

A Mini Project Report  
On

## **SPLIT BILLS**

CLASS: B.E 2/4 CSE AIML Sem: III

By

**RITHI BHAMIDIPATI**

**16020-21-748-036**

**VAMSI KRISHNA DESINEEDI**

**1602-21-748-059**



**Department of Computer Science & Engineering**

**Vasavi College of Engineering (Autonomous)**

**Ibrahimbagh, Hyderabad-31 2023**

## ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people made it possible and whose encouragement and guidance have made our efforts with success.

We are indebted to the **Internal Guide, Mrs. E.Shailaja**, ASSOCIATE PROFESSOR, Computer Science & Engineering, VASAVI COLLEGE OF ENGINEERING, Ibrahimbagh, hyd-31 for her support and guidance throughout the project.

We are also indebted to the **Head of the Department, Dr. T. Adilakshmi**, PROFESSOR, Computer Science and Engineering, VASAVI COLLEGE OF ENGINEERING, Ibrahimbagh, hyd-31 for her support and guidance throughout the project.

We extend our deep sense of gratitude to the **Principal, S. V. RAMANA RAO**, VASAVI COLLEGE OF ENGINEERING, Ibrahimbagh, hyd-31 permitting us to undertake this project.

## TABLE OF CONTENTS

| <b>S. no.</b> | <b>Content</b>                             | <b>Page no.</b> |
|---------------|--|-----------------|
| I             | Abstract                                   | 5               |
| 1.            | Introduction                               | 6               |
|               | 1.1 Overview of the project                | 6               |
|               | 1.2 Motivation                             | 6               |
|               | 1.3 Problem definition                     | 6               |
|               | 1.4 Objectives                             | 6               |
|               | 1.5 Concepts involved                      | 7               |
|               | 1.6 Scope                                  | 7               |
| 2.            | Software Requirements Specifications (SRS) | 8               |
| 3.            | Sample Code                                | 9               |
| 4.            | Result                                     | 13              |
| 5.            | Conclusion and Future Scope                | 15              |
| 6.            | References                                 | 15              |

## **List Of Figures**

| <b>S. no.</b> | <b>Name of the Figure</b>                   | <b>Page no.</b> |
|---------------|---|-----------------|
| 3.1           | Output Screen when u run the code           | 13              |
| 3.2           | Output Screen after addition of few events  | 13              |
| 3.3           | Final Output-Splitting up of Total Expenses | 14              |

## **I. ABSTRACT**

Split Bills is a financial management system that helps groups to easily share and track expenses. This Allows users to create a group and add expenses, which are then used to calculate the amount that each member owes. Split Bills is a useful tool for anyone looking to keep track of shared expenses and payments in an organized and efficient way. It Can be used with multiple people and across multiple trips. Avoids awkward conversations about money. Calculating who owes what to whom. Tracking shared expenses and debts.

# **1. INTRODUCTION**

## **1.1 OVERVIEW OF THE PROJECT:**

Split Bills is a tool used mainly to split Group Expenses among Friends or Roommates on daily basis or over few Days. We have used Java programming language. It can also be used to Split Group Expenses when group of friends are going on a trip for few days. We take the Number of People as the input and number of people on trip along with their names and they can add events and specify who is paying how much amount.

## **1.2 MOTIVATION:**

We, as a team, have thought of taking up the challenge to contribute our learning from education to serve the humanity and try to protect it from adversities. Being a computer science student, we have the ability to shape the future towards the right path and a golden future.

We have applied our knowledge, techniques taught by our beloved faculty and also understood how the programming languages are used in real life solutions.

## **1.3 PROBLEM DEFINITION:**

Split Bills aims to solve the problem of accurately and fairly splitting expenses among a group of people, such as roommates, friends, or family members. This can often be a difficult and time-consuming task, as people may forget who paid for what or disagree on how to split the costs. It also allows for easy reconciliation and reporting, making it simple for everyone to stay on top of expenses and avoid any confusion or disagreements and allows users to easily see who owes what and to whom.

## **1.4 OBJECTIVES:**

The main objective of this project is to Split Bills among a Group as it involves many people and many Events, there is high possibility of confusion at the End. To prevent this, we developed this tool.

## **1.5 CONCEPTS INVOLVED:**

This project has been implemented with the usage of “Java” language which is an object-oriented language and also considered to be one of the most powerful languages.

## **1.6 SCOPE:**

This project can only be used to Split Group expenses at present and can be developed more in the future with many features like Settling debts and tracking payments, Ability to export data to CSV for tracking and budgeting and many more.

## **2. SOFTWARE REQUIREMENTS SPECIFICATIONS**

### Hardware Specifications

- Windows: 7 or newer
- Processor: Minimum 1 GHz; Recommended 2GHz or more
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above
- Hard Drive: Minimum 16 GB; Recommended 32 GB or

### more Software Specifications

- Java Virtual Machine (JVM), which is usually a software-based interpreter.



### 3. SAMPLE CODE

Main.java

```
import java.util.*;
public class Main
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int x,y,i,j;
        System.out.println("Enter no. of Friends:");
        x=sc.nextInt();
        System.out.println("Enter their names:");
        Player arr[]= new Player[x];
        int gpaid[] = new int[x];
        for(i=0;i<x;i++)
        {
            arr[i] = new Player(sc.next(),x);
        }
        System.out.println("\n\n\t\t*****\n\n");
        for(i=0;i<x;i++)
        {
            System.out.println(arr[i].n+": "+(i+1));
        }
        System.out.println("\n\n");
        int k,d=0,t;
        j=1;
        while(1==1)
        {
            System.out.println("\n\n\t\t*****\n\n");
            System.out.println("EVENT-"+j);
            System.out.println("Who is paying?");
            y=sc.nextInt();
            System.out.println("Enter Amount:");
            k = (sc.nextInt());
            arr[y-1].eval=arr[y-1].eval-k;
            System.out.println("Who all are getting paid?\n(-1:exit,0:all)");
            while((t=sc.nextInt())!=-1)
            {
                if(t==0)
                {
                    for(i=0;i<x;i++)
                    {
                        arr[i].eval=arr[i].eval+ (k/x);
                    }
                    break;
                }
                else
                {
                    gpaid[t]=1;
                    d++;
                }
            }
        }
    }
}
```

```

        if(t!=0)
    {
        for(i=0;i<x;i++)
        {
            if(gpaid[i]==1)
            {
                arr[i].eval=arr[i].eval + (k/d);
            }
        }
    }
    for(i=0;i<x;i++)
    {
        gpaid[i]=0;
    }
    d=0;
    System.out.println("Enter 0 to exit, any other to continue:");
    if (sc.nextInt()!=0)
    {
        j++;
    }
    else
    {
        break;
    }
}
for(i=0;i<x;i++)
{
    if(arr[i].eval==0)
    {
        continue;
    }
    else if(arr[i].eval>0)
    {
        for(j=i+1;j<x;j++)
        {
            if(arr[j].eval<0)
            {
                if(arr[i].eval>(-arr[j].eval))
                {
                    arr[i].eval=arr[i].eval+arr[j].eval;
                    arr[i].a[j]=- (arr[j].eval);
                    arr[j].a[i]=(arr[j].eval);
                    arr[j].eval=0;
                }
            }
            else
            {
                arr[i].a[j]=(arr[i].eval);
                arr[j].a[i]=- (arr[i].eval);
                arr[j].eval=arr[j].eval+arr[i].eval;
                arr[i].eval=0;
            }
        }
    }
}

```

```

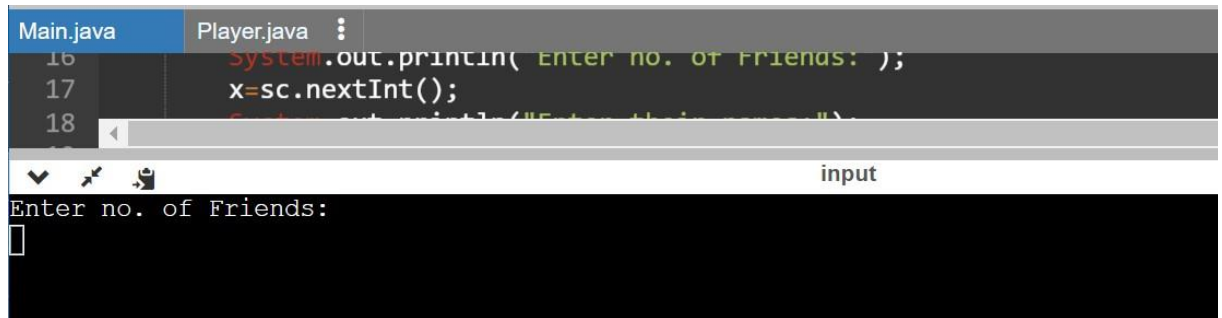
        if(arr[i].eval==0)
        {
            break;
        }
    }
}
else
{
    for(j=i+1;j<x;j++)
    {
        if(arr[j].eval>0)
        {
            if(arr[i].eval>(-arr[j].eval))
            {
                arr[j].eval=arr[j].eval+arr[i].eval;
                arr[j].a[i]=-(arr[i].eval);
                arr[i].a[j]=(arr[i].eval);
                arr[j].eval=0;
            }
            else
            {
                arr[j].a[i]=(arr[j].eval);
                arr[i].a[j]=-(arr[j].eval);
                arr[i].eval=arr[i].eval+arr[j].eval;
                arr[j].eval=0;
            }
            if(arr[i].eval==0)
            {
                break;
            }
        }
    }
}
}
for(i=0;i<x;i++)
{
    System.out.println(arr[i].n+":-");
    for(j=0;j<x;j++)
    {
        if(arr[i].a[j]!=0)
        {
            if(arr[i].a[j]<0)
            {
                System.out.println(arr[j].n+" has to pay "+-(arr[i].a[j])+" to "+arr[i].n+".");
            }
            else if(arr[i].a[j]>0)
            {
                System.out.println(arr[j].n+" will receive "+(arr[i].a[j])+" from "+arr[i].n+".");
            }
        }
    }
}
}
}
}

```

## Player.Java

```
public class Player
{
    int eval;
    String n;
    int a[];
    Player(String name,int x)
    {
        n=name;
        eval=0;
        a = new int[x];
    }
}
```

## 4. RESULT:



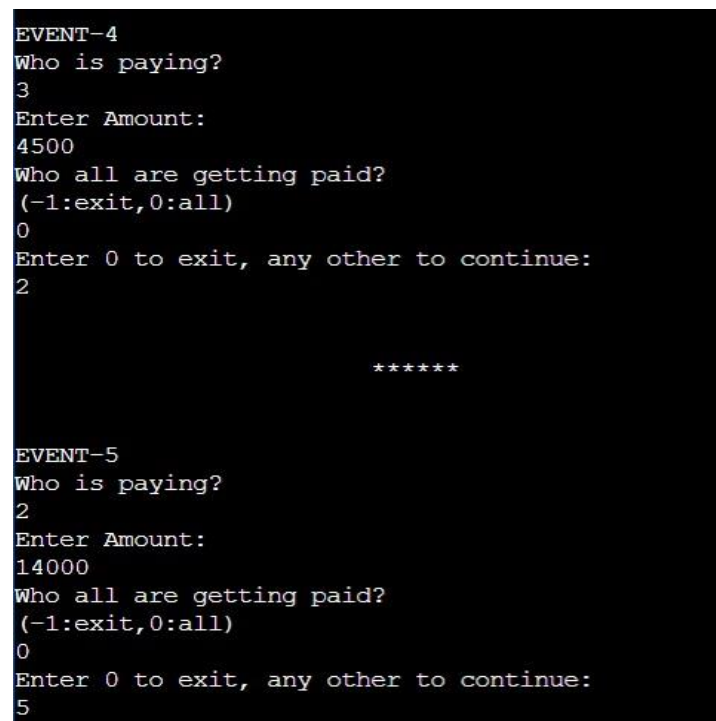
The screenshot shows a Java IDE with two tabs: 'Main.java' and 'Player.java'. The 'Player.java' tab is active, displaying the following code:

```
16 System.out.println( "Enter no. of Friends: ");
17 x=sc.nextInt();
18 sc.nextPrintln("Enter the amount:");
```

Below the code editor, there is a console window titled 'input'. It shows the output of the program:

```
Enter no. of Friends:
1
```

Fig 4.1. Output Screen when u run the code



The screenshot shows a terminal window with the following output:

```
EVENT-4
Who is paying?
3
Enter Amount:
4500
Who all are getting paid?
(-1:exit,0:all)
0
Enter 0 to exit, any other to continue:
2

*****

EVENT-5
Who is paying?
2
Enter Amount:
14000
Who all are getting paid?
(-1:exit,0:all)
0
Enter 0 to exit, any other to continue:
5
```

Fig 4.2. Output Screen after addition of few events

```
EVENT-10
Who is paying?
1
Enter Amount:
50000
Who all are getting paid?
(-1:exit,0:all)
0
Enter 0 to exit, any other to continue:
0
vamsi:-
krishna has to pay 12152 to vamsi.
rithi has to pay 9946 to vamsi.
krishna:-
vamsi will receive 12152 from krishna.
ram:-
rithi:-
vamsi will receive 9946 from rithi.
```

Fig 4.3. Final Output-Splitting up of Total Expenses

## **5. CONCLUSION AND FUTURE SCOPE**

- Hence, we have successfully developed Split Bills using Java.
- Further, we can also try to add some more features to this project such as option for receipt Scanning and Personal Finance Tracking.
- Through this project we learnt how to use Java in real life applications.
- A feature of Personal Tracking.
- Option to view the history of expenses and debts.
- Option to connect with bank account and track expenses automatically.

## **6. REFERENCES**

1. <https://www.youtube.com/watch?v=EnaIWAitmEU>