

# 6308 - JAVA PROGRAMMING

## JAVA MINI PROJECT

### Title : 9 to 9 online supermarket

#### TEAMMATES:

- ✧ AAKASH K (2018103502)
- ✧ ADITHYA M (2018103505)

## PROJECT DESCRIPTION

---

This 9 to 9 online supermarket is a system developed using java, which is associated with a store ( in our case , A-square Shopping ) where the admin is the owner of the store , and the users are customers who want to purchase items from the store. This is a product delivery system , where a user can view the products available in the store and place orders by giving their delivery address, the mode of payment is only COD. The admin can add products into the system, can delete / update the stocks of any products from the system, and can check the orders placed by the customers . The user can check for products available in the store from their homes through the system and can add the desired products into cart and place orders.

In our project , we have categorised the products as electronic products, mobiles and kids sections. 9 to 9 means the delivery of products occurs between 9am and 9pm .

## CONCEPTS OF JAVA INCORPORATED

---

- ❖ Classes and Objects
- ❖ Arrays and other data structures
- ❖ Java Swings and awt
- ❖ Core Java
- ❖ JDBC
- ❖ Exception Handling

# 1. MODULES

---

1. Front-end :

a. JAVA Swing/AWT


2. Back-end :

a. SQLite – DB , JDBC.

# 2. FORMS

---

1. Panel for ADMIN

 Administrator Panel

New EntryCheck StockBilling

Product Details

Brand:

Model:


Price:

Quantity:


Category:

Electronics

Product Description:



Choose photo:

 Update

## 2. Product information while adding a new product

Product Information

Brand:

Model:

Price:

In Stock:

Features:

Qty: 1

Add to cart

## 3. Account status

Account Information

You're signed in as:

Sign Out

Go Back

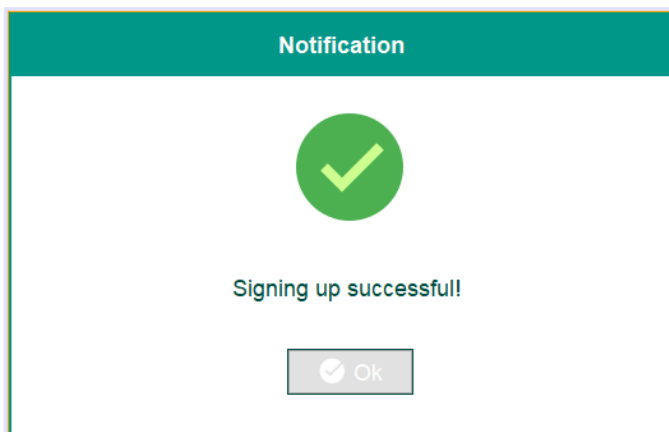
## 4. Notification after purchasing the products

Notification

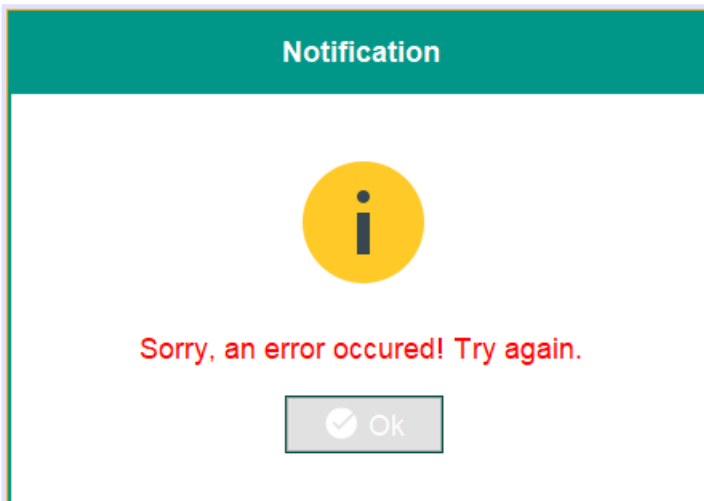
Your order is being shipped.  
Thanks for shopping with us!

Ok

5. Notification after signing up



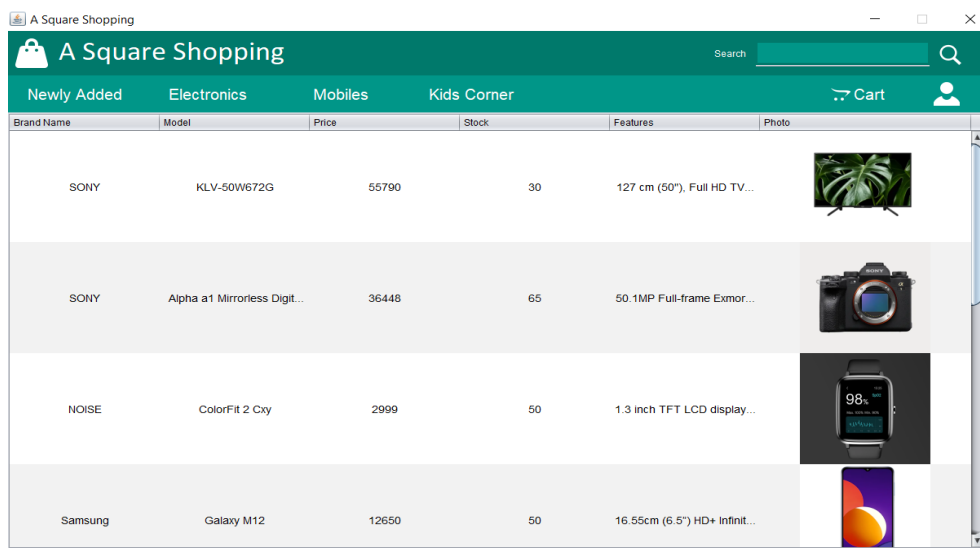
6. Notification if any error occurred







## USER POINT OF VIEW

### 1. HOME PAGE :

- Newly Added products page will be our Home page
- It contains the top Three items of recently added electronic products ,Mobiles and items of kids corner by Admin .
- The user can see products and the stocks available and can also add those desired items into the cart, but cannot buy if he is not signed in.

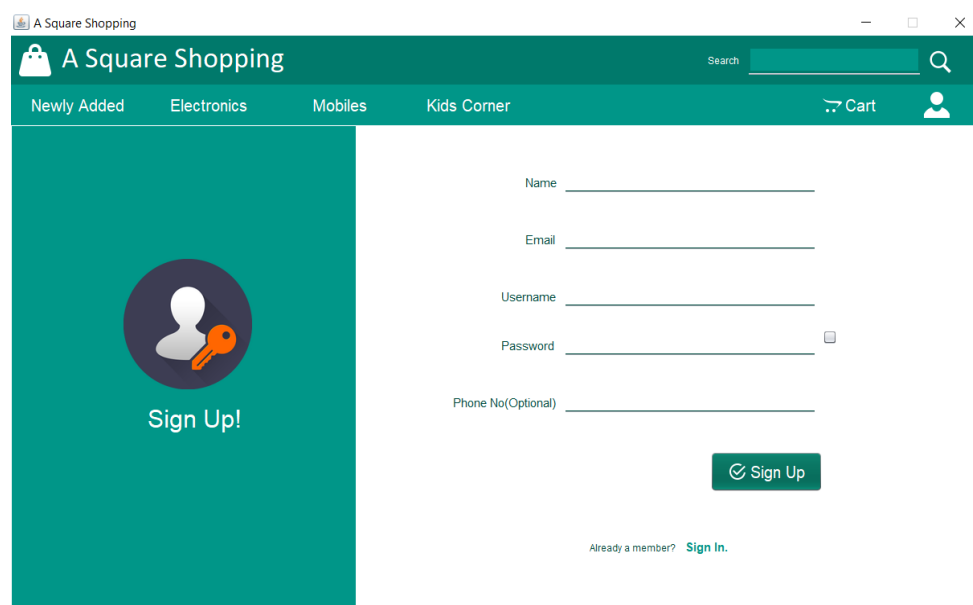


The screenshot shows the 'A Square Shopping' website's home page. The header is teal with a search bar and navigation links: 'Newly Added', 'Electronics', 'Mobiles', and 'Kids Corner'. A 'Cart' icon and a user profile icon are on the right. Below the header is a table of products:

Brand Name	Model	Price	Stock	Features	Photo
SONY	KLV-50W672G	55790	30	127 cm (50"), Full HD TV...	
SONY	Alpha a1 Mirrorless Digit...	36448	65	50.1MP Full-frame Exmor...	
NOISE	ColorFit 2 Cxy	2999	50	1.3 inch TFT LCD display...	
Samsung	Galaxy M12	12650	50	16.55cm (6.5") HD+ Infnit...	

### 2. Sign-up page :

- The user can sign up if he/she is not registered.
- The user must sign up and after signing , it automatically redirects to the sign-in/login page.



The screenshot shows the 'A Square Shopping' website's sign-up page. The header is teal with a search bar and navigation links: 'Newly Added', 'Electronics', 'Mobiles', and 'Kids Corner'. A 'Cart' icon and a user profile icon are on the right. The main content area is split into two sections. On the left is a teal box with a white user icon and a key icon, with the text 'Sign Up!'. On the right is a white box with a sign-up form:

Name

Email

Username

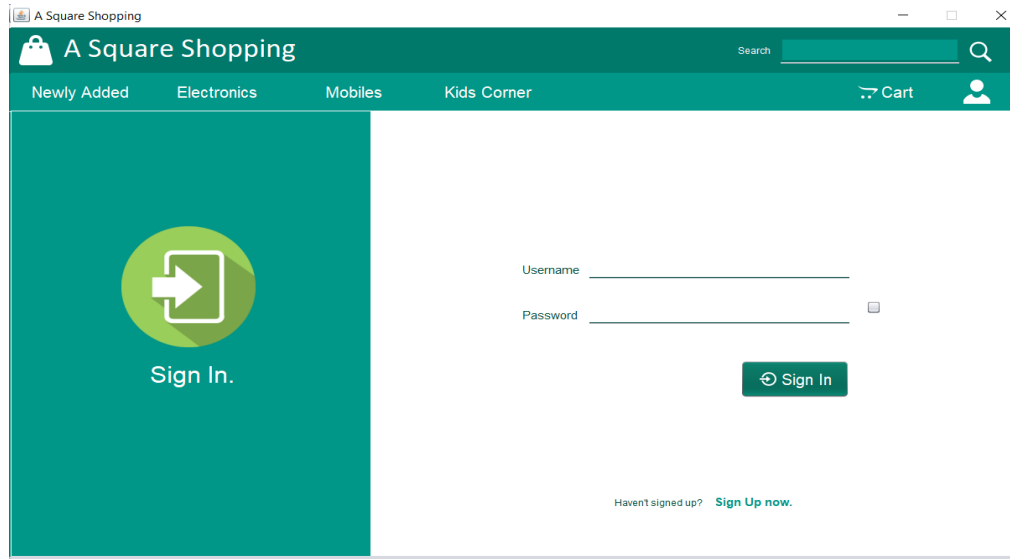
Password

Phone No(Optional)

Already a member? [Sign In.](#)

### 3. Login/Sign-in page :

- The user can login using the username and password which he used to signup.
- The user must sign in in order to purchase products.



### 4. Electronic Products page :

- This page contains a list of Mobile products added by the admin.
- The brand-name , model number, price , stock available , features of the product and its image are the descriptions available for any product that a user can see before adding it to cart.




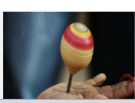
The screenshot shows the 'Electronic Products' page of 'A Square Shopping'. The header is green with the store name, a search bar, and navigation links: 'Newly Added', 'Electronics', 'Mobiles', and 'Kids Corner'. On the right of the header are links for 'Cart' and a user profile icon. Below the header is a table listing mobile products. The table has columns for 'Brand Name', 'Model', 'Price', 'Stock', 'Features', and 'Photo'. The products listed are:

Brand Name	Model	Price	Stock	Features	Photo
boAt	Rockez 335	1600	75	30H Nonstop Audio Bliss...	
boAt	Airdopes 441	2500	19	IPX7 Water & Sweat Resi...	
NOISE	ColorFit 2 Cxy	2999	50	1.3 inch TFT LCD display...	
SONY	Alpha a1 Mirrorless Digit...	36448	65	50.1MP Full-frame Exmor...	

## 5. Kids Corner page





- This page contains a list of kids products added by the admin.
- The brand-name , model number, price , stock available , features of the product and its image are the descriptions available for any product that a user can see before adding it to cart.
- This page consists of the list of all electronics products added by the Admin.
- The brand-name , model number, price , stock available , features of the product and its image are the descriptions available for any product that a user can see before adding it to cart.

A Square Shopping

A Square Shopping					
Search					
Newly Added Electronics Mobiles Kids Corner Cart					
Brand Name	Model	Price	Stock	Features	Photo
Yo Yo	NewYoYo	1000	34	-Captain America feature...	
Bag of marbles	GlassMarbles	200	22	-Marbles	
Gulli	o1	100	40	Durable & flexible	
Spinner	S1	50	44	Rope1 Spinner	

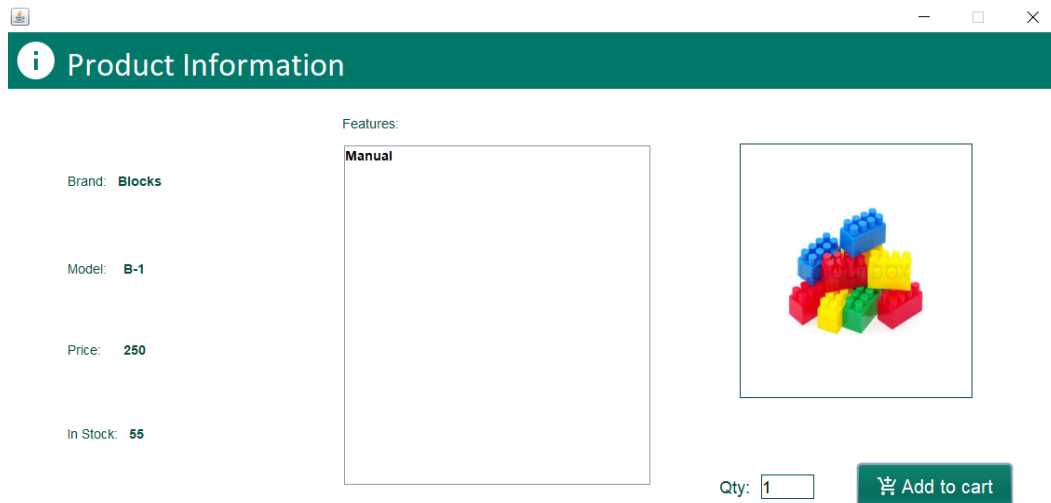
## 6. Mobiles pages :

A Square Shopping

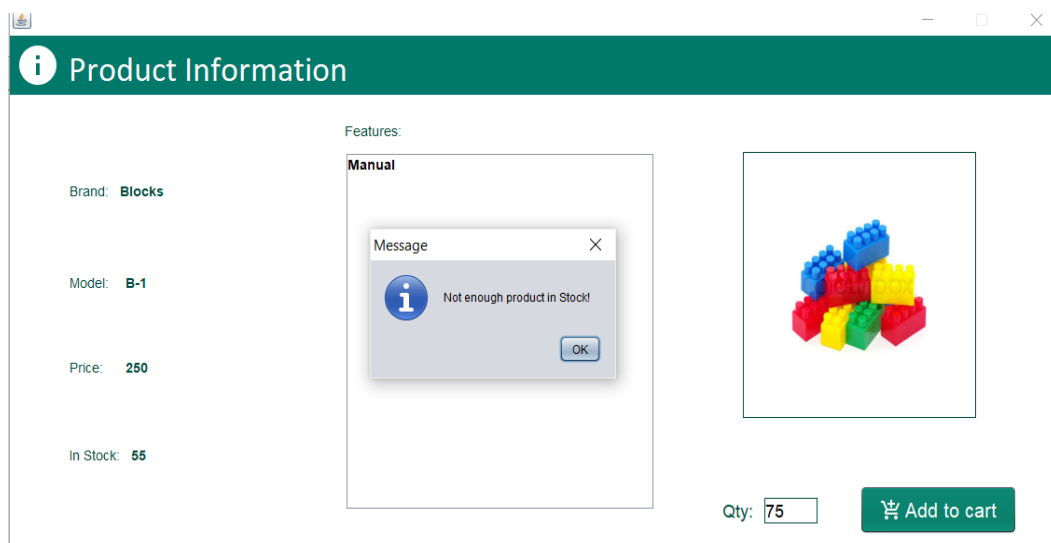
A Square Shopping					
Search					
Newly Added Electronics Mobiles Kids Corner Cart					
Brand Name	Model	Price	Stock	Features	Photo
Samsung	S20 FE	49799	25	Samsung Galaxy's most a...	
Mi	10i 5G	19999	40	Mi 10i 5G - Midnight Blac...	
One Plus	8T	41000	20	OxygenOS based on And...	
One Plus	Nord 5G	26999	25	Android v10 (Q)Qualcom...	

## 7. Product information and quantity to purchase

- For every product in the Electronics/Mobile/Kids Corner Section of the project. The following image is displayed with all its features like brand-name , model number, price , stocks available , features of the product and its image. The user can see the product's details before adding it to cart.
- The user can update the quantity that he wants and the Add to Cart button , adds them to the cart.



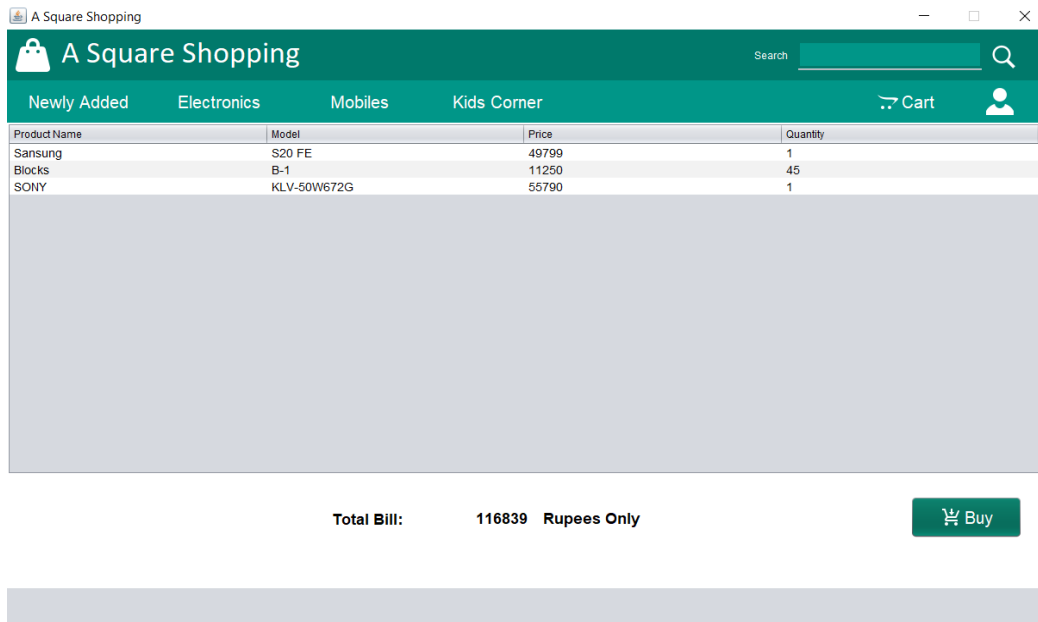
- If a user wants a product above the stock available, it generates the following notification, and the items are not added into the cart.





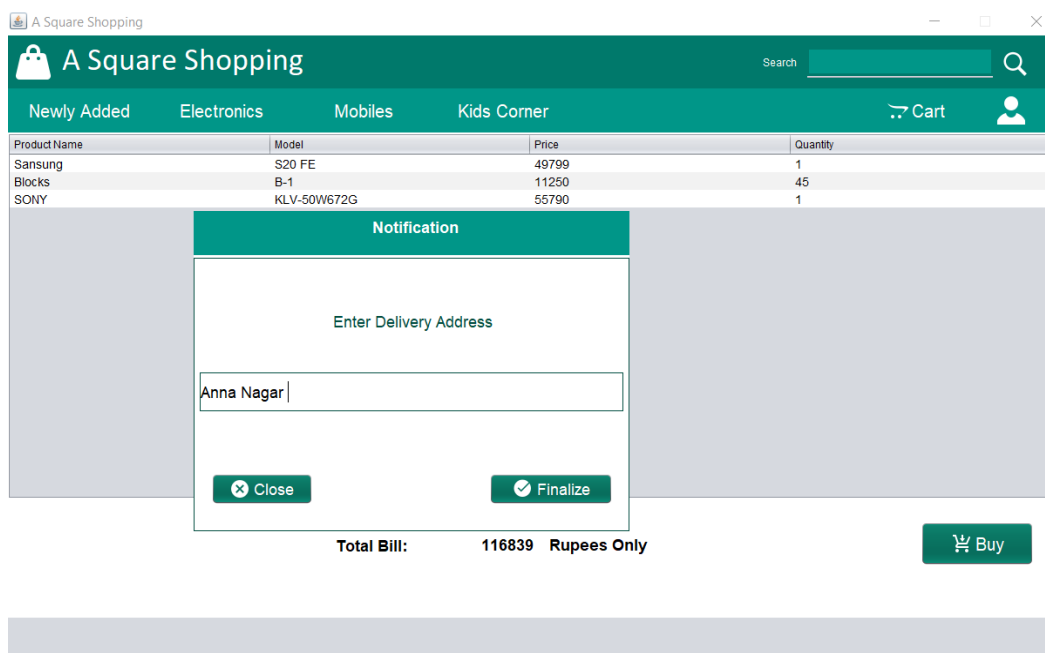
## 8. CART page :

- This page contains a list of products which the user has added to cart.
- Product name , model , price (calculated , with price of 1 \* quantity) and quantity of each product added to cart is shown.
- After adding the desired products , the user can see the total bill , and proceed further by selecting the BUY button.



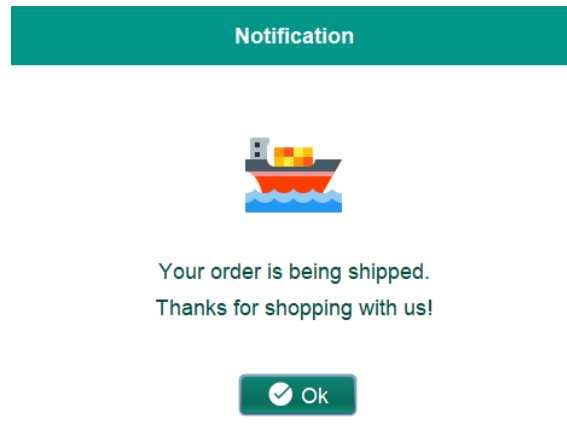
## 9. Delivery address page :

- After selecting the desired product , users can proceed further by entering the DELIVERY ADDRESS and finalizing the selected products.



10. Notification for shipping after finalize the delivery address and placing the order :

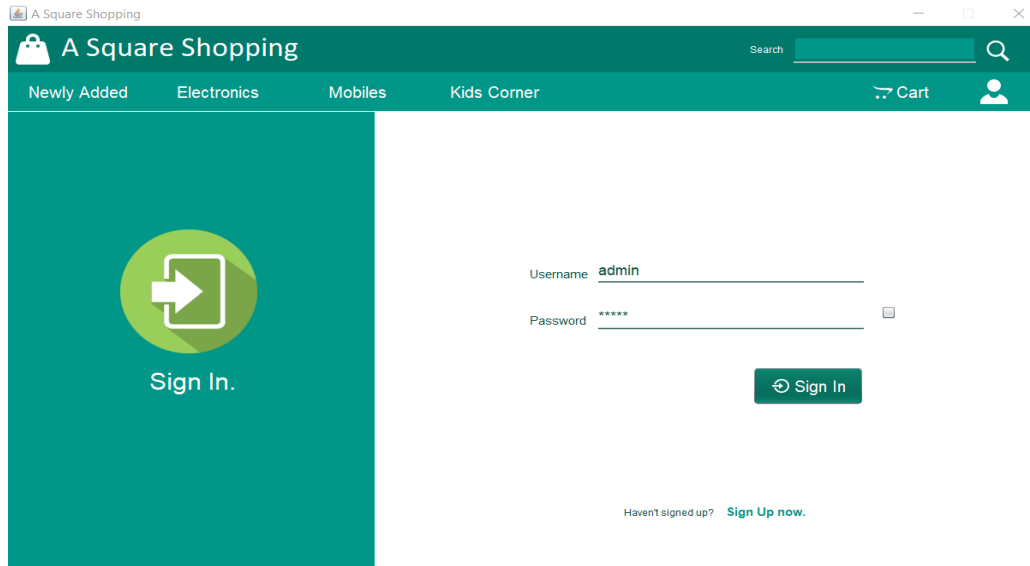
- If the order is placed, then the stocks of all those products are updated , i.e decreased from the available stocks .



## ADMIN POINT OF VIEW

### 1. Login panel for ADMIN :

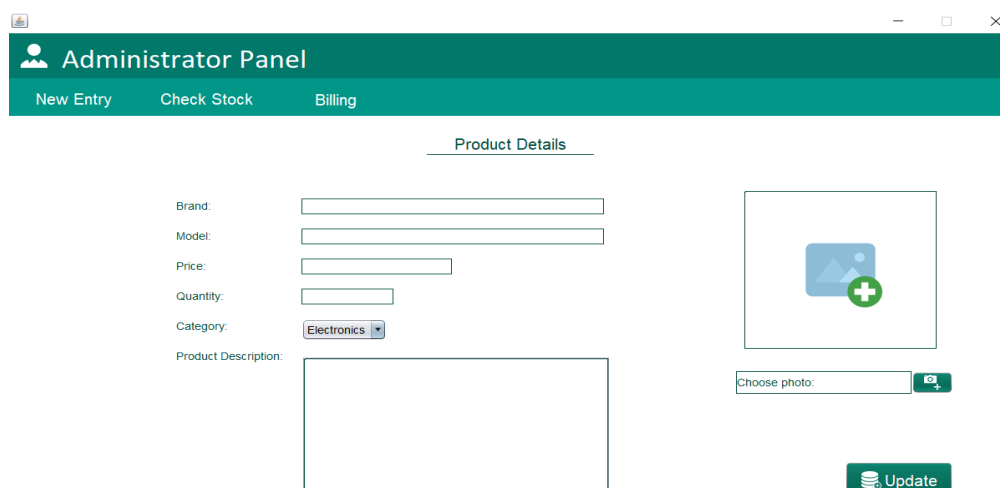
- Username and Password for admin is default set as “admin” and “admin”



The screenshot shows the 'A Square Shopping' website interface. The header includes a search bar and navigation links for 'Newly Added', 'Electronics', 'Mobiles', and 'Kids Corner'. A shopping cart icon and a user profile icon are also present. The main content area is split into two sections. On the left, there is a large green box with a white right-pointing arrow icon and the text 'Sign In.'. On the right, there is a login form with fields for 'Username' (containing 'admin') and 'Password' (containing '\*\*\*\*\*'). Below the password field is a 'Sign In' button. At the bottom of the form, there is a link that says 'Haven't signed up? Sign Up now.'

### 2. New Entry page :

- The Admin can add new products into the Shopping cart.
- The category drop list is present where the admin can choose which category (electronics, mobile , kids ) of the product the admin wants to add .
- The brand, model number , quantity , price , product description and the product's image must be added by the admin .
- The update button will update the database and the newly added items can be seen by the user.



The screenshot shows the 'Administrator Panel' of the website. The header has a user profile icon and the title 'Administrator Panel'. Below the header are three tabs: 'New Entry', 'Check Stock', and 'Billing'. The 'New Entry' tab is active. The main content area is titled 'Product Details' and contains a form for adding a new product. The form includes fields for 'Brand:', 'Model:', 'Price:', 'Quantity:', 'Category:' (with a dropdown menu showing 'Electronics'), and 'Product Description:'. To the right of the form is a large square area for the product image, with a 'Choose photo:' button and a camera icon. At the bottom right of the form is an 'Update' button.

### 3. Check Stocks for Admin :

- The admin can add stock , delete entry(product) and view all details of the present products available in the shop. Add stocks can update the number of stocks available for a product, the model number will act as the primary key.

Administrator Panel

New EntryCheck StockBilling

Brand Name	Model	Stock	Category
boAt	Rockerz 335	75	Electronics
boAt	Airdopes 441	19	Electronics
NOISE	ColorFit 2 City	50	Electronics
SONY	Alpha a1 Mirrorless Digital Camera	65	Electronics
SONY	KLV-50W672G	29	Electronics
Samsung	S20 FE	24	Mobiles
MI	10i 5G	40	Mobiles
One Plus	8T	20	Mobiles
One Plus	Nord 5G	25	Mobiles
Realme	X 128GB	14	Mobiles
Samsung	Galaxy M12	50	Mobiles
Yo Yo	NewYoYo	34	Kids
Bag of marbles	GlassMarbles	22	Kids
Gutti	o1	40	Kids
Spinner	S1	44	Kids
Gameboy	G5	47	Kids
Water Gun	WG-1	35	Kids
Blocks	B-1	55	Kids
Balloons	B-2	39	Kids

Add to Stoc... Enter model:

Electronics

Set Quantity:

↑ Update Stock

Delete Entry: Enter model:

Electronics

✕ Delete Entry

### 4. Billing panel for Admin :

- Admin can view and delete all previous billings done by the users with the time and date of purchase.

Administrator Panel

New EntryCheck StockBilling

Serial	Username	Total Bill	Time & Date
1	anish1	25	Wed May 19 15:22:04 IST 2021
2	bruce	202500	Wed May 19 15:36:51 IST 2021
3	dj	116839	Sun May 23 11:27:32 IST 2021

✕ Delete Log

### 3. TABLES in DATABASE

---

#### 1. Userinfo ( login details )

	username	name	email	password	phone
	Filter	Filter	Filter	Filter	Filter
1	asd	sad	sad@gmail.com	dsa	
2	jsdg	jha	jhag@gmail.com	ashgf	7418529635
3	tony	Tony Stark	iamironman@gmail.com	tony	
4	steve	Steve Smith	ss@gmail.com	steve	
5	ak	Aakash	akwayne@gmail.com	ak	
6	anish	anish	anishyemma@gmail.com	anish	9685741232
7	ak4	ak	ak1@gmail.com	ak	
8	adicj	adi	adicj@gmail	adicj	9856985471
9	dj	dj	dj@gmail.com	dj	
10	dj1	dj1	dj1@gmail.com	dj1	
11	dj2	dj2	dj2@gmail.com	dj2	6352417496

#### 2. BillingDB ( details of orders placed )

	id	uname	bill	date
	Filter	Filter	Filter	Filter
1	1	anish1	25	Wed May 19 15:22:04 IST 2021
2	2	bruce	202500	Wed May 19 15:36:51 IST 2021
3	19	dj	116839	Sun May 23 11:27:32 IST 2021

#### 3. electronicsDB ( list of electronic products available )

	id	mbrand	mmodel	mprice	mquantity	mdescription	mphoto
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	boAt	Rockerz 335	1600	75	30H Nonstop Audio Bliss...	boat-wire.png
2	2	boAt	Airdopes 441	2500	19	IPX7 Water & Sweat Resistance...	boat-wireless.png
3	3	NOISE	ColorFit 2 Cxy	2999	50	1.3 inch TFT LCD display...	smart-watch.png
4	4	SONY	Alpha a1 Mirrorless Digital Camera	36448	65	50.1MP Full-frame ...	sony-cam.png
5	5	SONY	KLV-50W672G	55790	29	127 cm (50"), Full HD TV...	sony-tv.png

#### 4. kidsDB

	id	mbrand	mmodel	mprice	mquantity	mdescription	mphoto
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Yo Yo	NewYoYo	1000	34	-Captain America featured Yo...	yoyo.jpg
2	2	Bag of marbles	GlassMarbles	200	22	-Marbles	marbles-bag.jpg
3	3	Gulti	o1	100	40	Durable & flexible...	Gulti.jpg
4	4	Spinner	S1	50	44	Rope...	Spinner.jpg
5	5	Gameboy	G5	300	47	3 game...	gameboy.jpg
6	6	Water Gun	WG-1	150	35	Fires water at a long distant....	water gun.jpg
7	7	Blocks	B-1	250	55	Manual...	Blocks.jpg
8	8	Balloons	B-2	5	39	10 piece packet...	Balloons.jpg

#### 5. mobileDB

	id	mbrand	mmodel	mprice	mquantity	mdescription	mphoto
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	Sansung	S20 FE	49799	24	Samsung Galaxy's most advanced ...	S-s20-fe.png
2	2	MI	10I 5G	19999	40	Mi 10i 5G - Midnight Black...	mi-10i.png
3	3	One Plus	8T	41000	20	OxygenOS based on Android 11...	1plus8T.png
4	4	One Plus	Nord 5G	26999	25	Android v10 (Q)...	1plus-nord.png
5	5	RealMe	X 128GB	200000	14	16.59 cm (6.53 inch) FHD+ Super ...	realme-x.png
6	6	Samsung	Galaxy M12	12650	50	16.55cm (6.5") HD+ Infinity-V Displa...	S-M12.png

## 4. CLASSES

---

### 1. BillObject

```
package project;

public class BillObject {
    private int id;
    private String uname;
    private int bill;
    private String date;

    public BillObject(int id, String uname, int bill, String date) {
        this.id = id;
        this.uname = uname;
        this.bill = bill;
        this.date = date;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getUname() {
        return uname;
    }

    public void setUname(String uname) {
        this.uname = uname;
    }

    public int getBill() {
        return bill;
    }

    public void setBill(int bill) {
        this.bill = bill;
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }
}
```

## 2. ProductList

package project;

import java.awt.Image;

import javax.swing.ImageIcon;

```
public class ProductList {
    private String brand;
    private String model;
    private int price;
    private int qty;
    private String description;
    private String mimage;

    public ProductList(String brand, String model, int price, int qty, String description, String image) {
        this.brand = brand;
        this.model = model;
        this.price = price;
        this.qty = qty;
        this.description = description;
        this.mimage = image;
    }

    public String getBrand() {
        return brand;
    }

    public void setBrand(String brand) {
        this.brand = brand;
    }

    public String getModel() {
        return model;
    }

    public void setModel(String model) {
        this.model = model;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }

    public int getQty() {
        return qty;
    }

    public void setQty(int qty) {
        this.qty = qty;
    }
}
```



```

public String getDescription() {
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

public String getMimage() {

    return mimage;
}

public void setMimage(String mimage) {
    this.mimage = mimage;
}
}

```

### 3. Search.java

```

public class Search {
    static int electronics = 0;
    static int mobile = 0;
    static int kids = 0;
    public static ArrayList<ProductList> mobileSearch(String model){
        ArrayList<ProductList> list = new ArrayList<>();
        try {
            Connection con = DriverManager.getConnection("jdbc:sqlite:DBs/electronicsDB.db");
            PreparedStatement ps = con.prepareStatement("SELECT * FROM electronics WHERE mbrand=? COLLATE NOCASE OR mmodel=? COLLATE NOCASE");
            ps.setString(1, model);
            ps.setString(2, model);
            ResultSet rs = ps.executeQuery();

            ProductList pl, gl, kl=null;

            while(rs.next()){
                pl = new ProductList(rs.getString("mbrand"),rs.getString("mmodel"),
                    rs.getInt("mprice"),rs.getInt("mquantity"),rs.getString("mdescription"),
                    rs.getString("mphoto"));
                electronics++;

                list.add(pl);
            }
            con.close();

            con = DriverManager.getConnection("jdbc:sqlite:DBs/mobileDB.db");
            ps = con.prepareStatement("SELECT * FROM mobiles WHERE mbrand=? COLLATE NOCASE OR mmodel=? COLLATE NOCASE");
            ps.setString(1, model);
            ps.setString(2, model);
            rs = ps.executeQuery();

```

```

        while(rs.next()){
            gl = new ProductList(rs.getString("mbrand"),rs.getString("mmodel"),
                rs.getInt("mprice"),rs.getInt("mquantity"),rs.getString("mdescription"),
                rs.getString("mphoto"));
            mobile++;
            list.add(gl);

        }
        con.close();

        con = DriverManager.getConnection("jdbc:sqlite:DBs/kidsDB.db");
        ps = con.prepareStatement("SELECT * FROM kids WHERE mbrand=? COLLATE NOCASE OR mmodel=?
COLLATE NOCASE");
        ps.setString(1, model);
        ps.setString(2, model);
        rs = ps.executeQuery();

        while(rs.next()){
            kl = new ProductList(rs.getString("mbrand"),rs.getString("mmodel"),
                rs.getInt("mprice"),rs.getInt("mquantity"),rs.getString("mdescription"),
                rs.getString("mphoto"));
            kids++;
            list.add(kl);

        }
        con.close();

    } catch (SQLException ex) {
        Logger.getLogger(MobileDB.class.getName()).log(Level.SEVERE, null, ex);
    }
    return list;
}
}

```

## 5. DATABASE OPERATIONS

---

### 1. dbconnector.java :

```
public static int insertToUser(String username, String name, String email, String password, String card) {
    try {
        Connection con = DriverManager.getConnection("jdbc:sqlite:DBs/bbs.db");

        String sql = "INSERT INTO userinfo"
            + "(username,name,email,password,phone)"
            + "VALUES(?,?,?,?,?)";

        PreparedStatement ps = con.prepareStatement(sql);

        ps.setString(1, username);
        ps.setString(2, name);
        ps.setString(3, email);
        ps.setString(4, password);
        ps.setString(5, card);
        ps.executeUpdate();

        JFrame notification = new NotificationForm();

    } catch (SQLException e) {
        if(e.getErrorCode()==19) //Duplicate username
            return 19;

        JFrame errorOc = new ErrorOccured();
    }

    return 0;
}
```

- Used for inserting data of new users into the table.
- The data which are provided by the user through SIGN-IN form.

```
public static boolean signer(String username, String pass){
    try {
        Connection con = DriverManager.getConnection("jdbc:sqlite:DBs/bbs.db");
        String check = "SELECT username, password FROM userinfo WHERE username=? AND password=?";
        PreparedStatement ps = con.prepareStatement(check);

        ps.setString(1, username);
        ps.setString(2, pass);
        ResultSet rs = ps.executeQuery();

        return rs.next();

    } catch (SQLException ex) {
        Logger.getLogger(DbConnector.class.getName()).log(Level.SEVERE, null, ex);
    }

    return false;
}
```

- This query is used for LOG-IN.
- It compares the username and password given by the user with the username and password existing in the data-base.

## 2. kidsDB.java , ElectronicsDB.java , mobileDB.java :

Note: The following database operations are common for the products of all categories.

```
public static void insertIntoKidsDB(String brand, String model, int price, int qty, String description, String imagePath){
    try {
        Connection con = DriverManager.getConnection("jdbc:sqlite:DBs/kidsDB.db");

        PreparedStatement ps = con.prepareStatement("INSERT INTO kids(mbrand, mmodel, mprice, "
            + "mquantity, mdescription, mphoto) VALUES(?,?,?,?,?,?)");

        ps.setString(1, brand);
        ps.setString(2, model);
        ps.setInt(3, price);
        ps.setInt(4, qty);
        ps.setString(5, description);
        ps.setString(6, imagePath);

        if(ps.executeUpdate()==1)
            JOptionPane.showMessageDialog(null, "Entry successful!");
    } catch (SQLException ex) {
        Logger.getLogger(MobileDB.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

- The admin can insert new products into the system under the kids , electronics and mobile section using this query by giving those respective values in the product information form.

```
public static void updateKidsDB(String model, int qty){
    try {
        Connection con = DriverManager.getConnection("jdbc:sqlite:DBs/kidsDB.db");

        PreparedStatement ps = con.prepareStatement("UPDATE kids SET mquantity=? WHERE mmodel=?");

        ps.setInt(1, qty);
        ps.setString(2, model);

        if(ps.executeUpdate()==0)
            JOptionPane.showMessageDialog(null, "Entry does not exist!");
        else if(ps.executeUpdate()==1 && flag){
            JOptionPane.showMessageDialog(null, "Stock updated successfully!");
            flag = false;
        }
    } catch (SQLException ex) {
        Logger.getLogger(KidsDB.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

- The Admin can update the stocks of the products if they are low , through the admin panel.

```

public static ArrayList<ProductList> TableGenerator(){
    ArrayList<ProductList> list = new ArrayList<>();
    try {
        Connection con = DriverManager.getConnection("jdbc:sqlite:DBs/kidsDB.db");
        Statement ps = con.createStatement();
        ResultSet rs = ps.executeQuery("SELECT mbrand, mmodel, mprice,mquantity, mdescription, mphoto FROM kids");

        ProductList pl;

        while(rs.next()){
            pl = new ProductList(rs.getString("mbrand"),rs.getString("mmodel"),
                                rs.getInt("mprice"),rs.getInt("mquantity"),rs.getString("mdescription"),
                                rs.getString("mphoto"));

            list.add(pl);
        }

        con.close();
    } catch (SQLException ex) {
        Logger.getLogger(MobileDB.class.getName()).log(Level.SEVERE, null, ex);
    }
    return list;
}

```

- This query is used to display all the product information from the data-base , in the main page of user under each category.

```

public static ArrayList<ProductList> homePageContent(){
    ArrayList<ProductList> list = new ArrayList<>();
    try {
        Connection con = DriverManager.getConnection("jdbc:sqlite:DBs/kidsDB.db");
        Statement ps = con.createStatement();
        ResultSet rs = ps.executeQuery("SELECT mbrand, mmodel, mprice,mquantity, mdescription, mphoto FROM kids ORDER BY id DESC LIMIT 3");

        ProductList pl;

        while(rs.next()){
            pl = new ProductList(rs.getString("mbrand"),rs.getString("mmodel"),
                                rs.getInt("mprice"),rs.getInt("mquantity"),rs.getString("mdescription"),
                                rs.getString("mphoto"));

            list.add(pl);
        }

    } catch (SQLException ex) {
        Logger.getLogger(MobileDB.class.getName()).log(Level.SEVERE, null, ex);
    }
    return list;
}

```

- This Query is used to display the top 3 latest added products under each category for the NEWLY ADDED section in the main page of the user.

```

public static ArrayList<ProductList> checkStock(){
    ArrayList<ProductList> list = new ArrayList<>();
    try {
        Connection con = DriverManager.getConnection("jdbc:sqlite:DBs/kidsDB.db");
        Statement ps = con.createStatement();
        ResultSet rs = ps.executeQuery("SELECT mbrand, mmodel, mprice, mquantity FROM kids");

        ProductList pl;

        while(rs.next()){
            pl = new ProductList(rs.getString("mbrand"),rs.getString("mmodel"),
                                0, rs.getInt("mquantity"),null, null);

            list.add(pl);
        }

    } catch (SQLException ex) {
        Logger.getLogger(MobileDB.class.getName()).log(Level.SEVERE, null, ex);
    }
    return list;
}

```

- The admin can view the stocks available for each products using this query to fetch data from each DB corresponding to the Mobile , Electronics and Kids section.

```

public static void delete(String model){
    try {
        Connection con = DriverManager.getConnection("jdbc:sqlite:DBs/kidsDB.db");
        PreparedStatement ps = con.prepareStatement("DELETE FROM kids WHERE mmodel=?");
        ps.setString(1, model);
        if(ps.executeUpdate()==0)
            JOptionPane.showMessageDialog(null, "Entry does not exist!");
        else
            JOptionPane.showMessageDialog(null, "Entry deleted successfully!");
    } catch (SQLException ex) {
        Logger.getLogger(ElectronicsDB.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

- Admin can delete a product , if there any exist , using this query from the admin panel.

### 3. search.java

```

public static ArrayList<ProductList> mobileSearch(String model){
    ArrayList<ProductList> list = new ArrayList<>();
    try {
        Connection con = DriverManager.getConnection("jdbc:sqlite:DBs/electronicsDB.db");
        PreparedStatement ps = con.prepareStatement("SELECT * FROM electronics WHERE mbrand=? COLLATE NOCASE OR mmodel=? COLLATE NOCASE");
        ps.setString(1, model);
        ps.setString(2, model);
        ResultSet rs = ps.executeQuery();

        ProductList pl, gl, kl=null;

        while(rs.next()){
            pl = new ProductList(rs.getString("mbrand"),rs.getString("mmodel"),
                rs.getInt("mprice"),rs.getInt("mquantity"),rs.getString("mdescription"),
                rs.getString("mphoto"));
            electronics++;
            list.add(pl);
        }
        con.close();

        con = DriverManager.getConnection("jdbc:sqlite:DBs/mobileDB.db");
        ps = con.prepareStatement("SELECT * FROM mobiles WHERE mbrand=? COLLATE NOCASE OR mmodel=? COLLATE NOCASE");
        ps.setString(1, model);
        ps.setString(2, model);
        rs = ps.executeQuery();

        while(rs.next()){
            gl = new ProductList(rs.getString("mbrand"),rs.getString("mmodel"),
                rs.getInt("mprice"),rs.getInt("mquantity"),rs.getString("mdescription"),
                rs.getString("mphoto"));
            mobile++;
            list.add(gl);
        }
        con.close();

        con = DriverManager.getConnection("jdbc:sqlite:DBs/kidsDB.db");
        ps = con.prepareStatement("SELECT * FROM kids WHERE mbrand=? COLLATE NOCASE OR mmodel=? COLLATE NOCASE");
        ps.setString(1, model);
        ps.setString(2, model);
        rs = ps.executeQuery();

        while(rs.next()){
            kl = new ProductList(rs.getString("mbrand"),rs.getString("mmodel"),
                rs.getInt("mprice"),rs.getInt("mquantity"),rs.getString("mdescription"),
                rs.getString("mphoto"));
            kids++;
            list.add(kl);
        }
        con.close();
    } catch (SQLException ex) {
        Logger.getLogger(MobileDB.class.getName()).log(Level.SEVERE, null, ex);
    }
    return list;
}

```

- This query is used to search a product by entering it's Model or Brand name.
- There are 3 queries under this class , which is used to search all 3 databases , which are Mobile DB , Kids DB and Electronics DB.

#### 4. BillingDB.java

```
public static void insertIntoBillingDB(String username, int price, String date){
    try {
        Connection con = DriverManager.getConnection("jdbc:sqlite:DBs/billingDB.db");

        PreparedStatement ps = con.prepareStatement("INSERT INTO billing(uname, bill, date) VALUES(?,?,?)");

        ps.setString(1, username);
        ps.setInt(2, price);
        ps.setString(3, date);
        ps.executeUpdate();

    } catch (SQLException ex) {
        Logger.getLogger(MobileDB.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

- This query is used to insert the bill details into the bill database after purchasing.

```
public static ArrayList<BillObject> billings(){
    ArrayList<BillObject> customers = new ArrayList<>();

    try {
        Connection con = DriverManager.getConnection("jdbc:sqlite:DBs/billingDB.db");
        Statement ps = con.createStatement();
        ResultSet rs = ps.executeQuery("SELECT id, uname, bill, date FROM billing");

        BillObject p1;

        while(rs.next()){
            p1 = new BillObject(rs.getInt("id"),rs.getString("uname"),
                                rs.getInt("bill"),rs.getString("date"));
            customers.add(p1);
        }
    } catch (SQLException ex) {
        Logger.getLogger(MobileDB.class.getName()).log(Level.SEVERE, null, ex);
    }
    return customers;
}
```

- The admin can view all the billing details of the customers by using this query.

```
public static void deleteBillings(){
    try {
        Connection con = DriverManager.getConnection("jdbc:sqlite:DBs/billingDB.db");
        PreparedStatement ps = con.prepareStatement("DELETE FROM billing");
        ps.executeUpdate();
        JOptionPane.showMessageDialog(null, "All entries have been deleted!");

    } catch (SQLException ex) {
        Logger.getLogger(ElectronicsDB.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

- The admin can delete all billing logs if he wants to , by using this query.