**CASE STUDY**

**IN**

**ADVANCE**

**DATABASE**

Submitted by:

**ALIMURUNG, Marco M.**

**SANTOS, Stephen Kyle**

**VILLANUEVA, Joanna**

*Bachelor of Science in Computer Science Students*

Section C

Submitted to:  
  
**Mr. Arjay D. Balberan***Instructor*

**Table of Contents**

**1. Introduction ----------------------------------------------------------------------------------------** Page 4

* 1. Background of the store **--------------------------------------------------------------------** Page 4
  2. Importance of an Inventory Management System **--------------------------------------** Page 5
  3. Objectives of the System **-------------------------------------------------------------------** Page 6
     1. General objective **--------------------------------------------------------------------** Page 6
     2. Specific objectives **------------------------------------------------------------------** Page 6

1. **Literature Review --------------------------------------------------------------------------------** Page #
   1. Existing Inventory Management Systems in Restaurants **------------------------------** Page #
   2. Best Practices in Inventory Management **-------------------------------------------------** Page #
   3. Relevant Technologies **----------------------------------------------------------------------** Page #

**3. System Requirements and Design -------------------------------------------------------------** Page 9

3.1 System Requirements **------------------------------------------------------------------------** Page 9

3.1.1 Hardware Requirements **--------------------------------------------------------------** Page 9

3.1.2 Software Requirements **---------------------------------------------------------------** Page 9

3.1.3 Network Requirements **---------------------------------------------------------------** Page 9

3.2 Functional Requirements **--------------------------------------------------------------------** Page 9

3.3 Database Design **----------------------------------------------------------------------------** Page 10

3.3.1 Entity-Relationship Diagram (ERD) **----------------------------------------------** Page 10

3.3.2 Data Flow Diagram (DFD) **---------------------------------------------------------** Page 10

3.4 Database Tables **-----------------------------------------------------------------------------** Page 10

3.5 System Architecture **------------------------------------------------------------------------** Page 10

**4. Development Process -----------------------------------------------------------------------------** Page

4.1 Planning **----------------------------------------------------------------------------------------** Page #

4.2 Analysis **----------------------------------------------------------------------------------------** Page #

4.3 Design **------------------------------------------------------------------------------------------** Page #

4.4 Implementation **--------------------------------------------------------------------------------** Page #

4.5 Testing **------------------------------------------------------------------------------------------** Page #

4.6 Deployment **------------------------------------------------------------------------------------** Page #

**5. User Testing and Feedback ----------------------------------------------------------------------** Page #

5.1 Testing Procedures **----------------------------------------------------------------------------** Page #

5.2 Issues Identified **-------------------------------------------------------------------------------** Page #

5.3 Solutions Implemented **-----------------------------------------------------------------------** Page #

5.4 Feedback from Users **-------------------------------------------------------------------------** Page #

5.5 Final Adjustments **-----------------------------------------------------------------------------** Page #

**6. Grades** **-----------------------------------------------------------------------------------------------** Page #

**7. References** ------------------------------------------------------------------------------------------- Page #

**Introduction**

* 1. **Background of Andoks Concepcion**

Chowking Mabalacat, a branch of the popular Chowking chain, operates as a fast-food restaurant known for its Chinese-Filipino fusion cuisine. Like many restaurants, it relies on managing a steady flow of perishable ingredients and supplies, requiring a reliable system for inventory tracking.

Currently, the branch employs a manual inventory system, where records of stock levels and product expiration dates are tracked using paper-based methods. This system involves the staff manually writing down stock levels, dates of replenishment, and expiration on physical logs. While simple, this approach has several limitations, including frequent errors in data entry, lack of real-time updates, and the absence of automated stock monitoring and alerts.\

**Issues faced with the current system:**

**Human Error:** The manual process is prone to mistakes, such as incorrect stock counts or missed expiration dates.

**Time-Consuming:** Manually updating stock levels and reviewing inventory reports takes considerable time.

**Lack of Real-Time Data:** Managers cannot access real-time updates on stock levels, which leads to delays in ordering new supplies.

**Limited Reporting:** The system does not generate reports automatically, which hinders decision-making for stock replenishment and waste reduction.

#### Importance of an Inventory Management System

General Overview of Inventory Management: Inventory management is a key process in any business, especially in the restaurant industry, where it involves tracking stock levels, managing orders, and ensuring products are available when needed. A robust inventory management system helps businesses maintain an optimal balance between having enough products to meet demand while minimizing excess stock that may spoil or go unused.

Why Accurate Inventory Tracking is Vital for Restaurant Operations: For restaurants like Chowking Mabalacat, accurate inventory tracking is crucial for several reasons:

**Prevents Wastage:** By keeping track of expiration dates and stock levels, the restaurant can reduce waste from spoiled products.

**Ensures Stock Availability:** It helps ensure that essential ingredients are always in stock, preventing menu items from being unavailable.

**Improves Financial Efficiency:** Proper inventory management helps avoid overstocking or understocking, leading to better cost control and minimizing unnecessary expenses.

Streamlines Operations: Automated inventory systems reduce the time spent on manual stock counts, allowing staff to focus on other essential tasks.

* 1. **Objectives of the System**
     1. **General Objective:**

The general objective of this project is to develop an inventory management system for Chowking Mabalacat to enable efficient tracking of stock levels and expiration dates while reducing reliance on manual methods.

* + 1. **Specific Objectives:**

**Reduce Human Errors:** Implement a system that minimizes the chances of human error in stock counts and expiration tracking.

**Streamline Ordering and Stock Replenishment:** Simplify the process of ordering new stock by providing accurate, real-time data on current inventory levels and predicting when replenishment is needed.

**Provide Real-Time Inventory Updates and Reports:** Offer managers access to real-time data on inventory status and generate reports automatically, making it easier to monitor stock levels, expiration dates, and trends in product usage.

**Literature Review**

In their 2019 study titled "IoT Based Food Inventory Tracking System," S. P. Lakshmi Narayan, E. Kavinkartik, and E. Prabhu highlight the importance of inventory control in effective kitchen management. Monitoring kitchen inventory facilitates better planning and decision-making. As technology rapidly evolves, individuals increasingly prefer using smart devices for everyday tasks over manual recording. Managing common food inventory poses significant challenges for households, restaurants, and food chains, particularly in timely replenishment and tracking food expiry. Busy professionals and restaurants often struggle with this due to the need for timely human intervention. The proposed IoT-based food inventory tracking system allows for real-time monitoring of kitchen stock, enabling users to analyze consumption data and predict usage patterns. Users can also check real-time inventory status and consumption history via an Android app. The system comprises a microcontroller, load cell, wireless module, MQTT broker, and both desktop and android applications, providing a completely wireless and reliable solution for domestic and commercial use.

In their 2020 research titled “The Effect of Inventory Turnover on Financial Performance in the US Restaurant Industry: The Moderating Role of Exposure to Commodity Price Risk,” Eunhye (Olivia) Park and Woo-Hyuk Kim address the critical yet often overlooked area of inventory management in the restaurant sector. Despite its significance, satisfactory inventory turnover in the industry has not been well documented. This study explores the connection between inventory turnover and firm performance, while also examining how exposure to commodity price risk moderates this relationship. Data was gathered from publicly traded US restaurant companies using the Mergent Online database, covering the period from 1999 to 2015. The findings reveal a positive correlation between inventory turnover and corporate financial performance. Additionally, the study highlights the interaction between commodity price risk and the relationship between inventory turnover and profitability. This research is notable for being the first empirical investigation into inventory management practices within the restaurant industry, underscoring the importance of effective inventory management tailored to specific restaurant types.

In the 2021 study titled “Operational Effects of Using Restaurant Management System: An Assessment According to Business Features,” Ernel Memis Kocaman investigates the advantages and disadvantages of Restaurant Management Systems (RMS) from the perspectives of restaurant staff. The research focuses on how RMS impacts operations based on various business characteristics, analyzing differences among restaurant types. A total of 385 employees from active restaurants participated in the study, with data collected through a structured questionnaire. Participants from RMS-utilizing restaurants reported that the system improved operations management, boosted sales, and enhanced service standards. Conversely, staff in restaurants without RMS indicated greater difficulty in system usage (p < 0.05). Additionally, employees in chain restaurants perceived RMS to have a more significant positive impact on sales and service standards (p < 0.05). Notably, employees in smaller restaurants (10 or fewer staff) reported the lowest positive effects regarding RMS on operations management and sales increase. This suggests a correlation between restaurant size and the necessity for RMS implementation.

**System Requirements and Design**

**3.1 System Requirements**

**3.1.1 Hardware Requirements:**

Essential devices include barcode scanners for tracking, desktop computers for management tasks, and mobile devices for on-the-go access.

**3.1.2 Software Requirements:**

**Operating Systems:** Must support environments like Windows or Linux.

**Web Servers:** Apache to host the application for web access.

**Database Management Systems:** MySQL for structured data storage, with PHP for backend scripting.

**3.1.3 Network Requirements:**

A reliable internet connection for online functionalities.

A local network setup to connect devices within the store.

Cloud storage may be utilized for data backup and remote access.

**3.2 Functional Requirements**

**User Authentication and Role Management:** Secure login for users with role-specific permissions.

**Real-Time Inventory Updates and Reports:** Immediate updates on inventory status and generation of analytical reports.

**Alerts for Low Stock Levels:** Notifications triggered when stock falls below predefined thresholds.

**Supplier and Order Management:** Tools for managing supplier information and processing orders efficiently.

**3.3 Database Design**

**3.3.1 Entity-Relationship Diagram (ERD):** This diagram illustrates the relationships among entities like products, suppliers, orders, and transactions to clarify data interconnections.

**3.3.2 Data Flow Diagram (DFD):** Shows how data flows within the system, highlighting inputs, processes, and outputs.

**3.4 Database Tables:**

Food\_items, inventory, users

Food\_items table Includes fields such as id, name, image\_url, price. Inventory table includes id, item\_name, amount, expiration\_date, category, image\_url. User table includes id, email, password, role, firstname, lastname.

**3.5 System Architecture**

**Frontend:** The user interface is designed for ease of use, enabling staff to manage inventory intuitively through web pages. Features include a dashboard for real-time updates and alerts.

**Backend:** Comprises a MySQL database for data storage and an Apache server that processes requests via PHP scripts, ensuring smooth data transactions and business logic execution.

**Development Process**

**4.1 Planning**

**Define Goals, Timeline, and Scope:** The goal is to create an inventory management system that allows efficient tracking of products in the chiller, freezer, and stock room. The system must handle two user roles: the manager (who adds and manages products) and the customer (who views inventory). The timeline for development is set, with specific milestones for completion.

**Initial Research on Chowking Mabalacat's Requirements:** Research is conducted to identify the specific needs of the restaurant, such as the current manual system’s shortcomings, the type of products managed, and the necessary reports and alerts. This stage ensures the system addresses Chowking’s unique operational needs.

**4.2 Analysis**

**Study the Existing System:** The manual inventory process is reviewed to identify inefficiencies such as data entry errors, lack of real-time updates, and difficulties in tracking stock levels and expiration dates.

**Interviews with Stakeholders:** Store managers, staff, and other key personnel are interviewed to gather specific requirements. These interviews highlight essential features such as real-time updates, low-stock alerts, and user-friendly reporting tools, which are necessary for the system’s effectiveness.

**4.3 Design**

**System Diagrams and UI Mockups:** Detailed system diagrams such as Entity-Relationship Diagrams (ERD) and Data Flow Diagrams (DFD) are created to visualize how data will flow through the system. Mockups of the user interface are also developed to outline how managers and customers will interact with the system.

**Database Structure and Feature Planning:** The database structure is designed to ensure all essential data is captured. Tables for products, suppliers, transactions, and users are planned, ensuring the system can handle large datasets efficiently. Features such as real-time updates and stock alerts are mapped out in detail.

**4.4 Implementation**

**System Development:** Using the selected technologies (PHP, MySQL, HTML, CSS, and JavaScript), the system is developed. The backend is set up to handle data processing, while the frontend ensures that the user interface is intuitive and accessible.

**Coding Key Features:** Specific functionalities like real-time inventory updates, low-stock alerts, product expiration tracking, and automated report generation are coded. Each feature is tested incrementally to ensure it works as intended.

**4.5 Testing**

**Unit Testing:** Individual components, such as the user login, product management, and reporting tools, are tested to ensure they function correctly. This testing checks each part of the system in isolation to identify and fix any issues.

**System Testing:** The entire system is tested to ensure all modules work together seamlessly. During this phase, testers simulate real-world scenarios, such as adding new products, generating reports, and triggering stock alerts.

**User Feedback:** Classmates acting as users (staff, managers) provide feedback on the system’s usability and functionality during testing. This feedback is used to refine features, fix bugs, and improve the overall user experience.

**4.6 Deployment**

**Full-Scale Launch:** The system is officially launched. All features are made available, ensuring that inventory can be managed efficiently.

**Classmate Testing:** Feedback and testing were performed by classmates, simulating the roles of staff and managers. This allowed for assessing the system's functionality in a controlled environment, focusing on its effectiveness and usability.

**User Testing and Feedback**

**5.1 Description of User Testing Stages:**

**Alpha Testing:** Conducted by the development team to identify initial functionality issues, focusing on the manager's ability to log in and add products, as well as customer view accuracy.

**Beta Testing:** Involved classmates as testers who used the system to provide real-world feedback on usability and functionality.

**Testing Criteria:**

**Usability:** Evaluated how intuitive the interface was for both the manager and customer.

**Accuracy of Inventory Updates:** Ensured that updates reflected the correct quantities and expiration dates after new product entries.

**Report Generation:** Assessed the effectiveness of the reports generated for inventory status.

**5.2 Issues Identified**

**Stock Alerts Not Triggered Correctly:** Some classmates noted that alerts for low stock levels were inconsistent.

**Slow Loading Time:** The system lagged when displaying a large number of items.

**Report Misalignment:** Reports did not always accurately reflect the current inventory status.

**5.3 Solutions Implemented**

**Recalibration of Stock Alerts:** Adjusted the alert thresholds to ensure timely notifications for low stock and nearing expiration products.

**Performance Optimization:** Enhanced the database queries to reduce loading times when displaying inventory.

**Report Fixes:** Updated the report generation process to improve accuracy in summarizing stock levels and expiration dates.

5.4 Feedback from Users

**Classmate Feedback:** "The interface is user-friendly, and adding products is straightforward. However, the alerts could be more reliable."

**Usability Insights:** Many testers appreciated the clear layout of available products but suggested improvements for quicker access to information.

**5.5 Final Adjustments**

**UI Enhancements:** Made minor adjustments to the user interface for improved navigation.

**Finalizing Stock Alerts:** Conducted a last round of adjustments to stock alert settings based on feedback, ensuring reliability.

**Grades**

**Day 1**

**Day 2**

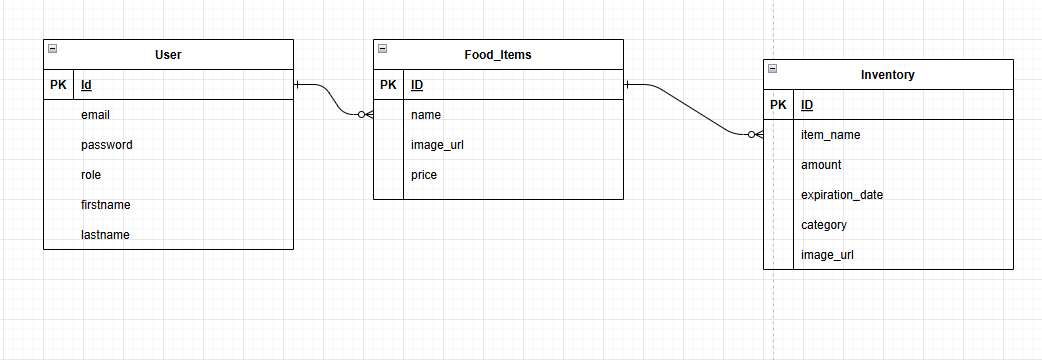
**Day 3**

**References**

<https://link.springer.com/chapter/10.1007/978-981-13-5758-9_4>

<https://journals.sagepub.com/doi/abs/10.1177/1354816620923860>

https://www.sciencedirect.com/science/article/abs/pii/S1878450X21001074

****