ESE 499 Capstone Design Project Final Report
Submitted to Professor Trobaugh and the Department of Electrical and Systems Engineering

Chelsea Branson (Systems Engineering, chelseabranson@wustl.edu)
Matthew Stern (Systems Engineering, matthew.stern@wustl.edu)

Advisors:
Professor Zachary Feinstein (zfeinstein@wustl.edu)
Professor Jason Trobaugh (jasont@wustl.edu)
Craig Goldford, Director of Operations (craig.goldford@OperationFoodSearch.org)
Julie Mann, Finance Manager (julie.mann@OperationFoodSearch.org)
Julia Fuller, Agency Relations Manager (julia.fuller@OperationFoodSearch.org)

Submitted April 30th 2017

Inventory Management and Distribution Decisioning for Operation Food Search
September 2016 - April 2017
# Table of Contents

**Abstract**  
2

**Introduction**  
3  
  - Background Information  
  - Problem Statement  
  - Aims and Objectives

**Methods**  
5

**Data**  
7

**Results**  
11

**Discussion**  
12

**Conclusions**  
14

**Deliverables**  
14  
  - Labels  
    - Nonperishable Inventory Manager  
    - Perishable Inventory Manager  
    - Distribution Decisioning Tool

**Schedule & Timeline**  
22

**Appendix**  
24  
  - Inventory Entry Macro  
  - Inventory Removal Macro  
  - Inventory Transfer Macro  
  - Inventory Override Entry Macro  
  - Agency Pickup Macro
Abstract

Operation Food Search (OFS) is a St. Louis nonprofit that connects over 330 food pantries and charities to goods donated by local businesses, food pantries, and the community. Currently, the operations of the business are only monitored by a human walkthrough each day, inventory is not recorded, and there is no record of distribution. To modernize operations, we plan to develop a functional inventory management system on Excel that will allow a real-time view of the entire warehouse. Next, we want to help them collect data on the distribution of goods to agencies with a simple form. Then, we plan to use the data collected in distribution to create an integer programming problem to decide which items from the warehouse should be pulled for the next day of distribution. Using Excel Solver, we will construct a problem where the constraints are based on the agencies arriving on any particular day. This should save OFS a significant amount of time, allow them to serve large agencies more effectively, and will help them use more inventory before it expires.
Introduction

Background Information

Since 1981, Operation Food Search (OFS) has been accepting donations from local grocery stores, restaurants, food manufacturers, and the community to distribute to those in need. Feeding 200,000 a month and distributing over $35 million of food in a year, OFS plays a vital role in serving the hungry in the St. Louis bi-state area. Most donations from stores are taken directly to agencies, however, large-scale donations are sent to their warehouse in Overland. This warehouse was just opened in late 2016 and offers double the square footage as their previous warehouse. Additionally, there is now a walk in cooler and freezer that agencies can pick goods from directly.

In this new warehouse, OFS is utilizing vertical space for the first time. Because the shelving can hold 3 pallets vertically, it is much harder to remember when any of the 504 shelves were stocked. The shelving lacks definitive sections, labels, or organization. Since OFS uses this warehouse for multiple functions, organization is necessary. Large scale donations, food drive donations, food drive supplies, programming supplies, and the backpack program are all operated out of the same warehouse.

Each weekend during the school year, the Backpack Program sends 4,000 backpacks of food home with children of struggling families. This program is large but has a definitive flow of goods in and out because the demand is constant and the supply is under control of the OFS staff. However, the other donations that arrive each day are not so simple. Every day brings something new, whether it’s a truckload of Lysol wipes or produce from the Schnucks distribution facility. After the goods fill the warehouse, distributing them is the next task.

The 330 agencies served by OFS are unique in their selection of donors and programs. Donations from the OFS warehouse are only one piece of the food supply that feeds the people they serve. Population served varies from agency to agency, some serve ten times more people than others. Despite that fact, all agencies receive the same amount of food for any given pickup. When agencies come to pick up food, no information about their preferences or pickup are recorded. Many agencies ask for a summary of the support they have received from OFS through the year, but that information is not currently known.

OFS has many operational tasks to complete and has no particular strategy for completing them. Recently, they looked at inventory management software on the market but have been unable to purchase expensive commercial products with their nonprofit budget. Any aid in the efficiency and efficacy of OFS operations directly impacts those in need. By improving operations, food can be distributed while it is the freshest, inventory loss can be minimized, and customizing to agency things can free up more space for donations.

Problem Statement

In its current form, the OFS warehouse relies on one person to know the contents of 504 shelf positions, 60 overflow positions, the cooler, and the freezer. This person has to walk
around and figure out what is in the inventory before deciding what items to pull for the agencies arriving that day.

Because that process is so intensive, the current distribution protocol is to give each agency the same amount of food. Some agencies serve ten times other agencies and would benefit immensely from the ability to take more items from Operation Food Search. Additionally, some smaller agencies rely too heavily on OFS support and should possibly receive fewer items so they can create stronger bonds with other resources in the community. At times, OFS has a surplus of inventory and no good protocol to distribute it. Some donations are then discarded before agencies even know that it is in stock.

Despite the fact that each agency picks up similar quantities each day, there is no distribution record. Some agencies desire to know how much support they have received through OFS, but OFS does not have any data to answer those questions. Overall donations are recorded, but there is no way to tell which parts of the community receive the most help. Data would be beneficial for all of the agencies and OFS.

Aims and Objectives

With this project, our aim is to increase the efficiency of operations. Weekly meetings with leaders of OFS exposed pertinent operational obstacles. The issues of inventory management, data creation, decisioning, and data storage are addressed by our deliverables and protocols. By improving these operational aspects, we will increase the capacity for donations, the freshness of food delivered, and decrease waste.

To address inventory management, we developed an organization protocol for the warehouse and an inventory management system to keep track of the contents. The organization procedure established a name for each pallet position and a color code for each section. Our inventory management tool allows OFS to add, remove, or transfer pallets into sections. There is an override function if the warehouse experiences a surplus in inventory and needs to store pallets in unconventional locations. Additionally, the warehouse contents are summarized and easily navigable in a convenient table. This tool is utilized as frequently as a forklift--any move results in use of the system.

Another frequently utilized tool is our data collection form. This form is completed each time an agency picks up items and saves the submission in a separate workbook. This allows agencies and OFS to have data on each pickup from the warehouse and to observe trends between visits of agencies.

After observing this initial implementation, we expanded to the cooler and freezer. We established a separate inventory manager for the perishable items as well as a decisioning tool for these items. This tool prompts the user for a few pieces of information, evaluates the contents of the cooler and freezer, and then uses integer programming to choose the most ideal pallets to be distributed for the day.
Methods

In the development of our deliverables, inventory theory was the key motivation and Excel was the most important software. All of our tools were built in Excel, since the licensing is already owned by OFS. We used the principles of inventory theory to build the tools in such a manner that discourages leaving items in the warehouse for long periods of time. This twofold effect aids in creating open space for more deliveries as well as increasing the freshness of the goods distributed.

Our first deliverable, the Inventory Management System, first required us to understand the warehouse. The non-perishable sector of 504 pallet positions houses non-food items such as household supplies and clothing, food drive donations, industrial-scale pallet donations, and the Backpack Program. Before developing the tool, we set up a naming convention and color-coded organization system. Then, we created labels for each pallet position. With a unique name for each pallet position, we were then able to start working on our inventory management system. This tool utilizes excel macros and user-friendly input sheets to add items, remove items, transfer items, and override the organizational category. One sheet displays all of the contents of the warehouse in a simple, sorted format. These all complement the master view of the warehouse, which shows all 504 pallet positions and 60 overflow positions formatted just like the actual warehouse.

Next, we developed another inventory management system for the cooler and freezer. This tool has all the same features as the other one, but has been developed with a decisioning tool in mind. Between the cooler and freezer, there is a maximum of 150 pallet positions, including floor space. While this number may seem large, OFS can easily run out of space. In the summer OFS coordinates 100,000 meals for Meals on Wheels in the area, which is a great strain on their cooler space. Between Meals on Wheels and the spectrum of cold storage space at the agencies, deciding what to remove from the cooler and freezer can be quite the challenge. To aid with this decision, we have developed a model to optimize the removal of pallets from the cooler and freezer.
The model has been constructed such that the contents of the cooler and freezer are the decision variables $x_i$. Generally, whole pallets are removed, so we have made all $x_i$ binary, as seen in constraint 8. The objective function has been crafted to pull the oldest pallets out of the inventory first, which is the primary concern for this perishable stock. In constraints 1 and 2, we define a maximum shelf life for items in the cooler and freezer. Refrigerated items should be removed if they have spent more than $f$ days in the inventory and freezer items should be removed after $d$ days have passed. Supply of perishables experiences a lot of variance and occasionally a large amount of space must be cleared for a new delivery. Constraint 4 allows for the user to determine $n_j$, the amount of empty space needed for the next delivery. All of the quantities in this model are considered in cases and pallets. Cases have a wide variance in size, so constraint 3 limits the agency capacity to 90% of the recorded size. In the event that the cases in the inventory are larger than normal, that cutoff should account for the size discrepancy.
Variables Used

\[ x_i = \begin{cases} 1 & \text{if contents at position } i \text{ are to be removed, } 0 & \text{otherwise} \\ a_i = \text{Number of days units in position } i \text{ have been in the warehouse} \\ f = \text{Max number of days units can remain in the freezer} \\ d = \text{Max number of days units can remain in the cooler} \\ r_i = \begin{cases} 1 & \text{if position } i \text{ is in the freezer, } 0 & \text{otherwise} \\ c_i = \text{Number of units in position } i \\ m = \text{Maximum capacity for today's agencies} \\ s_{ij} = \begin{cases} 1 & \text{if position } i \text{ is in section } j, \ 0 & \text{otherwise} \\ v_{ik} = \begin{cases} 1 & \text{if contents at position } i \text{ are of category } k, \ 0 & \text{otherwise} \\ u_j = \text{Number of spaces currently free in section } j \\ n_j = \text{Number of spaces needed in section } j \\ b_k = \text{Agency size limit for category } k \\ y_k = \text{Minimum needed positions for items in category } k \text{ in the warehouse} \\
\end{cases}
\]

Figure 2: Freezer/Cooler Removal Decisioning Model Variable Descriptions

Though removing items from the inventory is often the priority, OFS must carefully distribute their inventory. Keeping a baseline of inventory can be a concern with such an irregular supply. It would be unwise to distribute all the meat in one day if there are no meat donations for the rest of the week. In constraint 6, we allow OFS to control baselines for each category of food. The same logic applies to distribution, as OFS strives to provide complete nutrition. This OFS can also control how many items from each category leaves, so that there is a balance between meat, dairy, produce, and frozen food for distribution. In our implementation, we have made these parameters accessible to OFS. Thus, the problem can be changed as needed to satisfy dynamic demands.

Constraint Descriptions

1. All freezer items must be removed after \( f \) days have passed since entry
2. All cooler items must be removed after \( d \) days have passed since entry
3. The number of cases removed must not exceed 90 percent of the agencies’ available space
4. The number of spaces needed in each section are made available
5. The total number of units leaving for each category does not exceed preset limits
6. The number of spaces remaining for each category must exceed preset minimum
7. Any space with zero units cannot be removed
8. Any space is either removed or not removed

Figure 3: Freezer/Cooler Removal Decisioning Model Constraint Descriptions

Data

A tremendous portion of this effort to improve operations concerns data. For the decisioning tool to be successful, it requires information about each agency that picks up food from the warehouse. Agencies that pick up on Tuesdays were recorded by the Agency Relations Manager earlier on in the project. A sample of that data can be seen below in Figure 5. We used the capacities of their coolers and freezers as well as their pickup capacity to
construct values for our constraints. One limitation is that we only received this data for Tuesday, so values for other pickup days were extrapolated from the data collected from mid-March through mid-April. This data was collected using the Agency Pickup Form that is filled in with case quantities of each category when an agency picks items up from the warehouse. A portion of this data is visible in Figure 6. Upon our calculations the pickup form had been completed over 320 times. We were able to take both data sources and combine them to arrive at estimates for pickup capacities of each pickup day.

Figure 4: Agency Pickup Entry Form
Data collected by these forms is useful for both us and Operation Food Search. Agencies have contacted the Agency Relations Manager many times in the past to inquire about the support they received. The Agency Pickup Form ports data to a Distribution Data sheet each time it is filled out. On the Distribution Data sheet, there is a pivot table that offers an easy, dynamic view of the goods distributed to each agency. We spent a meeting teaching Julia how to operate pivot tables to find the answers she looks for. A sample of the data in the pivot table can be seen in Figure 7. This information will help OFS keep more robust records and serve agencies more effectively.
<table>
<thead>
<tr>
<th>Agency</th>
<th>Sum of Sum Total Cases</th>
<th>Sum of Sum Frozen</th>
<th>Sum of Sum Food Cases</th>
<th>Count of Sum Nonfood Cases</th>
<th>Sum of Sum Refrigerated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alton Christian Food Pantry SVDP</td>
<td>192</td>
<td>9</td>
<td>112</td>
<td>3</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>3/16/17 8:06</td>
<td>55</td>
<td>1</td>
<td>35</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3/16/17 8:30</td>
<td>60</td>
<td>1</td>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4/6/17 8:35</td>
<td>77</td>
<td>1</td>
<td>47</td>
<td>1</td>
</tr>
<tr>
<td>Agape SDA Pantry</td>
<td>180</td>
<td>3</td>
<td>100</td>
<td>3</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>3/16/17 10:17</td>
<td>52</td>
<td>1</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3/20/17 10:56</td>
<td>55</td>
<td>1</td>
<td>25</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4/6/17 10:44</td>
<td>73</td>
<td>1</td>
<td>43</td>
<td>1</td>
</tr>
<tr>
<td>Alton Guardian Angel Pantry</td>
<td>185</td>
<td>4</td>
<td>92</td>
<td>4</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>3/15/17 9:34</td>
<td>37</td>
<td>1</td>
<td>27</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3/22/17 9:31</td>
<td>51</td>
<td>1</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3/29/17 9:58</td>
<td>56</td>
<td>1</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4/6/17 9:49</td>
<td>21</td>
<td>1</td>
<td>17</td>
<td>1</td>
</tr>
<tr>
<td>Alton United Hands</td>
<td>79</td>
<td>1</td>
<td>35</td>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>3/21/17 7:38</td>
<td>79</td>
<td>1</td>
<td>35</td>
<td>1</td>
</tr>
<tr>
<td>Arizon Health (residential meal program)</td>
<td>240</td>
<td>4</td>
<td>168</td>
<td>4</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>3/13/17 10:03</td>
<td>59</td>
<td>1</td>
<td>49</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3/17/17 10:35</td>
<td>49</td>
<td>1</td>
<td>33</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4/3/17 10:39</td>
<td>78</td>
<td>1</td>
<td>48</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4/10/17 11:16</td>
<td>54</td>
<td>1</td>
<td>38</td>
<td>1</td>
</tr>
<tr>
<td>Arnold Food Pantry</td>
<td>248</td>
<td>3</td>
<td>146</td>
<td>5</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>3/14/17 7:39</td>
<td>35</td>
<td>1</td>
<td>25</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3/21/17 7:40</td>
<td>96</td>
<td>0</td>
<td>26</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3/28/17 7:39</td>
<td>34</td>
<td>0</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4/4/17 7:36</td>
<td>83</td>
<td>1</td>
<td>59</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4/11/17 7:40</td>
<td>40</td>
<td>1</td>
<td>26</td>
<td>1</td>
</tr>
<tr>
<td>Baptist Ministries South Center</td>
<td>256</td>
<td>4</td>
<td>139</td>
<td>4</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>3/16/17 7:37</td>
<td>49</td>
<td>1</td>
<td>37</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3/16/17 7:36</td>
<td>79</td>
<td>1</td>
<td>34</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>3/21/17 7:42</td>
<td>53</td>
<td>1</td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4/6/17 7:41</td>
<td>75</td>
<td>1</td>
<td>45</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 7: Agency Pickup Form Pivot Table
Results

OFS was able to implement some of our tools and now has more efficient operations! Before our inventory manager was implemented on March 10th, the labels we produced were placed on all 504 pallet positions. The same week, our Agency Pickup Form was put into action and began collecting data. We established a naming convention for the cooler and freezer and were able to implement an inventory manager for the perishable items as well. Finally, we were able to create a preliminary model of a decisioning tool for their perishable items. All of these changes added benefit to OFS operations.

We observed our first result as the labels were applied in early March. By designating specific sections, the warehouse has become more organized. Labels clearly color code and name each pallet position. Finding inventory is simpler with designated sections and created a foundation for our Inventory Manager. With our nonperishable inventory manager, all items are conveniently tracked. User-friendly interfaces guide the addition, removal, and transfer of items in the system. The sections are enforced but can be overridden in times of severe surplus in the warehouse. Our view of the warehouse conveniently highlights any items that have been in the warehouse for 30 days, flagging them for removal. One feature of the tool is a pivot table that shows all of the items in each category, organized to where the oldest items appear at the top of the list. The nonperishable inventory tool has added a lot of benefit to OFS.

Though our added value to OFS is difficult to qualify, their Director of Operations, Craig Goldford, has informed us of the changes they have experienced thus far. Firstly, overall warehouse organization has improved with our work. Creating a system and communicating that to the employees has been a strong driver of change. Inventory placement and the use of their racks has been prioritized. Following the new protocol has improved the accuracy of the First In, First Out (FIFO) inventory strategy. With the nonperishable inventory manager, the inventory for the backpack program can be assessed more easily. New inventory can now be reordered in advance of running out. Another area impacted by our work is the sorting and distribution of Food Drive food. With the new system, it is much easier to schedule the right number of volunteers to sort items donated to food drives. The bins that are sorted are also now managed under a FIFO system. Improving the management of these tasks is also associated with a cost savings (C. Goldford, April 27, 2017). These changes seem small at first, but each day, delivery, and pickup makes a large difference when over $35 million of food is distributed by OFS in a given year.

Each time an agency picks up food, the Agency Pickup Form is completed. That data is ported to the Distribution Data sheet, where OFS staff can quickly analyze distribution. Agencies often inquire about the donations they receive, as some grant programs will match the contributions from food banks like OFS. Because accurate pickup data is now available, the agencies can now have accurate data for grant application. This gives larger St. Louis agencies access to life-changing grants, donors, and other funding sources (J. Fuller, April 26, 2017).
Discussion

When generating our mathematical formulation for our decisioning tool, we wanted to ensure that the tool was simple enough for the staff at OFS to use on a regular basis and to ensure that the tool returned a response for any feasible scenario, yet robust enough to provide actual insights into what should be removed to fit their day-to-day demand and necessities (rather than simply going down the list of what has been in the warehouse the longest and removing it). Ultimately, we decided that the most important criteria to consider were those which allowed the maximum amount of cases to exit the warehouse while ensuring minimal waste and maximal variation between food types across agencies (rather than simply giving only produce or only dairy for a given visit). To ensure optimal usage of storage space, we decided to make our formulation a binary integer problem rather than a standard integer problem, since we wanted to ensure that no positions were left storing single cases of items, when the locations could be better utilized for storing multiple cases.

Despite ensuring that our model should run successfully for nearly all feasible scenarios, we did observe a variance in runtimes depending on the conditions set and the current contents of the warehouse. This can likely be attributed to certain constraints becoming seemingly redundant depending on the contents and quantity of each category in the warehouse, and the number of valid positions (meaning positions that are not automatically determined by constraints 1, 2, and 7). For example, constraint 6 is nearly always satisfied for scenarios with a large number of valid positions across all categories, since the system can only remove a certain number of positions due to constraint 3. Likewise, constraint 5 is nearly always satisfied for scenarios with a small number of valid positions across all categories, since it is nearly impossible to surpass the individual category limits. However, when the situations are reversed, each of these constraints make the problem more difficult to resolve due to the increasing number of subproblems. From our testing, we have observed runtimes as quick as under 10 seconds, and as long as over 5 minutes. However, we have not encountered scenarios where the system fails to run, since the only scenario that will break the system is if there simply is not enough available in the warehouse to distribute for each category, which hopefully is a scenario which will never be encountered.

<table>
<thead>
<tr>
<th>Valid Positions</th>
<th>Runtimes (s)</th>
<th>Iterations</th>
<th>Subproblems</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>12.332886</td>
<td>5</td>
<td>1454</td>
</tr>
<tr>
<td>65</td>
<td>29.147506</td>
<td>3</td>
<td>5670</td>
</tr>
<tr>
<td>70</td>
<td>80.601591</td>
<td>2</td>
<td>8184</td>
</tr>
<tr>
<td>75</td>
<td>29.787847</td>
<td>3</td>
<td>3108</td>
</tr>
</tbody>
</table>
While we chose to make our decision model based around a binary integer solution, the problem could theoretically be relaxed to an LP solution with minimal constraint changes. The decision variables $x_i$ would now represent the number of cases to remove from position $i$ rather than whether or not to remove the cases. To ensure that certain equations still work, we would need to create a set of helper variables $t_i$, which would have corresponding constraints to ensure that $t_i$ equals 1 if a nonzero amount of items is removed from position $i$ and equals 0 otherwise. Constraints 1 and 2 would have to be modified such that if the items in position $i$ have exceeded their maximum date, $x_i$ must equal $c_i$. For constraints 3 and 5, we would simply need to remove the $c_i$ from the equation, since the $x_i$ already takes care of the number of crates. For constraint 4, we would replace the summation of nonzero $x_i$ in the proper with a summation of the number of positions in the proper section where $x_i$ equals $c_i$. Constraint 6 would likely have to be modified such that a predetermined number of crates of each category remain in the warehouse rather than a number of positions. Constraint 7 would remain the same, and constraint 8 would be modified such that $0 <= x_i <= c_i$. While this formulation is possible to create and would return a valid result when run, it would likely be infeasible for OFS to manage such specific quantities, especially since many of the crate counts for each position are approximations more than hard counts.

Figure 8: Decisioning Tool Sample Run Time Analysis Data

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>94.519612</td>
<td>1</td>
<td>10730</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>86.128726</td>
<td>4</td>
<td>10282</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>13.486543</td>
<td>5</td>
<td>1002</td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>92.542414</td>
<td>1</td>
<td>12750</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>28.167573</td>
<td>9</td>
<td>1816</td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>175.19344</td>
<td>4</td>
<td>19988</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>56.091876</td>
<td>5</td>
<td>5050</td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>61.477432</td>
<td>1</td>
<td>7732</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>19.189844</td>
<td>5</td>
<td>1670</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>25.2245</td>
<td>6</td>
<td>2254</td>
<td></td>
</tr>
</tbody>
</table>
Working with OFS has been incredibly rewarding. This positive experience has caused us to ponder future projects. As this course closes, we will be establishing a formalized partnership to foster future endeavors to improve OFS. Personnel scheduling, vehicle routing, and scheduling backpack program orders are potential opportunities we have identified. The Gephardt Institute at Washington University aids in formalizing partnerships between students and the community. Beth Martin is a part of the Gephardt Institute and supports these engagements. She introduced us to the resources to solidify the connection between OFS and engineering design. The future research of this project will then be completed by qualified students in the Capstone Design Project course.

By the creation of our inventory managers, decision tools, and a partnership with OFS, we have fulfilled our promises to Operation Food Search. OFS has user-friendly tools that allow them to operate more efficiently. Constructing these tools required everyone to evaluate the operational procedures of OFS. New insights arose from taking the time to assess their procedures. The happiness of our client is a great strength of this project.

Conclusions

As a result of our work with OFS, they now have tools to manage their inventory, collect data, and distribute their stock. Since OFS did not have a means of doing any of these before, measuring our impact is a little difficult. However, OFS has implemented our inventory manager and data collection tool. Though we have established a decisioning tool for them, they are currently not ready for the implementation of this decision-maker. However, we hope to establish a lasting partnership with OFS where another round of students can come in and use data collected long-term to make a more informed decisioning tool. If that tool is not what works best for OFS, we hope that these students will tackle another issue like personnel scheduling, applying inventory theory to develop a calendar for ordering food for the Backpack Program, or another problem that we are currently unaware of.

Though our projects for OFS have been beneficial to both the Washington University and the food bank, this implies that nonprofits do not have access to the operational tools and knowledge they need. Consulting fees and software licensing fees for implementing and customizing operational tools are significant prohibitive costs to nonprofit businesses. Without our help, OFS would have had to fundraise for years to purchase an inventory manager. Future partnerships with nonprofits and charities in the area could offer students unparalleled real-world experience while contributing to the local community. Many of these projects would be at an appropriate level for student knowledge because of the prohibitive costs of operational tools.

Deliverables

Labels

Labels were ordered for all areas of the warehouse. These identify the pallet positions in the nonperishable, overflow, cooler, and freezer sections of the warehouse.
Figure 9: Label Proofs from Hi/Tec
Nonperishable Inventory Manager

The Nonperishable Inventory Manager allows for the addition, removal, and transfer of items into their sections. Warehouse sections can be overridden in times of surplus. Items over 30 days old are highlighted yellow. Views include a master sheet, pivot table of the contents, and a sheet for entry, removal, override, and transfer. Raw data is visible to both OFS and the team, but is locked for editing.

Figure 10: Labels in Use in the Warehouse (multiple photos)

Figure 11: Nonperishable Inventory Master Sheet View
Figure 12: Inventory Entry and Override Entry Forms

Figure 13: Inventory Entry Removal and Transfer Forms

Figure 14: Nonperishable Inventory Maximum Category Capacity and Valuation
Perishable Inventory Manager

The Perishable Inventory Manager also allows for entry, removal, and transfer of goods. Override is not necessary for the cooler and freezer because there are not isolated sections.
**Figure 16: Perishable Inventory Master Sheet View**

<table>
<thead>
<tr>
<th>Pallet Position</th>
<th>Section (Current)</th>
<th>Category</th>
<th># Of Units</th>
<th>Arrival Date</th>
<th>Override Section</th>
<th>Max Quantity</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1K3</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1K2</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1K1</td>
<td>Freezer</td>
<td>Deer Meat</td>
<td>30</td>
<td>4/20/17</td>
<td>None</td>
<td>30</td>
<td>Cases</td>
</tr>
<tr>
<td>F1K0</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1J3</td>
<td>Freezer</td>
<td>Meat</td>
<td>50</td>
<td>4/26/17</td>
<td>None</td>
<td>50</td>
<td>Cases</td>
</tr>
<tr>
<td>F1J2</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1J1</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1I0</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1I1</td>
<td>Freezer</td>
<td>Meat</td>
<td>40</td>
<td>4/5/17</td>
<td>None</td>
<td>50</td>
<td>Cases</td>
</tr>
<tr>
<td>F1I0</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1H3</td>
<td>Freezer</td>
<td>Frozen Food</td>
<td>50</td>
<td>4/21/17</td>
<td>None</td>
<td>50</td>
<td>Cases</td>
</tr>
<tr>
<td>F1H2</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1H1</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1H0</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1G3</td>
<td>Freezer</td>
<td>Frozen Food</td>
<td>50</td>
<td>4/16/17</td>
<td>None</td>
<td>50</td>
<td>Cases</td>
</tr>
<tr>
<td>F1G2</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1G1</td>
<td>Freezer</td>
<td>Deer Meat</td>
<td>30</td>
<td>4/10/17</td>
<td>None</td>
<td>30</td>
<td>Cases</td>
</tr>
<tr>
<td>F1G0</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1F3</td>
<td>Freezer</td>
<td>Meat</td>
<td>35</td>
<td>4/4/17</td>
<td>None</td>
<td>50</td>
<td>Cases</td>
</tr>
<tr>
<td>F1F2</td>
<td>Freezer</td>
<td>Meat</td>
<td>40</td>
<td>4/22/17</td>
<td>None</td>
<td>50</td>
<td>Cases</td>
</tr>
<tr>
<td>F1F1</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1F0</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1E3</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1E2</td>
<td>Freezer</td>
<td>Deer Meat</td>
<td>30</td>
<td>4/4/17</td>
<td>None</td>
<td>30</td>
<td>Cases</td>
</tr>
<tr>
<td>F1E1</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1E0</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1D3</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1D2</td>
<td>Freezer</td>
<td>Meal</td>
<td>49</td>
<td>4/15/17</td>
<td>None</td>
<td>50</td>
<td>Cases</td>
</tr>
<tr>
<td>F1D1</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1D0</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1C3</td>
<td>Freezer</td>
<td>Meat</td>
<td>37</td>
<td>4/21/17</td>
<td>None</td>
<td>50</td>
<td>Cases</td>
</tr>
<tr>
<td>F1C2</td>
<td>Freezer</td>
<td>Frozen Food</td>
<td>50</td>
<td>4/26/17</td>
<td>None</td>
<td>50</td>
<td>Cases</td>
</tr>
<tr>
<td>F1C1</td>
<td>Freezer</td>
<td>Meal</td>
<td>50</td>
<td>4/12/17</td>
<td>None</td>
<td>50</td>
<td>Cases</td>
</tr>
<tr>
<td>F1C0</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1B3</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1B2</td>
<td>Freezer</td>
<td>Deer Meat</td>
<td>20</td>
<td>4/23/17</td>
<td>None</td>
<td>30</td>
<td>Cases</td>
</tr>
<tr>
<td>F1B1</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1B0</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1A3</td>
<td>Freezer</td>
<td>Meal</td>
<td>50</td>
<td>4/5/17</td>
<td>None</td>
<td>50</td>
<td>Cases</td>
</tr>
<tr>
<td>F1A2</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1A1</td>
<td>Freezer</td>
<td>None</td>
<td>0</td>
<td>None</td>
<td>None</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>F1A0</td>
<td>Freezer</td>
<td>Meal</td>
<td>45</td>
<td>4/22/17</td>
<td>None</td>
<td>50</td>
<td>Cases</td>
</tr>
</tbody>
</table>

**Figure 17: Perishable Inventory Raw Data**

**Distribution Decisioning Tool**

This tool allows for constraint values to be input by the user, then runs solver to determine which items to pull from the cooler and freezer. It is built into the Perishable Inventory Manager.
### Figure 18: Perishable Distribution Decisioning Tool Constraint/Decision Variable View

<table>
<thead>
<tr>
<th>Constraint/Decision Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max # of days a pallet can remain in the freezer</td>
<td>30</td>
</tr>
<tr>
<td>Max # of days a pallet can remain in the cooler</td>
<td>5</td>
</tr>
<tr>
<td>Total maximum capacity of all agencies today</td>
<td>900</td>
</tr>
<tr>
<td># of freezer spaces needed</td>
<td>10</td>
</tr>
<tr>
<td># of cooler spaces needed</td>
<td>10</td>
</tr>
<tr>
<td>Max # of outgoing units of frozen food</td>
<td>175</td>
</tr>
<tr>
<td>Max # of outgoing units of meat</td>
<td>225</td>
</tr>
<tr>
<td>Max # of outgoing units of deer meat</td>
<td>100</td>
</tr>
<tr>
<td>Max # of outgoing units of produce</td>
<td>300</td>
</tr>
<tr>
<td>Max # of outgoing units of dairy</td>
<td>250</td>
</tr>
<tr>
<td>Max # of outgoing units of RL milk</td>
<td>125</td>
</tr>
<tr>
<td>Max # of outgoing units of Pepsico</td>
<td>100</td>
</tr>
<tr>
<td>Minimum # of remaining frozen food positions</td>
<td>2</td>
</tr>
<tr>
<td>Minimum # of remaining meat positions</td>
<td>2</td>
</tr>
<tr>
<td>Minimum # of remaining deer meat positions</td>
<td>0</td>
</tr>
<tr>
<td>Minimum # of remaining produce positions</td>
<td>2</td>
</tr>
<tr>
<td>Minimum # of remaining dairy positions</td>
<td>2</td>
</tr>
<tr>
<td>Minimum # of remaining RL milk positions</td>
<td>2</td>
</tr>
<tr>
<td>Minimum # of remaining Pepsico positions</td>
<td>2</td>
</tr>
</tbody>
</table>

### Figure 19: Perishable Distribution Decisioning Tool Constraint Value Input Form
Schedule & Timeline

**Figure 20:** Project Outline Fall 2016

**Figure 21:** Project Outline Spring 2017

This project was completed over the full 2016-2017 academic year. We were able to establish a relationship with OFS and build intelligence on operations in the Fall semester as seen in **Figure 20**. In the Spring, we began to build and implement tools according to OFS needs, as seen in **Figure 21**. Through this process, Matt was the quantitative expert and Chelsea was the qualitative expert. For example, Matt leveraged his experience in VBA to build out the first macros, Chelsea learned some VBA along the way to build the one for the Agency Pickup Form. Matt championed the building of the inventory management systems, Chelsea
helped in developing the concept and making it both visually appealing and easy to use. Through the academic year, the work was balanced between the two.

Matt lead the work on certain sections of the project. This included the layout and macros for the non-perishable and perishable inventory managers. He put the mathematical formulation into excel solver and arranged the perishable inventory manager accordingly. Matt was quick to implement any changes needed by the OFS team and implemented every desired feature into the inventory managers.

Chelsea lead the effort on the Agency Pickup Form and the Distribution Data. She also created the part of the macro to report all inventory entries to a history sheet. She developed the primary ideas for the naming conventions and color codes in the warehouse and made the labels for each of the pallet positions. Much of the report writing was completed by her, as well as writing directions for the tools. She also adjusted the formatting of the tools to make them easier to read and more user-friendly.

Overall, we made a great team. Our workloads were balanced and we were able to reach out to each other for support. The mathematical formulation was a collaboration and involved a couple weeks of ideation and iteration. Weekly meetings with OFS and twice-weekly workdays ensured the completion of this project. Through much time and effort, this project stayed largely on schedule and we were able to have a great impact on OFS.
Appendix

Inventory Entry Macro

Sub InventoryEntry()

    Dim section As String
    Dim category As String
    Dim crates As Integer
    Dim position As String
    Dim todayis As Date
    Dim i As Integer
    section = Range("B3").Value
    category = Range("B5").Value
    crates = Range("B9").Value
    position = Range("B11").Value
    todayis = Now
    If section = vbNullString Then
        MsgBox "Please select a section."
        Exit Sub
    End If
    If category = vbNullString Then
        MsgBox "Please select a category."
        Exit Sub
    End If
    If crates = 0 Then
        MsgBox "Please select a number of units."
        Exit Sub
    End If
    If position = vbNullString Then
        MsgBox "Please select a category."
        Exit Sub
    End If
    MsgBox crates & " units of type " & category & " will be placed into position " & position & " in the " & section & " section."
    For i = 2 To 550
        If position = Worksheets("Raw Data").Cells(i, 1) Then
            Worksheets("Raw Data").Cells(i, 3).Value = category
            Worksheets("Raw Data").Cells(i, 4).Value = crates
            Worksheets("Raw Data").Cells(i, 5).Value = todayis
        End If
    Next i
    Application.ScreenUpdating = False
ChDir "V:"
Workbooks.Open Filename:="V:\EntryHistory.xlsx"
Cells(Application.ActiveCell.Row, 1).Select
ActiveCell.Value = todayis
Cells(Application.ActiveCell.Row, 2).Select
ActiveCell.Value = crates
Cells(Application.ActiveCell.Row, 3).Select
ActiveCell.Value = category
Cells(Application.ActiveCell.Row, 4).Select
ActiveCell.Value = section
ActiveWorkbook.Save
ActiveWindow.Close
Application.ScreenUpdating = True

Range("B3").Value = ""
Range("B5").Value = ""
Range("B9").Value = ""
Range("B11").Value = ""
ActiveWorkbook.Save

End Sub

Inventory Removal Macro

Sub InventoryRemoval()

Dim section As String
Dim category As String
Dim position As String
Dim max As Integer
Dim crates As Integer
Dim i As Integer
section = Range("B3").Value
category = Range("B5").Value
position = Range("B7").Value
crates = Range("B11").Value

If section = vbNullString Then
    MsgBox "Please select a section."
    Exit Sub
End If

End Sub
If category = vbNullString Then
    MsgBox "Please select a category."
    Exit Sub
End If
If position = vbNullString Then
    MsgBox "Please select a position."
    Exit Sub
End If
max = Range("B9").Value
If crates = 0 Then
    MsgBox "Please select a number of units."
    Exit Sub
End If
MsgBox crates & " units of " & category & " will be removed from position " & position & " in the " & section & " section."
For i = 2 To 550
    If position = Worksheets("Raw Data").Cells(i, 1) Then
        If max = crates Then
            Worksheets("Raw Data").Cells(i, 3).Value = "None"
            Worksheets("Raw Data").Cells(i, 4).Value = 0
            Worksheets("Raw Data").Cells(i, 5).Value = ""
            Worksheets("Raw Data").Cells(i, 6).Value = "None"
        Else
            Worksheets("Raw Data").Cells(i, 4).Value = max - crates
        End If
    End If
Next i
Range("B3").Value = ""
Range("B5").Value = ""
Range("B7").Value = ""
Range("B11").Value = ""
ActiveWorkbook.Save
End Sub

**Inventory Transfer Macro**

Sub InventoryTransfer()
    Dim oldPosition As String
    Dim oldCategory As String
    Dim crates As Integer
    Dim newPosition As String
Dim oldDate As Date
Dim i As Integer

oldPosition = Range("C5").Value
oldCategory = Range("C3").Value
newPosition = Range("C7").Value
If oldCategory = vbNullString Then
    MsgBox "Please select a category."
    Exit Sub
End If
If oldPosition = vbNullString Then
    MsgBox "Please select an original position."
    Exit Sub
End If
If newPosition = vbNullString Then
    MsgBox "Please select a new position."
    Exit Sub
End If

crates = Range("F5").Value
oldDate = Range("G5").Value
MsgBox "The " & oldCategory & " items will be moved from position " & oldPosition & " to position " & newPosition
For i = 2 To 550
    If newPosition = Worksheets("Raw Data").Cells(i, 1) Then
        Worksheets("Raw Data").Cells(i, 3).Value = oldCategory
        Worksheets("Raw Data").Cells(i, 4).Value = crates
        Worksheets("Raw Data").Cells(i, 5).Value = oldDate
    End If
    If oldPosition = Worksheets("Raw Data").Cells(i, 1) Then
        Worksheets("Raw Data").Cells(i, 3).Value = "None"
        Worksheets("Raw Data").Cells(i, 4).Value = 0
        Worksheets("Raw Data").Cells(i, 5).Value = ""
        Worksheets("Raw Data").Cells(i, 6).Value = "None"
    End If
Next i

Application.ScreenUpdating = False
Range("C3").Value = ""
Range("C5").Value = ""
Range("C7").Value = ""
ActiveWorkbook.Save
End Sub
Inventory Override Entry Macro

Sub OverrideEntry()
    Dim actualSection As String
    Dim category As String
    Dim placementSection As String
    Dim crates As Integer
    Dim position As String
    Dim todayis As Date
    Dim i As Integer
    actualSection = Range("B3").Value
    category = Range("B5").Value
    placementSection = Range("B7").Value
    crates = Range("B11").Value
    position = Range("B13").Value
    todayis = DateValue(Now)
    If actualSection = vbNullString Then
        MsgBox "Please select an Actual Section."
        Exit Sub
    End If
    If category = vbNullString Then
        MsgBox "Please select a category."
        Exit Sub
    End If
    If placementSection = vbNullString Then
        MsgBox "Please select a Placement Section."
        Exit Sub
    End If
    If actualSection = placementSection Then
        MsgBox "The placement section and actual section must differ. If you are placing an item into the correct section, please use the regular entry form."
        Exit Sub
    End If
    If crates = 0 Then
        MsgBox "Please select a number of units."
        Exit Sub
    End If
    If position = vbNullString Then
        MsgBox "Please select a category."
        Exit Sub
    End If
    MsgBox crates & " units of type " & category & " will be placed into position " & position & " in the " & placementSection & " section."
    For i = 2 To 550
End Sub
If position = Worksheets("Raw Data").Cells(i, 1) Then
    Worksheets("Raw Data").Cells(i, 3).Value = category
    Worksheets("Raw Data").Cells(i, 4).Value = crates
    Worksheets("Raw Data").Cells(i, 5).Value = todayis
    Worksheets("Raw Data").Cells(i, 6).Value = actualSection
End If
Next i

Application.ScreenUpdating = False

ChDir "V:"
Workbooks.Open Filename:="V:\EntryHistory.xlsx"
Cells(Application.ActiveCell.Row, 1).Select
ActiveCell.Value = todayis
Cells(Application.ActiveCell.Row, 2).Select
ActiveCell.Value = crates
Cells(Application.ActiveCell.Row, 3).Select
ActiveCell.Value = category
Cells(Application.ActiveCell.Row, 4).Select
ActiveCell.Value = actualSection
ActiveWorkbook.Save
ActiveWindow.Close
Application.ScreenUpdating = True

Range("B3").Value = ""
Range("B5").Value = ""
Range("B7").Value = ""
Range("B11").Value = ""
Range("B13").Value = ""

ActiveWorkbook.Save

End Sub

**Agency Pickup Macro**

Sub AgencyPickUp()

    Dim agency As String
    Dim bakery As Double
Dim beverage As Double
Dim cannedgood As Double
Dim dairy As Double
Dim deer As Double
Dim dry As Double
Dim foodservice As Double
Dim meat As Double
Dim mixed As Double
Dim produce As Double
Dim snacks As Double
Dim clothingunits As Double
Dim clothingcases As Double
Dim good360units As Double
Dim good360cases As Double
Dim household As Double
Dim personalhygiene As Double
Dim RLmilk As Double
Dim frozenfood As Double
Dim presstime As Date
' time and date of button press recorded
Dim i As Integer

cannedgood = Range("B7").Value
dry = Range("B9").Value
snacks = Range("B11").Value
beverage = Range("B13").Value
mixed = Range("B15").Value
frozenfood = Range("B17").Value
meat = Range("B19").Value
dairy = Range("B21").Value
produce = Range("B23").Value
bakery = Range("B25").Value
foodservice = Range("B27").Value
deer = Range("B29").Value
RLmilk = Range("B31").Value

' Define nonfood items
clothingcases = Range("B35").Value
clothingunits = Range("B37").Value
good360cases = Range("B39").Value
good360units = Range("B41").Value
household = Range("B43").Value
personalhygiene = Range("B45").Value
agency = Range("C3").Value
presstime = Now

' Define error messages
If agency = vbNullString Then
    MsgBox "Please select an agency."
    Exit Sub
End If
If IsEmpty(Range("B7").Value) Then
    MsgBox "Please enter the canned goods picked up by the agency, or enter 0 if no canned goods were taken"
    Exit Sub
End If
If IsEmpty(Range("B9").Value) Then
    MsgBox "Please enter the dry goods picked up by the agency, or enter 0 if no dry goods were taken"
    Exit Sub
End If
If IsEmpty(Range("B11").Value) Then
    MsgBox "Please enter the snacks picked up by the agency, or enter 0 if no snacks were taken"
    Exit Sub
End If
If IsEmpty(Range("B13").Value) Then
    MsgBox "Please enter the beverages picked up by the agency, or enter 0 if no beverages were taken"
    Exit Sub
End If
If IsEmpty(Range("B15").Value) Then
    MsgBox "Please enter the mixed cases picked up by the agency, or enter 0 if no mixed cases were taken"
    Exit Sub
End If
If IsEmpty(Range("B17").Value) Then
    MsgBox "Please enter the frozen food picked up by the agency, or enter 0 if no frozen food was taken"
    Exit Sub
End If
If IsEmpty(Range("B19").Value) Then
    MsgBox "Please enter the meat picked up by the agency, or enter 0 if no meat cases were taken"
    Exit Sub
End If
If IsEmpty(Range("B21").Value) Then
    MsgBox "Please enter the dairy picked up by the agency, or enter 0 if no dairy was taken"
    Exit Sub
End If
If IsEmpty(Range("B23").Value) Then
    MsgBox "Please enter the produce picked up by the agency, or enter 0 if no produce was taken"
    Exit Sub
End If
If IsEmpty(Range("B25").Value) Then
    MsgBox "Please enter the bakery trays picked up by the agency, or enter 0 if no bakery goods were taken"
    Exit Sub
End If
If IsEmpty(Range("B27").Value) Then
    MsgBox "Please enter the food service cases picked up by the agency, or enter 0 if no food service goods were taken"
    Exit Sub
End If
If IsEmpty(Range("B29").Value) Then
    MsgBox "Please enter the deer meat picked up by the agency, or enter 0 if no deer meat was taken"
    Exit Sub
End If
If IsEmpty(Range("B31").Value) Then
    MsgBox "Please enter the Rolling Lawns milk picked up by the agency, or enter 0 if no R.L. milk was taken"
    Exit Sub
End If
If IsEmpty(Range("B35").Value) Then
    MsgBox "Please enter the clothing cases picked up by the agency, or enter 0 if no clothing was taken"
    Exit Sub
End If
If IsEmpty(Range("B37").Value) Then
    MsgBox "Please enter the clothing units picked up by the agency, or enter 0 if no clothing was taken"
    Exit Sub
End If
If IsEmpty(Range("B39").Value) Then
    MsgBox "Please enter the cases of Good360 picked up by the agency, or enter 0 if no Good360 cases were taken"
    Exit Sub
End If
If IsEmpty(Range("B41").Value) Then
MsgBox "Please enter the Good360 units picked up by the agency, or enter 0 if no Good360 units were taken"
Exit Sub
End If
If IsEmpty(Range("B43").Value) Then
    MsgBox "Please enter the household cases picked up by the agency, or enter 0 if no household items were taken"
    Exit Sub
End If
If IsEmpty(Range("B45").Value) Then
    MsgBox "Please enter the personal hygiene cases picked up by the agency, or enter 0 if no personal care items were taken"
    Exit Sub
End If

MsgBox "Submitting pickup form for " & agency & "."

Application.ScreenUpdating = False
ChDir "\ofs-hq-srv01\Inventory"
Workbooks.Open Filename:="\ofs-hq-srv01\Inventory\DistributionData2.0.xlsx"
Cells(Application.ActiveCell.Row, 1).Select
ActiveCell.Value = presstime
Cells(Application.ActiveCell.Row, 2).Select
ActiveCell.Value = agency
Cells(Application.ActiveCell.Row, 3).Select
ActiveCell.Value = cannedgood
Cells(Application.ActiveCell.Row, 4).Select
ActiveCell.Value = dry
Cells(Application.ActiveCell.Row, 5).Select
ActiveCell.Value = snacks
Cells(Application.ActiveCell.Row, 6).Select
ActiveCell.Value = beverage
Cells(Application.ActiveCell.Row, 7).Select
ActiveCell.Value = mixed
Cells(Application.ActiveCell.Row, 8).Select
ActiveCell.Value = frozenfood
Cells(Application.ActiveCell.Row, 9).Select
ActiveCell.Value = meat
Cells(Application.ActiveCell.Row, 10).Select
ActiveCell.Value = dairy
Cells(Application.ActiveCell.Row, 11).Select
ActiveCell.Value = produce
Cells(Application.ActiveCell.Row, 12).Select
ActiveCell.Value = bakery
Cells(Application.ActiveCell.Row, 13).Select
ActiveCell.Value = foodservice
Cells(Application.ActiveCell.Row, 14).Select
ActiveCell.Value = deer
Cells(Application.ActiveCell.Row, 15).Select
ActiveCell.Value = RLmilk
Cells(Application.ActiveCell.Row, 16).Select
ActiveCell.Value = clothingcases
Cells(Application.ActiveCell.Row, 17).Select
ActiveCell.Value = clothingunits
Cells(Application.ActiveCell.Row, 18).Select
ActiveCell.Value = good360cases
Cells(Application.ActiveCell.Row, 19).Select
ActiveCell.Value = good360units
Cells(Application.ActiveCell.Row, 20).Select
ActiveCell.Value = household
Cells(Application.ActiveCell.Row, 21).Select
ActiveCell.Value = personalhygiene

ActiveWorkbook.Save
ActiveWindow.Close

Range("B7").Value = ""
Range("B9").Value = ""
Range("B11").Value = ""
Range("B13").Value = ""
Range("B15").Value = ""
Range("B17").Value = ""
Range("B19").Value = ""
Range("B21").Value = ""
Range("B23").Value = ""
Range("B25").Value = ""
Range("B27").Value = "0"
Range("B29").Value = "0"
Range("B31").Value = "0"
Range("B35").Value = ""
Range("B37").Value = ""
Range("B39").Value = ""
Range("B41").Value = ""
Range("B43").Value = ""
Range("B45").Value = ""
Range("C3").Value = ""

Application.ScreenUpdating = True

End Sub