Expense Manager Mobile App Report

Name: Hitarth Anand Rohra
Roll No: AM.EN.U4EAC21032

Introduction

The Expense Manager App is designed to aid individuals and businesses in effectively managing their finances. It offers a user-friendly interface and a variety of functionalities that cater to the tracking, categorization, and analysis of income and expenses.

Features

- **Income/Expense Tracking:** Users can easily record and categorize their financial transactions, which helps in maintaining an organized financial record.
- **Reports and Analytics:** The app provides daily and monthly reports, allowing users to track and analyze their financial activities over time.
- **Category Management:** Transactions can be categorized under multiple headings, allowing for detailed financial management and review.
- **User Interface:** Utilizes Bottom Sheet Fragments for a modern and intuitive navigation experience, enhancing user interaction through Custom Alert Dialogs.

Technologies Used

- **Frontend:** Developed using XML for layout design.
- **Backend:** Java is used for the backend logic, integrating with MongoDB for data management.
- **Database:** Realm database is employed for local data storage, facilitating quick data retrieval and storage without needing an internet connection.
- **Architecture:** Follows the Model-View-ViewModel (MVVM) architecture, promoting a clean and maintainable codebase.

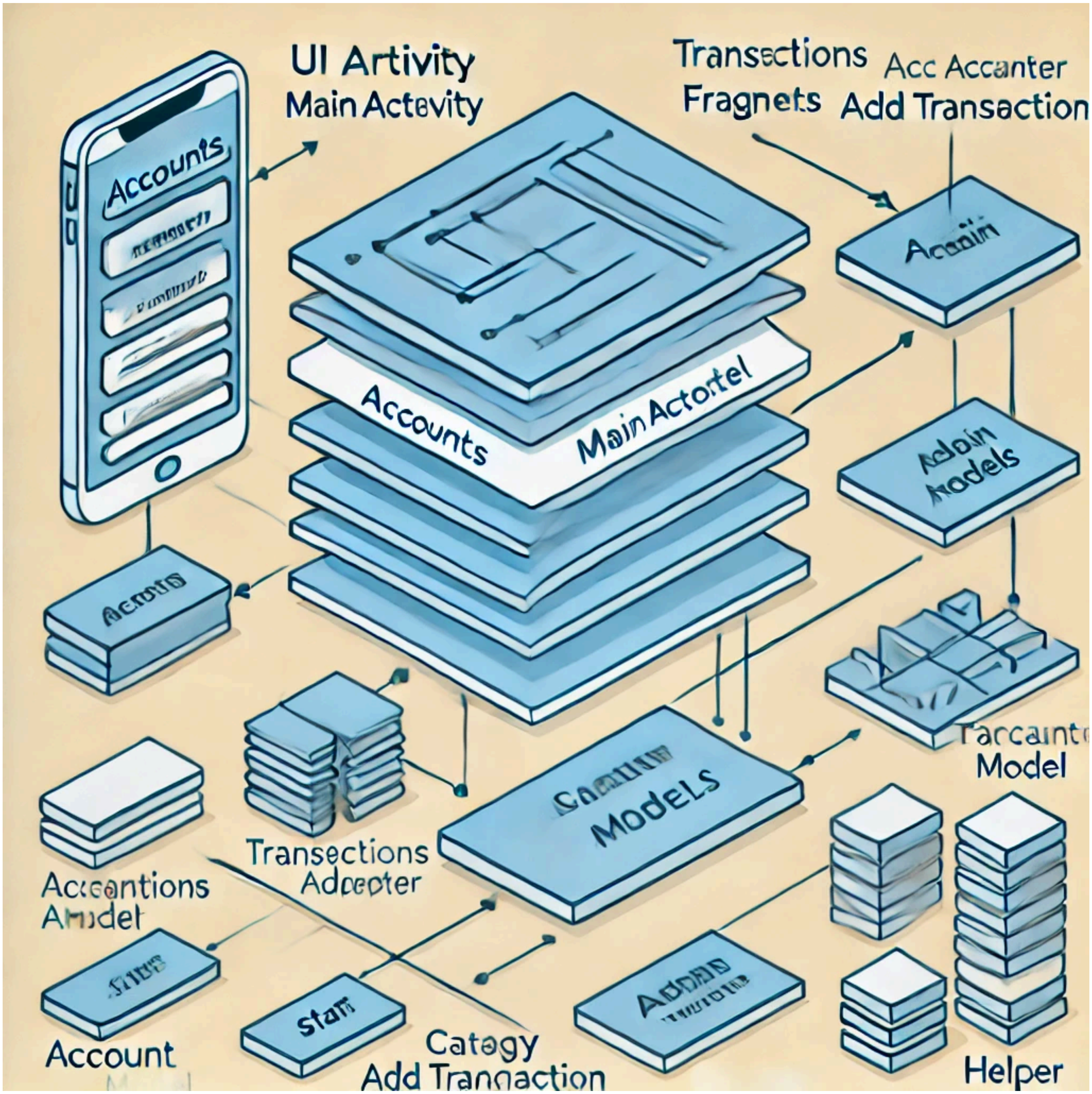
Installation and Usage

The installation process involves cloning the repository and setting up the project in Android Studio. Once the project is set up, it can be run on an Android device or emulator to test its functionalities.

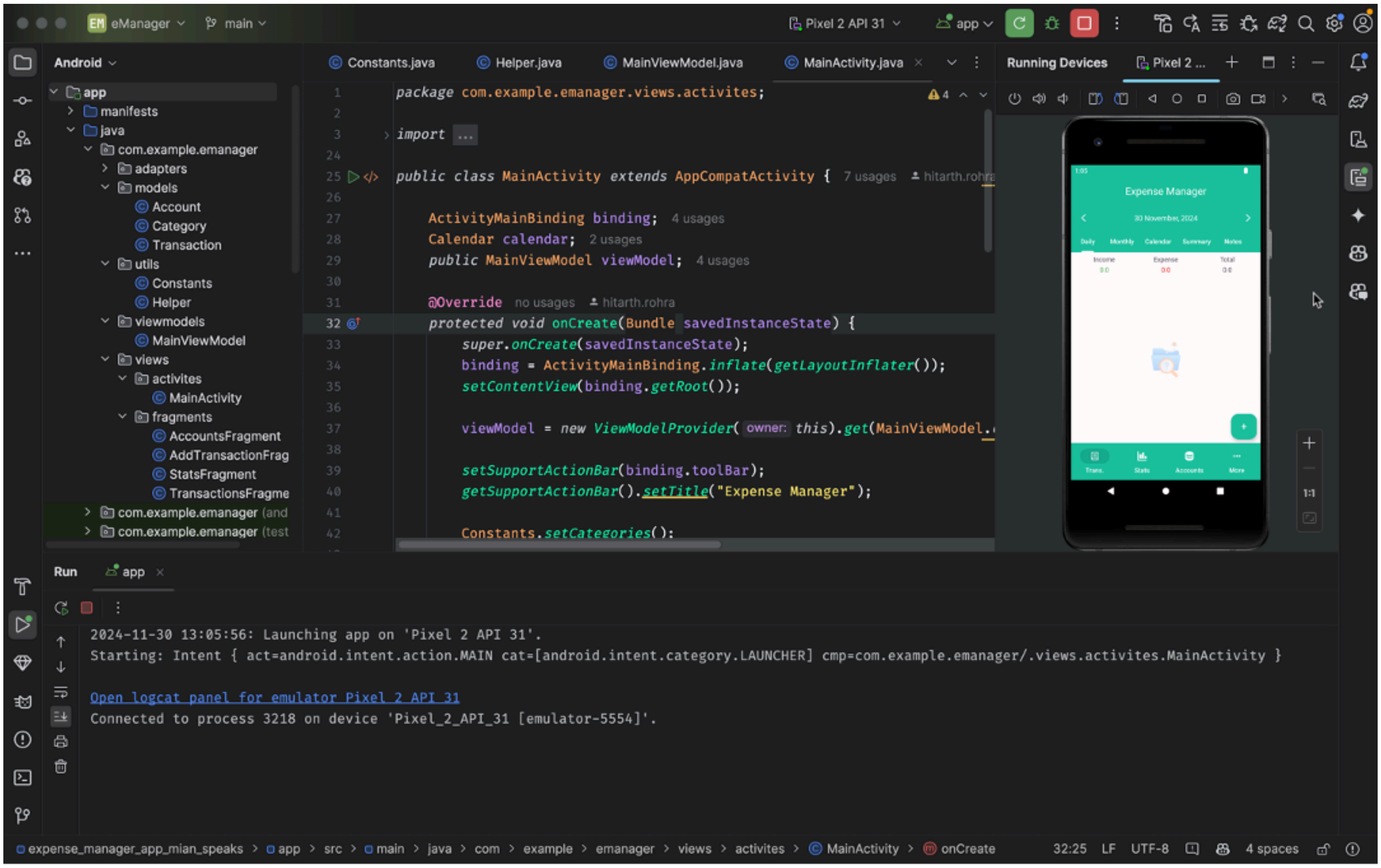
Challenges and Enhancements

- **Data Security and Privacy:** Ensuring the security and privacy of user data remains a paramount concern, particularly with financial information involved.
- **Scalability:** Future enhancements could include scaling the app to support a larger user base and integrating more complex financial management tools.
- **User Experience:** Continuous improvements can be made in the user interface to make it more engaging and easier to use.

App Design:


UI Artivity
Main Actevity
Transections
Fragnets
Acc Accanter
Add Transaction
Accounts
Accounts
Main Actortel
Acoain
Adoin
Models
Tarcaint
Model
Models
Transections
Adcepter
Acceantions
Amodel
Account
Stan
Categy
Add Trannaction
Helper

Code and Implementation:



```java
package com.example.emanager.views.activites;

import ...

public class MainActivity extends AppCompatActivity {

    ActivityMainBinding binding;
    Calendar calendar;
    public MainViewModel viewModel;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        viewModel = new ViewModelProvider(this).get(MainViewModel.

        setSupportActionBar(binding.toolBar);
        getSupportActionBar().setTitle("Expense Manager");

        Constants.setCategories();
```

```
2024-11-30 13:05:56: Launching app on 'Pixel 2 API 31'.
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.example.emanager/.views.activites.MainActivity }

Open logcat panel for emulator Pixel 2 API 31
Connected to process 3218 on device 'Pixel_2_API_31 [emulator-5554]'.
```

Data Storage and Handling in the App:

-- Accounts Table (to store different accounts/wallets)

```
CREATE TABLE accounts (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    account_name TEXT NOT NULL,

    account_type TEXT NOT NULL,  -- (cash, bank, credit card, etc.)

    initial_balance REAL DEFAULT 0.0,

    current_balance REAL DEFAULT 0.0,

    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,

    color_code TEXT,  -- for UI representation

    is_active INTEGER DEFAULT 1  -- boolean for soft delete

);
```

```sql
-- Categories Table (for expense/income categories)
CREATE TABLE categories (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    category_name TEXT NOT NULL,
    category_type TEXT NOT NULL,  -- (expense/income)
    icon_name TEXT,  -- for UI representation
    color_code TEXT,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    is_default INTEGER DEFAULT 0  -- to mark pre-defined categories
);

-- Transactions Table (main transactions table)
CREATE TABLE transactions (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    amount REAL NOT NULL,
    transaction_type TEXT NOT NULL,  -- (expense/income/transfer)
    description TEXT,
    category_id INTEGER,
    account_id INTEGER,
    transaction_date DATETIME NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    is_recurring INTEGER DEFAULT 0,
    recurring_type TEXT,  -- (monthly, weekly, yearly)
    recurring_date INTEGER,  -- day of recurrence
    FOREIGN KEY (category_id) REFERENCES categories(id),
    FOREIGN KEY (account_id) REFERENCES accounts(id)
);


-- Account Transfers Table (for tracking money transfers between accounts)
CREATE TABLE account_transfers (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    from_account_id INTEGER,
    to_account_id INTEGER,
    amount REAL NOT NULL,
    transfer_date DATETIME NOT NULL,
    description TEXT,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (from_account_id) REFERENCES accounts(id),
    FOREIGN KEY (to_account_id) REFERENCES accounts(id)
);
-- Budget Table (optional - for setting category-wise budgets)
CREATE TABLE budgets (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    category_id INTEGER,
    amount REAL NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (category_id) REFERENCES categories(id)
);
```

Conclusion

The Expense Manager App stands out as a robust tool for managing finances, suitable for both personal and professional use. Its development approach using modern technologies and architecture ensures a reliable and efficient user experience. Future updates may focus on expanding features such as real-time financial data integration and enhanced analytical tools.

For more details on the app, its setup, and its functionalities, you can visit the GitHub repository.