“UNIVERSITY OF GHANA

SCHOOL OF PHYSICAL & MATHEMATICAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

A blue and yellow emblem

Description automatically generated

DCIT 308: DATA STRUCTURES AND ALGORITHMS II

INVENTORY MANAGEMENT SYSTEM

ID: 10865289

DATE: 15/07/2023.

Contents

[PROJECT OVERVIEW 3](#_Toc140566773)

[INTRODUCTION 3](#_Toc140566774)

[TASK AND PURPOSE OF THE SYSTEM 3](#_Toc140566775)

[IMPLEMENTATIONS 4](#_Toc140566776)

[TEST & EVALUATION 5](#_Toc140566777)

# PROJECT OVERVIEW

This project aims to design and implement a software application that can efficiently store, track and manage the inventory of various products in a shop. The project will use different data structures such as lists, stacks, arrays, queues, trees, sets-maps, hash maps, hash tables and graphs to organize and manipulate the inventory data. The project will also provide features such as searching, sorting, filtering, updating, and reporting on the inventory data. The project will use Java as the programming language and MySQL as the database management system.

# INTRODUCTION

An inventory system is a system that tracks the quantity and value of the items that a business sells or uses. It can help to optimize the inventory levels, reduce costs, improve customer service, and prevent stockouts or overstocking. An inventory system can be manual or automated, periodic, or perpetual, and use different methods such as FIFO, LIFO, or weighted average to value the inventory.

# TASK AND PURPOSE OF THE SYSTEM

The inventory system tracks the quantity and value of the items in stock, records the transactions of purchasing and selling, and generates reports on inventory performance. The purpose of the system is to help the business manage its inventory efficiently, reduce costs, improve customer satisfaction, and support decision making. 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 include:**

• A login interface that accepts an admin’s username and password 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 vendor’s 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

This inventory management system was programmed using the JAVA programming language. Java is a powerful general-purpose programming language often used to develop desktop and mobile applications, big data processing, embedded systems, and so on. The IDE used in creating the system was NetBeans. NetBeans lets you quickly and easily develop Java desktop, mobile, and web applications using Swing Framework.

The data Structures used in the creation of this system were Stacks, Queues, and Array List. A stack is a linear data structure that follows the principle of Last In First Out (LIFO). This means the last element inserted inside the stack is removed first. Stacks were used to add and remove goods to and from the database. A queue is a linear data structure that stores the elements sequentially. It uses the FIFO approach (First in First Out) for accessing elements. Queues were implemented in the addition of vendors to the database and the deletion of vendors from the database. ArrayList is a general-purpose resizable array. It provides several useful features that operate on the data it contains.

Some of the ArrayList methods used in the project include add(), get(), and remove().Hash maps was used in storing of the vendors in the system. The vendor stored in the hash table or hash map is retrieved using the key value which was stored. Binary search methods were used to search the list of vendors and goods according to their IDs to retrieve matching data. Selection sort was used to sort goods by price or category and to sort vendors by name.

# TEST & EVALUATION

Testing that the program can be compiled and run using the IntelliJ IDE.

|  |  |
| --- | --- |
| 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 jTextField jComboBox 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 jComboBoxj ,jRadioButton |
| Actual Result | Item get added after filling JTextField jComboBoxj, JRadio Button 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. |

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) |

LIMITATIONS

Implementation of suitable algorithms could not be done efficiently.”