## EZWallet

**0 Informal description**

EZWallet (read EaSy Wallet) is a software application designed to help individuals and families keep track of their expenses. Users can enter and categorize their expenses, allowing them to quickly see where their money is going. EZWallet is a powerful tool for those looking to take control of their finances and make informed decisions about their spending.

**1 Business model**

V1: Free: no ads, no payments needed, the app is used only for test purposes.

V2: Freeware (adware) and pro versions: in the free version the interface show advertisement banners refreshed via Internet, in pro version (a fee to pay) all ads are hidden, and the user unlocks extra features.

**2 Stakeholders**

* User: A person who uses the application to manage its expenses
* COO: A worker whose goal is to administrate the evolution of the project
  + Developer: A worker whose goal is to design and create the GUI and the frontend part of the project
  + Designer: A worker whose goal is to develop the code of the project and manage the backend part
* Database: A tool used to manage the data of the application, in this case it is MongoDB

**3 Context diagram**

Immagine che contiene diagramma

Descrizione generata automaticamente

**4 Interfaces**

|  |  |  |
| --- | --- | --- |
| **Actor** | **Physical Interface** | **Logical Interface** |
| User | Device (with browser), Internet connection | GUI (TBD) |
| COO | Device (with browser), Internet connection | Bash, Docker, Postman |
| Developer | PC | GUI, WebStorm, VSCode, StarUML |
| Designer | PC | Browser, StarUML, GUI Designer App |

**5 Stories and Personas**

**//OLD VERSION**

Persona 1: A foreign student, female, 20 yo, no income

who uses the application to manage his expenses and to keep track of his transactions.

Persona 2: A parent, young professional, 30, needs to be careful of their expanses.

A family who wants to manage their expanses that wants to reduce their superfluous expanses.

Persona 3: A parent with teenager, entrepreneur, high income, self-made

Persona 4: young financial expert, works as financial consultant, medium income, do some stock exchanges as hobby

A professional trader that wants to control the profit earned by his investment.

Persona 5: Man, middle aged, loses his work.

**//FINAL VERSION**

**Persona 1: student, female, 20 yo, no income**

Story 1:

A foreign student has moved in a new city to apply to the university. She doesn't have any time to work and so the only form of income she receives is from her parents. They send her money every month to cover the house's expenses and since the money are only the strictly necessary, she decides to use the EZWallet web application to keep track of the expenses and avoid the risk of remaining with some expenses uncovered and, if possible, dedicate some remaining money to personal purposes. Every month she discuss with her parents the situation in order to adjust it as soon as possible.

Story 2:

A student has moved into a new city in order to be nearer to the university she applied to. She has 2 roommates and they record any of their expenses made for the house or for the common needs on the virtual wallet to have the possibility to balance weekly. By doing so they avoid bad situations in which someone owes money to the others for a long period of time.

**Persona 2: parent, young professional, 30 yo**

Story 1:

A recently formed family parent wants to reach a more stable financial situation. The need of organizing in the best way the resources leads to the choice of keeping track of all the expenses on the EZWallet web application. Doing so makes possible to analyze the habits and modify them in order to reach the goal set.

Story 2:

A young family parent wants to save money monthly in order to be able to cover important expected expenses such as a move or the care of a child. This is made possible by keeping track of all the expenses and decide every time what is appropriate to cut on in order to maximize the results.

**Persona 3: parent with teenagers, entrepreneur, high income, self-made**

Story 1:

A parent wants to educate his growing teenagers on the matter of money management. This decision is supported by asking them to track the family expenses on a daily basis so that they can realize the importance of proper decisions.

**Persona 4: young financial expert, works as financial consultant, medium income, do some stock exchanges as hobby**

Story 1:

A young financial expert has a good knowledge about the management of cash flows, so he decides to track the transactions performed in his investment activity in order to have an appropriate and precise report on the trend of his business. Since he uses different tools and applications to perform his trading, he decides to rely on a third-party wallet to have a general complete overview.

Story 2:

A young man who travels very much due to his work needs to keep track of his expenses in order to adapt to different situations quickly, he decides to use a web application such as EZWallet thanks to its portability such as the possibility to access it anytime from anywhere with any device.

**Persona 5: man, middle aged, victim of a financial crisis**

Story 1:

A middle-aged man who loses his job due to the recession and the effect of a crisis need to quickly modify and adapt his habits in order to properly manage his resources during a possible unemployment period. He starts to keep track of the expenses in order to realize what is the forecasted period he can cover without any income also considering some unexpected expenses.

Story 2:

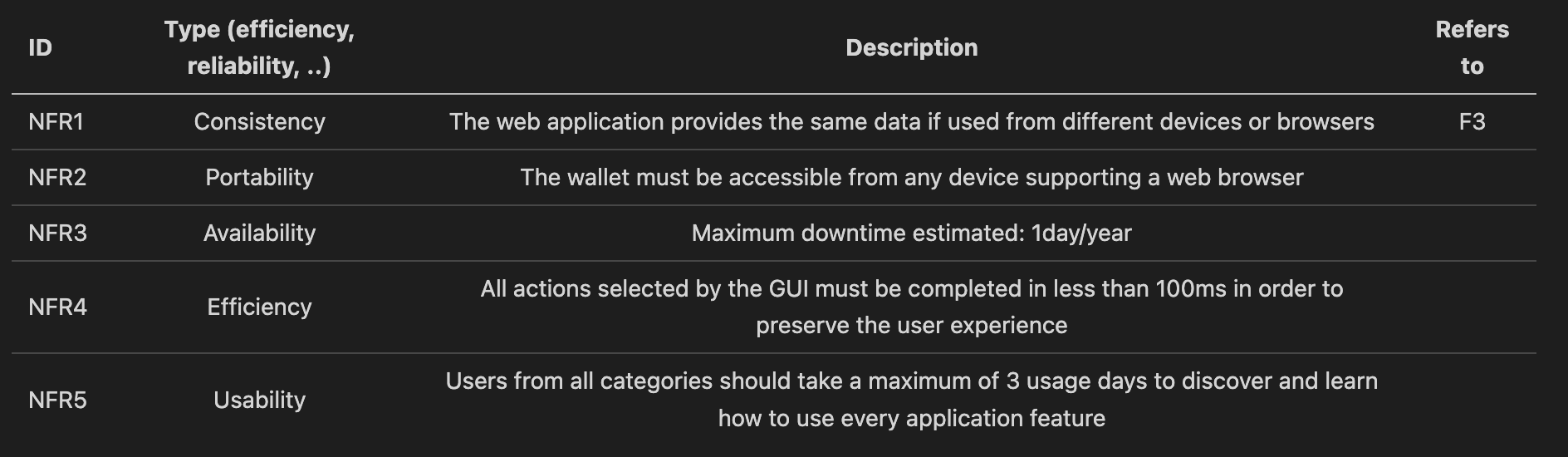
As an effect of the crisis a man notices that the cost of the life is constantly increasing so he decides to start keeping track of the expenses performed using the EZWallet application in order to construct an history and have the possibility to make comparisons with past months and directly organize his personal finance.

**6 FR and Table of Right**

Immagine che contiene testo

Descrizione generata automaticamente

**7 NFR**

****

**8 Use Case Diagram and Use Cases**

Immagine che contiene diagramma

Descrizione generata automaticamente

UC 1 : Register

UC 2 : Login

UC 3 : Logout

UC 4 : View other users information

Scenario1 : see other users information

Scenario 2 : see your own information

UC 5 : Display catergories

UC 6 : Get labels

UC 7 : Create a new category

UC 8 : CRUD actions on the transactions :

Scenario 1 : Create transaction

Scenario 2 : Read transaction

Scenario 3 : Delete transaction

Use case 1, UC1, **Sign up**

| **Actors Involved** |  |
| --- | --- |
| Precondition | User has no account |
| Post condition | User has an account |
| Nominal Scenario | The user enters his information to create a new account |
| Variants |  |
| Exceptions |  |

| **Scenario 1.1** | **The user creates a new account** |
| --- | --- |
| Precondition | User has no account |
| Post condition | User has an account |
| Step# | Description |
| 1 | User clicks the button to sign up |
| 2 | System asks for username, email and password |
| 3 | User enters the information |
| 4 | System checks if the email is already in use |
| 5 | System stores account |

Use case 2, UC2, **Login**

| **Actors Involved** |  |
| --- | --- |
| Precondition | User has an account |
| Post condition | User is logged in |
| Nominal Scenario | The user enters his credentials to log in |
| Variants |  |
| Exceptions | Wrong password, user already logged in |

| **Scenario 2.1** | **The user logs in** |
| --- | --- |
| Precondition | User has an account |
| Post condition | User is logged in |
| Step# | Description |
| 1 | User clicks the button to log in |
| 2 | System asks for email and password |
| 3 | User enters the information |
| 4 | System checks if the user is already logged in |
| 5 | System checks if the user is not registered |
| 6 | System checks if the password is correct |
| 7 | System creates the access token |
| 8 | System creates the refresh token |
| 9 | System stores the refresh token |
| 10 | System sends the access token to the client |

Use case 3, UC3, **Logout**

| **Actors Involved** |  |
| --- | --- |
| Precondition | User is logged in |
| Post condition | User is logged out |
| Nominal Scenario | The user logs out |
| Variants |  |
| Exceptions | User not found, user is already logged out |

| **Scenario 3.1** | **The user logs out** |
| --- | --- |
| Precondition | User is logged in |
| Post condition | User is logged out |
| Step# | Description |
| 1 | User clicks the button to log out |
| 2 | System checks if the user is logged in controlling access token and refresh token |
| 3 | System searches the user using the refresh token |
| 4 | System deletes the refresh token and the access token |
| 5 | System saves the status of the logout |

Use case 4, UC4, **View other users information**

| **Actors Involved** |  |
| --- | --- |
| Precondition | User must have admin privileges or must be logged in |
| Post condition | User sees other users information or its own information |
| Nominal Scenario | The user with admin privileges wants to see all users information, the user searches for its own information |
| Variants | Admin sees all users information, user sees its own information |
| Exceptions | User is not found, user is searching for other users information |

| **Scenario 4.1** | **User sees all users information** |
| --- | --- |
| Precondition | User must have admin privileges |
| Post condition | User sees all users information |
| Step# | Description |
| 1 | User clicks the button to view all users information |
| 2 | System checks if the user has admin privileges |
| 3 | System returns the information of all users |

| **Scenario 4.2** | **User searches for its own information** |
| --- | --- |
| Precondition | User must be logged in |
| Post condition | User sees its own information |
| Step# | Description |
| 1 | User clicks the button to view its own information |
| 2 | System asks for the username to search |
| 3 | User enters its username |
| 4 | System checks if the user is logged in |
| 5 | System searches the user |
| 6 | System checks if the username searched is the same of the logged user |
| 7 | System returns the information of the user |

Use case 5, UC5, **Display categories**

| **Actors Involved** |  |
| --- | --- |
| Precondition | User is logged in |
| Post condition | User sees all categories |
| Nominal Scenario | The user wants to see all categories |
| Variants |  |
| Exceptions |  |

| **Scenario 5.1** | **User sees all categories** |
| --- | --- |
| Precondition | User is logged in |
| Post condition | User sees all categories |
| Step# | Description |
| 1 | User clicks the button to view all categories |
| 2 | System checks if the user is logged in |
| 3 | System returns the information of all categories |

Use case 6, UC6, **Get labels**

| **Actors Involved** |  |
| --- | --- |
| Precondition | User is logged in |
| Post condition | User sees all labels |
| Nominal Scenario | The user wants to see all labels |
| Variants |  |
| Exceptions |  |

| **Scenario 6.1** | **User sees all labels** |
| --- | --- |
| Precondition | User is logged in |
| Post condition | User sees all labels |
| Step# | Description |
| 1 | User clicks the button to view all labels |
| 2 | System checks if the user is logged in |
| 3 | System returns the information of all labels |

Use case 7, UC7, **Create a new category**

| **Actors Involved** |  |
| --- | --- |
| Precondition | User is logged in |
| Post condition | User creates the new category |
| Nominal Scenario | The user wants to create a new category |
| Variants |  |
| Exceptions |  |

| **Scenario 7.1** | **User can create a new category** |
| --- | --- |
| Precondition | User is logged in |
| Post condition | User creates the new category |
| Step# | Description |
| 1 | User clicks the button to create a new category |
| 2 | System asks for type and color of the category |
| 3 | User enters the type and the color |
| 4 | System checks if the user is logged in |
| 5 | System creates the new category |

Use case 8, UC8, **CRUD actions on transactions**

| **Actors Involved** |  |
| --- | --- |
| Precondition | User is logged in |
| Post condition | User creates, reads, deletes transactions |
| Nominal Scenario | The user wants to create, read, delete transactions |
| Variants |  |
| Exceptions |  |

| **Scenario 8.1** | **User creates a new transaction** |
| --- | --- |
| Precondition | User is logged in |
| Post condition | User creates the new transaction |
| Step# | Description |
| 1 | User clicks the button to create a new transaction |
| 2 | System asks for the name, amount and type of the transaction |
| 3 | User enters the name, amount and type of the transaction |
| 4 | System checks if the user is logged in |
| 5 | System creates the new transaction |

| **Scenario 8.2** | **User reads a transaction** |
| --- | --- |
| Precondition | User is logged in |
| Post condition | User reads the transaction |
| Step# | Description |
| 1 | User clicks the button to read a transaction |
| 2 | System asks for the name, the type, the amount and the date of the transaction |
| 3 | User enters the name, the type, the amount and the date of the transaction |
| 4 | System checks if the user is logged in |
| 5 | System returns the transaction |

| **Scenario 8.3** | **User deletes a transaction** |
| --- | --- |
| Precondition | User is logged in |
| Post condition | User deletes the transaction |
| Step# | Description |
| 1 | User clicks the button to delete a transaction |
| 2 | System asks for the name, the type, the amount and the date of the transaction |
| 3 | User enters the name, the type, the amount and the date of the transaction |
| 4 | System checks if the user is logged in |
| 5 | System deletes the transaction |

**9 Glossary**

**Immagine che contiene diagramma

Descrizione generata automaticamente**

**10 System Design**

**Immagine che contiene diagramma

Descrizione generata automaticamente**

**11 Deployment Diagram**

**Immagine che contiene diagramma

Descrizione generata automaticamente**