Contents

[System requirement 2](#_Toc469477275)

[Files and folders 2](#_Toc469477276)

[Folders: 2](#_Toc469477277)

[Files: 2](#_Toc469477278)

[Methods from ShopDatabase.java 4](#_Toc469477279)

[public static void createDbAndConnect() 4](#_Toc469477280)

[public static void ConnectToDB() 4](#_Toc469477281)

[public static void createShopTable() 4](#_Toc469477282)

[public static void insertItem(String item,Double unitPrice,int quantity) 4](#_Toc469477283)

[public static void viewIncomboBox(JComboBox comboBox) 4](#_Toc469477284)

[public static void view(JTextField txt,JComboBox comboBox,DefaultTableModel md) 4](#_Toc469477285)

[public static void viewReturned(JTextField txt,JComboBox comboBox,DefaultTableModel md) 4](#_Toc469477286)

[public static void viewAvailable(DefaultTableModel model) 4](#_Toc469477287)

[public static void deleteStock(JTextField item) 4](#_Toc469477288)

[public static void UpdateStock(JTextField item,JTextField Quan,JTextField Newprice) 5](#_Toc469477289)

[public static void viewTotal(DefaultTableModel model) 5](#_Toc469477290)

[public static void viewQuantity(JTextArea textArea) 5](#_Toc469477291)

# System requirement

Operating System: windows 7/8/10

CPU:500 MHz

RAM: 70 Mb

Java version 7/8

Java JDK version jdk1.8.0\_111/jdk1.8.0\_101

# Files and folders

* All the files and folder are included in the directory shopping app

### Folders:

Client\_shopping\_lib –these are libraries used by the client jar executable

Shopping\_app\_lib- these are libraries used by the Admin jar executable

Grosseri-folder containing database used by the business

Jdbc-swing-gui.zip-the zip folder contains Source code, lib and database files

To run the application in eclipse, red hat extract the jdbc-swing-gui.zip in your workspace and then import existing java project

### 

### Files:

Deby-log file of the database being accessed by the applications  
Guide to client & admin shopping-Guide telling how to use both admin and client application

Technical Guide-this is the file which will guide the user technically

Admin password-A file containing Admin password and username

Client\_shopping

When you want to run the program as a consumer you should run the Client\_shopping

.jar

Admin\_shopping

When you are the manager of the shop and you want to make some changes in the business run the Admin\_shopping.jar

# Methods from ShopDatabase.java

### **public** **static** **void** createDbAndConnect()

this is the method mostly used by the administrator when creating a new database in the business and connecting to it

### **public** **static** **void** ConnectToDB()

this method is just a connection to an existing database in the system

### **public** **static** **void** createShopTable()

After running the createDbAndConnect() method this method is called this where a table is created in the database in which will be create by the administrator and the consumer will only have access to the items to the database

### **public** **static** **void** insertItem(String item,Double unitPrice,**int** quantity)

Administrator is responsible for entering data into the database and this method is responsible for interacting the application and the database when entering new item into the shop

### **public** **static** **void** viewIncomboBox(JComboBox comboBox)

method returning the item in the database, this method show the consumer which products are in the shop and it shows them in a comboBox form giving the user a choice to click the product or write description of the product

### **public** **static** **void** view(JTextField txt,JComboBox comboBox,DefaultTableModel md)

when consumer buys products and adds it to the chart this function updates the Quantity of the stock according to the number of item bought of the specific product

### **public** **static** **void** viewReturned(JTextField txt,JComboBox comboBox,DefaultTableModel md)

updates of the Quanity if ever a consumer returns the items back in the shop

### **public** **static** **void** viewAvailable(DefaultTableModel model)

method that helps with monitoring the business database, this helps the administrator to view the items stored in their database

### **public** **static** **void** deleteStock(JTextField item)

this method is meant for the administrator to delete products in the database if the item is no longer sold in the shop

### **public** **static** **void** UpdateStock(JTextField item,JTextField Quan,JTextField Newprice)

if stock reaches 0 in the shop and new stock has been delivered to the shop, this function is called giving the administrator the right to update the Quantity of the stock available in the shop

### **public** **static** **void** viewTotal(DefaultTableModel model)

returns the subtotal of the products bought by the consumer

### **public** **static** **void** viewQuantity(JTextArea textArea)

This method return the remaining stock available in the shop to the consumer so they know when they have reached their limit

The source codes of the program are situated in the Zip folder, extract jdbc-swing-java folder then to path jdbc-swing-gui\src\aa

Files in jdbc-swing-gui\src\aa

* InvoiceClient.java -Source code of the Consumer jar file
* Invoicelogin.java -Administrator GUI login window
* jTableRow.java - Administrator management system GUI window
* shopDatabase.java -Source code that sets connection of the application and database by performing specific tasks