

Use Case Diagram And Use Cases

Summary of the Use Cases:

Use Case Diagram

Use Case 1, UC1 - Transactions

Scenario 1.1 (Nominal): create new transaction

Scenario 1.2 (Nominal): delete transaction

Scenario 1.3 (Nominal): display transactions

Scenario 1.4 (Nominal): display labels

Scenario 1.5 (Nominal): get all transactions by type

Scenario 1.6 (Nominal): sort transactions by date

Scenario 1.7 (Nominal): get transactions by day

Scenario 1.8 (Nominal): modify a transaction

Scenario 1.9 (Exception): unauthorized creation

Scenario 1.10 (Exception): unauthorized removal

Scenario 1.11 (Exception): unauthorized display

Scenario 1.12 (Exception): transaction to modify not found

Use Case 2, UC2 - Categories

Scenario 2.1 (Nominal): create new category

Scenario 2.2 (Nominal): display categories

Scenario 2.3 (Nominal): edit a category

Scenario 2.4 (Nominal): get category by type

Scenario 2.5 (Nominal): notify when category exceeds limit

Scenario 2.6 (Nominal): update a category's limit

Scenario 2.7 (Exception): unauthorized in creation

Scenario 2.8 (Exception): unauthorized in display

Scenario 2.9 (Exception): category to modify not found

Scenario 2.10 (Exception): user doesn't have a premium account

Use Case 3, UC3 - Sign up

Scenario 3.1 (Nominal): sign up

Scenario 3.2 (Exception): email already used

Scenario 3.3 (Exception): missing parameters

Scenario 3.4 (Exception): password not valid

Scenario 3.5 (Exception): passwords don't match

Use Case 4, UC4 - Authentication

Scenario 4.1 (Nominal): login

Scenario 4.2 (Nominal): forgotten password

Scenario 4.3 (Exception): account not found

Scenario 4.4 (Exception): wrong credentials

Scenario 4.5 (Exception): forgotten password, email not associated to an account

Use Case 5, UC5 - Users

Scenario 5.1 (Nominal): display all users

Scenario 5.2 (Nominal): display user profile by username

[Scenario 5.3 \(Nominal\): switching from free to premium account](#)

[Scenario 5.4 \(Nominal\): change password](#)

[Scenario 5.5 \(Exception\): unauthorized display all users](#)

[Scenario 5.6 \(Exception\): unauthorized display user profile](#)

[Scenario 5.7 \(Exception\): unsuccessful payment](#)

[Scenario 5.8 \(Exception\): old password incorrect](#)

[Scenario 5.9 \(Exception\): new password not valid](#)

[Use Case 6, UC6 - Logout](#)

[Scenario 6.1 \(Nominal\): logout](#)

[Scenario 6.2 \(Exception\): account not found](#)

[Use Case 7, UC7 - Analytics](#)

[Scenario 7.1 \(Nominal\): get total expenditure by month](#)

[Scenario 7.2 \(Nominal\): get total expenditure by category](#)

[Scenario 7.3 \(Nominal\): get total expenditure by category and month](#)

[Scenario 7.4 \(Nominal\): get total expenditure by group](#)

[Scenario 7.5 \(Nominal\): get total expenditure by group and month](#)

[Scenario 7.6 \(Nominal\): get total expenditure by group and category](#)

[Scenario 7.7 \(Nominal\): get total expenditure by group, month and category](#)

[Scenario 7.8 \(Exception\): user unauthorized](#)

[Use Case 8, UC8 - Groups](#)

[Scenario 8.1 \(Nominal\): create a group](#)

[Scenario 8.2 \(Nominal\): add user to a group](#)

[Scenario 8.3 \(Nominal\): add transaction to a group](#)

[Scenario 8.4 \(Nominal\): add a category to a group](#)

[Scenario 8.5 \(Nominal\): edit a group](#)

[Scenario 8.6 \(Nominal\): remove user from a group](#)

[Scenario 8.7 \(Nominal\): delete transaction from a group](#)

[Scenario 8.8 \(Nominal\): delete category from a group](#)

[Scenario 8.9 \(Nominal\): display all categories of a group](#)

[Scenario 8.10 \(Nominal\): display all transactions of a group](#)

[Scenario 8.11 \(Nominal\): modify a transaction/category](#)

[Scenario 8.12 \(Exception\): category/transaction not present](#)

[Scenario 8.13 \(Exception\): user unauthorized](#)

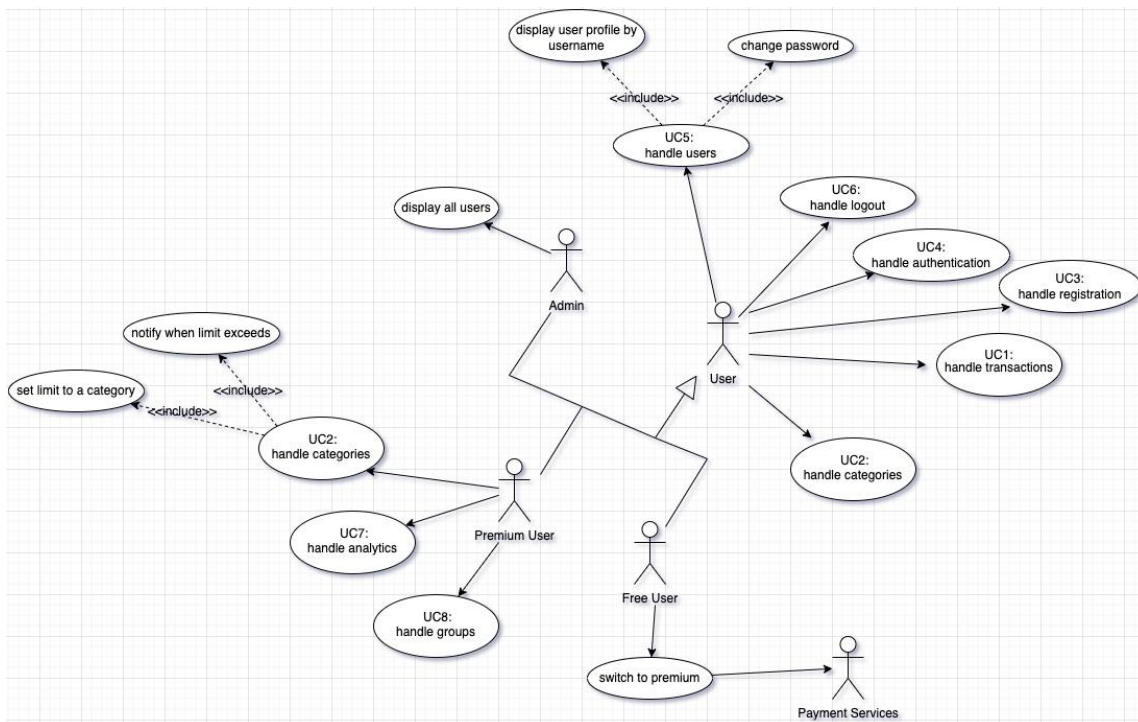
[Scenario 8.14 \(Exception\): user to add has not premium account](#)

[Scenario 8.15 \(Exception\): user to add not found](#)

[Scenario 8.16 \(Exception\): user has not a premium account](#)

[Scenario 8.17 \(Exception\): user to remove not found](#)

Use Case Diagram



Use Case 1, UC1 - Transactions

handle transactions

Actors Involved	User
Precondition	user must be logged in
Post condition	
Nominal Scenarios	create new transaction, remove transaction, display transactions, display labels, get transactions by type, sort transactions by date, modify transaction
Variants	
Exceptions	unauthorized creation, unauthorized removal, unauthorized display, transaction to modify not found

Scenario 1.1 (Nominal): create new transaction

The user, after making an expenditure on a specific date, wants to insert the amount spent inside the application in a specific category with a small description of the shopping made.

The scenario ends correctly, the transaction is stored in the system.

Step	Description
Precondition	user logged in
Post condition	transaction stored in the system

1	user asks to create a new transaction
2	system checks access token and its validity
3	system asks name, type of expense and amount spent
4	user inserts name, type and amount
5	system checks parameters and stores the transaction in the system

Scenario 1.2 (Nominal): delete transaction

The user, after inserting a new transaction, wants to delete the latter for some reason in order to remove that expenditure from the total of expenses.

The scenario ends correctly, the transaction is removed.

Step	Description
Precondition	user logged in & transaction must be already stored
Post condition	transaction removed from the system
1	user asks to remove a stored transaction
2	system checks access token and its validity
3	system asks id transaction
4	user inserts id transaction to remove
5	system delete transaction with id inserted

Scenario 1.3 (Nominal): display transactions

The user wants to visualize all the expenses recorded in the application.

The scenario ends correctly, all the transactions are displayed.

Step	Description
Precondition	user logged in
Post condition	all transactions stored are displayed
1	user asks to display all the transactions stored in the system
2	system checks access token and its validity
3	system displays all the transactions stored in the system

Scenario 1.4 (Nominal): display labels

The user wants to visualize all the labels of the categories and the transactions previously inserted.

The scenario ends correctly, all the labels are displayed.

Step	Description
Precondition	user logged in
Post condition	all labels stored are displayed

1	user asks to display all the labels, common to the categories and to the transactions stored
2	system checks access token and its validity
3	system displays all the labels

Scenario 1.5 (Nominal): get all transactions by type

The user wants to visualize all the transactions stored with a type the user has to specify.

The scenario ends correctly, all the transactions are displayed.

Step	Description
Precondition	user logged in
Post condition	all transactions with the type requested are displayed
1	the user sends a request to get all the transactions specifying a type
2	the system checks session token and its validity
3	the system displays all the transactions stored with the type requested

Scenario 1.6 (Nominal): sort transactions by date

The user wants to sort all the transactions stored by date (from the latest to the oldest).

The scenario ends correctly, the system sorts all the transactions stored.

Step	Description
Precondition	user logged in
Post condition	all the transactions stored are displayed by date (new to old)
1	the user sends a request to sort all the transactions stored
2	the system displays all the transactions stored ordering them by date

Scenario 1.7 (Nominal): get transactions by day

The user wants to get all the transactions done in a specific day.

The scenario ends correctly, the system displays all the transactions stored.

Step	Description
Precondition	user logged in
Post condition	all the transactions done in a specific date are displayed
1	the user sends a request to get all the transactions executed in a specific day
2	the system displays all the transactions stored in a specific day

Scenario 1.8 (Nominal): modify a transaction

The user wants to change some attributes of a transaction already stored. The user sends a request with the id of the transaction to modify and then the specific attributes to change.

The scenario ends correctly, the parameters of the transaction specified have been changed.

Step	Description
Precondition	user logged in & transaction specified must be already stored
Post condition	the transaction specified has been modified
1	the user sends a request to modify a specific transaction giving its id
2	the system checks session token and its validity
3	the system then checks if the transaction with the name specified is stored
4	the system asks the attributes to change
5	the user inserts the attributes to change and then modifies them changing their values
6	the system stores the new attributes of the transaction, which is modified correctly

Scenario 1.9 (Exception): unauthorized creation

The user, after making an expenditure on a specific date, wants to insert the amount spent inside the application in a specific category with a small description of the shopping made.

The scenario ends with an error, the transaction cannot be stored because the user has logged off (session token has expired) or the user has never logged in.

Step	Description
Precondition	user logged in but access token expired or user not logged in
Post condition	system returns error
1	user asks to create a new transaction
2	system checks access token and its validity
3	there's no access token or expired, system returns an error

Scenario 1.10 (Exception): unauthorized removal

The user, after inserting a new transaction, wants to delete the latter for some reason in order to not count that expenditure in the total expenses.

The scenario ends with an error, the transaction cannot be removed because the user has logged off (session token has expired) or the user has never logged in.

Step	Description
Precondition	user logged in but access token expired or user not logged in
Post condition	system returns error
1	user asks to remove a stored transaction
2	system checks access token and its validity
3	there's no access token or expired, system returns an error

Scenario 1.11 (Exception): unauthorized display

The user wants to visualize all the expenses recorded in the application.

The scenario ends with an error, the transactions cannot be displayed because the user has logged off (session token has expired) or the user has never logged in.

Step	Description
Precondition	user logged in but access token expired or user not logged in
Post condition	system returns error
1	user asks to display all the transactions stored in the system
2	system checks access token and its validity
3	there's no access token or expired, system returns an error

Scenario 1.12 (Exception): transaction to modify not found

The user wants to change some attributes of a transaction already stored. The user sends a request with the name of the transaction to modify and then the specific attributes to change. The scenario ends with an error, the system hasn't found any transaction with the name specified by the user.

Step	Description
Precondition	user logged in & transaction specified must not be stored
Post condition	the system returns an error
1	the user sends a request to modify a specific transaction giving its name
2	the system checks session token and its validity
3	the system then checks if the transaction with the name specified is stored
4	the system hasn't found the transaction with the name specified
5	the system returns an error

Use Case 2, UC2 - Categories

handle categories

Actors Involved	User
Precondition	user must be logged in
Post condition	
Nominal Scenarios	create new category, display categories
Variants	
Exceptions	unauthorized in creation, unauthorized in display

Scenario 2.1 (Nominal): create new category

The user wants to create a new category in which the expenses can be identified. The scenario ends correctly, the category is stored in the system.

Step	Description
Precondition	user must be logged in
Post condition	category stored in the system
1	user sends a request for creating a new category
2	system checks session token and its validity, user authorized
3	system asks category type and color
4	user inserts the two parameters
5	system stored the category with type and color chosen

Scenario 2.2 (Nominal): display categories

The user wants to display all the categories stored previously in which the expenses can be identified.

The scenario ends correctly, all the categories are displayed.

Step	Description
Precondition	user must be logged in && at least one category stored in the system
Post condition	system shows all categories stored
1	user sends a request to display all the categories
2	system checks the session token and its validity, user authorized
3	system displays all the categories found

Scenario 2.3 (Nominal): edit a category

The user wants to modify some attributes of a category already stored.

The scenario ends correctly, the category's attributes specified by the user have been modified with the values inserted by the user.

Step	Description
Precondition	user logged in && category specified already stored
Post condition	category's attribute specified has been changed
1	user sends a request to edit a category
2	system checks session token and its validity, user authorized
3	system checks if the category specified is already stored
4	system asks which parameter needs to be updated
5	user selects the parameter and inserts the new value
6	system correctly changes the parameter of the category specified

Scenario 2.4 (Nominal): get category by type

The user wants to display the category with a specified type.

The scenario ends correctly, the system shows only the category stored with the type specified by the user.

Step	Description
Precondition	user logged in
Post condition	system displays the category stored with the type specified
1	user sends a request to display the category stored with a specific type he inserts
2	system checks session token and its validity, user authorized
3	system displays the category found

Scenario 2.5 (Nominal): notify when category exceeds limit

The notification for when a category exceeds the limit (inserted previously by the user during the creation/modification of the category) is set to on. Every time a new transaction with the same type of the category is stored, the system checks the total spent of the category and the limit decided by the user.

In this scenario the total spent exceeds the limit established.

The scenario ends correctly, the system notifies the user that the category exceeds the limit.

Step	Description
Precondition	user has a premium account & notification is on & a new transaction is created
Post condition	notification sent to user because the category exceeded its limit
1	system retrieves category limit by the type of the newly created transaction
2	system computes new total spent = total spent + transaction amount
3	system checks if new total spent is less than category limit, is not
4	system sends a notification to user

Scenario 2.6 (Nominal): update a category's limit

The premium user wants to change the value of a category's limit. The user sends a request to the system and then specifies the new value.

The scenario ends correctly, the system changes the value of the category's limit specified with the new value inserted.

Step	Description
Precondition	user is logged in with a premium account & category specified must be already stored
Post condition	the value of the category's limit has been changed
1	user sends a request to the system to change the limit of a category
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system then checks the existence of the category, it's present so the system retrieves it
5	system asks for the new value of the category's limit

6	user inserts the new value of the category's limit
7	system correctly changes the value of the category's limit

Scenario 2.7 (Exception): unauthorized in creation

The user wants to create a new category in which the expenses can be identified.

The scenario ends with an error, the category cannot be stored because the user has logged off (session token has expired) or the user has never logged in.

Step	Description
Precondition	user must be logged in
Post condition	system returns an error
1	user sends a request for creating a new category
2	system checks session token and its validity, user unauthorized because session token not valid or expired
3	system returns an error

Scenario 2.8 (Exception): unauthorized in display

The user wants to display all the categories stored previously in which the expenses can be identified.

The scenario ends with an error, the categories cannot be displayed because the user has logged off (session token has expired) or the user has never logged in.

Step	Description
Precondition	user must be logged in && at least one category stored in the system
Post condition	system returns an error
1	user sends a request to display all the categories
2	system checks session token and its validity, user unauthorized because session token not valid or expired
3	system returns an error

Scenario 2.9 (Exception): category to modify not found

The user wants to modify some attributes of a category already stored.

The scenario ends with an error, the category's name specified by the user has not been found by the system.

Step	Description
Precondition	user logged in
Post condition	system returns an error
1	user sends a request to edit a category
2	system checks session token and its validity, user authorized

3	system checks if the category specified is already stored, not found
4	system returns an error

Scenario 2.10 (Exception): user doesn't have a premium account

The premium user wants to change the value of a category's limit. The user sends a request to the system and then specifies the new value.

The scenario ends with an error, the user does not have a premium account.

Step	Description
Precondition	user is logged in & category specified must be already stored
Post condition	system returns an error
1	user sends a request to the system to change the limit of a category
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does not
7	system returns an error

Use Case 3, UC3 - Sign up

sign up

Actors Involved	User
Precondition	user has no account
Post condition	user has an account and it is registered
Nominal Scenarios	in order to register, the user has to insert an email not already used and a password. After this procedure, the system will create and store the account
Variants	
Exceptions	email already used, missing parameters, passwords don't match

Scenario 3.1 (Nominal): sign up

The user wants to create an account in order to store and track all the expenses. In order to do so, the user sends a request to sign up with specific parameters (username, email and password). The password will be inserted twice to check the correctness and to confirm it.

The scenario ends correctly, the email inserted has not already been used and the account has been stored in the system. The user is now registered.

Step	Description
Precondition	user inserts an email not associated to an account (not already used) & password must respect specific constraints
Post condition	user has an account associated to the email he has inserted
1	user asks to sign up

2	system asks username, email and valid password twice
3	user inserts username, email and valid password twice
4	system check if the email is not already in use & password respects constraints & if the two password match
5	system encrypts password and stores account

Scenario 3.2 (Exception): email already used

The user wants to create an account in order to store and track all the expenses. In order to do so, the user sends a request to sign up with specific parameters (username, email and valid password).

The scenario ends with an error, the email inserted has already been used and the account cannot be created. The user cannot be registered.

Step	Description
Precondition	user inserts an email already used and associated to an account
Post condition	system returns an error
1	user asks to sign up
2	system asks username, email and password
3	user inserts username, email already associated to an account and password
4	system check if the email is not already in use, email already used
5	system returns an error

Scenario 3.3 (Exception): missing parameters

The user wants to create an account in order to store and track all the expenses. In order to do so, the user sends a request to sign up but with some missing parameters (email or username or both).

The scenario ends with an error, the email or the username were not inserted and the account cannot be created. The user cannot be registered.

Step	Description
Precondition	user inserts an email not associated to an account (not already used)
Post condition	system returns an error
1	user asks to sign up
2	system asks username, email and password
3	user inserts username and password or email and password
4	system check the parameters, but some are missing
5	system returns an error

Scenario 3.4 (Exception): password not valid

The user wants to create an account in order to store and track all the expenses. In order to do so, the user sends a request to sign up with specific parameters (username, email and password).

The scenario ends with an error, the password inserted does not respect the constraints established. The user cannot be registered.

Step	Description
Precondition	user inserts an email not associated to an account (not already used) & password does not respect specific constraints
Post condition	user has an account associated to the email he has inserted
1	user asks to sign up
2	system asks username, email and valid password
3	user inserts username, email and valid password
4	system check if the email is not already in use & password respects constraints
5	system returns an error, password inserted does not respect constraints

Scenario 3.5 (Exception): passwords don't match

The user wants to create an account in order to store and track all the expenses. In order to do so, the user sends a request to sign up with specific parameters (username, email and password). The password will be inserted twice to check the correctness and to confirm it.

The scenario ends with an error, the field 'password' and the field 'confirm password' (used to check that the password has been entered correctly) don't match. All parameters must be re-entered.

Step	Description
Precondition	user inserts an email not associated to an account (not already used) & password must respect specific constraints
Post condition	system returns an error
1	user asks to sign up
2	system asks username, email and valid password twice
3	user inserts username, email and valid password twice
4	system check if the email is not already in use & password respects constraints & if the two password match, they don't
5	system returns an error

Use Case 4, UC4 - Authentication

handle user authentication

Actors Involved	User
-----------------	------

Precondition	user has already an account
Post condition	
Nominal Scenarios	login, credentials recovery
Variants	
Exceptions	account not found, wrong credentials

Scenario 4.1 (Nominal): login

The user wants to sign in in order to retrieve all past transactions stored and all the categories created. In order to do so, the user sends a request to sign in with specific parameters (email and password).

The scenario ends correctly, the parameters inserted are correct and they have been found in the system. The user is now logged in and the cookie token is set, so the session is established.

Step	Description
Precondition	user must have an account
Post condition	User is logged in := auth token cookie is set in the client browser
1	user sends a request to login
2	system asks email and password
3	user inserts email and password
4	system checks if the credentials are correct, they are
5	system creates an auth token
6	user is logged in, the auth token cookie is set in the response

Scenario 4.2 (Nominal): forgotten password

The user forgets the password, so the user sends a request to recover it.

The scenario ends correctly, the system asks for the email associated with the account and then sends a new password, generated randomly, to the account's email.

Step	Description
Precondition	user must have an email associated to an account
Post condition	password has been recovered
1	user sends a request for recovering the password
2	system asks for email associated to the user's account
3	user inserts email associated to the account
4	system checks if the email is correctly associated to an account, it is
5	system sends a new random password to the account's email and so to the user
6	password is recovered

Scenario 4.3 (Exception): account not found

The user wants to sign in in order to retrieve all past transactions stored and all the categories created. In order to do so, the user sends a request to sign in with specific parameters (email and password).

The scenario ends with an error, the parameters inserted are not correct because they have not been found in the system, therefore the user doesn't have an account. The user cannot be logged in.

Step	Description
Precondition	user doesn't have an account
Post condition	system returns an error
1	user sends a request to login
2	system asks email and password
3	user inserts email (not associated to an account) and password
4	system checks if the credentials are correct, email not stored
5	system returns an error

Scenario 4.4 (Exception): wrong credentials

The user wants to sign in in order to retrieve all the past transactions stored and all the categories created. In order to do so, the user sends a request to sign in with specific parameters (email and password).

The scenario ends with an error, the parameters inserted are not correct because they have been found in the system but one or more parameters are not corresponding. The user cannot be logged in.

Step	Description
Precondition	user must have an account && credentials inserted are not correct (email/password)
Post condition	system returns an error
1	user sends a request to login
2	system asks email and password
3	user inserts email and password (one or both incorrect)
4	system checks if the credentials are correct, they are not
5	system returns an error

Scenario 4.5 (Exception): forgotten password, email not associated to an account

The user forgets the password, so the user sends a request to recover it.

The scenario ends correctly, the system asks for the email associated with the account and then sends the password to the account's email.

Step	Description
------	-------------

Precondition	user does not have an email associated to an account
Post condition	system returns an error
1	user sends a request for recovering the password
2	system asks for email associated to the user's account
3	user inserts email associated to the account
4	system checks if the email is correctly associated to an account, it is not
5	system returns an error

Use Case 5, UC5 - Users

handle users

Actors Involved	Admin, User, Payment Services
Precondition	user logged in and has admin privileges
Post condition	
Nominal Scenarios	display all users, get user by username, switching from free to premium user
Variants	
Exceptions	unauthorized display, unauthorized get user by username

Scenario 5.1 (Nominal): display all users

The administrator, which is a logged user with Admin privileges, wants to display all the accounts stored in the system, so the admin user sends a request to the system which, after verifying the session token, displays all the accounts registered.

The scenario ends correctly, the system lists all the accounts found.

Step	Description
Precondition	user logged in with admin privileges
Post condition	system displays all users stored in the system
1	admin asks the system to display all users
2	system checks access token and its validity
3	system displays all users found in the system

Scenario 5.2 (Nominal): display user profile by username

The logged in user wants to get all the information stored in the system about one user, so the user sends a request to the system which, after verifying the session token, displays all the information about the account registered.

The scenario ends correctly, the system displays all the information found about the account searched.

Step	Description
------	-------------

Precondition	user logged in
Post condition	system displays all the information found about the account searched
1	user asks the system to get all user's information providing a username, which must be already associated to an account
2	system checks access token, its validity and username
3	system displays all user's information found in the system

Scenario 5.3 (Nominal): switching from free to premium account

The user wants to switch from free to premium account. The user sends a request and inserts a valid payment method.

The scenario ends correctly, the user account has been changed to a premium account.

Step	Description
Precondition	user logged in with a free account
Post condition	user has now a premium account
1	user sends a request for switching to premium
2	system checks access token and its validity
3	system asks for a payment method
4	user inserts payment method
5	system elaborates the payment, successful payment
6	user account is now premium

Scenario 5.4 (Nominal): change password

The user wants to change the password of the account. The user, already logged in, sends a request and inserts the old password, then the new password respecting the constraints.

The scenario ends correctly, the account's password has been changed.

Step	Description
Precondition	user logged in & knows the old password & new password respects the constraints
Post condition	password has been changed
1	user sends a request for changing the password
2	system checks access token and its validity
3	system asks for the old password and the new one
4	user inserts the old and the new password
5	system checks the old password's correctness and if the password respects the constraints, it does
6	password is changed

Scenario 5.5 (Exception): unauthorized display all users

The administrator, which is a logged user with Admin privileges, wants to display all the accounts stored in the system, so the admin user sends a request to the system which, after verifying the session token, displays all the accounts registered.

The scenario ends correctly, the system lists all the accounts found.

Step	Description
Precondition	user logged in with admin privileges
Post condition	system displays all users stored in the system
1	admin asks the system to display all users
2	system checks access token and its validity
3	system displays all users found in the system

Scenario 5.6 (Exception): unauthorized display user profile

The logged in user wants to get all the information stored in the system about one user, so the user sends a request to the system which, after verifying the session token, displays all the information about the account registered.

The scenario ends correctly, the system displays all the information found about the account searched.

Step	Description
Precondition	user logged in
Post condition	system displays all the information found about the account searched
1	user asks the system to get all user's information providing a username, which must be already associated to an account
2	system checks access token, its validity and username
3	system displays all user's information found in the system

Scenario 5.7 (Exception): unsuccessful payment

The user wants to switch from free to premium account. The user sends a request and inserts a payment method.

The scenario ends with an error, the payment failed.

Step	Description
Precondition	user logged in with a free account
Post condition	system return an error
1	user sends a request for switching to premium
2	system checks access token and its validity
3	system asks for a payment method
4	user inserts payment method

5	system elaborates the payment, unsuccessful payment
6	system returns an error

Scenario 5.8 (Exception): old password incorrect

The user wants to change the password of the account. The user, already logged in, sends a request and inserts the old password but it's not correct.

The scenario ends with an error, the password cannot be changed.

Step	Description
Precondition	user logged in & does not know the old password
Post condition	system returns an error
1	user sends a request for changing the password
2	system checks access token and its validity
3	system asks for the old password and the new one
4	user inserts the old and the new password
5	system checks the old password's correctness and if the password respects the constraints, old password is not correct
6	password cannot be changed, system returns an error

Scenario 5.9 (Exception): new password not valid

The user wants to change the password of the account. The user, already logged in, sends a request and inserts the old password, then the new password but does not respect the constraints on passwords.

The scenario ends with an error, the password cannot be changed.

Step	Description
Precondition	user logged in & knows the old password & new password doesn't respect constraints
Post condition	system returns an error
1	user sends a request for changing the password
2	system checks access token and its validity
3	system asks for the old password and the new one
4	user inserts the old and the new password
5	system checks the old password's correctness and if the password respects the constraints, new password is not valid
6	password cannot be changed, system returns an error

Use Case 6, UC6 - Logout

logout

Actors Involved	User
Precondition	user has already an account and user is logged in
Post condition	user is logged out
Nominal Scenarios	user, previously logged in, logs out
Variants	
Exceptions	account not found

Scenario 6.1 (Nominal): logout

The user, previously logged in, wants to log out, so the user sends a request in order to log off.

The scenario ends correctly, the user is logged out.

Step	Description
Precondition	user must have an account and be logged in
Post condition	user is logged out := cookie token is set to empty string
1	user sends a request to log out
2	system checks access token and its validity, user is logged in
3	system logs out the user

Scenario 6.2 (Exception): account not found

The user, previously logged in, wants to sign off, so the user sends a request in order to log off.

The scenario ends with an error, the account cannot be found.

Step	Description
Precondition	the access token does not have a user associated
Post condition	system returns an error
1	user sends a request to log out
2	system checks access token
3	system cannot find a user associated to the access token
4	system returns an error

Use Case 7, UC7 - Analytics

handle analytics

Actors Involved	Premium User
Precondition	user logged in & has a premium account
Post condition	user can see all the analytics about his own account

Nominal Scenarios	get total expenditure by month, by category, by category and month, by group, by group and month, by group and category, by group and category and month
Variants	
Exceptions	user unauthorized

Scenario 7.1 (Nominal): get total expenditure by month

The premium user wants to know the total expenditure of a specific month. The user sends a request specifying the month.

The scenario ends correctly, the system displays the total expenditure by month specified.

Step	Description
Precondition	user logged in & has a premium account
Post condition	system get the total expenditure by month specified
1	user sends a request to get total expenditure specifying month
2	system checks session token and its validity, user authorized
3	system displays the total expenditure by month specified

Scenario 7.2 (Nominal): get total expenditure by category

The premium user wants to know the total expenditure of a specific category. The user sends a request specifying the category.

The scenario ends correctly, the system displays the total expenditure by category specified.

Step	Description
Precondition	user logged in & has a premium account
Post condition	system get the total expenditure by category specified
1	user sends a request to get total expenditure specifying category type
2	system checks session token and its validity, user authorized
3	system displays the total expenditure by category specified

Scenario 7.3 (Nominal): get total expenditure by category and month

The premium user wants to know the total expenditure of a specific month and category. The user sends a request specifying the category and the month.

The scenario ends correctly, the system displays the total expenditure by category and month specified.

Step	Description
Precondition	user logged in & has a premium account
Post condition	system get the total expenditure by month and category specified
1	user sends a request to get total expenditure specifying category type and month
2	system checks session token and its validity, user authorized

3	system displays the total expenditure by category and month specified
---	---

Scenario 7.4 (Nominal): get total expenditure by group

The premium user wants to know the total expenditure of a specific group. The user sends a request specifying the group.

The scenario ends correctly, the system displays the total expenditure by group specified.

Step	Description
Precondition	user logged in & has a premium account
Post condition	system get the total expenditure by group specified
1	user sends a request to get total expenditure specifying group name
2	system checks session token and its validity, user authorized
3	system displays the total expenditure by group specified

Scenario 7.5 (Nominal): get total expenditure by group and month

The premium user wants to know the total expenditure of a specific month and group. The user sends a request specifying the month and group.

The scenario ends correctly, the system displays the total expenditure by month and group specified.

Step	Description
Precondition	user logged in & has a premium account
Post condition	system get the total expenditure by group and month specified
1	user sends a request to get total expenditure specifying group name and month
2	system checks session token and its validity, user authorized
3	system displays the total expenditure by group and month specified

Scenario 7.6 (Nominal): get total expenditure by group and category

The premium user wants to know the total expenditure of a specific group and category. The user sends a request specifying the category and group.

The scenario ends correctly, the system displays the total expenditure by group and category specified.

Step	Description
Precondition	user logged in & has a premium account
Post condition	system get the total expenditure by group and category specified
1	user sends a request to get total expenditure specifying group name and category type
2	system checks session token and its validity, user authorized
3	system displays the total expenditure by group and category specified

Scenario 7.7 (Nominal): get total expenditure by group, month and category

The premium user wants to know the total expenditure of a specific month, group and category. The user sends a request specifying the month, group and category.

The scenario ends correctly, the system displays the total expenditure by month, group and category specified.

Step	Description
Precondition	user logged in & has a premium account
Post condition	system get the total expenditure by group, month and category specified
1	user sends a request to get total expenditure specifying group name, month and category type
2	system checks session token and its validity, user authorized
3	system displays the total expenditure by group, month and category specified

Scenario 7.8 (Exception): user unauthorized

The user wants to display the total expenditure by something but it's not logged in or the session token is expired.

The scenario ends with an error, the user is not authorized to display the total expenditure.

Step	Description
Precondition	user not logged in
Post condition	system returns an error
1	user sends a request to get total expenditure by something
2	system checks session token and its validity, user unauthorized
3	system returns an error

Use Case 8, UC8 - Groups

handle groups

Actors Involved	Premium User
Precondition	use logged in & has a premium account
Post condition	
Nominal Scenarios	create group, delete group, add user to group, add transaction to group, add category to group, edit group, remove user from group, delete transaction from group, delete category from group
Variants	
Exceptions	unauthorized user, user to add not found, user to remove not found, transaction to remove not found, category to remove not found, user to add has not a premium

	account, user has not a premium account,
--	--

Scenario 8.1 (Nominal): create a group

The premium logged in user wants to create a group to share transactions and categories with other members.

The user sends a request to create the group.

The scenario ends correctly, after the user inserts name and a picture, the system creates the group.

Step	Description
Precondition	user logged in & premium account
Post condition	group is created
1	user sends a request to create a group
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system asks group name, picture and color
5	user inserts group name, picture and color that represents the group
6	system creates the group with the parameters specified

Scenario 8.2 (Nominal): add user to a group

The premium logged in user wants to add a member to a group already created.

The scenario ends correctly, the user inserts the username of the account to add to the group, which must be a premium account. The system adds the account associated with the username inserted to the group specified in the request.

Step	Description
Precondition	user logged in & premium account & member of the group & member to add must have premium account too
Post condition	member is added to the group specified
1	user sends a request to add a member to a group specifying the group name
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system checks if the user who asked is a member of the group specified
5	system asks for the user to add
6	user inserts username of the account to add to the group
7	system checks if username exists, account found
8	system checks if the account to add is a premium account, the account is premium
9	system adds the chosen account to the group's members

Scenario 8.3 (Nominal): add transaction to a group

The premium logged in user wants to add a transaction to a group already created.

The scenario ends correctly, the transaction is added to the group specified in the request by the user, now every member of the group is able to see the transaction created.

Step	Description
Precondition	user logged in & premium account & member of the group
Post condition	transaction is added to the group
1	user sends a request to add a transaction to a specific group
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system checks if the user is a member of the group specified
5	system asks for transaction's parameters to create it
6	user inserts all the values for each parameter of the transaction
7	system adds the transaction to the group

Scenario 8.4 (Nominal): add a category to a group

The premium logged in user wants to add a category to a group already created.

The scenario ends correctly, the category is added to the group specified in the request by the user, now every member of the group is able to see the category created.

Step	Description
Precondition	user logged in & premium account & member of the group
Post condition	category is added to the group
1	user sends a request to add a category to a specific group
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system checks if the user is a member of the group specified
5	system asks for category's parameters to create it
6	user inserts all the values for each parameter of the category
7	system adds the category to the group

Scenario 8.5 (Nominal): edit a group

The premium logged in user, who is a member of a group, wants to modify one of the group's parameters.

The scenario ends correctly, the user chooses the parameter to change and inserts the new value and the system modifies it.

Step	Description
Precondition	user logged in & premium account & member of the group

Post condition	group's parameter modified
1	user sends a request to modify the parameters' values of a specific group
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system checks if the user is a member of the group specified
5	system asks which parameters to modify and their new values
6	user chooses the parameters to modify and inserts the new values
7	system changes the values of the parameters specified

Scenario 8.6 (Nominal): remove user from a group

The premium logged in user, who is a member of the group, wants to remove a user from the group.

The scenario ends correctly, the user inserts username of the account to remove and the system removes the account from the group.

Step	Description
Precondition	user logged in & premium account & member of the group & user to remove is a member of the group
Post condition	user specified is removed from the group
1	user sends a request to remove a user from a group specified
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system checks if the user is a member of the group specified
5	system asks username of the account to remove
6	user inserts username
7	system checks if the account is a member of the group
8	system removes the account from group's members

Scenario 8.7 (Nominal): delete transaction from a group

The premium logged in user, who is a member of the group, wants to delete a transaction from a group.

The scenario ends correctly, the user inserts transaction id to remove and the system removes the transaction from the group, if present.

Step	Description
Precondition	user logged in & premium account & member of the group & transaction is related to the group
Post condition	transaction specified is deleted from the group
1	user sends a request to delete a transaction from a group specified
2	system checks session token and its validity, user authorized

3	system checks if the user has a premium account, it does
4	system checks if the user is a member of the group specified
5	system checks if the transaction is added to the group
6	system removes the transaction from group

Scenario 8.8 (Nominal): delete category from a group

The premium logged in user, who is a member of the group, wants to delete a category from a group.

The scenario ends correctly, the user inserts category type to remove and the system removes the category from the group, if present.

Step	Description
Precondition	user logged in & premium account & member of the group & category is added to the group
Post condition	category specified is deleted from the group
1	user sends a request to delete a category from a group specified
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system checks if the user is a member of the group specified
5	system checks if the category is added to the group
6	system removes the category from group

Scenario 8.9 (Nominal): display all categories of a group

The premium logged in user, who is a member of the group, wants to display all the categories from a group.

The scenario ends correctly, the user can visualize all the categories created inside the group. .

Step	Description
Precondition	user logged in & premium account & member of the group
Post condition	all the categories of a specific group are displayed
1	user sends a request to display all the categories from a group specified
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system checks if the user is a member of the group specified
6	system displays all the categories from the group

Scenario 8.10 (Nominal): display all transactions of a group

The premium logged in user, who is a member of the group, wants to display all the transactions from a group.

The scenario ends correctly, the user can visualize all the transactions created inside the group.

Step	Description
Precondition	user logged in & premium account & member of the group
Post condition	all the transactions of a specific group are displayed
1	user sends a request to display all the transactions from a group specified
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system checks if the user is a member of the group specified
6	system displays all the transactions from the group

Scenario 1.11 (Nominal): modify a transaction/category

The premium user, a member of a group, wants to change some attributes of a transaction/category already stored inside a group. The user sends a request with the id of the transaction/category to modify and then the specific attributes to change.

The scenario ends correctly, the parameters of the transaction/category specified have been changed.

Step	Description
Precondition	user logged in & premium account & member of the group & transaction/category specified must be already stored
Post condition	the transaction specified has been modified
1	the user sends a request to modify a specific transaction/category giving its id
2	the system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system checks if the user is a member of the group specified
5	the system then checks if the transaction/category with the id specified is stored
6	the system asks the attributes to change
7	the user inserts the attributes to change and then modifies them changing their values
8	the system stores the new attributes of the transaction/category, which is modified correctly

Scenario 8.12 (Exception): category/transaction not present

The premium logged in user, who is a member of the group, wants to delete a transaction/category from a group.

The scenario ends with an error, the category/transaction inserted is not present in the group.

Step	Description
Precondition	user logged in & premium account & member of the group

Post condition	system returns an error
1	user sends a request to delete a transaction/category from a group specified
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system checks if the user is a member of the group specified
5	system checks if the transaction/category is added to the group, it is not
6	system returns an error

Scenario 8.13 (Exception): user unauthorized

The premium user wants to create a group to share transactions and categories with other members.

The user sends a request to create the group.

The scenario ends with an error, the user is not logged in or the session token is expired.

Step	Description
Precondition	user has premium account
Post condition	system returns an error
1	user sends a request to create a group
2	system checks session token and its validity, user NOT authorized
3	system returns an error

Scenario 8.14 (Exception): user to add has not premium account

The premium logged in user wants to add a member to a group already created.

The scenario ends with an error, the system cannot add the account specified by the user because the account to add is not a premium one.

Step	Description
Precondition	user logged in & premium account & member of the group
Post condition	system returns an error
1	user sends a request to add a member to a group specifying the group name
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system checks if the user is a member of the group specified
5	system asks for the user to add
6	user inserts username of the account to add to the group
7	system checks if username exists, account found
8	system checks if the account to add is a premium account, the account is NOT premium
9	system returns an error

Scenario 8.15 (Exception): user to add not found

The premium logged in user wants to add a member to a group already created.

The scenario ends with an error, the system cannot add the account specified by the user because the account to add does not exist.

Step	Description
Precondition	user logged in & premium account & member of the group
Post condition	system returns an error
1	user sends a request to add a member to a group specifying the group name
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system checks if the user is a member of the group specified
5	system asks for the user to add
6	user inserts username of the account to add to the group
7	system checks if username exists, account NOT found
8	system returns an error

Scenario 8.16 (Exception): user has not a premium account

The premium logged in user wants to add a transaction to a group already created.

The scenario ends with an error, the user is not a member of the group specified.

Step	Description
Precondition	user logged in & premium account
Post condition	system returns an error
1	user sends a request to add a transaction to a specific group
2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it doesn't
4	system returns an error

Scenario 8.17 (Exception): user to remove not found

The premium logged in user, who is a member of the group, wants to remove a user from the group.

The scenario ends with an error, the user to remove is already not a member of the group specified.

Step	Description
Precondition	user logged in & premium account & member of the group
Post condition	system returns an error
1	user sends a request to remove a user from a group specified

2	system checks session token and its validity, user authorized
3	system checks if the user has a premium account, it does
4	system checks if the user is a member of the group specified
5	system asks username of the account to remove
6	user inserts username
7	system checks if the account is a member of the group, it is NOT
8	system returns an error