**EXPENSE MANAGER**

**A PROJECT REPORT**

**Submitted By**

**MEENU DEVI**

**University Roll No. 190029014959**

**Submitted in partial fulfillment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATIONS**

**Under the Supervision of**

**Neelam.rawat**



**Submitted to**

**Department Of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**(FEBRUARY 2021)**

**CERTIFICATE**

Certified that **Meenu devi (University Roll No 1900290149059),**  **,** have carried out the project work having “**Expense Manager**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU**)** (formerly UPTU),Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**Meenu Devi**

**University Roll No1900290149059**

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

**Date:**

**Neelam.rawat**

**Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Signature of Internal Examiner Signature of External Examiner**

**Dr. Ajay Shrivastava**

**Head, Department of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**ABSTRACT**

Mobile applications stood top among usability and user convenience. Many applications are available in the market to manage personal and group expenses. Not many applications provides a comprehensive view of both use cases. In this project, we develop a mobile application that keeps track of user personal expenses, his/her personal contribution towards group expenses; maintain monthly incomes, recurring and adhoc payments. It provides information of "who owes who and by how much". The proposed application would eliminate sticky note, spreadsheet and ledger that cause confusions, data inconsistency problems while recording and splitting of expenses. With our application user can manage his expenses more effectively. This application will not only helps users to manage their expenses but also help marketing executives to plan marketing according to the needs of users.

**EXPENSE MANAGER** As the name itself suggests, this project is an attempt to manage our daily expenses in a more efficient and manageable way. Sometime we can’t remember where our money goes. And we can’t handle our cash flow.

For this problem, we need a solution that everyone can manage their expenses. So we decided to find an easier way to get rid of this problem. So, our application attempts to free the user with as much as possible the burden of manual calculation and to keep the track of the expenditure.

Instead of keeping a diary or a log of the expenses, this application enables the user to not just keep the control on the expenses but also to generate and save reports.

With the help of this application, the user can manage their expenses on a daily, weekly and monthly basis. Users can insert and delete transactions as well as can generate and save their reports.

The graphical representation of the application is the main part of the system as it appeals to the user more and is easy to understand.

**ACKNOWLEDGEMENTS**

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, Neelam rawatfor his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications**, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**Meenu Devi**

**TABLE OF CONTENTS**

Chapter 1 - Introduction

1.1 Project description

1.2 Project Scope

1.3 Hardware / Software used in Project

Chapter 2 – System Design

2.1 Introduction

2.2 System Architecture

2.3 Modules in the System

Chapter 3 - Form Design

3.1 Input / Output Form (Screenshot)

Chapter 4 - Coding

Chapter 5 - Testing

Chapter 6 - Refrences

**CHAPTER 1**

**INTRODUCTION**

* 1. **PROJECT DESCRIPION**

Mobile applications stood top among usability and user convenience. Many applications are available in the market to manage personal and group expenses. Not many applications provides a comprehensive view of both use cases. In this project, we develop a mobile application that keeps track of user personal expenses, his/her personal contribution towards group expenses; maintain monthly incomes, recurring and adhoc payments. It provides information of "who owes who and by how much". The proposed application would eliminate sticky note, spreadsheet and ledger that cause confusions, data inconsistency problems while recording and splitting of expenses. With our application user can manage his expenses more effectively. This application will not only helps users to manage their expenses but also help marketing executives to plan marketing according to the needs of users.

Mobile applications stood top among usability and user convenience. Many applications are available in the market to manage personal and group expenses. Not many applications provides a comprehensive view of both use cases. In this project, we develop a mobile application that keeps track of user personal expenses, his/her personal contribution towards group expenses; maintain monthly incomes, recurring and adhoc payments. It provides information of "who owes who and by how much". The proposed application would eliminate sticky note, spreadsheet and ledger that cause confusions, data inconsistency problems while recording and splitting of expenses. With our application user can manage his expenses more effectively. This application will not only helps users to manage their expenses but also help marketing executives to plan marketing according to the needs of users.

The Expense Manager is a mobile application intended to run on android device namely smart phone. Expense Manager is designed to efficiently cater the needs of users by eliminating imparting costs and settling vows to friends. The application encourages corresponding users help in who owes who, and for what. Aim is use better approaches to help users and their companions to share expenses easily. This new application will let bunch users and their companions to have detailed view inside this application around individual costs. The app allows its users to add a remark to an expense, click on the expense name in any expense list. Bill posting will have space for comments and notes container with a "Post" catch underneath. The Expense Manager has notification option to notify each time somebody adds a remark to an expense user is on, or user can withdraw to posted bill. The additional feature that we are going to add in this application that enable us to collect the sample data of users expenses and use this to study patterns of expenses in certain area or by specific kinds of spending for market analysis. These patterns can be derived using some data mining techniques [1][2][3][4] such as clustering, classification and association.

2. **BACKGROUND STUDY**

The idea of developing this project in platform arises with the frequent problems being experienced by people in sharing among them. RESEARCH ARTICLE OPEN ACCESS International Journal of Computer Techniques -– Volume 3 Issue 2, Mar- Apr 2016 ISSN :2394-2231 http://www.ijctjournal.org Page 61 Some of the concerns related dividing expenses are like maintaining a personal expense is a BIG problem, splitting the expenses among group is confusing. Some of the conventional methods used to tackle this problem in normal circumstances are like making use of a sticky note by normal users, Proficient people deal with this kind problems by using spreadsheet to record expenses and using a ledger to maintain large amounts data by especially by experts. As this shows that it is variable methods used by different people. This makes using this data inconsistent. There are still problems in areas like there is no assurance for data consistency, there are chances of critical inputs can be missed and the manual errors may creep in. The Data recorders are not always handy and it could be hectic process to have overall view of those expenses. We believe a handy design a handy mobile application which handles these problems. Such that app is capable of recording the expenses and giving comprehensive view with easy to use user interface and this app is intelligent enough to answer : ‘ Who owes who ?And by how much ??’

**3. RELATED WORK**

The mobile applications that are available in the market are very useful to the smartphone users and make their life easy. The expenses manager is also one of those applications, which much scope in daily life. As there are many similar applications available today we added some innovative features to make our application unique, easy to use and efficient. Apart from adding unique features like combining group expenses and personal expenses in to one application, we also added features like trends, estimations. Here, we have an idea of making use of application for the purpose of survey in the field of expenditures of user. This idea serves as main objective of research project. The research also includes syncing of the applications with some social networks and emails as well[5][6].

4. **METHODOLOGY**

This section of paper is very important and this will guide our team to successfully accomplish the goals set for research. Here, the research project methodology describes the steps and approaches to be fallowed to attain final product. As explained above our project is of splitting the expenses between the groups and also to efficiently manage the personal expenses as well. However, our projects will have additional features included as part of our research so that it makes our project unique in the market. These features would make the project more efficient and very useful for our users. Apart from the benefits user gets and there is an important use of the system that enables us to use the data of the user with his prior permissions for the purpose of data mining for several other functionalities to be applied in market by analyzing user expenses[7][8][9].

4.1 User registration/creation This application like most of the applications will have user login screen and option for registration. The user must register in this application when he/she is using for first time. However, the user who is already registered can login to the application using his/her login credentials that are created by the user at the time of registration.

4.2 Creating, alter of user groups

## Tools and Technologies

**Hardware**

**Processor**

Intel ® Core™ i3-2370 CPU @2.40GHz

**Installed Memory (RAM)**

1.00 GB or above

**System Type**

1. 32/64 bit Operating System
2. **Software Interface**

**Client-Side**

Android Mobile

**Software**

Android Studio

**Database Server**

SQLite Database

**Flaws in the current system**

* No offline data storage
* Overcrowded interface and inappropriate color schemes
* Unable to create multiple accounts
* Users get interrupted by annoying advertisements
* No privacy function
* Unable to generate PDF reports
* Unable to set budget mode (Weekly/Monthly)

## Features of Daily Expense Tracker Android App Project

* Create multiple accounts/budget
* Delete account
* Background color
* Modify Transactions
* Offline datastore
* Passcode security
* Selecting budget mode(Weekly/Monthly)
* Generate reports as PDF files
* Fully customizable categories
* Cash flow (Pie/Bar/Graph)
* Expenses percentage
* Carryover
* Show transaction note
* Currency Symbol

## Modules of Daily Expense Tracker Android App Project

The modules which are currently covered are:

**Add income/add expense**

This module deals with adding income and expenses. The user has both options available for adding income and expense. But there is a condition if the user hasn’t entered the amount yet then the user can’t enter expenses. When the user enters any transaction then that transaction will be added in both Spending and Transaction tabs. If the user wants to delete that transaction then the user has to long click the transaction available in the spending tab then that transaction will be deleted from both tabs.

**Modify Transactions**

If the user wants to delete that transaction then the user has to click the transaction available in the spending tab then that transaction will be deleted from both tabs.

**Filter Transaction view**

In the transaction tab, the user can filter the transactions. In the Spinner, users can select the day, month and year and then click the filter button and according to the day, month and year transactions will appear. If the user wants to filter the transactions only on the basis of day, for example, user-selected Monday then all transactions will appear that were made on Monday.

**PDF Report**

In the transaction, the tab user has an option available for creating a report in PDF. Users click on the PDF button then PDF report will be generated and the user can view that report and that report will be automatically saved in the device.

**Multiple Accounts**

Users can create multiple accounts. In the account tab. User has the option available for creating a new account.  Users will click the “+” sign button then a dialog will appear on the screen and the user can enter the name of the account then that name will be saved in the account tab. If a user wants to delete the particular account then the user has to l click the account name user want to delete. Then that account will be deleted.

**Transactions overview as Pie/Bar/Graph**

The user has three options available for graphical representation. When the user rotates the device then the pie chart will appear on the screen and also switch is available on the screen when the user will click on the bar chart will appear on the screen and when the user clicks on graph then Graph will appear on the screen.

**Themes**

At the top bar, the user has a setting option when the user clicks that then background option will appear user can select different background colors. After selecting the particular color background color will be changed.

**Passcode**

The passcode is available in setting option at the top bar. When the user clicks on the passcode switch when the user switches on then the passcode screen will appear and the user can choose the password and that password will be saved in the database. After that when the user will open the application user have to enter the passcode and that passcode will be matched with passcode saved in the database. If the user entered the wrong passcode then the error message will appear.

**Currency Symbol**

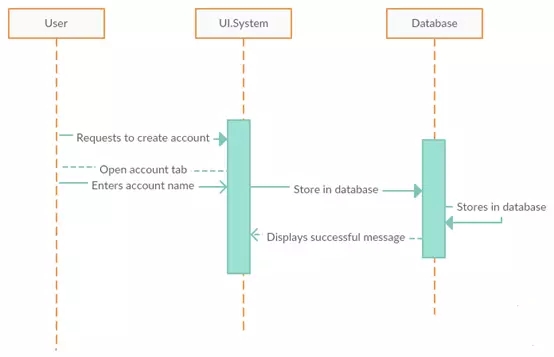
The currency symbol option is available at the top bar setting button. Users can select different currency symbols. If the user selects the dollar symbol then that symbol will appear on the spending tab.

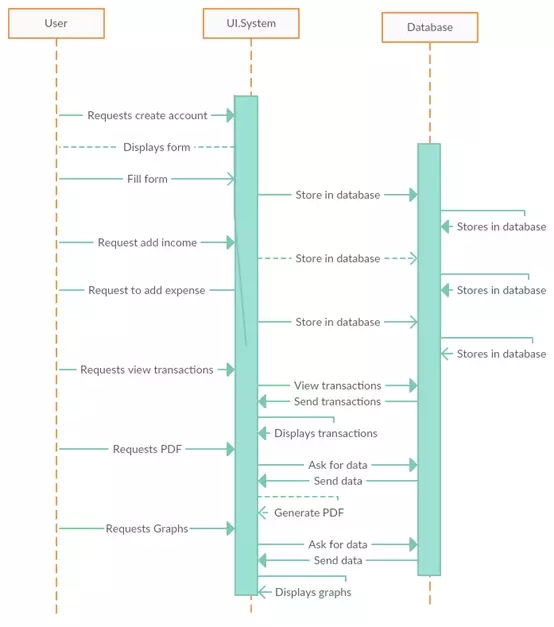
**Functional Requirements**

|  |  |
| --- | --- |
| **Identifier** | **Requirement** |
| **Req:1**Add transaction | This application will allow adding transaction. |
| **Req:2**Delete transaction | This application will allow the deleting transactions. |
| **Req:3**Amount spent in categories | This application will allow adding the amount spent in a particular category. |
| **Req:4**View all transactions | This application will allow viewing all previous transactions |
| **Req:5**Total amount | This application will allow seeing the total amount, the amount spent in different categories and balance left. |
| **Req:6** Overview | This application will allow viewing overall transactions. |
| **Req:7**Graph representation | This application will show the graph which will help the users to visualize the budget. |
| **Req:8**Pie representation | This application will show the pie. |
| **Req:9**Bar representation | This application will show the bar. |
| **Req:10**Change background | This application has the option to change the background. |
| **Req:11**Passcode | This application has the option to set a passcode for security. |
| **Req:12**Add multiple accounts | This application |
| **Req:13**Transaction time/date | This application has the ability to show the transaction time along with the date on which it was created. |
| **Req:14**Currency symbol | This application has many currency symbols as per user requirements. |
| **Req:15**Reminder | This application has the option to set a reminder to make the transaction. |
| **Req:16**Delete account | This application will generate PDF reports of the transactions. |
| **Req:17**PDF report | This application has the option to view and filter transactions by day, month and year. |
| **Req:18**Note | This application has the option to add a note about income and expenses. |

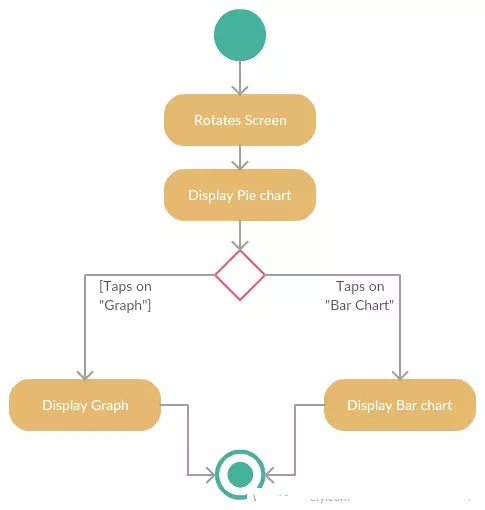
## Sequence Diagram of Daily Exp ense Tracker Android App Project

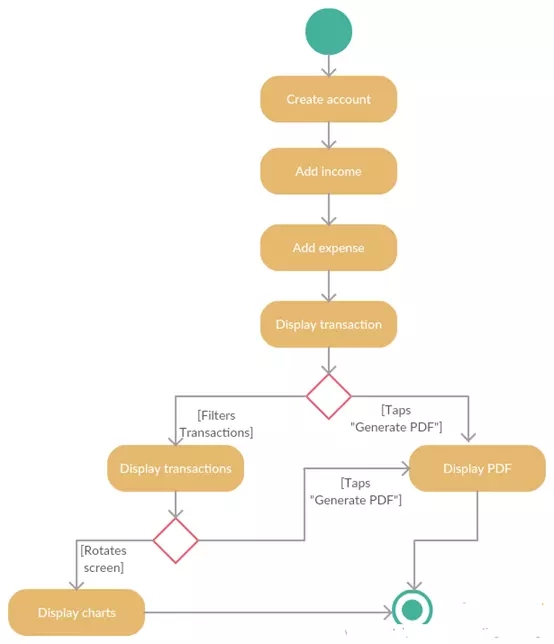
## 

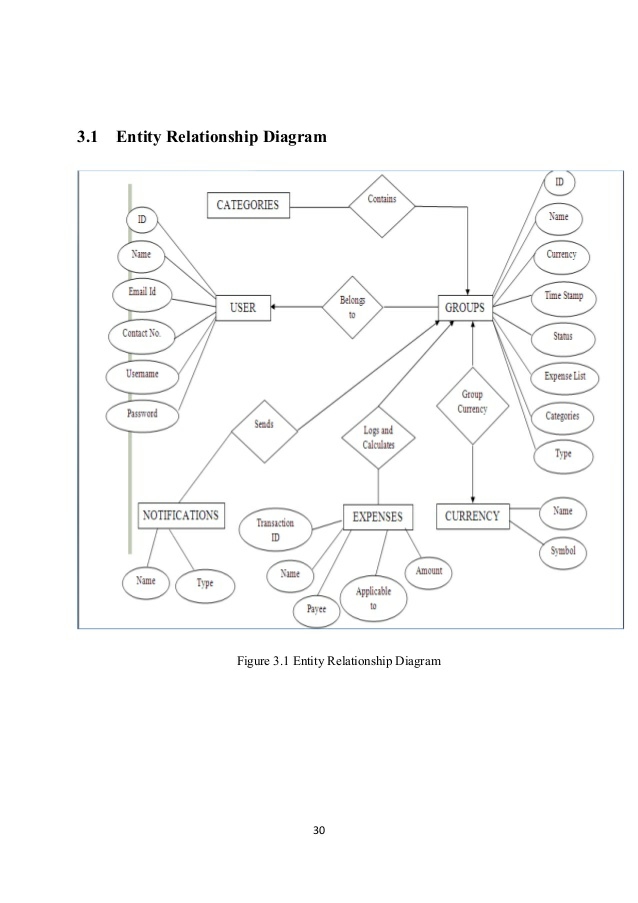
****

****

## Activity Diagram of Daily Expense Tracker Android App Project

\_\_\_

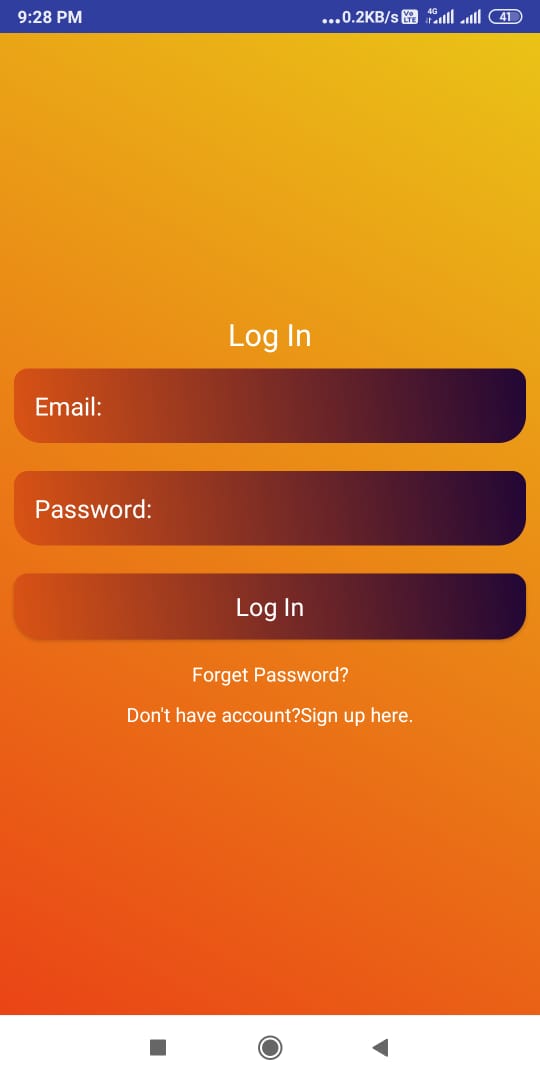


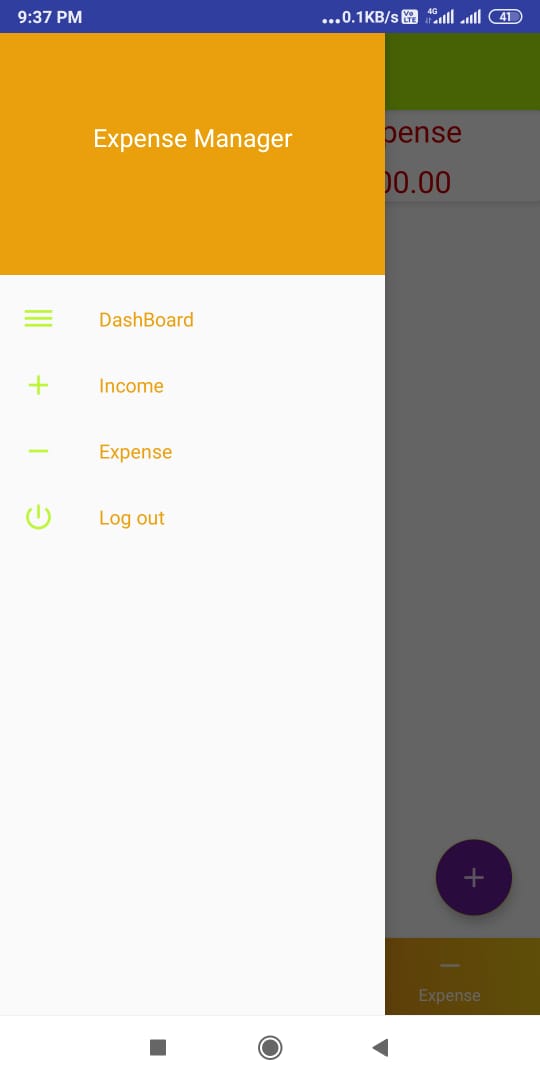


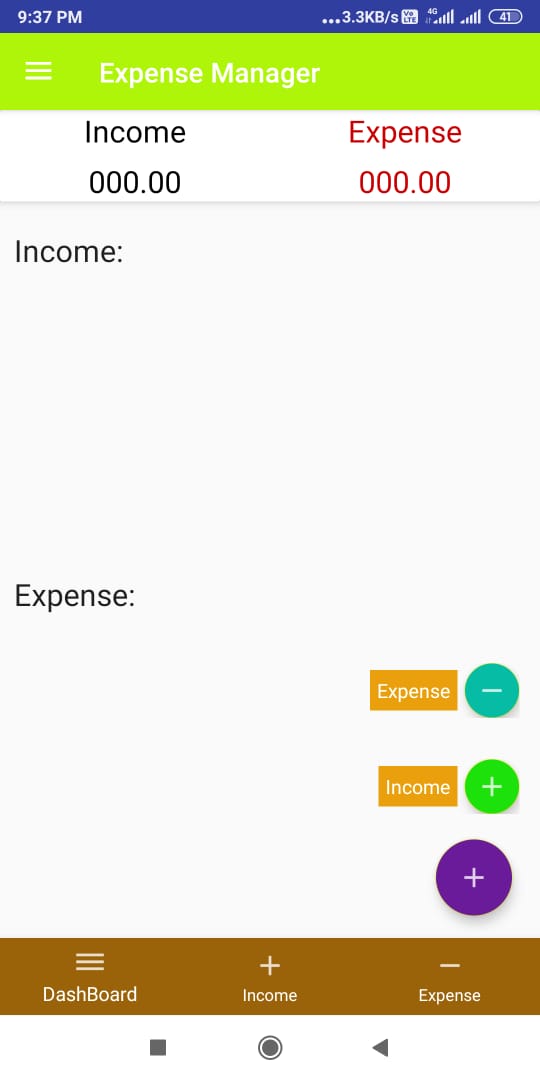
**CHAPTER 3**

**Form Design**

**3.1 Input / Output Form (Screenshot)**

****

****

****

**CHAPTER 4**

**Coding**

|  |
| --- |
|  |
|  |  |
|  | import android.content.Context; |
|  | import android.net.Uri; |
|  | import android.os.Bundle; |
|  | import android.support.annotation.Nullable; |
|  | import android.support.v4.app.Fragment; |
|  | import android.view.LayoutInflater; |
|  | import android.view.View; |
|  | import android.view.ViewGroup; |
|  | import android.widget.EditText; |
|  | import android.widget.Spinner; |
|  | import android.widget.Toast; |
|  |  |
|  | /\*\* |
|  | \* A simple {@link Fragment} subclass. |
|  | \* Activities that contain this fragment must implement the |
|  | \* {@link AddExpenseFragment.OnFragmentInteractionListener} interface |
|  | \* to handle interaction events. |
|  | \*/ |
|  | public class AddExpenseFragment extends Fragment { |
|  |  |
|  | private OnFragmentInteractionListener mListener; |
|  |  |
|  |  |
|  |  |
|  | public AddExpenseFragment() { |
|  | // Required empty public constructor |
|  | } |
|  |  |
|  |  |
|  | @Override |
|  | public View onCreateView(LayoutInflater inflater, ViewGroup container, |
|  | Bundle savedInstanceState) { |
|  | // Inflate the layout for this fragment |
|  | return inflater.inflate(R.layout.fragment\_add\_expense, container, false); |
|  | } |
|  |  |
|  |  |
|  | @Override |
|  | public void onActivityCreated(@Nullable Bundle savedInstanceState) { |
|  | super.onActivityCreated(savedInstanceState); |
|  | getView().findViewById(R.id.buttonAddExpense).setOnClickListener(new View.OnClickListener() { |
|  | @Override |
|  | public void onClick(View v) { |
|  | EditText editTextExpenseName = (EditText)getView().findViewById(R.id.editTextExpenseName); |
|  | EditText editTextAmount = (EditText)getView().findViewById(R.id.editTextAmount); |
|  | Spinner spinnerCategory = (Spinner)getView().findViewById(R.id.spinnerCategory); |
|  | if (editTextExpenseName.getText().toString().equals("") ){ |
|  | Toast.makeText(getActivity(), "Missing field Expense Name", Toast.LENGTH\_LONG).show(); |
|  | } |
|  | else if ( editTextAmount.getText().toString().equals("")){ |
|  | Toast.makeText(getActivity(), "Missing field Amount", Toast.LENGTH\_LONG).show(); |
|  | } |
|  | else{ |
|  | Expense expense = new Expense(); |
|  | expense.setExName(editTextExpenseName.getText().toString()); |
|  | expense.setExAmt(editTextAmount.getText().toString()); |
|  | expense.setExCategory(spinnerCategory.getSelectedItem().toString()); |
|  | expense.setExDate(); |
|  | mListener.gotoExpenseAppFragment(expense); |
|  | //Toast.makeText(getActivity(),"Added Expense",Toast.LENGTH\_SHORT).show(); |
|  | if (getActivity().getSupportFragmentManager().getBackStackEntryCount() > 0) { |
|  | getActivity().getSupportFragmentManager().popBackStack(); |
|  | } |
|  | } |
|  | } |
|  | }); |
|  |  |
|  | getView().findViewById(R.id.buttonCancel).setOnClickListener(new View.OnClickListener() { |
|  | @Override |
|  | public void onClick(View v) { |
|  | if (getActivity().getSupportFragmentManager().getBackStackEntryCount() > 0) { |
|  | getActivity().getSupportFragmentManager().popBackStack(); |
|  | } |
|  | } |
|  | }); |
|  | } |
|  |  |
|  | @Override |
|  | public void onAttach(Context context) { |
|  | super.onAttach(context); |
|  | if (context instanceof OnFragmentInteractionListener) { |
|  | mListener = (OnFragmentInteractionListener) context; |
|  | } else { |
|  | throw new RuntimeException(context.toString() |
|  | + " must implement OnFragmentInteractionListener"); |
|  | } |
|  | } |
|  |  |
|  | @Override |
|  | public void onDetach() { |
|  | super.onDetach(); |
|  | mListener = null; |
|  | } |
|  |  |
|  | /\*\* |
|  | \* This interface must be implemented by activities that contain this |
|  | \* fragment to allow an interaction in this fragment to be communicated |
|  | \* to the activity and potentially other fragments contained in that |
|  | \* activity. |
|  | \* <p> |
|  | \* See the Android Training lesson <a href= |
|  | \* "http://developer.android.com/training/basics/fragments/communicating.html" |
|  | \* >Communicating with Other Fragments</a> for more information. |
|  | \*/ |
|  | public interface OnFragmentInteractionListener { |
|  | // TODO: Update argument type and name |
|  | // void onFragmentInteraction(Uri uri); |
|  | void gotoExpenseAppFragment(Expense expense); |
|  | } |
|  | } |

**TESTING**

* 1. **INTRODUCTION**

Testing is the integral part of any System Development Life Cycle insufficient and interested application tends to crash and result in loss of economic and manpower investment besides user’s dissatisfaction and downfall of reputation.

“Software Testing can be looked upon as one among much process, an organization performs, and that provides the last opportunity to correct any flaws in the developed system. Software Testing includes selecting test data that have more probability of giving errors.” The first step in System testing is to develop the plan that all aspect of system .Complements, Correctness, Reliability and Maintainability.

Software is to be tested for the best quality assurance, an assurance that system meets the specification and requirement for its intended use and performance.

System Testing is the most useful practical process of executing the program with the implicit intention of finding errors that makes the program fail.

# Types of Testing

## Black Box (Functional) Testing:

Testing against specification of system or components. Study it by examining its inputs and related outputs. Key is to devise inputs that have a higher likelihood of causing outputs that reveal the presence of defects. Use experience and knowledge of domain to identify such test cases. Failing this a systematic approach may be necessary. Equivalence partitioning is where the input to a program falls into a number of classes, e.g. positive numbers vs. negative numbers. Programs normally behave the same way for each member of a class. Partitions exist for both input and output. Partitions may be discrete or overlap. Invalid data (i.e. outside the normal partitions) is one or more partitions that should be tested.

Internal System design is not considered in this type of testing. Tests are based on requirements and functionality.

This type of test case design method focuses on the functional requirements of the software, ignoring the control structure of the program. Black box testing attempts to find errors in the following categories:

* Incorrect or missing functions.
  + 1. Interface errors.
    2. Errors in data structures or external database access.
    3. Performance errors.
    4. Initialization and termination errors.

## White Box (Structural) Testing:

Testing based on knowledge of structure of component (e.g. by looking at source code). Advantage is that structure of code can be used to find out how many test case need to be performed. Knowledge of the algorithm (examination of the code) can be used to identify the equivalence partitions. Path testing is where the tester aims to exercise every independent execution path through the component. All conditional statements tested for both true and false cases. If a unit has no control statements, there will be up to 2n possible paths through it. This demonstrates that it is much easier to test small program units than large ones. Flow graphs are a pictorial representation of the paths of control through a program (ignoring assignments, procedure calls and I/O statements). Use flow graph to design test cases that execute each path. Static tools may be used to make this easier in programs that have a complex branching structure. Tools support. Dynamic program analyzers instrument a program with additional code. Typically this will count how many times each statement is executed. At end print out report showing which statements have and have not been executed. Problems with flow graph derived testing:

* + 1. Data complexity could not take into account.
    2. We cannot test all paths in combination.
    3. In really only possible at unit and module testing stages because beyond that complexity is too high.

This testing is based on knowledge of the internal logic of an application’s code. Also known as a Glass Box Testing .Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

## Unit Testing:

Unit testing concentrates on each unit of the software as implemented in the code. This is done to check syntax and logical errors in programs. At this stage, the test focuses on each module individually, assuring that it functions properly as a unit. In our case, we used extensive white-box testing at the unit testing stage.

A developer and his team typically do the unit testing do the unit testing is done in parallel with coding; it includes testing each function and procedure.

## Incremental Integration Testing:

Bottom up approach for testing i.e. continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately done by programmers or by testers.

## Integration Testing:

Testing of integration modules to verify combined functionality after integration .Modules are typically code modules, individual applications, client and server and distributed systems.

## Functional Testing:

This type of testing ignores the internal parts and focus on the output is as per requirement or not .Black box type testing geared to functionality requirements of an application.

## System Testing:

Entire system is tested as per the requirements. Black box type test that is based on overall requirement specifications covers all combined parts of a system.

**CHAPTER 6**

**REFERENCES**

[1] Fundamentals of Database systems – book by Elmasri and Navathe.

[2] Scrum Methodology & Agile Scrum Methodologies.

[3] Developer.android.com [4]http://www.appbrain.com/app/expensemanager/com.expensemanager .

[5]<http://expense-manager.com/how-expensesoftware/> .

[6] <https://www.splitwise.com/terms>.

[7]Textbook-Data Mining: Concepts and Techniques (3rd Edition)by J. Han, M. Kamber, and J. Pei -- Morgan Kaufmann Publ. 2012 ISBN: 978-0-12-381479-1

[8]IEEE Transactions on software engineering, vol. 31, No. 3, March 2005 [9]R.Pressman, software engineering A practitioner’s approach. Fifth edition McGrawHill, 2001. [10]<https://www.xpenditure.com/en>?

[11]<https://markushintersteiner.at/>

[12]http://expense-manager.com/how-expensesoftware/.