TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING THAPATHALI CAMPUS

Proposal
On
Stock Management System on C

Submitted by:

[Prithak Lamsal] [THA081BCT023]

[Pratyush Khatiwada] [THA081BCT022]

Submitted to:

Department of Electronics and Computer Engineering

Thapathali Campus Kathmandu, Nepal

ABSTRACT

This project presents a Stock Management System developed using the C programming language. The system is designed to streamline inventory management by enabling users to efficiently add, update and delete stock items. It maintains a structured database that stores essential product details, including name, quantity, price. By automating stock tracking, the system minimizes manual errors and enhances the accuracy of inventory control. The implementation of file handling and data structures ensures data integrity and efficient retrieval of information, making the system both reliable and effective.

The project is built with a menu-driven interface, ensuring ease of use and accessibility for users with varying levels of technical expertise. It demonstrates key programming concepts such as data management, structured programming, and user interaction, making it a valuable learning experience in software development. The Stock Management System serves as a practical solution for inventory control, contributing to the optimization of stock monitoring and management processes. Through this project, the application of C programming in real-world problem-solving is effectively showcased, highlighting its relevance in system automation and data processing.

Table of Contents

ABSTRACT i

INTRODUCTION 1

1.1 Background Introduction 1

1.3 Problem Definition 2

1.4 Objectives 2

1.5 Scope and Applications 3

2. LITERATURE REVIEW 3

2.1 The history of the Stock Management System 3

2.2 The History and Development of the C Programming Language 4

2.3 The Methods and Techniques of Creating a Stock Management System Using C Program that Can Run on a Terminal 6

3. PROPOSED SYSTEM ARCHITECTURE 7

3.1 Included Header Files 7

3.2 Functionality Breakdown 7

4. Feasibility Analysis 9

Reference 9

INTRODUCTION

1.1 Background Introduction

A language is a system of communication. Programming language is a type of language which is used to communicate with computer. There are many types of programming language by which we can instruct the computers to do some tasks. The history of programming language goes back to the 19th century. The first programming language used in 1950 was FORTAN. This is used for the engineering calculations.

Programming language are divided into 2 types i.e low level programming language and high level programming language based upon the varying with their closeness to hardware of the machine.

1. Low level Language: low level language is programming language which deals with the hardware components of a computer. The code written in low level languages are machine dependent that means which works only on specific machines. Its further divided into machine and assembly language. In machine level language, instructions are written in the form of 0 and 1, where 0 means the absence of current and 1 means the present of current. This codes are difficult to understand. In assembly language, we used symbols of machine code, but cannot directly understood by the computer.
2. High level language: A high level language is a user friendly and machine independent type of language. They are more similar to natural language (such as English). High level language must be translated by interpreter or compiler to understand by the computer. High level language includes C++, C, java etc.

Today there is wide uses different type of programming language in different purpose including java for app development and for web development html, css etc. There are many other programming languages in use today, each with its strengths and weaknesses. Choosing the right programming language for a particular task depends on a variety of factors, including the requirements of the project, the developer's experience and expertise, and the availability of tools and libraries.

In 1960s programs such as COBOL, BASIC was developed to write applications for business and research. Similarly in 1970s C and Pascal was developed to write about operating system and applications. In 1990s development of scripting language such as python, perl which are used for web development.

1.2 Motivation

As new ideas grew up, we decided to enter in simple management system development. We looked out for tutorial videos on YouTube and started to learn more about it. This lead us to this project untimely. It was a quite good opportunity for us to test ourselves in early phase of our learning. In the decision phase of the project, we decided to develop Stock Management System which was best for us to learn something new. We believe the project will be big success for us and will further build our confidence level for upcoming new projects in later years.

1.3 Problem Definition

Manual stock management is prone to errors, inefficiencies, and data inconsistencies, making inventory tracking time-consuming and unreliable. Issues such as overstocking, shortages, and inaccurate records can disrupt operations and lead to financial losses.

To address these challenges, this project introduces a Stock Management System using C programming. The system automates inventory tasks, including adding, updating, deleting, and searching for stock items. With structured data storage and file handling, it ensures accuracy, efficiency, and easy retrieval of stock information, improving overall inventory control.

1.4 Objectives

Throughout this project, we intend to accomplish the following goals:

- To develop an efficient stock management system that automates inventory tasks such as adding, updating, deleting, and searching for stock items.
- To ensure accuracy and reliability in stock tracking by implementing structured data storage and file handling.
- To minimize manual errors and inefficiencies associated with traditional inventory management methods.
- To provide a user-friendly, menu-driven interface for ease of use and accessibility.
- To enhance data retrieval and management for better decision-making in inventory control.
- To demonstrate the practical application of C programming in solving real-world inventory management challenges.

By undertaking this endeavor, participants can anticipate gaining valuable insights related to foundational C programming principles alongside honing strategic reasoning and resolution tactics through an entertaining medium. Concurrently, the accessibility factor ensures compatibility across diverse computing devices containing a terminal and C compiler.

1.5 Scope and Applications

This project focuses on teaching novice programmers the fundamentals of the C programming language and enhancing their problem-solving skills by building a simple but engaging program. This Stock Management System automates inventory processes, including adding, updating, deleting, and searching stock items. It ensures accuracy through structured data storage and file handling while providing a user-friendly interface for efficient stock tracking. The system is designed for small to medium-sized businesses or institutions requiring basic inventory control.

2. LITERATURE REVIEW

The literature review of this project will cover the following topics:

- The history of the Stock Management System.

- The history and development of the C programming language.

- The methods and techniques of creating Stock Management System using C program that can run on a terminal.

2.1 The history of the Stock Management System

The concept of stock management dates back to ancient times when traders and merchants manually recorded inventory using clay tablets, tally sticks, and handwritten scrolls. These early systems were simple but highly prone to errors, leading to difficulties in tracking goods accurately. As commerce expanded, civilizations developed basic accounting systems to maintain records of goods traded, stored, and sold. However, these manual methods remained inefficient, requiring meticulous bookkeeping and frequent stock audits to prevent mismanagement.

During the Industrial Revolution (18th–19th century), large-scale manufacturing and global trade created a greater need for efficient inventory management. Businesses started using mechanical registers, punch cards, and early bookkeeping machines to streamline stock tracking. While these methods were an improvement, they still required extensive human effort and were unable to provide real-time updates. By the mid-20th century, technological advancements led to the introduction of computing systems and databases, significantly enhancing inventory control. Companies began using electronic record-keeping, allowing them to store and retrieve stock information with greater accuracy and efficiency.

A significant breakthrough occurred in the 1970s with the introduction of barcodes and scanning technology. Barcodes allowed businesses to automate inventory tracking, minimizing human error and accelerating the process of stock monitoring. With barcode scanners integrated into point-of-sale (POS) systems, companies could instantly update inventory records, reducing discrepancies between recorded and actual stock levels. Throughout the 1980s and 1990s, businesses adopted Enterprise Resource Planning (ERP) systems, which integrated inventory management with other business operations, providing real-time stock updates and analytics.

With the rise of the internet and cloud computing in the 21st century, stock management became even more sophisticated. Cloud-based systems enabled businesses to track inventory remotely, access real-time data from multiple locations, and automate stock replenishment through predictive analytics. Advanced AI-driven inventory management systems now use machine learning algorithms to analyze sales patterns, forecast demand, and optimize stock levels. These innovations have significantly improved efficiency, reducing waste, overstocking, and financial losses.

The Stock Management System developed in C aligns with these advancements by offering a structured, automated approach to inventory control. While it may not include complex AI-driven features, it serves as a fundamental and practical solution for small businesses, educational institutions, and organizations needing a reliable stock management tool. By integrating structured data storage, file handling, and a user-friendly interface, this system demonstrates the importance of automation in inventory tracking and management, contributing to the ongoing evolution of stock management technology.

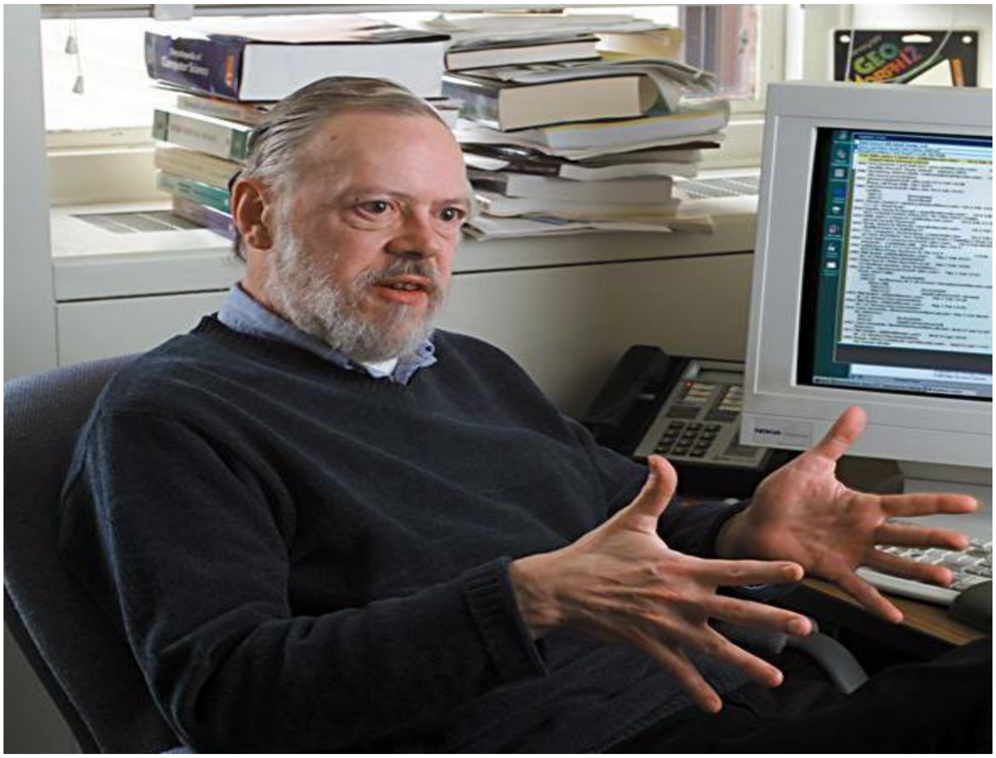
2.2 The History and Development of the C Programming Language

The C language is general purpose and procedural oriented program type of language. It is structured and machine independent language. It was developed by **Dennis Ritchie** in 1972 at the AT&T Bell Laboratories. It was developed along with the UNIX operating system, and is strongly linked with UNIX operating system. History of C language revolves around development as a system implementation language to write an operating system. In terms of the history of C language, its main features include low-level memory access as well as high-level memory access (so it is a middle-level programming language), a handy set of keywords, and a neat and clean style, these features make C programming language suitable for system programming. C supports a wide variety of built-in functions, standard libraries and header files. It follows a top-down approach. Many languages have derived syntax directly or indirectly from the C programming language. For example, C++ is closely a superset of the C language. Also, C programming language is very popular for system-level apps.

C got its name ‘C’ as it incorporates features from its predecessor “B” also originating at Bells labs. C significantly influences the development of many other programming language especially C++ , designed as an extension of C. The programming C is maintained by the standard committees such as the ISO C . C has various standards that meticulously define its syntax, features and behavior including ANSI C(C89 or C90), C99, C11, C17 and C18.

History timeline of C programming:

1972	C	C was first released.
1978	K&R	The first edition of “The C Programming Language” book by Brian Kernighan and Den Ritchie, often referred to as K&R C. It served as a reference until a formal standard w established.
1989	C-89/ ANSI C	ANSI standardized the language (ISO/IEC 9899:1990), introducing standard libraries a features like function prototypes, void pointers, etc.
1990	C-90	Minor updates and fixes on C89.
1999	C-99	Support for variable length arrays, new data types, and inline functions (ISO/IEC 9899:199
2011	C-11	Introduction of additional data types, multi-threading, and improved Unicode (ISO/I 9899:2011).
2018	C-18	Minor updates and fixes on C11 (ISO/IEC 9899:2018).



2.3 Methods and Techniques of Creating a Stock Management System Using C Program that Can Run on a Terminal

1. This stock management system in C uses sqlite3 for database management and openssl/evp.h for password hashing with SHA-256.

2. It includes user authentication (Login, Signup, Logout) and stock operations (CheckStocks, AddStocks, SellStocks). sqlite3 handles data storage, ensuring stocks and users are managed securely.

3. The main menu (mainscreen) directs users to login/signup, while Dashboard provides stock management options.

4. The system ensures data integrity through prepared SQL statements and error handling, allowing users to securely add, check, and sell stocks.

5. The program uses system("clear") for a clean console interface.

3. PROPOSED SYSTEM ARCHITECTURE

Included Header Files

#include <stdio.h> → Used for standard input and output (e.g., printf, scanf).
#include <stdlib.h> → Used for memory allocation, system commands (system("clear")), and exit functions.
#include <string.h> → Provides string manipulation functions (strcmp, strcpy, etc.).
#include <unistd.h> → Provides system call functions like sleep().
#include <sqlite3.h> → SQLite3 database is used for storing user authentication and stock data.
#include <openssl/evp.h> → OpenSSL library is used for password hashing with SHA-256.

Functionality Breakdown

The system consists of different functionalities categorized into authentication, stock management, and user interaction.

Authentication System

Signup (Signup function)

1. Users enter a username and password (password entered twice for confirmation).
2. The password is hashed using SHA-256 for security (sha256() function).
3. The user is stored in the Users table in users.db.

Login (Login function)

1. Users enter a username and password.
2. The password is hashed and compared with the stored hashed password in the Users table.
3. If a match is found, the user is taken to the Dashboard.

Password Hashing (sha256 function)

1. Uses OpenSSL EVP API to hash passwords securely before storing them.

Database Setup

1. The SQLite3 database (users.db) is used to store users and stock information.
2. Two tables are used:
   Users table → Stores username and hashed passwords.
   Stocks table → Stores stock items, quantities, and unit prices.
3. The createTable() function ensures that the Users table exists before storing user data.

Stock Management Functions

Checking Stocks (CheckStocks function)

1. Fetches all stock data from the Stocks table.
2. Uses sqlite3_exec() with a callback function to display stock records.
3. Provides navigation options to return to the Dashboard or log out.

Adding Stocks (AddStocks function)

1. Allows users to enter a new stock name, quantity, and unit price.
2. If the Stocks table does not exist, it is created.
3. Inserts the stock data using sqlite3_prepare_v2() and sqlite3_step().

Selling Stocks (SellStocks function)

1. Users specify the stock name and quantity to sell.
2. The price per unit is retrieved from the database table.
3. The stock quantity in the table is updated.
4. A billing system calculates the total cost for all sold items and displays a formatted bill.

Navigation & User Interface

1. The mainscreen() function presents the login/signup menu.
2. The Dashboard() function provides options for:
   - Checking Stocks
   - Adding Stocks
   - Selling Stocks
   - Logging Out
3. Users navigate using integer inputs.

Key Features

Secure Authentication:

1. Uses SHA-256 hashing for storing passwords securely.
2. Prevents storing plain-text passwords.

SQLite3 Database Integration:

1. Persistent storage for users and stocks.
2. sqlite3_prepare_v2(), sqlite3_step(), and sqlite3_finalize() are used for database queries.

Stock Operations:

1. Users can check available stocks.
2. Add new stock with unique names.
3. Sell stock and generate bills.

User-Friendly Console Interface:

1. Uses system("clear") for clean UI.
2. Provides navigation options for dashboard access.

Summary of Execution Flow

1. Program starts (main()) → Calls mainscreen().
2. User logs in / signs up (using Login() or Signup()).
3. Dashboard options allow:

Checking stock (CheckStocks())
Adding stock (AddStocks())
Selling stock (SellStocks())
Logging out (Logout())

4. Stock operations interact with the database (users.db).
5. User is continuously prompted until they exit.

4. Feasibility Analysis

Technical Feasibility: Developing a basic version of the stock management system using the programming language C is a highly achievable goal from both technical and economic standpoints. From a technical perspective, C is widely recognized as one of the most suitable languages for program development due to its flexibility, efficiency, and wide range of libraries and frameworks available for gaming applications. Moreover, this specific project focuses on implementing fundamental concepts of the C language, making it relatively easy for beginners to comprehend and execute.

Economic Feasibility: In addition to being technically feasible, creating a Stock Management System using C is also financially within reach. To get started, all that is required is a C compiler, which can be acquired at no cost whatsoever. One can either download a C compiler from various sources on the internet or utilize an online compiler without having to install anything on their local machine. Therefore, there are virtually no expenses involved in developing this project, making it an excellent choice for those who may have limited financial resources.

Operational Feasibility: Lastly, operational feasibility is another aspect to consider while building a Stock Management System using C. After writing and compiling the code successfully, running the game on a computer equipped with a C compiler should pose no difficulties. To ensure ease of maintenance, the project's code will be thoroughly documented and structured appropriately, enabling smooth execution, modification, and updating in the future if necessary. With these factors in mind, developing a basic version of the iconic game Pacman using C seems like a promising endeavor worth pursuing.

Reference:

- freecodecamp.org
- GeeksforGeeks | A computer science portal for geeks
- GitHub · Build and ship software on a single, collaborative platform · GitHub
- Stock Management System Project using C language - Studytonight