SijoitusSovellus Development Guide

1. Table of Contents

1. Introduction
2. Software Components
3. App Structure and Connections
4. Known Problems and Development Challenges
5. Development Ideas

1. Introduction

Android application SijoitusSovellus is an application for easy and casual bookkeeping for traders and investors. The major point of the application is to keep everything as clean and simple as possible making using it fast and easy.

In the application the user enters information about the investments he takes, the software calculates total balance of money spent. When exiting positions, the information on profits and losses appears on the graph in the homepage.

So far the application supports Finnish and English languages. To switch languages swipe from left to right to open menu to choose correct localization (menu icon not working yet).

In the graph of the homepage, X-Axis is set to update daily and all profits/losses of a single day go together as a single entry. In Y-Axis we have scale in whatever currency the trader engages in.

2. Software Components

Classes

A. MainActivity

The main activity of the app, responsible for displaying the user interface and handling user interactions.

B. AddCashFlowAsyncTask

An asynchronous task for adding cash flow transactions to the database.

C. Converters

Utility class containing methods to convert between different data types, used by the Room database.

D. InvestmentAdapter

An adapter class that handles the display and management of investment items in a RecyclerView.

E. SecondaryDatabase

A secondary Room database class, houses CashFlow entries which are used to display profit/loss.

F. CashFlow

A data class representing a cash flow transaction, including properties like amount and date.

G. Investment

A data class representing an investment, including properties like name, purchase price, and quantity.

H. CashFlowDao

An interface defining the data access methods for cash flow transactions, used by the Room database.

I. BuyActivity

An activity responsible for handling the process of buying investments, including user input and database updates.

J. InvestmentViewHolder

A ViewHolder class for managing the display of investment items within a RecyclerView.

K. DeleteInvestmentAsyncTask

An asynchronous task for deleting investment entries from the database.

L. InvestmentDao

An interface defining the data access methods for investments, used by the Room database.

M. AppDatabase

The main Room database class that provides access to the app's data, including Investment entities.

N. SellActivity

An activity responsible for handling the process of selling investments, including user input and database updates.

Interfaces

A. CashFlowDao

An interface defining the data access methods for cash flow transactions, used by the Room database.

B. InvestmentDao

An interface defining the data access methods for investments, used by the Room database.

Libraries and Dependencies

- **androidx.appcompat:appcompat:1.6.1**: Provides backwards compatibility for the Android ActionBar and other Material Design UI elements, making it easier to create a consistent look across different Android versions.

- **com.jjoe64:graphview:4.2.2**: A library for creating interactive and customizable graphs and charts in your Android app.

- **com.google.android.material:material:1.8.0**: The Material Components library, which provides Material Design components and styles for your app, ensuring a modern and consistent UI.

- **androidx.constraintlayout:constraintlayout:2.1.4**: A library for creating responsive and flexible UI layouts using a flat view hierarchy and constraints.

- **com.github.PhilJay:MPAndroidChart:v3.1.0**: A powerful charting library for Android that supports a variety of chart types, such as line, bar, and pie charts.

- **junit:junit:4.13.2**: A Java testing framework used for writing and executing unit tests.

- **androidx.test.ext:junit:1.1.5**: An extension of the JUnit library for Android, which provides additional functionality for testing Android apps.

- **androidx.test.espresso:espresso-core:3.5.1**: A testing library for Android that enables you to write UI tests for your app and simulate user interactions.

- **androidx.recyclerview:recyclerview:1.3.0**: A library providing the RecyclerView component, which is an efficient and flexible way to display large data sets or data that changes frequently in your app.

- **androidx.room:room-runtime:2.5.1**: The Room persistence library's runtime component, which provides an abstraction layer over SQLite and helps you create a robust and efficient local database for your app.

- **androidx.room:room-compiler:2.5.1**: The Room annotation processor, which generates the necessary code for your app's database based on your Room annotations.

3. App Structure and Connections

MainActivity

- **MainActivity**: The main screen of the application, displays chart and balances and options for further actions.
- Database connections: InvestmentDao: retrieves information from AppDatabase, CashFlowDao: Accesses and retrieves cash flow data from the SecondaryDatabase.
- initializeSums: Initializes the total investment value and saves it to SharedPreferences. FetchCashFlowAsyncTask: Fetches the cash flow data and populates the LineChart.

- onCreate: Sets up the user interface, initializes the InvestmentDao and CashFlowDao, and sets the click listeners for buttons.
- onResume: Updates the total investment value and fetches cash flow data.
- onActivityResult: Handles the result from BuyActivity, updates the total investment, and fetches cash flow data.
- updateTotalInvestment: Updates the total investment value displayed on the screen.
- setupLineChart: Sets up the LineChart UI component.
- populateLineChart: Populates the LineChart with cash flow data.
- setLocale: Changes the app's locale based on the selected language.
- loadLocale: Loads the saved locale from SharedPreferences and applies it.

SellActivity

- Database and DAO:

  - InvestmentDao: Accesses and retrieves investment data from the AppDatabase.

- AsyncTask Class:

  - GetInvestmmentsAsyncTask: Fetches all investments from the InvestmentDao.

- Methods:

  - onCreate: Sets up the user interface, initializes InvestmentDao, and executes GetInvestmmentsAsyncTask.
  - onPostExecute (inside GetInvestmmentsAsyncTask): Sets up the InvestmentAdapter for the RecyclerView with fetched investments.

BuyActivity

-User Interface Elements:

- TextInputEditText: Provides input fields for stock price, stock name, and stock amount.
- Button: btnEnter and btnCancel handle the investment saving and activity cancel actions, respectively.

-Database and DAO:

- InvestmentDao: Accesses and stores investment data in the AppDatabase.

-AsyncTask Class:

- InsertInvestmentTask: Inserts a new investment into the InvestmentDao.

-Methods:

- onCreate: Sets up the user interface, initializes InvestmentDao, and sets click listeners for btnEnter and btnCancel.
- saveInvestment: Validates input, creates an Investment object, executes InsertInvestmentTask, and displays a success message.
- clearFields: Clears the input fields.
- doInBackground (inside InsertInvestmentTask): Inserts a new investment into the database.

4. Known Problems and Development Challenges

Pressing main menu icon doesn’t function properly. To open a menu swiping from left to right is required.

Balance in the main menu doesn’t display new balances when adding new investments in BuyActivity. To display new balance either restarting the application or selling an investment is needed.

Loading locale in onCreate of MainActivity.java causes stuttering and problems with starting the application, is disabled until fix is found.

5. Development Ideas

In the homepage display percentile of losses/profits compared to total balance. Further functionality to line chart by increasing or decreasing time scope. Realtime analysis of losses and profits by connecting to trading API (ie. Tradingview).