



DCIT308 PROJECT

Individual Project

TABLE OF CONTENT

PROJECT OVERVIEW.....	1
FUNCTIONALITY OF THE SYSTEM.....	3
IMPLEMENTATION.....	5
TESTING.....	7

University Of Ghana
Department Of Computer Science
DCIT308 – Data Structures and Algorithms II
Individual project

PROJECT OVERVIEW

One of the basic requirements for running a business is an inventory management system. Management of inventories is essential to any organization. Lack of such a system could lead to a number of issues, including missing out on important sales opportunities, holding onto too much inventory, or not having enough stock on hand. Inventory management system for a provision store using data structures (Lists, stacks, arrays, queues, trees, sets-maps, hash maps, hash tables and graphs.

INTRODUCTION

For companies that deal with products or goods, an inventory management system is an essential tool. The processes involved in managing inventory, from purchase through storage, tracking, and sales, are meant to be streamlined and optimized. With the help of this system, businesses can see their inventory levels in real-time, enabling them to make wise decisions and successfully satisfy client requests.

Businesses can minimize inventory shortages, surplus inventory, and inefficient inventory transfer between sites with the aid of IMSs. A company's organization is maintained through effective inventory management, which also provides vital information that enables companies to adapt to trends, prevent supply chain management failures, and preserve profitability.

TASK AND PURPOSE OF THE SYSTEM

Our inventory management system is designed to handle the addition and deletion of goods and vendors. It works by recording the following information about each good and vendor.

Id: Good and vendor IDs will be an integer obtained from a JTextField in the code. Each ID will be unique to that specific good or vendor

.

Name: Good and Vendor Name will be a String also obtained from a JTextField in the code. Each name will be registered in the inventory management system every time a new good or vendor is added.

Categories: This applies only for goods and will be a string obtained from JComboBox. Each good will fall under one of our specified categories.

Price: Price of goods will vary according to the good in question. Price will also be obtained from a JTextField with a data type of integer.

Phone: This refers to vendor phone numbers that will be collected into the system

Address: This refers to vendor address that will be collected as part of the system.

The features that the inventory management system will have includes:

- A login interface that accepts an admin's username and password in order to access the main system.
- A homepage from which the admin can navigate to either the goods page, vendor page or bills page.
- A goods page, where the admin can add, delete or edit goods to or from a database table, displaying for the admin's view all the information about the good collected.
- A vendors page that will allow the admin to view the vendors the store purchases from as well as add vendors when there is a new vendor, edit vendor information and delete a vendor when the store no longer works with said vendor.

IMPLEMENTATIONS

The JAVA programming language was used to create our inventory management system. Strong general-purpose programming languages like Java are frequently used to create desktop and mobile applications, massive data processing, embedded devices, and other things.

2020 (Programiz)

NetBeans was the IDE used to build the system. Swing Framework-based Java desktop, mobile, and web apps may be created quickly and easily with NetBeans.

In the development of this system, stacks, queues, and array lists were used as data structures.

A stack is a type of linear data structure that adheres to the Last In First Out (LIFO) tenet. This implies that the last piece to be added to the stack gets eliminated first. To add and remove items from the database, stacks were used.

A queue is a linear data structure that progressively stores the entries. It accesses elements using the FIFO method (First In First Out). When adding vendors to the database and removing vendors from the database, queues were used.

A versatile resizable array is an arrayList. It offers a number of beneficial functions that work with the data it holds. The project uses several ArrayList methods, such as add(), get(), and remove().

The vendors were stored in the system using hash maps. Using the key value that was saved, the vendor stored in the hash table or hash map is retrieved.

In order to find matching information, binary search techniques were employed to search the list of vendors and products according to their IDs.

Products were sorted by price or category using selection sort, while sellers were sorted using names.

TEST, EVALUATION & RECOMMENDATIONS

Objective	Compile and run the java program from Apache NetBeans
Action	Executing the java program through Apache NetBeans by
opening software	going to the location of the program and
compiling and running the java package.	
Expected Result	The system should compile and run without any errors
Actual Result	The program was compiled and run successfully.
Conclusion	Test Successful

Login with admin username and password

Objective	Login to Admin Panel when correct username and password
entered	
Action	Admin username and password were entered into text field and
Login Button was clicked	
Expected Result	Successful Login
Actual Result	As correct username and password were entered Logged in
Conclusion	Test Successful

Adding goods to the table

Objective	Add Item in Table when all the jTextField, jCombobox
jRadioButton were filled and Add Item Button get clicked	
Action	After filling all the jTextFieldj, ComboBox, jRadioButton item
was added in Table as Add Item Button was clicked	
Expected Result	Item should be added in table as Add Item Button was clicked
after filling all jTextField, jComboBox ,jRadioButton	
Actual Result	Item get added after filling jTextField, jComboBox, RadioButton
and clicked in Add Item Button	
Conclusion	Test Successful

Deleting goods from table

Objective	Deleting a row from table.
Action	Selecting a row from a table and clicking on Delete button and

clicking on Yes.	
Expected Result	The row must be deleted without any errors.
Actual Result	The row was deleted without any errors.
Conclusion	Test Successful.

LIMITATIONS

Complex Data Management: Managing an inventory includes dealing with a lot of data, such as product information, stock levels, sales information, and more. It can be difficult to manage and organize this data, especially when there are many different goods, variations, and locations involved. In order to ensure effective data management and retrieval, thorough planning and design are required.

TIME COMPLEXITY

Data Structure Time Complexity Space Complexity

Stack	$O(1)$
Queue	$O(1)$
Lists	$O(n)$
Binary Search	$O(1)$ $O(1)$
Selection sort	$O(1)$ $O(1)$