ACM 321 Project Report: Inventory Management System

# Team Information

* **Section:** 3
* **Team Name:** InvenTech
* **Team Members:**
  + Rauf Kutay Akyıldız - Frontend and GUI Developer
  + Eren Acar - Backend and Database Developer

# Project Overview

## Objective

The objective of our work is to design an Inventory Management System for a kitchenware (Züccaciye) shop. With this system the store can control its products, sales and suppliers better and in a more organised and efficient way. It will enable the store to keep up stock levels, process sales very quickly and maintain reliable inventory information. Furthermore, the system will facilitate to the users to handle orders and products, which would make the shop more efficient to run

## Store Type

Our team has chosen a kitchenware store for this project. We believe that such a store is ideal for an inventory management system as it has a wide selection of products, such as cookware, dishware, and kitchen tools. This range of goods enables us to introduce various inventory categories (e.g., utensils, appliance, decor). In addition, a kitchenware store must offer excellent goods flow management so that the product is all the time available to customers, and therefore, this project is a good example of an Inventory Management System as it can function within an actual store.

# Design and Architecture

## System Architecture

Our inventory management system includes three basic operators for small retail kitchenware: managing inventory, sales and customer relations. The system also includes three basic components:

### **2.1.1. Graphical User Interface (GUI):** Built using Java Swing for user-friendly interaction.

### **2.1.2. Database Management:** SQLite is a relational database, and for reliable data storage.

### **2.1.3**. **Business Logic Layer:** Patients at the minimum gateway of standard functionalities, including CRUD functionalities, inventory control and sales control.

## Class Diagram

The class diagram depicts the main classes representing the system functionalites i.e.

**Customer:** Manages customer data, including name, address, and city.

**Product:** Stores product information (code, description, category, and price).

**Category:** Represents product categories.

**Invoice:** Handles invoice generation and customer payments.

**ListOfItems:** Tracks the products and quantities associated with each invoice.

**Class Relationships:**

* Customer is associated with Invoice via a one-to-many relationship.
* Product relates to Category to organize items.
* The invoice is linked to ListOfItems (list of purchased goods or products) and number of items purchased.

## Database Schema

The database schema is organized in a way that it allows to stage, and thus correctly associate, data for materials in stock, customers and sales. Following is a schema description for the corresponding diagram.

### **2.3.1 CustomerTable:**

* **CustomerID (integer, primary key):** Unique identifier for each customer.
* **CustomerName (text):** Name of the customer.
* **CustomerAddress (text):** Address of the customer.
* **CustomerCity (text):** City where the customer resides.
* **CustomerCounty (text):** County of the customer.

### **2.3.2 CategoryTable:**

* **CategoryID (integer, primary key):** Unique identifier for each category.
* **CategoryName (text):** Name of the product category.

### **2.3.3 ProductTable:**

* **ProductCode (text, primary key):** Unique code for each product.
* **ProductDescription (text):** Description of the product.
* **CategoryName (text, foreign key):** Links the product to its category.
* **ProductPrice (decimal):** Price of the product.

### **2.3.4 InvoiceTable:**

**InvoiceID (integer, primary key):** Unique identifier for each invoice.

**CustomerID (integer, foreign key):** Links the invoice to the customer.

**Payment (decimal):** Total payment amount for the invoice.

### **2.3.5 ListOfItemsTable:**

* InvoiceID (integer, foreign key): Links the item list to an invoice.
* ProductCode (text, foreign key): Links the item to a product.
* Quantity (integer): Quantity of the product in the invoice.

Attached Diagram: The database schema is the same as the drawing presented and it describes both the object and relational aspects of the database model, such as the use of foreign key constraints. Every table is built so as to obtain the highest data retrieval speed, as well as so as to implement the business logic tier in an appropriate manner.

metin, ekran görüntüsü, multimedya yazılımı, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

# Features and Functionality

## Key Features

[List the main features implemented in your application, e.g.: - Adding,

updating, and deleting inventory items. - Importing and exporting inventory data. - Managing supplier and sales information.]

## Customization

[Describe the customizations made for your chosen store type and their impact on design.]

# Application Walkthrough

## GUI Overview

[Provide screenshots of your GUI with brief descriptions of the functionality of each screen.]

## Sample Workflow

[Explain a sample use case scenario, e.g., how a user would add a new inventory item.]

# Object-Oriented Principles

## Use of Classes and Objects

[Discuss how you applied classes and objects in your project.]

## Inheritance and Polymorphism

[Describe examples of inheritance and polymorphism used in your project.]

## Interfaces and Abstract Classes

[Explain how interfaces and abstract classes are used in your implementation.]

# Database Integration

## Database Operations

[Explain the CRUD operations implemented in your project.]

## Sample Queries

[Provide examples of SQL queries used in your application.]

# File I/O

## Import/Export Functionality

[Describe the file formats used and how data import/export is implemented.]

## Error Handling

[Discuss how your application handles errors during file operations.]

# Challenges and Solutions

[Describe the challenges your team faced during the project and how you addressed them.]

# Future Improvements

[List potential improvements or additional features that could be added to the project.]

# Conclusion

[Summarize your experience working on the project and the skills your team developed.]

# Appendix

## User Manual

[Provide a detailed user manual, including installation instructions and usage guidelines.]

## References

[List any references, tools, or resources used during the project.]