



**Faculty of Engineering & Technology**  
**Electrical & Computer Engineering Department**  
**Comp 333, Data Base Systems**  
**Phase 2 – Requirements Document & ERD**  
**Pets & things Store Management System**

---

**Prepared by:**

**Name:** Yahya Sarhan

**Number:** 1221858

**Name:** Mohammad Abu Adiya

**Number:** 1220270

**Instructor:** Bassem Sayrafi

**Section:** sec 2

**Date:** 11/18/2025

## Table Of Contents

Project Scope and Description .....	3
Client Information.....	5
System Requirements and Entities.....	6
1.System Modules.....	6
2. Main Entities .....	7
3. ER Diagram .....	8
4. Relationships Between Entities.....	9
Technologies to Be Used .....	10
Sample Queries .....	11
Summary .....	12

## Project Scope and Description

The *Pets & Things Store Management System* is a desktop application designed for **Pets & Things Pet Store**, located in *Ein Musbah, Ramallah, Palestine*.

The store has two floors:

- The first floor sells products for both cats and dogs, such as food, treats, toys, and accessories.
- The second floor provides a cat hotel service, where customers can book rooms for their cats for a specific period of time. Each room in the hotel can accommodate **more than one cat**, depending on their **size** and **comfort level**.

The system will include a warehouse management feature that tracks how many units of each product are stored in the warehouse. When the store purchases products from suppliers, the quantities are added to the warehouse. When sales occur or shelves are restocked, the warehouse stock decreases automatically. This helps the store maintain accurate inventory, prevent out-of-stock situations, and view low-stock alerts.

The store is managed by **Mr. Mohammad Sarhan**, who runs the business with one employee in the morning and Owner in the evening. Currently, most tasks like recording sales, purchases, inventory, and bookings are done manually, which takes time and can cause mistakes. This project will create a simple digital system that helps manage everything in one place.

The system will help the store:

- Handle product sales and purchases
- Manage warehouse inventory and supplier records
- Manage cat hotel bookings and room availability
- Store customer, employee, and pet information
- Generate useful reports and summaries

## Smart Feature Ideas

Area	Smart Feature Ideas
Reports	<ul style="list-style-type: none"><li>- Total sales per day, week, or month</li><li>- Best-selling products</li><li>- Total revenue from the hotel</li><li>- Product reorder suggestions</li></ul>
Inventory Automation	<ul style="list-style-type: none"><li>- Automatically reduce product quantity after a sale</li><li>- Notification for “Low stock” (e.g., when quantity &lt; 10)</li></ul>
User Roles	<ul style="list-style-type: none"><li>- Employee login (cashier, manager)</li><li>- Different permissions (manager can edit, cashier can only sell)</li></ul>
Pet Tracking & Booking Details	<ul style="list-style-type: none"><li>- When a customer books a hotel room for their cat, they must provide:<ul style="list-style-type: none"><li>• Cat name, age, and breed</li><li>• Food preferences and special care needs</li><li>• Customer contact and emergency phone number</li></ul></li><li>- Track the number of cats currently staying in the hotel</li></ul>
Booking Management	<ul style="list-style-type: none"><li>- View all bookings for a specific date</li><li>- Show which rooms are occupied or free</li><li>- Prevent double booking</li></ul>
Printable Documents	<ul style="list-style-type: none"><li>- Print invoices for product sales</li><li>- Print booking confirmations for hotel guests</li></ul>
Search & Filters	<ul style="list-style-type: none"><li>- Search for products by name or category</li><li>- Filter bookings by date or customer</li></ul>

## Bonus Features

These extra features can make the system more advanced and professional if time allows:

- Dashboard page showing total sales, available rooms, and low-stock alerts
- Search bar to quickly find customers or products
- Login and password system for security
- Charts and graphs for reports (sales trends, booking activity)
- Email or message notification for booking confirmations or reminders

The *Pets & Things Store Management System* will be developed as a **desktop application using Python and MySQL**, with a simple interface that both employees can use. It will improve daily work, save time, and make managing the business easier and more reliable.

## Client Information

**Client Name:** Mohammad Sarhan

**Business Name:** Pets & Things Store

**Location:** Ein Musbah, Ramallah, Palestine

**Contact Information:**

Phone: 0599597959

Email: petstore.pal@gmail.com

**Business Description:**

Pets & Things Pet Store is a medium-sized pet shop located in Ein Musbah, Ramallah. The store sells a wide range of products for both cats and dogs, including dry food, wet food, treats, toys, accessories, and litter boxes. It also offers a cat hotel service on the second floor, where customers can book rooms for their cats for short or long stays.

The store also has a small warehouse inside the same building where extra product stock is stored. When new products arrive from suppliers, they are first placed in the warehouse before being moved to the shelves. This warehouse helps the owner keep backup stock, restock products during the day, and avoid running out of popular items. The business is managed by Mr. Mohammad Sarhan, who works the evening shift, while one employee covers the morning shift. The store aims to provide quality pet products and comfortable hotel services for cats while maintaining good customer relationships.

# System Requirements and Entities

## 1.System Modules

The *Pets & Things Store Management System* will include several main modules. Each module will handle a specific area of the store's daily operations.

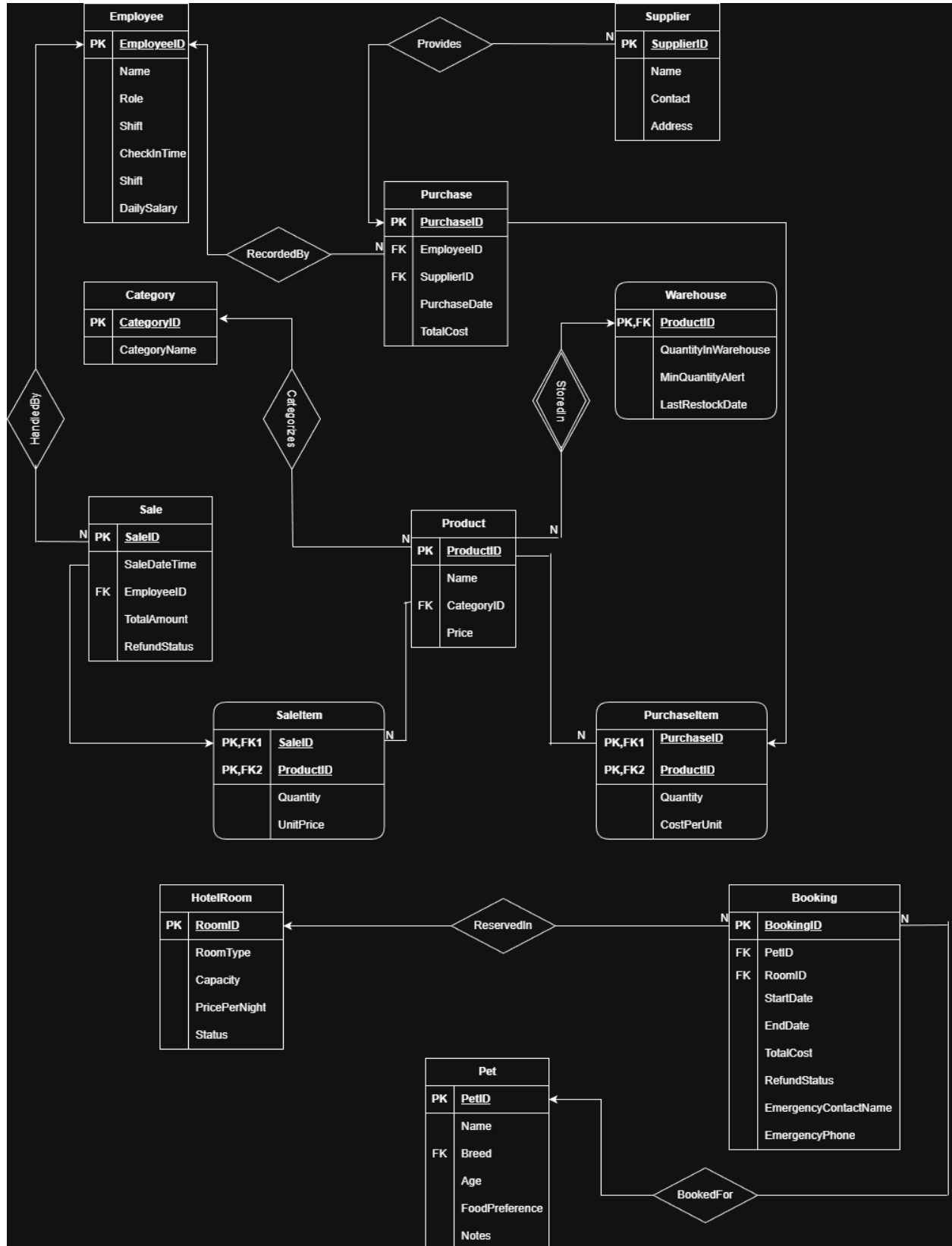
Module	Description
<b>Sales Management</b>	Records product sales, tracks daily revenue, and updates stock automatically when a sale is made. It also allows refunds for returned products and adjusts stock accordingly.
<b>Purchases Management</b>	Tracks products purchased from suppliers and updates the warehouse stock.
<b>Warehouse Management</b>	The warehouse module manages all product stock stored in the warehouse. When the store purchases items from suppliers, the products are added to the warehouse. When employees restock the shelves or make sales, quantities are reduced from warehouse inventory. This module helps track available stock, storage capacity, restocking history, and prevents running out of essential products. The warehouse is the central source of truth for all product quantities in the store.
<b>Cat Hotel Management</b>	Handles room bookings, pet and customer details, check-in and check-out times, and automatically calculates the total amount due when a customer checks out. If a booking is cancelled before the cat stays any night, the system allows a full refund.
<b>Employee Management</b>	Stores employee information such as name, role, and work shift. The system also tracks employee check-in and check-out times and calculates their daily salary based on their working hours.
<b>Reports and Statistics</b>	Generates reports such as total sales, hotel revenue, best-selling products, and stock alerts.

## 2. Main Entities

The database will include several entities that represent different parts of the store and its operations. Each entity will have its own attributes, and relationships will connect them (for example, one pet can have many bookings).

Entity	Description	Example Attributes
<b>Employee</b>	Staff members who manage sales or hotel services. The system stores their personal details, shift times, and daily salary.	EmployeeID, Name, Role, CheckInTime, CheckOutTime, HoursWorked, DailySalary
<b>Supplier</b>	Companies that supply products to the store.	SupplierID, Name, Contact, Address
<b>Product</b>	Items available for sale.	ProductID, Name, CategoryID, Price, Quantity
<b>Category</b>	Product type classification (e.g., food, accessories).	CategoryID, CategoryName
<b>Warehouse</b>	This entity tracks inventory levels for each product, updating quantities with purchases and sales.	WarehouseID, Location, Capacity, ProductID, Quantity, CategoryID
<b>Sale</b>	Sales transactions recorded in the system, including refund status if a product is returned	SaleID, Date, TotalAmount, RefundStatus
<b>SaleItem</b>	Details of each product sold in a sale.	SaleID, ProductID, Quantity, UnitPrice
<b>Purchase</b>	Purchases from suppliers.	PurchaseID, SupplierID, Date, TotalCost
<b>PurchaseItem</b>	Details of products purchased from suppliers.	PurchaseID, ProductID, Quantity, CostPerUnit
<b>HotelRoom</b>	Rooms available for the cat hotel. Each room can host more than one cat, depending on their size and comfort.	RoomID, RoomType, Capacity, PricePerNight, Status
<b>Pet</b>	Information about customer pets.	PetID, Name, Breed, Age, FoodPreference
<b>Booking</b>	Cat hotel bookings made by customers. The system records all booking details and allows refunds if the cat does not stay any night.	BookingID, PetID, RoomID, StartDate, EndDate, TotalCost, RefundStatus, EmergencyContact

### 3. ER Diagram





## 4. Relationships Between Entities

The ERD includes the following key relationships:

### Category — Product

A Category categorizes many Products (1-to-Many).

This relationship is implemented through the foreign key Product.CategoryID.

### Product — Warehouse (*Weak Relationship*)

Each Product has one warehouse stock record (1-to-1).

Warehouse.ProductID serves as both the primary key and foreign key, making Warehouse a weak entity.

### Supplier — Purchase

A Supplier *provides* many Purchases (1-to-Many).

Implemented via Purchase.SupplierID.

### Employee — Purchase / Employee — Sale

An Employee may record many purchases and handle many sales (1-to-Many).

Implemented using EmployeeID as a foreign key in both Purchase and Sale.

### Purchase — PurchaseItem — Product (*Associative Weak Entity*)

Purchase and Product have a many-to-many relationship resolved through the weak entity PurchaseItem.

PurchaseItem contains the composite primary key (PurchaseID, ProductID).

### Sale — SaleItem — Product (*Associative Weak Entity*)

Sale and Product also form a many-to-many relationship.

SaleItem links them using the composite key (SaleID, ProductID).

### HotelRoom — Booking

A hotel room may have multiple bookings over time (1-to-Many).

Booking.RoomID represents this relationship.

### Pet — Booking

A pet may have many bookings (1-to-Many).

Implemented via Booking.PetID.

## Technologies to Be Used

The *Pets & Things Store Management System* will be developed as a **desktop application**. It will use modern tools and technologies that make the system efficient, easy to use, and suitable for managing store operations.

Component	Technology / Tool	Description
Database	MySQL	Used to store all data about products, sales, purchases, employees, and hotel bookings.
Programming Language	Python	The main language for building the system. It will be used to create the application logic and connect with the MySQL database.
GUI Framework	Tkinter	Provides a simple and user-friendly desktop interface for employees to interact with the system.
Database Design Tool	MySQL Workbench	Used to design the database schema, ER diagram, and relationships between tables.

## Sample Queries

### Sales & Revenue

1. Show total sales for a specific **day**.
2. Show total sales for each **week** (grouped by week).
3. Show total sales for each **month** (grouped by month).
4. List the **best-selling products** (by total quantity sold).
5. Show **total sales per product category**.
6. Show **days where sales exceeded** a chosen amount (e.g., 500 ILS).
7. List **refunded sales** with date, product(s), and amount.
8. Calculate **net sales** for a date range (**total sales – refunded amounts**).

### Hotel & Bookings

9. List **all available hotel rooms** (status = available).
10. Show **all bookings between two dates**.
11. Show **which rooms are occupied/free** on a chosen date.
12. List **cats currently staying** in the hotel **today**.
13. Show **total revenue from hotel bookings** for the current month.
14. Show the **total number of bookings per room** (room popularity).
15. List **refunded hotel bookings** (bookings cancelled before any night stayed), with reason/date/amount.
16. Calculate **net hotel revenue** for a date range (**booking charges – booking refunds**).

### Inventory & Purchases (Warehouse)

17. List all products in the “**Food**” category.
18. Show all products with **quantity < 10** (low-stock alert).
19. List products that were **restocked in the last 7 days** (from purchases).
20. Show **total purchase amount per supplier** (sum of purchases).
21. Show all products in the warehouse with `QuantityInStock < MinimumRequired` (low-stock alert).
22. List all products and their warehouse quantities, sorted by category
23. Show all items that were restocked this week (from purchases to warehouse).
24. Show warehouse capacity usage and remaining space.

## Summary

The *Pets & Things Store Management System* is designed to help **Pets & Things Pet Store** manage all its daily work in one easy system. It replaces manual records with a digital solution that saves time, reduces errors, and improves organization.

The system will manage product sales, purchases, warehouse inventory, hotel bookings, and employee records. It also supports **refunds** for returned products or cancelled hotel bookings. Employees will be able to check in and check out, and the system will calculate their **daily salary** automatically.

Reports will help the owner understand how the business is performing, including total sales, hotel revenue, best-selling products, and stock levels. The system also keeps track of low-stock products, pet information, and booking details such as customer contact and emergency numbers.

This project will be developed as a **desktop application using Python and MySQL**, providing a simple and friendly interface for both employees. In later phases, the system will be expanded with a detailed **ER diagram**, a **database design**, and a **working prototype** that connects all modules together.

The final goal of the project is to give the store owner, **Mr. Mohammad Sarhan**, a complete and reliable management system that makes running the store easier, faster, and more accurate every day.