**Fix Bugs of the Application**

**Description:-** In a FixBugApp by taking expenditure and perform retrieve, delete, add, sort, search operations on it by using loops and switch cases. In this app sorting and searching algorithm are also used.

**Following are the steps for fix bug app**

Step1:- Inside the main function the following display occur

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:

Step2:- Entering the choice as per switch cases the following operation takes place

Case1- saved expanses are listed

Case2- add expanses and update it

Case3- delete expenses

Case4- Sort expanses

Case5- Search expanses

Case6- close application

Otherwise “You have made invalid choice”

Step3:- After the execution of one of the cases “Closing your application” should be display

**Sprint Planning:-**

|  |  |
| --- | --- |
| Analysis | Day 1 |
| Design | Day 2 |
| Implementation | Day 3 |
| Test | Day 4 |
| Report | Day 5 |

**Algorithm:-**

public class BugsFixApp {

public static void main() {

“Welcome to TheDesk”

}

private static void {

{

"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"

};

{

(arr[i]);

}

"\nEnter your choice:\t"

for(int j=1;j<=slen;j++){

if(options==j){

switch (options){

case 1:

"Your saved expenses are listed below:”

break;

case 2:

"Enter the value to add your Expense: \n"

"Your value is updated\n"

break;

case 3:

"You are about the delete all your expenses! \nConfirm again by selecting the same option...\n”

"All your expenses are erased!\n"

} else {

"Oops... try again!"

}

break;

case 4:

sortExpenses

break;

case 5:

searchExpenses

break;

case 6:

closeApp();

break;

default:

"You have made an invalid choice!"

break;

}

}

}

}

private static void closeApp() {

“Closing your application... \nThank you!"

}

private static void searchExpenses

"Enter the expense you need to search:\t"

//Linear Search

for(int i=0;i<leng;i++) {

if(arrayList.get(i)==input) {

"Found the expense " + input + " at " + i + " position");

}

}

}

**private** **static** **void** sortExpenses

//Complete the method. The expenses should be sorted in ascending order.

Collections.*sort*();

"Sorted expenses: "

}

}

}