

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

**GitHub Username:** aku-sama

# ExpTracker

## Description

If you have some problems with management your expenses, ExpTracker app is all that you need. You can add your purchases to different expenses categories and see the full statistic about your money usage. You can add cheques to the app and you don't forget what you buy.

## Intended User

All men and women, who interested in expenses monitoring and need to plan their budget. All people who want to know the answer to the question "Where is my money? I really don't remember what I bought".

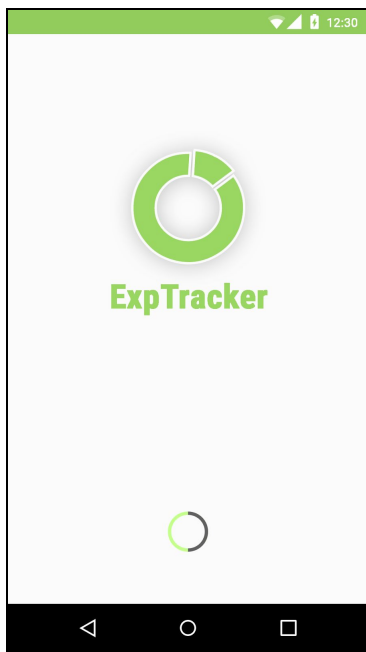
## Features

- Add purchase with cheque attachment
- Select purchase category
- Protect app access by PIN - code
- View expenses statistic for selected period
- View expenses detalization for selected period

## User Interface Mocks

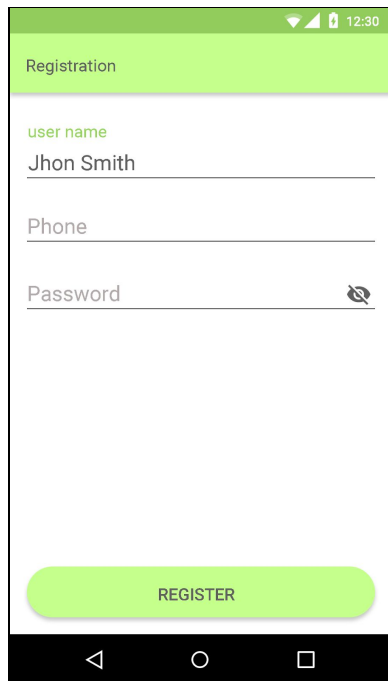
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

### Screen 1



Splash screen: loading screen, show after the app start.

## Screen 2



Registration

user name  
Jhon Smith

Phone

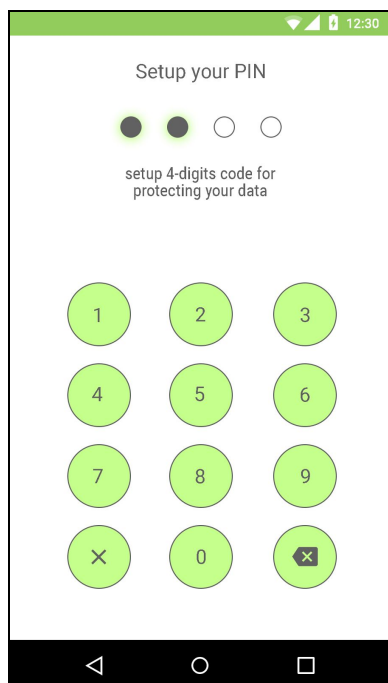
Password

REGISTER

The registration screen features a light green header with the title 'Registration'. Below the header, there are three input fields: 'user name' (containing 'Jhon Smith'), 'Phone', and 'Password' (with a toggle icon). A green 'REGISTER' button is positioned at the bottom of the form. The screen is framed by a black Android navigation bar at the bottom.

Authorization screen: for the app usage and user identification, first time you need create user account.

## Screen 3



Setup your PIN

● ● ○ ○

setup 4-digits code for  
protecting your data

1 2 3

4 5 6

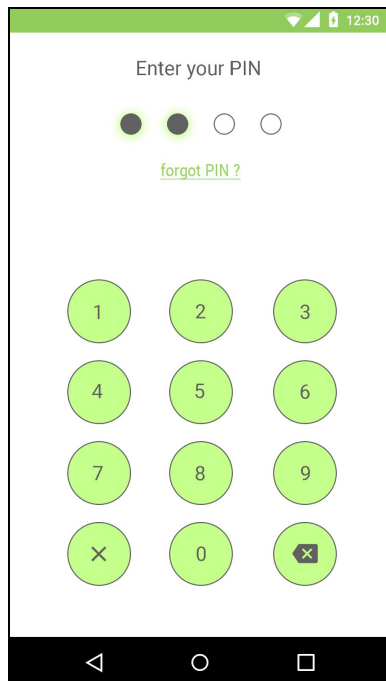
7 8 9

X 0 X

The 'Setup your PIN' screen has a light green header with the title 'Setup your PIN'. Below the header, there are four circles representing the PIN digits, with the first two filled and the last two empty. Below the circles, the text 'setup 4-digits code for protecting your data' is displayed. A numeric keypad is shown below the text, consisting of a 4x3 grid of circles containing the numbers 1 through 9, 0, and two 'X' symbols for backspace. The screen is framed by a black Android navigation bar at the bottom.

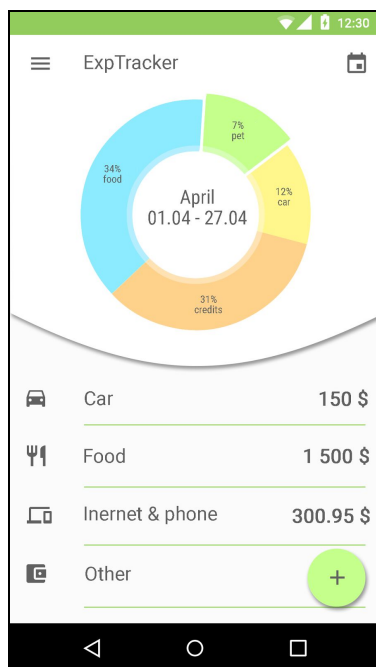
Set PIN: a user should set up 4-digits access code for protecting access to him/her expenses information.

## Screen 4



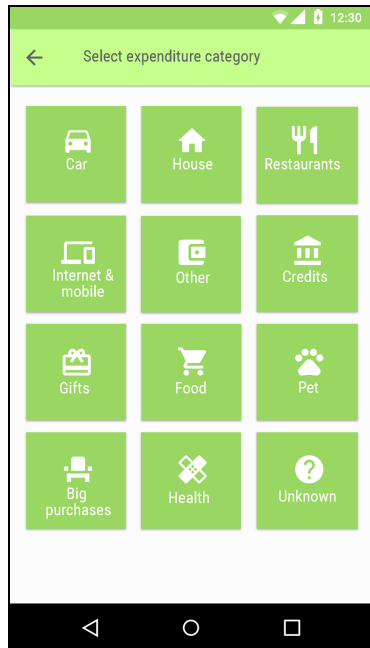
Enter PIN: after authorization and PIN setup, a user doesn't need to login every time. Users should use this screen for entering PIN and access to the app data.

## Screen 5



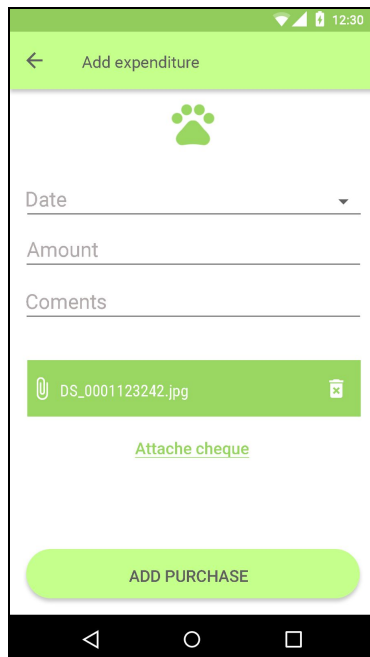
Main screen: Home screen of the app. A user can view statistics about expenses in categorized chart and summary expenses list. A user can filter date range of statistics. User can add new purchase by FAB

## Screen 6



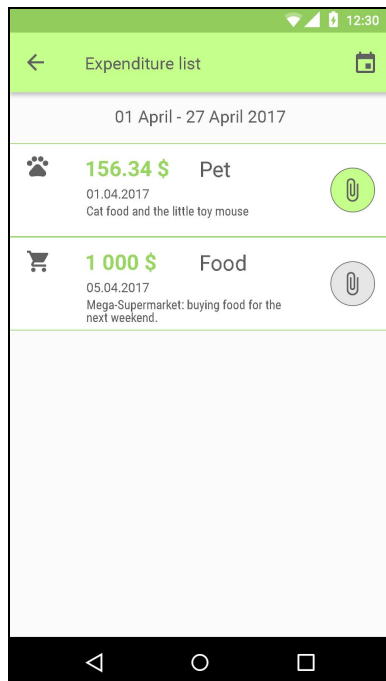
Category select screen: When user add new purchases , first he or she need to select expenditure category.

## Screen 7



Add expenditure step 2: after purchase category select, a user should set date (today by default) and an amount of purchase. Additionally user can attach cheque photo from gallery or camera.

## Screen 8



Expenditure detalization list: user can view full details about purchases in selected period. User can change a period filter. The user can view all attached cheques. This feature available via Navigation Drawer menu.

## Key Considerations

### How will your app handle data persistence?

I will create my own content provider for data storing and handling.

### Describe any corner cases in the UX.

If user forgot her/ his PIN code, the app will do logout from user account and user can re-login and create new PIN. User data don't be lost in this case.

If the user wants to delete any added purchase, he/ she need use swipe-to-delete option.

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

Glide to handle the loading and caching of images.

Butterknife for view injection.

Mosby for handle view states (screen orientation change and etc.)

MPAndroidChart for create statistics diagram.

EverythingMe/easy-content-providers for create own ContentProvider

JUnit for unit testing.

**Describe how you will implement Google Play Services.**

The app don't need Google Play Services.

## Next Steps: Required Tasks

### Task 1: Project Setup

- Create Git repository
- Create and configure new Android studio project
- Configure libraries
- "Cut" design sources.

### Task 2: Implement UI for Each Activity and Fragment for phone and Tablets

- Build UI for Splash screen
- Build UI for Registration screen
- Build UI for PIN set
- Build UI for PIN enter screens
- Build UI for MainActivity
- Build UI for Navigation Drawer menu
- Build UI for Expenditure List
- Build UI for Adding purchase: category selection
- Build UI for Adding purchase: information form.
- Build UI for app dialogs : PIN forgot, Purchase adding result, Purchase delete result.

### Task 3: Create Base classes

- Create parent class for all Activities and Fragments
- Create base classes for all MVP object. (BaseView, Base Presenter and etc.)

#### **Task 4: Create Content Provider**

#### **Task 5: Create full app logic**

- Create business logic unit tests
- Create content provider
- Create all app logic

#### **Task 6: Add animations**

#### **Task 7: Test the app**