How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: “**Capstone_Stage1**”
3. Replace the text in green

GitHub Username: bendaf

Spip - SPlit trIP expenses

Description

The purpose of the app is to split travel bills between the travelers in a way that everyone participate only at their own expenses. The user can create trips with participants and share their expenses between them easily. The app includes a widget where you can add expenses instantly

Intended User

Group members who pay bills for the others and others for him and would like to pay off debts easily. The app will have English and Hungarian language support.

Features

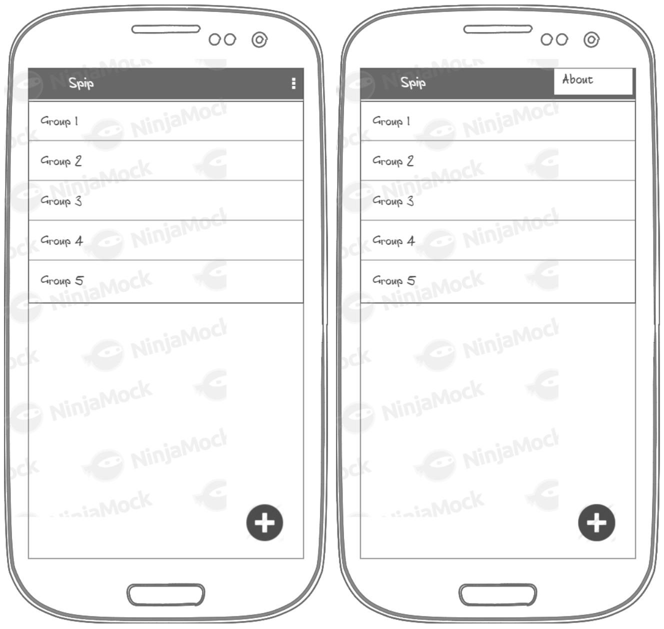
The main features of Spip.

- Manages Groups and Group members
- Stores and lists expenses
- Displays the personal balances and average spendings
- Have a widget to add expenses easily
- Integrates Firebase auth to authenticate user
- Java based application
- Uses Google analytics to help examine user behaviour

User Interface Mocks

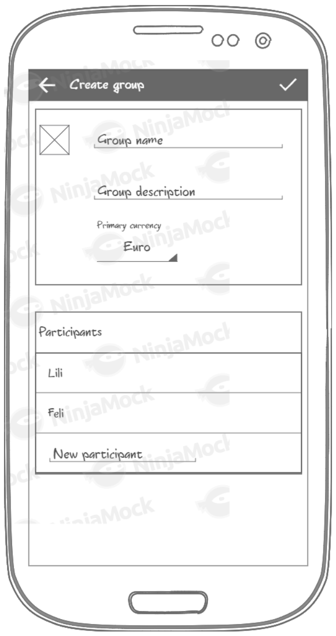
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.
I have created my mocks in ninjamock.com: https://www.ninjamock.com/s/CM78LTx

Screen 1



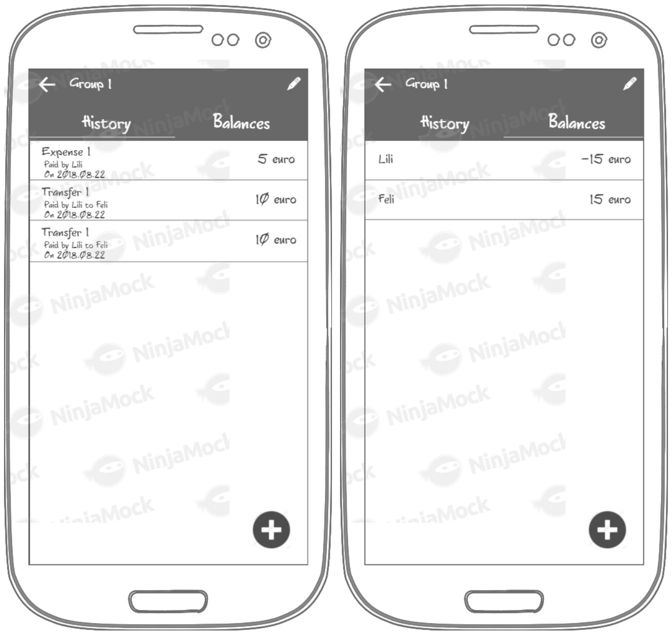
The main screen listing the groups. Clicking on a group takes us to Screen 3 while tha FAB button takes us to Screen 2

Screen 2



Create group screen. Every detail can be filled up. If a new participan name is typed than a bonus row is added to the participants recyclerview. Clicking on the back button it takes us to Screen 1 while the pipe takes us to Screen 2

Screen 3



Expense 1 Paid by Lili On 2018.08.22 5 euro

Transfer 1 Paid by Lili to Feli On 2018.08.22 10 euro

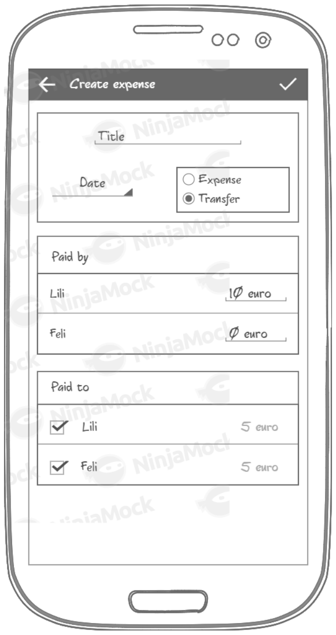
Transfer 1 Paid by Lili to Feli On 2018.08.22 10 euro

Lili -15 euro

Feli 15 euro

The two tab of screen 3: One where the balances are listed and the other where the history of the expenses. Clicking to the back button takes us to Screen 1 while the edit icon takes us to Screen 2. Clicking on the FAB takes us to Screen 4.

Screen 4



Lili 10 euro

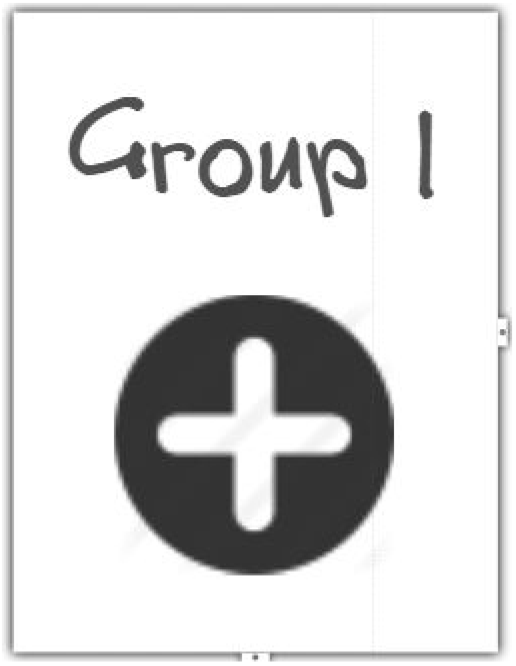
Feli 0 euro

Lili 5 euro

Feli 5 euro

Add new expense tab. Every detail can be set than clicking on the pipe or the back arrow icon takes us to Screen 3 the difference is that the pipe adds the new expense (or edits it) while the back arrow don’t.

Widget



Width the help of the widget users can easily add new expenses to a group.

Key Considerations

How will your app handle data persistence?

I will build a content provider for my database which will store the groups and expenses and members in different tables.

Describe any edge or corner cases in the UX.

The app will not support tablets, since the mobiles are more handy for this purpose.
The rotation will also be disabled, most of the screens require keyboard, and they would not be usable in landscape mode.
The network will be used by only the Google play services and they will be optional, so the app should work without network connection.
I will enable RTL switching and make special attention to accessibility.

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

I will use Android Studio 3.0.1 with gradle 3.1.4.

I will use the android support libraries version 27.1.1.
I use schematic version 0.7.0 to generate content provider code for me.
I will use timber version 4.7.1 to simplify loging.

Describe how you will implement Google Play Services or other external services.

I will use Google analytics to log the user behaviour, detect how many groups and transactions are added by an average user and which are the most used features in the app.
I will also use google Auth to authenticate the user if they want to autocomplete their username.

Next Steps: Required Tasks

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

Task 1: Project Setup

- Configure libraries
- Configure AndroidMainfest

Task 2: Create Content Provider

Create the database and content provider for the expense management.

- Create tests
- Create SQLite database
- Create Content Provider with the help of Schematic
- Test the Content Provider

Task 3: Implement UI for Screen 1

- Create layout for Screen 1
- Implement screen logic
- Use Loader for data loading
- Add Google Analytics to screen

Task 4: Implement UI for Screen 2

- Create layout for Screen 2
- Implement screen background logic
- Add Google Analytics to screen

Task 5: Implement UI for Screen 3

- Create layout for Screen 3
- Implement screen logic
- Use loader for data loading
- Add Google Analytics to screen

Task 6: Implement UI for Screen 4

- Create layout for Screen 4
- Implement screen logic
- Add Google Analytics to screen

Task 7: Implement Google Authentication possibility for the user

- Create an option to authenticate and automatically fill the name of the first group member.
- Make authentication Async to not disturb the main thread with an AsyncTask.

Task 8: Create widget

- Save the currently active group in a shared pref
- Create a widget which contains a link to add expenses easily to the group

Task 8: Test the application

- Test the ui functionality
- Test stability

Add as many tasks as you need to complete your app.

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
- Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”