Module Code & Module Title

CS5004NA Emerging Programming Platforms and Technologies

Assessment Weightage & Type

30% Group Coursework

Title (Where Required): Mobile Information System

Year and Semester

2020-21Autumn

Group Name:			
SN	**Student Name**	**College ID**	**University ID**
1.	**Sunab Shrestha**	**19031737**	**NP01CP4A190408**

Table of Contents

Table of Figures

Table of Tables

1. Proposal

We have decided to build a "Mobile IS" which will store and display all the mobile phone brands and models. This system will help user to know, search and store different smart phones of different brands.

- **List of data:**

Brand: Manufactures of different types of smart phone. It accepts string datatype. To retrieve data combo box is used.

Model Name: Name of smartphones. It accepts string data type. To retrieve data text field is used.

Model Id: Each smartphone has a unique id. It accepts integer data type. To retrieve data text field is used

color: Color as per required by the user. It accepts string data type. To retrieve data radio button is used.

Price: Price of smartphone. It accepts integer data type. To retrieve data text field is used.

- **List of features:**
  We can search mobile using their brands.
  We can search mobile using the price.
  We can add, and also delete data from the table.
  We can import, export values.

- Tools used for developing
  NetBeans:
  NetBeans is an open-source integrated development environment (IDE) for developing with Java, PHP, C++, and other programming languages. NetBeans is also referred to as a platform of modular components used for developing Java desktop applications. (Techopedia, n.d.)

  Balsamiq:
  Balsamiq Wireframes is a user interface design tool for creating wireframes (sometimes called mockups or low-fidelity prototypes). You can use it to generate digital sketches of your idea or concept for an application or website, to facilitate discussion and understanding before any code is written. (Balsamiq, n.d.)

2. Introduction

This assignment contributes 30% to the overall marks for this module and involves group work. For this coursework we are required to develop an information system of your own which should keep records, store it on table and also display it. The project requires to use NetBeans IDE to create a Java based menu information system.

Here we have created a “Mobile Information System” using NetBeans and Balsamiq wireframe. It is a user-friendly information system which helps user to Add, Delete, Import, export, and search products. We used features like Radio buttons, Combo Box, text fields and labels, Table, menu bar to create this program.

3. Binary Search

3.1. Working mechanism of Binary Search Algorithm

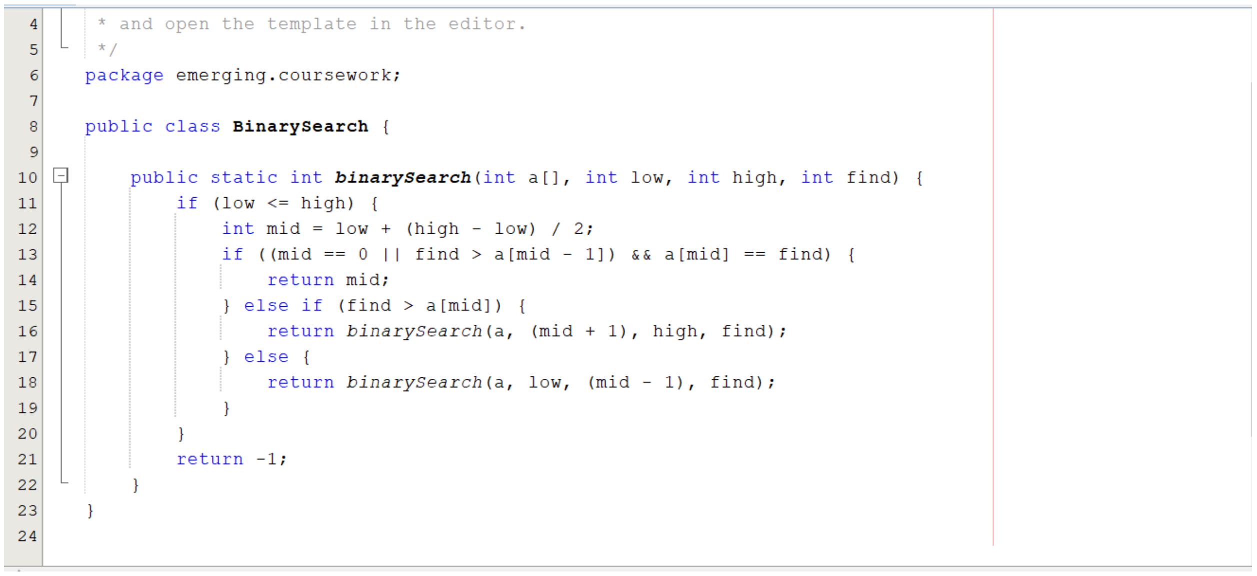
Binary search is used to search a key element from multiple elements. It is a fast search algorithm with run-time complexity of O(log n). This search algorithm works on the principle of divide and conquer.

Binary search looks for a particular item by comparing the middle most item of the collection. If a match occurs, then the index of item is returned. If the middle item is greater than the item, then the item is searched in the sub-array to the left of the middle item. Otherwise, the item is searched for in the sub-array to the right of the middle item. This process continues on the sub-array as well until the size of the subarray reduces to zero. (Tutorials Point, n.d.)

3.2. Implementation of Binary search in program

For smooth running, and efficient running of the program binary search was implemented while searching for a price and while adding a product. Binary search was used to get the index of an element in a sorted array. After knowing the index of an element program was written to search for the exact price.

Similarly, while adding the product all the value of model ID column was added to an array and was sorted using merge sort. Then binary search was implemented to search for the value entered in the model ID text field. If the **binary Search** method returns -1 then the product will be added else an error will be displayed.



```
 * and open the template in the editor.
 */
package emerging.coursework;

public class BinarySearch {

    public static int binarySearch(int a[], int low, int high, int find) {
        if (low <= high) {
            int mid = low + (high - low) / 2;
            if ((mid == 0 || find > a[mid - 1]) && a[mid] == find) {
                return mid;
            } else if (find > a[mid]) {
                return binarySearch(a, (mid + 1), high, find);
            } else {
                return binarySearch(a, low, (mid - 1), find);
            }
        }
        return -1;
    }
}
```

Figure 1: implementation of Binary search

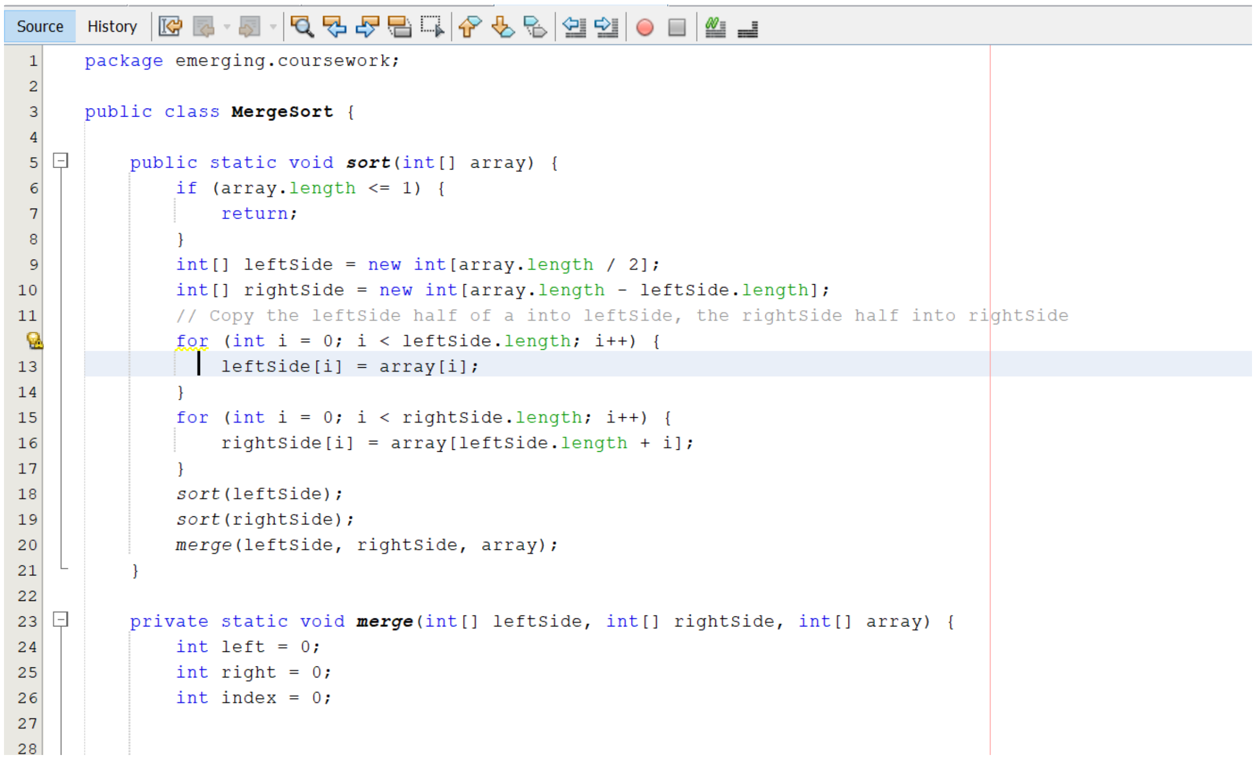
4. Sorting Algorithm Name

4.1. Working mechanism of the Merge Sort Program

Merge sort is one of the most efficient sorting algorithms. It works on the principle of Divide and Conquer. Merge sort repeatedly breaks down a list into several subsists until each subsist consists of a single element and merging those subsists in a manner that results into a sorted list. (Interviewbit, n.d.)

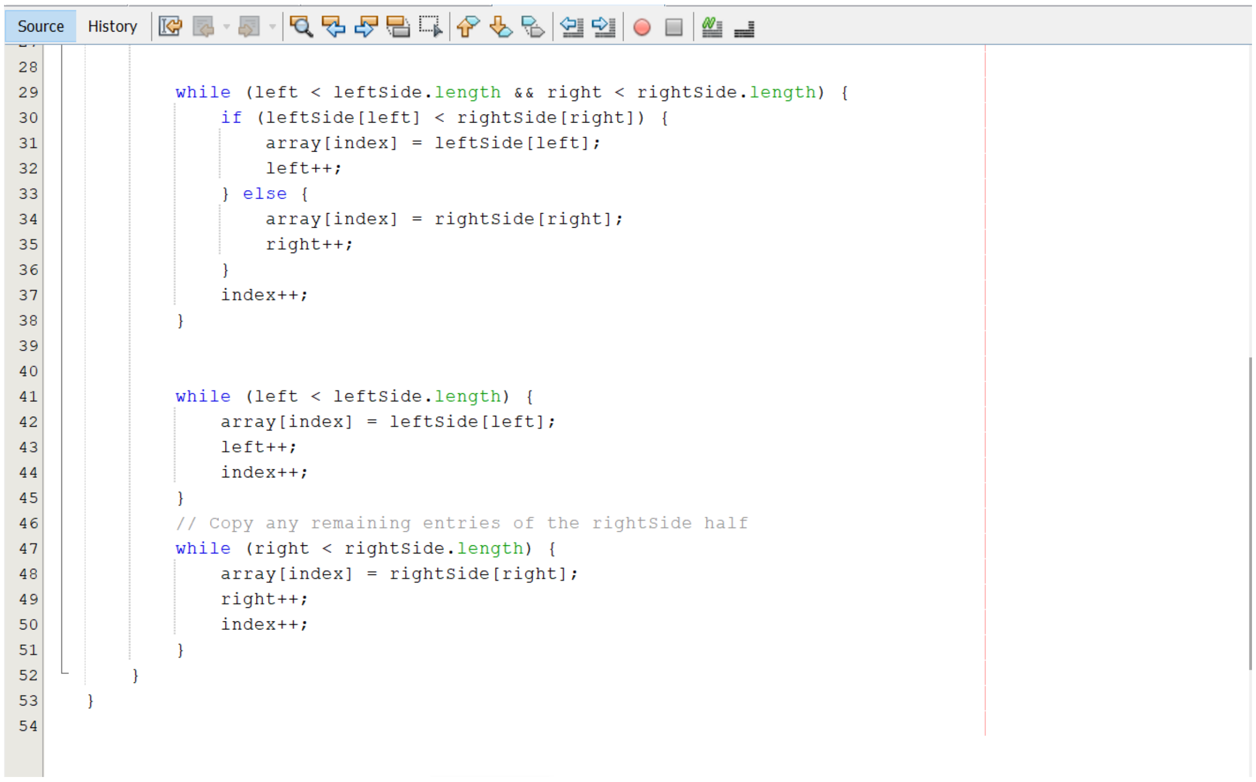
4.2. Implementation of Merge sort in program

Merge sort was implemented in program when an array of element was to be sorted. For merge sort two method **sort** and **merge** was used in the program. Merge sort was mostly used while adding the product and while searching of product based on price. While adding the product merge sort was implemented on array of elements present on model ID column then the array was sorted in ascending order for carrying out binary Search algorithm. While searching the product merge sort was implemented on array of elements present on price column then the array was sorted in ascending order for carrying binary Search algorithm.



```java
package emerging.coursework;

public class MergeSort {

    public static void sort(int[] array) {
        if (array.length <= 1) {
            return;
        }
        int[] leftSide = new int[array.length / 2];
        int[] rightSide = new int[array.length - leftSide.length];
        // Copy the leftSide half of a into leftSide, the rightSide half into rightSide
        for (int i = 0; i < leftSide.length; i++) {
            leftSide[i] = array[i];
        }
        for (int i = 0; i < rightSide.length; i++) {
            rightSide[i] = array[leftSide.length + i];
        }
        sort(leftSide);
        sort(rightSide);
        merge(leftSide, rightSide, array);
    }

    private static void merge(int[] leftSide, int[] rightSide, int[] array) {
        int left = 0;
        int right = 0;
        int index = 0;

```

Figure 2: implementation of merge sort



```java

        while (left < leftSide.length && right < rightSide.length) {
            if (leftSide[left] < rightSide[right]) {
                array[index] = leftSide[left];
                left++;
            } else {
                array[index] = rightSide[right];
                right++;
            }
            index++;
        }


        while (left < leftSide.length) {
            array[index] = leftSide[left];
            left++;
            index++;
        }
        // Copy any remaining entries of the rightSide half
        while (right < rightSide.length) {
            array[index] = rightSide[right];
            right++;
            index++;
        }
    }
}

```

Figure 3: implementation of merge sort

5. Method Description

Method Name	Method Description
public MobileInfo()	This is the method of a class in which the method where component are built is called
private void closeActionPerformed(java.awt.event.ActionEvent evt)	This is the private method with void which is non-return type method, which is called when a close menu item is clicked.
private void searchBrandBTNActionPerformed(java.awt.event.ActionEvent evt)	This is the private method with void which is non-return type method, which is called when search brand button is clicked.
public void search(String findBrand)	This is public method with void which is non-return type method, which is called by searchBrandBTNActionPerformed method. In this method actual search function takes place.
private void importBTNActionPerformed(java.awt.event.ActionEvent evt)	This is the private method with void which is non-return type method, which is called when import button is clicked.
private void addBTNActionPerformed(java.awt.event.ActionEvent evt)	This is the private method with void which is non-return type method, which is called when Add button is clicked.
private void clearBTNActionPerformed(java.awt.event.ActionEvent evt)	This is the private method with void which is non-return type method, which is called when Clear button is clicked.
private void modelIdTextKeyPressed(java.awt.event.KeyEvent evt)	This is the private method with void which is non-return type method, which compels the program to not accept a string value as an input.
private void modelIdTextKeyReleased(java.awt.event.KeyEvent evt)	This is the private method with void which is non-return type method, this method makes model ID text editable.
private void priceTextKeyPressed(java.awt.event.KeyEvent evt)	This is the private method with void which is non-return type method, which compels the program to not accept a string value as an input.
private void priceTextKeyReleased(java.awt.event.KeyEvent evt)	This is the private method with void which is non-return type method, this method makes model ID text editable.

private void searchPriceTextKeyPressed(java.awt.event.KeyEvent evt)	This is the private method with void which is non-return type method, which compels the program to not accept a string value as an input.
private void searchPriceTextKeyReleased(java.awt.event.KeyEvent evt)	This is the private method with void which is non-return type method, this method makes model ID text editable.
private void exportBTNActionPerformed(java.awt.event.ActionEvent evt)	This is the private method with void which is non-return type method, this method exports the table to CSV file by using export button.
private void searchbyPriceBTNActionPerformed(java.awt.event.ActionEvent evt)	This is the private method with void which is non-return type method, this method helps to search the product by its price.
private void deleteBtnActionPerformed(java.awt.event.ActionEvent evt)	This is the private method with void which is non-return type method, This method helps to delete selected row.
private void importTableMenuActionPerformed(java.awt.event.ActionEvent evt)	This is the private method with void which is non-return type method, This method helps to import data from CSV file to table by using import button
private void exportTableMenuActionPerformed(java.awt.event.ActionEvent evt)	This is the private method with void which is non-return type method, This method helps to export CSV file by clicking menu Item.
private void openMenuActionPerformed(java.awt.event.ActionEvent evt)	This is the private method with void which is non-return type method, This method helps to open Help file.
public String getBrand()	This is a public method which returns string value of selected items from brand combo box.
public String getmodelName()	This is a public method which returns string value of model name text field.
public int getmodelID()	This is a public method which returns Integer value of model ID text field.
public int getprice()	This is a public method which returns Integer value of Price text field.
public String getcolour()	This is a public method which returns string value from radio buttons of colors.
public static void main(String args[])	This is the main method for this class.

Table 1: method description

6. Testing

6.1. Test: 1

Test case	1
Action	The project was opened and runed.
Expected Result	It was expected that GUI of project would open.
Actual Result	The GUI opened.
Test Result	The test was successful.

Table 2: test 1

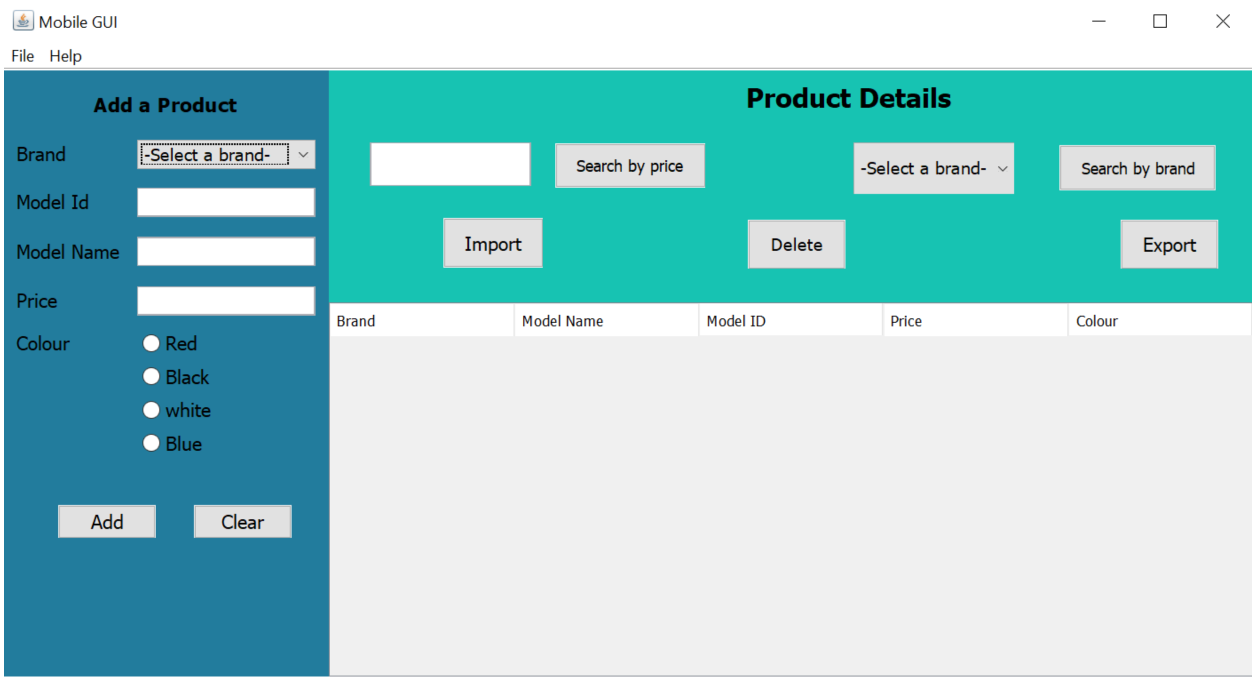
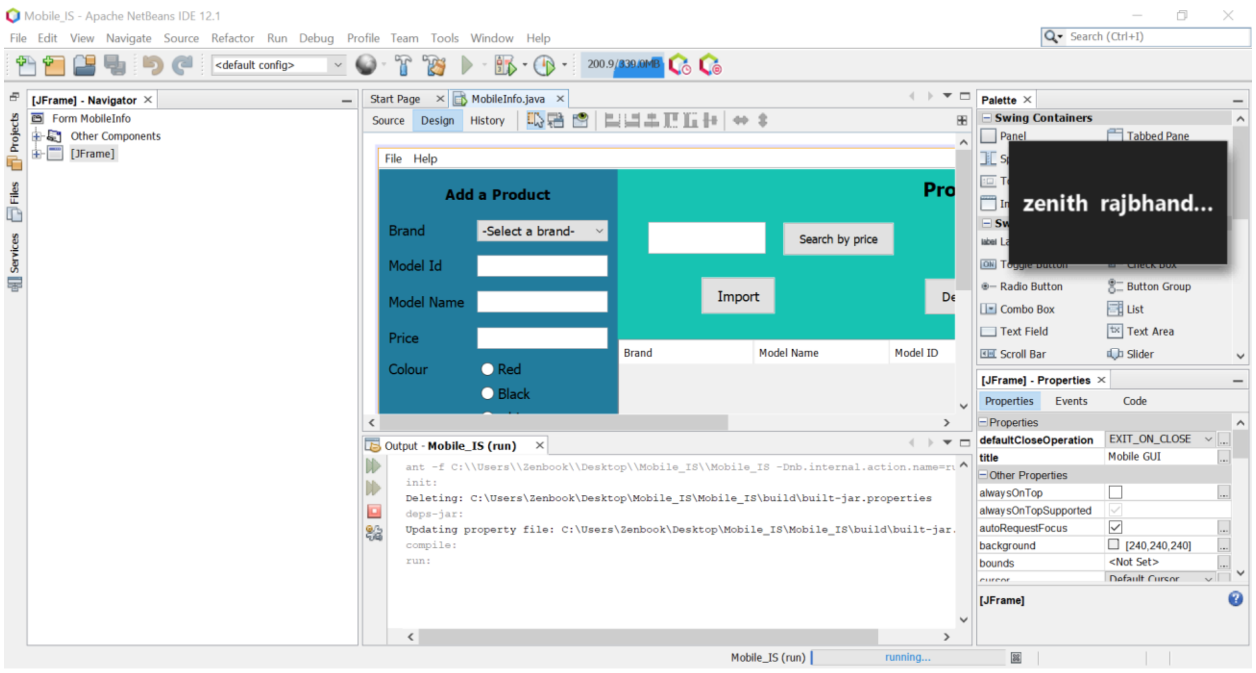


Figure 4: test 1



Output - Mobile_IS (run)

```
ant -f C:\\Users\\Zenbook\\Desktop\\Mobile_IS\\Mobile_IS -Dnb.internal.action.name=ru
init:
Deleting: C:\Users\Zenbook\Desktop\Mobile_IS\Mobile_IS\build\built-jar.properties
deps-jar:
Updating property file: C:\Users\Zenbook\Desktop\Mobile_IS\Mobile_IS\build\built-jar.
compile:
run:
```

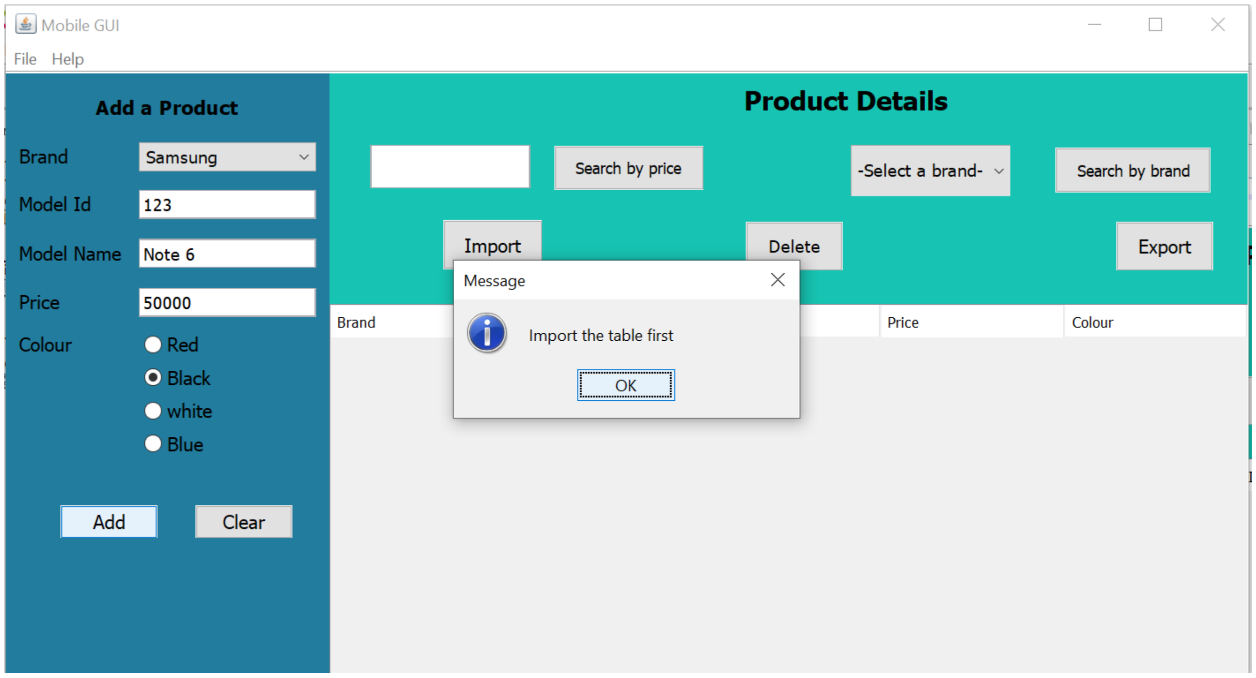
Properties	
defaultCloseOperation	EXIT_ON_CLOSE
title	Mobile GUI
Other Properties	
alwaysOnTop	☐
alwaysOnTopSupported	✓
autoRequestFocus	✓
background	[240,240,240]
bounds	<Not Set>
cursor	Default Cursor

Figure 5: test 1

6.2. Test: 2

Test case	2
Action	Add button was clicked without importing table.
Expected Result	It was expected that the product would not be added before importing table.
Actual Result	The product was not added.
Test Result	The test was successful.

Table 3: test 2



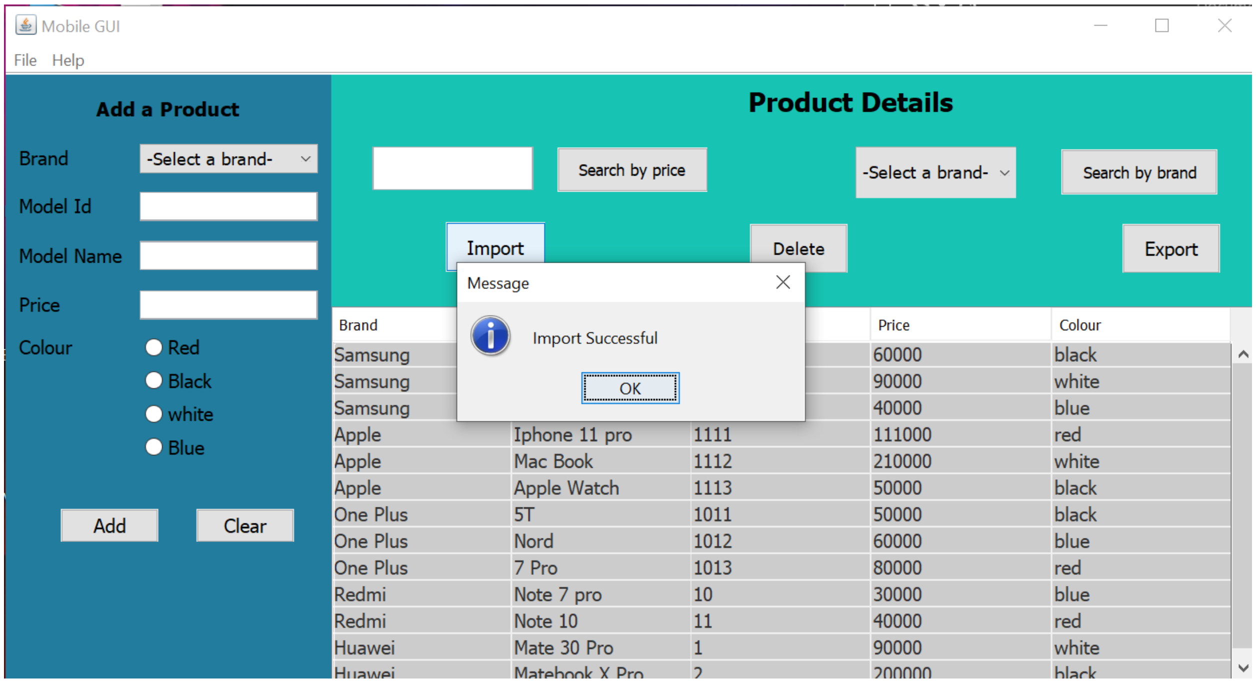
Import the table first

Figure 6: test 2

6.3. Test: 3

Test case	3
Action	Importing table.
Expected Result	It was expected to import table that was exported.
Actual Result	The table was imported
Test Result	The test was successful.

Table 4: test 3



Import Successful

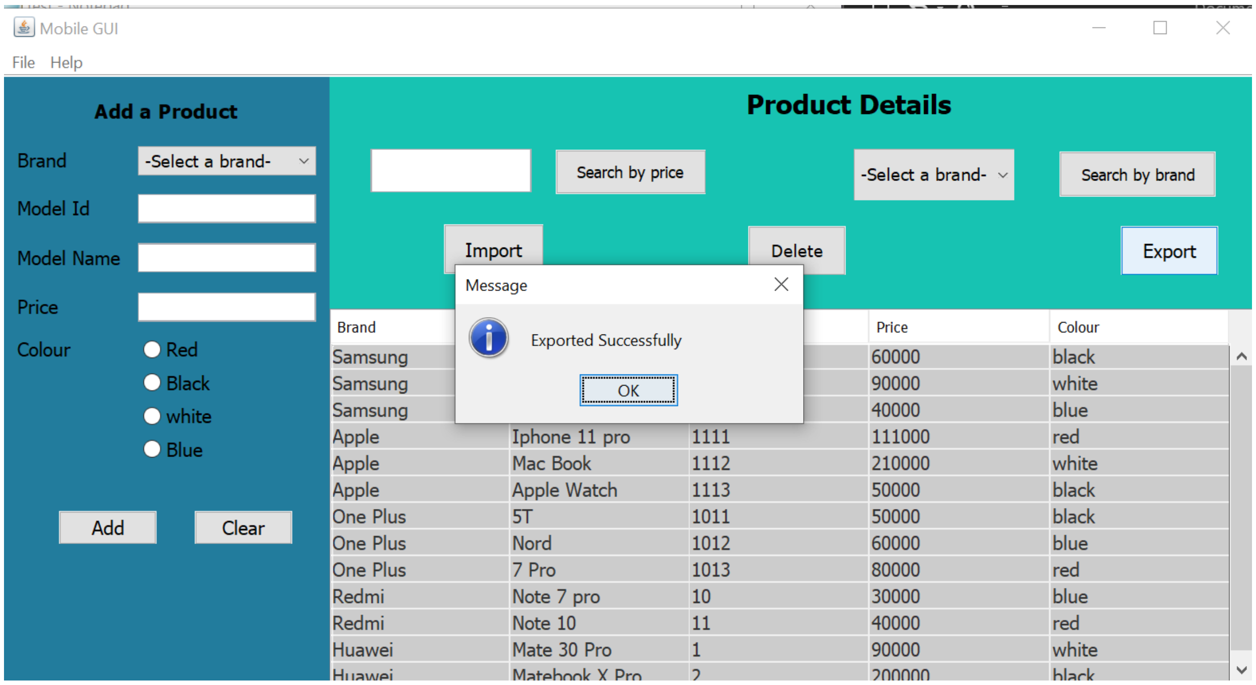
Brand			Price	Colour
Samsung			60000	black
Samsung			90000	white
Samsung			40000	blue
Apple	Iphone 11 pro	1111	111000	red
Apple	Mac Book	1112	210000	white
Apple	Apple Watch	1113	50000	black
One Plus	5T	1011	50000	black
One Plus	Nord	1012	60000	blue
One Plus	7 Pro	1013	80000	red
Redmi	Note 7 pro	10	30000	blue
Redmi	Note 10	11	40000	red
Huawei	Mate 30 Pro	1	90000	white
Huawei	Matebook X Pro	2	200000	black

Figure 7:test 3

6.4. Test: 4

Test case	4
Action	Exporting table.
Expected Result	It was expected to export the table.
Actual Result	The table was Exported
Test Result	The test was successful.

Table 5: test 4



Exported Successfully

Brand			Price	Colour
Samsung			60000	black
Samsung			90000	white
Samsung			40000	blue
Apple	Iphone 11 pro	1111	111000	red
Apple	Mac Book	1112	210000	white
Apple	Apple Watch	1113	50000	black
One Plus	5T	1011	50000	black
One Plus	Nord	1012	60000	blue
One Plus	7 Pro	1013	80000	red
Redmi	Note 7 pro	10	30000	blue
Redmi	Note 10	11	40000	red
Huawei	Mate 30 Pro	1	90000	white
Huawei	Matebook X Pro	2	200000	black

Figure 8: test 4

6.5. Test: 5

Test case	5
Action	Opening documentation in help or pressing CTRL+O.
Expected Result	It was expected a PDF document would pop out.
Actual Result	The document was popped out.
Test Result	The test was successful.

Table 6: test 5

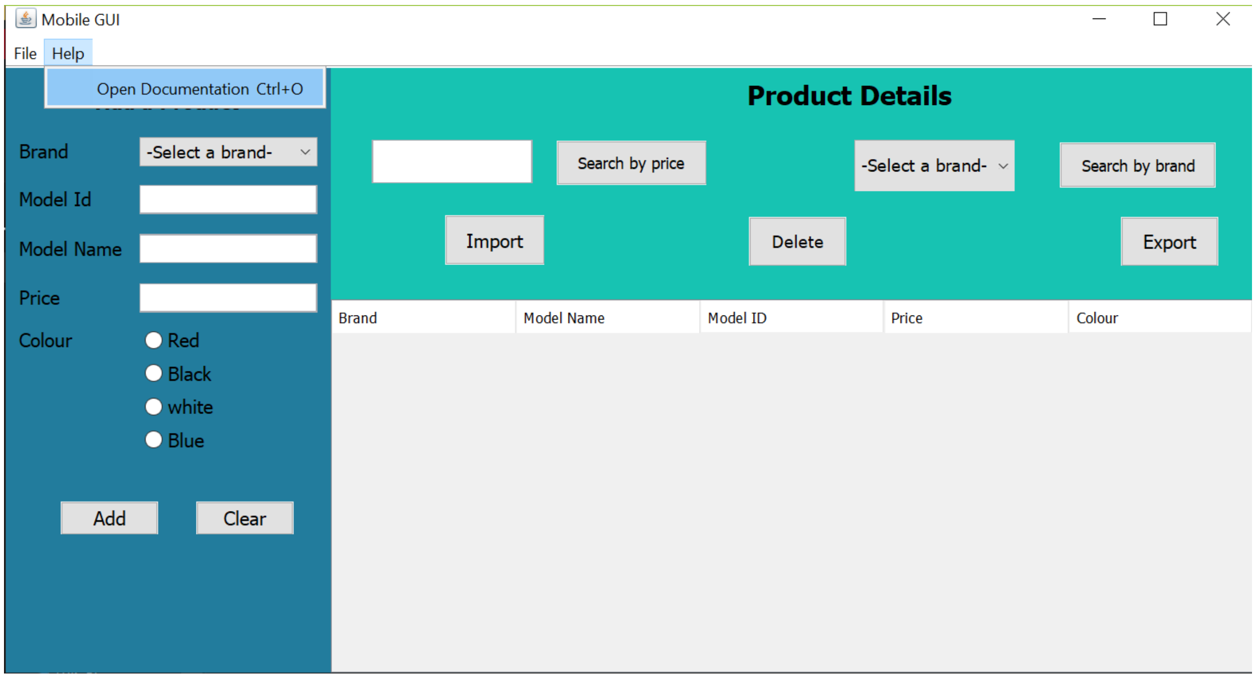
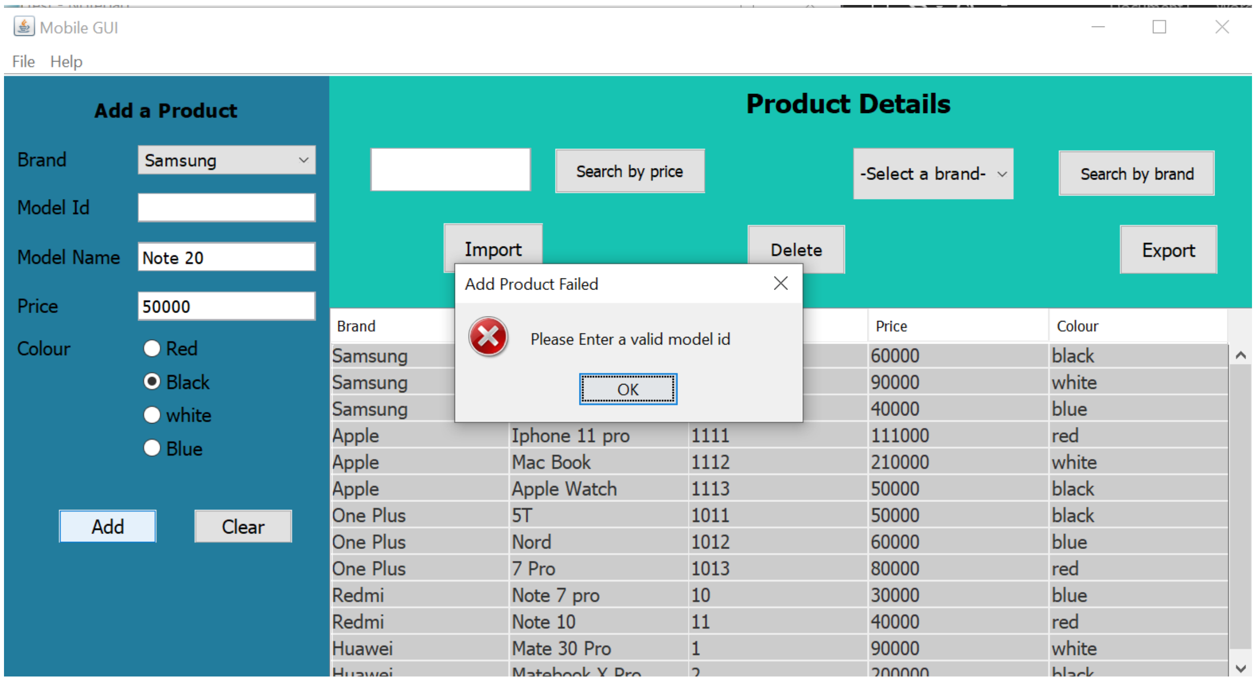


Figure 9: test 5

6.6. Test: 6

Test case	6
Action	Adding product keeping Model ID empty.
Expected Result	An error would pop out saying enter valid model ID
Actual Result	The Error popped out.
Test Result	The test was successful.

Table 7: test 6



Mobile GUI

File Help

Add a Product

Brand: Samsung

Model Id:

Model Name: Note 20

Price: 50000

Colour: Red [✓ Black] white Blue

Add Clear

Product Details

Search by price | -Select a brand- | Search by brand

Import | Delete | Export

Add Product Failed

Please Enter a valid model id

OK

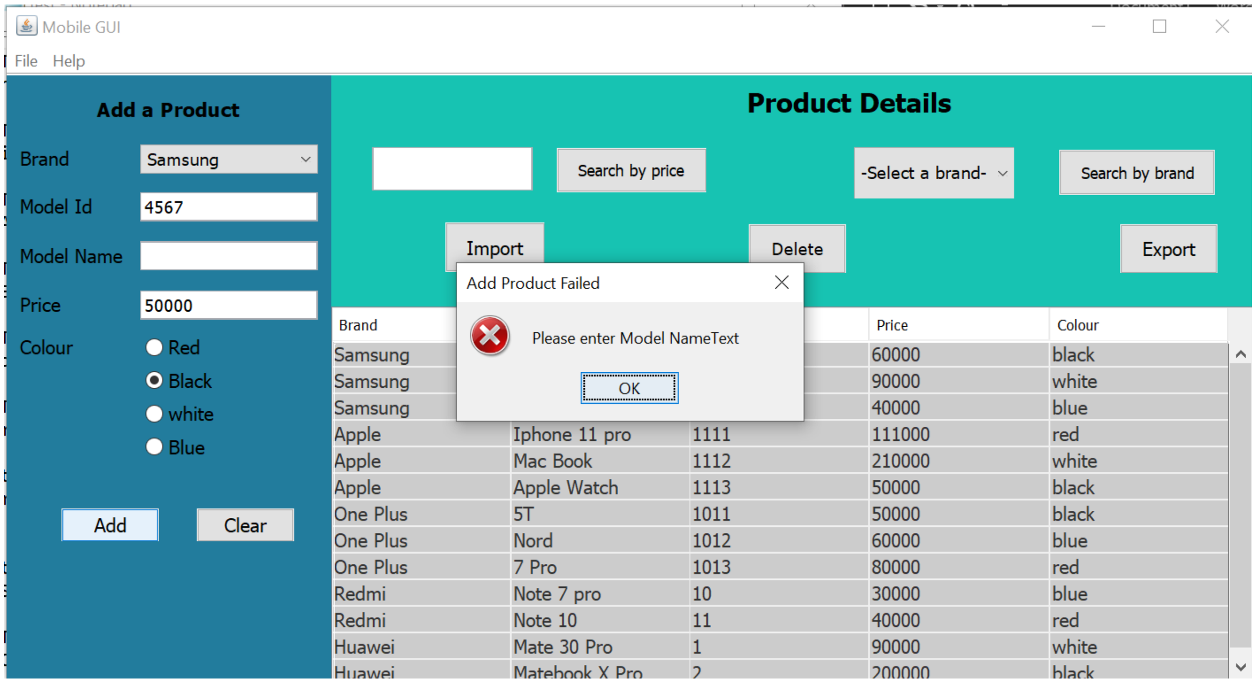
Brand			Price	Colour
Samsung			60000	black
Samsung			90000	white
Samsung			40000	blue
Apple	Iphone 11 pro	1111	111000	red
Apple	Mac Book	1112	210000	white
Apple	Apple Watch	1113	50000	black
One Plus	5T	1011	50000	black
One Plus	Nord	1012	60000	blue
One Plus	7 Pro	1013	80000	red
Redmi	Note 7 pro	10	30000	blue
Redmi	Note 10	11	40000	red
Huawei	Mate 30 Pro	1	90000	white
Huawei	Matebook X Pro	2	200000	black

Figure 10: test 6

6.7. Test: 7

Test case	7
Action	Adding product keeping Model Name empty.
Expected Result	An error would pop out saying enter model name
Actual Result	The Error popped out.
Test Result	The test was successful.

Table 8: test 7



Brand: Samsung
Model Id: 4567
Model Name:
Price: 50000
Colour: Red [✓ Black] white Blue

Add Product Failed

Please enter Model NameText

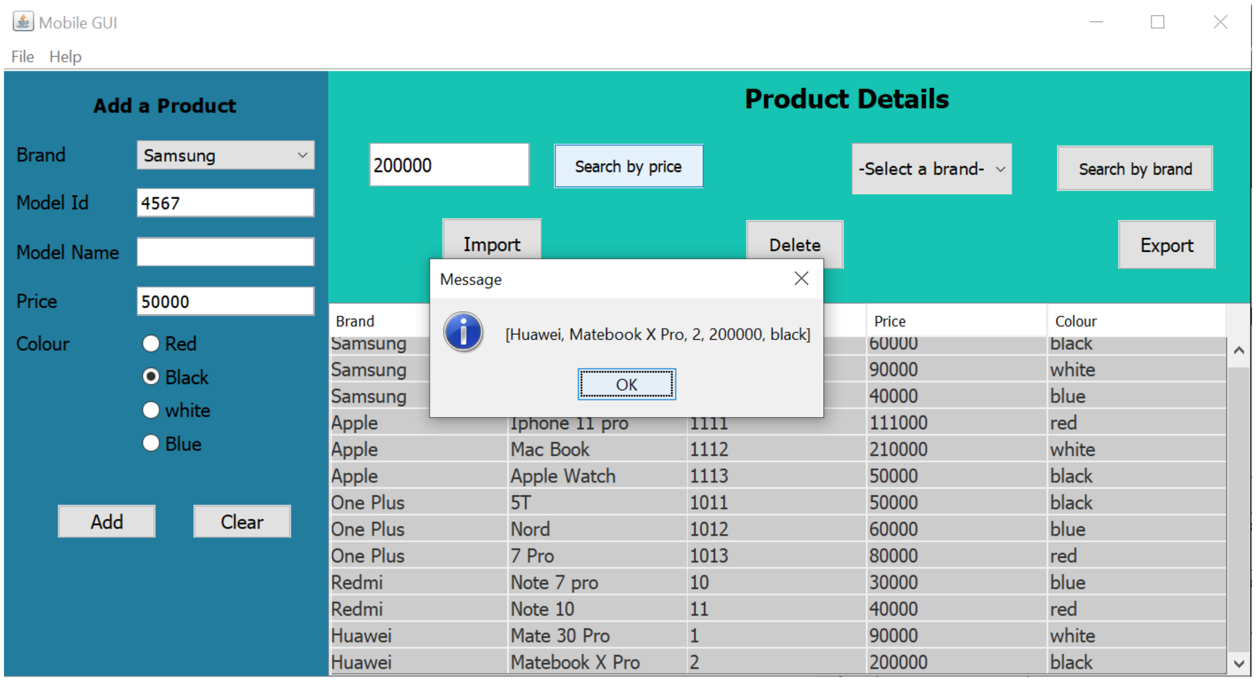
Brand			Price	Colour
Samsung			60000	black
Samsung			90000	white
Samsung			40000	blue
Apple	Iphone 11 pro	1111	111000	red
Apple	Mac Book	1112	210000	white
Apple	Apple Watch	1113	50000	black
One Plus	5T	1011	50000	black
One Plus	Nord	1012	60000	blue
One Plus	7 Pro	1013	80000	red
Redmi	Note 7 pro	10	30000	blue
Redmi	Note 10	11	40000	red
Huawei	Mate 30 Pro	1	90000	white
Huawei	Matebook X Pro	2	200000	black

Figure 11: test 7

6.8. Test: 8

Test case	8
Action	Searching Product by their Price.
Expected Result	The details of the searched product would be displayed
Actual Result	The details were displayed.
Test Result	The test was successful.

Table 9: test 8



Brand: Samsung
Model Id: 4567
Model Name:
Price: 50000
Colour: Red [✓ Black] white Blue

200000

[Huawei, Matebook X Pro, 2, 200000, black]

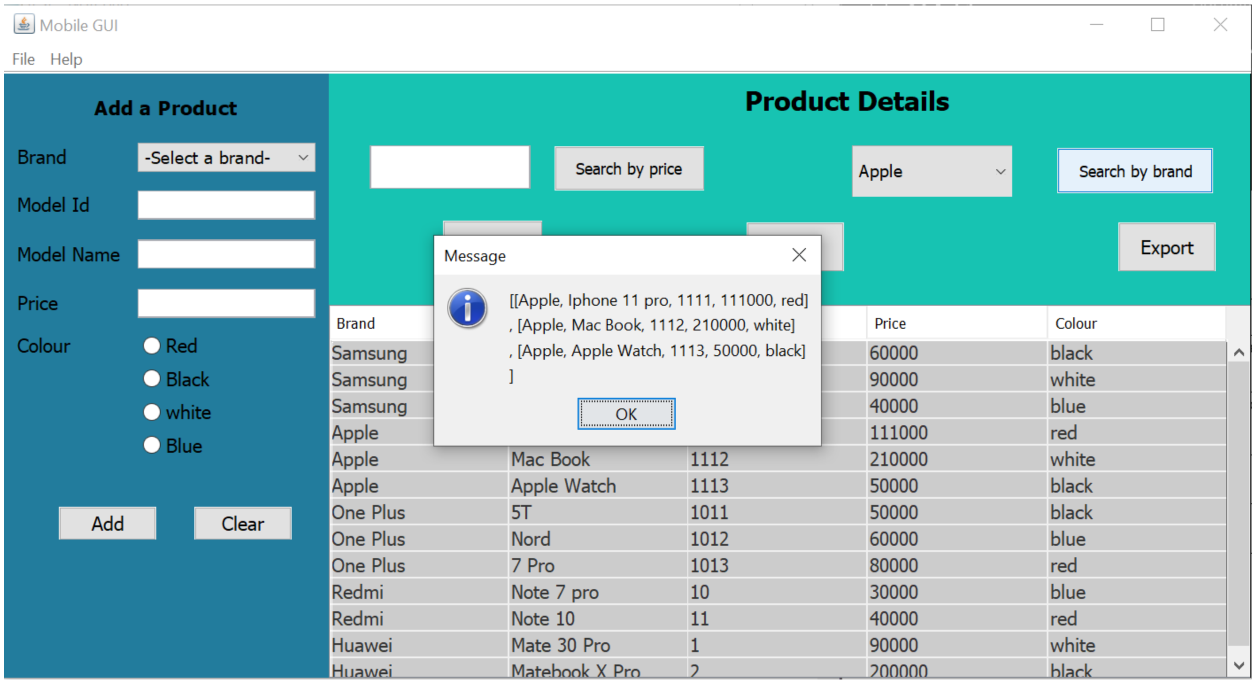
Brand			Price	Colour
Samsung			60000	black
Samsung			90000	white
Samsung			40000	blue
Apple	Iphone 11 pro	1111	111000	red
Apple	Mac Book	1112	210000	white
Apple	Apple Watch	1113	50000	black
One Plus	5T	1011	50000	black
One Plus	Nord	1012	60000	blue
One Plus	7 Pro	1013	80000	red
Redmi	Note 7 pro	10	30000	blue
Redmi	Note 10	11	40000	red
Huawei	Mate 30 Pro	1	90000	white
Huawei	Matebook X Pro	2	200000	black

Figure 12: test 8

6.9. Test: 9

Test case	9
Action	Searching Product by their Brand
Expected Result	The details of the searched product would be displayed by the brand.
Actual Result	The details were displayed.
Test Result	The test was successful.

Table 10: test 9



Brand: -Select a brand-
Model Id
Model Name
Price
Colour: Red, Black, white, Blue

Apple

[[Apple, Iphone 11 pro, 1111, 111000, red]
, [Apple, Mac Book, 1112, 210000, white]
, [Apple, Apple Watch, 1113, 50000, black]
]

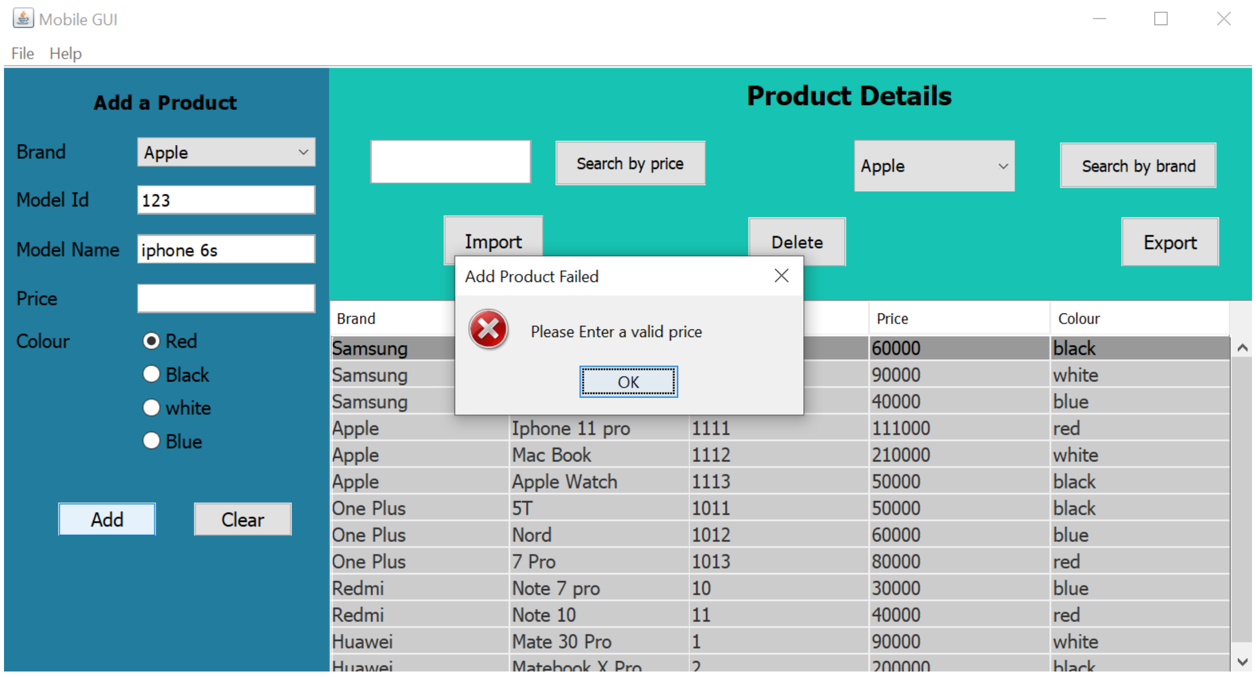
Brand			Price	Colour
Samsung			60000	black
Samsung			90000	white
Samsung			40000	blue
Apple			111000	red
Apple	Mac Book	1112	210000	white
Apple	Apple Watch	1113	50000	black
One Plus	5T	1011	50000	black
One Plus	Nord	1012	60000	blue
One Plus	7 Pro	1013	80000	red
Redmi	Note 7 pro	10	30000	blue
Redmi	Note 10	11	40000	red
Huawei	Mate 30 Pro	1	90000	white
Huawei	Matebook X Pro	2	200000	black

Figure 13: test 9

6.10. Test: 10

Test case	10
Action	Adding product keeping price empty.
Expected Result	The product would not be added popping to enter valid price.
Actual Result	The product was not added and box popped out asking to enter valid price.
Test Result	The test was successful.

Table 11: test 10



Brand: Apple
Model Id: 123
Model Name: iphone 6s
Price:
Colour: [✓ Red] Black white Blue

Search by brand: Apple

Add Product Failed

Please Enter a valid price

OK

Brand			Price	Colour
Samsung			60000	black
Samsung			90000	white
Samsung			40000	blue
Apple	Iphone 11 pro	1111	111000	red
Apple	Mac Book	1112	210000	white
Apple	Apple Watch	1113	50000	black
One Plus	5T	1011	50000	black
One Plus	Nord	1012	60000	blue
One Plus	7 Pro	1013	80000	red
Redmi	Note 7 pro	10	30000	blue
Redmi	Note 10	11	40000	red
Huawei	Mate 30 Pro	1	90000	white
Huawei	Matebook X Pro	2	200000	black

Figure 14: test 10

7. Conclusion

Overall, this coursework was a new and fascinating experience where we learned a lot about creating an Information System. We also got to know about how to use NetBeans and also its different features. During the process of completing the project, we met with various confusions and difficulties. These confusions were cleared with research and immense help provided by our tutors.

8. References

Balsamiq. (n.d.). Retrieved from Balsamiq: https://balsamiq.com/wireframes/desktop/docs/intro/

Interviewbit. (n.d.). Retrieved from interviewBit: https://www.interviewbit.com/tutorial/merge-sort-algorithm/#:~:text=Merge%20sort%20is%20one%20of,results%20into%20a%20sorted%20list.

Techopedia. (n.d.). Retrieved from Techopedia: https://www.techopedia.com/definition/24735/netbeans

Tutorials Point. (n.d.). Retrieved from tutorialspoint.com: https://www.tutorialspoint.com/data_structures_algorithms/binary_search_algorithm.htm

9. Appendix A

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package emerging.coursework;

import javax.swing.JOptionPane;
import java.awt.Color;
import java.awt.Desktop;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.lang.System.Logger;
import java.lang.System.Logger.Level;
import java.util.Arrays;
import java.util.LinkedList;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author Zenbook
 */
public class MobileInfo extends javax.swing.JFrame {
```

```
/**
 * Creates new form MobileGUI
 */
public MobileInfo() {
   initComponents();
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

   colourButton = new javax.swing.ButtonGroup();
   addProduct = new javax.swing.JPanel();
   jLabel1 = new javax.swing.JLabel();
   brand = new javax.swing.JLabel();
   modelId = new javax.swing.JLabel();
   modelName = new javax.swing.JLabel();
   price = new javax.swing.JLabel();
   colour = new javax.swing.JLabel();
   brandCombobox = new javax.swing.JComboBox<>();
   modelIdText = new javax.swing.JTextField();
   modelNameText = new javax.swing.JTextField();
   priceText = new javax.swing.JTextField();
```

```
redButton = new javax.swing.JRadioButton();
blackButton = new javax.swing.JRadioButton();
whiteButton = new javax.swing.JRadioButton();
blueButton = new javax.swing.JRadioButton();
addBTN = new javax.swing.JButton();
clearBTN = new javax.swing.JButton();
jPanel2 = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
productTable = new javax.swing.JTable();
jLabel7 = new javax.swing.JLabel();
searchPriceText = new javax.swing.JTextField();
searchbyPriceBTN = new javax.swing.JButton();
searchBrandCombo = new javax.swing.JComboBox<>();
searchBrandBTN = new javax.swing.JButton();
importBTN = new javax.swing.JButton();
exportBTN = new javax.swing.JButton();
deleteBtn = new javax.swing.JButton();
menuBar = new javax.swing.JMenuBar();
fileMenu = new javax.swing.JMenu();
importTableMenu = new javax.swing.JMenuItem();
exportTableMenu = new javax.swing.JMenuItem();
close = new javax.swing.JMenuItem();
helpMenu = new javax.swing.JMenu();
openMenu = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Mobile GUI");

addProduct.setBackground(new java.awt.Color(34, 124, 157));
```

```
    jLabel1.setFont(new java.awt.Font("Tahoma", 1, 15)); // NOI18N
    jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel1.setText("Add a Product");

    brand.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
    brand.setText("Brand");

    modelId.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
    modelId.setText("Model Id");

    modelName.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
    modelName.setText("Model Name");

    price.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
    price.setText("Price");

    colour.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
    colour.setText("Colour");

    brandCombobox.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
    brandCombobox.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "-Select a
brand-", "Samsung", "Apple", "One Plus", "Redmi", "Huawei" }));

    modelIdText.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
    modelIdText.addKeyListener(new java.awt.event.KeyAdapter() {
      public void keyPressed(java.awt.event.KeyEvent evt) {
        modelIdTextKeyPressed(evt);
      }
    public void keyReleased(java.awt.event.KeyEvent evt) {
        modelIdTextKeyReleased(evt);
    }
});

modelNameText.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N

priceText.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
priceText.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyPressed(java.awt.event.KeyEvent evt) {
        priceTextKeyPressed(evt);
    }
    public void keyReleased(java.awt.event.KeyEvent evt) {
        priceTextKeyReleased(evt);
    }
});

colourButton.add(redButton);
redButton.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
redButton.setText("Red");
redButton.setName("Red"); // NOI18N
redButton.setOpaque(false);

colourButton.add(blackButton);
blackButton.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
blackButton.setText("Black");
blackButton.setName("black"); // NOI18N
blackButton.setOpaque(false);
```

```
colourButton.add(whiteButton);
whiteButton.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
whiteButton.setText("white");
whiteButton.setName("white"); // NOI18N
whiteButton.setOpaque(false);

colourButton.add(blueButton);
blueButton.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
blueButton.setText("Blue");
blueButton.setName("blue"); // NOI18N
blueButton.setOpaque(false);

addBTN.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
addBTN.setText("Add");
addBTN.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        addBTNActionPerformed(evt);
    }
});

clearBTN.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
clearBTN.setText("Clear");
clearBTN.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        clearBTNActionPerformed(evt);
    }
});

javax.swing.GroupLayout addProductLayout = new javax.swing.GroupLayout(addProduct);
```

```
        addProduct.setLayout(addProductLayout);

        addProductLayout.setHorizontalGroup(

            addProductLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
addProductLayout.createSequentialGroup()

                .addContainerGap()

.addGroup(addProductLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

                    .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
addProductLayout.createSequentialGroup()

.addGroup(addProductLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

                            .addComponent(colour, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                            .addComponent(price, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                            .addComponent(modelName, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 87, Short.MAX_VALUE)

                            .addComponent(brand, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                            .addComponent(modelId, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(addProductLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

                            .addComponent(brandCombobox, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)

                            .addComponent(redButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                            .addComponent(priceText)

                            .addComponent(modelNameText)
                    .addComponent(modelIdText)
                    .addComponent(blackButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(whiteButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(blueButton, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
            .addContainerGap())
          .addGroup(addProductLayout.createSequentialGroup()
            .addGap(42, 42, 42)
            .addComponent(addBTN, javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(29, 29, 29)
            .addComponent(clearBTN, javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(28, Short.MAX_VALUE))
        );
        addProductLayout.setVerticalGroup(
          addProductLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(addProductLayout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(addProductLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(brand, javax.swing.GroupLayout.PREFERRED_SIZE, 24,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(brandCombobox, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
```

```
.addGroup(addProductLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(modelId, javax.swing.GroupLayout.PREFERRED_SIZE, 26,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(modelIdText, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(addProductLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(modelName, javax.swing.GroupLayout.PREFERRED_SIZE, 26,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(modelNameText, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(addProductLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(price, javax.swing.GroupLayout.PREFERRED_SIZE, 26,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(priceText, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(addProductLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(redButton)
                .addComponent(colour, javax.swing.GroupLayout.PREFERRED_SIZE, 26,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(0, 0, 0)
            .addComponent(blackButton)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(whiteButton)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(blueButton)
            .addGap(34, 34, 34)
```

```
.addGroup(addProductLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(clearBTN)
                .addComponent(addBTN))
              .addContainerGap(104, Short.MAX_VALUE))
        );

        redButton.setActionCommand("red");
        blackButton.setActionCommand("black");
        whiteButton.setActionCommand("white");
        blueButton.setActionCommand("blue");

        jPanel2.setBackground(new java.awt.Color(23, 195, 178));

        productTable.setBackground(new java.awt.Color(204, 204, 204));
        productTable.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
        productTable.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        productTable.setForeground(new java.awt.Color(51, 51, 51));
        productTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
                "Brand", "Model Name", "Model ID", "Price", "Colour"
            }
        ) {
            boolean[] canEdit = new boolean [] {
                false, false, false, false, false
            };

            public boolean isCellEditable(int rowIndex, int columnIndex) {
                return canEdit [columnIndex];
            }
        });
        productTable.setOpaque(false);
        productTable.setRowHeight(20);
        productTable.setSelectionBackground(new java.awt.Color(153, 153, 153));
        productTable.setSelectionForeground(new java.awt.Color(0, 0, 0));
        productTable.setShowGrid(false);
        productTable.getTableHeader().setResizingAllowed(false);
        productTable.getTableHeader().setReorderingAllowed(false);
        jScrollPane1.setViewportView(productTable);
        if (productTable.getColumnModel().getColumnCount() > 0) {
            productTable.getColumnModel().getColumn(0).setResizable(false);
            productTable.getColumnModel().getColumn(1).setResizable(false);
            productTable.getColumnModel().getColumn(2).setResizable(false);
            productTable.getColumnModel().getColumn(3).setResizable(false);
            productTable.getColumnModel().getColumn(4).setResizable(false);
        }

        jLabel7.setFont(new java.awt.Font("Tahoma", 1, 20)); // NOI18N
        jLabel7.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel7.setText("Product Details");

        searchPriceText.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        searchPriceText.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                searchPriceTextKeyPressed(evt);
            }
            public void keyReleased(java.awt.event.KeyEvent evt) {
                searchPriceTextKeyReleased(evt);
            }
        });

        searchbyPriceBTN.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
        searchbyPriceBTN.setText("Search by price");
        searchbyPriceBTN.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                searchbyPriceBTNActionPerformed(evt);
            }
        });

        searchBrandCombo.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
        searchBrandCombo.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "-Select a
    brand-", "Samsung", "One plus", "Redmi", "Apple", "Huawei" }));

        searchBrandBTN.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N
        searchBrandBTN.setText("Search by brand");
        searchBrandBTN.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                searchBrandBTNActionPerformed(evt);
            }
        });

        importBTN.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        importBTN.setText("Import");
        importBTN.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                importBTNActionPerformed(evt);
            }
        });

        exportBTN.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
        exportBTN.setText("Export");
        exportBTN.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                exportBTNActionPerformed(evt);
            }
        });

        deleteBtn.setFont(new java.awt.Font("Tahoma", 0, 13)); // NOI18N
        deleteBtn.setText("Delete");
        deleteBtn.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                deleteBtnActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
            .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 218,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(196, 196, 196))
          .addGroup(jPanel2Layout.createSequentialGroup()
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
              .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(31, 31, 31)
                .addComponent(searchPriceText, javax.swing.GroupLayout.PREFERRED_SIZE, 121,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(searchbyPriceBTN)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 111,
Short.MAX_VALUE))
              .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(86, 86, 86)
                .addComponent(importBTN)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(deleteBtn, javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(201, 201, 201)
                .addComponent(exportBTN, javax.swing.GroupLayout.PREFERRED_SIZE, 73,
javax.swing.GroupLayout.PREFERRED_SIZE))
              .addGroup(jPanel2Layout.createSequentialGroup()
                .addComponent(searchBrandCombo, javax.swing.GroupLayout.PREFERRED_SIZE, 121,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(34, 34, 34)
                .addComponent(searchBrandBTN)))
            .addGap(27, 27, 27))
        );
        jPanel2Layout.setVerticalGroup(
          jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
            .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 41,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(13, 13, 13)
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                      .addComponent(searchPriceText, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
                      .addComponent(searchbyPriceBTN, javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE))
                  .addGap(23, 23, 23)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                      .addComponent(exportBTN, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)
                      .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(deleteBtn, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(importBTN, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                  .addGap(25, 25, 25)
                  .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 0,
Short.MAX_VALUE))
              .addGroup(jPanel2Layout.createSequentialGroup()
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(searchBrandBTN, javax.swing.GroupLayout.PREFERRED_SIZE, 34, 
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(searchBrandCombo, javax.swing.GroupLayout.PREFERRED_SIZE, 39, 
javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGap(0, 0, Short.MAX_VALUE))))
        );

        fileMenu.setText("File");


importTableMenu.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_I, 
java.awt.event.InputEvent.CTRL_DOWN_MASK));
        importTableMenu.setText("Import table");
        importTableMenu.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                importTableMenuActionPerformed(evt);
            }
        });
        fileMenu.add(importTableMenu);


exportTableMenu.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_E, 
java.awt.event.InputEvent.CTRL_DOWN_MASK));
        exportTableMenu.setText("Export table");
        exportTableMenu.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                exportTableMenuActionPerformed(evt);
            }
        });
        fileMenu.add(exportTableMenu);


        close.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_C,
java.awt.event.InputEvent.CTRL_DOWN_MASK));
        close.setText("Close");
        close.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                closeActionPerformed(evt);
            }
        });
        fileMenu.add(close);


        menuBar.add(fileMenu);


        helpMenu.setText("Help");


        openMenu.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_O,
java.awt.event.InputEvent.CTRL_DOWN_MASK));
        openMenu.setText("Open Documentation");
        openMenu.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                openMenuActionPerformed(evt);
            }
        });
        helpMenu.add(openMenu);


        menuBar.add(helpMenu);


        setJMenuBar(menuBar);
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(addProduct, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGap(0, 0, 0)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(addProduct, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );

        pack();
    }// </editor-fold>

    private void closeActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        System.exit(0);
    }

    private void searchBrandBTNActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
```

```
int searchBrand = searchBrandCombo.getSelectedIndex();


if (productTable.getRowCount() == 0) {
   JOptionPane.showMessageDialog(this, "Table not imported");
} else {
   switch (searchBrand) {
      case 1:
         search("Samsung");
         break;


      case 2:
         search("One plus");
         break;


      case 3:
         search("Redmi");
         break;


      case 4:
         search("Apple");
         break;
      case 5:
         search("Huawei");
         break;


      default:
         JOptionPane.showMessageDialog(this, "Select a brand to search");
         break;
   }
```

```
        }
    }
    public void search(String findBrand) {
        int count = 0;
        LinkedList resultList = new LinkedList();
        Object[] value = new Object[productTable.getColumnCount()];
        for (int i = 0; i < productTable.getRowCount(); i++) {
            if (productTable.getValueAt(i, 0).toString().equals(findBrand)) {
                for (int j = 0; j < productTable.getColumnCount(); j++) {
                    value[j] = productTable.getValueAt(i, j);
                }
                count++;
                resultList.add(Arrays.deepToString(value) + "\n");
            }
        }
        if (count == 0) {
            JOptionPane.showMessageDialog(this, "Searched Product Not found");
        } else {
            JOptionPane.showMessageDialog(this, resultList);
        }
    }
    private void importBTNActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String filePath = "TableData.csv";
        if (productTable.getRowCount() == 0) {
            File file = new File(filePath);
            try {
                FileReader fileReader = new FileReader(file);
                BufferedReader bufferedReader = new BufferedReader(fileReader);
                DefaultTableModel dt = (DefaultTableModel) productTable.getModel();

                Object[] lines = bufferedReader.lines().toArray();

                for (int i = 0; i < lines.length; i++) {

                    String[] eachData = lines[i].toString().split(",");

                    dt.addRow(eachData);

                }

                JOptionPane.showMessageDialog(this,"Import Successful");

            } catch (FileNotFoundException ex) {

java.util.logging.Logger.getLogger(MobileInfo.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);

            }

        } else {

            JOptionPane.showMessageDialog(this, "Table already imported");

        }

    }


    private void addBTNActionPerformed(java.awt.event.ActionEvent evt) {

        int colCount = productTable.getColumnCount();


       if (productTable.getRowCount() != 0) {

            if (brandCombobox.getSelectedIndex() != 0 || !modelNameText.getText().trim().isEmpty() ||
!modelIdText.getText().trim().isEmpty()

                    || !priceText.getText().trim().isEmpty() || colourButton.getSelection() != null) {

                if (!getBrand().equals("0") && getmodelID() != 0 && !getmodelName().equals("0") &&
getprice() != 0 && !getcolour().equals("0")) {

                    int[] result = new int[productTable.getRowCount()];

                    for (int i = 0; i < productTable.getRowCount(); i++) {

                        int v = Integer.parseInt(productTable.getValueAt(i, 2).toString());
                        result[i] = v;
                    }
                    MergeSort.sort(result);
                    if (BinarySearch.binarySearch(result, 0, result.length - 1,
Integer.parseInt(modelIdText.getText().trim())) == -1) {
                        Object data[] = {getBrand(), getmodelName(), getmodelID(), getprice(), getcolour()};
                        DefaultTableModel dt = (DefaultTableModel) productTable.getModel();
                        dt.addRow(data);
                        JOptionPane.showMessageDialog(this, "Product Added");
                    } else {
                        JOptionPane.showMessageDialog(this, "Model ID Already Exists");
                    }
                }
            } else {
                JOptionPane.showMessageDialog(this, "Please Enter all the value", "Add product Failed",
JOptionPane.ERROR_MESSAGE);
            }
        }else{
            JOptionPane.showMessageDialog(this,"Import the table first");
        }


    }


    private void clearBTNActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        brandCombobox.setSelectedIndex(0);
        modelIdText.setText("");
        modelNameText.setText("");
        priceText.setText("");
```

```
        colourButton.clearSelection();
    }


    private void modelIdTextKeyPressed(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        char p = evt.getKeyChar();
        if (!(Character.isDigit(p))) {
            //8= backspace, 46=., 37=%,127=
            if (evt.getKeyCode() == 8 || evt.getKeyCode() == 46 || evt.getKeyCode() == 37 ||
evt.getKeyCode() == 127) {
                modelIdText.setEditable(true);
            } else {
                modelIdText.setEditable(false);
            }
        }


    }


    private void modelIdTextKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        modelIdText.setEditable(true);
    }


    private void priceTextKeyPressed(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        char p = evt.getKeyChar();
        if (!(Character.isDigit(p))) {
            //8= backspace, 46=., 37=%,127=
            if (evt.getKeyCode() == 8 || evt.getKeyCode() == 46 || evt.getKeyCode() == 37 ||
evt.getKeyCode() == 127) {
                priceText.setEditable(true);
            } else {
                priceText.setEditable(false);
            }
        }

    }

    private void priceTextKeyReleased(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        priceText.setEditable(true);
    }

    private void searchPriceTextKeyPressed(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        char p = evt.getKeyChar();
        if (!(Character.isDigit(p))) {
            //8= backspace, 46=., 37=%,127=
            if (evt.getKeyCode() == 8 || evt.getKeyCode() == 46 || evt.getKeyCode() == 37 ||
evt.getKeyCode() == 127) {
                searchPriceText.setEditable(true);
            } else {
                searchPriceText.setEditable(false);
            }
        }

    }
```

```
private void searchPriceTextKeyReleased(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    searchPriceText.setEditable(true);
}

private void exportBTNActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String filePath = "TableData.csv";
    File file = new File(filePath);
    try {
        FileWriter fileWriter = new FileWriter(file);
        BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);

        for (int i = 0; i < productTable.getRowCount(); i++) {
            for (int j = 0; j < productTable.getColumnCount(); j++) {
                bufferedWriter.write(productTable.getValueAt(i, j) + ",");
            }
            bufferedWriter.newLine();
        }
        bufferedWriter.close();
        fileWriter.close();
        JOptionPane.showMessageDialog(this, "Exported Successfully");
    } catch (IOException ex) {
        java.util.logging.Logger.getLogger(MobileInfo.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        JOptionPane.showMessageDialog(this, "File not Found", "Export Failed",
JOptionPane.ERROR_MESSAGE);
    }
```

```
  }


  private void searchbyPriceBTNActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int[] result = new int[productTable.getRowCount()];


    Object[] value = new Object[productTable.getColumnCount()];
    if (productTable.getRowCount() != 0) {
      if (!searchPriceText.getText().isEmpty()) {
        for (int i = 0; i < productTable.getRowCount(); i++) {
          int v = Integer.parseInt(productTable.getValueAt(i, 3).toString());
          result[i] = v;
        }
        MergeSort.sort(result);
        int index = BinarySearch.binarySearch(result, 0, result.length - 1,
Integer.parseInt(searchPriceText.getText().trim()));


        if (index == -1) {
          JOptionPane.showMessageDialog(this, "Product Not Found", "Search Failed",
JOptionPane.ERROR_MESSAGE);
        } else {
          for (int i = 0; i < productTable.getRowCount(); i++) {
            if (Integer.parseInt(searchPriceText.getText().trim()) ==
Integer.parseInt(productTable.getValueAt(i, 3).toString())) {
              for (int j = 0; j < productTable.getColumnCount(); j++) {
                value[j] = productTable.getValueAt(i, j);
              }
            }
          }
          JOptionPane.showMessageDialog(this, Arrays.deepToString(value));
```

```
            }
        } else {
            JOptionPane.showMessageDialog(this, "Enter a valid price");
        }
    } else {
        JOptionPane.showMessageDialog(this, "Import the table first");
    }
}

private void deleteBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    DefaultTableModel dt = (DefaultTableModel) productTable.getModel();
    if (productTable.getRowCount() != 0) {
        if (productTable.getSelectedRow() >= 0) {
            dt.removeRow(productTable.getSelectedRow());
            JOptionPane.showMessageDialog(this, "Row Deleted");
        } else {
            JOptionPane.showMessageDialog(this, "select a row to delete");
        }
    } else {
        JOptionPane.showMessageDialog(this, "Please Import the table first");
    }
}

private void importTableMenuActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String filePath = "TableData.csv";
    if (productTable.getRowCount() == 0) {
        File file = new File(filePath);
        try {
            FileReader fileReader = new FileReader(file);
            BufferedReader bufferedReader = new BufferedReader(fileReader);

            DefaultTableModel dt = (DefaultTableModel) productTable.getModel();
            Object[] lines = bufferedReader.lines().toArray();
            for (int i = 0; i < lines.length; i++) {
                String[] eachData = lines[i].toString().split(",");
                dt.addRow(eachData);
            }
        } catch (FileNotFoundException ex) {
java.util.logging.Logger.getLogger(MobileInfo.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
        }
      } else {
        JOptionPane.showMessageDialog(this, "Table already imported");
      }
    }

    private void exportTableMenuActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      String filePath = "TableData.csv";
      File file = new File(filePath);
      try {
        FileWriter fileWriter = new FileWriter(file);
        BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);

        for (int i = 0; i < productTable.getRowCount(); i++) {
            for (int j = 0; j < productTable.getColumnCount(); j++) {
                bufferedWriter.write(productTable.getValueAt(i, j) + ",");
            }
            bufferedWriter.newLine();
        }
        bufferedWriter.close();
        fileWriter.close();
        JOptionPane.showMessageDialog(this, "Exported Successfully");
    } catch (IOException ex) {
        java.util.logging.Logger.getLogger(MobileInfo.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        JOptionPane.showMessageDialog(this, "File not Found", "Export Failed",
JOptionPane.ERROR_MESSAGE);
    }
}


private void openMenuActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:


    try {
        File file = new File("documentation.pdf");
        Desktop.getDesktop().open(file);
    } catch (FileNotFoundException e) {
        JOptionPane.showMessageDialog(this, "No file");
    } catch (IOException ex) {
        JOptionPane.showMessageDialog(this, "File not found");
    }
}
public String getBrand() {
    if (brandCombobox.getSelectedIndex() == 0) {
        JOptionPane.showMessageDialog(this, "Please Select a valid category", "Add Product Failed",
JOptionPane.ERROR_MESSAGE);
        return "0";
    } else {
        return brandCombobox.getSelectedItem().toString();
    }
}

public String getmodelName() {
    if (modelNameText.getText().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Please enter Model NameText", "Add Product Failed",
JOptionPane.ERROR_MESSAGE);
        return "0";
    } else {
        return modelNameText.getText().trim();
    }
}

public int getmodelID() {
    if (modelIdText.getText().trim().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Please Enter a valid model id", "Add Product Failed",
JOptionPane.ERROR_MESSAGE);
        return 0;
    } else {
        return Integer.parseInt(modelIdText.getText().trim());
    }
}

public int getprice() {
    if (priceText.getText().trim().isEmpty()) {
        JOptionPane.showMessageDialog(this, "Please Enter a valid price", "Add Product Failed",
JOptionPane.ERROR_MESSAGE);
        return 0;
    } else {
        return Integer.parseInt(priceText.getText().trim());
    }
}

public String getcolour() {
    if (colourButton.getSelection() == null) {
        JOptionPane.showMessageDialog(this, "Please Select a Color", "Add Product Failed",
JOptionPane.ERROR_MESSAGE);
        return "0";
    } else {
        return colourButton.getSelection().getActionCommand();
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Windows".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;

                }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(MobileInfo.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(MobileInfo.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(MobileInfo.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(MobileInfo.class
                    .getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
                new MobileInfo().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton addBTN;
private javax.swing.JPanel addProduct;
private javax.swing.JRadioButton blackButton;
private javax.swing.JRadioButton blueButton;
private javax.swing.JLabel brand;
private javax.swing.JComboBox<String> brandCombobox;
private javax.swing.JButton clearBTN;
private javax.swing.JMenuItem close;
private javax.swing.JLabel colour;
private javax.swing.ButtonGroup colourButton;
private javax.swing.JButton deleteBtn;
private javax.swing.JButton exportBTN;
private javax.swing.JMenuItem exportTableMenu;
private javax.swing.JMenu fileMenu;
private javax.swing.JMenu helpMenu;
private javax.swing.JButton importBTN;
private javax.swing.JMenuItem importTableMenu;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel7;
private javax.swing.JPanel jPanel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JMenuBar menuBar;
private javax.swing.JLabel modelId;
private javax.swing.JTextField modelIdText;
private javax.swing.JLabel modelName;
private javax.swing.JTextField modelNameText;
private javax.swing.JMenuItem openMenu;
private javax.swing.JLabel price;
private javax.swing.JTextField priceText;
private javax.swing.JTable productTable;
private javax.swing.JRadioButton redButton;
private javax.swing.JButton searchBrandBTN;
private javax.swing.JComboBox<String> searchBrandCombo;
private javax.swing.JTextField searchPriceText;
private javax.swing.JButton searchbyPriceBTN;
private javax.swing.JRadioButton whiteButton;
// End of variables declaration
}
```

10. Appendix B

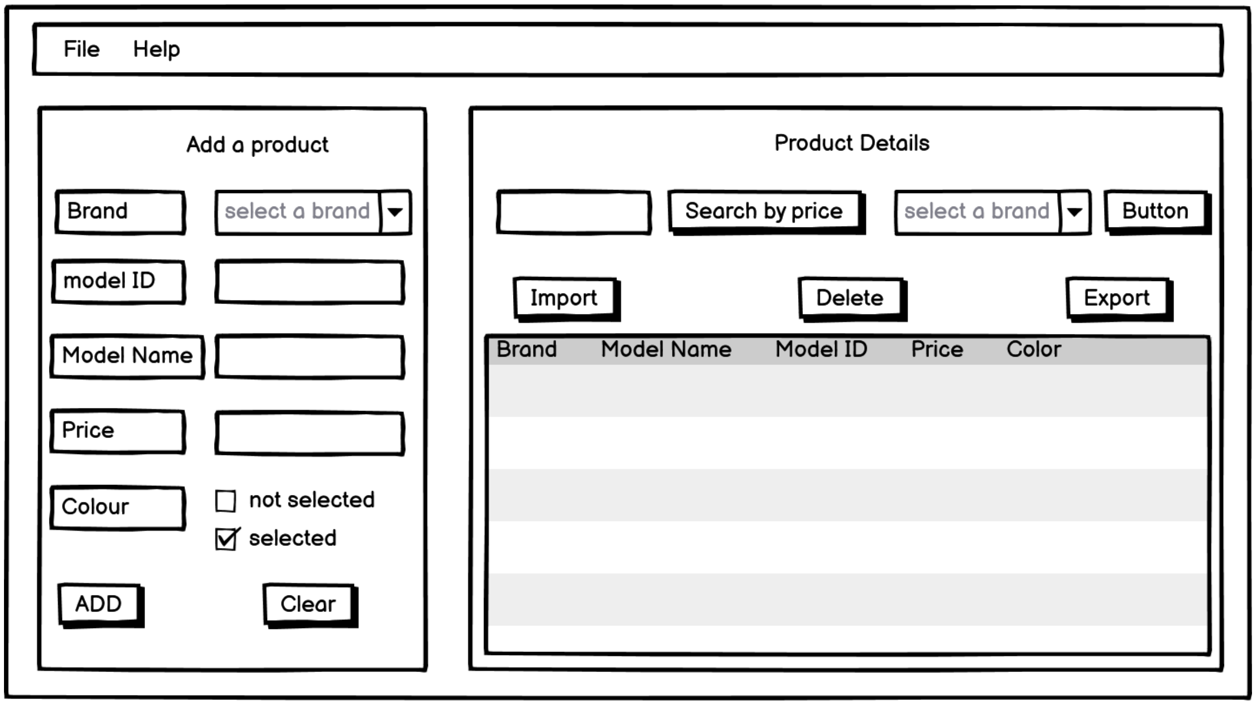


Figure 15: Wireframe