

# SYNOPSIS

## ON

### STOCK-MATE(INVENTORY- MANAGEMENT)



#### Submitted-By:

Bikash Kumar Giri

Bhagyalaxmi Palai

Banitarani Das

Abhipsa Dasadhikari

#### Guided-By:

Prof.Samir Kumar Patro

Promod Kumar Jena

## UNITED SCHOOL OF BUSINESS MANAGEMENT

37/A, Infocity Ave, Chandaka Industrial Estate, Patia, Bhubaneswar, Odisha 751024

January,2025

## OVERVIEW:

StockMate is an innovative SaaS platform built using the powerful MERN stack (MongoDB, Express.js, React, Node.js). Designed specifically for inventory owners, it offers an intuitive and efficient solution to streamline stock management.

With StockMate, users can minimize stock wastage, track inventory trends, and make data-driven decisions. The platform adapts to businesses of all sizes, from small retailers to large-scale warehouses.

## MODULES:

- 1.Add-Product Module
- 2.Add-Purchase Module
- 3.Add-Sale Module
- 4.Add-Store Module
5. AvatarMenu Module
6. MultiSelect –Menu Module
7. UpdateProduct- Module
8. UpdatePurchase –Module
9. UpdateSale- Module

## Add-Product Module:

- State Management: Use to manage the visibility of the modal and the product data.
- Form Handling: Captures input changes and updates the state accordingly.

- **Submit Function:** Sends the product data to the server via an API call, handles success and error responses, and refreshes the product list.
- **Modal Toggle:** Shows and hides the modal based on user interactions.

### Add-Purchase Module:

It captures the product name, supplier, quantity, and price, and sends this data to the server when the form is submitted.

- **State Management:** We'll add state to manage the purchase details.
- **Form Handling:** We'll create a form to capture purchase details.
- **Submit Function:** We'll send the purchase data to the server.

### Add-Sale Module:

A sale module in your system would allow users to record and manage sales transactions. It would capture details such as the product sold, quantity, sale price, customer information, and the date of sale.

This data would then be sent to your server for storage and further processing.

Essentially, the sale module helps track and document every sale made, ensuring accurate inventory management and sales reporting.

### Add-Store Module:

A store module in your system would be designed to manage different store locations, capturing details such as store name, address, contact information, and inventory specific to each location. It would allow for the organization and tracking of products, stock levels, and sales data across multiple stores, facilitating better inventory management and reporting.

This module ensures seamless coordination between various store locations, making it easier to monitor and manage the overall supply chain.

### AvtarMenu Module:

An AvatarMenu module is designed to provide a user interface element where users can interact with their profile avatar.

This module typically includes functionalities such as viewing and editing profile information, accessing user settings, and performing account-related actions.

It enhances user experience by offering a convenient, visually appealing way to manage personal preferences and account details.

### MultiSelect-Menu Module:

A MultiSelect-Menu module allows users to select multiple options from a dropdown menu or list.

It enhances user interaction by providing a way to choose multiple items simultaneously, which is useful in scenarios like selecting tags, categories, or multiple preferences.

This module typically includes features like searchability, checkboxes, and clear selection indicators to improve usability and user experience.

### UpdateProduct- Module:

The Module is designed to allow users to modify existing product information within the inventory system.

It provides an interface where users can edit details such as product name, manufacturer, stock levels, and other attributes.

This ensures that the inventory remains accurate and up-to-date, reflecting any changes or corrections made to the product data.

## UpdatePurchase –Module:

This Module allows users to modify existing purchase records within the system.

It provides an interface for editing details such as product name, supplier, quantity, and purchase price.

This module ensures that purchase records are accurate and up-to-date, reflecting any necessary corrections or changes.

This functionality is essential for maintaining precise procurement data and facilitating effective inventory management.

## UpdateSale-Module:

This Module enables users to edit existing sales transactions within the system. It provides an interface for modifying details such as product sold, quantity, sale price, customer information, and sale date.

It ensures that sales records are accurate and current, reflecting any necessary adjustments or corrections.

This functionality is crucial for maintaining precise sales data, aiding in better sales tracking and reporting.

## Software-requirements:

### Frontend:

1. React.js: For building the user interface.
2. Tailwind CSS: For styling the components.
3. Axios: For making API requests.

### Backend:

1. Node.js: For server-side logic.
2. Express.js: For creating the server and handling routes.
3. MongoDB: For database management.
4. Mongoose: For object data modeling (ODM) with MongoDB.

### Additional Tools:

1. Postman: For API testing.
2. Git: For version control.
3. npm: For managing dependencies.

### Development Environment:

1. Code Editor: Such as Visual Studio Code.

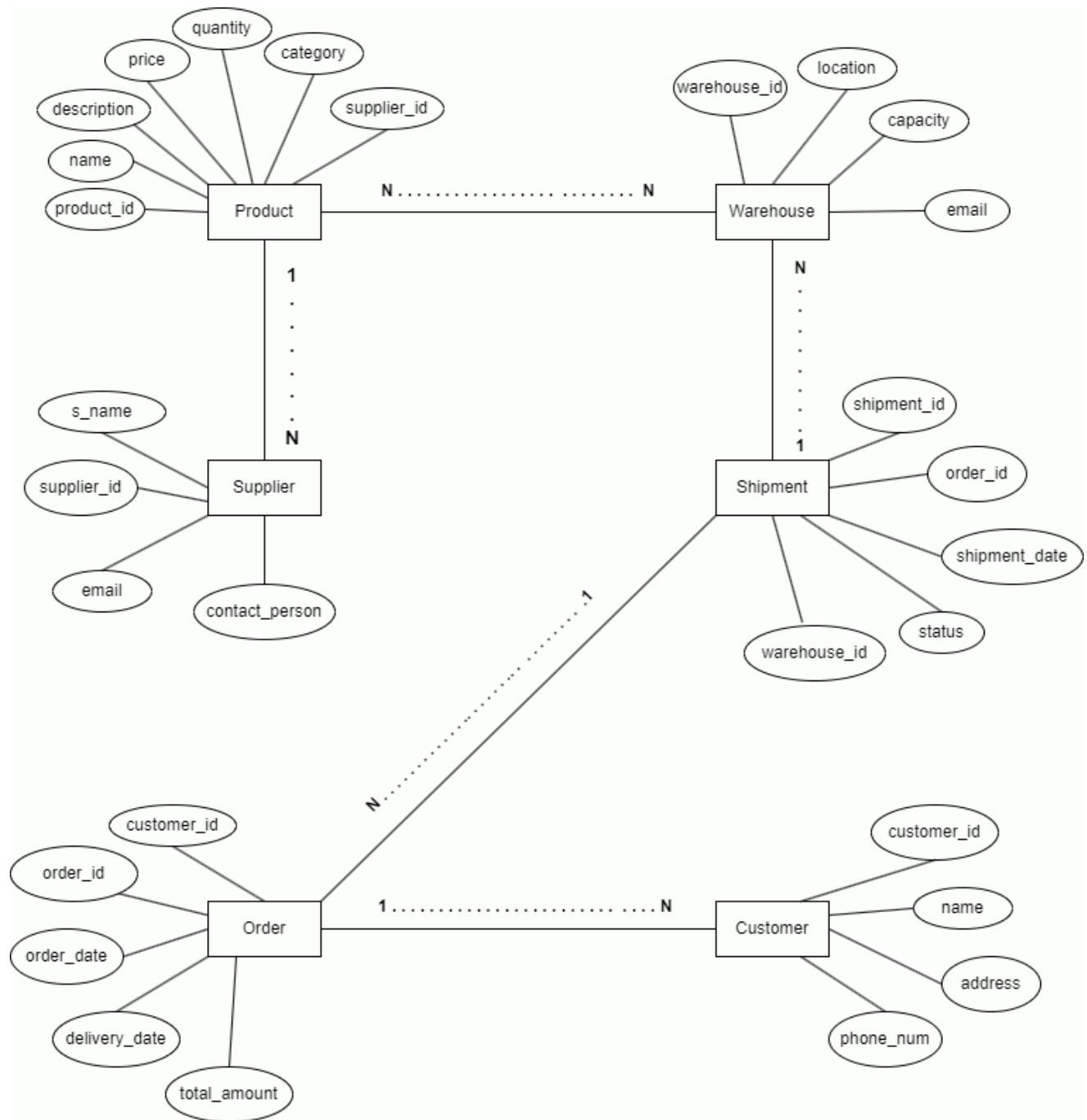
### Hardware-Requirements:

Processor: Multi-core processor

RAM: At least 8 GB of RAM, but 16 GB or more is recommended for smoother performance, especially when running multiple development tools simultaneously.

Storage: SSD with at least 256 GB of storage to ensure fast read/write speeds. More storage may be needed depending on the size of your projects and data.

## ER-DIAGRAM



## DFD-Diagram:

DFDs are used to visually represent the flow of data within a system, showcasing how inputs are transformed into outputs through various processes.

### Components:

Processes: Represent the actions taken to transform data.

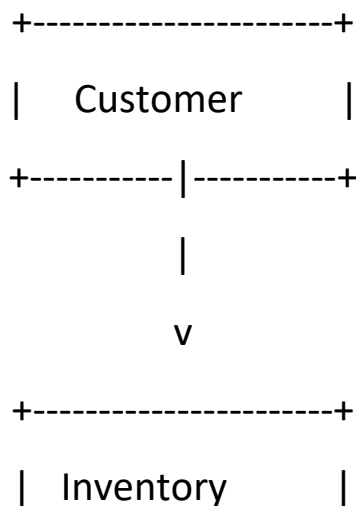
Data Stores: Where data is stored within the system.

External Entities: Sources or destinations of data outside the system.

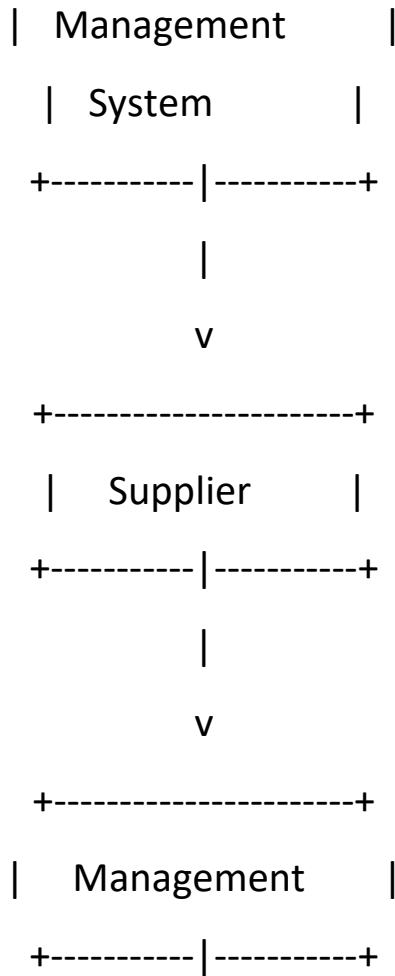
Data Flows: Arrows that indicate the movement of data between entities, processes, and data stores.

DFDs are a helpful tool for understanding and documenting the functional aspects of a system.

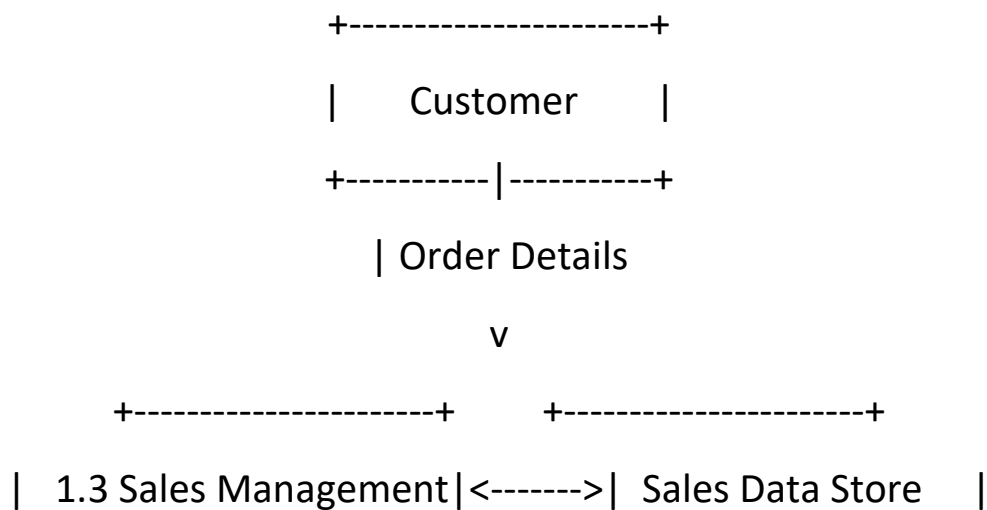
### Level 0 DFD for Inventory Management System:

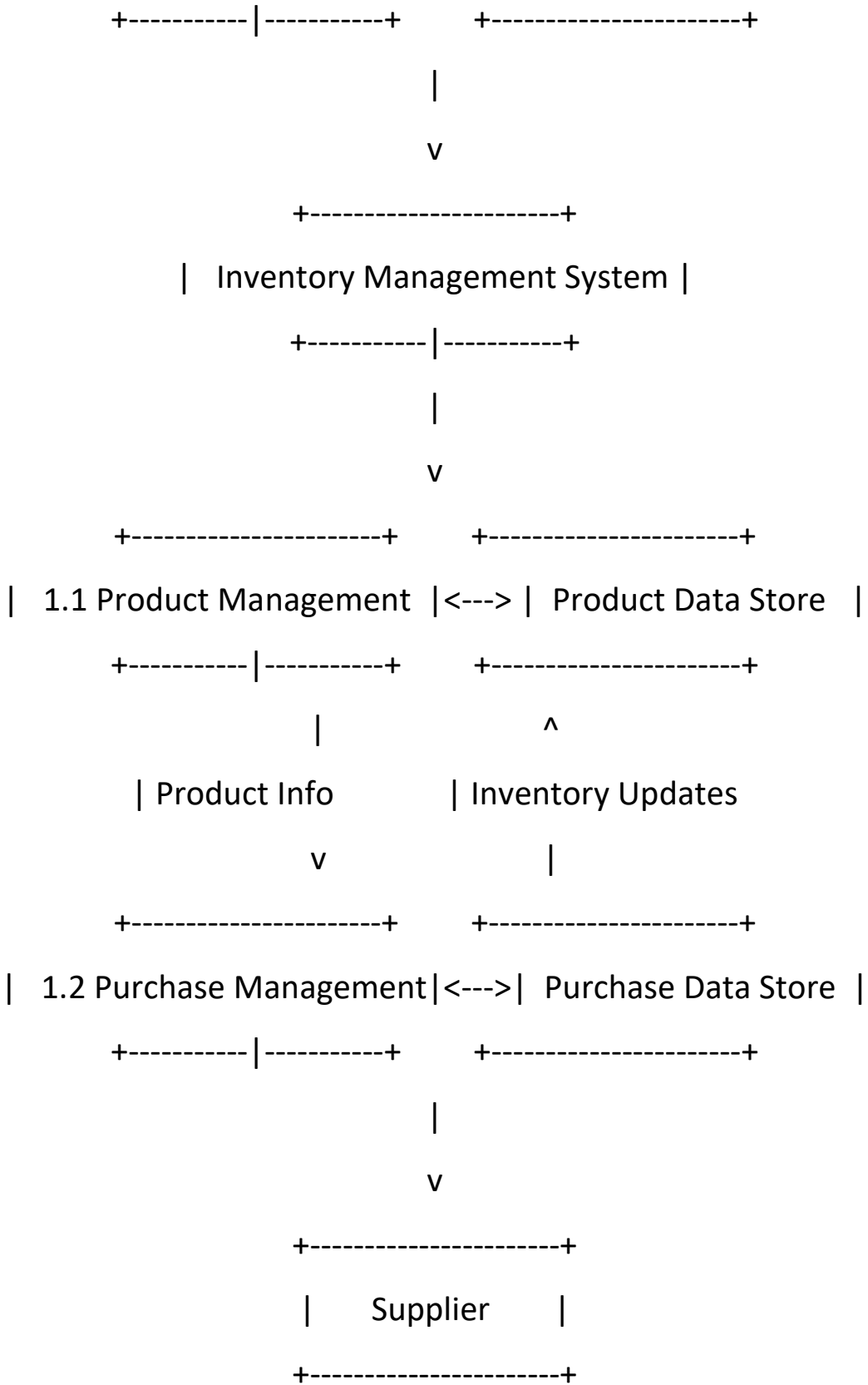






Level 1 DFD for Inventory Management System:





|

v

+-----+

| Management |

+-----|-----+

| Report Requests

v

+-----+

| 1.4 Inventory Reporting |

+-----|-----+

| Reports

v

+-----+

| Management |

+-----+

### Future Scope:

Expanding an Inventory Management System holds a lot of potential to evolve into an even more robust and intelligent solution.

By incorporating these innovations, an Inventory Management System can become a powerful tool for businesses, helping them operate and more .

### Conclusion:

An Inventory Management is essential for efficiently managing stock, purchases, sales, and supplier relations. By leveraging advanced technologies and planning for future enhancements, businesses can streamline their operations, reduce costs, and improve overall efficiency.

### Bibliography:

<https://www.w3schools.com/REACT/DEFAULT.ASP>

<https://www.w3schools.com/nodejs/>

<https://www.w3schools.com/mongodb/>

<https://www.w3schools.com/express/>