Project Group 1: Sprint 1 Summary

Weiqiang (Kevin) Huang, Chih-Ming Sun, Kareem Abdelshafy, Todd Brown

Resource:

repo: https://github.ccs.neu.edu/huangw25/CS5500-Project-Group-1
Trello board: https://trello.com/b/BpvBRp4O/cs5500

Snapshot of Progress

- We implemented the code to deliver a CSV with nearly all the model factors being driven by file/configuration

- We implemented very similar to our Sequence diagram that was presented with a couple of small changes.

    - We introduced MVC pattern to separate concerns of data generation and file writing

    - We introduced StoreModel interface and implementation to hide the complexities of multi-model approach

    - We (spearheaded by Kevin) discussed and did a small proof of concept to refactor the approach of three independent models to reflect encapsulating logic based on Customer Types. We decided the work was too much to take on during this sprint and may revisit later.

- We originally planned on adding some more variance to the project regarding making arrival times less uniform and total customers by day vary slightly. In the end we did not get to those stories – and remain a question of whether they need to be prioritized in future sprints.

Initial User Stories

Story Name	Completed
01-Generate-a-basic-CSV	yes
03-Enhance-CSV-to-accurately-relfect-Non-Holidays	Yes
02-Enhance-CSV-to-accurately-relfect-the-day-of-a-holiday	Yes
Enhance CSV to accurately reflect Day before a Holiday (Story was archived, just recovered – old name)	yes
04-Enhance CSV to accurately reflect the week before a holiday	yes
05-Enhance-CSV-to-accurately-reflect-Dinner-Time-rush	Yes
06-Enhance-CSV-to-accurately-reflect-Lunch-Hour-rush	yes
07-Enhance-CSV-to-accurately-reflect-Senior-Discount-time	yes
08-Enhance-CSV-to-accurately-reflect-Grab-and-Go-traffic-on-Very-Nice-Days	yes

Added Stories (Mid Sprint)

Story Name	Completed
10-gaussian-distribution-time-in-store	yes
11-read-the-average-time-in-store-from-csv	Yes
13-read-holidays-from-csv	Yes
14-read-weather-forecast-from-csv	yes
15-refactor-project-to-introduce-store-model-concept	yes
16-vary-senior-time-in-store-from-45-to-60	Yes
20-add-reasonable-holidays-to-holiday-calendar	yes
21-take-dates-as-console-input	yes
23-remove-average-traffic-map	yes
24-fix-parsers-windows-bug	Yes
25-add-remaining-traffic-factors	Yes
26-add-java-doc-where-missing	Yes
27-refactor-storemodel-reduce-coupling	Yes

Remaining Backlog

Story Name	Completed
09-poisson-distribution-customer-arrival	No
12-make-average-customers-per-day-vary-slightly	No
17-drive-main-from-command-line	Partial
18-senior-hours-as-input?	NO

Details and Challenges

Estimation

We elected to estimate the stories out on the order of "Days" and as we developed the software it became evident that our estimates were wrong. We didn't revise our estimation method, nor did we reflect on the estimates as we started a stories.

Process

We initially started the project in Scrum by writing and organizing the stories for our sprint. We formed late in "Sprint 0" as a result of two teams being combined. As we started Sprint 1 we were still "forming, storming and norming" and didn't really discuss another approach besides Scrum. We set **our goal** to be "delivering a CSV at the end of the sprint that takes in of the managers requirements as possible". We also discussed and debated "deterministic models vs stochastic models" and added cards for "gaussian distribution for time in store" and "Poisson distribution for customer arrivals" which became a guardrails for our approach.

Over the course of the sprint we transitioned (unintentionally) to a **Kanban approach**. We all have different availability and Kanban worked naturally for us (we didn't change the board). We employed considerable **Pair programming** and **Mobbin**g over the course of the sprint as well as discussing Code MR and Unit Test Coverage metrics on occasion.

Software Design

Our initial design was based on three independent models which could be coded and tested on their own. They covered three pieces of the customer visit:

- When they arrive,
- What type of customer they are, and
- How long they spend in the store.

Kevin challenged that approach during the sprint, delivering a proof of concept that we considered moving to. That pivot was deemed too large at that point in time we introduced a key component: concept of the "Store Model". This allowed us to abstract the implementation of generating Customer Visits from the controller, removed the need of the controller to have to interact with three components of the model, and in the future allow us to refactor or introduce alternative StoreModels without rewriting the Controller (or the view).

We tried to make the customer arrival flow as a gaussian distribution. However, because of the information in the requirements <average (25 min), minimum (6 min) and maximum (75 min)>, we chose not to use the symmetric distribution around the mean. Instead we chose a Chi-Squared distribution with k=3 and the mean=25 with some check to keep potential outliers between 6 minutes and 75 minutes.

Another item that challenged us was **removing hard coded factors** (“magic numbers”) from the models into a configuration files. We moved the Holiday Schedule, Weather Forecast and other model factors into three separate files (and corresponding classes). These three classes are passed into the StoreModel for reference.

CodeMR analysis shows that the relationship between TrafficFactors (one of the configuration classes) and related class (CustomerArrivalModel) as having “medium-high” lack if cohesion. WE tried a POC to break the TrafficFactors into smaller classes and it didn’t change the metric, therefore it was reverted. We were able to use deMR to find spaces to refactor smaller changes, such as reducing the coupling on StoreModel by introducing the StoreModelBuIlder.

The conversation on **determinism vs stochasticism** is one we came back to regularly throughout the sprint. One example is the increases in customers during lunch and dinner rush. We assumed 15% more traffic at lunch and 10% more traffic at dinner. During the final sprint review, we discussed the large variance in lunch numbers:

day-of-week	Mon															
Count of type	Column Labels															
Row Labels	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Grand Total
23	52	53	53	53	53	53	61	53	53	53	53	58	56	53	43	800
Dinner												1	2			3
Lunch							13									13
NonSenior	47	46	44	49	47	41	39	44	42	48	44	49	47	44	37	668
Senior	5	7	9	4	6	12	9	9	11	5	9	8	7	9	6	116
24	52	53	53	53	53	53	61	53	53	53	53	58	56	53	43	800
Dinner												5	2			7
Lunch							9									9
NonSenior	48	49	45	43	47	45	40	45	44	46	45	45	44	45	40	671
Senior	4	4	8	10	6	8	12	8	9	7	8	8	10	8	3	113
25	52	53	53	53	53	53	61	53	53	53	53	58	56	53	43	800
Dinner												12	2			14
Lunch							5									5
NonSenior	43	41	47	45	46	41	47	44	50	43	43	43	49	44	36	662
Senior	9	12	6	8	7	12	9	9	3	10	10	3	5	9	7	119
26	52	53	53	53	53	53	61	53	53	53	53	58	56	53	43	800
Dinner												7	1			8
Lunch							9									9
NonSenior	45	44	45	47	41	47	47	46	44	44	44	45	45	43	40	667
Senior	7	9	8	6	12	6	5	7	9	9	9	6	10	10	3	116
27	61	61	61	61	61	61	69	61	61	61	61	66	64	61	50	920
Dinner												6	7			13
Lunch							10									10
NonSenior	50	49	45	48	53	47	47	51	51	46	50	57	48	52	46	740
Senior	11	12	16	13	8	14	12	10	10	15	11	3	9	9	4	157
Grand Total	269	273	273	273	273	273	313	273	273	273	273	298	288	273	222	4120

This pivot table shows traffic over a five-week period (month of June extending into May and July to capture full weeks), filtered to Monday traffic only. Our assumptions (15% increase in lunch traffic) indicates an average of 8 additional customers at lunch on Mondays. As seen in the screen shot, lunch traffic varies between 5 and 13 users with a mean of 9.

We are comfortable with the result and feel we model the store accurately. Testing over this variance is a challenge. For the unit tests we ran high numbers of iterations to achieve a large enough sample to validate our results. The integration test of StoreModel could not rely on a direct increase in iterations. instead we had to increase the average number of customers per day, which increased the number of iterations behind the scenes and achieved the same affect.

Still, we would want to present these results (variance and potential options) to the manager for feedback.

Bugs / Questions

One outstanding bug exists parsing the various input files from a windows machine. We have elected to keep this as an outstanding issue for two reasons. Primarily the requirements dictate that we can control the environment in which this software is executed as the store manager only wants a CSV and not the software. Secondly, we can overcome the hurdles within the team as only one of us is working on a windows device and we have a high degree of pairing inherent in our Sprint 1 process.

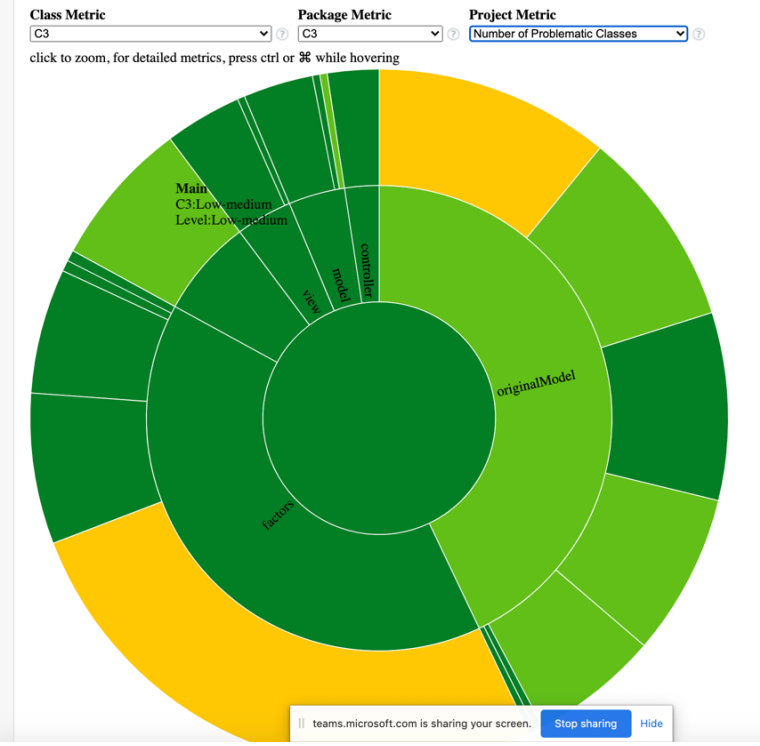
An outstanding question with the manager is regarding the variance in the output file (e.g. Lunh on Mondays described above). We want to know whether that is within the managers expectations or not.

Sample Output

```
1,2020-05-31T06:00:11,52,Senior
2,2020-05-31T06:00:22,60,NonSenior
3,2020-05-31T06:00:33,60,NonSenior
4,2020-05-31T06:00:44,55,Senior
5,2020-05-31T06:00:55,60,NonSenior
6,2020-05-31T06:01:06,60,NonSenior
7,2020-05-31T06:01:17,60,NonSenior
8,2020-05-31T06:01:28,60,NonSenior
9,2020-05-31T06:01:39,50,Senior
```

Metrics

Complexity, Coupling and Lack of Cohesion snapshot



Test Coverage Snapshot