**Project Report**

# Object-Oriented Programming Using Java

# (ECSE102L)

# EXPENSE MANAGER


**Bennett University**

**School of Engineering & Applied Sciences**

**Department of Computer Sciences & Information Technology**

**Submitted by:**

**BLOCK OVERFLOW (EB09)**

PRANJAL BHARDWAJ (E19CSE432)

ATINDRA SHEKHAR (E19CSE187)

SADHIL CHHABRA (E19CSE299)

KASARU SIVA KUMAR (E19CSE027)

ADUPA SANJAY BHARGAV (E19CSE068)

**June 2020**

# ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and my university professors. I would like to extend my sincere thanks to all of them.

I am highly indebted to Pranjal Bhardwaj for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.

I would like to express my gratitude towards my parents and my team members Pranjal Bhardwaj, Sadhil Chhabra, Kasaru Siva Kumar and Adupa Sanjay Barghav for their kind cooperation and encouragement which help me in completion of this project.

I would like to express my special gratitude for giving me such attention and time.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

A special gratitude I give to our professors namely Dr. Arpit Bhardwaj, Dr. Mayank Swarankar, Mr. Mohit Sajwaan, Mr. Deepak Singh, and Mr. Vijaypal Singh Rathor whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project especially in writing this report.

**No codes have been taken from GitHub or any other internet sources. All of the codes have been self-created by putting all of our minds together into this project.**

# ABSTRACT

In this project, we develop an Android application and a JAVA Application that keeps track of user personal expenses, his/her personal contribution towards group expenses, and maintain monthly incomes.

Mobile applications stood top among usability and user convenience. Many applications are available in the market to manage personal and group expenses. EXPENSE MANAGER is an android application. This monitors your earnings and your expenditures. This application helps you to monitor your everyday earnings and the details about where the money was spent. Every information is registered to the database along with the automatic Time Stamp. The database can be recovered by oneself and other trusted clients as well.

PERSONAL EXPENSE MANAGER is a JAVA Application that can run on any Operating Systems whether it's Windows, macOS or even Linux. This application is basically a JAR File and a compilation of all the modules created in JAVA. This application allows the user to Add Category and presents the categories added by the user. This application also allows expense entry in the selected categories by the user. After completing all the formalities the user can now view the expenses category wise, monthly and even yearly by automatically sorting in ascending order of the months and year. All of this data is stored by Java Virtual Machine in a file which is serializable. The files are stored namely in the form of 'category.ser' and 'expense.ser'. These files are stored onto the hard drive of the computer. The user can access these data in the files by just running the jar file from the command line of the computer.

# Table of Contents

# LIST OF FIGURES

# 1 Introduction

We are making an expense manager application which is specifically designed for the university expenses and makes it easier for a student to manage finances. It is designed to be a simple, **intuitive**, **stable** and **feature-rich application**. This application helps enable users to keep track of their expenses.

## 1.1 Problem Statement

If you don't check your spending and create a budget, you will have no control whatsoever on your money. Instead, **money will control you**, and you will either have perpetual lack of funds or you will end up steeped in debt. If you are clueless about how much is your **inflow** and how much you are **spending**, you will not know at the end of the month what happened to your money. If you don't have great financial management skills, you will not know how to categorize your expenses. In this day and age, when expenses are going through the roof, we need an **Expense Manager App.**

## 1.2 Objectives

Expense Manager helps in improving finance management control. It generates expenditure by different categories. It helps in reducing paper-work. The main objective of this system is to improve the database management performance and provide easy and fast results in maintaining the employee details. To provide high level security and to reduce the time in generating the report.

## 1.3 Importance and Need of your Project:

a) Prioritize Your Spending
b) Becoming Aware of Poor Spending Habits
c) Identify Fraud
d) Take Control of Your Finances
e) Savings and Investments

# 2 Proposed Solution/Approach/Technique

So, we have made an **Expense Manager** Application. This Expense Manager application will help one manage and report one's expenses. It has been designed to be a **great budget manager** tool for anyone who is concerned about their spending. It's designed to be a feature rich application with numerous tools such as **adding categories**, **expense entry**, **expense listing**, **monthly expense listing** and **yearly expense listing**. One can also add **remarks** beside their spent category to know specifically for what purpose the money was spent.

## List of the java files created and their explanations:

### i) StartApp.java

This file in Java is the startup of the application. This file creates the object and then the 'PEMService' constructor is called. The object will call the 'showMenu' method and ask the user for a choice.

### ii) PEMService.java

This file in Java contains all the functionalities interlinked between various other files written in Java. This file contains the '**showMenu**' method which is written using the switch - case.

This file also contains the '**onAddCategory**' which has functionalities to add categories by the user by creating an object of Category. The file '**Category.java**' stores the categories with unique **categoryID** which is automatically generated by the JVM in terms of milliseconds. The categoryID is generated in order to recognise every category differently, else the compiler will confuse and the Java application will crash.

The file also contains '**onCategoryList**' which has the functionalities to show the categories listed by the user. It will list all the categories by extracting the data from the '**clist**' object. The file will list all the categories in the form of a famous data structure in

JAVA known as '**Lists**'. Basically, the idea is that the categories are stored in the 'Category.java' file in the form of **'ArrayList**'.

In addition to all these facilities, the file contains '**onExpenseEntry**'. It has functionalities which allows the user to add expenses according to a user selected **specific category**. The user will be able to **add expenses** in both **floating points** as well as **integer**. The user will be able to add a **remark** for which the expense was made. The user has to add the **date** in the format of **DD/MM/YYYY**, this gives the user to add the expenses before the current day as well, in case he forgets to add on that day.

The file contains '**onExpenseList**' which displays the expense entry along with category ID, amount spent, remark and the date on which the expense was made.

To display the expenses monthly wise, yearly wise, and category wise this file also contains '**onMonthlyExpenseList**', '**onYearlyExpenseList**' and '**onCategorisedExpenseList**'.

To store the data to the main partition of the computer where the Operating System is managed, the java object should be given a special permission by the JVM to write it onto the hard disk of the computer. So in order to issue special permits to these objects '**repo.expList**' and '**repo.catList**' we have to make them serializable. Here we apply the concept of Serialization (Interfaces).
We made a reusable method '**serialize**' which would write and read the object.

**Note:**

*For a better understanding of all of these things we suggest you **visit our Java code**. For better viewing, we recommend you to use **IntelliJ IDE by JetStorm**. We have also done the **source code documentation** and added alot of **comments** beside the codes. You can also refer to Proposed Methodology in this document.*

## 2.1 Proposed Methodology

### (I) JAVA Application (Computer)

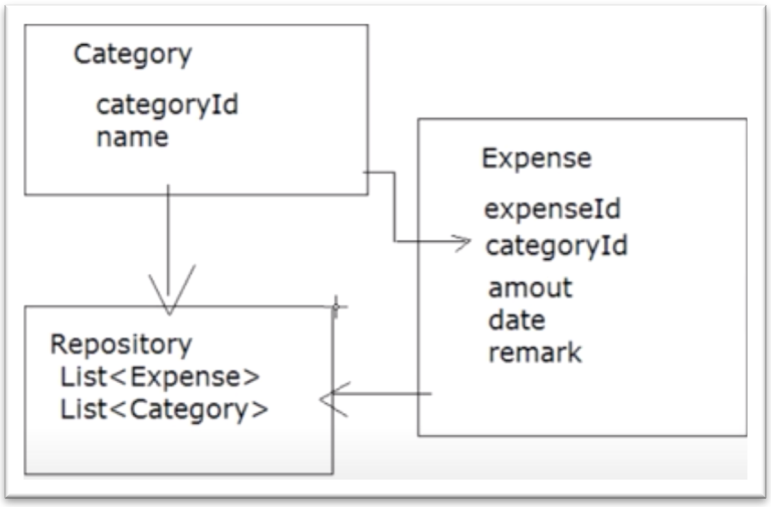


Fig 2.1 (a): Creating classes: Category, Expenses, Repository, PEMService and Main Class.

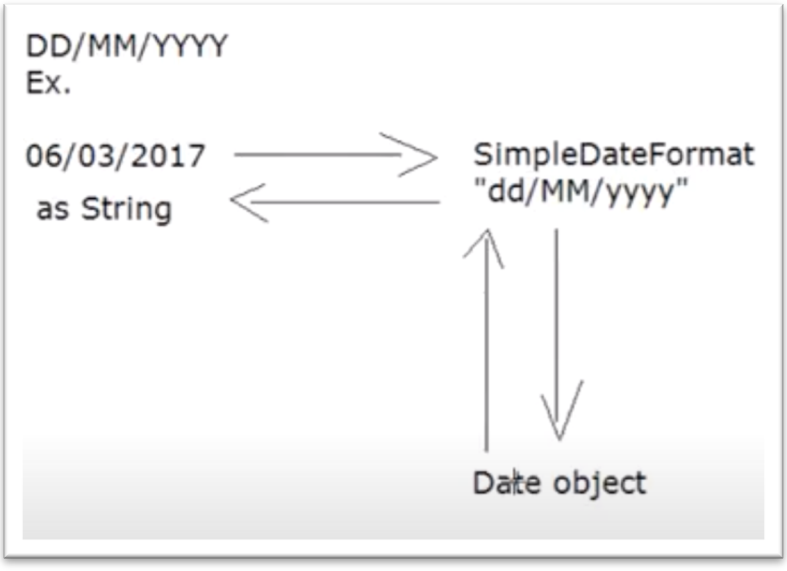


Fig 2.1 (b): Date Formatting using SimpleDateFormat object

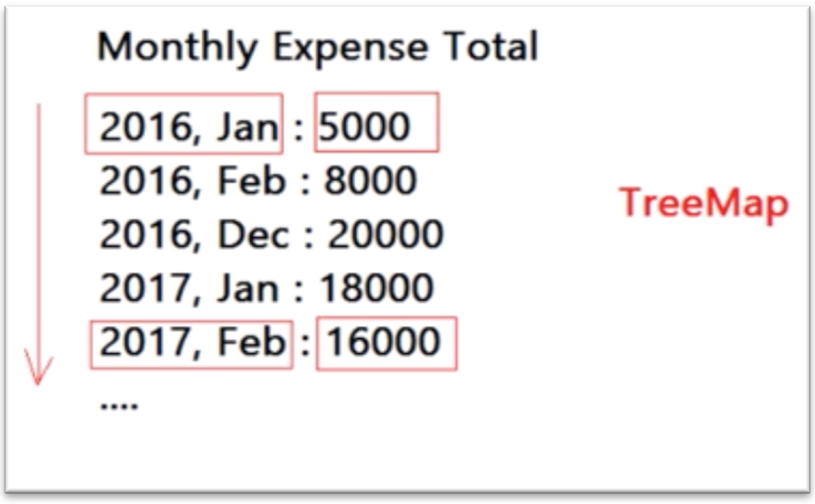


Fig 2.1 (c): Month Wise Expense Total Report

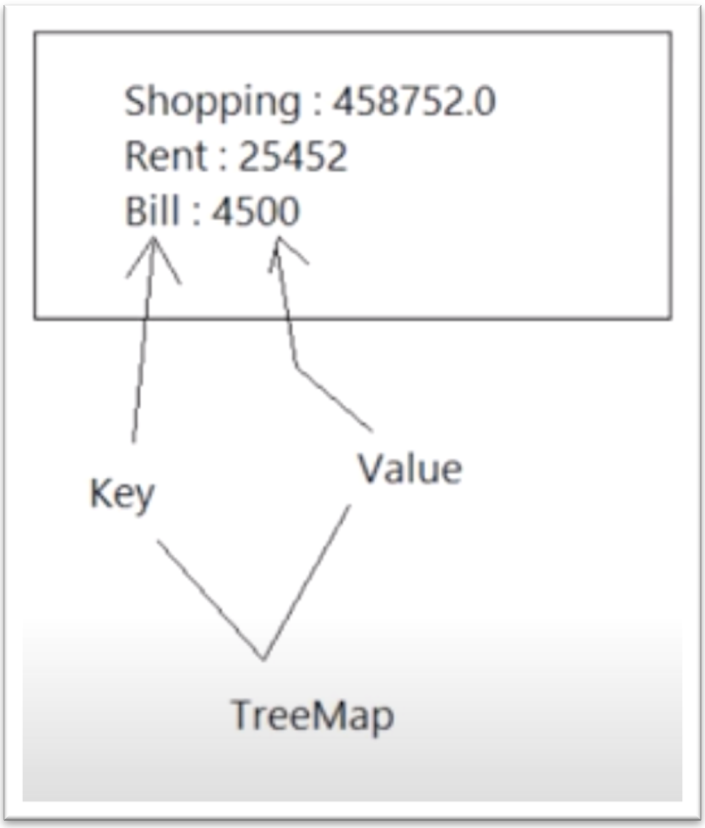


Fig 2.1 (d): Category Wise Expense Total Report

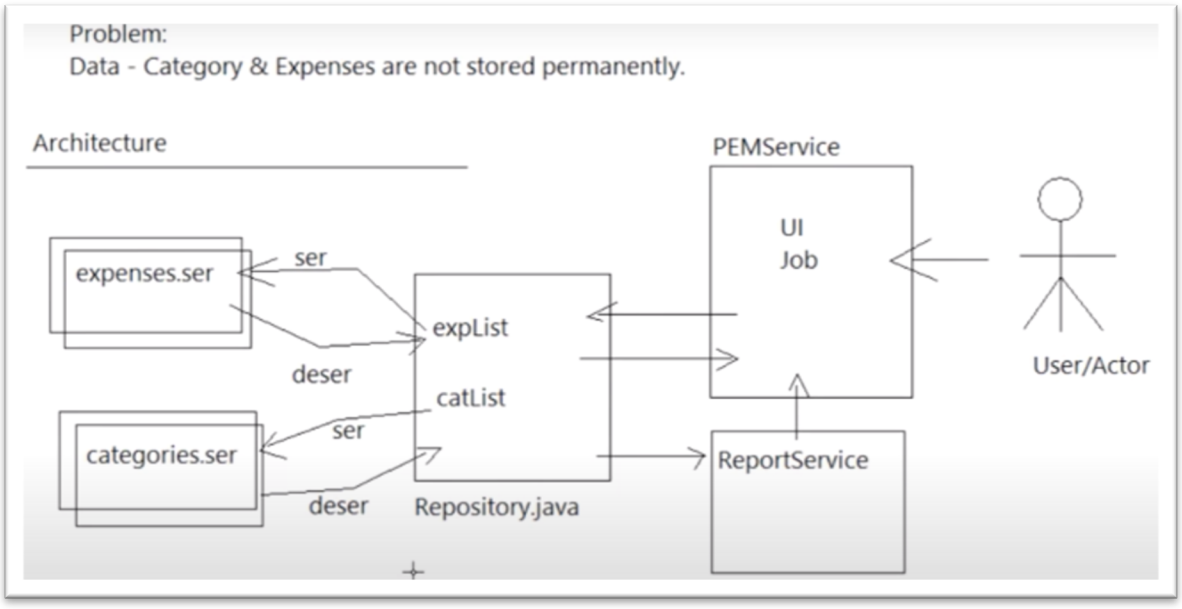


Fig 2.1 (e): Storing data permanently in file using Java Interface

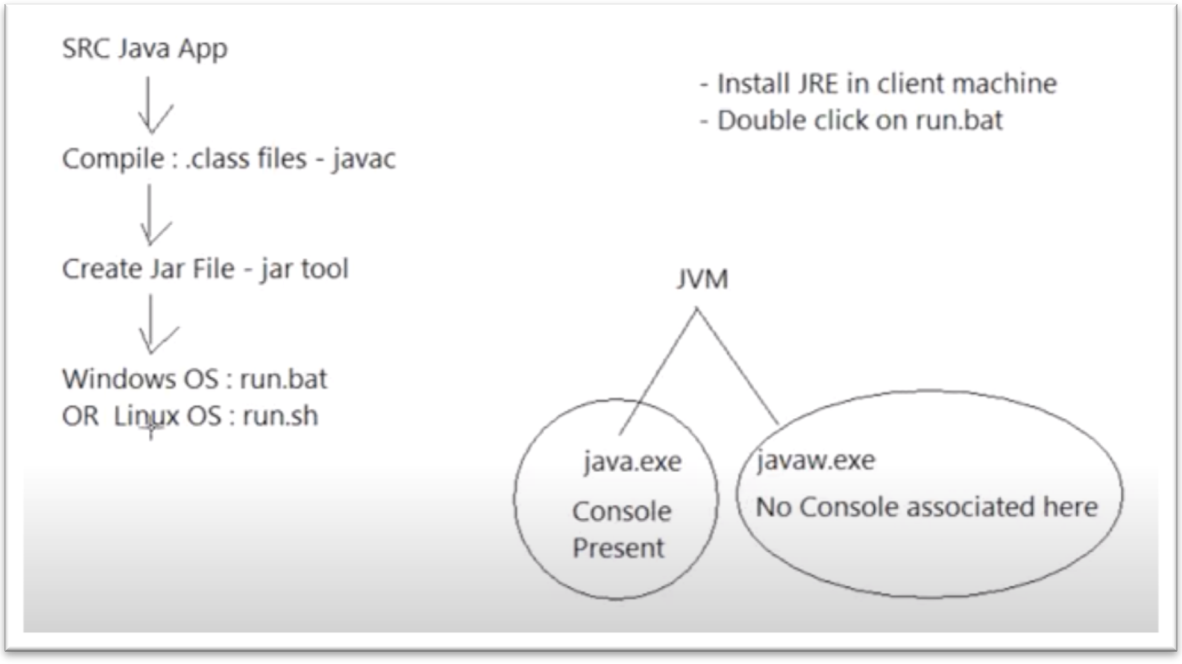


Fig 2.1 (f): Extensions for different Operating System

# 3 Project Execution

## 3.1 Project Setup

a) **Hardware Requirements**:

We have designed this project so that it can work without the need of any specific hardwares. But our project needs to be implemented on a computer irrespective of its specific system requirements and a smartphone.

**b) Software Requirements:**

We have designed our application to work on both computers and smartphones.

i) **Computer:**

For running the application on a computer, Oracle JDK 13 should be installed along with Oracle JRE 1.8.0. We have designed our application in such a way that it can work on any operating system which supports JAVA (Example: Microsoft Windows (Windows 7 (Version SP1) above), macOSX (10.13 and above), Linux (Red Hat Enterprise (Version 7.x) above).

ii) **Smartphone:**

For running the application on a smartphone, the smartphone should be running with Android OS (Lollipop and above).

## 3.2 Results and discussion

### a) Expense Manager Application (Java)

We have developed a desktop application using JAVA so that any individual can track his/ her expenses. We have developed this application using the Character User Interface without using any graphical tools.

So in order to get started with the designing of our big project, we had to plan out certain important features which our application will perform.

**A) Features**:

i) **Category Master**: User can add / manage categories.

For example: (Students) can have expenses for various categories such as rent, fees, stationary, lunch, dinner, party, etc.

ii) **Expense Entry**: Users can add expenses by selecting specific categories.

iii) **Expense List:** Raw listing of the expenses added by the user.

iv) **Reports:**

a) **Monthly Expense Report**
b) **Yearly Expense Report**
c) **Category wise Reports**

**B) User Interface:**

**Character User Interface:**

We have designed the user interface using the Object Oriented Programming Concepts.

The Character User Interface has been made using the basic switch case functions.

**C) Database:**

**File Based Repository:**

To store the expenses and the categories we have used the Interface Concepts of JAVA commonly known as Serialization (read and write Objects).

A file will be created in the main partition of the computer where the Operating System is managed by JVM ‘expenses.ser’ which will store all the expenses. It will write the object (serialise) from ‘expList’ in Repository.java file. To access the expenses, JVM will read the object from ‘expenses.ser’ (desterilize).

A file will be created in the main partition of the computer where the Operating System is managed by JVM ‘categories.ser’ which will store all the categories. It will write the object (serialise) from ‘catList’ in Repository.java file. To access the expenses, JVM will read the object from ‘categories.ser’ (desterilize).

# b) Expense Manager Application (Android)

An android application has also been developed so that the user can have an amazing UI experience. Great GUI components of the IDE Android Studio have been used in order to get the best UI possible. The database that has been used in the application is the Firebase from Google which is a cloud database storage service with a realtime database and lets the application know if there are any changes in the realtime.

**A) Features**:

i) **Registration** Login **Page**: The user can register and login through the login page. The service has been provided by the Google FireAuth service. This feature provides separate data log and entry for separate users providing a "rich" experience.

ii) **Detailed Expense List**: The application has a page that shows each and every payment made in the past. The application uses List view and using ArrayList adds the retrieved payments and incomes to the list view.

iii) **Expense Adder Page:** This page allows the user to add new payments/expenses to the list with automatic date and description so that the user can completely track the payments or expenses.

iv) **Income Adder Page:** This page allows the user to add new income to a list with automatic date and description so that the user can track all of the incoming money.

**Database -**

The Database that has been used in the android application is the google Firebase Realtime Database. This allows users to store data in the cloud at no expense. The database allows users to add income and payments with description and date with the amount in double which can be as accurate as the user wants.

# 4 Conclusion and Future Work

In this project, we develop a mobile application that keeps track of user personal expenses, his/her personal contribution towards group expenses, and maintain monthly incomes.

**Using our application, users can manage their expenses more effectively.**

As part of the project, we considered adding certain features to the application to make it more useful to the user.

Some of the extra features are like enabling users to register to the application using an existing email or social network account, it will synchronize the users profile information to the application.

**Thus the JAVA Application and the Android App will act as a pocket diary. It will store all the expenses in the form of a Database and can be edited independent of the location and the time.**

Later on, we have planned to deploy this application on a Cloud platform and a Chabot. In addition to this, making the application available to iOS Devices. There are also plans of working with Artificial Intelligence which will analyse and give graphical results of incomes and expenses.

# 5 Major Contributions

| Members Name | Major Contributions |
|---|---|
| Pranjal Bhardwaj | Android App Development<br>Project Summary<br>Project Report Documentation<br>Project PowerPoint Presentation<br>Poster Work<br>Video Editing |
| Atindra Shekhar<br><br>Sadhil Chhabra | Java Application Development<br>Project Summary<br>Project Report Documentation<br>Project PowerPoint Presentation<br>Poster Work<br>Video Editing |
| Kasaru Siva Kumar<br><br>Adupa Sanjay Bharghav | Poster Work<br>Project Summary |

# References

[1] Dr. Arpit Bhardwaj:

https://www.bennett.edu.in/faculties/dr-arpit-bhardwaj/

[2] Dr. Mayank Swarankar:

https://www.bennett.edu.in/faculties/mayank-swarnkar/

[3] Mr. Deepak Singh:

https://www.bennett.edu.in/faculties/deepak-singh/

[4] Mr. Mohit Sajwan

https://www.bennett.edu.in/faculties/mohit-sajwan/

[5] Mr. Vijaypal Singh Rathor

https://www.bennett.edu.in/faculties/vijaypal-singh-rathor/

[6] Developer.android.com

https://developer.android.com/

[7] Oracle JAVA:

https://www.oracle.com/java/technologies/

[8] IntelliJ Professional:

https://www.jetbrains.com/idea/