Takealot Inventory Problem

Jocelyn Atwell

School of Natural and Applied Sciences

Sol Plaatje University

Kimberley, South Africa

201800014@spu.ac.za

Abstract Every aspect of life has a possible problem we could address. They differ from sector to sector. The most interesting problems to solve would be to make use of inventory control systems. There are many benefits in using these systems as well as drawbacks. In this project, we will dive into the use of inventory control systems as it is implemented to an ecommerce retailer problem.

Keywords: inventory control systems, algorithms, events, simulations.

Introduction

According to Inc.com, an inventory system can be described as a system that encompasses all the aspects of a company's inventory management which includes purchases, receiving, shipping, warehousing, tracking, turnover, or profits, reordering as well as storage.

The importance of an inventory system is to ensure that all the associated activities of a company's inventory management are performed within the correct sequence to have an effective and efficient inventory control system. Computerised inventory simulations such as the MATLAB simulation allows the company to integrate various functioning subsystems that operate under one single system.(Inc.com, 2020)

There are three types of inventory systems according to Upkeep, namely: the periodic inventory system, the perpetual inventory system as well as the manual inventory. The manual inventory relies heavily on manual record keeping methods and are mainly used for small business that require no special tools, but this is very time consuming. The perpetual inventory system focuses on real time data and the processing of it. Many special tools are required for use such as RFID tags or scanners which are automatically linked to a spreadsheet system or database for data collection and storage. It is very fast and effective and leaves little to no room for human error, however the equipment required for this inventory system are expensive to purchase and maintain. Finally, the periodic inventory system makes use of the physical counting of data on a periodic basis which could be weekly, monthly, quarterly, or yearly. This inventory system also makes use of scanners and RFID tags to scan, collect and store data fast and effective, however it is time consuming as it is not done in real time and on a daily basis as the bulk of the data collection and storage would have to be done for a longer time per period session.(UpKeep, 2021)

According to Barcode Direct, the first readable punch card was invented in the 1880s and was used to record data. However, it was only that in the 1980s where the first barcoding system was introduced. It was later improved in the 1960s where the first barcode system was used to track inventory using scanners. In the 1980s computer system software was introduced to create an inventory tracking system operating hand in hand with the barcode system. It was not until the early 2000s until RFID tags and barcode technology was properly introduced in which it worked fast and efficiently unlike the years before. This was when data could be stored and analysed for feature extraction and analysis of results to improve on their businesses.(Direct, 2015)

Takealot is an ecommerce retailer that sell items from various departments and deliver their products directly to their customers. They would like to know, when a customer adds items to their online shopping cart, considering the VAT, shipping costs, etc; what would the grand total of an online order placed be? Secondly, how long would it take for the package to be delivered? Would they make a profit or a loss per order?

Data would be collected from the customer that would be able to answer all three research questions. These results would be separated into three parts, namely: the order details which include the grand total results, the estimated time for each delivery per order placed as well as the profits made per customer order. These results would be simulated and represented in graphical format with visual representations.

Takealot Business

What is Takealot?

Takealot is an ecommerce retailer in South Africa. When it was created initially, it run under the business name Take2 in the year 2011 and was initially based in the United States of America. Takealot is one of the largest ecommerce retailers on the African continent. This business has over 21 departments to select from when searching for an item to purchase online. These departments include Lifestyle, Media and Gaming, Fashion as well as Electronics. Takealot has their own logistics company as well which eliminates the middleman and allows them to deliver their products safely and directly to the customer. Through being the largest ecommerce retailer in South Africa, Takealot prides itself in having more than 2000 employees.(Takealot.com, 2021a)

There are three strategically placed warehouses in South Africa, namely in Cape Town, Durban, and Johannesburg. For inventory purposes only I have excluded the Durban and Johannesburg warehouses from this experiment and included a Pretoria warehouse as well as a Bloemfontein warehouse. The reason for this is to have a warehouse located on both lateral sides of the country as well as a central based warehouse. Having warehouses that are not too far from the customers delivery address ensures that a customer does not wait too long for their parcel to be delivered if they would be ordering items from the nearest warehouse.

Figure 1:One of the Takealot warehouses

All the online purchases from this business could be done through the Takealot app which could be purchased on the App-Store or it could be done through their official website which is https://www.takealot.com. (Takealot.com, 2021b)

Other essential information to know about Takealot will be mentioned below. Takealot has sales for many items available to purchase online, especially through their application. There are many Gift Vouchers that could be purchased through their company as well. The many benefits of being a regular customer would be to obtain discount coupons of a specific value which could overall reduce the purchase value you would pay as a customer. Takealot also permits their customers to make exchanges and return damaged parcels. Once an order is placed by a customer, it could be tracked through a user interactive system that keeps the customer updated on the whereabouts of their parcel as soon as it is purchased until it is expected to arrive on the specified day. Takealot also allows you to sell items through their company. They also host competitions in which customers could win gift vouchers and more. One promotion that was introduced quite recently is that with every first order placed by a customer would include free shipping as well as a free mask promoting safety against the Coronavirus.

Since March 2020, when the Coronavirus struck South Africa, a new system was introduced to prevent customers from physically signing documents verifying that their parcel was received. Instead, they would send the customer a link via instant messaging in which the customer would open the webpage and click on parcel received. This method constitutes with the social distance coronavirus protocol and ensures that their staff and customers remain as safe as possible.

Design and Methodology

System Parts

A. Python

The use of the python programming language in this project was simply just for data acquisitional purposes. A data frame was created that consisted of all the customers credentials which include the following:

	Column Name	***Description***	***Data Type***
1	Nr	Indicate order to the data	Integer
2	Customer Name	Name of the customer	String
3	Gender	The gender of the customer	String
4	Customer Location	The city in which the customer resides	String
5	Discount Percentage	The discount of the coupon provided	Float
6	Item Names	Names of the products added to shopping cart	String
7	Item Prices	Cumulative prices of the items added to shopping cart	Integer
8	Purchase Prices	Factory purchased prices by company per cumulative order	Float
9	Profits	Profits company makes per customer order	Float
10	Warehouse	The nearest warehouse allocated to customer	String
11	Shipping	Shipping prices per condition	Integer
12	Shipping Var	Shipping conditions which determine shipping prices	String
13	Grand Total	Final price customers pay per order online	Float
14	GPS Estimate Final	Shortest distance between customer location and warehouse (in minutes)	Integer
15	GPS Estimate BFN	Estimated distance	Integer

		between customer location and Bloemfontein warehouse	
16	GPS Estimate CPT	Estimated distance between customer location and Cape Town warehouse	Integer
17	GPS Estimate PTA	Estimate between customer location and Pretoria warehouse	Integer
18	Collection	Indicating if a customer will collect their order at the warehouse specified	Integer (Boolean Type)
19	Payment Reference	Automatic random number generated for payment reference	Integer

After generating the basis of the data frame, various functions were used to determine the elements within certain columns that are dependent on the results of another. The following functions were generated:

1. **Table["GPS_Estimate_Final"] = Table[["GPS_Estimate_BFN","GPS_Estimate_CPT","GPS_Estimate_PTA"]].min(axis=1)**

The above function was used to determine the nearest warehouse to each customer. Choosing a warehouse that is closer to the customers location is beneficial to both the company as well as the customer. The customer would not have to pay a large sum of money for delivery fees and the company would maintain their trust between their customers through delivering their parcels fast and effectively. To get the GPS estimates between the customers location and each warehouse, I manually searched for the time it would take between each source and destination. This was very tedious and could have possibly been done using a simpler method.

2. **Table.loc[Table["Item_Prices"] >= 500, "Shipping"] = 0**
   Table.loc[Table["Item_Collection"] == 1, "Shipping"] = 0
   Table.loc[Table["Shipping_Var"] == 1, "Shipping"] = 50
   Table.loc[Table["Shipping_Var"] == 3, "Shipping"] = 100
   Table.loc[Table["Shipping_Var"] == 4, "Shipping"] = 250

The above code calculates the shipping costs each customer would pay depending on specific conditions. The first condition states that if the cumulative price of the order placed by a customer is more than R500.00, their shipping costs would be free of charge. The second condition is that if the customer is collecting their order they would not have to pay shipping costs as well. The third, fourth and fifth conditions operate under the following conditions: "sc","sp" and "dp" where "sc" is assigned the value 1, "sp" is assigned the value 3 and "dp" is assigned the value 4. Referring to the third condition, if the shipping variable is equal to "sc" which is 1, it means that the customer resides in the same city as the warehouse allocated. In condition four, if the shipping variable is "sp" or 3, the customer resides in the same province as the warehouse allocated. The final condition refers to if the shipping variable is equal to "dp" or 4, the customer resides in a different province that where the allocated warehouse is. If the customer lives in the same city where the warehouse is located, they would be charged R50.00 for delivery cost. If the customer lives in the same provide where the warehouse is located, the customer would be charged R100.00 for delivery cost and if they resided in a different province, they would pay R250.00.

3. **Table[“Purchase_Prices”] = Table[“Item_Prices”] * 0.35**

Function 3 above was used to calculate the prices Takealot paid to purchase the sold items from a factory at 35% of the selling price.

4. **Table[“Profits”] = Table[“Item_Prices”] – Table[“Purchase_Prices”]**

Function 4 was used to calculate the estimated profits Takealot should expect per cumulative customer order.

5. **VAT = (Table[“Item_Prices”] *0.15) + Table[“Item_Prices”]**
   Table[“Discount_Percentage”] = Table[“Discount_Percentage”] / 100

Function 5 was used to add the VAT tax to the selling price of the products ordered followed by converting the discount coupon values to a percentage.

6. **Table[“Grand_Total”] = (VAT + Table[“Shipping”]) * (1- Table[“Discount_Percentage”])**

Function 6 was used to calculate the grand total a customer would pay for the order placed online which consists of the cumulative selling price of the items in the cart, the vat calculated in function 5, the shipping costs as well as the discount on the order if a coupon was provided.

7. **import random**
   Table[“Payment_Reference”] = random.sample(range(1000),20)

Function 7 was used to randomly generate a payment reference the customer would use to make an online payment. These reference numbers are unique as no duplicates exist within the data frame. Random numbers were generated between 1 and 1000 for the 20 rows within the data frame.

8. **Table.loc[Table[“GPS_Estimate_Final”] == Table[“GPS_Estimate_BFN”],”Warehouse”] = “BFN”**
   Table.loc[Table[“GPS_Estimate_Final”] == Table[“GPS_Estimate_CPT”],”Warehouse”] = “CPT”
   Table.loc[Table[“GPS_Estimate_Final”] == Table[“GPS_Estimate_PTA”],”Warehouse”] = “PTA”

Function 8 was used to determine which warehouse was allocated to the customer by comparing the value within the final estimated column and the other columns as well and inserting the abbreviation of the warehouse under the column warehouse.

Furthermore, three separate tables were constructed using the columns found in this large data frame. The tables created were specific to the simulation events in mind for my MATLAB simulations. The table names were CustomerEventsTable, DeliveryEventsTable and ProfitsEventsTable. The CustomerEventsTable contains the results the customer would see before making a payment for their items within the shopping cart. The DeliveryEventsTable contains the results the logistics department would have to determine how long it would take to deliver the parcel to the customer and where it should be delivered. The ProfitsEventsTable contains the results the finance department would analyse to determine how much profit they would be making per customer purchase.

The CustomerEventsTable consists of the columns: Nr, Customer_Name, Discount_Percentage, Item_Names, Item_Prices, Shipping, “Grand_Total” and the Payment_Reference column as well. The DeliveryEventsTable consists of the columns: Nr, Customer_Name, Customer_Location, Warehouse, GPS_Estimate_Final, Collection as well as the Payment_Reference column. Finally, the ProfitsEventsTable consists of the columns: Nr, Item_Prices, Purchase_Prices as well as the Profits column.

Assumptions

There are many assumptions that were made throughout the process of constructing a solution to the problem. Here are a few of the assumptions that were made:

There are three main warehouses only, in Cape Town, Pretoria and Bloemfontein. The Bloemfontein and Pretoria warehouses were replaced by the Durban and Johannesburg warehouses for hypothetical purposes. By spacing out the warehouses within the geographical locations of the country, each city in every province would have a nearest warehouse to them with the estimated time it would take to travel between the two points less than 12 hours or 720 minutes. This allows the logistics department to deliver their parcels within the same day the parcels were packaged and cleared for delivery, or it would be delivered the following day.

The second assumption is that shipping costs differ depending on whether the customer lives in the same city, the same province, or a different province and whether they would be collecting their parcel on site or not. Another factor that contributes to the shipping cost is that if any order place has a cumulative total price greater than or equal to R500.00, shipping costs would be free of charge.

Discount coupons reduces the grand total or the final payment price of the customer payment by a certain percentage, however if no coupon is presented, the default value would be 1. By having a default value of 1, when it is multiplied by the rest of function 6, it would not have any significant effect on the final grand total payment amount.

The fourth assumption is that all three warehouses have the items added to the cart in stock. If the warehouse does not have that item in stock, the customer would be notified that the items are out of stock and would be referred to a related product to consider purchasing. If the allocated Bloemfontein warehouse does not have a particular item in stock, they are unable to refer the customer to another warehouse.

The Google GPS map was used to determine the time taken to travel by vehicle from a particular source point which is the allocated warehouse to the destination which is the customers location. A manual search was done to obtain these results.

Once an order is placed and paid for online, it takes 2 to 3 working days to package, seal and clear and order to the delivered from the warehouse to the customers location. There would be depot point at which the parcels would be delivered before it reached the customer, especially if the customer lives in a different province.

For this experiment only, all the orders were placed on the same day by 20 different customers from 20 different locations.

The weight of the parcels was not considered in this experiment and do not contribute any meaningful values to the results as well.

Businesses operate for approximately 9 hours from 08:00 am to 17:00pm. Items packaged and cleared for delivery that takes an estimated 12 or more hours to arrive at the customers destination should be moved to two days thereafter. This assumption only applies if there is only one or two of the three warehouses in operation, but especially if only one of the warehouses are in operation only.

Limitations

Using the robots.txt extension at the end of the ecommerce retailer website, I wanted to clarify if I had permission to scrape the website to get the exact prices for their products within each department as well as their sales and much more. Unfortunately, I was not granted access to do so. For this experiment, all the prices for the items bought online were not the actual prices available on the website, but rather a rough estimate of it.(Takealot.com, no date)

I was also unable to use the robots.txt extension to clarify if I could scrape the google maps GPS website to extract the estimates between each warehouse and the customers location. The estimates mentioned within the dataset were manually searched for and noted.(Google, 2021)

B. MATLAB

Simulink in MATLAB was used to created simulations of how the model would work with the data and display meaningful and significant results that the company could use and interpret to answer the research questions mentioned under the Introduction.

According to the MathWorks Help Center, Simulink can be described as a block diagram environment that can be used for multidomain simulation as well as model-based design.

Simulink is used for system-level design, simulation, automatic code generation as well as continuous testing and verifying of embedded systems. Simulink also allows the user to create various projects, interact with the source control as well as managing shared model components. During simulations, the user can design models for dynamic systems, run their models and review the results thereof as well as validating the behaviour of the system.

The code “Simulink.sdi.view” was used to import all the columns from the python tables as vectors to be able to implement them within the simulation.

MATLAB Simulink was used to apply the data generated through my python datasets to create a simulation of how the system would work if implemented. It would provide real time results from the data which would be manipulated through the simulation as well.(Center, 2021)

Algorithmic diagrams were drawn to represent how the simulations would be conducted the three events created.

The algorithmic diagram below represents how data could be collected within the Customer Event Simulation:

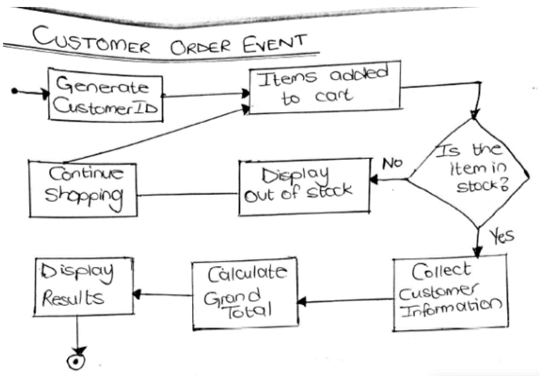

Figure 2: Customer Order Event

As you can see, the algorithm illustrates that the initial step of the simulation would be to generate an automatic customer id which is unique to each customer. Thereafter the customer would select a particular item they are interested in. The system would verify if that item were in stock or not. If the item is not in stock, a message would be displayed that the item is out of stock and the customer could continue shopping, however if the item is in stock, it would be added to the cart. The next step would be to collect the customers information. Using the information collected, a function would be used to calculate the grand total of the online payment to be made. Once this is completed, the results would be displayed for the customer to review and make their final payment.

The algorithmic diagram below represents how the simulation would be conducted under the Customer Delivery Event:

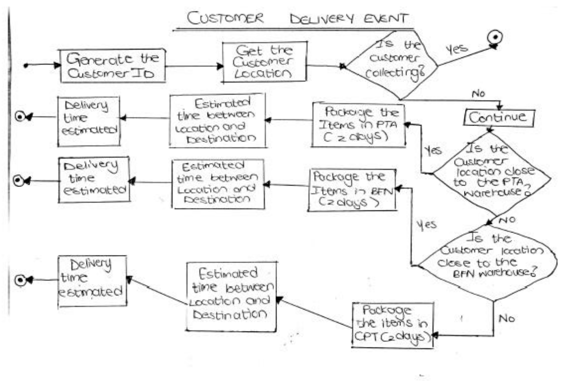

Figure 3: Customer Delivery Event

This algorithm starts off the simulation by generating a unique customer id. Thereafter the customers location is recorded. The system would then verify if a customer were collecting their parcel at the warehouse or not. If they are the algorithm terminates immediately, however if they are not the system would verify if the customer lived close to the Pretoria warehouse or not. If the customer does live close to that warehouse, the parcel would be packaged, and the estimated time recorded to get to the customers location would be used to assist in delivering the parcel.

If, however, the customer does not live close to the Pretoria warehouse, the system would verify if the customer lived close to the Bloemfontein warehouse. If they do, the same procedure would be followed until the parcel is delivered to the customer. Finally, if the customer does not live close to the Pretoria or Bloemfontein warehouse, the Cape Town warehouse would be allocated to the customer and the same procedure would follow thereafter.

The algorithmic diagram below indicates how the simulation will be conducted under the Takealot Profits Event:

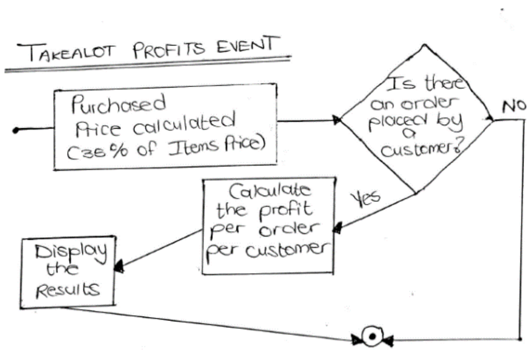

Figure 4: Profits Event

Initially the system would calculate the price Takealot paid to purchase the products from a factory. The items were purchased for 35% of the selling price. Thereafter the system would verify if an order was placed by a customer. If an order was not placed by a customer, the system would terminate, but if an order was placed, the system would calculate the profits made per customer order and display the results.

Analysis of Results

A. Python Datasets and Excel Charts

To collect my data, I used Python. Since I was unable to use realistic data, I had to manually create my own. To do this I used Pandas and NumPy.

Pandas was used to create my data frame or table in which I was able to add all my necessary columns and row elements. NumPy was used to implement mathematical functions to determine the results for some of the columns which were dependent on the results of another column such as the shipping costs being dependent on the whether a customer lives in the same city, same province, or different province to mention a few.

The data collected within the table was used to create three separate tables, each tables data used to address the three research questions.

Using Excel I was able to generate graphs displaying the profits a Takealot warehouse would make per customer sale within a day. The results were as follows.

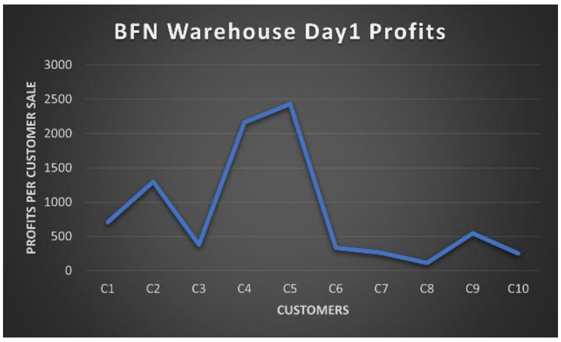

Figure 5: BFN Warehouse Day 1 Profits

The above graph shows the profits generated by the Bloemfontein Warehouse on Day 1 of their sales.

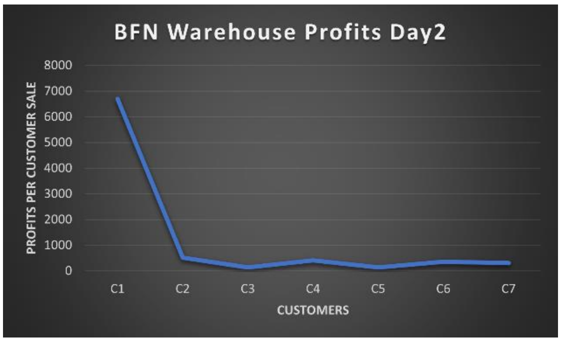

Figure 6: BFN Warehouse Profits Day 2

The above graph shows the profits generated by the Bloemfontein Warehouse on Day 2 of their sales.

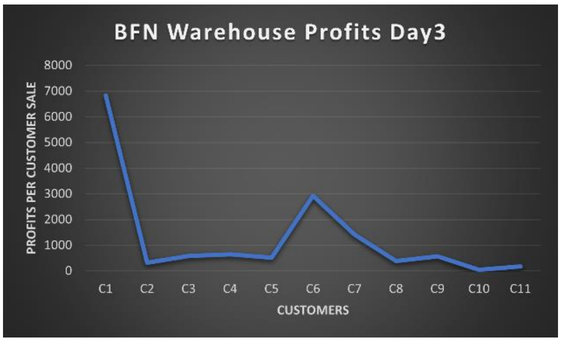

Figure 7: BFN Warehouse Profits Day 3

The above graph shows the profits generated by the Bloemfontein Warehouse on Day 3 of their sales.

Figure 8: BFN Warehouse Profits Over 3 Days

The above graph displays the profits made over the course of three days by the Bloemfontein Warehouse. The profits generated below 1000 rand were from the customers who have placed an order for one or two fairly affordable items. Visually it is clear to see that on Day 3 more profits were made than compared to Day 2 or Day 1. This could also be the case as Day 3 had the most purchases from the Bloemfontein Warehouse due to customers close by being allocated to that specific warehouse.

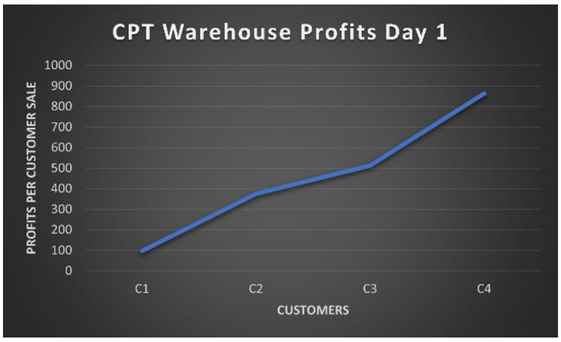

Figure 9: CPT Warehouse Profits Day 1

The above graph shows the profits generated by the Cape Town Warehouse on Day 1 of their sales.

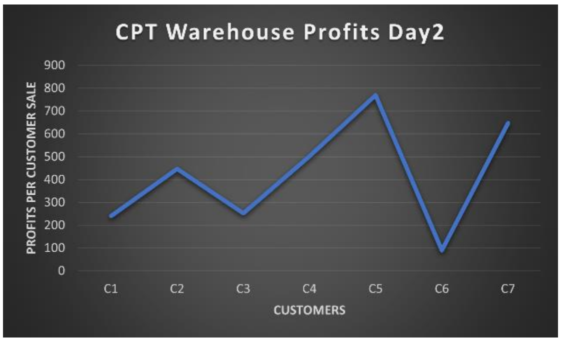

Figure 10: CPT Warehouse Profits Day 2

The above graph shows the profits generated by the Cape Town Warehouse on Day 2 of their sales.

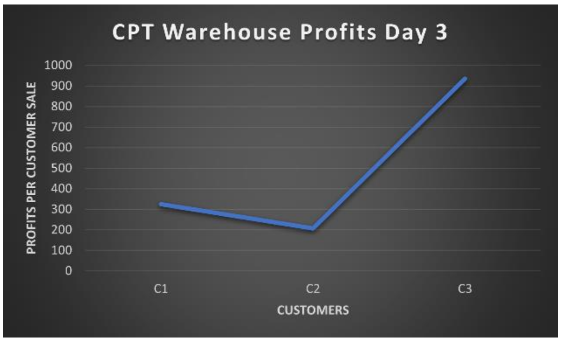

Figure 11: CPT Warehouse Profits Day 3

The above graph shows the profits generated by the Cape Town Warehouse on Day 3 of their sales. On this day there were only three customers who made a purchase from their warehouse.

Figure 12: CPT Warehouse Profits Over 3 Days

The above graph displays the profits made over the course of three days by the Cape Town Warehouse. Visually Day 3 was when the most profits were generated as seen by the two spikes in the profit increases. The plots where the line graph intercepts with the zero line are when no sales were made then so no profits were generated. We can also say that Day 1 had three sales lesser than Day 2 and Day 3.

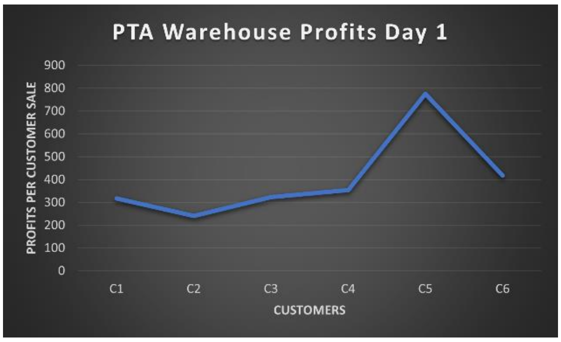

Figure 13: PTA Warehouse Profits Day 1

The above graph shows the profits generated by the Pretoria Warehouse on Day 1 of their sales.

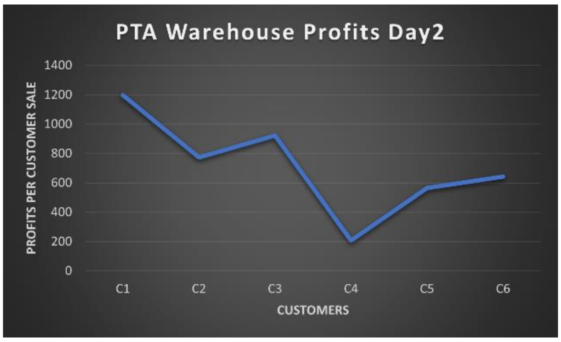

Figure 14: PTA Warehouse Profits Day 2

The above graph shows the profits generated by the Pretoria Warehouse on Day 2 of their sales.

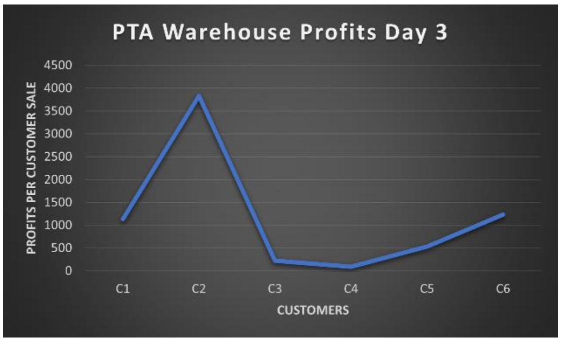

Figure 15: PTA Warehouse Profits Day 3

The above graph shows the profits generated by the Pretoria Warehouse on Day 3 of their sales.

Figure 16: PTA Warehouse Profits Over 3 Days

The above graph displays all the profits for the first three days of the Pretoria Warehouse. The visuals indicate that there was a spike in profits on Day 3. The least profits generated were on Day 1. The most profits that were generated on Day 3 were a profit of approximately R4 000 for the specific purchase of a particular customer.

B. MATLAB Simulation Results

To load data into MATLAB Simulink to run it through my simulation subsystems, I used the command code "Simulink.sdi.view" to open the data inspector tab where I would be able to import my data.

<u>The Simulation of the Takealot Inventory System:</u>

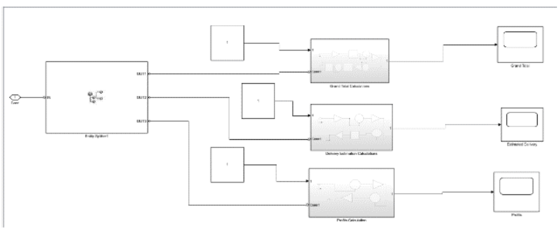

Figure 17: Takealot Inventory System Simulation

Before the simulation could be created, the necessary data had to be imported. The code "Simulink.sdi.view" was used to import all the columns from the python tables as vectors to be able to implement them within the simulation.

The above simulation has three subsystems that each have specific tasks to complete. Initially, the simulation is suppose to calculate the grand total payment the customer should pay when buying items online and this output should be displayed for the customers to see. Secondly, the simulation should use the shortest estimated distance between the allocated warehouse and the customers location and display it for the customers to see as their estimated time of delivery which is two days after the items are packaged, sealed and cleared to be delivered. The final part of the simulation should calculate the profits each warehouse would make per customer sale for each day. However, there were a few complications in running the simulation. The successful run of the simulation will be listed under the list of future improvements.

<u>Grand Total Calculation Subsystem:</u>

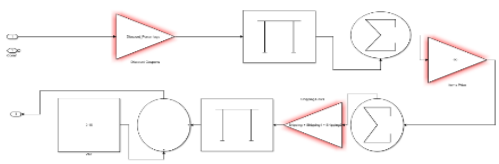

Figure 18: Customer Event Subsystem Simulation

The simulation consists of three subsystems. The first subsystem which is the customer event system, aims to provide the customer with the information they need to do their online payment and ensure that they are paying for the items they have added to their cart. This subsystem simulation would be used by the web development department of the company.

<u>Delivery Estimation Calculation Subsystem:</u>

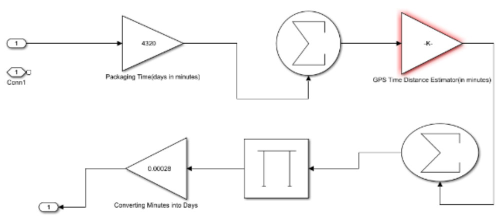

Figure 19: Delivery Event Subsystem Simulation

The second subsystem is the delivery event system. This would be used by the logistics department of the company. The subsystem aims to provide the mobile logistics technician with the information they need to determine where the parcel should be delivered and how long it would take to reach that destination.

<u>Profit Calculation Subsystem:</u>

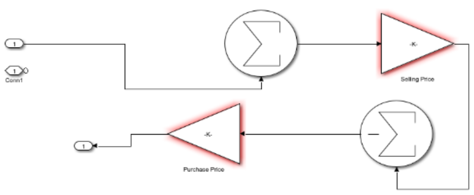

Figure 20: Profits Event Subsystem Simulation

The final subsystem is the profits event system. This subsystem would be used by the finance department of the company. This subsystem aims to provide the company with real time results to interpret how much profit the company would make per order placed by a customer on a particular day.

Applications of Inventory Models (Literature of Results)

An application of preservation technology in inventory control system with price dependent demand and partial backlogging

This article addresses the issue of product preservation through using an inventory control system. The inventory control systems are used to measure the non-instantaneous deterioration of products through using preservation technology as well. The swarm optimisation QPSO is introduced to overcome the non-linear optimisation problems. The demand of a product is dependent on the selling price of the product. The algorithm revealed through the inventory control system that the demand parameters chosen a and b by the QPSO optimisation algorithm would result in a hight profit and the cost of preservation would have a reverse effect. The suggested solution from these results would be for the retailers to sell their products as early as they can to reduce the cost of preservation and ultimately increase their profits as well.(SubhashChandra Das, 2020)

Inventory Simulation and Modelling Newspaper Sellers Problem

This video introduced a Newspaper Sellers problem that had the following characteristics. The newspaper company would buy newspapers in bundles of ten for 33 cents per paper and would sell them for 50 cents per paper. All the scrap paper or leftovers would be sold for the salvage value of 5 cents each.

There were three types of days which affected the sales of the newspapers per day. The three types of days were good, fair, and poor which each had a probability of 0.35, 0.45 and 0.20 respectively. It was also mentioned that the distribution of the papers demanded of each of these days were 2.15.

The problem statement comprised of determining the optimal number of papers the newspaper seller should purchase on a particular day. The solution approach was to simulate the demands for 10 days and record the profits from the sales of each day.(ItsNowOrNever, 2016)

The distribution of newspapers demanded on each day was:

Demand	**Good**	**Fair**	**Poor**
40	0.03	0.10	0.44
50	0.05	0.18	0.22
60	0.15	0.40	0.16
70	0.20	0.20	0.12
80	0.35	0.08	0.06
90	0.15	0.04	0.00
100	0.07	0.00	0.00

Random digits were generated for the types of news days which were: 58,17,21,45,43,36,27,73,86 and 19. The random digits generated for the demand were as follows: 93,63,31,19,91,75,84,37,23 and 02.

This system operated under the assumption that the seller buys 70 newspapers each day.

For the simulation the following data was recorded.

RDA for the type of Newsday:

Type of Newsday	Probability	Cumulative Probability	RDA
Good	0.35	0.35	00-35
Fair	0.45	0.80	36-80
Poor	0.20	1.00	81-00

Demand	**Probability**			**Cumulative Probability**			**RDA**		
	Good	**Fair**	**Poor**	**Good**	**Fair**	**Poor**	**Good**	**Fair**	**Poor**
40	0.03	0.10	0.44	0.03	0.1	0.44	0-3	0-10	0-44
50	0.05	0.18	0.22	0.08	0.28	0.66	4-8	11-18	45-66
60	0.15	0.40	0.16	0.23	0.68	0.82	9-23	29-68	67-82
70	0.20	0.20	0.12	0.43	0.88	0.94	24-43	69-88	83-94
80	0.35	0.08	0.06	0.78	0.96	1	44-78	89-96	95-00
90	0.15	0.04	0.00	0.93	1	1	79-93	96-00	
100	0.07	0.00	0.00	1	1	1	94-00		

Profit = (Revenue from the sales) – (cost of the newspapers) – (lost profit from excess demand) + (salvage from the sale of scrap).

The simulation table results using the above tables were as follows:

Day	**Type of Newsday**	**Demand**	**Revenue from the**	**Lost profit from the**	**Salvage from sale of the**	**Daily Profit**
1	Fair	80	35	1.7		10.20
2	Good	80	35	1.7		10.20
3	Good	70	35			11.9
4	Fair	50	25		1.00	2.9
5	Fair	80	35	1.7		10.20
6	Fair	70	35			11.9
7	Good	90	35	3.4		8.5
8	Fair	60	30		0.5	7.40
9	Poor	40	20		1.5	-1.60
10	Good	40	20		1.5	-1.60

Newspaper Sellers Problem Inventory Management System Simulation and Modelling

The example presented in the video address the newspaper seller’s problem, however it used different values and expected different outcomes as well. This inventory control system addresses purchasing and sales of newspapers. The newspaper seller buys papers for 5tk each and sells them for 8tk each. Newspapers are bought in bundles of 5. The newspaper that is not sold at the end of the day will be sold for 1tk each at salvage price.

There are three types of days on which the seller operates, namely: good, fair, and poor in which their probabilities of occurring are 0.50,0.30 and 0.20 respectively.

The problem statement is comprised of determining the optimal number of papers the newspaper seller should buy. The solution comprises of simulating demands for 10 days and then recording the profits from the sales of each day. A simulation table should be prepared for the decision to buy 65 papers.(Ahamed, 2020)

Profits = (sale price) – (cost of the newspapers) – (lost profit from the excess demand) + (salvage from the sale of scrap papers)

The distribution table for the daily demand:

Demand	Good	Fair	Poor
50	01-08	01-10	01-44
60	09-23	11-28	45-66
70	24-43	29-68	67-82
80	44-78	69-88	83-94
90	79-93	89-96	95-00
100	94-00	97-00	

Random values chosen for the demand are as follows: 70,17,36,83,54,4,42,20,67,58,9

Distribution for the type of the sale day:

Newsday	Probability	Cumulative Probability	RD Assignment
Good	0.20	0.20	01-20
Fair	0.50	0.70	21-70
Poor	0.30	1.00	71-00

Random values chosen for the Newsday are as follows: 89,45,84,3,64,55,84,25,6,70,34

Simulation Table for 10 days to buy 65 newspapers:

Days	RD for Type	Type of Day	RD for	Demand	Sale Price	Excess	Salvage	Daily Profit
1	89	Poor	70	70	520	15	0	180
2	45	Fair	17	60	480	0	5	160
3	84	Poor	36	50	400	0	15	85
4	3	Good	83	90	520	75	0	120
5	64	Fair	54	70	520	15	0	180
6	55	Fair	4	50	400	0	15	85
7	84	Poor	42	50	400	0	15	85
8	25	Fair	20	60	480	0	5	160
9	6	Good	67	80	520	45	0	150
10	70	Fair	58	70	520	15	0	180
				650				

Inventory System Discrete Event Simulation in Dynamic in Python (Event-Scheduling)

The video under this review of literature focused on a model like the Takealot Inventory Control System of this report. Within this model problem description selling products in stock is only allowed. The stock products would sell for r = 100 each. The customer interarrival time can be described as: d ~ expon (λ =5). Each customers demands can be described as: D ~ uniform (1,4) products. The order policy can be described as (s,S): when the inventory falls to x < s, order y= S-x. The costs can be calculated by c(y) = 50y to order. The delay delivery can be described as l=2 days until delivery. There is a holding cost of h=2 per item per day.(Grogan, 2016)

The following is a representation of the simulation system:

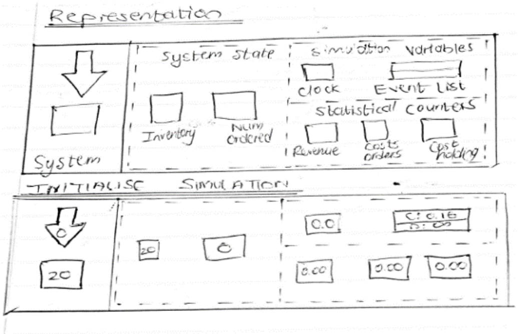

Figure 21: Simulation System Representation

The figure below shows the simulation and the results for t=0.16.

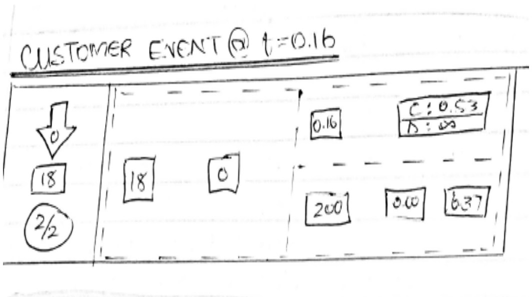

Figure 22: Simulation for t=0.16

The following algorithmic diagram represents the Advanced Time/Handle Event and how it would run during the simulation:

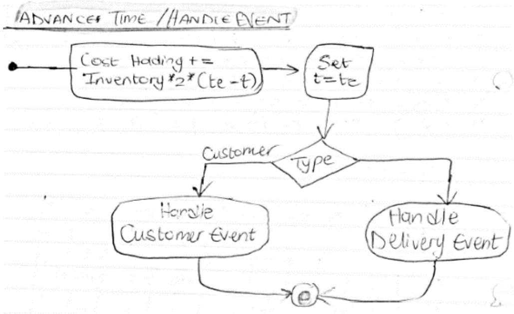

Figure 23: Advanced Time/Handle Event

The algorithmic diagram below represents how the simulation would run the Handle Customer Event and in what order execution would be done:

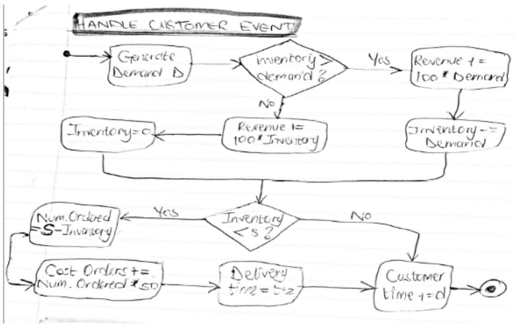

Figure 24: Handle Customer Event

The algorithmic diagram below represents how the simulation would run, and which steps would be taken to complete the Handle Delivery Event when simulated:

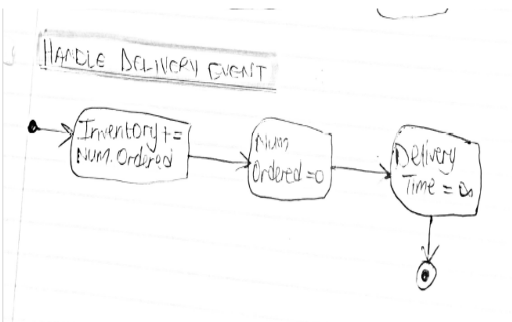

Figure 25: Handle Delivery Event

Inventory Control System by Using Vendor Managed Inventory (VMI)

The aim of this study is to build an inventory control system that can manage goods within a retail business in which the Vendor Managed

Inventory method. This method provides information such as the sales data and goods for the supplier. The inventory system is used to calculate safety stock and reorder stock of goods based on the sales data. The results favoured should be that the goods should be available at approximately at 70 percent and the accumulation of the goods should be reduced to approximately 30 percent. (Sabila, Mustafid and Suryono, 2018)

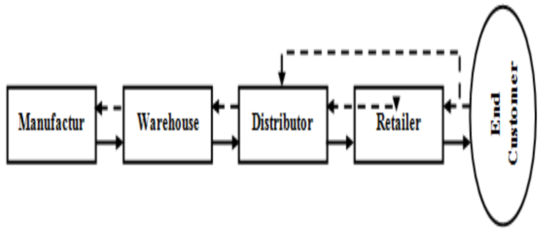

Figure 26: Supply chain of VMI

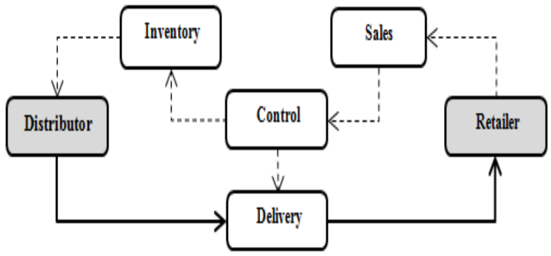

Figure 27: Model of inventory control system

Conclusion

In conclusion we have found that inventory systems are very useful in our lives, especially in the retailer sector. It can be applied to any department within the retailer, whether it includes stock taking in the warehouse department, calculating the daily profits within the finance department, or estimating the shortest route and time to deliver parcels to the customer by the logistics department. Inventory control systems can be applied anywhere.

For this Takealot Inventory Problem, one discovery that was made is that the retail business is bound to make a profit on their items sold whether it is a large value or not. The advantage the business has is that they are able to deliver to any and every location specified by the customer which increases the customer convenience and in turn boosts their sales daily. Another discovery is that the central warehouse was able to generate more profits as more customers were buying from the Bloemfontein Warehouse due to the closest warehouse allocation done before the simulation started.

For future recommendations, I would like to investigate possible solutions to automatically record the estimated time it would take for the mobile logistics technician to reach the customers destination within manually searching for the results and recording it. I would also like to make use of a larger dataset which could increase the accuracy of the results in the end. Another future improvement would be to correct the errors in the simulation in order for it to produce the expected outputs which were achieved through using excel functions.

Another recommendation would be to create a function that calculates approximately how many days it would take for a parcel to be delivered and at what time it would arrive at the customers location.

References

Ahamed, S. (2020) *Newspaper Sellers Problem, Inventory Management Syetem, Simulation and Modeling*. Bangladesh: Youtube.com. Available at: https://www.youtube.com/watch?v=WfiuBSvxPCw .

Center, M. help (2021) *Simulink*. Available at: https://www.mathworks.com/help/simulink/index.html?s_tid=srchtitle_simulink_1 (Accessed: 27 September 2021).

Direct, B. (2015) *The Barcode Blog*. Available at: https://www.barcodedirect.com/the-evolution-of-inventory-management/ (Accessed: 27 September 2021).

Google (2021) *Google Maps*. Available at: https://www.google.co.za/maps/@-34.6283864,27.2516951,5z (Accessed: 27 September 2021).

Grogan, P. (2016) *Inventory System Discrete Event Simulation in Python (Event-scheduling)*. Youtube.com. Available at: https://www.youtube.com/watch?v=7LuN_6m7h2o&t=4s.

Inc.com (2020) *Inventory Control Systems*. Available at: https://www.inc.com/encyclopedia/inventory-control-systems.html (Accessed: 27 September 2021).

ItsNowOrNever (2016) *System Modeling and Simultion: Newspaper Seller's Problem*. India: Youtube.com. Available at: https://www.youtube.com/watch?v=OH8MRT8eqRI.

Sabila, A. D., Mustafid, M. and Suryono, S. (2018) 'Inventory Control System by Using Vendor Managed Inventory (VMI)', *E3S Web of Conferences*, 31, pp. 4–7. doi: 10.1051/e3sconf/20183111015.

SubhashChandra Das, A. M. Z. (2020) 'An application of preservation technology in inventory control system with price dependent demand and partial backlogging', *Alexandria Engineering Journal*. Available at: https://www.sciencedirect.com/science/article/pii/S1110016820301058.

Takealot.com (2021a) *More about our journey*. Available at: https://www.takealot.com/about/our-journey (Accessed: 27 September 2021).

Takealot.com (2021b) *Takealot*. Available at: https://www.takealot.com/ (Accessed: 27 September 2021).

Takealot.com (no date) *Robots.txt file*. Available at: https://www.takealot.com/robots.txt (Accessed: 27 September 2021).

UpKeep (2021) *What are the different types of inventory systems?* Available at: https://www.onupkeep.com/answers/inventory-management/types-of-inventory-systems (Accessed: 27 September 2021).