

# Pdf Club

## REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT  
FOR

Six Month Industrial Training

at

**Auribises Technologies, Ludhiana**  
**(from June 5,2017 to December 5,2017)**

Submitted By

Navjot Singh (1411289)

Manpreet Singh (1411281)

Deepak Pruthi(1411248)



---

Information Technology Department  
**GURU NANAK DEV ENGINEERING COLLEGE**  
**LUDHIANA, INDIA**

---

## Acknowledgement

We are highly grateful to the Dr. M.S. Saini, Director, Guru Nanak Dev Engineering College (GNDEC), Ludhiana, for providing this opportunity to carry out the major project work at Guru Nanak Dev Engineering College.

The constant guidance and encouragement received from Dr. K.S. Mann H.O.D. IT Department, GNDEC Ludhiana has been of great help in carrying out the project work and is acknowledged with reverential thanks.

We would like to express a deep sense of gratitude and thanks profusely to Prof. Sandeep Kumar Singla, without his wise counsel and able guidance, it would have been impossible to complete the project in this manner.

We express gratitude to other faculty members of Information Technology department of GNDEC for their intellectual support throughout the course of this work.

Finally, we are indebted to all whosoever have contributed in this report work.

**Navjot Singh**  
**Manpreet Singh**  
**Deepak Pruthi**

---

## Abstract

Modern life offers a plethora of options of services and goods for consumers. As a result, peoples expenses have gone up dramatically, e.g., compared to a decade ago, and the cost of living has been increasing day by day. Thus it becomes essential to keep a check on expenses in order to live a good life with a proper budget set up.

The Android OS smartphones is one of the top-selling in the world ,it is apparent that people have been using smartphones as an organizational tool. .

Expense Tracker Mobile Application (Monitary) was developed for Android users to keep track of their expenses and determine whether they are spending as per their set budget. Potential users need to input the required data such as the expense amount, merchant, category, and date when the expense was made. Optional data such as sub-category and extra notes about the expense can be entered as well.The application allows users to track their expenses daily, weekly, monthly, and yearly in terms of summary, bar graphs, and pie-charts.

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Introduction to Organization . . . . .	1
1.2	Introduction To Project . . . . .	1
1.3	Project Category(Internet based, Application or System Development, Research based ,Industry Automation, Network or System Administration) . . . . .	2
1.4	Objectives . . . . .	2
1.5	Problem Formulation . . . . .	2
1.6	Identification/Reorganization of Need . . . . .	2
1.7	Existing System . . . . .	2
1.8	Proposed System . . . . .	2
1.9	Unique Features of the System . . . . .	2
<b>2</b>	<b>Requirement Analysis and System Specification</b>	<b>3</b>
2.1	Feasibility Study (Technical,Economical,Operational) . . . . .	3
2.1.1	Types of Feasibility . . . . .	4
2.1.1.1	Technical Feasibility . . . . .	4
2.1.1.2	Economic Feasibility . . . . .	5
2.1.1.3	Operational Feasibility . . . . .	5
2.2	Software Requirement Specification Document . . . . .	6
2.2.1	Data Requirement . . . . .	6
2.2.2	Performance Requirement . . . . .	6
2.2.3	Functional Requirement . . . . .	6
2.3	Intended User . . . . .	6
2.4	Features . . . . .	6
2.5	Expected hurdles . . . . .	7
2.6	SDLC model to be used . . . . .	7
2.6.1	Agile Model . . . . .	7
<b>3</b>	<b>System Design</b>	<b>8</b>
3.1	Function Oriented . . . . .	8
3.2	Detail Design . . . . .	8

<b>4</b>	<b>Implementation ,Testing and Maintenance</b>	<b>9</b>
4.1	Introduction to Languages,IDEs,Tools and Technologies used for Implementation	9
4.1.1	Technologies used . . . . .	9
4.1.1.1	Firebase . . . . .	9
4.1.1.2	Android . . . . .	11
4.1.2	Language used . . . . .	13
4.1.2.1	JAVA . . . . .	13
4.1.2.2	XML . . . . .	15
4.1.3	IDE used . . . . .	15
4.1.3.1	Android Studio . . . . .	15
4.1.4	Introduction to L <sup>A</sup> T <sub>E</sub> X . . . . .	16
4.1.4.1	Typesetting . . . . .	17
4.1.4.2	Installing L <sup>A</sup> T <sub>E</sub> X on System . . . . .	17
4.1.4.3	Pdftscreen L <sup>A</sup> T <sub>E</sub> X . . . . .	18
4.2	Coding standards of Language used . . . . .	19
4.2.1	Introduction . . . . .	19
4.2.2	Source file basics . . . . .	19
4.3	GANTT chart . . . . .	19
4.4	Testing Techniques and Test Plans . . . . .	19
4.4.1	xUnit Framework . . . . .	19
4.4.2	JUnit . . . . .	19
4.4.3	Robotium Android Testing Tool . . . . .	20
4.4.4	MonkeyRunner Android App Testing . . . . .	20
<b>5</b>	<b>Results and Discussions</b>	<b>21</b>
5.1	User Interface Representation (Of Respective Project) . . . . .	21
5.1.1	Brief Description of Various Modules of the system . . . . .	21
5.2	Snapshots of system with brief detail of each . . . . .	21
5.3	Back Ends Representation (Database to be used) . . . . .	21
5.3.1	Snapshots of Database Tables with brief description . . . . .	21
<b>6</b>	<b>Conclusion and Future Scope</b>	<b>22</b>
6.1	Conclusion . . . . .	22
6.2	Future Scope . . . . .	22

LIST OF FIGURES

2.1	Agile Model . . . . .	7
4.1	Firebase Console . . . . .	9
4.2	Firebase Products . . . . .	10
4.3	Android Anatomy . . . . .	11
4.4	Activity Life Cycle . . . . .	12
4.5	Android Studio . . . . .	15
4.6	L <sup>A</sup> T <sub>E</sub> X Logo . . . . .	16
4.7	Donald Knuth, Inventor Of T <sub>E</sub> X typesetting system . . . . .	16

## 1.1 Introduction to Organization

Forget about not remembering where you spent your money! Keep track of your expenses and get a lot of statistics to know in what you spend more and you can evaluate where you can cut some expenses! Take advantage of this tool that will help you to manage your money and be informed all the time. You can also configure to get notifications when you have to make some payments every month so you don't forget! This Application is made for any user who wants or wish to keep a better track of their expenses in order to keep them in a same place.

## 1.2 Introduction To Project

With the launch and increase in sales of smartphones over the last few years, people are using mobile applications to get their work done, which makes their lives easier. Mobile applications comprise various different categories such as Entertainment, Sports, Lifestyle, Education, Games, Food and Drink, Health and Fitness, Finance, etc. This Expense Tracker application falls in the Finance Category and serves the important purpose of managing finances which is a very important part of one's life. The software product went through the design, development, and the testing phase as a part of the Software Development Lifecycle.

The applications interface is designed using custom art elements, the functionality is implemented using Android SDK, and the phase of testing the product was accomplished successfully. The application is not much user intensive but just comprises of having them enter the expense amount, date, category, merchant and other optional attributes (taking picture of the receipts, entering notes about the expense, adding subcategories to the categories). With this entered information, the user is able to see the expense details daily, weekly, monthly, and yearly in figures, graphs. All these topics have been explained in detail in their respective chapters

- 1.3 Project Category(Internet based, Application or System Development, Research based ,Industry Automation, Network or System Administration)
- 1.4 Objectives
- 1.5 Problem Formulation
- 1.6 Identification/Reorganization of Need
- 1.7 Existing System
- 1.8 Proposed System
- 1.9 Unique Features of the System



## CHAPTER 2

# REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

### 2.1 Feasibility Study (Technical,Economical,Operational)

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Carrying out a feasibility study involves information assessment, information collection and report writing. The information assessment phase identifies the information that is required to answer the three questions set out above. Once the information has been identified, you should question information sources to discover the answers to these questions Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

A feasibility study is designed to provide an overview of the primary issues related to a business idea. The purpose is to identify any make or break issues that would prevent your business from being successful in the marketplace. In other words, a feasibility study determines whether the business idea makes sense. A thorough feasibility analysis provides a lot of information necessary for the business plan. For example, a good market analysis is necessary in order to determine the projects feasibility. This information provides the basis for the market section of the business plan.

The document provide the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards.

Objectives of feasibility study are listed below.

- To analyze whether the software will meet organizational requirements

- To determine whether the software can be implemented using the current technology and within the specified budget and schedule
- To determine whether the software can be integrated with other existing software.

### 2.1.1 Types of Feasibility

Various types of feasibility that are commonly considered include technical feasibility, operational feasibility, and economic feasibility.

#### 2.1.1.1 Technical Feasibility

Technical feasibility is one of the first studies that must be conducted after the project has been identified. In large engineering projects consulting agencies that have large staffs of engineers and technicians conduct technical studies dealing with the projects.

When writing a feasibility report, the following should be taken to consideration:

- A brief description of the business to assess more possible factors which could affect the study
- The part of the business being examined
- The human and economic factor
- The possible solutions to the problem

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed. Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. Technical feasibility also performs the following tasks.

- Analyzes the technical skills and capabilities of the software development team members
- Determines whether the relevant technology is stable and established
- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

#### **2.1.1.2 Economic Feasibility**

The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

Software is said to be economically feasible if it focuses on the issues listed below.

- Cost incurred on software development to produce long-term gains for an organization.
- Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis).
- Cost of hardware, software, development team, and training.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

#### **2.1.1.3 Operational Feasibility**

If the system is not easy to operate, than operational process would be difficult. The operator of the system should be given proper training. The system should be made such that the user can interface the system without any problem.

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture, and existing business processes.

Operational feasibility also performs the following tasks.

- Determines whether the problems anticipated in user requirements are of high priority.
- Determines whether the solution suggested by the software development team is acceptable.

- Analyzes whether users will adapt to a new software.
- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?
- The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

This system proposed will have a very user friendly interface, a nave user will be able to understand the user interface in seconds. Itll have basic yet intriguing interface with a list various pdf categories, fab button which indicates that pdf upload using it and many other functions which is easily understandable.

## 2.2 Software Requirement Specification Document

### 2.2.1 Data Requirement

For this project, no specific data collections were required, all that was needed for testing the project. Initial Pdf or Ebook was uploaded to check sytem works in right way or not.

### 2.2.2 Performance Requirement

Basis on resource requirements, this application will be able to run on pretty low-end devices with 512MB of ram and 800Mhz processor.

### 2.2.3 Functional Requirement

The application will have different kind of functions like uploading Ebooks of various categories. User can download and share that uploaded content.

## 2.3 Intended User

This Application is made for any user who wants or wish to share Pdf or Ebook of various categories. Easy platform for the user who love to read ebook instead of buying it.

## 2.4 Features

- Upload and share pdf easily.
- Arranging Pdf or Ebook in various categories.
- User have access to delete uploading content(content uploaded by him/her self only).
- Feedback option given so that user can share his views.

## 2.5 Expected hurdles

- An algorithm for working of feedback notification to admins is one of the biggest hurdle in the project.
- Libraries for Android studio needs to be added into the imported from various sources such as GitHub, Firebase.
- Dependencies of the different libraries used in the project.

## 2.6 SDLC model to be used

### 2.6.1 Agile Model

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release. An Agile software life cycle is much different as compared to traditional software development frameworks like Waterfall. In Agile, more emphasis is given to sustained and quick development of product features rather than spending more time during the initial project planning, and analyzing the actual requirements. The Agile team develops the product through a series of iterative cycles known as sprints. Besides development activity, other aspects pertaining to development such as product analysis, designing the product features, developing the functionality, and testing the development for bugs are also carried out during the sprints. The incremental cycles should always produce a shippable product release that can be readily deployed.

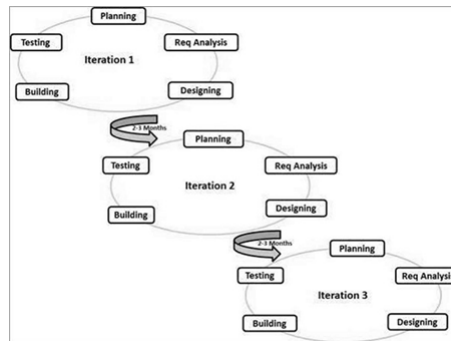


Figure 2.1: Agile Model

### 3.1 Function Oriented

Object-oriented design is the process of planning a system of interacting objects for the purpose of solving a software problem. It is one approach to software design. An object contains encapsulated data and procedures grouped together to represent an entity. The 'object interface' defines how the object can be interacted with. An object-oriented program is described by the interaction of these objects. Object-oriented design is the discipline of defining the objects and their interactions to solve a problem that was identified and documented during object-oriented analysis.

### 3.2 Detail Design

## CHAPTER 4

# IMPLEMENTATION ,TESTING AND MAINTENANCE

## 4.1 Introduction to Languages,IDEs,Tools and Technologies used for Implementation

### 4.1.1 Technologies used

#### 4.1.1.1 Firebase

Firebase helps you build better mobile apps and grow your business.

Firebase is a mobile platform from Google offering a number of different features that you can pick n mix from. Specifically, these features revolve around cloud services, allowing users to save and retrieve data to be accessed from any device or browser. This can be useful for such things as cloud messaging, hosting, crash reporting, notifications, analytics and even earning money through AdMob.

It works with Android apps, iOS apps and web apps and best of all: its free!

- **Setting up a project**

Before you can do anything with Firebase, you first need to create an account. You can do this over at [firebase.google.com](https://firebase.google.com/).

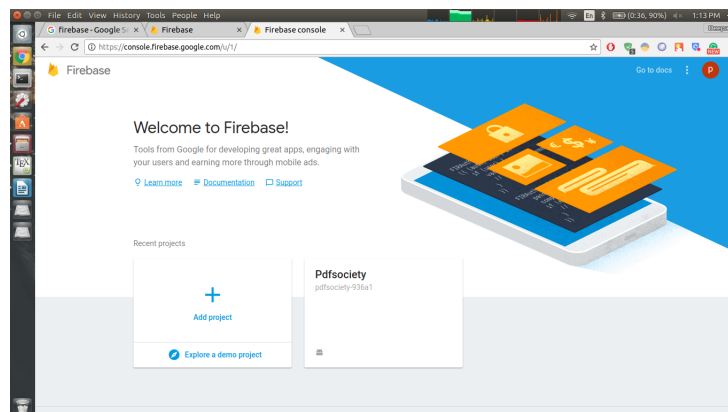


Figure 4.1: Firebase Console

## • Firebase Products

Mix and match complementary products



Figure 4.2: Firebase Products

### – Firebase Authentication

Firebase Authentication aims to make building secure authentication systems easy, while improving the sign-in and onboarding experience for end users. It provides an end-to-end identity solution, supporting email and password accounts, phone auth, and Google, Twitter, Facebook, and GitHub login, and more.

Most apps need to know the identity of a user. Knowing a user's identity allows an app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices. Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

Firebase Authentication integrates tightly with other Firebase services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend.

### – Firebase Realtime Database

Store and sync data with our NoSQL cloud database. Data is synced across all clients in realtime, and remains available when your app goes offline.

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

### – Cloud Storage



Cloud Storage is built for app developers who need to store and serve user-generated content, such as photos or videos.

Cloud Storage for Firebase is a powerful, simple, and cost-effective object storage service built for Google scale. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads for your Firebase apps, regardless of network quality. You can use our SDKs to store images, audio, video, or other user-generated content. On the server, you can use Google Cloud Storage, to access the same files.

#### 4.1.1.2 Android

Android provides a rich application framework that allows you to build innovative apps and games for mobile devices in a Java language environment. The documents listed in the left navigation provide details about how to build apps using Android's various APIs. The various fundamental concepts about the Android app framework:



Figure 4.3: Android Anatomy

Apps provide multiple entry points. Android apps are built as a combination of distinct components that can be invoked individually. For instance, an individual activity provides a single screen for a user interface, and a service independently performs work in the background. From one component you can start another component using an intent. You can even start a component in a different app, such as an activity in a maps app to show an address. This model provides multiple entry points for a single app and allows any app to behave as a user's "default" for an action that other apps may invoke.

Apps adapt to different devices, Android provides an adaptive app framework that allows you to provide unique resources for different device configurations. For example, you can create different XML layout files for different screen sizes and the system determines which layout to apply based on the current device's screen size. You can query the availability of device features at runtime if any app features require specific hardware such as a camera. If necessary, you can also declare features your app requires so app markets such as Google Play Store do not allow installation on devices that do not support that feature. Android comes with an Android market which is an online software store. It was developed by Google.

It allows Android users to select, and download applications developed by third party developers and use them. There are around 2.0 lack+ games, application and widgets available

on the market for users. Android applications are written in java programming language. Android is available as open source for developers to develop applications which can be further used for selling in android market. There are around 200000 applications developed for android with over 3 billion+ downloads. Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. For software development, Android provides Android SDK (Software development kit).

- **Activity Lifecycle** Activities in the system are managed as an activity stack. When a new activity is started, it is placed on the top of the stack and becomes the running activity the previous activity always remains below it in the stack, and will not come to the foreground again until the new activity exits. An activity has essentially four states: If an activity in the foreground of the screen (at the top of the stack), it is active or running.

If an activity has lost focus but is still visible (that is, a new non-full-sized or transparent activity has focus on top of your activity), it is paused. A paused activity is completely alive (it maintains all state and member information and remains attached to the window manager), but can be killed by the system in extreme low memory situations.

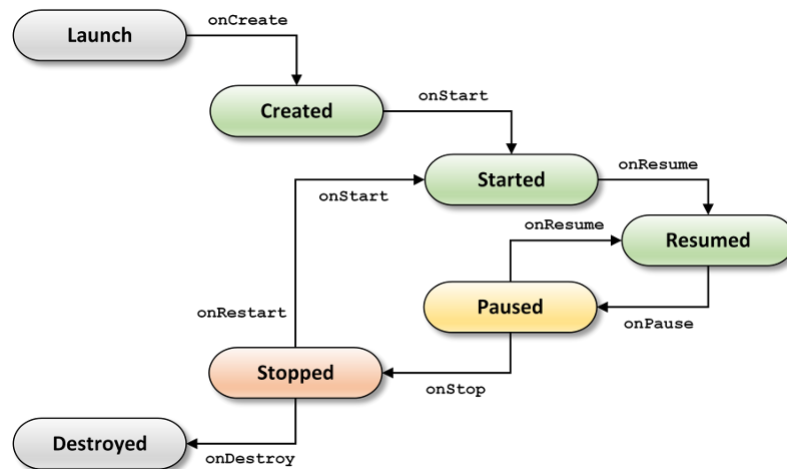


Figure 4.4: Activity Life Cycle

If an activity is completely obscured by another activity, it is stopped. It still retains all state and member information, however, it is no longer visible to the user so its window is hidden and it will often be killed by the system when memory is needed elsewhere.

If an activity is paused or stopped, the system can drop the activity from memory by either asking it to finish, or simply killing its process. When it is displayed again to the user, it must be completely restarted and restored to its previous state.

- **Libraries used in making this application**

- **Android Support Library** Android Support Library, Card View, Recycler View, Material : The application will follow Material design guidelines.

## 4.1.2 Language used

### 4.1.2.1 JAVA

Java is a platform-independent programming language used to create secure and robust application that may run on a single computer or may be distributed among servers and clients over a network.

Java features such as platform-independency and portability ensure that while developing Java EE enterprise applications, you do not face the problems related to hardware , network , and the operating system.

Java was started as a project called "Oak" by James Gosling in June 1991. Gosling's goals were to implement a virtual machine and a language that had a familiar C like notation but with greater uniformity and simplicity than C/C++. The First publication of Java 1.0 was released by Sun Microsystems in 1995. It made the promise of "Write Once, Run Anywhere", with free runtimes on popular platforms. In 2006-2007 Sun released java as open source and andplatform independent soft-ware. Over time new enhanced versions of Java have been released. The current version of Java is Java 1.7 which is also known as Java 7. The Java virtual machine (JVM) is a software implementation of a computer that executes programs like a real machine. The Java virtual machine is written specifically for a specific operating system, e.g. for Linux a special implementation is required as well as for Windows.

Java programs are compiled by the Java compiler into bytecode. The Java virtual machine interprets this bytecode and executes the Java program. The Java runtime environment (JRE) consists of the JVM and the Java class libraries and contains the necessary functionality to start Java programs. The JDK contains in addition the development tools necessary to create Java programs. The JDK consists therefore of a Java compiler, the Java virtual machine, and the Java class libraries.

The characteristics and features of java are as follows :

- **Simple** Simple Java is a simple language because of its various features, Java Doesn't Support Pointers , Operator Overloading etc. It doesn't require unreferenced object because java support automatic garbage collection. Java provides bug free system due to the strong memory management.
- **OOPS** Object-Oriented Programming Language (OOPs) is the methodology which provide software development and maintenance by using object state, behavior Programming Language must , and have properties. the Object Oriented following characteristics.
  - Encapsulation
  - Polymorphism
  - Inheritance
  - Abstraction

As the languages like Objective C, C++ fulfill the above four characteristics yet they are not fully object oriented languages because they are structured as well as object oriented languages. In java everything is an Object. Java can be easily extended since it is based on the Object model.

- **Secure** Secure Java is Secure Language because of its many features it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption. Java does not support pointer explicitly for the memory. All Program Run under the sandbox.
- **Robust** Robust Java was created as a strongly typed language. Data type issues and problems are resolved at compile-time, and implicit casts of a variable from one type to another are not allowed.
- **Platform-independent** Platform-independent Java Language is platform-independent due to its hardware and software environment. Java code can be run on multiple platforms e.g. windows, Linux, sun Solaris, Mac/Os etc. Java code is compiled by the compiler and converted into byte code. This byte code is a platform independent code because it can be run on multiple platforms i.e. Write Once and Run Anywhere(WORA).
- **Architectural Neutral** Architecture neutral It is not easy to write an application that can be used on Windows , UNIX and a Macintosh. And its getting more complicated with the move of windows to non Intel CPU architectures. Java takes a different approach. Because the Java compiler creates byte code instructions that are subsequently interpreted by the java interpreter, architecture neutrality is achieved in the implementation of the java interpreter for each new architecture.
- **Portable** Portable Java code is portable. It was an important design goal of Java that it be portable so that as new architectures(due to hardware, operating system, or both) are developed, the java environment could be ported to them. In java, all primitive types(integers, longs, floats, doubles, and so on) are of defined sizes, regardless of the machine or operating system on which the program is run. This is in direct contrast to languages like C and C++ that leave the sized of primitive types up to the compiler and developer. Additionally, Java is portable because the compiler itself is written in Java.
- **Dynamic** Dynamic Because it is interpreted , Java is an extremely dynamic language, At runtime, the java environment can extends itself by linking in classes that may be located on remote servers on a network(for example, the internet) At runtime, the java interpreter performs name resolution while linking in the necessary classes. The Java interpreter is also responsible for determining the placement of object in memory. These two features of the Java interpreter solve the problem of changing the definition of a class used by other classes.
- **Interpreted** We all know that Java is an interpreted language as well. With an interpreted language such as Java, programs run directly from the source code. The interpreter program reads the source code and translates it on the fly into computations. Thus, Java as an interpreted language depends on an interpreter program. The versatility of being platform independent makes Java to outshine from other languages. The source code to be written and distributed is platform independent. Another advantage of Java as an interpreted language is its error debugging quality. Due to this any error occurring in the program gets traced. This is how it is different to work with Java.
- **High performance** High performance For all but the simplest or most infrequently used applications, performance is always a consideration for most applications, includ-

ing 21graphics-intensive ones such as are commonly found on the world wide web, the performance of java is more than adequate.

- **Multithreading** Writing a computer program that only does a single thing at a time is an artificial constraint that lived with in most programming languages. With java, we no longer have to live with this limitation. Support for multiple, synchronized threads is built directly into the Java language and runtime environment. Synchronized threads are extremely useful in creating distributed, network-aware applications. Such as application may be commu-nicating with a remote server in one thread while interacting with a user in a different thread.
- **Distributed** Java facilitates the building of distributed application by a collection of classes for use in networked applications. By using javas URL (Uniform Resource Locator) class, an application can easily access a remote server. Classes also are provided for establishing socket-level connections.

#### 4.1.2.2 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 through use of tags that can be created and defined by users. Much like natural language is extensible (that is, can grow) when speakers create new words and agree on what they mean, XML is a markup language that can grow when users create new elements and agree on what they mean. For example, XML can markup machine-readably that apples and bananas are types of fruit, which is semantically deeper than the purpose of HTML. However, HTML is useful for display of content; often HTML is used to display XML content after transformation with XSL.

#### 4.1.3 IDE used

##### 4.1.3.1 Android Studio



Figure 4.5: Android Studio

Android Studio is the official integrated development environment (IDE) for Google’s Android operating system, built on JetBrains’ IntelliJ IDEA software and designed specifically

for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It 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. The current stable version is 2.3.3, released in June 2017. Next major update, version 3.0, is in preview stage as of September 2017.

It offer tools custom-tailored for Android developers, including rich code editing, debugging, testing, and profiling tools.

- **System Requirements** Android application development on either of the following operating systems

Microsoft Windows 10/8/7/Vista/2003 (32 or 64-bit)  
Mac OS X 10.8.5 or higher, up to 10.9 (Mavericks)  
GNOME or KDE desktop

Second point is that all the required tools to develop Android applications are open source and can be downloaded from the Web. Following is the list of software's you will need before you start your Android application programming.

Java JDK5 or later version  
Java Runtime Environment (JRE) 6  
Android Studio

#### 4.1.4 Introduction to $\text{\LaTeX}$



Figure 4.6:  $\text{\LaTeX}$  Logo

$\text{\LaTeX}$ , I had never heard about this term before doing this project, but when I came to know about it's features, it is just excellent.  $\text{\LaTeX}$ (pronounced /letk/, /letx/, /ltx/, or /ltk/) is a document markup language and document preparation system for the  $\text{\TeX}$  typesetting program. Within the typesetting system, its name is styled as  $\text{\LaTeX}$ .

Within the typesetting system, its name is styled as  $\text{\LaTeX}$ . The term  $\text{\LaTeX}$  refers only to the language in which documents are written, not to the editor used to write those documents. In order to create a document in  $\text{\LaTeX}$ , a .tex file must be created using some form of text editor. While most text editors can be used to create a  $\text{\LaTeX}$  document, a number of editors have been created specifically for working with  $\text{\LaTeX}$ .



Figure 4.7: Donald Knuth, Inventor Of  $\text{\TeX}$  typesetting system

$\text{\LaTeX}$  is most widely used by mathematicians, scientists, engineers, philosophers, linguists, economists and other scholars in academia. As a primary or intermediate format, e.g., translating DocBook and other XML-based formats to PDF,  $\text{\LaTeX}$  is used because of the high quality of typesetting achievable by  $\text{\TeX}$ . The typesetting system offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout and bibliographies.

$\text{\LaTeX}$  is intended to provide a high-level language that accesses the power of  $\text{\TeX}$ .  $\text{\LaTeX}$  essentially comprises a collection of  $\text{\TeX}$  macros and a program to process  $\text{\LaTeX}$  documents. Because the  $\text{\TeX}$  formatting commands are very low-level, it is usually much simpler for end-users to use  $\text{\LaTeX}$ .

#### 4.1.4.1 Typesetting

$\text{\LaTeX}$  is based on the idea that authors should be able to focus on the content of what they are writing without being distracted by its visual presentation. in preparing a  $\text{\LaTeX}$  document, the author specifies the logical structure using familiar concepts such as chapter, section, table, figure, etc., and lets the  $\text{\LaTeX}$  system worry about the presentation of these structures. it therefore encourages the separation of layout from content while still allowing manual typesetting adjustments where needed.

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\LaTeX}
\begin{document}
  \maketitle
  \LaTeX{} is a document preparation system
  for the \TeX{} typesetting program.
  \par
   $E=mc^2$ 
\end{document}
```

#### 4.1.4.2 Installing $\text{\LaTeX}$ on System

Installation of  $\text{\LaTeX}$  on personal system is quite easy. As i have used  $\text{\LaTeX}$  on Ubuntu 13.04 so i am discussing the installation steps for Ubuntu 13.04 here:

- Go to terminal and type

*sudo apt-get install texlive-full*

- Your Latex will be installed on your system and you can check for manual page by typing.

*man latex*

in terminal which gives manual for latex command.

- To do very next step now one should stick this to mind that the document which one is going to produce is written in any type of editor whether it may be your most common usable editor Gedit or you can use vim by installing first vim into your system using command.

*sudo apt-get install vim*

- After you have written your document it is to be embedded with some set of commands that Latex uses so as to give a structure to your document. Note that whenever you wish your document to be looked into some other style just change these set of commands.
- When you have done all these things save your piece of code with .tex format say test.tex. Go to terminal and type

*latex path of the file test.tex Or pdflatex path of the file test.tex*

*eg: pdflatex test.tex*

for producing pdf file simultaneously.

After compiling it type command

*evince filename.pdf*

*eg: evince test.pdf*

To see output pdf file.

#### 4.1.4.3 Pdfscreen $\LaTeX$

There are some packages that can help to have unified document using  $\LaTeX$ . Example of such a package is pdfscreen that let the user view its document in two forms-print and screen. Print for hard copy and screen for viewing your document on screen. Download this package from [www.ctan.org/tex-archive/macros/latex/contrib/pdfscreen/](http://www.ctan.org/tex-archive/macros/latex/contrib/pdfscreen/). Then install it using above mention method.

To test it the test code is given below:-

Just changing print to screen gives an entirely different view. But for working of pdfscreen another package required are comment and fancybox.

The fancybox package provides several different styles of boxes for framing and rotating content in your document. Fancybox provides commands that produce square-cornered boxes with single or double lines, boxes with shadows, and round-cornered boxes with normal or



bold lines. You can box mathematics, floats, center, flushleft, and flushright, lists, and pages.

Whereas comments package selectively include/excludes portions of text. The comment package allows you to declare areas of a document to be included or excluded. One need to make these declarations in the preamble of your file. The package uses a method for exclusion that is pretty robust, and can cope with ill-formed bunches of text.

So these extra packages needed to be installed on system for the proper working of pdfscreen package.

## 4.2 Coding standards of Language used

### 4.2.1 Introduction

This document is the definition of Google’s coding standards for source code in the Java Programming Language. A Java source file is described as being in Google Style if and only if it adheres to the rules herein. Like other programming style guides, the issues covered span not only aesthetic issues of formatting, but other types of conventions or coding standards as well. However, this document focuses primarily on the hard-and-fast rules that we follow universally, and avoids giving advice that isn’t clearly enforceable (whether by human or tool).

### 4.2.2 Source file basics

- File name, the source file name consists of the case-sensitive name of the top-level class it contains (of which there is exactly one), plus the .java extension.
- File encoding: UTF-8 Source files are encoded in UTF-8. Aside from the line terminator sequence, the ASCII horizontal space character (0x20) is the only whitespace character that appears anywhere in a source file. This implies that:
- All other whitespace characters in string and character literals are escaped.
- Tab characters are not used for indentation.

## 4.3 GANTT chart

## 4.4 Testing Techniques and Test Plans

### 4.4.1 xUnit Framework

xUnit is the collective name for several unit testing frameworks that derive their structure and functionality from Smalltalk’s SUnit. SUnit, designed by Kent Beck in 1998, was written in a highly structured object-oriented style, which lent easily to contemporary languages such as Java and C. Following its introduction in Smalltalk the framework was ported to Java by Kent Beck and Erich Gamma and gained wide popularity, eventually gaining ground in the majority of programming languages in current use. The names of many of these frameworks are a variation on "SUnit", usually replacing the "S" with the first letter (or letters) in the

name of their intended language ("JUnit" for Java, "RUnit" for R etc.). These frameworks and their common architecture are collectively known as "xUnit".

#### 4.4.2 JUnit

JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks which is collectively known as xUnit that originated with SUnit. JUnit is linked as a JAR at compile-time; the framework resides under package junit.framework for JUnit 3.8 and earlier, and under package org.junit for JUnit 4 and later. A research survey performed in 2013 across 10,000 Java projects hosted on GitHub found that JUnit, (in a tie with slf4j-api), was the most commonly included external library. Each library was used by 30.7 percentage of projects.

#### 4.4.3 Robotium Android Testing Tool

Robotium is one the first and frequently utilized automated testing tools for software supported on Android. Robotium is a free Android UI testing tool. It is suitable for tests automation for different Android versions and sub-versions. Software developers often describe it as Selenium for Android. Tests created by Robotium are written in Java. In fact, Robotium is a library for unit tests. But it takes much time and efforts to create tests by means of Robotium, as one must work with the program source code in order to automate tests. The tool is also unsuitable for interaction with system software; it cannot lock and unlock a smartphone or a tablet. There is no Record and Play function in Robotium, and it does not provide screenshots.

#### 4.4.4 MonkeyRunner Android App Testing

MonkeyRunner Android App Testing MonkeyRunner is one of popular Android Testing tools used for automating of functional tests for Android software. This tool is more low-level than Robotium is. One does not have to deal with the source code in order to automate tests. The tests are written in Python, one may use a recording tool for creating tests. MonkeyRunner can run tests on real devices connected to a PC or emulators. The tool has an API what allows it to control a smartphone, a tablet or an emulator from outside of Android code. A significant disadvantage of the mobile app testing tool is that it is necessary to write scripts for each device. Another problem of MonkeyRunner is that the tests require adjustments each time when user interface of the tested program is changed.

## CHAPTER 5

## RESULTS AND DISCUSSIONS

### **5.1 User Interface Representation (Of Respective Project)**

#### **5.1.1 Brief Description of Various Modules of the system**

### **5.2 Snapshots of system with brief detail of each**

### **5.3 Back Ends Representation (Database to be used)**

#### **5.3.1 Snapshots of Database Tables with brief description**

## CHAPTER 6

## CONCLUSION AND FUTURE SCOPE

### 6.1 Conclusion

Earlier user records are maintained in a diary which leads to the wastage of paper. Thus makes the system not ecofriendly. It .After working on the Android money management we would like to conclude that with the help of this app, maintenance of user money records becomes simple and easier.

### 6.2 Future Scope

The app uses android technology which has evergreen scope. The app obviously has a bright future scope as there is test which includes different level and type of questions .. Moreover in Future, one can see his/her earlier records regarding the payments and extra diets. The platform used is android. Nowadays Android has become very popular which is an open-source, Linux-based operating system mainly designed by Google for smart-phones and tablets.

Many mobile Apps development industries are considering Android Application Development as one of the best business opportunities, for this they need to hire a lot of knowledgeable mobile application developer in future. This adds a big sign of scope of mobile Apps in future.

In the current job market of mobile application development, the need for inventive App developers is huge and still increasing. Android Apps development can also be taken up as a part time job. You can create your own applications at home and submit it to the Google Play store which can be downloaded by smart-phone users.

## BIBLIOGRAPHY

- [1] Introduction to Android: <http://developer.android.com/guide/index.html>
- [2] Android Training: <http://developer.android.com/training/index.html>.
- [3] Firebase Documentation: <https://firebase.google.com/docs/android/setup/>