Team Work : Elkham , Yernur

SE-2439

2. Architecture & Design – Pet Store

2.1 System Architecture

The Pet Store project is implemented as a monolithic web application, where all business logic is concentrated in one Go backend service.

Client (Web Browser)

↓ HTTP (REST API)

Go Backend (Monolith)

↓

Relational Database (PostgreSQL / MySQL)

Components

Client (Browser)

Sends requests to view animals, feeds, and accessories, create orders, register, and authenticate. Receives responses in JSON format.

Go Backend (Monolith)

-Implements the REST API

-Manages users, products, and orders

-Handles business logic: authentication, CRUD operations, order and inventory management

The database (PostgreSQL/MySQL)

-Stores users, products, orders, and categories.

-Maintains data integrity and relationships between entities.

2.2 Use-Case Diagram

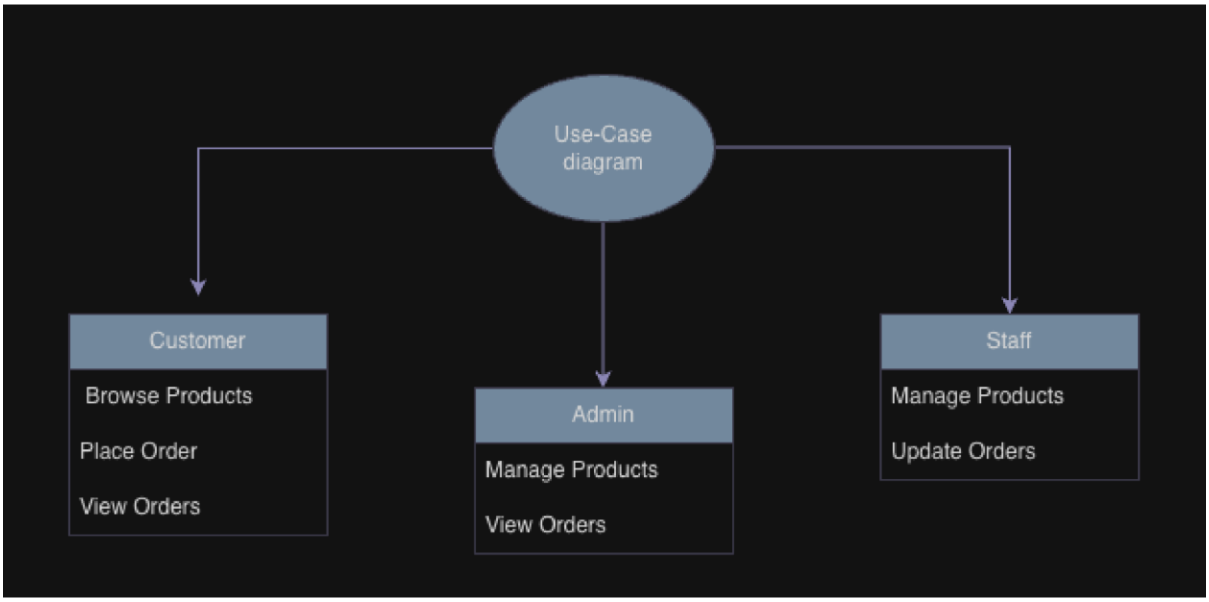
Customer – the buyer

Admin – owner or general manager

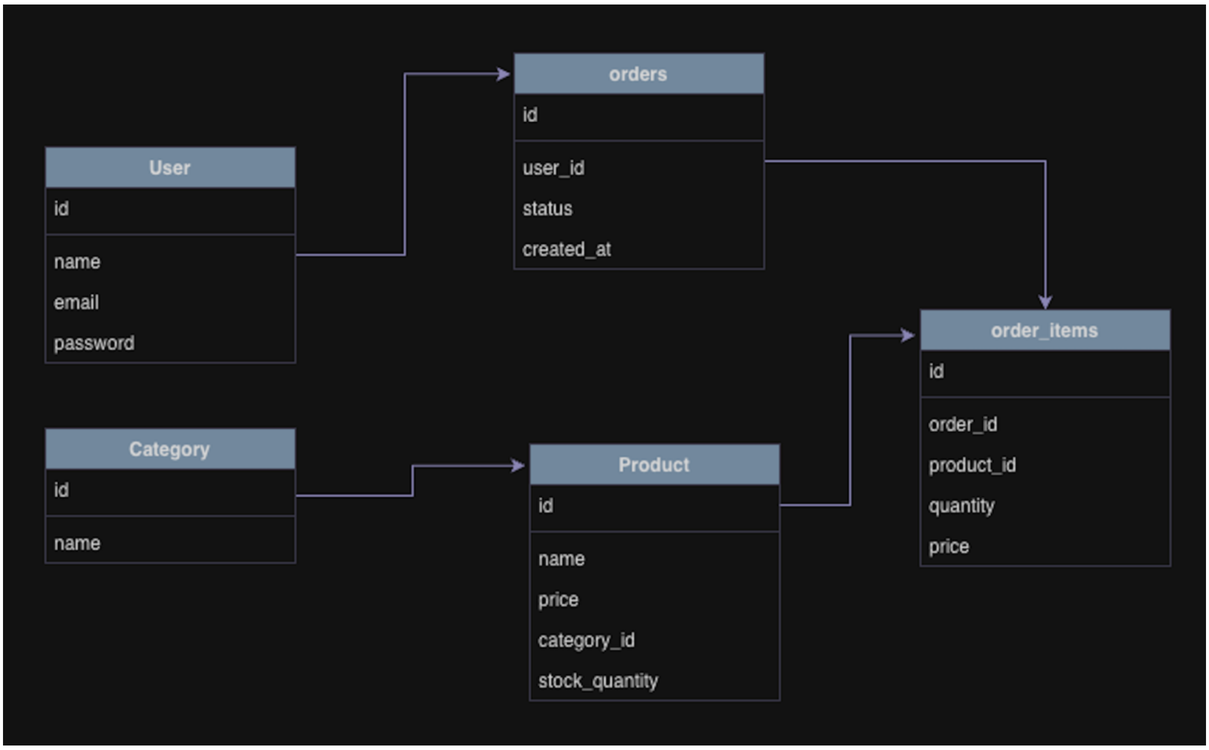
Staff – Administrative Assistant

Use Cases

- Customer: Browse Products, Place Order, View Orders
- Admin: Manage Products, View Orders
- Staff: Manage Products, Update Orders



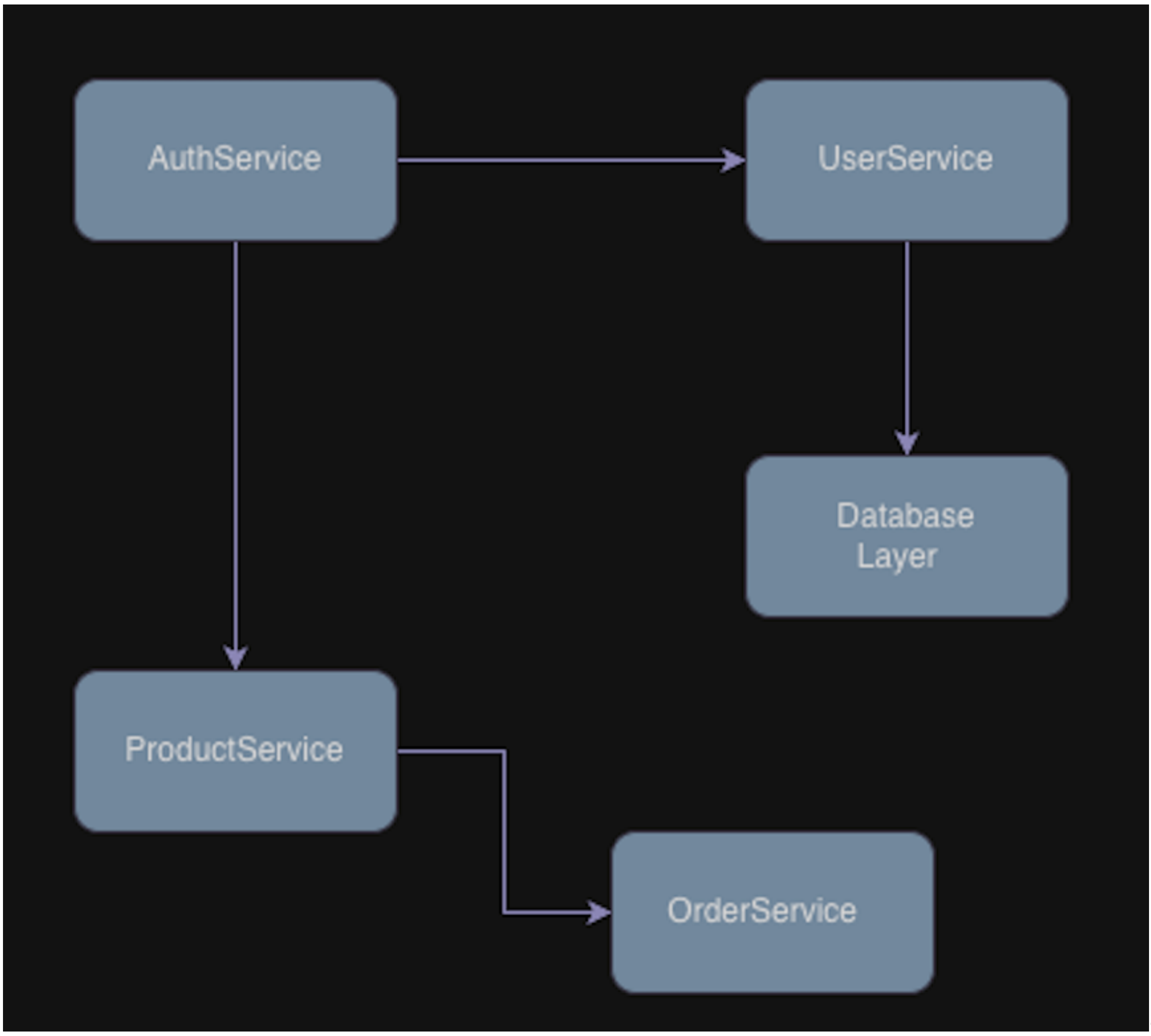
2.3 ERD (Entity Relationship Diagram)



- users (id, name, email, password)
- categories (id, name)
- products (id, name, price, category_id, stock_quantity)
- orders (id, user_id, status, created_at)
- order_items (id, order_id, product_id, quantity, price)

- User → Orders (1-to-many)
- Orders → Order_items (1-to-many)
- Product → Order_items (1-to-many)
- Category → Product (1-to-many)

2.4 UML Diagram



e

2.5 Modules & Responsibilities

Module: Auth

Responsibility: User authentication, login

Module: User

Responsibility: User profile management, role handling

Module: Product

Responsibility: CRUD operations for products, categories, and animals

Module: Order

Responsibility: Order creation, order management, order items

Module: Database

Responsibility: Database access and query execution