Milestone 1 – Group 7

Project Name:

Inventory Management System

Problem Statement:

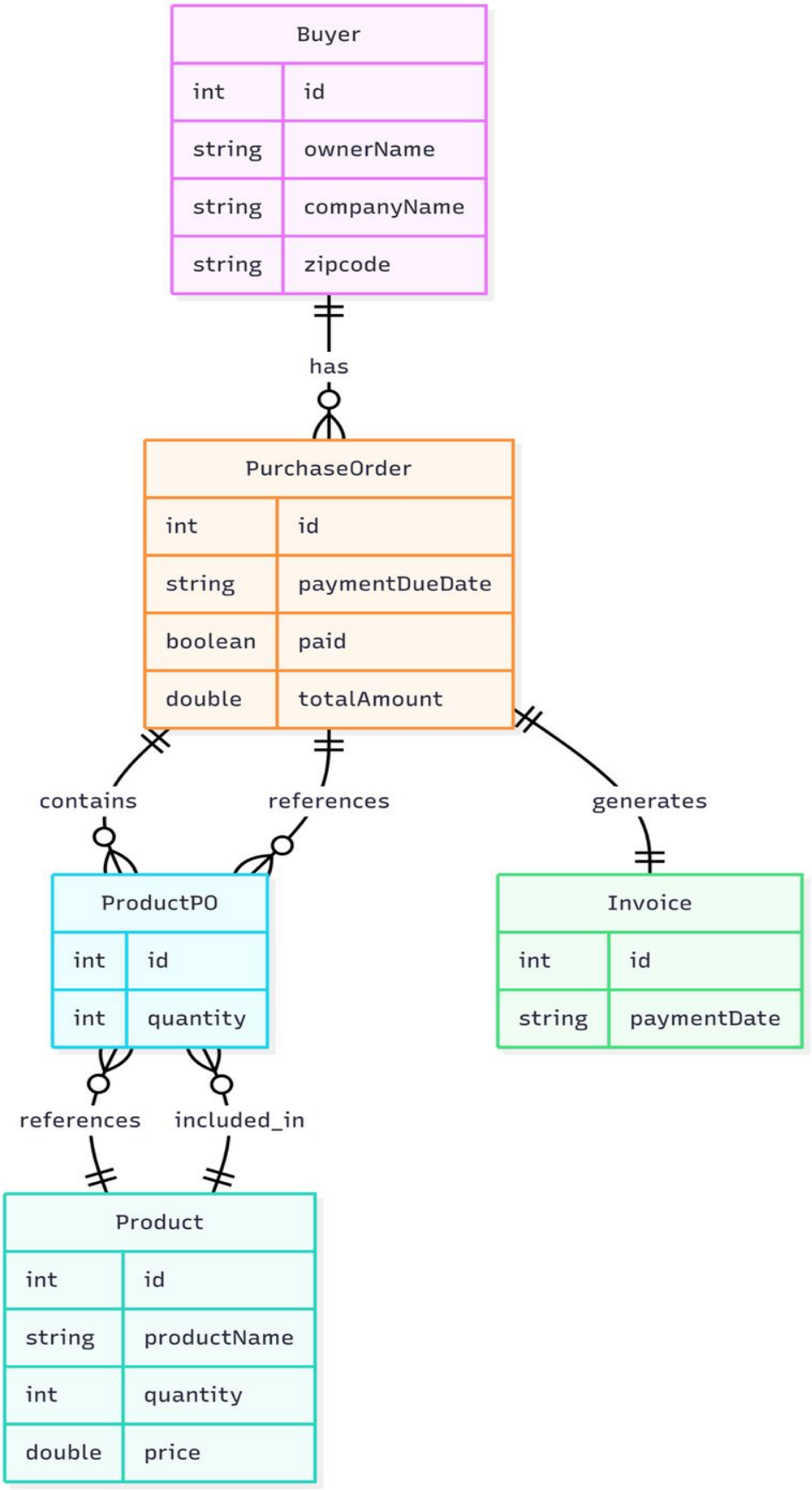
In today's fast-paced business environment, organizations must efficiently manage their inventory to ensure smooth operations, minimize costs, and meet customer demands. Traditional manual inventory management methods are often error-prone, time-consuming, and lack real-time visibility, leading to issues such as stockouts, overstocking, delayed order processing, and inaccurate record-keeping. These inefficiencies can result in lost sales, increased operational costs, and poor customer satisfaction.

To address these challenges, the Inventory Management System provides a comprehensive, web-based solution that automates and integrates all aspects of inventory control and related business operations. The system allows users to seamlessly add, update, and track products, manage buyer and employee records, process purchase orders, and generate invoices from a unified interface. Features like real-time synchronization, PDF invoice generation, low-stock alerts, and secure authentication improve operational efficiency and ensure better decision-making.

Design Patterns to be Implemented:

Design Pattern	Purpose / Use Case
Command	Encapsulate tasks as objects, useful for undo/redo operations or action queuing
Decorator	Dynamically add features to products or carts (e.g., discount feature)
Facade	Provide unified access to subsystems (e.g., PDF generation, messaging)
Factory	Create various product or communication objects flexibly
Observer	Implement stock alert system and buyer notifications
State	Manage inventory status (in-stock, low-stock, out-of-stock) with state-specific behavior
Strategy	Allow different strategies for buyer, employee, product, invoice handling

UML Diagram:


Buyer
int
id
string
ownerName
string
companyName
string
zipcode
has
PurchaseOrder
int
id
string
paymentDueDate
boolean
paid
double
totalAmount
contains
references
generates
ProductPO
int
id
int
quantity
Invoice
int
id
string
paymentDate
references
included_in
Product
int
id
string
productName
int
quantity
double
price

Tech Stack

- **Backend:** Java, Spring Boot
- **Frontend:** React.js
- **Database:** MySQL or MongoDB
- **Authentication:** Spring Security, JWT
- **PDF Generation:** iTextPDF
- **Email Notifications:** JavaMailSender
- **Tools:** Docker, Swagger, Postman, GitHub

Functionalities Planned for Milestone 2

Category	Functionality
User Authentication	Secure login, JWT-based auth, role-based access (Admin, Employee)
CRUD Operations	Manage Products, Buyers, Employees
Inventory & Orders	Stock tracking, purchase order creation & updates
PDF & Email	Generate PDF invoices, send order/stock email alerts
Dashboards	Role-specific dashboard with summary info
API & Testing	Swagger API documentation, unit & integration testing

Contributions

Team Member	Contribution
Rudra Patel	Wrote the problem statement, finalized design pattern mapping to use cases, and formatted the milestone documentation.
Janaki Rama Raju Vadapalli	Designed and created the UML/ER diagram based on the database schema and relationships.
Smit Patel	Organized the tech stack section and listed out planned functionalities for Milestone 2.
Omkar Nate	Defined and validated the entity relationships in code, ensured ER diagram compliance.
Dhruv Baraiya	Compiled team contributions and verified milestone formatting.