GitHub Username: henryragua

Personal Finances

Description

On the Personal Finance application, you can create a savings plan that will continually review and you'll know if you achieve your goal, also will help you record your financial transactions, shopping lists and expense projections.

You can project a savings plan and know at all times whether you are meeting or if you are spending more than necessary to fulfill your savings.

Everything you do in your application, you can do it intuitively, thanks to modern query tools and data editing.

Intended User

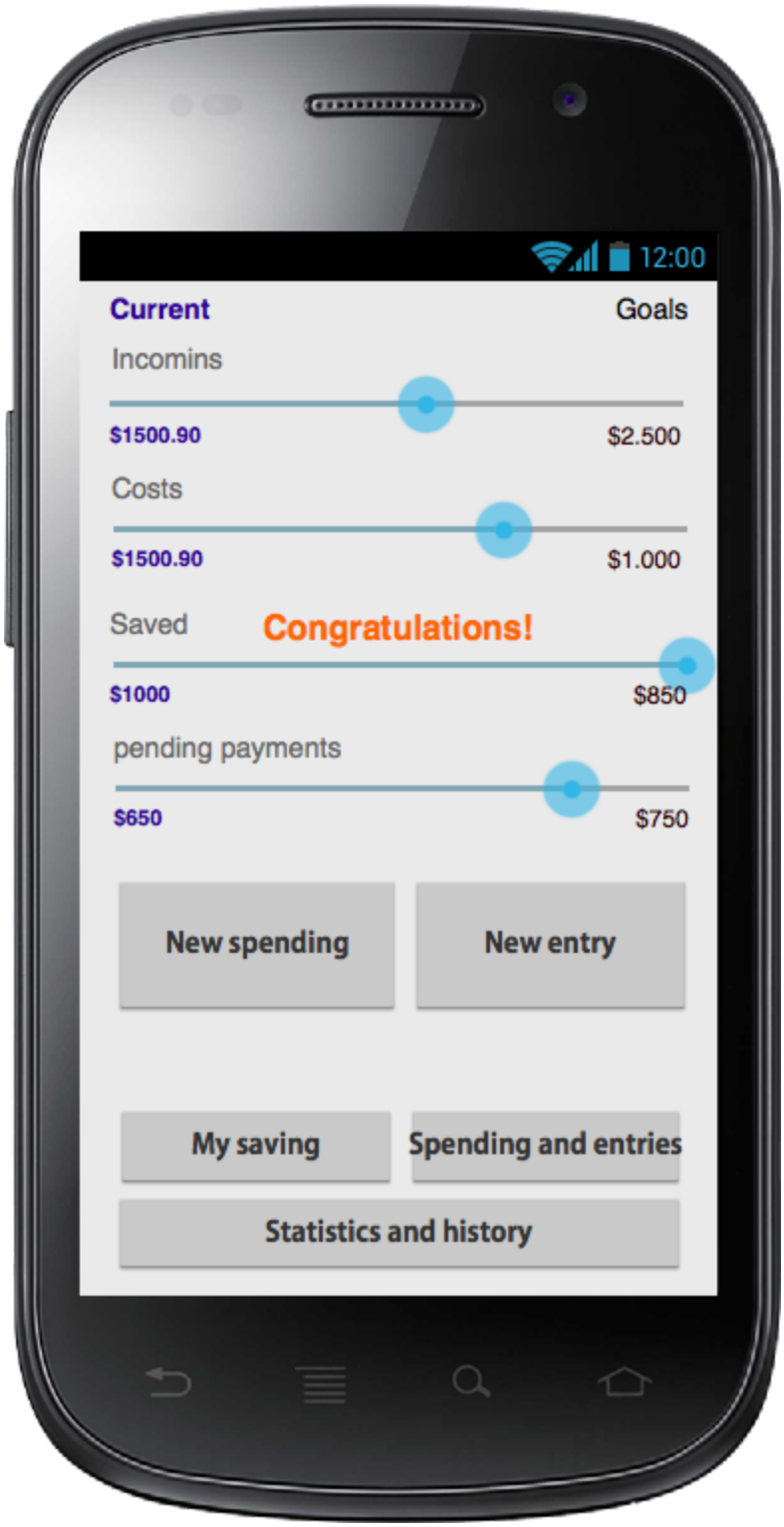
A person with income from salary, business, investment, etc., preferably having at your expense.

Features

- Check overall financial state
- Register spending money
- Register cash income
- Configure saving money
- Configure recurring costs
- Configure regular income
- Check state savings
- Check costs money
- Check incomes
- Check recurrent costs
- Check periodic income
- Statistics and history

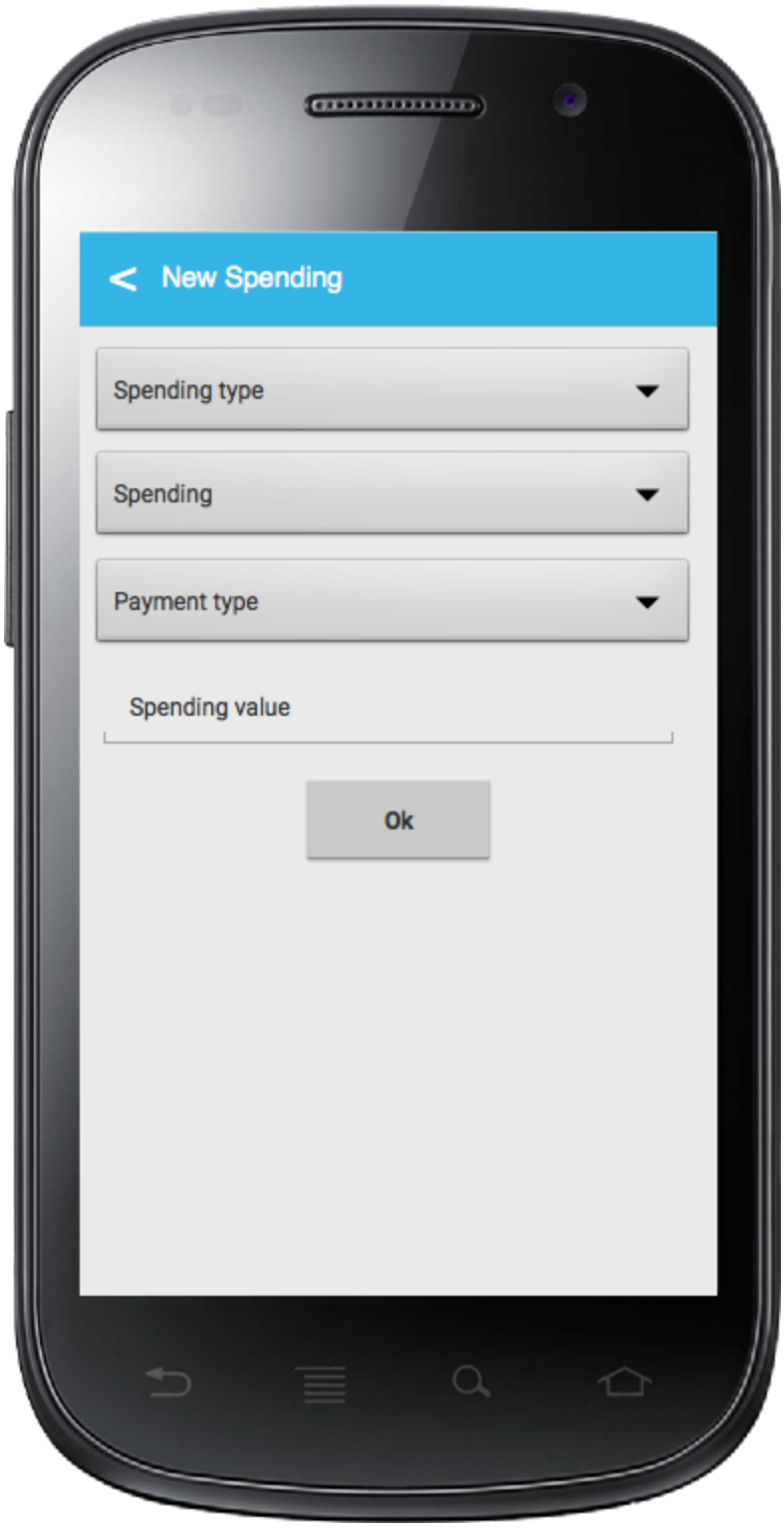
User Interface Mocks

Main screen



Main screen and the main options

New Spending

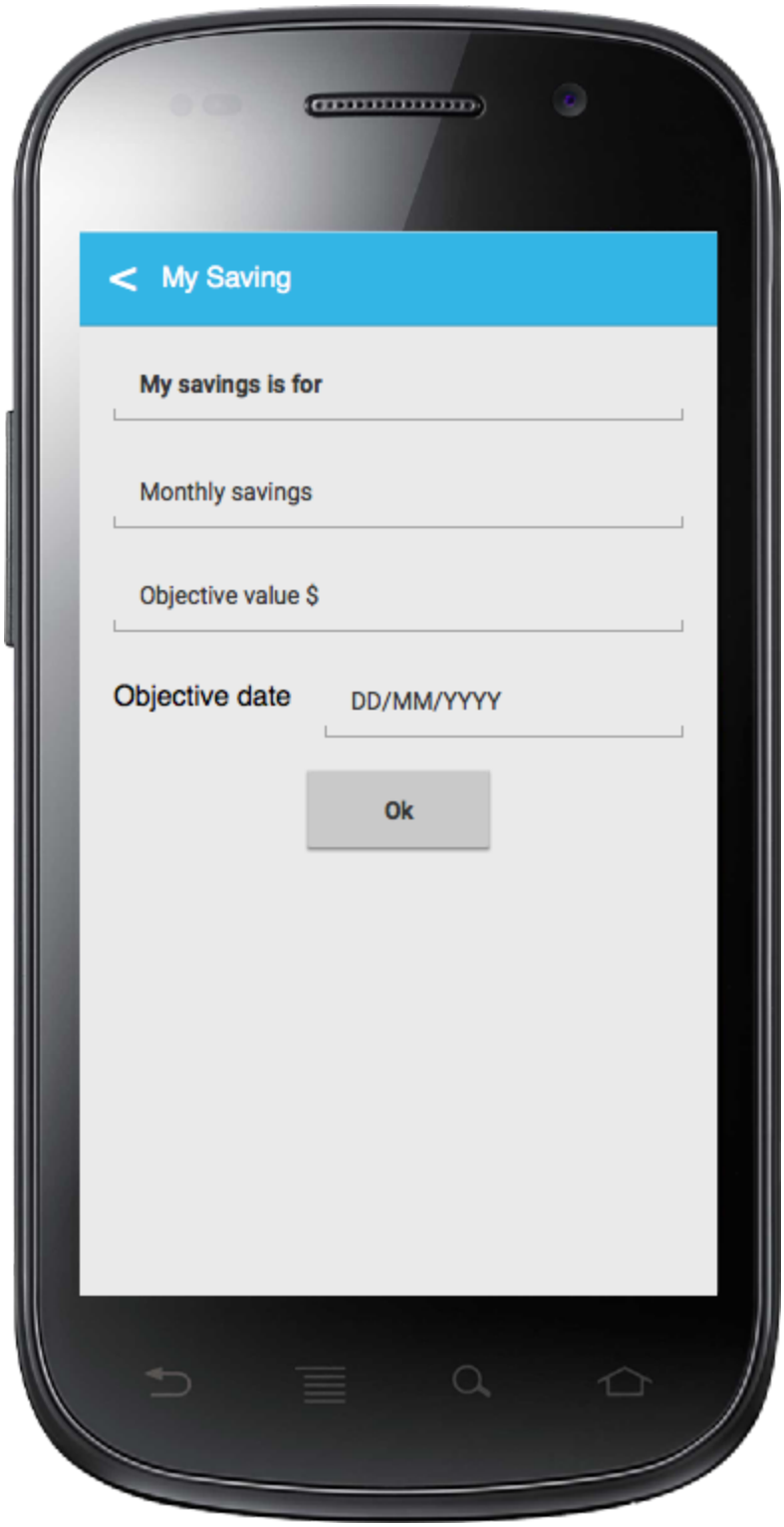


Register a new spending

New Entry

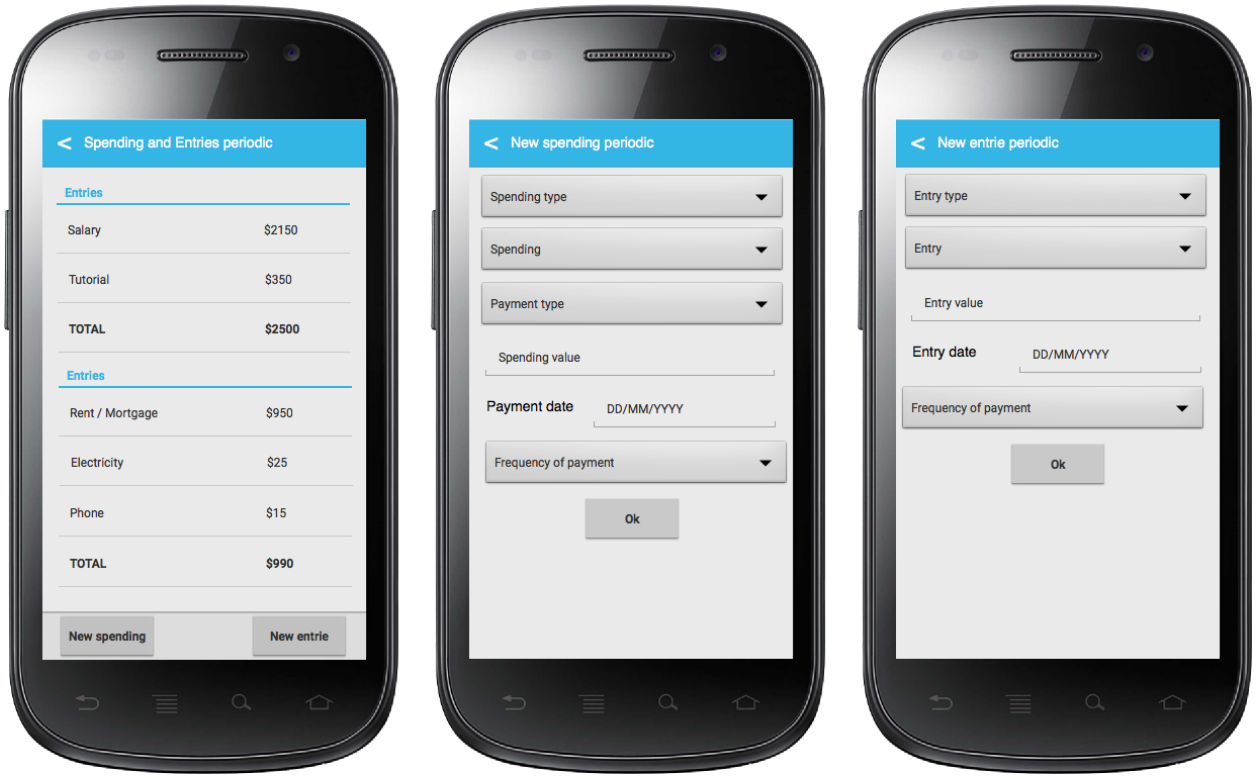
Register a new entry

My Saving



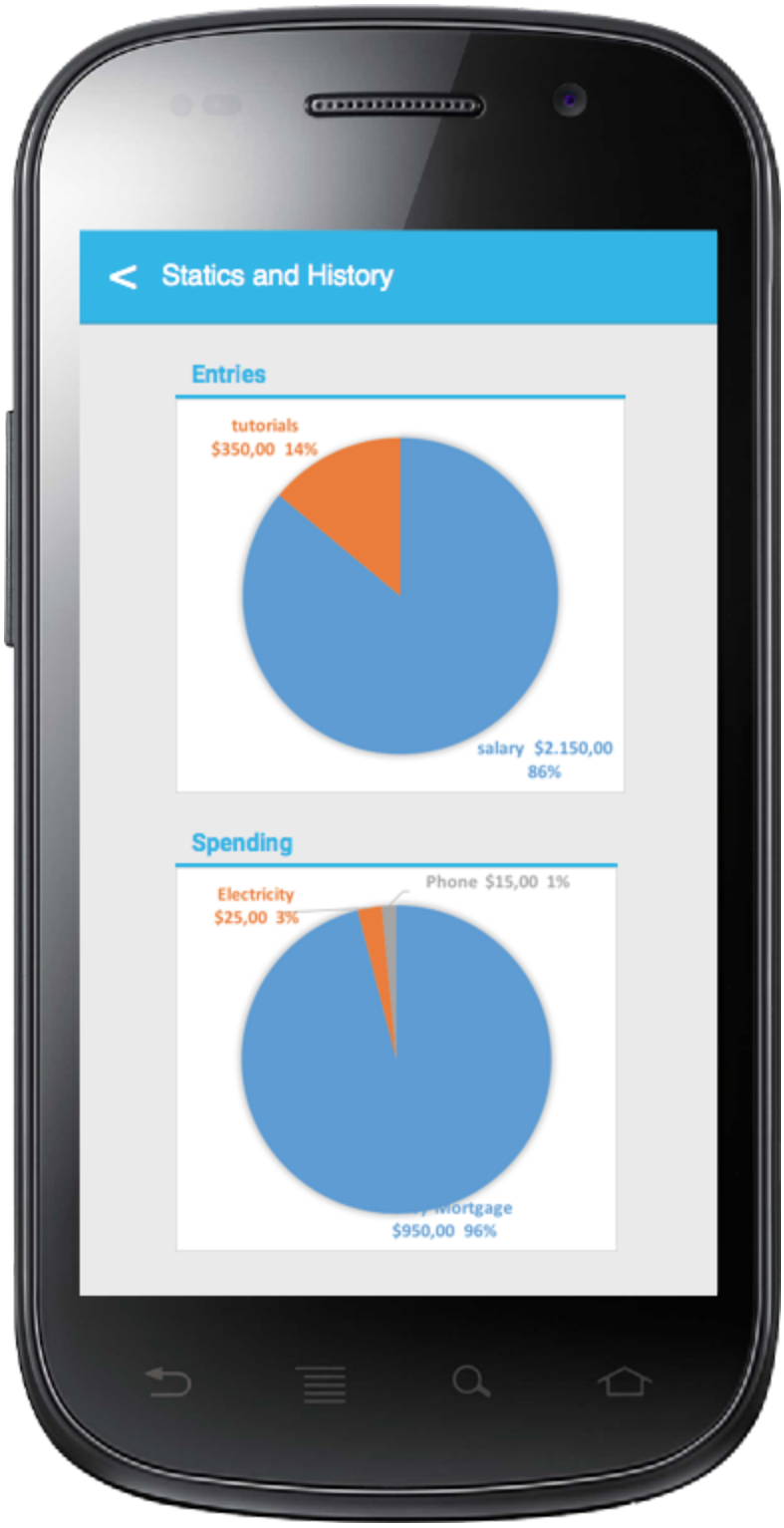
Configure the saving

Spending and entries periodic



Configures and queries the spending and entries periodic

Statics and History



Queries the detail of the transactions

Key Considerations

How will your app handle data persistence?

Persist application data via a Content Provider

Describe any corner cases in the UX.

The states of the screen whenever a pause in the activities is made will be saved, and values will be restored in the resume activity

Describe any libraries you'll be using and share your reasoning for including them.

- MPAndroidChart: For handling charts in reports
- Picasso: For loading images
- Butter Knife: For bind Android views and callbacks to fields and methods.
- Spinner Material: For support of the spinner with the hint text.
- CompactCalendarView: For queries by date.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Creating the Project with compatibility with Android API enter: minimum 16.
- Android create colors Material.
- Create text files for display messages for languages English (default) and Spanish.
- The libraries are set to be used in the application.
- Try running the application.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for New spending fragment

- Build UI for New entry fragment
- Build UI for My Saving fragment
- Build UI for the transaction item layout
- Build UI for the list of the Spending and entries fragment
- Build UI for the new spending fragment
- Build UI for the new entry fragment
- Build UI for the Statics and History main fragment
- Build UI for the saving history detail fragment
- Build UI for the spending history detail fragment
- Build UI for the entry history detail fragment

Task 3: Creation of ContentProvider

- Crear el contrato de la base de datos.
- Desarrollo del DBHelper.
- Creación del ContentProvider.
- Configuración del ContentProvider en el AndroidManifest.
- Configuración de los UriMatcher en el ContentProvider.
- Implementación de las funciones: query, insert, bulkInsert, update y delete en el ContentProvider.
- Test al ContentProvider.

Task 4: Implementation of activities

- The instance of the views on the controller classes is performed.
- The information on the views from the ContentProvider, through LoaderCallbacks is loaded.
- The Providers are created for lists to deploy the application.
- Update functions data through ContentProvider develop.

Task 5: Implementation

- It verified that the functions defined are implemented in the application.
- The application is installed on different devices of different characteristics and are tested all the functionalities.