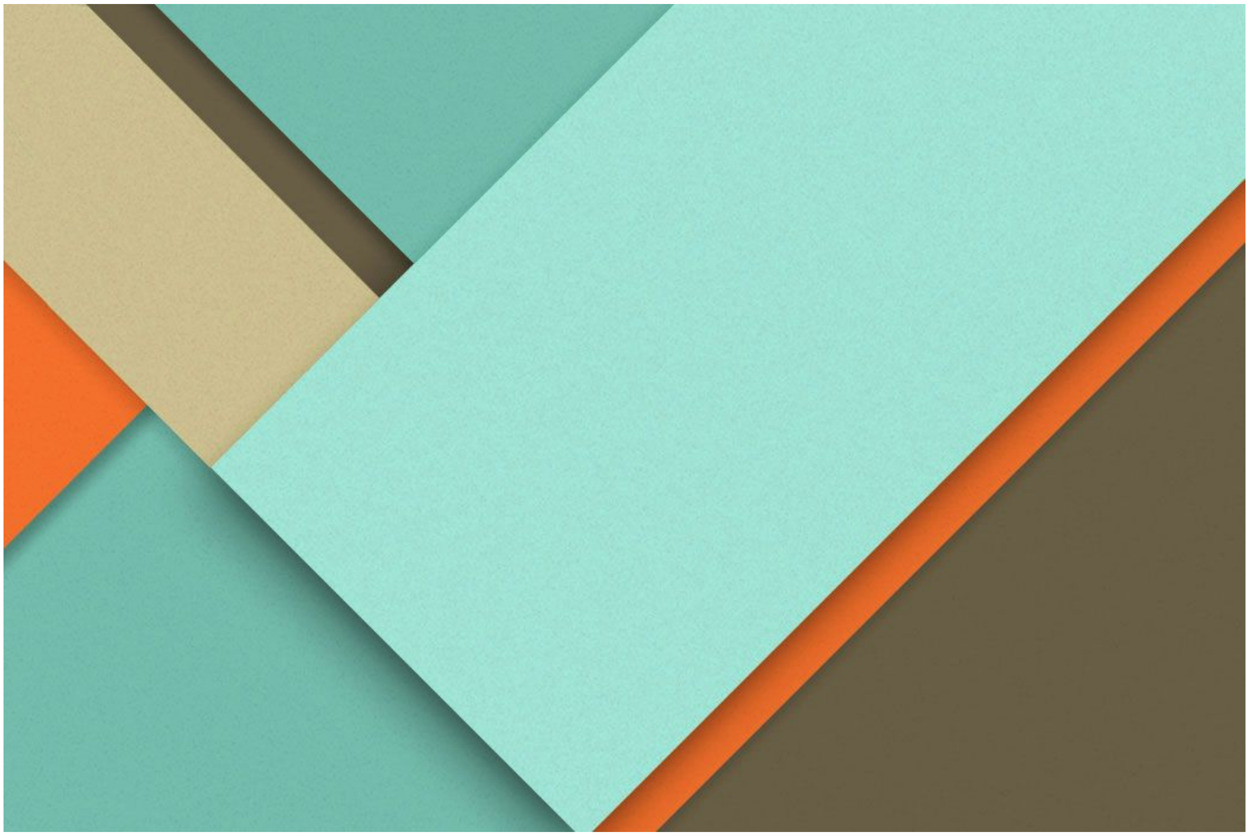


# XReport

Final Report

Rakshith Kunchum

Yashas Devaraju

Group 10
Mobile App Development

# I. Introduction

Expense reporting is an important duty of any organization. In today's word of stringent accounting and auditing, it is mandatory to record expenses borne by employees of an organization. On the other hand, for a person attending business meetings or academic conferences and who incurs expenses that is paid from his own pocket, it is important that he files the expenses in a proper manner so that he can be reimbursed.

Expense report is a document which contains all the expenses that an individual or a team has incurred as part of a business operation. For example: A professor attending a conference along with two of his students is bound to incur a lot of expenses like travelling, staying at hotels, food, entry fee at the conference etc. Usually the professors will be entitled to reimburse the entire expenses incurred for this business operation from their department at the university.

We have developed an Android application, namely **XReport**, which will help the professors or employees to add and manage the expenses that they incur as part of their business. They can later consolidate all these expenses into one single report and submit them to the authorizing person or the approver at the department or office in order to get the amount reimbursed.

We have developed an android interface that enables the authorizing person to go through the reports that have been submitted to him and to either approve them or reject them with added comments.

The following **goals and specifications** were kept in mind when developing the application.

1. Users should be able to add expenses that they incur on their business operations.

2. Users should be able to consolidate all the expenses incurred into one single report and submit it for approval to a higher authorizing officer.

3. The authorizing officer must be able to go through all the submitted reports in order to be able to approve the reports and reimburse the expenses or to request them for additional information on the costs made during that business operation.

4. Users should be able to login with their credentials.

5. When the users are adding an expense, they will be able to enter the amount, the date and place (GPS locations) at which the expense was incurred. They can also attach the image of the receipt (camera) of the expense that they committed.

6. The users will be generating a pdf document aggregating all the expenses that they committed. They will also be able to submit this report to the approving authority.

7. The approving authority will get a notification whenever a report is submitted. They will be able to go through the submitted report. If they are satisfied with the expenses incurred, they can approve the report. If they find any inconsistencies, then they can reject the report and request for explanatory comments from the submitter.

# II. Design Process

First we came up with the domain classes and the database schema required to hold the data. Nouns were identified to provide us with the class or domain object names, adjectives to provide us with the attributes and verbs with the responsibilities. Some of the nouns and verbs identified were:

**Nouns**: Expense, User, Report, Location, Receipt, Expense Item, etc.

**Verbs**: Add Expense, add expense item, get location, generate report, approve report, etc.

Based on the above, domain classes, responsibilities and collaborators were worked upon and the below design decisions were made.

**Database Schema**:

Expense

| ExpenseID | Name | Event Type | Date | Days | Approver | Submit Status | Comments |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

Transaction

| TransactionID (Primary Key) | ExpenseID (Foreign Key) | Name | Category | Amount | Currency |
|---|---|---|---|---|---|
| | | | | | |

| Date | Location.x | Location.y | Receipt Image | Comments | Vendor |
|---|---|---|---|---|---|
| | | | | | |

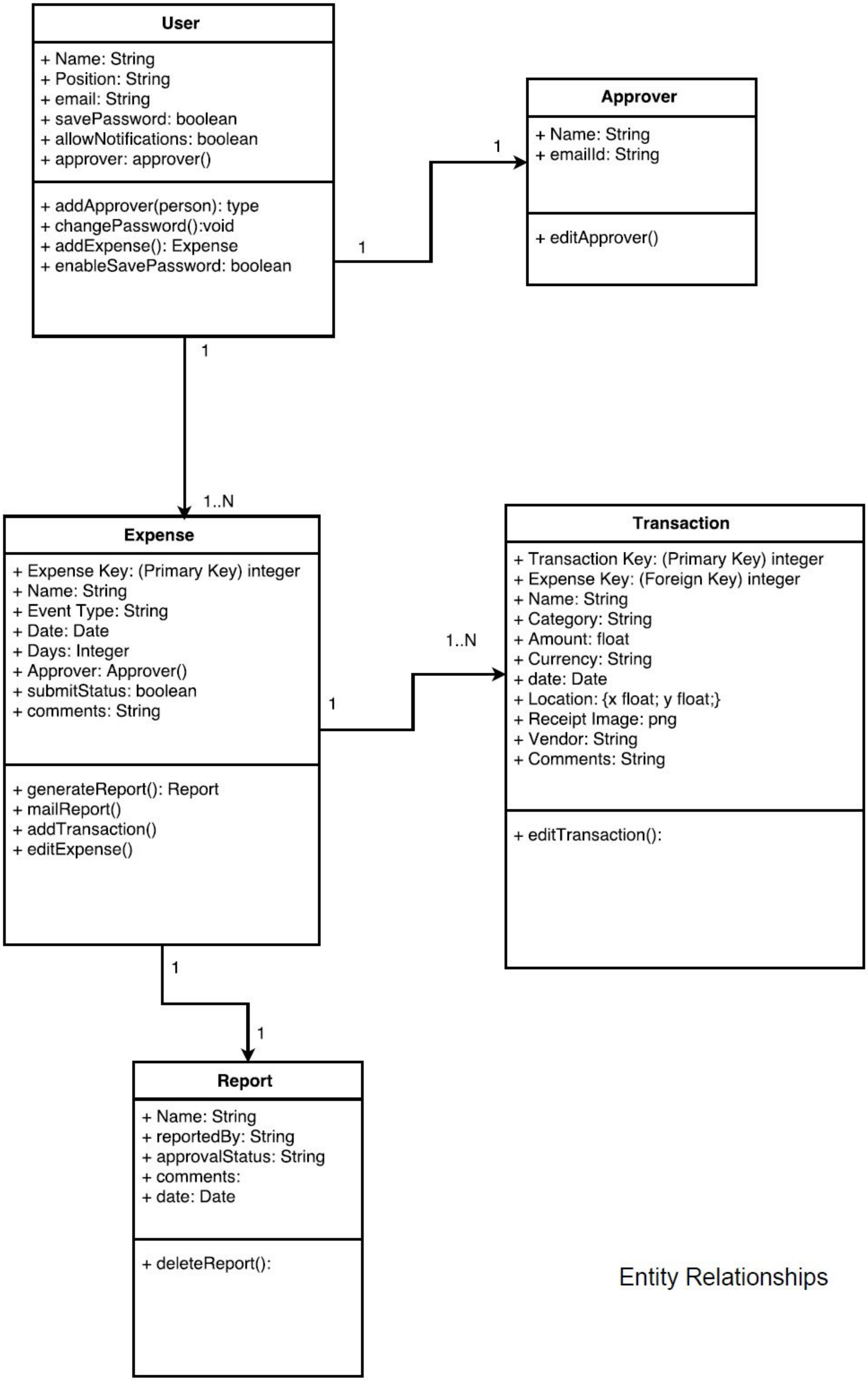
List of Approvers

| PID | Name | emailID |
|---|---|---|
| | | |

List of Reports

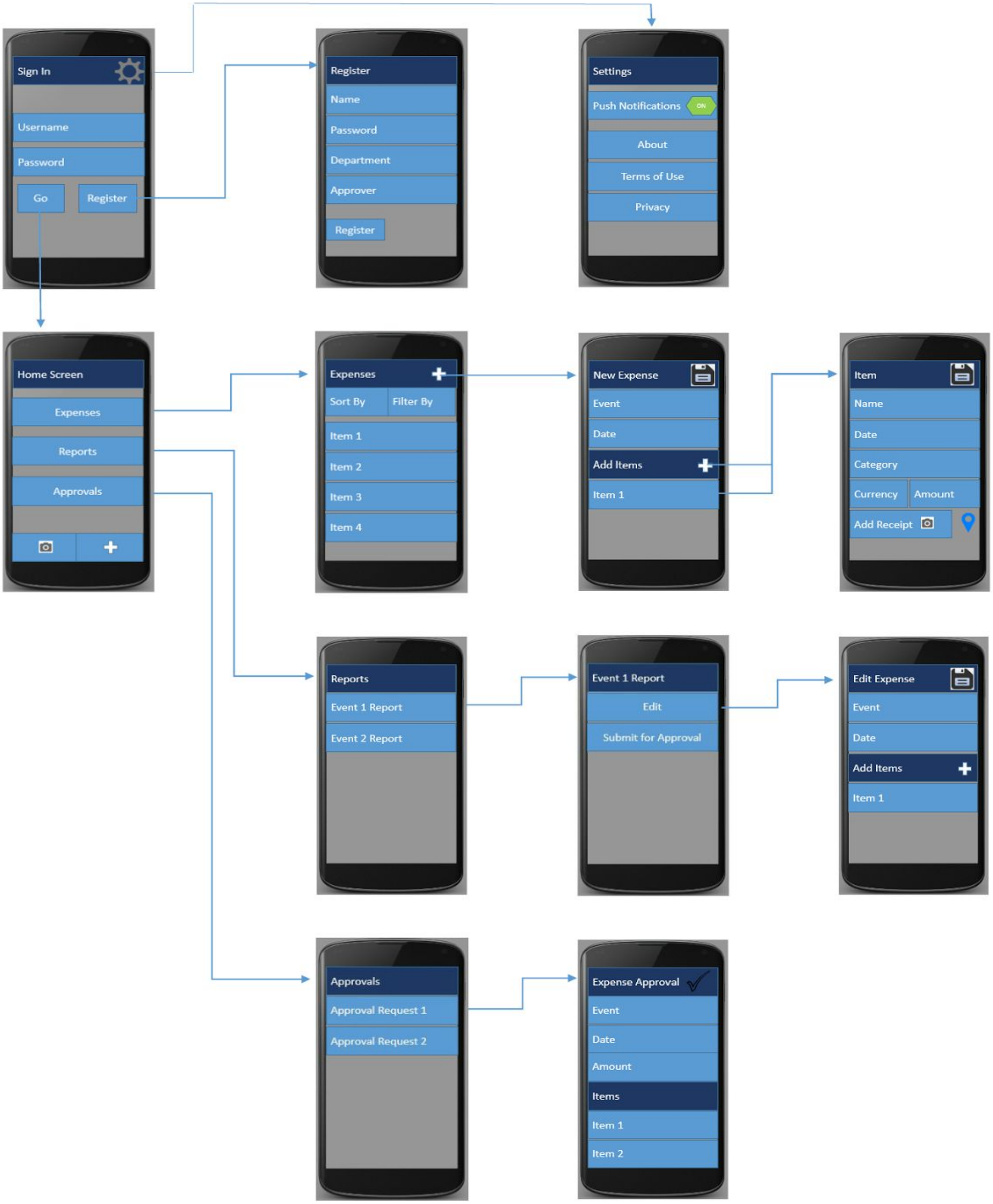
| PID | Report Name | Submitted By | Submitted On | Status | Comments |
|---|---|---|---|---|---|
| | | | | | |

Next, the entity relationships were worked out to arrive at the below implementation.



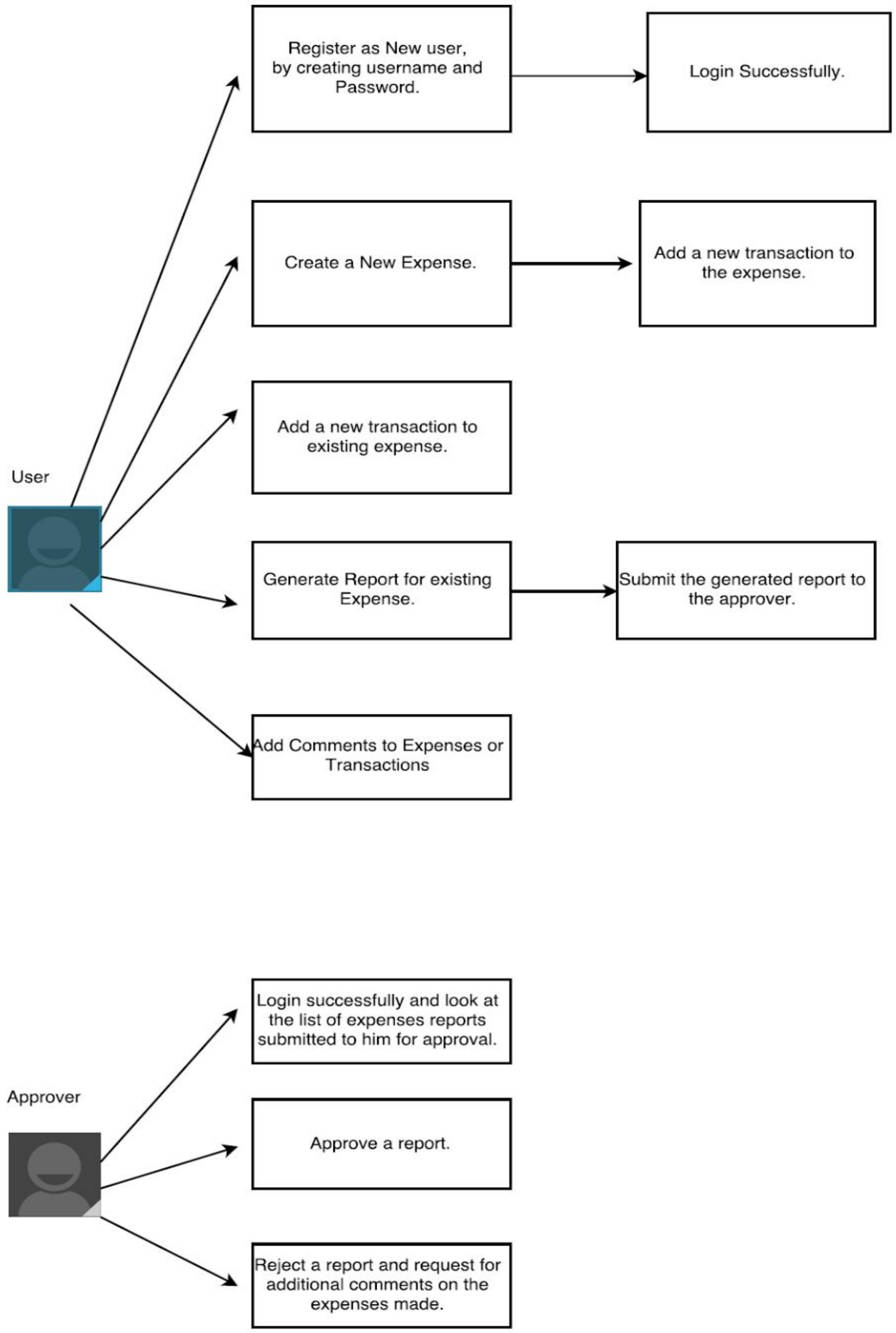
Entity Relationships

A preliminary screen flow diagram was worked out so that we could get started with the xml resources.



**Screen Flow Diagram**

Similarly, a use case diagram was created to understand the flow that the application should take**.**


Register as New user, by creating username and Password.
Login Successfully.
Create a New Expense.
Add a new transaction to the expense.
Add a new transaction to existing expense.
User
Generate Report for existing Expense.
Submit the generated report to the approver.
Add Comments to Expenses or Transactions
Login successfully and look at the list of expenses reports submitted to him for approval.
Approver
Approve a report.
Reject a report and request for additional comments on the expenses made.

Use Case Diagram

# III. Translating Design to Implementation

## MODEL-VIEW-CONTROLLER DESIGN (MVC DESIGN)

### MODEL

#### DATABASE SCHEMA DESIGN

In order to build the database schema for our project we have created a class called **MySQLiteHelper**. It extends the class SQLiteOpenHelper and overrides the functions onCreate and onUpgrade. The class maintains a database-version number to keep track of the schema changes that we make. onCreate method is called to create the schema tables for the first. onUpgrade is called when wish to change the database schema version. Whenever we change the schema version, the underlying data stored is erased.

#### DATASOURCE DESIGN

For each table created in our schema we have a separate datasource class file in our project. This class contains all the methods to establish connection with the database. This class also contains the methods to add, query, update and delete rows on the table. This class also contains user specific methods query rows based on criteria from the table. Example: In case of expense item table we will have to extract rows based on given expense id.

Hence we have dedicated datasource files for each of the tables User, Expense, Expense-Item, Reports. The methods implemented in each class are specific to querying or modifying the corresponding tables.

#### CLASS OBJECT DESIGN

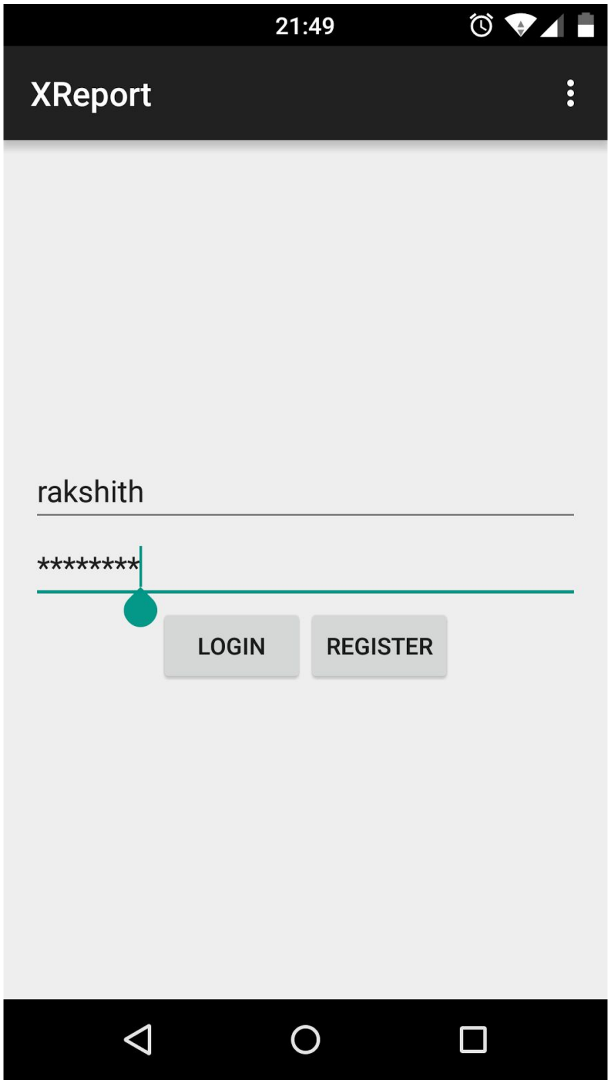
We tried to implement the granular designing concepts of '**Model-View-Controller**' Model (MVC Model). We have created necessary class files in our project in order to represent the objects belonging to different tables. This design will help us to modify single instance at a given time. This will further also help us to persist the data to the underlying database tables.

# VIEW

## ACTIVITY FLOW DESIGN

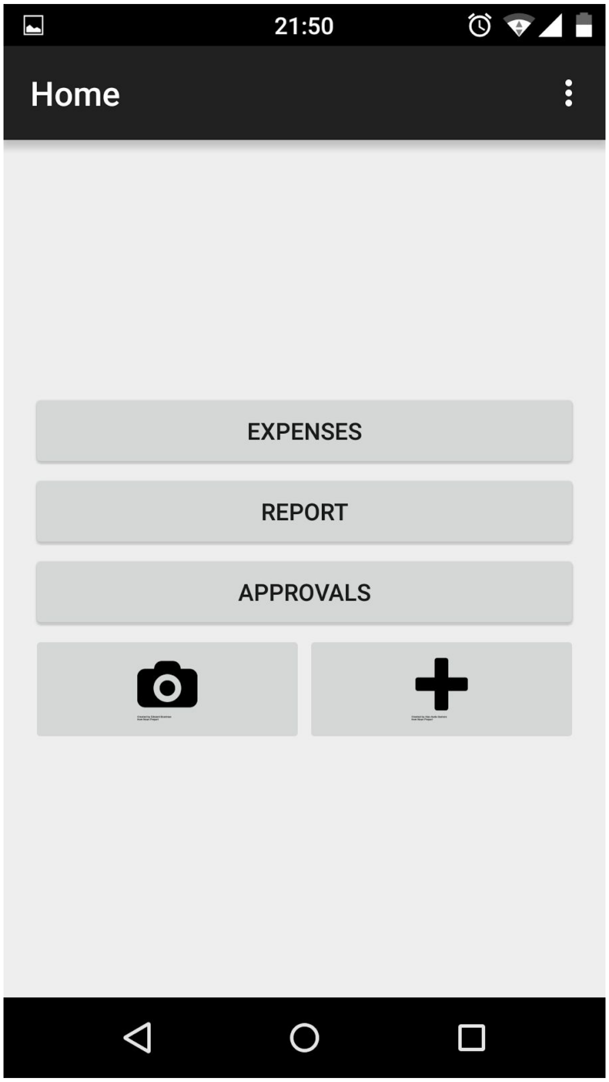
### REGISTRATION AND LOGIN ACTIVITY

The startup activity of our application is the Login Activity. The user will be able to enter his username and password to proceed. In case of new user, there is opportunity for the user to navigate to Register Activity. User is able to enter his credentials and register as a new user. After successful registration the user is redirected to Login page, where one can login to the application to avail the services.



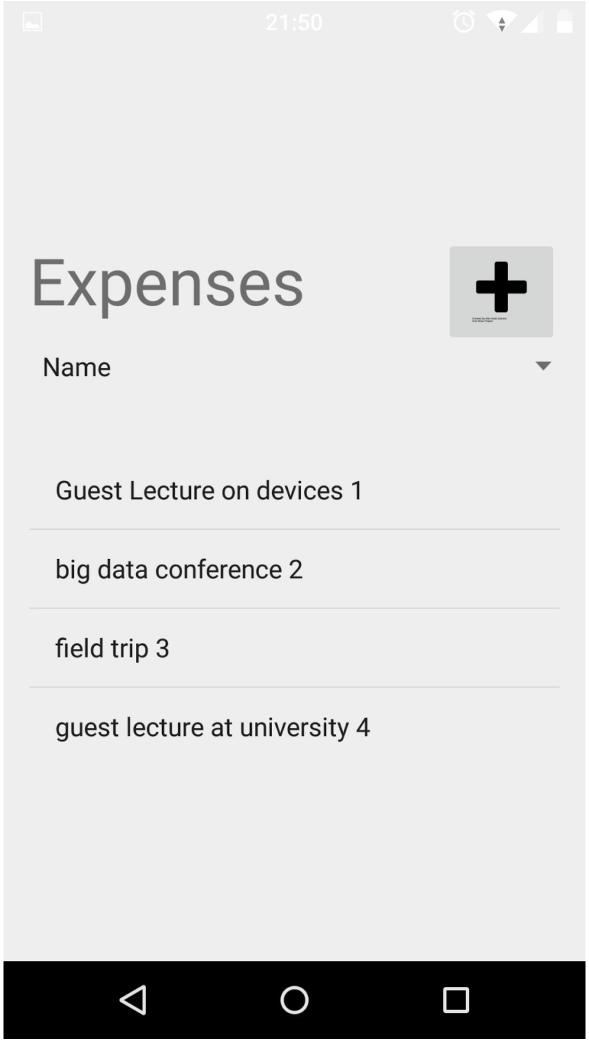
rakshith

********



## HOME ACTIVITY

As the name says it's the home page for a logged in user. From this activity the user will be able to navigate to the expense activity page for the list of expenses created. The user will also be able to navigate to the reports activity for the list of expense reports that have been submitted so far. The user can also navigate to Approvals activity where he can look the list of reports submitted to him for his approval. We have also added to shortcut buttons on home activity page to create expense items quickly. The user can also initiate creation of a new expense item just by clicking on the camera button, which will open the back camera to add a new expense item bill.



Expenses

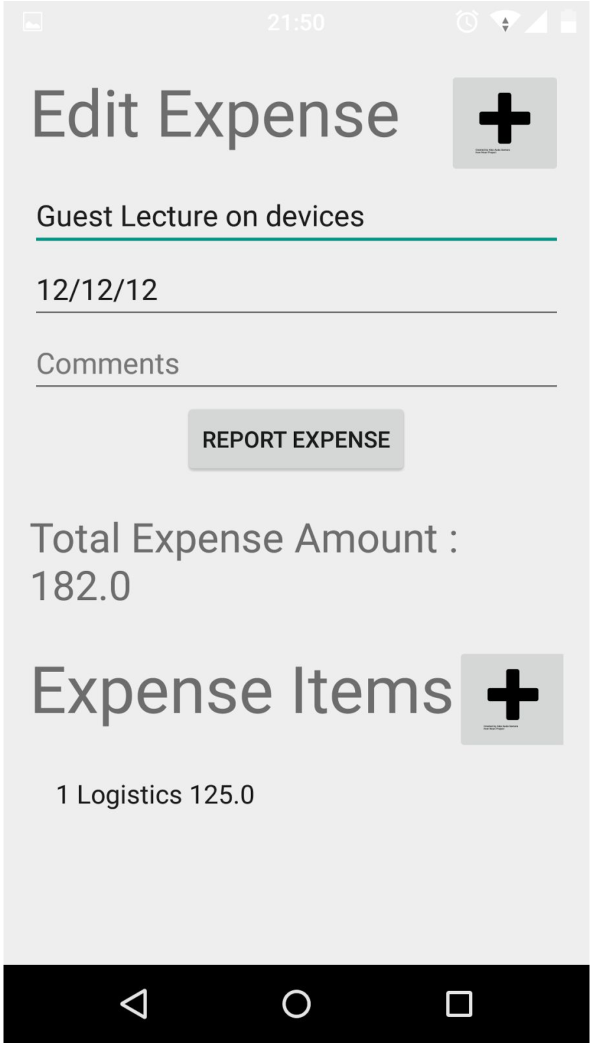
Name

Guest Lecture on devices 1

big data conference 2

field trip 3

guest lecture at university 4



Edit Expense

Guest Lecture on devices

12/12/12

Comments

Total Expense Amount : 182.0

Expense Items

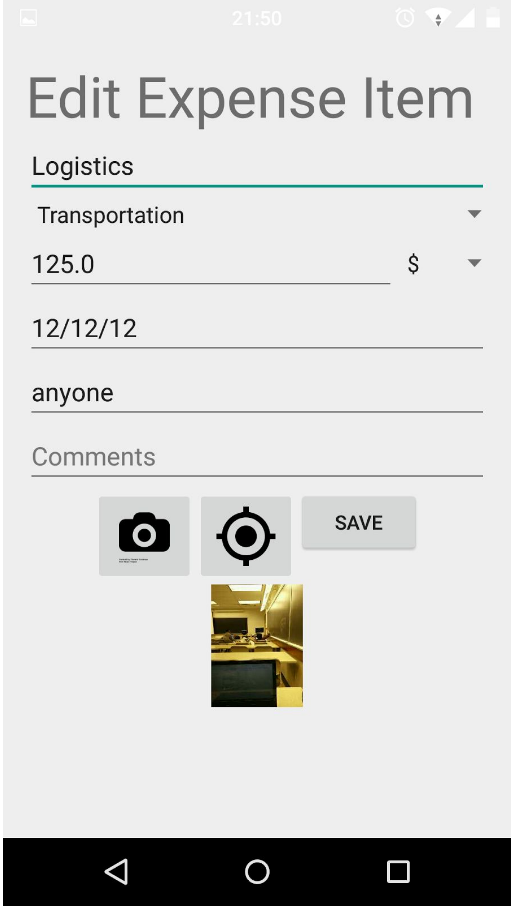
1 Logistics 125.0

## EXPENSES AND EXPENSE-ITEM ACTIVITY

In the Expenses activity, the list of expenses created so far are enlisted. The rows in the list are click enabled. Add button enables the user to add new expense. The list of expenses can be further reordered by name of expense, date of expense or by amount spent during the expense. After clicking on an expense, the name, date and such particulars regarding the selected expense will be displayed. The list of expense items incurred under the selected expense will be shown as a list. Clicking on the expense-item will take us to Edit expense item page where user will be able to edit the details of the expense item.

Camera button will enable the user to take photo of the bill and attach it to the expense item. Upon saving the expense item, the user list of expense items will be updated instantly.

In the expense activity the report button will consolidate all the expense items made under the given expense and create a pdf report and open the mailer to send the report to the concerned approver.



21:50

Edit Expense Item

Logistics

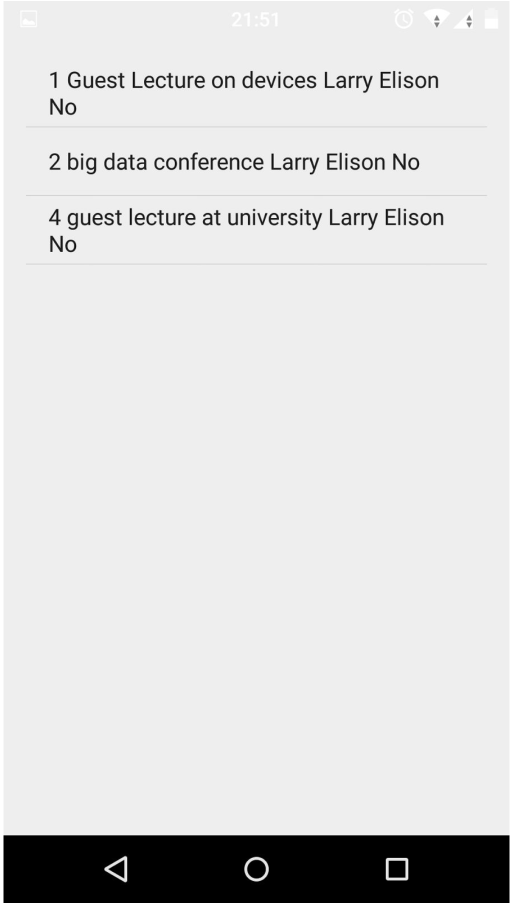
Transportation

125.0 $

12/12/12

anyone

Comments



21:51

1 Guest Lecture on devices Larry Elison No

2 big data conference Larry Elison No

4 guest lecture at university Larry Elison No

### REPORT ACTIVITY

The list of reports submitted so far can be viewed on this page. The status of the reports (either approved or not) can be tracked easily using this activity.

# CONTROLLER

## BUSINESS LOGIC IMPLEMENTATION

### USER LOGIN

During Login the credentials entered will be validated. As soon as the login is successful the user context is set in the application. The user can even set his preferences by navigating to the Settings Activity.

### CREATING NEW EXPENSES AND NEW EXPENSE ITEMS

Creating new expense will create a new expense instance. The details entered by the user will be populated accordingly in the class instance. When user clicks on save, the entered details will be persisted to the database. Similar logic is followed in case of creating and persisting newly created expense items.

### CREATING EXPENSE REPORTS

When the user clicks on the report expense button, a pdf is created in which all the expenses made under the given expense is listed as formatted text. The bill images are also attached in the report. Also a draft mail will open using default email client where the pdf will be attached to the mail.

**What worked :** All of the functionalities that we listed like login, register, creating expenses, expense items, reporting them were implemented.

**What did not work** : While doing expense reporting we were initially unable to create and save documents in pdf format.  However we need a mechanism to save the expenses made as formatted text and save them in a document.

**How we bridged it** : We made use of the pdf supported libraries from itext in order to create and save pdf documents. [3]

## TESTING THE APPLICATION

### JUNIT TESTING

We used JUnits Framework in order to test our activity flow and user input validations.

We created JUnits to validate the data being persisted on the database.

# IV.  Suggested Changes and Improvements

XReport App is designed and developed by following all the design guidelines an Android project needs to follow including Test driving the project, enhancing the performance, making it available to the targeted audience and utilizing the maximum resources and minimizing the power consumption.

However, we see that the following changes or improvements can be made in the app:

1. The application can be enhanced to store all the data on a **central repository** rather than the user's mobile device. This can help in running aggregated reports that can help in overall accounting.
2. The expense reports can be formatted in a better way so as to provide barcode information in order to identify the report quickly.
3. Finally, the app could be released on **Google Play**.

## V. Acknowledgement

Firstly, we would like thank **Prof. Champion** for helping us throughout the semester whenever we needed help.

We would also like to thank our friends and classmates who provided us with their valuable feedback. This helped us in developing a better mobile app.

Lastly, we would like to mention that it was a fantastic experience working on the XReport application. All the team members contributed equally towards each phase of the application, checkpoint and the report in order to successfully complete the project.

# VI. References

[1] Bill Phillips, Chris Stewart, Brian Hardy, and Kristin Marsicano, Android Programming: **The Big Nerd Ranch Guide**, Big Nerd Ranch LLC, 2015.

[2] Rajiv Ramnath, Roger Crawfis, and Paolo Sivilotti, **Android™ 3 SDK Programming for Dummies®**, Wiley, 2011.

[3] http://www.vogella.com/tutorials/JavaPDF/article.html

[4] http://www.vogella.com/tutorials/AndroidCamera/article.html

[5] http://www.vogella.com/tutorials/AndroidLocationAPI/article.html

[6] http://www.vogella.com/tutorials/AndroidListView/article.html

[7] http://www.vogella.com/tutorials/AndroidSQLite/article.html