Sourcecode:

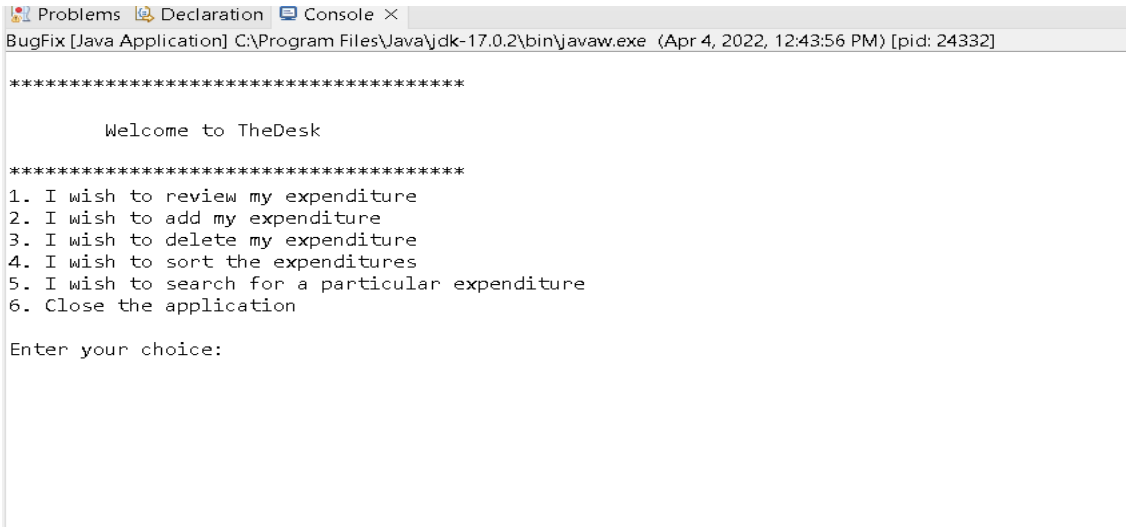
```
package com.simplilearn.demo;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class BugFix {

    public static void main(String[] args) {
        /*System.out.println("Hello World!");*/
        System.out.println("\n*************************************\n");
        System.out.println("\tWelcome to TheDesk \n");
        System.out.println("*************************************");
        optionsSelection();

    }
    private static void optionsSelection() {
        String[] arr = {"1. I wish to review my expenditure",
                "2. I wish to add my expenditure",
                "3. I wish to delete my expenditure",
                "4. I wish to sort the expenditures",
                "5. I wish to search for a particular expenditure",
                "6. Close the application"
        };
        int[] arr1 = {1,2,3,4,5,6};
        int  slen = arr1.length;
        for(int i=0; i<slen;i++){
            System.out.println(arr[i]);
            // display the all the Strings mentioned in the String array
        }
        ArrayList<Integer> arrlist = new ArrayList<Integer>();
        ArrayList<Integer> expenses = new ArrayList<Integer>();
        expenses.add(1000);
        expenses.add(2300);
        expenses.add(45000);
        expenses.add(32000);
        expenses.add(110);
        expenses.addAll(arrlist);
        System.out.println("\nEnter your choice:\t");
        Scanner sc = new Scanner(System.in);
        int  options =  sc.nextInt();
        for(int j=1;j<=slen;j++){
            if(options==j){
                switch (options){
                    case 1:
                        System.out.println("Your saved expenses are listed below:
\n");
                        System.out.println(expenses+"\n");
                        optionsSelection();
                        break;
                    case 2:
                        System.out.println("Enter the value to add your Expense:
\n");
                        int value = sc.nextInt();
                        expenses.add(value);
                        System.out.println("Your value is updated\n");
                        expenses.addAll(arrlist);
```

```
                        System.out.println(expenses+"\n");
                        optionsSelection();

                        break;
                    case 3:
                        System.out.println("You are about the delete all your
expenses! \nConfirm again by selecting the same option...\n");
                        int con_choice = sc.nextInt();
                        if(con_choice==options){
                               expenses.clear();
                            System.out.println(expenses+"\n");
                            System.out.println("All your expenses are erased!\n");
                        } else {
                            System.out.println("Oops... try again!");
                        }
                        optionsSelection();
                        break;
                    case 4:
                        sortExpenses(expenses);
                        optionsSelection();
                        break;
                    case 5:
                        searchExpenses(expenses);
                        optionsSelection();
                        break;
                    case 6:
                        closeApp();
                        break;
                    default:
                        System.out.println("You have made an invalid choice!");
                        break;
                }
            }
        }

    }
    private static void closeApp() {

        System.out.println("Closing your application... \nThank you!");

    }
    private static void searchExpenses(ArrayList<Integer> arrayList) {

       Scanner sc=new Scanner(System.in);
        int leng = arrayList.size();
        System.out.println("Enter the expense you need to search:\t");
        int exp=sc.nextInt();
        boolean check = arrayList.contains(exp);

        if (check)
            System.out.println("Congratulations The list contains " +exp+ "!!
Happy Shopping!!");
        else
            System.out.println("Sorry but the list does not contains " +exp+
".Please enter the correct expenses. ");

    }
    private static void sortExpenses(ArrayList<Integer> arrayList) {
```

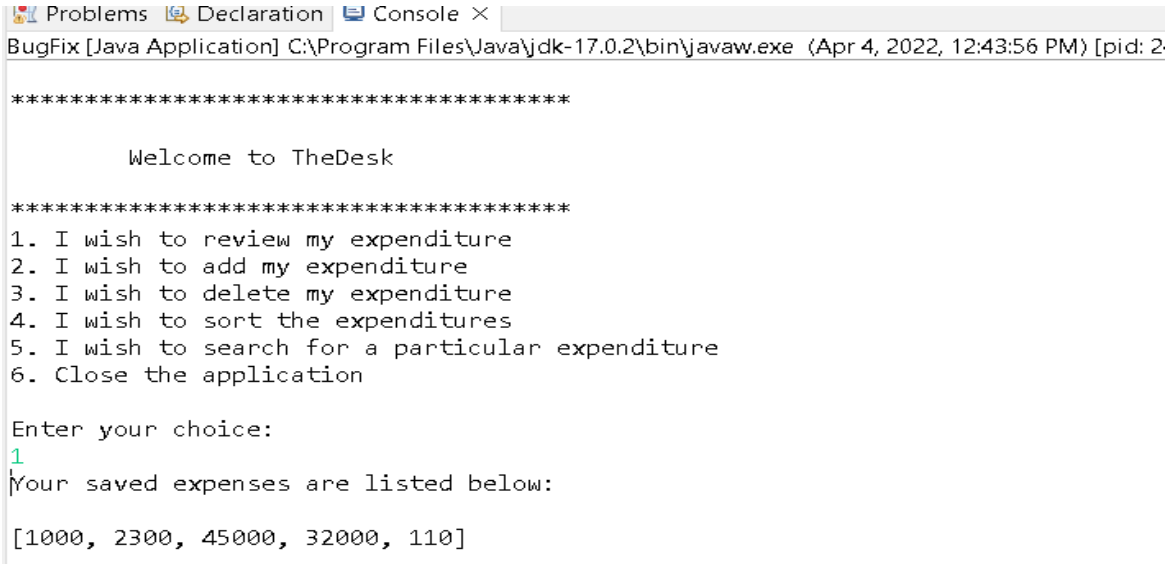
```
        int arrlength =  arrayList.size();
        Collections.sort(arrayList);
        System.out.println("After Sorting: "+ arrayList);

    }

}
```

Output:

Welcome Screen:



1.When the user wants to review its expenditure:



2.When the user wants to add to its expenditure:

i.Here the user added 475 to its expenditure list:

```
BugFix [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe  (Apr 4, 2022, 1:13:54 PM) [pid: 7728]
**************************************

        Welcome to TheDesk

**************************************
1. I wish to review my expenditure
2. I wish to add my expenditure
3. I wish to delete my expenditure
4. I wish to sort the expenditures
5. I wish to search for a particular expenditure
6. Close the application

Enter your choice:
2
Enter the value to add your Expense:

475
Your value is updated

[1000, 2300, 45000, 32000, 110, 475]
```

The expenditure list:

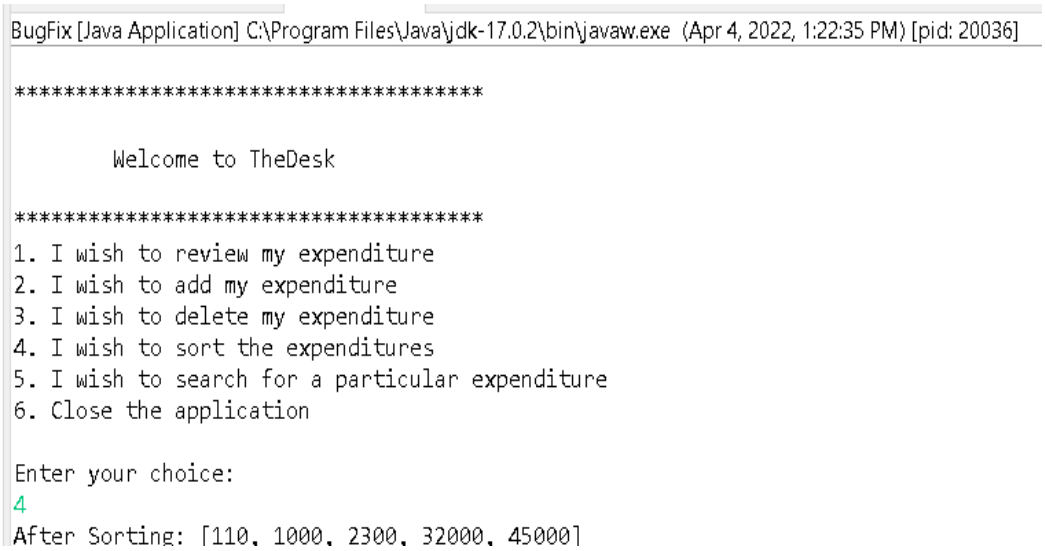
[1000, 2300, 45000, 32000, 110]

3.When the user wants to delete its expenses:

i.Here the entire list has been cleared

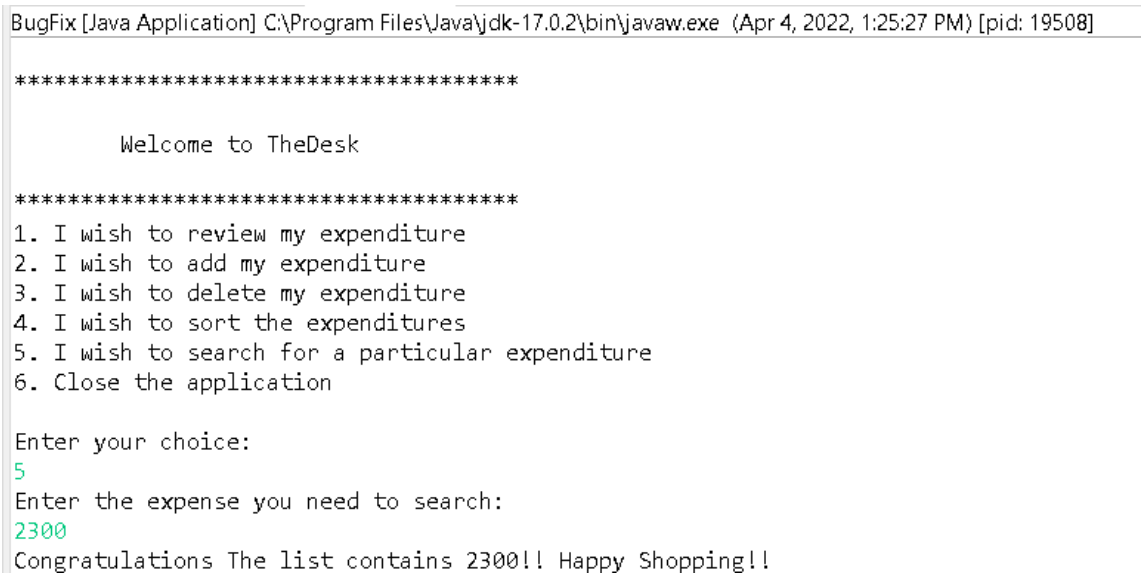
```
BugFix [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe  (Apr 4, 2022, 1:18:40 PM) [pid: 24944]

        Welcome to TheDesk

**************************************
1. I wish to review my expenditure
2. I wish to add my expenditure
3. I wish to delete my expenditure
4. I wish to sort the expenditures
5. I wish to search for a particular expenditure
6. Close the application

Enter your choice:
3
You are about the delete all your expenses!
Confirm again by selecting the same option...

3
[]

All your expenses are erased!
```

4.When the user wants to sort the expenses in ascending order:



5.When the user wants to search the expense from the expenditure list:

i.When the searched element is in the list:



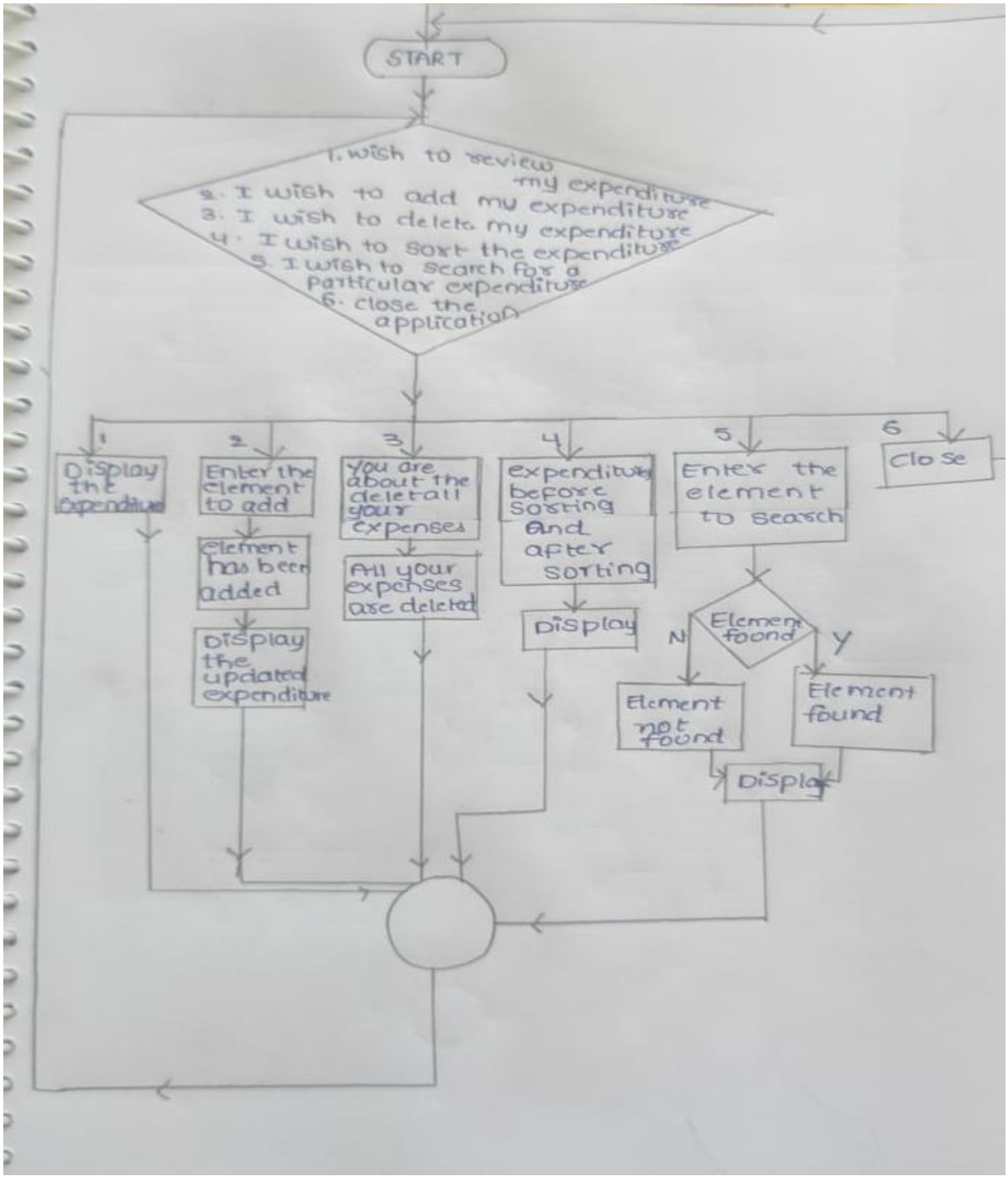
ii.When the searched element is not in the list:

```
BugFix [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe  (Apr 4, 2022, 1:27:31 PM) [pid: 21876]

**************************************

        Welcome to TheDesk

**************************************
1. I wish to review my expenditure
2. I wish to add my expenditure
3. I wish to delete my expenditure
4. I wish to sort the expenditures
5. I wish to search for a particular expenditure
6. Close the application

Enter your choice:
5
Enter the expense you need to search:
2575
Sorry but the list does not contains 2575.Please enter the correct expenses.
```

6.When the user wants to close the application:

```
<terminated> BugFix [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe  (Apr 4, 2022, 1:29:24 PM – 1:29:27 PM) [pid: 4772]

**************************************

        Welcome to TheDesk

**************************************
1. I wish to review my expenditure
2. I wish to add my expenditure
3. I wish to delete my expenditure
4. I wish to sort the expenditures
5. I wish to search for a particular expenditure
6. Close the application

Enter your choice:
6
Closing your application...
Thank you!
```

Flowchart:


START
1. Wish to review my expenditure
2. I wish to add my expenditure
3. I wish to delete my expenditure
4. I wish to sort the expenditure
5. I wish to search for a particular expenditure
6. close the application
1
Display the expenditure
2
Enter the element to add
element has been added
Display the updated expenditure
3
You are about the delete all your expenses
All your expenses are deleted
4
expenditure before sorting and after sorting
Display
5
Enter the element to search
Element found
N
Y
Element not found
Element found
Display
6
Close