W200 – Spring 2020 – Sat. 8 AM
Professor Gerry Benoit
Samuel Gomez
February 29, 2020

Project 1 Proposal

The goal of Project 1 is to create a warehouse management system that sends and receives product to and from various production systems. At a high-level, a warehouse management system tracks product throughout the production lifecycle. The purpose of this project is to create an object-oriented production environment and WMS in python. Emphasis will be placed on how to arrange bins in the cold storage as well as follow a first in first out (FIFO) method from the receiving line to the wash line production processes.

User Interaction

Before describing the user interaction, it is helpful to understand the receiving and production processes. At the very start of the process, Tasteful Selections receives trailers mixed with 50,000lbs of baby potatoes and dirt/foreign material. Once trailers arrive on site, they are stored in trailer locations until processed by the Receiving Line. The receiving line consumes trailers by the variety in bulk and outputs sized product by continuously filling bins. There are eight size profiles produced during the sizing process.

After a bin is created, it is moved to a cold storage where it is held at 38 degrees for a few weeks before utilized as a wash line input. The scope of this project focuses on the inputs and outputs to and from the receiving line and wash line as well as the tote storage locations.

The program will start by allowing the user to initialize a factory (e.g. amount of totes, cold storages, a receiving line, and a wash line). The user will have the ability to create orders which will dictate production inputs and output destinations. Totes will be generated by receiving line orders and sent to the their storage destination. The cold storage class will organize the product in an aisle, columns, and row format. It will also allow the user to view the capacity of the room and the contents of each bin if requested. The program will incorporate certain rules—such as FIFO, etc.—for storing bins into the cold storage and consuming bins from the cold storage to the wash line. A random function will be used to create the bins produced by the receiving line based on the order. If time permits, the program will generate a weighted average consistent with expected size profiles.

Classes:

1) Factory – processing facility
    a) Name
    b) Location
    c) Number of cold storages
    d) One receiving line
    e) One wash line

2) Tote – Sized product held in a bin.
   a) Attributes
      i) Bin Id
      ii) Variety
      iii) Size
      iv) Weight
      v) Time stamp
      vi) Lot
   b) Methods
      i) Multiple getter and setter functions for the user to interact with.

3) Receiving Line Order – Production line that sorts unsized product into sized product.
   a) Attributes
      i) Product variety
      ii) Quantity in pounds
      iii) Storage destination
      iv) Order number
      v) Date
      vi) Time
   b) Input
      i) Unsized product measured in pounds.
   c) Output
      i) Bins that weigh 1200 – 1250 pounds
   d) Methods
      i) Execute_a – Place totes in Cold Storage A
      ii) Execute_b – Place totes in Cold Storage B
      iii) Execute_c – Place totes in Cold Storage C
      iv) Execute_d – Place totes in Cold Storage D

4) Cold Storage – Refrigerated storage room to hold product before it is consumed by the
   wash line.
   a) Attributes
      i) Name
      ii) Number of bin locations
      iii) Bin locations
      iv) Temperature
   b) Methods
      i) Get_location – find next available location
      ii) Display – display contents of the cold storage
      iii) Display_size – display cold storage contents of a certain size
      iv) Display_variety – display cold storage contents of a certain variety
      v) Display_variety_size – display cold storage contents of a cetain variety and size
      vi) Get_tote – return tote using FIFO method

5) Wash Line Order
    a) Attributes
        i) Order number
        ii) Variety
        iii) Size
        iv) Bin Quantity
    b) Methods
        i) Execute – remove totes from the cold storage to the wash line
        ii) Represent

Flow Chart

Receiving Line Order

Attributes
- Class
  - Order count
- Instance
  - Product variety
  - Quantity in pounds
  - Storage destination
  - Order number
  - Date
  - Times

Methods
- Execute_a
- Execute_b
- Execute_c
- Execute_d
- Represent

Tote

Attributes
- Class
  - Tote count
- Instance
  - Tote ID - hidden
  - Variety
  - Size
  - Weight
  - Location
  - Date/Time

Methods
- Multiple getter and setter functions for the user to interact with the attributes if need be
- Represent

Cold Storage

Attributes
- Instance
  - Name
  - Aisle
  - Column
  - Row

Methods
- Get_location
- Display
- Display_size
- Display_variety
- Display_variety_size
- Get_tote

Wash Line Order

Attributes
- Class
  - Count
- Instance
  - Variety
  - Size
  - Totes

Methods
- Execute
- Represent