Documentation. ( REMOVE ONCE DONE )

▪ Software process and technical details up to this point (4 Pages)

**Introduction ( REMOVE IF NOT ENOUGH SPACE)**

The problem

A study led by the University of Leeds found that unhealthy foods, such as pastries, caffeinated drinks and alcoholic beverages contribute a total of 23.6% to food-related emissions. Meanwhile, non-vegetarian diets produced 59% more greenhouse gas emissions than vegetarian diets. (Impact of Climate Change and Adaptation Measures, n.d.)

One of the conclusions drawn from said study was that a healthy diet based on unprocessed, largely plant-based foods is a sustainable one, pointing to the 2019 IPCC Climate Change report that suggests that adopting this kind of diet would reduce diet-related greenhouse gas emissions by 80%. (Henderson, 2021)

Our solution

A responsive web application that encourages users to learn and cook healthy meals with preferably low-carbon and locally produced ingredients instead of travelling outside to eat.

The application allows users to input food ingredients that they possess in their household and suggests healthy meals that the user could create.

Essentially, the application acts as a recipe book that would show the user the meals that they can create using their existing ingredients. This will encourage the user to utilise their leftover ingredients efficiently without wasting them.

**Software Processes**

We have broken down our software processes into four main portions when applicable, the initial plan, our experience when implementing it, any technical details as well as our conclusion or learning point.

**Architecture of the Software**

Initial Plan

We initially decided to stick with frameworks and applications that we have learnt in class. Hence, we have decided on the following,

Project Tracking : Jira

Front-end : React JavaScript Library

Back-end: Java Spring Boot

Version-Control : Github / Git

Database : H2 Database

Hosting : AWS

Api : Spoonacular

Experience

However, after discussion, it was decided that a localized database would not fit with our use cases that well. Thus, we planned to switch over to a cloud-based database such as google.

Conclusion / Learning Point

What was initially decided might not be the only or best option. If needed, we should be able to switch to the better option.

**Project Tracking - Jira**

Initial Plan

As mentioned above, we chose to go with applications that we learnt in class whenever possible to make our experience as smooth as possible.

Experience

The first sprint with Jira did not go too well. Some of us were not too comfortable with the website since we only had minimal experience with it in class. Most of the features of Jira were not too intuitive. For example, some of us decided to start the sprint and then add on the tasks and user stories. However, we quickly ran into errors and had to restart.

Fortunately, the second sprint went much more smoothly without much errors.

Technical Details

<https://forksmash.atlassian.net/jira/software/c/projects/CS203G1T4/boards/4>

Conclusion / Learning Point

Familiarize yourself with the application that you chose to use to minimize delays or errors.

**Front-End - React**

Initial Plan

After consulting the professor, we found out that React self-help materials would be released in elearn. In addition, since some of us had good experience with React, we chose this as our Front-End.

Experience

??

Technical Details

??

Conclusion / Learning Point

??

**Back-End – Java Spring Boot & Version Control – Github/Git**

The backend and version control was combined as most of the comments regarding one was related to the other.

Initial Plan

Since this was the application that was learnt extensively in class, we decided unanimously on this Java Spring Boot and Github.

Experience

Most of the difficulties in implementing the backend comes from not knowing how to coordinate the code. The workload was mostly split in 3 people doing the backend, 2 people doing the front-end as well as 1 person doing the database. Thus, there were 4 people coordinating in a single folder.

Since this was the first semi-large scale production project for most of us, we pushed most of our changes into the main branch instead of separating it into branches. This caused some dependency problems that could have been easily solved with git branches.

For example, code that was probably more suited to be pushed into a separate branch was pushed into the main branch causing certain failures.

Fortunately, this issue only occurred during the first sprint and it was a good learning experience for all of us.

Technical Details

Implemented AppUser with its controller, service and repository.

Implemented Registration with its controller and service.

Conclusion / Learning Point

Branches in Git are very helpful and should be used.

<https://github.com/kendricktty/cs203-g1t4>

**Database**

The database implementation was probably the most contentious point of our project.

Initial Plan

The initial plan was to use the innate h2 database in spring boot which was also what was taught in class. However, it was brought out that a cloud-based solution was probably a better option considering our application.

Experience

The main problem was that it was a lot harder to implement a cloud-based database compared to using the h2 database.

Thus, there was a bottleneck at the database since the other back-end features were depending on it in order to test their own implementation.

Google cloud database was tried but was ultimately given up due to the pricing and other factors.

Technical Details

????

Conclusion / Learning Point

???

**PUT THE ER DIAGRAM HERE?????**

**Hosting - AWS**

At this point, we have not implemented any hosting services yet. However, the plan is to use AWS.

**Conclusion**

Admittedly, some of the code implemented was referenced online with modifications made to fit the project. The proper references will be made in the final document once the project is finalized.

**References (REMOVE IF NOT ENOUGH SPACE)**

**https://documentation.divio.com/#**