Final Project

Dalton Murray

College of Business & I.T., Lawrence Technological University

INT 4203, Systems Analysis and Design

Dr. Anthony Padalino

December 6, 2022

**Final Project  
\*Please note that because of how word compresses images it makes it impossible for the diagram images to show up properly due to the size and resolution of them all. Because of this, I have compressed the folder with all of the images and also uploaded this folder to canvas\***

**Main objective:**

Our company’s goal is to ensure quality of products, fast processing speeds, offer quick delivery, and ensure that this all can be done at the convenience of the customer while retaining our primary goals. To solve these problems, we have almost entirely eliminated the human issue, we focus on having an entirely human-less process. In order to do this, we have implemented many features to our store such as drone delivery systems, RFID scanning, AI features to watch for inventory, robots to ask questions to, clean up messes, restock shelves, and handle almost all essential tasks.

**Transactional System**

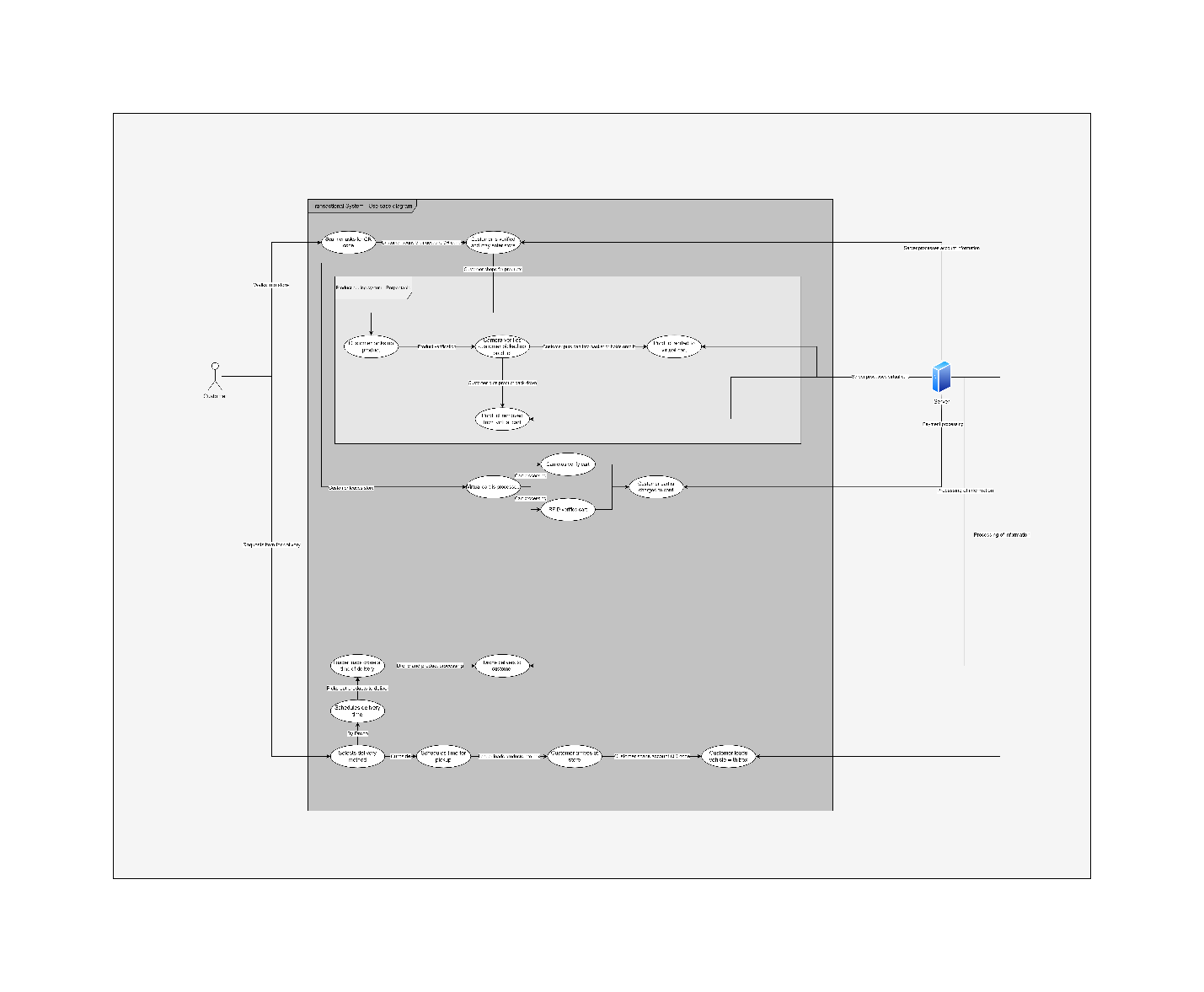
**Business Case:**

Our transactional system surpasses all existing competitors and potential future competitors. We have designed our systems to be as efficient as physically and logistically possible. We have designed this system in order to be able to serve our customers in the best way possible by preventing the amount of time it takes in order to get the products they need when they need it.

Some internal factors that will affect this system are if there are any sensors or cameras which become faulty or any autonomous system which somehow fails.

Some external factors that will affect this system are weather, weather will effectively make our drone delivery systems useless until a better way of making the drones weather resistant become available to us or we create a design for large amounts of rain or wind. A large portion of our company sales also depends on the honesty of our customers, however, all footage and sensor data is recorded so if mistakes are made it is possible to go back to them to review accuracy.

**Use case diagram:**

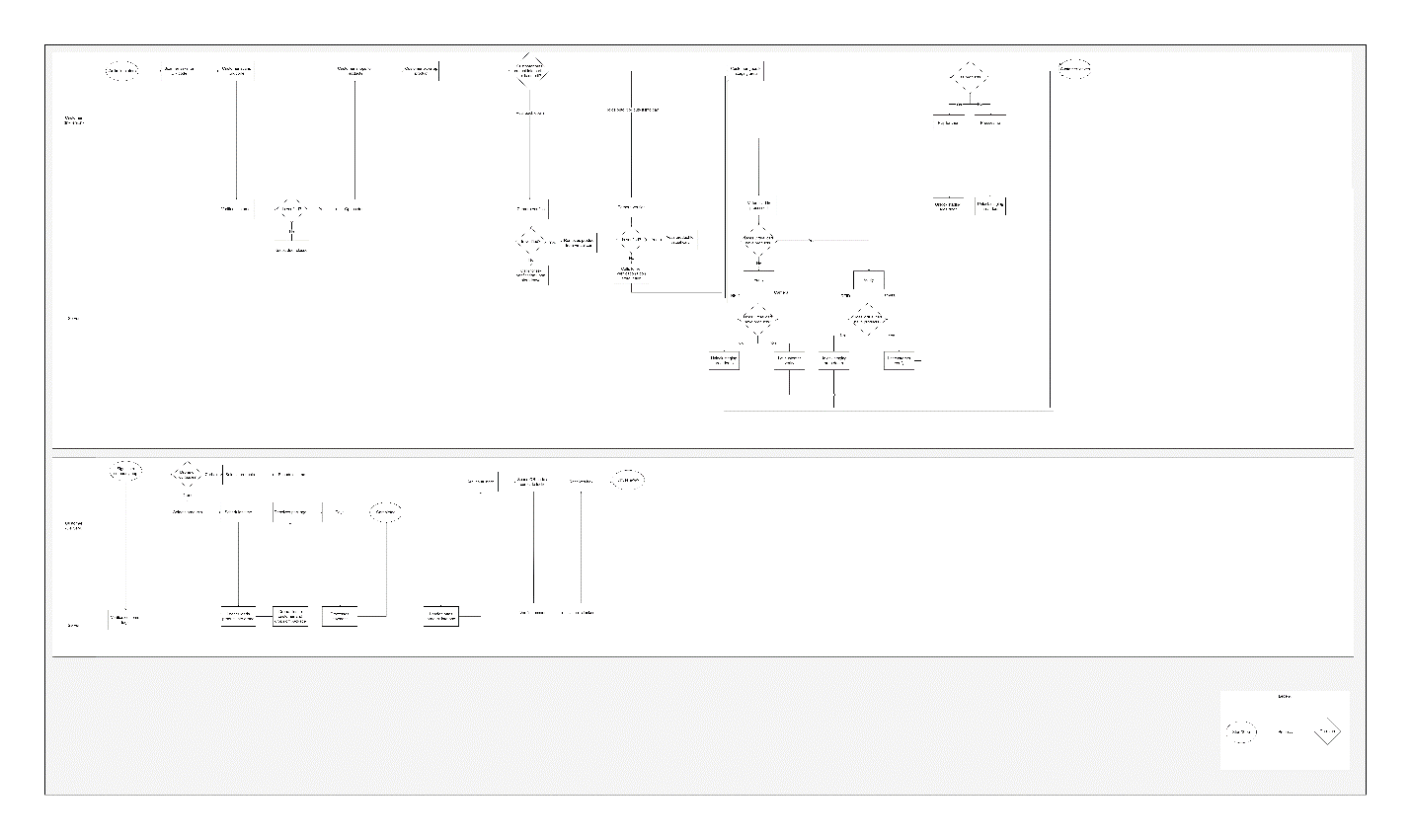
****

(Image is available in folder use case diagram for better viewing)

Brief discussion:

When creating the use case diagram for the transactional systems I immediately started to overcomplicate it without realizing that the amount of detail I was putting into it was too much. This made me want to simplify my design much further down until I have the current design which I am using now. I have simplified it down to how the customer is able to pick up and purchase products in the store as well as using delivery methods such as curbside delivery and drone delivery. The use case diagram shows how when the customer does something the system reacts to it, such as picking up product it verifying the product and adding it to their virtual cart until they leave the store or put down the product. This also shows how the drone and curbside deliveries work autonomously by using loading robots and putting the product into boxes and then putting it into a locker outside of the store or onto a drone and the drone determining the optimal flight path and then delivering the products to the customer and coming back.

**Business process diagram:**

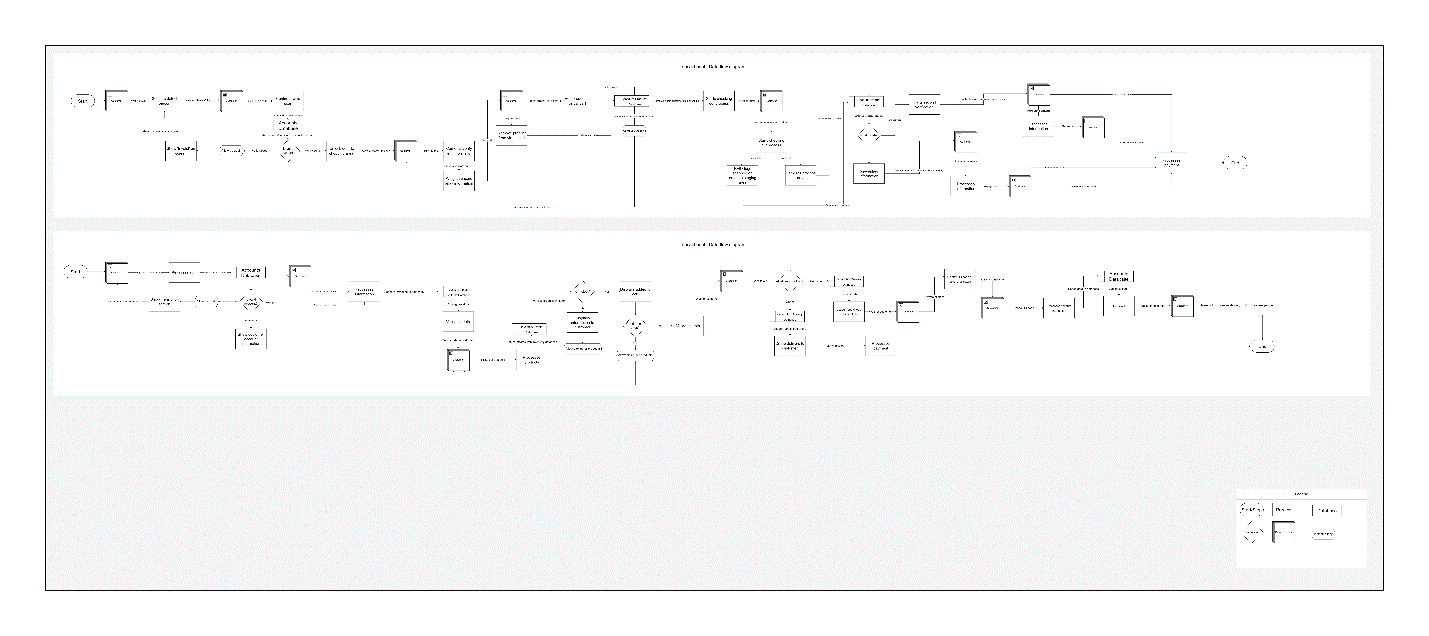
****

(Image is available in folder business process diagram for better viewing)

Brief discussion:

For the business process diagram, I went a little more complex than the use case diagram as I really wanted to show how a lot of interactions are handled between the customer and the server/systems. It is split in two sections of two parts, one for handling in-person interactions and another for handling curbside and drone delivery options, finally each of these are in two parts one part to show what the customer does and another part to show what the server does. In the transaction system a majority of work is being done by the server whereas the customer does not have to do much of work at all, and demonstrates how the server verifies and reverifies every action done and handles what to do after that such as further processing or waiting on the customer to do something like leave the store or to display actions to confirm to the customer.

**Data flow diagram:**



(Image is available in folder data flow diagram for better viewing)

Brief discussion:

The data flow diagram is also much more complex than the use case diagram, as I mostly used the use case diagram to oversimplify the system and then break it down with the business process diagram and data flow diagram. With the data flow diagram, it could be broken down much, much more however I decided to limit myself to what is displayed before it becomes too complex making it less understandable. We are able to see that the transaction system does many checks and writes to databases to log everything being done so that everything may be looked back to later, analyzed, verified, and corrected if needed. The data flow diagram shows all of the actions and the data as a result of it and what happens to the data all the way from walking into the store to processing payments, handling drone deliveries and curbside locker deliveries as well as how the customer (external entity) interacts with the system and changes data and how we process it.

**Human Resource System**

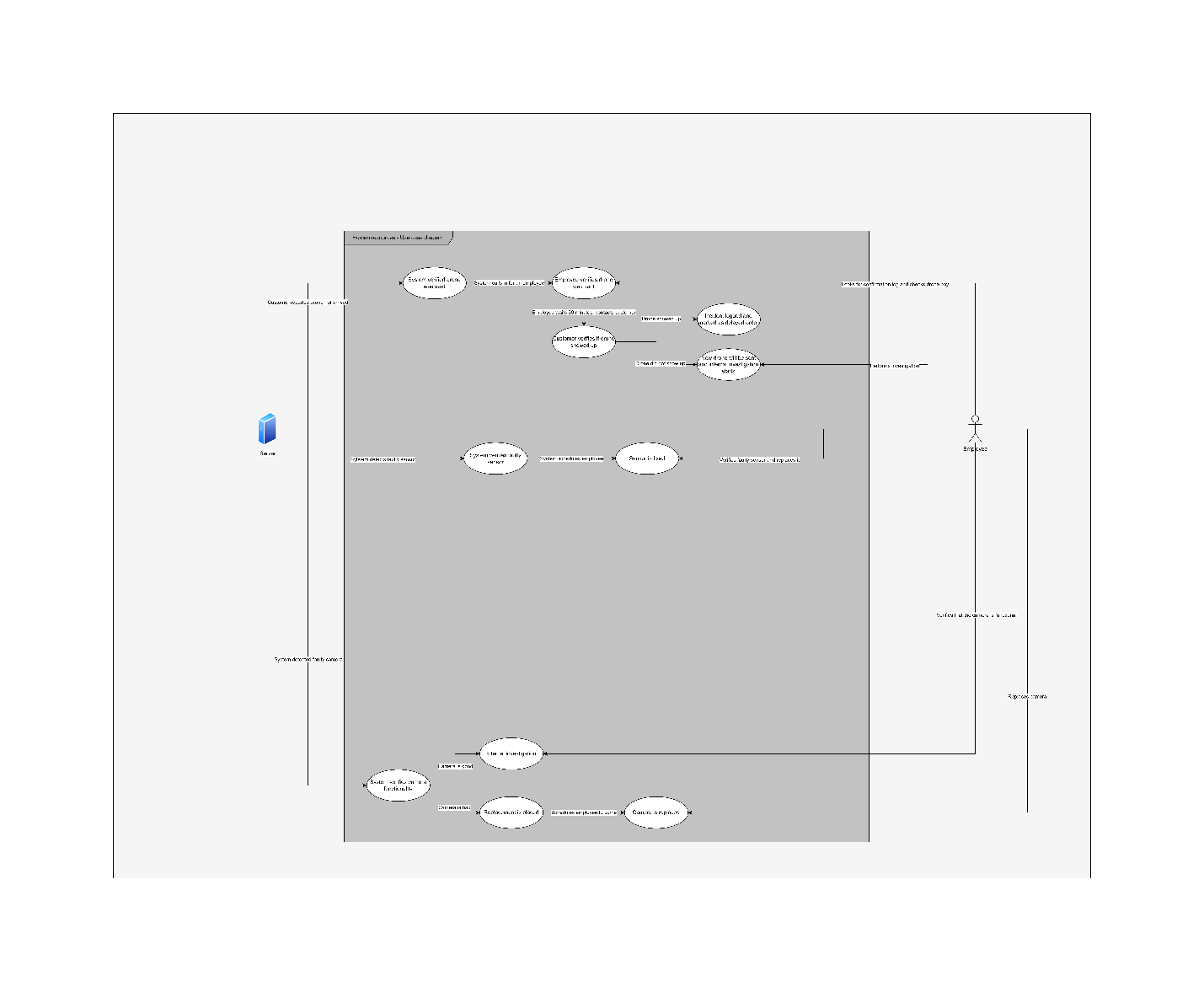
**Business Case:**

Our human resource system is a bit different than other companies because the vast majority of our stores focus on not having humans at all. Despite this, there are some situations where we would have no choice but to have humans such as faulty sensors or cameras. This is where the human resource system focuses on.

Some internal factors that will affect this system are how many employees are currently in the area. Since we are an autonomous focused company we do not have many employees handling the actual stores and most of them are in research and development, as a result of this there may be only a few employees who handle differing regions of stores.

Some external factors that will affect this system are of course weather and how employees are able to get to a store to fix issues when they arise. We also have replacement parts for everything at every store location, however, if something were to break and somehow the parts used to replace it are not logged then they may not be available if something breaks again needing that part.

**Use case diagram:**

****

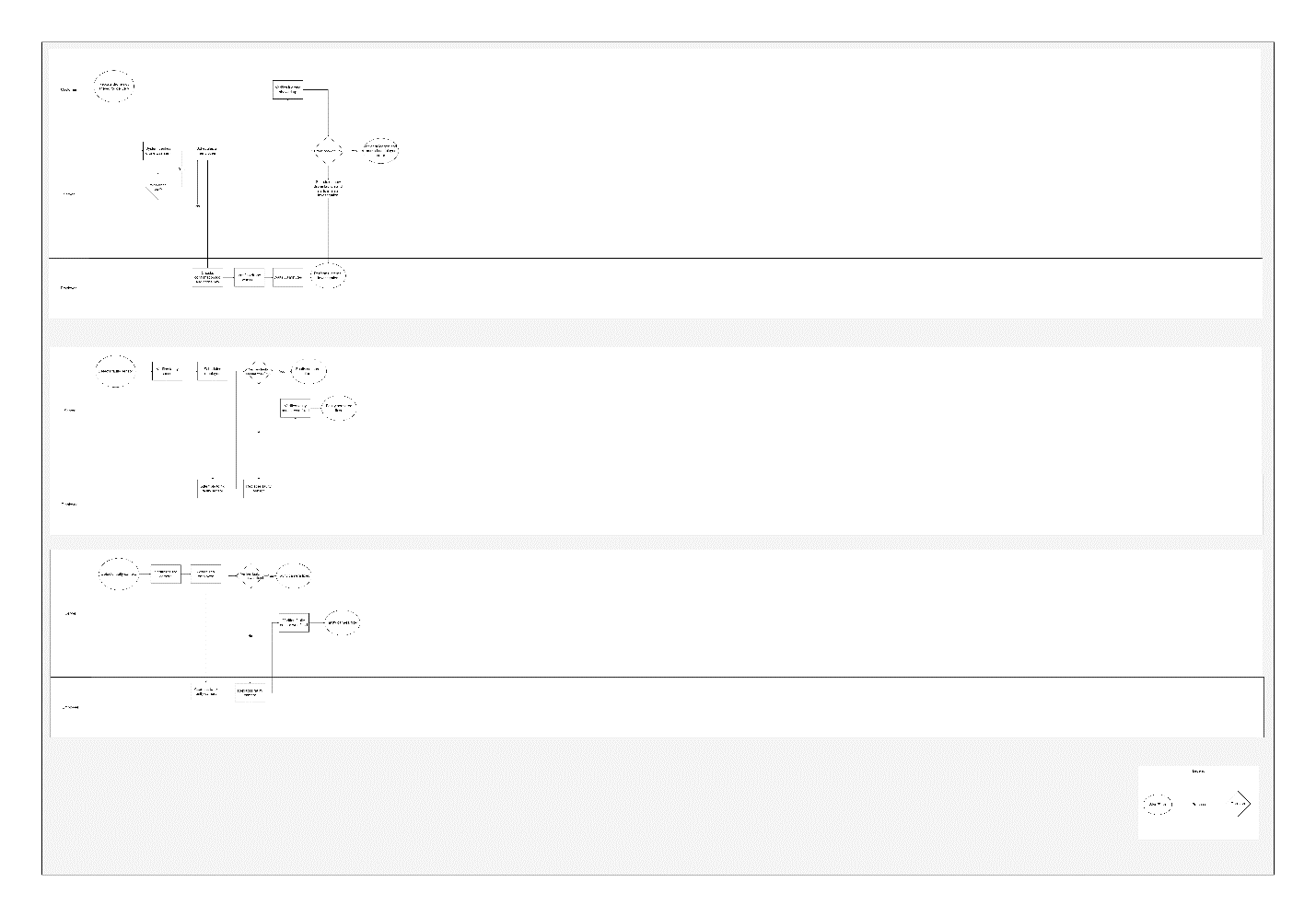
(Image is available in folder use case diagram for better viewing)  
Brief discussion:

Our company focuses on not needing humans, thus making the human resource system a bit unusual. There are only a few areas which our store actually needs humans and to use a human resource system, which is reflected in our use case diagram. The use case diagram has the system on the left side and the employees on the right side because the server sends information and interacts with the components in the store to then send information to the employee of what they need to do. The three things in the use case diagram are if a customer orders a delivery and then tells our system that the drone has not arrived yet, the second is detecting a faulty sensor, and the third is detecting a faulty camera.

If our customers report an order not received we need to verify that we actually sent out a drone, once we do this we can confirm if we did or did not, regardless of the result of this we then send out an employee to confirm that the system is accurate, finally we wait approximately 30 minutes to allow time for the customer to respond back if they have received a drone or not as they simply may be impatient or somehow the drone was delayed, after this if they report they received it we then end there, log it and mark it as a delayed order, however, if they have not received it we simply send out another drone with their product and start an internal investigation.

Our second and third actions in the use case diagram are simply for if the system detects a fault in our camera or sensor systems, if so it performs a diagnosis and communicates to send out an employee to first attempt to fix and then replace if the fix does not work.

**Business process diagram:**

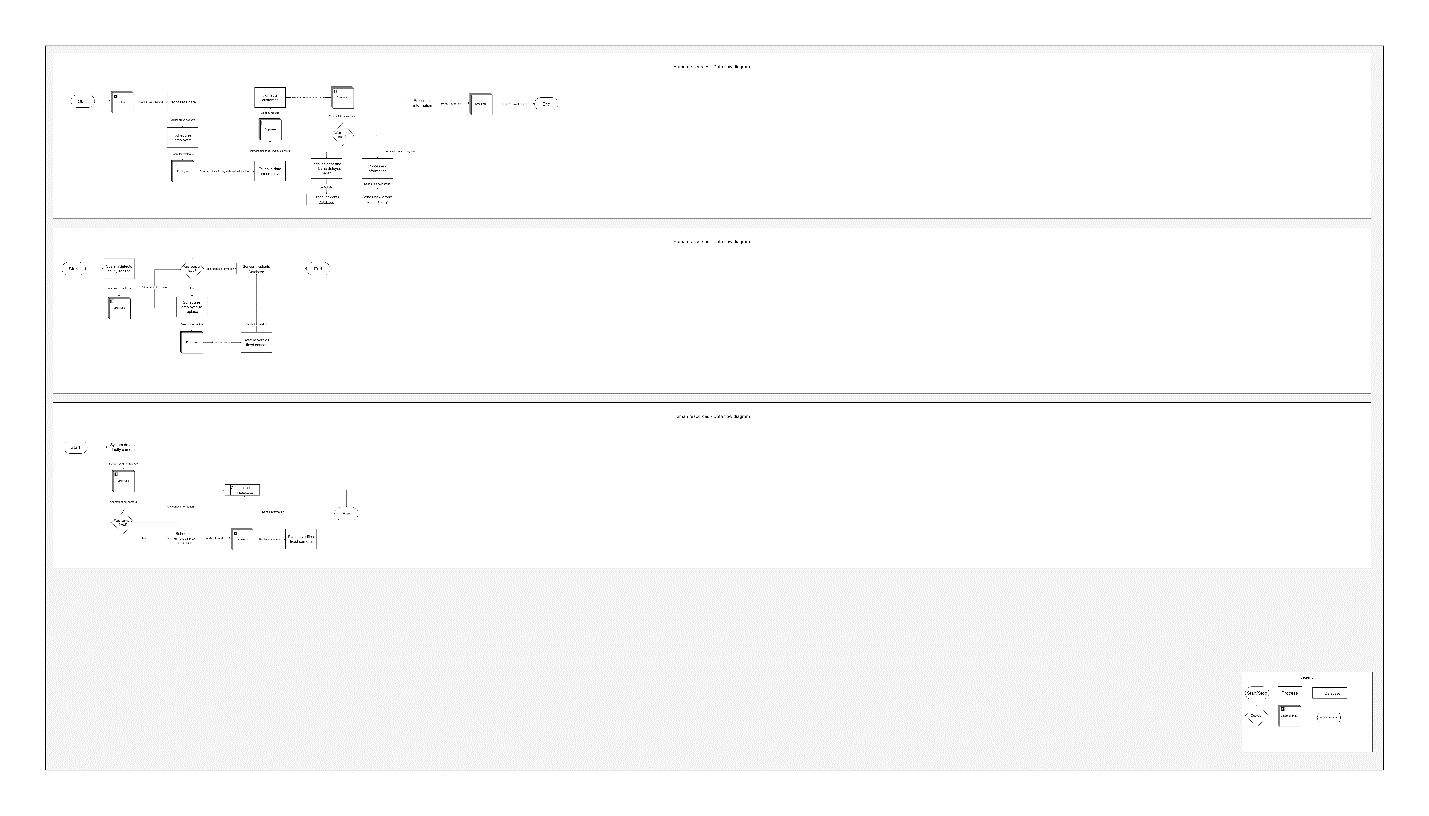
****

(Image is available in folder business process diagram for better viewing)

Brief discussion:

The business diagram for the human resource system is quite simple as there is not too much to it, however, room has been left to expand into greater detail if needed to fully understand the system. The business diagram has 3 possible things to it which are broken down into customer, server, and employee, and it shows the process of how we schedule for employees and detect faulty sensors and cameras. Once one has been detected it schedules an employee who then handles their side such as attempting to fix it, allowing the system to determine if faulty still or fixed, if faulty still it schedules a replacement and then verifies that the replacement has been fixed. It also uses customers to help verify if they have not gotten their drone delivery which allows us to then further proceed to see if it is just a delay for some reason or if a new delivery needs to be placed as well as showing an internal investigation being started if needed.

**Data flow diagram:**

****

(Image is available in folder data flow diagram for better viewing)

Brief discussion:

The human resources data flow diagram is also more simple compared to some other diagrams, however, is not as simple as the loyalty system diagram. With the human resources data flow diagram we show how the customer, system, and employee are able to interact with the systems and the resulting data and processing of them. Some examples of this is writing to a drone incidents database, sensor incidents database, as well as a camera incidents database so that we are able to record everything that has been done to each store and verify that everything is operating correctly. With the human resources data flow diagram it relies a lot on external entities such as an employee or a customer, and then verifies the actions taken have fixed what was wrong.

**Loyalty System**

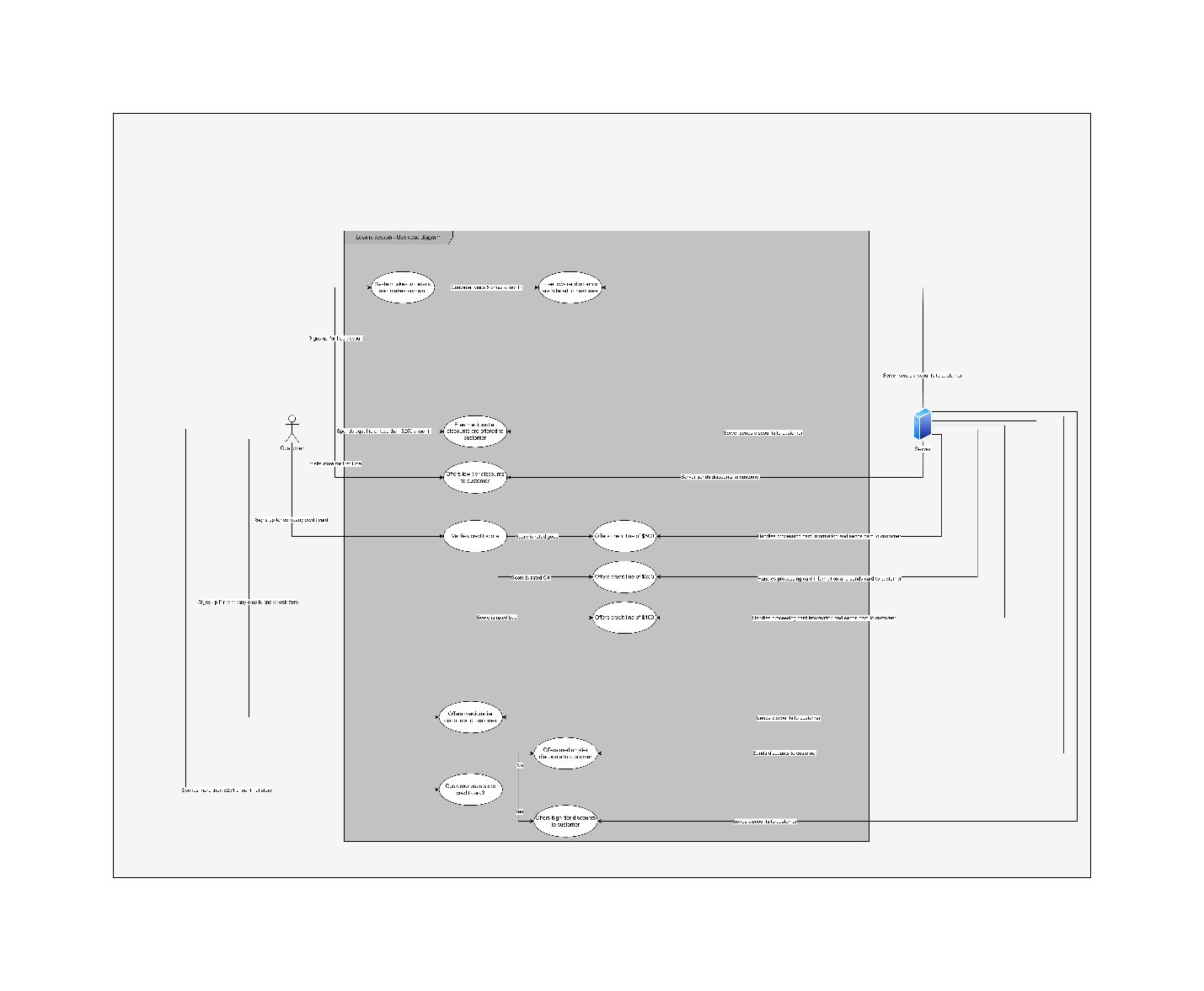
**Business Case:**

In order to make sure that our customers come back for not only our excellent service we want to give back to them. We are able to do this by offering free delivery for a certain amount of money spent in-store, offering coupons to them for products they are interested in, as well as offer discounts for signing up for newsletters, signing up for credit cards, and for using their store credit cards.

Some internal factors that will affect this system are that the systems are properly recording data about purchases as well as other events such as signing up for accounts and sending out data to make credit cards and ship to them.

Some external factors that will affect this system are dependent on the customer’s honesty, as well as them using their account and not signing up for multiple accounts, however, we have systems in place to attempt to prevent this, as well as weather for shipping out credit cards.

**Use case diagram:**

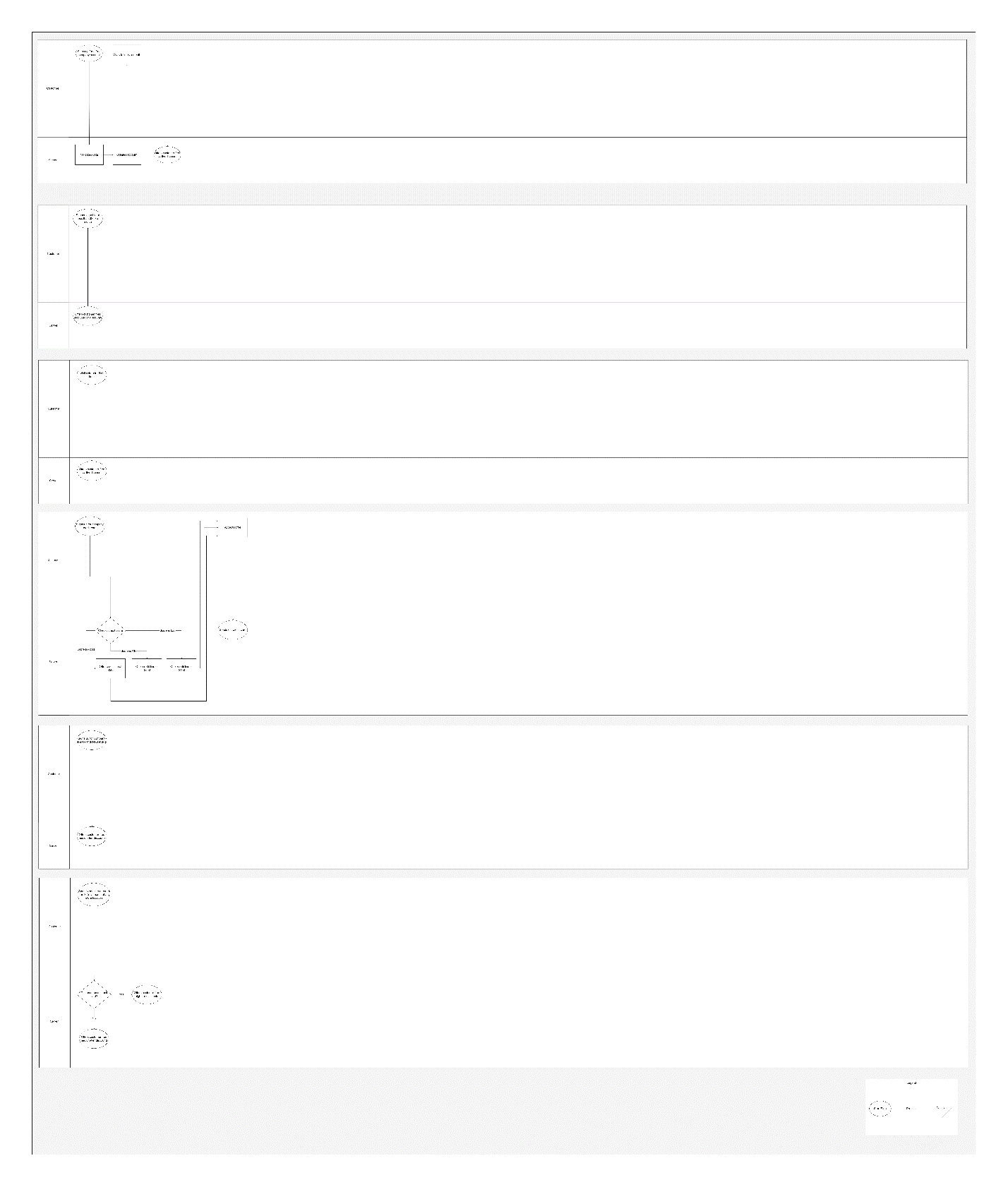
****

(Image is available in folder use case diagram for better viewing)

Brief discussion:

Our use case diagram shows how the customer is able to interact with the store and how the server and systems react to this. As our loyalty system is quite simple, there are only a few ways for customers to get discounts, however, these discounts can range from low-tier discounts from a couple cents off to high-tier discounts offering high percentages off from their entire purchase. Some examples of our loyalty system include detecting if they sign up for a free account and our systems sending out discounts, spending money on products without a store credit card, visiting for the first time, as well as spending money on products using a store card, or even signing up for newsletters in the form of email. Once a requirement is met our systems will automatically detect this and send out the discounts to the customers account.

**Business process diagram:**

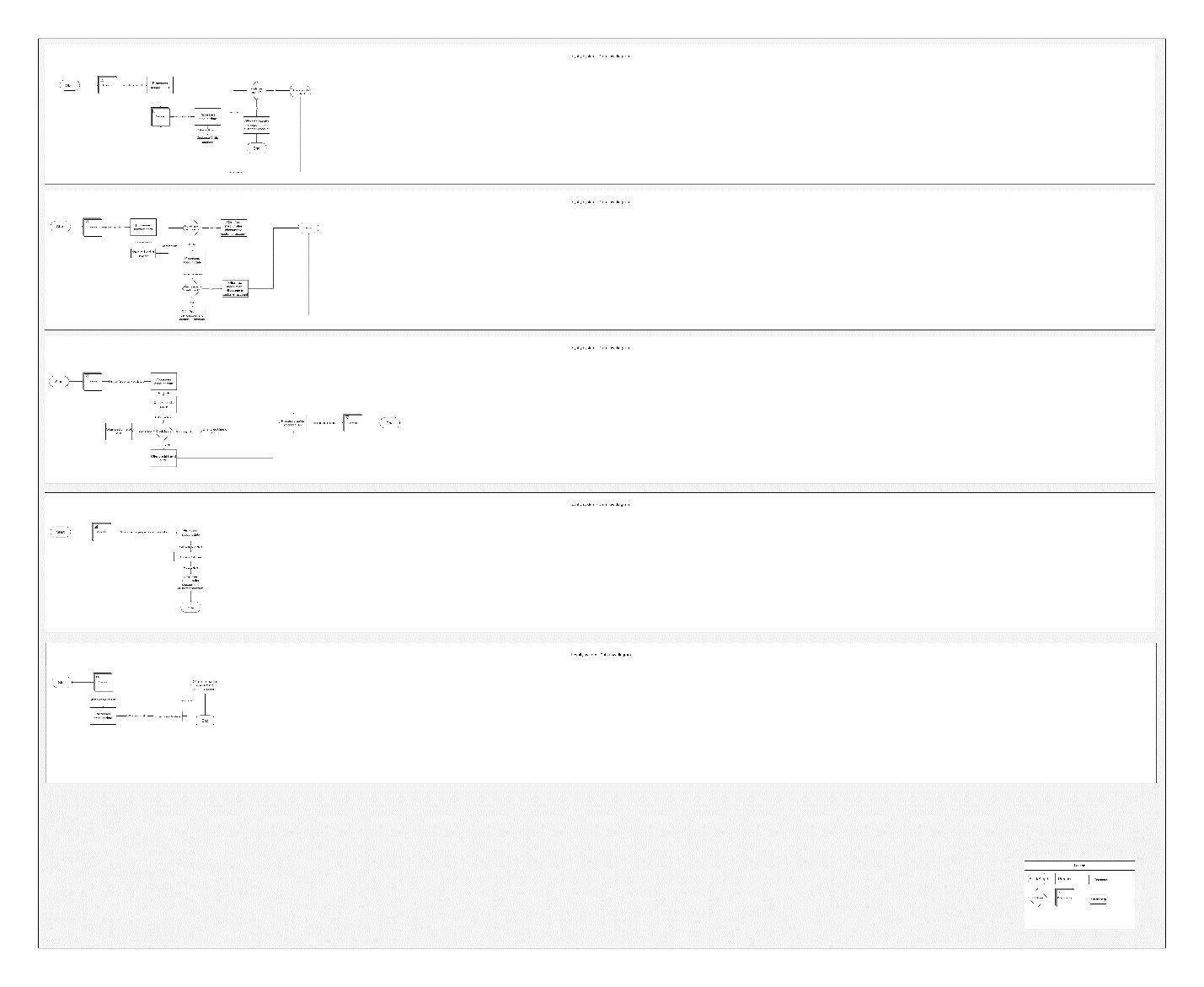
****

(Image is available in folder business process diagram for better viewing)

Brief discussion:

The loyalty system business process diagram is broken into 6 sections of 2 which are quite simple, however, could be broken down further if needed. It shows what actions are taken as a result of actions done by the customer such as visiting the store for the first time an then offering them free low-tier discounts, or signing up for a company credit card, performing a credit score check, and then determining what to do from there such as offering differing amounts of credit to the customer. The loyalty system business process diagram essentially visualizes and simplifies the use case diagram but shows further actions where necessary as well as being able to see paths taken such as if they spend greater than or equal to $251 a month on the store and if they use a store card or not and then offer them either medium-tier discounts if they don’t and high-tier discounts if they do.

**Data flow diagram:**

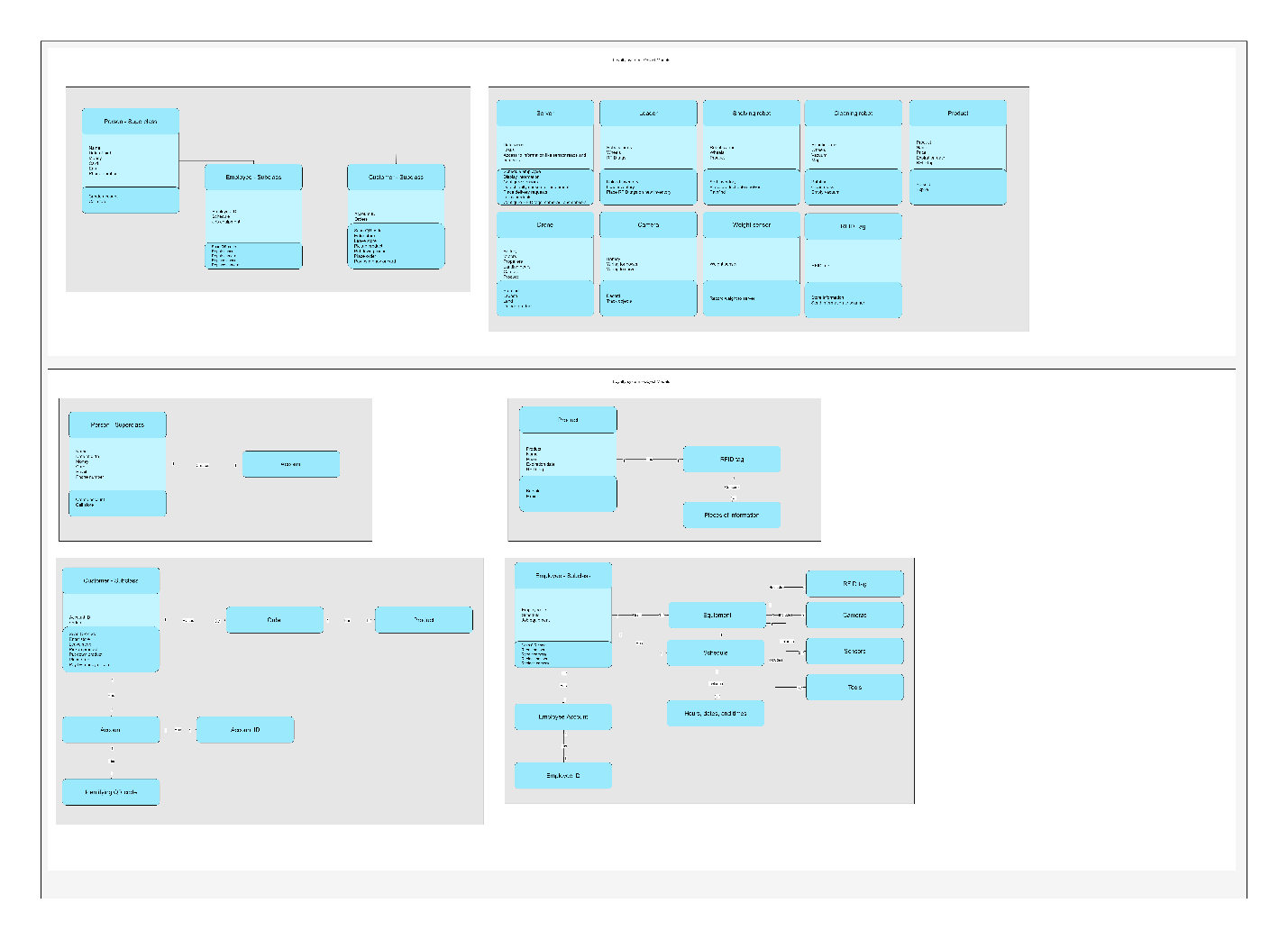
****

(Image is available in folder data flow diagram for better viewing)

Brief discussion:

The data flow diagram for the loyalty system is also quite simplified but could be broken down much further. The data flow diagram shows how the customer is able to interact with the system by doing different things and then the data it generates and how the system processes it, such as spending money at the store, checking the databases for how much they have spent monthly, and then determining if they are using a company credit card or not, and then showing the discounts we are able to offer to them. It shows the different decisions made and how it effects what discounts we give them, as well as if there are things they do which directly trigger discounts being given such as signing up for the company newsletter which does not need many checks to be done except for checking our accounts database to verify they have signed up for it, and then sending out discounts to them.

**Object models:**

****

(Image is available in folder object models for better viewing)

Brief discussion:

The object model diagram which I have made for the loyalty system is extremely simplified and is broken down into two parts. The first part is the classes and the superclass/subclasses such as person being a superclass and customer and employee being a subclass of the person superclass, which is in its own category and then has a few of the different components of the store in another category such as the server, loaders, shelving robots, cleaning robots, product itself, drones, cameras, weight sensors, and RFID tags. For each class I have listed the more important attributes it possesses and then some of the more important methods it has available to use. In the second section, I have listed some of the interactions between the objects, such as a single person able to create a single account, a single customer able to place multiple orders, and a single order being able to have one or more products in it. I then also do the same thing for the customers account, product, and some portions of the employee class.

**Supply chain system**

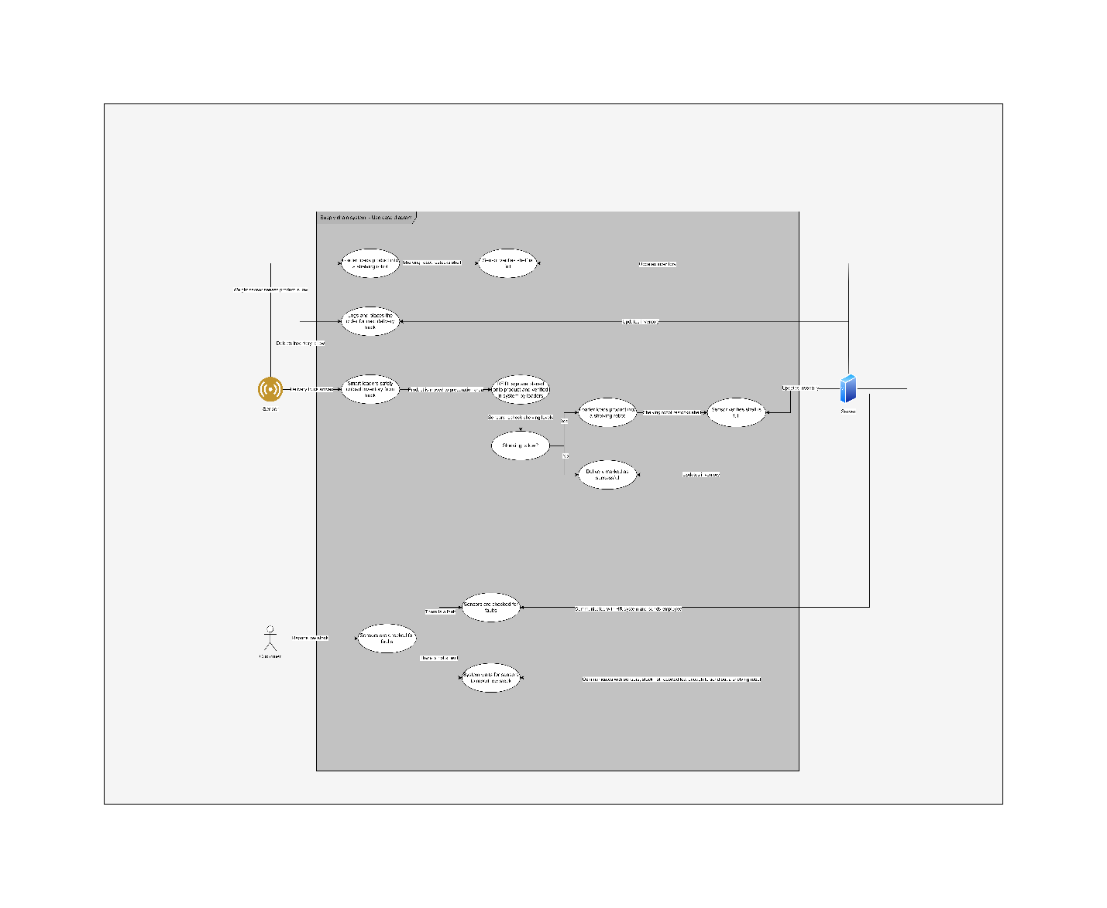
**Business Case:**

As our goal for the entirety of every store we own is to make them completely autonomous this does not limit our ability to have a proper supply chain and have products delivered, organized, prepared, and put onto shelves. In fact, our supply chain system given its complexity is able to work better than any other store managed by humans since they are able to with great accuracy predict shipping times and when we will run out of product.

Some internal factors that will affect this system are having faulty sensors or cameras which may be having problems with accurately predicting when to get deliveries placed.

Some external factors that will affect this system are of course delivery delays. Unfortunately, most of the time deliveries are unpredictable given the distance needed to ship products, however, our systems try to take into account unpredictability and add extra time for shipments when needed.

**Use case diagram:**

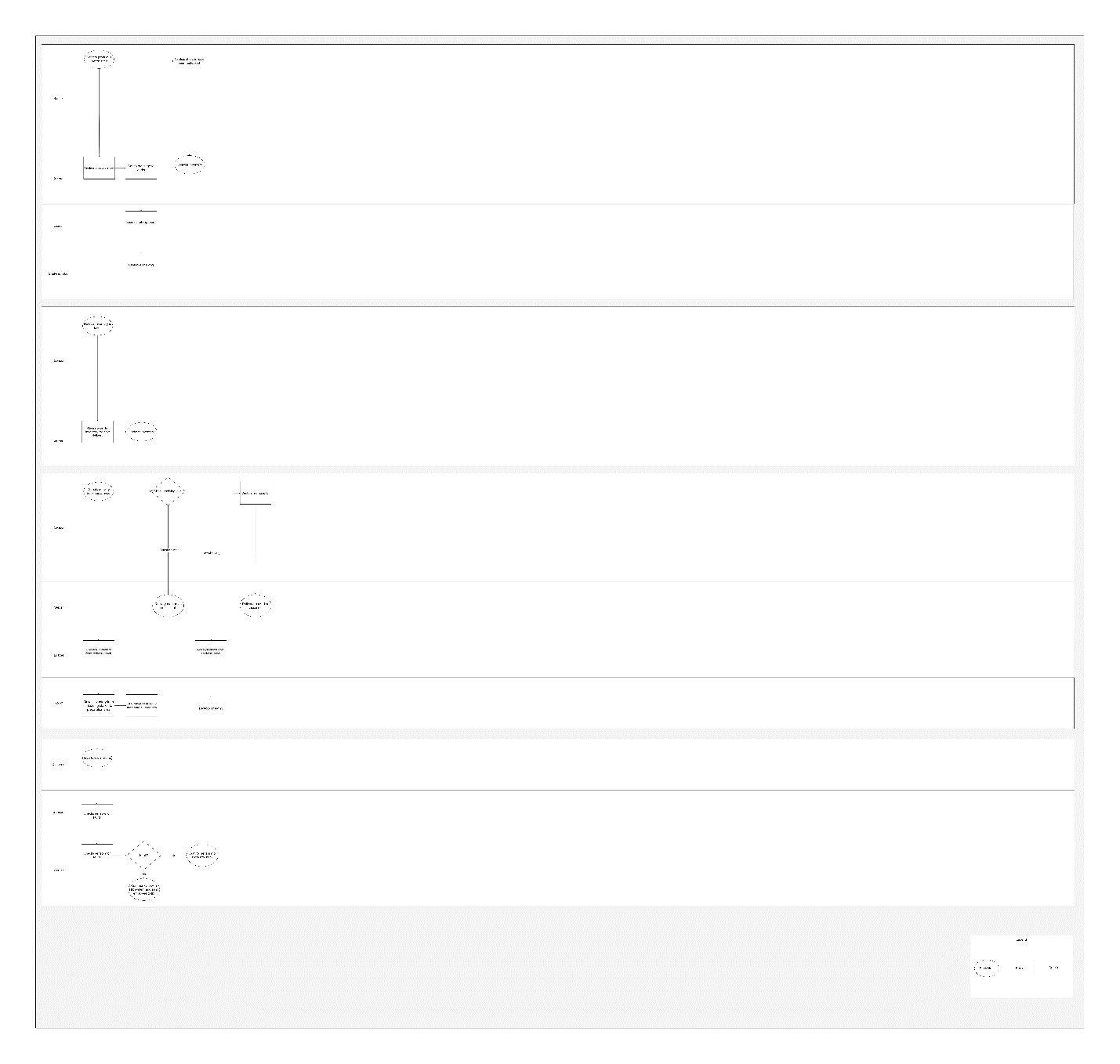
****

(Image is available in folder use case diagram for better viewing)

Brief discussion:

Our system for handling supply chain management is extremely complex, given this, I have almost oversimplified the system on the use case diagram. Firstly, in the use case diagram we have three main parts for handling sensors and its interactions with robots in our stores. The sensors are able to detect when product on shelving are low, and then tell loaders to load shelving robots which can then restock shelves and have sensors verify the shelves are full, and then the server updates its inventory logs across multiple databases. Secondly, sensors are able to detect when actual inventory is low and then accurately predict when we will run out and when we need to place a shipment for product and then place an order for delivery. Thirdly, sensors are able to detect when an actual delivery has arrived, safely unload the product from the delivery truck, sort and organize the product, place RFID tags on product and configure them, and then recheck shelving and restock where necessary. Lastly, we also have a customer interaction in the use case diagram which shows that they are able to report low stock. However, this is used only to assist in helping detect faulty cameras and sensors since it is all autonomous, after the report we communicate with the server to check for faults and then if there is none do nothing because we already know the shelving is low and has not gone under the threshold yet to be reshelved, but if we do have a fault we then communicate with HR system and run it through there.

**Business process diagram:**

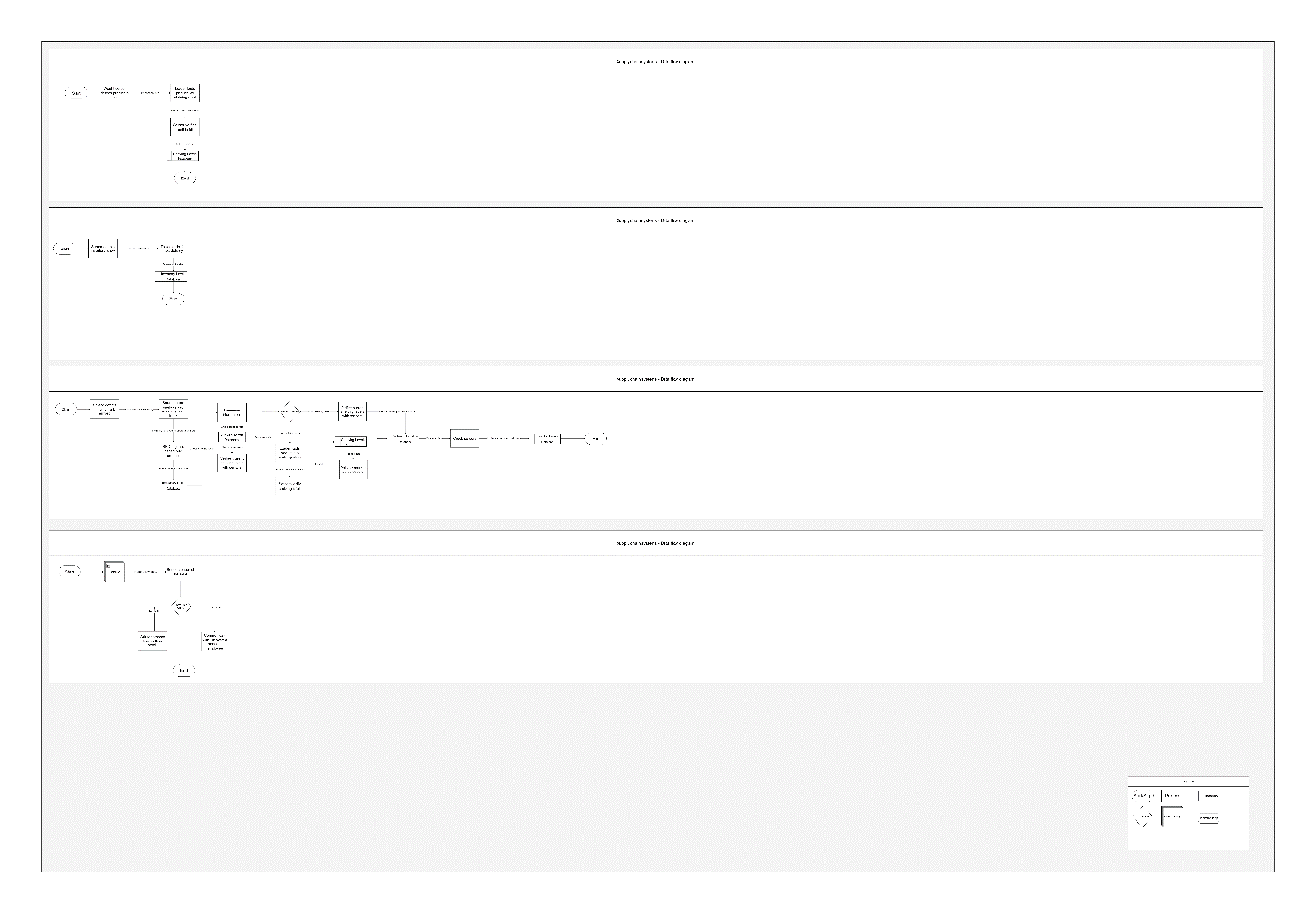
****

(Image is available in folder business process diagram for better viewing)

Brief discussion:

The supply chain business process diagram is also oversimplified, however, I believe, as well as all of the other diagrams, I have made it so that it is able to be understood easily without being in much more complexity into the system. The business process diagram shows interactions between the server, sensors, loaders, shelving robots, as well as the customers. It shows relatively easily the processes taking place when something happens such as a sensor detecting low product on a shelf, the server verifying this data, telling a loader to load a shelving robot, it then doing this, and then the shelving robot restocking the shelves and allowing the sensors to verify this and then the server updating its inventory. It also shows detecting overall inventory low and placing delivery orders, detecting delivery trucks and the process taking place, as well as the customers reporting low stock on shelves.

**Data flow diagram:**

****

(Image is available in folder data flow diagram for better viewing)

Brief discussion:

The data flow diagram for the supply chain system is also relatively simply, however, being broken into multiple parts to help visualization of the different parts of the system. It is able to show what databases and entities, such as the customer, are having influence upon the data of the system. Some databases being used are shelving level databases, delivery information databases, and inventory level databases so that we are able to verify everything for accuracy as well as so there is a log of everything happening in the store. These systems allow for the store to operate smoothly and always have product when they are needed and ensure the efficiency of everything running.

I pledge that on all academic work that I submit, I will neither give nor receive unauthorized aid, nor will I present another person's work as my own.

Dalton Murray