A Project Report

on

Expense Audit Application

Submitted in partial fulfillment of the
requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

GALGOTIAS
UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Dr. Vishwadeepak Singh Baghela
Professor

Submitted By

18021011609 – AMAN PATHAK
18021180023 -ASHISH PATHAK

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING /
DEPARTMENT OF COMPUTERAPPLICATION
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA

DECEMBER, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE’S DECLARATION

I/We hereby certify that the work which is being presented in the the project, entitled **“Expense Audit Application ”** in partial fulfillment of the requirements for the award of the **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING** –submitted in the **School of Computing Science and Engineering** of Galgotias University, Greater Noida, is an original work carried out during the period of **JULY-2021 to DECEMBER-2021**, under the supervision of **Dr. Vishwadeepak Singh Baghela, Professor, Department of Computer Science and Engineering** of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

18SCSE1010378 – AMAN PATHAK
18SCSE1180024 – ASHISH PATHAK

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor

Dr. Vishwadeepak Singh Baghela

Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **18SCSE1010378 – AMAN PATHAK, 18SCSE1180024 - ASHISH PATHAK** has been held on **_________________** and his/her work is recommended for the award **of BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

Acknowledgement

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude we give to our project Supervisor, Dr.Vishwadeepak Singh Baghela, whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this report. Furthermore, we would also like to acknowledge with much appreciation the crucial role of the staff of Prime College, who gave the permission to use all required equipment and the necessary materials to complete the task. We are thankful and fortunate enough to get constant support from our seniors and every teaching staff of B.Tech, department which helped us successfully complete our project. We would also like to extend our regards to all the non-teaching staff of SCSE department for their timely support. Special thanks go to my team mates Ashish Pathak, who helped me to assemble the parts and gave suggestion about the task of designing, developing and documentation. Last but not least, many thanks go to the guide of the project, Dr.Vishwadeepak Singh Baghela who have invested her full effort in guiding the team in achieving the goal. We have to appreciate the guidance given by other supervisor as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comment and advices. Our thanks and appreciations also go to each and every one of our colleagues for their encouragement and support in developing the project.

Abstract

This Expense Audit Application is a Software that can be installed in Computer. It is created in JAVA language using NetBeans IDE. This application will track all the spending records of daily life. In this you can add the spending category wise and by Date. You can add the amount of your spending. And on first screen it will show last 20 days Spending as a table form by id, Date, category and amount of your last 20 days Spending. And you can delete any data of your spending of last 20 days directly from here. And it will show the total amount of last 20 days spending. In the other window it will show a box in which you can add your category and view all your category which is added by You and also remove the category. In the next window u can view your spending date wise between two dates and total amount of date wise spending. And in that in another part you can also see your spending category wise between two days, And It will sow the total amount. In the menu bar all the options are there to open all these windows and exit option also there. It is using JDBC for Database Connectivity using MySQL database.

Hence this application will be useful to Record the spending and the amount of your Spending

Contents

List of Figures

Acronyms

B.Tech.	Bachelor of Technology
BCSE	Bachelor of Science Engineering
JSE	Java Standard Edition
SCSE	School of Computing Science and Engineering
JEE	Jakarta Enterprise Edition

CHAPTER-1

Introduction

1.1 Background

Expense Audit Application is a refined system which allows user to efficiently manage his/her spendings with ease. Tracking spendings daily can really help to us save lot of money. Once we start off by tracking our spendings each day, we will be able to get a better idea where you are spending your money, so you stay in control and achieve your goal. It will be able to generate your expense and saving report as time duration you selected. There will be a reminder that will help to save money for your pre-defined spendings.

1.2 Literature Reviews/Comparative study

To create this software, we will use Java Programming Language. MySQL to store the data. And JDBC to connect the database to our software.

Programming Language-

Our application is developed in JAVA Programming language.

Java programming language was originally developed by Sun Microsystems which was initiated by James Gosling and released in 1995 as core component of Sun Microsystems' Java platform (Java 1.0 [J2SE]).

The latest release of the Java Standard Edition is Java SE 8. With the advancement of Java and its widespread popularity, multiple configurations were built to suit various types of platforms. For example: J2EE for Enterprise Applications, J2ME for Mobile Applications.

The new J2 versions were renamed as Java SE, Java EE, and Java ME respectively. Java is guaranteed to be **Write Once, Run Anywhere.**

Java is −

- **Object Oriented** − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- **Platform Independent** − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.
- **Simple** − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
- **Secure** − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- **Architecture-neutral** − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

- **Portable** − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

- **Robust** − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.

- **Multithreaded** − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.

- **Interpreted** − Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.

- **High Performance** − With the use of Just-In-Time compilers, Java enables high performance.

- **Distributed** − Java is designed for the distributed environment of the internet.

- **Dynamic** − Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

Database Software-

We will use MySQL to store the data.

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

- Relational Database **Management** System (RDBMS) MySQL is a relational database **management** system. ...
- Easy to use. MySQL is easy to use. ...
- It is secure. MySQL consists of a solid data security layer that protects sensitive data from intruders. ...
- Client/ Server **Architecture**. ...
- Free to download. ...
- It is scalable. ...
- Speed. ...
- High Flexibility.

Java JDBC-

We can use JDBC API to access tabular data stored in any relational database. By the help of JDBC API, we can save, update, delete and fetch data from the database.

1.3Problem Formulation

Every earning people are mostly obsessed at the end of the month as the they cannot remember where all of their money has gone when they have spent and ultimately have to sustain in little money minimizing their essential needs. There is no as such complete solution present easily or we should say free of cost which enables a person to keep a track of its daily expenditure easily and notify them if they are going to have money shortage. To do so a person has to keep a log in a diary or in a computer, also all the calculations needs to be done by the user which may sometimes results in errors leading to losses. Due to lack of a complete tracking system, there is a constant overload to rely on the daily entry of the expenditure and total estimation till the end of the month.

This software is to solve this problem, by using this software we can record our spending record category wise, amount, and Date also. We can review all our spending we can add, delete see out spending date wise category wise also. For this problem we created this Software.

When we will install this software on the first window, we can see the format to add amount, Date, Category and we can add out spending.

Last 20 days Spending will be shown on the first window. And there is a button to delete our category and amount. Last 20 days total amount will be shown there.

1.4Objectives

The objective of this system is:

a) To keep track of daily spendings and budgeting;

b) To save money for pre-defined spendings which will help planning on your future investments

1.5 Scope and Limitations

1.4.1 Scope

This application can take a good market as it is usable by anyone who are willing to manage their spendings and aiming to save for the future investments and many more. There is not any range criteria or any kind of profession or gender are focused, it will used hugely. 1.4.2

1.4.2 Limitations

- User have to entry every record manually.
- The category divided may be blunder or messy.
- Person who is handling system must have some technnical knowledge.

1.6 Report Organization

Chapter 2:

This chapter covers all the history, methods, requirement specification and feasibility analysis and structured system requirements.

Chapter 3:

Design of DET project is explained in detail with all the necessary diagrams and brief functionality.

Chapter 4:

Process of implementation and testing is described along with all the tools used for the development. 4

Chapter 5:

Conclusion and future scope of the application are explained.

Chapter 6:

The previous reports which helped in our project are listed

CHAPTER-2

Literature Survey/Project Design

To create this software, we will use Java Programming Language. MySQL to store the data. And JDBC to connect the database to our software.

Programming Language-

Our application is developed in JAVA Programming language.

Java programming language was originally developed by Sun Microsystems which was initiated by James Gosling and released in 1995 as core component of Sun Microsystems' Java platform (Java 1.0 [J2SE]).

The latest release of the Java Standard Edition is Java SE 8. With the advancement of Java and its widespread popularity, multiple configurations were built to suit various types of platforms. For example: J2EE for Enterprise Applications, J2ME for Mobile Applications.

The new J2 versions were renamed as Java SE, Java EE, and Java ME respectively. Java is guaranteed to be **Write Once, Run Anywhere.**

Java is −

- **Object Oriented** − In Java, everything is an Object. Java can be easily extended since it is based on the Object model.
- **Platform Independent** − Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

- **Simple** − Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.
- **Secure** − With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.
- **Architecture-neutral** − Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.
- **Portable** − Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.
- **Robust** − Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking.
- **Multithreaded** − With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.
- **Interpreted** − Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.
- **High Performance** − With the use of Just-In-Time compilers, Java enables high performance.

- **Distributed** − Java is designed for the distributed environment of the internet.
- **Dynamic** − Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

Database Software-

We will use MySQL to store the data.

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

- Relational Database **Management** System (RDBMS) MySQL is a relational database **management** system. ...
- Easy to use. MySQL is easy to use. ...
- It is secure. MySQL consists of a solid data security layer that protects sensitive data from intruders. ...
- Client/ Server **Architecture**. ...
- Free to download. ...
- It is scalable. ...
- Speed. ...
- High Flexibility.

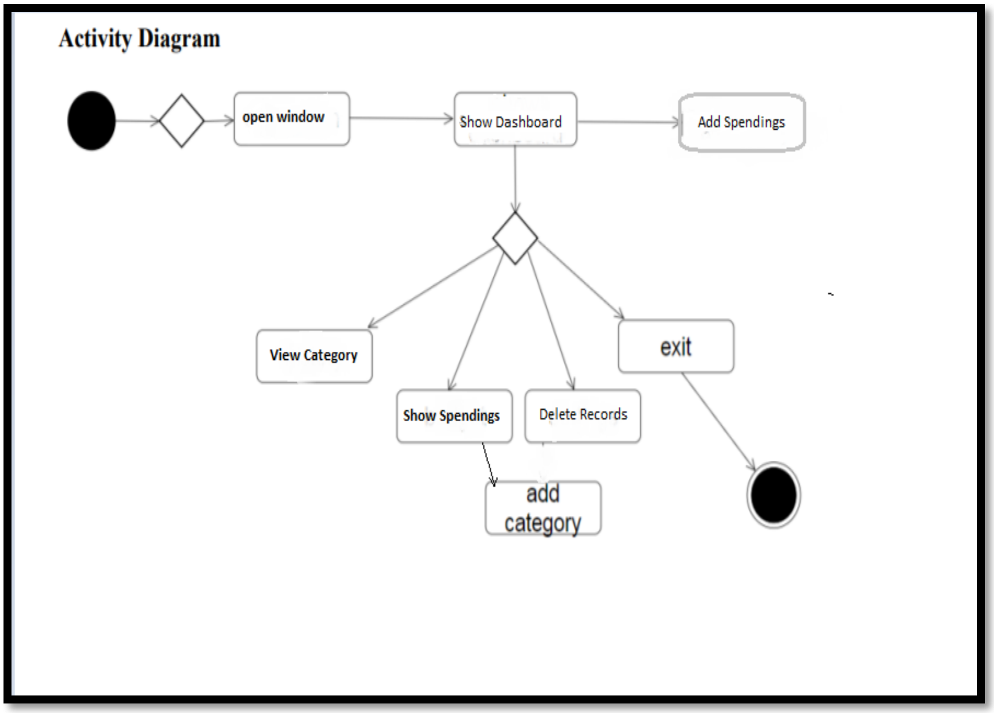
Java JDBC-

We can use JDBC API to access tabular data stored in any relational database. By the help of JDBC API, we can save, update, delete and fetch data from the database.

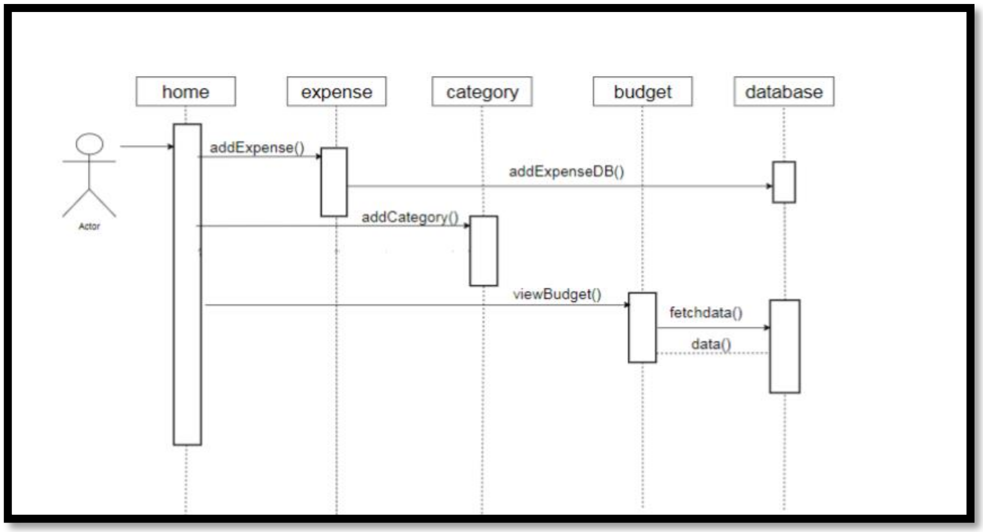
\

Architecture Diagram for Proposed method

Activity Diagram:



Sequence Diagram:



Sequence diagram

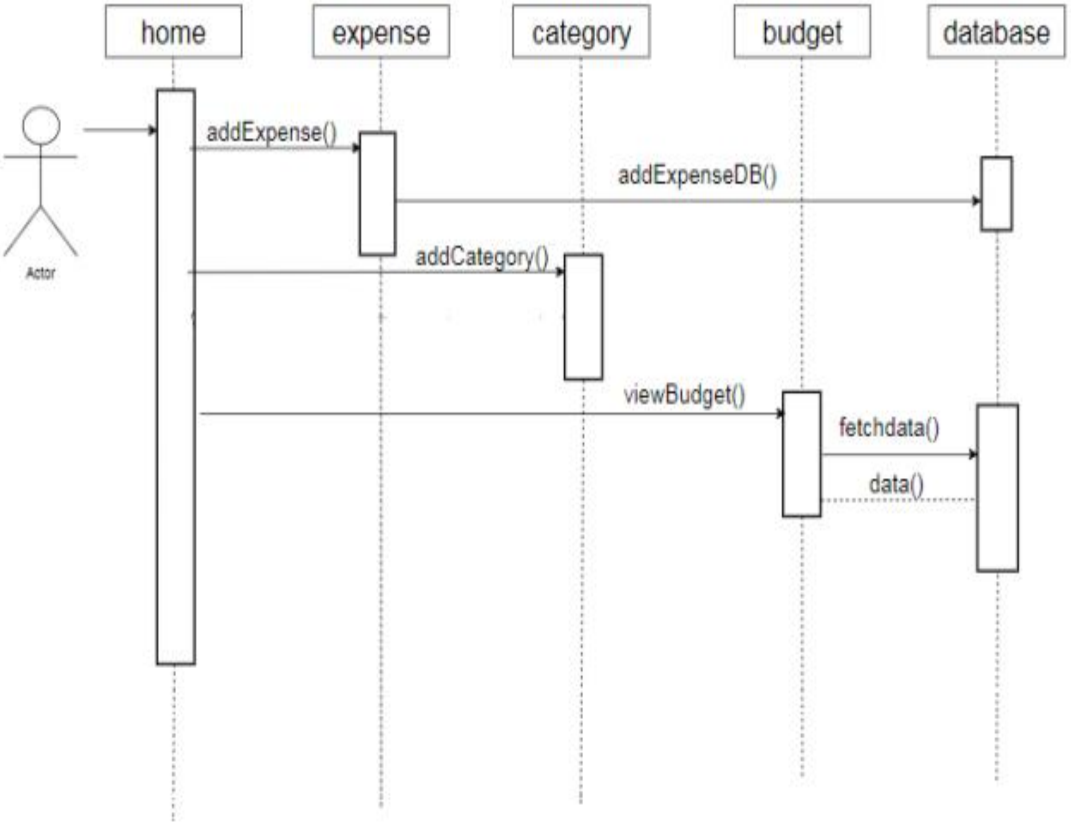
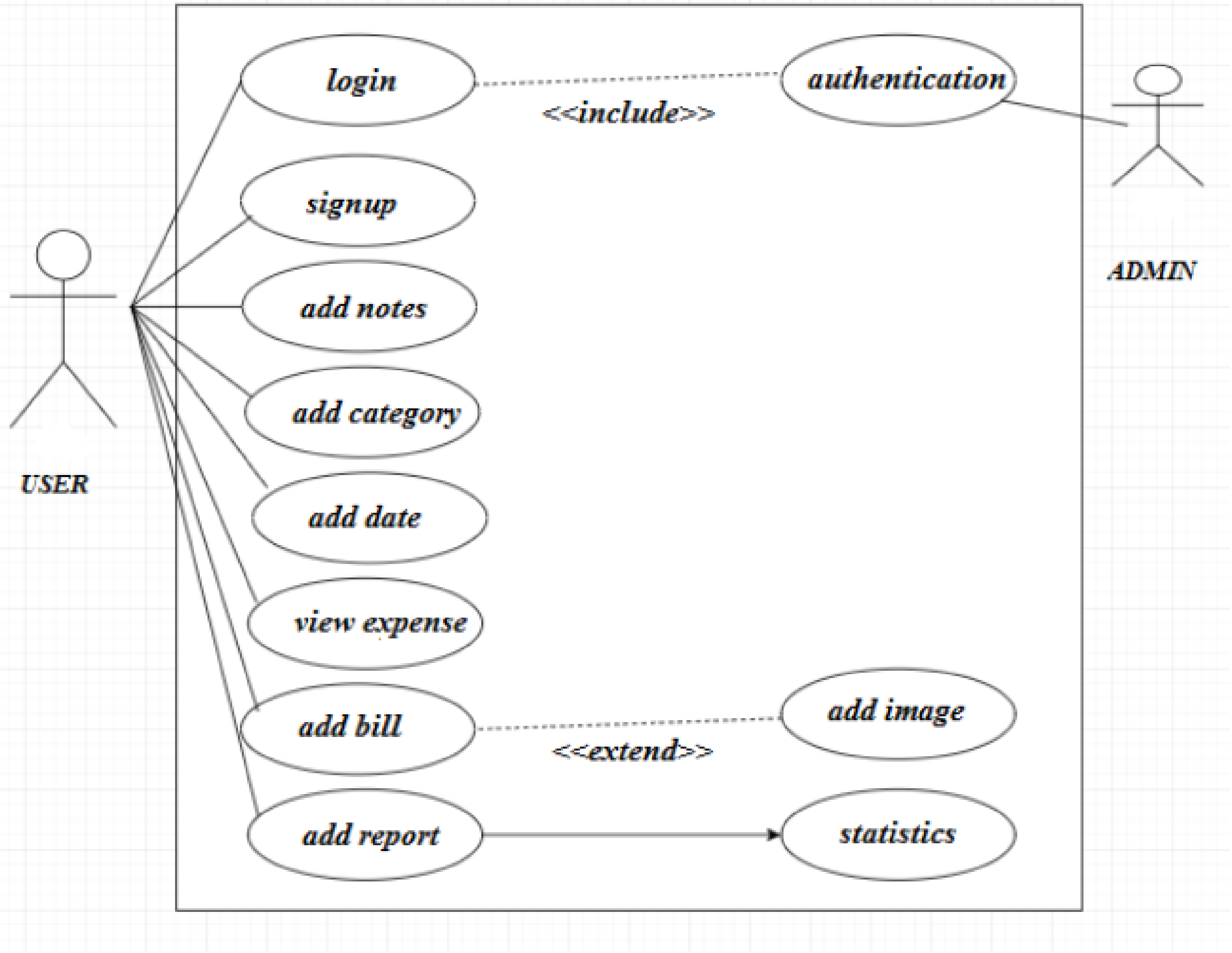


Figure 8: Sequence Diagram

Use Case Diagram



Non-Functional requirements

1. Usability There is a consistency in all the modules and webpages. To ease the navigation there is a back tab to provide access to previous page. There is proper instruction on each page. 2. Reliability Each data record is stored on a well-built efficient database schema. There is no risk of data loss. The internal evaluation of data is well coded. 3. Supportability The system is well built to support any machine. Maintainability of the system is easy. 4. Performance In order to ease the accessibility, the types of spendings are categorized along with an option to name on the own.

Throughput of the system is increased due to light weight database support. 5. Availability The system is available all the time, no time constraint.

2.3 Feasibility Analysis

2.4.1 Technical Feasibility

This assessment focuses on the technical resources available. It helps to determine whether the technical team is capable of converting the ideas into working systems. It also involves evaluation of the hardware, software and other technology requirements of the proposed system.

Hardware Specification:

- 10 MB memory

Software Specification:

- Front End : XHTML
- Back End : Java and SQlite

Chapter 3
Functionality/Working of Project

3.1 System design and summary

We have developed the specified system that works while not net. To use this technique, we want an information, app and therefore the user. algorithmic program Used algorithmic program for mining frequent item sets for Boolean association rules. Apriorism uses a "bottom up" approach, wherever frequent subsets square measure extended one item at a time (a step referred to as candidate generation, and teams of candidate’s square measure tested against the info. JDBC is intended to control on information containing transactions (for example, collections of things bought by customers, or details of a web site frequentation).

3.2-System style

Systems style is that the method of shaping the design, modules, interfaces, and knowledge for a system to satisfy nominal needs. Systems style can be seen because the application of systems theory to development.

Sequence diagram

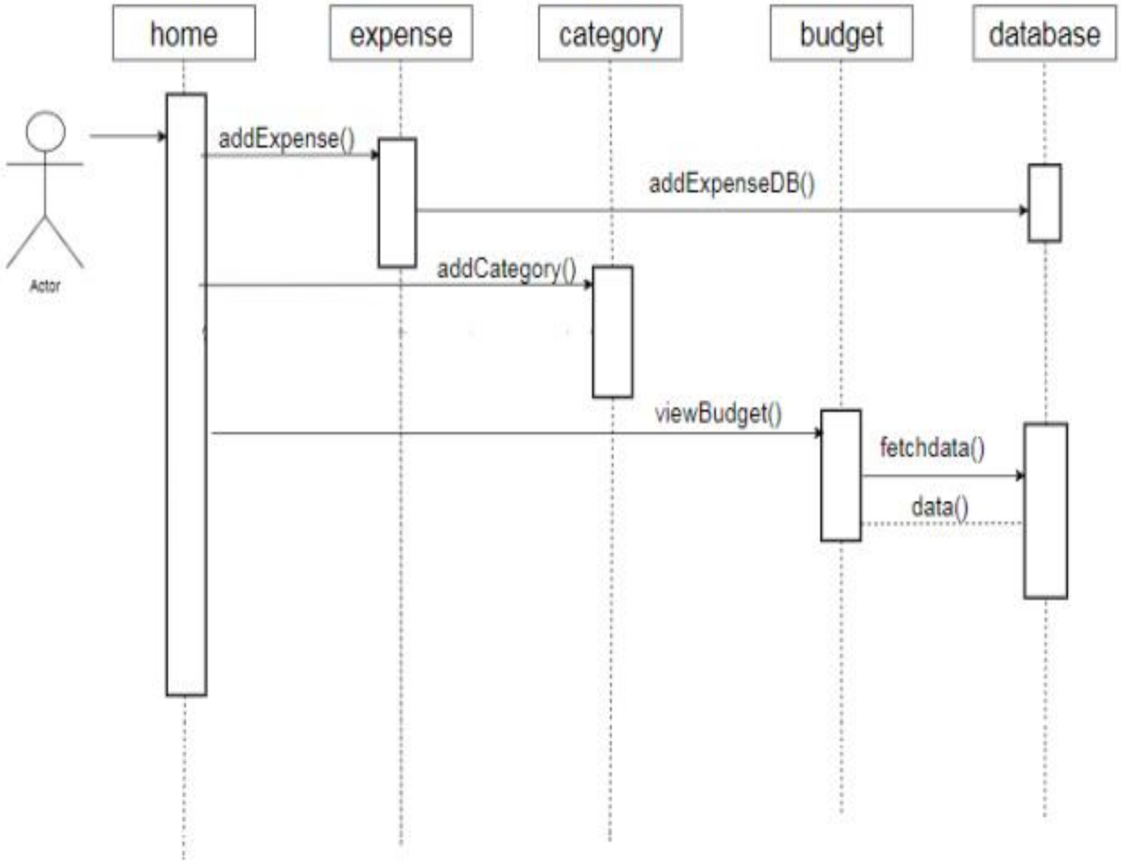
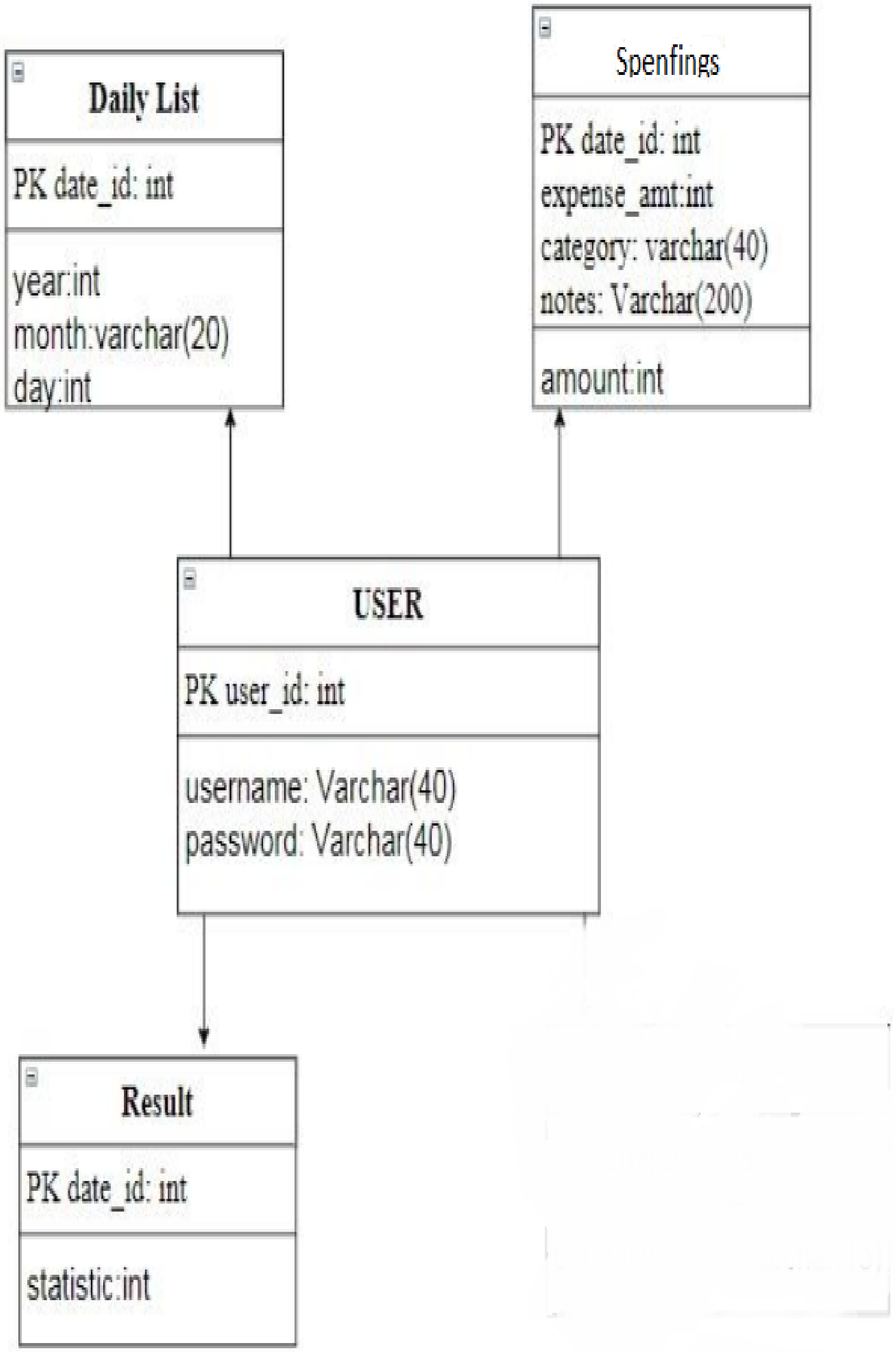


Figure 8: Sequence Diagram

3.2.1 Database Schema



Integration Testing

The Integration testing a part of a testing methodology is that the testing of the various modules/components that are with success unit tested once integrated along to perform specific tasks and activities. The check is usually done on each the interfaces between the elements and therefore the larger structure being created, if its quality property cannot be assessed from its elements. when integration the necessities, we have a tendency to tested it, it absolutely was fine and satisfactory.

System Testing

The system testing a part of a testing methodology involves testing the whole system for errors and bugs. This check is dole out by interfacing the hardware and computer code elements of the whole system, so testing it as an entire.

Overview Process Model Used

The body of water model could be a ordered approach, wherever every basic activity of a method painted as a separate section, organized in linear order. In the body of water model, you need to set up and schedule all of the activities before beginning functioning on them (plan driven process). body of water model is employed for the project as a result of all the necessities square measure clear as this project isn't addressing the shoppers and thus beforehand coming up with may be created concerning the way to do every section of development.

4.2 Tools Used

4.2.1 Back End Tools

JAVA: Java could be a all-purpose computer-programming language that's coinciding, class-based, object-oriented, and specifically designed to possess as few implementation dependencies as potential. because the back-end tools java is employed to supply practicality to the attributes displaying in UI.

MySQL:

It is an electronic information service management system contained in an exceedingly C programming library. In distinction to several different management systems, SQLite isn't a client–server information engine. Rather, it's embedded into the tip program. So, SQLITE 3.8.2 is employed for manipulating knowledge from information and visualizing to the user.

A modularization consists of well-defined manageable units with well-defined interfaces among the units.

Desirable property of modular system includes

a) Each module is a well-defined sub-system.

b) Single, well – defined purpose of each module.

c) Modules can be separately compiled and stored in a library.

d) Modules can use another module.

e) Modules should be easier to use than to build.

f) Modules should be simpler from outside then from inside.

The project can be decomposed in following modules:

- **Add Amount date wise and Category wise**: This module is responsible to enable the user to add a new bill. And it will show all last recent spending also.
- **Add Category or Delete Category**: This module is responsible to add a new Category or Delete a category.
- **Delete the bill**: This module is responsible for the pre-defined bill.
- **View Spending**: This module is responsible for viewing all the spendings in detail added to category wise and Date wise.
- **Edit Module**: This module is responsible for editing a pre-defined bill.
- **Categories module**: This module is responsible for various options. In this app users have options of selecting various basic expense categories and currency according to which were added by User.

The classes used for this project are:

- Category.java
- SpendingTracker.java
- ViewSpending.java

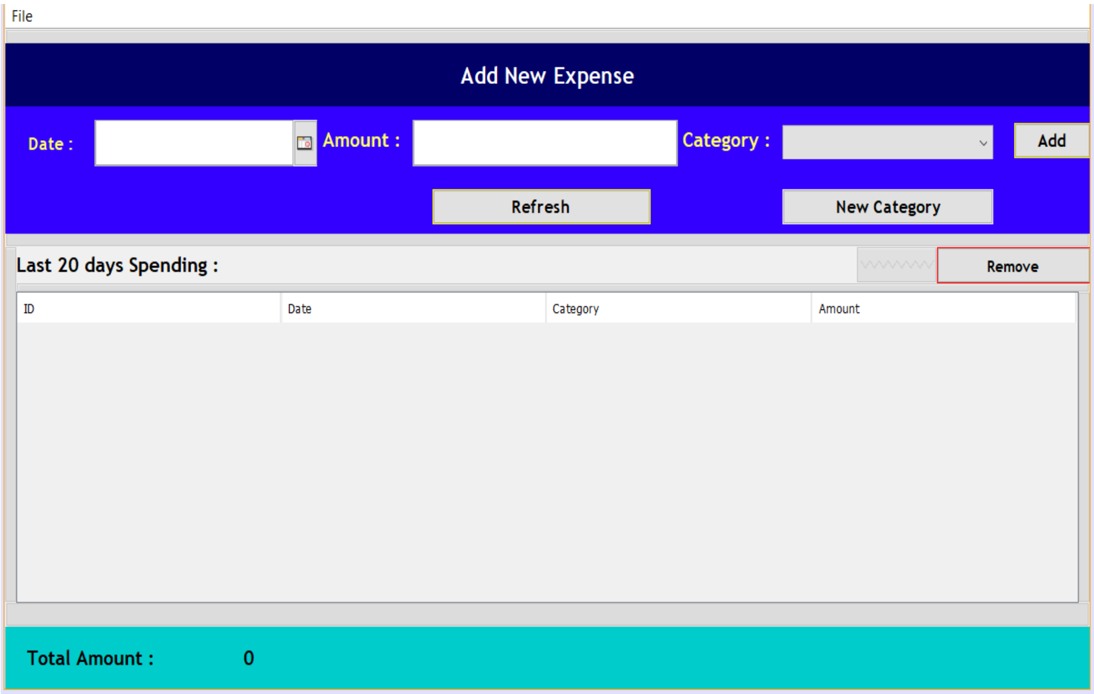
Methods are:

- getEntries()
- displayCategory()
- Many Swings and AWT Methods

System style

Systems style is that the method of shaping the design, modules, interfaces, and knowledge for a system to satisfy nominal needs. Systems style can be seen because the application of systems theory to development.

SpendingTracker.java

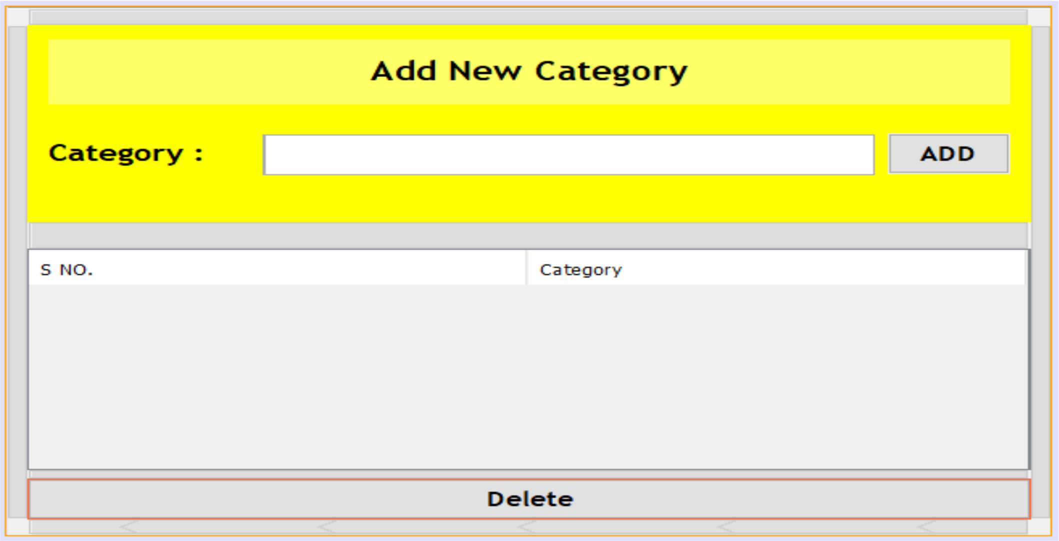


```
 public SpendingTracker() {
 initComponents();
displayCategory();
d.setSelectableDateRange(null, new java.util.Date());
```

```
  getEntries();

  this.setExtendedState(JFrame.MAXIMIZED_BOTH);

  d.setDate(new java.util.Date());

}
        private void displayCategory(){
            try{
              category.removeAllItems();
              ResultSet rs=db.DbConnect.st.executeQuery("select * from category_info");
              while(rs.next()){
                category.addItem(rs.getString("category"));
              }
            }catch(Exception ex){
              JOptionPane.showMessageDialog(null, ex);
            }
          }
          private void getEntries(){
            try{
              javax.swing.table.DefaultTableModel dtm=(javax.swing.table.DefaultTableModel)table.getModel();
              int rc=dtm.getRowCount();
              while(rc--!=0){
                dtm.removeRow(0);
              }
              java.time.LocalDate cd=java.time.LocalDate.now();
              java.time.LocalDate bd=cd.minusDays(20);
              ResultSet rs=db.DbConnect.st.executeQuery("select * from spendings where sdate<='"+cd+"'and sdate>='"+bd+"'");
              int total=0;
              while(rs.next()){
                total+=rs.getInt("amount");
                Object o[]={rs.getInt("sid"),rs.getDate("sdate"),rs.getString("category"),rs.getInt("amount")};
                dtm.addRow(o);
              }
              totalAmount.setText(total+"");
            }
            catch(Exception ex){
              JOptionPane.showMessageDialog(null, ex);
            }
          }
        private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
            try{
              java.util.Date dt=d.getDate();
              String s1=a.getText();
              String c=(String)category.getSelectedItem();
              if(dt!=null && !s1.equals("") && !c.equals("")){
                int amount=Integer.parseInt(s1);
                java.sql.Date date=new java.sql.Date(dt.getTime());
                db.DbConnect.st.executeUpdate("insert into spendings (category,sdate,amount)
        values('"+c+"','"+date+"',"+amount+")");
                JOptionPane.showMessageDialog(null, "Expense Added Successfully");
                getEntries();
              }
              else{
                JOptionPane.showMessageDialog(null, "Plz fill all details");
              }
            }
            catch(Exception ex){
              JOptionPane.showMessageDialog(null, ex);
            }
          }
         private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
```

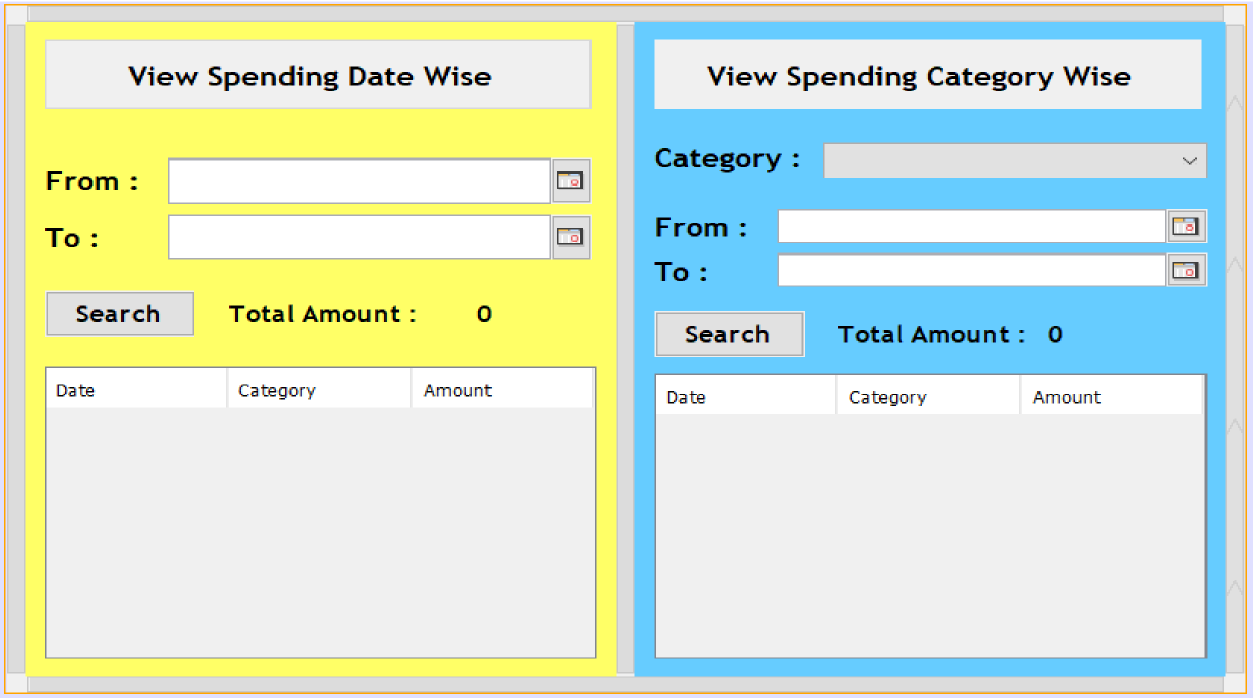
```
        int ri=table.getSelectedRow();
        if(ri!=-1){
            int r=JOptionPane.showConfirmDialog(null,"Do you really want to delete it","Deletion
Conformation",JOptionPane.YES_NO_OPTION);
              if(r==JOptionPane.YES_OPTION){
            int id=(int)table.getValueAt(ri, 0);
        try{
            db.DbConnect.st.executeUpdate("delete from spendings where sid="+id);
            JOptionPane.showMessageDialog(null, "Successfully Deleted");
            getEntries();
        }
        catch(Exception ex){
           JOptionPane.showMessageDialog(null, ex);
       }
      }
        }
    }
```

Category.java



```
package gui;
import java.sql.*;
import javax.swing.JOptionPane;
public class Category extends javax.swing.JFrame {

    public Category() {
        initComponents();
        getEntries();
    }
    private void getEntries(){
        try{
            javax.swing.table.DefaultTableModel dtm=(javax.swing.table.DefaultTableModel)table.getModel();
            int ro=dtm.getRowCount();
            while(ro--!=0){

                dtm.removeRow(ro);
            }
        ResultSet rs=db.DbConnect.st.executeQuery("select * from category_info");
        int sno=0;
        while(rs.next()){
        String category=rs.getString("category");
        //Object o[]={++sno,category};
        //dtm.addRow(o);
        java.util.Vector row=new java.util.Vector();
        row.add(++sno);
        row.add(category);
        dtm.addRow(row);
        }
        }
        catch(Exception ex){
            JOptionPane.showMessageDialog(null, ex);

        }
    }
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jPanel2 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        t = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        table = new javax.swing.JTable();
        jButton2 = new javax.swing.JButton();
        jMenuBar1 = new javax.swing.JMenuBar();

        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("Category");

        jPanel1.setBackground(new java.awt.Color(255, 255, 0));

        jPanel2.setBackground(new java.awt.Color(255, 255, 102));

        jLabel1.setFont(new java.awt.Font("Trebuchet MS", 1, 18)); // NOI18N
        jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel1.setText("Add New Category");

        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()
                .addComponent(jLabel1, 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()
                .addComponent(jLabel1)
                .addContainerGap(12, Short.MAX_VALUE))
        );

        jLabel2.setFont(new java.awt.Font("Trebuchet MS", 1, 16)); // NOI18N
        jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
        jLabel2.setText("Category :");
        t.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                tActionPerformed(evt);
            }
        });

        jButton1.setFont(new java.awt.Font("Trebuchet MS", 1, 14)); // NOI18N
        jButton1.setText("ADD");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(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)
                    .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 103,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(t, javax.swing.GroupLayout.PREFERRED_SIZE, 304,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jButton1)))
                .addContainerGap())
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel2)
                    .addComponent(t, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton1))
                .addGap(31, 31, 31))
        );
```

ViewSpending.java



```
package gui;
import java.sql.*;
import javax.swing.JOptionPane;
public class ViewSpending extends javax.swing.JFrame {
    public ViewSpending() {
        initComponents();
 d1.setSelectableDateRange(null, new java.util.Date());
        d1.setDate(new java.util.Date());
        d2.setDate(new java.util.Date());
        displayCategory();
        dd1.setSelectableDateRange(null, new java.util.Date());
        dd1.setDate(new java.util.Date());
        dd2.setDate(new java.util.Date());
    }
    private void displayCategory(){
            try{
            ResultSet rs=db.DbConnect.st.executeQuery("select * from category_info");
            while(rs.next()){
                category.addItem(rs.getString("category"));
            }
        }
    catch(Exception ex)
    {
    JOptionPane.showMessageDialog(null, ex);
}
    }
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jPanel2 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        d1 = new com.toedter.calendar.JDateChooser();
        d2 = new com.toedter.calendar.JDateChooser();
        jButton1 = new javax.swing.JButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        table1 = new javax.swing.JTable();
        jLabel4 = new javax.swing.JLabel();
        totalAmount1 = new javax.swing.JLabel();
        jPanel3 = new javax.swing.JPanel();
        jPanel4 = new javax.swing.JPanel();
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        dd1 = new com.toedter.calendar.JDateChooser();
        dd2 = new com.toedter.calendar.JDateChooser();
        jButton2 = new javax.swing.JButton();
        jScrollPane2 = new javax.swing.JScrollPane();
        table2 = new javax.swing.JTable();
        jLabel9 = new javax.swing.JLabel();
        totalAmount2 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        category = new javax.swing.JComboBox<>();
        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        setTitle("View Spending");

        jPanel1.setBackground(new java.awt.Color(255, 255, 102));
        jLabel2.setFont(new java.awt.Font("Trebuchet MS", 1, 16)); // NOI18N
        jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel2.setText("View Spending Date Wise");


        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()
            .addComponent(jLabel2, javax.swing.GroupLayout.DEFAULT_SIZE, 258,
    Short.MAX_VALUE)
            .addGap(20, 20, 20))
        );
        jPanel2Layout.setVerticalGroup(
          jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
    jPanel2Layout.createSequentialGroup()
            .addContainerGap(12, Short.MAX_VALUE)
            .addComponent(jLabel2)
            .addContainerGap())
        );

        jLabel1.setFont(new java.awt.Font("Trebuchet MS", 1, 14)); // NOI18N
        jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
        jLabel1.setText("Total Amount :");

        jLabel3.setFont(new java.awt.Font("Trebuchet MS", 1, 16)); // NOI18N
```

```
jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
jLabel3.setText("To :");

d1.addPropertyChangeListener(new java.beans.PropertyChangeListener() {
    public void propertyChange(java.beans.PropertyChangeEvent evt) {
        d1PropertyChange(evt);
    }
});

jButton1.setFont(new java.awt.Font("Trebuchet MS", 1, 14)); // NOI18N
jButton1.setText("Search");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

table1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {


    },
    new String [] {
        "Date", "Category", "Amount"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false
    };

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

```
      }
    });
    table1.getTableHeader().setReorderingAllowed(false);
    jScrollPane1.setViewportView(table1);

    jLabel4.setFont(new java.awt.Font("Trebuchet MS", 1, 16)); // NOI18N
    jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
    jLabel4.setText("From :");

    totalAmount1.setFont(new java.awt.Font("Trebuchet MS", 1, 14)); // NOI18N
    totalAmount1.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
    totalAmount1.setText("0");

    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.L
EADING, false)
          .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jButton1)
            .addGap(18, 18, 18)
            .addComponent(jLabel1)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(totalAmount1, javax.swing.GroupLayout.PREFERRED_SIZE,
61, javax.swing.GroupLayout.PREFERRED_SIZE))
          .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
          .addGroup(jPanel1Layout.createSequentialGroup()
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
              .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
              .addComponent(d2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
              .addComponent(d1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
      .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 291,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(19, 19, 19))
    );
    jPanel1Layout.setVerticalGroup(
      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(29, 29, 29)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
          .addComponent(jLabel4)
          .addComponent(d1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
          .addComponent(jLabel3)
          .addComponent(d2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
          .addComponent(jButton1)
          .addComponent(jLabel1)
          .addComponent(totalAmount1))
        .addGap(18, 18, 18)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 172,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

    jPanel1Layout.linkSize(javax.swing.SwingConstants.VERTICAL, new
java.awt.Component[] {d1, d2, jButton1, jLabel1, jLabel3, jLabel4, totalAmount1});

    jPanel3.setBackground(new java.awt.Color(102, 204, 255));

    jLabel6.setFont(new java.awt.Font("Trebuchet MS", 1, 16)); // NOI18N
    jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
```

```
        jLabel6.setText("View Spending Category Wise");

        javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
        jPanel4.setLayout(jPanel4Layout);
        jPanel4Layout.setHorizontalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel4Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT_SIZE, 258,
Short.MAX_VALUE)
                .addGap(20, 20, 20))
        );
        jPanel4Layout.setVerticalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel4Layout.createSequentialGroup()
                .addContainerGap(12, Short.MAX_VALUE)
                .addComponent(jLabel6)
                .addContainerGap())
        );

        jLabel7.setFont(new java.awt.Font("Trebuchet MS", 1, 14)); // NOI18N
        jLabel7.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
        jLabel7.setText("Total Amount :");

        jLabel8.setFont(new java.awt.Font("Trebuchet MS", 1, 16)); // NOI18N
        jLabel8.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
        jLabel8.setText("To :");

        dd1.addPropertyChangeListener(new java.beans.PropertyChangeListener() {
            public void propertyChange(java.beans.PropertyChangeEvent evt) {
                dd1PropertyChange(evt);
            }
        });

        jButton2.setFont(new java.awt.Font("Trebuchet MS", 1, 14)); // NOI18N
        jButton2.setText("Search");
```

```
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

table2.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {

    },
    new String [] {
        "Date", "Category", "Amount"
    }
) {
    boolean[] canEdit = new boolean [] {
        false, false, false
    };

    public boolean isCellEditable(int rowIndex, int columnIndex) {
        return canEdit [columnIndex];
    }
});
table2.getTableHeader().setReorderingAllowed(false);
jScrollPane2.setViewportView(table2);

jLabel9.setFont(new java.awt.Font("Trebuchet MS", 1, 16)); // NOI18N
jLabel9.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
jLabel9.setText("From :");

totalAmount2.setFont(new java.awt.Font("Trebuchet MS", 1, 14)); // NOI18N
totalAmount2.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
totalAmount2.setText("0");

jLabel11.setFont(new java.awt.Font("Trebuchet MS", 1, 16)); // NOI18N
jLabel11.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
jLabel11.setText("Category :");

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(
      jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel3Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
          .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 0,
Short.MAX_VALUE)
          .addGroup(jPanel3Layout.createSequentialGroup()
            .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(dd1, javax.swing.GroupLayout.DEFAULT_SIZE, 226,
Short.MAX_VALUE))
          .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING, false)
            .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(jPanel3Layout.createSequentialGroup()
              .addComponent(jButton2)
              .addGap(18, 18, 18)
              .addComponent(jLabel7)
              .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATE
D)
              .addComponent(totalAmount2, javax.swing.GroupLayout.PREFERRED_SIZE,
75, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(jPanel3Layout.createSequentialGroup()
              .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 61,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
              .addComponent(dd2, javax.swing.GroupLayout.DEFAULT_SIZE, 226,
Short.MAX_VALUE))
            .addGroup(jPanel3Layout.createSequentialGroup()
              .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE, 85,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
              .addComponent(category, 0, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))))
        .addContainerGap())
```

```
    );
    jPanel3Layout.setVerticalGroup(
      jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel3Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.B
ASELINE)
          .addComponent(jLabel11)
          .addComponent(category, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
          .addComponent(dd1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jLabel9, javax.swing.GroupLayout.Alignment.TRAILING))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.T
RAILING)
          .addComponent(dd2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jLabel8))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.L
EADING)
          .addComponent(totalAmount2)
```

```
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                        .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 27,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(jLabel7)))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED_SIZE, 168,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );

        jPanel3Layout.linkSize(javax.swing.SwingConstants.VERTICAL, new
java.awt.Component[] {jButton2, jLabel7, totalAmount2});

        jPanel3Layout.linkSize(javax.swing.SwingConstants.VERTICAL, new
java.awt.Component[] {category, jLabel11});

        jPanel3Layout.linkSize(javax.swing.SwingConstants.VERTICAL, new
java.awt.Component[] {dd1, dd2});

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, 311,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
                    .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addContainerGap())
        );

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

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        try{
            javax.swing.table.DefaultTableModel
dtm=(javax.swing.table.DefaultTableModel)table1.getModel();
        int rc=dtm.getRowCount();
        while(rc--!=0){
            dtm.removeRow(0);
        }
        java.sql.Date dt1=new java.sql.Date(d1.getDate().getTime());
        java.sql.Date dt2=new java.sql.Date(d2.getDate().getTime());
        ResultSet rs=db.DbConnect.st.executeQuery("select * from spendings where
sdate>='"+dt1+"'and sdate<='"+dt2+"' order by sdate asc");
        int total=0;

            while(rs.next()){
                 int t=rs.getInt("amount");
                 total+=t;
                Object o[]={rs.getDate("sdate"),rs.getString("category"),t};
                dtm.addRow(o);
            }
            totalAmount1.setText(total+"");
        }
        catch(Exception ex){
            JOptionPane.showMessageDialog(null, ex);
```

Chapter-4

Result and Discussion

After making this application we assure that this application will help its users to manage the cost of their daily expenditure. It will guide them and aware them about their daily spendings. It will prove to be helpful for the people who are frustrated with their daily budget management, irritated because of amount of spendings and wishes to manage money and to preserve the record of their daily cost which may be useful to change their way of spending money. In short, this application will help its users to overcome the wastage of money.

Recommendation:

DET app is usable by anyone who are willing to manage their spendings and aiming to save for the future investments. This app has no range criteria or any kind of profession or gender are focused so it will used hugely by any other person.

Future Scope:

In further days, there will be mails and pay mode embedded with the app. Also, backup details will be recorded on clou

Scope and Limitations

Limitations

- User have to entry every record manually.
- The category divided may be blunder or messy.
- Person who is handling system must have some technical knowledge.

Future Scope

In further days, there will be mails and pay mode embedded with the app. Also, backup details will be recorded on cloud.

Chapter-5

Conclusion and Future Scope

After making this application we assure that this application will help its users to manage the cost of their daily expenditure. It will guide them and aware them about their daily spending. It will prove to be helpful for the people who are frustrated with their daily budget management, irritated because of amount of spending and wishes to manage money and to preserve the record of their daily cost which may be useful to change their way of spending money. In short, this application will help its users to overcome the wastage of money.

1.4.1 Scope

This application can take a good market as it is usable by anyone who are willing to manage their spendings and aiming to save for the future investments and many more. There is not any range criteria or any kind of profession or gender are focused, it will used hugely. 1.4.2

References

MySQL

[1] www.dev.mysql.com

[2] www.mysql.com

Java

[1] www.w3schools.com

[2] www.javatpoint.com