



# FUNCTIONAL ANALYSIS

Kala, your incomes and expenses management app

## Short Description

Problem analysis, behaviour specification of the android app that is being developed and specification of the requirements that must do the software.

By KandV008

## 0. Table of Contents

1. Table of References.....	4
2. Table of Illustrations .....	4
3. Problem Analysis .....	5
4. Potential users .....	5
5. Data source.....	5
6. Entities.....	6
6.1. User .....	6
6.1.1. Traits .....	6
6.2. Month .....	6
6.2.1. Traits .....	6
6.3. Money Exchange.....	6
6.3.1. Traits .....	6
7. Requirements.....	7
7.1. Functional Requirements.....	7
7.1.1. Create Account.....	7
7.1.2. Delete Account.....	7
7.1.3. Log in .....	7
7.1.4. Recover password .....	7
7.1.5. Log out .....	8
7.1.6. Enter money exchange .....	8
7.1.7. Define type of the money exchange .....	8
7.1.8. Define scope of the money exchange.....	8
7.1.9. Read record of money exchange .....	9
7.1.10. Edit introduced money exchange .....	9
7.1.11. Delete introduced money exchange .....	9
7.1.12. Read summary of the introduced money exchange.....	9
7.1.13. Read mensual report.....	10
7.1.14. Access to Help page .....	10
7.1.15. Change language of the application .....	10
7.1.16. Show error login pop-up .....	10
7.1.17. Access to Home Page.....	10
7.1.18. Access to Configuration page.....	11
7.1.19. Define value of the exchange .....	12
7.1.20. Define description of money exchange .....	12
7.2. Future Functional Requirements .....	13
7.2.1. Change the currency.....	13

7.2.2. Send verification email.....	13
7.2.3. Edit Account.....	13
7.2.4. Show Terms and Conditions .....	13
7.2.5. Show Data Protection clause.....	14
7.3. Non Functional Requirements.....	15
7.3.1. Oriented to smarth phones.....	15
7.3.2. The app must run in Android 8 or upper .....	15
7.3.3. The software must be implemented with Kotlin .....	15
7.3.4. The application needs to connect with a cloud data base .....	15
7.3.5. Languages Required .....	15
7.3.6. GUI must be minimalist and user-friendly .....	16
7.3.7. Needs an-email system.....	16
7.3.8. Protection and Security for Registered User Data.....	16
7.3.9. Use of Jetpack Compose .....	16
8. Use Case Diagrams .....	17
8.1. Actor definition .....	17
8.1.1. Unregistered User.....	17
8.1.2. Registered User .....	17
8.2. Access System.....	17
8.3. Home Page .....	18
8.4. Configuration Page.....	18
8.5. CRUD of Money Exchange .....	18
9. Class Diagram .....	20
9.1. Classes.....	20
9.1.1. User .....	20
9.1.2. MonthInformation .....	20
9.1.3. MoneyExchange.....	20
9.1.4. MoneyExchangeType.....	21
9.1.5. MoneyExchangeScope .....	21
9.1.6. Date .....	21

## 1. Table of References

Table 1. FR-1 Create Account	7
Table 2. FR-2 Delete Account	7
Table 3. FR-3 Log in	7
Table 4. FR-4 Recover password	7
Table 5. FR-5 Log out	8
Table 6. FR-6 Enter money exchange	8
Table 7. FR-7 Define type of the money exchange	8
Table 8. FR-8 Define scope of the money exchange	8
Table 9. FR-9 Read record of money exchange	9
Table 10. FR-10 Edit introduced money exchange	9
Table 11. FR-11 Delete introduced money exchange	9
Table 12. FR-12 Read summary of the introduced money exchange	9
Table 13. FR-13 Read mensual report	10
Table 14. FR-15 Access to Help Page	10
Table 15. FR-17 Change language of the application	10
Table 16. FR-18 Show error login pop-up	10
Table 17. FR-19 Access to Home Page	11
Table 18. FR-20 Access to Configuration Page	11
Table 19. FR-24 Define value of the exchange	12
Table 20. FR-25 Define description of money exchange	12
Table 21. FR-14 Change the currency	13
Table 22. FR-16 Send verification email	13
Table 23. FR-21 Edit Account	13
Table 24. FR-22 Show Terms and Conditions	14
Table 25. FR-23 Show Data Protection clause	14
Table 26. NFR-1 Oriented to smarth phones	15
Table 27. NFR-2 The app must run in Android 5 or upper	15
Table 28. NFR-3 The software must be implemented with Kotlin	15
Table 29. NFR-4 The application needs to connect with a cloud data base	15
Table 30. NFR-5 Languages Required	16
Table 31. NFR-6 GUI must be minimalist and user-friendly	16
Table 32. NFR-7 Needs an-email system	16
Table 33. NFR-8 Protection and Security for Registered User Data	16
Table 34. NFR-9 Use of Jetpack Compose	16

## 2. Table of Ilustrations

Use Case Diagram 1. Access system .....	17
Use Case Diagram 2. Home Page .....	18
Use Case Diagram 3. Configuration Page .....	18
Use Case Diagram 4. CRUD of money exchange .....	19
 Class Diagram 1. Main.....	 20

### 3. Problem Analysis

The client, who is me, need an application to control all the expenses and incomes during a month, including cash and deposit money.

The users can create an account to follow all the exchanges of money using their smartphone, having a fast and simple access to the application.

The idea is to introduce the expense or the income and after that, see a summary of the exchange of money along the month.

### 4. Potential users

The android application focusses mainly to adults, because is the age when the people start to concert about their money.

The app will have two types of users, that will be specificized afterwards (see [section 6.1](#)).

### 5. Data source

The users will be created during the software life cycle. There will not be default users nor other data.

The system will keep the information in a cloud database. This is since the registered user needs access to his information in different smartphones.

## 6. Entities

All entities must be persistent.

All entities will have common traits, and some of them will have subentities with special traits.

### 6.1. User

#### 6.1.1. Traits

- Identifier: Number
- Email: String of characters
- Set of Months

### 6.2. Month

#### 6.2.1. Traits

- Identifier: String of character (Month's name with the year)
- Set of Money Exchange
- Expensed money: Number
- Incomed money: Number

### 6.3. Money Exchange

#### 6.3.1. Traits

- Identifier: Number
- Money Value: Number
- Type of money exchange: Expense or Income
- Scope of the money: Food, Leisure, Useful, Medicine u others
- Description: String of character
- Date: DD/MM/YYYY (DD → Day; MM → Month; YYYY → Year)

## 7. Requirements

### 7.1. Functional Requirements

#### 7.1.1. Create Account

<b>Name</b>	Create Account
<b>Id</b>	FR-1
<b>Type</b>	Required
<b>Description</b>	The unregistered user can create an account, and must introduce the name, the email and a password.
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

Table 1. FR-1 Create Account

#### 7.1.2. Delete Account

<b>Name</b>	Delete Account
<b>Id</b>	FR-2
<b>Type</b>	Required
<b>Description</b>	The registered user can delete his account
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

Table 2. FR-2 Delete Account

#### 7.1.3. Log in

<b>Name</b>	Log in
<b>Id</b>	FR-3
<b>Type</b>	Required
<b>Description</b>	The registered user can access to the system using his name or email and password
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

Table 3. FR-3 Log in

#### 7.1.4. Recover password

<b>Name</b>	Recover password
<b>Id</b>	FR-4
<b>Type</b>	Optional
<b>Description</b>	The registered user can recover his password using their email to send an email
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

Table 4. FR-4 Recover password

#### 7.1.5. Log out

<b>Name</b>	Log out
<b>Id</b>	FR-5
<b>Type</b>	Required
<b>Description</b>	The registered user can log out of the system
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

*Table 5. FR-5 Log out*

#### 7.1.6. Enter money exchange

<b>Name</b>	Enter money exchange
<b>Id</b>	FR-6
<b>Type</b>	Required
<b>Description</b>	The registered user can enter information about the money exchange
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

*Table 6. FR-6 Enter money exchange*

#### 7.1.7. Define type of the money exchange

<b>Name</b>	Define type of the money exchange
<b>Id</b>	FR-7
<b>Type</b>	Required
<b>Description</b>	Now to enter the money exchange, the registered user must define if it is an expense or an income
<b>Requirements</b>	Access to the persistent system
<b>Dangers</b>	Low

*Table 7. FR-7 Define type of the money exchange*

#### 7.1.8. Define scope of the money exchange

<b>Name</b>	Define scope of the money exchange
<b>Id</b>	FR-8
<b>Type</b>	Required
<b>Description</b>	Now to enter the money exchange, the registered user must define the scope, that can be food, leisure, useful, medication or other. (Open to increment or decresase the scopes)
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

*Table 8. FR-8 Define scope of the money exchange*



#### 7.1.9. Read record of money exchange

<b>Name</b>	Read record of money exchange
<b>Id</b>	FR-9
<b>Type</b>	Required
<b>Description</b>	The registered user can access to a record of the previous money exchange introduced
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

Table 9. FR-9 Read record of money exchange

#### 7.1.10. Edit introduced money exchange

<b>Name</b>	Edit introduced money exchange
<b>Id</b>	FR-10
<b>Type</b>	Required
<b>Description</b>	The registered user can edit previous money exchange
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

Table 10. FR-10 Edit introduced money exchange

#### 7.1.11. Delete introduced money exchange

<b>Name</b>	Delete introduced money exchange
<b>Id</b>	FR-11
<b>Type</b>	Required
<b>Description</b>	The registered user can delete a previous money exchange
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

Table 11. FR-11 Delete introduced money exchange

#### 7.1.12. Read summary of the introduced money exchange

<b>Name</b>	Read summary of the introduced money exchange
<b>Id</b>	FR-12
<b>Type</b>	Required
<b>Description</b>	When the registered user enters to the system, need to see a summary of the expense and income money along the month
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

Table 12. FR-12 Read summary of the introduced money exchange

#### 7.1.13. Read mensual report

<b>Name</b>	Read mensual report
<b>Id</b>	FR-13
<b>Type</b>	Required
<b>Description</b>	The registered user can see a mensual report with the expense and income of the different scopes
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

Table 13. FR-13 Read mensual report

#### 7.1.14. Access to Help page

<b>Name</b>	Access to Help page
<b>Id</b>	FR-15
<b>Type</b>	Required
<b>Description</b>	The user can access to this page to see information about the application, some help to use the software and understand the functionality
<b>Requirements</b>	Access to the text file
<b>Danger</b>	Low

Table 14. FR-15 Access to Help Page

#### 7.1.15. Change language of the application

<b>Name</b>	Change language of the application
<b>Id</b>	FR-17
<b>Type</b>	Optional
<b>Description</b>	The unregistered and registered user can change the software language
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Medium

Table 15. FR-17 Change language of the application

#### 7.1.16. Show error login pop-up

<b>Name</b>	Show error login pop-up
<b>Id</b>	FR-18
<b>Type</b>	Required
<b>Description</b>	When the registered user sends incorrect credentials, the application will warn the error
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

Table 16. FR-18 Show error login pop-up

#### 7.1.17. Access to Home Page

<b>Name</b>	Access to Home Page
<b>Id</b>	FR-19
<b>Type</b>	Required
<b>Description</b>	After the registered user send the correct credentials, the registered user will access the home page

<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

*Table 17. FR-19 Access to Home Page*

#### 7.1.18. Access to Configuration page

<b>Name</b>	Access to Configuration page
<b>Id</b>	FR-20
<b>Type</b>	Required
<b>Description</b>	The registered user can enter to configuration to edit some values of the application
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

*Table 18. FR-20 Access to Configuration Page*

## 7.1.19. Define value of the exchange

<b>Name</b>	Define value of the exchange
<b>Id</b>	FR-24
<b>Type</b>	Required
<b>Description</b>	Now to enter the money exchange, the registered user must define the value of the exchange.
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

*Table 19. FR-24 Define value of the exchange*

## 7.1.20. Define description of money exchange

<b>Name</b>	Define description of money exchange
<b>Id</b>	FR-25
<b>Type</b>	Optional
<b>Description</b>	Now to enter the money exchange, the registered user can add a description of the exchange.
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Low

*Table 20. FR-25 Define description of money exchange*

## 7.2. Future Functional Requirements

During the development, some functional requirements changed priority for various reasons.

The following requirements will not be implemented in version 1.0 but will be added during the different updates.

The documents compiling the updates will be independent of the artifacts presented in this file.

### 7.2.1. Change the currency

<b>Name</b>	Change the currency
<b>Id</b>	FR-14
<b>Type</b>	Improve
<b>Description</b>	<p>The application will let the registered user to change the currency. Default value will be €, “euro”; but you can change it to \$, “dolar”.</p> <p>This would be a visual change, not a currency conversion. The following may be added in the future.</p>
<b>Requirements</b>	
<b>Danger</b>	Low

Table 21. FR-14 Change the currency

### 7.2.2. Send verification email

<b>Name</b>	Send verification email
<b>Id</b>	FR-16
<b>Type</b>	Improve
<b>Description</b>	When the unregistered user creates an account, the app must send a verification email
<b>Requirements</b>	Email system
<b>Danger</b>	Medium

Table 22. FR-16 Send verification email

### 7.2.3. Edit Account

<b>Name</b>	Edit Account
<b>Id</b>	FR-21
<b>Type</b>	Improve
<b>Description</b>	The registered user can change his account information, as the name or the email
<b>Requirements</b>	Access to the persistent system
<b>Danger</b>	Medium

Table 23. FR-21 Edit Account

### 7.2.4. Show Terms and Conditions

<b>Name</b>	Show Terms and Conditions
<b>Id</b>	FR-22
<b>Type</b>	Improve

<b>Description</b>	The application will show up a pop-up with the terms and conditions of the software when the registered user log in for the first time.  It is necessary to accept to continue using the program
<b>Requirements</b>	
<b>Danger</b>	Low

Table 24. FR-22 Show Terms and Conditions

#### 7.2.5. Show Data Protection clause

<b>Name</b>	Show Data Protection clause
<b>Id</b>	FR-23
<b>Type</b>	Improve
<b>Description</b>	The application will show up a pop-up with the Data Protection clause of the software when the registered user had accepted the Terms and Conditions.  It is necessary to accept to continue using the program
<b>Requirements</b>	
<b>Danger</b>	Low

Table 25. FR-23 Show Data Protection clause

### 7.3. Non Functional Requirements

#### 7.3.1. Oriented to smarth phones

<b>Name</b>	Oriented to smarth phones
<b>Id</b>	NFR-1
<b>Type</b>	Required
<b>Description</b>	The application is focused to smath phone users, because the objective of the application is an easy access to the data about the money exchange of the registered user

*Table 26. NFR-1 Oriented to smarth phones*

#### 7.3.2. The app must run in Android 8 or upper

<b>Name</b>	The app must run in Android 8 or upper
<b>Id</b>	NFR-2
<b>Type</b>	Required
<b>Description</b>	For the application to be used by the largest number of Android users, it must be in older version of Android

*Table 27. NFR-2 The app must run in Android 5 or upper*

#### 7.3.3. The software must be implemented with Kotlin

<b>Name</b>	The software must be implemented with Kotlin
<b>Id</b>	NFR-3
<b>Type</b>	Required
<b>Description</b>	Kotlin is, currently, the programming language more adequate to create and Android software due to the support it has in this platform

*Table 28. NFR-3 The software must be implemented with Kotlin*

#### 7.3.4. The application needs to connect with a cloud data base

<b>Name</b>	The application needs to connect with a cloud data base
<b>Id</b>	NFR-4
<b>Type</b>	Required
<b>Description</b>	To keep the data introduced by the user, this information must be saved in some cloud data base, to be possible to access easily with another smarth phone with the application

*Table 29. NFR-4 The application needs to connect with a cloud data base*

#### 7.3.5. Languages Required

<b>Name</b>	Languages Required
<b>Id</b>	NFR-5
<b>Type</b>	Required
<b>Description</b>	The mainly languages available are Spanish and English.  It is possible to extends the catalogue of languages available

Table 30. NFR-5 Languages Required

#### 7.3.6. GUI must be minimalist and user-friendly

<b>Name</b>	GUI must be minimalist and user-friendly
<b>Id</b>	NFR-6
<b>Type</b>	Required
<b>Description</b>	The GUI must be minimalist to be simple and intuitive.  To be user-friendly, all type of people, can use this application without any physic or mentally limitation

Table 31. NFR-6 GUI must be minimalist and user-friendly

#### 7.3.7. Needs an-email system

<b>Name</b>	Needs an-email system
<b>Id</b>	NFR-7
<b>Type</b>	Required
<b>Description</b>	To send any notification to the registered user, is necessary an-email system

Table 32. NFR-7 Needs an-email system

#### 7.3.8. Protection and Security for Registered User Data

<b>Name</b>	Protection and Security for Registered User Data
<b>Id</b>	NFR-8
<b>Type</b>	Required
<b>Description</b>	All the data introduced by the registered user will be secure and protected

Table 33. NFR-8 Protection and Security for Registered User Data

#### 7.3.9. Use of Jetpack Compose

<b>Name</b>	Use of Jetpack Compose
<b>Id</b>	NFR-9
<b>Type</b>	Required
<b>Description</b>	This technology will be used to create the user interface.

Table 34. NFR-9 Use of Jetpack Compose



## 8. Use Case Diagrams

The next Use Cases Diagrams portray the different requirements and their relations with actors.

All the functional requirements have associated a use case with their id (see Section 5.1).

All the actors are briefly described at the Section 6.1 of this document.

### 8.1. Actor definition

#### 8.1.1. Unregistered User

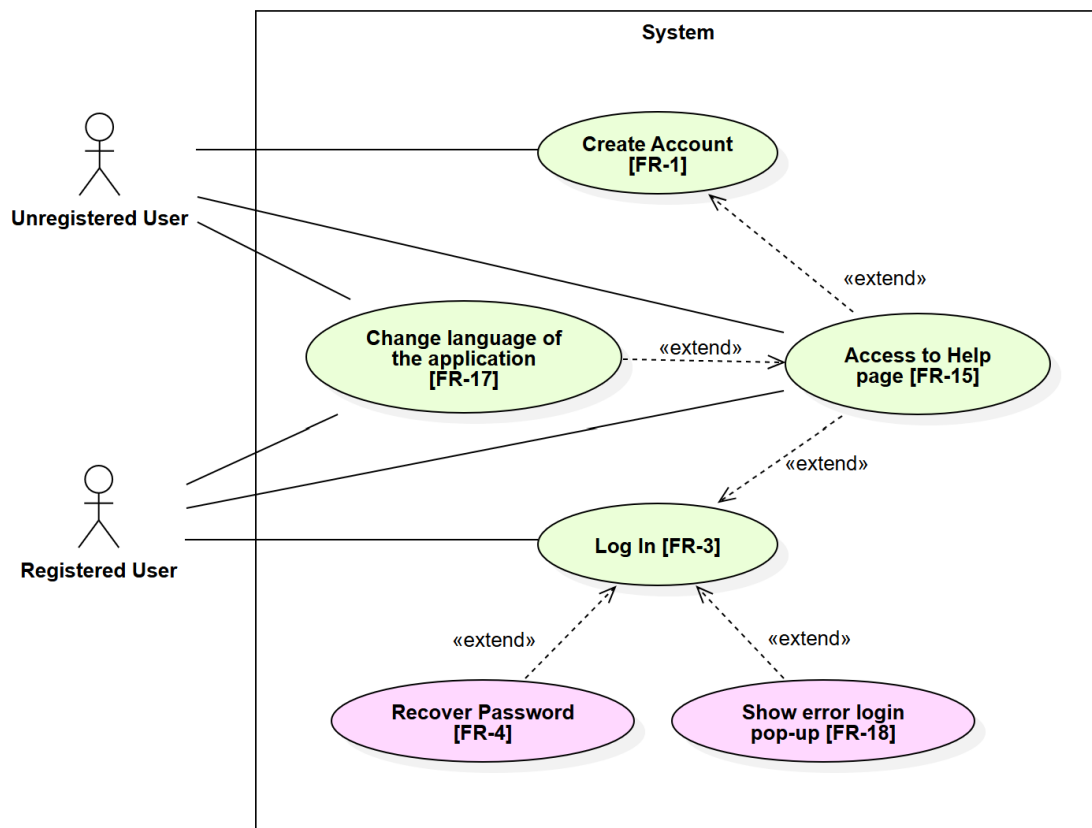
This type of user only can create an account and change the language of the application.

#### 8.1.2. Registered User

This type of user can use all the functionality of the software without any limitation. Before that, they need to log in.

### 8.2. Access System

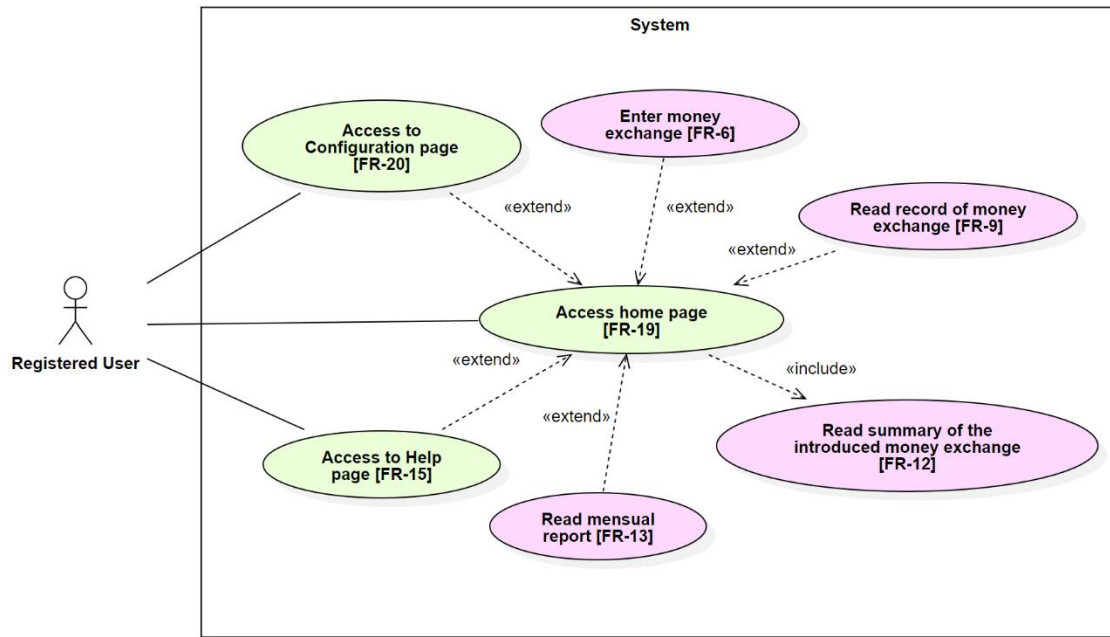
This use case diagram represents the relation with the system of the two different users, being mainly about the access system.



Use Case Diagram 1. Access system

### 8.3. Home Page

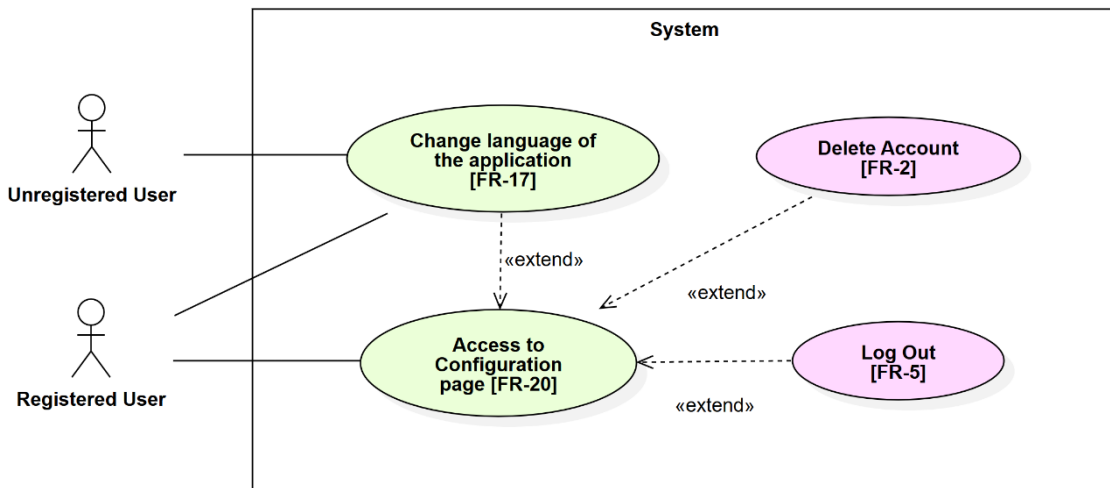
The next use case diagram shows the functionality of the home page of Kala, with the principal action, FR-6: Enter money exchange, the core of the application.



Use Case Diagram 2. Home Page

### 8.4. Configuration Page

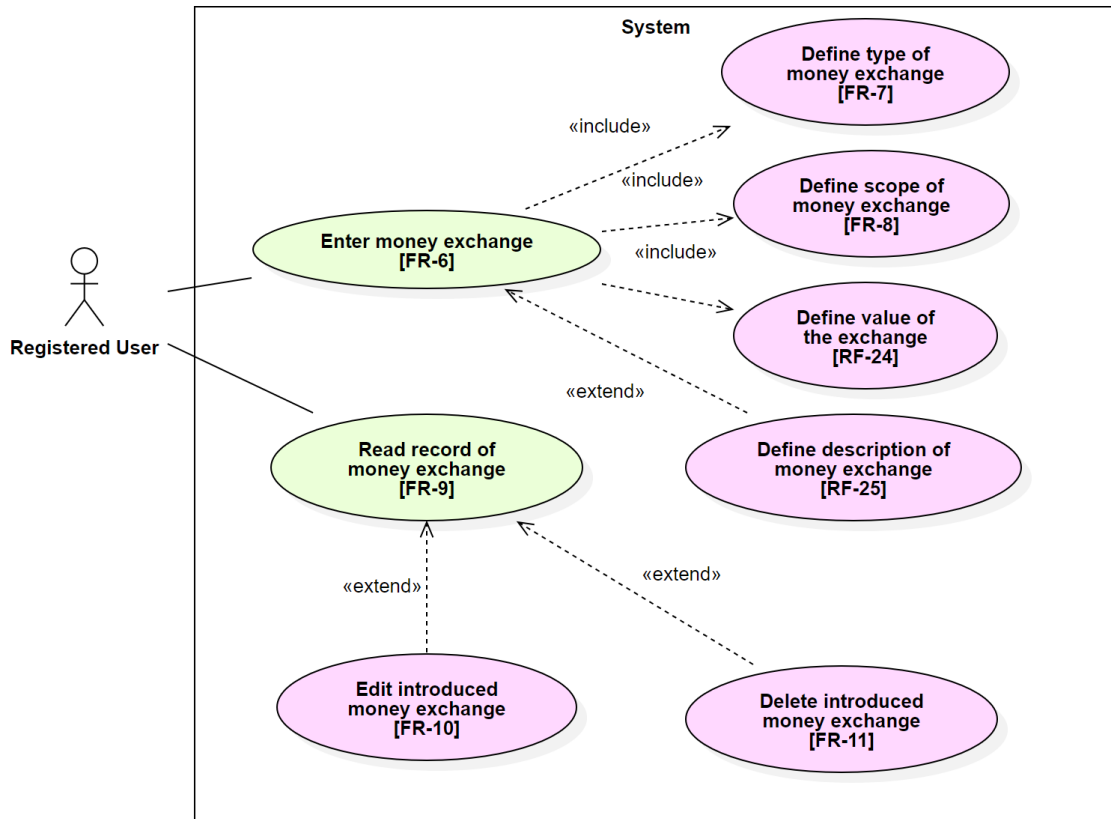
The following use case diagram displays all the actions that registered user, can do in the configuration page.



Use Case Diagram 3. Configuration Page

### 8.5. CRUD of Money Exchange

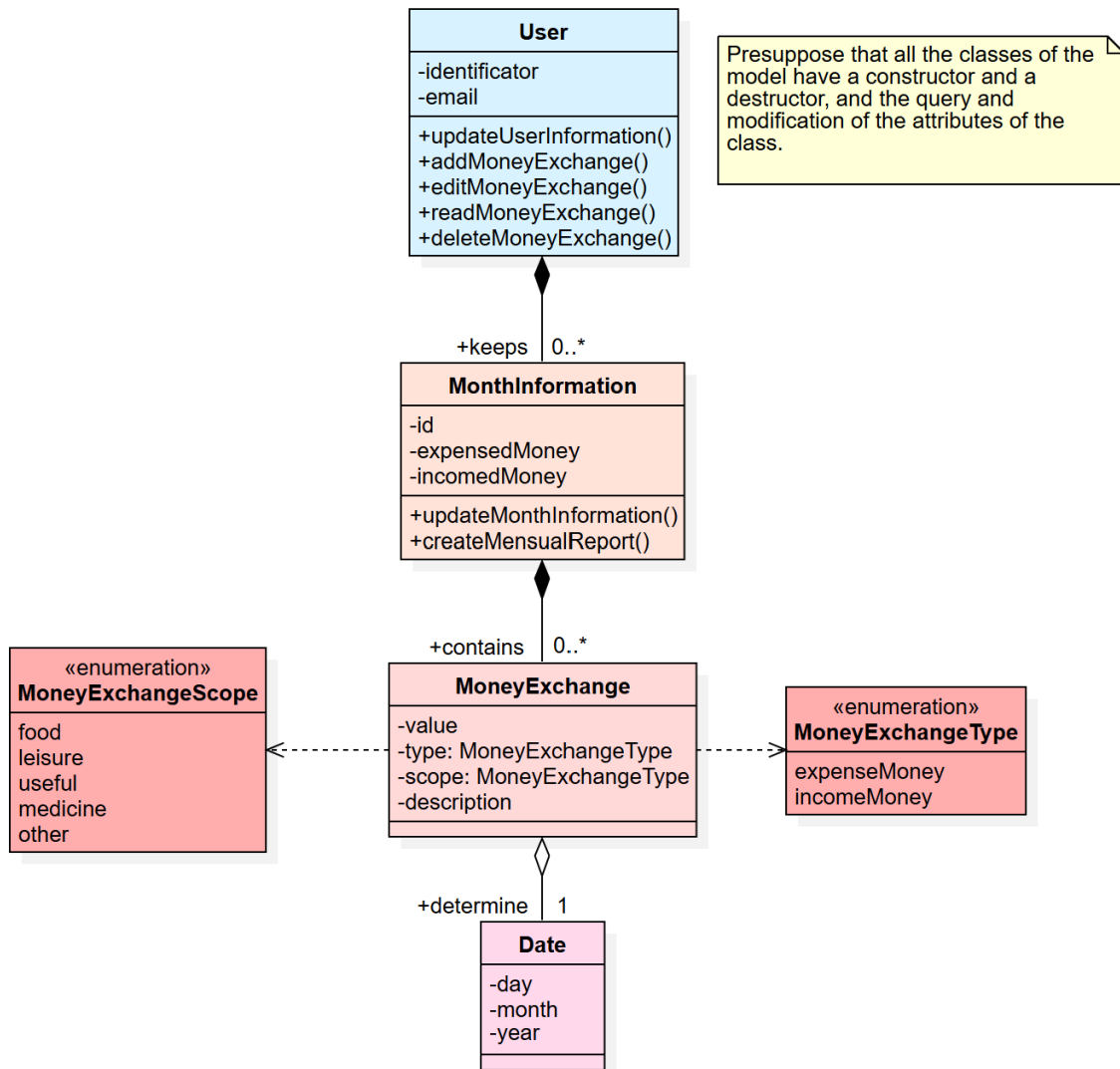
This use case diagram explains the functionality of the CRUD system of the software, and how the registered user is related to it.



Use Case Diagram 4. CRUD of money exchange

## 9. Class Diagram

This class diagram represents the body of the android application, and how everything is related among them.



Class Diagram 1. Main

### 9.1. Classes

#### 9.1.1. User

The User class will gather all the MonthInformation with their corresponding MoneyExchange. It will keep the name and email address, with a unique ID.

#### 9.1.2. MonthInformation

The MonthInformation class will have all the money exchange along the month, with the mensual report.

#### 9.1.3. MoneyExchange

The MoneyExchange class will storage all the information about one money exchange.



#### 9.1.4. MoneyExchangeType

The MoneyExchangeType enum will determine the only two types of money exchange: expensed or income money, to classify them.

#### 9.1.5. MoneyExchangeScope

The MoneyExchangeScope enum will decide the scopes of money exchange. It is possible to extend in the future the scopes, but currently they are the next ones: Food, Leisure, Useful, Medicine and Others.

#### 9.1.6. Date

The Date class will allow the organization of months of the application.