POLITECNICO
MILANO 1863

Split Your Expenses

Design Document

Design and Implementation of Mobile Application A.Y. 2021/2022

Alessandro D’Onofrio

Danilo Castiglia

Contents

Introduction

1.1 Why Split Your Expenses?

The idea of this application comes from the possibility to have a facility that would help us in the common life of university roommates that have to split some of the expenses for the house they share. For Instance, you have to go to the supermarket to buy some products for house cleaning and you want to divide the expense's amount with some of your housemates, but suddenly you realize that one of them lend you some money the day before, in this case we are in front of a give-have cycle that could be difficult to manage.

Thinking about ways of solving this problem, we have understood that we can extend this paradigm to a vastness of aspect of common real life. We are speaking about travel expenses, couple shared expenses or also the organization of a specific public or private event. This idea had to be realized due to the fact that could help people in many different ways in the management of their wallets.

The purpose of SYE is to manage a certain group of expenses in a fast and smart way, giving also some hints about the splitting of these ones and some stats about the group and the people that compose it.

1.2 Functionalities

In order to realize the purpose of this project, the functionalities currently implemented in the app are:

- **Create a Group of Expenses**: the user can create a new group choosing name, description, currency of the group, category and the people involved in it;
- **Edit a Group**: the user can edit name, description and user of a group;
- **Delete a Group**: the user can delete a group and all the expenses associated;
- **Import a Group**: the user can import a group using a specific link to it;
- **Add a new Expense**: the user can create a new expense choosing title, amount, currency of the expense, date, payer and list of users that participate to it;
- **Edit an Expense**: the user can edit an expense;
- **Delete an Expense**: the user can delete an expense;
- **Visualize Users' Balances**: the user can visualize the balances of each member of the group related with what they have paid or not. A positive balance means the user has to receive money back, a negative balance the opposite;
- **Visualize Group Statistics**: the user can visualize group statistics such as the total amount of all the expenses in the group and who is the member that spend the large amount of money among the participants of the group.

1.3 Implementation Choices

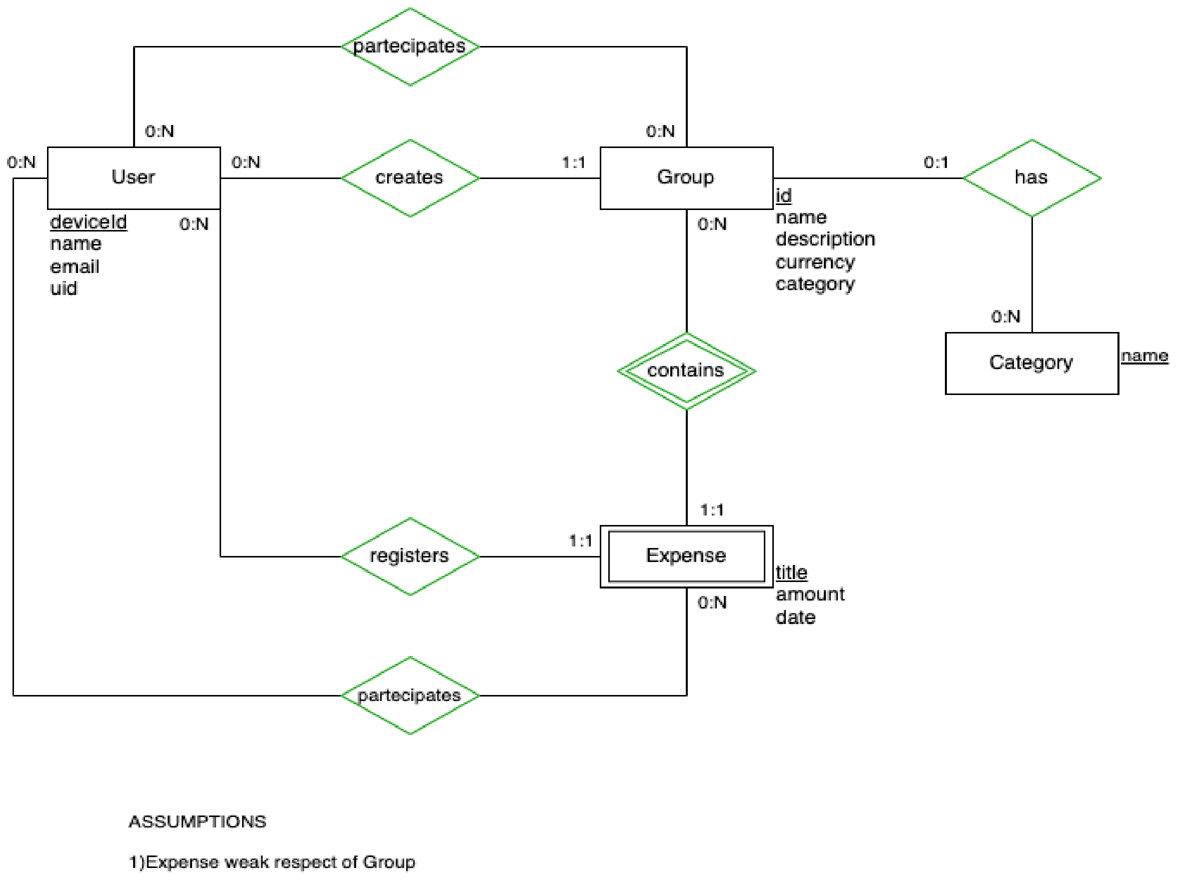
While implementing the app functionalities, the choice was to focus more on the usability and on making the app usage as much intuitive as possible, more than implementing a lot of features.
Thus the key feature of the whole application is to share knowledge gathered from students' experience in such a way that it won't get lost into the chaos generated by a complex app.

A lot of importance has been given at making a responsive and gladly to see interface. To perform this we have chosen Flutter as developing framework to deal with few but effective widgets that optimize the User Interface. We need also fast access to data, that should be stored as they are, without being normalized; to do this we have chosen Firebase as database and backend manager.

Architectures

2.1 Data Model

It is not easy to deal with huge amount of data without having their structure and their connections in mind. Although we are not using a relational database to store and manage the data, we preferred to represent them in a common and clear model called ER diagram. This will help us understand how data works together:



2.2 Data Management

The data is stored on Google Firestore, and is retrieved by the application via future builders and native method of the FlutterFireUI liberary. This means that every change of the data into the database is immediately notified to the application that update the user interface accordingly.

We implemented the db.dart abstract class that contains a set of static methods that communicate with FireBase, the DB class has these methods:

- AddUser(): create a reference to the device id in the database.
- GetExpenseList(): return the expenses list of a specific group.
- GetUsersList(): return the list of group's participants.
- IsGroupPresent(): check if a group is already associated with a user.

- GetExpense(): returne a single expense.
- AddExpense(): create a new expense into a group.
- AddGroups(): create a new group associated with the creator.
- ImportGroup(): import an existing group through a link.
- EditExpense(): edit a single expense in the database.
- GetGroup(): return a single group from the database.
- DeleteExpense(): delete an expense from the database.
- DeleteGroup(): delete a group from the database.
- EditGroup(): edit a group.
- RegisterUser(): saves the uid of a logged user.
- GetTotalGroupBalance(): return a Map of all the expenses amount associated with payer.
- GetBalances(): return a Map of users' balances through all the expenses in a group.

2.3 External libraries

- **FlutterFireUI**

This library provides useful widgets to render the elements contained in the DB. We used this library to manage query on DB and to render list of elements.

- **FlutterSlidable**

Implements the swipe to delete functionality. We used it in both the Groups List and Expenses List.

- **ReactiveForms**

Enable reactive paradigm in flutter forms, manages the validators and check the user input. We used it in the New Expense form.

- **FirebaseAuth**

Manages Registration and Login/Logout of the user, saving password in a secure way.

- **DevicePreview**

Enables the testing on different devices with both iOs and Android and with different sizes.

- **CurrencyPicker**

This library offers a beautiful selector with currencies from all over the world. Note that this library only offers the list of the currencies, not the current value.

- **SyncFusion Flutter Chart**

This library, written completely in Dart, offers us the possibility to manage and display different type of charts. The library let you handle of the aspect of chart visualization, rom the type of diagrams to render to the data they are analysing.

It is completed with animations and smart rendering of the data. The main widget of the library are the chart, that let you display any type of chart, fixing their dimension and position in the parent widget and the Series widget (ColumnSeries and PieSeries) that control the visualization of the data in the chart, displaying labels and legends of them.

In the application we use the SfCartesianChart and the SfCircularChart to display balances data.

2.4 External services

- **FreeCurrencyAPI**

We used “freecurrencyapi.net” APIs to retrieve the real-time value of all the currencies and convert one value into each other.

- **FirebaseDatabase**

It is the DB provider that we decided to use to maintain the Group state saved and updated between devices.

3. Design

3.1 User Interface

The use of Flutter creates a User Interface where everything works together to create beautiful and attractive layouts and renders. Our implementation is easy to understand but still attractive and not tedious for the user. This because we want to guarantee a painless experience that doesn’t bother the customer during the usage of the app.

3.1.1 Smartphone UI

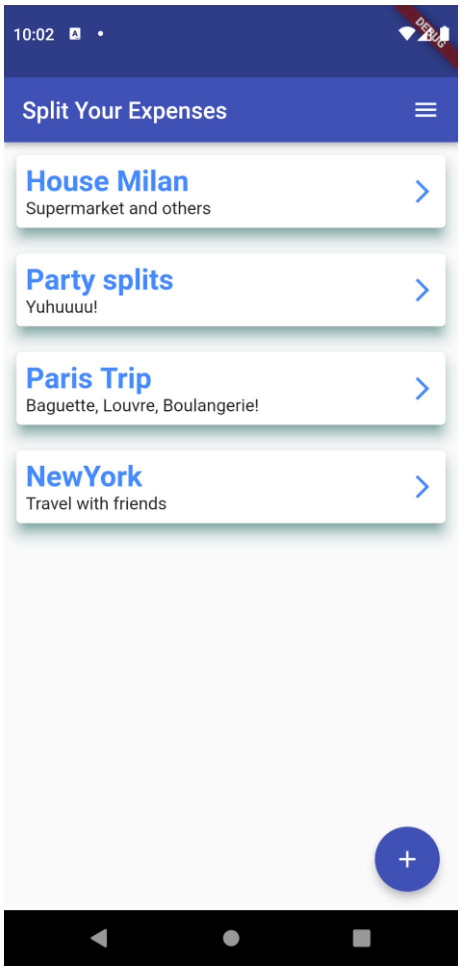
The App was initially designed for smartphone and later we decided to add the support for a different layout optimized for tablet.

- **Group List:**

This page returns a Scaffold widget with an App Bar and a body that contains the list of groups created the specific device that is visualizing them. In fact, the application saves the device id the first time the user opens the application and match each further element to it. In this way a user will always visualize his own groups without a login.

In this page we download groups data from firebase and render them in a specific widget, the GroupVisualizer, that shows us the name of the group and its description. It is a tappable item that leads the user to the expenses list of the group.

The GroupVisualizer is wrapped in a SwipableItem that let the user delete the group only swiping left. In the bottom right side of the screen, we can see a FloatingActionButton that let the user navigate in the creation group form.



House Milan
Supermarket and others

Party splits
Yuhuuuu!

Paris Trip
Baguette, Louvre, Boulangerie!

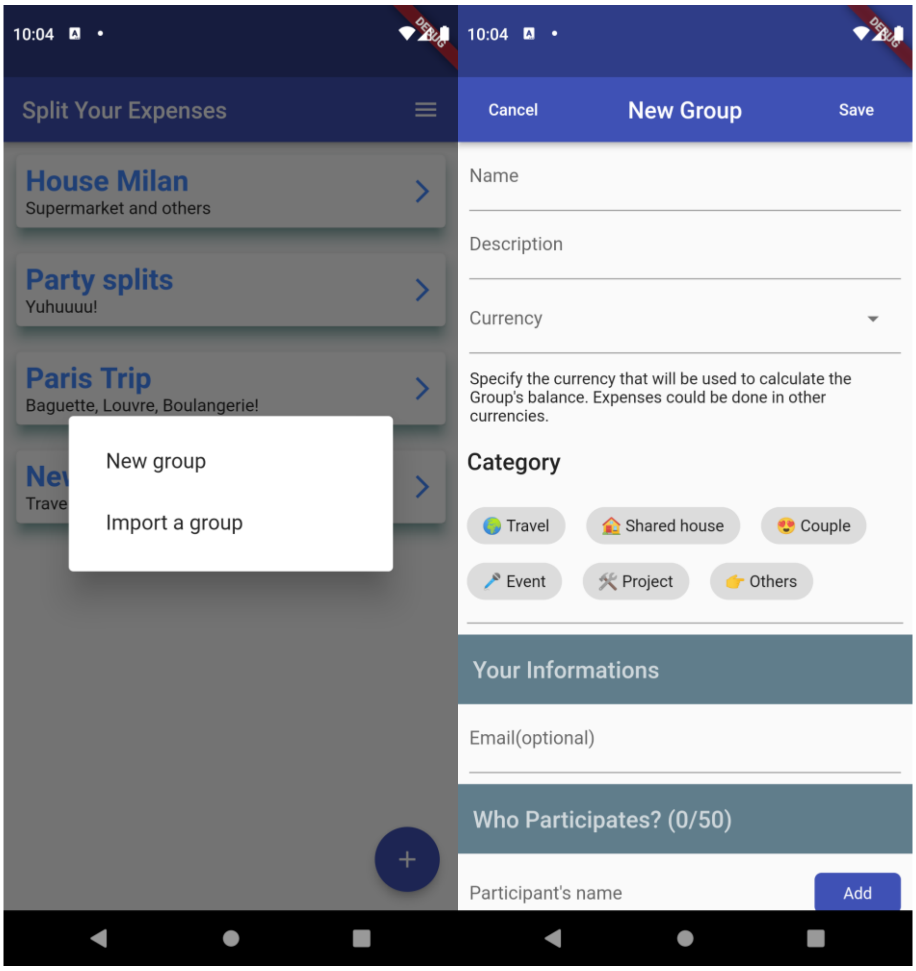
NewYork
Travel with friends

- **Group Creation Form:**

Clicking the **+** button in the group list page open a selection between New Group and Import a Group.

The Group Creation Form shows the user a form that let him create a new user. For this specific page we have used a specific library that manages form in a smart and better way than Flutter does. The user can insert the title, the description, a currency among all the ones provided by the FreeCurrencyAPI and a category among the six provided by the app. Then the user can insert, optionally, an email for a better tracking of the group and then insert the list of participants.

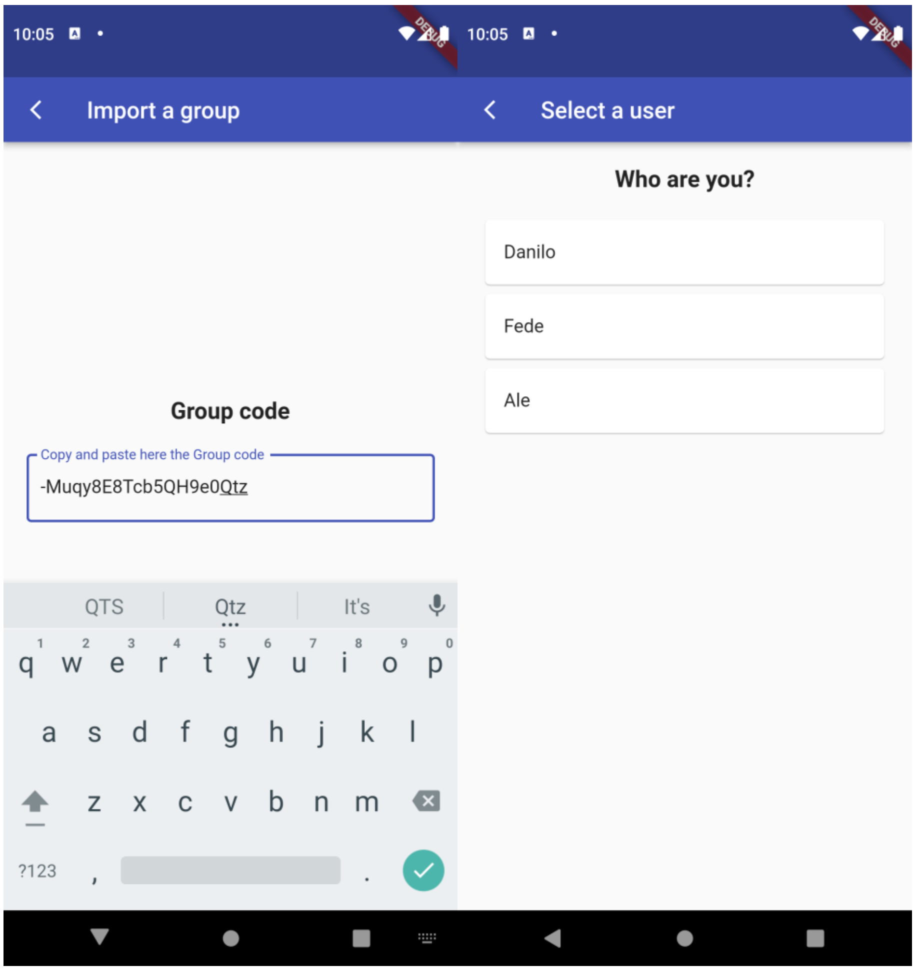
Once all data are inserted the user can tap on the Save button in the App Bar to send data to Firebase and save them in the database. Once it is done, he is redirected to the grouplist page.



Specify the currency that will be used to calculate the Group's balance. Expenses could be done in other currencies.

- **Import Group:**

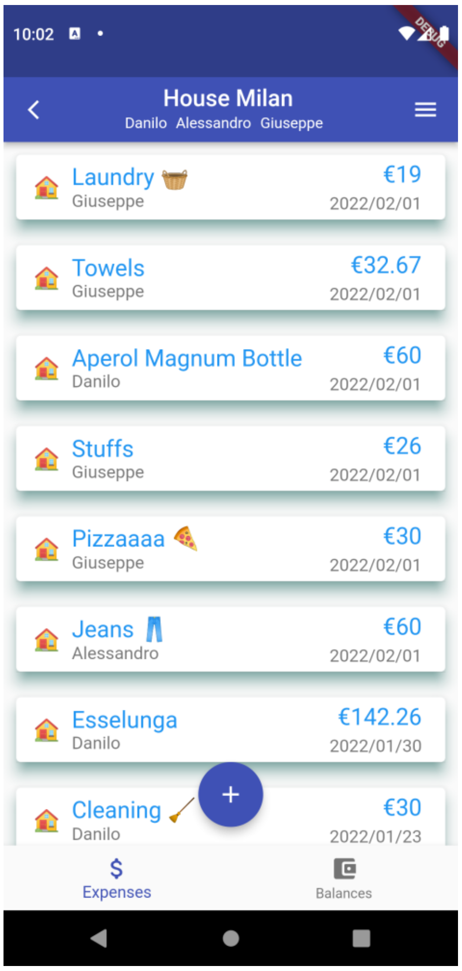
Clicking on "Import a group" opens the Import Group form. The user can enter the link or the ID of the Group, shared by a user of the group, normally the creator.



- **Expenses List:**

This page uses the FlutterFireUI library to retrieve from the DataBase all the expenses of the selected group and renders them in a scrollable list. The list is synchronized with the DB: when a user adds, modify or remove an expense, this is added, modified or removed from the list of all the other participants.

All the expenses implement a Swipe to delete function and are clickable to access the detail page of the selected expense.


10:02

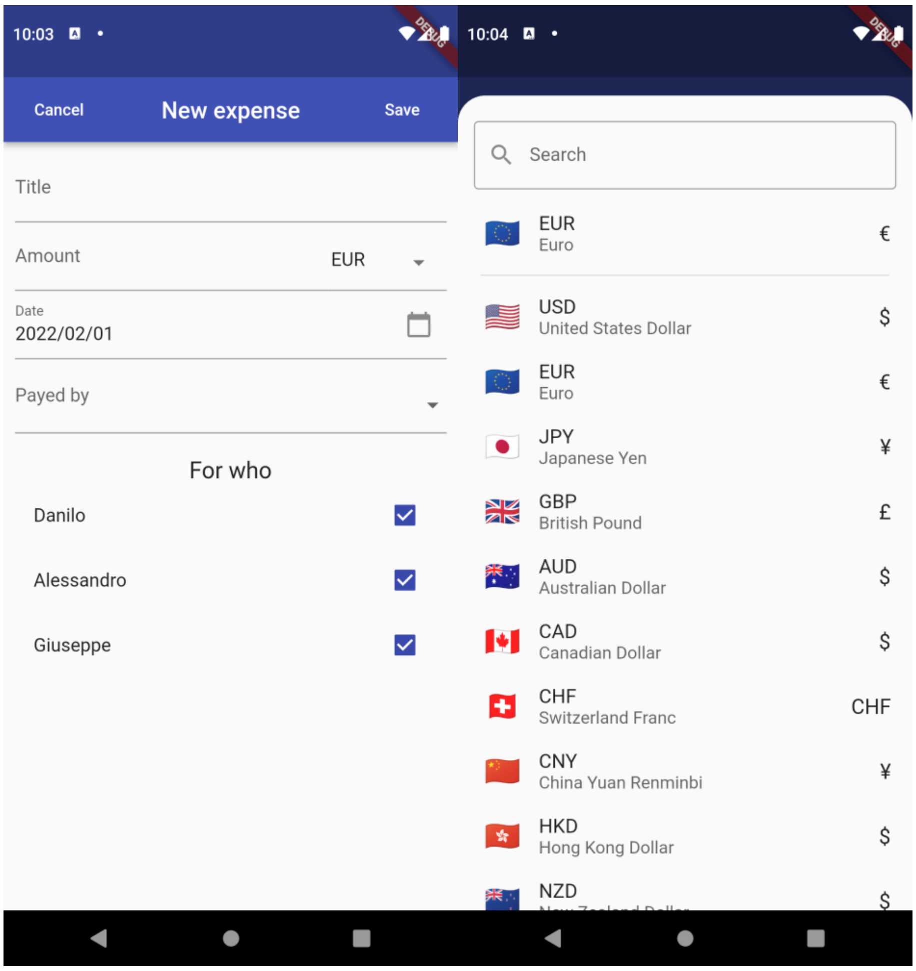
House Milan

Danilo Alessandro Giuseppe

Laundry 🧺	€19	
Giuseppe	2022/02/01	
Towels	€32.67	
Giuseppe	2022/02/01	
Aperol Magnum Bottle	€60	
Danilo	2022/02/01	
Stuffs	€26	
Giuseppe	2022/02/01	
Pizzaaaa 🍕	€30	
Giuseppe	2022/02/01	
Jeans 👖	€60	
Alessandro	2022/02/01	
Esselunga	€142.26	
Danilo	2022/01/30	
Cleaning 🧹	€30	
Danilo	2022/01/23	

- **Expense Creation Form:**

It's the form used to create a new expense; it's created using the Reactive Forms library. The user can insert the Title of the expense, the amount, the date, select who payed and for who. The user can also select the currency of the expense, this is achieved thanks to the CurrencyPicker library, that renders the list of currencies, and the FreeCurrencyAPI. The latter is reached by HTTP GET requests and return the converted amount.



Title

Amount EUR

Date
2022/02/01

Payed by

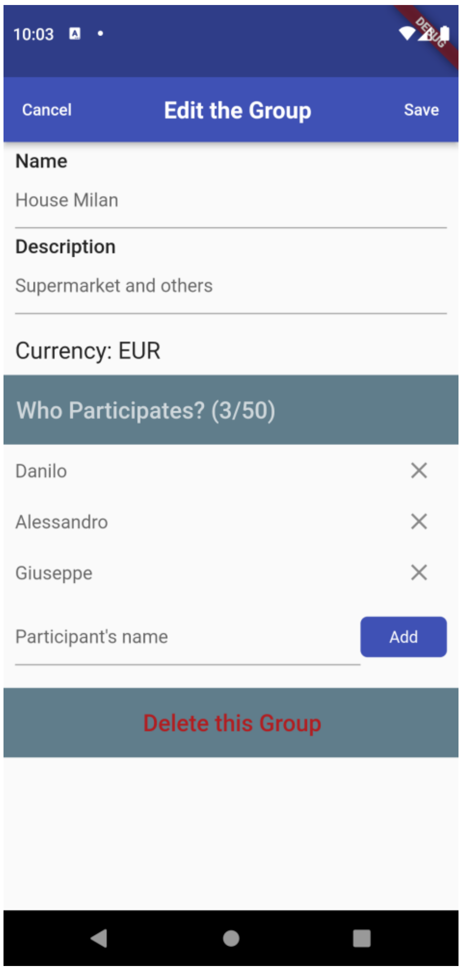
For who

Danilo	✓
Alessandro	✓
Giuseppe	✓

EUR	Euro	€
USD	United States Dollar	$
EUR	Euro	€
JPY	Japanese Yen	¥
GBP	British Pound	£
AUD	Australian Dollar	$
CAD	Canadian Dollar	$
CHF	Switzerland Franc	CHF
CNY	China Yuan Renminbi	¥
HKD	Hong Kong Dollar	$
NZD		$

- **Edit Group Form:**

You can access the group editing by tapping on its name in the Expenses List page or with the top right button in the same page (to give a more intuitive UI experience). In this page we use a form to let the user edit the group. He can modify the title, the description and the list of participants in the group in the same way as it was seen in the group creation page.



Name

House Milan

Description

Supermarket and others

Currency: EUR

Who Participates? (3/50)

Danilo

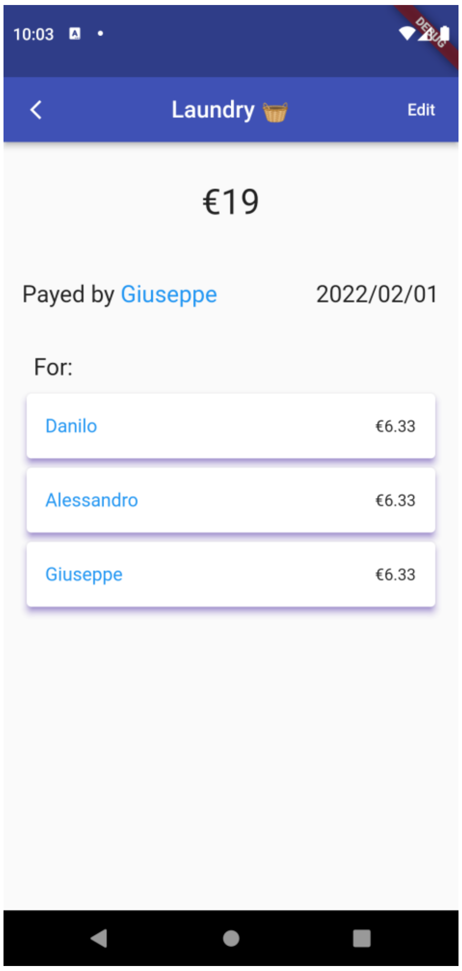
Alessandro

Giuseppe

Participant's name

- **Expense Detail:**

This page is reached by clicking on an expense from the expense list. From this page it’s possible to edit the expense and to view all the details.



€19

Payed by Giuseppe 2022/02/01

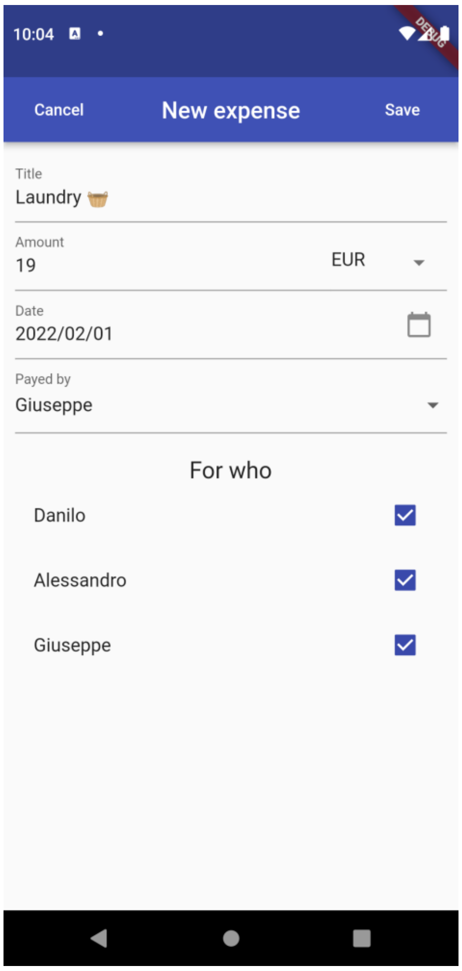
For:

Danilo	€6.33
Alessandro	€6.33
Giuseppe	€6.33

- **Edit Expense Form:**

Like the Expense Creation Form, this form allows the user to edit some or all the information of the chosen expense.

The Edit Expense Form is accessed by clicking on the Edit button in the top right corner of the Expense Detail Page.


10:04

Cancel **New expense** Save

Title
Laundry 🧺

Amount
19 EUR

Date
2022/02/01

Payed by
Giuseppe

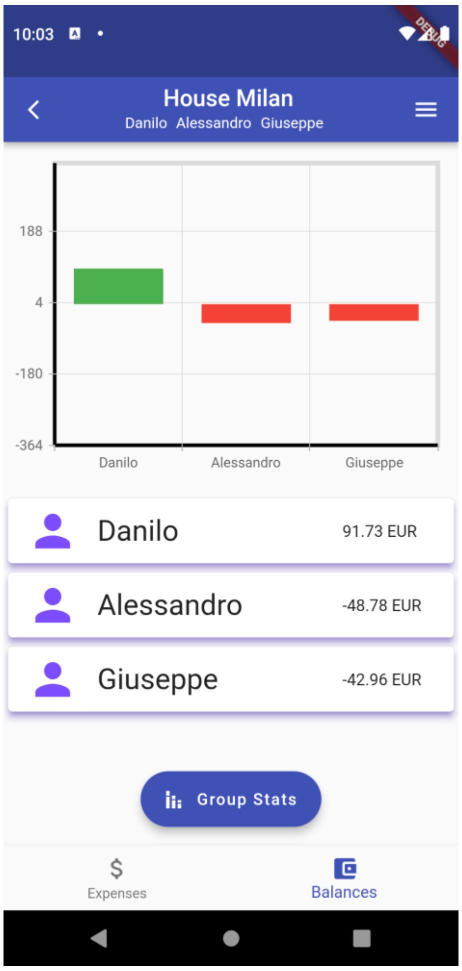
For who

Danilo	✓
Alessandro	✓
Giuseppe	✓

- **Balances Page:**

The balances page downloads the expenses of the group and analyses the balances of all the participants. In this page we can visualize a column chart SfColumnChart that compares the balances and display who have a positive one from the negatives. We use the red and green colours to distinguish the difference. Each time a user made an expense, the amount of it is added to his balance and the split of it is subtracted to all the users that participate to the expense.

A user with a positive balance has to be paid, the user with a negative balance need to pay someone.



Danilo 91.73 EUR

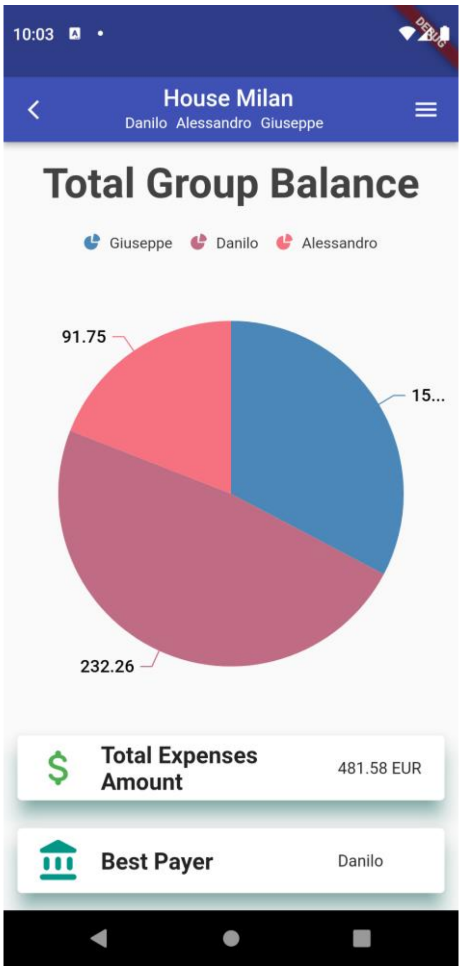
Alessandro -48.78 EUR

Giuseppe -42.96 EUR

- **Group Statistic:**

This page is a further analysis to the group data, here we use SfcircularChart, a cake chart, to display the total amount of the expenses and wo paid them.

We use two cards to display in a clear way the total amount and the best payer, the user that made the greatest buy in the group. The chart is animated and displays different colours to distinguish different users in the cake.



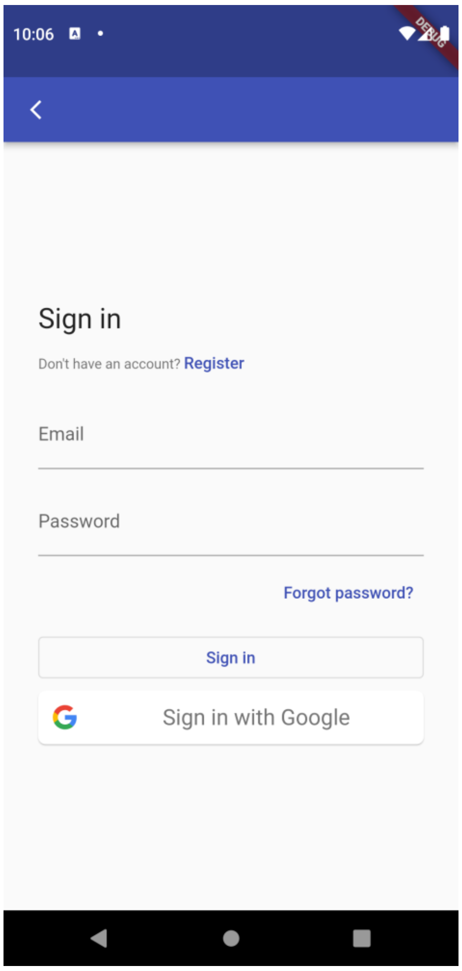
Total Group Balance

Total Expenses Amount 481.58 EUR

Best Payer Danilo

- **Login Page:**

We implemented the Registration Form, Login Form and LogOut button to manage the User state. We used the Firebase Auth Library. The password is obviously case sensitive. It must be long more than 8 characters and must include at least one letter and one number.

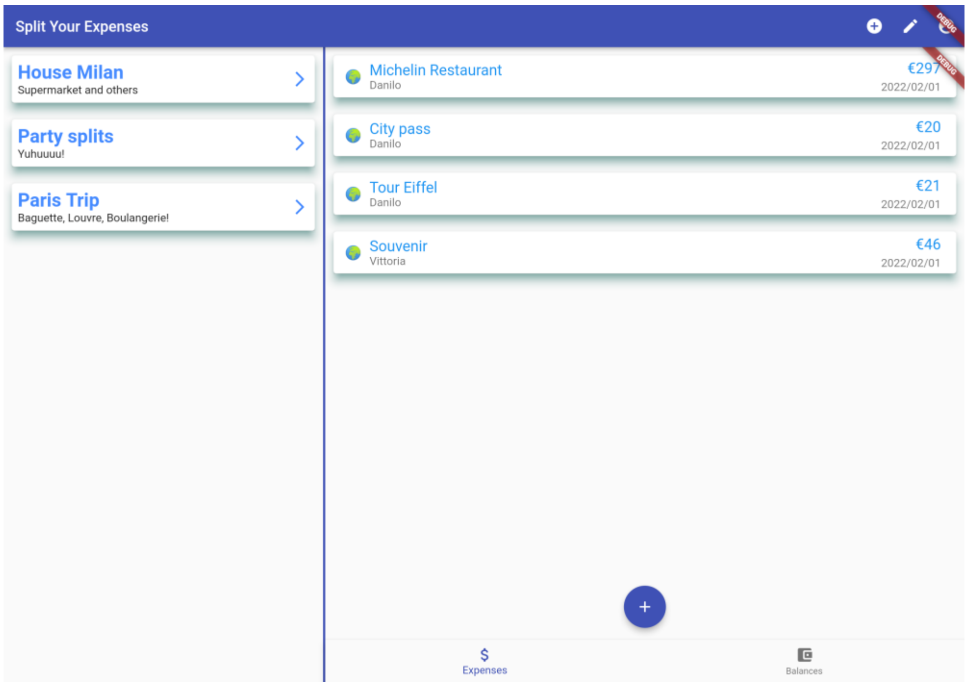


3.1.2 Tablet UI

Split Your Expenses automatically detect if the device is a smartphone or a tablet and if it is in Portrait mode or Landscape mode.

The tablet UI is different from the smartphone one because the device is bigger, and the space can be better used. We tried to reuse widgets as much as possible.

In the left side of the interface there is the list of all the Groups and on the right side the detail of the selected group.



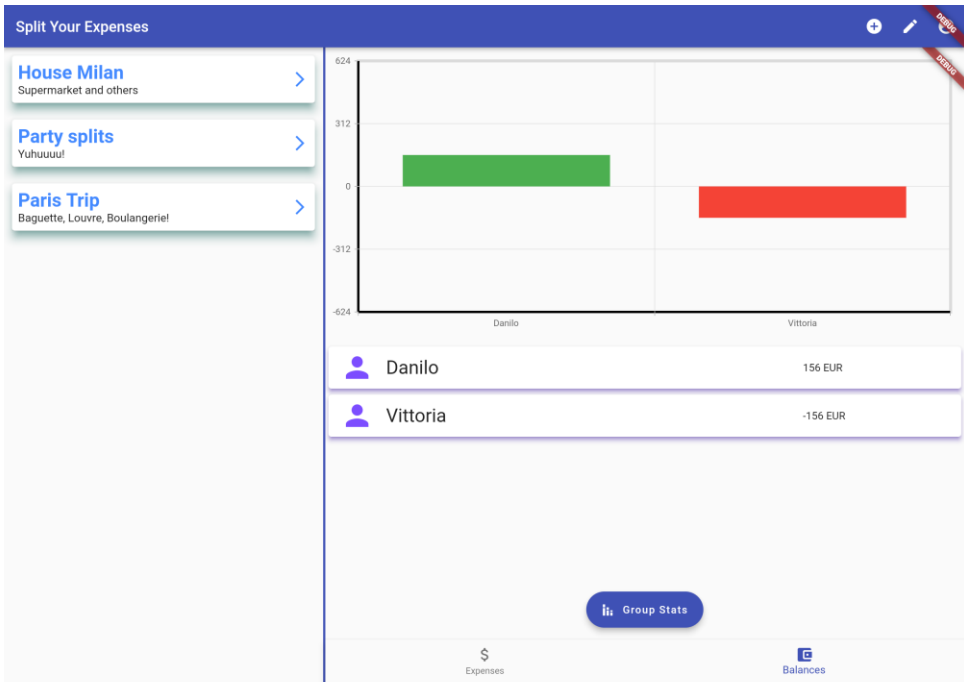
Split Your Expenses

House Milan
Supermarket and others

Party splits
Yuhuuuu!

Paris Trip
Baguette, Louvre, Boulangerie!

Michelin Restaurant	Danilo	€297
		2022/02/01
City pass	Danilo	€20
		2022/02/01
Tour Eiffel	Danilo	€21
		2022/02/01
Souvenir	Vittoria	€46
		2022/02/01



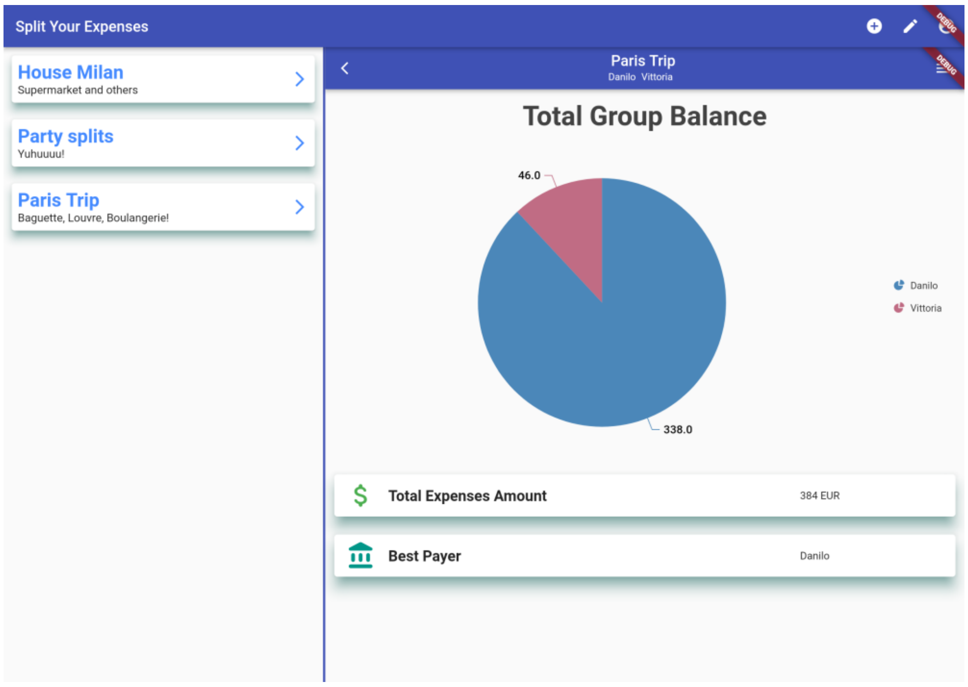
Split Your Expenses

House Milan
Supermarket and others

Party splits
Yuhuuuu!

Paris Trip
Baguette, Louvre, Boulangerie!

Danilo	156 EUR
Vittoria	-156 EUR


Split Your Expenses

House Milan
Supermarket and others

Party splits
Yuhuuuu!

Paris Trip
Baguette, Louvre, Boulangerie!

Paris Trip
Danilo Vittoria

Total Group Balance

Total Expenses Amount 384 EUR

Best Payer Danilo

Testing

Our testing campaigns made use of automated unit and widget tests. We made 50 automated tests using the Flutter Test environment. The test set covers almost all the classes and checks that all the widgets are properly visualized and work properly.

The strong presence of the DB in all the pages of our app makes hard Integration Testing so we decided to manually tests that all the parts of our app are correctly integrated and work correctly together.

Conclusions

The app can be downloaded on iOs and Android devices, both smartphones and tablets.

The goal of the app is to help people divide their expenses and to easily visualize them on graphs. Thanks to the real-time updates from the DB, the user experience is fluid and dynamic. The strengths of the app are its easy, elegant and smooth user interface. The app is very simple, so it can be used by people from all the ages. The graphs make easy to understand the sales balance to everyone.