Warehouse Group 2: Samantha Miller, Jacques Troussard, Taylor Dohmen

Requirements Checklist

## 2.3 Functional Requirements

*2.3.1. Log into System*

*Description:* A user accessing the site shall provide a email and password to use the application.

*Main Flow:*

1. User accesses site by Internet browser.
2. System displays login screen, prompting email and password fields, and a Login button.
3. User types email and password and clicks Login.
4. System verifies matching user information and displays the Home page of the site.

*Alternate Flow A:*

1. User accesses site page that requires login, like the Home or Import page.
2. The system redirects any user without a session to the Login page.

*Alternate Flow B:*

3. The user types either the email or password. The Login button is disabled until both fields are filled.

*Alternate Flow C:*

3. The user types a wrong email or password.

4. The system checks the provided information and returns a error message proclaiming the email or password was invalid.

5. Repeat until Main Flow 4 is met.

*2.3.2. Log out of System*

*Description:* A logged-in user shall exit their logged-in session on the site.

*Main flow:*

1. A logged-in user clicks the Logout button on the menu.
2. The system removes the user from the session and displays the Login page of the application.

*2.3.3. Search for Product*

*Description:* Any user shall search for products based on warehouse location, product name, or product number.

*Main Flow:*

1. A logged-in user navigates to the Search page via the menu bar.
2. The system displays the Search page, which offers options to search for products by warehouse location, product name, or product number, as well as Search and Reset buttons.
3. The user enters requirements in the form and clicks Search.
4. The system queries all related information and displays all product information in the table.

*Alternate Flow A:*

3. The user clicks the Reset button instead of Search.

4. The system clears the fields of the search boxes.

*Alternate Flow B:*

5. The user clicks the Reset button after searching.

6. The system clears the returned table and search fields.

*Note:* This functionality is on the Products page. Fields clear after Search button is clicked.

*2.3.4. Import Inventory File*

*Description*: Any user shall upload a Comma Separated Values (CSV) file into the system which specifies what changes are being made to the warehouse inventory. The system shall then update the product database to account for the specified changes.

*Main Flow*:

1. User navigates to an inventory management page and selects the option to upload an inventory management file. (See Appendix B.1 for preliminary import configuration.)
2. System displays an Upload window including a File Explorer and Upload button.
3. User navigates the File Explorer, selects the desired file, and clicks the Upload button.
4. The system parses the file and updates the inventory database based on the file’s specification. If the operation is time intensive, the system should display a progress indicator to the user.
5. The system displays a message informing the user that the operation was executed successfully.

*Alternate Flow A*:

1. The system parses the file and finds that it is improperly formatted.
2. The system displays a message informing the user that the file is improperly formatted. Return to Main Flow 1.

*Alternate Flow B*:

1. The system parses the file and finds that there is an invalid operation.
2. The System displays a message informing the user of bad operation. Return to Main Flow 1.

*2.3.5. Create Invoice*

*Description*: A sales associate shall select a subset of their mobile warehouse’s inventory and create and invoice for the sale of those items.

*Main Flow*:

1. Sales associate navigates to a Create Invoice page.
2. System displays form with a field for entering customer information, a list of the products in the sales associate’s mobile warehouse with areas to specify the sold quantity of each item, and a submit button.
3. Sales associate enters the customer’s information, specifies the quantity of each item being sold, and selects submit.
4. System validates that the quantities are not larger than the quantity of the item in stock.
5. System generates invoice based on the price and quantities of the items sold and customer information given and saves the sale record in the database.
6. System updates inventory database to account for the sold items.
7. System displays message indicating that the sale was successfully documented.

*Alternate Flow A*:

1. System finds that a sale quantity is larger than the inventory quantity and displays a message indicating this error. Return to Main Flow 2.

*2.3.6. Search for Invoice (by Sales Associate, Customer, Date)*

*Description*: Any user should search for and review records of previous sales.

*Main Flow*:

1. User navigates to Sales page.
2. System displays a list of previous sales, sorted by date (most recent first), a search field, a drop-down menu to select search field, and a search button .
3. User selects what field they want to search by, enters a term into the search field, and selects search.
4. System queries the sales database to and displays a list of sales records that match the search terms.
5. User selects a particular search result.
6. System displays invoice and an option to download the invoice as a txt file for that particular sale.

*Note:* This page is found under the Invoice menu.

*2.3.7. Generate Sales Metrics (by associate between range of dates)*

*Description*: A manager should generate a report detailing what each sales associate sold, given a date range.

*Main Flow*:

1. Manager navigates to Sales Metrics page.
2. System displays input fields for start date and end date and a generate report button.
3. Manager fills in the date fields and selects the Generate Report button.
4. System queries the sales database and compiles a report of metrics breakdown including total sales and sales per item each associate sold during the time period.
5. System displays a list of associates, their totals, and a sublist of items the quantity of each that the associate sold.

*Note:* This functionality is provided by the Inventory Search page, if the user searches for an employee and a time frame.

*2.3.8. Create a User Account*

*Description*: The administrator shall create any user accounts and the manager should create sale associate accounts, allowing unique sales associates to access their inventory stores.

*Main Flow (Admin):*

1. Administrator log into the postgresql database as an administrative user.
2. Administrator creates an inventory manager role with a preselected list of privileges allowing them to access and modify the ‘users’ table as well as all other tables necessary to facilitate the manager's functions.

*Main Flow (Manager):*

1. Manager logs into their account.
2. System compares credentials to SQL users table.
3. Manager navigates to manage users management page.
4. System queries users database and renders a table with users and check boxes
5. Manager selects ‘Create User.’
6. System renders web form containing all the necessary fields required to insert a new user.
7. Manager finishes form and selects submit (reset simply clears form).
8. System receives post request from manager user, scrubs input and updates SQL users table.
9. System generates password and renders success page with username and password.

*Alternate Flow (Manager):*

1. System generates non-generic error and renders error page.

*Note:* This functionality is found under the Manager menu under Accounts.

*2.3.9. Update a User Account*

*Description*: The manager should update the various attributes associated with any a user account of lesser privilege level.

1. Manager logs into their account.
2. System compares credentials to SQL user tables.
3. Manager navigates to user page.
4. System queries users database and renders a table with users and check boxes.
5. Manager selects user checkbox and clicks ‘Update.’
6. System queries database for user and attributes.
7. System renders a web form preloaded with user attributes.
8. System logs date, time, manager id, and user’s details.
9. Manager refills desired user attribute fields and selects ‘Submit.’
10. System receives post request from manager user, scrubs input and updates SQL users table and appends changes to log.
11. System generates success page.

*Alternate Flow:*

10. System generates non-generic error and renders error page

*Note:* This functionality is found under the Manager menu under Accounts.

*2.3.10. Disable a User Account*

*Description:* The manager should modify the access status to the system of any user account of lesser privilege level.

1. Manager logs into their account
2. System compares credentials to SQL user tables.
3. Manager navigates to user page.
4. System queries users database and and renders a table with users and check boxes.
5. Manager selects user checkbox and clicks ‘Update.’
6. System queries database for user and attributes.
7. System renders a web form preloaded with user attributes.
8. System logs date, time, manager id, and user’s details.
9. Manager toggles Disable/Enable radio button and selects submit.
10. System receives post request from manager user, scrubs input and updates SQL users table and appends changes to log.
11. System generates success page.

*Alternate Flow:*

10. System generates non-generic error and renders error page.

*Note:* This functionality is found under the Manager menu under Accounts.

*2.3.11. Update Product Attributes*

*Description:* Inventory manage should access list of products queried from database and modify attributes individually.

*Main Flow:*

1. Manager logs into their account.
2. System compares credentials to SQL user tables.
3. Manager navigates to inventory management page.
4. System renders management page with link to modify item.
5. Manager click modify item link.
6. System renders page with part description field and submit button.
7. Manager enters item key terms (part number, name, etc.)
8. System queries database and renders table with possible matches.
9. Manager selects item and clicks on ‘Modify.’
10. System renders item page, a preloaded web form with item attributes.
11. Manager modifies data and clicks submit.
12. System receives post request from manager user, scrubs input and updates SQL users table and appends changes to log.

*Alternate Flow:*

12. System generates non-generic error and renders error page.

## 2.4 Constraints

2.4.1 Access & connectivity

The software’s classification as a web application is subject to some innate access and connectivity constraints, chiefly an internet connection. The entire system’s operations are based upon a connection with high integrity. Since the application is hosted in the cloud, the very nature of its existence is dependent upon a connection. Furthermore the connection must continuously reliable because almost every function the application performs fetches data from a database over the internet.

2.4.2 Virtualized Components

Performance in this application is theoretically constrained by the capacities of the cloud provider, Google. The application itself and database will be hosted in the cloud. The entire system could be configured to grow its resources (computing/database capacities) if the main system detects it necessary, there still technically exists an upper threshold to these resources, but more realistically, however unlikely Google Cloud services are still vulnerable to physical as well as virtual complications.

2.4.3 Scalability

This system has virtually unlimited depth scalability. The database can be copied and moved into a larger system without many problems. However its breadth, that is appending new attributes to any existing table, is constrained by the database schema and is intentionally so to limit failures and increase reliability of the entire system.