativePath: "/fibres", controllers.GetFibreContents)
    router.GET( relativePath: "/listing-statuses", controllers.GetListingStatuses)
    router.POST( relativePath: "/swaps", authMiddleware, controllers.AddSwap)
    router.GET( relativePath: "/swaps", controllers.GetSwaps)
    router.POST( relativePath: "/users", authMiddleware, controllers.AddUserDetails)
    router.GET( relativePath: "/users", controllers.GetUsers)
    router.GET( relativePath: "/user/:id", authMiddleware, controllers.GetUserProfile)
    err := router.Run( addr...: "0.0.0.0:8080")
    if err != nil : ↵

}
```

5.5 Deployment Infrastructure

There are many services available that allow deployment of a web application. Some of these services include Heroku, Netlify, AWS and Red Hat OpenShift. There are several factors to consider when deciding what the best option is including cost, reliability, security, and longevity of service. It is important to choose a service that fits with your goals and will last the test of time. In terms of being able to react quickly if your provider leaves the market, utilising containerisation can be of great benefit.

Containerising your application allows you to package it up with all its dependencies so that it can easily be moved and deployed elsewhere if necessary. Docker is a popular tool that is used to do this. For this application, 2 Docker images have been created, one of the backend and one of the frontend of the application. These have been deployed to a Red Hat OpenShift Cluster where a route is exposed for the application to be made available.

Figure 27 - Yarn Swap Deployment on Red Hat OpenShift



The frontend and backend of the application are deployed to a Red Hat OpenShift Cluster via the OC CLI (OpenShift Command Line Interface). For each part of the application (frontend and backed), this deployment process creates a deployment, a service, and a route. These all come together in preparing the application to be accessed via the users browser.

*Please note: The OpenShift Cluster will not be always available; I have included the complete deployment procedure at Appendix D.

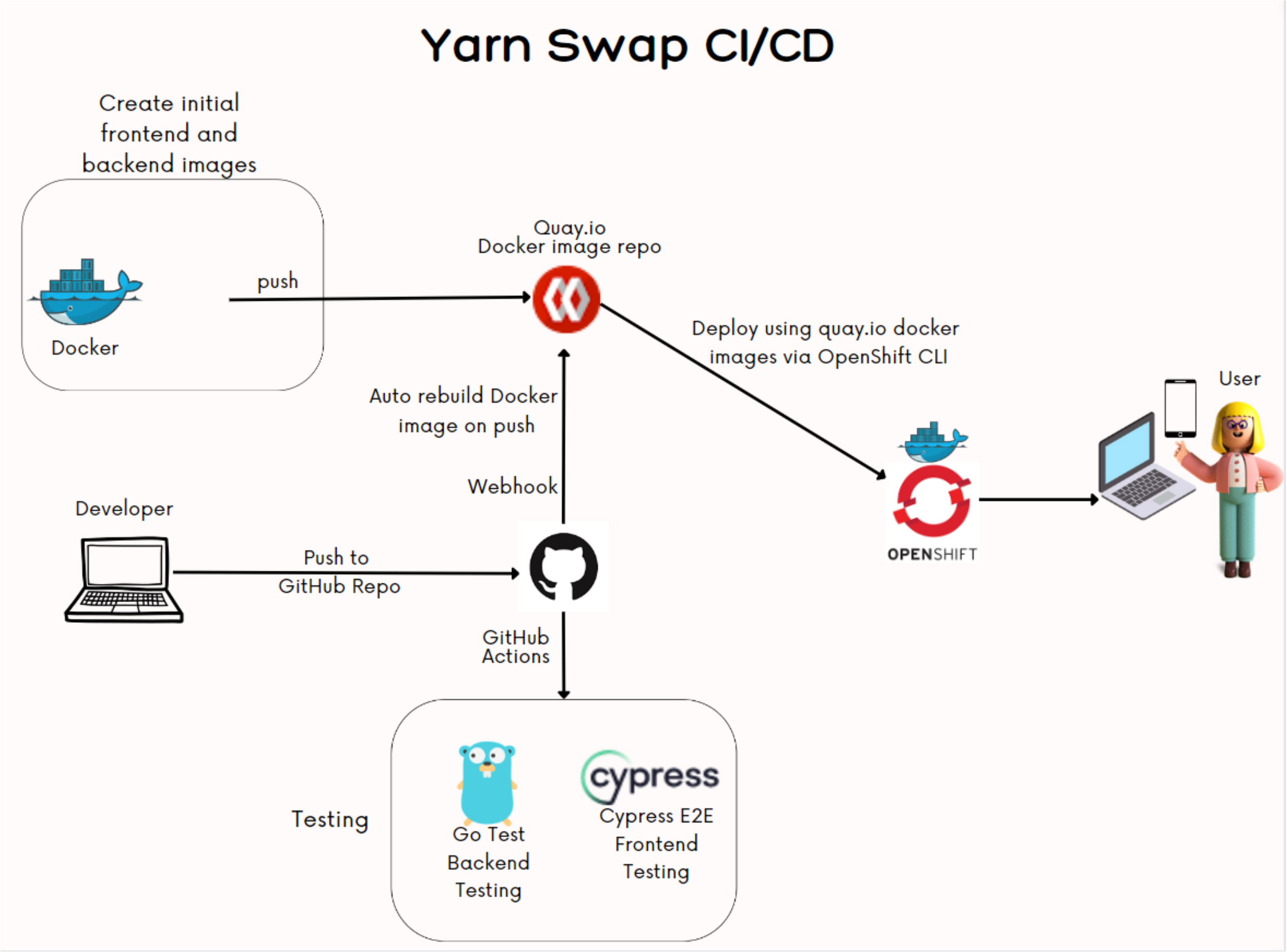
5.6 Continuous Integration/Continuous Deployment

“A continuous integration and continuous deployment (CI/CD) pipeline is a series of steps that must be performed in order to deliver a new version of software. CI/CD pipelines are a practice focused on improving software delivery throughout the software development life cycle via automation.” (RedHat.com, 2022)

For this project, the aim was to create a pipeline that will take the project source code from its GitHub repository, build it, test it, and deploy it on Red Hat OpenShift.

Below is a diagram of the integration and deployment of the completed project:

Figure 28 - Yarn Swap CI/CD



Yarn Swap CI/CD
Create initial frontend and backend images
Docker
push
Quay.io
Docker image repo
Deploy using quay.io docker images via OpenShift CLI
Auto rebuild Docker image on push
Webhook
Developer
Push to GitHub Repo
GitHub Actions
OPENSHIFT
User
Testing
Go Test Backend Testing
cypress
Cypress E2E Frontend Testing

Technologies used:

- Quay.io
- Github Actions
- Golang Testing Package
- Cypress
- Docker
- Red Hat OpenShift & OC CLI

Quai.io is a container registry, where container images can be stored. It is possible to connect via a webhook to a GitHub repository for a new image to be built. This has been configured to rebuild on push of a commit to the YarnSwap backend and/or frontend GitHub repos. This workflow builds a new docker image which is then used to deploy the application to a Red Hat OpenShift Cluster.

GitHub Actions is a CI/CD platform where “you can create workflows that build and test every pull request to your repository.” (GitHub Docs, 2023) This has been configured to run tests covering all YarnSwap API endpoints for the backend and a Cypress End to End tests for the frontend, on push to the GitHub repos.

Golang Testing Package provides support for the automated testing of Go packages. It has been used in the Yarn Swap project backend to test all API endpoints.

Cypress is a JavaScript frontend testing tool that allows creation and running of unit tests as well as end to end tests. It has been used in the Yarn Swap project frontend to test some of the applications workflows.

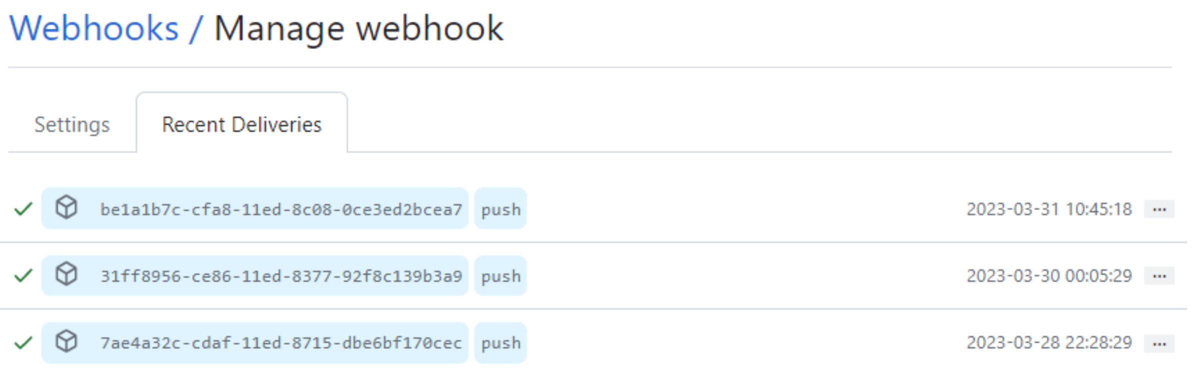
Docker allows an application to be contained, along with all its dependencies so that it can be deployed anywhere. Docker has been used to create container images of both the frontend and backend of the Yarn Swap application.

Red Hat OpenShift allows building and deployment of containerised applications. It is built on top of Kubernetes and allows for added benefits including increased security and a user-friendly UI. Kubernetes is the captain of the ship - it manages the containerised application.

In planning this project, the intention was to create a CI/CD pipeline using the open-source automation server, Jenkins. A number of decisions along the way changed this route and instead, the following CI/CD process was put into place.

Docker was used to manually create initial frontend and backend application images that were pushed to the Quay.io image repository; Webhooks were then configured through the respective GitHub repositories so that new image/s are built by Quay.io when a commit is pushed.

Figure 29 - Backend webhook configured to update Docker image via Quay.io



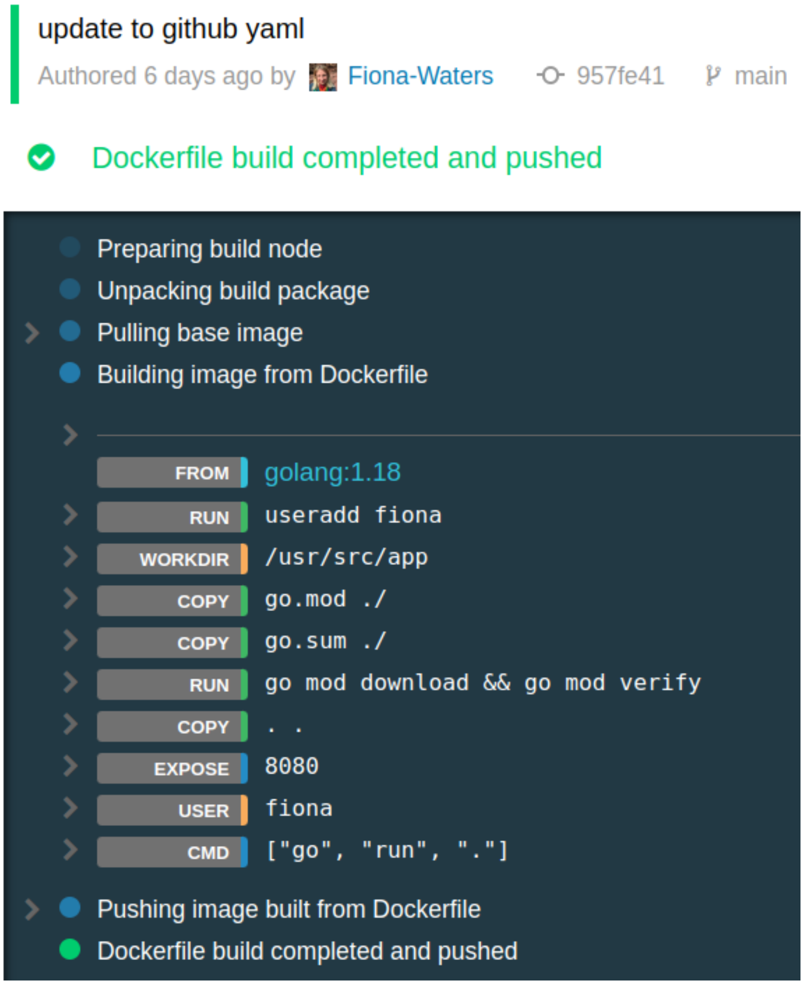
Webhooks / Manage webhook

✓ be1a1b7c-cfa8-11ed-8c08-0ce3ed2bcea7 push 2023-03-31 10:45:18 ...

✓ 31ff8956-ce86-11ed-8377-92f8c139b3a9 push 2023-03-30 00:05:29 ...

✓ 7ae4a32c-cdaf-11ed-8715-dbe6bf170cec push 2023-03-28 22:28:29 ...

Figure 30 - Quay.io building new image after push to GitHub repo



update to github yaml

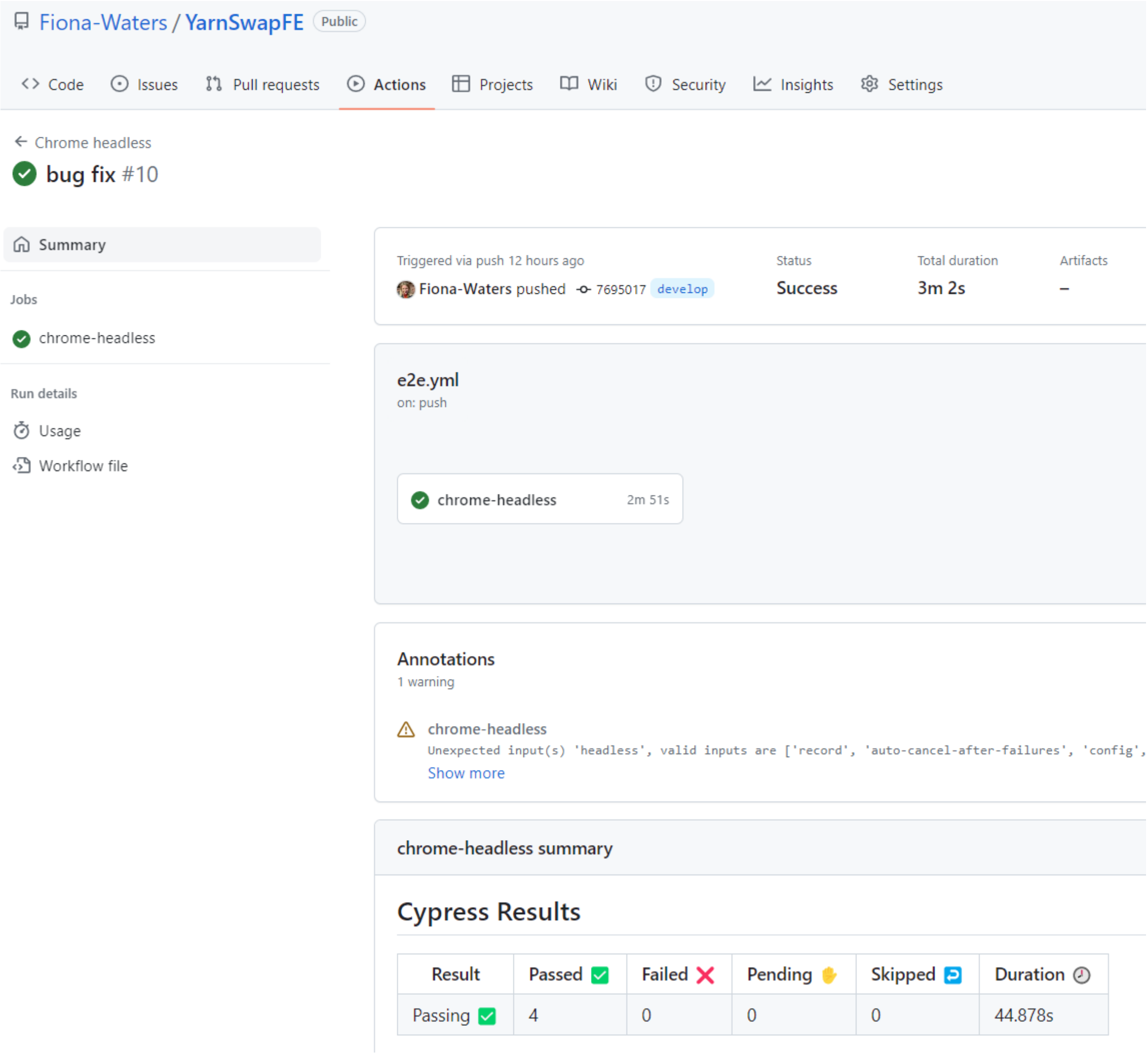
Authored 6 days ago by Fiona-Waters 957fe41 main

Dockerfile build completed and pushed

```
Preparing build node
Unpacking build package
Pulling base image
Building image from Dockerfile

FROM     golang:1.18
RUN      useradd fiona
WORKDIR  /usr/src/app
COPY     go.mod ./
COPY     go.sum ./
RUN      go mod download && go mod verify
COPY     . .
EXPOSE   8080
USER     fiona
CMD      ["go", "run", "."]

Pushing image built from Dockerfile
Dockerfile build completed and pushed
```

In addition, GitHub actions have been created for both project repos. On the backend this allows for the API endpoints to be tested. This is triggered when a new commit is pushed to the repository. On the frontend this allows for some Cypress End to End tests to run, again triggered by a push to the repo.

Figure 31 – GitHub Action running Cypress Tests on Push



bug fix #10

Triggered via push 12 hours ago
Fiona-Waters pushed 7695017 develop

Status: Success
Total duration: 3m 2s
Artifacts: –

e2e.yml
on: push

chrome-headless 2m 51s

Annotations
1 warning

chrome-headless

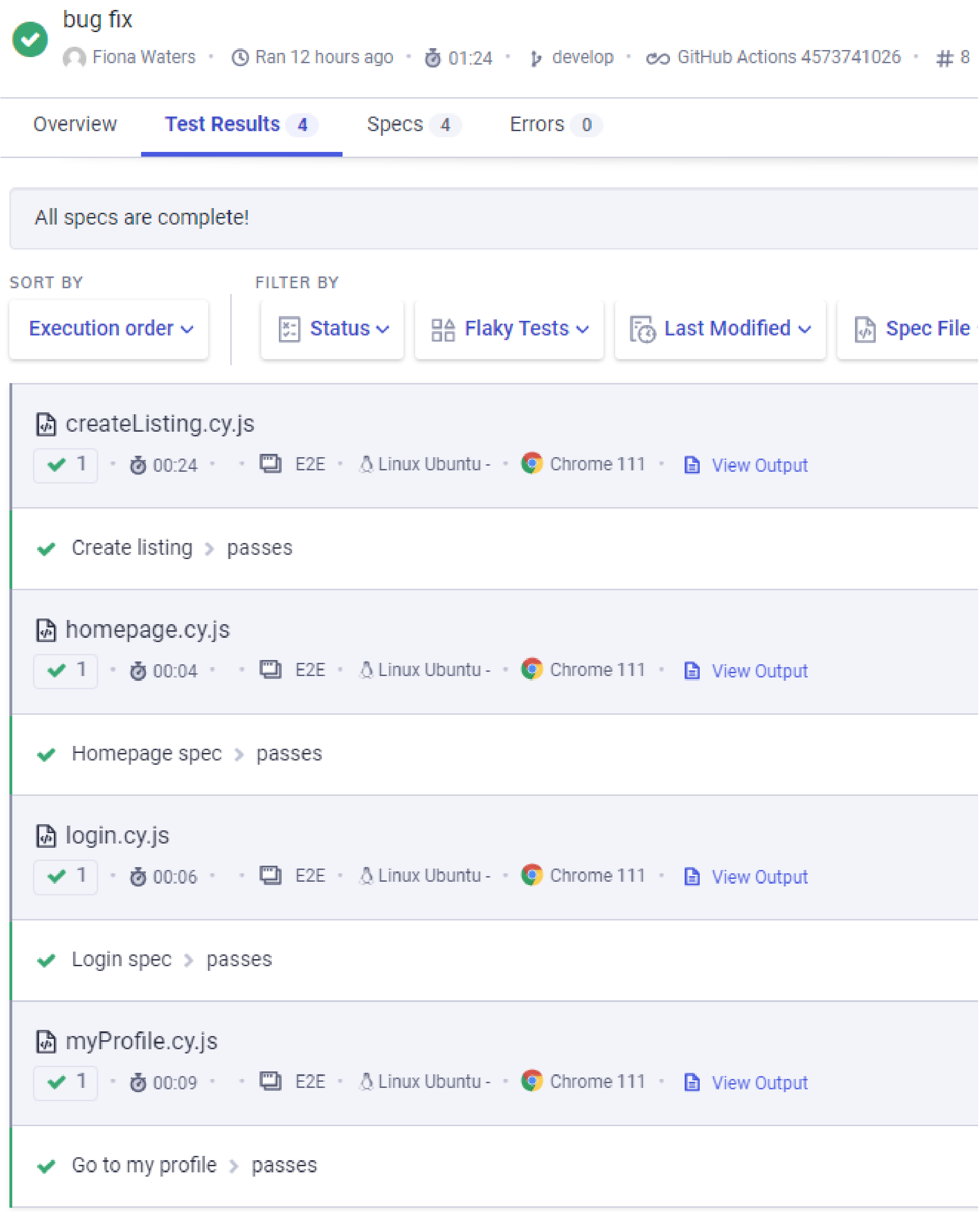
```
Unexpected input(s) 'headless', valid inputs are ['record', 'auto-cancel-after-failures', 'config',
```

chrome-headless summary

Cypress Results

Result	Passed ✅	Failed ❌	Pending ✋	Skipped ↩	Duration 🕓
Passing ✅	4	0	0	0	44.878s

Figure 32 – Corresponding Test Results on Cypress Dashboard



bug fix

Fiona Waters • Ran 12 hours ago • 01:24 • develop • GitHub Actions 4573741026 • # 8

All specs are complete!

createListing.cy.js
✓ 1 • 00:24 • • E2E • Linux Ubuntu - • Chrome 111 • View Output

✓ Create listing > passes

homepage.cy.js
✓ 1 • 00:04 • • E2E • Linux Ubuntu - • Chrome 111 • View Output

✓ Homepage spec > passes

login.cy.js
✓ 1 • 00:06 • • E2E • Linux Ubuntu - • Chrome 111 • View Output

✓ Login spec > passes

myProfile.cy.js
✓ 1 • 00:09 • • E2E • Linux Ubuntu - • Chrome 111 • View Output

✓ Go to my profile > passes

These items all contribute towards the creation of a continuous integration and continuous deployment environment.

To complete the process and ensure automation, an Image Stream Tag can be created in the pod attached to the deployment on the Red Hat OpenShift Cluster which tracks the image and updates it. However, currently if the tests attached to the GitHub action fail, quay.io will still build a new docker image of the application, which would then be used to re-deploy the application. Given more time, further research would have been carried out in this area to correct the issue, therefore at this time the image update on cluster is not automated but can be updated manually.

6 Reflection

6.1 Problems Encountered

Testing

It was my intention to follow the TDD (Test Driven Development) model and when I started creating frontend React components, I also created a Storybook Story for each one. Unfortunately, this did not last too long as worry that I would not complete the project in time got to me and I decided to just push ahead with the main functionality. This means that only some components have corresponding Storybook files.

When it came near the end of the allocated time for the project I looked again at testing. I had thought that testing the backend, and the Firebase functionality would be straight forward but after some research it became clear that the testing of functions for Realtime Database would be extremely difficult and that testing of the use Cloud Firestore would be more accessible with the existence of functions that allow mocking of the platform functions. I investigated the possibility of moving from Realtime Database to Cloud Firestore but came to the conclusion that it would waste too much time.

Instead, I set about testing the API endpoints via the Gin Framework. This turned out to be straightforward and covered what I wanted to test. I was able to set up a GitHub action to run these tests each time changes are pushed to the GitHub repo.

In terms of frontend testing, I decided use Cypress, which is a front end testing tool. I looked first at unit tests but it became clear that I did not have time to complete enough unit tests to properly test the code base. I looked then at Cypress E2E testing and was able to complete a sample set of tests that ran through a couple of the applications processes. A GitHub action has also been set up to run these tests each time changes are pushed to the GitHub repo.

Solution: From this I have learned to make testing a priority at the start of the process.

CI/CD Pipeline - Jenkins

It was my intention to create a CI/CD pipeline that would run on the OpenShift Cluster on a Jenkins instance. Honestly, I was quite worried about this and was not confident that I would be able to do it. As it turned out the technologies I was using as part of this process could do almost all of the things I needed, with some others added in, GitHub webhooks etc.

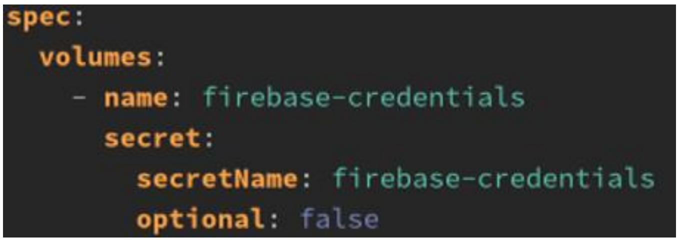
Solution: The solution in this case was to use other more recent technologies that require less management than running and maintaining a Jenkins instance and pipeline. I would also say that conducting more thorough research into such things at the beginning of any future project, might alleviate such issues in the future. I am disappointed that I did not get to complete what I had intended but am happy with the resulting outcome.

Adding Firebase Credentials to OpenShift Cluster

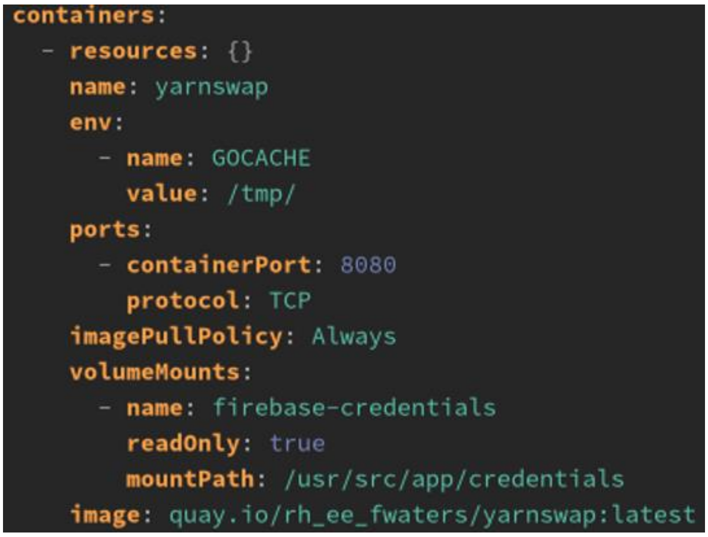
When researching the steps to deploy the application on the OpenShift cluster, the process of adding the secret containing the Firebase Credentials seemed straightforward but when I followed this process, the application would not run as expected and the following error was outputted in the backend pod logs:

> Error initialising database client: cannot read credentials file: open yarnswap-firebase.json: no such file or directory

The firebase credentials could not be found. Following some research via official Red Hat OpenShift documentation I found that containers are not persistent by default and on restart their contents are cleared. I needed to attach a volume to the container so that the credentials would not be lost. This is done in the deployment yaml file like this:



```yaml
spec:
  volumes:
    - name: firebase-credentials
      secret:
        secretName: firebase-credentials
        optional: false
```



```yaml
containers:
  - resources: {}
    name: yarnswap
    env:
      - name: GOCACHE
        value: /tmp/
    ports:
      - containerPort: 8080
        protocol: TCP
    imagePullPolicy: Always
    volumeMounts:
      - name: firebase-credentials
        readOnly: true
        mountPath: /usr/src/app/credentials
    image: quay.io/rh_ee_fwaters/yarnswap:latest
```

According to Kubernetes.io (2023) a volume is ‘directory containing data, accessible to the containers in a pod’.

‘This is a directory on disk or in another container. For any kind of volume in a given pod, data is preserved across container restarts.’

Solution: ‘To use a volume, specify the volumes to provide for the Pod in .spec.volumes and declare where to mount those volumes into containers in .spec.containers[*].volumeMounts’.

I tried this approach as per the Kubernetes documentation and thankfully it worked, a solution was found. Following this update the application deployment completed successfully and was accessible as required.

6.2 Learning

Personal

Backend Development

I preferred working on the backend development. While I like the concept of being able to visually see something tangible and how it looks etc while creating/writing it, I enjoyed creating the backend API for this project more. This may be because I have been doing this kind of work and working with Golang on work placement, I have less experience in working with React. It will be interesting to see where this learning leads me.

Planning

I love to plan, but I underestimated the level of planning that was required for this type of project. I was used to working on module assignments where we work from a base of labs, starting from scratch as a one woman band was a very different story. In hindsight I should have earmarked an entire week for planning and research at the beginning of the first sprint. I was disappointed to not have adequate testing included in my project which may have been different had I planned more thoroughly. Testing is a must! It is not an option to wait until the end of a project to add tests, this needs to be done at the start. I will know to prioritise this for future projects also.

Time

Time management has been so important over the course of this project and the course of study in general, I am hoping that I have learned something about this that I can take with me into the future!

Explanation

I learned a lot simply by having to explain what I was implementing. It is one thing to be able to complete a task but another thing entirely to be able to explain how it works to someone else, or in a written report.

Technical

Webhooks

I had used one or two webhooks previously but had not realised how powerful they are. Having the ability to connect technologies and platforms together in this way was a key element to the success of the deployment and CI/CD area of my project. I plan to use many more webhooks in the future.

GitHub Actions

Being able to connect and run my testing suites using GitHub actions was also a revelation. I know that there is a world of features in this area, and I look forward to delving into them and learning more about what they can do.

React

I had used React for the Movies App Assignment for the ICT Skills module last summer but did not have any experience of using it other than this. This turned out to be a massive learning curve, and even so I have only hit the tip of the React iceberg. I found it to be quite complex and complicated at times and sometimes spent too much time trying to get to the bottom of things. I am not sure what the solution to this is, other than spending a prolonged period getting used to React and its intricacies.

Cypress

I had never used Cypress for testing before but had heard good things about it. I am so glad that I decided to try it, it is very intuitive and satisfying to watch the tests run – as long as they are passing! I will look to use Cypress more in the future.

Wireframes

I enjoyed working with Balsamiq to create wireframe screen mock-ups for Yarn Swap. This is not something I had done before, as in other assignments it was not a requirement, but enjoyed learning about this side of things. It was a nice gentle process in contrast to a lot of the other work involved in this project.

OpenShift Deployment

This was a massive learning curve but an enjoyable one. I have used OpenShift a lot on work placement but not specifically for setting up deployments manually. This allowed me to learn more about its inner workings and to understand more clearly what I am working on while on work placement. This is invaluable learning.

6.3 Achievements

I achieved my goal for this project for the most part. While some parts were implemented differently than planned, and there are many improvements that could be made, I am happy and proud of the outcome.

I was able to achieve the development of a Progressive Web App created using React and Chakra UI that would allow the craft community to share yarn with each other. I was able to create a backend API using Golang and the Gin Framework that speaks to a Firebase Database in order to store and retrieve data. I was also able to create a CI/CD plan that allows for the app to be updated and deployed with few manual steps, which is something that could be improved on in the future.

6.4 Future Development

The following items could be added to Yarn Swap in the future to enhance its capabilities:

- Wishlist functionality where users can add their favourite listings, to contemplate the swap before proceeding with it. This could also involve the categorisation of listings into potential craft projects.
- Forum functionality where users can discuss relevant topics and ask for advice on all things yarn and related projects. This is something that may be difficult to implement in terms of monitoring and moderation. This would require a lot of planning and forethought.
- User Profile – add dark/light mode as a user profile setting allowing the user to persist this setting.
- Add a reCAPTCHA on adding username to ensure the user is a human.
- Use MQTT or similar publish/subscribe mechanism, or email to notify users of swap updates etc.
- Tokens – add the option to purchase a token for a small fee if the user does not have any tokens. This allows for larger community inclusivity.
- Loyalty – add a loyalty service where a user receives an extra token each time, they upload five items that are swappable, for example. This would need more thought to determine specifics.
- User ratings – This would allow trust to be built between users and would allow a user to prove that they are trustworthy by being a responsible community member.
- PWA – to allow for further caching and increased capabilities while the app is offline.

8 Citations

[1] developer.mozilla.org. (n.d.). Cross-Origin Resource Sharing (CORS) [Online]

Available at: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS#:~:text=Cross%2DOrigin%20Resource%20Sharing%20(CORS)%20is%20an%20HTTP%2D.

[2] Firebase. (2019). Choose a Database: Cloud Firestore or Realtime Database [Online]

Available at: https://firebase.google.com/docs/database/rtdb-vs-firestore

[Accessed 2023]

[3] Frost. (2013). Atomic Design. [Online]

Available at: https://bradfrost.com/blog/post/atomic-web-design/

[Accessed 2023]

[4] A.Fu (2023). Vite Plugin PWA. [Online]

Available at: : https://vite-pwa-org.netlify.app/guide/

[Accessed 2023]

[5] GitHub Docs. (2023) Understanding GitHub Actions. [Online]

Available at: https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions

[Accessed 2023]

[6] Javatpoint (2011). Pros and Cons of ReactJS – Javatpoint. [Online]

Available at: https://www.javatpoint.com/pros-and-cons-of-react

[Accessed 2023].

[7] Jenkins.io. (2023). Creating your first Pipeline. [Online] Available at: https://www.jenkins.io/doc/pipeline/tour/hello-world/ [Accessed 2023]

[8] Kubernetes (2019). Production-Grade Container Orchestration. [Online]

Available at: https://kubernetes.io/

[Accessed 2023]

[9] Kubernetes.io. (2019). Volumes. [Online]

Available at: https://kubernetes.io/docs/concepts/storage/volumes/ [Accessed 2023]

[10] MongoDB (2019). NoSQL Databases Explained. [Online]

Available at: https://www.mongodb.com/nosql-explained

[Accessed 2023].

[11] Oracle.com. (2022). What is NoSQL? [Online]

Available at: https://www.oracle.com/ie/database/nosql/what-is-nosql/

[Accessed 2023]

[12] RedHat.com. (2022). What is a CI/CD pipeline? [Online]

Available at: https://www.redhat.com/en/topics/devops/what-cicd-pipeline#:~:text=A%20pipeline%20is%20a%20process,for%20how%20software%20is%20released

[Accessed 2023]

[13] Rehkoph, M. (2019). User Stories | Atlassian. [Online].

Available at: https://www.atlassian.com/agile/project-management/user-stories

[Accessed 2023].

[14] Scarfone, K. (2014). The 4 Factors of Planning a Cloud Deployment. [Online].

Available at: https://edtechmagazine.com/higher/article/2014/11/4-factors-planning-cloud-deployment

[Accessed 2023]

9 References

[1] Chakra UI

https://chakra-ui.com/getting-started

[2] Using ChakraUI with Vite

https://chakra-ui.com/getting-started/vite-guide

[3] How to Set Up a React App with Vite

https://www.digitalocean.com/community/tutorials/how-to-set-up-a-react-project-with-vite

[4] React-router

https://www.freecodecamp.org/news/how-to-use-react-router-version-6/#:~:text=To%20install%20React%20Router%2C%20all,%2Drouter%2Ddom%406%20.

[5] React.js Project Structure

https://www.xenonstack.com/insights/reactjs-project-structure

[6] React-router Layouts

https://reactrouter.com/en/main/start/tutorial

[7] Gin Web Service

https://go.dev/doc/tutorial/web-service-gin

[8] How to Build a Web App with Go, Gin, and React

https://www.freecodecamp.org/news/how-to-build-a-web-app-with-go-gin-and-react-cffdc473576/

[9] CORS gin middleware

https://github.com/gin-contrib/cors

[10] React Query

https://react-query-v3.tanstack.com/quick-start

[11] Retrieving data from Firebase Realtime Database

https://firebase.google.com/docs/database/admin/retrieve-data#ordering-by-key

[12] Database Normalisation

https://www.youtube.com/playlist?list=PLTd6ceoshpreIS_2qHSfKe-iBwW-6eltD

[13] Intro to Firebase Realtime Database

https://firebase.google.com/docs/database#:~:text=The%20Firebase%20Realtime%20Database%20lets,end%20user%20a%20responsive%20experience.

[14] What is NoSQL

https://www.oracle.com/ie/database/nosql/what-is-nosql/

[15] Firebase - Realtime Database VS Firestore

https://firebase.google.com/docs/database/rtdb-vs-firestore

[16] Creating a Jenkins Pipeline

https://www.edureka.co/blog/jenkins-pipeline-tutorial-continuous-delivery

[17] Balsamiq

https://balsamiq.com/

[18] SendBird

https://sendbird.com/

[19] PWA Vite Plugin

https://vite-pwa-org.netlify.app/

[20] firebaseui-web

https://github.com/firebase/firebaseui-web

[21] Go Testing Package

https://pkg.go.dev/testing

[22] Firebase Admin SDK

https://firebase.google.com/docs/admin/setup/#go

[23] Red Hat OpenShift Container Platform Documentation – Persistent Storage

https://docs.openshift.com/container-platform/3.11/dev_guide/volumes.html

[24] Stack Overflow – React Error

https://stackoverflow.com/questions/69719601/getting-error-digital-envelope-routines-reason-unsupported-code-err-oss

[25] Atomic Web Design

https://bradfrost.com/blog/post/atomic-web-design/

[26] Color Mode

https://chakra-ui.com/docs/styled-system/color-mode

[27] Kubernetes Patterns

https://www.redhat.com/en/topics/cloud-native-apps/introduction-to-kubernetes-patterns

[28] Docker Golang Base Image

https://hub.docker.com/_/golang

[29] Go React Docker

https://levelup.gitconnected.com/go-and-react-development-with-docker-pt-1-9f734a5991fc

[30] React Images Uploading Plugin

https://www.npmjs.com/package/react-images-uploading

[31] Gin Middleware - Auth

https://sosedoff.com/2014/12/21/gin-middleware.html
https://chenyitian.gitbooks.io/gin-tutorials/content/tdd/21.html

[32] Gin Testing

https://gin-gonic.com/docs/testing/
https://circleci.com/blog/gin-gonic-testing/

[33] Cypress Testing

https://docs.cypress.io/guides/end-to-end-testing/writing-your-first-end-to-end-test

[34] Dockerizing a React App

https://jsramblings.com/dockerizing-a-react-app/

[35] Web Manifest Generator PWA

https://www.simicart.com/manifest-generator.html/

[36] react-pwa-installer-prompt

https://github.com/shnaveen25/react-pwa-installer-prompt

[37] react-detect-offline

https://www.npmjs.com/package/react-detect-offline

[38] Adding Firebase Credentials to GitHub Action

https://stackoverflow.com/questions/73965176/authenticating-firebase-connection-in-github-action.

[39] Responsive Nav Bar

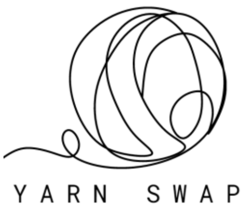
https://www.jimraptis.com/blog/create-a-navbar-with-chakra-ui-react

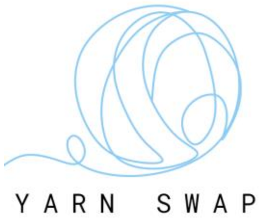
Appendix A – Yarn Swap Logo Evolution

“A logo is an effective way to inform your customers about your company without using any words.” (Goldstein, 2021)

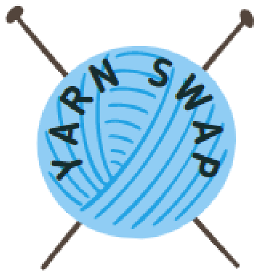
The idea behind the Yarn Swap Logo was a simple one; one that would immediately allow the user to understand what the application is all about. Canva.com was used to create the logos below. Option one includes an abstract ball of yarn and is simple and clean. The yarn ball may not have been completely recognisable, so the colour was changed to make that clearer. Option 2 includes a circle with lines and knitting needles to show that it is a ball of yarn. In all cases the Yarn Swap name is clearly shown.

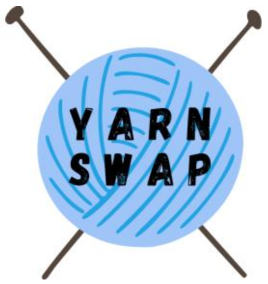
Option one





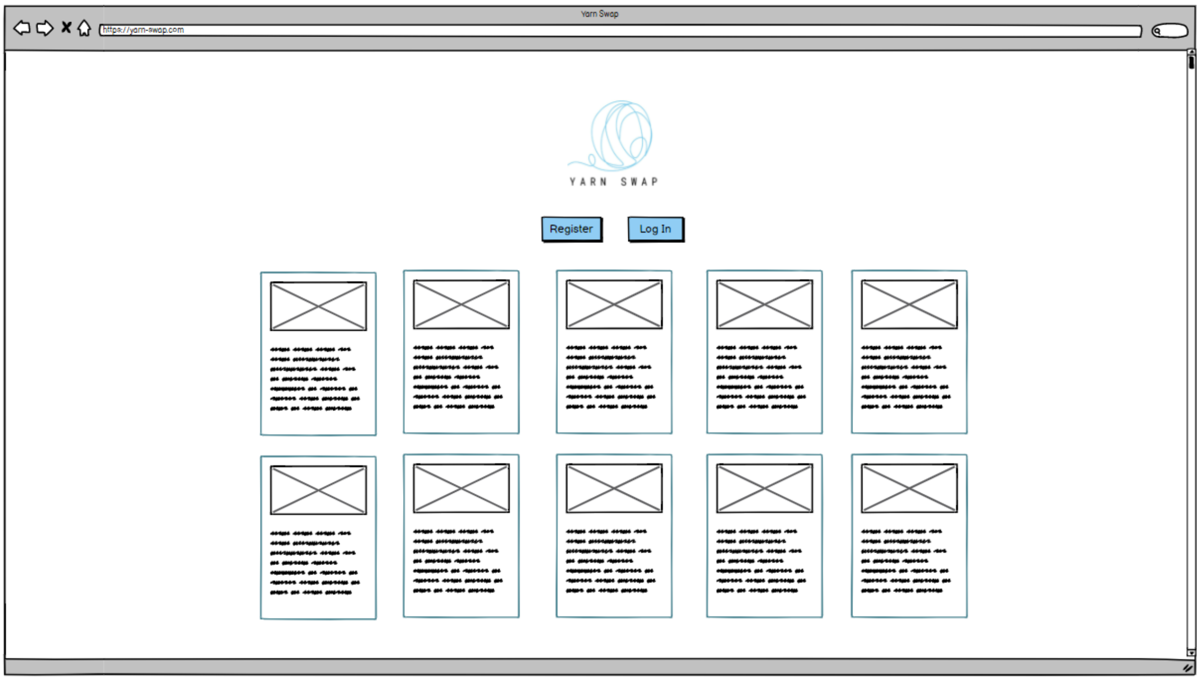
Option two



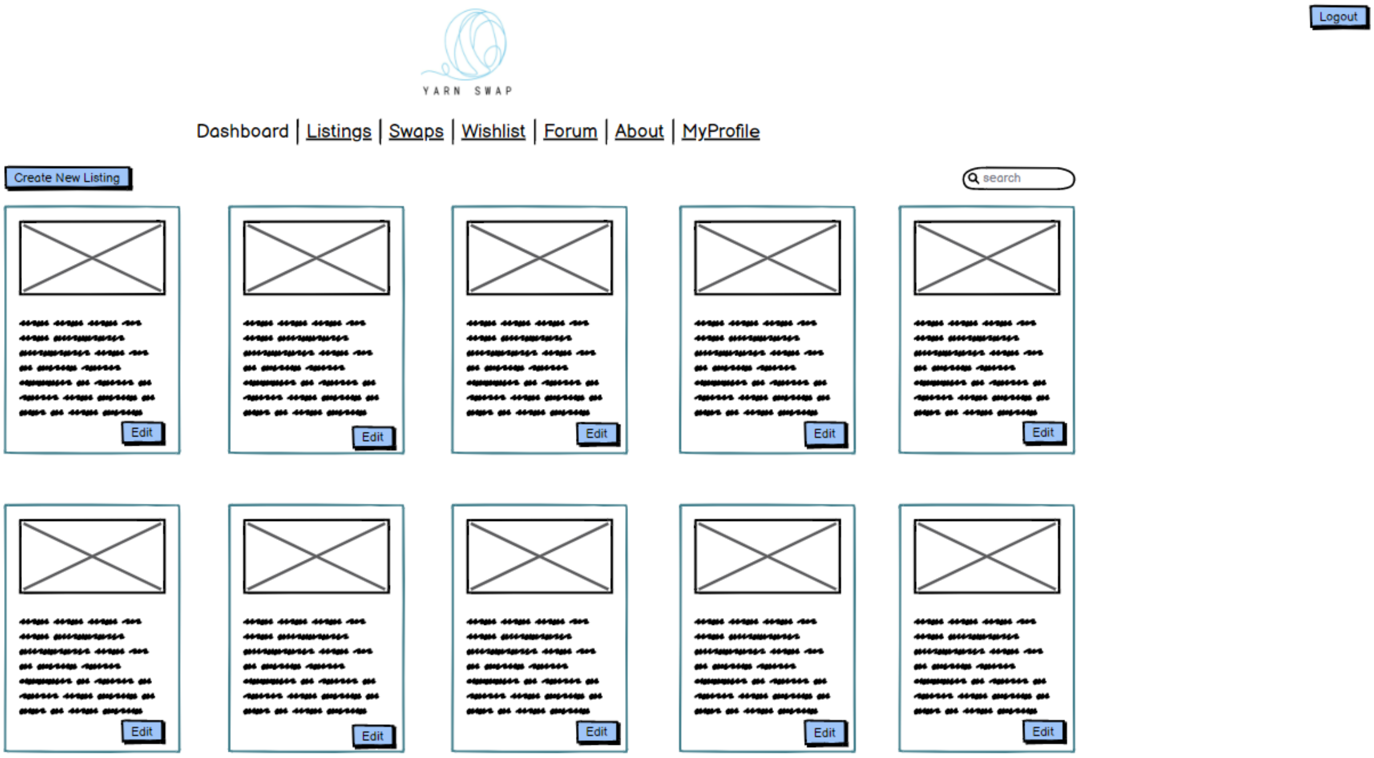


Appendix B – Wireframes

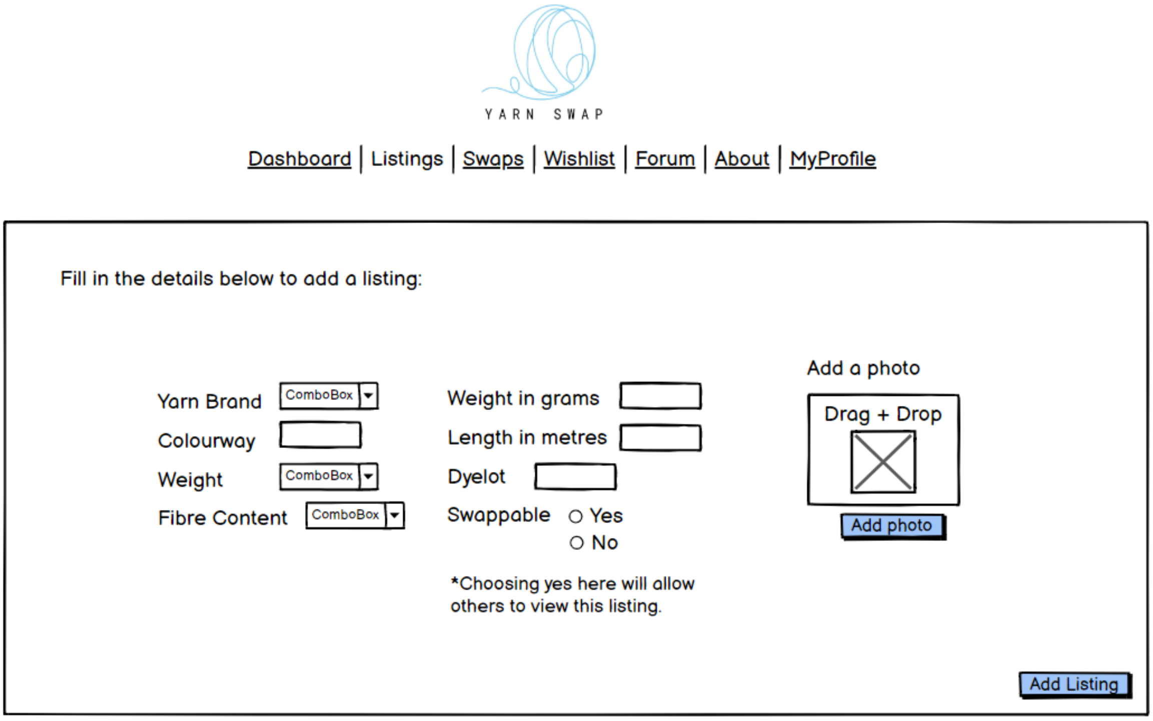
Homepage



User Dashboard


Logout
YARN SWAP
Dashboard | Listings | Swaps | Wishlist | Forum | About | MyProfile
Create New Listing
search
Edit

Create New Listing



YARN SWAP

Dashboard | Listings | Swaps | Wishlist | Forum | About | MyProfile

Fill in the details below to add a listing:

Yarn Brand ComboBox ▾

Colourway

Weight ComboBox ▾

Fibre Content ComboBox ▾

Weight in grams

Length in metres

Dyelot

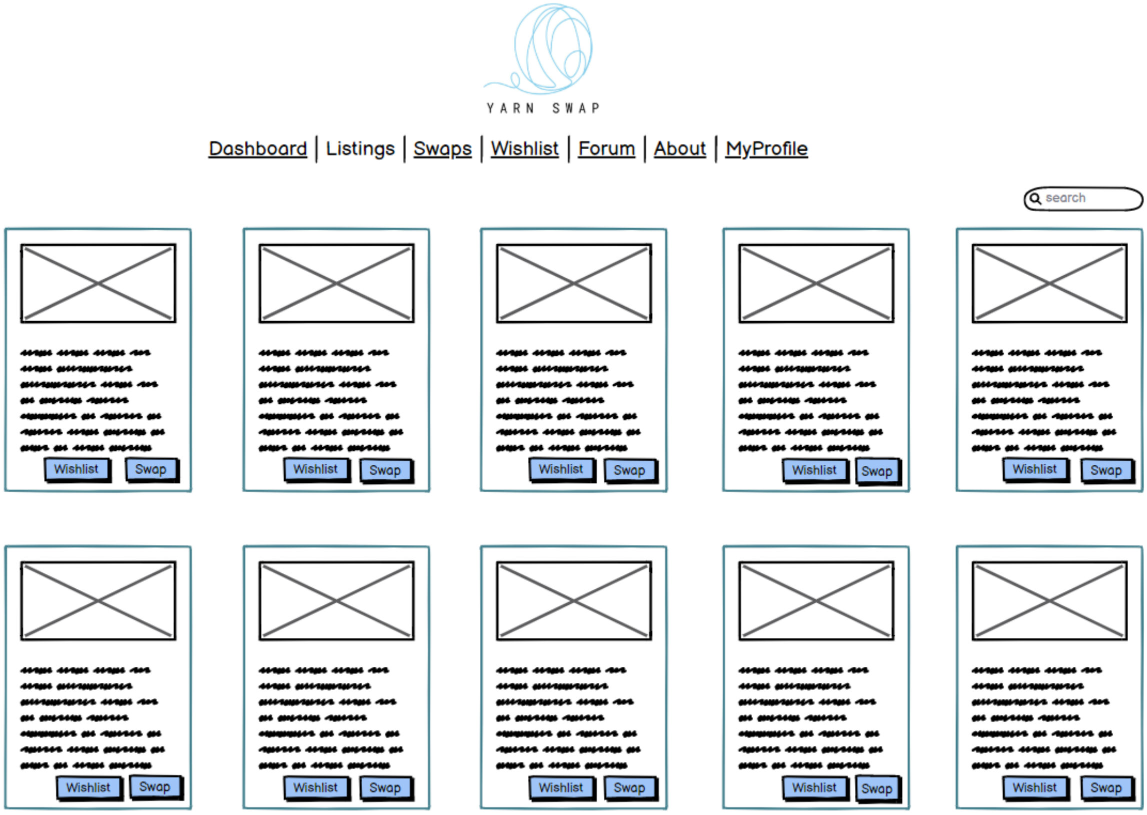
Swappable ○ Yes ○ No

*Choosing yes here will allow others to view this listing.

Add a photo

Drag + Drop

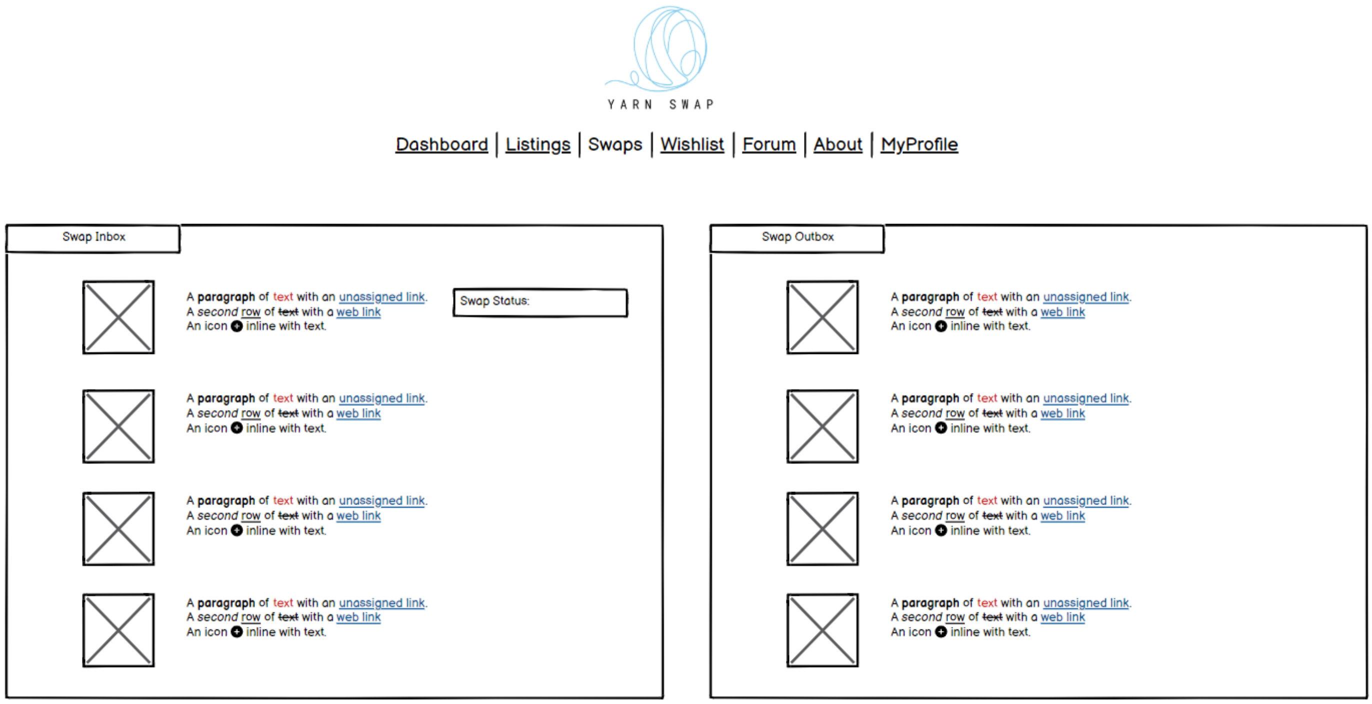
All Listings



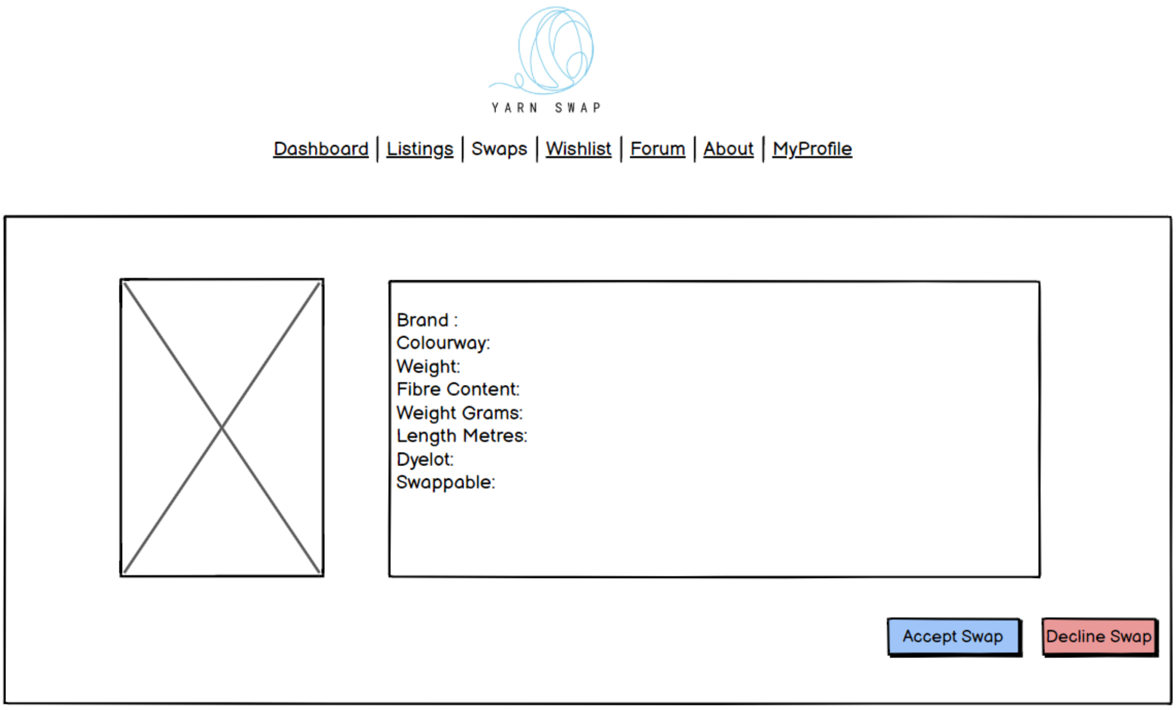
YARN SWAP

Dashboard | Listings | Swaps | Wishlist | Forum | About | MyProfile

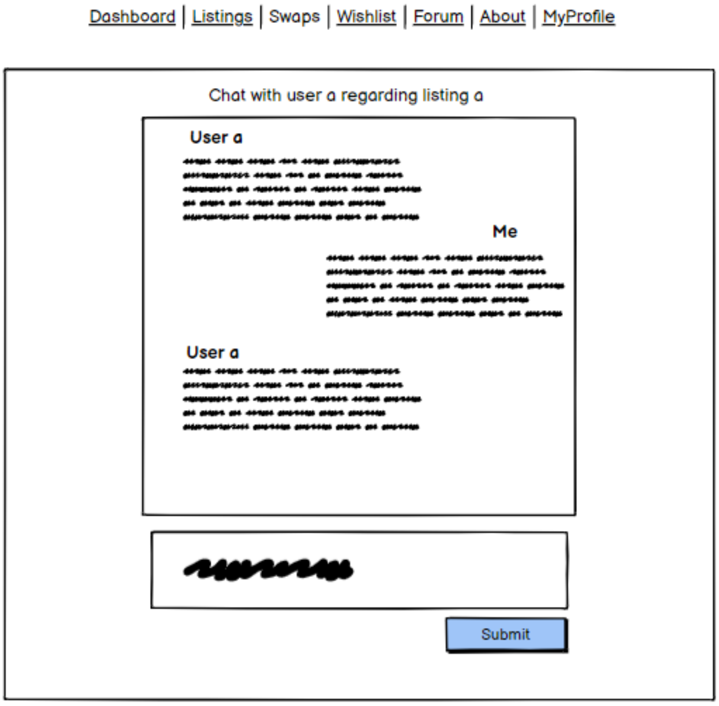
Swaps Page


YARN SWAP
Dashboard | Listings | Swaps | Wishlist | Forum | About | MyProfile
Swap Inbox
A paragraph of text with an unassigned link.
A second row of ~~text~~ with a web link
An icon ● inline with text.
Swap Status:
A paragraph of text with an unassigned link.
A second row of ~~text~~ with a web link
An icon ● inline with text.
A paragraph of text with an unassigned link.
A second row of ~~text~~ with a web link
An icon ● inline with text.
A paragraph of text with an unassigned link.
A second row of ~~text~~ with a web link
An icon ● inline with text.
Swap Outbox
A paragraph of text with an unassigned link.
A second row of ~~text~~ with a web link
An icon ● inline with text.
A paragraph of text with an unassigned link.
A second row of ~~text~~ with a web link
An icon ● inline with text.
A paragraph of text with an unassigned link.
A second row of ~~text~~ with a web link
An icon ● inline with text.
A paragraph of text with an unassigned link.
A second row of ~~text~~ with a web link
An icon ● inline with text.

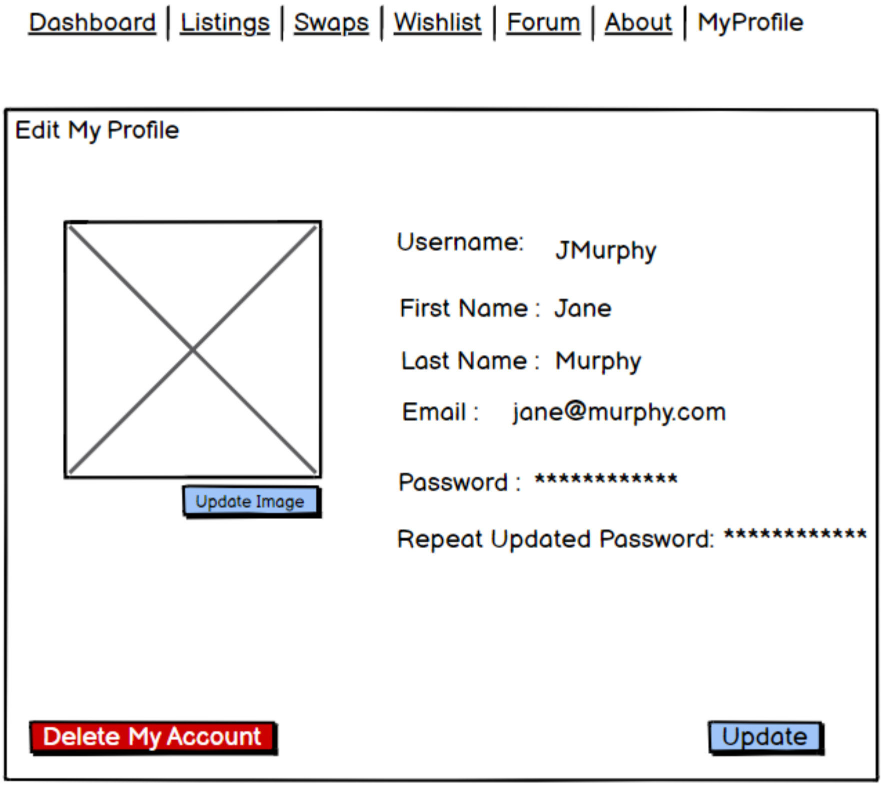
Swap Details



Swap Chat


Dashboard | Listings | Swaps | Wishlist | Forum | About | MyProfile
Chat with user a regarding listing a
User a
Me
User a
Submit

Edit User Profile


Dashboard | Listings | Swaps | Wishlist | Forum | About | MyProfile
Edit My Profile
Update Image
Delete My Account
Update

Username: JMurphy

First Name : Jane

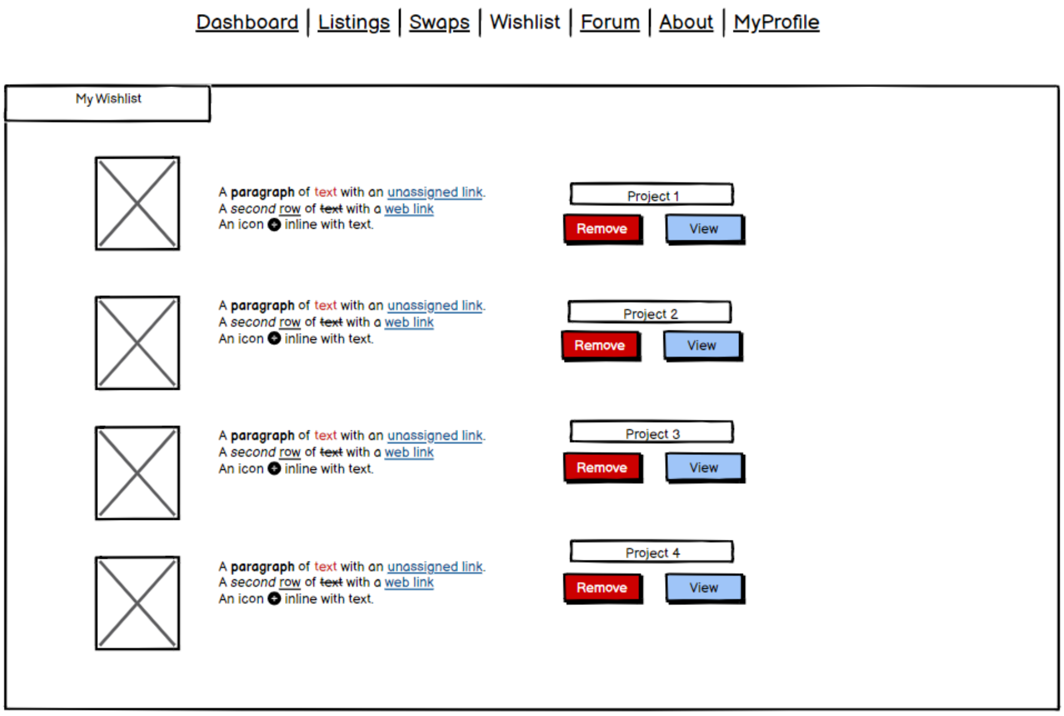
Last Name : Murphy

Email : jane@murphy.com

Password : ************

Repeat Updated Password: ************

Wishlist



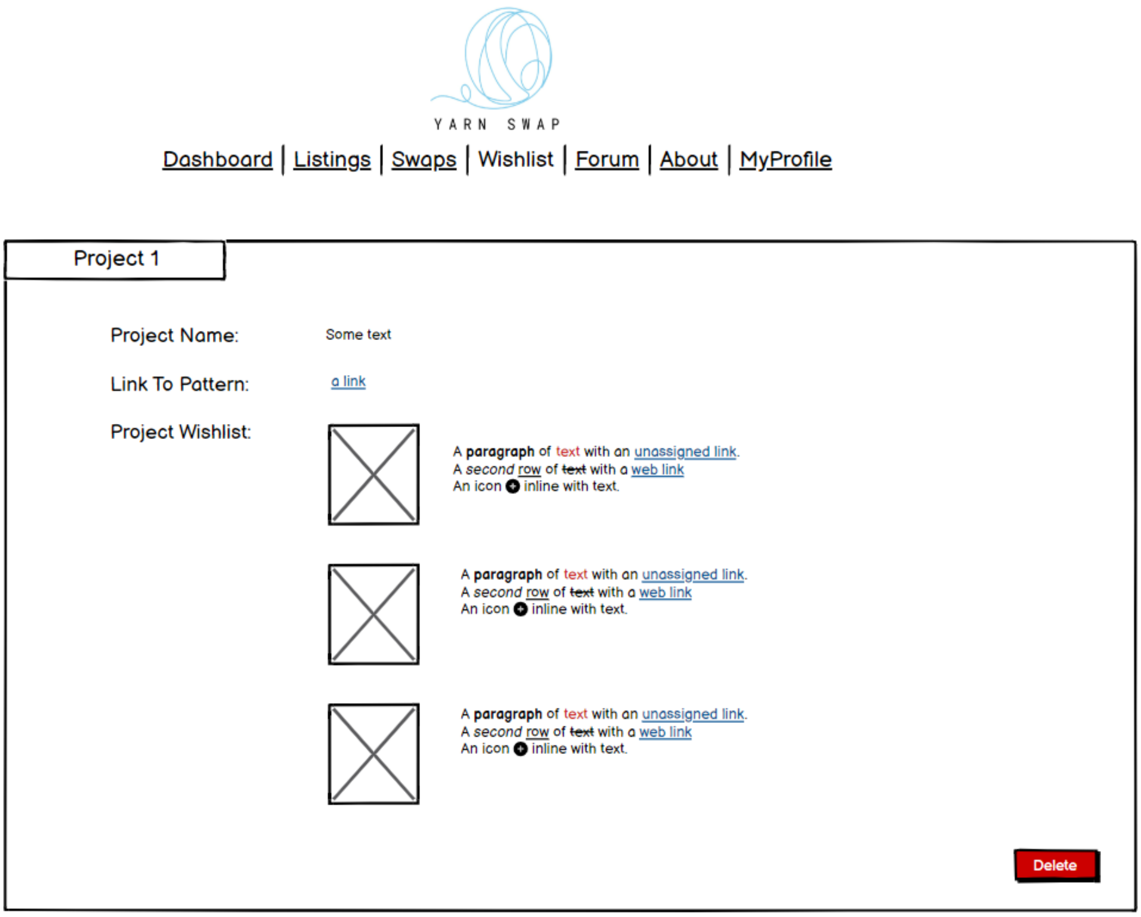
A **paragraph** of text with an unassigned link.
A *second* row of ~~text~~ with a web link
An icon ● inline with text.

A **paragraph** of text with an unassigned link.
A *second* row of ~~text~~ with a web link
An icon ● inline with text.

A **paragraph** of text with an unassigned link.
A *second* row of ~~text~~ with a web link
An icon ● inline with text.

A **paragraph** of text with an unassigned link.
A *second* row of ~~text~~ with a web link
An icon ● inline with text.

Project


YARN SWAP
Dashboard | Listings | Swaps | Wishlist | Forum | About | MyProfile
Project 1
Delete

Project Name: Some text

Link To Pattern: a link

Project Wishlist:

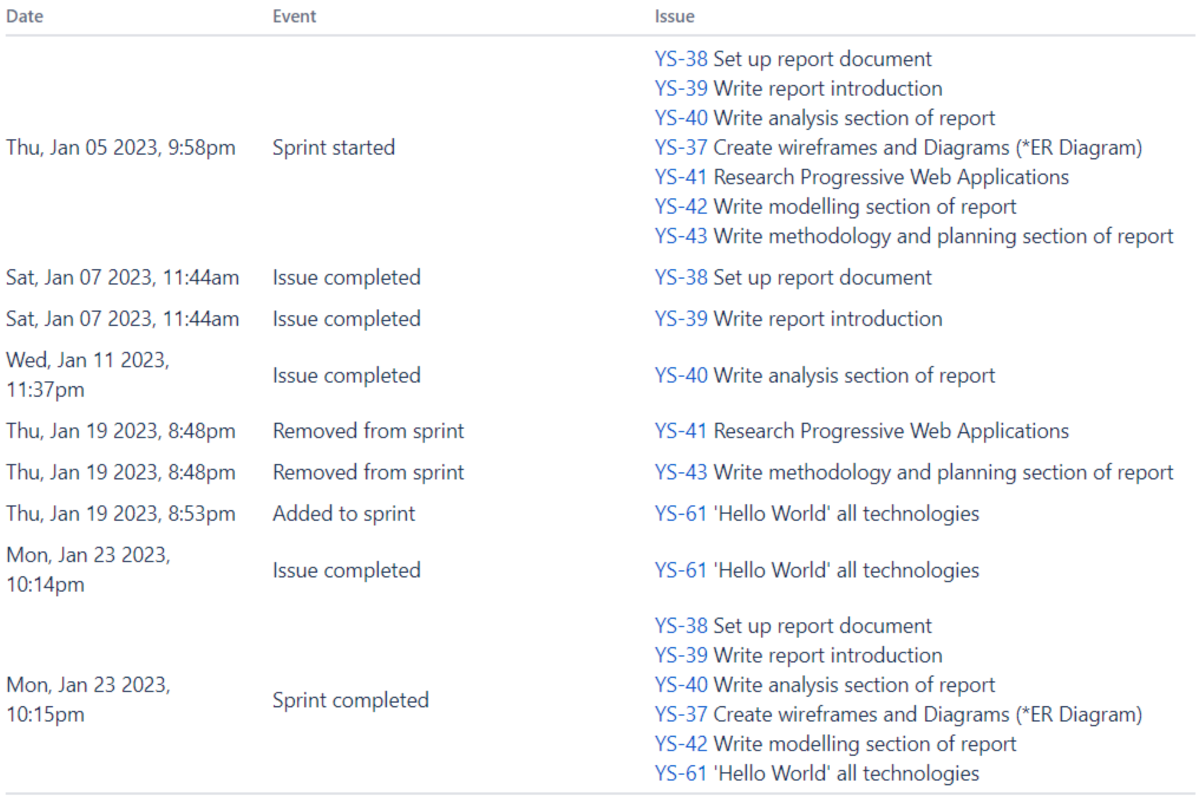
A **paragraph** of text with an unassigned link.
A *second* row of ~~text~~ with a web link
An icon ● inline with text.

A **paragraph** of text with an unassigned link.
A *second* row of ~~text~~ with a web link
An icon ● inline with text.

A **paragraph** of text with an unassigned link.
A *second* row of ~~text~~ with a web link
An icon ● inline with text.

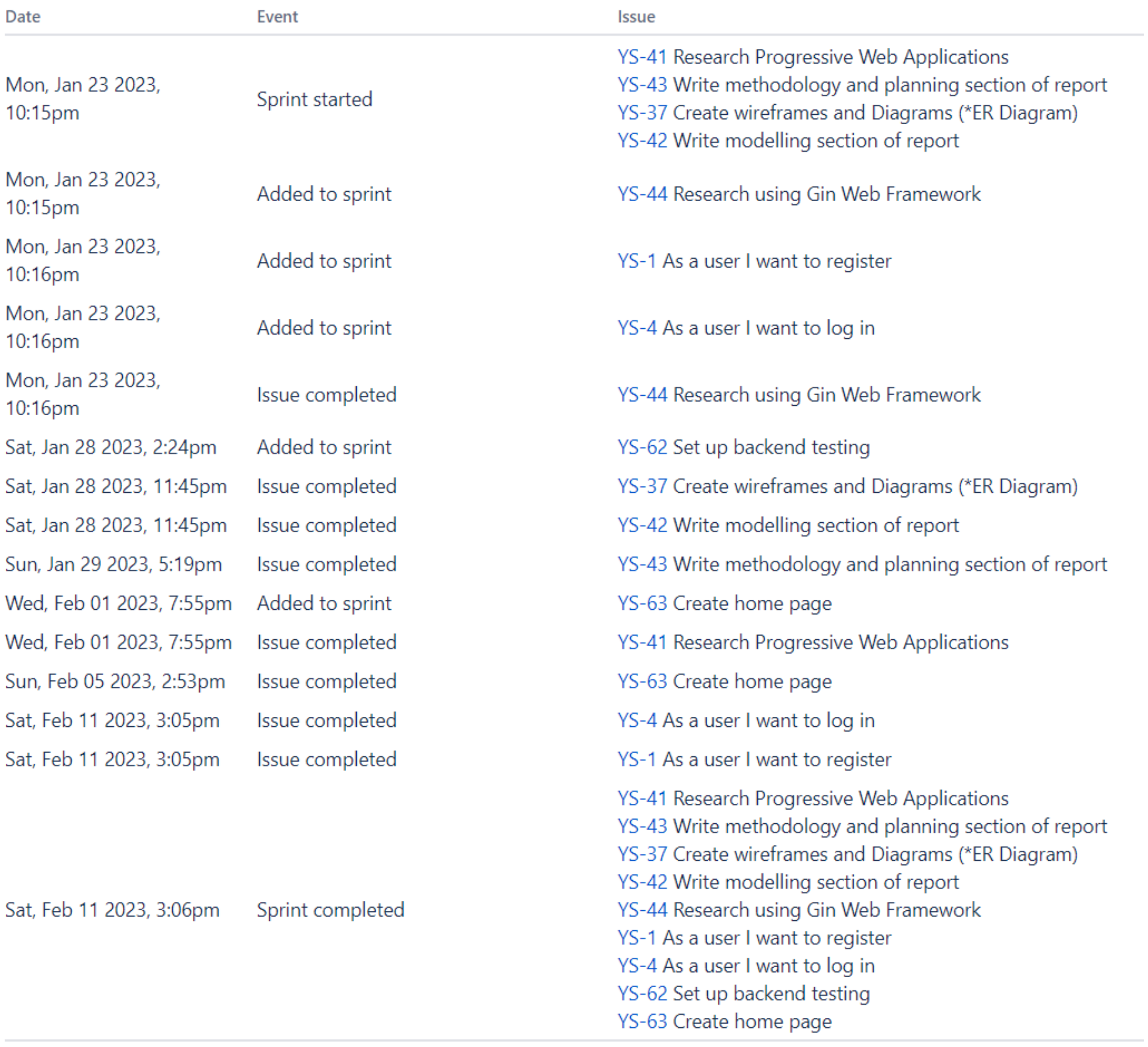
Appendix C – Jira Burnup Report by Sprint

Sprint 1



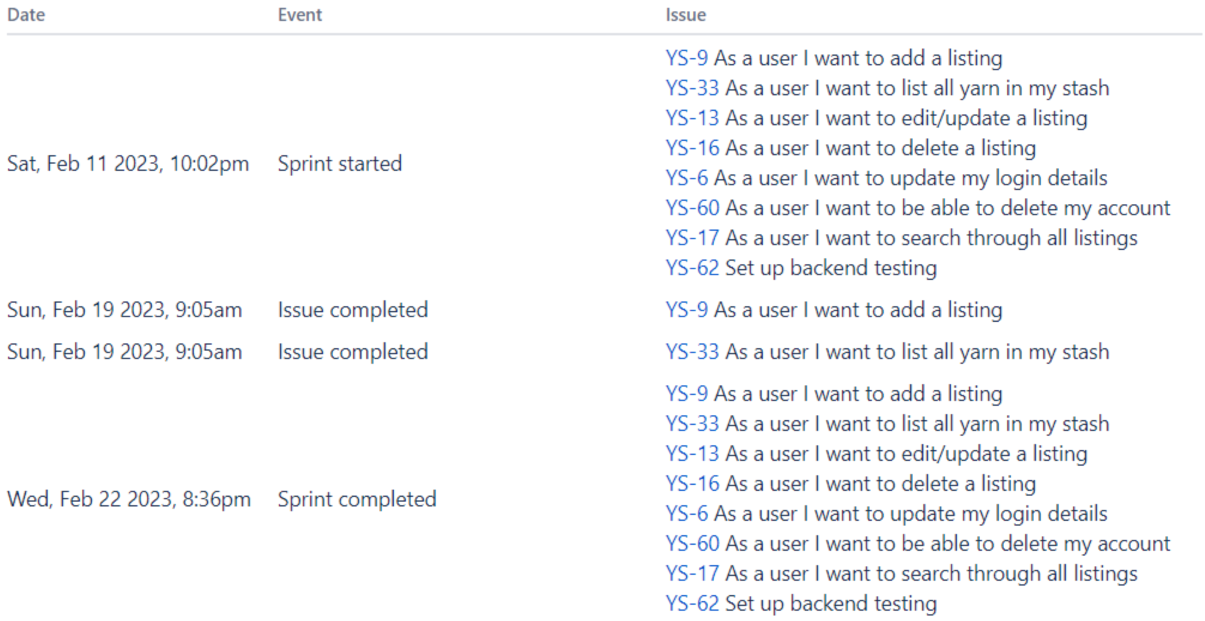
Date	Event	Issue
Thu, Jan 05 2023, 9:58pm	Sprint started	YS-38 Set up report document YS-39 Write report introduction YS-40 Write analysis section of report YS-37 Create wireframes and Diagrams (*ER Diagram) YS-41 Research Progressive Web Applications YS-42 Write modelling section of report YS-43 Write methodology and planning section of report
Sat, Jan 07 2023, 11:44am	Issue completed	YS-38 Set up report document
Sat, Jan 07 2023, 11:44am	Issue completed	YS-39 Write report introduction
Wed, Jan 11 2023, 11:37pm	Issue completed	YS-40 Write analysis section of report
Thu, Jan 19 2023, 8:48pm	Removed from sprint	YS-41 Research Progressive Web Applications
Thu, Jan 19 2023, 8:48pm	Removed from sprint	YS-43 Write methodology and planning section of report
Thu, Jan 19 2023, 8:53pm	Added to sprint	YS-61 'Hello World' all technologies
Mon, Jan 23 2023, 10:14pm	Issue completed	YS-61 'Hello World' all technologies
Mon, Jan 23 2023, 10:15pm	Sprint completed	YS-38 Set up report document YS-39 Write report introduction YS-40 Write analysis section of report YS-37 Create wireframes and Diagrams (*ER Diagram) YS-42 Write modelling section of report YS-61 'Hello World' all technologies

Sprint 2



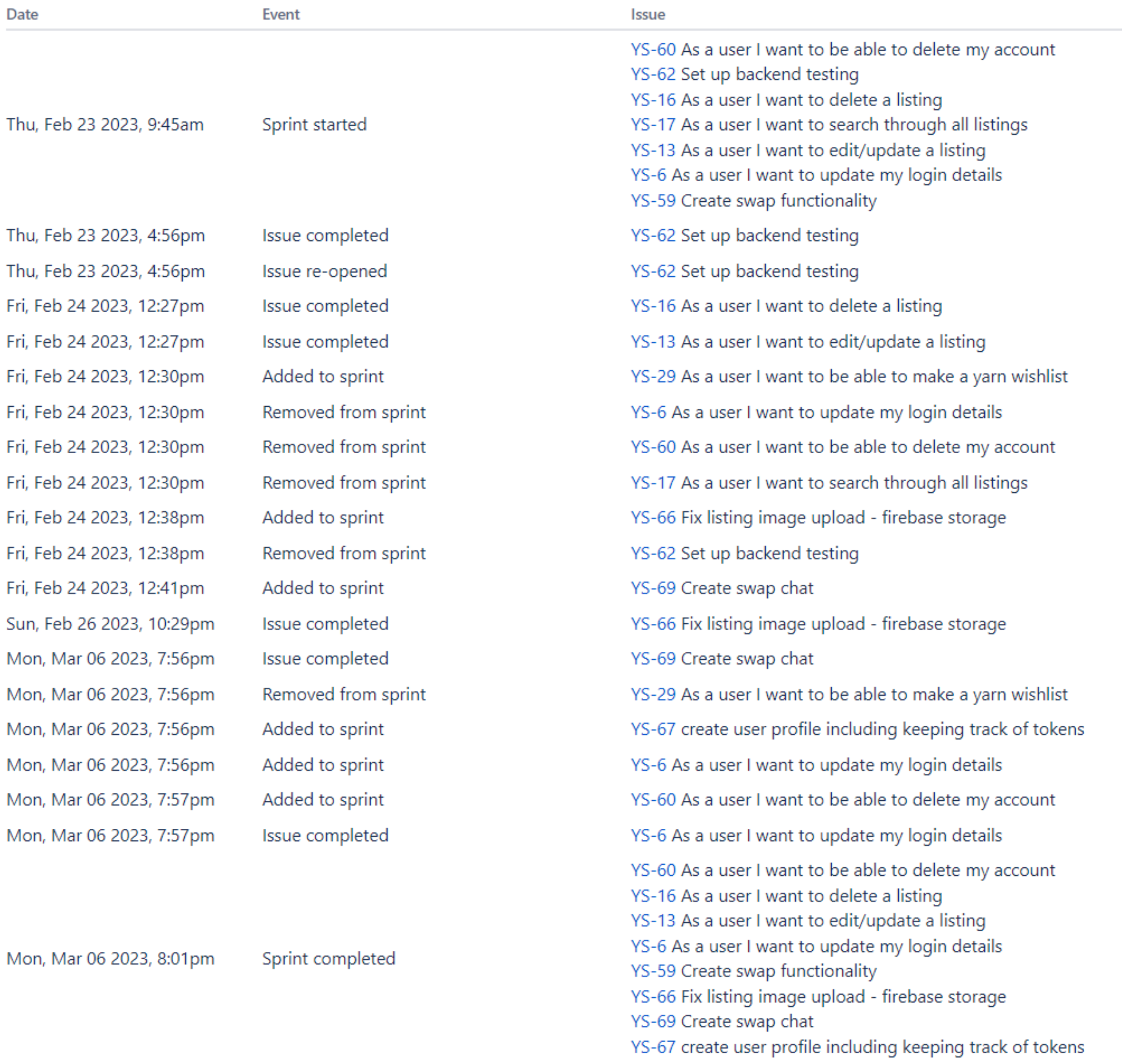
Date	Event	Issue
Mon, Jan 23 2023, 10:15pm	Sprint started	YS-41 Research Progressive Web Applications YS-43 Write methodology and planning section of report YS-37 Create wireframes and Diagrams (*ER Diagram) YS-42 Write modelling section of report
Mon, Jan 23 2023, 10:15pm	Added to sprint	YS-44 Research using Gin Web Framework
Mon, Jan 23 2023, 10:16pm	Added to sprint	YS-1 As a user I want to register
Mon, Jan 23 2023, 10:16pm	Added to sprint	YS-4 As a user I want to log in
Mon, Jan 23 2023, 10:16pm	Issue completed	YS-44 Research using Gin Web Framework
Sat, Jan 28 2023, 2:24pm	Added to sprint	YS-62 Set up backend testing
Sat, Jan 28 2023, 11:45pm	Issue completed	YS-37 Create wireframes and Diagrams (*ER Diagram)
Sat, Jan 28 2023, 11:45pm	Issue completed	YS-42 Write modelling section of report
Sun, Jan 29 2023, 5:19pm	Issue completed	YS-43 Write methodology and planning section of report
Wed, Feb 01 2023, 7:55pm	Added to sprint	YS-63 Create home page
Wed, Feb 01 2023, 7:55pm	Issue completed	YS-41 Research Progressive Web Applications
Sun, Feb 05 2023, 2:53pm	Issue completed	YS-63 Create home page
Sat, Feb 11 2023, 3:05pm	Issue completed	YS-4 As a user I want to log in
Sat, Feb 11 2023, 3:05pm	Issue completed	YS-1 As a user I want to register
Sat, Feb 11 2023, 3:06pm	Sprint completed	YS-41 Research Progressive Web Applications YS-43 Write methodology and planning section of report YS-37 Create wireframes and Diagrams (*ER Diagram) YS-42 Write modelling section of report YS-44 Research using Gin Web Framework YS-1 As a user I want to register YS-4 As a user I want to log in YS-62 Set up backend testing YS-63 Create home page

Sprint 3



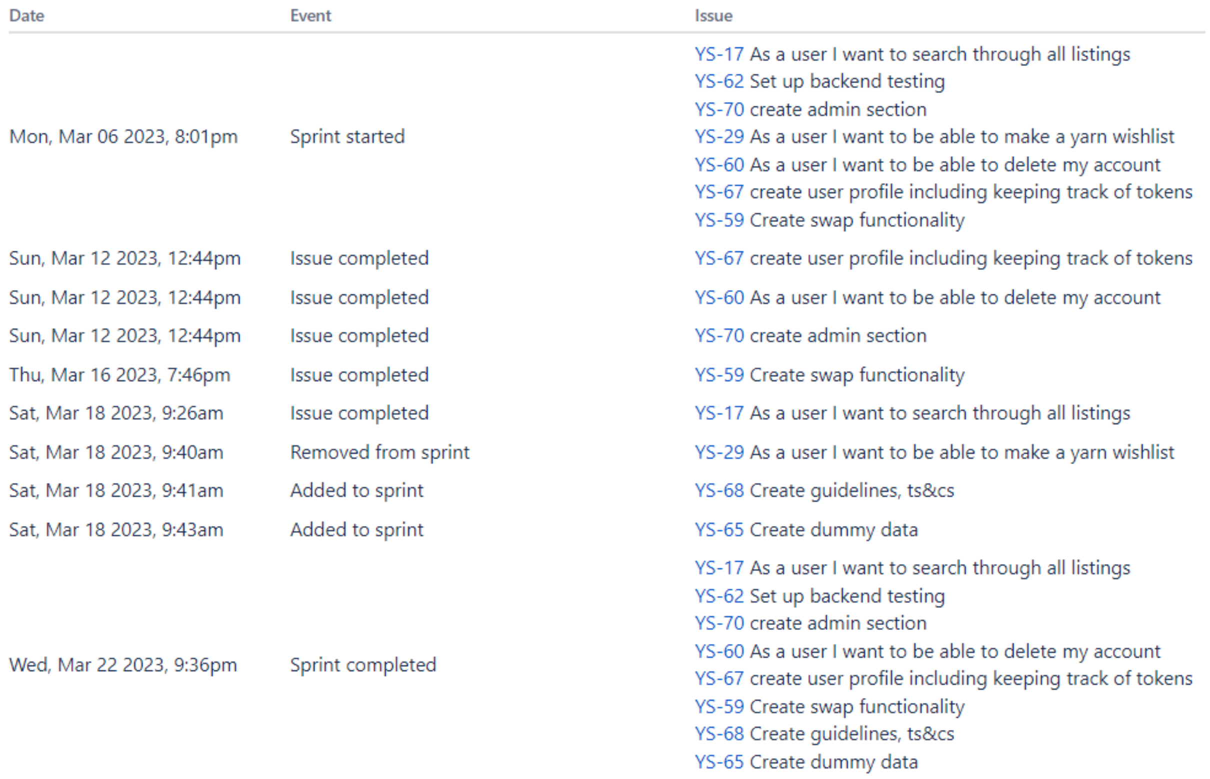
Date	Event	Issue
Sat, Feb 11 2023, 10:02pm	Sprint started	YS-9 As a user I want to add a listing YS-33 As a user I want to list all yarn in my stash YS-13 As a user I want to edit/update a listing YS-16 As a user I want to delete a listing YS-6 As a user I want to update my login details YS-60 As a user I want to be able to delete my account YS-17 As a user I want to search through all listings YS-62 Set up backend testing
Sun, Feb 19 2023, 9:05am	Issue completed	YS-9 As a user I want to add a listing
Sun, Feb 19 2023, 9:05am	Issue completed	YS-33 As a user I want to list all yarn in my stash
Wed, Feb 22 2023, 8:36pm	Sprint completed	YS-9 As a user I want to add a listing YS-33 As a user I want to list all yarn in my stash YS-13 As a user I want to edit/update a listing YS-16 As a user I want to delete a listing YS-6 As a user I want to update my login details YS-60 As a user I want to be able to delete my account YS-17 As a user I want to search through all listings YS-62 Set up backend testing

Sprint 4



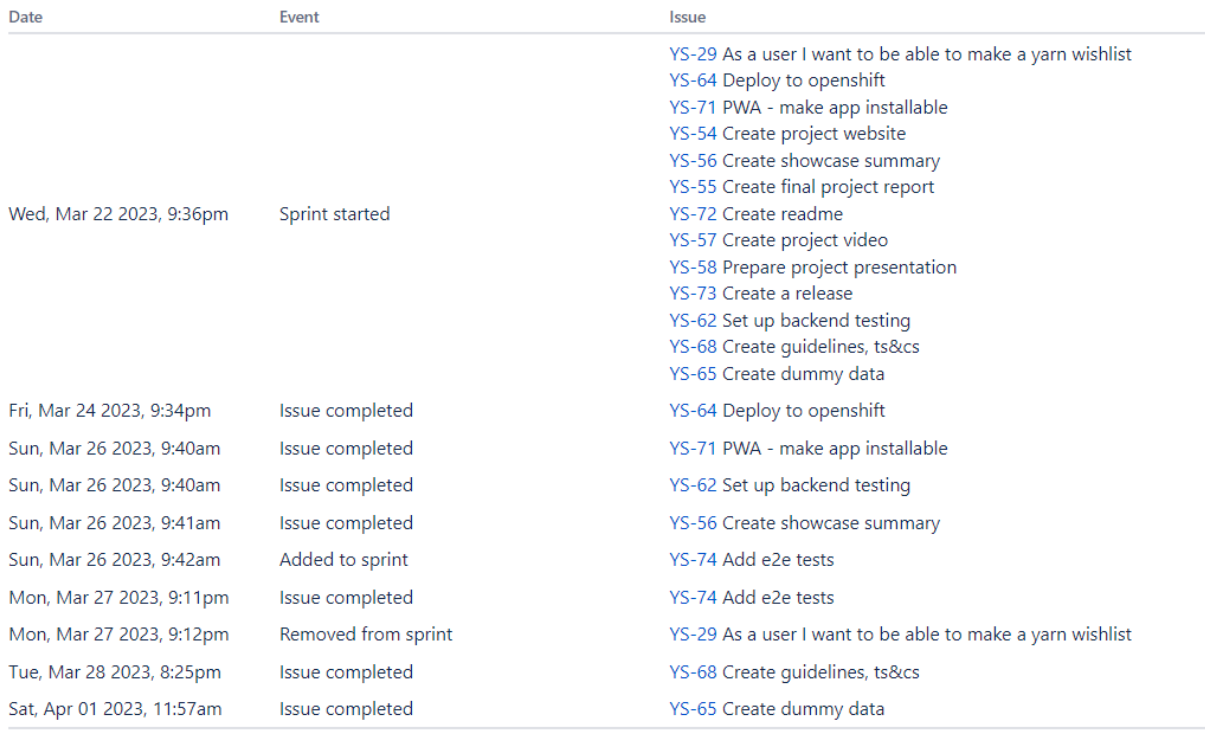
Date	Event	Issue
Thu, Feb 23 2023, 9:45am	Sprint started	YS-60 As a user I want to be able to delete my account YS-62 Set up backend testing YS-16 As a user I want to delete a listing YS-17 As a user I want to search through all listings YS-13 As a user I want to edit/update a listing YS-6 As a user I want to update my login details YS-59 Create swap functionality
Thu, Feb 23 2023, 4:56pm	Issue completed	YS-62 Set up backend testing
Thu, Feb 23 2023, 4:56pm	Issue re-opened	YS-62 Set up backend testing
Fri, Feb 24 2023, 12:27pm	Issue completed	YS-16 As a user I want to delete a listing
Fri, Feb 24 2023, 12:27pm	Issue completed	YS-13 As a user I want to edit/update a listing
Fri, Feb 24 2023, 12:30pm	Added to sprint	YS-29 As a user I want to be able to make a yarn wishlist
Fri, Feb 24 2023, 12:30pm	Removed from sprint	YS-6 As a user I want to update my login details
Fri, Feb 24 2023, 12:30pm	Removed from sprint	YS-60 As a user I want to be able to delete my account
Fri, Feb 24 2023, 12:30pm	Removed from sprint	YS-17 As a user I want to search through all listings
Fri, Feb 24 2023, 12:38pm	Added to sprint	YS-66 Fix listing image upload - firebase storage
Fri, Feb 24 2023, 12:38pm	Removed from sprint	YS-62 Set up backend testing
Fri, Feb 24 2023, 12:41pm	Added to sprint	YS-69 Create swap chat
Sun, Feb 26 2023, 10:29pm	Issue completed	YS-66 Fix listing image upload - firebase storage
Mon, Mar 06 2023, 7:56pm	Issue completed	YS-69 Create swap chat
Mon, Mar 06 2023, 7:56pm	Removed from sprint	YS-29 As a user I want to be able to make a yarn wishlist
Mon, Mar 06 2023, 7:56pm	Added to sprint	YS-67 create user profile including keeping track of tokens
Mon, Mar 06 2023, 7:56pm	Added to sprint	YS-6 As a user I want to update my login details
Mon, Mar 06 2023, 7:57pm	Added to sprint	YS-60 As a user I want to be able to delete my account
Mon, Mar 06 2023, 7:57pm	Issue completed	YS-6 As a user I want to update my login details
Mon, Mar 06 2023, 8:01pm	Sprint completed	YS-60 As a user I want to be able to delete my account YS-16 As a user I want to delete a listing YS-13 As a user I want to edit/update a listing YS-6 As a user I want to update my login details YS-59 Create swap functionality YS-66 Fix listing image upload - firebase storage YS-69 Create swap chat YS-67 create user profile including keeping track of tokens

Sprint 5



Date	Event	Issue
Mon, Mar 06 2023, 8:01pm	Sprint started	YS-17 As a user I want to search through all listings YS-62 Set up backend testing YS-70 create admin section YS-29 As a user I want to be able to make a yarn wishlist YS-60 As a user I want to be able to delete my account YS-67 create user profile including keeping track of tokens YS-59 Create swap functionality
Sun, Mar 12 2023, 12:44pm	Issue completed	YS-67 create user profile including keeping track of tokens
Sun, Mar 12 2023, 12:44pm	Issue completed	YS-60 As a user I want to be able to delete my account
Sun, Mar 12 2023, 12:44pm	Issue completed	YS-70 create admin section
Thu, Mar 16 2023, 7:46pm	Issue completed	YS-59 Create swap functionality
Sat, Mar 18 2023, 9:26am	Issue completed	YS-17 As a user I want to search through all listings
Sat, Mar 18 2023, 9:40am	Removed from sprint	YS-29 As a user I want to be able to make a yarn wishlist
Sat, Mar 18 2023, 9:41am	Added to sprint	YS-68 Create guidelines, ts&cs
Sat, Mar 18 2023, 9:43am	Added to sprint	YS-65 Create dummy data
Wed, Mar 22 2023, 9:36pm	Sprint completed	YS-17 As a user I want to search through all listings YS-62 Set up backend testing YS-70 create admin section YS-60 As a user I want to be able to delete my account YS-67 create user profile including keeping track of tokens YS-59 Create swap functionality YS-68 Create guidelines, ts&cs YS-65 Create dummy data

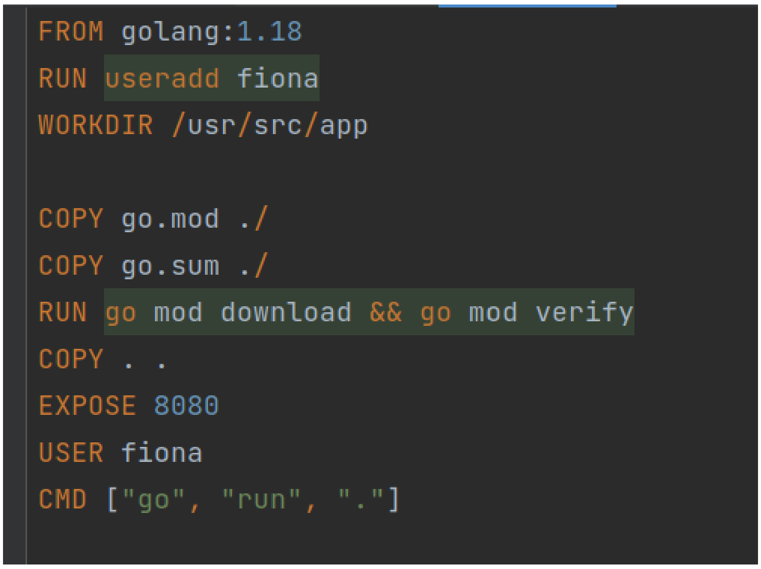
Sprint 6



Date	Event	Issue
Wed, Mar 22 2023, 9:36pm	Sprint started	YS-29 As a user I want to be able to make a yarn wishlist YS-64 Deploy to openshift YS-71 PWA - make app installable YS-54 Create project website YS-56 Create showcase summary YS-55 Create final project report YS-72 Create readme YS-57 Create project video YS-58 Prepare project presentation YS-73 Create a release YS-62 Set up backend testing YS-68 Create guidelines, ts&cs YS-65 Create dummy data
Fri, Mar 24 2023, 9:34pm	Issue completed	YS-64 Deploy to openshift
Sun, Mar 26 2023, 9:40am	Issue completed	YS-71 PWA - make app installable
Sun, Mar 26 2023, 9:40am	Issue completed	YS-62 Set up backend testing
Sun, Mar 26 2023, 9:41am	Issue completed	YS-56 Create showcase summary
Sun, Mar 26 2023, 9:42am	Added to sprint	YS-74 Add e2e tests
Mon, Mar 27 2023, 9:11pm	Issue completed	YS-74 Add e2e tests
Mon, Mar 27 2023, 9:12pm	Removed from sprint	YS-29 As a user I want to be able to make a yarn wishlist
Tue, Mar 28 2023, 8:25pm	Issue completed	YS-68 Create guidelines, ts&cs
Sat, Apr 01 2023, 11:57am	Issue completed	YS-65 Create dummy data

Appendix D – Deployment Procedure

The initial set up for the complete deployment of the Yarn Swap Progressive Web Application involved building frontend and backend application images using Docker. Dockerfiles were added to both repos. These give the instructions needed by Docker to build the images. This is the backend Dockerfile:



```
FROM golang:1.18
RUN useradd fiona
WORKDIR /usr/src/app

COPY go.mod ./
COPY go.sum ./
RUN go mod download && go mod verify
COPY . .
EXPOSE 8080
USER fiona
CMD ["go", "run", "."]
```

Build and Push the Images

On the command line the docker build and docker push commands were used to build the Docker images and push them to the already created Quay.io image repositories.

Provision a Red Hat OpenShift Cluster

The cluster was provisioned via the OpenShift Cluster Manager Command Line Interface (OCM CLI) using the following command along with a cluster configuration JSON file:

```
ocm post /api/clusters_mgmt/v1/clusters --body=$CLUSTER_FILE
```

The configuration file contains information regarding cluster configuration as below. The cluster name, AWS region, number of nodes and machine type is all we need to set initially.

```
{
  "name": "fwaters",
  "region": {
    "id": "eu-west-1"
  },
  "nodes" : {
    "compute" : 4,
    "compute_machine_type" : {"id" : "m5.xlarge" }
  }
}
```

Once the cluster is ready, we can deploy the application.

The application is deployed via the OpenShift Command Line Interface (OC CLI).

Deploy the Backend

On command line. You must be logged in to your cluster.

Create and select a project/namespace where the application will reside:

```
oc new-project yarn-swap
```

Create the application, set an environment variable for saving the GOCACHE, specify the location of the Docker image and specify the namespace where you want the application to be placed.

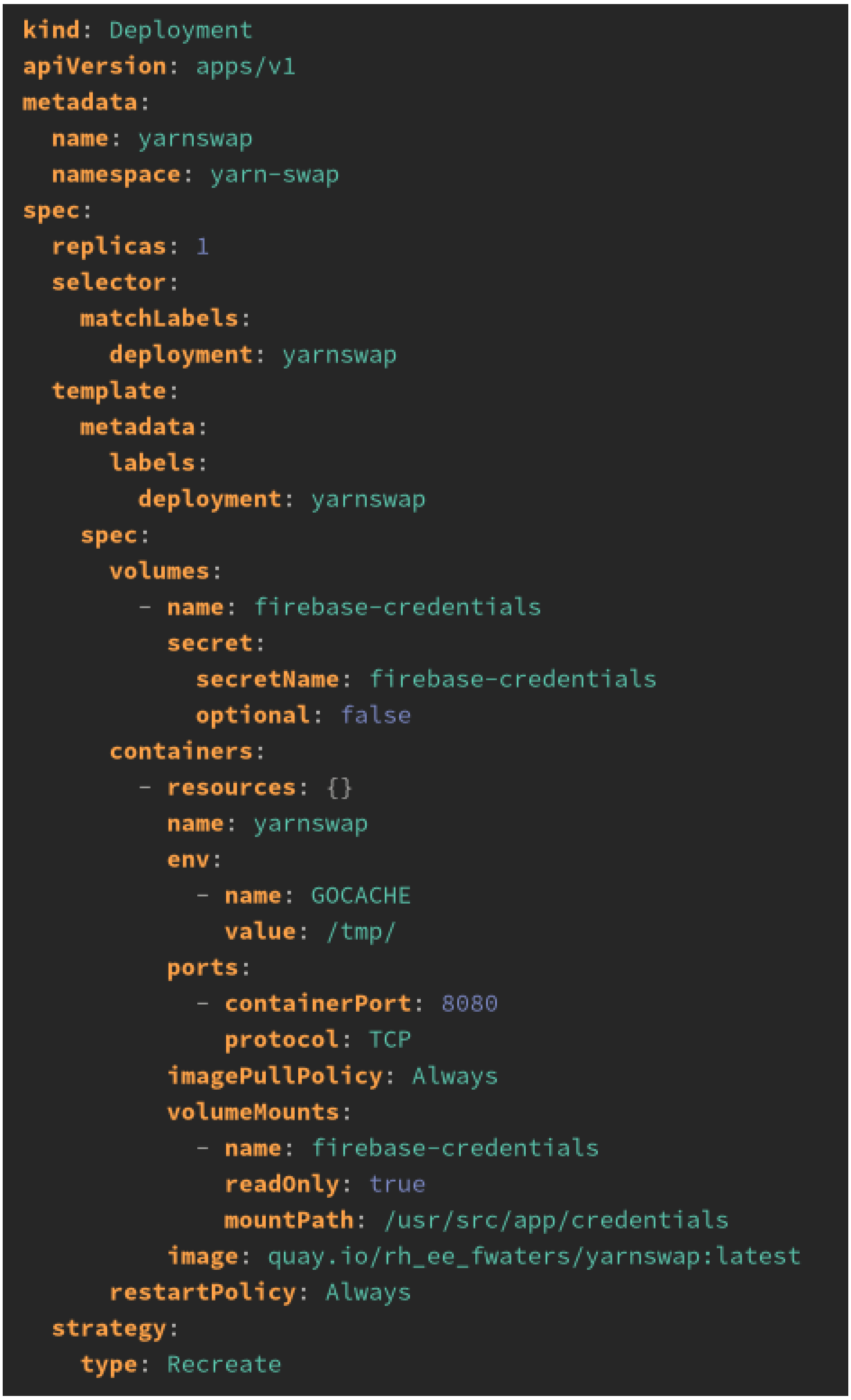
```
oc new-app -e GOCACHE=/tmp/ quay.io/rh_ee_fwaters/yarnswap:main -n yarn-swap
```

The oc new-app function creates a build config, a deployment, a service and a route.

Expose the route:

```
oc expose service/yarnswap
```

A yaml file has been prepared to apply some extra items that are required. As the backend is connecting directly with Firebase, a secret is required. This secret contains the Firebase credentials. This also means that the cluster deployment requires additional configuration to access and these credentials. Volumes and VolumeMounts have been added. Also, the restartPolicy has been set to always, so that the pod will restart when the image is updated. This is the complete deployment yaml file:

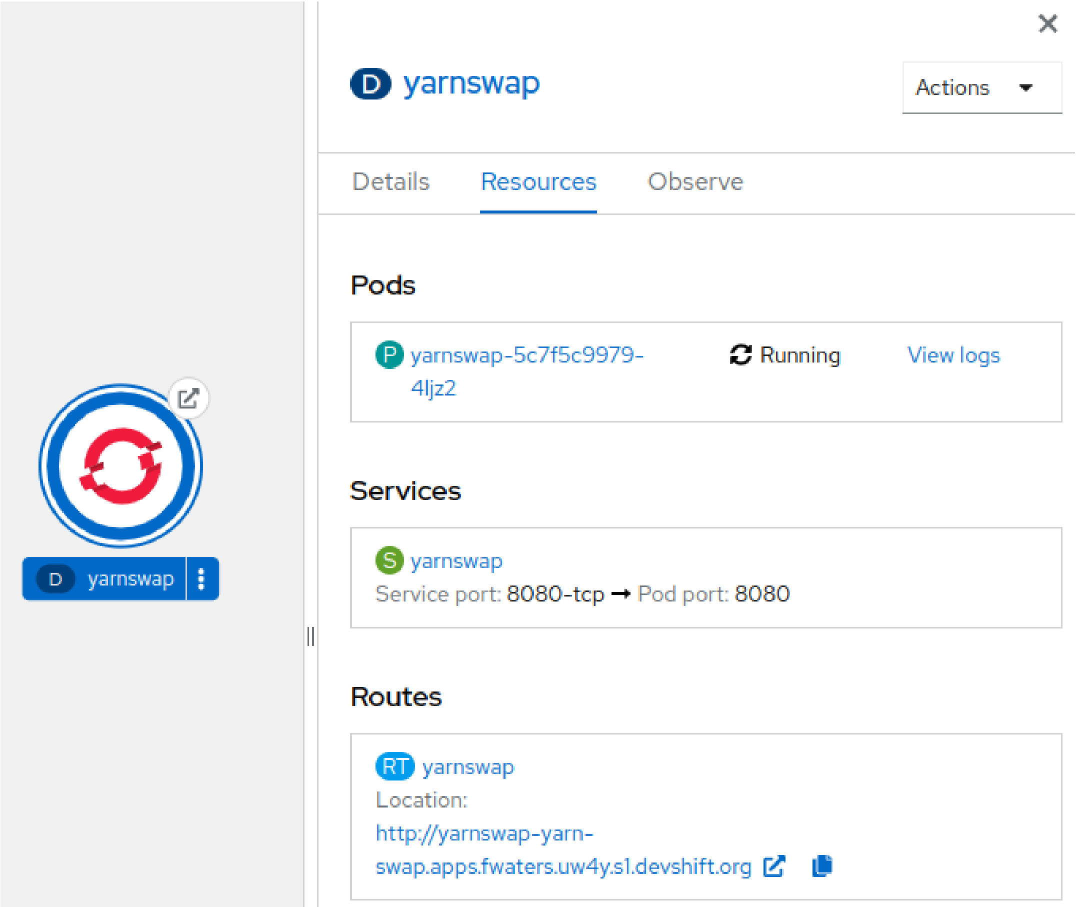


```yaml
kind: Deployment
apiVersion: apps/v1
metadata:
  name: yarnswap
  namespace: yarn-swap
spec:
  replicas: 1
  selector:
    matchLabels:
      deployment: yarnswap
  template:
    metadata:
      labels:
        deployment: yarnswap
    spec:
      volumes:
        - name: firebase-credentials
          secret:
            secretName: firebase-credentials
            optional: false
      containers:
        - resources: {}
          name: yarnswap
          env:
            - name: GOCACHE
              value: /tmp/
          ports:
            - containerPort: 8080
              protocol: TCP
          imagePullPolicy: Always
          volumeMounts:
            - name: firebase-credentials
              readOnly: true
              mountPath: /usr/src/app/credentials
          image: quay.io/rh_ee_fwaters/yarnswap:latest
      restartPolicy: Always
  strategy:
    type: Recreate
```

The secret is applied to the cluster in the same way. I will not share the file for security reasons.

Once these steps have been completed the application pod should be running.

This screenshot shows the Topology in the OpenShift Dedicated UI. You can see the pod is running and can click to view the logs. You can also see Services and Routes details.



D yarnswap

Pods

P yarnswap-5c7f5c9979-4ljz2 Running

Services

S yarnswap
Service port: 8080-tcp → Pod port: 8080

Routes

RT yarnswap
Location:
http://yarnswap-yarn-swap.apps.fwaters.uw4y.s1.devshift.org

The backend is now deployed.

Deploy the Frontend

The frontend deployment is slightly simpler as no secrets are required and therefore no extra configuration. You have already created a Docker image of the application, have pushed it to a repo on Quay.io and have created and logged in to your OpenShift Cluster as per the instructions above.

On command line.

Select the yarn-swap project/namespace where you want the application to reside:

```
oc project yarn-swap
```

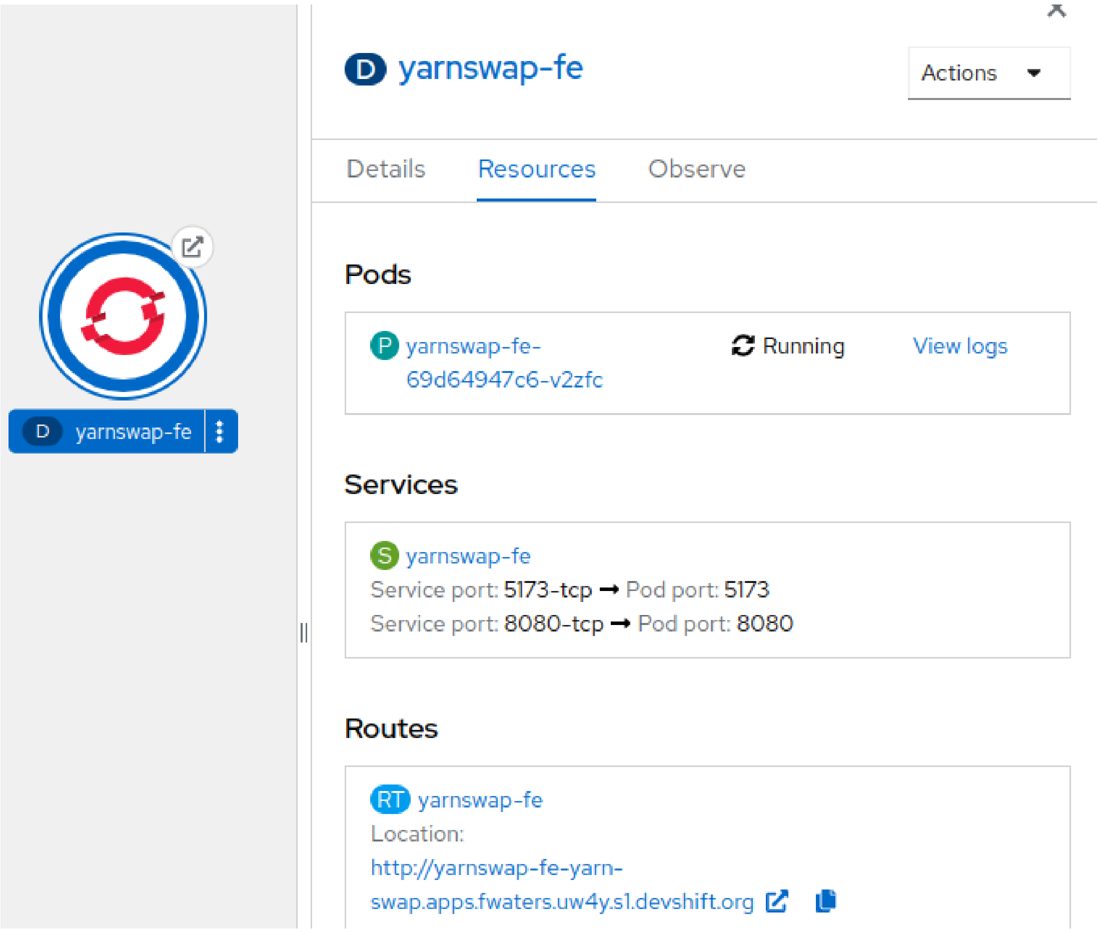
Create the application, specify the location of the Docker image, and specify the namespace where you want the application to be placed.

```
oc new-app quay.io/rh_ee_fwaters/yarnswap-fe:develop -n yarn-swap
```

Expose the route:

```
oc expose service/yarnswap
```

The application is now deployed. This screenshot shows the Topology in the OpenShift Dedicated UI. You can see the pod is running and can click to view the logs. You can also see Services and Routes details. Clicking on the Location URL under Routes will render the frontend of the application in the browser.



Pods

P yarnswap-fe-69d64947c6-v2zfc Running

Services

S yarnswap-fe
Service port: 5173-tcp → Pod port: 5173
Service port: 8080-tcp → Pod port: 8080

Routes

RT yarnswap-fe
Location:
http://yarnswap-fe-yarn-swap.apps.fwaters.uw4y.s1.devshift.org

Appendix E – Ethics Questionnaire Responses

See next page.

School of Science and Computing Research Ethics Committee

My Home / Modules / School of Science and Computing Research Ethics Committee

/ School of Science and Computing Ethics Checklist (2022-2023)

/ Ethics Checklist for Undergraduate, Taught Postgraduate and Research Projects in the School of Science and Computing (2022-2023)

Your response

Respondent: **Fiona Waters** (Group: CM-HDIPCS) Submitted on: Monday, 5 December 2022, 12:39 PM

Ethics Checklist for Undergraduate, Taught Postgraduate and Research Projects in the School of Science and Computing

All students in the School of Science and Computing who are either (1) in the final year of an undergraduate/BSc degree, or (2) on a taught postgraduate/MSc programme **must complete this Ethics Checklist before conducting their project** regardless of the project type or discipline. The Checklist should also be completed by anyone (whether staff member or student) conducting a **research project** (whether programmatic or not) within the School.

The purpose of this Ethics Checklist is to **identify projects that will require formal ethical approval** from the School Research Ethics Committee, or the SETU Research Ethics Committee, before they can proceed.

Students/applicants should note that this Ethics Checklist is a **formal declaration**, and great care must be taken to **answer all questions accurately**. Students should consult with their project supervisors/advisors regarding any aspects or questions that they are unsure of before completing and submitting the Ethics Checklist.

Students/applicants must **answer all questions** presented to them until the Checklist questionnaire is completed.

Feedback Report

No human experimentation issues (TPG/RP).

No animal experimentation issues (N/A).

No issues regarding the use of human tissues.

No animal tissue or biological fluids issues.

No ionising radiation issues.

No primary data collection issues (N/A).

No underage/vulnerable people issues (TPG/RP).

No issues regarding existing/secondary data use (N/A).

No controversial data issues.

No issues related to the collection of rare or protected plants.

No issues regarding the use of genetically modified (GM) plant material.

Instructions:

1. If the above feedback is **entirely** **green** then, based on your answers, there is **no need to apply for ethical approval** for your project.
2. If **any** part of the above feedback is **yellow/amber**, then there is at least one issue with your project that needs to be reviewed and **you must apply for ethical approval** to continue your project.
3. If **any** part of the above feedback is **red** then there is a serious ethical issue and **you cannot continue your project** as currently planned.

It is recommended that you print this Feedback Report to a PDF file for your records. You should also forward and discuss this Feedback Report PDF with your project supervisor. They will be able to advise if you have any further questions or if you need to apply for ethical approval.

1 * Are you a student on a **final year undergraduate** programme, a **taught postgraduate** programme, or are you conducting a **research project**?

○ Final Year Undergraduate
● Taught Postgraduate
○ Postgraduate Research Project
○ Other Research Project

2 * What is the **working title** of your project?

Yarn Swap

3 * Who are the project **supervisors/advisors/principal investigators**?

Colm Dunphy/Peter Windle/TBC

4 * Does your project involve **human experimentation**? (e.g. clinical trials)

○ Yes ● No

5 * Does your project involve **animal experimentation**?

○ Yes ● No

(6) * Is the planned animal experimentation limited to **non-invasive procedures only** (such as feeding, weighing, or taking hair samples), and does **not** involve any invasive procedures (such as taking blood) from live animals?

○ Yes ○ No

7 * Does your project involve the use of **human** remains/cadavers/tissues/cells/biological fluids/embryos/foetuses?

○ Yes ● No

(8) * Do you intend to only use established **commercial human cell lines**, and no other **human** remains/cadavers/tissues/cells/biological fluids/embryos/foetuses in your project?

○ Yes ○ No

9 * Does your project involve the use of **animal tissues** or **biological fluids**?

○ Yes ● No

(10) * Do you intend to only use (1) **established commercial animal cell lines**, or (2) **slaughterhouse-derived tissues/fluids**, or (3) **fluids collected as part of routine animal husbandry** (e.g. milk) and no other animal tissues or biological fluids in your project?

○ Yes ○ No

11 * Does your project involve the **collection of rare or protected plants**?

○ Yes ● No

12 * Does your project involve the generation or use of **genetically modified (GM) plant material**?

○ Yes ● No

(13) * Do you agree to (1) only use **established genetically modified (GM) plant cell lines, seeds, or plant products** in your project, (2) **not generate new plant mutations** using chemical or other means, and (3) follow specified SETU **containment and use protocols** for GM plant materials at all times?

○ Yes ○ No

14 * Does your project involve the use of **ionising radiation**? (e.g. use of gamma ray spectrometry)

○ Yes ● No

(15) * Do you agree to carefully **follow the instructions** of the SETU designated **Radiation Protection Officer (RPO)**, and **adhere to all legal requirements** as set out in the Radiological Protection Act 1991 (Ionising Radiation) Regulations (2019), regarding the use of ionising radiation materials and equipment?

○ Yes ○ No

16 * Does your project involve the **collection of any new (or primary) data** from **individual people or groups**?

○ Yes ● No

(17) * Does your project involve the **collection of any new (or primary) individual or group data** that is **personally or uniquely identifying**? (e.g. data about people or organisations/companies/groups that could be used to identify those individuals or groups; data collection might take any form, including internet and social media data, etc.)

○ Yes ○ No

(18) * Will you ensure that participants who you are collecting data from are provided with **fair warning** and must provide **explicit informed consent** for any data collected?

○ Yes ○ No

(19) * Will you ensure that any project-related data collection, data storage, and data use is in **full compliance** with the **EU General Data Protection Regulation (GDPR)** and the **Data Protection Act (2018)**?

○ Yes ○ No

(20) * Does any of the data that you intend to collect include **sensitive or private personal information** about individuals, or **commercially sensitive information** about organisations/companies/groups?

○ Yes ○ No

21 * Does your project involve **persons under the age of 18 years** (i.e. minors), or **any vulnerable groups**? (e.g. prisoners, refugees, those in care, addiction service users, etc.)

○ Yes ● No

22 * Does your project involve the use of **existing (or secondary) data**? (i.e. data originally collected for another purpose)

○ Yes ● No

(23) * Is the existing or secondary data you intend to use either (1) **anonymous/non-personally identifying** and in the **public domain**, or (2) available with **explicit and specific informed consent or permission** for the data to be **legally** reused in the way you intend?

○ Yes ○ No

(24) * Are any aspects of the primary/secondary data you intend to use for the project **controversial** in nature?

○ Yes ○ No

25 * Before you submit the Ethics Checklist, you must **confirm all of the following**:

☑ I understand that the Ethics Checklist is a formal declaration.
☑ I have answered all questions on the Ethics Checklist carefully and truthfully.
☑ The supervisor/advisor (or principal investigator) for the project is present as the Ethics Checklist is being submitted, or they have given me explicit permission to submit it in their absence.
☑ I have had adequate ethics training and/or instruction prior to completing the Ethics Checklist.
☑ I understand, and agree to abide by, the general ethical principle of "do no harm" for this project.
☑ I will follow the instructions given in the Feedback Report.

26 * Authentication Code (ask your project supervisor/advisor for this code)

Enter Student Number: ▭

Enter the Authentication Code below and click "Verify Code" [Verify Code]

Note: If an INVALID authentication code is used then this submission is NULL and VOID

2727

◀ Announcements   Jump to...   Ethics Checklist Feedback Questionnaire (***optional***) ▶

You are logged in as Fiona Waters (Log out)
School of Science and Computing Research Ethics Committee

Get the mobile app