

**University of Central Punjab**

**BS Computer Sciences**

**Final Report**

**Programming Fundamentals**

**Submitted by:**

**Group Members:**

***Afifa Muhammad* L1F22BSCS0478 .**

***Muhammad Abdullah***  **L1F22BSCS0032**

***Saad ur rehman (head)* L1F22BSCS1074**

**Submitted to:**

***Sir Hafiz Bilal***

**UNIVERSITY OF CENTRAL PUNJAB**

**This Report is submitted to the Faculty of Information Technology University of the Central Punjab, to fulfill the Requirement of BS**

**(morning) degree in Computer Sciences.**

**Batch 2022- 2026**

**ACKNOWLEDGEMENT:**

All praise to Allah Almighty, the most beneficent and the most merciful. One of the greatest pleasures of writing this is to display our gratitude to those who played a significant role in the completion of our final report. Most of them may never appear on the scene but this acknowledgement is to thank them. Firstly, our parents who were always there for us from the very beginning then comes our respected and learned Mentor Sir Hafiz Bilal whose assistance, valuable comments and guidance helped us in the completion of our task.

**Table of Contents:**

|  |  |  |
| --- | --- | --- |
| **Contents** | **Page #** |  |
| Problem statement………………………………………………………………... | 05 |
| Abstract….………………………………………………………………………... | 05 |
| Introduction………………………………………………………………………. | 06 |
| Methodology……………………………………………………… | 07 |
| Methods………………………………………………… | 09 |
| Errors………………………………………………………………... | 09 |
| Future plains……………………………………………………………………. | 11 |
| Code explanation………………………………………………………………. | 13 |
| Flow chart………………………………………………………………………... | 14 |
| Conclusion…………………………………………………………………... | 15 |

**Problem Statement:**

The objective of this project is to create a C++ department store management system. By offering an effective and automated solution, the initiative seeks to streamline department store operations. The store currently confronts a number of difficulties, including manual inventory management, trouble tracking sales and profitability, and the lack of a centralized system for managing customers and employees. These problems lead to inefficiencies, mistakes, and laborious procedures. The goal of this program is to develop a comprehensive system that will allow the store to efficiently manage its inventory, track sales, generate reports, handle customers. We seek to decrease manual effort and errors while increasing productivity, accuracy, and client satisfaction through the development of this departmental store system of management.

**Abstract:**

A C++ program called the Departmental Store Management System (DSMS) evolved to help department stores manage their operations with greater effectiveness. The project efforts to provide a computerized approach for problems like manually handling inventory, tracking revenue, and managing consumers and also admin. The store may simplify its activities, reduce errors, and enhance efficiency by putting the DSMS into place. Sophisticated inventory management, precise sales tracking, thorough reporting, centralized customer information management, and effective employee administration are just a few of the features that the system offers. Overall, the DSMS offers a strong and simple solution to improve departmental store operations, make the most use of resources, and raise customer satisfaction.

**Introduction:**

The Departmental Store Management System (DSMS) is a cutting-edge C++ software system that intends to revolutionize department shops' management and operations. The issues that traditional stores encounter, where manual operations result in inefficiencies, mistakes, and fragmented data, are addressed by this project. Department shops may improve customer satisfaction, increase efficiency, and optimize operations by implementing the DSMS.

In order to get around common store management obstacles, the DSMS provides a user-friendly interface and a number of capabilities. The inventory management module automates stock monitoring, classification, and low-stock item warnings. The sales monitoring module equips store administrators with the information they need to make wise decisions by providing real-time sales statistics, automated invoicing, and thorough reporting. Additionally, the system has modules for managing customers that centralize data and enable customized services.

The DSMS accelerates processes while simultaneously guaranteeing data security through strong control of access and authentication measures. The system maintains the integrity and confidentiality of consumer, sales, and employee data by preserving sensitive information, fostering trust and confidence in the store's operations.

As a whole, the DSMS transforms department store management by automating processes, cutting down on errors, and offering insightful data. Stores may improve sales monitoring, manage inventory more effectively, and provide great customer service by putting this system in place. The DSMS will revolutionize how department shops run in the changing retail environment.

**Methodology:**

The methodologies are:

**Requirements Gathering:**

The first step in the methodology is to gather the requirements for the Departmental Store Management System (DSMS). This involves conducting interviews and discussions with key stakeholders, such as store managers and employees, to understand their needs and expectations from the system. The requirements will include functionalities such as inventory management, sales tracking, reporting, customer management, and administration.

**System Design:**

Once the requirements are gathered, the next step is to design the system architecture and user interface. This involves creating a high-level design that outlines the modules, their interactions, and the flow of data within the system. The design should take into consideration factors such as scalability, usability, and security.

**Development:**

With the system design in place, the development phase begins. This involves writing code in C++ to implement the various modules and functionalities of the DSMS. The development process should follow best coding practices, including modular and well-documented code, error handling, and adherence to coding standards.

**Testing:**

After the development phase, thorough testing of the DSMS is conducted to ensure that it meets the specified requirements and functions correctly. This includes unit testing of individual modules as well as integration testing to verify the interaction between different components of the system. Test cases are designed to cover various scenarios and edge cases to ensure the reliability and accuracy of the system.

**Deployment and Training:**

Once the DSMS passes the testing phase, it is ready for deployment in the departmental store environment. The system is installed on the designated hardware, and necessary configurations are made. Additionally, training sessions are conducted to familiarize store administrators and employees with the functionalities and usage of the DSMS.

**Maintenance and Support:**

After deployment, ongoing maintenance and support are provided to ensure the smooth operation of the DSMS. This includes bug fixes, software updates, and addressing any issues that may arise during usage. Regular backups and data security measures are implemented to safeguard the system and its data.

The methodology outlined above provides a systematic approach to developing the Departmental Store Management System (DSMS) in C++. It ensures that the system is developed according to the specified requirements, thoroughly tested, deployed successfully, and maintained for optimal performance.

**Methods:**

The methods are:

* 1 D Array
* Pointers
* User define functions
* If else condition
* Loops
* Switch statement Cases
* Char array
* File handling
* Dynamic array
* Data types like (int char etc.)
* Variables

**Errors:**

When developing the Departmental Store Management System (DSMS), it is important to anticipate and address potential errors or bugs that may arise during the implementation and usage of the system. Here are three types of errors or bugs that can be expected:

**Input Validation Errors:**

One common type of error stems from inadequate input validation. If the DSMS does not properly validate user inputs, it may result in unexpected behavior or system crashes. For example, if the system allows for invalid or incorrect data to be entered, it could lead to inaccurate inventory records, incorrect calculations, or corrupted data. To mitigate these errors, robust input validation mechanisms should be implemented, including checks for data type, range, and format to ensure that only valid inputs are accepted.

**Logic Errors:**

Logic errors can occur when there are mistakes or flaws in the program's algorithm or decision-making logic. These errors may not cause the system to crash but can result in incorrect outputs or undesired behavior. For instance, if the sales tracking module has a logic error, it may miscalculate the total sales or generate inaccurate reports. To identify and resolve logic errors, thorough testing, including unit testing and integration testing, should be conducted to verify the correctness of the system's algorithms and logic.

**Data Integrity Issues:**

Maintaining the integrity of the data within the DSMS is crucial. Data integrity issues can occur when data is improperly stored, updated, or accessed. This can lead to inconsistencies, missing data, or incorrect calculations. For instance, if the inventory management module fails to update stock levels accurately, it can result in discrepancies between the recorded inventory and the actual stock available. Implementing proper data validation, error handling, and transactional mechanisms can help mitigate data integrity issues.

**System Compatibility and Integration Challenges:**

The Departmental Store Management System (DSMS) may encounter issues related to system compatibility and integration. This arises from the need to integrate the DSMS with existing hardware and software systems used in departmental stores, such as barcode scanners, cash registers, and accounting software. Incompatibilities can lead to data synchronization problems, communication failures, and disruptions in the workflow. Thorough testing, standardized data formats, and established communication protocols can help mitigate these challenges and ensure seamless integration between the DSMS and external systems.

**Mitigate:**

To mitigate and address these errors and bugs, thorough testing and quality assurance practices should be followed throughout the development process. This includes conducting functional testing, stress testing, and edge case testing to ensure that the system behaves as expected under different scenarios. Additionally, regular software updates, bug fixes, and user feedback mechanisms should be implemented to address any issues that may arise during the usage of the DSMS.

**Future Plains:**

* Scalability challenges as the departmental store expands and handles larger volumes of data and transactions.
* Security vulnerabilities that may arise due to potential cyber threats and unauthorized access to sensitive data.
* Technology obsolescence as newer hardware and software systems emerge, requiring updates and adaptations to maintain compatibility.
* Changing regulatory requirements and compliance standards that may necessitate modifications to the DSMS.
* User resistance and training issues during the adoption and implementation of the new system.
* Data backup and disaster recovery strategies to safeguard against potential data loss or system failures.
* Ongoing maintenance and support requirements to address bugs, software updates, and evolving user needs.
* Integration challenges with third-party systems or services, such as payment gateways or e-commerce platforms.
* Evolution of customer preferences and market dynamics, requiring continuous enhancements to meet changing demands.
* Adapting to technological advancements, such as the Internet of Things (IoT) or artificial intelligence, to leverage new opportunities and improve efficiency.
* Will move to the GUI.

**Code explanation:**

The DSMS would typically involve multiple modules such as inventory management, sales tracking, reporting, customer management, and employee administration.

**Inventory Management:**

The inventory management module handles tasks such as adding products to the inventory, updating stock levels, and generating alerts for low stock items. It may include functions for categorizing products, tracking product attributes (e.g., price, quantity etc.).

**Sales Tracking:**

The sales tracking module records sales transactions, including details like the products sold, quantities, prices, and payment methods. It may also calculate totals, generate invoices, and update inventory levels accordingly. Additionally, it can provide reporting features to analyze sales data and identify sales trends.

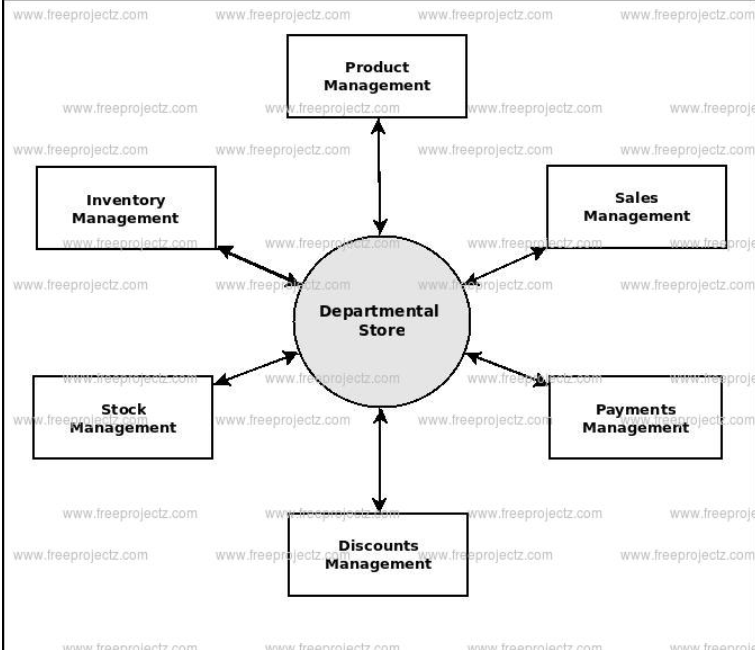
**Reporting:**

The reporting module generates various reports, including sales reports, inventory reports, profit and loss statements, and customer analytics. These reports provide insights into the store's performance, helping store administrators make informed decisions and formulate strategies to optimize operations.

**Customer Management**:

The customer management module maintains a centralized database of customer information, including profiles, purchase history, loyalty program details, and contact information. It enables personalized services, targeted marketing campaigns, and loyalty reward programs, enhancing customer satisfaction and retention.

**Flowchart:**



**Conclusion:**

In conclusion, departmental stores serve as comprehensive retail destinations, providing a wide range of products, efficient services, and convenient shopping experiences to customers. By effectively managing their operations, inventory, and customer relationships, departmental stores can thrive in the competitive retail landscape and continue to meet the evolving needs of consumers.