**Group 2 Specification Sheet**

* 1. **<<Project Title>>**

Grocery Inventory & Order Management System (GIOMS)

Developed by: Clean Sweep Solutions (CSS)

Developed for: Whole Foods Market Inc. (WFMI)

**1.2 <<GIOMS>>**

Tasfia Hossain (Captain)

Yudava Roopnarine

Ivan Jimenez

Rakib Rary

Samuel Orellana

Matthew Murillo

Kevin Ren

**1.3 <<Project Description>>**

This specification document outlines the functionality and requirements for the Grocery

Inventory & Order Management System (GIOMS). The software will be licensed to

Whole Foods Market Inc. (WFMI) for use in their grocery warehouses and will be used

to keep track of product inventory as well as keep track of incoming orders and item

fulfillment requests. Fulfillment requests are updated during processing and after

completion. GIOMS will be accessed through a web application.

**1.4** **<<Purpose>>**

GIOMS will provide WFMI with a much faster and more accurate inventory and order

management system. WFMI’s current inventory and order management software was

internally audited and demonstrated discrepancies such as not providing Whole Food

stores with accurate delivery times, warehouse inventory quantities not reflecting

true-to-life inventory, and order quantity errors which would result in stores receiving

more or less products than they ordered. GIOMS aims to reduce these discrepancies by

allowing faster read/write access than other competitive software allowing WFMI

warehouses to quickly address any issues that may arise and have more data

consistency than is present with their current software.

\*\*GIOMS software is intended for logistical use by the grocery industry. Software can be

modified to function in any business industry.\*\*

**1.5 <<Business Model>>**

GIOMS aims to earn a name for itself by decreasing financial losses for companies that

deal with logistical issues by increasing accuracy and consistency. Costs as result of

logistical issues are projected to decrease by 16% during the first year of WFMI warehouses adopting GIOMS and 25% each subsequent year. The cost decrease will be less during the first year because of the costs and training associated with adopting new software. It is this kind of success that will allow GIOMS to gain more market share through use in other large companies in the grocery industry which will lead to increased adoption throughout other retail industries. These increased revenue streams will provide funding for potential future projects to address other logistical and stagnation issues in different industries.

**1.6 <<Scope>>**

GIOMS will provide software scalable for up to 5 locations (preliminary). The software will allow editing of inventory levels, order contents and order status. The end-user can assign permissions as necessary to additional users to extend equal or restricted read/write level access to aforementioned functions. GIOMS does not offer a mobile form of the software or retail-level order fulfillment. This is specifically intended for back-of-house purposes.

**1.7 <<Definitions/Acronyms>>**

1. CSS – a third party company that compiles logistics and software solutions for retail industry clients
2. STC (Simple Traffic Control) – preliminary software title for retail release
   1. **<<Use Case Diagram>>**

| **Actors** | Warehouse manager(s), permitted warehouse staff, Store logistics manager |
| --- | --- |
| **Preconditions** | At least 10 items fit ‘low stock’ criteria at a given location |
| **Triggers** | Store logistics manager submits fulfillment request |
| **Main success scenarios** | All items requested are available and order is fulfilled, warehouse inventory numbers adjusted |
| **Alternative paths** | 1. Order can only be partially fulfilled 2. Order canceled by store |
| **Post condition** | Store receives fulfillment order and in-store inventory is updated |

The actors in the table are the store logistics managers. They will be the one to submit orders on behalf of the employing store. Warehouse managers are also actors in the system. They will mark the progress of the orders. Additionally, any other warehouse members that are authorized can also be actors.

The preconditions for the use cases are that at least 10 items must fit the ‘low stock’ condition at a given store. This condition can vary depending on the store and product.

The triggers are the submission of an order request by store managers at participating locations.

The main success scenario:

A store submits an order request. Every item is available in the requested quantities and the warehouse manager marks the order as ‘Accepted’. The warehouse fulfills the request and warehouse staff prepares the items for shipping. The order is updated as complete. Come shipment day, the order is loaded for shipping and the order request is marked as shipped. The order is received and is accurate to the order and is marked as ‘Received – no errors’.

Alternative Paths:

A store logistics manager submits an order. The warehouse manager reviews the order and notes on the order that inventory levels are insufficient to fulfill order. Warehouse manager creates a counterorder visible to the ordering store indicating that warehouse inventory is insufficient to fulfill original order. The store logistics manager can accept the counterorder that is then marked as a submitted request. Warehouse manager notes this update and updates the status to processing. The counterorder is processed the same manner an order would be. Original order marked as ‘CO’ (counterorder) and the reason for the counterorder is specified. Optionally, the inquiring store can also reject the counterorder. This ends the order process, and the process is marked as ‘Canceled’.

Post conditions:

If an order or counterorder is processed and fulfilled, the inquiring store receives and processes the order. The store inventory levels are updated to reflect new quantities. The order is marked as completed.

* 1. **<<Acceptance Criteria>>**

1. **As** a store logistics manager,

**I want** to be able to submit orders quickly and easily

**So that** I spend less time in the ordering process and more time on other responsibilities.

**Scenario**: On any given day, the warehouse receives approximately a dozen orders. If the ordering and fulfillment process can be simplified on the software end, that would allow warehouse managers to allot more time to spend on staff training, safety protocols, email communications, etc. This also applies to warehouse managers.

1. **As** a logistics manager,

**I want** to be able to modify or cancel orders

**So that** I don’t submit too small an order to warrant shipping costs if my original order cannot be fulfilled.

**Scenario**: Warehouse managers cannot provide a counterorder on current software. If a store orders an item and we cannot fulfill any part of it, a member of staff would have to email the store manager directly to inform them of this. Sometimes, the staff is unable to do even this due to the workload on busy days. It would be more efficient to see when an order is unfulfillable from the moment it’s received rather than manually confirm quantities and verify the order and then jump to another software to send communications regarding any issues with said order. This avoids us having unhappy stores with what they receive in their order when it doesn’t line up with their original orders or holding up the shipping process by waiting to receive communication from them.

1. **As** a warehouse manager,

**I want** to give logistics managers the option to add quantity options for their orders within the software

**So that** I don’t waste employee hours to begin fulfilling an hour that will ultimately be canceled or worse, send an unsatisfactory fulfillment to a store.

**Scenario**: An order is received prior to Thanksgiving that includes 100 turkeys. We only have 35 available. On the current software, there is no option to include a range of acceptable quantities or the ability to leave notes for logistics. These have to be implemented separately via email as it stands. If the logistics team does not have the opportunity to read through the couple dozen or so daily emails and find the email containing said special requests, the order will be processed as the current software displays it or as close as inventory permits. If the store requesting fulfillment only had the turkey on high priority and the rest of the shipment could wait, it would have been more efficient to wait for a restock closer to their deadline, something that is also not currently displayable on given software. As such, it is a waste of company shipping costs and creates dissatisfaction between DC (distribution center) and stores.

1. **As** a warehouse manager,

**I want** to know immediately if an incoming order is fulfillable with current stock levels

**So that** I can respond immediately and accurately with an in-progress update or a counterorder.

**Scenario**: As mentioned prior, current software does not keep track of plausibility of incoming orders. Instead, they must be manually verified. The ideal solution would be a software that can analyze incoming orders and compare items and quantities to current stock levels available for shipment and validate whether they can be fulfilled or not. Ideally, a ‘low stock’ warning when an item requested is 10% away from depleting the warehouse stock of said item to 0 i.e., if ‘cereal A’ has 50 units left in the warehouse and a store requests 45, the software should display a ‘low stock’ warning alongside that item.

1. **As** a warehouse manager,

**I want** to allow staff of my choosing to access orders and inventory to process and modify orders in my absence

**So that** I am not the only employee who can fulfill my role and not every employee can access my permissions.

**Scenario**: A warehouse manager is not always on site but there is always at least a warehouse logistics lead or assistant manager. As such, I want to be able to assign them the same permissions I have and access the same orders I have access to so that they may be able to carry out my role in my absence without them having access to my personal details i.e., create several logins with shared access to the same orders. Additionally, I should be able to assign employees access to read current orders but not modify or counteroffer/create new orders i.e., restricted access in the event that a member of management is not available to fulfill the role or is not present for some unforeseen circumstance.

* 1. **<<Assumptions and Constraints>>**

Current business resource limitations estimate scaling for 5 locations but if more time is needed to ensure functional base software addresses customer needs, that scalability may decrease. We are constrained to deliver functional software by the beginning of the first quarter so that the client, Whole Foods Market, Inc., can trial run GIOMS during their lightest quarter. We assume we will have access to appropriate staff for testing and questioning. We assume Whole Foods Market Inc. has suitable network infrastructure to support the software.

**2.4 <<Platform Requirements Specification>>**

**Windows Systems:**

| Operating System | Windows 2000 – Service Pack 4  Windows XP – Service Pack 2  Windows 2003 R2 |
| --- | --- |
| CPU Type and Speed | Pentium IV 2 GHz MHz or faster |
| Memory | 1 GB RAM |
| Display | 65536 colors set to at least 1280 x 1024 |
| Hard Drive | 65 MB if you already have JDK 1.5  137 MB if you do not have JDK 1.5 |
| Java SDK | JDK 8.0 Update 351 or later |

**Linux Systems:**

| Operating System | Red Hat Enterprise Linux 6.0  Fedora Core 8 |
| --- | --- |
| CPU Type and Speed | Pentium IV 2 GHz MHz or faster |
| Memory | 1 GB RAM |
| Display | 65536 colors set to at least 1280 x 1024 |
| Hard Drive | 137 MB |
| Java SDK | JDK 8.0 Update 351 or later |

**Mac OS X Systems:**

| Operating System | Apple Mac OS X Version 10.4.x |
| --- | --- |
| CPU Type and Speed | Dual 1.25 GHz G4/G5 (11 GHz G4 minimum) |
| Memory | 1.4 GB RAM or 1.1 GB minimum |
| Display | “Thousands” of colors |
| Hard Drive | 137 MB |
| Java SDK | Sun J2SE 1.5 release 1 |

**<<Process Flowchart>>**

Diagram

Description automatically generated

**<<Journey Flowchart>>**

Diagram

Description automatically generated