Project Report on
Sarthak's Shopping Cart in Java

In partial fulfilment of the award for the degree of Master of Computer Applications

2022-2024

Supervised by:
Dr. Rajinder Singh

Submitted by:
Sarthak Gupta(2224)

Panjab University Swami Sarvanand Giri Regional Centre
Una Road, Bajwara, Hoshiarpur, Punjab 146021

Index

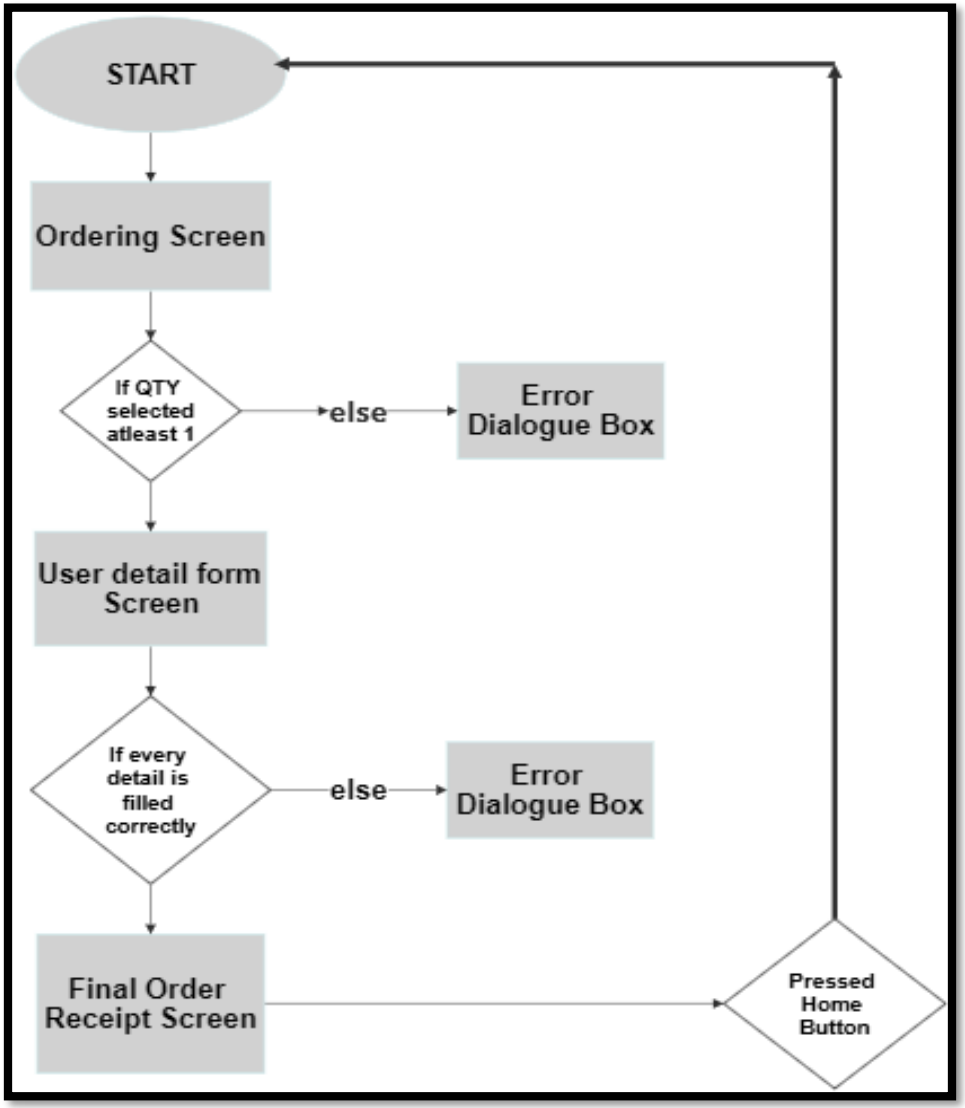
Introduction

*The project entitled **Sarthak's Shopping Cart** is completely based on **java and JDBC** or **java database connectivity** through **MYSQL**. It is like a shopping software application which allows you to order anytime without any error in the transaction.*

*Some of it's **features** are:-*

- ***Creating account with password.***
- ***Changing user details.***
- ***Forget password.***
- ***OTP verification with GSM modem.***
- ***Message to user after every transaction, if committed successfully.***
- ***No technical error.***

Data Flow Diagram (DFD)

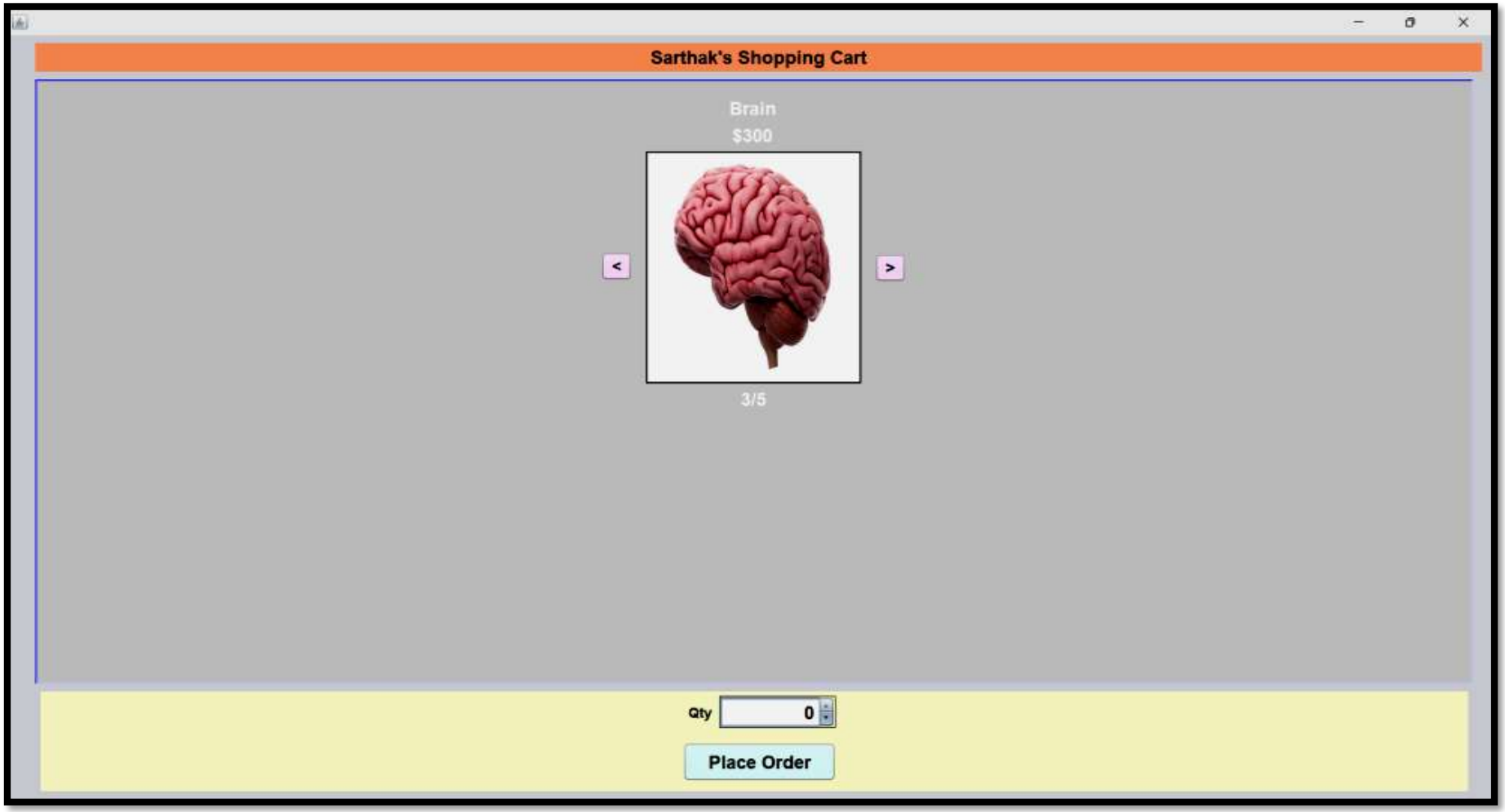


How to run the project

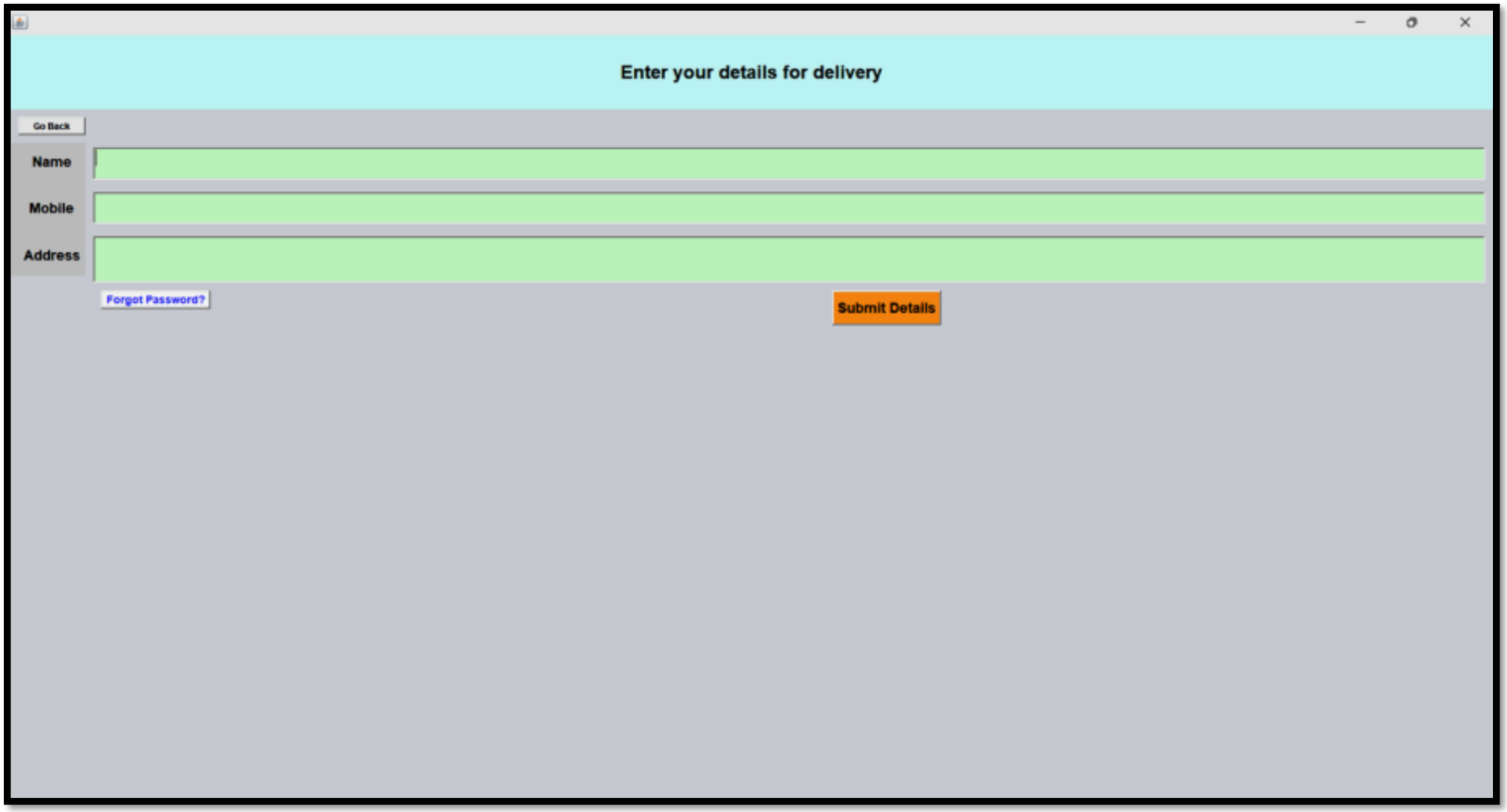
1. *Scan the QR code with Google lens and follow the instructions below:-*

2. *Now download and install the "GSM_modem.apk" and "GSM_helper.apk" in your Smartphone.*
3. *Then, download "Shopping.rar" in your pc and extract the folder named "Shopping".*
4. *Open the project in "Apache Netbeans IDE 17" and create a MYSQL database/schema named "sarthakshoppingcart" in "MYSQL Workbench 8.0". Also, setup MYSQL database connection in Netbeans . I have mentioned some videos for the reference in the bibliography page.*
5. *Now, open GSM_helper in your smart phone and keep it running in the background. Then, open GSM_modem and click on start. Also, make sure your Smartphone device is connected to Wi-Fi (LAN network).*
6. *Now, open the "UserDetailsJForm.java" file in Netbeans and Press "CTRL+F" to find "192.168.1.164:8090" and replace with your IP address shown in GSM_modem application.*
7. *Right click on "ShoppingJFrame.java" and select run file.*
8. *Now, you can test and enjoy my project!*

Ordering Screen

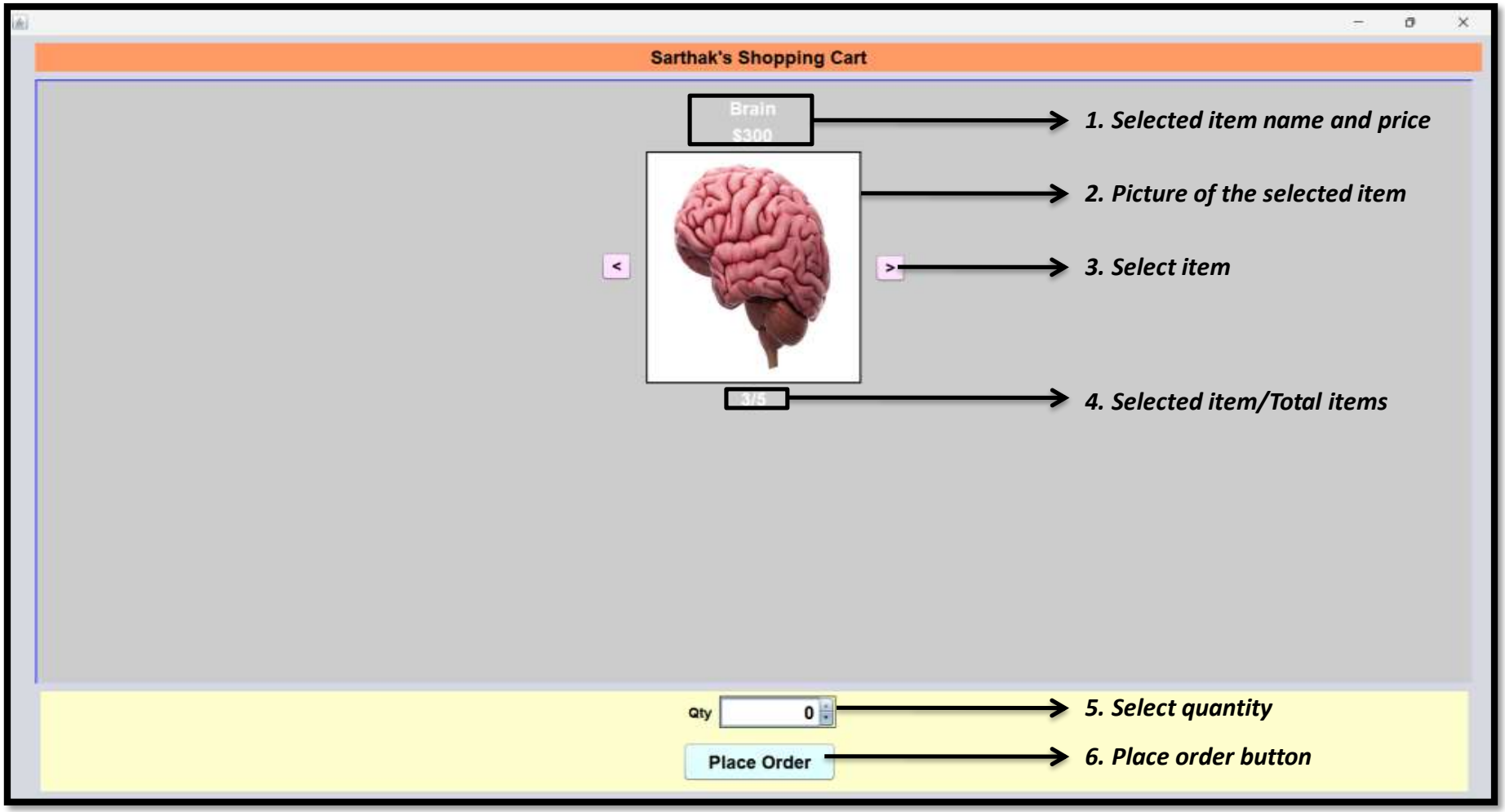


User Detail Form Screen



Final Order Receipt Screen

Ordering Screen Technical Features



Ordering Screen Code snippets

1. Selected item name and price:

```
//File names are picked automatically with this snippet and are displayed as item name in the screen and if item.
      price is set to “itemOutOfStock” then user can’t proceed and place order button is setVisible to false.

// Get image files in ascending/descending order according to the last modified (time).
void GetAllImageFiles() {
      File folder = new File("photos/");
      files = folder.listFiles();
      Arrays.sort(files, (f1, f2) -> Long.compare(f2.lastModified(), f1.lastModified()));
}

//Set image name as item price.
itemNamejLabel.setText(files[imageIndex].toString().substring(7,files[imageIndex].toString().length() - 4));

//if item out of stock then don’t allow user to order else set the item price as provided in the code.
if (itemPrice[imageIndex].equals(itemOutOfStock)) {
         proceedBtn.setVisible(false);
         itemPrice_jLabel.setText(String.valueOf(itemPrice[imageIndex]));
}
else
     itemPrice_jLabel.setText("$" + String.valueOf(itemPrice[imageIndex]));
```

2. Picture of the selected item:

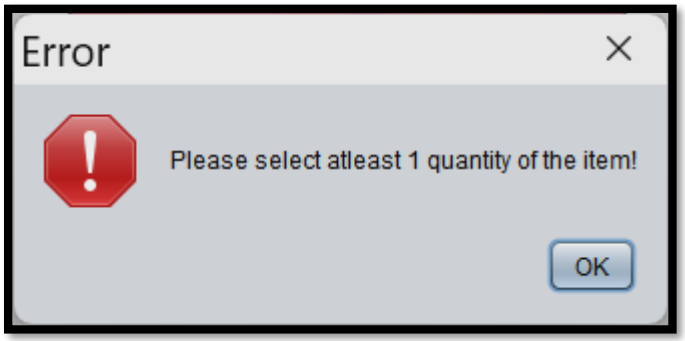
```
//Display proper scaled image in the Jlabel component by passing the image name in the parameter.
 void ImagePick(String imageName) {
   ImageIcon imageIcon = new ImageIcon(imageName);
   Image image = imageIcon.getImage();
   Image scaledImage = image.getScaledInstance(jLabel1.getWidth(), jLabel1.getHeight(),  Image.SCALE_SMOOTH);
   imageIcon = new ImageIcon(scaledImage);
   jLabel1.setIcon(imageIcon);
}
```

3. Select item and 4. Selected item/Total items:

```
//Next item button action to be performed with all technical errors kept in mind Index  of the item is also updated
accordingly.
private void nextImgBtnActionPerformed(java.awt.event.ActionEvent evt) {
                if (imageIndex < files.length - 1) {
                                imageIndex++;
                                proceedBtn.setVisible(true);
                                if (itemPrice[imageIndex].equals(itemOutOfStock)) {
                                proceedBtn.setVisible(false);
                                itemPrice_jLabel.setText(String.valueOf(itemPrice[imageIndex]));

                }
                else
                                itemPrice_jLabel.setText("$" + String.valueOf(itemPrice[imageIndex]));
        ImagePick(files[imageIndex].toString());
        itemCountjLabel.setText((imageIndex + 1) + "/" + files.length);
        newFileName = files[imageIndex].toString().substring(files[imageIndex].toString().length() -
                                files[imageIndex].toString().length(), files[imageIndex].toString().length() - 4);
        itemNamejLabel.setText(newFileName.substring(7, newFileName.length()));
    }
}
```

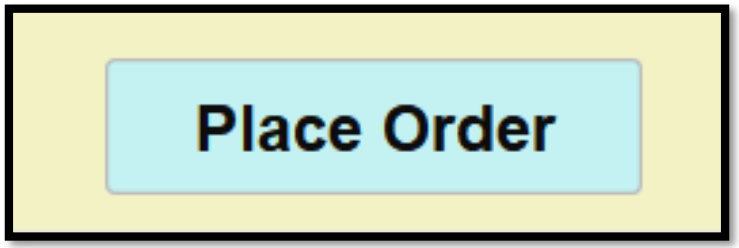
```
//Previous item button action to be performed with all technical errors kept in mind. Index  of the item is also
updated accordingly.
private void nextImgBtnActionPerformed(java.awt.event.ActionEvent evt) {
                if (imageIndex > 0) {
                                imageIndex--;
                                proceedBtn.setVisible(true);
                                if (itemPrice[imageIndex].equals(itemOutOfStock)) {
                                                proceedBtn.setVisible(false);
                                                itemPrice_jLabel.setText(String.valueOf(itemPrice[imageIndex]));
                }
                 else
                                itemPrice_jLabel.setText("$" + String.valueOf(itemPrice[imageIndex]));

                ImagePick(files[imageIndex].toString());
                itemCountjLabel.setText((imageIndex + 1) + "/" + files.length);
                newFileName = files[imageIndex].toString().substring(files[imageIndex].toString().length() -
                                files[imageIndex].toString().length(), files[imageIndex].toString().length() - 4);
                itemNamejLabel.setText(newFileName.substring(7, newFileName.length()));
                }
}
```

5. <u>Select quantity:</u> This works when the "<u>Place Order</u>" button is pressed.

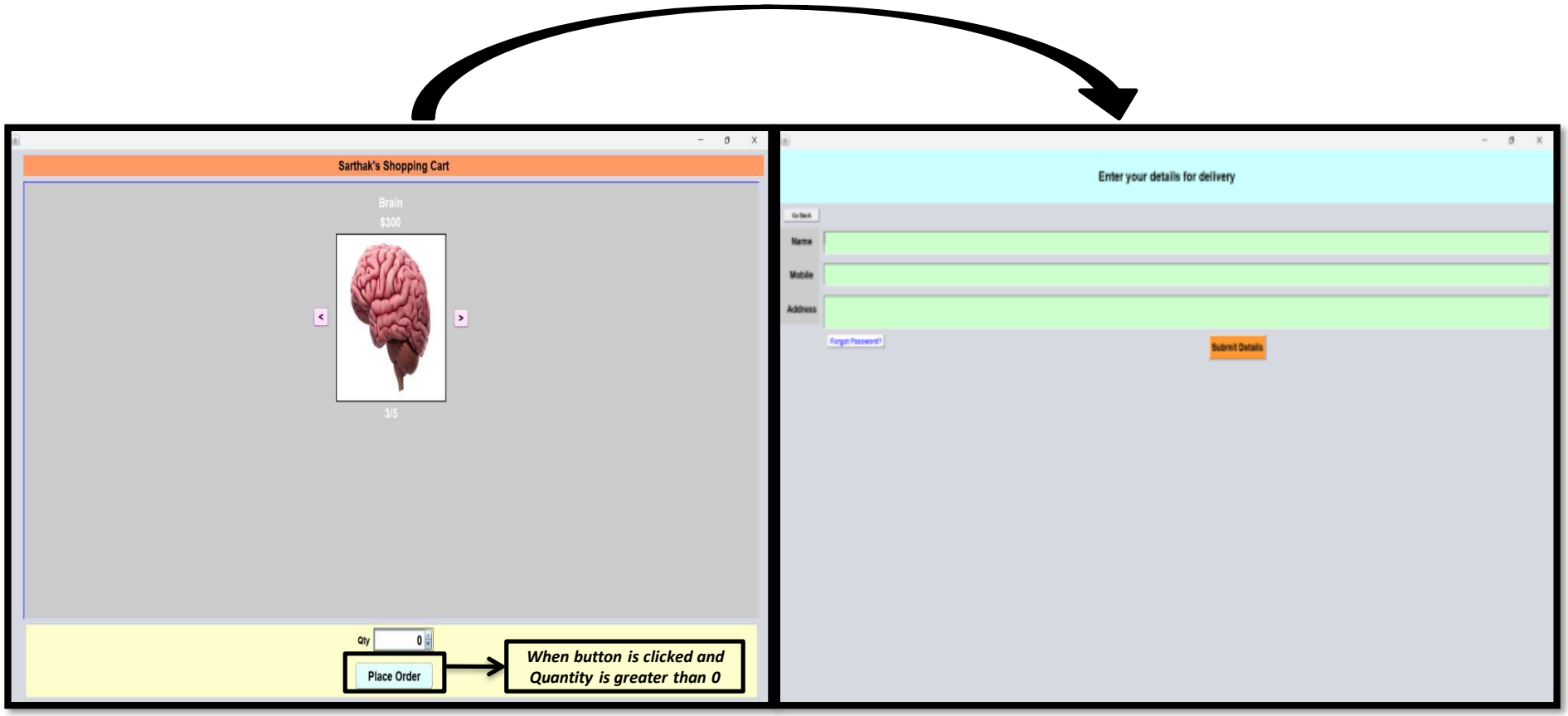


6. *Place order button:*

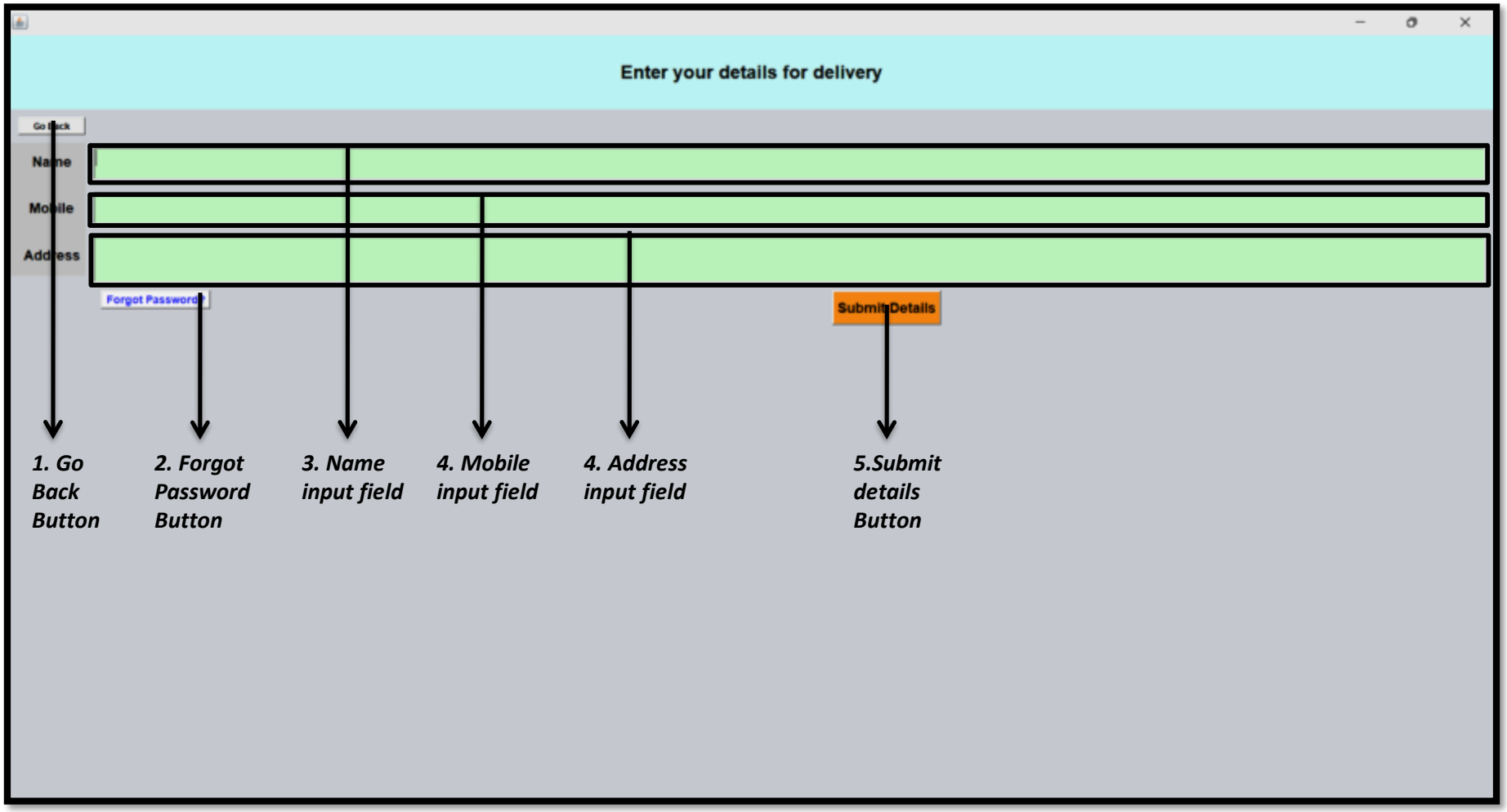
```
//Before the order is placed the condition  for at least 1 quantity selected is checked and next screen is displayed
with all the values associated with item transferred to the screen.
private void proceedBtnActionPerformed(java.awt.event.ActionEvent evt) {
                proceedBtn.setEnabled(false);
                if (Integer.parseInt(jSpinner1.getValue().toString().trim()) >= 1) {
                                jSpinner1.setEnabled(false);
                                nextImgBtn.setEnabled(false);
                                previousImgBtn.setEnabled(false);
                                UserDetailsJForm userDetailFormObj = new
                               UserDetailsJForm(this.itemNamejLabel.getText(), this.itemPrice_jLabel.getText(),
                               this.jSpinner1.getValue().toString(),
String.valueOf((Integer.parseInt(jSpinner1.getValue().toString().trim()) * Integer.parseInt(itemPrice[imageIndex])));
                               userDetailFormObj.setExtendedState(userDetailFormObj.MAXIMIZED_BOTH);
                               userDetailFormObj.setVisible(true);
                               this.setVisible(false);
                                this.dispose();
                }
                else {
                                JOptionPane.showMessageDialog(this, "Please select atleast 1 quantity of the item!",
                                                                "Error", JOptionPane.ERROR_MESSAGE);
                                proceedBtn.setEnabled(true);

                }
}
```



Ordering Screen Transition To User Detail Form Screen



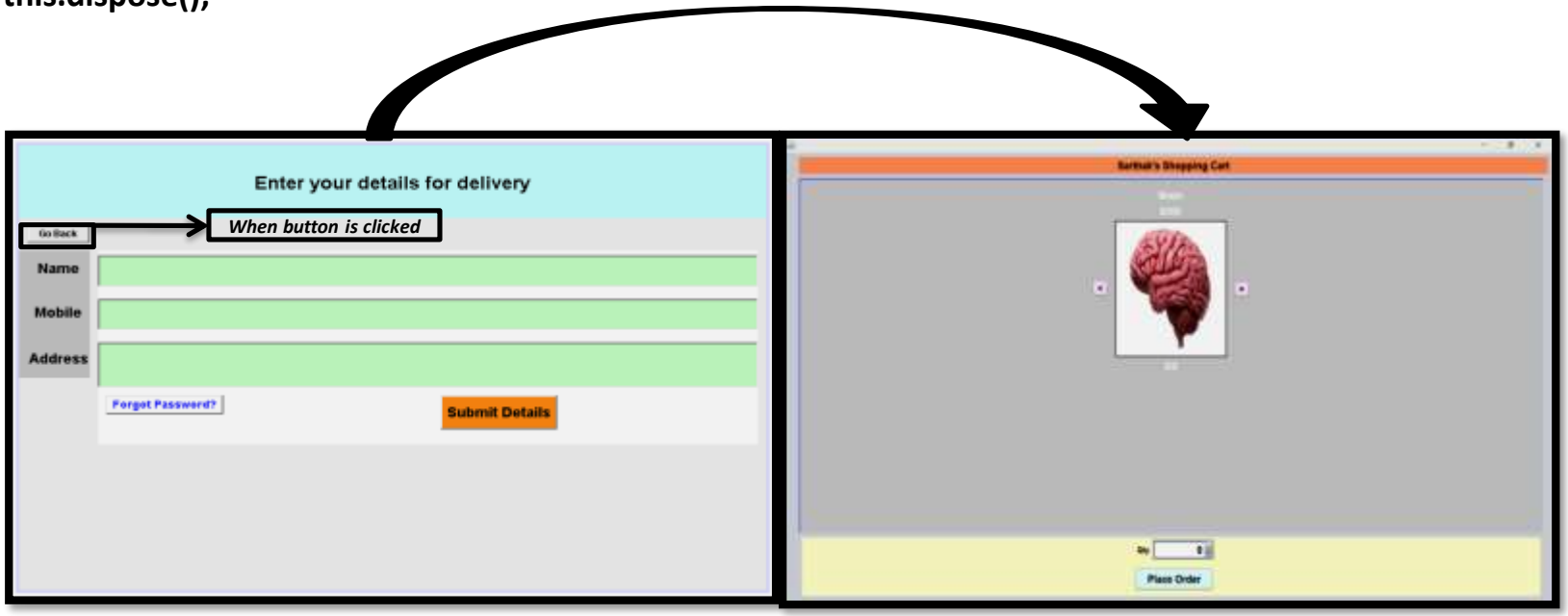
User Detail Form Screen Technical Features



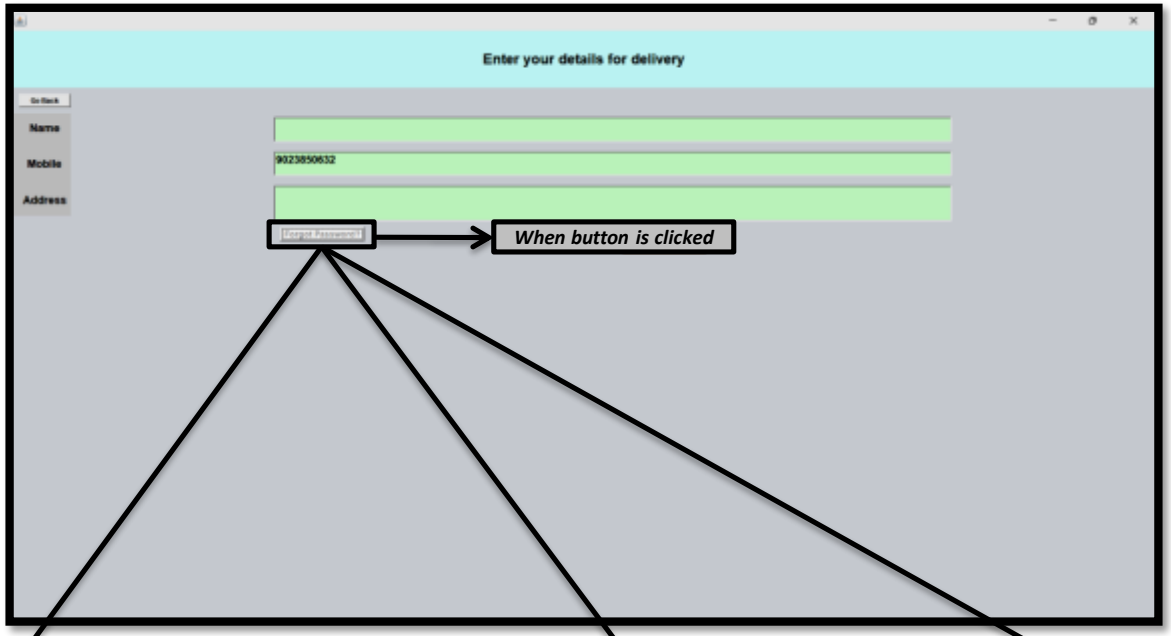
User Detail Form Screen Code snippets

1. Go Back Button:

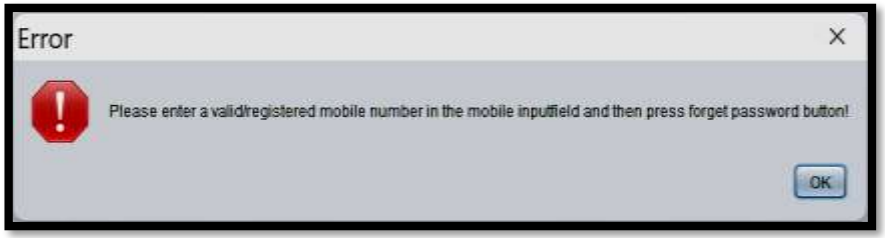
```
//Previous screen is loaded as new and the current is disposed/closed when the button is pressed.
private void goBackBtnActionPerformed(java.awt.event.ActionEvent evt) {
    SubmitDetailsBtn.setEnabled(false);
    forgetPassBtn.setEnabled(false);
    goBackBtn.setEnabled(false);
    new ShoppingJFrame().setVisible(true);
    this.dispose();
}
```



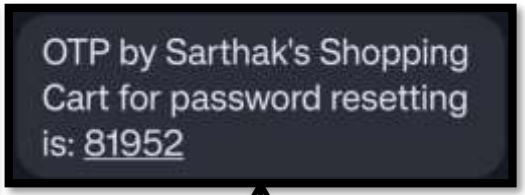
2. Forgot Password Button condition check:

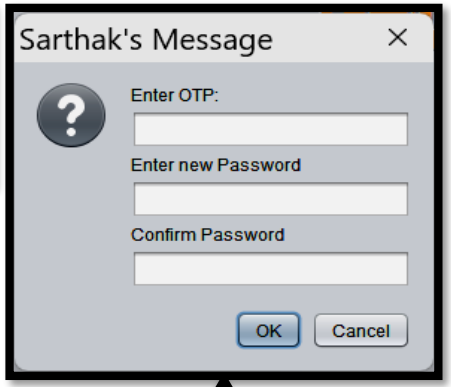


if (Mobiletxtfld.getText().length() != 10)



if (Mobiletxtfld.getText().length() == 10 && rs.next())

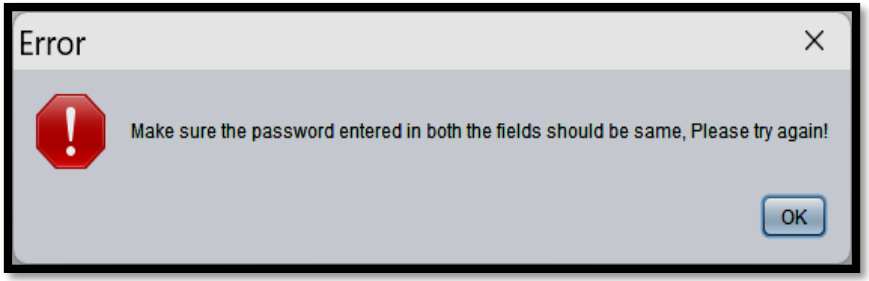


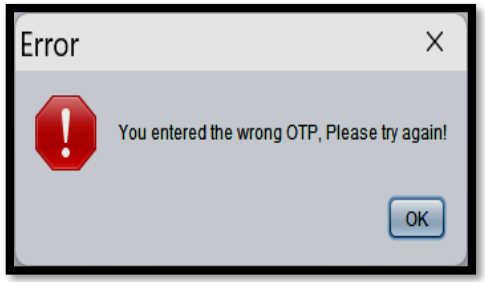


2. <u>Forgot Password Button working (Frontend, Java):</u> When the forgot password button is clicked then the otp is sent to user's mobile from my mobile number as my phone acts as GSM modem with the help of the following code snippet:-

```
//Enter the message to be sent in the string. %20 is added after every space as it represents concatenation + sign in URL encoding.
String getMessage(String str) {
      return "http:" + "//192.168.1.164:8090/SendSMS?username=Sarthak&password=Lucky@1325&phone=?" +
                Mobiletxtfld.getText().trim() + "&message=" + str.replace(" ", "%20");
}
//Establish connection and execute URL  given in the getMessage method above.
URL oracle = new URL(getMessage("OTP by Sarthak's Shopping Cart for password resetting is: " + otpRecorded));
URLConnection yc = oracle.openConnection();
BufferedReader in = new BufferedReader(new InputStreamReader(yc.getInputStream()));
```

Before sending the otp , it is recorded in the otpRecorded variable to verify the otp entered by the user is correct or not. If it is correct then the message is shown else an error message is shown if both the password fields do not match the conditions or the otp is incorrect:-





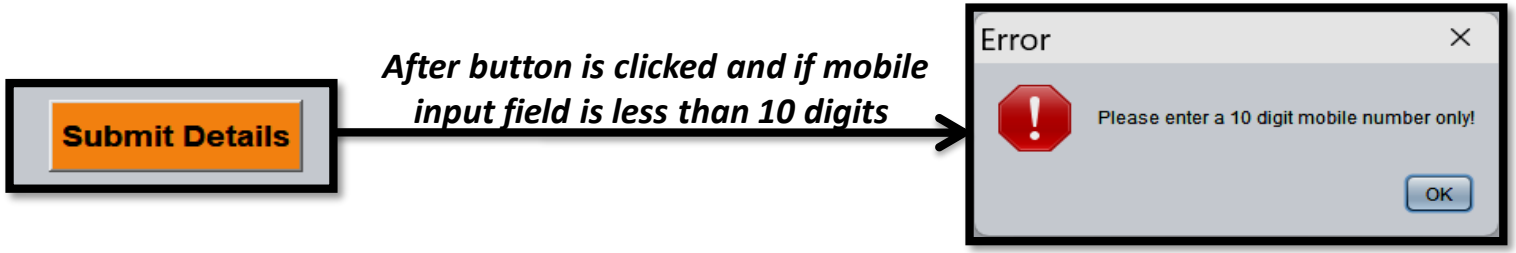
2. <u>Forgot Password Button working (Backend, Java+MYSQL):</u> When the forgot password button is clicked and success message is shown then the data is record in the MYSQL workbench with the help of the following code snippet:-

```
//Connection is established and MYSQL query is executed  to find whether the mobile entered already exist in the database or not because a user who is not registered cannot forget the password.
Connection con;
String query = "SELECT * FROM shoppingcarttb WHERE shoppingcarttb.Mobile = ?";
PreparedStatement stmt = con.prepareStatement(query);
stmt.setString(1, Mobiletxtfld.getText());
ResultSet rs = stmt.executeQuery();

//if resultset gets the mobile number then the changed password is updated in the database and a message for new password is sent to the user, same way as the otp is sent.
If(rs.next()){
            String insertStatement = "UPDATE shoppingcarttb SET Password=? WHERE Mobile=?";
            stmt = con.prepareStatement(insertStatement);
            stmt.setString(1, text2.getText().trim());
            stmt.setString(2, Mobiletxtfld.getText());
            stmt.executeUpdate();

            URL oracle = new URL(getMessage("The new password for your Sarthak's Shopping Cart is: " +
            text2.getText().trim()));
            URLConnection yc = oracle.openConnection();
            BufferedReader in = new BufferedReader(new InputStreamReader(yc.getInputStream()));
}
```

3. Name input field: You can enter your name normally.

4. Mobile input field: You can only enter a 10 digit mobile number. Also it is the primary key in the database as mobile number can't be same for two different person.
The following code snippet is used to do this:-

```
private void MobiletxtfldKeyReleased(java.awt.event.KeyEvent evt) {
// When the key is released it is check that whether the input field has only integer values which can be typed in the input field.
            Mobiletxtfld.setEditable(true);
             if ((evt.getKeyChar() >= '0' && evt.getKeyChar() <= '9' || evt.getKeyCode() ==
                KeyEvent.VK_BACK_SPACE) && Mobiletxtfld.getText().length() < 10 || (evt.getKeyCode() ==
                KeyEvent.VK_BACK_SPACE && Mobiletxtfld.getText().length() == 10))
                        Mobiletxtfld.setEditable(true);

            else {
                        Mobiletxtfld.setEditable(false);
                        Mobiletxtfld.setText(Mobiletxtfld.getText().substring(0, Mobiletxtfld.getText().length()));
                        Mobiletxtfld.setCaretPosition(Mobiletxtfld.getText().length());
            }
}
```

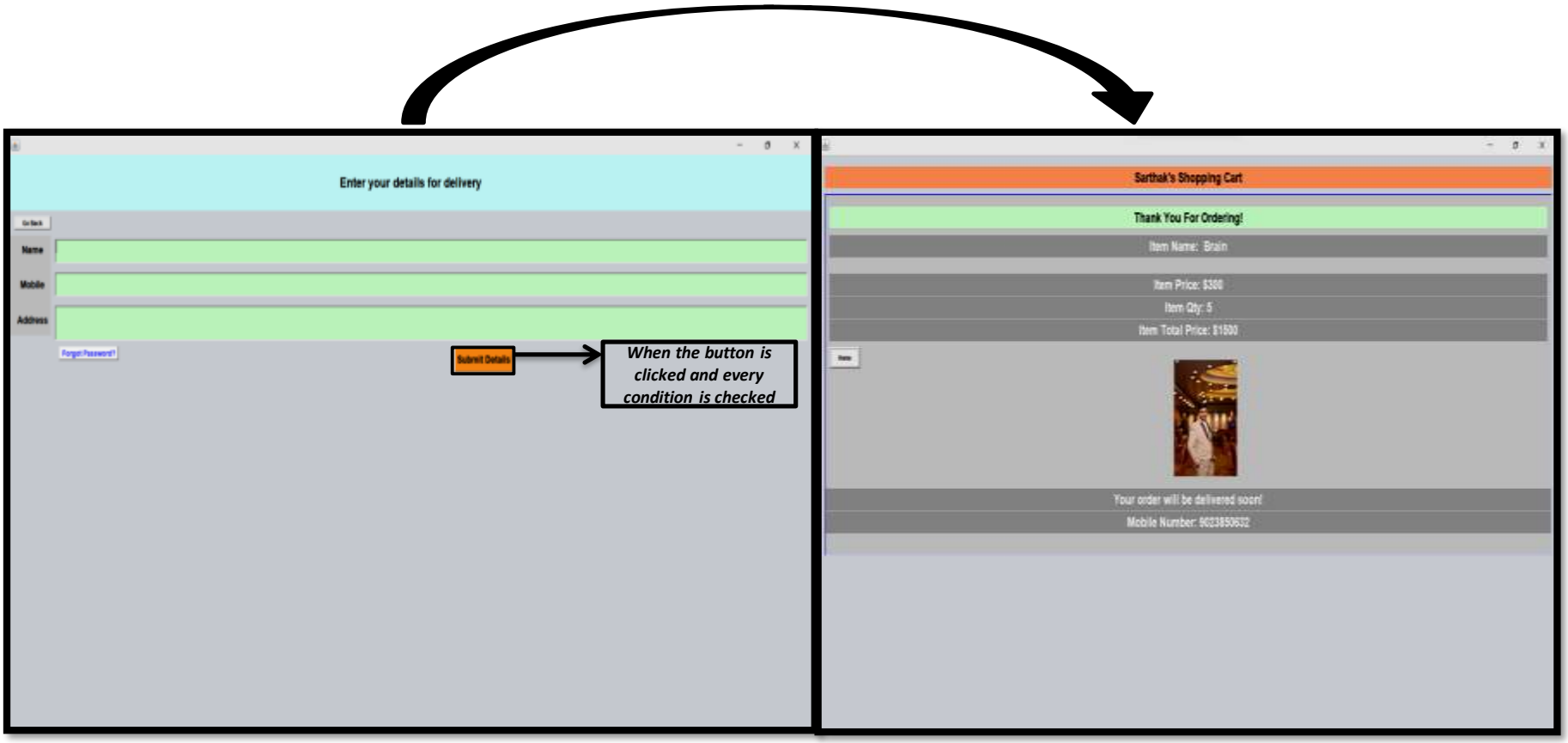


5. Address input field: You can enter your address normally.

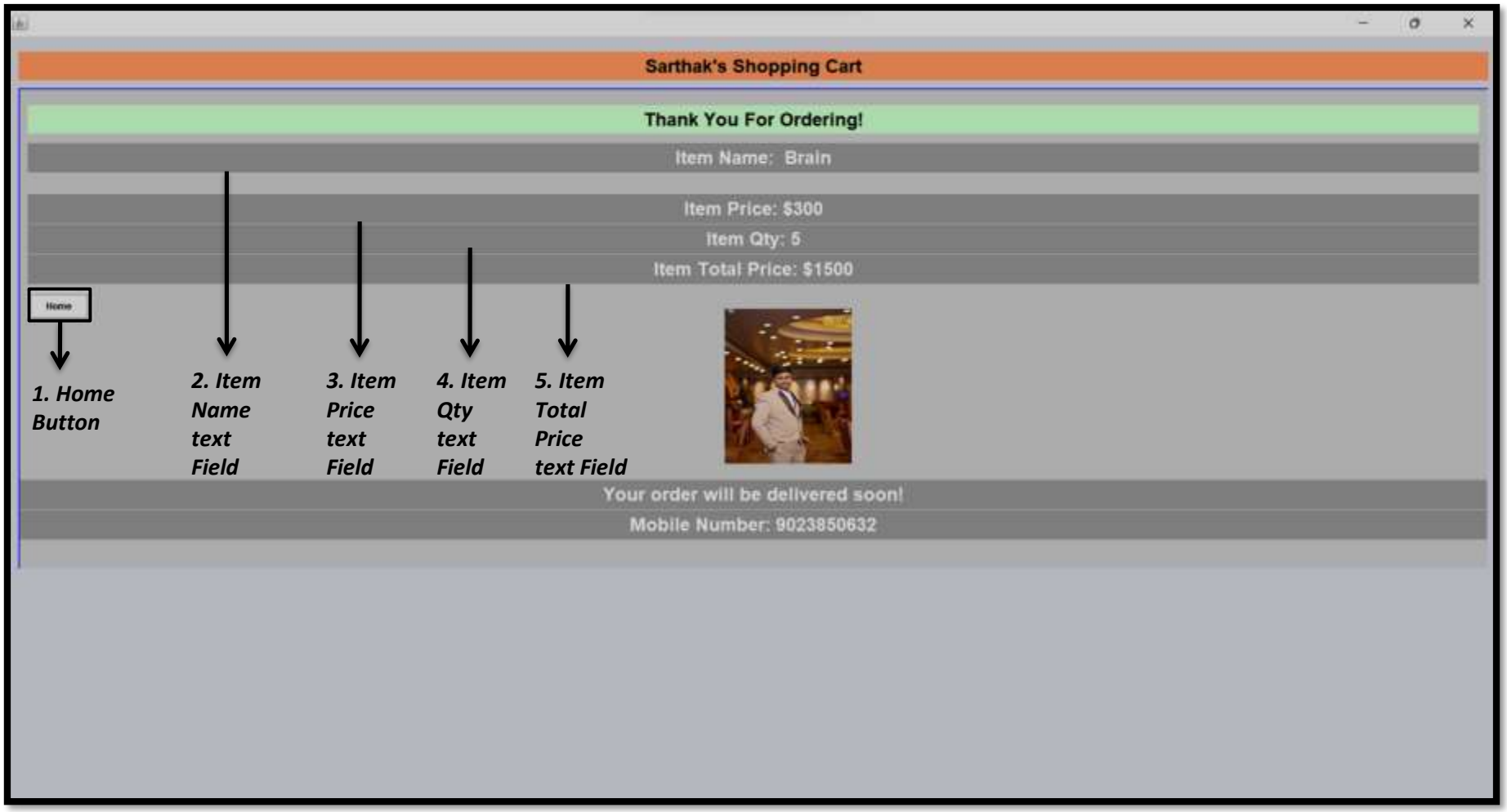
5. <u>Submit details Button:</u> Now this button has the main role in updating data in the database. When this button is clicked all the commands and queries are executed and the respective database in MYSQL is updated. The following are some of the code snippets that are used to update the database:-

```
// If mobile is not present in the database then OTP is sent otherwise user can login with his/her password.
if (!Nametxtfld.getText().isEmpty() && !Addresstxtfld.getText().isEmpty() && !Mobiletxtfld.getText().isEmpty()) {
JTextField text0 = new JTextField(16);
        JTextField text1 = new JPasswordField(16);
        JTextField text2 = new JPasswordField(16);
Object[] fieldObj= {"Enter OTP sent on the entered mobile number", text0,"Enter Password", text1,"Confirm Password", text2 };
        Object[] fieldObj2= {"Enter Password", text1};
         String checkIfMobilePresent = "SELECT * FROM shoppingcarttb WHERE shoppingcarttb.Mobile = ?";
         PreparedStatement stmt = con.prepareStatement(checkIfMobilePresent);
         stmt.setString(1, Mobiletxtfld.getText().trim());
         ResultSet rs = stmt.executeQuery();
         if (!rs.next()) {
                  otpRecorded = new Random().nextInt(99999) + 10000;
 URL oracle = new URL(getMessage("OTP for Sarthak's Shopping Cart for creating new account is: " + otpRecorded));
                 URLConnection yc = oracle.openConnection();
                  BufferedReader in = new BufferedReader(new InputStreamReader(yc.getInputStream()));
JOptionPane.showConfirmDialog(null, fieldObj, "Sarthak's Message", JOptionPane.OK_CANCEL_OPTION);
String insertStatement = "INSERT INTO shoppingcarttb(Name, Mobile, Address, ItemName, ItemQty, ItemPrice, ItemTotalPrice, Date,
Time, Password) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
stmt = con.prepareStatement(insertStatement); LocalDate date = LocalDate.now();DateTimeFormatter formatter =
DateTimeFormatter.ofPattern("dd/MM/yyyy"); String formattedDate = date.format(formatter); LocalTime currentTime =
LocalTime.now();formatter = DateTimeFormatter.ofPattern("hh:mm:ss a"); String formattedTime = currentTime.format(formatter);
                  stmt.setString(1, Nametxtfld.getText().trim()); stmt.setString(2, Mobiletxtfld.getText().trim());
                  stmt.setString(3, Addresstxtfld.getText().trim()); stmt.setString(4, itemName);
                  stmt.setString(5, itemQty); stmt.setString(6, itemPrice);
                  stmt.setString(7, itemTotalPrice); stmt.setString(8, formattedDate);
                  stmt.setString(9, formattedTime); stmt.setString(10, text1.getText().trim());
                  stmt.executeUpdate();
                  }
}
```

User Detail Form Screen Transition To Final Receipt Screen



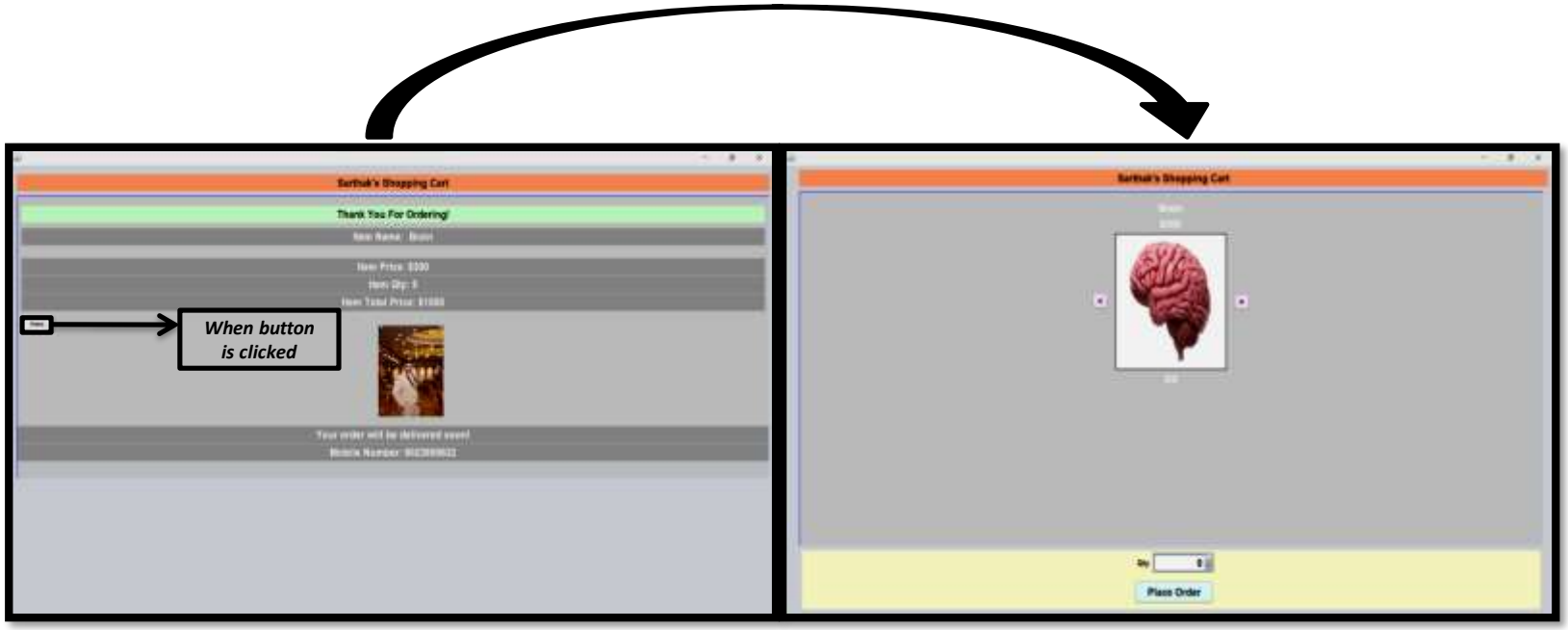
Final Receipt Screen Technical Features



Final Receipt Screen Code snippets

1. Home Button:

```
//First screen i.e. Ordering screen is loaded as new and the current is disposed/closed when the button is pressed.
private void goToHomeBtnActionPerformed(java.awt.event.ActionEvent evt) {
    goToHomeBtn.setEnabled(false);
    new ShoppingJFrame().setVisible(true);
    this.dispose();
}
```



2. Item Name text Field, 3. Item Price text Field, 4. Item Qty text Field and 5. Item Total Price text Field:

```
// When this constructor is called the values that are to  be transferred are passed as arguments.
public FinalReceiptJFrame(String itemName, String price, String qty, String totalPrice) {
    initComponents();
    this.setExtendedState(this.MAXIMIZED_BOTH);

    itemNameJLabel.setText("Item Name:  "+itemName);
    itemPriceJLabel.setText("Item Price: "+price);
    itemQtyJLabel.setText("Item Qty: "+qty);
    itemTotalPriceJLabel1.setText("Item Total Price: $"+totalPrice);
}
```

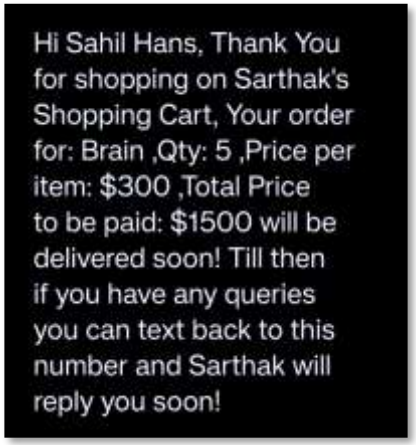
MYSQL Representation of Data

This is how the data is displayed in the MYSQL database.

Name	Mobile	Address	ItemName	ItemQty	ItemPrice	ItemTotalPrice	Date	Time	Password	MessageSent
Sahil Hans	6280334155	Sector-15, Panchkukla Haryana 134103	Brain	5	$300	1500	04/07/2023	05:40:36 PM	87654321	Yes

Message sent to the client's mobile

This is how the message is sent to the client's mobile.



Bibliography

Scan the QR Codes with Google lens

Documentation: ***https://docs.oracle.com/javase/tutorial/jdbc/basics/index.html***

Video Reference link: ***https://www.youtube.com/watch?v=7LkB5p-HzTo***