

# Sprint 2: Scoping

*Project Group 8: Yao-Fu Yang, Qiushi Li, and Rong Chang*

*Trello board:* <https://trello.com/b/61ULBOVX/grocery-store-data-support>

*Git:* [https://github.ccs.neu.edu/yangyaof/cs5500\\_sum2020\\_group8](https://github.ccs.neu.edu/yangyaof/cs5500_sum2020_group8)

## User stories

In this phase, overall, the store manager wants to know how the factors such as holiday, discount hours, lunch/dinner rush hours, and weather may change patterns of customers shopping, so that he can plan the operations more efficiently.

More specifically,

1. He needs to know **the traffic of customers in the top 3 hours of a day** over a week, and in addition to answer the questions such as,
  - which day has the most shoppers and which day has the fewest shoppers;
  - how this traffic may change on the different days of week;
  - how this traffic may change due to the weather, holiday, and discount event.
2. He needs to understand how the **weather** can affect the shopping volume and entry time, by exploring
  - ~ whether a bad weather on a Saturday may result in less shoppers than a Tuesday with a special senior discount event or even a normal weekday?
3. He needs to observe how much **the senior discount event** may affect the traffic of customers and shopping duration, for example, with the senior shopping hours moved to Wednesday 1-3pm, how the customers will change on Tuesday and Wednesday?
4. He needs to observe how the traffic of customers and shopping duration change if the dinner **rush hours** shift to different times in a day, such as 4-5 pm.
5. He needs all of the information from the csv file stored into a suitable **database** (either SQL or NoSQL) to support more sophisticated data analysis.

## Work planning

In this phase, the user wants to dig the data with more understanding on how the environmental factors may affect customers' shopping pattern. Therefore, our work on the data aims to expand the flexibility of the code and class methods. Such that, it may allow the user to observe how the shopping patterns change in association with the changes in holiday, discount hours, lunch/dinner rush hours, and weather.

In addition, we plan to store the csv data into a NoSQL database. Although we first considered using a SQL database because our current data presents information in a structured relational

(table) format, we opted for the NoSQL database to support evolving data requirements. Specifically, we imagine that there may be additional impacts of COVID-19 that are not included in the current dataset, requiring additional fields later. We also want to account for changes to the dataset when the situation with COVID-19 improves and/or resolves over time. Additionally, we feel that a key-value database may be suitable for our current dataset, where the key represents a predefined interval in hour(s) or day(s), and the value represents information for all corresponding shopper visits. Please see the listed plan of this phase of work below.

### ***Part I: development of methods and classes***

1. We will revise our assumptions on the daily amount of customers, distribution of entry time, and duration.
  - a. Only keep the normal daily pattern.
  - b. Create special patterns for special events/days.
  - c. Allow these special patterns to be applied flexibly.
2. We will add methods such as applySpecialDiscount(), applyBadWeather(), and applyNiceWeather(). Once a day is identified for any special event/day, onto the normal day's pattern, we will call these methods to add more customers with the identified entry time and identified shopping duration of these customers. We expect this to reduce the degree of hard-coding for special events/days, and increase the flexibility in applying these effects to different days/times of the client's choice.

### ***Part II: maintaining code integrity***

1. We will revise our code and include checks for erroneous inputs that might break the system. This will include customized exception classes with customized messages.
2. We will include additional unit tests that aim to cover edge cases. We will also explore integration testing once our unit tests are completed.

### ***Part III: move csv to a NoSQL database***

1. We will set up our environments to support a NoSQL database and make the necessary configurations to transform our data from the csv file into the desired structure in the database. We are currently considering a key-value database for data storage for now.