(1) Personal contribution to group project

System

- I am the domain expert for this project, given my experience with childcares as a parent. I also came up with the domain expert rules for this system.
- Priyansh coded a web scrapping code for this project to scrape reviews from KiasuParents, Priya pass the reviews from Priyansh's code through a model to determine the sentiment of the reviews and Edwin built Decision Trees with Genetic Algorithm to determine the likelihood parents will accept the offer from a particular childcare. Priyansh also coded a script to determine rough distance between two postal codes. When building the systems, I incorporated the above into the systems.
- I built three systems from scratch and set up MySQL database connections for this project.
    - Childcare Matcher system for childcare centres: Python/Flask system with user interface. The user interface adapted a login page from this on codepen and freelancer bootstrap template. The adaptation includes Jinja template such that information can be pass to or from python/flask server. This system is connected to MySQL to grab parents' data.
    - KIE server for parent recommendation system: This is a Springboot application with KIE rules, linked to Google API to get distance between postal codes and linked to MySQL database to get childcare information.
    - Parent recommendation system: This is a python/flask system with user interface. The user interface adapted a template from freelancer bootstrap template. The adaptation includes Jinja template such that information can be pass to or from python/flask server. This system is connected to MySQL to send/receive parents' data and also linked to KIE server.

Documentation

- Contributed to the project report
- Video the system for the project video

(2) what learnt is most useful for you

The building of the KIE server is most novel for me, as I am quite used to building python/flask systems. This Java programming experience will be useful for me if I wish to take on jobs that requires such knowledge, not to mention incorporating rules in the KIE server. I also typically build standalone applications, so this is the first time I build multiple systems and allow them to interact with each other via API. This hands-on experience building APIs will be useful as well.

(3) how you can apply the knowledge and skills in other situations or your workplaces

Currently, the system we have in my workplace is huge and essentially a standalone platform with many modules (think of small software) that reside on it. Multiple developers interact with this platform and in time, it takes a lot of time developing on it as we are concerned what our changes will affect other modules. Hence, perhaps separating the modules and allowing them to interact with each other through API might be a better solution, than hosting all modules essentially together.

Luckily for us, the rules for these modules are quite stagnant so we just code it into the system instead of having a separate rule engine. However, it is likely in time we will have to produce a module with frequently changing rules. If so, we can consider using rules engine.