***Contributors: Minh Hieu Bui, Phuc Minh Thao Pham - 989468542***

***Lecturer: Xun Xu Class MSBA 232 – Programming for Data Science***

***12/5/2024***



**Pastry Store Management System Using Python**

Table of Contents

[**Abstract** 2](#_Toc184292261)

[**1. Introduction** 2](#_Toc184292262)

[**2. Project Objective** 4](#_Toc184292263)

[**2.1. Automating Common Tasks** 4](#_Toc184292264)

[**2.2. Reducing Human Error in Inventory Management** 4](#_Toc184292265)

[**2.3. Improving Decision-Making Through Financial Summaries** 5](#_Toc184292266)

[**2.4. Enhancing User Experience with a Robust Interface** 5](#_Toc184292267)

[**2.5. Providing a Scalable and Flexible Framework** 5](#_Toc184292268)

[**3. Data Analytics** 6](#_Toc184292269)

[**3.1 Data** 6](#_Toc184292270)

[**3.2 Methods** 7](#_Toc184292271)

[**3.3 Results of Problem Analytics** 8](#_Toc184292272)

[**4. Implications and Conclusions** 9](#_Toc184292273)

[**4.1. Real-Time Updates for Better Decision-Making** 10](#_Toc184292274)

[**4.2. Financial Insights for Strategic Planning** 10](#_Toc184292275)

[**4.3. Operational Efficiency and Reduced Workload** 11](#_Toc184292276)

[**4.4. Future Scalability and Adaptability** 11](#_Toc184292277)

[**4.5. Conclusion** 11](#_Toc184292278)

[**5. Idea Sharing** 12](#_Toc184292279)

[**5.1. Reinforcing Python’s Value in Business Applications** 12](#_Toc184292280)

[**5.2. Potential Extensions for Future Applications** 13](#_Toc184292281)

[**5.3. Key Takeaways and Reflections** 14](#_Toc184292282)

[**6. References** 15](#_Toc184292283)

[**7. Appendix** 15](#_Toc184292284)

# **Abstract**

This project aims to develop a Python-based program designed to optimize the management of a pastry store's inventory and menu. The system provides a range of functionalities, including the addition, editing, removal, and display of menu items, as well as the calculation of total inventory value and potential revenue per item. The program's interactive design enables efficient data entry, real-time inventory updates, and streamlined access to critical operational insights. By automating essential inventory management tasks, the program reduces manual errors, enhances decision-making, and improves overall business efficiency. This system serves as a practical solution for small business owners, offering a scalable framework that could be adapted to similar retail contexts. Through this project, the integration of programming tools into inventory management processes demonstrates the potential to enhance operational workflows and drive productivity in small-scale enterprises.

**Keywords:**

Pastry Store, Inventory Management, Python Program, Menu System, Business Application

# **1. Introduction**

Efficient inventory management is a cornerstone of success for small businesses, particularly in industries where perishable goods, such as pastries and baked goods, are involved. The ability to maintain optimal stock levels is critical to preventing overstocking, which leads to waste, or understocking, which results in lost sales and dissatisfied customers. However, many small businesses rely on traditional, manual methods of inventory tracking, which are not only time-consuming but also prone to human error. These inefficiencies can significantly impact a business's profitability and operational workflow.

This project addresses the need for an interactive and automated inventory and menu management system, specifically designed for a pastry store. In such a business, where multiple items with varying prices and quantities need to be tracked simultaneously, a well-structured system becomes essential. Traditional approaches, such as handwritten logs or basic spreadsheets, can struggle to handle the complexities of real-time updates, price adjustments, and stock-level calculations. Moreover, these methods often lack the scalability required to adapt to a growing menu or fluctuating inventory demands.

The primary goal of this project is to develop a Python-based program that simplifies and streamlines inventory and menu management. The system integrates essential functionalities, including:

* **Adding new items**: Enabling the smooth introduction of new products into the inventory.
* **Editing existing items**: Allowing price changes, quantity adjustments, or item renaming without complications.
* **Removing discontinued items**: Ensuring the inventory remains relevant and up-to-date.
* **Displaying inventory and menu details**: Providing clear and organized views of the current stock and pricing information.
* **Financial summaries**: Offering insights into potential revenue through calculations of total item values.

The program’s interactive design ensures user-friendly operation, minimizing the learning curve for users with minimal technical expertise. Built-in error handling and validation further reduce the likelihood of mistakes, ensuring data integrity throughout the system.

By automating these core inventory management processes, this project aims to enhance operational efficiency, reduce manual workload, and support better decision-making for business owners. The system not only addresses the immediate needs of a pastry store but also provides a scalable and adaptable framework that can be applied to similar small businesses in the food and retail sectors. Through the integration of Python programming, this project highlights the potential of technology to solve practical challenges and improve business processes.

# **2. Project Objective**

The objective of this Python-based program is to develop an efficient and interactive solution for managing the inventory and menu of a pastry store. The system is designed to address the limitations of manual inventory management methods by automating key processes and providing tools that streamline operations, reduce errors, and support better decision-making. The program focuses on achieving the following specific objectives:

## **2.1. Automating Common Tasks**

The program automates repetitive and time-consuming tasks, such as adding new items, updating quantities, adjusting prices, and removing discontinued products. This automation reduces the manual workload on store managers and staff, allowing them to focus on customer service and other strategic activities. By integrating these features, the system eliminates the inefficiencies associated with traditional manual methods.

## **2.2. Reducing Human Error in Inventory Management**

Manual data entry and record-keeping are prone to errors that can lead to incorrect stock levels, mispricing, and financial discrepancies. The program incorporates error-checking mechanisms, such as input validation and structured workflows, to minimize these risks. For example, it ensures that prices and quantities entered are within acceptable ranges and that duplicate or empty entries are flagged. This reduces inaccuracies and ensures reliable inventory records.

## **2.3. Improving Decision-Making Through Financial Summaries**

The program provides financial insights by calculating the total potential revenue for each item based on its price and quantity. These calculations allow store managers to identify the most profitable items, monitor inventory turnover, and make data-driven decisions, such as adjusting pricing strategies or planning promotions. Additionally, the clear and organized display of menu and inventory data enables quick assessments of stock levels and product availability, supporting proactive decision-making.

## **2.4. Enhancing User Experience with a Robust Interface**

The system is designed to be intuitive and user-friendly, even for users with limited technical expertise. By using interactive menus and prompts, the program ensures smooth navigation and operation. Built-in help options and error messages guide users through the processes, ensuring that the system is accessible to all levels of staff.

## **2.5. Providing a Scalable and Flexible Framework**

While the program is initially designed for a pastry store, its flexible architecture allows it to be easily adapted to other small businesses with similar inventory management needs. The system can accommodate a growing menu, fluctuating stock levels, or additional features, such as sales tracking or customer preferences, making it scalable as the business evolves.

In summary, this project aims to create a Python program that not only addresses the immediate challenges of managing a pastry store's inventory but also delivers a sustainable and scalable solution that supports the long-term growth and efficiency of the business.

# **3. Data Analytics**

## **3.1 Data**

The data for this project is structured as a nested dictionary, providing an efficient and intuitive way to store and access menu and inventory details. Each item in the inventory serves as a key in the main dictionary, with its associated details—price and quantity—stored as a sub-dictionary. This approach enables seamless operations, such as adding, editing, or removing items, as well as performing calculations and generating reports.

The attributes for each item are as follows:

* Price: The cost of a single unit of the item, stored as a float for precision in financial calculations.
* Quantity: The number of units currently in stock, stored as an integer to track inventory levels accurately.

Example Data Structure:

menu\_inventory = {

"Cheesecake": {"price": 6.99, "quantity": 50},

"Croissant": {"price": 1.99, "quantity": 70},

"Strawberry Short Cake": {"price": 6.99, "quantity": 39},

"Cupcake": {"price": 1.99, "quantity": 40},

"Chocolate Muffin": {"price": 1.99, "quantity": 0}

}

This structure ensures that:

* Data is easily accessible and modifiable using dictionary operations.
* New items can be added dynamically by inserting new key-value pairs.
* Existing items can be updated or removed with minimal code.
* The entire dataset can be iterated over for display or analytical purposes.

## **3.2 Methods**

The program utilizes several Python functions to manage and analyze the inventory effectively. Each function is designed to perform specific tasks, ensuring modularity and readability. The key functions include:

* display():
  + Displays the inventory, menu, or both based on user preferences.
  + Provides real-time updates on stock levels and pricing, distinguishing between available items and those that are sold out.
* addItem():
  + Allows users to add new items to the inventory by entering the item name, price, and quantity.
  + Includes input validation to ensure that entries are valid, such as preventing negative prices or empty item names.
* removeItem():
  + Enables the removal of items that are discontinued or no longer relevant.
  + Guides the user through selecting the correct item for removal with clear prompts.
* editItem():
  + Facilitates editing of an item's attributes, including name, price, and quantity.
  + Ensures that updates are valid and do not conflict with existing data (e.g., duplicate item names).
* totalAmountItem():
  + Calculates the total revenue potential for each item by multiplying its price by its quantity.
  + Outputs the calculated amounts in a clear format for financial analysis.
* validateItemSelection():
  + Ensures that user inputs for selecting items (e.g., item numbers) are valid and within range.
  + Prevents errors caused by invalid or out-of-range inputs.

## **3.3 Results of Problem Analytics**

The results of the data analysis are presented in an interactive and user-friendly format. Below are sample outputs demonstrating the program's functionality:

Inventory Display:

1. Cheesecake: 50

2. Croissant: 70

3. Strawberry Short Cake: 39

4. Cupcake: 40

5. Chocolate Muffin: 0

Menu Display:

1. Cheesecake: $6.99

2. Croissant: $1.99

3. Strawberry Short Cake: $6.99

4. Cupcake: $1.99

5. Chocolate Muffin: Sold Out

The system dynamically updates the inventory and menu based on user inputs. For example:

* Adding a new item will automatically display it in the updated menu and inventory.
* Items with zero quantity are marked as "Sold Out" in the menu.
* Financial summaries generated by totalAmountItem() provide insights into the revenue potential of each item.

This analytical framework ensures that the pastry store's inventory is managed efficiently, supporting informed decision-making and minimizing errors.

# **4. Implications and Conclusions**

This program significantly improves the efficiency and accuracy of inventory management for a pastry store by automating key processes and providing real-time updates. The system eliminates reliance on traditional manual methods, which are often time-consuming and prone to errors, such as miscalculations, misrecorded stock levels, or missed updates. By automating these tasks, the program reduces the risk of human error and ensures that inventory data remains consistent and reliable.

## **4.1. Real-Time Updates for Better Decision-Making**

One of the most valuable features of the program is its ability to provide real-time updates on inventory levels and menu availability. This allows store managers to:

* **Identify Sold-Out Items:** Items with zero stock are immediately flagged as "Sold Out" in the menu display. This ensures that managers and staff are aware of stockouts and can restock items promptly to avoid disappointing customers.
* **Monitor Best-Selling Products:** By analyzing inventory turnover, managers can identify which products are in high demand. This insight allows for better allocation of resources, such as focusing on producing more of the best-selling items or promoting complementary products.
* **Plan Inventory Adjustments:** The ability to track stock levels dynamically ensures that overstocking or understocking issues are minimized. Managers can order new supplies or adjust production schedules based on accurate inventory data.

## **4.2. Financial Insights for Strategic Planning**

The program calculates the total revenue potential for each item by multiplying the item’s price by its quantity in stock. These financial insights enable managers to:

* **Evaluate Profitability:** By understanding which items contribute the most to potential revenue, managers can focus on promoting these products or adjusting prices to maximize profit margins.
* **Strategize Promotions:** Items with slower turnover or nearing expiration can be identified and discounted to encourage sales, reducing waste and improving revenue streams.
* **Set Strategic Pricing:** The program provides a clear overview of how pricing affects revenue, allowing managers to experiment with price adjustments or create bundled offers.

## **4.3. Operational Efficiency and Reduced Workload**

The automated nature of this system saves significant time and effort for store managers and staff. Instead of manually updating records, calculating financial summaries, or cross-checking inventory, all these tasks are handled efficiently by the program. This allows managers to focus on other areas of the business, such as customer service, product innovation, or marketing efforts.

## **4.4. Future Scalability and Adaptability**

The program’s modular design ensures that it can grow with the business. For example:

* Additional features, such as sales tracking or integration with point-of-sale systems, can be implemented without major rework.
* The system can easily accommodate an expanding menu or inventory by simply adding new items to the data structure.
* Data collected through the program can be analyzed further to identify long-term trends, helping managers plan for future growth.

## **4.5. Conclusion**

Overall, this program addresses the core challenges of inventory management in a pastry store by automating processes, providing actionable insights, and enabling better decision-making. The financial summaries and real-time updates empower store managers to maintain optimal stock levels, promote high-demand items, and improve profitability. By combining automation and data-driven insights, this program enhances operational workflows, reduces errors, and supports the long-term success of the business. It serves as a practical example of how technology can transform small-scale retail operations, making them more efficient and competitive.

# **5. Idea Sharing**

This project highlighted the critical role of Python in addressing real-world business challenges, particularly in the context of small-scale retail operations. The process of designing and implementing a Python-based inventory management system demonstrated the language’s versatility, accessibility, and power in solving practical problems.

## **5.1. Reinforcing Python’s Value in Business Applications**

Python proved to be an ideal choice for this project due to its:

* **Simplicity:** Python’s straightforward syntax makes it accessible even to individuals with limited programming experience, enabling them to understand and modify the program as needed.
* **Rich Libraries:** The language’s extensive libraries and frameworks—such as pandas for data analysis, matplotlib for visualization, and tkinter for graphical user interfaces—provide ready-to-use solutions for common programming tasks. These libraries significantly accelerate development while maintaining code quality.
* **Community Support:** Python’s active and diverse user community ensures that developers have access to a wealth of tutorials, documentation, and forums, reducing the learning curve and providing guidance for troubleshooting.

## **5.2. Potential Extensions for Future Applications**

This project also opened up opportunities for extending the program with advanced features, transforming it from a basic inventory management tool into a comprehensive business management system. Some possible extensions include:

1. **Sales Forecasting:**
   * By incorporating historical sales data, the program could use machine learning models or statistical methods to predict future demand for menu items.
   * This feature would help managers plan production schedules, optimize stock levels, and prepare for seasonal fluctuations or special events.
2. **Integration with Point-of-Sale (POS) Systems:**
   * Connecting the inventory management system with POS systems would enable real-time updates as items are sold, providing an accurate and dynamic view of inventory levels.
   * Integration could also automate sales tracking, allowing managers to generate detailed reports on revenue, profit margins, and customer purchasing patterns.
3. **Data Visualization Tools:**
   * Adding data visualization capabilities would enhance the program’s usability by presenting inventory trends, sales performance, and financial summaries through graphs and charts.
   * Libraries like matplotlib or seaborn could be used to create dashboards that provide managers with at-a-glance insights into key metrics.
4. **Mobile or Web-Based Access:**
   * Extending the program to a web or mobile application would improve accessibility, enabling managers to monitor inventory and sales remotely.
   * Cloud-based solutions could facilitate real-time collaboration among staff and managers, streamlining workflows across multiple store locations.
5. **Customer Insights and Loyalty Features:**
   * Adding customer feedback or loyalty tracking would provide valuable insights into customer preferences, helping managers tailor their offerings to meet demand.
   * Features such as tracking the most frequently purchased items or creating personalized discounts could enhance customer satisfaction and retention.

## **5.3. Key Takeaways and Reflections**

Through the development of this project, several valuable lessons were learned:

* **Importance of Automation:** Automating routine tasks, such as inventory updates or financial calculations, not only saves time but also minimizes the risk of errors, ensuring more reliable and actionable data.
* **Flexibility of Python:** The flexibility of Python’s data structures and its ability to scale make it an excellent tool for developing systems that can grow alongside a business.
* **Real-World Impact:** By solving practical challenges, programming projects like this can have a direct and meaningful impact on small businesses, improving their efficiency, profitability, and competitiveness

# **6. References**

* Python Official Documentation: <https://docs.python.org>
* Online resources on Canvas – MSBA 232

# **7. Appendix**

| **Time** | **Contents** |
| --- | --- |
| November 25, 2024 | Group Created |
| December 3, 2024 | Coding Framework Developed |
| December 5, 2024 | Final Project Paper Submission |