**Group 18 – CITADEL**

**Course CODE: CSC 316**

**ANALYSIS AND DESIGN OF INVENTORY MANAGEMENT SYSTEM OF POINT-1 LOUNGE**

**GROUP MEMBERS AND ROLES**

* 210805095 – Opadeji Abdullah Ololade – Senior Systems Analyst
* 230805542 – Giwa Kehinde Ahmad – Secretary
* 210805094 – Aweroro Ayomide Abdullahi – Data Model Engineer
* 210805132 – Kolawole Usman Ariyo – Software Designer
* 210805165 – Ishola Abdullahi Oladapo – Logistics Manager

**CHAPTER 1**

**Introduction**

**1.1 Background of the Study**

An important aspect of all businesses dealing with goods and supplies is *inventory management.* Our case study, **Point 1 Lounge,** which serves as a relaxation center for students and staff, currently tracks inventory manually. Employees record stock levels and sales in handwritten logs, making it difficult to monitor inventory, analyze sales trends, and assess profitability accurately.

To address these issues, our team, Team Citadel, is developing an **Inventory Management System (IMS)** tailored for **Point 1 Lounge**. This system will automate inventory tracking and generate stock and sales reports in order to enhance the business efficiency.

Following the **System Development Life Cycle (SDLC) using the Waterfall Model**, our team will systematically analyze the existing system, identify issues, define requirements, and design a structured solution.

**1.2 Problem Statement**

The manual inventory system at Point 1 Lounge presents several issues:

* **Inefficiency** – Tracking inventory by hand is slow and prone to errors.
* **Lack of real-time updates** – Employees must manually verify stock levels.
* **Difficulty in generating reports** – Trends in sales and inadequate stock are not easily identified
* **Accountability issues** – Without automation, proper record-keeping is not guaranteed, which increases the risk of human error.

A structured digital solution is required to improve inventory tracking, report generation, and overall efficiency.

**1.3 Objectives**

The proposed system would aim to:

* Automate inventory tracking to reduce errors.
* Generate **weekly and monthly reports** to help with stock analysis.
* Implement access control, ensuring **only authorized staff** can modify inventory records.

**1.4 Project Scope and Constraints**

The Inventory Management System will:

* Focus on managing **sales, inventory, and depleted stock** records.
* Generate real-time updates and **automated reports** for the business owner.
* Be restricted to **staff and management access only** (customers will not interact with the system).

Constraints:

* Internet connectivity is required for **data synchronization and report generation**.

**1.5 Alternative Solutions**

To solve the inventory management challenges, the following alternatives were considered:

1. **Manual Spreadsheet System:** Using Excel or Google Sheets for inventory tracking.
   * **Pros:** Simple and cost-effective.
   * **Cons:** Still prone to human error.
2. **Off-the-Shelf Inventory Software:** Purchasing existing inventory management software.
   * **Pros:** Pre-built and feature-rich.
   * **Cons:** May not fully align with Point 1 Lounge’s specific needs.
3. **Enterprise Solutions Software:** Using high-end enterprise inventory solutions.
   * **Pros:** Comprehensive functionality and scalability.
   * **Cons:** Expensive and complex to implement.
4. **Custom-Built Web Application (Chosen Solution):** Designing a tailored inventory management system.
   * **Pros:** Built specifically to match the business requirements.
   * **Cons:** Requires time and development effort.

**1.6 System Description**

The proposed **Inventory Management System (IMS)** will operate as follows:

* **Sales Action:** Staff inputs sales data, and the system updates stock levels accordingly.
* **Supply Action:** New stock is recorded into the system, updating the **inventory file**.
* **Price Action:** Management can update product pricing when necessary.
* **Depleted Stock Management:** The system automatically logs items that need restocking and generates **weekly and monthly reports** to assist in order planning.
* **Access Control:** Only **authorized staff** will have login credentials to interact with the system.

**1.7 Feasibility Assessment**

**1.7.1 Economic Feasibility**

**Tangible Benefits**

The estimated tangible benefits of the Inventory Management System over a 5-year period:

| **Tangible benefit** | **Estimated Value (₦) in 5 Years** |
| --- | --- |
| Reduced Overstocking & Understocking | 10,000,000 |
| Improved Supplier Management | 3,000,000 |
| Better Demand Forecasting | 10,000,000 |
| Error Reduction | 1,000,0000 |
| Efficient Decision-Making with Data Insights | 9,000,000 |
| **Total tangible benefits** | **₦ 33,000,000** |

**Intangible Benefits**

The intangible benefits of the Inventory Management System:

1. Increased employee productivity
2. Stronger supplier relationships
3. Reduced stress & workload
4. Greater business scalability

**Tangible Costs**

The estimated one-time costs for developing and implementing the Inventory Management System:

| **Tangible Cost** | **Estimated Cost (₦)** |
| --- | --- |
| Software Development | 500,000 |
| Hardware Installation | 200,000 |
| **Total** | **700,000** |

The estimated annual recurring costs for operating and maintaining the Inventory Management System:

| **Tangible Cost** | **Estimated Cost (₦)** |
| --- | --- |
| Cloud Hosting & Data storage | 750,000 |
| System Maintenance & Support | 600,000 |
| Total | 1,350,000 |

**Conclusion**

* The project is cost-effective compared to purchasing enterprise software.
* Development will be handled by the team, reducing outsourcing costs.

**1.7.2 Technical Feasibility**

The implementation of the Inventory Management System requires the following:

**Software Requirements:**

• Operating System: Any specification of Microsoft Windows XP or higher.

• Interface Design: Figma, Draw.io, Lucidchart.

• Front-end: React, TypeScript.

• Back-end: Java.

• Database: MySQL for inventory tracking and sales records.

**Hardware Requirements:**

• Processor of at least 2 GHz clock speed.

• RAM should be a minimum of 4GB.

• Hard disk of at least 256GB SSD.

• Monitor, keyboard, mouse, and barcode scanner for efficient inventory management

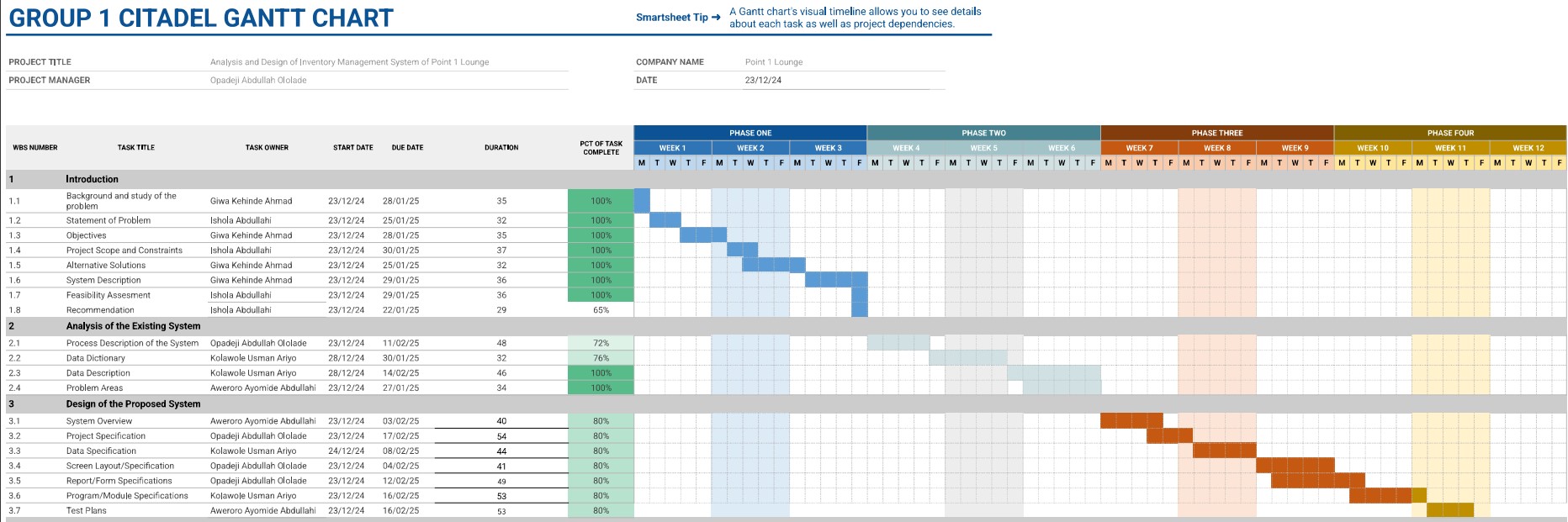
**1.7.3 Schedules, Timeline, and Resource Analysis**

* Project Start Date:
* Deadline:
* Key Milestones:

**Schedule Analysis**

The project follows a structured timeline using the Waterfall Model, ensuring each phase is completed before moving to the next.

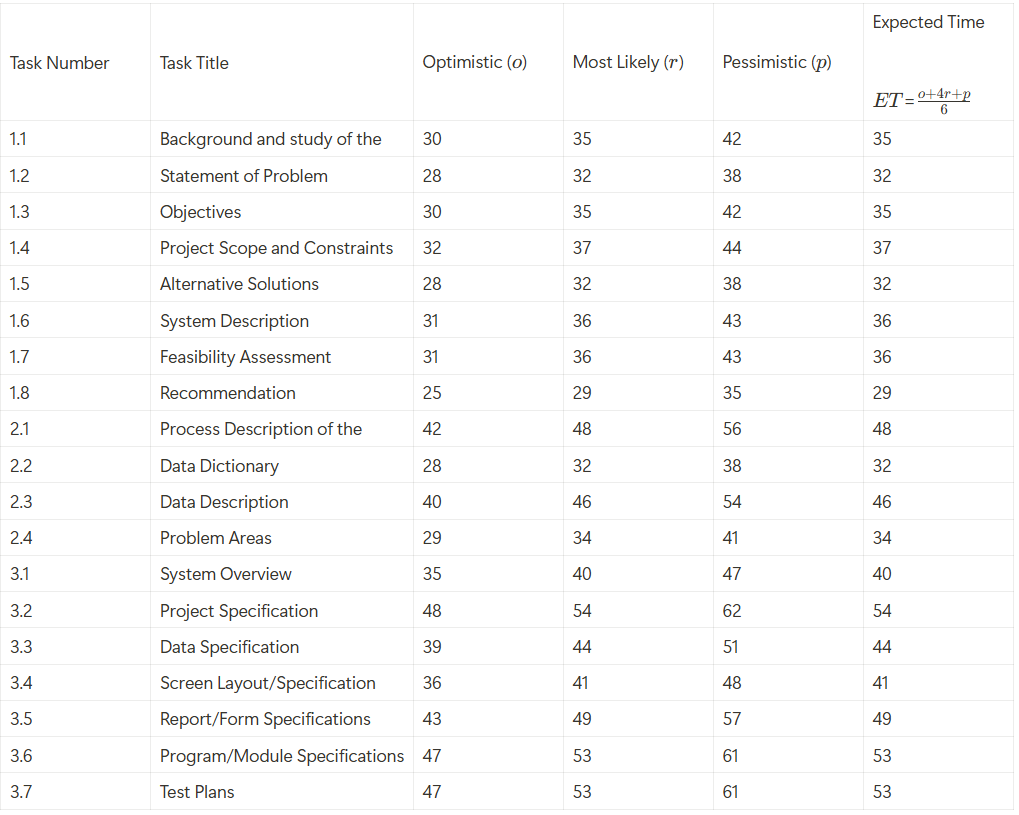
The Gantt chart depicting the tasks needed to be done and it's time for completion:

****

[Gantt Chart Link](https://docs.google.com/spreadsheets/d/139A9ggPWBvFcYCX4eTanHRoEaKlxs1Jn8QggOLWp_LE/edit?usp=sharing)

**Timeline Analysis**

The Project Evaluation Review Technique (PERT) was used to do the timeline analysis:



**Resource Analysis**

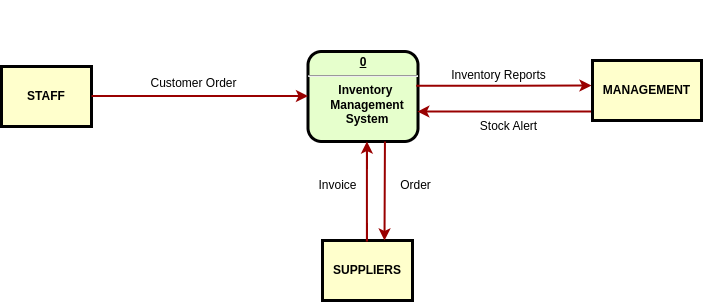
The project primarily utilized resources such as software tools (Figma, Draw.io, Lucidchart, React, Java, MySQL) and manual effort from team members. The major expenses were on data usage and transportation for meetings. Each team member contributed individually to these expenses.

**CHAPTER 2**

**ANALYSIS OF THE CURRENT SYSTEM**

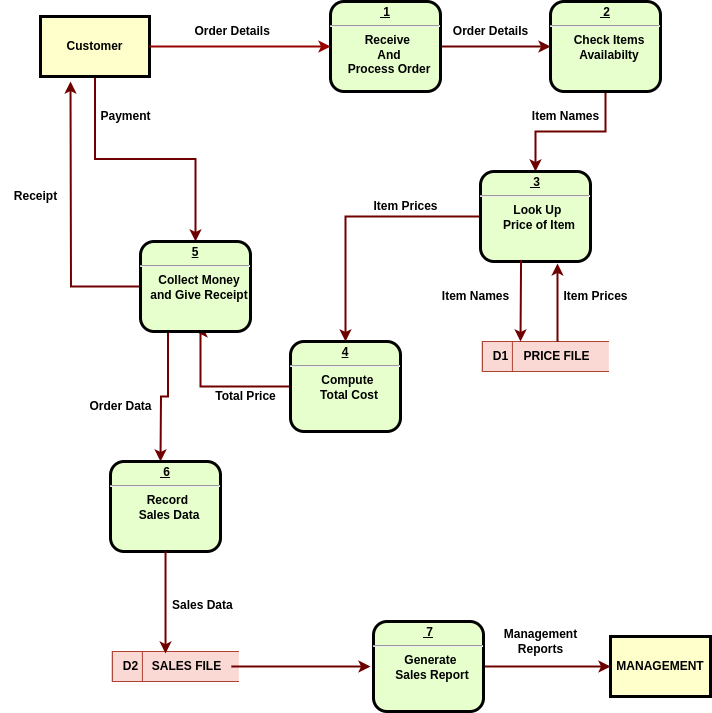
**2.1 Process Description of the System (using Context and DFDs)**

**CONTEXT LEVEL DIAGRAM OF THE CURRENT SYSTEM**

****

**2.1.1 Current Physical Data Flow Diagram (CPDFD)**

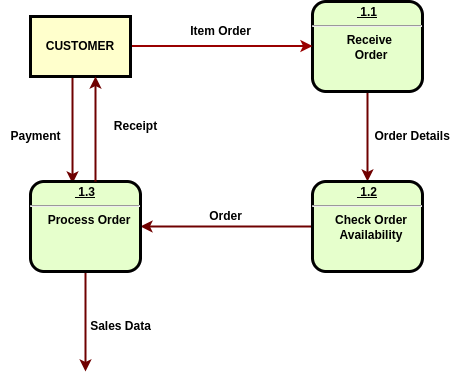
**PHYSICAL LEVEL 1 OF THE CURRENT SYSTEM**

****

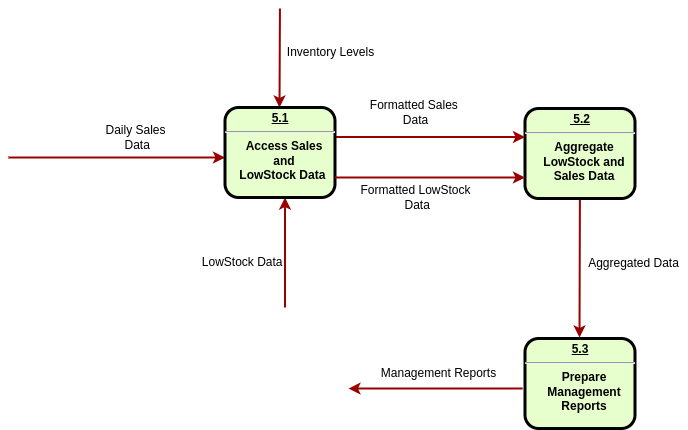
**2.1.2 Current Logical Flow Diagram**

**LOGICAL LEVEL 0 OF THE CURRENT SYSTEM**

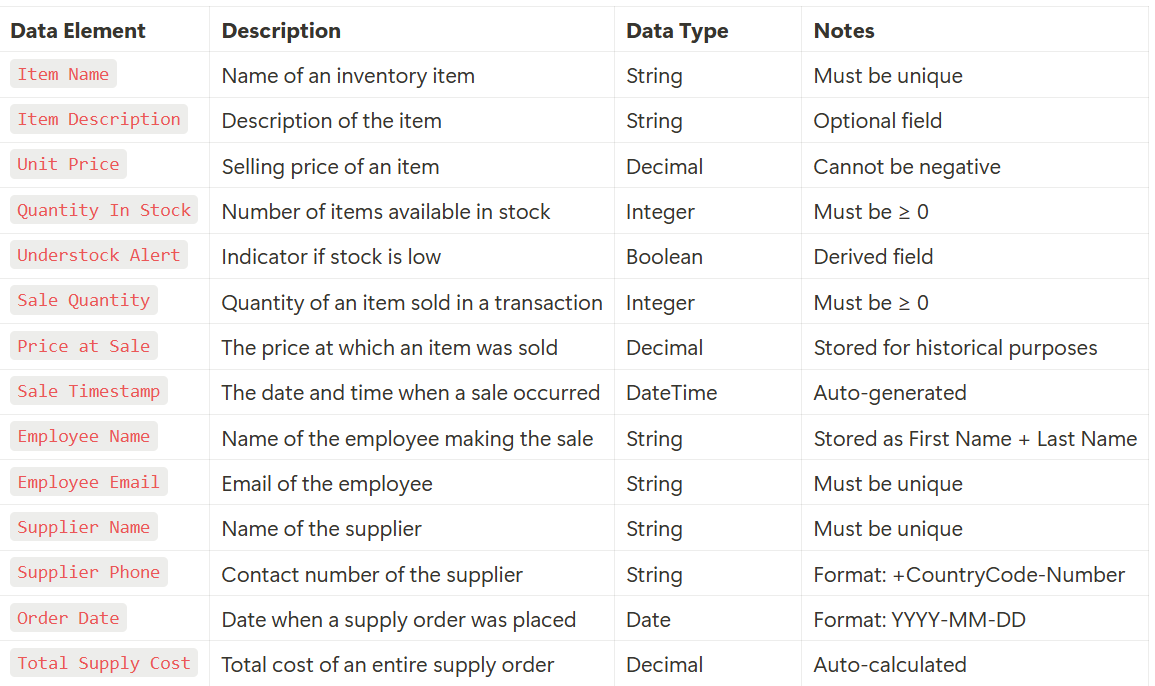
****

**LOGICAL LEVEL 1 OF THE CURRENT SYSTEM - PROCESS 1**

**LOGICAL LEVEL 1 OF THE CURRENT SYSTEM - PROCESS 5**

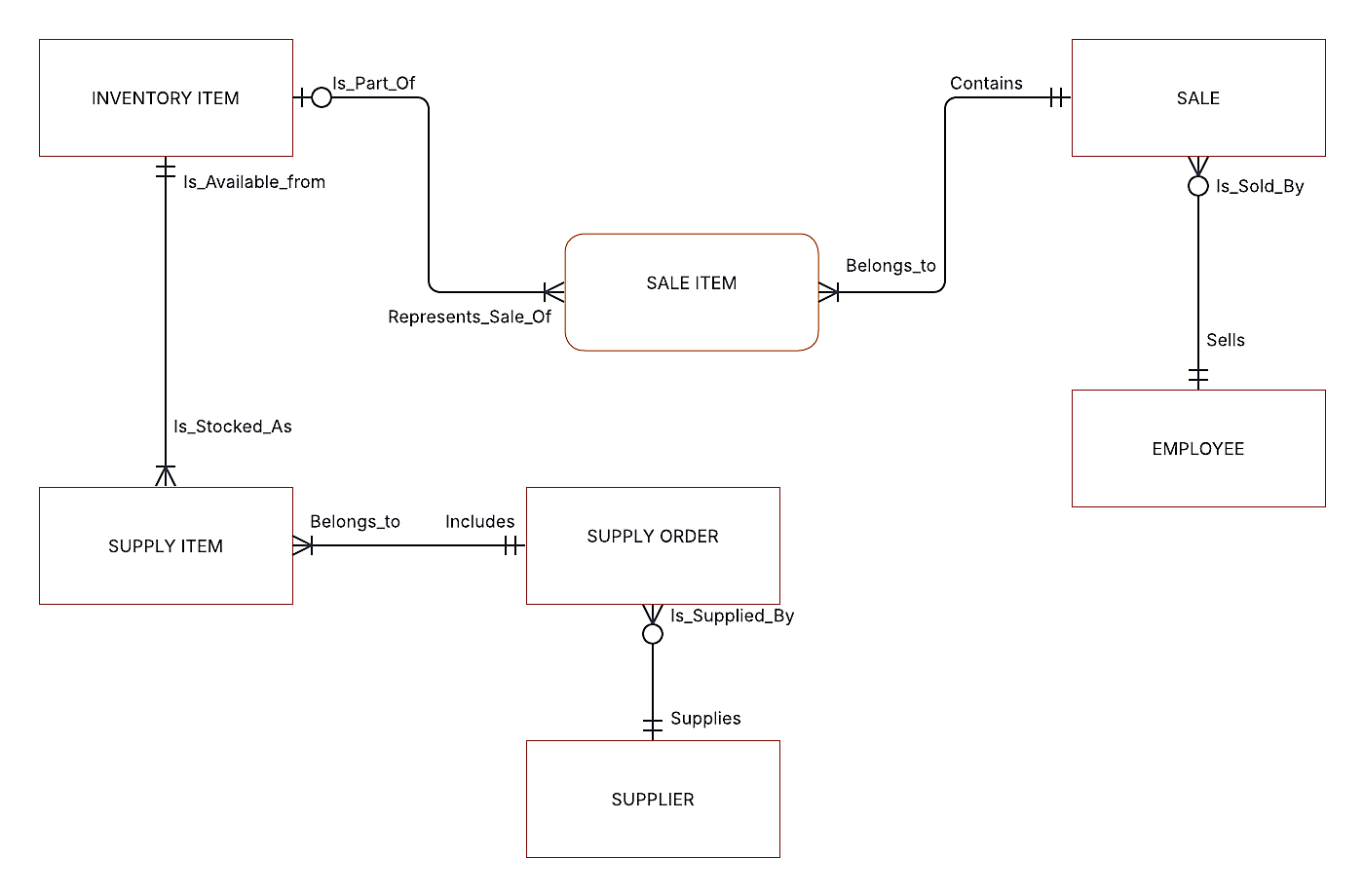
****

**2.2 Data Dictionary**

****

**2.3 Data Description of the System (Using ERDs)**

[Live Li nk](https://lucid.app/lucidchart/859564d4-a30e-4cb9-bbae-fca527019c65/edit?page=0_0&invitationId=inv_d7926c4c-6405-4481-a00e-96a0e619fdd5#)

****

**2.4 Problem Areas**

After careful analysis of the current system, several problems have been identified, Key problems include

1. **Error Prone:** Manual entry can easily lead to error in sales reporting and miscalculations
2. **Lack of Authentication:** There is no authentication or tracking of staff interactions, which can lead to records being altered or lost.
3. **Inefficient Stock Monitoring:** Identifying low-stock items is challenging, as staff must physically check all inventory.
4. **No Sales Trend Analysis:** There is no way to track sales trends.

**CHAPTER 3**

**DESIGN OF THE PROPOSED SYSTEM**

**3.1 System Overview**

**THE SYSTEM SHOULD HAVE THE FOLLOWING CAPABILITIES**

* **SALES ACTION:** Record sold item.
* **SUPPLY ACTION:** Record newly supplied (quantity) of items or add to existing items.
* **PRICE ACTION:** Update the prices of items.

**IN ANY CASE, THREE (3) SEPARATE FILES (DATABASE) WILL BE REQUIRED:**

* SALES FILES:
* RECORD Sold items
* GENERATE Weekly and Monthly sales report
* INVENTORY FILE:
* TRACK Inventory Items
* DEPLETED FILE
* TRACK items that need restocking
* GENERATE Demand order list
* GENERATE Weekly and Monthly report

**SALES ACTION**

**Info**

This will modify;

* INVENTORY FILE
* SALES FILE
* DEPLETED FILE

1. **A Customer**

* MAKES an order request

2. **An Authorized Staff**

* TAKE the order request
* INPUT the order on the System
  + TRIGGER: check if the item, and the required quantity is available
  + If the trigger action return NILL, we record the item in DEPLETED FILE
    - MODIFY the DEPLETED FILE by adding the depleted item
* if the trigger is SUCCESS:
  + MODIFY the INVENTORY FILE by decreasing the quantity of items bought
  + MODIFY the SALES FILE by adding the quantity of item bought such that
    - CREATE another row in the records if the old price have been updated

**SUPPLY ACTION**

This could be a weekly or bi-weekly action. Although it is possible to be a daily action based on daily or weekly sales report.

**Info**

This will modify;

* INVENTORY FILE
* DEPLETED FILE

#### An Authorized Staff OR Management

 *AGGREGATE* all the items in the *DEPLETED FILE* into a single record

 *COMPUTE* the required quantity for each item (manually)

 *ADD* new items that is not present in the aggregation if necessary

 *SEND* order request is then sent to the necessary suppliers

#### An Authorized Staff OR Management

 *CONFIRM* the order request against the order supplied

 *UPDATE* the *Inventory file*

 *ADD* to new quantity in the *INVENTORY FILE* if the item is present

 *ADD* a new item, its price and quantity if the item is not present

 *UPDATE* prices of item IF necessary

 *UPDATE* the *Depleted file*

 *REMOVE* the items whose order has been delivered

# **PRICE ACTION**

This will allow for the update of prices.



**Info**

This will modify;

* INVENTORY FILE

1. \*\*An Authorized Staff OR Management

 *UPDATE* the prices of items in *INVENTORY FILE*

**DATABASE FILES**

This section is an attempt to investigate the necessary column for each of the database files.

**INVENTORY** **FILE**

 **ID**: tracks the item number

 **NAME**: name of the item

 **PRICE**: price-per-unit of the item

 **QUANTITY**: available quantity of the item

 **DATE** **SUPPLIED**: date the item was supplied

**CATEGORY**: category of beverages the item belongs to



**SALES FILE**

 **ID**: a unique identifier for each transaction

 **ITEM ID**: a reference to the ID of the item in the INVENTORY FILE

 **QUANTITY SOLD**: quantity of the item sold

 **SALE PRICE**: total price of the item sold, based on the quantity

 **TIMESTAMP**: time of the transaction

**DEPLETED FILE**

 **ID**: an identifier for each recorded item

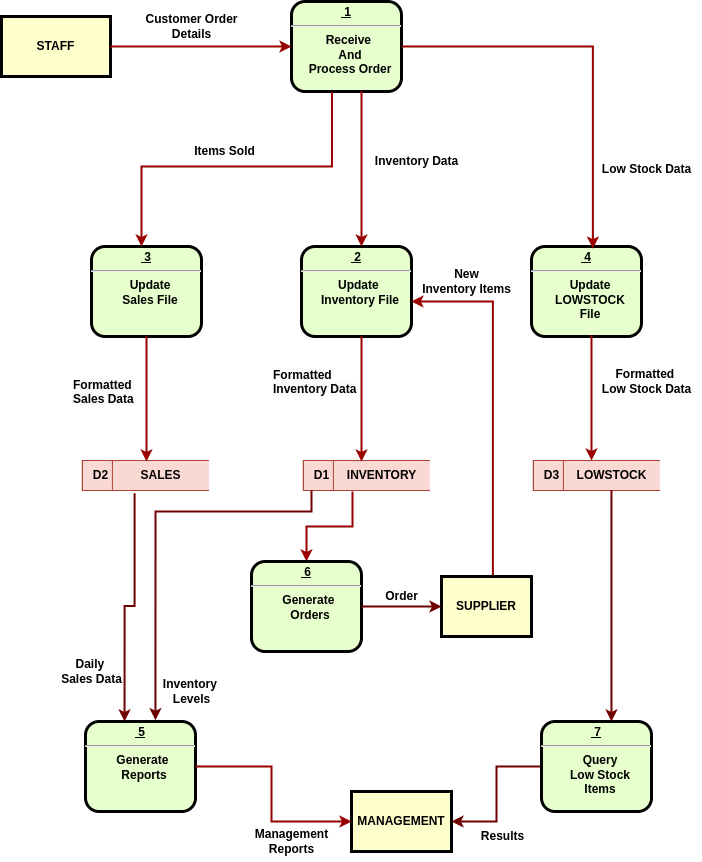
 **ITEM ID**: a reference to the ID of the item in the INVENTORY FILE

 **DATETIME**: time and date the item is added

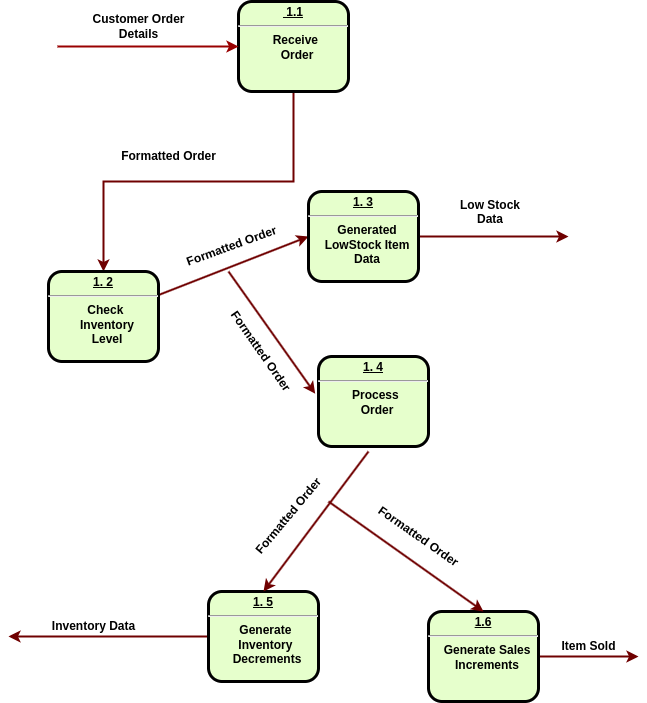
**3.2 Process Specification**

**3.2.1 Required Logical Flow Diagram**

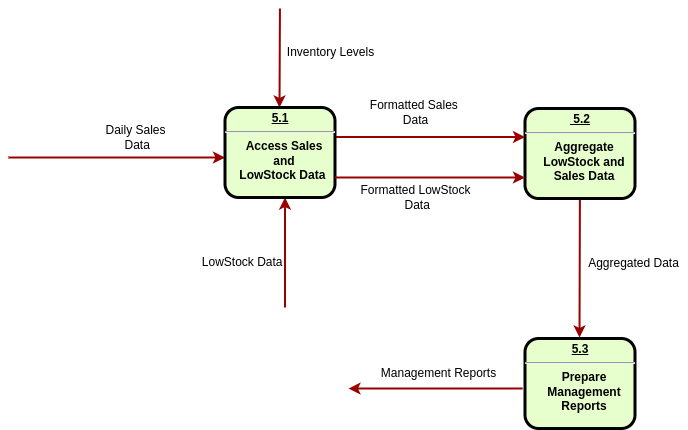
**LOGICAL LEVEL 0 DFD OF THE PROPOSED SYSTEM**

****

**LOGICAL LEVEL 1 DFD OF THE PROPOSED SYSTEM - PROCESS 1**

****

**LOGICAL LEVEL 1 DFD OF THE PROPOSED SYSTEM - PROCESS 5**

****

**3.2.2 Data Dictionary**

****

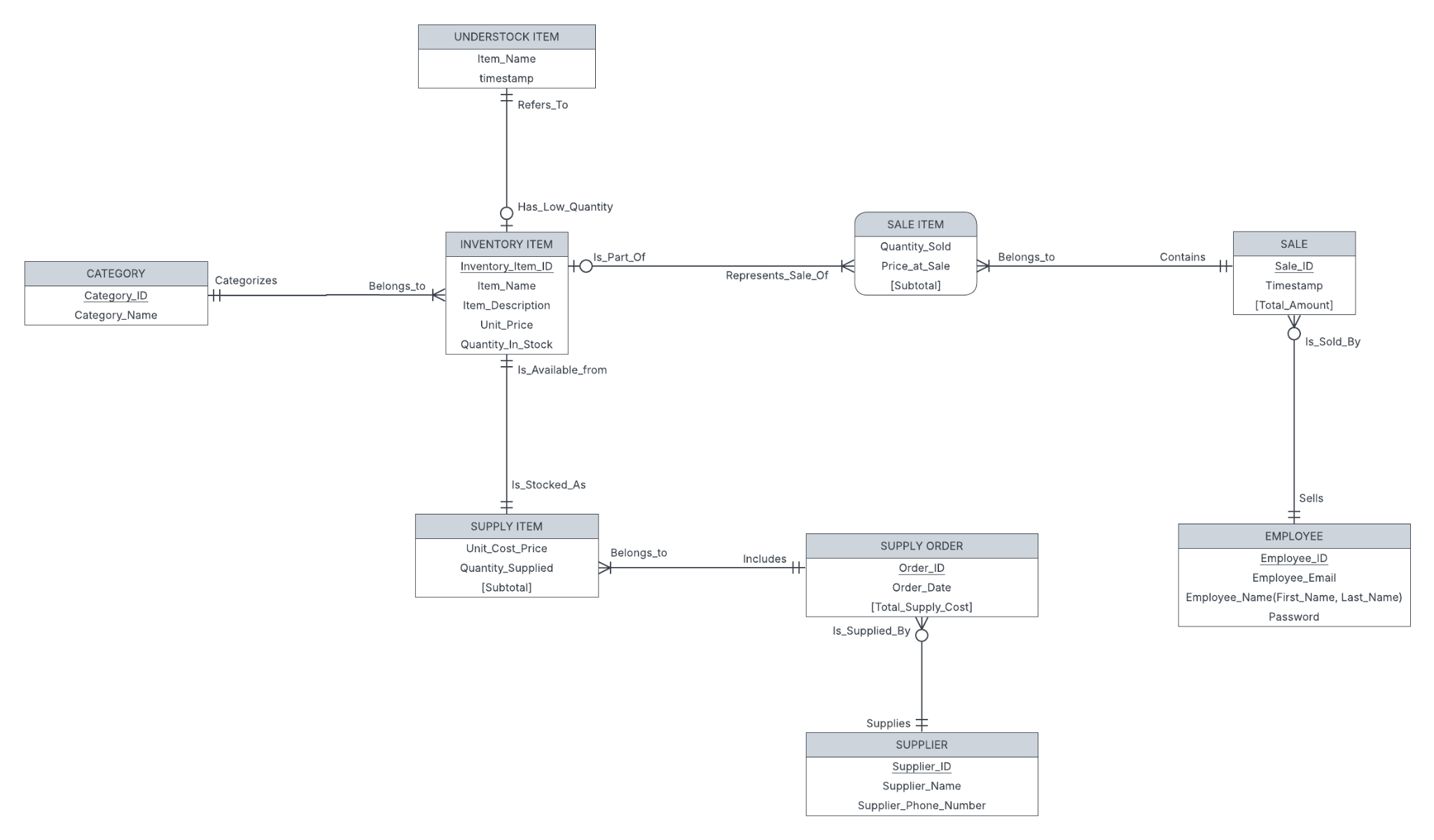
**3.2.3 Decision Tables**



**3.3 Data Specification**

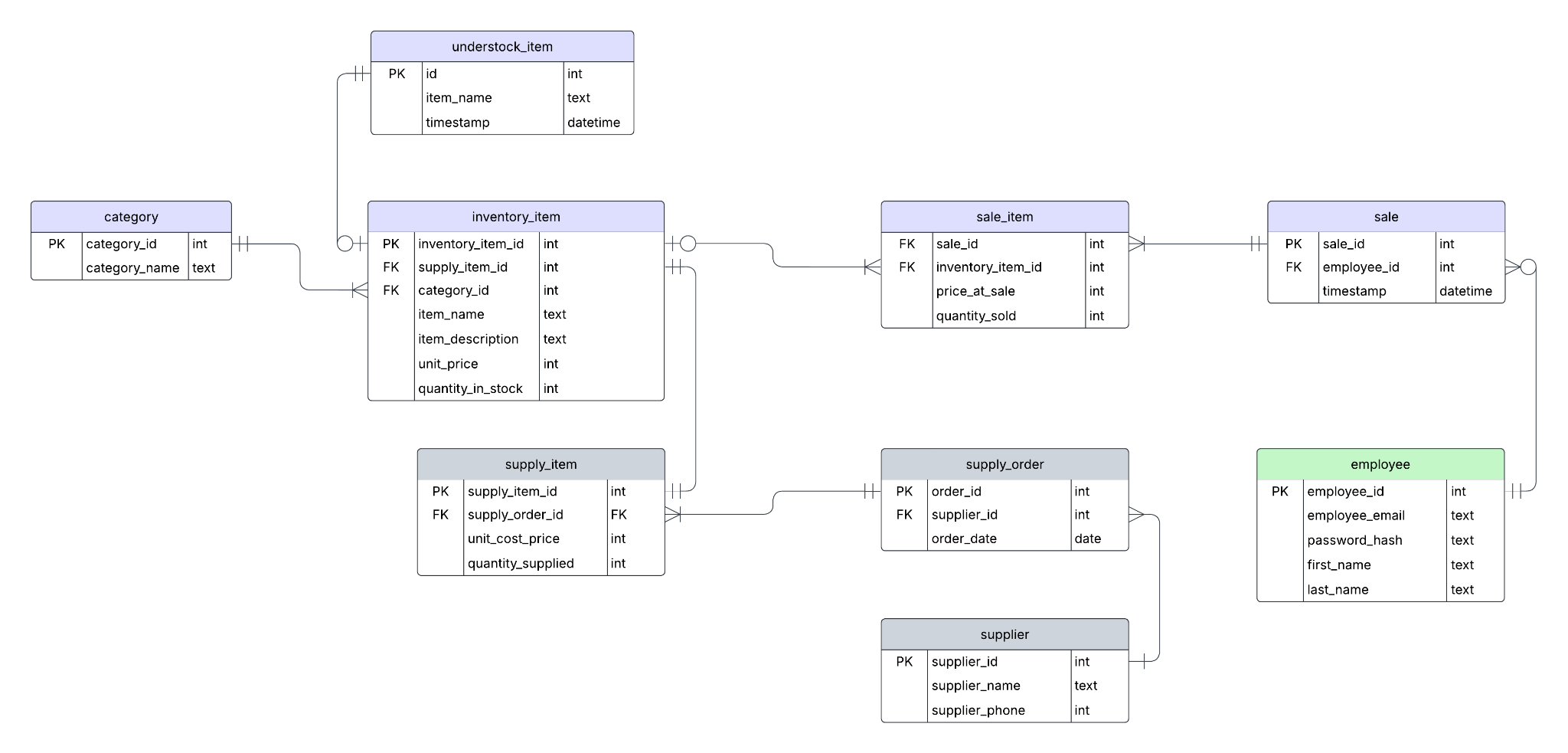
**3.3.1 Entity Relationship diagrams (ERD)**

[Live link](https://lucid.app/lucidchart/135308f0-b39d-4ce6-89c1-5b198ee0a189/edit?page=0_0&invitationId=inv_b71b354b-42d0-49dd-86b0-e5a0a02571fa#)

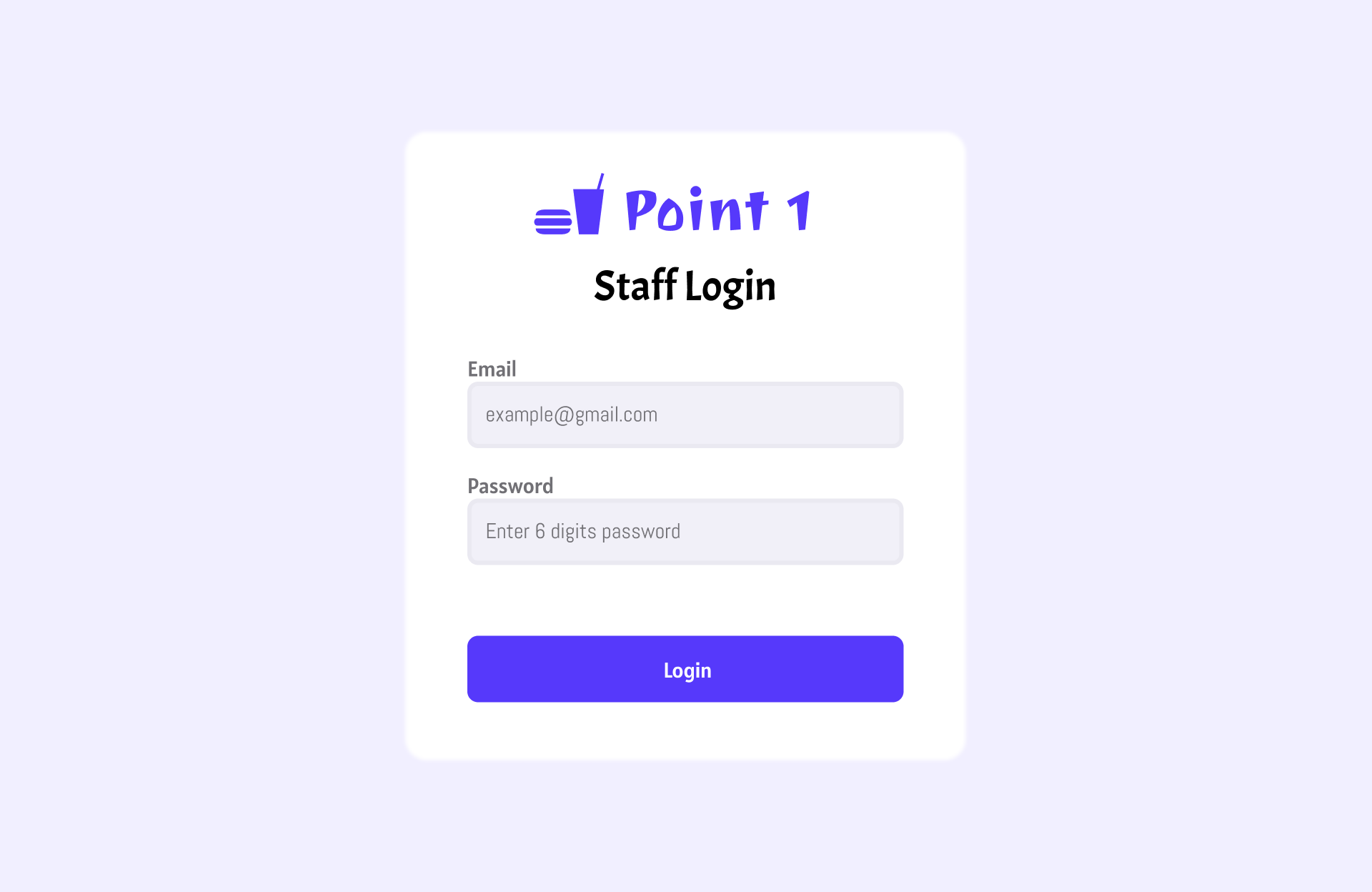
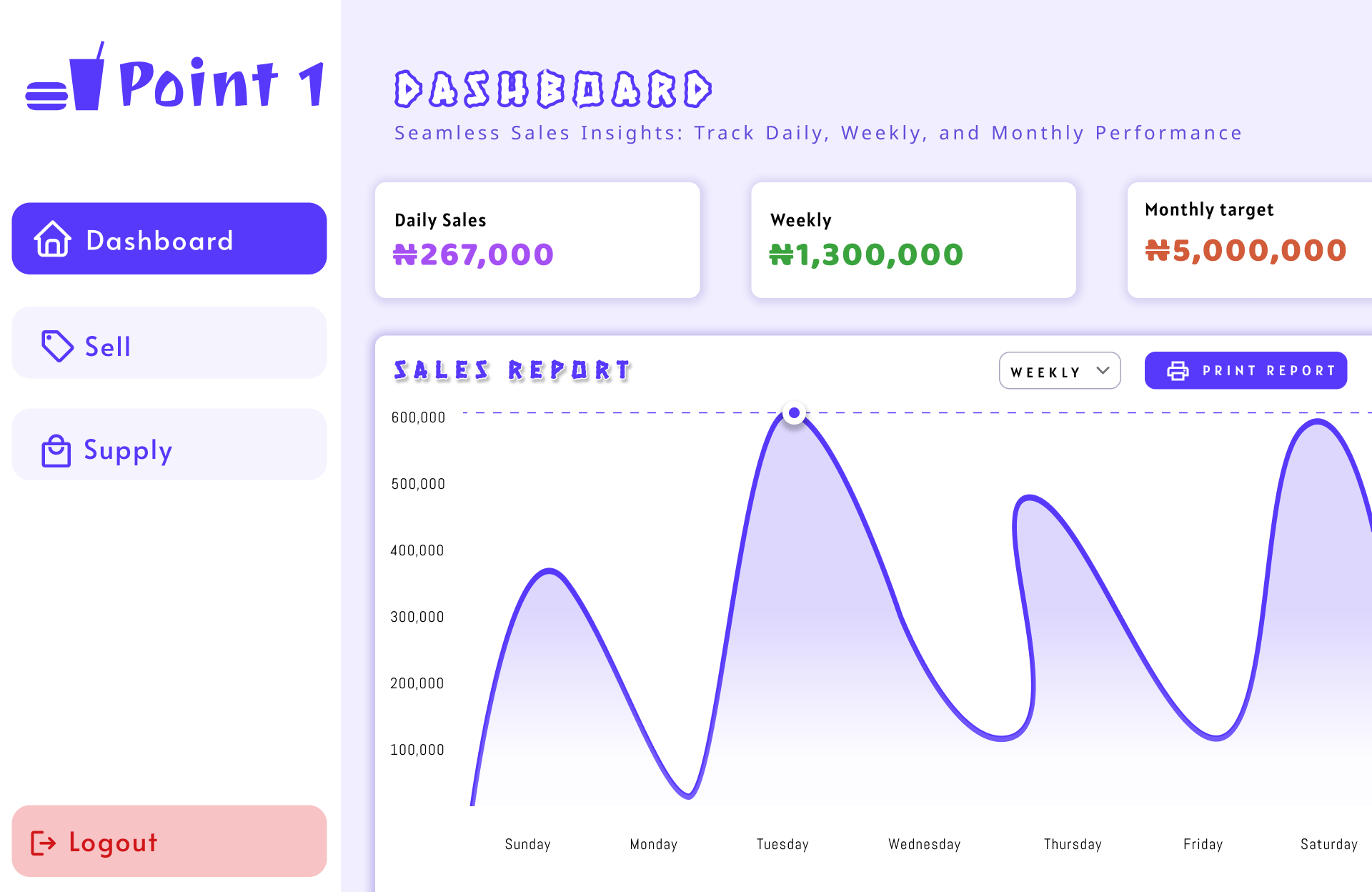
****

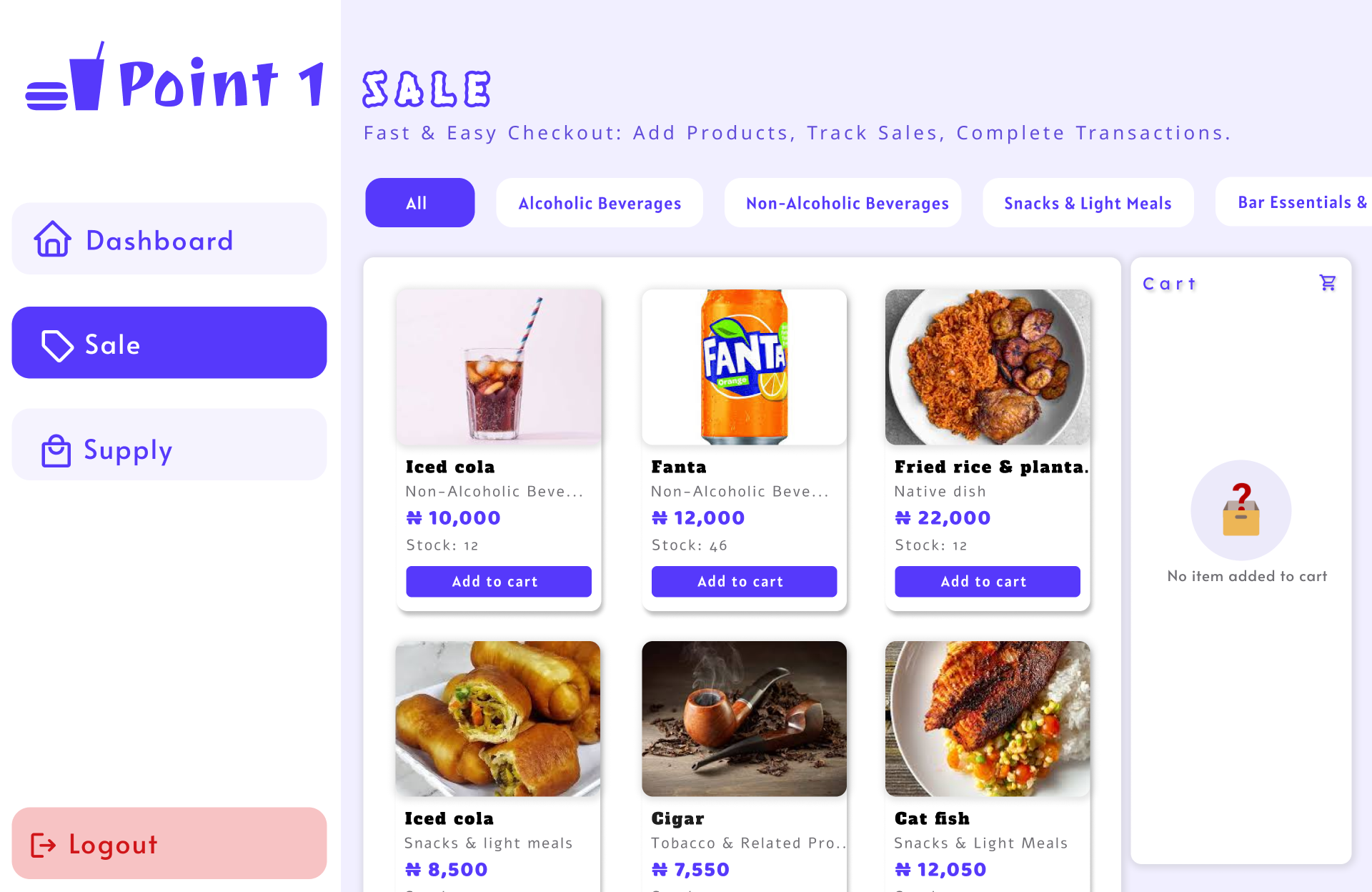
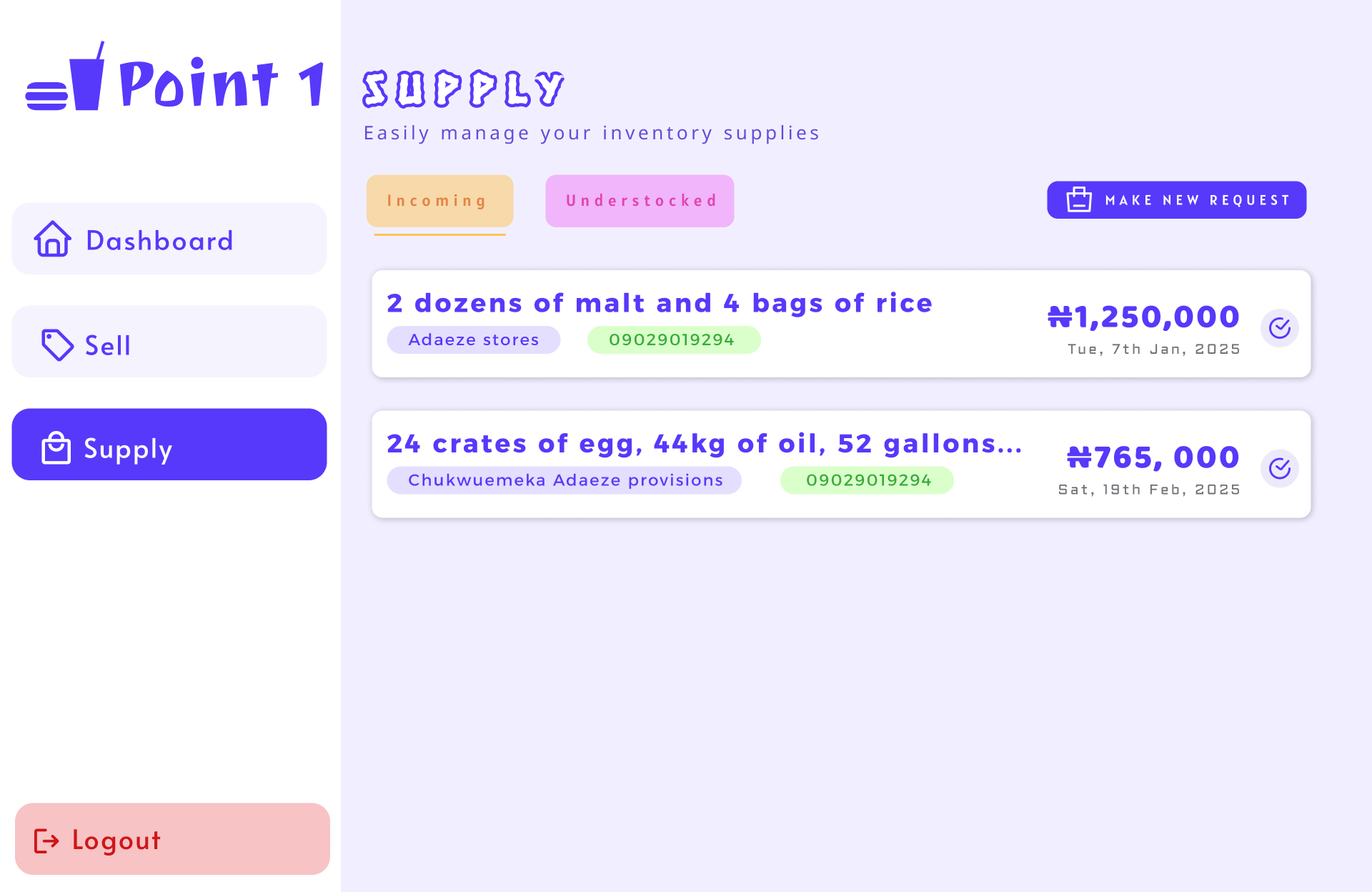
**3.3.2 Database Design**

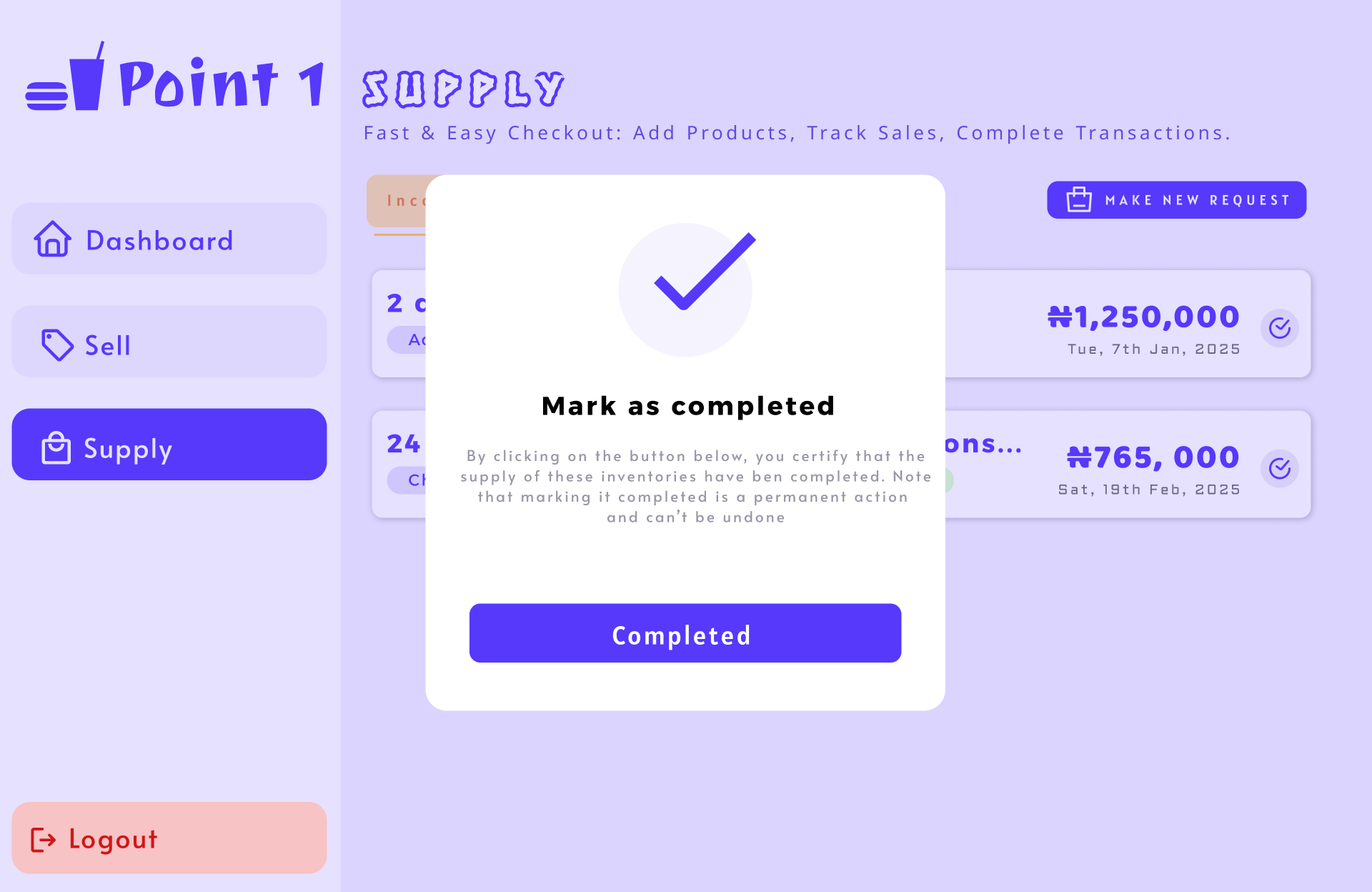
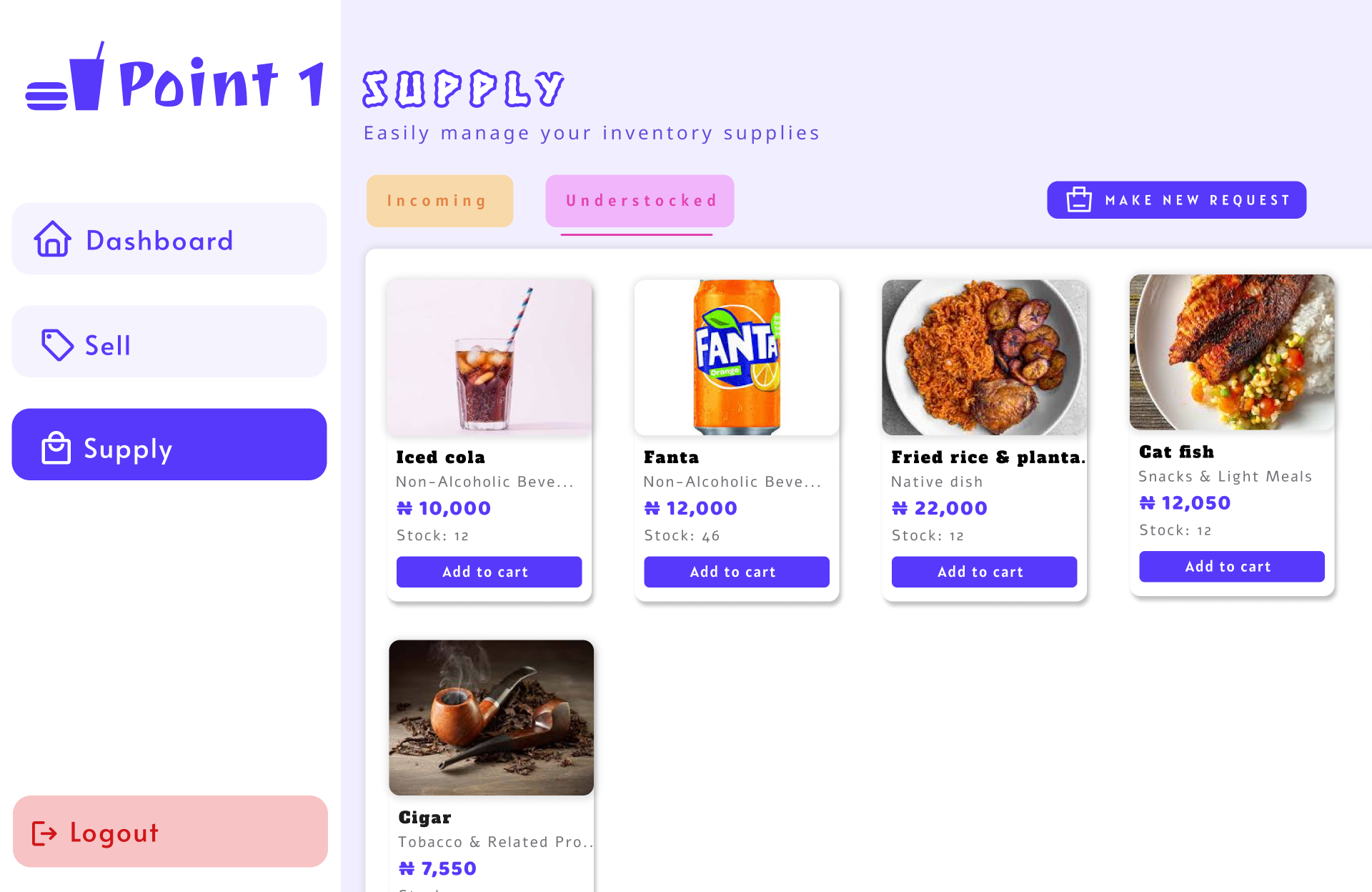
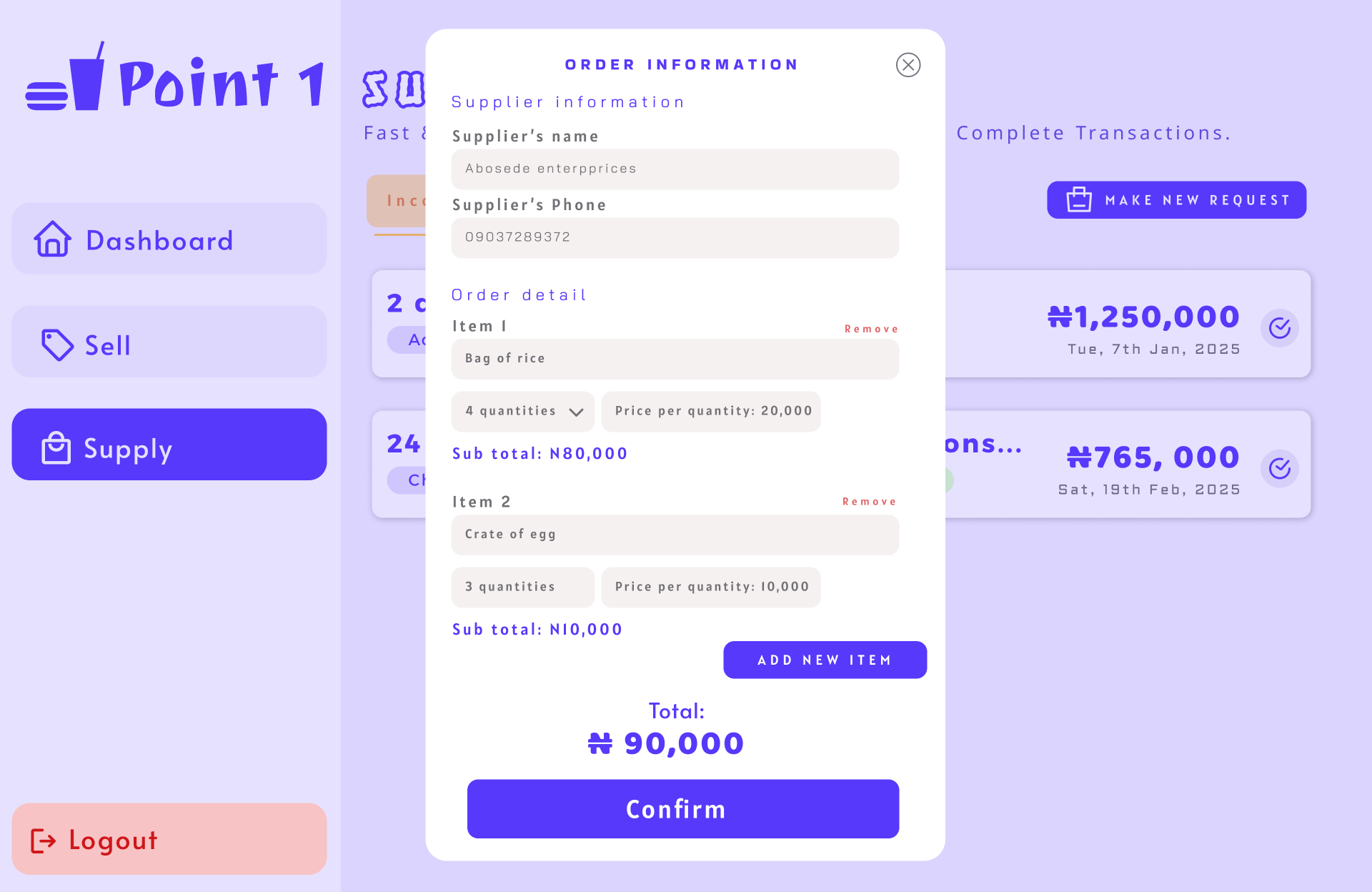
[Live Link](https://lucid.app/lucidchart/2ecf7327-fd0d-4ce9-9b8f-7cc04de9a4e5/edit?page=0_0&invitationId=inv_76b60b4e-b8c6-4c47-9118-3fede52e07bc#)



**3.4 Screen Layout and Specification**

** **

** **

** **

**3.5 Report/Form Specifications**

****

****

# **3.6 Program/Module Specification**

The proposed Inventory Management System (IMS) consists of the following modules:

### 1. The Ordering Module

* **Functionality:** Handles how staff process customer orders.
* **Procedure:**
  1. The staff logs into the system.
  2. The staff selects the items ordered by the customer.
  3. The staff confirms the order and generates a receipt.
  4. The system updates the Sales File and Inventory File automatically.

### 2. Inventory Module

* **Functionality:** Tracks stock levels, updates supply records, and flags low-stock items.
* **Procedure:**
  1. When a new supply arrives, staff enters the details into the system.
  2. The Inventory File is updated.
  3. If an item is low in stock, it is recorded in the Depleted File.

### 4. The Report Generating Module

* **Functionality:** Generates sales reports weekly or monthly and identifies best-selling items.
* **Procedure:**
  1. The system aggregates data from the Sales File and Inventory File.
  2. Generates reports in PDF format.
  3. Sends reports to the owner’s email automatically.

# **3.7 Test Plans**

**1. Functional Testing**

We would give the system to the management of Point 1 to verify that all core functions perform as expected.

Key Tests:

1. Adding, editing, and deleting inventory items.
2. Processing sales transactions and updating stock levels.
3. Triggering reorder alerts when stock falls below the set threshold.
4. Generating inventory and sales reports.
5. Validating supplier and purchase order integrations.

**2. Integration Testing**

We will ensure that various modules and external systems work seamlessly together.

Key Tests:

1. Data flow between the inventory module and the sales module.
2. Integration with the supplier management system.
3. Consistency of data during transactions across modules

**CHAPTER 4**

**Conclusion**

The new Inventory management system provides a more efficient way to automate stock tracking, improve sales monitoring, and enhance business efficiency. By replacing the existing system, the new system ensures real-time inventory updates, automated reporting, and better accountability.The system also enhances security by restricting access to authorized personnel only.

Upon completion, the IMS is expected to:

* Reduce human errors in inventory tracking.
* Improve efficiency in sales and stock management.
* Provide real-time data for decision-making.
* Automate report generation.

The system will undergo testing and evaluation to ensure it meets the intended functionality.

**APPENDICES**

[Project workbook](https://drive.google.com/drive/folders/1kWTO1nuUbcrJnyD7E3k_3hJaAFQ8ojW0?usp=sharing)

[Meeting Minutes](https://drive.google.com/drive/u/3/folders/1ZXXmcDTZCNyUHCRBfgqcS3fyDuZsq5Xw)