|  |
| --- |
| **Bug fix of the application**  **(Sprint work and Project specifications)** |

**Version History:**

|  |  |
| --- | --- |
| Author | Mohammed Musthaq S |
| Purpose | Sprint work and specification of project |
| Date | 4th March 2022 |
| Version | 1.0 |

Contents

[1. Module of the project 3](file:///C:\Users\HP\Desktop\Phase1%20assessment\Documentation\my%20project%20details%20and%20scrum%20details.docx#_Toc79611972)

[2. Sprint wise work: 3](file:///C:\Users\HP\Desktop\Phase1%20assessment\Documentation\my%20project%20details%20and%20scrum%20details.docx#_Toc79611973)

[3. Git hub link: 3](file:///C:\Users\HP\Desktop\Phase1%20assessment\Documentation\my%20project%20details%20and%20scrum%20details.docx#_Toc79611974)

[4. Project code: 4](file:///C:\Users\HP\Desktop\Phase1%20assessment\Documentation\my%20project%20details%20and%20scrum%20details.docx#_Toc79611975)

# Module of the project

* optionsSelection
* closeApp
* searchExpenses
* sortExpenses
* add
* clear

# 2. Sprint wise work:

|  |  |
| --- | --- |
| **Sprint Number** | **Modules** |
| 1 | OptionSelection:  The user has to select the operation to be performed |
| 2 | add:  the user can add new expense in the list |
| 3 | Clear:  The user can delete the expense list |
| 4 | sortExpense:  the user can sort the expense list in acending order |
| 5 | searchExpense:  the user can search the expense amount is present in the list |
| 6 | Close:  The user can close the application |

# 3. Git hub link:

|  |  |
| --- | --- |
| **Repository name** | [Simplilearn-phase-1](https://github.com/musthaqmd/Simplilearn-phase-1) |
| **Repository Link** | <https://github.com/musthaqmd/Simplilearn-phase-1.git> |
| **Folder name** | **Practice Programs ---> Bug fix of the application** |

# 4. Project code:

|  |
| --- |
| **Folder Structure** |
|  |
| **Main.java** |
| // Fix Bugs of the Application  **package** FixBugsOfTheApplication;  **import** java.util.ArrayList;  **import** java.util.Collections;  **import** java.util.Scanner;  **public** **class** Main {  **public** **static** **void** main(String[] args) {    //Add "Hello World" to the greeting message and menu.  System.***out***.println();  System.***out***.println("\tHello World! \n");  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 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!");  }    // method has been added to search particular expense in arraylist  **private** **static** **void** searchExpenses(ArrayList<Integer> arrayList) {  **int** leng = arrayList.size();  System.***out***.println("Enter the expense you need to search:\t");  Scanner sc = **new** Scanner(System.***in***);  **int** expense = sc.nextInt();  **for** (**int** i = 0; i < leng; i++) {  **if** (arrayList.get(i) == expense) {  System.***out***.println("Found the expense " + expense + " at " + i + " position");  }  }  }  //method has been added to sort the expenses in ascending order  **private** **static** **void** sortExpenses(ArrayList<Integer> arrayList) {  **int** arrlength = arrayList.size();  Collections.*sort*(arrayList);  System.***out***.println("Your sorted expenses in an ascending order:");  System.***out***.println(arrayList);  System.***out***.println();  }  } |