

Expenses Management- Android Application

By : Smit K. Patel

e-mail: smit.patel001@umb.edu

UMB ID : 01775717

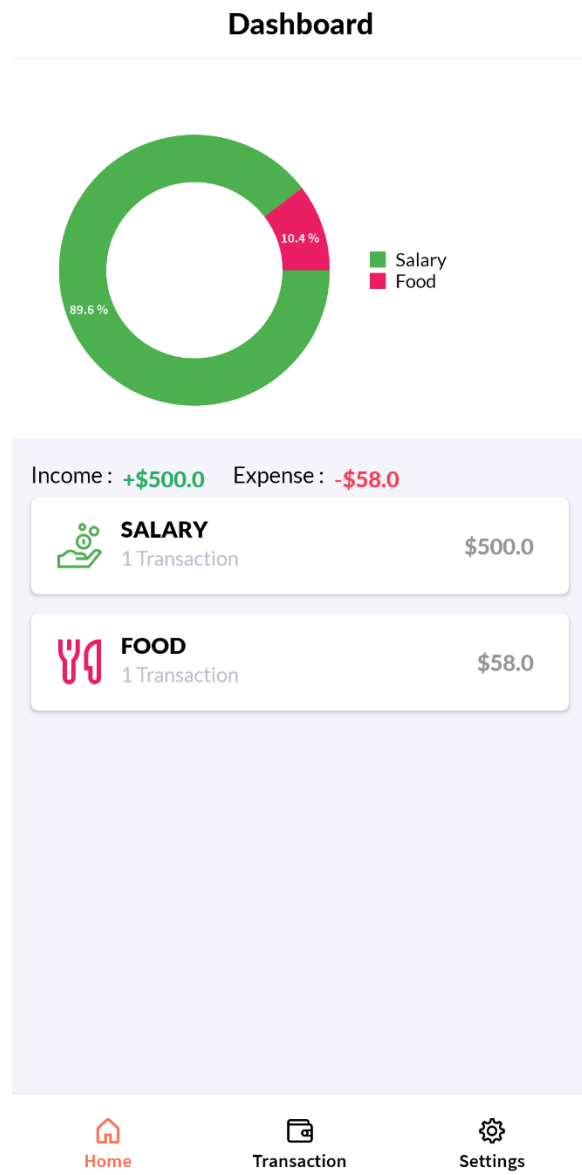
1.PROJECT STATEMENT

Daybook is an app for keeping track of personal spending. By tapping the add expenditure option in the app, users may enter the specifics of their income and expenses and classify them properly. Users will be able to see a live graphical depiction of expenses and income after adding the transaction to local database, which may be utilized for overall wealth management. Users will be able to keep track of their spending and income in a simple and organized manner with the aid of this application. Users may also plan investments and savings for momentous events such as weddings and house purchases. Users must provide storage permissions to save data locally in the android room(adv. SQLite) database.

2. APPLICATION DESIGN

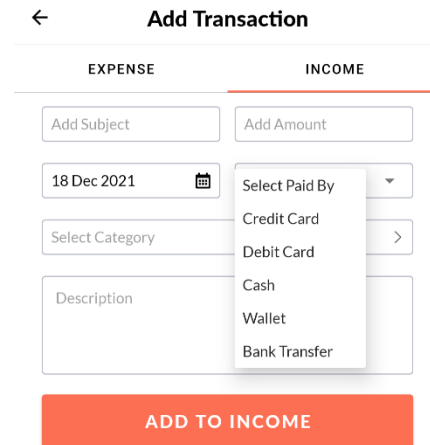
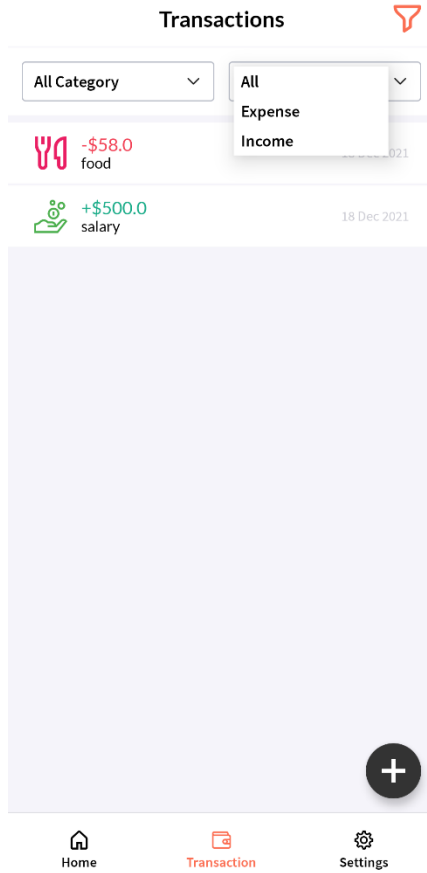
The app is planned to be divided in three sections as shown below :

a. Dashboard



This is the application's primary landing page. This displays a live graphical depiction of the user's earnings and expenses. Below the graphical depiction, the user may view a list of transactions. Each transaction can be clicked on to see the description of the transaction, if available.

b. Transaction Tab

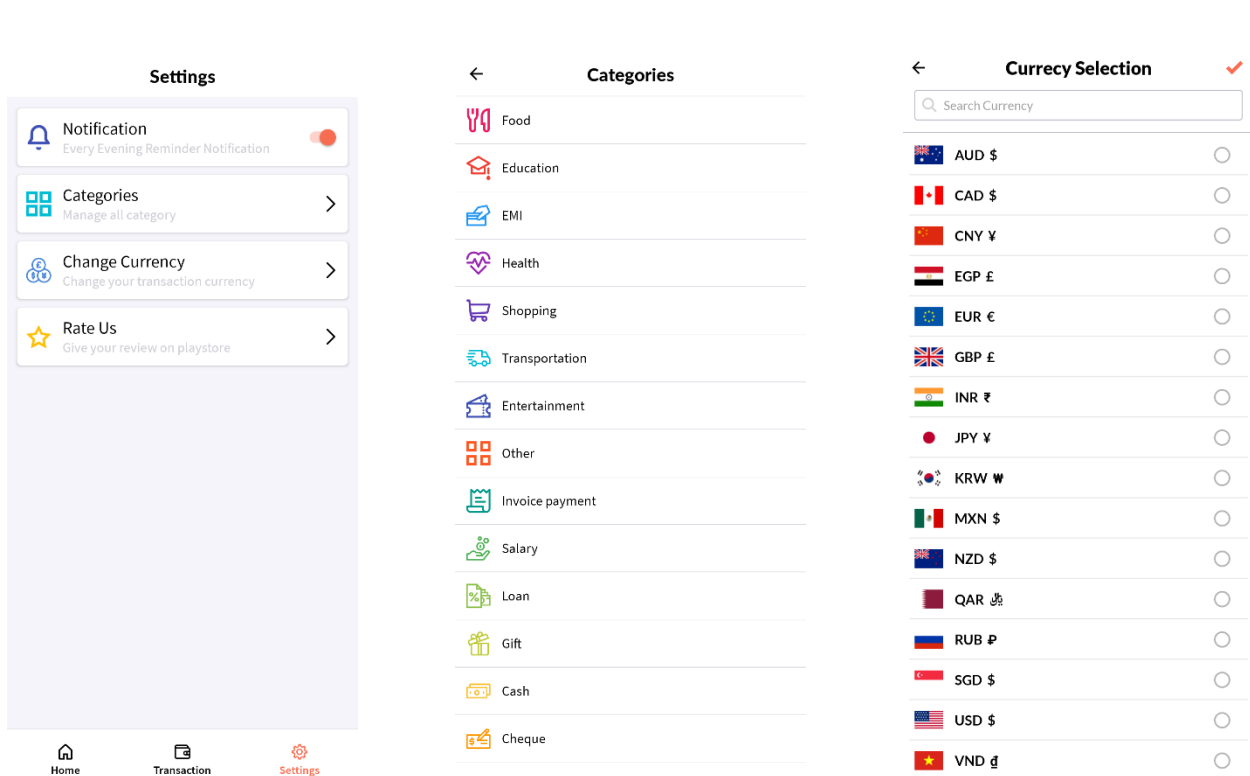


By arriving on this tab, the user may see all the transactions. These transactions may be sorted by transaction type and category.

- Add Transaction Section

User can click the floating add button in the Transaction tab to open add income/expense panel. To be able to click on the add transaction button, the user must first pick the tab of income or expense and then supply the necessary information, such as transaction data, transaction subject, and transaction category as shown in the figure above. The entered data is validated and inserted into the database. The user is then brought back to the transaction page, where the updated transaction list is displayed.

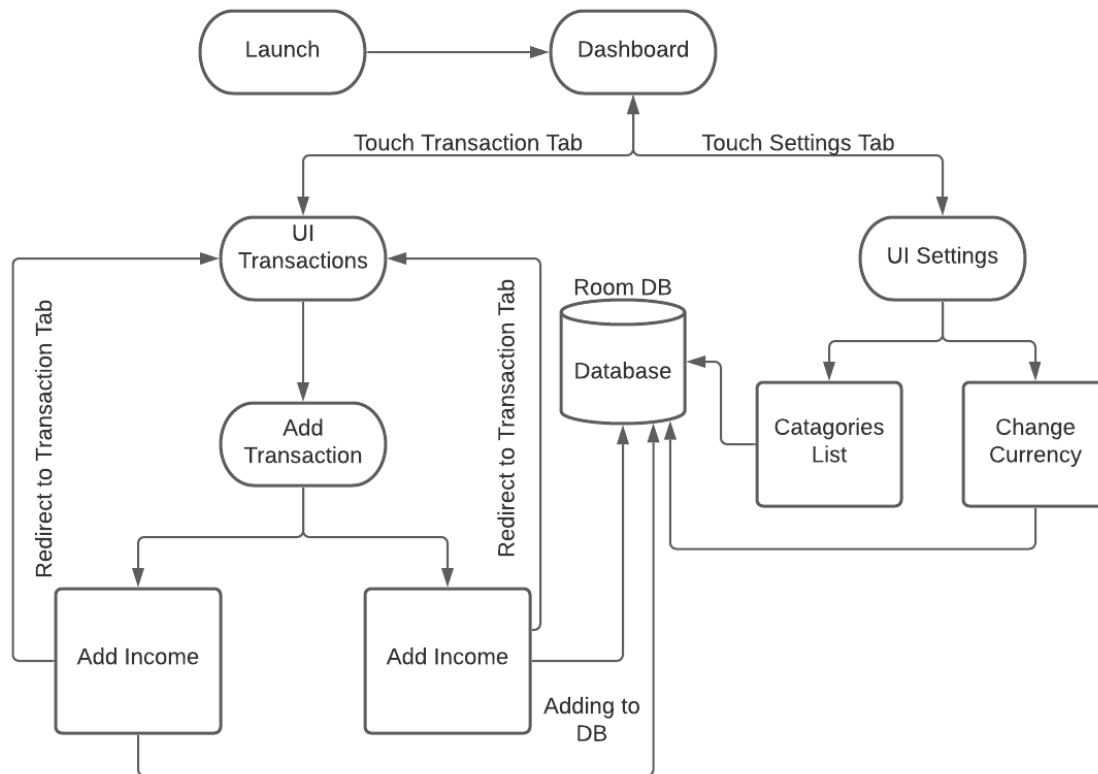
c. Settings Tab



The user can personalize the app's experience here.

- They can choose whether or not to get alerts.
- Change the currency by selecting it from the drop-down menu.
- User can check the categories in which the transaction can be added. User will be able to add new category(Future Scope).
- Rate Us button (Future Scope).

The flow of the application is as follow:



In this application, I used many modules, including activity, adapter, base, database, and fragment. It can be seen in the activity section; each module has its own set of rules. Additionally, this application makes advantage of the Android room database to save, update, remove, and retrieve the user's transactions in local storage. In Android, an adapter serves as a link between UI components and data-driven components that assists the UI component's data. Later data stored in the database is sent to the adapter, which then takes the data and displays it in the frontmost application. When it comes to fragments, I used the following six: TabDashboardFragment for dashboard views, TabTransactionFragment for transaction views, CatagoryListFragment for category views, TabProfileFragment for the setting view, AddNewTransactionFragment for adding transaction views, and HomeFragment for UI control. Using the custom view module, I attempted to make the application operate with swipe gestures to switch tabs. The util module is used to log the items that are important for troubleshooting. Also, there is a listener module that monitors items clicked and the child listener module, allowing users to switch between apps quickly. Finally, In Activity HomeActivityFragment is used to organize all the components into a well-organized application. Another point to mention is the database; Android Room is used to handle all the data-oriented tasks (android's advanced version of SQLite). All of these modules are handled in such a way that everything is accessible to each function, preventing future bugs.

3. APPLICATION IMPLEMENTATION AND EVALUATION

The application is subdivided in specific modules to separate the functionality of the classes:

Activity Module:

This section lists many types of activities that can be done in the app. i.e., Currency selection, Home Activity—to guide the user to the appropriate page.

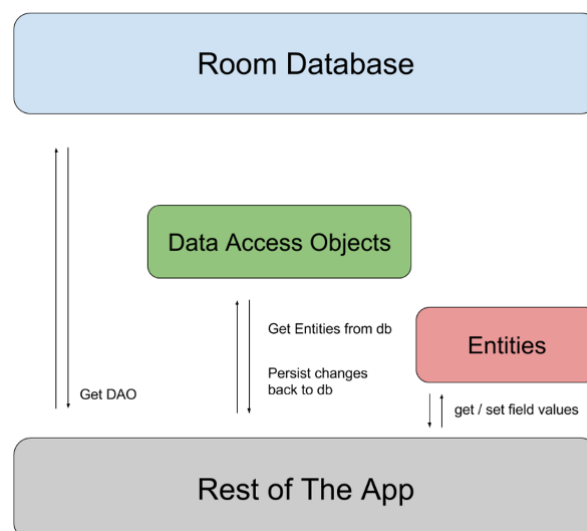
Adaptor Module :

The classes in this section serve as a link between UI and data-driven components.

- CatagoryListAdapter - Retrieves and displays a list of available categories.
- ChartCategoryAdaptor – The chart is displayed on the dashboard tab.
- CurrencySelectionListAdapter – Fetches a list of currencies and interacts with CurrencySelectionAdaptor to assist the user in selecting the appropriate currency.
- DashboardItemListAdaptor - Presents a transaction list on the dashboard.
- SettingsListItemsAdaptor – Displays a list of customization options accessible to the user.
- TabAdaptor - Assists users in switching between tabs and sharing information inside.
- TimeSpinnerAdaptor - Aids in the creation of a transaction UI for selecting an expense or income transaction.
- TransactionListAdaptor- Exhibits and filters a list of database transactions on the transaction tab.

Db Module:

This folder contains the queries and functions needed to perform SQL transactions on the local room persistence database.



Fragments Module:

This module groups all the fragments used in the application.

- TabDashboardFragment- for generating a dashboard display.
- TabTransactionFragment - Transaction view used to generate transaction listings on the transaction tab.
- TabProfileFragment – for presenting the settings options list view.
- CatagoryListFragment – for displaying the category list.
- AddNewTransactionFragment – for creating a new transaction view (income/expense).
- HomeFragment- Manages the UI of the fragments.

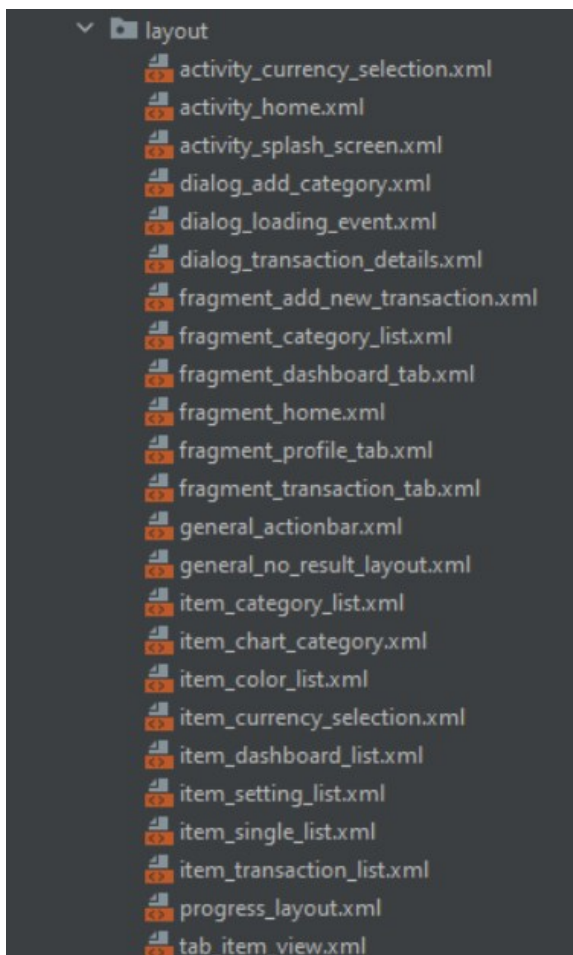
Listener Module:

This module contains the listener classes, where the clicked item is tracked, and which is the child listener to move between the app efficiently.

Utile Module :

I have used a module util for logging the things useful for debugging.

UI :



- The drawable folder includes all the application's images, icons, and xml path icons.
- The font folder comprises the typefaces used by the app's many components, streamlining the app's appearance.
- Finally, the layout folder contains all the User Interface designs for each component, where the primary appearance of the app is produced with careful attention to the margins and paddings for each component to operate correctly on any Android smartphone.
- Each of the fragments has its own layout. Fragment_profile_tab.xml, fragment_home.xml, and so forth.
- The attributes, colors, and strings list are all present in the subdirectory values.
- The raw JSON objects of flag, symbol, and short name for a certain nation are stored in the currency symbols list.

There is a minor issue in displaying the graphical representation of the transactions while testing the current version of the program. After adding a transaction to the database, the graph does not update in the dashboard tab. However, if the app is closed and reopened after adding the transaction, the dashboard graph shows the updated representation.

I am debugging the issue. In addition, I'm working on a new version to the app that will allow users to create their own categories via the settings tab.

4. REFERENCES

- [1] Android jetpack, Android room Database <https://developer.android.com/training/data-storage/room#java>
- [2] Mark O'Sullivan, SwipeRevealLayout java Class for swipe through tabs.
- [3] Professor Bo Sheng : University of Massachusetts- Boston, CS443 Lecture materials and homework assignments

5. EXPERIENCES AND THOUGHTS

I've always wanted to keep track of my expenditures and income, as well as prepare for future investments. For this project, I believed that creating an app that could be used to perform the same activity without the usage of an excel sheet would be beneficial as well as allow me to understand the app development process in action. I am looking forward to putting the app on the Android Play Store after it has been thoroughly debugged and some more features have been added. The most difficult part was the database integration, as I had never used android room Database. Throughout the course of this project, I have learnt several additional features, such as SwipeRevealLayout and fragments data sharing, in addition to room database. Overall, this project gave me the confidence to learn new things and continue to build the newly learned talents.