**University of Regina**

**CS 455 Term Project**

**Changxuan Zhao**

Mobile App for Android Device

**Tiny Finance**

****

[**Overview**](#_bvrzvga8mlaq) **2**

[**Configuration**](#_84tb86dy2u0r) **2**

[**Installation**](#_or23j7f0ygfs) **2**

[**Usage**](#_ar1yun29tfap) **4**

[**MVC Design Pattern**](#_ivm22senfmkf) **7**

[**Manifest**](#_9z3pd3uiu84y) **8**

[**Contact**](#_cna6dstt1thg) **10**

[**Bugs**](#_1gulihfu7n44) **10**

[**Troubleshooting**](#_jyty4q6v0b0e) **11**

[**Further Reading**](#_gc93vewbocya) **11**

[**Copyright**](#_i66dgq9xq81t) **11**

[**Acknowledgment**](#_ckjtr9asdtyj) **11**

### 

### 

### 

### 

### 

### **Overview**

Tiny Finance is an Android-based mobile app designed to deliver a quick and easy solution to record daily expenses. The app provides a bar chart and view list for users to review their expenses in the past. To better protect users’ privacy, a room database is implemented such that no personal information will be uploaded to the Internet, and users may delete the data at any time.

### **Configuration**

Prerequisites: You need to download the project to a PC which has Android Studio installed. Please refer to the [*Installation*](#_or23j7f0ygfs) for more details.

* Android Device:

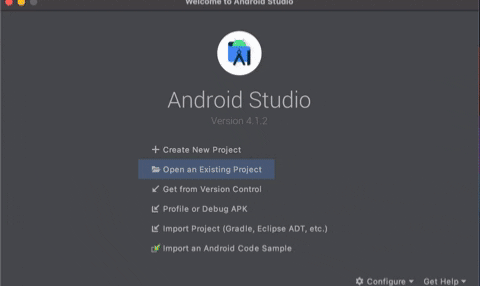
The mobile app is designed to work for all Android devices with Android 5.0 or higher

* Virtual Device:

Download any virtual devices with minimum API level 28 or above on Linux/MacOS/Windows machine

### **Installation**

To install the app, you need to download [Android Studio](https://developer.android.com/studio) on PC. Here is a [tutorial](https://developer.android.com/studio/install) of how to install Android Studio on PC with different operating systems. Once you have installed Android Studio, clone the project from Github. It is recommended to download the repository as a .zip file. Then, open Android Studio and select “Open an Existing Project”. Navigate to your local directory where the project is saved. Finally, click the “Run” button at the top bar.



* Run on emulator

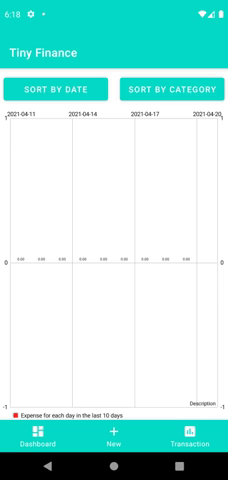
Click “AVD Manager” at the top right corner. Then, click on “+ Create Virtual Device” to choose a virtual device. After selecting a desired device, click “Next” to choose the API level. The minimum API level accepted is API 28. Now, click “Next” and “Finish” to set up the emulator. Compile and run the project, you will see the emulator popped out automatically.

* Run on Android device

First, connect the device to your PC via a USB cable. Next step is to enable “USB debugging” in the “Developer options” window. Open “Settings” on your phone and select “System” if the device uses Android v8.0 or higher. Otherwise, select “About phone” and tap “Build number” seven times. Now, return to the previous screen, choose “Developer options”. In this window, enable “USB debugging”. Make sure you have selected the device from Android Studio before you run Tiny Finance.

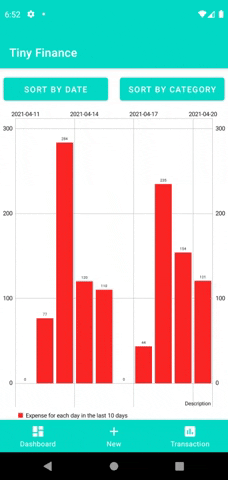
### **Usage**

When the app is first initialized, you will find the dashboard and view list are empty. Don’t worry. It is because the app does not create a local database. Tap “New” at the bottom navigation bar and record the first transaction. Remember to choose a category from the drop-down list.



* Dashboard

The dashboard is the default window to show as Tiny Finance is initialized. The window displays the expense over the past ten days. Click the buttons at the top of the screen to choose sorting records by date or category. By default, it is sorted by date.



* New

Follow the hint to enter all required fields. Make sure to select a proper category from the drop-down menu as the bar chart may be sorted by category. There is a validator to perform basic input validation, and it will pop up an error message at the bottom of this screen to remind you if unexpected input appeared. Click the save button after you have entered values for the attributes. Once a record is saved, the system will jump to the transaction window automatically.



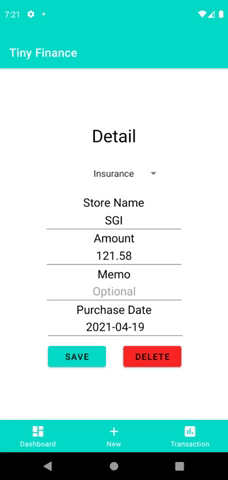
* Transaction

In the transaction window, it loads all records from the database and displays in a list of views sorted by date. The icon represents different categories. Other attributes except memo are shown around the icon to give users a general view. To check the details of a specific transaction, tap the header and the system will navigate to a new screen with details of the selected record.



* Detail

The detail screen is hidden from users and it can only be accessed from the transaction screen by selecting a specific record. The layout is similar with the window for creating a new record that you have seen before. The detail screen is designed to provide a method for users to modify or delete their existing records. Once a change has been made, there will be a reminder displayed at the bottom of the screen. To go back to the previous page, click the back button on your phone or tap the navigation bar.



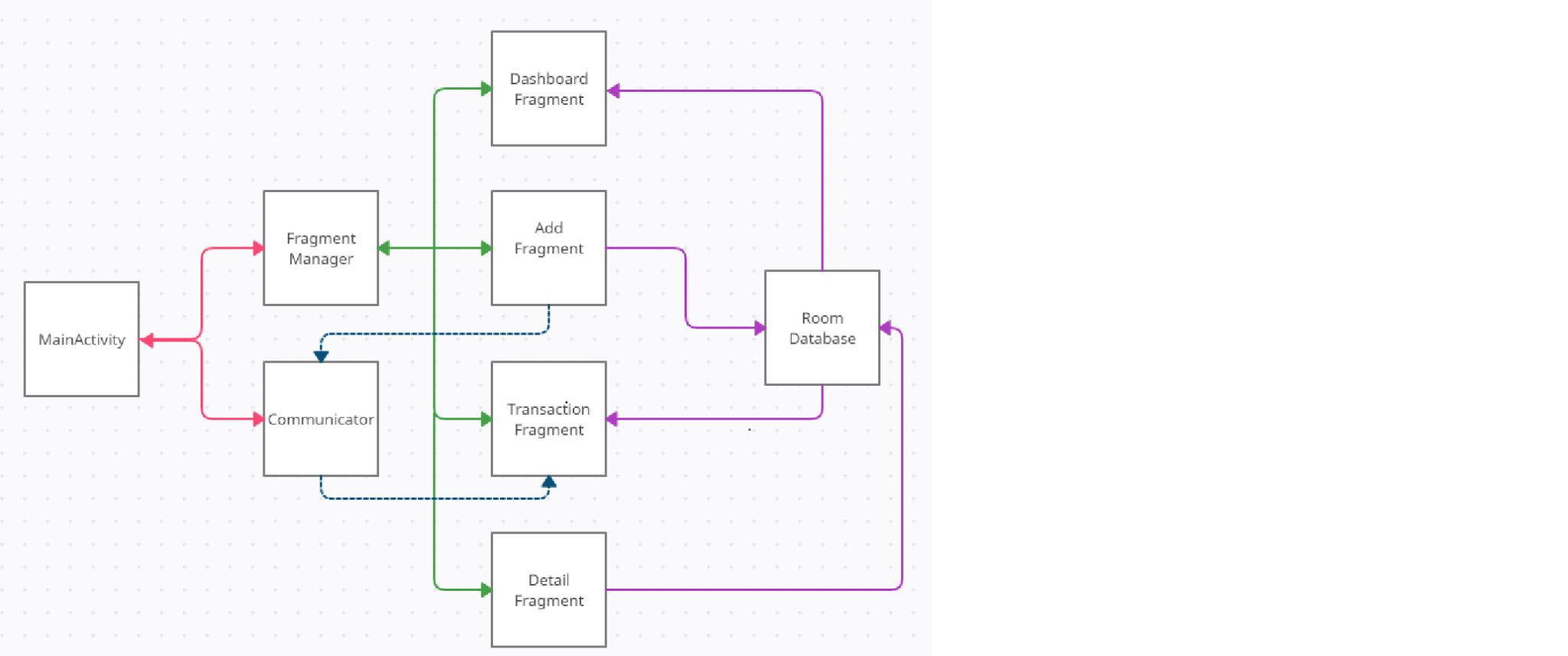
### **MVC Design Pattern**

Tiny Finance is implemented based on Model View Controller architecture. The MVC framework can be defined as:

“an architectural pattern that separates an application into three main logical components: Model, View, and Controller. ... Each architecture component is built to handle specific development aspects of an application. MVC separates the business logic and presentation layer from each other.”

As MVC enforces a separation of the business and presentation logic, it aids to enforce strict organization and separation of logic between components. For example, the code in view of the mobile app does not modify any data of the model directly. In order for the view to modify a model, it has to communicate with the controller first. This idea helps to reduce errors and improve readability by consolidating all computational logic in one place. Additionally, these aspects help to improve future maintainability. Having the code separated into models, views, and controllers improves the system’s ability to be grown and maintained.

Models are defined in Tiny Finance as the main activity used to store objects and data such as callbacks and messages. Another functionality of the main activity is to invoke the fragment manager that acts as the primary Controller. The fragment manager controls the fragments to display different screens. For users’ convenience, a bottom navigation bar that provides an interface for users to interact with the fragment manager is presented. The Views of the app are defined by the widgets embedded in each fragment. The DashboardFragment contains a bar chart that retrieves data from the room database and organizes them by date or category. The AddFragment allows users to create a new record. Once all input data has been validated, the fragment calls the Repository to save data into the database, and it passes a message to the main activity via Communicator which acts as the secondary Controller to load the TransactionFragment. After saving data successfully, the TransactionFragment added to the callback will replace the AddFragment and occupy the fragment container. The usage of TransactionFragment is to retrieve all data from the room database, then list records by following the chronological order in which the entries are created. From here, users may tap the header of a specific record to load the DetailFragment for reviewing, modification, or deleting the selected entry.



### **Manifest**

1. MainActivity.kt
2. RecordIntentApplication
3. Fragment
   1. DashboardFragment.kt
   2. AddFragment.kt
   3. TransactionFragment.kt
   4. DetailFragment.kt
   5. Communicator
   6. RecordDetailViewModel
   7. RecordListViewModel
4. Database
   1. Record
   2. RecordDao
   3. RecordDatabase
   4. RecordRepository
5. Drawable
   1. ic\_add.xml
   2. ic\_dashboard.xml
   3. ic\_transaction.xml
6. Layout
7. activity\_main.xml
8. fragment\_add.xml
9. fragment\_dashboard.xml
10. fragment\_detail.xml
11. fragment\_transaction.xml
12. list\_record.xml
13. Menu
14. menu.xml
15. Mipmap
16. education.png
17. entertainment.png
18. food.png
19. housing.png
20. insurance.png
21. others.png
22. personal.png
23. supplies.png
24. transportation.png
25. utilities.png
26. Values
27. colors.xml
28. strings.xml

### **Contact**

Changxuan Zhao - [zhao205@uregina.ca](mailto:zhao205@uregina.ca)

Project Link: <https://github.com/Maverickbear77/CS455Project-Tiny-Finance>

### **Bugs**

* Bar chart

The height of the first column is much narrower compared to other columns as the height of bars increase.

* Passing message

The message passed via Communicator from “New” window to “Transaction” window cannot be displayed through Toast command.

### **Troubleshooting**

If any errors occurred, uninstall Tiny Finance or delete the emulator depending on which way you are using to run the app. However, be sure to save the data before you do. Please refer to [*Further reading*](#_gc93vewbocya) to learn more about saving the room database to your PC.

### **Further Reading**

* [Kotlin](https://play.kotlinlang.org/#eyJ2ZXJzaW9uIjoiMS40LjMwIiwicGxhdGZvcm0iOiJqYXZhIiwiYXJncyI6IiIsImpzQ29kZSI6IiIsIm5vbmVNYXJrZXJzIjp0cnVlLCJ0aGVtZSI6ImlkZWEiLCJjb2RlIjoiLyoqXG4gKiBZb3UgY2FuIGVkaXQsIHJ1biwgYW5kIHNoYXJlIHRoaXMgY29kZS4gXG4gKiBwbGF5LmtvdGxpbmxhbmcub3JnIFxuICovXG5cbmZ1biBtYWluKCkge1xuICAgIHByaW50bG4oXCJIZWxsbywgd29ybGQhISFcIilcbn0ifQ==)
* [Android programming in Kotlin](https://developer.android.com/kotlin)
* [Room database](https://stackoverflow.com/questions/53958313/on-an-android-device-where-is-located-sqlite-database-created-by-room-library)

### **Copyright**

This project is not specifically distributed under a license. All rights are reserved by Changxuan Zhao. Please contact the author for any modifications or other usage.

### **Acknowledgment**

* [MVC architecture](https://www.guru99.com/mvc-tutorial.html)
* [Run your app](https://developer.android.com/training/basics/firstapp/running-app)
* [Bar chart](https://github.com/PhilJay/MPAndroidChart)
* [Bottom navigation bar](https://www.geeksforgeeks.org/how-to-add-a-floating-action-button-to-bottom-navigation-bar-in-android/)
* [Flaticon](https://www.flaticon.com/free-icon/utilities_1519162)
* [Spinner](https://www.youtube.com/watch?v=KVs2eKiM_uA)
* [Communicator](https://morioh.com/p/439e168920e9)