LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS5004NA Emerging Programming Platforms and Technologies

Assessment Weightage & Type

30% Group Coursework

Year and Semester

2019-20 Autumn

Group Name:			
SN	**Student Name**	**College ID**	**University ID**
1	**UTSHA SHRESTHA**	**np01cp4a180259**	**18030040**
2	**SRISHTI ULAK**	**np01cp4a180269**	**18030021**
3	**KSHITIZ MOKTAN TAMANG**	**np01cp4a180200**	**18030035**

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of Contents

Table of Table

Table of figure

Proposal

Information System based on java swing is required to develop. As given in the question “Beverage Information System” is developed where admin will able to input detailed information related to the beverages which includes beverage categories with its volume and total price. The admin is able to search data inside the system based on beverage category and beverage price based on the requirement of customer.

The main purpose of the development of this system is to help the beverage store to show detailed information regarding beverages available to their customer based on their requirement.

List of Data:

Beverage Number: Uniquely identifies each hospital(String)(Text Field)

Brand_Name: Name of Brands of Beverage(String)(Text Field)

Beverage_Type: Type of Beverage alcoholic and non- alcoholic(String)(Radio Button)

Beverage_Category: Different categories of Beverage (String)(Combo box)

Manufacturer_Country: Name of manufacturing country(String)(Text Field)

Volume: Volume of Beverage(String)(Text Field)

Price: Total price of the Beverage(String)(Text Field)

List of Features:

1. Admin can easily get list of information on the basis of category of beverages.
2. Admin can add information about the new data of beverages .
3. The new stored data of beverages information is shown in the tables will be refreshed every time the new data will be added.
4. It is easier to search data as admin can search through multiple attributes such as Price and Beverage_Category.

NetBeans:

In this project, we have used java as a programming language and Apache NetBeans as a Text editor. Apache NetBeans is an open-source integrated Developing environment which support all development of java programming including Java SE. Its frame works for the development of swing desktop application. It is easier to work in Apache NetBeans, as it contains to drag and drop swing items for making GUI of the program. Apache has many plugins which can be easily found and can be used for making our program more efficient. Feature like debugging is very helpful as we can allocate the errors easily (Filehippo.com, 2019).

Balsamiq

Balsamiq is a graphical user interface for building wire frame of the application. Balsamiq allows user to drag and drop pre-built widgets for building their own wireframe. The Balsamiq Mockups user interface is made up of five primary areas: the toolbar, the UI Library, the canvas, the navigator panel and the properties panel. The group of icons in the center of the toolbar is for commonly-performed canvas functions. These are actions that you are probably used to from text editors or other drawing tools, such as copy, paste, group, align and zoom. Hence, due to the above features we have used Balsamiq for designing wire frame to develop our system (balsamiq, 2019).

1. Introduction

This course work is about development of Java based Information System in which a “Beverage System” is developed for beverage store where user can store information regarding beverages that are available in their store.

As given in the requirement of this coursework, a table containing list of data is displayed whenever the system is run. This system contains three different panel containing different GUI components. The first panel contains table, containing list of beverages, the second panel contains user input. Here, the user can add value from user input in which all the detail information of any beverage category is input. The user can also update any row of table if required by selecting the row that need to be updated. This system has functionality to search the items based on the price. When there are items containing same price only the first matching item is displayed in the JOption message box and when there is no such item of that price then a meaningful message is displayed. For the sorting purpose merge sort algorithm is used . This system contains text field for searching by price purpose in which binary search algorithm is implemented. Similarly, this system contains text field for searching by categories of beverages that are water, milk, wine, whiskey, beer and juice in which total number of items of that categories is displayed in JOption message box.

Aims:

The main aim of this course work is to develop an information system for a beverage store in order to store the detail information of beverages.

Objectives:

By the use of this system beverage store owner can have systematic way of displaying the detailed information of beverages. It makes easier to update the data of beverages of any category. The data of beverages can be stored in separate text file by the use of this system which concludes data protection.

2. GUI manual:

This system consists of three different panel:

- List of Beverage

When this button is clicked panel containing list of beverages in table is displayed. The data from row of table can be retrieved to user input when the row of table is selected. In this panel information regarding beverages can be displayed by searching the respective beverage based on its price. The price is entered in the text field and by clicking on search by price button, the dialog box is displayed containing beverage information of that price. If there is beverage of same price then the information of the beverage that is at the top most is displayed instead of other one. Similarly, number of brand with its brand name of same category can be searched by selecting category in combo box and clicking the search by category. A dialog box is displayed containing information regarding the search by category of beverage.

- User input

When this button is clicked, panel containing user input to populate the table is displayed. This panel has two different panel, the bigger one contains user inputs in which input form user is taken to populate the table and the smaller one contains the buttons such as

- add button: To add the user input.
- clear button: To clear the user input field.
- update button: To update data of the selected row of table.
- Delete button: To delete data of the selected row of table.

- Beverage Image:

When this button is clicked, panel containing image of beverages is displayed.

Extra Features:

A Card Layout object is a layout manager for a container, and it treats each component in the container as a card. At one time Only one card is visible at a time, and the container acts as a stack of cards. The first component added to a Card Layout object is the visible component when the container is first displayed. To create the Card Layout in NetBeans the main panel in which to hold other panels for the Card Layout, the several panels within this Main Panel are created and insert code so that when a button or an action is generated, the panels are swapped. By the help of the card layout the different panel can be change by the help of different button. (Febrega, 2018)

3. Individual Work

3.1. Task-1:

Name: Utsha Shrestha

Responsibilities:

i. GUI design:

In GUI card layout was used as card layout arranges the components in sequential manner. All the components of GUI are placed into the queue in order they are added (Raja, 2019). The GUI contains main three panel and panel is changed by clicking the button. All the required components are added in the panel.

Problems faced and problem solving:

During GUI design, there was difficulties in managing the panel but with the proper concentration and understanding GUI design was successfully done.

ii. Sort

For searching the data using binary algorithm data was sorted using binary search

Problem faced and problem:

There was issue in understanding in merge sort but after lots of study this was solved.

iii. Delete :The row of table can be deleted when the row is selected and delete button is clicked.

Problem faced and problem solving:

During deleting the table there was problem in mouse clicked event for radio button and combo box, but after lots of research this problem was solved.

iv. Methods description of the sort, delete, GUI designs, add.

v. Validation for delete and update

vi. Description of Merge sort

3.2. Task-2:

Name: Srishti Ulak

Responsibilities:

i. Search by price:

In search by price binary search algorithm was used.

Problem faced and problem solving:

There was issue in understanding the binary search algorithm but with proper understanding and research this problem was solved.

ii. Search by category:

In search by category linear search algorithm used.

Problem faced and problem:

There was issue in error detection and solved by debugging.

iii. Update:

The selected row can be updated by clicking update button.

Problem faced and problem solving:

During implementation of this, there was issue in solving the error, but with the help of debugging tool this problem was solved.

iv. Clear :

The user input is cleared when clear button is clicked.

Problem faced and problem solving:

There was problem in clearing the radio button, but this was solved by the proper research.

v. Methods description of search by price, search by category, update and clear.

vi. Validation for search by price.

vii. Testing

3.3. Task-3:

Name: Kshitiz Moktan Tamang

Responsibilities:

i. Add:

The data is entered in the user input and added to the table by clicking the add button. The data can be viewed in panel containing table.

Problems faced and problem solving:

During this there was difficulties in understanding about Default table model. But, after proper research this problem was solved.

ii. Save file

The values are added to the table and the values of table will be save in the txt file by clicking in the saving menu.

Problem faced and problem:

Problem while location of file path by research.

iii. Open file:

The value of will be add to the table when the life in open

iv. Help_info;

The user manual will be open when the help file in open from table.

Methods description of save file, open file, help_info.

v. Wireframe for GUI

vi. Validation of add

vii. Description of Binary Search

4. GUI Design:

4.1. Wire frame of GUI

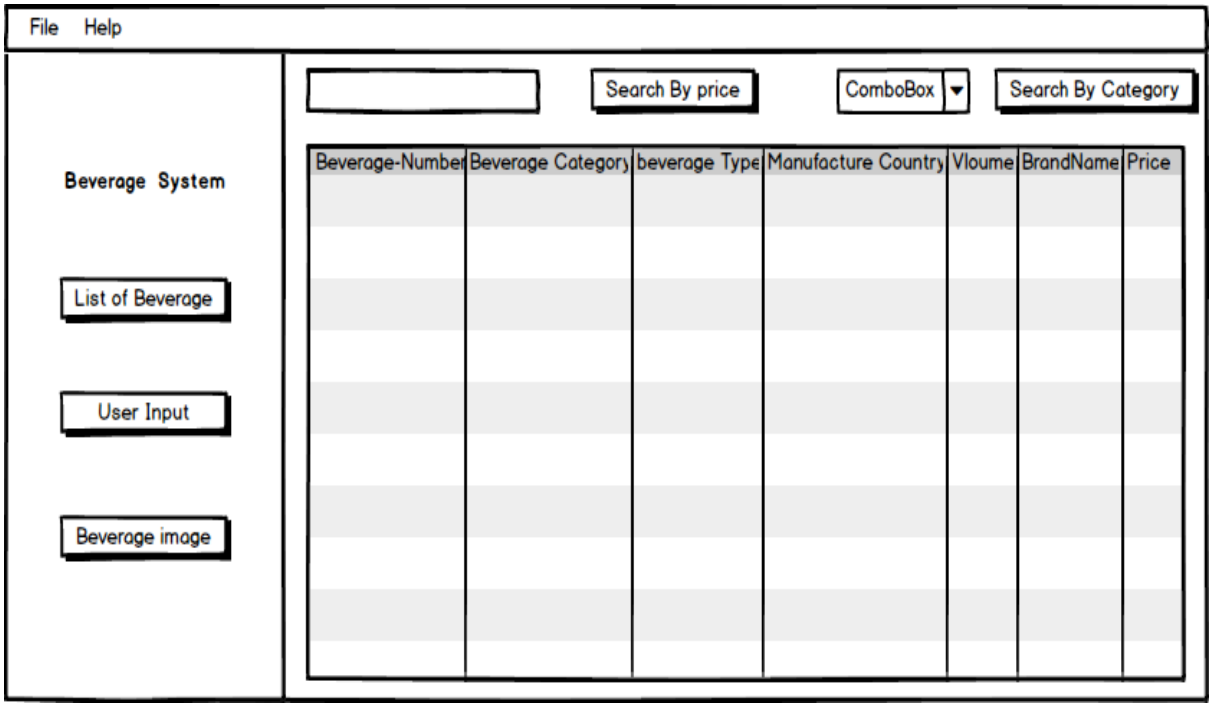


Figure 1: Wire frame of panel 1

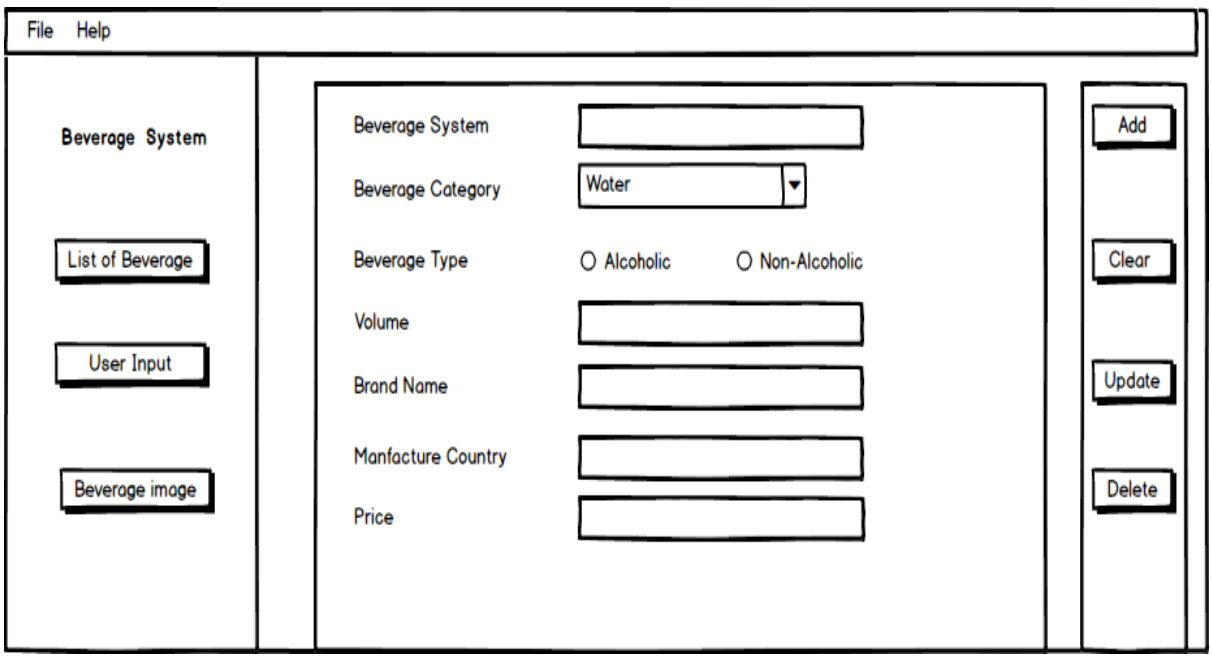


Figure 2: Wire frame of panel 2

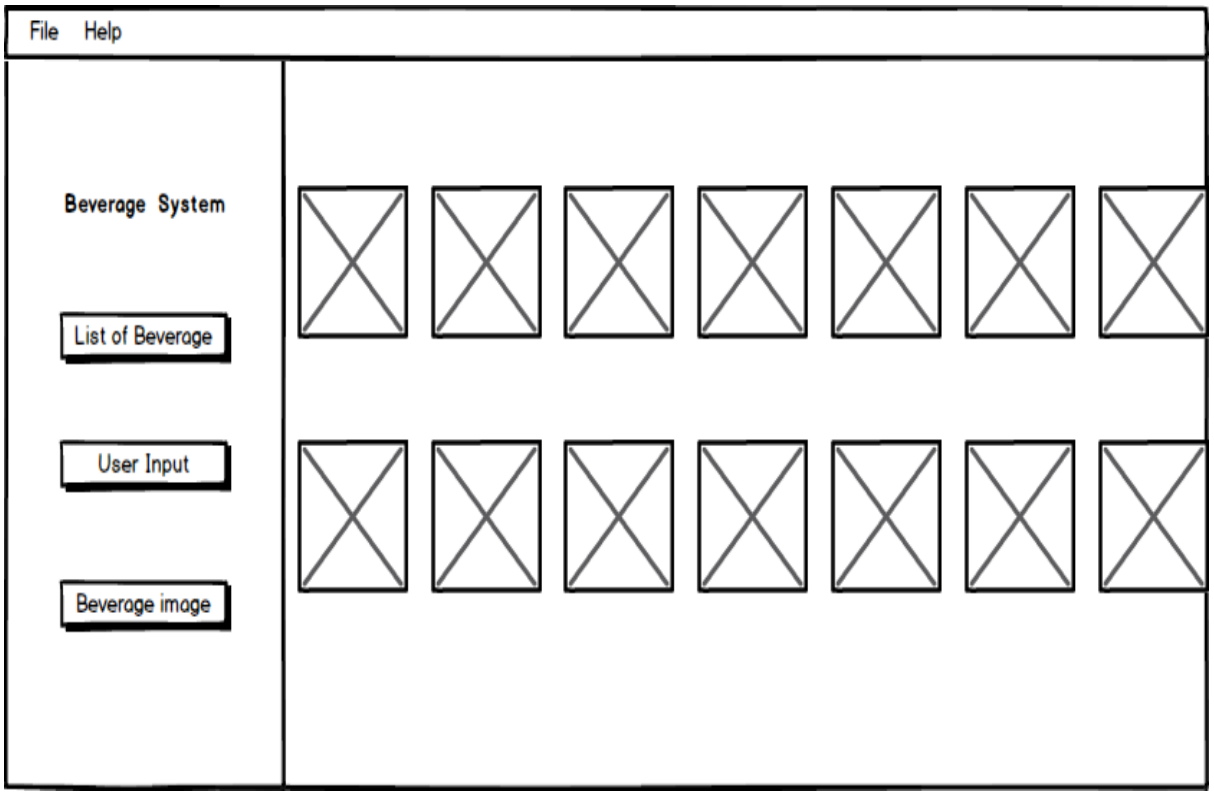


Figure 3: Wire Frame of panel 3

4.2. GUI

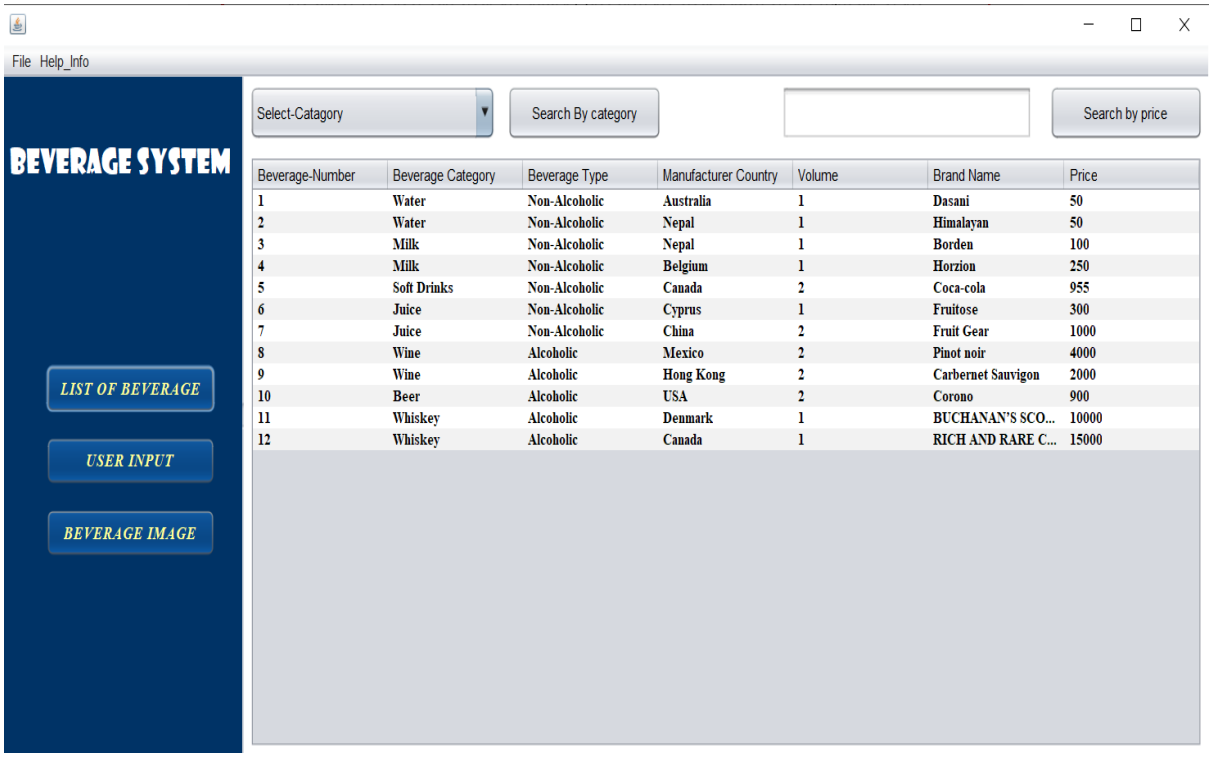


Figure 5: GUI of Panel 1

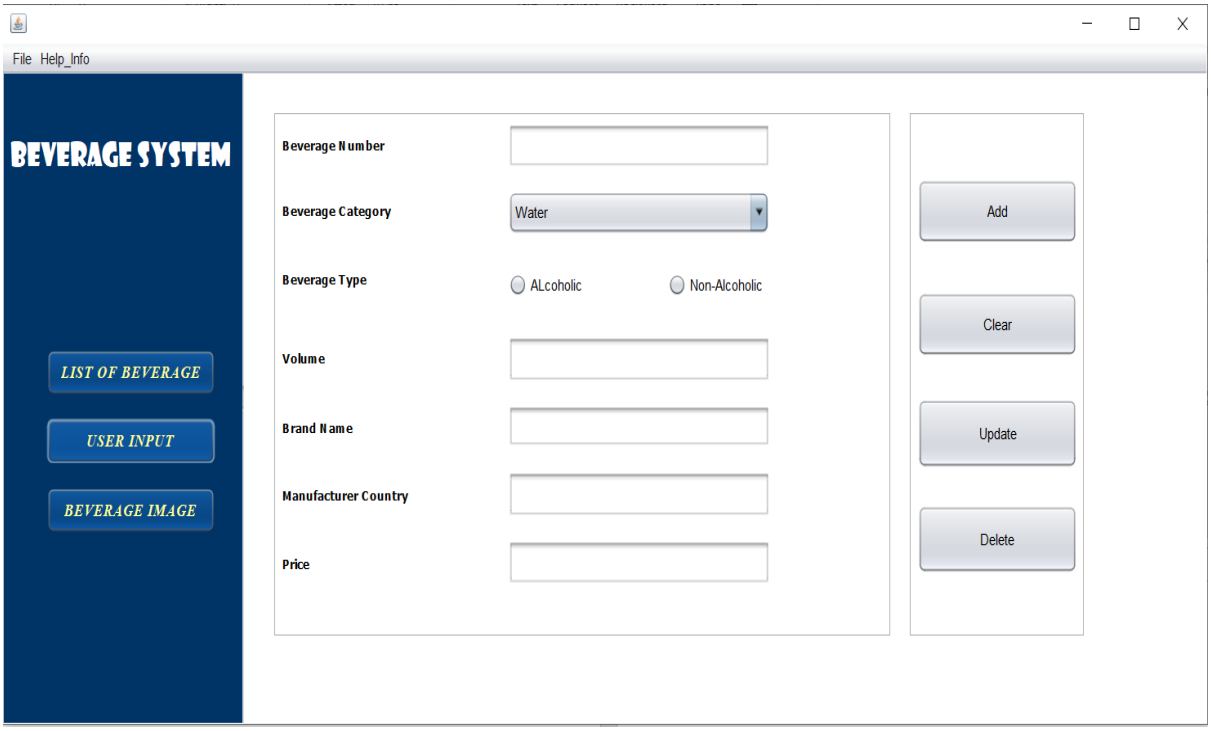


Figure 4: GUI of Panel 2

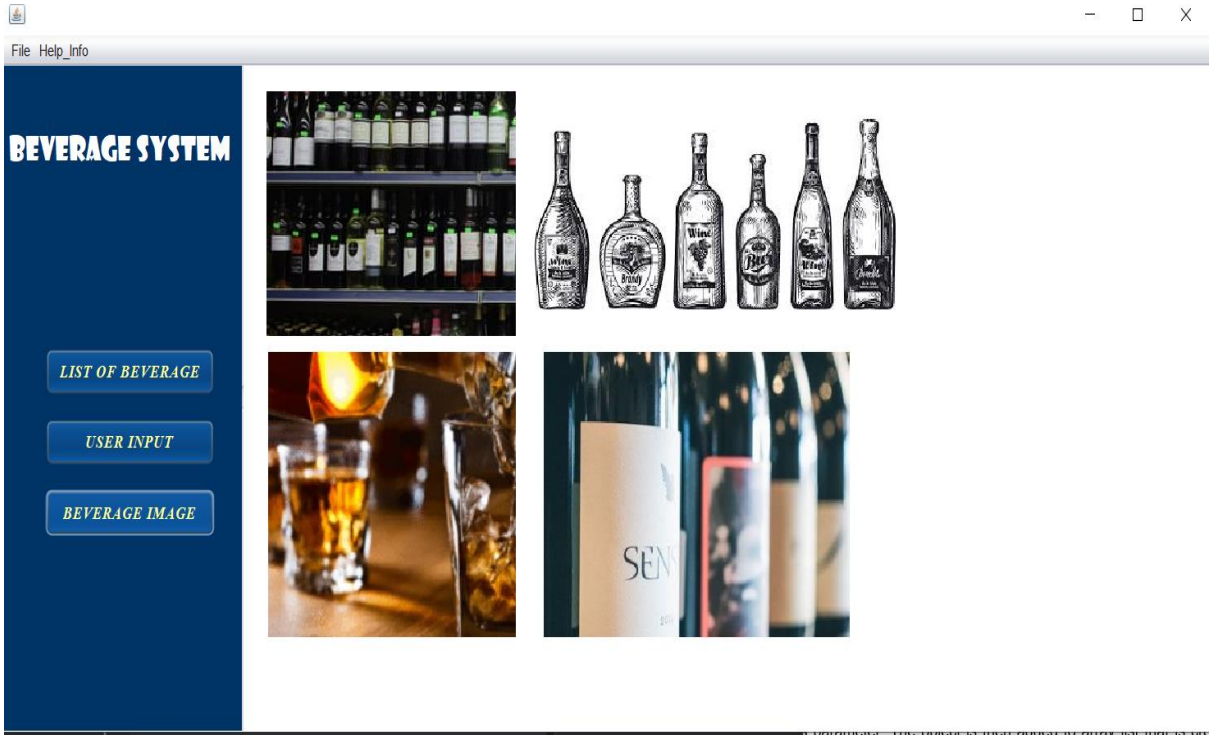


Figure 6: GUI of panel 3

5. Method Description:

i. BeverageList():

It is the method of array list in which new object of the class is created and value is passed in its parameter. The object is then added to array list that is created by importing java.util.ArrayList package using add() method. After adding all the created new object the array list is return. This method is made public so that java run time can execute this method.

ii. ArrayList BeverageList():

This method has void return type to add data in row of the JTable using DefaultTableModel. Array list is declared in the main class. The values of row are added by creating new object. The values are added in then array list. Array list. Arraylist name list is create for drinklist.

iii. Void addRowToJTable():

The array list values are store in the beverage list. The rows data are keeps in the store in the object. By the help of for loop the values of rows are store in their order. DefaultTableModel allows the full range of actions by the table model to be performed on the JTable (e.g, adding a row, inserting a row, removing a row, adding a column, etc.).

iv. void sort(int[] sortArray):

This method is to sort the data of int type array using merge sort. It has public access modifier which means it can be accessed by outside class. It is static non return type method, which means it can be called without creating an instance of a class. The array that contains all the data to be sorted is declared and it is named as sortArray. If the length of the array is less than 1 Then the length of the array is checked whether it is less than 1 or not if this is true array is returned. Then this array is divided into two half and each half is sorted. The first half of array(firstNum) and second half of

array(secondNum) is declared and populated. These method works as recursive method as it calls itself for sorting each half of array.

v. void merge(int[] firstNum, int[] secondNum, int[] sortArray):

This method merges the two sorted arrays in an array. It is private access modifier method. This method is static non return type method in which first half array, second half array and array to be merged are the parameter list. The next elements of first half of array i.e. firstElement, second half of array i.e. secondElement and next open position i.e. num are initialized. Each element of firstNum array and second array is compared and sorted in a order as long as the elements sort array comes to the end.

vi. void int sorting(int sortArray[ ], int y):

It is the method to sort the data in a order using binary search algorithm. The data to be sorted is stored in an array. The start and end element of array is initialized and checked whether the start element is greater than end element . Inside the while loop mid value is determined and checked whether it is equals to the y element. If its value is equal to y element the array is returned and if the y element is greater than the mid element, the value of start element has value of mid element increased by one. Otherwise, the value of end element has value of mid element decreased by one.

vii. Void jButton1ActionPerformed:

This method is used to change layout to the list of beverage from other layout. The Card Layout class manages the components in such a manner that only one component is visible at a time and It also treats each component as a card.

viii. Void jButton4ActionPerformed:

This method is used to change layout to the user input from other layout.

ix. Void jButton5ActionPerformed:

This method is used to change layout to the beverage image from other layout.

x. Void jButtonAddActionPerfrmed:

This is the method to add the data from the user inputs to JTable by bringing add button in action. The values from jTextFields are get using getText() method and stored in a global variable. The value form selected jRadioButton is get using isSelected() method, which is determined from if/else statement. If the radio button for alcoholic beverage type is selected, in btype variable “Alcoholic” value is set otherwise, “Non-Alcoholic” value is set. After getting all the user input value, it is checked whether the fields are empty or not using isEmpty() method. If the fields are empty a JOption dialog box is displayed containing error message otherwise the input values are added to the JTable using getModel() method. To insert row in Jtable, getModel() method is used and kept in DefaultTableModel variable, for this purpose object type casting is done. The data obtained from the user input is stored in the array and using addRow() method and JOption dialog box is displayed containing success message. After the data is successfully added to JTable, all the fields are cleared by using setText() method for text field clearSelection() method for radio button, setSelectedItem() method in combo box.

xi. void jTextFieldBNumKeyPressed(java.awt.event.KeyEvent evt):

In this method of void jTextFieldBNumKeyPressed doesn’t allow the enter string values. Keypress event is used to for the validation of text-field. In this method text-field value will be store in string. The length of the input is check. The length is the Character should be smaller than 10.

xii. void jTextFieldBrandKeyPressed (java.awt.event.KeyEvent evt)

In this method of void jTextFieldBrandKeyPressed only allow the enter string values. This method is the for keypress does not allow to enter character is letter, space and is control char that allow to edit but the iso control for edit operation(delete key and backspace is allowed)

xiii. void jTextFieldcountryKeyPressed(java.awt.event.KeyEvent evt)

In this method of void jTextFieldcountryKeyPressed only allow the enter string values. This method is the for keypress does not allow to enter character is letter, space and is control char that allow to edit but the iso control for edit operation(delete key and backspace is allowed)

xiv. void jTextFieldPriceKeyPressed(java.awt.event.KeyEvent evt)

In this method of void jTextFieldPriceKeyPressed doesn't allow the enter string values. Keypress event is used to for the validation of text-field. In this method text-field value will be store in string. The length of the input is check. The length is the Character should be smaller than 10.

xv. void jButtonUpdateActionPerformed(java.awt.event.ActionEvent evt):

It is the method to bring update button in action that updates the value of jTable row. This method is made public so that its value can be access and its type is void i.e. non-return type. ActionEvent parameter of this method is an event object that represents an event i.e. a button click. This parameter contains the information about the event. The row that is to be updated is clicked that is done using mouse clicked event. Here, data of jTable is get using getModel() method of DefaultTableModel. If the single row is selected the value of all the user input is get using getText() method for text fileds and isSelected() method for radio button. Then, all the updated value is set in jTable row using setValueAt() method. In parameter of setValueAt() method the variable containing value of user input, getSelectedRow() method(to get value from jtable) and index of column of jtable, then a JOption message box containing success message displayed. If the no row is selected error message is displayed. Then all the text fields are cleared using setText() method passing the empty string as its parameter value and radio buttons are cleared using clearSelection() method and combo box are cleared using setSelectionItem() method in which value of first item is set.

xvi. void jTableBeverageMouseClicked(java.awt.event.MouseEvent evt)

This method is for event of mouse click in a component. getModel() method is used to get the value from jTable. The data is set to text field when a row is selected. All the data form the row of table is get using getValueAt() method and stored in the respective String variables. Then these data are set in the user input using setText() for text fields in which respective variable are passed as a parameter value. Similarly, the

xvii. void jButtonClearActionPerformed(java.awt.event.ActionEvent evt):

This method is to bring the clear button in action that clears all text fields using setText() method having empty parameter. By the help of getSelctedRowToDelete the row index is selected, and the index values should be greater than zero. TableModel.removeRow method is used to remove that particular row form the table. The text-fields, combo-box and radio button are made clear. The Exception e is used to catch any error in the program.

xviii. void jButtonDeleteActionPerformed(java.awt.event.ActionEvent evt)

This method is used to delete the row form the table. DefaultTableModel allows the full range of actions by the table model to be performed on the JTable (e.g, adding a row, inserting a row, removing a row, adding a column, etc.).

xix. void jButtonPSearchActionPerformed(java.awt.event.ActionEvent evt)

By this help of this method the search method is run with price value. The row values is get by the help of the getRowCount method. The price is in the 6^{th} index. From the jTextFieldSPrice of price the search value is get. If the value of the search price is equal the dialog box is show the information of that product. The values of row is 0 then the value field is empty. If the textfield is not empty then the string value of the price is converted into integer. The integer value is check if value lies in the table value. If the values of the text field is not same as the table value then the message will be shown.

xx. jButtonCsearchActionPerformed(java.awt.event.ActionEvent evt)

By this help of this method the search method is done by the help of drop-down box. The value of category is store in string. DefaultTableModel allows the full range of actions by the table model to be performed on the JTable (e.g, adding a row, inserting a row, removing a row, adding a column, etc.). If the category is not selected the JOptionPane will display message. If the category is selected then the information will be display of that category.

xxi. void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt)

In this method the help file is open. Runtime.getRuntime() method returns the runtime object associated with the current Java application. The methods of class Runtime are instance methods and must be invoked with respect to the current runtime object. File name is store in string form. The exception handling is used to give path to the for the file.

xxii. void main(String args[ ])

It is the main method in which GUI is run. This method is made public so that java runtime can execute this method. It is the static method and accepts single argument of String type array.

xxiii. void SaveActionPerformed(java.awt.event.ActionEvent evt)

In this method the saved file is open where the object of both file writer and buffered writer are made. The path for the save file is given in the file writer object whereas the buffered writer object creates a buffered character output which uses the default size for an output buffer. The exception handling is used to give path to the for the file.

xxiv. void jMenuItem2ActionPerformed (java.awt.event.ActionEvent evt)

In this method we use exit method to exit the application.

xxv. void jMenuItem3ActionPerformed (java.awt.event.ActionEvent evt)

In this method the file which need to be open is done. The object of buffered reader class is made which read the text from an input stream by buffering characters that read character from array or lines. The expectation handling is used to give path to the for the file

6. Merge sort

Merge sort approach is the methodology which uses recursion mechanism and works on the principle of Divide and Conquer algorithm. Merge Sort divides input array in two halves, calls itself for the two halves and then merges the two sorted halves. Merge() function is used to divide the array into two equal half. This step is carried out recursively for all the half arrays until there are no more half arrays to divide. In conquer step we sort and merge the divided arrays from bottom to top and get sorted array (Landup, 2019).

MergeSort(arr[], s, r)

If r>s

First middle point of array is divided into two halves.

middle m = (s+r)/2

MergeSort for first half part is called.

Call mergeSort(arr, s, m)

MergeSort for second half part is called.

Call mergeSort(arr, m+1, r)

First half and second half of Merge the two halves sorted in step 2 and 3:

Call merge(arr, s, m, r)

Conceptually, merge sort works as follows in recursive fashion:

- Divide the unsorted list into two array of about half the size
- Sort each of the two array
- Merge the two sorted array back into one sorted list (HackerEarth, 2020)

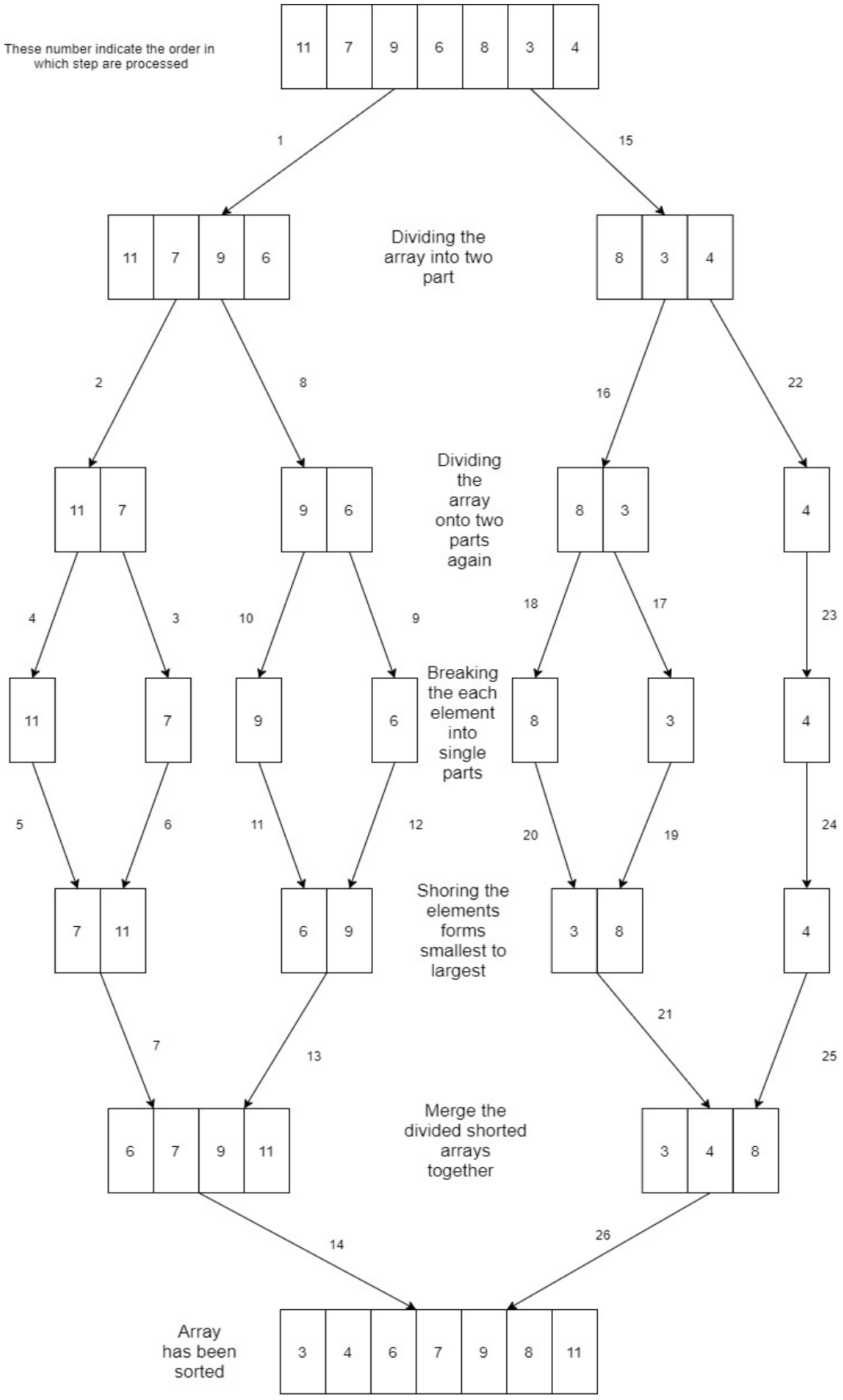


Figure 7: Merge Sort Algorithm

7. Binary Search:

Binary search works on the principle of divide and conquer, data collection should be in sorted sequence either in ascending or in descending order. In binary search the middle element of the list with the selected element will be compared. If the selected values matched with the middle element then its position in the list is returned. If the selected values doesn't match with middle element, then the list will be divided into two part. First half consists of the first element to middle element whereas the second half consists of the element next to the middle element to the last element. (Williams, 2018)

When the binary search operation is performed in the sorted set, the number of iterations can always be reduced based on the value that is being searched.

Following are the steps of implementation for binary search algorithm :

- ✓ Compare the middle element if the target value is equal to the middle element of the array, then return the index of the middle element.
- ✓ If target value matches with middle element, we return the mid index otherwise target value is greater than the mid element.
- ✓ If the target value is greater than the number in the middle index, then elements is picked to the right of the middle index. If the target value is less than the number in the middle index, then the elements is picked to the left of the middle index (Lewis, 2018).

When a target match is found then it will return the index of the element matched. If not then it will return in -1.

Let us consider the following array Sorted :

Array A

0	1	2	3	4	5	6	7	8	9

Before starting the search, the start and end of the range of given array should be known. Let us consider them as Low and High,

Low = 0

Hight = n-1

Now, the search value A is compared with the element situated at the median of the lower and Upper bounds. If the value A is greater than the lower bound is increased, else the upper bounds is decreased.

Left = 0 right = 9

Array A

0	1	2	3	4	5	6	7	8	9

Item= 3 mid=4

A[mid] = 4

From the above table, the lower bound is 0 and the upper bound is 8. The median is found by the formula (x+y)/2 where x= lower bounds and y= upper bounds. Here the median value is a [4] = 4. The value 4>2, which is the value being searched. The elements beyond 4 doesn't need conduct a search as the element will be greater than 2 obviously.

Therefore, the upper bound of the array is drop to the position of the element Now, following the same procedure on the same array with the values:

Low =0

High =3

This procedure is repeated recursively until low> high. If at any iteration, we get a[mid] = key, value of mid is return which is the position of key in the array. If the key is not present in the array, the value -1 is return. (HackerEarth, 2020)

Implementation:

```
int binary Search (int low, int high, int key)
{
  while(low<=high)
  {
    int mid=(low + high)/2;
    if(a[mid]<key)
    { low=mid+1;   }
    else if(a[mid]>key){
        high=mid-1;  }
    else
    {     return mid; }
  }
  return -1;
 }
```

The Recursive Code of Binary Search

```
int binarySearch(double b, double X[], int left, int right)
{if (left == right)
if (b==X[left]) return left;
else return -1;
int mid = (left+right)/2;
if (b==X[mid]) return mid;
```

if (b < X[mid]) return binarySearch (b, X, left, mid-1);

if (b > X[mid]) return binarySearch(b, X, mid+1, right);} (Williams, 2018)

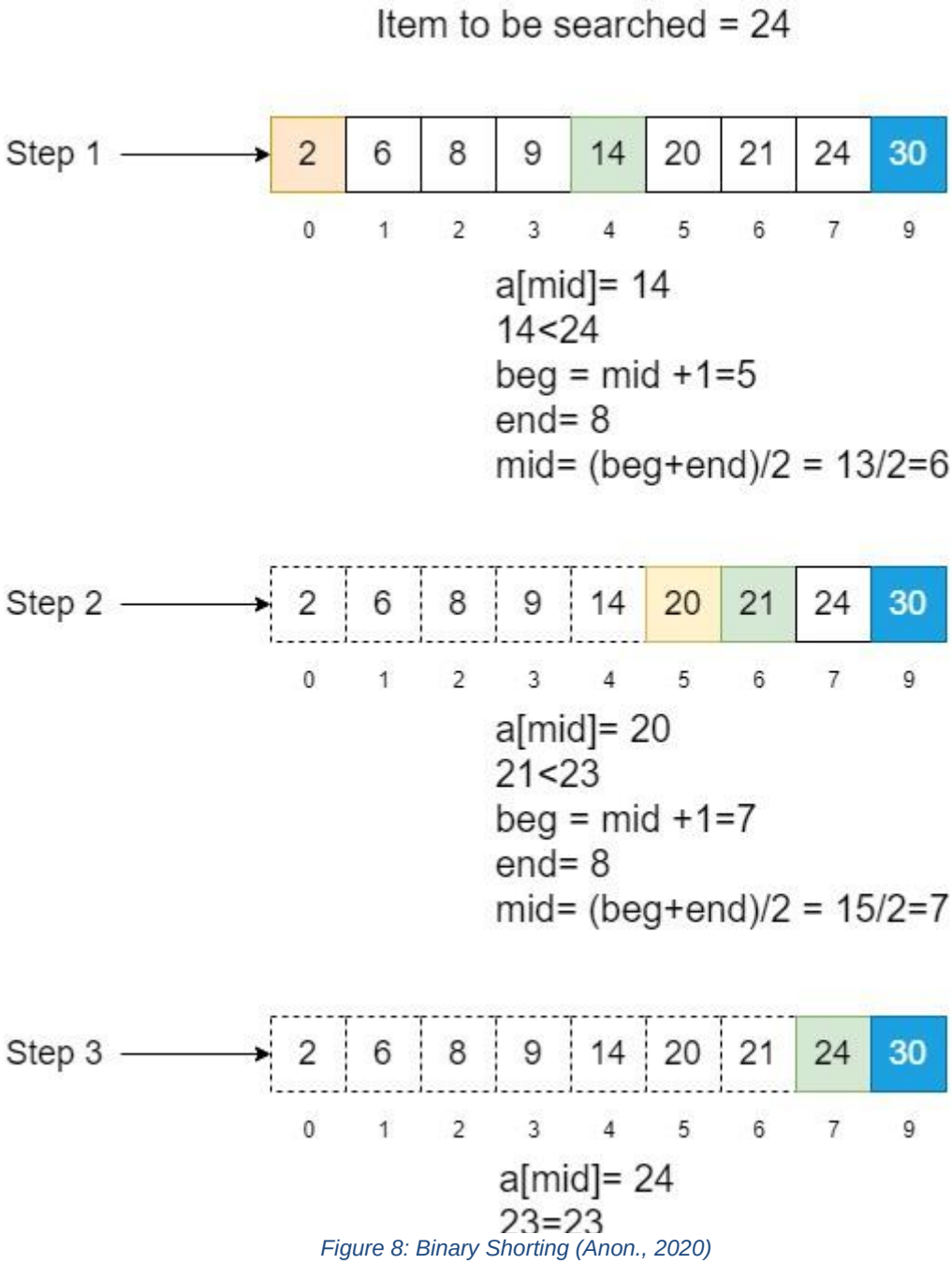


Figure 8: Binary Shorting (Anon., 2020)

8. Testing:

a. Run GUI successfully

Objective	To run GUI successfully
Action	1. Run GUI
Expected Result	To open GUI
Actual Result	To open GUI
Conclusion	Test successful.

Table 1: Run GUI successfully

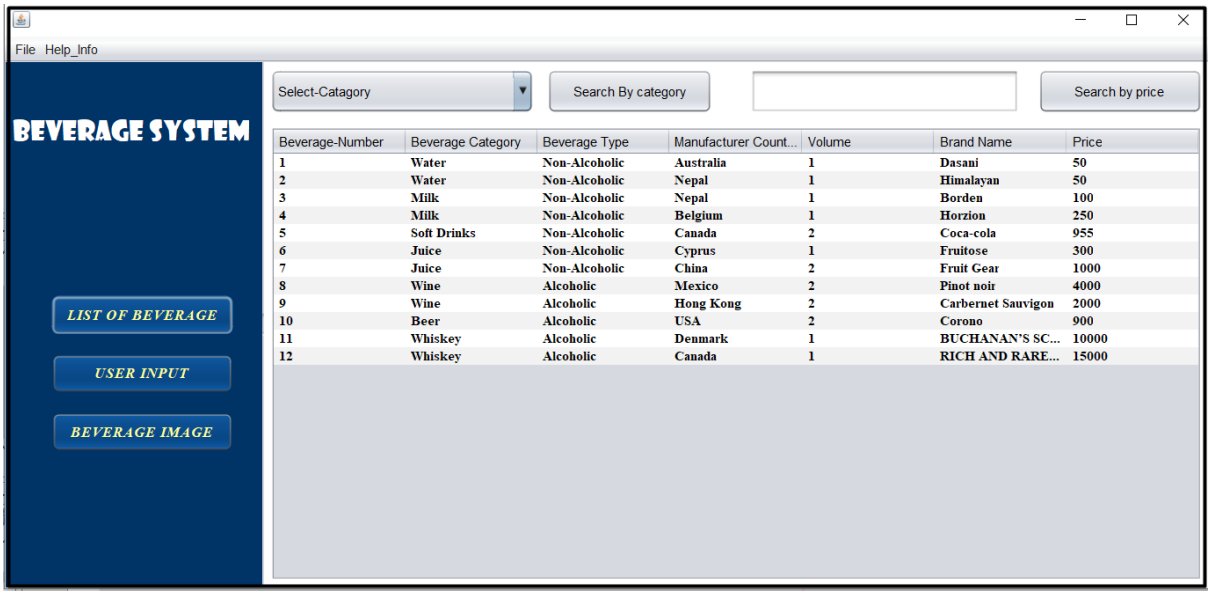


Figure 9: Run GUI successfully

b. Restriction of Character

Objective	To restrict character in Number input.
Action	2. To restrict the alphabet 3. Keypress event is used to check character
Expected Result	To restrict the character from text-filed
Actual Result	To restrict the character from text-filed
Conclusion	Test successful.

Table 2: Restriction of Character

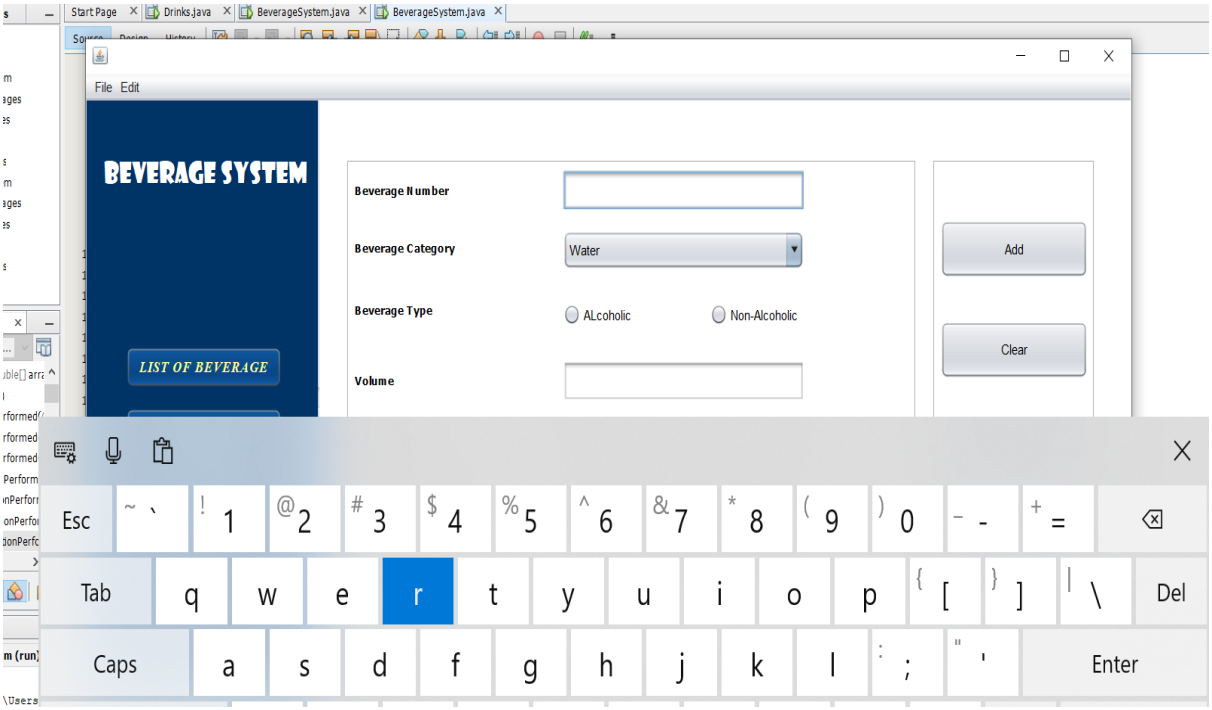


Figure 10: Restrict character

c. Restriction of Number

Objective	To restrict number in character input.
Action	1. To restrict the number 2. Keypress event is used to check character
Expected Result	To restrict the number from text-filed
Actual Result	To restrict the number from text-filed
Conclusion	Test successful.

Table 3: Restriction of Number

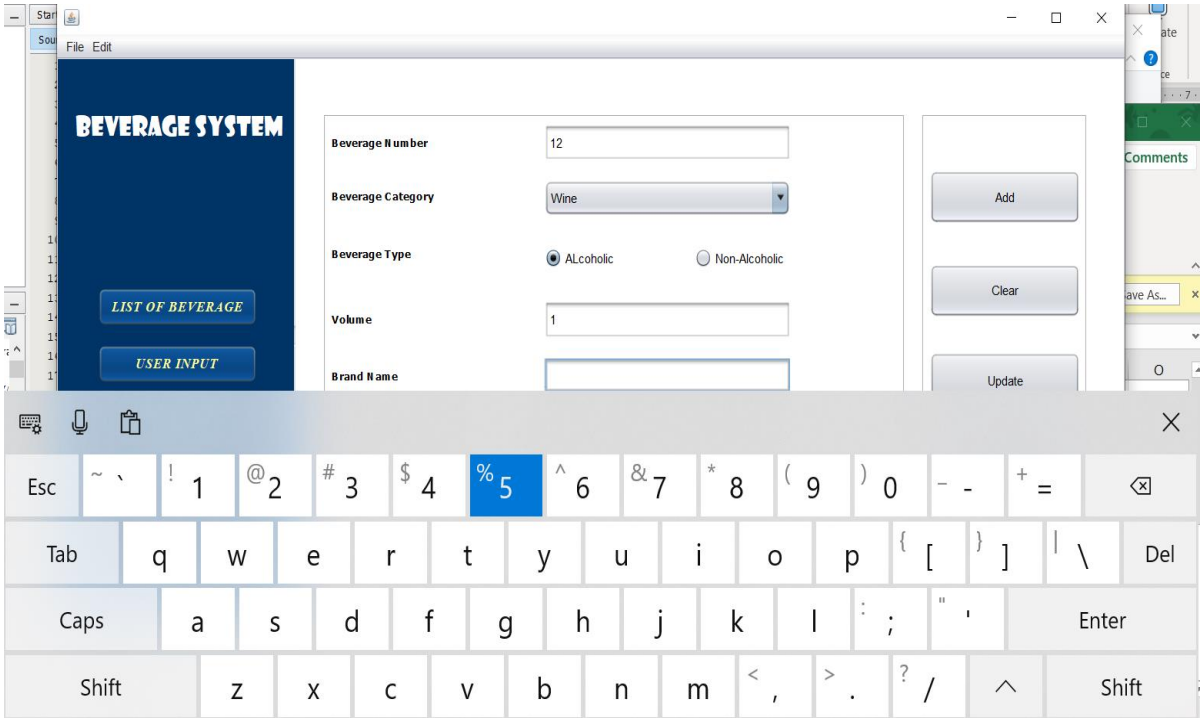


Figure 11: Restrict number in character input.

d. Checking empty Text-field

Objective	To check table is empty
Action	To add without entering value.
Expected Result	To display dialog box
Actual Result	To display dialog box in with message of empty fields.
Conclusion	Test successful.

Table 4: Checking empty Text-field

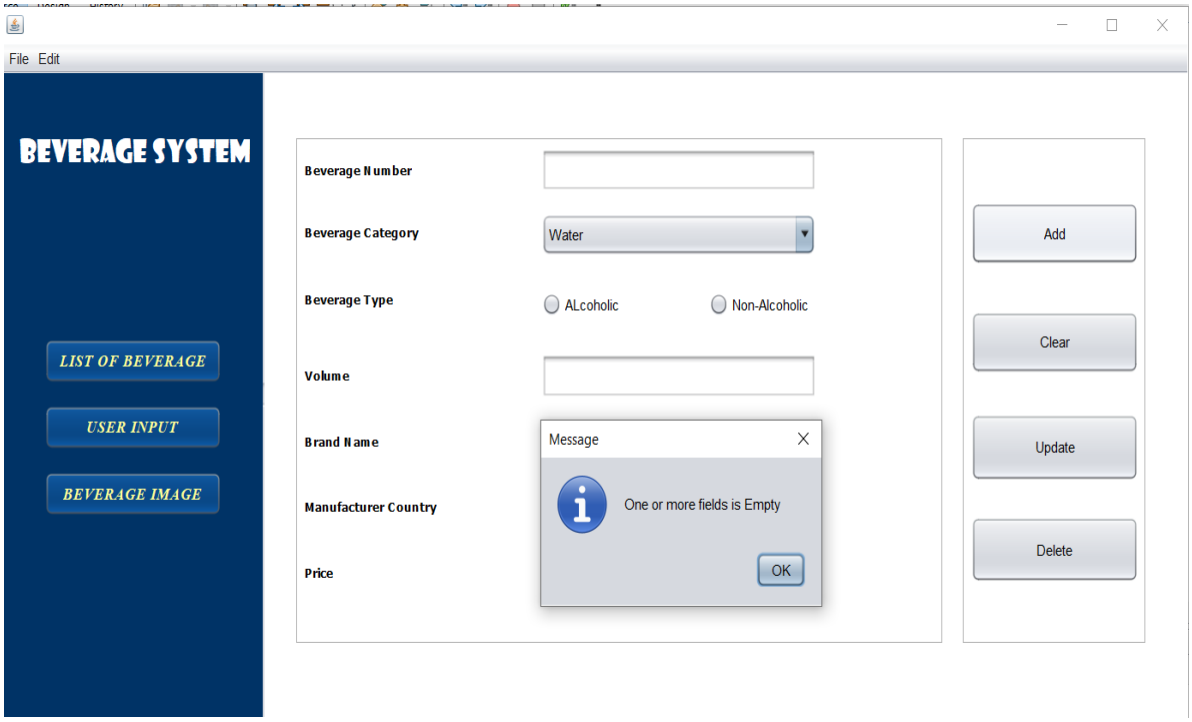


Figure 12: Checking empty text field

e. Adding values in table

Objective	To add value in text field
Action	1. Inserting values in field 2. Adding values in table
Expected Result	Adding values to the table
Actual Result	Add value in table and displaying dialog box.
Conclusion	Test successful.

Table 5: Adding values in table

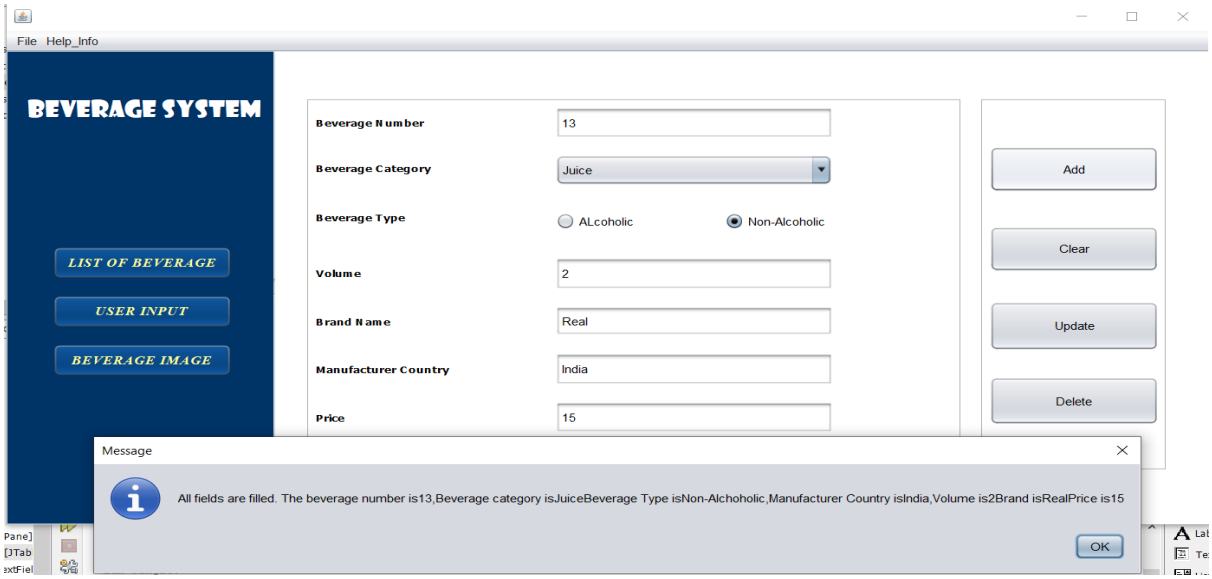


Figure 13: Inserting value in table

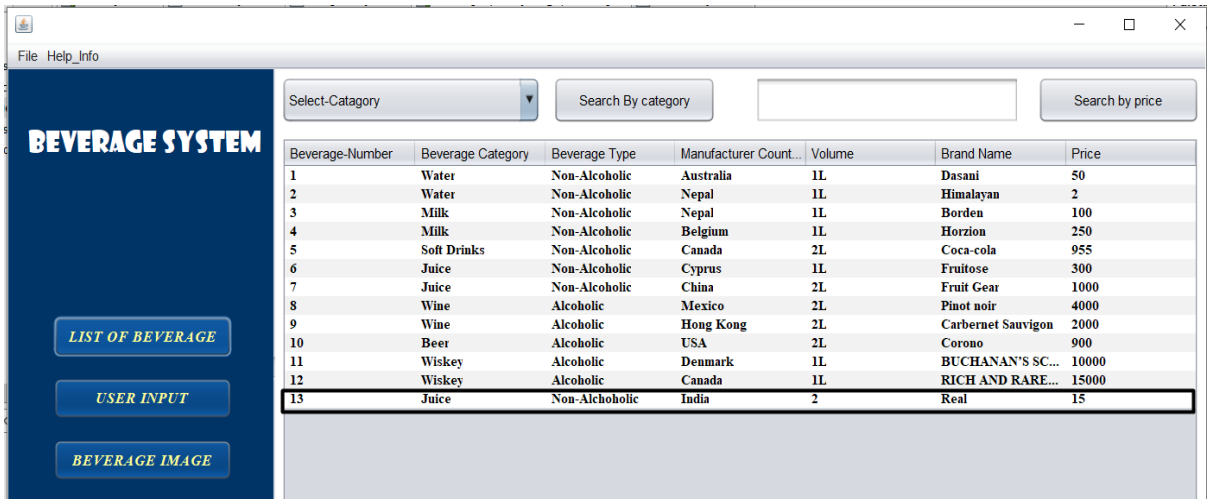


Figure 14: Adding values into table

f. Searching by category:

Objective	To search by category.
Action	1. Selecting category to search its data. 2. Clicking search by category button.
Expected Result	Searching by category
Actual Result	Searching by category
Conclusion	Test successful.

Table 6: Searching by category

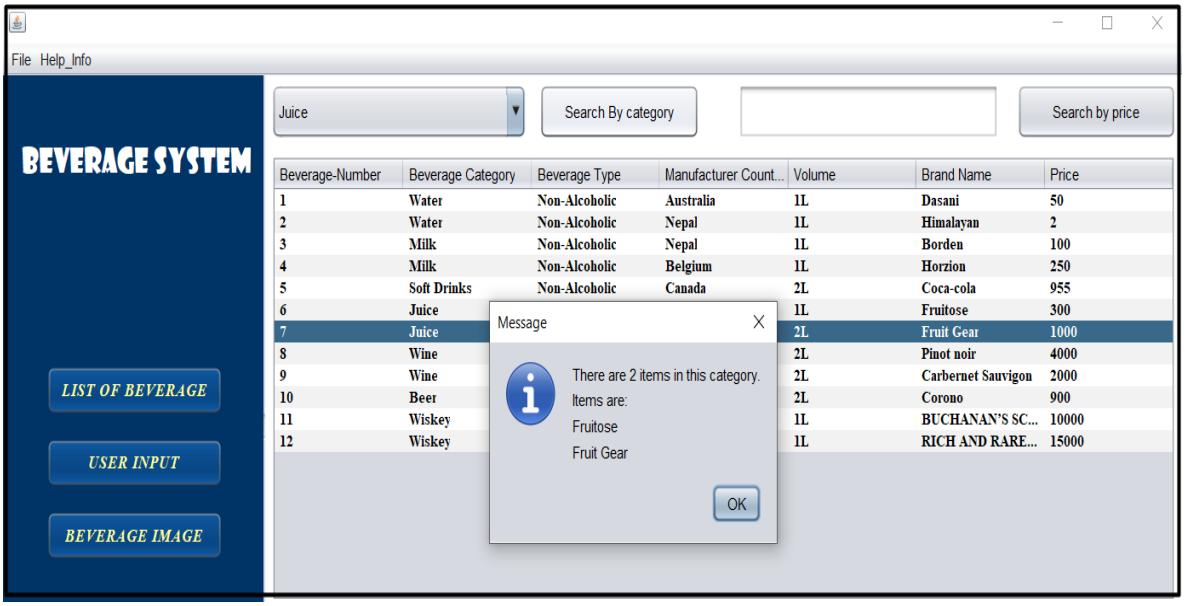


Figure 15: Searching by category

g. Searching by Price:

Objective	To search by price
Action	1. Searching by price
Expected Result	Searching by price and searching result comes in dialog-box
Actual Result	Searching by price and searching result comes in dialog-box
Conclusion	Test successful.

Table 7:Searching by Number

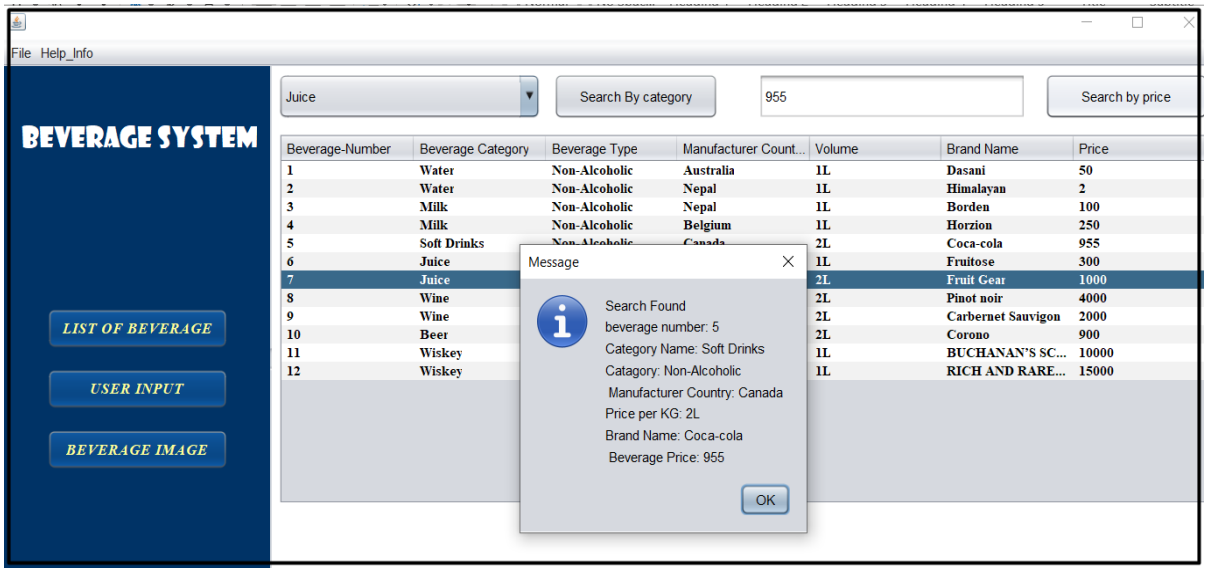


Figure 16: Searching By number

h. Entry of duplicate data

Objective	To enter duplicate data
Action	1. Checking same values
Expected Result	Checking same values and searching result comes in dialog-box
Actual Result	Checking same values and searching result comes in dialog-box
Conclusion	Test successful.

Table 8: Entry of duplicate data

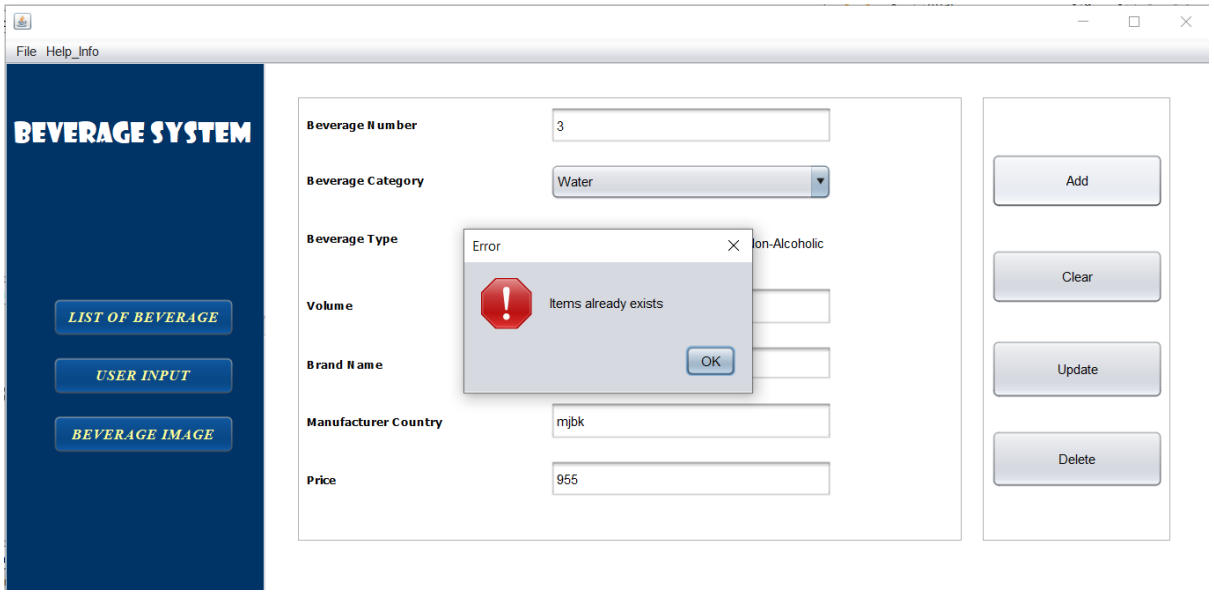


Figure 17: Entry of duplicate data

i. Open help File

Objective	To open help file
Action	1. Open help file from menu bar
Expected Result	PDF file to open
Actual Result	PDF file to open
Conclusion	Test successful.

Table 9: Open help File

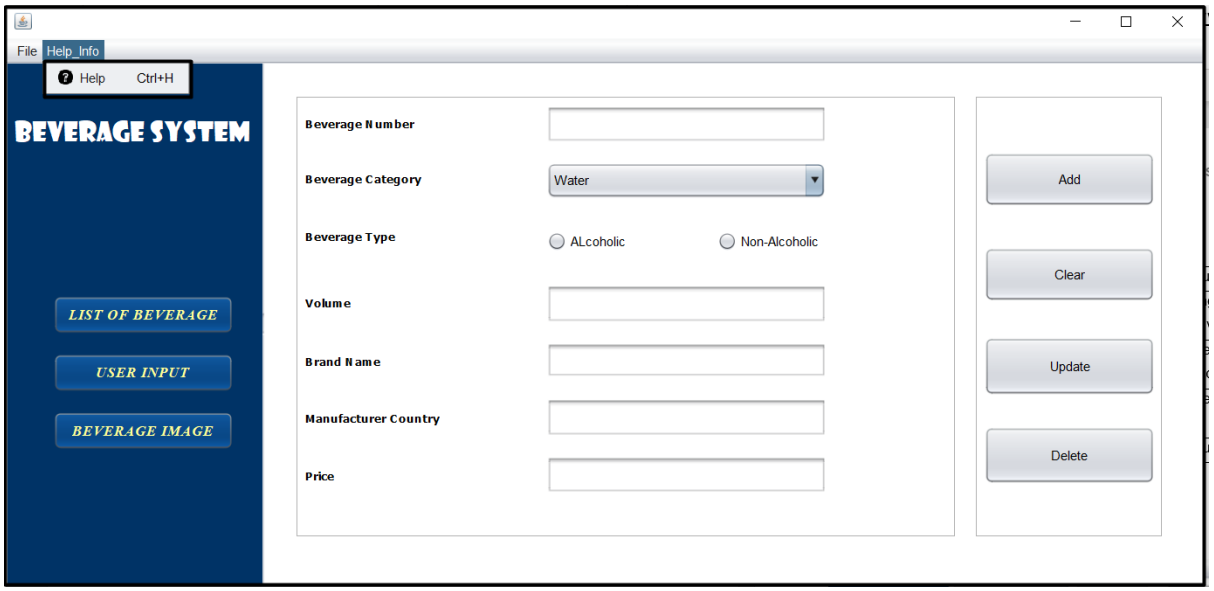


Figure 18: Open help from menu bar

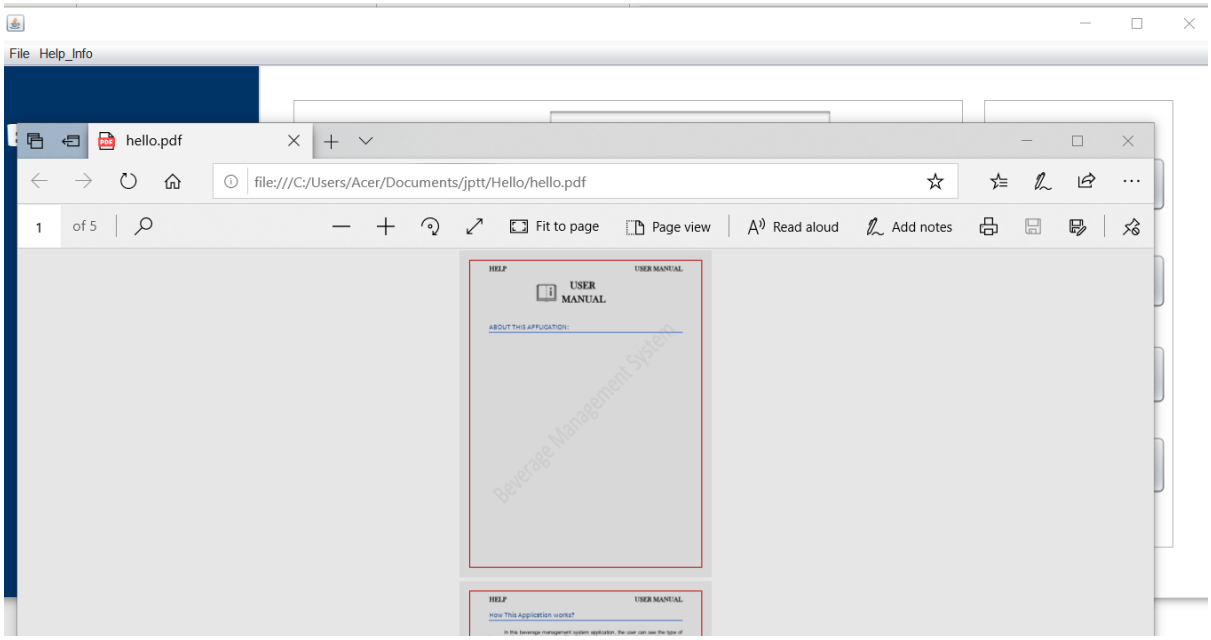
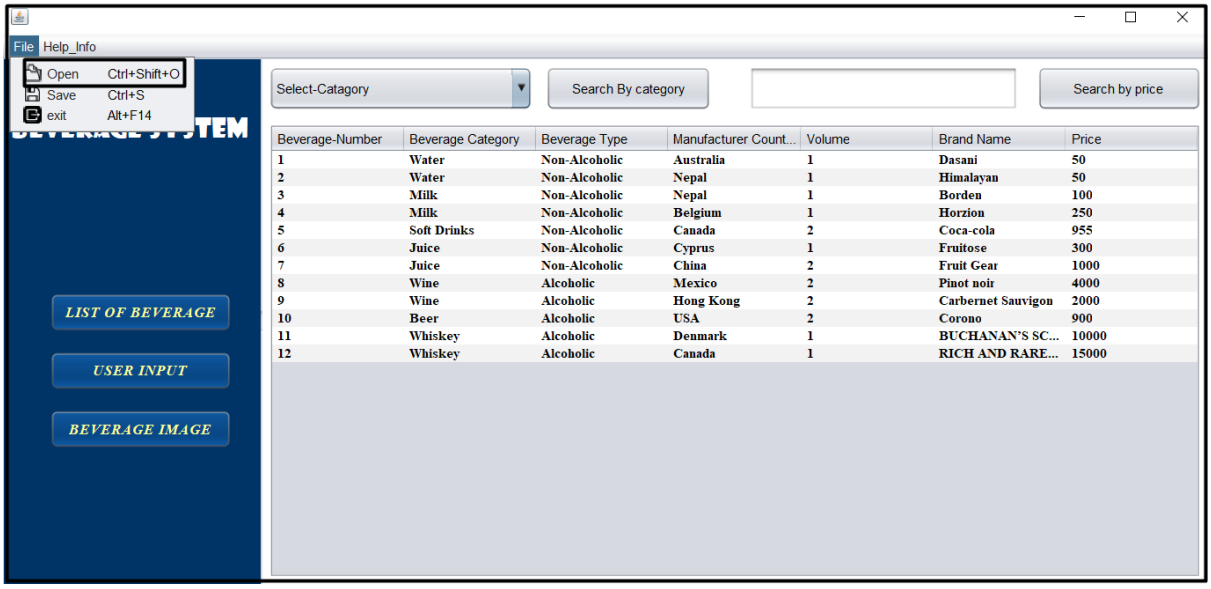


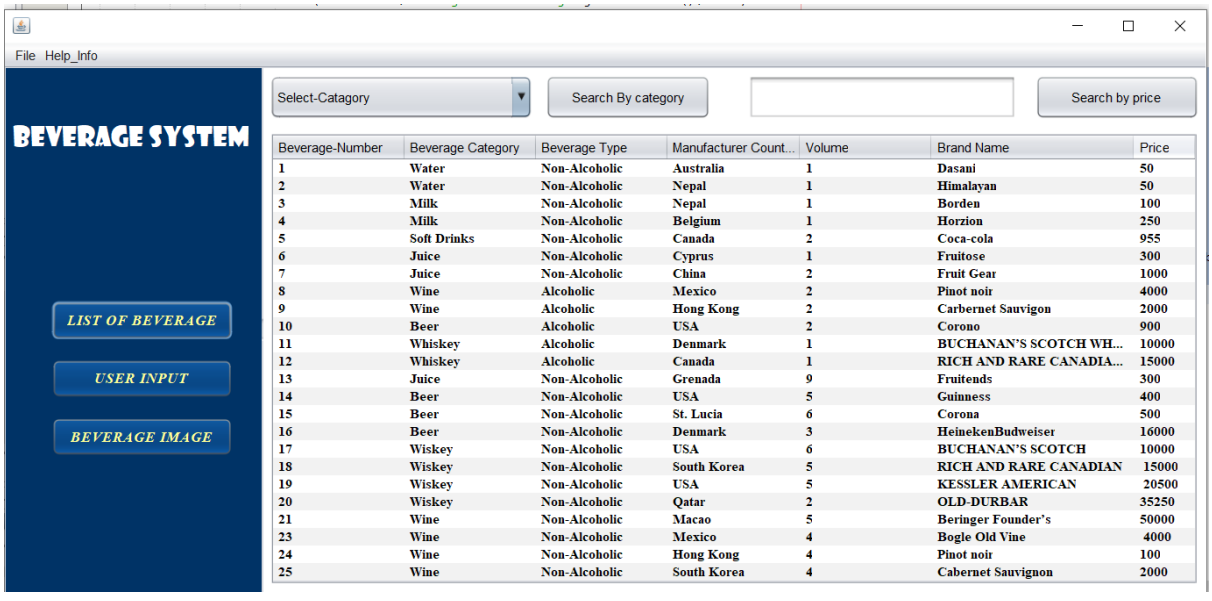
Figure 19: Open PDF file

j. Add value in table by text file

Objective	To add value in table
Action	1. Add value in table by text field
Expected Result	Add values in table
Actual Result	Add values in table
Conclusion	Test successful.

Table 10: Add value in table by text file





k. Updating the value of table

Objective	To update value in text field
Action	1. Updating table values 2. Adding values in table
Expected Result	Updating values to the table with popping out of dialog-box.
Actual Result	Updating values in table and displaying dialog box.
Conclusion	Test successful.

Table 11: Updating the value of table

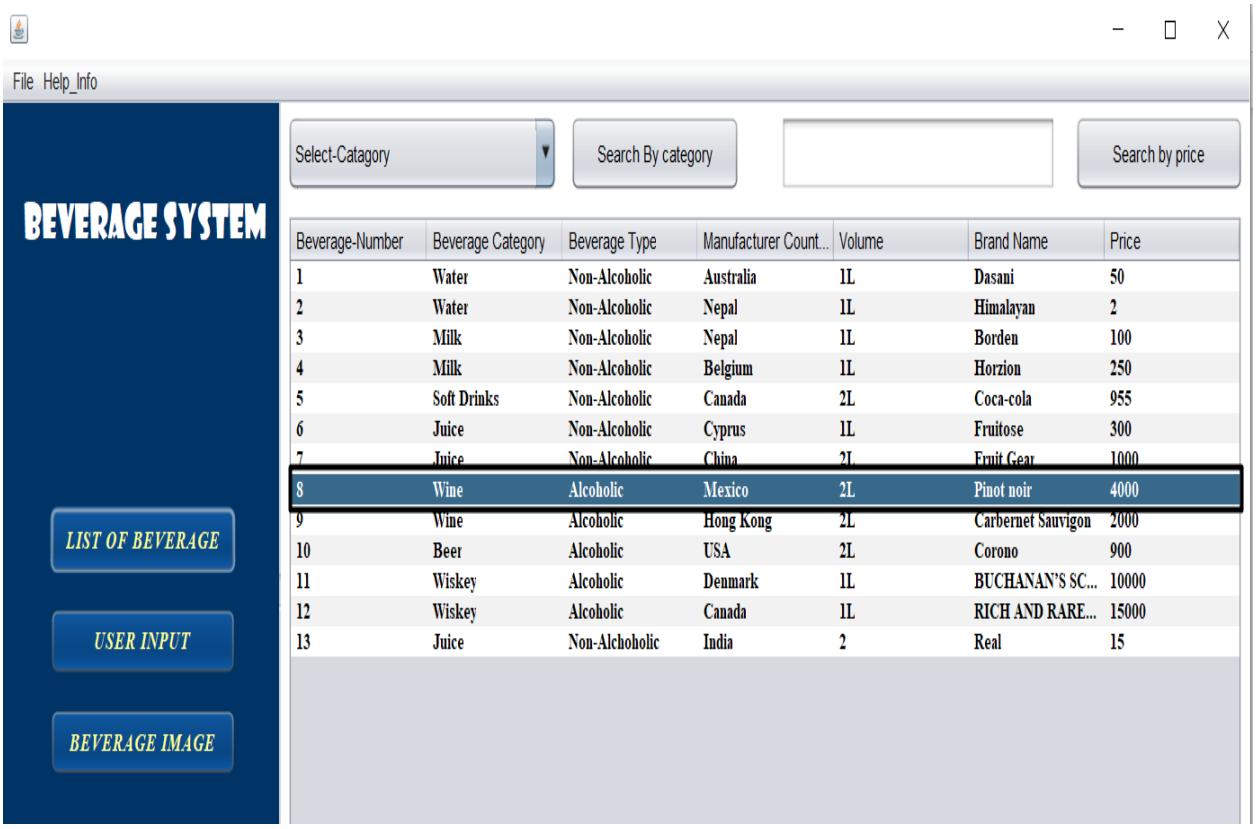


Figure 20: Updating this table

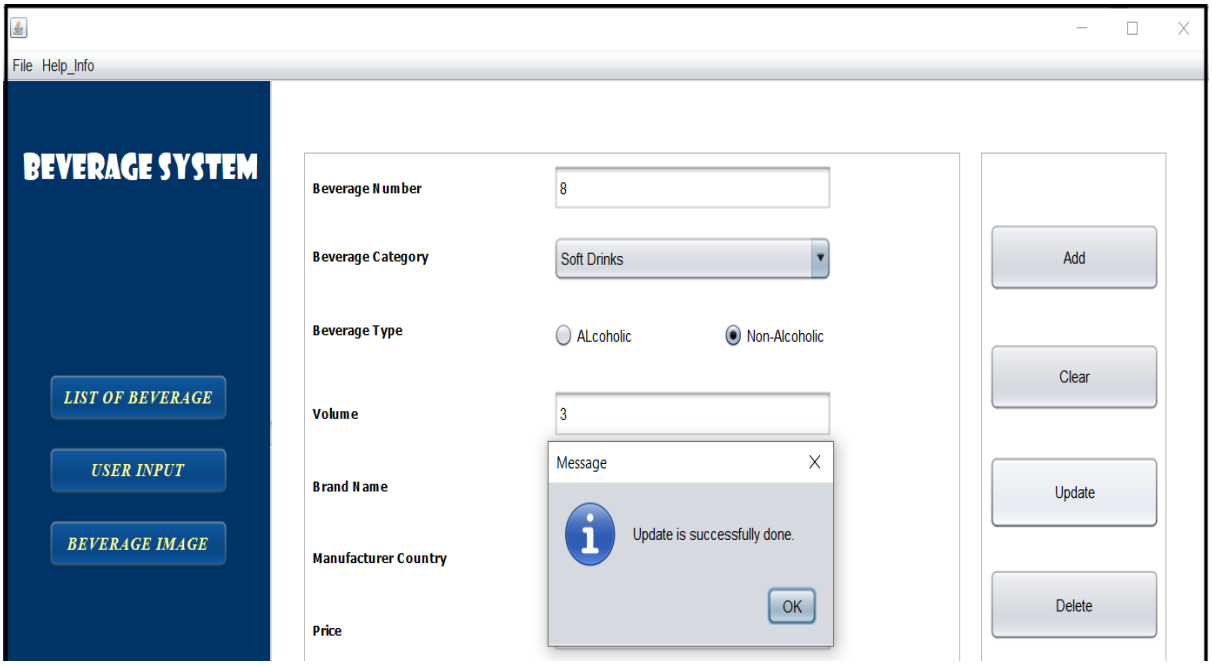


Figure 22: Changing the value of text field

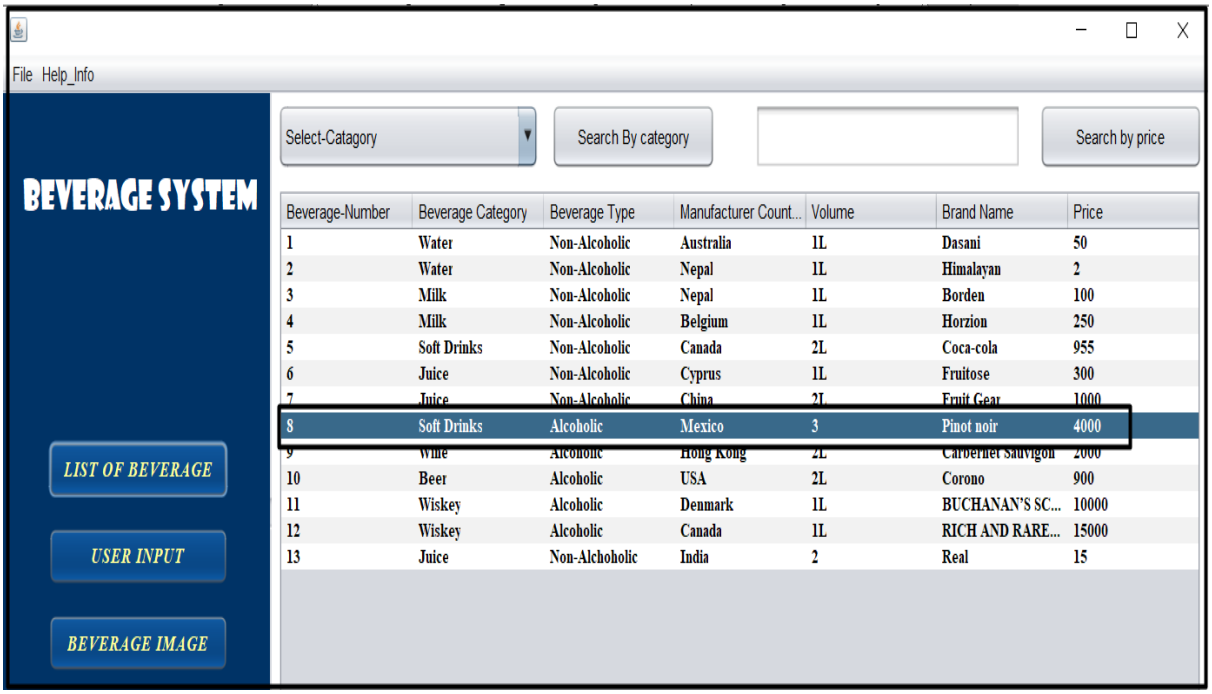


Figure 21: Table values is updated

I. Clearing values

Objective	To Clear values of text field.
Action	1. Clearing value from table
Expected Result	Clearing the values in table
Actual Result	Clearing the values in table
Conclusion	Test successful.

Table 12: Clearing values

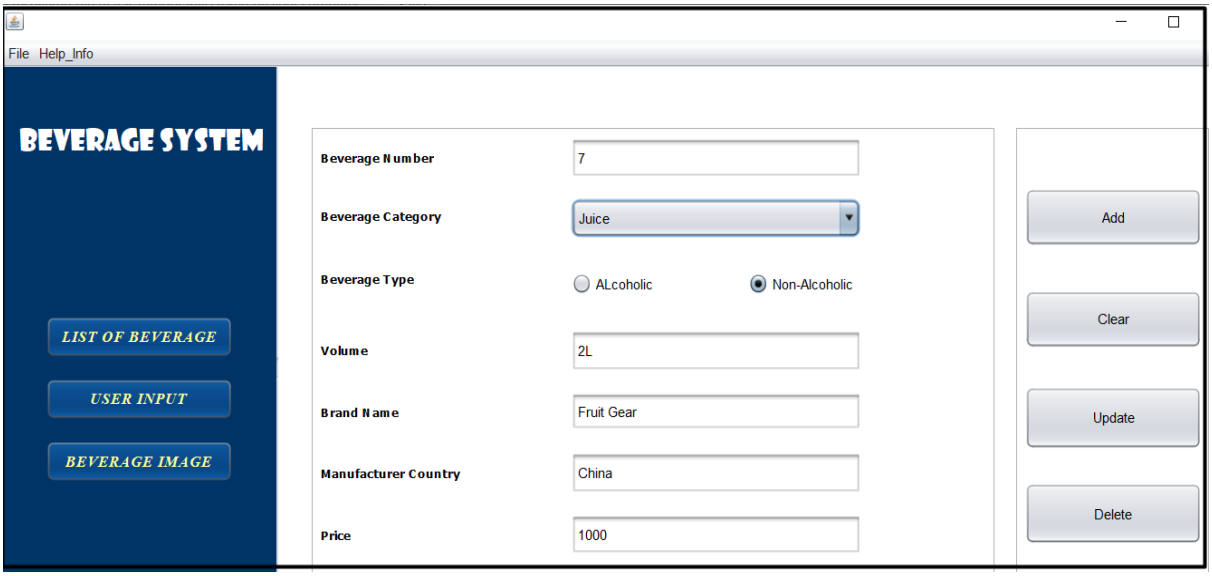


Figure 23: Entering the mistake values

9. Conclusion.

This system is developed to store information regarding beverages of different type and category of a beverage store which is very user-friendly. In this system, user can display data of beverages, add, delete and update data in table. Merge sort is used in this coursework in which the array which need to be sorted is divided into two halves to sort data and then the sorted arrays are merged into single array containing sorted data. It is based on Divide and Conquer algorithm. In this system merge sort algorithm is used to sort the data of price of beverage. Similarly, in this system binary search algorithm is used to search the data of beverage based on the category and price.

Before starting the development process of this system, lots of research was done. Together with the group discussion, proper planning was done for the development of this system. As it was a group work of three members, for each members of group individual task was assigned for both development and documentation. During development phase of the system numerous errors occurred. These errors were detected by prebuilt error detection tool which consumed lots of time. However, with the help of debugging tool and with lots of research this problem was solved. We also faced problem regarding understanding of merge sort and binary search. Hence, with the proper guidelines of teachers and study, this coursework was completed in given short period of time with determination of all the group members.

However, by this coursework we got to learn about the java programming more detailly. This coursework has taught us to work in a group with proper planning and understanding. In near the future, we would be able to develop any kind information system. We also learned about different algorithms to sort and search the data. Furthermore, we learned about the default table model that is used to add, display, update data in the row of table.

Bibliography

Anon., 2020. *Java Binary Search Program | Binary Search In Java Example.* [Online]
Available at: https://appdividend.com/2019/08/19/java-binary-search-program-binary-search-in-java-example/
[Accessed 15 January 2020].

balsamiq, 2019. *Balsamiq Mockups 3 Application Overview - Balsamiq for Desktop Documentation | Balsamiq.* [Online]
Available at: https://balsamiq.com/wireframes/desktop/docs/overview/?fbclid=IwAR0dV8WycQl_5S0avgLFB_7HREnnu_KuERPK5REX2vO_dVP06CTmbdrthHQ
[Accessed 16 DECEMBER 2019].

Febrega, M., 2018. *Card Layout.* [Online]
Available at: https://medium.com/@MarcoAFabrega_44518/4-benefits-to-using-a-card-based-design-3376ed24a130
[Accessed 17 Jan 2019].

Filehippo.com, 2019. *Download NetBeans IDE 11.1 for Windows - Filehippo.com.* [Online]
Available at: https://filehippo.com/download_netbeans/?fbclid=IwAR0R7zjNaSPjnkB3_6yWlQfd0y5LMlhhx1_tuf0bs5QLc7iCds_O3CAhp3A
[Accessed 16 DECEMBER 2019].

HackerEarth, 2020. *Binary Search Tutorials & Notes | Algorithms | HackerEarth.* [Online]
Available at: https://www.hackerearth.com/practice/algorithms/searching/binary-search/tutorial/?fbclid=IwAR1Mwb7LdLpYBHC9sOLcmQxf-00_6PILn6EMAJ5lFiOjgrvc3s8UWxDj4oo
[Accessed 12 JAN 2020].

HackerEarth, 2020. *Merge Sort Tutorials & Notes | Algorithms | HackerEarth.* [Online]
Available at: https://www.hackerearth.com/practice/algorithms/sorting/merge-sort/tutorial/
[Accessed 15 Jan 2020].

Landup, D., 2019. *Merge Sort in Java.* [Online]
Available at: https://stackabuse.com/merge-sort-in-java/
[Accessed 12 JAN 2020].

Lewis, J., 2018. *JavaScript Algorithms: What Is Binary Search, A Detailed Step-By-Step, And Example Code.* [Online]
Available at: https://medium.com/@jeffrey.allen.lewis/javascript-algorithms-

explained-binary-search-25064b896470
[Accessed 16 jan 2020].

Williams, A., 2018. *The Binary Search Algorithm in JavaScript.* [Online] Available at: https://code.tutsplus.com/tutorials/the-binary-search-algorithm-in-javascript--cms-30003
[Accessed 15 January 2020].

Appendix:

```
import java.awt.CardLayout;
import java.awt.Desktop;
import java.awt.HeadlessException;
import java.awt.event.KeyEvent;
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.util.logging.Level;
import java.util.logging.Logger;
import java.util.ArrayList;
import java.util.Arrays;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/*
 * 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.
 */
```

```
/**
 *
 * @author utsha
 */
public final class BeverageSystem extends javax.swing.JFrame {
    ArrayList<DrinksList> list = new ArrayList<>();// declearing arraylist


    /**
     * Creates new form BeverageSystem
     */
    CardLayout cardLayout;//decleraing of cardlayout
    public BeverageSystem() {
        initComponents();
        addRowToJTable();



        cardLayout =(CardLayout)(panelCard.getLayout());


    }
    /**
     * This is the class for displaying the data in table
    */
    public class DrinksList{
        public String beverageNumber;
        public String brandName;
        public String beverageType;
```

```
    public String beverageCategory;
    public String manufacturerCountry;
    public String volume;
    public String price;

    public DrinksList(String beverageNum, String bCategory,String bType,String bVolume,String brand, String bCountry, String bPrice )
    {
        this.beverageNumber =beverageNum;
        this.brandName = brand;
        this.beverageType =bType;
        this.beverageCategory= bCategory;
        this.manufacturerCountry =bCountry;
        this.volume = bVolume;
        this.price = bPrice;
    }
}
/**
 * This is the method for adding the value in array list.
*/
public ArrayList BeverageList()
{
    // Adding value in Array list
    DrinksList beverage1 = new DrinksList("1","Water","Non-Alcoholic","Australia", "Dasani", "1","50" );
    DrinksList beverage2 = new DrinksList("2","Water","Non-Alcoholic","Nepal", "Himalayan", "1","50" );
```

```
DrinksList beverage3 = new DrinksList("3","Milk","Non-Alcoholic","Nepal", "Borden", "1","100" );

DrinksList beverage4 = new DrinksList("4","Milk","Non-Alcoholic","Belgium", "Horzion", "1","250" );

DrinksList beverage5 = new DrinksList("5","Soft Drinks","Non-Alcoholic","Canada", "Coca-cola", "2","955" );

DrinksList beverage6 = new DrinksList("6","Juice","Non-Alcoholic","Cyprus", "Fruitose", "1","300" );

DrinksList beverage7 = new DrinksList("7","Juice","Non-Alcoholic","China", "Fruit Gear", "2","1000" );

DrinksList beverage8 = new DrinksList("8","Wine","Alcoholic","Mexico", "Pinot noir", "2","4000" );

DrinksList beverage9 = new DrinksList("9","Wine","Alcoholic","Hong Kong", "Carbernet Sauvigon", "2","2000" );

DrinksList beverage10 = new DrinksList("10","Beer","Alcoholic","USA", "Corono", "2","900");

DrinksList beverage11 = new DrinksList("11","Whiskey","Alcoholic","Denmark", "BUCHANAN'S SCOTCH WHISKY", "1","10000" );

DrinksList beverage12 = new DrinksList("12","Whiskey","Alcoholic","Canada", "RICH AND RARE CANADIAN WHISKY", "1","15000");

list.add(beverage1);// add values in array

list.add(beverage2);

list.add(beverage3);

list.add(beverage4);

list.add(beverage5);

list.add(beverage6);

list.add(beverage7);

list.add(beverage8);

list.add(beverage9);

list.add(beverage10);
```

```
        list.add(beverage11);
        list.add(beverage12);
        System.out.println(list);
        return list;// returing values
    }
    /**
     * This is to add the value from array to rows of table.
    */
    public void addRowToJTable()
    {
        DefaultTableModel model =(DefaultTableModel)jTableBeverage.getModel();
        ArrayList <DrinksList> list = BeverageList();
        Object rowData[] = new Object[7];
        for(int i = 0; i <list.size(); i++)
            {
            rowData[0] =list.get(i).beverageNumber;
            rowData[1] =list.get(i).beverageCategory;
            rowData[2] =list.get(i).beverageType;
            rowData[3] =list.get(i).volume;
            rowData[4] =list.get(i).manufacturerCountry;
            rowData[5] =list.get(i).brandName;
            rowData[6] =list.get(i).price;
            model.addRow(rowData);
        }
    }
     /**
```

```
    * This method is to sort the array.
    * @param sortArray
    */
   public static void sort(int[] sortArray)
  {
     if (sortArray.length <= 1) {
         return;
     }
     int[] firstNum = new int[sortArray.length / 2];
     int[] secondNum = new int[sortArray.length - firstNum.length];
     // Copying the first half of a into firstNum, the second half into secondNum
     for (int i = 0; i < firstNum.length; i++)
     {
        firstNum[i] = sortArray[i];
     }
     for (int i = 0; i < secondNum.length; i++)
     {
        secondNum[i] = sortArray[firstNum.length + i];
     }
     sort(firstNum);
     sort(secondNum);
     merge(firstNum, secondNum, sortArray);

  }

     /**
```

```
    * To merge two sorted arrays in an array
    * @param firstNum
    * @param secondNum
    * @param sortArray
    */
  private static void merge(int[] firstNum, int[] secondNum, int[] sortArray)
  {
     int firstElement = 0; // Next element to consider in the first array i.e. firstNum
     int secondElement = 0; // Next element to consider in the second array i.e. secondNum
     int num = 0; // Next open position in sortArray

     // As long as neither firsNum nor secondNum is past the end, move
     // the smaller element into sortArray

     while (firstElement < firstNum.length && secondElement < secondNum.length)
     {
        if (firstNum[firstElement] < secondNum[secondElement])
        {
           sortArray[num] = firstNum[firstElement];
           firstElement++;
        }
        else
        {
           sortArray[num] = secondNum[secondElement];
           secondElement++;
        }
```

```
        num++;
      }
    }
   /**
    * Implementation of binary search
    * @param sortArray
    * @param y
    * @return
    */
    private int sorting(int sortArray[], int y)
    {

        Arrays.sort(sortArray);
        sort(sortArray);
        System.out.println(Arrays.toString(sortArray));
        sort(sortArray);
        int start = 0;
        int end = sortArray.length -1;

        while(start <= end){

            int mid = (start+end)/2;

            //Checking if the element y is present in th middle of array
            if(sortArray[mid] == y){
                return sortArray[mid];
```

```
            }
        // checking if y element is greater than right elements from mid element of array and those elements are ignored
        else if(y > sortArray[mid]){
            start = mid +1;
        }
        else{
            end = mid-1;
        }

    }
    return -1;// returns -1 as the element is not present in the array

}



/**
 * 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() {

    buttonGroup1 = new javax.swing.ButtonGroup();
    buttonGroup2 = new javax.swing.ButtonGroup();
```

```
jSplitPane1 = new javax.swing.JSplitPane();
jPanel3 = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();
jButton1 = new javax.swing.JButton();
jButton4 = new javax.swing.JButton();
jButton5 = new javax.swing.JButton();
panelCard = new javax.swing.JPanel();
panelCard1 = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
jTableBeverage = new javax.swing.JTable();
jTextFieldSPrice = new javax.swing.JTextField();
jButtonPSearch = new javax.swing.JButton();
jComboBoxScategory = new javax.swing.JComboBox<>();
jButtonCsearch = new javax.swing.JButton();
panelCard2 = new javax.swing.JPanel();
jPanel1 = new javax.swing.JPanel();
jLabelPrice = new javax.swing.JLabel();
jTextFieldcountry = new javax.swing.JTextField();
jTextFieldBrand = new javax.swing.JTextField();
jLabelcountry = new javax.swing.JLabel();
jLabelBtype = new javax.swing.JLabel();
jComboBoxCategory = new javax.swing.JComboBox<>();
jTextFieldBNum = new javax.swing.JTextField();
jLabelBCa = new javax.swing.JLabel();
jLabelVol = new javax.swing.JLabel();
jRadioButtonAlc = new javax.swing.JRadioButton();
```

```
jRadioButtonNAlc = new javax.swing.JRadioButton();
jTextFieldPrice = new javax.swing.JTextField();
jLabelBrand = new javax.swing.JLabel();
jLabelBNum = new javax.swing.JLabel();
jTextFieldVolume = new javax.swing.JTextField();
jPanel2 = new javax.swing.JPanel();
jButtonAdd = new javax.swing.JButton();
jButtonUpdate = new javax.swing.JButton();
jButtonClear = new javax.swing.JButton();
jButtonDelete = new javax.swing.JButton();
panelCard3 = new javax.swing.JPanel();
jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
jMenuBar1 = new javax.swing.JMenuBar();
jMenu1 = new javax.swing.JMenu();
Open = new javax.swing.JMenuItem();
save = new javax.swing.JMenuItem();
exist = new javax.swing.JMenuItem();
jMenu2 = new javax.swing.JMenu();
jMenuItem1 = new javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(0, 51, 102));
```

```
jSplitPane1.setDividerSize(1);

jPanel3.setBackground(new java.awt.Color(0, 51, 102));

jLabel1.setFont(new java.awt.Font("Showcard Gothic", 0, 24)); // NOI18N
jLabel1.setForeground(new java.awt.Color(255, 255, 255));
jLabel1.setText("Beverage System");

jButton1.setBackground(new java.awt.Color(0, 51, 102));
jButton1.setFont(new java.awt.Font("Times New Roman", 3, 14)); // NOI18N
jButton1.setForeground(new java.awt.Color(255, 255, 153));
jButton1.setText("LIST OF BEVERAGE");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton4.setBackground(new java.awt.Color(0, 51, 102));
jButton4.setFont(new java.awt.Font("Times New Roman", 3, 14)); // NOI18N
jButton4.setForeground(new java.awt.Color(255, 255, 153));
jButton4.setText("USER INPUT");
jButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton4ActionPerformed(evt);
    }
```

```
        });

        jButton5.setBackground(new java.awt.Color(0, 51, 102));
        jButton5.setFont(new java.awt.Font("Times New Roman", 3, 14)); // NOI18N
        jButton5.setForeground(new java.awt.Color(255, 255, 153));
        jButton5.setText("BEVERAGE IMAGE");
        jButton5.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton5ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
                    .addGroup(jPanel3Layout.createSequentialGroup()
                        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                        .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE,                    222,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel3Layout.createSequentialGroup()
```

```
                .addGap(42, 42, 42)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING, false)

                .addComponent(jButton1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addComponent(jButton5,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addComponent(jButton4,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

                .addGap(0, 0, Short.MAX_VALUE)))

            .addContainerGap())

        );

        jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(jPanel3Layout.createSequentialGroup()

                .addContainerGap()

                .addComponent(jLabel1,    javax.swing.GroupLayout.PREFERRED_SIZE,
121, javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(85, 85, 85)

                .addComponent(jButton1,    javax.swing.GroupLayout.PREFERRED_SIZE,
35, javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(18, 18, 18)

                .addComponent(jButton4,    javax.swing.GroupLayout.PREFERRED_SIZE,
35, javax.swing.GroupLayout.PREFERRED_SIZE)

                .addGap(18, 18, 18)

                .addComponent(jButton5,    javax.swing.GroupLayout.PREFERRED_SIZE,
35, javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
            .addContainerGap(87, Short.MAX_VALUE))
        );

        jSplitPane1.setLeftComponent(jPanel3);

        panelCard.setLayout(new java.awt.CardLayout());

        panelCard1.setBackground(new java.awt.Color(255, 255, 255));

        jTableBeverage.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
        jTableBeverage.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {

            },
            new String [] {
                "Beverage-Number", "Beverage Category", "Beverage Type", "Manufacturer Country", "Volume", "Brand Name", "Price"
            }
        ) {
            boolean[] canEdit = new boolean [] {
                false, false, false, false, false, false, false
            };

            public boolean isCellEditable(int rowIndex, int columnIndex) {
                return canEdit [columnIndex];
            }
```

```
        });
        jTableBeverage.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jTableBeverageMouseClicked(evt);
            }
        });
        jScrollPane1.setViewportView(jTableBeverage);

        jButtonPSearch.setText("Search by price");
        jButtonPSearch.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButtonPSearchActionPerformed(evt);
            }
        });

        jComboBoxScategory.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "Select-Catagory", "Water",
"Milk", "Juice", "Soft Drinks", "Beer", "Whiskey", "Wine" }));

        jButtonCsearch.setText("Search By category");
        jButtonCsearch.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButtonCsearchActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout panelCard1Layout = new
javax.swing.GroupLayout(panelCard1);
```

```
        panelCard1.setLayout(panelCard1Layout);
        panelCard1Layout.setHorizontalGroup(
panelCard1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
            .addGroup(panelCard1Layout.createSequentialGroup()
                .addContainerGap()
.addGroup(panelCard1Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
                    .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 923, Short.MAX_VALUE)
                    .addGroup(panelCard1Layout.createSequentialGroup()
                        .addComponent(jComboBoxScategory,
javax.swing.GroupLayout.PREFERRED_SIZE,                    240,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jButtonCsearch,
javax.swing.GroupLayout.PREFERRED_SIZE,                    150,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jTextFieldSPrice,
javax.swing.GroupLayout.PREFERRED_SIZE,                    244,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(jButtonPSearch,
javax.swing.GroupLayout.PREFERRED_SIZE,                    149,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap())
        );
```

```
        panelCard1Layout.setVerticalGroup(

panelCard1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)

            .addGroup(panelCard1Layout.createSequentialGroup()

                .addContainerGap()

.addGroup(panelCard1Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.BASELINE)

                    .addComponent(jTextFieldSPrice,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    40,
javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addComponent(jButtonPSearch,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    40,
javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addComponent(jButtonCsearch,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    40,
javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addComponent(jComboBoxScategory,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    40,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

                .addComponent(jScrollPane1,   javax.swing.GroupLayout.DEFAULT_SIZE,
373, Short.MAX_VALUE)

                .addContainerGap())

        );


        panelCard.add(panelCard1, "panelCard1");


        panelCard2.setBackground(new java.awt.Color(255, 255, 255));
```

```
jPanel1.setBackground(new java.awt.Color(255, 255, 255));
jPanel1.setBorder(javax.swing.BorderFactory.createEtchedBorder());
jPanel1.setForeground(new java.awt.Color(204, 204, 204));

jLabelPrice.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabelPrice.setText("Price");

jTextFieldcountry.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyPressed(java.awt.event.KeyEvent evt) {
        jTextFieldcountryKeyPressed(evt);
    }
});

jTextFieldBrand.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyPressed(java.awt.event.KeyEvent evt) {
        jTextFieldBrandKeyPressed(evt);
    }
});

jLabelcountry.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabelcountry.setText("Manufacturer Country");

jLabelBtype.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
jLabelBtype.setText("Beverage Type");
```

```
        jComboBoxCategory.setModel(new
javax.swing.DefaultComboBoxModel<>(new String[] { "Water", "Milk", "Juice", "Soft
Drinks", "Beer", "Whiskey", "Wine" }));

        jTextFieldBNum.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                jTextFieldBNumKeyPressed(evt);
            }
        });

        jLabelBCa.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabelBCa.setText("Beverage Category");

        jLabelVol.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabelVol.setText("Volume");

        buttonGroup1.add(jRadioButtonAlc);
        jRadioButtonAlc.setText("ALcoholic");

        buttonGroup1.add(jRadioButtonNAlc);
        jRadioButtonNAlc.setText("Non-Alcoholic");

        jTextFieldPrice.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                jTextFieldPriceKeyPressed(evt);
            }
        });
```

```
        jLabelBrand.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabelBrand.setText("Brand Name");

        jLabelBNum.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N
        jLabelBNum.setText("Beverage Number");

        jTextFieldVolume.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyPressed(java.awt.event.KeyEvent evt) {
                jTextFieldVolumeKeyPressed(evt);
            }
        });

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING, false)
                        .addComponent(jLabelPrice,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
                .addComponent(jLabelBrand,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addComponent(jLabelVol,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addComponent(jLabelBtype,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 100, Short.MAX_VALUE))

            .addComponent(jLabelcountry,
javax.swing.GroupLayout.PREFERRED_SIZE,                    133,
javax.swing.GroupLayout.PREFERRED_SIZE)


.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
TRAILING, false)

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

                .addComponent(jLabelBCa,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 108, Short.MAX_VALUE)))


.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,      88,
Short.MAX_VALUE)


.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING, false)

            .addComponent(jTextFieldBNum)

            .addComponent(jTextFieldBrand)

            .addGroup(jPanel1Layout.createSequentialGroup()
```

```
                .addComponent(jRadioButtonAlc,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    91,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

                .addComponent(jRadioButtonNAlc,
javax.swing.GroupLayout.PREFERRED_SIZE,                                   100,
javax.swing.GroupLayout.PREFERRED_SIZE))

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

            .addComponent(jTextFieldPrice,
javax.swing.GroupLayout.DEFAULT_SIZE, 256, Short.MAX_VALUE)

            .addComponent(jTextFieldcountry,
javax.swing.GroupLayout.DEFAULT_SIZE, 256, Short.MAX_VALUE)

            .addComponent(jTextFieldVolume))

        .addContainerGap(117, Short.MAX_VALUE))

    );

    jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel1Layout.createSequentialGroup()

            .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING, false)

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

            .addComponent(jTextFieldBNum,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    34,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(18, 18, 18)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

            .addComponent(jLabelBCa,
javax.swing.GroupLayout.PREFERRED_SIZE,                    30,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addComponent(jComboBoxCategory,
javax.swing.GroupLayout.PREFERRED_SIZE,                    33,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

            .addGroup(jPanel1Layout.createSequentialGroup()

              .addComponent(jLabelBtype,
javax.swing.GroupLayout.PREFERRED_SIZE,                    35,
javax.swing.GroupLayout.PREFERRED_SIZE)

              .addGap(0, 0, Short.MAX_VALUE))

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

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

        .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING, false)

            .addComponent(jTextFieldVolume,
javax.swing.GroupLayout.DEFAULT_SIZE, 35, Short.MAX_VALUE)

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

        .addGap(18, 18, 18)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)

                .addComponent(jTextFieldBrand,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    32,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(jLabelBrand,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    32,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(18, 18, 18)


.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)

                .addComponent(jTextFieldcountry,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    35,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(jLabelcountry,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    34,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(18, 18, 18)


.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)

                .addComponent(jTextFieldPrice,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    34,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addComponent(jLabelPrice,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    34,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGap(38, 38, 38))

        );


        jPanel2.setBackground(new java.awt.Color(255, 255, 255));
```

```
jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder());
jPanel2.setForeground(new java.awt.Color(255, 255, 255));

jButtonAdd.setText("Add");
jButtonAdd.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonAddActionPerformed(evt);
    }
});

jButtonUpdate.setText("Update");
jButtonUpdate.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonUpdateActionPerformed(evt);
    }
});

jButtonClear.setText("Clear");
jButtonClear.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButtonClearActionPerformed(evt);
    }
});

jButtonDelete.setText("Delete");
jButtonDelete.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonDeleteActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addContainerGap()
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jButtonDelete, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jButtonUpdate, javax.swing.GroupLayout.DEFAULT_SIZE, 156, Short.MAX_VALUE)
                .addComponent(jButtonAdd, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jButtonClear, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addContainerGap())
    );
    jPanel2Layout.setVerticalGroup(
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```

```
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addContainerGap(49, Short.MAX_VALUE)
            .addComponent(jButtonAdd,
javax.swing.GroupLayout.PREFERRED_SIZE,                    49,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(38, 38, 38)
            .addComponent(jButtonClear,
javax.swing.GroupLayout.PREFERRED_SIZE,                    49,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(33, 33, 33)
            .addComponent(jButtonUpdate,
javax.swing.GroupLayout.PREFERRED_SIZE,                    53,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(29, 29, 29)
            .addComponent(jButtonDelete,
javax.swing.GroupLayout.PREFERRED_SIZE,                    52,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(48, 48, 48))
    );

    javax.swing.GroupLayout panelCard2Layout = new
javax.swing.GroupLayout(panelCard2);
    panelCard2.setLayout(panelCard2Layout);
    panelCard2Layout.setHorizontalGroup(

panelCard2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
        .addGroup(panelCard2Layout.createSequentialGroup()
            .addGap(30, 30, 30)
            .addComponent(jPanel1,    javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
            .addGap(18, 18, 18)
            .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        panelCard2Layout.setVerticalGroup(
            panelCard2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(panelCard2Layout.createSequentialGroup()
            .addGap(30, 30, 30)
            .addGroup(panelCard2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

        panelCard.add(panelCard2, "panelCard2");

        panelCard3.setBackground(new java.awt.Color(255, 255, 255));

        jLabel2.setBackground(new java.awt.Color(255, 255, 255));
        jLabel2.setIcon(new javax.swing.ImageIcon(getClass().getResource("/image/5227.jpg"))); // NOI18N
```

```
        jLabel3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/image/5.jpg"))); // NOI18N


        jLabel4.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/image/1.jpg"))); // NOI18N
        jLabel4.setText("jLabel4");


        jLabel6.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/image/3.jpg"))); // NOI18N


        javax.swing.GroupLayout panelCard3Layout = new
javax.swing.GroupLayout(panelCard3);
        panelCard3.setLayout(panelCard3Layout);
        panelCard3Layout.setHorizontalGroup(

panelCard3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
            .addGroup(panelCard3Layout.createSequentialGroup()
                .addGap(23, 23, 23)

.addGroup(panelCard3Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.TRAILING)
                    .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE,
242, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE,
243, javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(panelCard3Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
                    .addGroup(panelCard3Layout.createSequentialGroup()
```

```
                .addGap(18, 18, 18)
                .addComponent(jLabel3))
            .addGroup(panelCard3Layout.createSequentialGroup()
                .addGap(27, 27, 27)
                .addComponent(jLabel6,
javax.swing.GroupLayout.PREFERRED_SIZE,                    380,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(270, Short.MAX_VALUE))
    );
    panelCard3Layout.setVerticalGroup(
panelCard3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
        .addGroup(panelCard3Layout.createSequentialGroup()
            .addGap(19, 19, 19)
.addGroup(panelCard3Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING, false)
                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE,
184, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel3,    javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addGroup(panelCard3Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
                .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE,
214, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE,
214, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(17, Short.MAX_VALUE))
```

```
        );

        panelCard.add(panelCard3, "panelCard3");

        jSplitPane1.setRightComponent(panelCard);

        jMenu1.setText("File");

Open.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.
VK_O, java.awt.event.InputEvent.SHIFT_MASK |
java.awt.event.InputEvent.CTRL_MASK));
        Open.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/image/open file.png"))); // NOI18N
        Open.setText("Open");
        Open.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                OpenActionPerformed(evt);
            }
        });
        jMenu1.add(Open);

save.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.
VK_S, java.awt.event.InputEvent.CTRL_MASK));
        save.setIcon(new  javax.swing.ImageIcon(getClass().getResource("/image/save
file.png"))); // NOI18N
        save.setText("Save");
        save.addActionListener(new java.awt.event.ActionListener() {
```

```
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            saveActionPerformed(evt);
        }
    });
    jMenu1.add(save);

exist.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.
VK_F14, java.awt.event.InputEvent.ALT_MASK));
    exist.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/image/exit.png"))); // NOI18N
    exist.setText("exit");
    exist.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            existActionPerformed(evt);
        }
    });
    jMenu1.add(exist);

    jMenuBar1.add(jMenu1);

    jMenu2.setText("Help_Info");

jMenuItem1.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.Key
Event.VK_H, java.awt.event.InputEvent.CTRL_MASK));
    jMenuItem1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/image/help.png"))); // NOI18N
    jMenuItem1.setText("Help");
```

```
        jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem1ActionPerformed(evt);
            }
        });
        jMenu2.add(jMenuItem1);

        jMenuBar1.add(jMenu2);

        setJMenuBar(jMenuBar1);

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jSplitPane1)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jSplitPane1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT_SIZE, 502, Short.MAX_VALUE)
        );

        pack();
        setLocationRelativeTo(null);
    }// </editor-fold>
```

```
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        cardLayout.show(panelCard, "panelCard1");
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        cardLayout.show(panelCard, "panelCard2");
    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        cardLayout.show(panelCard, "panelCard3");
    }
    DefaultTableModel model;
    String bNumber;
    String bCategory;
    String bType;
    String bName;
    String bCountry;
    String bvolume;
    String bprice;
    /**
     * Add data to the table
     * @param evt
     */

    private void jButtonAddActionPerformed(java.awt.event.ActionEvent evt) {
```

```
try{
    bNumber = jTextFieldBNum.getText();
    if(jRadioButtonAlc.isSelected()){
      bType ="Alcoholic";
    }else if(jRadioButtonNAlc.isSelected()){
      bType="Non-Alchoholic";
    }
    bvolume = jTextFieldVolume.getText();
    bCategory = jComboBoxCategory.getSelectedItem().toString();
    bName = jTextFieldBrand.getText();
    bCountry = jTextFieldcountry.getText();
    bprice = jTextFieldPrice.getText();

    //Checking if one or more field is empty
    if(bNumber.isEmpty()|| bType.isEmpty()||bvolume.isEmpty() ||
bCategory.isEmpty()||bName.isEmpty()||bCountry.isEmpty()||bprice.isEmpty()){//||(!j
RadioButtonAlc.isSelected() == false )&&(jRadioButtonNAlc.isSelected() == false))){
        JOptionPane.showMessageDialog(null, "One or more fields is Empty");

    }else{

        DefaultTableModel model
=(DefaultTableModel)jTableBeverage.getModel();
        int row = jTableBeverage.getRowCount();
        boolean exists = false;
        for (int i = 0; i < row; i++) {
```

```
            if (jTableBeverage.getValueAt(i, 0)!=null || jTableBeverage.getValueAt(i, 5)!=null) {
                if (jTableBeverage.getValueAt(i, 0).equals(bNumber) || jTableBeverage.getValueAt(i, 5).equals(bName)) {
                    JOptionPane.showMessageDialog(rootPane, "Items already exists", "Error", JOptionPane.ERROR_MESSAGE);
                    exists=true;
                    break;
                }
            }
        }
        if (!exists) {
            String[] arrays={bNumber,bCategory, bType, bCountry, bvolume, bName, bprice};
            model.addRow(arrays);
            JOptionPane.showMessageDialog(null, "All fields are filled. The beverage number is"+ bNumber+","+"Beverage category is"+ bCategory
            +"Beverage Type is"+bType+","+"Manufacturer Country is"+ bCountry+","+"Volume is"+ bvolume+"Brand is"+ bName+"Price is"+ bprice);
        }
        jTextFieldBNum.setText("");
        jTextFieldBrand.setText("");
        jTextFieldcountry.setText("");
        jTextFieldPrice.setText("");
        jTextFieldVolume.setText("");
```

```
            buttonGroup1.clearSelection();
            jComboBoxCategory.setSelectedItem("Water");
        }



    }catch(Exception e){
        JOptionPane.showMessageDialog(null, "Something went wrong.Please check properly");
    }
}



private void jTextFieldBNumKeyPressed(java.awt.event.KeyEvent evt) {
    String num = jTextFieldBNum.getText();
    int length = num.length();
    char c = evt.getKeyChar();
    if(evt.getKeyChar()<='9'){
    // check for length not more that 10 digite
        if(length<10){
            jTextFieldBNum.setEditable(true);
        }else{
            jTextFieldBNum.setEditable(false);
        }
    }else{
        if(evt.getExtendedKeyCode()==KeyEvent.VK_BACK_SPACE || evt.getExtendedKeyCode()==KeyEvent.VK_DELETE){
            jTextFieldBNum.setEditable(true);
```

```
        }else{
            jTextFieldBNum.setEditable(false);
        }
    }
}

private void jTextFieldBrandKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    char c = evt.getKeyChar();
    if(Character.isLetter(c)||Character.isWhitespace(c)||Character.isISOControl(c)){
      // iso control for edit operation(delete key and backspace is allowed)
      //if enter character is letter, space and isocontrol char that allow to edit
      jTextFieldBrand.setEditable(true);
    }else{
      jTextFieldBrand.setEditable(false);
    }
}

private void jTextFieldPriceKeyPressed(java.awt.event.KeyEvent evt) {
    String price = jTextFieldPrice.getText();
    int length = price.length();
    char c = evt.getKeyChar();
    if( evt.getKeyChar()>='0' && evt.getKeyChar()<='9'){
    // check for length not more that 10 digite
        if(length<10){
            jTextFieldPrice.setEditable(true);
```

```
            }else{
                jTextFieldPrice.setEditable(false);
            }
        }else{
            if(evt.getExtendedKeyCode()==KeyEvent.VK_BACK_SPACE                    ||
evt.getExtendedKeyCode()==KeyEvent.VK_DELETE){
              jTextFieldPrice.setEditable(true);
            }else{
                jTextFieldPrice.setEditable(false);
            }
        }
    }

    private void jTextFieldcountryKeyPressed(java.awt.event.KeyEvent evt) {
        char c = evt.getKeyChar();
        if(Character.isLetter(c)||Character.isWhitespace(c)||Character.isISOControl(c)){
          // iso control for edit operation(delete key and backspace is allowed)
          //if enter character is letter, space and isocontrol char that allow to edit
          jTextFieldBrand.setEditable(true);
        }else{
          jTextFieldBrand.setEditable(false);
        }
    }
    /**
     * Updates selected row of table
     * @param evt
     */
```

```
    private void jButtonUpdateActionPerformed(java.awt.event.ActionEvent evt) {
        //get table model
        try{
            DefaultTableModel tableModel =(DefaultTableModel)jTableBeverage.getModel();
            if(jTableBeverage.getSelectedRowCount() == 1){
                //if single row is selected thn update
                String num = jTextFieldBNum.getText();
                String brand = jTextFieldBrand.getText();
                String country = jTextFieldcountry.getText();
                String price = jTextFieldPrice.getText();
                String category = jComboBoxCategory.getSelectedItem().toString();
                String volume = jTextFieldVolume.getText();
                if(jRadioButtonAlc.isSelected()){
                    String type ="Alcoholic";
                }else if(jRadioButtonNAlc.isSelected()){
                    String type="Non-Alcoholic";
                }

                //set udated vlue on table row

                tableModel.setValueAt(num, jTableBeverage.getSelectedRow(), 0);
                tableModel.setValueAt(category, jTableBeverage.getSelectedRow(), 1);

                tableModel.setValueAt(country, jTableBeverage.getSelectedRow(), 3);
                tableModel.setValueAt(volume, jTableBeverage.getSelectedRow(), 4);
                tableModel.setValueAt(brand, jTableBeverage.getSelectedRow(), 5);
```

```
            tableModel.setValueAt(price, jTableBeverage.getSelectedRow(), 6);
            //update message display

            JOptionPane.showMessageDialog(this, "Update is successfully done.");
            jTextFieldBNum.setText("");
            jTextFieldBrand.setText("");
            jTextFieldcountry.setText("");
            jTextFieldPrice.setText("");
            jTextFieldVolume.setText("");
            buttonGroup1.clearSelection();
            jComboBoxCategory.setSelectedItem("Water");
        }else{
            if(jTableBeverage.getRowCount()==0){
                // if the table is empty
                JOptionPane.showMessageDialog(this, "The table is empty.");
            }else{
                //if no row is selected or mutiple row is selected of table.
                JOptionPane.showMessageDialog(this, "No row is selected.");
            }
        }
    }catch(Exception e){
        JOptionPane.showMessageDialog(null, "Something went wrong.Please check
properly");
    }
  }
  /**
   * Mouse click event for selecting row
```

```
     * @param evt
     */
    private void jTableBeverageMouseClicked(java.awt.event.MouseEvent evt) {
        //display data from Jtable in input fileds
        try{
            DefaultTableModel tableModel =(DefaultTableModel)jTableBeverage.getModel();
            // set data to ext field when raw is selected

            String tblNumber= tableModel.getValueAt(jTableBeverage.getSelectedRow(), 0).toString();
            String tblCategory= tableModel.getValueAt(jTableBeverage.getSelectedRow(), 1).toString();
            String tblType= tableModel.getValueAt(jTableBeverage.getSelectedRow(), 2).toString();
            String tblCountry= tableModel.getValueAt(jTableBeverage.getSelectedRow(), 3).toString();
            String tblVolume= tableModel.getValueAt(jTableBeverage.getSelectedRow(), 4).toString();
            String tblName= tableModel.getValueAt(jTableBeverage.getSelectedRow(), 5).toString();
            String tblprice= tableModel.getValueAt(jTableBeverage.getSelectedRow(), 6).toString();

            //set to textField
            jTextFieldBNum.setText(tblNumber);
            jTextFieldBrand.setText(tblName);
            jTextFieldcountry.setText(tblCountry);
            jTextFieldVolume.setText(tblVolume);
            jTextFieldPrice.setText(tblprice);
```

```
            if(tblType.equals("Alcoholic")){
                jRadioButtonAlc.setSelected(true);
            }
            else if(tblType.equals("Non-Alcoholic")){
                jRadioButtonNAlc.setSelected(true);
            }
            //comboBox.getSelectedItem()
            for(int i=0; i< jComboBoxCategory.getItemCount(); i++){
               if(jComboBoxCategory.getItemAt(i).equalsIgnoreCase(tblCategory)){
                jComboBoxCategory.setSelectedIndex(i);


                }
            }
        }catch(HeadlessException e){
           JOptionPane.showMessageDialog(null, "Something went wrong.Please check
properly");
        }



    }
    /**
     * Clear user input
     * @param evt
     */
    private void jButtonClearActionPerformed(java.awt.event.ActionEvent evt) {
        jTextFieldBNum.setText("");
        jTextFieldBrand.setText("");
```

```
        jTextFieldcountry.setText("");
        jTextFieldPrice.setText("");
        jTextFieldVolume.setText("");
        buttonGroup1.clearSelection();

    }

    private void jButtonDeleteActionPerformed(java.awt.event.ActionEvent evt) {
        //to delete selescted row
            try{
                DefaultTableModel tableModel
=(DefaultTableModel)jTableBeverage.getModel();

                int getSelctedRowToDelete = jTableBeverage.getSelectedRow();
                // checking if their row is selected
                if(getSelctedRowToDelete>=0){
                    tableModel.removeRow(getSelctedRowToDelete);
                    JOptionPane.showMessageDialog(null,"The row is deleted..");
                    jTextFieldBNum.setText("");
                    jTextFieldBrand.setText("");
                    jTextFieldcountry.setText("");
                    jTextFieldPrice.setText("");
                    jTextFieldVolume.setText("");
                    buttonGroup1.clearSelection();

                }else{
                    JOptionPane.showMessageDialog(null,"The row is unable to delete..");
```

```
            }
        }catch(HeadlessException e){
            JOptionPane.showMessageDialog(null,"Error message!!!!");
        }
    }
    /**
     * Search by price using binary search
     * @param evt
     */
    private void jButtonPSearchActionPerformed(java.awt.event.ActionEvent evt) {
        try{
            int rows = jTableBeverage.getRowCount();
            int searchIndex = 6;//Index of column of price is 6
            String search =jTextFieldSPrice.getText();//getting value form text field of serach by price
            int datarows = 0;//lopping ko lagfii
            for(int i = 0;i< rows ; i++)
            {
                if((jTableBeverage.getValueAt(i, searchIndex)) == null) //Check if field is empty
                {
                    break;
                }
                datarows++;
            }
           if(!"".equals(search) && datarows != 0)
```

```
        {

            int findItem = Integer.parseInt(jTextFieldSPrice.getText());
            int data[] = new int[datarows];

            for(int i = 0;i< datarows ; i++){
               String a = (String) jTableBeverage.getValueAt(i, searchIndex);
               data[i]= Integer.parseInt(a);
             }

            System.out.println(Arrays.toString(data));

            int searchResult= (int) sorting(data, findItem);

            System.out.println(searchResult);

            if(searchResult != -1)
            {
               for (int i = 0; i < jTableBeverage.getRowCount(); i++)

                  if(Integer.parseInt((String) jTableBeverage.getValueAt(i,
searchIndex))== searchResult)
                  {
                     String bNum = (String) jTableBeverage.getValueAt(i, 0);
                     String bCategory = (String) jTableBeverage.getValueAt(i, 1);
                     String btype = (String) jTableBeverage.getValueAt(i, 2);
                     String bMcountry = (String) jTableBeverage.getValueAt(i, 3);
```

```
                String bvolume = (String) jTableBeverage.getValueAt(i, 4);
                String bBrand = (String) jTableBeverage.getValueAt(i, 5);
                String bPrice = (String) jTableBeverage.getValueAt(i, 6);

                JOptionPane.showMessageDialog(rootPane,"Search Found\nbeverage number: "+bNum+"\nCategory Name: "+bCategory+"\nCatagory: "+btype+"\n Manufacturer Country: "+bMcountry+"\n volume "+bvolume+"\nBrand Name: "+bBrand+"\n Beverage Price: "+bPrice,"Message",JOptionPane.INFORMATION_MESSAGE);
                break;
            }
        }
        else
        {
         JOptionPane.showMessageDialog(rootPane,"Enter the price of beverage!","Message",JOptionPane.WARNING_MESSAGE);
        }
      }
      else
      {
      JOptionPane.showMessageDialog(rootPane,"Enter the price of beverage!","Message",JOptionPane.WARNING_MESSAGE);
      }
      jTextFieldSPrice.setText("");
    }catch(HeadlessException | NumberFormatException e){
        JOptionPane.showMessageDialog(null,"Error message!!!!");
    }
  }
  /**
```

```
     * Search by category using linear search
     * @param evt
     */
    private void jButtonCsearchActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String categorySearch = (String)(jComboBoxScategory.getSelectedItem());
        int categoryCount = 0;
        String itemsFound = "";
        if (categorySearch.equals("Select a category"))
        {
            JOptionPane.showMessageDialog(rootPane, "Please choose a category.");

        }
        else
        {

            DefaultTableModel model = (DefaultTableModel)jTableBeverage.getModel();
            int row = model.getRowCount();
            for (int i = 0; i < row; i++)
            {
                String temp = (String)(jTableBeverage.getValueAt(i, 1));
                if (temp.equals(categorySearch)) {
                    categoryCount++;
                    itemsFound = itemsFound + "\n" + ((jTableBeverage.getValueAt(i, 5)) );
                }
            }
```

```
            if (categoryCount == 0)
            {
                JOptionPane.showMessageDialog(rootPane, "No items found in this category");
            }
            else
            {
                JOptionPane.showMessageDialog(rootPane,"There are " + categoryCount + " items in this category." + "\nItems are: " + itemsFound);
            }
        }
    }
    /**
     * to open Help file
     * @param evt
     */
    private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
        Runtime open = Runtime.getRuntime();
        String File = "help.pdf";
        try{
            File myFile = new File(System.getProperty("user.dir")+"\\src\\"+File);
            Desktop.getDesktop().open(myFile); // Returns the Desktop instance of the current browser context.

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

```
    }
/**
 * to exist
 * @param evt
 */
    private void existActionPerformed(java.awt.event.ActionEvent evt) {
        System.exit(0);

    }
/**
 * to open file
 * @param evt
 */

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


        try {
            BufferedReader br = new BufferedReader(new FileReader("src/open.txt"));

            DefaultTableModel hello = (DefaultTableModel) jTableBeverage.getModel();

            Object[] tableLines = br.lines().toArray();

            for (Object tableLine : tableLines) {
```

```
            String line = tableLine.toString().trim();
            String[] dataRow = line.split(",");
            hello.addRow(dataRow);
        }

    }

    catch (FileNotFoundException ex) {
        Logger.getLogger(BeverageSystem.class.getName()).log(Level.SEVERE,
null, ex);
    }
    Open.setEnabled(false);
  }
/**
 * to save values in text file
 * @param evt
 */
  private void saveActionPerformed(java.awt.event.ActionEvent evt) {

    try{
        FileWriter fr = new FileWriter("src/Savefile.txt");//setting path

       BufferedWriter bw =new BufferedWriter(fr);


        for(int k = 0; k<jTableBeverage.getRowCount();k++){
            for (int l=0; l<jTableBeverage.getColumnCount();l++){
```

```
                if (l==6) {
                    bw.write(jTableBeverage.getModel().getValueAt(k,l).toString());
                }else{
                    bw.write(jTableBeverage.getModel().getValueAt(k,l)+",");
                }

              }
              bw.write("\n");

            }
            bw.close();
            fr.close();
            JOptionPane.showMessageDialog(null,"Data Saved");
        }catch(IOException ex){
            Logger.getLogger(BeverageSystem.class.getName()).log(Level.SEVERE,
null, ex);
        }

    }
/**
 * validation of volume
 * @param evt
 */
    private void jTextFieldVolumeKeyPressed(java.awt.event.KeyEvent evt) {
        String num = jTextFieldVolume.getText();
        int length = num.length();
        char c = evt.getKeyChar();
```

```
        if(evt.getKeyChar()<='9'){
        // check for length not more that 10 digite
            if(length<10){
                jTextFieldVolume.setEditable(true);
            }else{
                jTextFieldVolume.setEditable(false);
            }
        }else{
            if(evt.getExtendedKeyCode()==KeyEvent.VK_BACK_SPACE                    ||
evt.getExtendedKeyCode()==KeyEvent.VK_DELETE){
              jTextFieldVolume.setEditable(true);
            }else{
                jTextFieldVolume.setEditable(false);
            }
        }
    }



    /**
     * @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 ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(BeverageSystem.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(BeverageSystem.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(BeverageSystem.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(BeverageSystem.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>
```

```
        /* Create and display the form */

        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new BeverageSystem().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JMenuItem Open;
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.ButtonGroup buttonGroup2;
    private javax.swing.JMenuItem exist;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton4;
    private javax.swing.JButton jButton5;
    private javax.swing.JButton jButtonAdd;
    private javax.swing.JButton jButtonClear;
    private javax.swing.JButton jButtonCsearch;
    private javax.swing.JButton jButtonDelete;
    private javax.swing.JButton jButtonPSearch;
    private javax.swing.JButton jButtonUpdate;
    private javax.swing.JComboBox<String> jComboBoxCategory;
    private javax.swing.JComboBox<String> jComboBoxScategory;
    private javax.swing.JLabel jLabel1;
```

```
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabelBCa;
    private javax.swing.JLabel jLabelBNum;
    private javax.swing.JLabel jLabelBrand;
    private javax.swing.JLabel jLabelBtype;
    private javax.swing.JLabel jLabelPrice;
    private javax.swing.JLabel jLabelVol;
    private javax.swing.JLabel jLabelcountry;
    private javax.swing.JMenu jMenu1;
    private javax.swing.JMenu jMenu2;
    private javax.swing.JMenuBar jMenuBar1;
    private javax.swing.JMenuItem jMenuItem1;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JRadioButton jRadioButtonAlc;
    private javax.swing.JRadioButton jRadioButtonNAlc;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JSplitPane jSplitPane1;
    private javax.swing.JTable jTableBeverage;
    private javax.swing.JTextField jTextFieldBNum;
    private javax.swing.JTextField jTextFieldBrand;
    private javax.swing.JTextField jTextFieldPrice;
```

```
    private javax.swing.JTextField jTextFieldSPrice;
    private javax.swing.JTextField jTextFieldVolume;
    private javax.swing.JTextField jTextFieldcountry;
    private javax.swing.JPanel panelCard;
    private javax.swing.JPanel panelCard1;
    private javax.swing.JPanel panelCard2;
    private javax.swing.JPanel panelCard3;
    private javax.swing.JMenuItem save;
    // End of variables declaration

}
```