Project Report: Emma Bakery Inventory Management System

Executive Summary

The Emma Bakery Inventory Management System is a Python-based application designed to simplify and automate the process of managing inventory in a bakery setting. This project encapsulates essential functions such as adding new items, updating existing items, and displaying inventory in a user-friendly, text-based interface. While it currently operates as a standalone application without persistent storage, it lays a strong foundation for a more comprehensive inventory management solution.

Project Overview

Objective

The primary goal of this project is to create a simple and effective inventory management tool tailored for Emma Bakery. The key objectives include: - Managing item details including item code, name, quantity, weight, unit, and price. - Providing an interactive interface for users to add, update, and view inventory items. - Enhancing operational efficiency and accuracy in inventory management.

Scope

The system focuses on basic inventory management without delving into advanced features like order processing, sales tracking, or integration with other systems.

Technical Details

Technology Stack

- **Programming Language**: Python
- **Development Environment**: Text-based console application

System Architecture

- **Class `Items`**: Serves as the blueprint for inventory items.
- **Utility Functions**: Facilitate various operations like displaying inventory and updating item details.
- **Main Function**: Acts as the control center for user interactions and workflow management.

Key Features

- **Adding New Items**: Users can add items to the inventory with unique item codes.
- **Updating Items**: Provides options to modify item details.
- **Displaying Inventory**: Lists all items in a tabular format.

Implementation

Development Process

The development followed a structured approach: 1. **Requirement Analysis**: Identified key features needed for a basic inventory management system. 2. **Design**: Outlined the class structure and defined necessary functions. 3. **Coding**: Implemented the system in Python, focusing on readability and simplicity. 4. **Testing**: Conducted basic tests to ensure functionality.

Challenges and Solutions

- **User Input Validation**: Implemented basic validation and error messages to guide user input.
- **Data Persistence**: As a future improvement, integrating a database or file-based storage is recommended.

Usage

The system is designed for bakery staff with basic computer skills. It allows for easy management of inventory through simple commands and inputs.

Project Documentation

Class: `Items`

- **Purpose**: Represents individual items in the inventory.
- **Attributes**:
  - `item_code_index`: Class variable acting as an index to track item codes.
  - Private instance variables: `__item_code`, `__name`, `__quantity`, `__weight`, `__unit`, `__price`.
- **Methods**:
  - `__init__`: Constructor to initialize an item with code, name, quantity, weight, unit, and price.
  - `create_object_by_user_input`: Class method to create an item object based on user input.
  - `update_name`, `update_quantity`, `update_weight`, `update_weight_unit`, `update_price`: Methods to update respective attributes.
  - `get_item_code`, `get_name`, `get_quantity`, `get_weight`, `get_unit`, `get_price`: Getter methods for respective attributes.

    - `print_item_info`: Prints detailed information about the item.
- **Usage**: Used to manage individual items in the bakery's inventory.

Function: `print_items_data`

- **Purpose**: Prints a formatted table of all items in the inventory.
- **Parameters**:
    - `items_list`: A list of `Items` objects.
- **Behavior**: Calculates column widths and prints each item's details in a tabular format.

Function: `main_menu`

- **Purpose**: Displays the main menu and gets user choice.
- **Returns**: User's choice as a string.

Function: `find_item_by_code`

- **Purpose**: Finds an item in the database by its code.
- **Parameters**:
    - `database`: The list of all `Items` objects.
    - `item_code`: The code of the item to find.
- **Returns**: The `Items` object if found, otherwise `None`.

Function: `update_item`

- **Purpose**: Provides a submenu for updating various attributes of an item.
- **Parameters**:
    - `item`: The `Items` object to be updated.

Function: `main`

- **Purpose**: Entry point for the program. Manages the overall workflow and user interactions.
- **Behavior**: Implements a loop to handle user choices for adding, updating, and printing items, and exiting the program.

Execution

- **Entry Point**: The `if __name__ == "__main__":` block calls the `main` function to start the program.

Notes

- The program heavily relies on user input for data entry and updates.
- It manages inventory items by their unique item code.
- The program does not have persistent data storage; data is lost when the program exits.

- Error handling is primarily done through print statements.

Potential Improvements

- Implement data validation for user inputs.
- Add functionality to delete items from the inventory.
- Integrate persistent data storage (e.g., database or file system).
- Enhance the user interface, potentially by using a GUI or a web interface.

Conclusion and Future Work

The Emma Bakery Inventory Management System successfully provides a foundational platform for inventory management. Future enhancements could include: - Integration with a database for data persistence. - Implementation of a graphical user interface (GUI). - Expansion of features to include sales tracking and reporting.