**Class Stock**

**package** Main;

**public** **class** Stock {

**private** String Name;

**private** String ProductID;

**private** **double** Price;

Stock(){

**this**("null","null",00.00);

}

Stock(String Name, String ProductID, **double** Price){

**this**.Name = Name;

**this**.ProductID = ProductID;

**this**.Price = Price;

}

**public** **void** setName(String newName) {

**this**.Name = newName;

}

**public** **void** setProductID(String newProductID) {

**this**.ProductID = newProductID;

}

**public** **void** setPrice(**double** newPrice) {

**this**.Price = newPrice;

}

//get method

**public** String getName() {

**return** **this**.Name;

}

**public** String getProductID() {

**return** **this**.ProductID;

}

**public** **double** getPrice() {

**return** **this**.Price;

}

**public** String toString() {

**return** "<html>Name: " + getName() + "<br>Product ID: " +

getProductID() + "<br>Price: RM "+ getPrice() ;

}

}

**Class Fabric**

**package** Main;

**public** **class** Fabric **extends** Stock{

**private** **int** Quantity;

Fabric(){

}

Fabric(String Name, String ProductID, **double** Price, **int** Quantity){

**super**(Name,ProductID,Price);

**this**.Quantity = Quantity;

}

//set method

**public** **void** setQuantity(**int** newQuantity) {

**this**.Quantity = newQuantity;

}

//get method

**public** **int** getQuantity() {

**return** **this**.Quantity;

}

**public** String getType() {

**return** "Fabric";

}

**public** String toString() {

**return** **super**.toString() + "<br>Quantity: " + getQuantity() + "<br>Type: " + getType() +"</span></html>";

}

}

**Class Electronic**

**package** Main;

**public** **class** Electronic **extends** Stock{

**private** **int** Quantity;

Electronic(){

}

Electronic(String Name, String ProductID, **double** Price, **int** Quantity){

**super**(Name,ProductID,Price);

**this**.Quantity = Quantity;

}

//set method

**public** **void** setQuantity(**int** newQuantity) {

**this**.Quantity = newQuantity;

}

//get method

**public** **int** getQuantity() {

**return** **this**.Quantity;

}

**public** String getType() {

**return** "Electronic";

}

**public** String toString() {

**return** **super**.toString() + "<br>Quantity: " + getQuantity() + "<br>Type: " + getType() +"</span></html>";

}

}

**Class User**

**package** Main;

**public** **class** User {

**private** String username;

**private** String password;

User(String username,String password){

**this**.username = username;

**this**.password = password;

}

**public** String getUsername() {

**return** **this**.username;

}

**public** String getPassword() {

**return** **this**.password;

}

}

**GUI And Logic : Login**

package Main;

import java.awt.EventQueue;

import java.awt.Toolkit;

import javax.swing.JFrame;

import javax.swing.JPanel;

import java.awt.Color;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.ImageIcon;

import java.awt.Font;

import javax.swing.JTextField;

import javax.swing.JPasswordField;

import javax.swing.JButton;

import java.awt.event.FocusAdapter;

import java.awt.event.FocusEvent;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

public class Login {

public JFrame frame;

private JTextField usernametext;

private JPasswordField passwordField;

private static List<User> User= new ArrayList<User>(Arrays.asList(new User("Admin","Admin"),new User("Worker","Worker")));

private char[] passwordvalue;

private String usernamevalue;

private boolean isSuccessfull = false;

//This class for login purpose. This class is the initialization of the application

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

Menu.Data.add(new Electronic("Redmi 3S", "E001", 600,10));

Menu.Data.add(new Fabric("Adidas Cap @Blue", "F001", 30,9));

Menu.Data.add(new Electronic("Iphone X", "E002", 3000,8));

Menu.Data.add(new Fabric("Jubah Naruto", "F002", 55.90,100));

Menu.Data.add(new Electronic("Electric Cooker", "E003", 100,10)); //initialize five products

EventQueue.invokeLater(new Runnable() { //The Application will run in this class first

public void run() {

try {

Login window = new Login(); //go to Login frame

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\*/

public Login() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

private void initialize() {

//customize jFrame

frame = new JFrame();

frame.setResizable(false);

frame.getContentPane().setBackground(new Color(255, 255, 255));

frame.setIconImage(Toolkit.getDefaultToolkit().getImage(Menu.class.getResource("/res/warehouse.png")));

frame.setTitle("Stock Management System");

frame.setBounds(450, 200, 900, 700);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

//left side panel

JPanel panel = new JPanel();

panel.setBackground(new Color(128, 128, 128));

panel.setBounds(0, 0, 375, 665);

frame.getContentPane().add(panel);

//icon label

JLabel icon = new JLabel("");

icon.setBounds(55, 190, 265, 272);

icon.setIcon(new ImageIcon(Login.class.getResource("/res/warehouse.png")));

//title label

JLabel lblStockManagementSystem = new JLabel("STOCK MANAGEMENT\r\n");

lblStockManagementSystem.setBounds(24, 92, 326, 42);

lblStockManagementSystem.setForeground(new Color(255, 255, 255));

lblStockManagementSystem.setFont(new Font("Bebas Neue", Font.PLAIN, 50));

panel.setLayout(null);

panel.add(lblStockManagementSystem);

panel.add(icon);

//title label

JLabel lblSystem = new JLabel("SYSTEM");

lblSystem.setForeground(Color.WHITE);

lblSystem.setFont(new Font("Bebas Neue", Font.PLAIN, 50));

lblSystem.setBounds(126, 147, 122, 42);

panel.add(lblSystem);

//user name label

JLabel lblUsername = new JLabel("Username: ");

lblUsername.setFont(new Font("Century Gothic", Font.BOLD, 21));

lblUsername.setBounds(454, 250, 115, 43);

frame.getContentPane().add(lblUsername);

//user name textField

usernametext = new JTextField();

usernametext.addFocusListener(new FocusAdapter() {

@Override

public void focusLost(FocusEvent e) {

usernamevalue = usernametext.getText(); //get user name from usernametext

}

});

usernametext.setFont(new Font("Brandon Grotesque Light", Font.PLAIN, 21));

usernametext.setBounds(581, 250, 222, 43);

frame.getContentPane().add(usernametext);

usernametext.setColumns(10);

//password label

JLabel lblPassword = new JLabel("Password:");

lblPassword.setFont(new Font("Century Gothic", Font.BOLD, 21));

lblPassword.setBounds(454, 331, 115, 43);

frame.getContentPane().add(lblPassword);

//password field

passwordField = new JPasswordField();

passwordField.addFocusListener(new FocusAdapter() {

@Override

public void focusLost(FocusEvent e) {

passwordvalue = passwordField.getPassword(); //get password from passwordField

}

});

passwordField.setFont(new Font("Brandon Grotesque Light", Font.PLAIN, 21));

passwordField.setBounds(581, 331, 222, 43);

frame.getContentPane().add(passwordField);

//login title

JLabel Login = new JLabel("LOGIN\r\n");

Login.setForeground(new Color(255, 127, 80));

Login.setFont(new Font("Brandon Grotesque Black", Font.PLAIN, 38));

Login.setBounds(454, 149, 147, 59);

frame.getContentPane().add(Login);

//button login

JButton btnLogin = new JButton("Login");

btnLogin.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

for(int i=0; i<User.size(); i++) {

//check the user name and password match in the database or not

if(User.get(i).getUsername().equals(usernamevalue) && User.get(i).getPassword().equals(String.valueOf(passwordvalue))) {

JOptionPane.showMessageDialog(null, "Login Successful");

isSuccessfull = true; //if matched isSuccessfull=true

break;

}

}

if(isSuccessfull==true) { //if isSuccessful is true, the user can access the menu

try{

Menu menu =new Menu();

frame.dispose();

menu.frame.setVisible(true);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

else if(isSuccessfull==false){ //if not show dialog login unsuccessful

JOptionPane.showMessageDialog(null, "Login Unsuccessful");

passwordField.setText("");

usernametext.setText("");

}

}

});

btnLogin.setFont(new Font("Brandon Grotesque Bold", Font.PLAIN, 21));

btnLogin.setForeground(new Color(255, 255, 255));

btnLogin.setBackground(new Color(255, 69, 0));

btnLogin.setBounds(707, 417, 106, 43);

frame.getContentPane().add(btnLogin);

JLabel ask = new JLabel("Do not have username?");

ask.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

JOptionPane.showMessageDialog(null, "Please contact Administrator"); //showing the message to contact the administrator

}

});

ask.setFont(new Font("Century Gothic", Font.PLAIN, 13));

ask.setBounds(534, 429, 168, 16);

frame.getContentPane().add(ask);

JLabel aboutlabel = new JLabel("");

aboutlabel.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

try{

About about =new About(); //go to about frame

about.setVisible(true);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

aboutlabel.setIcon(new ImageIcon(Login.class.getResource("/res/support.png")));

aboutlabel.setBounds(821, 595, 73, 70);

frame.getContentPane().add(aboutlabel);

}}

**GUI And Logic : Menu**

package Main;

import java.awt.Color;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

import java.util.List;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import java.awt.GridLayout;

import java.awt.Font;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

public class Menu {

public JFrame frame;

private JButton btnAdd,btnUpdate,btnDisplay, btnDelete;

public static List<Stock> Data = new ArrayList<Stock>();

private JLabel aboutlabel;

//this frame is for showing the available menus

public Menu() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

private void initialize() {

//customize jFrame

frame = new JFrame();

frame.setIconImage(Toolkit.getDefaultToolkit().getImage(Menu.class.getResource("/res/warehouse.png")));

frame.setTitle("Stock Management System");

frame.setBounds(450, 200, 1000, 700);

frame.setResizable(false);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.getContentPane().setLayout(new GridLayout(0, 1, 0, 0));

//panel background

JPanel panel = new JPanel();

panel.setBackground(Color.LIGHT\_GRAY);

frame.getContentPane().add(panel);

btnAdd = new JButton("ADD");

btnAdd.setFont(new Font("Century Gothic", Font.BOLD, 18));

btnAdd.setForeground(new Color(0, 0, 0));

btnAdd.setBackground(new Color(255, 255, 255));

btnAdd.setBounds(386, 289, 210, 56);

btnAdd.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try{

Add add =new Add(); //go to Add frame

frame.dispose();

add.frame.setVisible(true);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

btnUpdate = new JButton("UPDATE");

btnUpdate.setFont(new Font("Century Gothic", Font.BOLD, 18));

btnUpdate.setBackground(new Color(255, 255, 255));

btnUpdate.setBounds(387, 380, 210, 56);

btnUpdate.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try{

Update update =new Update(); //go to Update frame

frame.dispose();

update.setVisible(true);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

btnDisplay = new JButton("DISPLAY");

btnDisplay.setFont(new Font("Century Gothic", Font.BOLD, 18));

btnDisplay.setBackground(new Color(255, 255, 255));

btnDisplay.setBounds(387, 469, 210, 56);

btnDisplay.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try{

Display display =new Display(); //go to Display frame

frame.dispose();

display.setVisible(true);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

btnDelete = new JButton("DELETE");

btnDelete.setFont(new Font("Century Gothic", Font.BOLD, 18));

btnDelete.setBackground(new Color(255, 255, 255));

btnDelete.setBounds(390, 557, 207, 56);

btnDelete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try{

Delete delete =new Delete(); //go to Delete frame

frame.dispose(); //dispose current window without terminating the application

delete.setVisible(true);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

panel.setLayout(null);

panel.add(btnUpdate);

panel.add(btnDelete);

panel.add(btnDisplay);

panel.add(btnAdd);

JLabel lblStockManagementSystem = new JLabel("Stock Management System");

lblStockManagementSystem.setForeground(new Color(255, 255, 255));

lblStockManagementSystem.setFont(new Font("Bebas Neue", Font.PLAIN, 85));

lblStockManagementSystem.setBounds(112, 35, 757, 89);

panel.add(lblStockManagementSystem);

JLabel icon = new JLabel(" ");

icon.setBounds(423, 137, 136, 138);

icon.setIcon(new ImageIcon(Menu.class.getResource("/res/warehouse128.png")));

panel.add(icon);

aboutlabel = new JLabel("");

aboutlabel.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

try{

About about =new About(); //go to About frame

about.setVisible(true);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

aboutlabel.setIcon(new ImageIcon(Menu.class.getResource("/res/support.png")));

aboutlabel.setBounds(909, 580, 73, 70);

panel.add(aboutlabel);

JLabel logout = new JLabel("");

logout.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

try{

Login login = new Login(); //go to Login frame

frame.dispose();

login.frame.setVisible(true);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

logout.setIcon(new ImageIcon(Menu.class.getResource("/res/logout-or-send-square-with-right-arrow-interface-symbol.png")));

logout.setBounds(931, 530, 39, 46);

panel.add(logout);

}

}

**GUI And Logic : Add**

package Main;

import javax.swing.JFrame;

import java.awt.Color;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import java.awt.Toolkit;

import javax.swing.JTextField;

import javax.swing.JButton;

import javax.swing.JCheckBox;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import java.awt.event.FocusAdapter;

import java.awt.event.FocusEvent;

public class Add {

public JFrame frame;

private JTextField name;

private JTextField productid;

private JTextField quantity;

private JTextField price;

private JButton btnSave;

private JButton btnBack;

private JLabel lblType;

private String namevalue;

private String idvalue;

private int quantityvalue;

private double pricevalue;

private int type;

private JButton btnClear;

private Boolean isIdEntered = false;

private Boolean isNameEntered = false;

//This Class is for function Add Item

public Add() {

initialize();

}

/\*\*

\* Initialize the contents of the frame.

\*/

private void initialize() {

//customize frame

frame = new JFrame();

frame.setResizable(false);

frame.setIconImage(Toolkit.getDefaultToolkit().getImage(Menu.class.getResource("/res/warehouse.png"))); // This so our program's icon

frame.setTitle("Stock Management System");

frame.getContentPane().setFont(new Font("Century Gothic", Font.BOLD, 21));

frame.getContentPane().setBackground(Color.LIGHT\_GRAY);

frame.setBounds(450, 200, 1000, 700);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

//label name

JLabel lblName = new JLabel("Name:");

lblName.setFont(new Font("Century Gothic", Font.BOLD, 21));

lblName.setBounds(203, 152, 119, 41);

frame.getContentPane().add(lblName);

//name textField

name = new JTextField();

name.setFont(new Font("Franklin Gothic Book", Font.PLAIN, 21));

name.addFocusListener(new FocusAdapter() {

@Override

public void focusLost(FocusEvent e) {

if(!(name.getText().equals(""))) {

namevalue = name.getText(); //get product name from JTextField

isNameEntered = true; //indicate name is entered

}

}

});

name.setBounds(334, 152, 426, 41);

frame.getContentPane().add(name);

name.setColumns(10);

//product id label

JLabel lblProductId = new JLabel("Product ID:");

lblProductId.setFont(new Font("Century Gothic", Font.BOLD, 21));

lblProductId.setBounds(203, 70, 119, 41);

frame.getContentPane().add(lblProductId);

//product id TextField

productid = new JTextField();

productid.setFont(new Font("Franklin Gothic Book", Font.PLAIN, 21));

productid.addFocusListener(new FocusAdapter() {

@Override

public void focusLost(FocusEvent e) {

for(int i = 0; i < Menu.Data.size(); i++) { //This loop is for checking if the new product already registered or not

if(!(Menu.Data.get(i).getProductID().equals(productid.getText()))) {

idvalue = productid.getText(); //get product id from JTextField

isIdEntered = true; //indicate product id is entered

}

else {

JOptionPane.showMessageDialog(null, "This product is already added"); //this dialog message will show if product already added

productid.setText("");

}

}

}

});

productid.setColumns(10);

productid.setBounds(334, 70, 426, 41);

frame.getContentPane().add(productid);

//quantity label

JLabel lblQuantity = new JLabel("Quantity:");

lblQuantity.setFont(new Font("Century Gothic", Font.BOLD, 21));

lblQuantity.setBounds(203, 231, 119, 41);

frame.getContentPane().add(lblQuantity);

//quantity textField

quantity = new JTextField();

quantity.setFont(new Font("Franklin Gothic Book", Font.PLAIN, 21));

quantity.addFocusListener(new FocusAdapter() {

@Override

public void focusLost (FocusEvent e) {

if(!(quantity.getText().equals(""))) {

try {

quantityvalue = Integer.parseInt(quantity.getText()); //get quantity value from JTextField

}

catch(NumberFormatException e1) { //catch the NumberFormatException if user enter string value

e1.printStackTrace();

JOptionPane.showMessageDialog(null, "Please enter numeric value"); //and this dialog message will be shown

quantity.setText("");

}

}

}

});

quantity.setColumns(10);

quantity.setBounds(334, 231, 426, 41);

frame.getContentPane().add(quantity);

//price label

JLabel pricelabel = new JLabel("Price:");

pricelabel.setFont(new Font("Century Gothic", Font.BOLD, 21));

pricelabel.setBounds(203, 400, 119, 40);

frame.getContentPane().add(pricelabel);

//price TextField

price = new JTextField();

price.setFont(new Font("Franklin Gothic Book", Font.PLAIN, 21));

price.addFocusListener(new FocusAdapter() {

@Override

public void focusLost(FocusEvent e1) {

if(!(price.getText().equals(""))) {

try {

pricevalue = Double.parseDouble(price.getText()); //get price value from JTextField

}

catch(NumberFormatException e2) { //catch the NumberFormatException if user enter string value

JOptionPane.showMessageDialog(null, "Please enter numeric value"); //and this dialog message will be shown

price.setText("");

}

}

}

});

price.setColumns(10);

price.setBounds(334, 400, 426, 40);

frame.getContentPane().add(price);

//save button

btnSave = new JButton("SAVE"); //this button is to save an object to ArrayList

btnSave.setFont(new Font("Consolas", Font.BOLD, 21));

btnSave.setBackground(new Color(255, 255, 255));

btnSave.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (type==1 && isIdEntered==true && isNameEntered==true) { //check type, product name, product are already entered or not

Stock Electronic = new Electronic(namevalue,idvalue,pricevalue,quantityvalue); //an object is created

Menu.Data.add(Electronic); //and store into the ArrayList

JOptionPane.showMessageDialog(null, "A product has been added"); //this dialog message will be appeared once the product is added

}

else if(type==2 && isIdEntered==true && isNameEntered==true) { //same as above

Stock Fabric = new Fabric(namevalue,idvalue,pricevalue,quantityvalue);

Menu.Data.add(Fabric);

JOptionPane.showMessageDialog(null, "A product has been added");

}

else {

JOptionPane.showMessageDialog(null, "Please fill in the required field"); //if required field is not completed, this dialog message will appear

}

}

});

btnSave.setBounds(146, 516, 180, 60);

frame.getContentPane().add(btnSave);

//back button

btnBack = new JButton("BACK");

btnBack.setFont(new Font("Consolas", Font.BOLD, 21));

btnBack.setBackground(new Color(255, 255, 255));

btnBack.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try{

Menu menu =new Menu(); //call back menu JFrame

frame.dispose();

menu.frame.setVisible(true);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

btnBack.setBounds(657, 516, 180, 60);

frame.getContentPane().add(btnBack);

//type label

lblType = new JLabel("Type:");

lblType.setFont(new Font("Century Gothic", Font.BOLD, 22));

lblType.setBounds(204, 311, 119, 40);

frame.getContentPane().add(lblType);

//Electronic check box

JCheckBox chckbxElectronic = new JCheckBox("Electronic");

chckbxElectronic.setFont(new Font("Verdana", Font.PLAIN, 21));

chckbxElectronic.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(chckbxElectronic.isSelected() == true) { //check if JCheckBox is selected or not

type=1; //indicate product type 1 which is electronic

}

}

});

chckbxElectronic.setBounds(335, 317, 164, 37);

frame.getContentPane().add(chckbxElectronic);

//Fabric check box

JCheckBox chckbxFabric = new JCheckBox("Fabric");

chckbxFabric.setFont(new Font("Verdana", Font.PLAIN, 21));

chckbxFabric.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(chckbxFabric.isSelected() == true) { //check if JCheckBox is selected or not

type=2; //indicate product type 2 which is fabric

}

}

});

chckbxFabric.setBounds(547, 315, 164, 40);

frame.getContentPane().add(chckbxFabric);

//clear button

btnClear = new JButton("Clear");

btnClear.setFont(new Font("Consolas", Font.BOLD, 21));

btnClear.setBackground(new Color(255, 255, 255));

btnClear.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) { //to clear all field

name.setText("");

productid.setText("");

quantity.setText("");

price.setText("");

chckbxElectronic.setSelected(false);

chckbxFabric.setSelected(false);

}

});

btnClear.setBounds(401, 516, 180, 60);

frame.getContentPane().add(btnClear);

}

}

**GUI And Logic : Update**

package Main;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.JButton;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import java.awt.Toolkit;

import javax.swing.JTextField;

import java.awt.Color;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import java.awt.event.FocusAdapter;

import java.awt.event.FocusEvent;

@SuppressWarnings("serial")

public class Update extends JFrame {

private JPanel contentPane;

private JTextField idtext;

private JTextField updateqty;

private JTextField updateprice;

private JButton saveprice;

private String idvalue;

private int index;

private int qtyvalue;

private Double pricevalue;

Boolean isEntered = false;

Boolean isError = false;

private JButton enterbutton;

//this jFrame is to update quantity or price

/\*\*

\* Create the frame.

\*/

public Update() {

//customize content pane

setResizable(false);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setIconImage(Toolkit.getDefaultToolkit().getImage(Menu.class.getResource("/res/warehouse.png")));

setTitle("Stock Management System");

setBounds(450, 200, 1000, 700);

contentPane = new JPanel();

contentPane.setBackground(Color.LIGHT\_GRAY);

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

//back button

JButton button = new JButton("BACK");

button.setFont(new Font("Consolas", Font.BOLD, 21));

button.setBackground(new Color(255, 255, 255));

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) { // call back menu JFrame

try{

Menu menu =new Menu();

dispose();

menu.frame.setVisible(true);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

button.setBounds(402, 521, 178, 70);

contentPane.add(button);

//product id label

JLabel lblEnterProductId = new JLabel("Enter Product ID:");

lblEnterProductId.setFont(new Font("Century Gothic", Font.BOLD, 21));

lblEnterProductId.setBounds(235, 95, 178, 41);

contentPane.add(lblEnterProductId);

//idtext for id

idtext = new JTextField();

idtext.setFont(new Font("Times New Roman", Font.PLAIN, 21));

idtext.addFocusListener(new FocusAdapter() {

@Override

public void focusLost(FocusEvent e) {

idvalue = idtext.getText(); //get product id from JTextField

}

});

idtext.setColumns(10);

idtext.setBounds(457, 99, 274, 41);

contentPane.add(idtext);

//quantity label

JLabel quantitylabel = new JLabel("Quantity:");

quantitylabel.setFont(new Font("Century Gothic", Font.BOLD, 21));

quantitylabel.setBounds(255, 291, 119, 41);

contentPane.add(quantitylabel);

//update quantity textField

updateqty = new JTextField();

updateqty.setFont(new Font("Times New Roman", Font.PLAIN, 21));

updateqty.addFocusListener(new FocusAdapter() {

@Override

public void focusLost(FocusEvent e) {

if(isEntered==true) { //check if product is entered or not

if(!(updateqty.getText().equals(""))) {

try {

qtyvalue = Integer.parseInt(updateqty.getText()); //get updated quantity value from JTextField

}

catch(NumberFormatException e1) { //catch the exception if user enter string value

isError = true; //indicate there is an error

e1.printStackTrace();

JOptionPane.showMessageDialog(null, "Please enter numeric value"); //dialog message appear ask user to enter numeric value

updateqty.setText("");

}

}

}

}

});

updateqty.setColumns(10);

updateqty.setBounds(433, 291, 284, 41);

contentPane.add(updateqty);

//price label

JLabel pricelabel = new JLabel("Price:");

pricelabel.setFont(new Font("Century Gothic", Font.BOLD, 21));

pricelabel.setBounds(255, 364, 119, 41);

contentPane.add(pricelabel);

//update price textField

updateprice = new JTextField();

updateprice.setFont(new Font("Times New Roman", Font.PLAIN, 21));

updateprice.addFocusListener(new FocusAdapter() {

@Override

public void focusLost(FocusEvent e) { //same as updated quantity function

if(isEntered==true ) {

if(!(updateprice.getText().equals(""))) {

try {

pricevalue = Double.parseDouble(updateprice.getText());

}

catch(NumberFormatException e2) {

isError = true;

JOptionPane.showMessageDialog(null, "Please enter numeric value");

updateprice.setText("");

}

}

}

}

});

updateprice.setColumns(10);

updateprice.setBounds(433, 368, 284, 41);

contentPane.add(updateprice);

//save price button

saveprice = new JButton("SAVE");

saveprice.setFont(new Font("Consolas", Font.BOLD, 21));

saveprice.setBackground(new Color(255, 255, 255));

saveprice.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if( isError == false) { //if there is no error this if will execute

if(isEntered==true ) { //check if product id is entered or not

if(Menu.Data.get(index) instanceof Electronic) { // check which subclass the object lies on

((Electronic)Menu.Data.get(index)).setPrice(pricevalue); //cast the object according to subclass to set new price value

}

else if(Menu.Data.get(index) instanceof Fabric) { //same as above

((Fabric)Menu.Data.get(index)).setPrice(pricevalue);

}

JOptionPane.showMessageDialog(null, "Product is updated");

}

else {

JOptionPane.showMessageDialog(null, "Enter the product ID first"); //ask user enter product id first

updateprice.setText("");

}

}

}

});

saveprice.setBounds(729, 371, 109, 38);

contentPane.add(saveprice);

//enter button

enterbutton = new JButton("Enter");

enterbutton.setFont(new Font("Consolas", Font.BOLD, 21));

enterbutton.setBackground(new Color(255, 255, 255));

enterbutton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) { //this function is to select product id

index = 0; //initialize index

for(int i = 0; i<Menu.Data.size(); i++) {

if( Menu.Data.get(i).getProductID().equals(idvalue)==true) { //check if product id entered exist in the database or not

index = i;

isEntered=true; //indicate product id is entered

JOptionPane.showMessageDialog(null, "The product has been selected");

break; //break the loop

}

}

int counter=0;

for(int i = 0; i<Menu.Data.size(); i++) {

if( Menu.Data.get(i).getProductID().equals(idvalue)==false ) { //check if product id entered exist in the database or not

counter++;

}

}

if(Menu.Data.isEmpty()==true || counter==Menu.Data.size()) {

idtext.setText("");

JOptionPane.showMessageDialog(null, "The product does not exist in database"); //this dialog message appeared when product id not exist in the database

}

}

});

enterbutton.setBounds(442, 178, 97, 59);

contentPane.add(enterbutton);

//save quantity button

JButton savequantity = new JButton("SAVE");

savequantity.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) { //same as function in save price button

if(isError == false) {

if(isEntered==true ) {

if(Menu.Data.get(index) instanceof Electronic) {

((Electronic)Menu.Data.get(index)).setQuantity(qtyvalue);

}

else if(Menu.Data.get(index) instanceof Fabric) {

((Fabric)Menu.Data.get(index)).setQuantity(qtyvalue);

}

JOptionPane.showMessageDialog(null, "Product is updated");

}

else {

JOptionPane.showMessageDialog(null, "Enter the product ID first");

updateqty.setText("");

}

}

}

});

savequantity.setFont(new Font("Consolas", Font.BOLD, 21));

savequantity.setBackground(Color.WHITE);

savequantity.setBounds(729, 291, 109, 41);

contentPane.add(savequantity);

}

}

**GUI And Logic : Display**

package Main;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.border.EmptyBorder;

import javax.swing.table.DefaultTableModel;

import java.awt.Color;

import java.awt.Dimension;

import javax.swing.JOptionPane;

import javax.swing.JLabel;

import java.awt.Font;

import java.awt.Toolkit;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import java.awt.event.FocusAdapter;

import java.awt.event.FocusEvent;

import javax.swing.JTable;

@SuppressWarnings("serial")

public class Display extends JFrame {

private JPanel contentPane;

private JTextField idtext;

private String textvalue;

private int index;

private JTable table;

private String[] Header = {"Product ID","Name","Quantity","Price per item","Type"}; //Initialize table header

private int qtyv;

private double pricev;

private String typev;

//this frame is for displaying the products

/\*\*

\* Create the frame.

\*/

public Display() {

//customize contentPane

setResizable(false);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setIconImage(Toolkit.getDefaultToolkit().getImage(Menu.class.getResource("/res/warehouse.png")));

setTitle("Stock Management System");

setBounds(450, 200, 1000, 700);

contentPane = new JPanel();

contentPane.setBackground(Color.LIGHT\_GRAY);

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

contentPane.setBounds(100, 100, 700, 500);

setContentPane(contentPane);

contentPane.setLayout(null);

contentPane.setLayout(null);

//product id label

JLabel lblEnterProductId = new JLabel("Enter Product ID:");

lblEnterProductId.setFont(new Font("Century Gothic", Font.BOLD, 21));

lblEnterProductId.setBounds(274, 42, 180, 34);

contentPane.add(lblEnterProductId);

DefaultTableModel tableModel = new DefaultTableModel(Header, 0){ //create tableModel with header

public boolean isCellEditable(int row, int column)

{

return false;//This causes all cells to be not editable

}

};

// customize product id textField

idtext = new JTextField();

idtext.addFocusListener(new FocusAdapter() {

@Override

public void focusLost(FocusEvent e) {

textvalue = idtext.getText(); //getText from idtext when the idtext lost focus

}

});

idtext.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

textvalue = idtext.getText(); //getText from idtext when user press enter

}

});

idtext.setBounds(466, 42, 244, 34);

contentPane.add(idtext);

//back button

JButton button = new JButton("BACK");

button.setFont(new Font("Consolas", Font.BOLD, 21));

button.setBackground(new Color(255, 255, 255));

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) { //back to Menu frame

try{

Menu menu =new Menu();

dispose();

menu.frame.setVisible(true);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

button.setBounds(413, 561, 156, 48);

contentPane.add(button);

//load button

JButton btnLoad = new JButton("Load");

btnLoad.setFont(new Font("Consolas", Font.BOLD, 18));

btnLoad.setBackground(new Color(255, 255, 255));

btnLoad.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) { //event handling for showing particular product id

index = 0;

for(int i = 0; i<Menu.Data.size(); i++) {

if( Menu.Data.get(i).getProductID().equals(textvalue)==true) { //check every index in ArrayList that contains what the user entered

index = i;

System.out.println(Menu.Data.get(index).toString());

if(Menu.Data.get(index) instanceof Electronic) { //this is for Electronic product

String productid = Menu.Data.get(index).getProductID();

String name = Menu.Data.get(index).getName();

int qty = ((Electronic)Menu.Data.get(index)).getQuantity();

Double price = ((Electronic)Menu.Data.get(index)).getPrice();

String type = ((Electronic)Menu.Data.get(index)).getType();

Object[] show= {productid,name,qty,price,type}; //create the object array of electronic product

tableModel.addRow(show); //and add it into tableModel as row

}

else if(Menu.Data.get(index) instanceof Fabric) { //same as above above but for Fabric product

String productid = Menu.Data.get(index).getProductID();

String name = Menu.Data.get(index).getName();

int qty = ((Fabric)Menu.Data.get(index)).getQuantity();

Double price = ((Fabric)Menu.Data.get(index)).getPrice();

String type = ((Fabric)Menu.Data.get(index)).getType();

Object[] show= {productid,name,qty,price,type};

tableModel.addRow(show);

}

break; //break when a row is added

}

}

int counter=0;

for(int i = 0; i<Menu.Data.size(); i++) {

if( Menu.Data.get(i).getProductID().equals(textvalue)==false ) {

counter++;

}

}

if(Menu.Data.isEmpty()==true || counter==Menu.Data.size()) {

JOptionPane.showMessageDialog(null, "The product does not exist in database"); ///showing dialog message if the product does not exist in database

}

}

});

btnLoad.setBounds(228, 99, 134, 34);

contentPane.add(btnLoad);

//customize table

table = new JTable(tableModel); //Initialize tableModel into table

table.setFont(new Font("Cambria Math", Font.PLAIN, 15)); //defined the font

table.setRowHeight(40); //set row height for the table

//scrollPane for scrolling purpose

JScrollPane scrollPane = new JScrollPane(table); //add the table into JScrollpane: to enable scrolling

table.getTableHeader().setPreferredSize(

new Dimension(scrollPane.getWidth(),40) //Row Height for header

);

table.getTableHeader().setFont(new Font("Times New Roman", Font.BOLD, 15));

scrollPane.setBounds(94, 166, 794, 362);

table.setFillsViewportHeight(true);

getContentPane().add(scrollPane);

//clear table button

JButton btnClear = new JButton("Clear");

btnClear.setFont(new Font("Consolas", Font.BOLD, 18));

btnClear.setBackground(new Color(255, 255, 255));

btnClear.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

tableModel.setRowCount(0); //clear all row

}

});

btnClear.setBounds(424, 99, 134, 34);

contentPane.add(btnClear);

//show all button

JButton btnShowAll = new JButton("Show All");

btnShowAll.setFont(new Font("Consolas", Font.BOLD, 18));

btnShowAll.setBackground(new Color(255, 255, 255));

btnShowAll.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) { //event handling for showing all data in ArrayList

for (int i = 0; i < Menu.Data.size(); i++){ //this loop for adding every elements in ArrayList into the table

String productid = Menu.Data.get(i).getProductID();

String name = Menu.Data.get(i).getName();

if(Menu.Data.get(i) instanceof Electronic) {

qtyv = ((Electronic)Menu.Data.get(i)).getQuantity();

pricev = ((Electronic)Menu.Data.get(i)).getPrice();

typev = ((Electronic)Menu.Data.get(i)).getType();

}

else if(Menu.Data.get(i) instanceof Fabric) {

qtyv = ((Fabric)Menu.Data.get(i)).getQuantity();

pricev = ((Fabric)Menu.Data.get(i)).getPrice();

typev = ((Fabric)Menu.Data.get(i)).getType();

}

Object[] show= {productid,name,qtyv,pricev,typev}; //create the object array containing the attributes of the element

tableModel.addRow(show); //add the object array into tableModel to form a row

}

}

});

btnShowAll.setBounds(615, 99, 135, 34);

contentPane.add(btnShowAll);

}

}

**GUI And Logic : About**

package Main;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import java.awt.Color;

import javax.swing.JLabel;

import java.awt.Font;

import java.awt.Toolkit;

@SuppressWarnings("serial")

public class About extends JFrame {

//This Class only shows our developers list names.

private JPanel contentPane;

public About() {

setResizable(false);

setIconImage(Toolkit.getDefaultToolkit().getImage(Menu.class.getResource("/res/warehouse.png")));

setTitle("Stock Management System: About");

setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

setBounds(700, 400, 450, 300);

contentPane = new JPanel();

contentPane.setBackground(new Color(0, 0, 0));

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

JLabel lblDevelopers = new JLabel("Developers: ");

lblDevelopers.setForeground(new Color(255, 255, 255));

lblDevelopers.setFont(new Font("Century Gothic", Font.BOLD, 15));

lblDevelopers.setBounds(170, 23, 92, 30);

contentPane.add(lblDevelopers);

JLabel lblMuhammadLaziemBin = new JLabel("Muhammad Laziem Bin Shafie (1621781)");

lblMuhammadLaziemBin.setForeground(new Color(255, 255, 255));

lblMuhammadLaziemBin.setFont(new Font("Century Gothic", Font.BOLD, 15));

lblMuhammadLaziemBin.setBounds(69, 60, 293, 30);

contentPane.add(lblMuhammadLaziemBin);

JLabel lblMuhammadSyafiqBin = new JLabel("Muhammad Syafiq Bin Mohd. Faudzy (1623765)");

lblMuhammadSyafiqBin.setForeground(new Color(255, 255, 255));

lblMuhammadSyafiqBin.setFont(new Font("Century Gothic", Font.BOLD, 15));

lblMuhammadSyafiqBin.setBounds(44, 87, 344, 30);

contentPane.add(lblMuhammadSyafiqBin);

JLabel lblNikAhmadFaiz = new JLabel("Nik Ahmad Faiz Bin Mohd Fauzi (1623399)");

lblNikAhmadFaiz.setForeground(new Color(255, 255, 255));

lblNikAhmadFaiz.setFont(new Font("Century Gothic", Font.BOLD, 15));

lblNikAhmadFaiz.setBounds(63, 116, 306, 30);

contentPane.add(lblNikAhmadFaiz);

JLabel lblMuhammadYusufBin = new JLabel("Muhammad Yusuf Bin Zainul Ariffin (1620815)");

lblMuhammadYusufBin.setForeground(new Color(255, 255, 255));

lblMuhammadYusufBin.setFont(new Font("Century Gothic", Font.BOLD, 15));

lblMuhammadYusufBin.setBounds(55, 145, 322, 30);

contentPane.add(lblMuhammadYusufBin);

}

}

**GUI And Logic : Delete**

package Main;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.JButton;

import java.awt.Color;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import java.awt.Font;

import java.awt.Toolkit;

import javax.swing.JTextField;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

import java.awt.event.FocusAdapter;

import java.awt.event.FocusEvent;

@SuppressWarnings("serial")

public class Delete extends JFrame {

private JPanel contentPane;

private JTextField idtext;

private String idvalue;

private int index ;

//This frame is for deletion of product

/\*\*

\* Create the frame.

\*/

public Delete() {

//customize contentPane

setResizable(false);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setIconImage(Toolkit.getDefaultToolkit().getImage(Menu.class.getResource("/res/warehouse.png")));

setTitle("Stock Management System");

setBounds(450, 200, 1000, 700);

contentPane = new JPanel();

contentPane.setBackground(Color.LIGHT\_GRAY);

contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

setContentPane(contentPane);

contentPane.setLayout(null);

//product id label

JLabel productidlabel = new JLabel("Enter Product ID:");

productidlabel.setFont(new Font("Century Gothic", Font.BOLD, 21));

productidlabel.setBounds(124, 280, 172, 41);

contentPane.add(productidlabel);

//customize id TextField

idtext = new JTextField();

idtext.setFont(new Font("Courier New", Font.PLAIN, 21));

idtext.addFocusListener(new FocusAdapter() {

@Override

public void focusLost(FocusEvent e) {

idvalue = idtext.getText();

}

});

idtext.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

idvalue = idtext.getText();

}

});

idtext.setBounds(313, 280, 283, 41);

contentPane.add(idtext);

idtext.setColumns(10);

//button back

JButton button = new JButton("BACK");

button.setBackground(new Color(255, 255, 255));

button.setFont(new Font("Consolas", Font.BOLD, 21));

button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try{

Menu menu =new Menu();

dispose();

menu.frame.setVisible(true);

} catch (Exception e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

button.setBounds(367, 523, 228, 54);

contentPane.add(button);

//delete product label

JLabel lblDeleteProduct = new JLabel("Delete Product");

lblDeleteProduct.setFont(new Font("Brandon Grotesque Black", Font.PLAIN, 46));

lblDeleteProduct.setBounds(340, 94, 302, 114);

contentPane.add(lblDeleteProduct);

//button delete

JButton btnDelete = new JButton("Delete");

btnDelete.setBackground(new Color(255, 255, 255));

btnDelete.setFont(new Font("Consolas", Font.BOLD, 21));

btnDelete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

index = 0;

int counter=0;

Boolean isDeleted = false;

for(int i = 0; i<Menu.Data.size(); i++) {

if( Menu.Data.get(i).getProductID().equals(idvalue)==true) { //check if product id entered exist in database or not

index = i;

Menu.Data.remove(index); //delete element on ArrayList which the product id lies on

JOptionPane.showMessageDialog(null, "The product has been deleted");

isDeleted = true; //indicate the deleted product

break;

}

else if(Menu.Data.get(i).getProductID().equals(idvalue)==false ) { //counter++ if if product id doesn't exist in particular element

counter++;

}

if(counter==Menu.Data.size()) {

idtext.setText("");

JOptionPane.showMessageDialog(null, "The product does not exist in database"); //show this dialog message if the product id doesn't exist

break;

}

}

if((Menu.Data.isEmpty()==true && isDeleted==false)) { //check if the ArrayList is empty

idtext.setText("");

JOptionPane.showMessageDialog(null, "The product does not exist in database"); //this dialog message will appear if ArrayList is empty

}

}

});

btnDelete.setBounds(607, 280, 178, 41);

contentPane.add(btnDelete);

}

}