

# EZWallet – Group 23 - V2

## Functional Requirements

### F1: Manage Users authentication

1. Log User in
2. Log User out

3. Credentials recovery

4. Change password

### F2: Manage Users

1. Get all user
2. Create a new User (Sign up)
3. Get user by username
4. Switch from free to premium account

### F3: Manage Transactions

1. Get all transactions
2. Create a new transaction
3. Delete an existing transaction
4. Get all labels
5. Sort transactions by date
6. Get all transactions by type

## 7. Update transaction

### F4: Manage Categories

1. Get all categories
2. Create a new category

#### 2.1 Get category by type

3. Update a category color and type
5. Push notification when total expenses of a category reaches its limit (PREMIUM)
6. Update a category limit (PREMIUM)

### F5: Manage analytics (PREMIUM)

1. Get the total expenditure by month
2. Get the total expenditure by category
3. Get the total expenditure by month and category
4. Get the total expenditure by group
5. Get the total expenditure by group and month
6. Get the total expenditure by group and category
7. Get the total expenditure by group and category and month

### F6: Manage groups (PREMIUM)

1. Create group
2. Add user to a group
3. Add transaction to a group
4. Add category to a group
5. Delete a user from a group
6. Get categories of a group
7. Get transactions of a group
8. Delete transaction from a group
9. Delete category from a group
10. Edit category of a group
11. Edit transaction of a group

## Table of access rights

This table contains newly added functions only (V2 functions).

	Logged out User	Logged in User	Premium User	Admin
<b>F2.4)</b> Switch from free to premium account	no	yes	no	yes
<b>F3.5)</b> Sort transactions by date	no	yes	yes	yes
<b>F3.6)</b> Get all transactions by type	no	yes	yes	yes
<b>F3.7)</b> Update transaction	no	yes	yes	yes
<b>F4.3)</b> Update a category color and name	no	yes	yes	yes

<b>F4.5) Push notification when total expenses of a category reaches its limit</b>	no	no	yes	yes
<b>F4.6) Update a category limit</b>	no	no	yes	yes
<b>F5.1) Get the total expenditure by month</b>	no	no	yes	yes
<b>F5.2) Get the total expenditure by category</b>	no	no	yes	yes
<b>F5.3) Get the total expenditure by month and category</b>	no	no	yes	yes
<b>F5.4) Get the total expenditure by group</b>	no	no	yes	yes

<b>F5.5) Get the total expenditure by group and month</b>	no	no	yes	yes
<b>F5.6) Get the total expenditure by group and category</b>	no	no	yes	yes
<b>F5.6) Get the total expenditure by group and category and month</b>	no	no	yes	yes
<b>F6.1) Create a group</b>	no	no	yes	yes
<b>F6.2) Add user to a group</b>	no	no	yes	yes
<b>F6.3) Add transaction to a group</b>	no	no	yes	yes
<b>F6.1) Add category to a group</b>	no	no	yes	yes

# Non functional requirements

## Usability

1. Users with little to none experience in smartphone use ( < 1 year of usage ) must be able to correctly use all functions after having the initial tutorial complete. The tutorial must have at least one help frame for each view and each function.
2. Errors in usage must not exceed the number of 3 per day for users that have completed the tutorial.

## Efficiency

1. All functions must return in less than 1 sec
2. All views must load in  $\leq 0.5$  sec

## Reliability

1. Max server downtime must be at most 5 min per month
2. The frontend service must be always up, at least to serve an information page to the user

## Maintainability

1. Fixing defects must take no more than 6 ph
2. Adding or updating a function must always take no more than 16 ph
3. Testing of an added or updated function must always take no more than 8ph
4. Deploying a new version of the server must take little to none human effort (<1ph)
5. Deploying a new version of the client must take little to none human effort (<1ph)

## Portability

1. The app must be usable on all kind of devices
2. The design must be responsive

## Security

1. Authentication must be handled to give access only to registered users.
2. Authorization must be handled: Access to parts of the system is granted only to authorized users. It must be clear to the user what he/she has access to, no more than 4 unauthorized accesses weekly from a normal user.
3. Security concerns about illicit activities on secured routes must be handled in no more than one week.
4. Account passwords must respect defined constraints (for UC5.9)

