INTRODUCTION

The aim of this project was to develop a program that would simulate the Easton Farmer's Market. The development goals were focused around building an efficient simulation to maximize the number of people going through the market and minimizing the time spent in the market by each customer. Meeting a majority of the needs and requirements of the customers was also focused upon. The analysis goals included analyzing the average time spent by customers in the market and each stall. Also, determining the optimal distribution of different types of stalls and the relative importance of each type of stall with respect to the customers' needs.

Several assumptions were made to simplify the task at hand and allow the programmer to focus on the key requirements and areas of importance. Some of them were as follows-

- Only one customer would arrive at the market at any given point of time.
- There were fixed types of stalls that could exist in the market.
- Each stall would only have a single queue, but multiple stalls of a type could exist.
- Any stall of a certain type would be identical to the others of its type.
- Customers' needs were not quantified in terms of any units and were, instead, reduced to simple yes/no criteria to indicate whether a certain customer needed a certain type of good or not.
- When a customer visited a certain type of stall his needs with regard to that type of good would be completely satisfied.
- As soon as a customer satisfies all his different types of needs he immediately leaves the market.
- There was no time to be taken by customers to travel between stalls. As soon as a customer entered the market he would join the queue for a stall and then, after leaving that stall, would immediately enter another queue for a different type of good corresponding to his needs.
- The entire market and each stall could only accommodate a specific number of people.
- The market would open and close at fixed times.
- The arrival times of customers, time spent in line for each customer and individual needs would be determined based on values from different Gaussian distributions. They are described as follows:

Time interval for arrival of new customers into the market-

Mean	Standard Deviation
61 seconds	31 seconds

Percentage of people who would need types of items-

	Mean	Standard Deviation
Baked	37%	17%
Meat	53%	13%
Dairy	59%	19%
Fruit	47%	13%
Vegetable	71%	29%
Beverage	43%	11%

Processing time for customer when they reach the front of a queue-

	Mean	Standard Deviation
Baked	29 seconds	13 seconds
Meat	83 seconds	41 seconds
Dairy	59 seconds	23 seconds
Fruit	83 seconds	31 seconds
Vegetable	119 seconds	29 seconds
Beverage	19 seconds	7 seconds

It was hypothesized that as the number of stalls for each type would be increased, the market would increase in efficiency, but only up to a certain point. After a point of adding more stalls, the addition of extra stalls would be wasteful as they would occupy market units while not handling a significant percentage of the customers. The time spent by a customer in the market would be expected to reduce as the efficiency of the market increased. The vegetable stalls were expected to have the most significant impact on the market efficiency because (according to constraint values) the most number of customers would need vegetables and the stalls would also take the longest to process each customer.

APPROACH

The developed solution was divided into several classes, each serving a different functionality-

- Person class- This class acts as a parent class to two sub-classes and provides functionality for storing a unique ID for every person.
- Customer class- This class is a child class of Person and provides functionality to create customers with random needs based on the Gaussian percentage constraints. The class can also store and retrieve the time when a customer arrives at the market or at any stall. A customers needs can be changed from this class after he visits a type of stall.
- Worker class – This class is a child class of Person and serves to represent a Worker working at any stall.
- Stall class – This class is the parent class for all the different types of stalls. It stores functionality for most of the operations and processing that occur at a stall. It can be used to add customers to the queue at a stall and subsequently remove them once they have been processed. The time taken to process each customer is based on the Gaussian time constraints. This class also stores the time taken for each customer when they enter the queue to when they are fully processed.
- BakedStall class – This class is a child class of Stall and it borrows most functionality from the parent.
- MeatStall class – This class is a child class of Stall and it borrows most functionality from the parent.
- DairyStall class – This class is a child class of Stall and it borrows most functionality from the parent.
- FruitStall class – This class is a child class of Stall and it borrows most functionality from the parent.
- VegetableStall class – This class is a child class of Stall and it borrows most functionality from the parent.

- BeverageStall class – This class is a child class of Stall and it borrows most functionality from the parent.
- Market class – This class serves to represent any market being simulated. The class can be set to contain any combination of different types of stalls. It provides functionality to add/remove a customer from the market and add/remove customers from various stalls. Customers are passed to the Stalls to actually be added in the queue. The market always tries to place customers in the least crowded stall of a given type. **Thus, if no customer is ever added to a stall that means that the other stalls were equally/more empty whenever customers were being added to a stall of that type. Such a stall would not be needed to improve efficiency given the fixed constraints.** This class also contains functionality for a waiting area in case there is no available space in any stall that corresponds to their types of needs. It also stores the total time spent by each customer in the market.
- MarketSimulation class – This class serves to run the actual simulation of a Market using simulated time. Each iteration through a loop counts as one second of simulation time. During each iteration of the simulation the waiting area is checked to try and place the waiting customers in a stall queue. Each queue in the market is also checked by the Stalls to see if the customer at the front of the queue has been processed. Customers are added to the market based on intervals determined by the Gaussian time constraints. This class additionally provides functionality for a text based interactive interface to run the simulation and change parameters.
- Queue class – This class provides functionality to implement a Java LinkedList as a first-in first-out queue.
- GaussianGenerator class – This class provides functionality to generate random numbers in a Gaussian distribution with a given mean and standard deviation.
- RandomGenerator class – This class provides functionality to generate random numbers within a given range of bounds.

Several levels of hierarchy were implemented to hide internal functionality of most of the classes. Starting from the Stall class up to the MarketSimulation class, abstraction and polymorphism was implemented. Similar development techniques were used for the Stall class with its children and the Person class with its children.

METHODS

The parameters used for the experiments are listed as follows-

1. Number of stalls selling baked goods
2. Number of stalls selling meat
3. Number of stalls selling dairy
4. Number of stalls selling fruits
5. Number of stalls selling vegetables
6. Number of stalls selling beverages

As described in the introduction, the statistical parameters to determine how frequently the market state changes were set as fixed constraints.

The simulations were run by varying the parameters sequentially. A total of five different market configurations were tested. For the initial configuration the parameters were set to their minimum values to obtain a benchmark of how each stall compared to each other. The parameters were then revised on each successive configuration in accordance with how much traffic each additional stall was receiving. If a stall did not receive any traffic then it would serve no purpose in the market. Such a method was utilized to slowly move towards an optimal configuration after testing each subsequent set of parameters.

The five configurations tested are listed as follows-

1. 1 stall of each type
2. 2 stalls of each type
3. 3 stalls of each type
4. 2 stalls for baked goods & beverages; 4 stalls for dairy, meat, fruit & vegetables
5. 2 stalls for baked goods & beverages; 3 stalls for dairy, meat & fruit; 5 stalls for vegetables

Each stall configuration was run 5 times to obtain **average values.** For each configuration an analysis was performed using the average times spent by customers at each stall and the number of customers that would enter/exit each stall. After analyzing the data for each configuration, a comparison of these configurations was also performed.

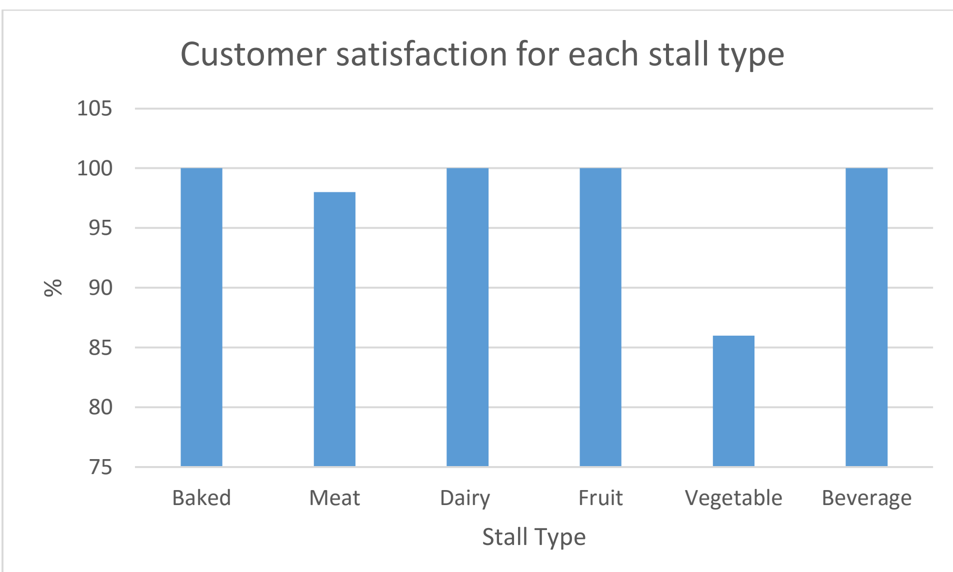
DATA & ANALYSIS

The data obtained has been described and analyzed as follows-

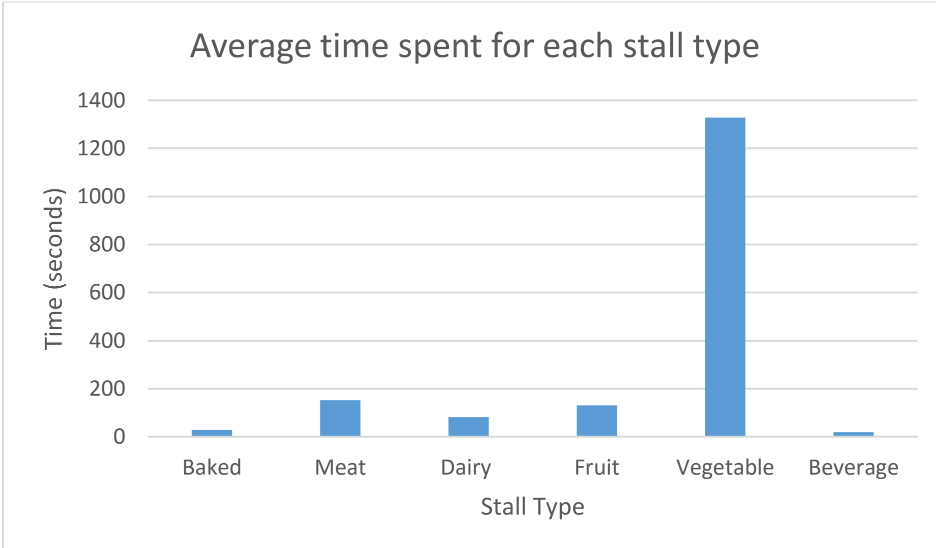
1) **<u>1 stall of each type-</u>**

Percentage of customers fully satisfied = 82%
Average time spent by a customer in the market = 1168 seconds

Percentage of customers satisfied for each stall type-



Average time spent by a customer for each stall type-

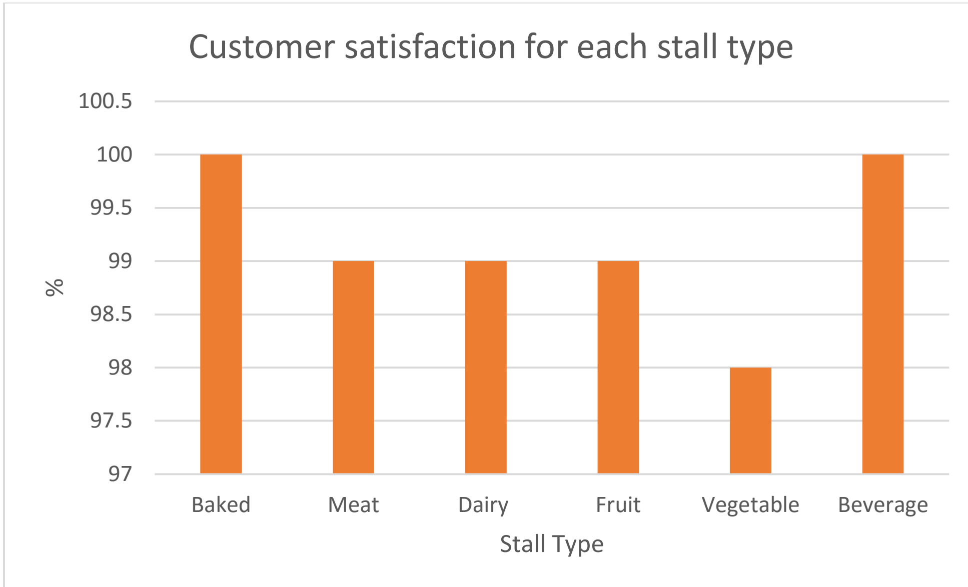


Using 1 stall of each type would could not be ideal based on the relatively large number of customers being served at each stall. This configuration provides a benchmark as we can notice how crowded each stall would be with this configuration. The average times spent in the market and at each stall are also particularly high. As the market would have ample units available, we can easily add one more stall of each type to notice how much of a difference such an addition would make.

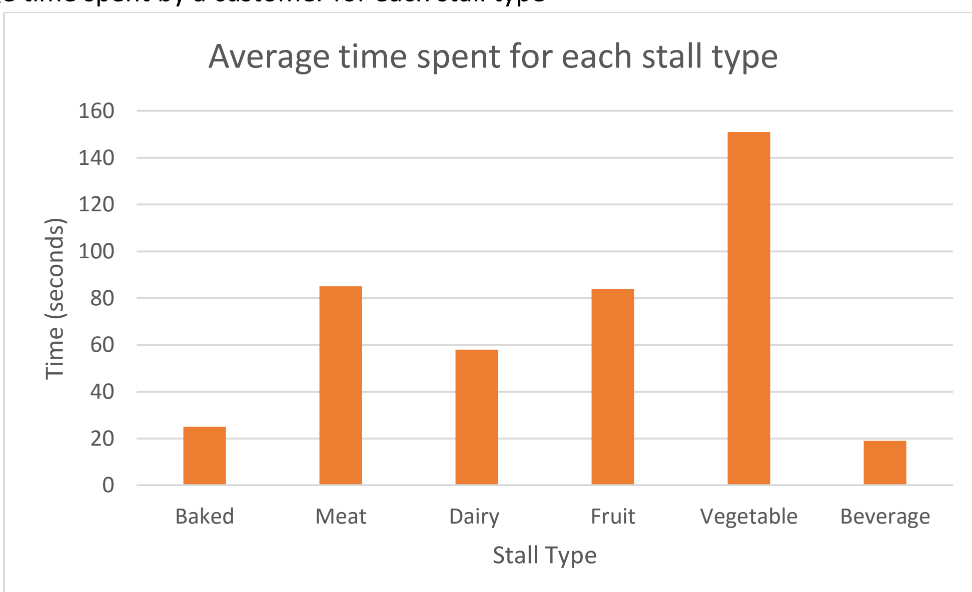
2) <u>2 stalls of each type-</u>

Percentage of customers fully satisfied = 97%
Average time spent by a customer in the market = 249 seconds

Percentage of customers satisfied for each stall type-



Average time spent by a customer for each stall type-

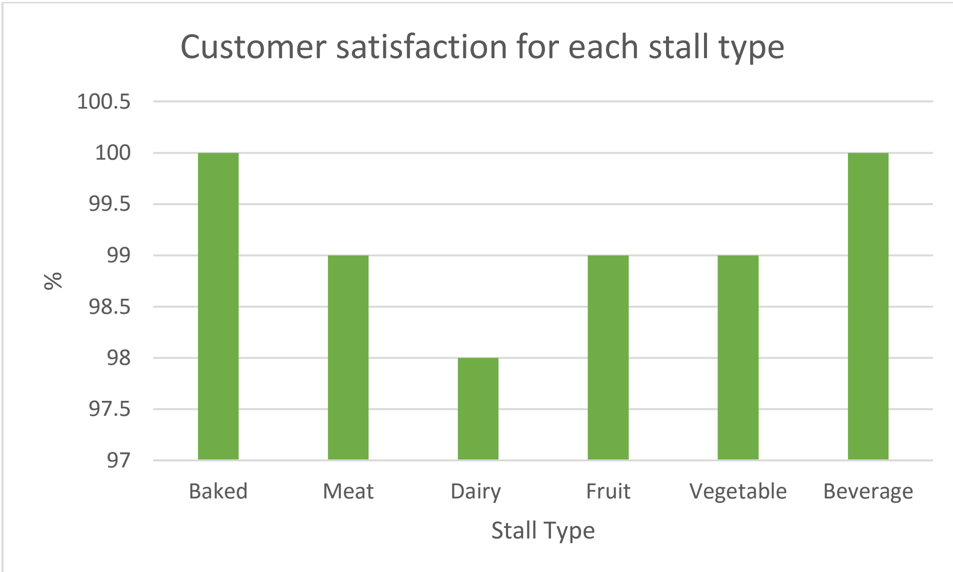


Using 2 stalls of each type presents a much more efficient configuration as compared to 1 of each, considering the total unit cost of implementing an extra set of stalls. The average time spent by the customer was reduced by around 900 seconds and the number of customers leaving the market fully satisfied was increased by around 30, leaving just 5 customers who were not fully satisfied. Certain stalls like Baked and Beverage are not catering to many customers in the second stalls, while, it can clearly be seen that the addition of a second Vegetable stall caters to a lot of traffic and greatly reduces the waiting time at the Vegetable stalls. A second stall of every type does seem to increase overall efficiency. We can add yet another set of stalls to see how it will effect traffic to each type of stall.

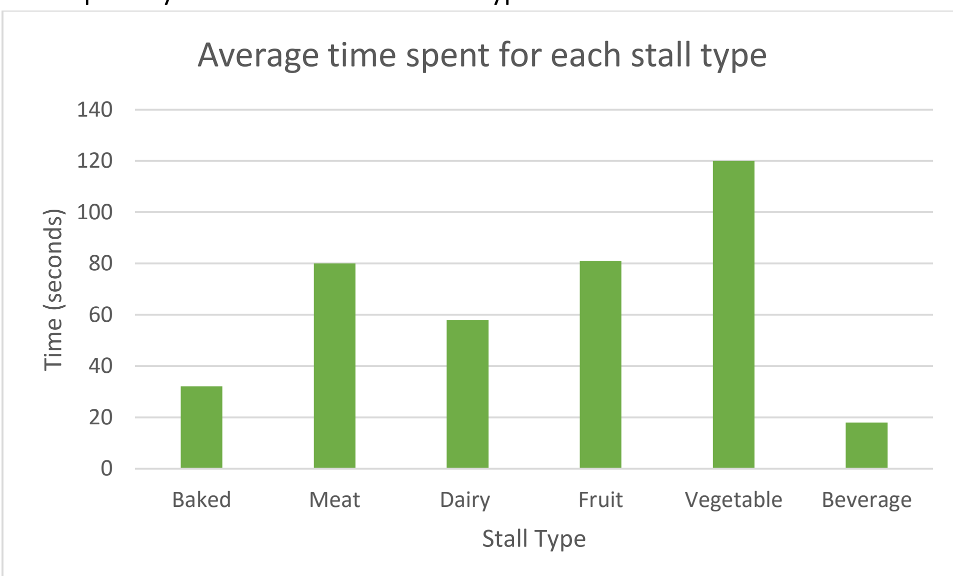
3) <u>3 stalls of each type-</u>

Percentage of customers fully satisfied = 98%
Average time spent by a customer in the market = 214 seconds

Percentage of customers satisfied for each stall type-



Average time spent by a customer for each stall type-

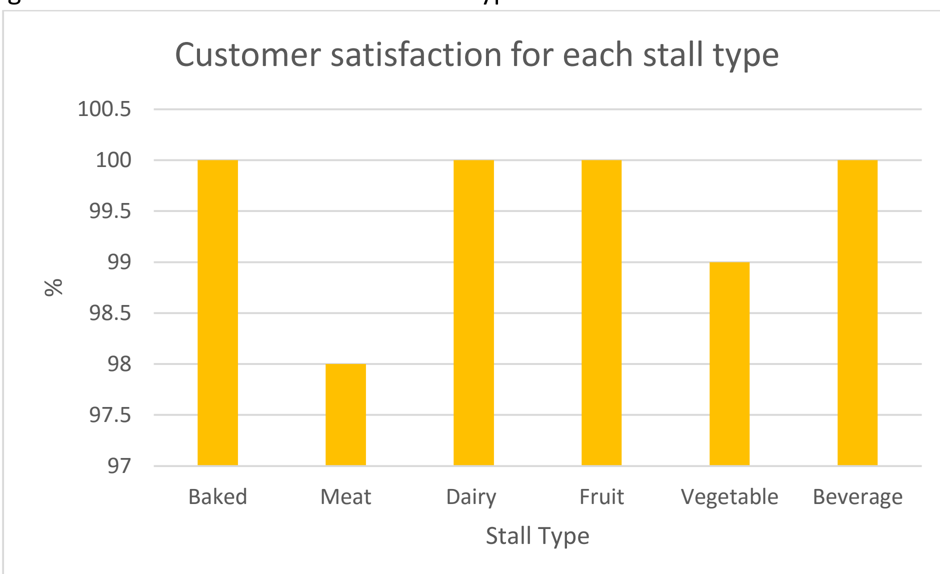


Using 3 stalls of each type does not provide as much of an improvement as increasing from 1 to 2 did, however, certain stalls do benefit from this. While the third vegetable handles the most amount of traffic amongst all the third stalls, meat, dairy and fruit do handle a small amount of customers. On average the third baked stall and beverage stall do not handle any customers, thus, we can conclude that a third instance of these stalls would not be necessary. The customers handled by the third set of stalls do help reduce the average time spent in the market by around 40 seconds and the slight reduction in time taken at stalls is most apparent for the vegetable stalls. Adding a 4th stall for Baked goods or Beverages would not be ideal and we can reduce the number back to 2. Instead we can add an additional set of stalls for the other 4 types to see how much traffic they handle.

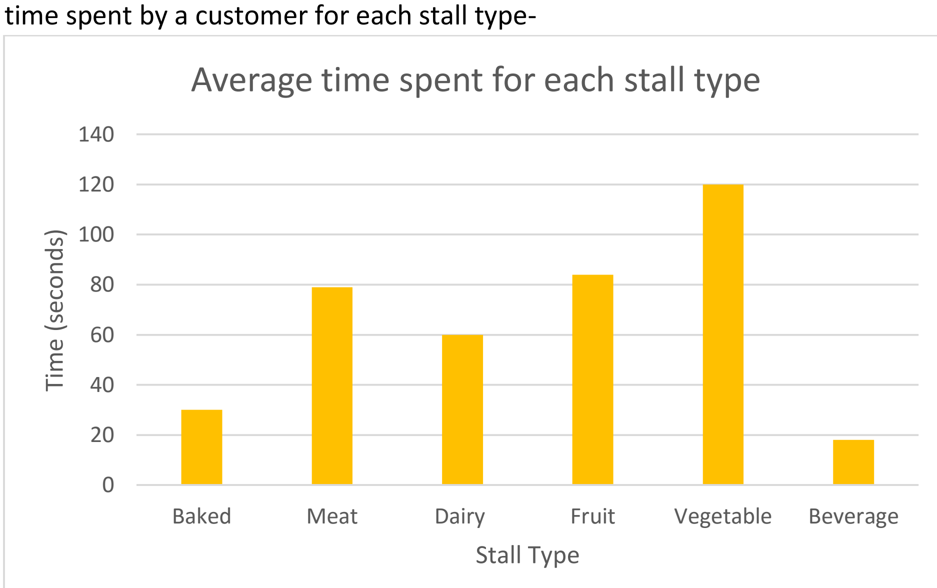
4) <u>2 stalls of Baked, Beverage; 4 stalls of Meat, Dairy, Fruit, Vegetable-</u>

Percentage of customers fully satisfied = 98%
Average time spent by a customer in the market = 220 seconds

Percentage of customers satisfied for each stall type-



Average time spent by a customer for each stall type-

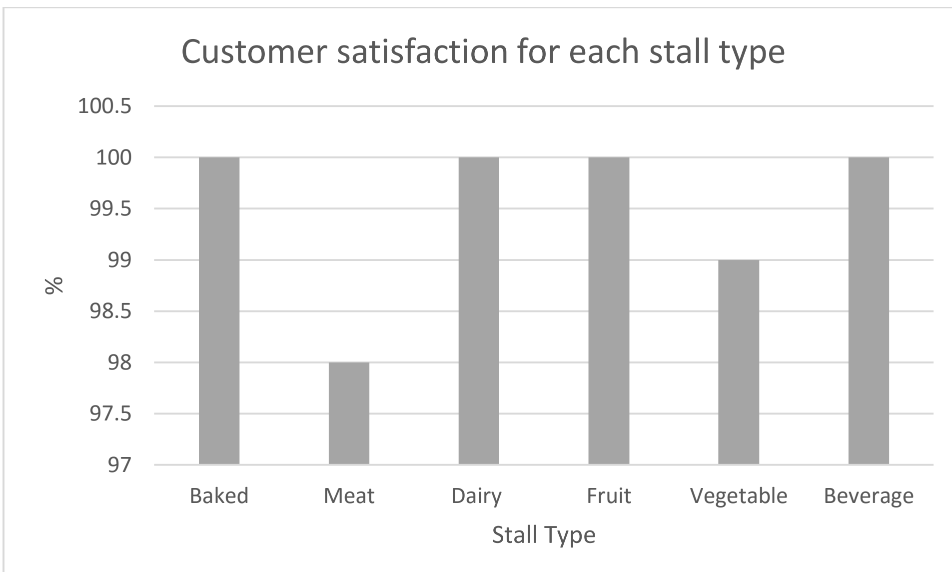


Using 4 stalls for Meat, Dairy, Fruit and Vegetables does not affect the efficiency of the market by a lot. This is because, on average, the 4th Meat, Dairy and Fruit stall do not need to cater to any customers at all. The Vegetable stall on the other hand does cater to a small number of customer, which once again helps reduce the customer load from the other Vegetable stalls. The overall time spent by customers in the market and at the stalls is almost the same. This indicates that we are close to our optimal distribution of stalls. We can still see the effect of adding a 5th vegetable stall, while removing the 4th stall for Dairy, Meat and Fruit.

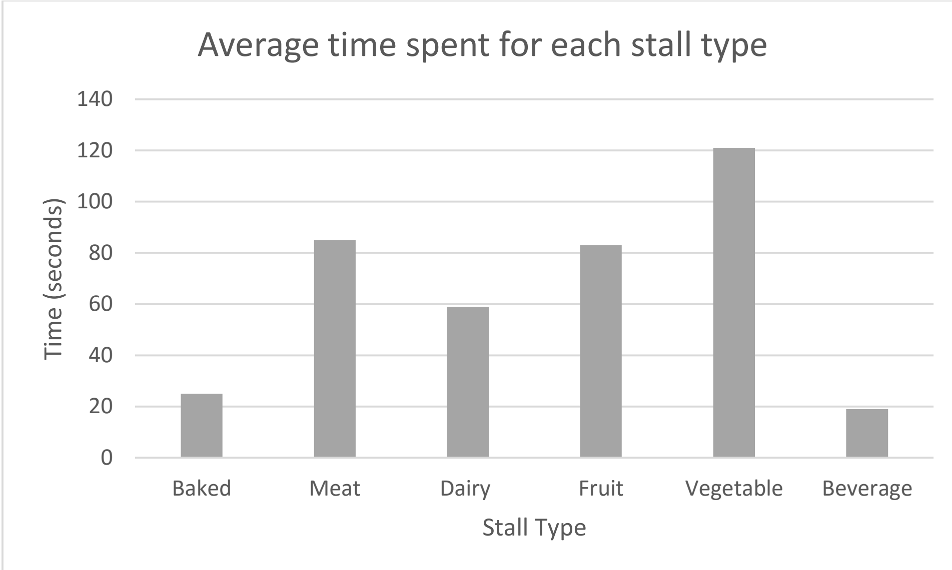
5) <u>2 stalls of Baked, Beverage; 3 stalls of Meat, Dairy, Fruit; 5 stalls of Vegetable-</u>

Percentage of customers fully satisfied = 98%
Average time spent by a customer in the market = 217 seconds

Percentage of customers satisfied for each stall type-



Average time spent by a customer for each stall type-

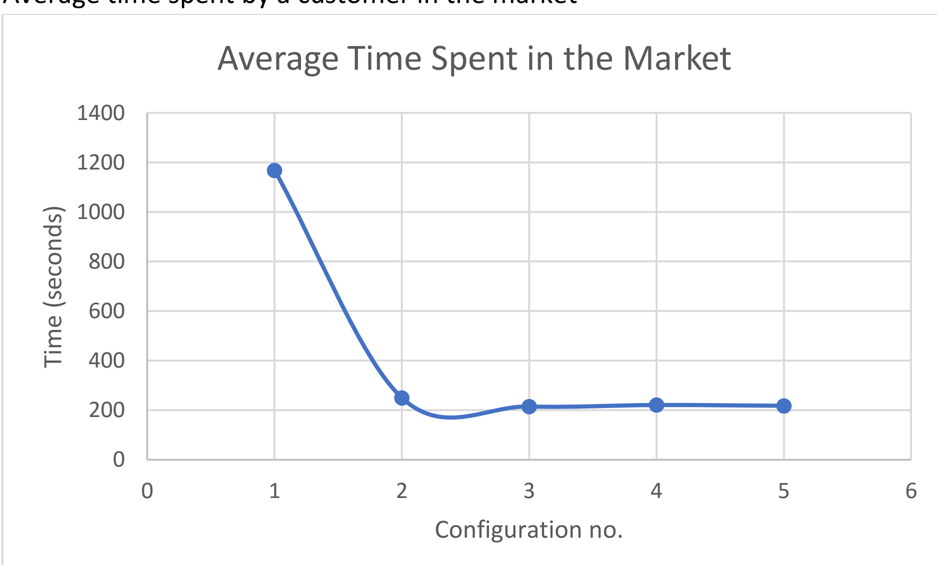


Using 2 stalls for Baked goods, Beverages; 3 stalls for Meat, Dairy, Fruit and 5 stalls for Vegetables maximizes efficiency while not wasting any market units on unnecessary stalls. The addition of a 5^{th} Vegetable stall handles a minute amount of the traffic towards all the vegetable stalls. We could argue that it is not as essential and having 4 stalls for vegetables would be sufficient, but keeping the 5^{th} stall would increase market efficiency by a small amount as it almost always handles some customers. The average time spent by a customer in the market and at each stall is similar to values obtained in the previous configuration, pointing to the fact that we have reached a stable threshold.

<u>Overall statistics-</u>

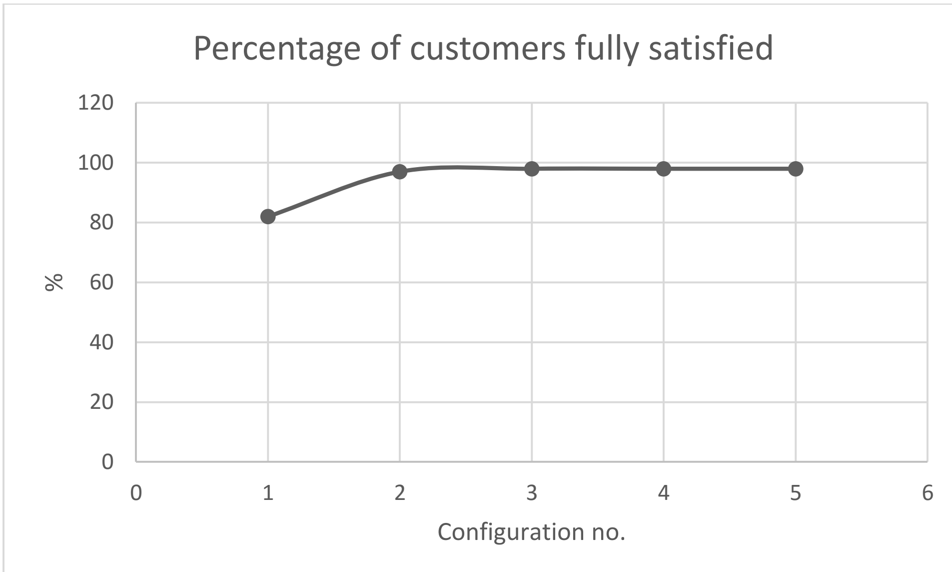
A comparison of the various statistics obtained from each of the configurations has been presented as follows-

- Average time spent by a customer in the market-



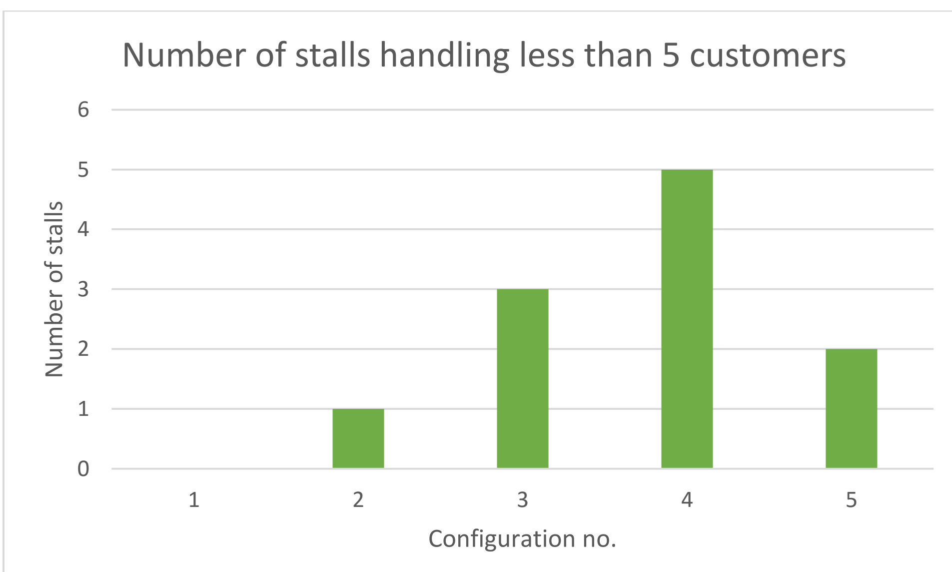
The time taken drop sharply after the first configuration and after the third configuration we see that it remains pretty stable.

- Percentage of customers fully satisfied by the market-



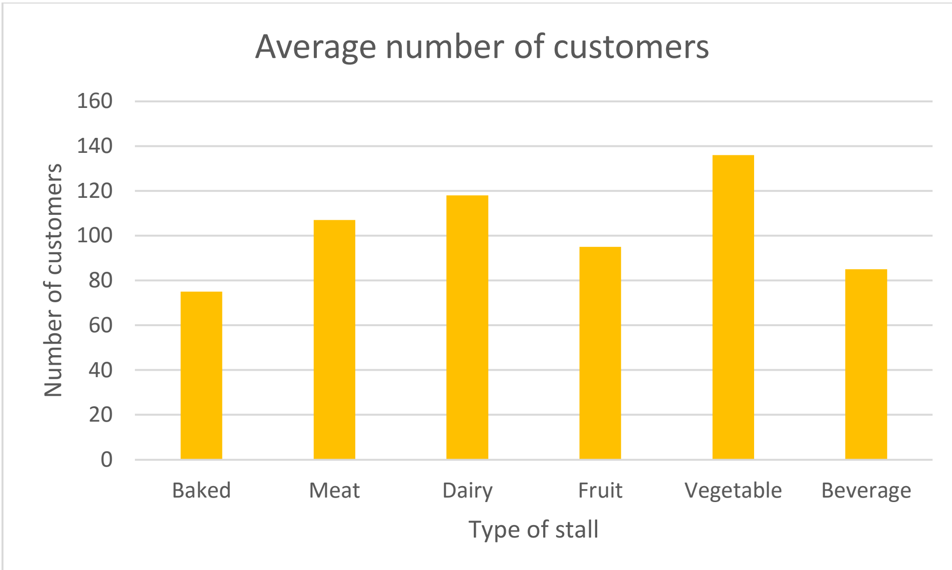
The percentage of customers fully satisfied is always on the higher side but after the second configuration we see that almost all the customers are always fully satisfied.

- Number of stalls handling less than 5 customers-



We see that configuration 4 has the most number of stalls operating at a low inflow of customers. Configuration 5 helps reduce wasted market units of this form.

- Average Number of customers visiting each type of stall-



The Vegetable stall clearly appears to deal with the largest traffic of customers.

CONCLUSION

After analyzing the data obtained from various configurations we were able to reach an optimal configuration on the fifth attempt.

Thus, the **optimum distribution of stalls** is listed below-

Baked Goods: 2 stalls
Meat: 3 stalls
Dairy: 3 stalls
Fruit: 3 stalls
Vegetables: 5 stalls
Beverages: 2 stalls

We also looked at the **average time** being spent by a customer in the market and saw that it stabilized at just a little above 200 seconds as we moved towards the optimal distribution.

In addition we realized that the Vegetable stall was the **most critical type of stall** in the market. On average, the Vegetable stalls were visited the most; close to 140 customers out of around 200 total customers would visit one of the Vegetable Stalls. The average time being spent by the customers on each stall was also the highest for all the vegetable stalls.

In conclusion, we also managed to **confirm the hypothesis** by confirming the general trend of increase in efficiency. We saw that adding stalls would stop increasing efficiency after a certain point, instead leading to wasted market units as some stalls would be operating with a very low inflow of customers.

We also saw average time spent in the market reduce with the increase in efficiency. The Vegetable Stalls also were, in fact, the most important in determining market efficiency as they were recognized to be the most critical type of stall.

REFERENCES

Weiss, Mark Allen. *Data Structures and Problem Solving Using Java*. Boston, MA: Pearson/Addison-Wesley, 2010. Print.

Java Practices - Generate Random Numbers. Hirondelle Systems, n.d. Web. 9 Oct. 2015. <http://www.javapractices.com/topic/TopicAction.do?Id=62>.

TimeUnit (Java Platform SE 7). Oracle, n.d. Web. 12 Oct. 2015. <http://docs.oracle.com/javase/7/docs/api/java/util/concurrent/TimeUnit.html>.

Talwar, Nakul. *Managing a Farmer's Market*. Vers. 1.0. N.p.: n.p., 2015. CS 150 Project.