Our group decided to use Ben's 'Fridge Recipe' EDR as the foundation for our project due to its clarity, suitability for a group setting and alignment with the project requirements. The chosen EDR uses the "SSH Camera" to scan different tenants' ingredients. Storing it on the 'cloud database'. It then takes these ingredients and using a mix of different equations, returns a list of recipes, in order of relevancy, that the tenant can then proceed to follow the instructions for and make.

During the initial planning, we made a few changes to the original design. For example, we modified the equation for quality. Rather than it using *e* and having a 'decay rate', to help better suit our idea of how the project would flow into each other – we simplified it and used the SSH camera to simulate the captured date of the ingredient. We also decided to ignore the user interface on the SSH Console Table as we decided that this would be irrelevant for a bare bones prototype.

Overall, we accomplished a functional prototype that effectively manages and stores data in database. It then returns accurate recipes for the user. The prototype meets the initial objectives we set and beyond just technical deliverables, the project strengthened many of our teamwork, problem-solving and coding skills.

We began the project by thoroughly analyzing the chosen EDR, to ensure that it meant our own project requirements. During our initial needs meeting, we identified the areas of modification which were necessary. Some of the main issues rose around the calculations, which we changed as needed.

**Calculations Plan**

Changing the calculations as we have come across some problems. The intention is to use the items that are going out of date rather than prioritising the fresher items (which is what the calculations do as of now).

New conditions being implemented too.

M: Recipe match rating, ranging from 0-100%
M first being based on a user having all the items, then secondarily based off the average quality \

Q: Quality of the ingredient, ranging from 0-100%
Q = (1 - Days since captured/Estimated Shelf Life + 1)

To streamline the process, we divided the tasks among the team based on individual strengths and affinity towards a certain task. Ben focused on the research for recipes and the potential for a JavaFX interface. Josh was to set up the database units and set up the repository to run smoothly. Jamal was to develop the logic behind calculating random quality and the logic behind ranking the recipes of the users. Arman was assigned to randomly generating items (mocking the SSH camera) and to generate the logic for assigning random amounts of the items. All aspects were worked on mainly by the person assigned to it – but also utilized the issues tab on GitHub to request help and input from everyone else involved.

⊙ 8 Open  ✓ 23 Closed

The prototype development was carried out in *PostgreSQL* and Java using the EDR as a blueprint. During the database creation phase, we ensured normalization to avoid redundancy and improve efficiency by outlining what tables we would have and their respective columns. However, our first issue arose in setting up the database. Josh set up an issue on GitHub to gain input from everyone. Jamal mentioned that docker would be able to help us to quickly set up our database, while it also running easily across all our machines. This show of collaboration and communication allowed us to quickly resolve an issue – and caused the flow of the project to be unstopped.

## Set up docker if needed #21
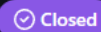
**Closed**   varsityyy opened this issue 2 weeks ago · 4 comments · Fixed by #31

**varsityyy** commented 2 weeks ago   Collaborator · · ·

Look into docker.

Will be useful for quick set up of our database we plan to include.

If agreed, setup must be done as soon as.

☺

**JoshuaRaybould** commented 2 weeks ago   Owner · · ·

I'm looking into it now, it does seem like it will be useful so we can set up the database quickly and easily on our machines. I'll try and have it set up either today or end of tomorrow at the latest.

☺

**JoshuaRaybould** commented 2 weeks ago   Owner · · ·

I'm pretty confident we'll use docker, so it would be a good idea for everyone to install it and the vscode extension for it when they get chance.

☺

A key aspect of the prototype was the accurate rankings of the recipes and their quality insurance. This required some problem-solving as we were wondering how to simulate the camera and the different shelf lives of the items. While the logic behind the code is in the repository, they were the result of extensive team collaboration and iterative improvements.

**varsityyy** commented 3 days ago · edited ▾   Collaborator · · ·

I'm looking into it now, yes I do not have any logic for actually ranking the recipes in that specified order, i rank them based on quality score. Something i completely missed.

☺

🔖 **varsityyy** linked a pull request 2 days ago that will close this issue

**Update Recipe logic** #90   Closed

**varsityyy** commented 2 days ago   Collaborator · · ·

I have managed to sort this out now.

☺

**varsityyy** commented 2 weeks ago • edited ▾     ( Collaborator )  •••

Started to change the quality logic to handle the intervals in days rather than minutes. This is to make it consistent with Arman's unit since he is using days aswell

☺

**JoshuaRaybould** commented 2 days ago     ( Owner )  •••

It looks like the random date captured can be either before or after the current date, where in reality we can only have captured it on or before the current date. This is leading to some quality scores greater than 1, also currently it seems the current date time stamp is being put in the database rather than the calculated one. I'm looking over this now and believe I can solve these issues, would you like me to go ahead with this @ArmanAli05 or would you prefer do this yourself?

☺

Through further meetings and conversations, we also managed to refine and better our prototype. For example, initially we were populating the recipes of the database manually in a CSV file, and then connecting to the database it would take the data from it. However, after some further meetings - we decided to edit some of the databases tables and realized not everything was concrete. So through the use of GitHub issues, we left some open for any more changes down the line. This led to the upgrade from CSV files to JSON files.

**JoshuaRaybould** commented last week     ( Owner ) ( Author )  •••

Whilst the main portion of the database is set up and has been confirmed to work, after we further discussed it became apparent we may want to remove some tables, so I'll leave this issue open until we've fully decided and completed this part.

☺

**JoshuaRaybould** commented last week     ( Owner ) ( Author )  •••

I think since we are planning to just hardcode recipes elsewhere we could do with removing the recipes table, I think there may be other stuff we want to clean up slightly so someone can do that as a bigger thing later on, in a dedicated branch, unless @benjoseph05 removes it when he is hardcoding the recipes in java. After this, this issue can be closed.

☺

**benjoseph05** commented yesterday     ( Collaborator )  •••

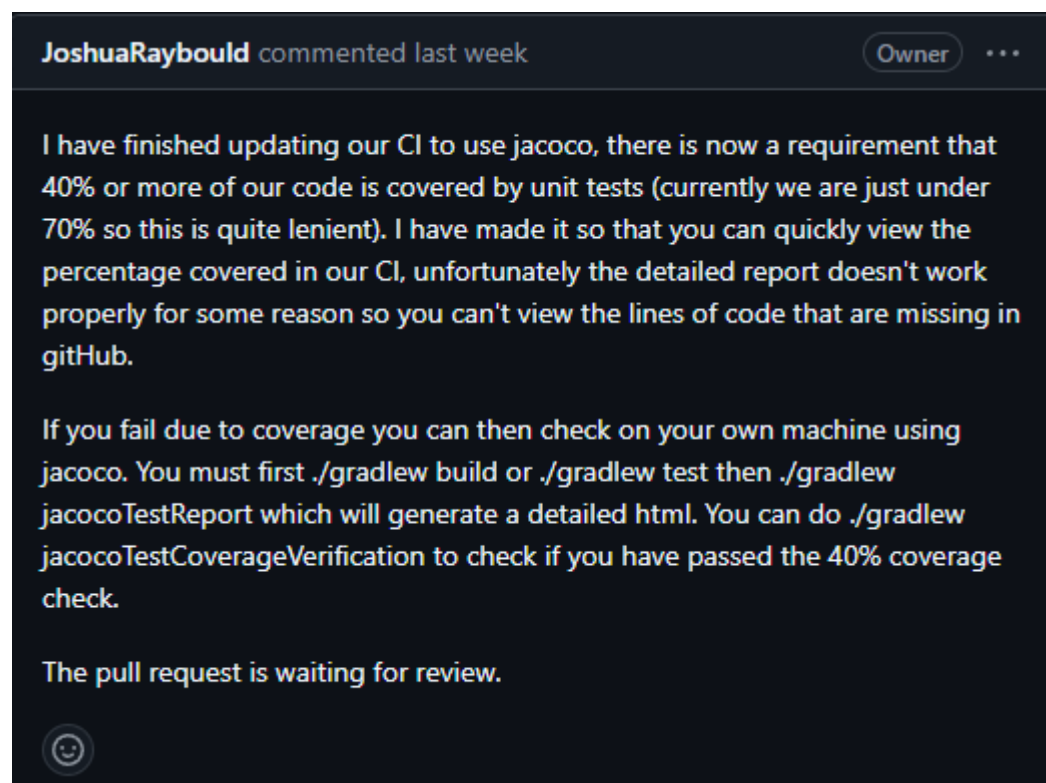With the JSON recipe file, we can remove the recipe csv file yes?

☺

**JoshuaRaybould** commented yesterday     ( Owner ) ( Author )  •••

Your right yes, I have removed the csv and other no longer necessary database tables.

☺

One of the problems we faced was the creation of a test user. The problem was that creating a test user and populating it's relative tables proved hard to accomplish before the unit testing of the database began – leading to constant failure even though the logic was right. Resolving this required team work and persaverance. Josh and Arman proposed many different ideas (such as using a sleep() function or @Before to make sure the user was created first) but many carried on to fail. However, through a lot of time and collaborative effort, they figured out that the database was building in the wrong order and managed to fix it within the gradle. This area displayed a sense of ownership as Josh was not assigned to this particular aspect of the project, but understoof the technical side. Allowing them to gain a correct fix for everyone.

Throughout the creation of the project, we conducted unit testing for our own areas. Josh decided that to make sure our code is consistent, he would add *Jacoco* to insure more rigorous testing. This made us aware of more areas to conducting unit testing and integration testing, to make sure our code succeeded in different areas. This also allowed us to identify areas that were wrong or created problems.

**JoshuaRaybould** commented last week     ( Owner )   • • •

I have finished updating our CI to use jacoco, there is now a requirement that 40% or more of our code is covered by unit tests (currently we are just under 70% so this is quite lenient). I have made it so that you can quickly view the percentage covered in our CI, unfortunately the detailed report doesn't work properly for some reason so you can't view the lines of code that are missing in gitHub.

If you fail due to coverage you can then check on your own machine using jacoco. You must first ./gradlew build or ./gradlew test then ./gradlew jacocoTestReport which will generate a detailed html. You can do ./gradlew jacocoTestCoverageVerification to check if you have passed the 40% coverage check.

The pull request is waiting for review.

☺

To help with our testing phase, Jamal looked into the implementation of mockito – as this would allow us to set up easy unit tests that are independent of other parts of the code to ensure that that individual logic was correct. Josh also looked into adding *piccoli* to make our program take command line arguments and try to add more integration tests for it.

**varsityyy** commented last week      ( Collaborator )   • • •

We need to explore how Mockito can be effectively integrated into our project to improve the quality of unit tests. Specifically, it will help us isolate dependencies and test critical logic like the rankRecipes method in the RecipeRanking class without relying on real database or external dependencies.

☺   👍 1

**varsityyy** commented last week • edited ▾      ( Collaborator ) ( Author )   • • •

Mockito pretty much allows us to simulate the behavior of external dependencies like databases or APIs, enabling us to focus on testing the logic of specific methods. This can improve the reliability and maintainability of our unit tests.

**JoshuaRaybould** commented last week      ( Owner )   • • •

I'm currently looking at how to improve my code for the sake of testing, I have realised many tests we've introduced rely on the database and so are more in line with integration tests. I think mockito as suggested by jamal, could solve this, but if not, we'll try to find another way. This isn't to say don't put the tests in just that we may want to think about what we are doing a bit more.

☺

**JoshuaRaybould** commented last week      ( Owner ) ( Author )   • • •

For most of my part unit tests don't make sense, so I'm going to add picolli and make our program take command line arguments and try to add to integration tests for it. These can then be added to our CI.
For **@varsityyy** if mockito isn't working I guess you could hardcode a list of ingredients for that tenant even though that's quite annoying. Then have some integration testing too. @ArmanAli05 should be able to create unit tests to check his values are reasonable. **@benjoseph05** it depends how you proceed so get back to me in the other issues and pull request please.

Throughout the project, we gained valuable insights into our technical skills, team work, project management and even leadership. One key takeway was the importance of communication and

constructive feedback. Through using GitHub issues, we managed to make the project flow seamlessly between all the different sections while also building in areas we weren't directly involved with – but were aware of what was happening.

## Josh's Reflections:

Working with Jamal has been a pleasure. Jamal communicated effectively with all the members of the group, keeping us all well connected. Jamal's main responsibilities were to ensure the main recipe matching logic and quality calculations for the recipes worked correctly. This required a lot of thought and multiple iterations to arrive at something we were satisfied with. Whilst there were certainly moments Jamal wasn't entirely sure on exactly how to do something he was always ready to collaborate and rectify any shortcomings. He was also extremely driven, striving to learn more outside of the confines of his unit, which was evident in his finding and recommendation of using mockito for testing. This also showed his attention to detail as he had realised our tests were all dependent on the database and wanted to make sure they were more self-contained. Throughout my time working with him, Jamal consistently showed great dedication and determination but also knew when to stop and alter his approach if things weren't working. Going forward, I feel he needs to be more confident in himself and make more suggestions, especially throughout the planning stages. Sometimes whilst the group were talking over something it was hard to gauge his opinion, which could have been very valuable moving forward. That isn't to say he didn't contribute or follow what was going on, Jamal always made sure to update us on his units of work appropriately so we knew how we could expect him to proceed.

Arman has been a joy to work with. He was especially good during the planning sessions of our work, providing valuable insight and often acting as a mediator when opinions on the work differed, helping us arrive at reasonable conclusions. I feel Arman often managed to lighten the mood when others were finding things difficult or frustrating which helped team morale. Arman's main role was to simulate the SSH camera getting details of user's fridge items. To make this as true to reality as possible we wanted this to generate random data which greatly increased the complexity. To achieve this, he suggested the addition of a new column in our database to differentiate between different types of food, such that the quantity values he generated could accurately match the recipes. When it came to completing his units of work, he would try his best to do what was required to a high standard. Although on occasion his updates on work were infrequent, when he did provide updates, they were often very important things to think about and keep up with and showed the depth of his thought process. I think it would have been helpful if he made a more consistent effort to answer other people's questions, for example on GitHub issues, or tried to look over code others had done to give them advice. I feel Arman has a variety of in-depth knowledge and understanding that would be highly beneficial to other team members.

I met Ben for the first time during this project and have enjoyed getting to know him. Ben's EDR was the one picked by our group which meant his thoughts and insights were especially useful during the initial planning phase. I felt he reacted to suggestions very well, not taking offence where we thought he had overlooked an aspect of design or wanted to change something. Ben's main responsibility was to simulate the system that would provide a list of recipes to the main backend, so that we could apply the recipe matching logic. Originally, we had planned to hard-code this information in java but later decided to use JSON. Despite not using JSON before Ben managed to adapt to the situation and fulfil his role to a high standard which was likely time consuming. Ben completing this on his own shows his dedication and willingness to learn more. I feel Ben could have done with trying to communicate more earlier on so that the team could be more aware of his situation and be able to

provide support sooner if needed. On the other hand, when it came to his own units of work, he was very good at keeping us updated on GitHub issues. Whenever he was unsure about an aspect of the design, he would always clarify to ensure what we were doing made logical sense and would ask for help when he needed it.

## Jamal's Reflections

I have really appreciated collaborating with Arman throughout this project. His proactive and innovative approach was apparent in the planning phase, where he contributed significantly by coming up with most of the ideas and leading the planning process. Arman's involvement was crucial in setting the direction for our work, particularly with the database setup and hardcoding. His ability to easily create complex logic for various project components was evident. Arman was always asking questions on GitHub issues, ensuring that he stayed on top of the work and kept the team aligned. His detailed feedback and questions helped clarify complex concepts, which was vital for maintaining the project's momentum. His deep understanding of the database side of our work allowed him to effectively contribute to the data handling and the development of quality calculation logic. This was particularly important in managing the "tenants_fridge_items" and "fridge_item" tables as part of the overall database layout, as well as the integration of the SSH Cloud Database. Arman's approach ensured that the team's plan was comprehensive and well-structured. His suggestions, especially regarding updates to the app and handling data from the fridge camera, were not only thoughtful but also practical, which helped address potential issues early on. Moving forward, I would encourage Arman to continue leveraging his ability to ask critical questions and his deep technical knowledge. By focusing on managing his workload efficiently and balancing his time between various tasks, he can further enhance his contributions and maintain the high standards we've established. Overall, Arman's dedication, innovative thinking, and collaborative spirit have been invaluable to our project.

Collaborating with Ben throughout this project has been a great experience. Ben was heavily involved in the planning phase, contributing many innovative ideas that helped shape the direction of our work. His focus on user interface design was particularly notable, as he proposed the use of JavaFX to create an intuitive and visually appealing application. This forward-thinking approach ensured that our project not only functioned effectively but was also user-friendly and accessible. One of Ben's key contributions was to utilize JSON for managing recipe data. He took on the idea that JSON offered flexibility and scalability compared to hardcoding recipes directly in Java. By leveraging JSON, we were able to easily add, edit, and structure recipes in a standardized format, which significantly organised data handling and improved maintainability. This practical recommendation demonstrated Ben's technical insight and his ability to identify tools that align with project goals. Additionally, Ben consistently demonstrated a keen interest in understanding how all the individual units of the project would integrate seamlessly. He frequently queried and questioned various aspects of the project on GitHub, ensuring his own contributions aligned with the overall structure. His proactive approach to developing his unit in a way that fit well with the team's work highlighted his collaborative spirit and attention to detail. Moving forward, I encourage Ben to continue building on his curiosity and technical adaptability. By further exploring integration strategies and honing his focus, he will undoubtedly enhance his already impressive contributions. Overall, Ben's dedication, problem-solving skills, and team-oriented mindset have been vital to our project's success.

Josh

# Arman's Reflections

Throughout this project, Josh has proven time to be an amazing collaborator to work with. Josh was a valuable member of our team and helped contribute significantly to the overall success of our work. Their main responsibility included setting up the Gradle and Docker, allowing us to all get our work done with little to no worries about how everything would tie together. From the start, he displayed a strong work ethic and serious dedication to ensuring the high-quality results we must present. One of his greatest strengths was displaying his leadership qualities. For example, he made sure that everyone was aware of their roles and responsibilities – ensuring that we all managed our own sections and were able to implement them thoroughly. Whenever an issue was raised on GitHub, he would go to the person directly and offer them whatever help he could, which allowed us to flow nicely into each other's code. Amounting knowledge of everyone's work allowed him to guide us all and allow the project to be completed with little to no headache. His understanding on how the more technical parts of the project should be handled meant that he was not able to implement it himself – but also teach us for any future projects we may have. Overall, Josh was an exceptional team player, whose efforts were invaluable to the project's success. Their combination of technical skill and leadership qualities made working with him a positive and productive experience.

Ben was an extremely valuable member of our team, and helped contribute to the success of our project. Their primary responsibilities included researching and gathering the recipes which we would display to the users, and the user-interface. His input during the initial preparation phase allowed us to streamline and seamlessly flow into each other. One of Ben's standout qualities was his attention to detail and insight into thinking of the future effects of the ideas being suggested. For instance, during the planning stage, he made many suggestions on how to implement the idea of the quality ranking and how we would store different data in the database. His key idea of implementing the recipes in a JSON file offered us flexibility and scalability when hardcoding them – improving maintainability and sustainability. This contribution was vital in maintaining the project's momentum and ensuring high-quality results. Additionally, Ben excelled in collaboration within the group. They were approachable with help and actively listened to the issues we presented on GitHub and quickly worked to solve and improve them wherever he saw fit. Not only that but he provided constructive feedback on other portions of the project to help improve our overall output. Their commitment to clear and consistent communication helped navigate the challenges of working on our own branches when they were dependent on his hardcoded recipes. Particularly when creating test cases – as he provided us samples of how the data would look like so we could proceed. If there was one thing I would change, it would be the intervals of communcation, as it would take a while to get a reponse. However, when he did it was extremely relevant and pivotal. Overall, Ben played a crucial role in the team's success. Their combination of hard work and teamwork made them a pleasure to collaborate with and achieve our goals.

Jamal was an integral part of our team, bringing both a sense of enthusiasm and hard work to the project. Their primary contribution included developing the quality system and displaying the recipes to the users in order of their availability. All of which were carried out with good commitment. Communication on Jamal's work was easy to understand through the use of issues on GitHub. I was able to gauge where he was at and understand how his area of the code would intertwine with mine. They consistently encourage everyone to communicate – ensuring that every member of the team's perspective. His ideas and contributions were critical to the testing phase as well, due to the addition of researching and adding *Mockito* to the work. This allowed us to run the testing phase especially smoothly with little to know problem, to further refine our work.

While performance in his own areas was consistently strong, Jamal could benefit from more collaboration between team members. By talking directly with people whose code were affected by his work and taking it upon himself to lend a hand in areas that would impact his own workload, it would allow for the project to run more smoothly and help stop problems from occurring further down the line. However, this did not overshadow their many contributions to the team. In summary, Jamal was a dedicated and reliable team member whose efforts were pivotal to the project's success. Their skillset, communicative nature and commitment to excellence made working with him a highly rewarding experience.

## Ben's Reflection

Josh was an integral part of the team who contributed his thoughts and strengths to bring success towards this project. As meeting Josh for the first time, I can say that he is someone who is very easy to collaborate with and a valuable colleague who shares guidance in areas where we might have some doubts on. One of Josh's roles was to set up Gradle and databases using Docker, which was critical to our project as all further methods would heavily rely upon it. By utilizing his technical strengths, it provided aid and strength to the integrity of the project and ushered further developments. One of Josh's assets that was displayed during this group project was his attention to detail and making sure that the plan moving forward was one that considered every factor and how it would replicate in the real world. Especially when planning on how to go about the project, Josh provided valuable suggestions that challenged certain aspects like the quality rating and the database structure, which sparked a much more innovative approach to the problem. This forward thinking allowed us to continuously adapt our implementations, ensuring that it was of the highest quality that could be proposed in an actual market. Moving forward, I hope Josh continues to exhibit his innovative thinking and be confident in challenging certain aspects of a project's outline, in order to provide a much more efficient and thorough implementation. Additionally, I hope he continues to strengthen his technical side, and offer creative ways to go about problems. Overall, Josh's contribution to this project has been critical to our development, and with his friendly demeanor and structural guidance towards the planning, I look forward to work with him again.

Collaborating with Arman has been a very valuable experience. Throughout this project, Arman has been very insightful with his views on how we should progress in our project and has shown himself to be reliable, in terms of getting handed a task and knowing it will be provided with a strong structure and of high quality. One of Arman's roles in the project was creating a user for our system. It involved creating a user with a given name and generating its ingredients, which included certain mandatory ones and randomly selecting others, so that it could always generate recipes that match with the user's ingredients. The method would allocate random quantities to each ingredient. As shown by this example, Arman has evidently displayed his strong knowledge in coding in Java and his ability to create such complex functions with ease. This contribution was definitely key when moving forward with the project in a succinct manner, as other functions depended on the user function to work. As well as his technical side, Arman provided an uplifting work environment, which was crucial during times when we came across adversities, such as tests not passing or confusion with how to move forward. This behavior was noted well as it motivated all of us to continue persevering, which is a key asset to have when in a group setting. Moving forward, I would urge Arman to continue to display his uplifting manner and exhibit his knowledge on Java programming. Additionally, he might want to consider updating his progress on tasks he has been given on GitHub regularly, so that everyone in the group has an idea on how he is progressing, and can offer help if need to, or start planning their own functions. Overall, it has been a great experience working with Arman and I look forward to work with him again, in the future.

Collaborating with Jamal has been a great experience. Since it was the first time I have met Jamal, I can say he was very friendly and calm and always offered great advice and guidance, which ensured the continual progress of the project. Jamal's main task was creating the quality rating function and returning recipes based on the availability. Jamal was able to provide such functions with good levels of complexity, which helped coincide with our functions. Throughout the project, Jamal displayed his strong knowledge in Java and was able to explain his code thoroughly, which was useful when understanding the flow of logic in his code. He especially carried this across by updating his progress on GitHub consistently, through the use of issues. One of Jamal's main assets that was notable throughout this project was his keen eye in research. With his in-depth research in Mokito and how it can aid with the testing aspect of our project, his ideas came into fruition and was critical in the development of our test case. As a result, his research contributed to the quality and professionalism of our project, which is a notable feat. As well as displaying his technical and R&D strengths, Jamal demonstrated a calming working environment, which was essential in a group setting. One example would be that Jamal was able to provide a calm response, when opinions differed, which is necessary in group projects, so as to mitigate any potential conflicts. Additionally, as a colleague, Jamal was very helpful in offering his guidance whenever I was confronted with problems with my Gradle and Docker, which once again highlights his friendly approach and his want to be a help to others. Moving forward, I would urge Jamal to be more confident with saying his views on aspects of our project, such as implementation, as they are opinions to consider and offer great importance, which could lead to further improvement in how we carry out our project. Overall, Jamal has been a great colleague to work with and I look forward to work with him again.