**Goal fulfilled through requirements:**

* The System should be reliable.
* The System should improve efficiency.
* The System should integrate with <<*Productivity Tool: Excel, Sheets, Word*>> to render reports.
* The System should be developed with the evolutionary capability to integrate with a website for customer interaction and business operations management.

**Follow-up Questions in RE spiral: *Follow-up questions for the customer are provided as comments.***

**Possible Required Classes to Implement (with priority level):**

* Inventory Item (p1)
* Product (p1)
* Pickup Location (p2)
* Customer *(from Abstract class Person) (p2)*
* Order (p3)
* User *(from Abstract class Person) (p3)*
* *Authentication (p3)*
* *TextUIs or AppGUIs (p4)*
* *Menu (p4)*

**Non-Functional Requirements:**

**Section/Module 0: (Priority 1)**

1. Accessibility and Reliability:
   1. The System shall provide on-demand accessibility to the authorized user.
   2. The System shall store all app data locally on the device for offline persistent storage. (requires PC, app), developer, IDE)
2. Infrastructure and Development:
   1. The System shall operate on <<*32-bit 4GB RAM Windows 10 PC*>> platform.
   2. The System should be developed using Java.
3. Safety and Security:
   1. The System shall allow authorized users to be created. (requires PC, app, create user page)
   2. The system shall allow user login and logout. (depends on 0.3.a, authentication)
   3. The System shall only allow fully authorized users to use the system. (depends on 0.3.b, authentication)
   4. The System should allow the authorized user to edit and reset their login credentials.
   5. The system shall require the user to be authenticated after a session goes idle for a user-specified time period in minutes.
   6. The System shall Encrypt all databases for privacy and security.
   7. The System shall make use of checksum and redundancies for reliability, to ensure data integrity and to prevent data loss.

**Functional USER Requirements:**

**Section/Module 1: (Priority 1)**

1. The system shall create product entries. (requires PC, app, item description)
2. The system shall display a list of products. (depends on 1.a)
3. The system shall allow the user to search for a product by criteria.
4. The system shall allow the user to edit product details.
5. The System shall allow the user to filter by specific criteria.
6. **System requirement activity:** *The System shall provide verification for “create” and “edit” actions in the database.*

**Section/Module 2: (Priority 3)**

1. The System shall create an order. (depends on 1.b, 6.d, 7.c)
2. The System shall allow the user to delete an order.
3. The System shall display a list of orders. (depends on 2.a.)
4. The System shall filter orders by select criteria.
5. The System shall allow the user to edit an order.
6. The System shall allow the user to search for an order.
7. **System requirement activity:** *The System shall provide verification for “create” and “edit” actions in the database.*

**Section/Module 3: (Priority 4)**

1. The System shall generate a weekly report of all orders at the end of previous week.
2. The System shall report on number of orders, order product names and percentages, order status percentages, delivery location percentages and inventory level statuses for all items.
3. The System shall allow the user to generate reports of orders by date.
4. The System shall provide notification of Low Stock.

**Section/Module 4: (Priority 3)**

1. The System shall calculate the number of available bracelets. (requires PC, app, formula)
2. The System shall record the number of each bead required for the various standard-sized bracelets. (requires PC, app, records)

**Section/Module 5: (Priority 1)**

1. The System shall allow the user to create an Inventory Item. (requires pc, app, physical items)
2. The System shall display inventory levels for all items. (depends on 5.a)  
   (beads, spacers, pendants, pouches)
3. The System shall allow the user to edit an Inventory Item. (depends on 5.b)
4. The System shall allow the user to delete an inventory item.
5. The System shall filter for all stock and low stock items.
6. **System requirement activity:** *The System shall provide verification for “create” and “edit” actions in the database.*

**OTHER Functional Requirements:**

**Section/Module 6: (Priority 2)**

1. The System shall allow the User to create pickup location. (requires PC, app)
2. The System shall allow the user to edit pickup location.
3. The System shall allow the user to delete pickup location.
4. The System shall display a list of available pickup locations. (depends on 7.a)
5. The System shall filter pickup locations based on customer parish.
6. **System requirement activity:** *The System shall provide verification for “create” and “edit” actions in the database.*

**Section/Module 7: (Priority 2)**

1. The System shall allow a user to create a customer record. (requires PC, app and customer)
2. The System shall allow a user to edit customer details.
3. The System shall display customers. (depends on 7.a)
4. **System requirement activity:** *The System shall provide verification for “create” and “edit” actions in the database.*
5. *The System* ***should*** *display all orders for a selected customer, order status and sorted by delivery date.*
6. *The system* ***should*** *display all closed orders in red.*

**Architectural Design Considerations:**

* MVC ([web-development](https://www.youtube.com/watch?v=E7Voso411Vs&t=2264s)) or Layered

***Design Idea/Consideration for Orders:***

* *Orders View displays 4 sections/tabs: Products, Open, Created, Delivered*
  + *Products section displays all bracelet types, stock, undelivered amount ordered, amount applied and the pending amount to be created. Tapping on a type will allow the authorized user to indicate how much he just created, giving them the option to manually or automatically apply to open orders. Assignments change relevant order statuses from “Open” to “Created” when all the products in that order are completed. It chooses order to add products based on first come first served basis.*
  + *Open section shows all orders that are not yet fully created. Indicating type and amounts pending in the order. Tapping order allows the authorized user to see full order details. Tapping Edit allows user to edit order details. Tapping cancel allows user to delete order.*
  + *Created section shows all the orders that are fully created and ready for delivery. This is arranged in order by delivery date. Tapping an order allows the user to see full order details. Tap “Delivered” button to change order status to delivered. Tapping Edit allows user to edit order details. Tapping cancel allows user to delete order.*
  + *Delivered shows all the fulfilled orders.*
  + *Excess products created are stored as Stock to be applied to next order. Excess due to editing order items, order cancellation or return.*