EXPENSES TRACKER

A PROJECT REPORT

Submitted by

LOGESHWARAN ELUMALAI (2116210701134)
MANJUNATHAN (2116210701147)
MOHAMED BASMAN (2116210701160)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING

COLLEGEANNA UNIVERSITY,

CHENNAI

MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Thesis titled **"EXPENSE TRACKER"** is the bonafide work of "**LOGESHWARAN ELUMALAI (210701134), MANJUNATHAN (210701147), MOHAMED BASMAN (210701160)**" who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Mrs., Ananthi S M.Tech.,

Assistant Professor (SG)

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai - 602 105

Internal Examiner **External Examiner**

ABSTRACT

This project focuses on the development of an expenses tracker application for Android devices, utilizing Kotlin as the programming language and Android Studio as the integrated development environment (IDE). The primary objective of the application is to provide users with an intuitive and efficient means to track and manage their personal expenses. Key features of the application include the ability to log daily expenditures, categorize expenses, set budget limits, and generate detailed financial reports. The application employs a modern, user-friendly interface designed with Material Design principles, ensuring a seamless user experience. It integrates with a local SQLite database for secure and efficient data storage, allowing users to access their financial records offline. Advanced functionalities such as data visualization through charts and graphs, as well as notifications and reminders for due payments, enhance the utility of the app. By leveraging Kotlin's concise syntax and advanced features, the development process is streamlined, resulting in a robust and maintainable codebase. The project also emphasizes adherence to the Model-View-ViewModel (MVVM) architectural pattern, ensuring a clear separation of concerns and facilitating easier testing and maintenance. Overall, this expenses tracker application serves as a comprehensive tool for personal finance management, helping users achieve better financial discipline and awareness through effective expense tracking and insightful analytics.

INTRODUCTION

In the modern world, effective personal finance management is crucial for achieving financial stability and meeting long-term financial goals. However, keeping track of daily expenses and managing budgets can be a daunting task without the right tools. To address this need, we present an expenses tracker application developed for Android devices using Kotlin and Android Studio. This application aims to simplify the process of monitoring personal finances by providing an intuitive and user-friendly platform for recording, categorizing, and analyzing expenses.

The core functionality of the expenses tracker includes the ability to log daily expenditures, categorize them under various headings such as groceries, utilities, and entertainment, and set budget limits to prevent overspending. Users can also generate detailed financial reports and visual representations of their spending habits, which can aid in making informed financial decisions.

Built with Kotlin, the application leverages the language's modern features and concise syntax to ensure a robust, efficient, and maintainable codebase. The use of Android Studio as the development environment facilitates seamless integration of various Android-specific functionalities and libraries, enhancing the overall performance and user experience of the application.

Adhering to the Model-View-ViewModel (MVVM) architectural pattern, the application ensures a clear separation of concerns, which improves code maintainability and testability. Additionally, the implementation of a local SQLite database guarantees secure and efficient storage of user data, allowing access to financial records even without an internet connection.

CODE

Transaction.kt

```
package dev.spikeysanju.expensetracker.model

import androidx.room.ColumnInfo
import androidx.room.Entity
import androidx.room.PrimaryKey
import java.io.Serializable
import java.text.DateFormat

@Entity(tableName = "all_transactions")
data class Transaction(

    @ColumnInfo(name = "title")
    var title: String,
    @ColumnInfo(name = "amount")
    var amount: Double,
    @ColumnInfo(name = "transactionType")
    var transactionType: String,
    @ColumnInfo(name = "tag")
    var tag: String,
    @ColumnInfo(name = "date")
    var date: String,
    @ColumnInfo(name = "note")
    var note: String,
    @ColumnInfo(name = "createdAt")
    var createdAt: Long =
        System.currentTimeMillis(),
    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "id")
    var id: Int = 0,
) : Serializable {
    val createdAtDateFormat: String
        get() = DateFormat.getDateTimeInstance()
            .format(createdAt) // Date Format: Jan 11, 2021, 11:30 AM
}
```

ExportCSVService.kt

```kotlin
package dev.spikeysanju.expensetracker.services.exportcsv

import android.content.Context
import android.net.Uri
import androidx.annotation.WorkerThread
import com.opencsv.CSVWriter
import com.opencsv.bean.StatefulBeanToCsvBuilder
import kotlinx.coroutines.flow.flow
import java.io.FileWriter
import javax.inject.Inject

class ExportCsvService @Inject constructor(
    private val appContext: Context
) {

    @WorkerThread
    fun <T> writeToCSV(csvFileUri: Uri, content: List<T>) = flow<Uri> {
        val fileDescriptor =
appContext.contentResolver.openFileDescriptor(csvFileUri, "w")
        if (fileDescriptor != null) {
            fileDescriptor.use {
                val csvWriter = CSVWriter(FileWriter(it.fileDescriptor))
                StatefulBeanToCsvBuilder<T>(csvWriter)
                    .withSeparator(CSVWriter.DEFAULT_SEPARATOR)
                    .build()
                    .write(content)
                csvWriter.close()
                emit(csvFileUri)
            }
        } else {
            throw IllegalStateException("failed to read fileDescriptor")
        }
    }
}
```

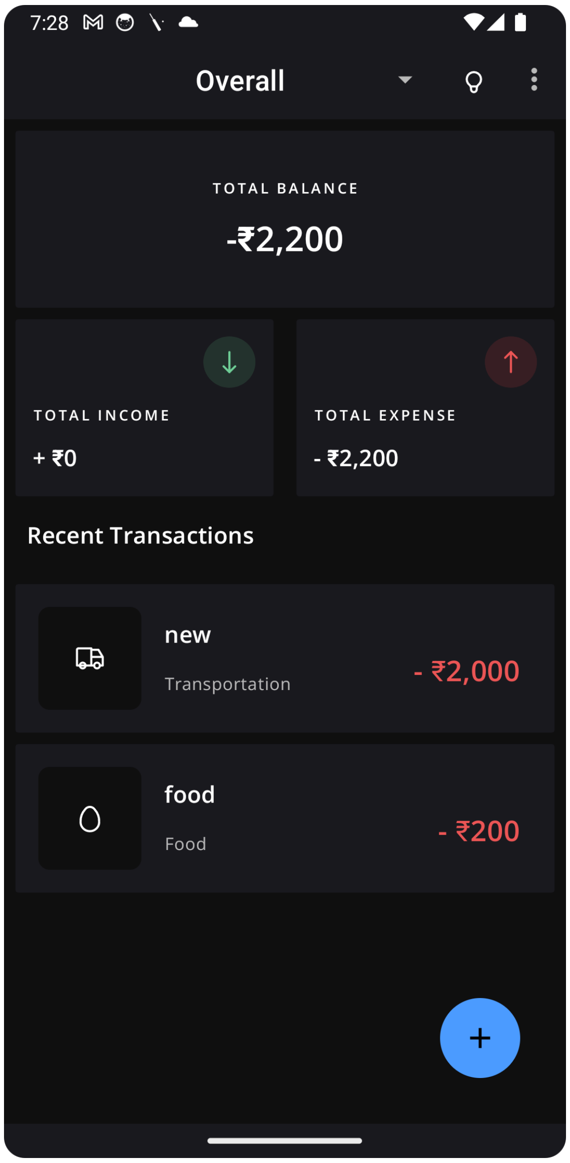
activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout xmlns:android="http://
schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".view.main.MainActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/appbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:backgroundTint="@color/surface"
        app:liftOnScroll="true"
        tools:ignore="UnusedAttribute">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?actionBarSize" />
    </com.google.android.material.appbar.AppBarLayout>

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/nav_host_fragment"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="?actionBarSize"
        app:defaultNavHost="true"
        app:navGraph="@navigation/nav_graph" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

dashboard.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout xmlns:android="http://
schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/background"
    tools:context=".view.dashboard.DashboardFragment">


    <androidx.core.widget.NestedScrollView
        android:id="@+id/main_dashboard_scroll_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fillViewport="true"
        android:scrollbars="none">


        <androidx.constraintlayout.widget.ConstraintLayout
            android:id="@+id/dashboard_group"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">


            <include
                android:id="@+id/total_balance_view"
                layout="@layout/total_balance_view"
                android:layout_width="match_parent"
                android:layout_height="124dp"
                android:layout_marginStart="@dimen/dimen_8"
                android:layout_marginTop="@dimen/dimen_8"
                android:layout_marginEnd="@dimen/dimen_8"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent" />
            <LinearLayout
                android:id="@+id/total_income_expense_view"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:baselineAligned="false"
                android:gravity="center_horizontal"
                android:orientation="horizontal"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@id/total_balance_view">

                <include
                    android:id="@+id/income_card_view"
                    layout="@layout/content_income_expense_card_layout"
                    android:layout_width="0dp"
                    android:layout_height="wrap_content"
                    android:layout_weight="1" />

                <include
                    android:id="@+id/expense_card_view"
                    layout="@layout/content_income_expense_card_layout"
                    android:layout_width="0dp"
                    android:layout_height="wrap_content"
                    android:layout_weight="1" />

            </LinearLayout>

            <TextView
                android:id="@+id/title_recent_transaction"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginStart="@dimen/dimen_16"
                android:layout_marginTop="@dimen/dimen_16"
                android:fontFamily="@font/open_sans_semibold"
                android:text="@string/text_recent_transactions"
                android:textAppearance="@style/
TextAppearance.MaterialComponents.Subtitle1"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@id/total_income_expense_view" />
            <androidx.recyclerview.widget.RecyclerView
                android:id="@+id/transaction_rv"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="@dimen/dimen_16"
                android:visibility="visible"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@id/title_recent_transaction"
                tools:listitem="@layout/item_transaction_layout" />

          </androidx.constraintlayout.widget.ConstraintLayout>
       </androidx.core.widget.NestedScrollView>

       <ViewStub
          android:id="@+id/emptyStateLayout"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_gravity="center"
          android:layout="@layout/content_empty_state_layout"
          android:visibility="gone"
          app:layout_constraintBottom_toBottomOf="parent"
          app:layout_constraintEnd_toEndOf="parent"
          app:layout_constraintStart_toStartOf="parent"
          app:layout_constraintTop_toTopOf="parent" />

       <com.google.android.material.floatingactionbutton.FloatingActionButton
          android:id="@+id/btn_add_transaction"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_gravity="bottom|end"
          android:layout_margin="@dimen/dimen_32"
          android:backgroundTint="@color/blue_500"
          android:contentDescription="@string/app_name"
          android:src="@drawable/ic_baseline_add"
          app:borderWidth="0dp"
          app:tint="@color/white"
          tools:ignore="UnusedAttribute" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

OUTPUT



7:28

Overall

TOTAL BALANCE

-₹2,200

TOTAL INCOME

+ ₹0

TOTAL EXPENSE

- ₹2,200

Recent Transactions

new

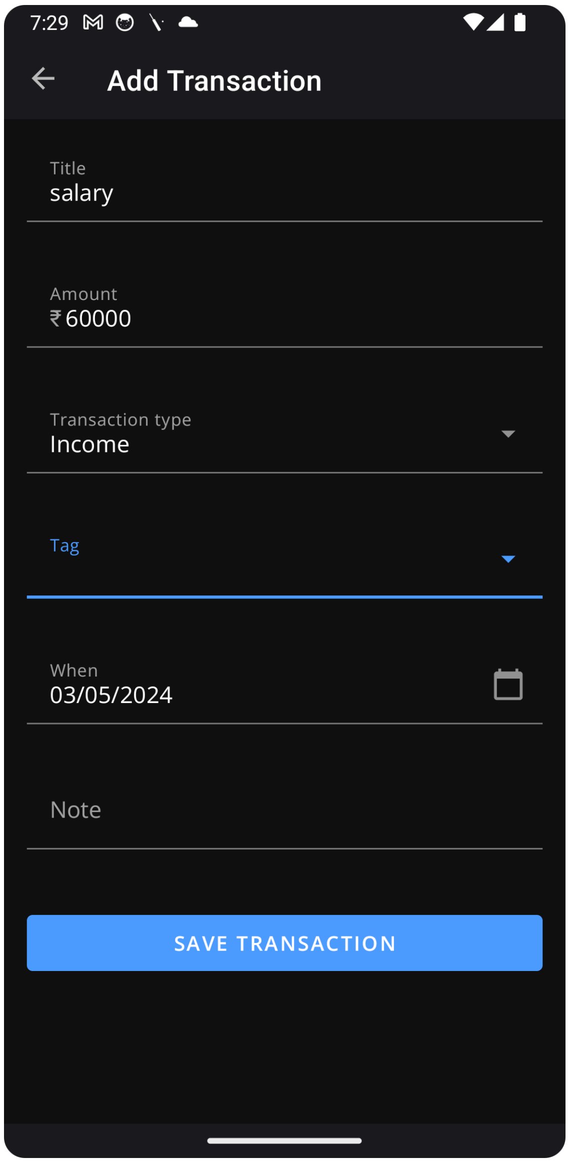
Transportation

- ₹2,000

food

Food

- ₹200



7:29

Add Transaction

Title
salary

Amount
₹ 60000

Transaction type
Income

Tag

When
03/05/2024

Note

SAVE TRANSACTION