

NAME: NSABIMANA Peter

REG NO: 221022891

COLLEGE: CBE

FACULTY: BIT

CLASS NO: 7

Group No: 7

Module: Advanced programming in java

PROJECT NAME: STOCK MANAGEMENT SYSTEM

PROJECT DESCRIPTION DOCUMENT

1. PLANNING

The goal of this project was to develop a stock management system that can be used to track ins and outs of products in our stock and generate an overview of stock status, which will be used for reporting purposes. The main objective of this stock management system was to create a user-friendly interface with functionalities that allows easy management of our stock and accurate tracking of the current stock status. This system also aims to solve the problem of manual tracking of our physical stock, which can lead to errors and inaccuracies, the system will save us a lot of time and will perform quick and accurate calculations in a few seconds. The future system will focus on the improvement of the user interface and adoption of more different technologies, which will improve reporting techniques and make the process easier than today.

2. Design

Our Stock Management System have a beautiful and user-friendly interface designed by using Apache NetBeans IDE 15 and backend codes are written using NetBeans as well. The system is composed by 8 different interfaces which are Login, categories, shop, suppliers, product, productIn, productOut and report which will show the current status of our stock. Stock Manager will login by using our login form which will first check if username and password any record in database, if it matches the form will direct stock manager to the report form or overview where he will get necessary real-time information about the current situation of stock. He will also be able to navigate to other interfaces to manage (record, update, delete) categories, shop, suppliers, product, productIn and productOut's information. All recorded data will be saved to mysql database and retrieved in tables at their

corresponding interface page. On categories interface Stock Manager will be able to record new category, update existing category and delete any category as he/she want. On shops page stock manager will also be able to record new shop if there is one, view the available shops, update shop information and delete some shop information as long as it is necessary. On all pages, also Stock manager will have ability to perform data management activities as he or she can on the above interface pages.

As we highlighted in planning stage the main purpose of the project was to develop stock management information to track ins and outs of product in our stock. In our case, we have a big stock and many shops to which our stock will supply products to be sold out, and the stock will take in products from different suppliers. So here the main functionality is Stock manager will record products received from suppliers by using productIn form found on productIn page and changes will be reflected to products page where we can see the available products in stock, to do that he will choose a supplier who is supplying products fill other information related to products like cost, quantities and so on and click save for that products to be added in stock. Also when some quantity of products is to be taken to one of our shops Stock manager will also use productOut form found on productOut page, he or she will first choose a shop where we are going to take products and fill other information like quantity and so on and the that quantity will be deducted from what was the current quantity that particular product.

Validation also works properly you can not take out quantity of the product higher than the quantity available, The system will give you a error message telling you quantity you want to take is not enough and it will show you the available quantity in that message.

As our system will be used by a single user who manage our stock to interact with the system he or she will need to login first by using his or her own login credentials, and then he can access all other functionalities like record information update or delete.

Functional and non-functional requirements are implemented in this system. For instance system have user friendly interface which is easy to navigate, security as only system administrator can log in the system, speed as the system process data very quickly and many more requirements.

3. Development

Our stock management system was developed by using Apache NetBeans IDE 15 and java programming language. NetBeans is java-based integrated development environment (IDE) and such it uses java swing library to create the user interface (UI) for java applications. Swing provides a rich set of user interface components such as buttons, password fields, text fields, tables, labels and many more, that can be easily added to java application using drag-and-drop functionality in NetBeans, with that functionality we have designed front-end system with a good looking user interface by drag-and-drop necessary components like if we needed a form we drag-and-drop components like labels, text fields, and buttons that will be used to submit form data even displaying data from database into tabale component was done well. NetBeans also provide source code text editor for writing and editing java code with syntax highlighting, code completion, error highlighting and refactoring. That is where we have written all back-end codes including functions to record retrieve and update data in the database as well as other logic to handle system's functionalities.

With the use of MYSQL Connector Jar file we was able to connect the back-end code with the database, allowing the system to communicate and perform operations like inserting data in and retrieving data from database. XAMMP server was used to design MySQL database called stock with 7 tables such as **admin** (with fields: id PK, firstname, lastname, username, phone, email, password, address and created_at), **category** (with fields: id PK, name, description, created_at), **product** (with fields: id PK, barcode PK, category_id FK, name, cost, quantity, total_cost, created_at), **supplier** (with

fields: id PK, firstname, lastname, phone, email, address, created_at), **shop** (with fields: id PK, name, address, created_at), **productIn** (with fields: id PK, barcode FK, supplier_id FK, quantity, cost, total, entry_date) and **productOut** (with fields: id PK, barcode FK, shop_id FK, quantity, cost, total, exit_date). The above tables were designed with their relationships to ensure effective operation and connection between back-end, front-end and database of our stock management system.

Libraries chosen include Java Standard Edition Libraries (java virtual machine JVM, java development kit JDK and java runtime environment JRE), Swing. Platform such as github is used for version controlling and change tracking. Database is designed by using XAMPP server and it is connected to backend code with the use of MYSQL Connector Jar File and JDBC and storage like github cloud storage was used to store application files and on local machine application and its data well stored correctly.

4. Testing

The system's performance was tested and monitored by using manual testing methods to ensure quick response time and data accuracy. During this stage, we found some serious defects and bugs and worked our best to fix them. Firstly, forms were able to submit empty text fields into database to solve this problem we first checked if user filled some data in text field to processed the process of inserting that new record in database. Secondly, clicking delete or update buttons without selecting a record to delete or update used to cause bugs in the background, to solve this problem before running update or delete code we first check a condition to see if there is exactly a record selected and if yes that specific record would now be deleted or updated and if no record selected the message is displayed telling a user to select a record to delete or update. After fixing the above two bugs system now runs correctly without any bug and it have high accuracy of data as we expected before developing this stock management system.

5. Deployment

Stock Management system was deployed on a local laptop for testing and performance monitoring. Starting from login form it authorize only user who type in username and password that matches any user database record if not it gives you error telling you that you enter incorrect username and or password. We also filled other remaining forms with data and submit them the result was what we expected because data were submitted into database tables exactly the way we wanted. The key functionality of our stock management system was to track ins and outs of products, so when you fill productIn form quantity is added to the current quantity of that product if any and its total cost is calculated by multiplying unit cost and quantity it is also added to what was total cost before saving new record. Again by filling out productOut form system will check if requested quantity of product is available if no it will display message saying Quantity is not enough and if yes it quantity and its total cost will be deducted from what was the quantity and total cost for that specific product.