<u>LAHORE GARRISON UNIVERSITY</u>

<u>Group Members</u>

Ahmad Hassan (001)

Zohaib Hassan (045)

Naveed Tariq (027)

Course Name;

“Software Construction and Development”

Project Title:

“Product Inventory Manager”

Submitted To:

“Sir Farhan Sarwar”

Project Overview

The Product Inventory Manager is a desktop application developed in Python using Tkinter for the graphical user interface (GUI) and SQL Server for database management. This application allows users to add, update, delete, and view products in an organized inventory system. The system is designed for small businesses to manage product inventories efficiently, offering real-time interaction with the database through an intuitive GUI.

Features

1. Add Product

Users can add new products with a name and price.

2. Update Product

Users can select an existing product and update its details.

3. Delete Product

Users can remove a product from the inventory.

4. View Products

Products are displayed in a table with real-time updates.

5. User-Friendly Interface

Interactive GUI with intuitive controls.

6. Data Validation

Ensures all required fields are filled, and price is numeric.

Technology Stack

Component	***Technology/Tool***
Programming	Language Python 3.x
GUI Framework	Tkinter
Database	Microsoft SQL

Database Connector	Server pyodbc
OS Compatibility	Windows

System Requirements

- Python 3.x installed on the machine
- Tkinter library (comes with Python)
- pyodbc library (pip install pyodbc)
- Microsoft SQL Server with a database named billing and a table product

Database Design

The application uses a single table to store product details:

Table: Products

Column Name	**Data Type**	**Description**
Id		INT (Primary Key, Identity) Unique Product ID
Name	NVARCHAR (100)	Name of the product
Price	FLOAT	Price of the product

SQL Script to Create Table:

```
CREATE TABLE products (
id INT IDENTITY (1,1) PRIMARY KEY,
name NVARCHAR (100) NOT NULL,
price FLOAT NOT NULL
);
```

Application Workflow

1. Launch Application

The GUI window opens, displaying the product table and input fields.

2. Add Product

Enter product name and price, then click Add Product.

3. Update Product

Select a product from the table, modify the fields, then click Update Product.

4. Delete Product

Select a product from the table and click Delete Product.

5. View Products

All products are displayed in a scrollable table that updates in real time

Code Structure

The project is organized in a single Python file with the following sections:

1. Database Connection

2. def connect ():

3. return pyodbc.connect(

4. f"DRIVER={{ODBC Driver 17 for SQL Server}};

5." f"SERVER={SERVER}; DATABASE={DATABASE};Trusted Connection=yes;Encrypt=no;")

6.Establishes a connection to SQL Server using pyodbc.

7. CRUD Functions

o fetch_products()

Retrieves and displays all products.

o add_product()

Adds a new product to the database.

o update_product()

Updates selected product details.

o delete_product()

Deletes the selected product from the database.

8. GUI Components

o Frames: Input frame, Button frame, Table frame

o Widgets: Labels, Entry fields, Buttons, Treeview (table) o Styles: Custom fonts, colors, and row height for a modern interface

9. Event Handling

o select_row(event)

Populates the entry fields when a product row is clicked.

User Interface

The GUI is simple and intuitive:

- **Input Section:** Text fields to enter product name and price.

- **Button Section**: Buttons to Add, Update, Delete products.

- **Table Section**: Displays all products in a scrollable table.

- **Footer**: Shows project name and developer information.

Screenshot Example: (Include a screenshot of your running GUI here)

Error Handling

- Alerts user if required fields are empty.

- Alerts user if price is not numeric.

- Warns user when no product is selected for update or delete.

Future Enhancements

- Add search **functionality** for products.

- Implement **category-wise** product management.

- **Add export to Excel or PDF functionality.**

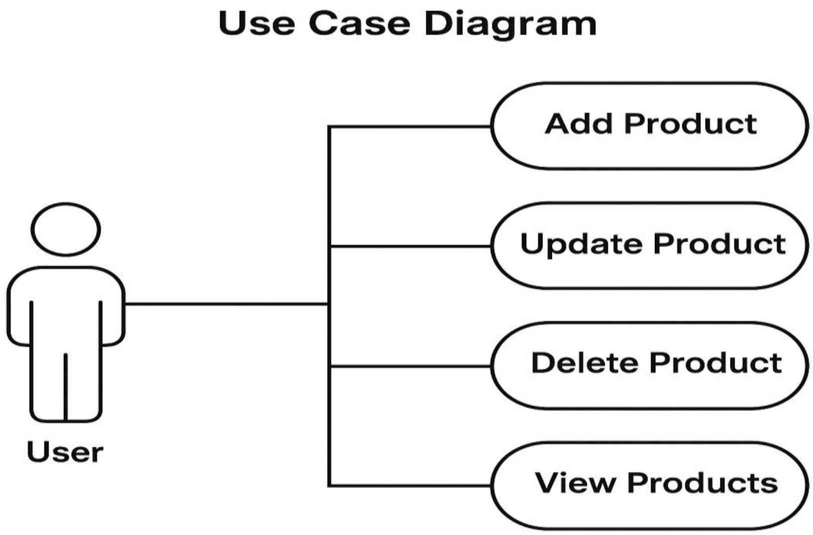
- Implement **user authentication** for multi-user access.

Conclusion

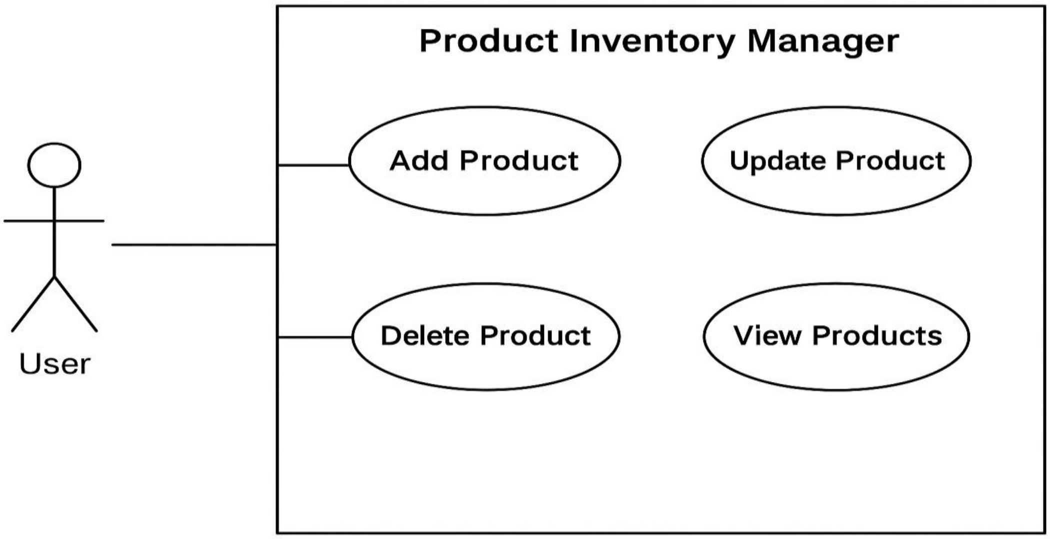
The **Product Inventory Manager** is a robust, user-friendly application that simplifies inventory management. It demonstrates practical use of **Python**, **Tkinter**, and **SQL Server**, integrating **GUI** development with database management effectively.

UML Diagram

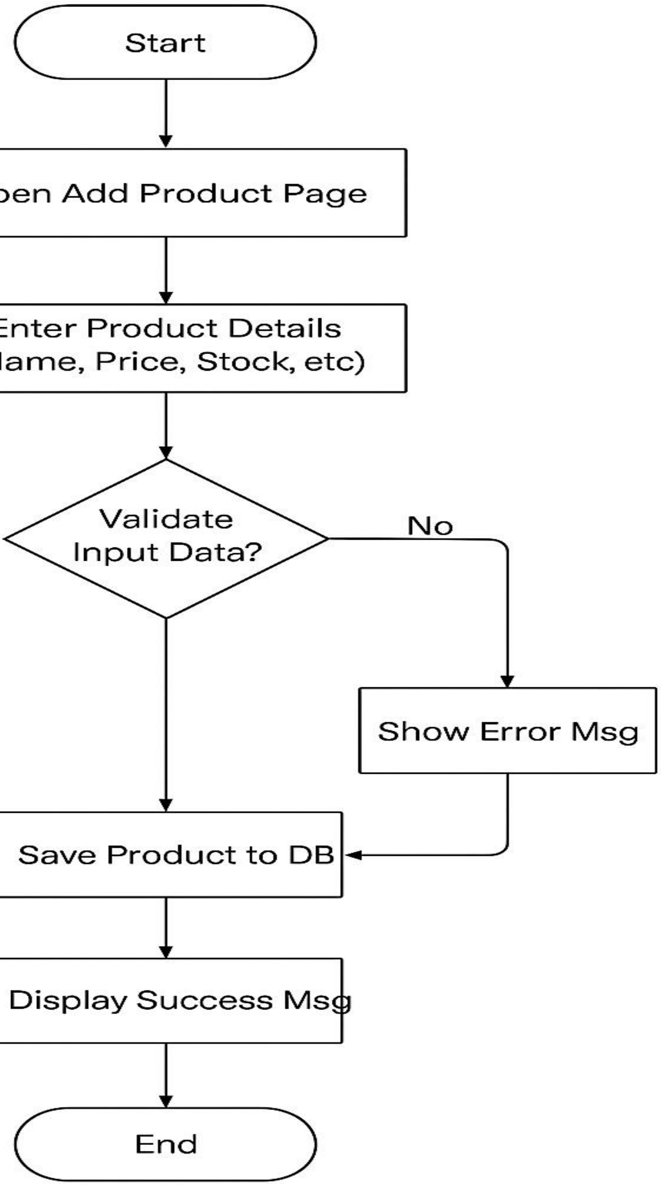
Use Case Diagrame



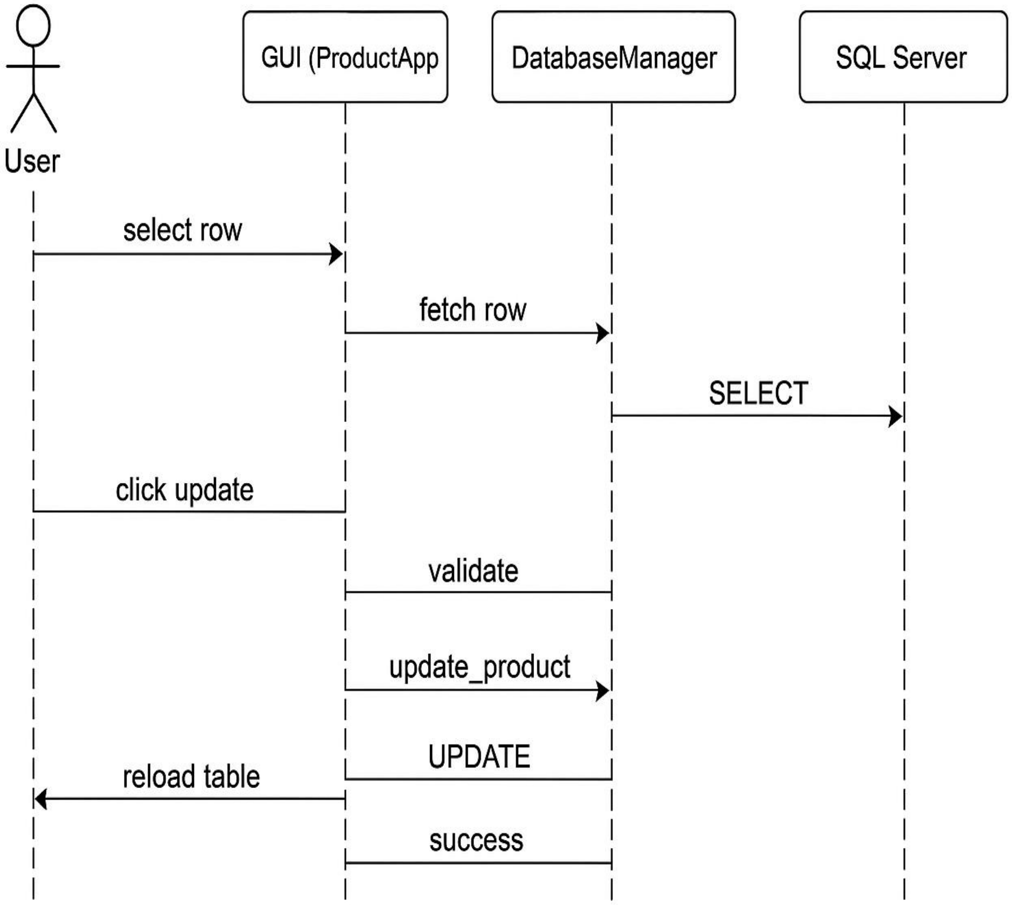
Class Diagram



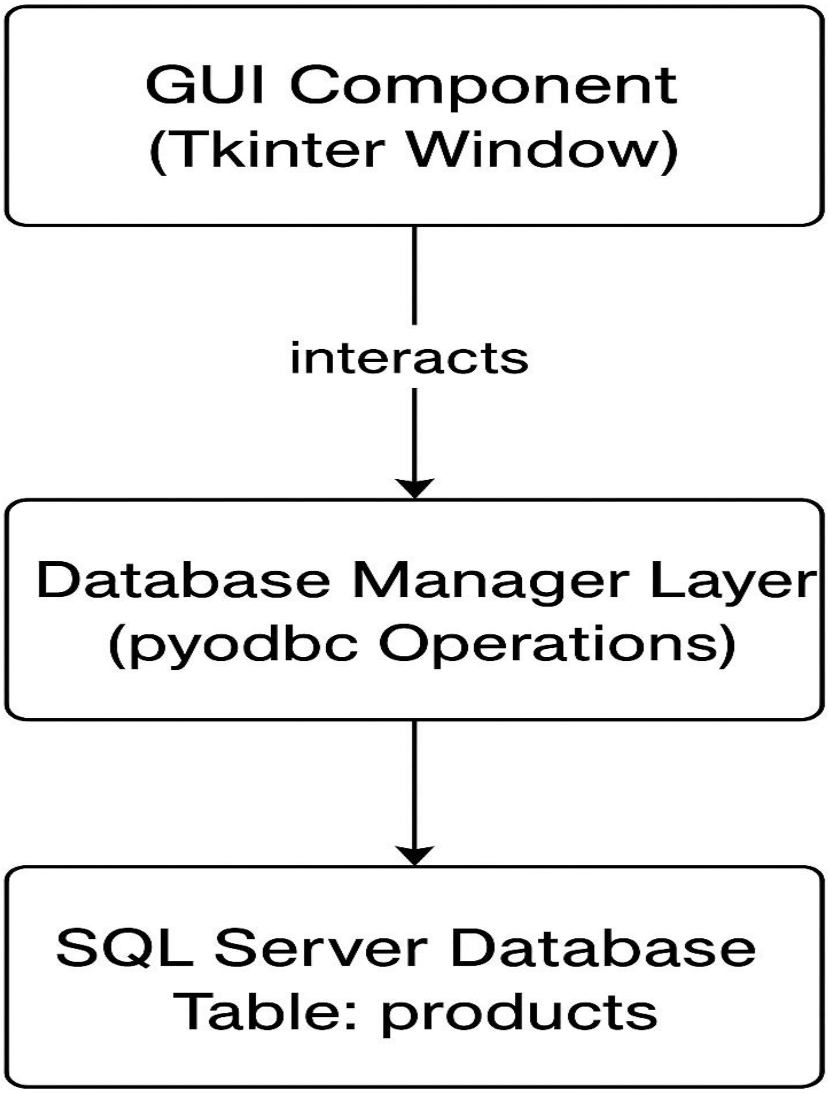
Activity Diagram


Start
Open Add Product Page
Enter Product Details
(Name, Price, Stock, etc)
Validate
Input Data?
No
Show Error Msg
Save Product to DB
Display Success Msg
End

Sequence Diagram



Component Diagram


GUI Component
(Tkinter Window)
interacts
Database Manager Layer
(pyodbc Operations)
SQL Server Database
Table: products