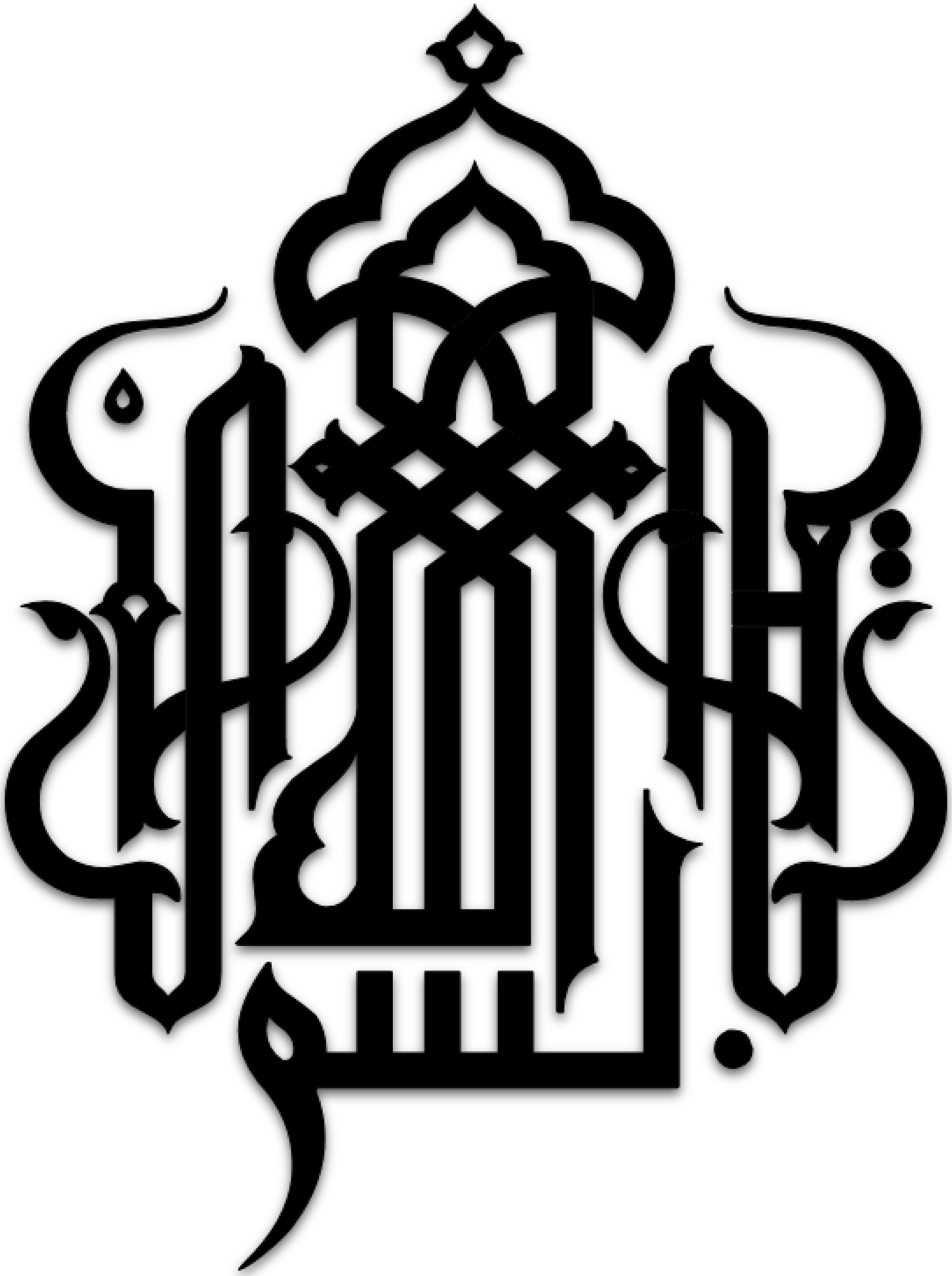


Personal Budget Management System

Session:
2016-2020

Submitted By:
Shehryar Kamran

Program:
BSCS

Registration No:
161-NUN-0086

Supervised By:

Syed Atif Ali Shah

Faculty of Engineering and Information Technology
NORTHERN UNIVERSITY NOWSHERA

CERTIFICATE OF APPROVAL

This is certified that we have studied this project report, Personal Budget Management System submitted by Mr. Shehryar Kamran, University Registration Number 161-NUN-0086. We conclude that this project report is of sufficient standard to warrant its acceptance by the Northern University, Nowshera for the award of the degree of BSCS.

Supervisor:

Syed Atif Ali Shah

Signature: _________________

HOD:
Muhammad Asif

Signature: _________________

DEDICATION

In the name of Allah, the Most Gracious and the Most Merciful Alhamdulillah, all praises to Allah for the strengths and His blessing in completing this thesis. I have to thank my parents for their love and support throughout my life. Thank you both for giving me strength to reach for the stars and chase my dreams. My brothers and sisters deserve my whole hearted thanks as well.

Special appreciation goes to my supervisor Syed Atif Ali Shah for his supervision and constant support. His invaluable help of constructive comments and suggestions throughout the experimental and thesis works have contributed to the success of this project. Without his support and hard work I was unable to complete this humble task.

To all my friends, thank you for your understanding and encouragement in my many, many moments of crisis. Your friendship makes my life a wonderful experience. I cannot list all the names here, but you are always on my mind.

Shehryar Kamran

ACKNOWLEDGMENT

First of all praise and thanks to ALLAH ALMIGHTY The most gracious and Merciful, real bless of the universe, whom I never heard ‘Nay, whenever I knocked at His door, who gave me the ability to contribute a drop of awareness and cognition from the existing ocean of knowledge and wisdom. I offer my countless salutation on the ‘HOLY PROPHET MUHAMMAD’ (PBUH) the entire source of guidance for humanity as a whole forever. I pay thanks to all our teachers and professors. I am extremely thankful to my supervisor Syed Atif Ali Shah who worked hard with me in each and every step of this project, guided me and encouraged me a lot to complete this project , in short I would not be able to complete this project without his kind support. It is a fact that I would not be successful in my aim without the sincere prayers, help and Encouragement of my Parents, Brother and sister.

THANKS

Shehryar Kamran

COPYRIGHTS NOTICE

Shehryar Kamran

BSCS

BRIEF CONTENTS

Chapter 1

Introduction

INTRODUCTION TO PERSONAL BUDGET MANAGEMENT SYSTEM

1.1 Introduction

A personal budget management is a finance plan that allocates future personal income towards expenses, savings and debt repayment. Past spending and personal debts are considered when creating a personal budget. Budgeting is the most basic and effective tool for managing our monthly income. But most people avoid doing it because they think keeping record of budget or expense is hectic task.

Recording monthly expenses on paper is very difficult as modification, removal and searching is not easy on paper. Similarly using any software which run on our computer like MS excel or any other tool is also clumsy. Because will have to note all expenses on paper then later record them all on tool installed on computer. Better solution use mobile application to keep record of every expense easily at any location.

In this project, your task is develop an Android based application which will be used to manage personal budget. Your application should fulfill following requirements.

1.2 Objectives

- ✓ Create personal budget.
- ✓ Record expense transactions.
- ✓ Record income transactions.
- ✓ Income can be updated or deleted in future.
- ✓ Remove any expense.
- ✓ Expense can be updated in future.
- ✓ Details of income can be seen.
- ✓ Expense which was created before with its information can be viewed later on.
- ✓ Filters for advanced transactions search.
- ✓ Data can be backup for future use.

1.3 Background

Managing expenses is important for our rapidly-growing daily life. Long gone are the days when you had to sit down with a notepad and a sky-high stack of bills and receipts to figure out your income, expenses, and overall budgetary plan. These days there are all kinds of spreadsheets and programs that are meant to make your financial planning a lot easier and more efficient. But short of carrying your laptop around with you or have to note all expenses on paper then later record them all on tool installed on computer, these comprehensive budgeting and tracking measures are largely confined to your home.

Managing personal expenses and incomes by an automated system really helps you stay disciplined to organize your finances, which is the first step of knowing your overall financial health. Maintaining a budget is like maintaining a bridge; Looking through your finances at regular intervals will allow to spot trends and see areas that can be improved way before it becomes a problem. Knowing your exact financial situation lifts a tremendous amount of stress off your shoulders because you don't have to worry about the unknown. But the Problem is how to maintain the budget efficiently and with ease and how to avoid overdraft fees, unexpected shortages, the minimum payment treadmill, and that sinking dread when you realize you actually couldn't afford to go out.

To solve this problem, we are developing this android application. This project reduces people time.

By using this application user can more efficiently plan their budget and lets you create customized categories in which it simply subtract from as you spend. Just add your expenses to the various categories and this app will tell you when you're running low on cash.

You can also break them down into several categories such as mortgage/rent, utilities, food, entertainment, and so on.

Chapter 2

Designing &

Implementation

DESIGNING & IMPLEMENTATION

System design is the phase where proposed system is actually designed keeping in the mind user requirements. It is necessary to consult the user of the system to be designed. The design phase that the user requirements can be worked out. The Design phase is best described by breaking it up into Logical Design and Physical Design sub phases. During the Logical Design phase, the system's analysts make use of the information collected in the Requirements phase to design the system independently of any hardware or software system. Once the higher-level Logical Design is complete, the systems analyst then begins transforming it into a Physical Design dependent on the specifications of specific hardware and software technologies. Creating and editing source files is the most important function that the IDE serves. After all, that is probably what you spend most of your day doing. The IDE provides a wide range of tools that can complement any developer's personal style, whether you prefer to code everything by hand or want the IDE to generate large chunks of code for you.

2.1 System Design

The System Design provided by the Java utility package is very powerful and perform a wide range of functions. In comparison, a System Design is a concrete implementation of the specification provided by an ADT. Some formal design methods and programming languages emphasize System Design, rather than algorithms, as the key organizing factor in software design. System Design is the first design activity, which results in less complex, modular and efficient program structure. The information domain model developed during analysis phase is transformed into System Design needed for implementing the software. The data objects, attributes, and relationships depicted in entity relationship diagrams and the information stored in data dictionary provide a base for System Design activity. During the System Design process, data types are specified along with the integrity rules required for the data.

2.2 Architectural Design

The most obvious representation: Java source code. However, writing source code before you fully understand an algorithm often leads to difficult-to-find bugs. One technique for overcoming those bugs involves flowcharts. A flowchart is a visual representation of an Architectural control design. Representation illustrates statements that need to execute, decisions that need to be made, logic flow (for iteration and other purposes), and terminals that indicate start and end points. To present that visual representation, a flowchart uses various symbols. Suppose you have a simple Architectural that initializes a counter to 0, reads characters until a new-line (\n) character is seen, increments the counter for each read digit character, and prints the counter's value after the new-line character has been read. An alternative to flowcharts is pseudo code*:* a textual representation of an algorithm that approximates the final source code. Pseudo code is useful for quickly writing down an algorithm's representation. Because syntax is not a concern, no rules define how to write pseudo code.

2.3 Sequence Model

The sequence structure is built into Java. Unless directed otherwise, the computer executes Java statements one after the other in the order in which they’re written—that is, in sequence. Java lets you have as many actions as you want in a sequence structure. Anywhere a single action may be placed, we may place several actions in sequence.

2.4 Data Flow Design

The Dataflow model provides a number of useful abstractions that insulate you from low-level details of distributed processing, such as coordinating individual workers, shading data sets, and other such tasks. These low-level details are fully managed for you by Cloud Data flow's runner services.

2.4.1 Level 0 DFD

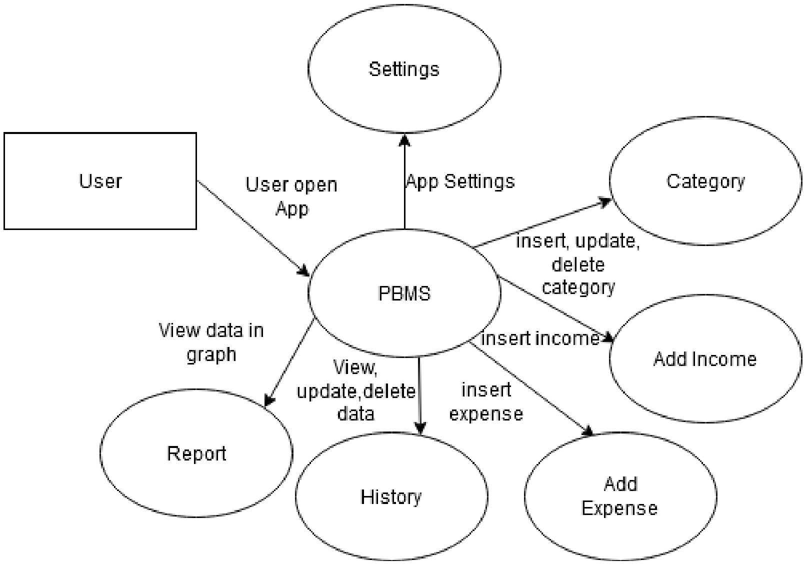


Figure 1: Level 0 Data flow Model

2.4.2 Level 1 DFD

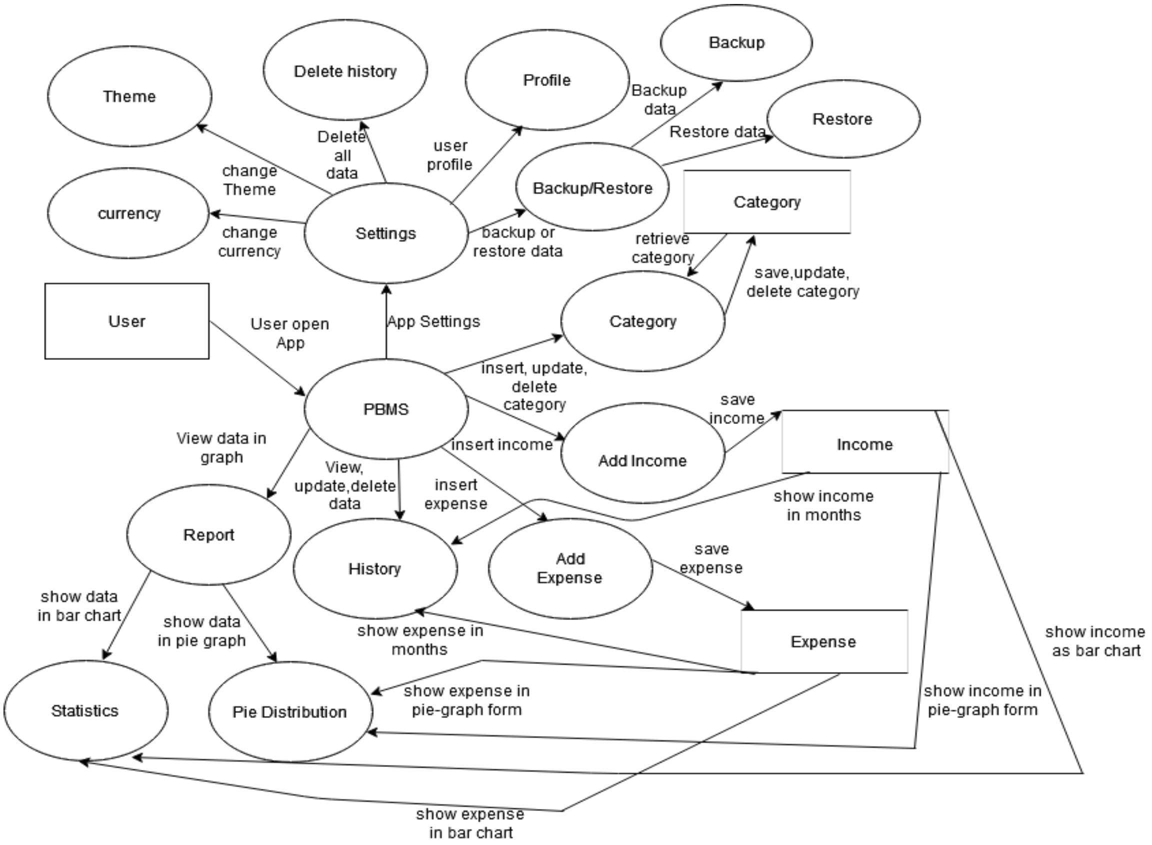


Figure 2: Level 1 Data flow Model

2.5 Entity Relationship Diagram

Entity relationship model is a logical representation of data in an organization. It views the entire system as a collection of entities related to one another. It is used to describe the elements of a system and their relationships. It was introduced by Peter Chen in 1976.

2.7.4 ERD

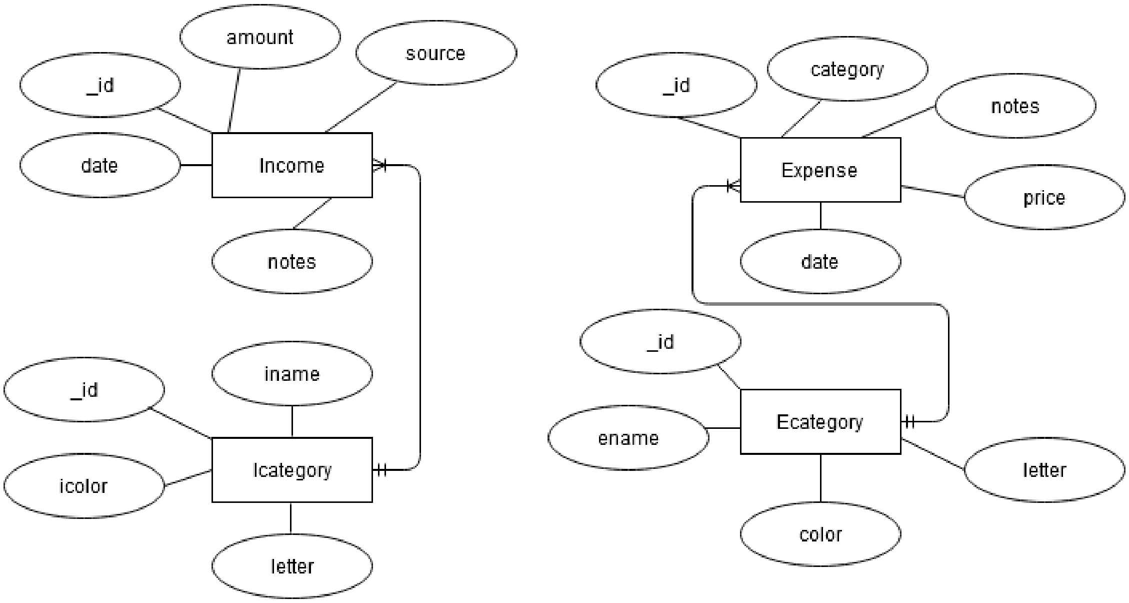


Figure 3: ER- Diagram

2.6 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

2.6.1 Use Case Diagram

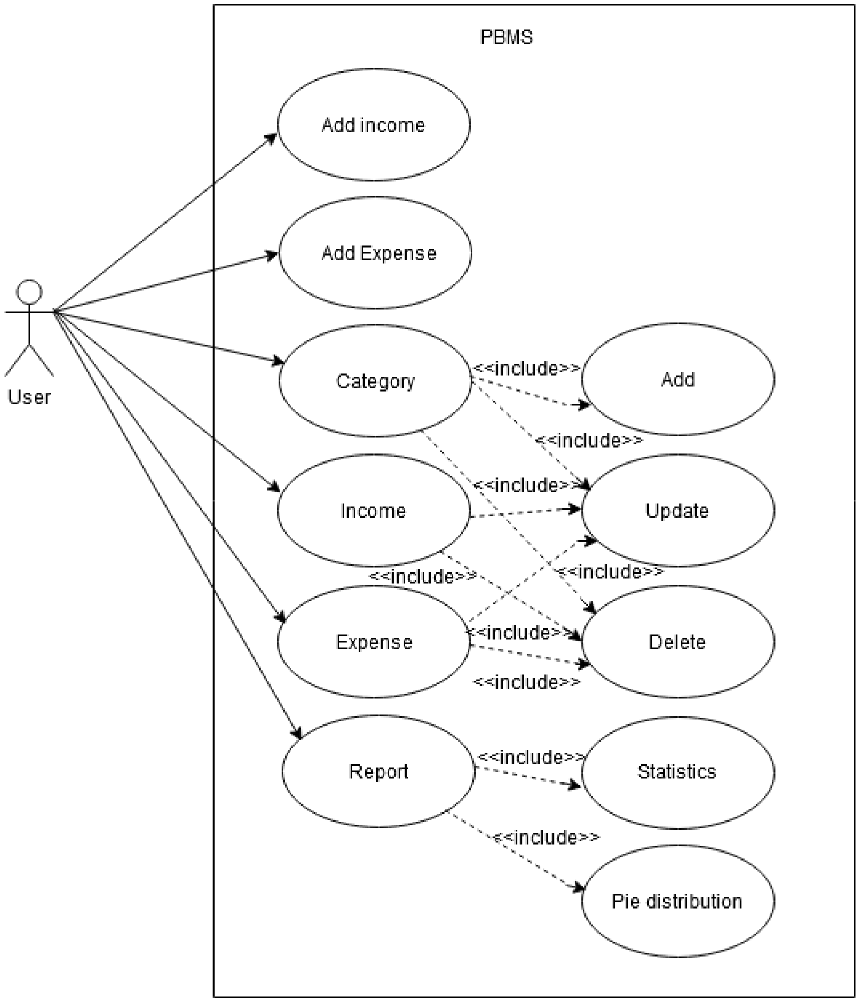


Figure 4: Use Case Diagram

2.7 Implementation

Software Implementation is the process of converting a new or a revised system design into an operational one. There are three types of implementation: Implementation of a computer system to replace a manual system, Implementation of a new computer system to replace an existing one, Implementation of a modified application to replace an existing one.

Since we are replacing a manual system with a new computerize system. Our implementation goes through the following phases.

- Data Conversion
- System Installation

2.7.1 System Installation

System installation is the process in which a new system is installed on computer. There are different techniques for the installation.

- Abrupt Cutover.
- Parallel Operation with a Single Cutover Point.

In our project we will use parallel approach where the new and the old system will be parallel operated. The new system is parallel operated for testing and comparison purpose.

2.7.2 Maintenance

Maintenance is the process of restoring something to its original positions. Sometime a program need changes, the change may be due to the additional needs of the user. Thus the system must be flexible enough to accept the changes. Maintenance covers a large range of activities including correction code, fixing designs errors, updating documentation, testing data and to improve performance.

Chapter 3

Tools &

Technologies

TOOLS & TECHNOLOGIES

3.1 Tools Used

- Android Studio (Platform)
- Language Used

3.1.1 Android Studio 3.5.3 (Android development tool)

Android Studio is the official integrated development environment (IDE) for Android platform development. It was announced on May 16, 2013 at the Google I/O conference. Android Studio is freely available under the Apache License 2.0.Android Studio was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.Based on Jet Brains' Interlay software, Android Studio is designed specifically for Android development. It is available for download on Windows, Mac OS X and Linux, and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development. New features are expected to be rolled out with each release of Android Studio. The following features are provided in the current stable version:

- Auto-recommend memory settings
- Build speed
- Apply changes
- Chrome OS support
- File access speed in disk I/O

3.1.2 Language use

I use this language for my project.

- JAVA
- XML

3.1.3 JAVA

Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere. Chances are you've downloaded a program that required the Java runtime, and so you probably have it installed it on your system. Java also has a web plug-in that allows you to run these apps in your browser. Java language has its own structure, syntax rules, and programming paradigm. The Java language's programming paradigm is based on the concept of OOP, which the language's features support. The Java language is a C-language derivative, so its syntax rules look much like C's. For example, code blocks are modularized into methods and delimited by braces ({ and }), and variables are declared before they are used. Structurally, the Java language starts with packages. A package is the Java language's namespace mechanism. Within packages are classes, and within classes are methods, variables, constants, and more.

3.1.4 XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The World Wide Web Consortium's XML 1.0 Specification of 1998 and several other related specifications all of them free open standards—define XML.

The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

Several schema systems exist to aid in the definition of XML-based languages, while programmers have developed many application programming interfaces (APIs) to aid the processing of XML data.

XML performs the same function in Android app development: describing data and introducing elements.

Specifically, XML sets the layout of things like buttons and images, and defines the font, color, and any text that shows by default. To make those buttons actually do anything, you will need to use a programming language like Java, though.

Chapter 4
Testing & Debugging

TESTING & DEBUGGING

Testing is the set of activities that are performed in order to uncover the errors in the software. Testing can be view as bastion from which quality can be assessed. There are different strategies for software testing. One extreme software test is conducted when software is fully constructed. At the other extreme the software is tested on daily basis. Both these approaches have their own advantages and disadvantages. Therefore I choose a software strategy that is falls between these two approaches. Software is tested incrementally, first testing the individual components and moving toward integration testing and then to the system testing.

4.1 Levels or strategies of Testing

I'm testing a project. My test strategy calls for implementing two entirely different methodologies for testing this project. The content-heavy site includes a fair amount of manual UI-based testing, ensuring the quality of the content as well as the flow and lay-out of content templates. It includes a base set of project tests, which checks for a series of positive and negative tests. Because it is similar to the search component on another site we own. Eventually each page will have a set of manual and automated test cases, developed in conjunction with the page development. There are four main stages of testing that need to be completed before a program can be cleared for use:

- Unit Testing.
- Integration Testing.
- System testing.
- Deployment testing.

4.1.1 Unit Testing

Unit testing is type of testing in which I focus on each individual component. In the event driven programming paradigm the subroutine that's execute for each event can be regarded as component. So I code each event individually and tested it before coding the next event. The local data structures, independent paths and error handling paths are tested for each component. For error handling I use the try catch method. In try block I write the code which is protected. If there any problem in try then in catch block I write the exception which show the output with dialoged box.

4.1.2 Integration Testing

Integration Testing are type of testing which focus on more than one component as they are integrated. After checking the individual component in the unit testing I integrate these components one by one. Every time a new component is added it is tested for its functionality. A test is needed to be design for checking all these sequences of events.

4.1.3 System Testing

To find such type of problem arises due to the integration of new component I need System testing. System testing is the re-execution of test that have already executed. Like I use the counter to increase or decrease the values. Sometimes I perform same task many time. To handling this I use System testing.

4.1.4 Deployment Testing

Deployment testing is type of testing that check whether the software work in different Operating environments or not. I have checked the software on different systems. The software runs and able to do all its function in these environments.

4.2 Debugging

Debugging is the process of removing errors from software. Debugging occurs as a consequence of successful testing, that is when a test case uncovers an error. Debugging is not testing but always occurs as a consequence of testing. The debugging process begins with executing of a test case. If the test case fails to provide the expected result then there are some errors present. The task of debugger is to locate these errors in code and remove it. As we use android studio (emulator) which provide and automatic debugging tool which locate most of the syntactic error present in the code. This made it easy to debug the code. But for logical error we still had to search the code to locate the error.

- Open the Options window.
- Click the Java Debugger tab in the Java category in the Options window. ...
- Select Visual Debugging in the list of categories and select Track locations of component hierarchy changes. ...
- Stop your debugging session (if one is running)

Chapter 5

Application Deployment & User Manual

APPLICATION DEPLOYMENT & USER MANUAL

As today's engineers try to meet industry challenges, such as the need to bring products to market faster and integrate new technologies into existing systems, a modular test software architecture has emerged as an ideal solution. The modularization of a test system extends to software by emphasizing the development of test system software using reusable and easy to maintain modules. One of the tasks a test system developer will need to accomplish when creating a modular test system is to deploy this test software to the production floor. Whether you choose to build an installer, use a network server to distribute your test software, or leverage a source code control package, you will want to use care to ensure the integrity of the test stations while at the same time working quickly to have the upgrade process be as seamless as possible. Therefore, an appropriate software deployment strategy will be needed. Software deployment is the process of managing and automating the packaging, testing, distribution and installation of software files and/or applications to test systems across an enterprise network or production floor. An effective software deployment strategy provides the confidence and reliability needed to ensure an effective software deployment. When facing software deployment, several challenges present themselves and need to be addressed such as:

- What method should I use to gather all my required files?
- How do I distribute those files to my test stations?
- How much control do I need or want over my distribution?
- How do I go about updating those test stations?
- Should I make this deployment process manual or automatic or even both?

To address these issues, several methods will be described below on how to effectively deploy your software. We will first discuss the methods of creating a deployment image. Once this deployment image has been created, we will cover the three methods for deploying this image to your test systems which are:

- File Copy Method
- Source Code Control Method
- Installer Method

5.1 Maintenance considerations

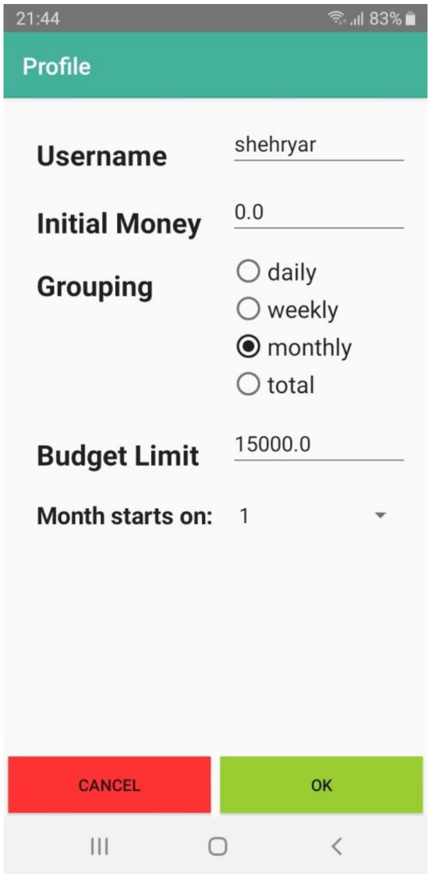
Factors are addressed that affect the total cost of developing, deploying and supporting software systems. The experiences are drawn from a large computerized facility incorporating over half a million lines of application code. The operational and support activities dominate the cost considerations over program life. Techniques are presented in this paper to ensure early design considerations of these operational and support activities and to reduce their cost. Thus, the focus is the life cycle cost factors of the system software.

5.2 User Guide

A user guide or user's guide, also commonly known as a manual, is a technical communication document intended to give assistance to people using a particular system. Most user guides contain both a written guide and the associated images. In the case of computer applications, it is usual to include screenshots of the human-machine interface and hardware manuals often include clear, simplified diagrams.

5.3 Profile

App run you will get Profile Screen first, for the first time only to setup your profile.
User will set his/her name, money that already it have(initial money), have budget record of which type(daily,weekly,monthly,total), budget limit and the date on which month start.



Username shehryar

Initial Money 0.0

Grouping
○ daily
○ weekly
● monthly
○ total

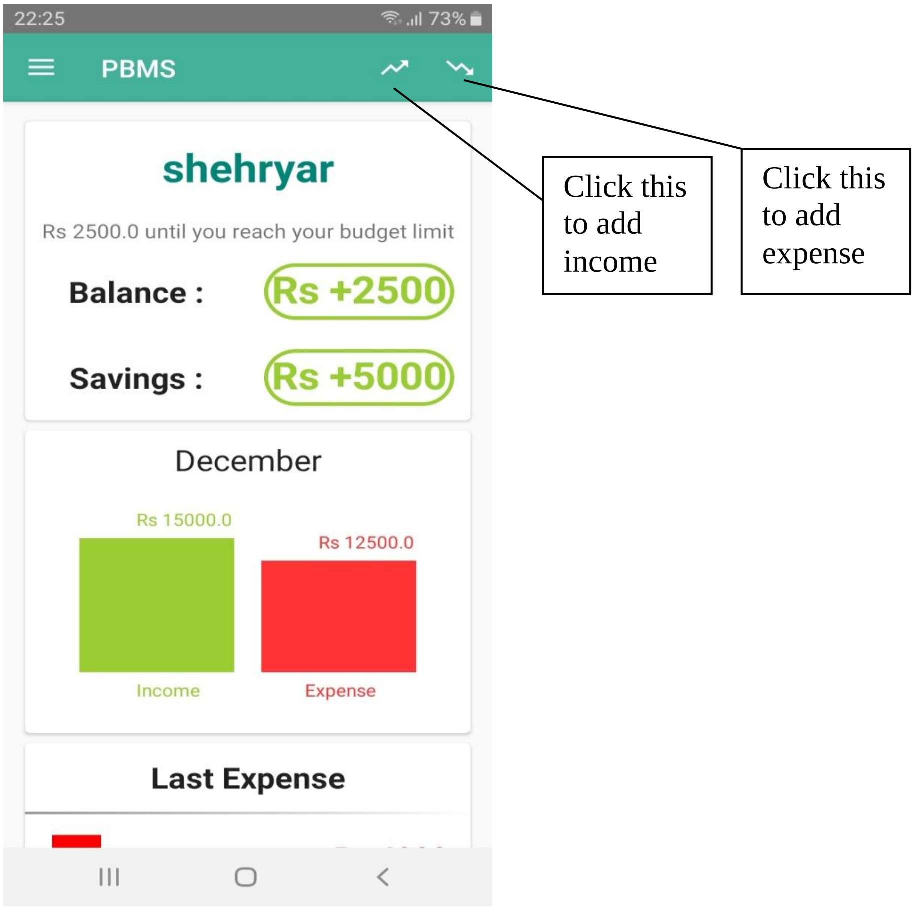
Budget Limit 15000.0

Month starts on: 1

Figure 5: Profile

5.4 Home

You will get the overview of your income, expense, saving and balance of current month. It also tells that how much money is left to reach your budget, if its over budget then warn the user, if budget reached then it also tell that also.



shehryar

Rs 2500.0 until you reach your budget limit

Balance : Rs +2500

Savings : Rs +5000

December

Last Expense

Click this to add income

Click this to add expense

Figure 6: Home

5.5 Navigation Bar

In Navigation Bar of App we have following items

- History
- Add Income
- Add Expense
- Pie Distribution
- Statistics
- Categories
- Settings
- Exit

By clicking one of them you will go to that screen in app.

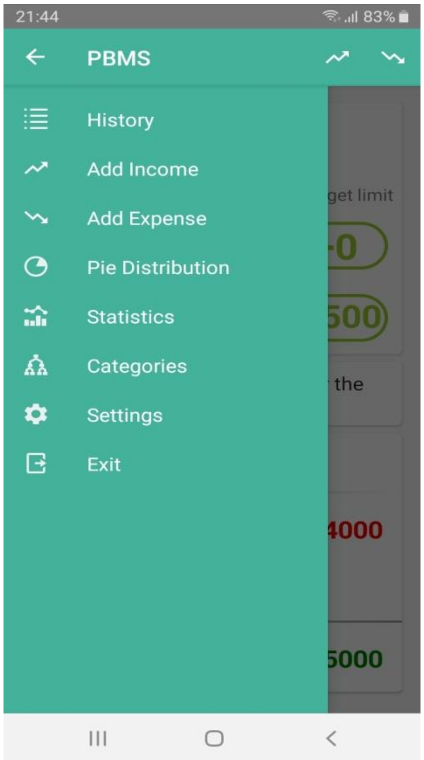
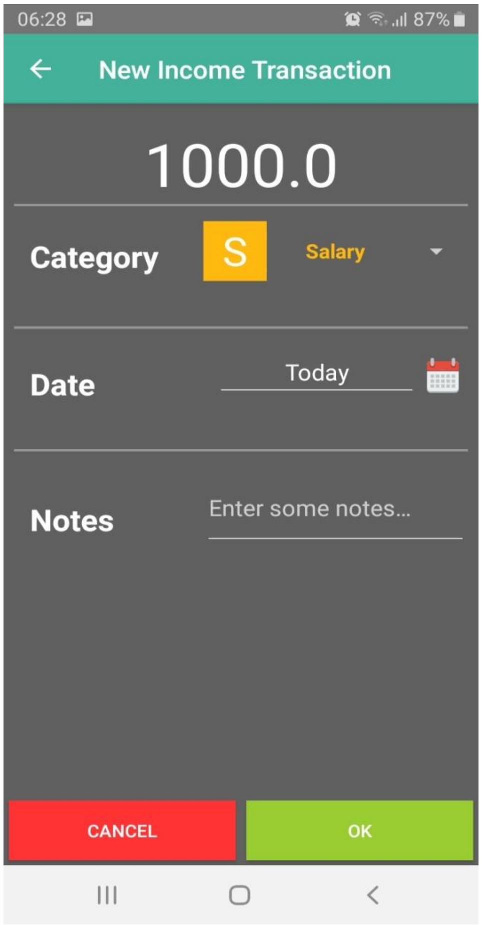


Figure 6: Navigation Bar

5.6 Add Income

User can add income by inserting amount, category, date, notes (optional).
Empty record cannot be inserted, Category can be changed by clicking on spinner and choosing from them.



New Income Transaction

1000.0

Category S Salary

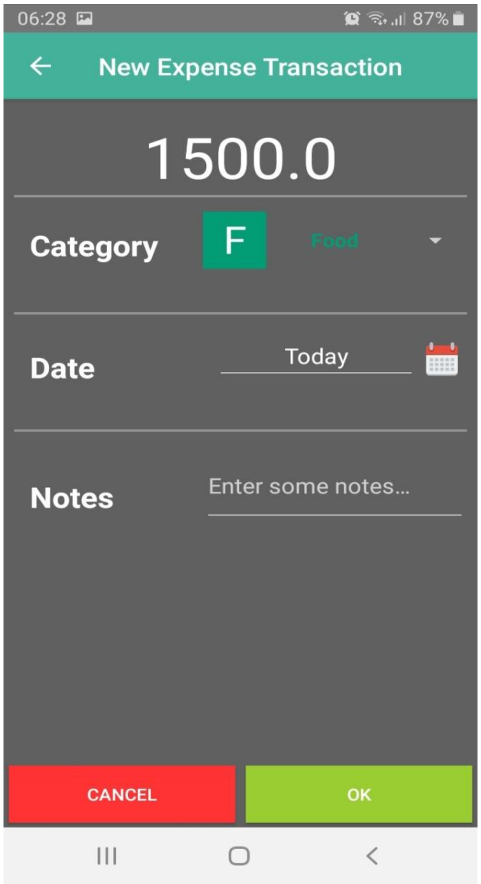
Date Today

Notes Enter some notes...

Figure 7: Add Income

5.7 Add Expense

User can add new expense by inserting amount, category, date, notes (optional).
Empty record cannot be inserted, Category can be changed by clicking on spinner and choosing from them.



New Expense Transaction

1500.0

Category F Food

Date Today

Notes Enter some notes...

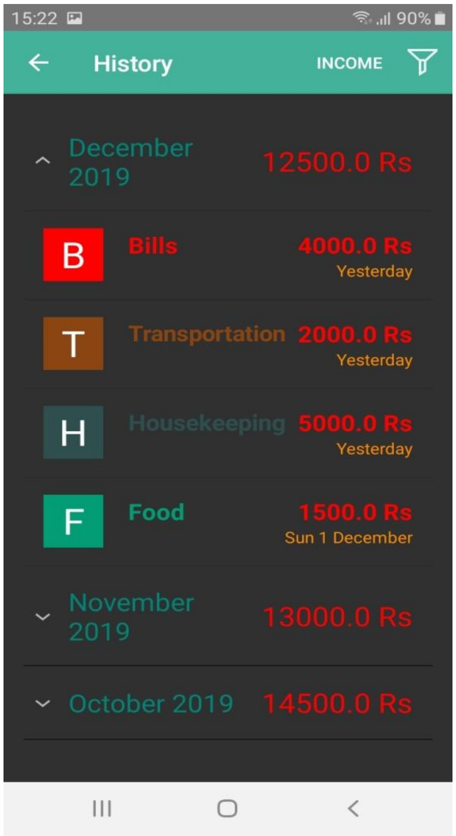
Figure 8: Add Expense

5.8 History

All Income/Expense history can be viewed from here as a whole or filtered.

By clicking single month user can view record according to that month,edit and delete a specific record can also be perform here.

Filteration can be done: By Most Recent, By Category, By Price, By Date, By Period(Custom Date).



15:22 90%

History INCOME

December 2019 12500.0 Rs

B Bills 4000.0 Rs Yesterday

T Transportation 2000.0 Rs Yesterday

H Housekeeping 5000.0 Rs Yesterday

F Food 1500.0 Rs Sun 1 December

November 2019 13000.0 Rs

October 2019 14500.0 Rs

Figure 9: History

5.9 Pie Distribution

Income/Expense can be viewed in pie graph and can be filtered.

Record tha shown can be of Monthly,weekly,Daily,Total,Custom Period(Custom Date).

Filteration can be according to categories.

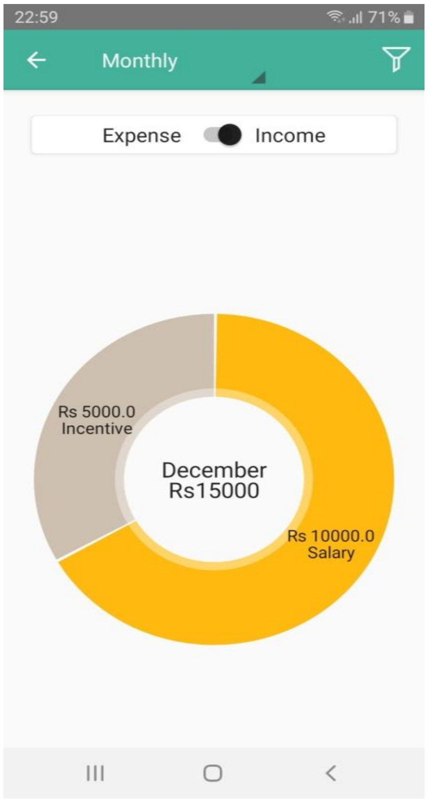


Figure 10: Pie Distribution

5.10 Statistics

Income and Expense in Bar chart form and can be filtered, red is for expense and green for income.

Show Record according to: Last 3 months, Last 6 weeks, Last 12 weeks, Total,Custom Period(Custom Date).

Filteration can be according to income or expense categories.

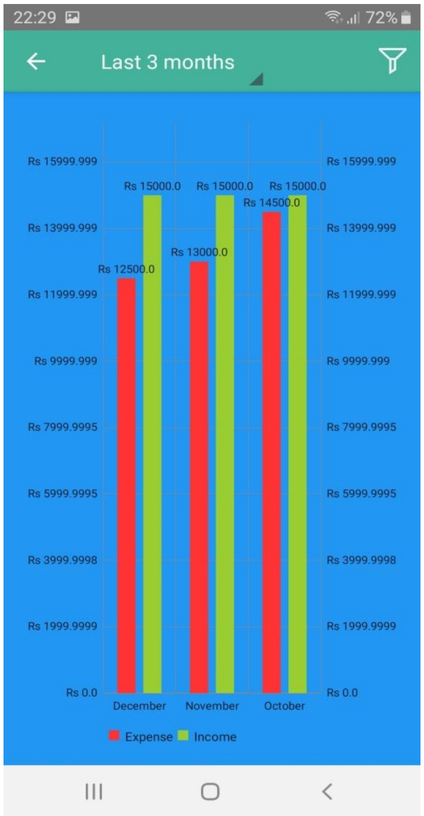


Figure 11: Statistics

5.11 Categories

Categories can be added, updated, deleted here for both income and expense.

Some predefine categories are already avalable here for user convience.

Category logo color can be what you want.

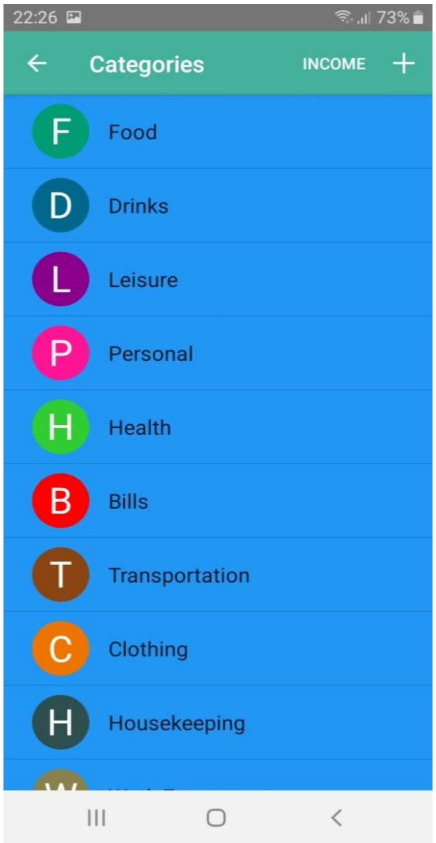


Figure 12: Categories

5.12 Settings

Here you can change currency, theme, profile. Delete all history at once or Backup/Restore data from your device.

In currency there are 7 currency options for different country people.

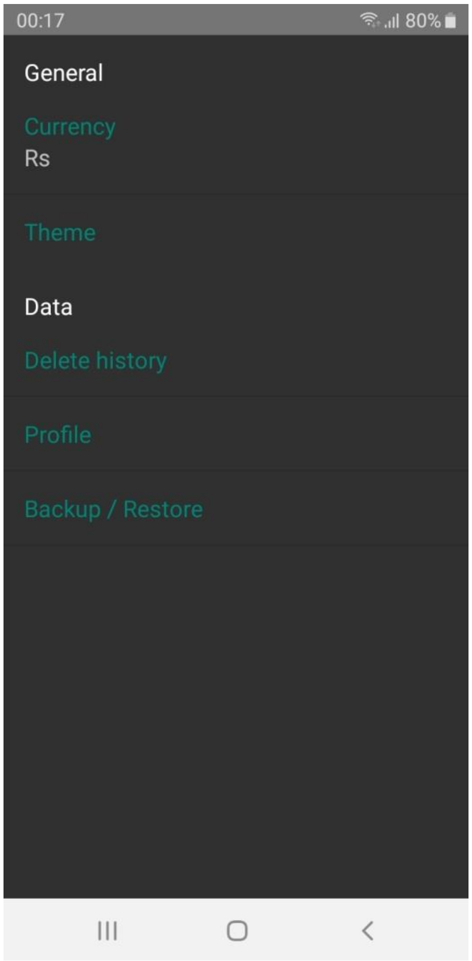
In theme there are four themes which are:white(default),black,grey,blue.

On delete history, it will ask you to confirm to delete all app data from database and on confirmation all record from database will be deleted.

Profile option will take you to profile screen.

In backup/restore you will be asked to backup or restore, if backup then all data is backup on device, restore is clicked then data that device contain will be imported in app.

If first time using backup/restore then permission is needed to access storage.



00:17 80%

General

Currency
Rs

Theme

Data

Delete history

Profile

Backup / Restore

Figure 13: Settings