A blue and yellow logo

Description automatically generated

Troy Indian Fast-Food Restaurant

By:

Sowmya Ramanujam

Shobhitha Ganesh Babu

Padmini Poojari

Ahsan Virk

Ravi Teja Lakkakula

Supervision:

Prof. Edward Williams

**Table of Contents:**

List of Figures ……………………………………………………………………………………………………………………… 3

List of Tables ………………………………………………………………………………………………………………………. 3

Introduction

Description of the enterprise

Description of the system to be simulated.

Description of the input and how it was obtained.

Approach to building the Simio model and description of it.

Description of the Simio output

Recommendations for improving the system.

Conclusion and recommendations for future work

References

**List of Figures:**

1

2

3

4

5

**List of Tables:**

1

2

3

**Introduction:**

In this case study, a simulation model of a fast-food restaurant will be created to better understand the system's behavior under certain conditions. The biggest issue with the fast-food business is the customers' excessive wait periods. Customers must wait in line to place their orders and then wait again for their orders to be prepared. According to the restaurant's managers, the average wait time for a customer is more than ten minutes. The restaurant's simulation model will be constructed to examine how and how much waiting times could be lowered in various scenarios.

**Description of the enterprise:**

It is a popular Indian fast-food restaurant located in Troy.

A diagram of a food system

Description automatically generated with medium confidence

The entities of the customer and the orders that they place are marked in dark and light green colors. Whereas for online orders we have given customers a dark purple color and for online orders the entity is shown as light purple color.

**Description of the system to be simulated:**

From this system we want to build a model which shows how the system behaves in the weekend differently from weekday with the same number of staff and what the utilization and wait time looks like which is going to be our scenario 1. In scenario 2 of the experiment the things that are going to change between weekday and weekend are online food order percentage, number of open hours, food order mix, order size and arrival rate of the customers. In scenario 3 we change the number of people at the cash register by adding an extra person and adding an extra person at either of the food stations for the weekend or adding another half hour to the shop and show how it eases out weekend congestions. The objective of the project is to model the difference in operations between weekday and weekend and by looking at staff allocation to improve efficiency and reduce customer waiting.

**Description of the input and how it was obtained:**

Based on observation for a day at the restaurant and by consulting the manager, workers, owner, and few customers we have come up with data that simulates real-world environment. The data for the cashier and cooking process is captured using a stopwatch, and the data is confirmed with the respective manager. By statistically evaluating the obtained data and discussing the results with restaurant management, the following distributions were discovered to represent each process best:

Variables of the system: Queue length, Service time, Average waiting time

Input parameters: one person is assigned at Chinese food station, 2 persons who are interchangeable at chat and sandwich station and 2 persons who are interchangeable at roti and dosa station.

Performance indicators:

Constraints of the system: Seating capacity in the dining area is 50.

Working hours: The restaurant opens at 11 am and closes at 9:30 pm at night during the weekdays and it is opened till 10:00 pm during weekends and the system is open till last customer is served but to finish the cleaning and all other parts we have kept the store opened till 11 pm at night.

**Approach to building the Simio model and description of it:**

Arrival mode is time varying arrival rate because the restaurant does not have the steady flow of customers for which we have defined the rate table for the weekdays. Since the weekends are much busier than weekends what we have done is, instead of defining another rate table, we have doubled the arrival rate of customers who come on weekdays, which is clearly visible from the number of sales that the store makes. For that we have defined a property of arrival rate scale factor. We have considered entities per arrival as 1. The way we have accounted for group arrivals is the size of the order.

A close-up of a computer code

Description automatically generated

Figure 1. State assignments before exiting entrance.

We have defined 25 percentage of orders come through online. When it is an online order the color of the entity changes to purple.

ModelEntity.Customerid, ModelEntity.ID

This is an important part because this is how we match the customer id with the orders generated by them so that when the delivery happens, we match whether the order belongs to the customer and then give it to them. Every model entity that is generated by Simio has an intrinsic property called modelentity.id hence we didn’t create anything new we just attributed that to customer id.

Once the customer passes through the entrance then he proceeds to the order counter where he has got choice of ordering any of the three categories in three order sizes.

A diagram of a diagram

Description automatically generated

Figure 2. Flow chart showing different components that make up the customer choice of ordering food.

As part of our data collection, we have identified that the percentage of orders that go to the chat sandwich station are 60 % in any given day followed by roti dosa station by 70 % and then the Chinese food station by 30 %. The summation of order perspective does not add up to 100 and that is because the orders contain items from multiple stations. It tells me within each food station how does my order size per transaction varies. The second thing that we have defined is the number of food categories that they order from. There are cases where a few customers order only from one station, some order from two and the rest can be from three. The values that we came up with are 50 percent of the orders in an average day go only to one station, 40 percentage goes to either of the two stations and the remaining 10 percentage of the population order from three stations.

A diagram of food and a diagram of food

Description automatically generated with medium confidence

Figure 3. Venn diagram depicting proportions of food items that customers order.

Order size influences the process time at each station. Based on talking to the people we have come up with small, medium, and party size orders proportions for each food station.

A table with numbers and text

Description automatically generated

Table 1. Table displaying order size of different food stations.

We have defined processing time based on the order size. If they are big orders, then it takes random triangular distribution of around 3 to 6 minutes. If the order size is small, then it’s going to take 2 to 3 minutes. Here ordering means we refer to ordering the item, making the payment, and leaving the counter.

We have created a balking decision at the order counter which defines that if the total number of customers at the order counter are more than 7 then they are 15 % chances that the new customers will leave and if it is more than 10 people then 30 % of the people will leave. We have also defined reneging where the customers who are waiting in the line to order food crosses 5 minutes and if they realize that the dining area is full then 20 % of the customers will leave. We have implemented that the system, either the POS or credit card machine fails every 2 weeks for which we must restart the system and it’s going to take 2 – 3 minutes to be up and running.

A connector which is a zero-time connection between two nodes is used for order counter for which we have made a separation logic as make copies for separation mode, copy quantity to be 3 and copy entity type is orders. Copies are nothing but orders of customers. These orders have the same metadata as customers. For every customer we have created three copies of what they order. For example, if the customer orders roti and chat then it’s going to be true for roti and chat, false for the Chinese food. We have created a process named AssigningPath\_DineinOrderCopies.

A screenshot of a computer

Description automatically generated

Figure 4. Process flow of dine in order copies.

The main logic here is that at the first station I check the first component in the copy and if it is true then only that food station processes it if it is false then I will destroy that copy in the same way for the other two. For example, if my first component in the copy and is Chinese and the first food counter is Chinese then if a customer places an order of only Chinese food item, then it’s going to create three copies where in the first copy executes at first counter and the rest copies will get destroyed because their values are false. This entire logic is present in the check node. Same thing for OnlineOrderSystem as well.

We have added a process at online orders to restrict the flow of online orders that goes to food counters to avoid the customers waiting at the queue who comes for dine in. because we have modeled in our system that once the online order is placed then the customer going to come only after 30 minutes of placing an order. For that reason, we check if the dine in queue is more than 5 customers then we delay the online order execution by 4 minutes and we assign a ticket of 1 to it. Once it waits for 4 minutes it goes through decision logic again to see if the queue is still more than 5 customers and if yes then it’s going to wait for another 4 minutes, and the ticket number increases to 2. Once the order waits for 8 minutes, we no longer keep it waiting so it goes further for the execution.

A diagram of a process

Description automatically generated

Figure 5. Delaying online orders if the dine in queue is more than5 customers.

At the order management there is going to be a person who calls out the order for which the processing time is 2 to 5 seconds, and the capacity is 1 at a time. Once the entities pass the check nodes then they change the color to light version to differentiate. We have created three paths with selection weights described from order management one each goes to different food counters. At each station we have modeled the processing time as Task sequence which means it is going to execute in a sequential way. Initially the food is prepared, which always happens and depending on the order size we have a different time.

A table with numbers and text

Description automatically generated

Table 2. Processing time for different size orders and packaging at three food stations.

For the different order sizes, we have considered it as random triangular distribution with unit time in minutes. Plating we refer to just place the food on a plate and serve it to the dine in customer. For plating and packing we took random uniform distribution with unit time in minutes.

In our model we have specified two station elements in our model for customers and online food to wait at a particular station unless being called upon. We have defined a process as to when the customer finishes ordering the food then he will be transferred to the customer waiting zone and once the food gets ready, he goes to the delivery and then to dine in area to have food.

A diagram of a transfer

Description automatically generated

Figure 6. Process to transfer customer to waiting area once he finishes placing an order.

We have defined a notional online wait time. When an online order appears on the system, the entity is created and waits for a random triangular distribution of 25, 33, and 40 minutes before collecting the food from online delivery.

For delivery, we used a combiner, which combines two things and produces one result. We are mixing orders and customers using a matching rule that includes both members and parents. The copies arrive at the member node, and the parent node is where the customer enters. When the completed food order arrives at the member node, we initiate a process that calls the customer from the waiting station.

A diagram of a customer delivery

Description automatically generated

Figure 7. Process to call dine in customer waiting at station to handover the food.

After collecting the food from the delivery node, the consumer proceeds to the dining area, where he spends roughly a random triangular distribution of 15, 25, 40 minutes. The dining room has a capacity of 50 people. When the diner has finished his meal, he exits through the exit. However, in the case of online delivery, whenever the customer arrives at the parent node, the process initiates, which seeks food from the online delivery food waiting and sends it over to the client after verifying that the customer id matches the customer id on order copy.

**Description of the Simio output**

**Recommendations for improving the system.**

**Conclusion and recommendations for future work**

The future work includes managing the parking lot and washroom at the restaurant. Managing the parking lot is important because if it is full then we lose many customers.

**References**

**Simio and Simulation - Modeling, Analysis, Applications - 6th Edition**

**Jeffrey S. Smith and David T. Sturrock**