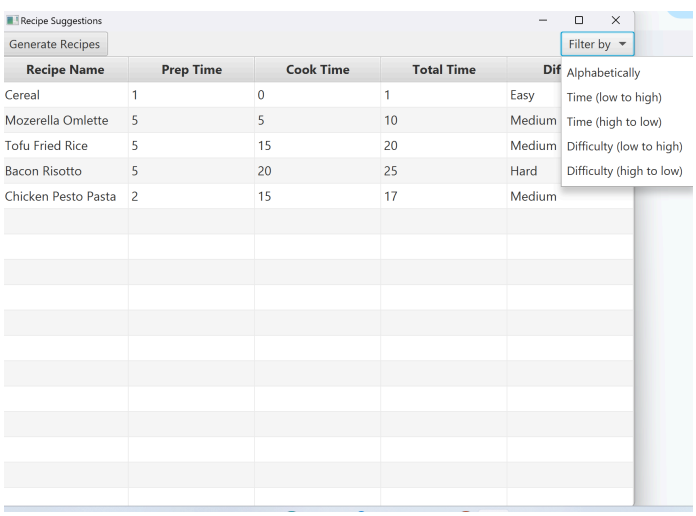
The Report

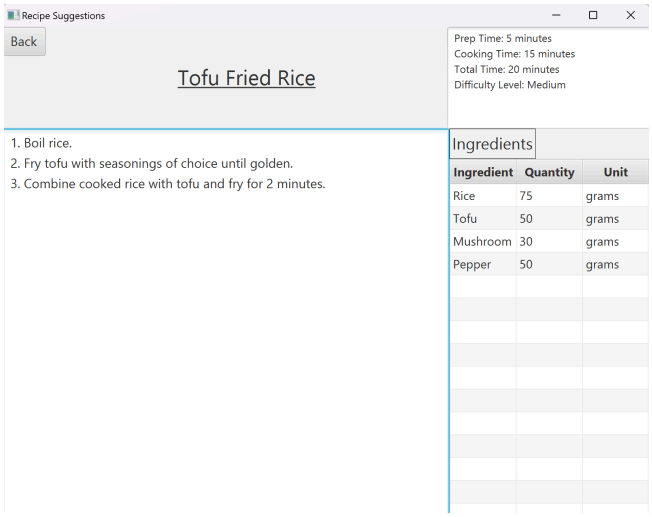
Authors: Jessica Greene, Elita Tuscano, Amelia Williams

What Was Accomplished

In this team project, we developed a fully functional prototype of the application described in Jessica Greene's EDR, showing that the design is viable. The overall technical accomplishment of our project was the implementation of a multi-threaded client-server application to provide a user with a list of suggested recipes. The prototype integrates an in-memory database using H2, a backend using Java, and a user-friendly interface built using JavaFX.

To evaluate the viability of our design, we considered both the EDR goals and the design's technical feasibility. The main aim of the design is to display a list of recipes, ordered by how well they match the ingredients available in the fridge. Additionally, clicking on any recipe should display further information, such as instructions and ingredients. An extension to the main goal is adding the ability to filter by user preferences. As shown below, our prototype demonstrates these essential features, therefore meeting the outlined goals. Furthermore, the prototype demonstrates the feasibility of the technologies described within the EDR, alongside additional technologies we selected to enhance the functionality. Therefore, we believe that we have successfully implemented a prototype of the chosen design.



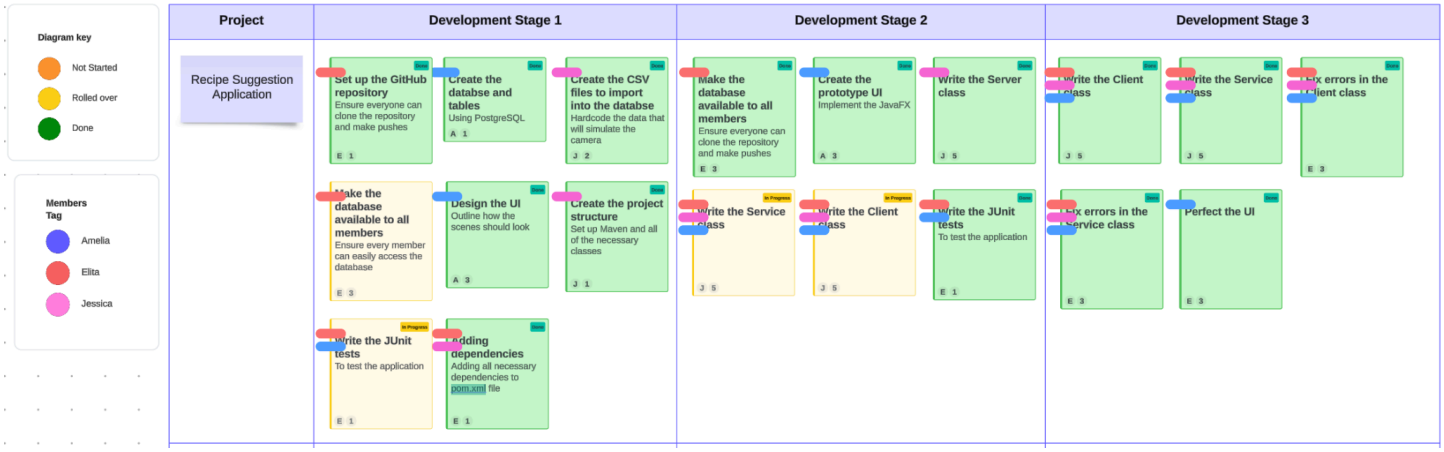


How it Was Accomplished

During our first group meeting, we discussed which methodology would be most suitable for our project. While we considered Agile methodologies, given their widespread use in project development, we ultimately decided to use the Waterfall methodology. We chose this based on the fact that the requirements were clearly laid out in the EDR and unlikely to change. This, paired with the small scale of the prototype, made Waterfall a better fit for this project.

Our planning stage started with an analysis of the design in the EDR, to create a product specification with clear requirements using the MoSCoW technique. Using this specification, we divided the project into distinct

and manageable tasks, which were organised into three week-long development stages and assigned to team members. We displayed this in a diagram, allowing us to easily update our individual task progress, and view the project's overall progression. While we aimed to follow these stages as much as possible, if a task was not completed, it was carried over to the next week.



During the development stages, we held regular group meetings to provide detailed updates about our individual tasks and discuss if any changes needed to be made to the plan. We tracked these updates in a shared document, which we later transferred to the issues section in our GitHub repository. At the end of each development stage, we held longer group meetings to review our work from that week and amend our development stages accordingly. For instance, when we were not able to complete our JUnit tests in time, we rolled this over to the next stage.

As for our project structure, we decided to use Maven as our build automation tool. Not only was it the build tool that we were most familiar with, but it also aided us with automating dependency management. This was especially useful when integrating multiple dependencies such as H2, JavaFX, and Mockito.

Using features such as branching and branch protection rules, we were able to develop together at ease without jeopardising the master branch. Specifically, we made it so that a pull request had to be reviewed and accepted by all members before it could be merged with the master. Implementing continuous integration prevented individual branches from significantly diverting away from the master branch, as each member integrated their changes multiple times a day. Using GitHub Actions, we ensured that before a pull request could be merged with the master, it was also required to pass integration testing to confirm the project worked as a whole.

We initially decided on using JUnit to test the functionality of individual methods, however, it proved difficult to separate the Client and Server components to allow for individual unit testing. As a result of this, we made JUnit tests to check methods that could be easily isolated, such as attending the Client request in the Service class, and testing the connection to our H2 database. Due to the limited number of JUnit tests we could write, we ultimately carried out the majority of our tests manually. This allowed us to perform thorough tests where we compared the actual and expected outputs from the database as well as evaluating how well the features implemented in the prototype met those detailed in the EDR. This ensured that we met all of the test requirements before moving on with any related code.

When we first made the database locally, using PostgreSQL as outlined in the EDR, we planned to implement version control for the database through Liquibase. However, after facing challenges with Liquibase implementation causing us to fall behind on our timeline, we decided to re-evaluate our development priorities. This ultimately led to us using H2, due to the ease of its implementation into our project. Although this decision limited the scalability of our prototype, our main priority was to ensure we could begin testing our application, which required connecting to and querying the database. Upon reflection, making use of Docker with our original PostgreSQL database would have made our implementation more robust and scalable, particularly for handling a larger dataset.

This prototype proves that the basic project structure outlined in the EDR, alongside additional tools and techniques, such as Maven, produces a functional prototype. Although we made the change to switch to H2, this was purely for ease of implementation, and we believe that the proposed design using PostgreSQL is still achievable with the use of Docker. This prototype shows the design's technical feasibility, as well as the functionality of the proposed features and therefore validates that the overall design is viable.

Reflections on the Members

Jessica about Amelia
Amelia was not only extremely dedicated to her allocated tasks, but keen to help wherever else she could. Regularly throughout implementing our prototype, she offered her assistance and asked if there were any tasks she could help me with. Throughout her design and creation of the User Interface, she consulted me and Elita to confirm that we were happy with the layout and presentation. She also provided us with detailed updates on her progress multiple times a day to ensure we knew the most current status of the User Interface, which proved particularly useful when working on the Client and Service methods to return data from the database to be printed to the User Interface. Amelia accomplished a fully-functioning User Interface to a high standard that met all of the requirements. She surpassed the initial requirements, constantly using her initiative to try different layouts and add additional features.

Towards the end of the project, I encountered unforeseen issues with the functionality of the Client class. Amelia willingly stepped up and helped me to debug my code, whilst also fixing one of our JUnit tests. This further showed me the value that she added to the team and I would definitely highlight teamwork and communication as two of her main strengths. In any further projects that she takes on she should continue to make use of these particular strengths, however I would encourage her to take care of herself and not take on an overwhelming workload by over offering her help to others.

Jessica about Elita
Elita proved herself to be an extremely valuable member of the team. She was eager to find new technologies, such as Liquibase and H2, to aid the progress of our prototype, especially when we ran into difficulties. She was extremely dedicated to her work and completed every one of her allocated tasks to the highest standard possible in the time available. Despite running into multiple issues, she always remained resilient and maintained a positive attitude. She spent multiple days integrating the new technologies into our project, and although this caused her to fall behind on one of her tasks, the work she carried out was vital in ensuring our prototype ran seamlessly.

Throughout the project, Elita was vocal about her progress and spoke up when she ran into any significant issues. After falling behind on one of her tasks, she immediately communicated this to the group, allowing us to decide together the next best possible route of action.

She explained in detail each of the new technologies she was researching to the team so that we could best understand the progression of the project as a whole and offer our assistance. I really appreciated this from her as I felt that I was constantly up to date with her current work. She should continue to do this in her future projects as it will be greatly appreciated by her colleagues. Although it is evident that Elita enjoyed researching the new software engineering techniques, I would encourage her to share this workload more so that she does not fall under unnecessary stress.

Amelia about Jessica

During our team project, Jessica consistently demonstrated her dedication to completing our project to the highest standard. She went above and beyond to ensure that we were on track with our plan, often putting in long hours to finish her assigned tasks. Jessica never needed reminders to complete her tasks, and often completed them ahead of time, allowing other team members to utilise her finished elements when developing and testing features elsewhere in the project. One of Jessica's main responsibilities was implementing the client-server architecture for the server. Her swift completion of this task aided me in developing various methods within the client class that were related to the UI.

Jessica also showed great flexibility during this project. She was always willing to assist in debugging code whenever the team encountered challenges, all while staying on top of her own tasks. On one particular occasion, Jessica and I collaborated to resolve a complex bug within the client class. We effectively exchanged ideas and shared our knowledge of the specific functions we had each written, eventually being able to identify a solution. I believe this experience highlights Jessica's ability to collaborate directly and communicate effectively with others.

In the future, I would encourage Jessica to seek help from her team members more. While her ability to complete tasks quickly is impressive, not taking on so many tasks at once and asking for help when needed would help to balance her workload and maintain her high standard of work in the long term.

Amelia about Elita

During our team project, Elita was a vital member of our team. She was extremely proactive in researching technologies beyond those we were initially familiar with, and implemented them with speed and efficiency, despite having no previous experience. In addition to this, Elita ensured that we were able to understand these technologies, making sure we were able to make use of them and use them in our tasks when applicable. Her technical knowledge and in-depth research enabled all members of the team to gain a thorough understanding of these, for which I am very grateful.

Throughout the development of our prototype, Elita maintained strong and open communication. When she realised that completing the JUnit tests in their allocated development stage was not possible, she swiftly communicated this. By being honest and proactive about this, we were able to assess and re-assign the task without hindering the project's progress significantly.

Another skill Elita demonstrated during our group project is her incredible persistence. During our project, we encountered a few frustrating bugs, particularly relating to receiving the output from the server and displaying it. Elita was responsible for debugging many of these, sometimes spending hours locating and fixing them. I deeply admired her perseverance in this and as a result of this, we were able to get our prototype running how we wanted it to.

In future projects, Elita should maintain her high-quality communication as it has proved very valuable to her teammates. She should also continue to research and explore different technologies that she can implement in her projects as this can add significant value.

Elita about Jessica

Jessica was an integral member of the team, consistently demonstrating a strong work ethic and playing a key role in the development of the prototype. She was responsible for creating the file structure for our application, as well as writing the Server, Service and Credentials classes. She also aided in the development of the Client class, alongside Amelia. Jessica always made sure that her work was completed to the highest possible standard and this is reflected in the quality of her work. While developing the Client class, she collaborated effectively with Amelia, who was developing the User Interface, and was able to update the Service and Client classes to handle the new features implemented in the UI accordingly. She made sure to communicate these changes clearly, keeping us constantly updated with her progress.

Throughout the development of the project, Jessica has remained flexible and supportive. She willingly took on multiple roles, and if we fell behind on a task during a development stage, Jessica took the initiative to ensure it could be rolled over to the next stage or completed it herself. She was also incredibly helpful at points where Amelia or I were stuck on an error, and helped us debug and resolve our errors.

She has shown incredible dedication to this project, and her positive attitude has kept us going. In particular, Jessica should continue to maintain her high standard of work, and showcase her collaboration and teamwork in any projects she may complete in the future.

Elita about Amelia

Throughout the development of this project, Amelia has proved herself to be a valuable member of the team, ensuring clear communication on her progress and taking initiative whilst completing her tasks. Amelia was responsible for developing the User Interface and maintaining the Client class to support these features. She successfully completed these tasks and went above and beyond to implement extra features. During development, she would regularly ask for feedback from myself and Jessica, to ensure that we were up to date with her latest changes, effectively communicating her progress.

Amelia has shown incredible resilience and initiative whilst managing a tough workload. These qualities were particularly evident when she could not complete her tasks during the first development stage, due to preparing for a placement interview. She communicated this early on, allowing us to adjust our development stages accordingly, demonstrating her professionalism.

During development, Amelia took on tasks that either Jessica or I had fallen behind on, showcasing her adaptability and dedication to the team. When I had fallen behind on one of my tasks, writing the JUnit tests, Amelia willingly stepped in although she was unfamiliar with JUnit. She dedicated extra time and effort to

learn and implement the technology, ultimately completing the task to a high standard. Going forward I would encourage Amelia to continue using her strong communication skills, and to maintain her proactive approach in any future projects.

References

Lucid Charts, Diagram 1 [Online]. Available at: https://lucid.app/lucidspark/978c0b25-2614-41ed-96a2-280f0bcf80bf/edit?invitationId=inv_c3518f03-629f-4aaa-95d5-5a9cab2ea93b&page=0_0# [Accessed on 20 November 2024]

Java Connect to H2 Database [Online]. Available at: https://www.codejava.net/java-se/jdbc/connect-to-h2-database-examples#google_vignette [Accessed on 1 December 2024]

Mockito and JUnit 5 [Online]. Available at: https://www.baeldung.com/mockito-junit-5-extension [Accessed on 1 December 2024]

Interface Cached Row Set [Online]. Available at: https://docs.oracle.com/javase/8/docs/api/javax/sql/rowset/CachedRowSet.html [Accessed on 8 December 2024]

Interface Result Set Meta Data [Online]. Available at: https://docs.oracle.com/javase/8/docs/api/java/sql/ResultSetMetaData.html [Accessed on 8 December 2024]

Retrieve Entire Row With Result Set [Online]. Available at: https://stackoverflow.com/questions/16882971/retrieve-entire-row-with-resultset [Accessed on 8 December 2024]

The structure of our code was taken from Assignment 3 of the the Full-Stack Application Development module in First Year. (University of Birmingham Computer Science, 2024)