COSC2299 Software Engineering: Processes & Tools

Project Report

BrokieHub

Group P3 Number 8

GitHub Repository: **https://github.com/cosc2299-sept-2023/team-project-group-p03-08**

Name	Student ID	Contribution
Kiran Kulkarni	s3943716	16.67%
Jamie Truong	s3947728	16.67%
Peter Fulton	s3896790	16.67%
Andy Chen	s3935474	16.67%
Benjamin Nippard	s3945124	16.67%
Tyler Humbert	s3947682	16.67%

Vision Statement

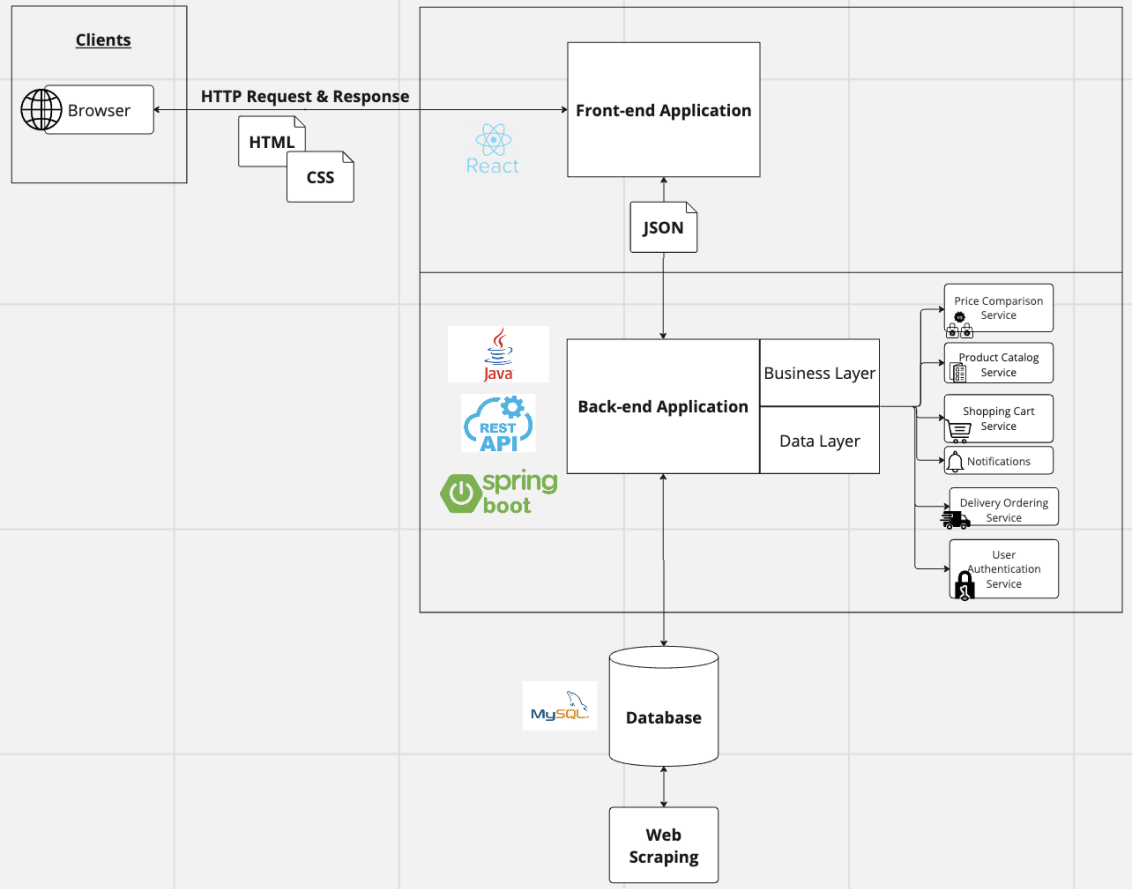
The vision for BrokieHub at its fundamental level was to create a price comparison website that could query local grocery stores alongside Coles and Woolworths and provide a comprehensive list of prices in the local area to users. The value in the project is that this website acts as a tool for users to make educated analysis of the prices in their local area to ensure they are getting the best bargains at any given time. This has become especially more important as the cost-of-living soars and ordinary Australians are struggling to get value for their daily groceries.

The project was envisioned to expand on the baseline premise by providing a wide array of tools that would be implemented in future sprints with features such as:

- Price History display, where the database would log and hold previous prices of a given product at a monthly and a trimonthly basis to give users an idea of price increases and make them aware of inflated prices
- Location Mapper, where products would also display where they are in relation to the user's address in their account. This would allow them to make choices about where to purchase a product from based on location in addition to prices.
- Notifications to alert users about their favourite products, especially when they have had a significant price decrease in comparison to previous time periods, or when they have limited time deals.

System Architecture Diagram

The system architecture that we have adopted for our project is the modern web application architecture that consists of a front-end application with a presentation layer and a back-end application consisting of the business and data layer.



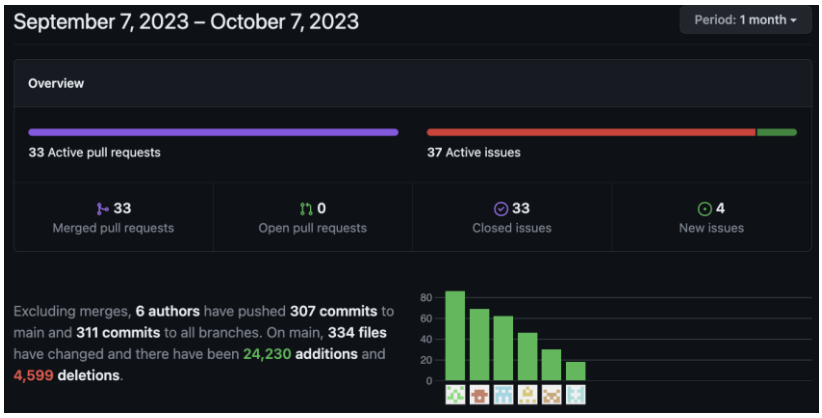
The front-end application consists of the presentation layer that enables users to interact with the back-end application through a browser that sends HTTP requests where our front-end application will return a response.

The back-end application consists of the business and data layer. The business layer consists of our models, controllers and services. The models contain the business concepts and manages the data of our application. The controllers handle HTTP requests and queries based on the business logic and models. The controller receives user inputs from the presentation layer, validates it and then passes that information to the models. The services contain the overarching business logic of our project. The other layer for our back-end application is the data layer consisting of our repositories which contains the persisting data of the database for our project. We have also included a connection from the database to web scraping in order to fill the product database with products from the web.

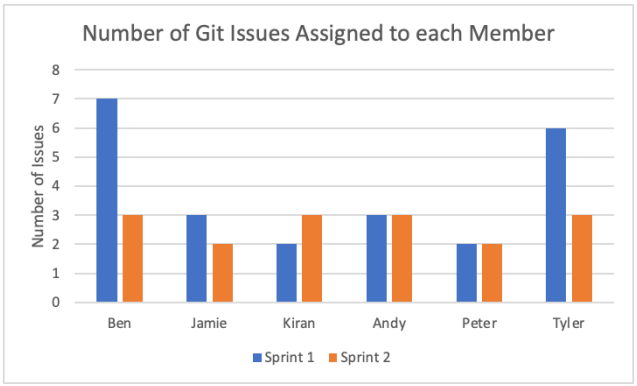
The technologies employed for the components of our software architecture include using React for the front-end framework, Spring Boot for the back-end framework and MySQL for data storage and querying.

Git Organisation

Overview of Git Organisation:



Number of Git Issues Assigned to each Member:



Graph of Code Frequency:

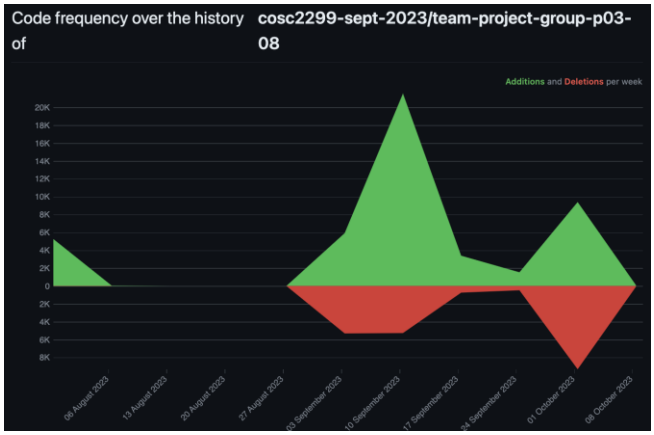
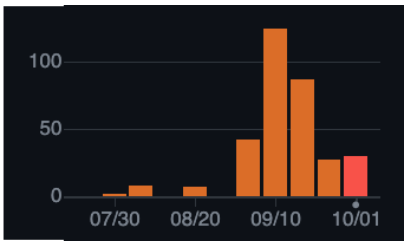


Chart on how often we Committed:



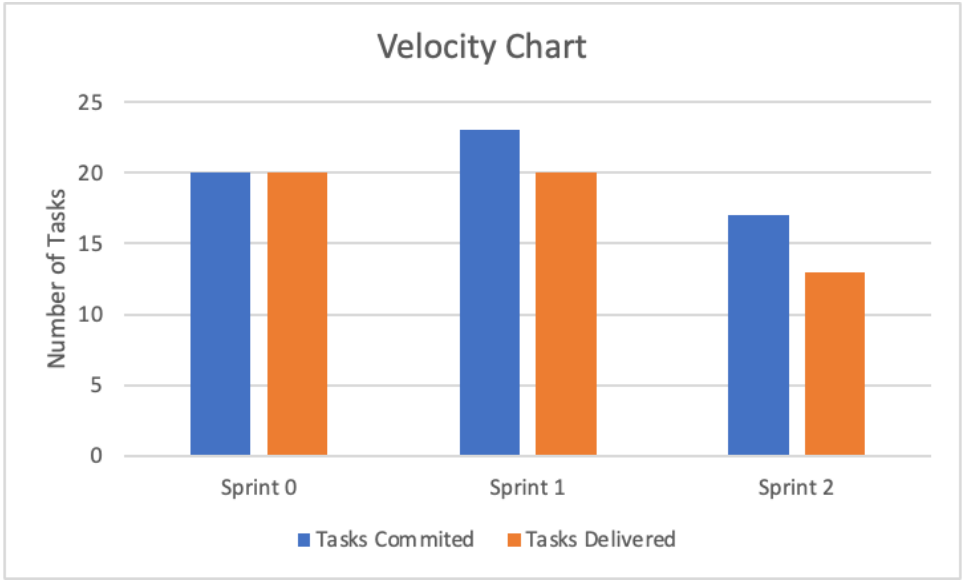
For our GitHub organisation we utilised branches for key features to ensure proper functionality before merging into our main branch within which we have continuous integration running to ensure that our code merges are stable. For key features we based these off our GitHub project board which outlines the requirements of the feature these tasks are issues which are then marked as resolved once the feature/s is completed.

Scrum Organisation

For this project our team followed the scrum process and framework that aims to help us address complex problems with high productivity and creativity to deliver an application that meets the needs of our users. The Scrum Master of our team was Kiran who ensured that our team followed the Scrum process. The remainder of our team participated in the development team where we aimed to execute the project.

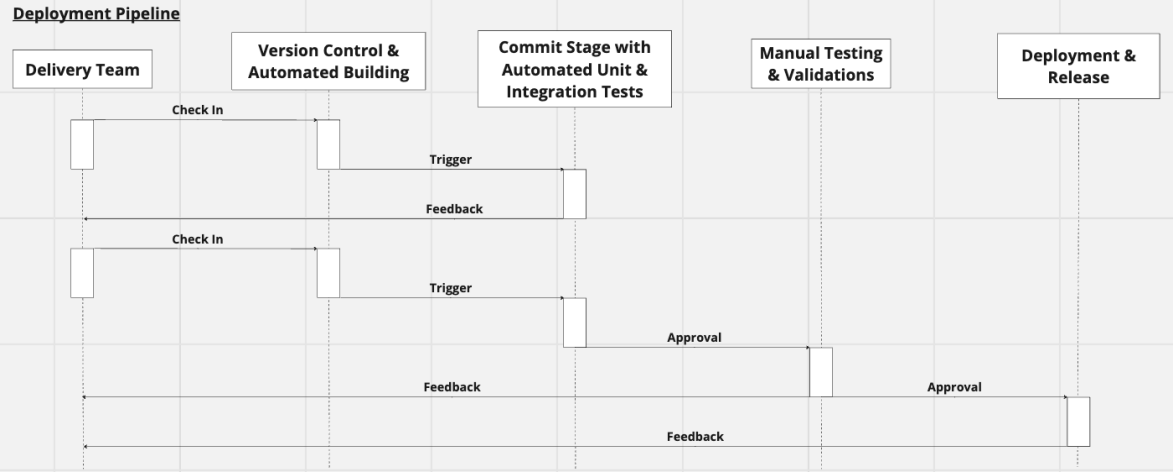
Our Scrum Process involved at the beginning of each sprint, conducting Sprint Planning where we defined the goal of our sprint, creating tasks from our developed user stories, adding tasks to our product backlog on our GitHub project board, estimating the duration of these tasks in hours and then allocating tasks to each member of the team.
Sprint stand ups were held twice a week (Tuesday and Thursday) where members described what they had completed, what they still had to do and any blockers that they were facing. Further details of each sprint stand up can be found in the minutes. At the end of each Sprint, we conducted a Sprint Review where our Scrum Master wrote a Sprint Retro reflecting on matters that worked well and didn't work well, and things that could be improved for the following sprint. In addition, we also received feedback from our product owner (tutor) at the end of each sprint that assessed our progress and indicated where improvements in our process that could be made.

The overall velocity of the team through all sprints is reflected in the following graph.



Deployment Pipeline

The Deployment Pipeline of our application involves the use of Continuous Integration and Continuous Development through GitHub Actions and utilises a set of practices that helps automate the building and testing of our software.



Specifically in the first stage of our deployment pipeline, we utilise Version Control that maintains a single source repository for our team. Also in this stage, we employ Automated Building using Maven that helps to ensure quality assurance and ensures a working build of the application whenever features are pushed to main. The next stage of our pipeline is the Commit Stage that ensures our application is working on the technical level by running a number of unit and integration tests. Next, in the deployment pipeline we conduct Manual Testing that involves ensuring that the application fulfils its requirements and detecting any errors that the automated tests did not detect. These manual tests predominantly involve user acceptance testing. Finally, we enter the Deployment & Release stage where the system is delivered to the end users and the application is deployed through either a production or staging environment.

Overall, we found the implementation of CI/CD extremely useful in our team as it helped keep our team in sync and additionally whenever there were any issues with any of our builds, CI/ CD promptly identified the issue and this allowed for us to quickly respond and fix it.