**Final Report for Year Long Internship**

Prepared and submitted by

| Name: | Ng Zen Yu | **P2123011** |
| --- | --- | --- |
| **School of Computing** | | |
| **Diploma in Applied AI & Analytics** | | |
| Company Name: | QueueCut Pte Ltd | |
| Co. Supervisor: | Tony Teoh | |

*In partial fulfilment of the requirements for the module*

*ST2302\_WSH : Year Long Internship*

**Lecturer: Dr Saw Vee Liem**

**Submission Date: 12/1/2023**

**Word Count: 1492 words**

**Table of Contents**

Background - Page 3

Solution Development - Page 3

Development Process - page 4

Final Phases of Project - page 5

Evaluate Findings - page 9

Conclusion - page 11

References - page 12

Appendix - page 12

**Background:**

I work for QueueCut, a food delivery company offering customers the option to skip queues by ordering food that is collected in person or delivered from our app. Our app provides zero commission fees for merchants, letting them sell worry-free on our platform.

After discussions with our boss, we decided that the interns would develop a chatbot and recommendation system for the app. I leveraged my Android Studio skills I learnt from previous projects to develop the front-end, and knowledge that I gained in python and AI models from our course to develop the recommendation system.

**Solution Development**

**Highlight how you have done the following:**

**• Resource Used (Manpower, tools and equipment)**

**• Artifacts and deliverables produced during the development phase.**

Deliverables:

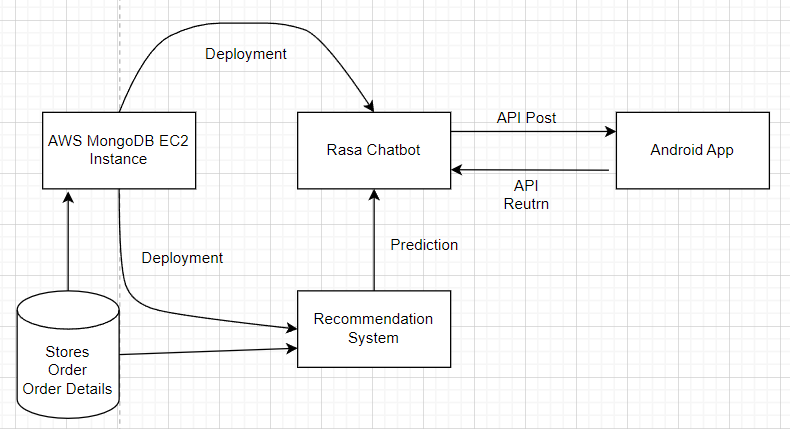
We split the workload amongst the 3 interns, where I was in charge of developing the app and recommendation system, Pratik, the chatbot development, and Samuel, the development of the backend server.

Firstly, we researched on what method of chatbot would be the most practical for our use case and decided on Rasa.

Secondly, I was tasked with developing the front end of the app as I had most prior experience with Android Studio.

Thirdly, a recommendation system that returns 5 different food items for the user. This is another component that I was in charge of.

Lastly, a server with APIs for the chatbot to interact with the android app by creating an AWS server/Instance and adding MongoDB to an EC2 Instance.



**Development Process:**

Prototype Phase:

I was tasked with developing the Android Studio app and linking the app to the chatbot.

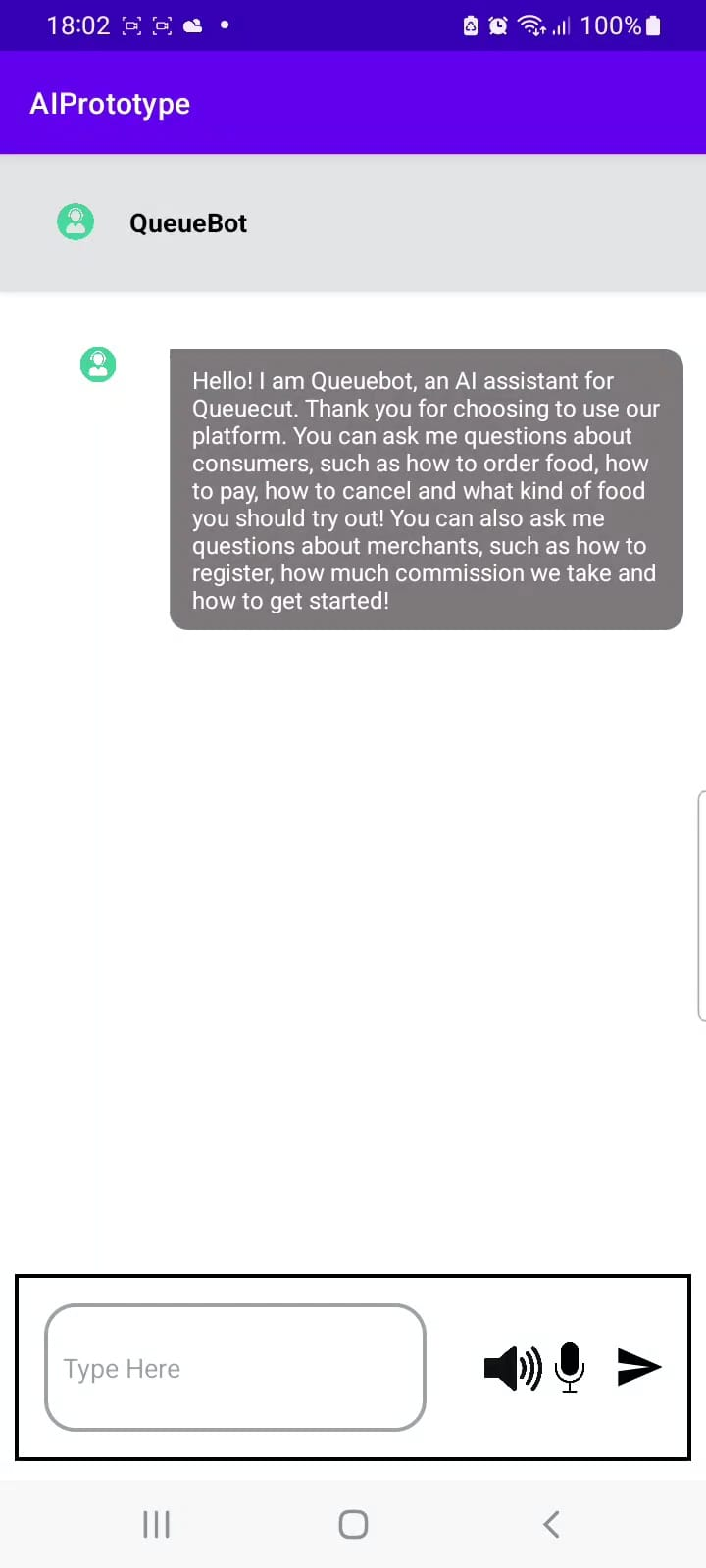
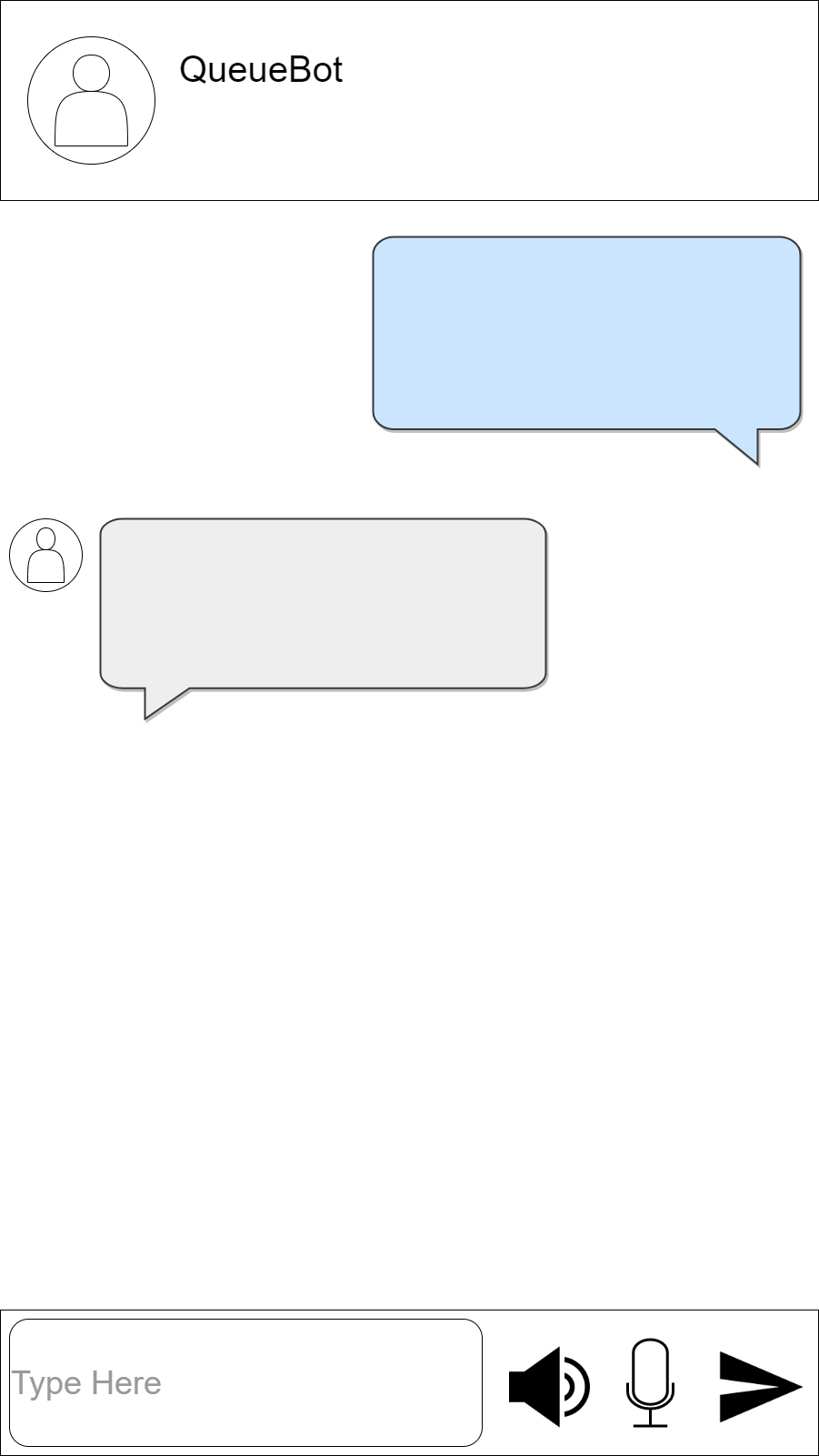
First Prototype:

Another intern, Pratik, had done a previous chatbot designed for end-to-end conversations between users. I referenced his code to construct a user-to-bot interface. I added 2 additional functions, a speech-to-text and text-to-speech function, for users to talk to the chatbot verbally, and for the chatbot to talk back to the user. Thereafter, I incorporated APIs hosted on Samuel’s computer into the app, enabling users to post the messages via API to the Rasa chatbot and for it to return the appropriate response. The app then processes the response and displays it as a speech bubble.

Explored Methods:

I experimented with diverse chatbot approaches, including using the torch module and NeuralNet for intent-based training. This chatbot was able to detect inputs asking for assistance and redirect their personal details to our customer service officer via email. However, as it was more difficult to integrate the recommendation system and was not as established a method as Rasa, I passed Pratik the code to use as reference.

Next, We discussed the design and came up with a wireframe below. I then built the app around the wireframe and added the Post API thereafter.



Solving Issues:

In Android Studio, I encountered crashes for unforeseen reasons, and even after reading the error logs, I was still not sure what was the problem. To solve this I used breakpoints, a feature in Android Studio that pauses code execution at specific points. This allowed me to troubleshoot errors much easier without having to constantly rerun the app.

**Describe the final phases of your project i.e. coding, testing and deployment.**

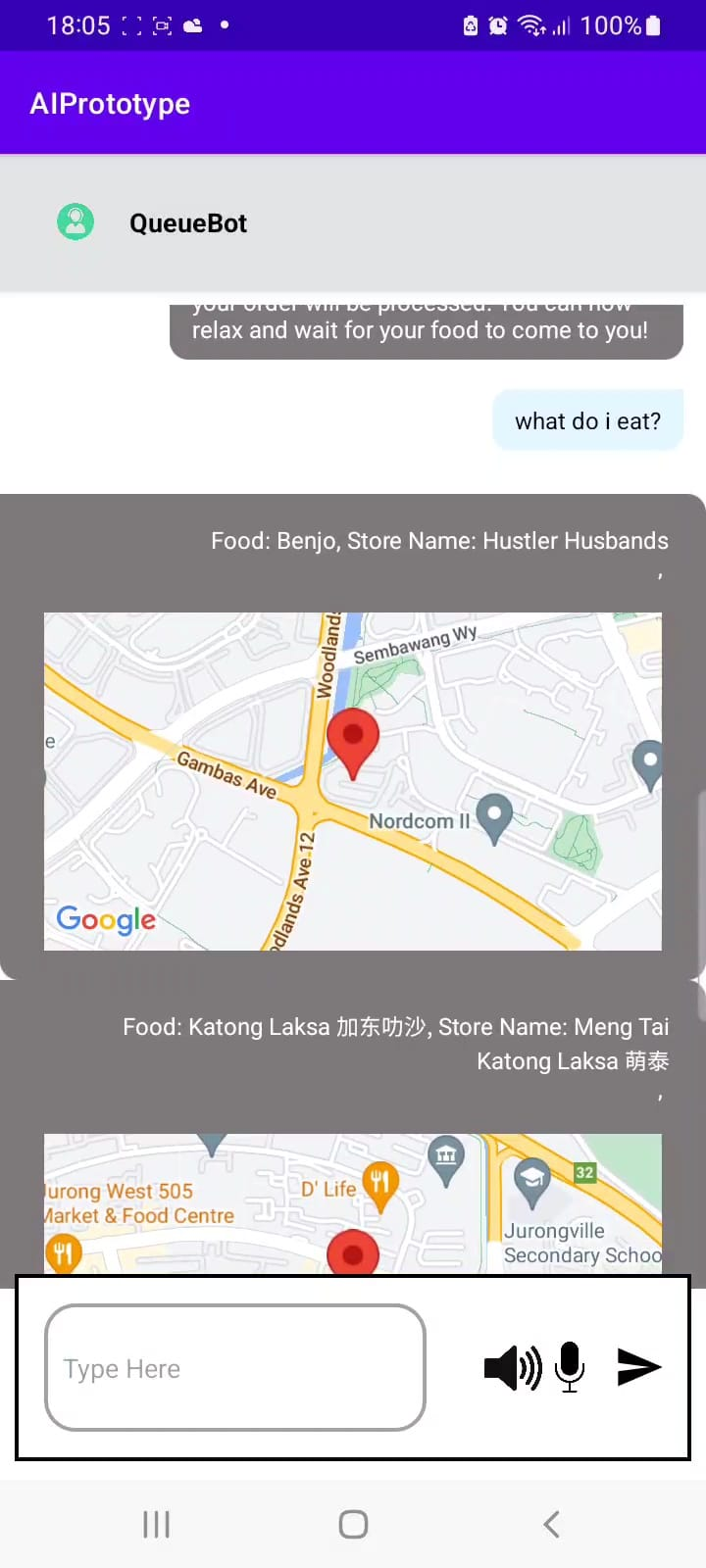
**o Methodology / Strategy used**

**o Techniques used in solving problems**

**o Set up and outcomes of testing and deployment**

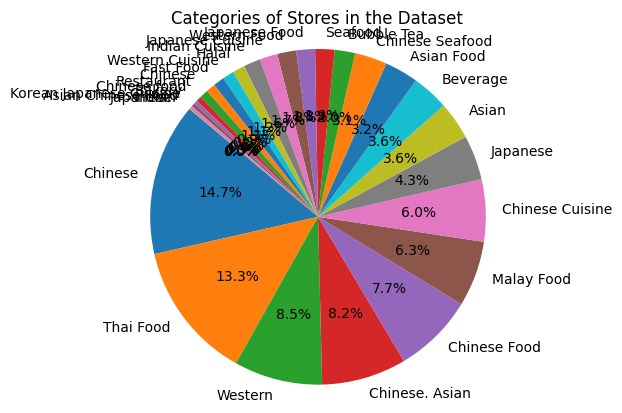
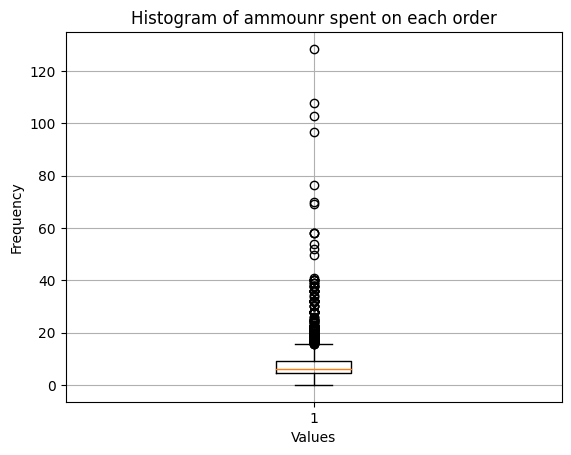
Further Development:

We focused on building a recommendation system as per our boss's request. With Pratik’s help, we settled for a python module called surprise, that specialises in recommendation systems. We integrated it with the chatbot, which triggers a custom action allocated to the recommendation system. It then provides a recommendation based on the question from the user. Additionally, I implemented a feature linking Google Maps to the app to allow the user to see a store to find out the store’s location.



Explored Methods:

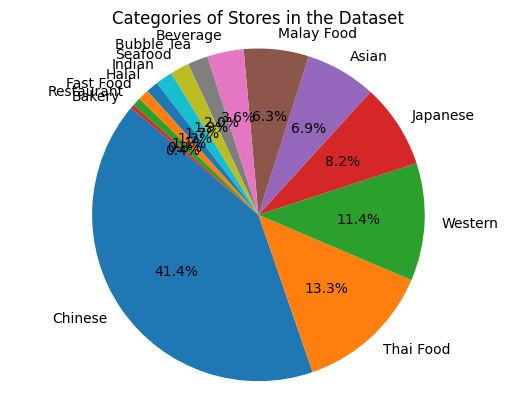
Firstly, I retrieved the data as an API from our database and imported it into my python file. I merged all the datasets by their respective keys and was left with a dataset with 173 columns and 2534 entries. To gain further insights, I used a box and whiskers plot to find the average amount a user would spend on a single transaction, and a pie chart to visualise the distribution of food categories along with their respective percentages.



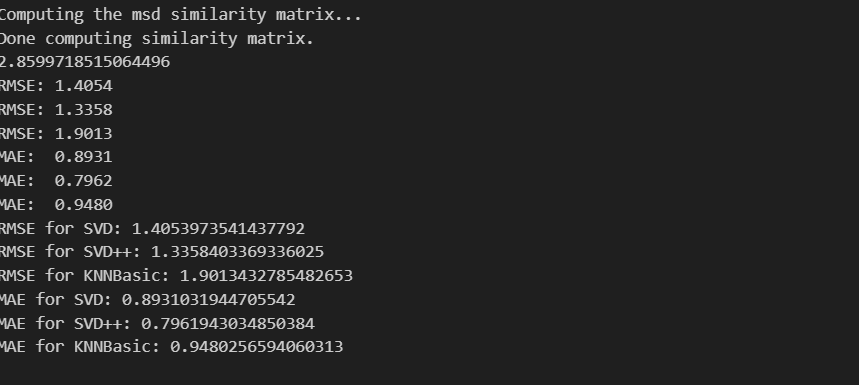
Solving Issues:

As the recommendation model would allow the user to filter the recommendation of food by the food categories, there were currently too many categories, many of them being very similar. Therefore, I summarised the categories by replacing the similar categories with just 1 category (eg. if the category contains the word ‘Chinese’, it is placed in the ‘Chinese’ category.)

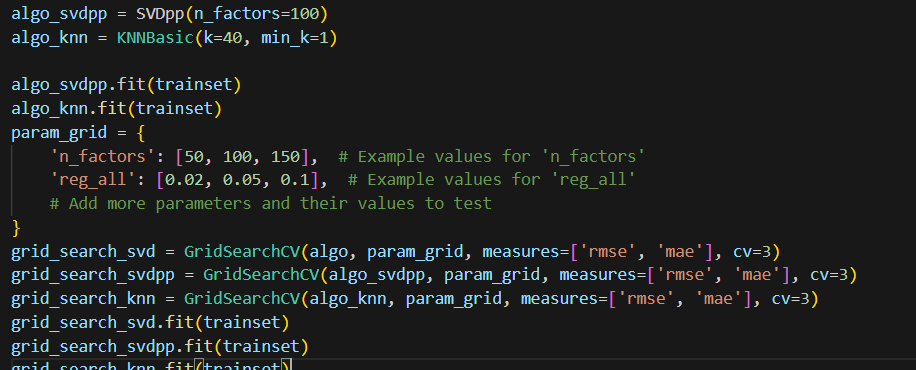




When developing the recommendation model, I was unsure of what training method to use in order to achieve the most optimal results. However, I was able to determine the most effective training methods by printing the RMSE and MAE scores to determine which training method was the most suitable. By this method, I was able to determine that SVD++ was the most accurate of the training methods that I tested.



Thereafter, I used GridSearchCV to conduct hyper parameter tuning to attain a better score. I defined a parameter grid with the respective parameters and predicted the model on the new set of parameters.



**Evaluate Findings**

**Things that went well**

* Despite being busy with other customer projects, I was able to complete the project smoothly.
* I was able to learn how to use the surprise package to create a working recommendation system.

**Things that went badly**

* Finding solutions for the recommendation system was hard as we had not constructed a recommendation system before, and had to learn it from scratch.
* Our data was not suited to build a recommendation system as there was no rating for the foods from each user. We eventually used the number of times a user bought a certain food as the rating system and trained the model on that basis.

**Discussion of problems encountered, and decisions made**

* Samuel had their computer compromised with ransomware in their mongodb. The data that was lost was not crucial and we were able to resolve the problem by cleaning the device.
* We had a lack of manpower and I had to juggle other customers’ projects as well. With this problem, I was able to manage my time between this and other projects well and complete them in a timely manner.

**Summary of changes to original scope or plans**

* Originally, our project aimed to deliver an Android app with a chatbot capable of recommending food. To enhance user experience, I suggested adding the location of the store on the app to allow the user to estimate delivery or travel time for in-person pickup.
* Furthermore, I added a feature that filters the prediction of the recommendation system by store category (.ie Western,Chinese etc.) .
* For the chatbot, our boss wanted a more versatile chatbot,to not only answer frequently asked questions(FAQs) but be able to have a conversation with the user. However, due to limitations with tools (python, java etc.) and manpower constraints, we were not able to develop a more complex chatbot.

**Potential improvements**

* Allowing the chatbot the ability to respond to more common questions by adding more intents into the chatbot instead of the current FAQs and basic greetings. Furthermore,we can develop a handover feature for customer service officers to take over the chat when the user requests for help.

**Next steps and follow-up actions**

We are currently handling the hand-over of the project to the other co-workers. To prepare for this, we documented our development process, how to set up and run the chatbot, and possible improvements for the project to move ahead in the future.

**Key messages for future projects**

* We implemented regular checkpoints, and a demonstration prototype ready similar to the Agile Methodology. This fostered accountability, ensuring deliverables and a working prototype was completed by the end of a certain timeframe.
* Working in a real-world scenario gave us differing perspectives to what we experienced in school. We realised that updating each other and our superiors was very important to ensure coordination and garner valuable feedback from our superiors.

**Reflection on alternative approaches**

* We could have used a Generative Pre-training Transformer Model (GPT Model) like ChatGPT instead of a Rasa chatbot. This allows for better Natural Language Processing(NLP).
* Using pre-loaded replies for the chatbot to limit the number of queries asked. This would aid the chatbot model prediction accuracy. We did not do this as the boss wanted the user to be able to answer open ended questions.

**Feedback from customer/user**

We consulted our supervisors and bosses during the development phase of the app for feedback. Overall, they were mostly pleasantly surprised by what they saw. Suggestions included UI changes, slight content changes for the chatbot and recommendation system, and a feature that allows the customer service officer to assist the customer when requested.

**Conclusion**

In conclusion, we successfully developed an Android app featuring a Rasa-powered chatbot and a recommendation system. Despite having to juggle between multiple projects and dealing with unforeseen issues, we were not only able to achieve our original objectives, but also added valuable enhancements like store location and category-based filtering for the recommendation system. Prioritising cooperation and feedback from supervisors ensured consistent improvement to our development and resulted in a polished final product.

# **References**

Flateman, J. (2021) *Using surprise in python with a recommender system*, *Medium*. Available at: <https://medium.com/@jonahflateman/using-surprise-in-python-with-a-recommender-system-2d6030140926/ >(Accessed: 11 January 2024).   
  
QueueCut 2023, QueueCut Website, Accessed 11 January 2024, <<http://queuecut.sg/>>.

**Appendix**