Lessons Learned

What you learned from it (challenges we met)

- This application involves some complex interactions between users (e.g. when to claim, deliver and accept items) so we should be more specific with the procedure our app will follow.
- Works from different teammates overlapped a lot with each other, so we learnt to make multiple branches to ease this problem and try to communicate more effectively.
- Through the working progress, we narrow down the scope of our app (from targeting general public to MIT students, setting fixed pickup locations and limited grocery stores).

Evaluation

What went well

- Time management
- Efficiency in deliverables and meetings
- Implementation of tasks assigned (in general)
- Communication after learning from initial mistakes
- Asking for help
- Code reviews and discussion of choices (design, implementation, etc.)
- Responsiveness of everyone
- Narrowing down of scope for the project and the time we had

What could have been improved

- Some design decisions can be made earlier so we won't waste time changing codes (such as not having the Ratings model, and settling all the different scenarios where requester rejects / deliverer does not show up etc.)
- We put the model logic in the routing at the beginning, and had to waste time changing the code halfway through our project
- Sometimes, we were in a rush to push our own changes that we didn't realize that we broke someone else's code. In the future, we should always check this and ask for help in merging the code if we're unsure on how to fix the code.

What we could do differently in the future

- Be clearer and more specific with design decisions and make sure everyone understands them
- Be clearer with expectations on good coding practices

- Discuss, simulate, and evaluate use cases earlier so that possible problems with the flow of user actions could be discussed earlier