

**School of Arts and Sciences**

**Computer Science Department**

**CSCI380 – Software Engineering**

**Group # 11 - Project:**

Money Management System

|  |  |
| --- | --- |
| **Prepared by:** | **Presented to:** |
| Hicham Baydoun – 11735296 | Miss Amal El Arid |
|  |  |
|  |  |
|  |  |

Summer 2018-2019

**Request for Information System Services**

|  |  |
| --- | --- |
| **Executive Sponsor:** | Sarah Hartman |
| **Organization:** | Telecom |
| **Title:** | Vice President |

|  |  |
| --- | --- |
| **Submitted by:** | Hicham Baydoun |
| **Title:** | President |
| **Request Date:** | July 1, 2010 |

**Type of Service Requested:**

|  |  |
| --- | --- |
| * **New** Development | * **Existing** Application Enhancement |
| * Business Process Analysis and **Redesign** | * Existing Application **Maintenance** |

**Brief Company Background and History:**

Our company was established by its founder Hicham Baydoun in 1998. The company started with only 1 branch in Beirut, and now it has over 5 branches throughout Lebanon. The company aims to help people by creating solutions for people’s problems through software.

**Brief Statement of Problems, Opportunities, or Directives**:

The main objective of this software is to solve the struggle faced by people in managing their financial assets, so some of problems that people face are:

**•** Managing their money and setting a budget.

**•** Tracking their daily spendings.

**•** Comparing their daily and monthly spendings.

**•** Running out of money or finishing their income before the month ends.

**•** Finding it hard to save up money.

**Brief Statement of Expected Solution:**

The main solution of this system is to help people to manage their money monthly and daily. Using the solutions this system offers, the user can now:

**•** Set budgets to items they purchase and use (e.g. Food, home supplies, clothes).

**•** Document their daily purchases (item and price) in order to trach their spending.

**•** Compare monthly/daily spending and expenses by viewing statistics (pie chart and histogram) made by the system.

**•** Spend their money wisely throughout the month using the managing service on the system.

**•** Save up money.

**Deliverability**: 12 months

**Phase 1:**

**Requirements Analysis**

**Context Diagram (Draw using Visio Software)**

****

**Tentative List of Requirements**

**Functional Requirements: (Minimum of 20 complete requirements)**

|  |
| --- |
| 1. The user creates an account by registering on the system |
| 1. The user can login into his/her account |
| 1. The user chooses the currency that he/she uses |
| 1. The user can chooses the language he/she wants |
| 1. The user can create (add) a wallet (e.g. Food wallet, Technology wallet, Clothes wallet) by entering wallet name and balance (budget) |
| 1. The user can create (add) an item (e.g. chicken sandwich, Samsung phone, t-shirt) |
| 1. The user can document a transaction by selecting a wallet, an item, and enter the price of the item; a transaction is when the user buys something in real life he documents it on the system to track his/her spendings |
| 1. The system subtracts the price of an item from the selected wallet when a transaction is made |
| 1. The user can view his/her daily transactions by showing the item purchased, its price, and from to wallet the item belongs to on the home page |
| 1. The user can view on the home page the total balance, which is the sum of all the balances of the wallets, in other words; the user can see how much money he still has from his total balance, that he did not yet spend this month |
| 1. The user can use the calendar on the system and select any day to see all transactions made that day |
| 1. The user can read instructions on how to use the app by selecting/pressing the info button found on the home page; this requirement ensures a user-friendly experience |
| 1. The user can view statistics (pie chart and histogram) about his spending on the statistics page |
| 1. The user can compare his spending by days, weeks, or months using the histogram |
| 1. The user can compare his daily spendings per item using the pie chart (in percentage form) |
| 1. The user can delete wallets, items, and transactions |
| 1. The user can edit the name and balance of a wallet at any time |
| 1. The user can edit the name of an item at any time |
| 1. The user can view a progress bar on the home page, that indicates how much he has spent from his total balance |
| 1. The user receives daily notifications from the system reminding him/her to document transactions in order to manage his money |
| 1. The user receives a notification when he reaches the wallet’s balance limit(he/she cannot spend anymore from this wallet) |
| 1. The user receives a notification each time he/she adds a wallet or an item, or makes a transaction |
| 1. The user receives daily tips on how to spend his money wisely |
| 1. The system automatically updates the balances and statistics when a change is made by the user |

(You can add as much as you need)

**Non-Functional Requirements: (Minimum of 5 requirements)**

|  |
| --- |
| 1. The system supports 3 languages: English, French, and Arabic |
| 1. The system is available any-time and does not need internet access (availability) |
| 1. The system restricts unregistered or unauthorized users from using it (security) |
| 1. Only the user can edit or add wallets, items, and transactions (security) |
| 1. The interface must be easy to use; it is user-friendly (usability) |
| 1. The system is always responsive and has a fast performance and response, it should pop up notifications in less than 2 seconds after the user’s action (performance) |
| 1. The system is updated and maintained regularly (maintainability) |
| 1. The system supports more than 300 transactions per day (performance) |
| 1. The system should be portable, so the best platform to implement the system is a mobile application (usability and performance) |

(You can add as much as you need)

**Phase 2:**

**Use-Case**

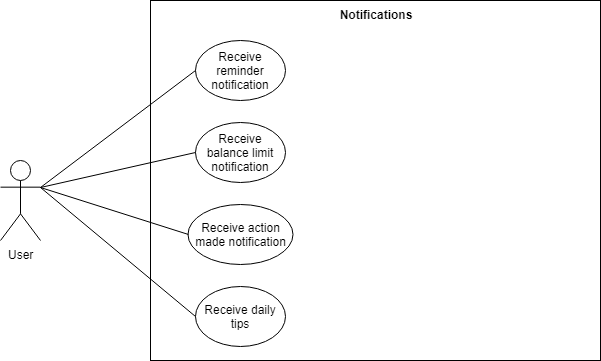
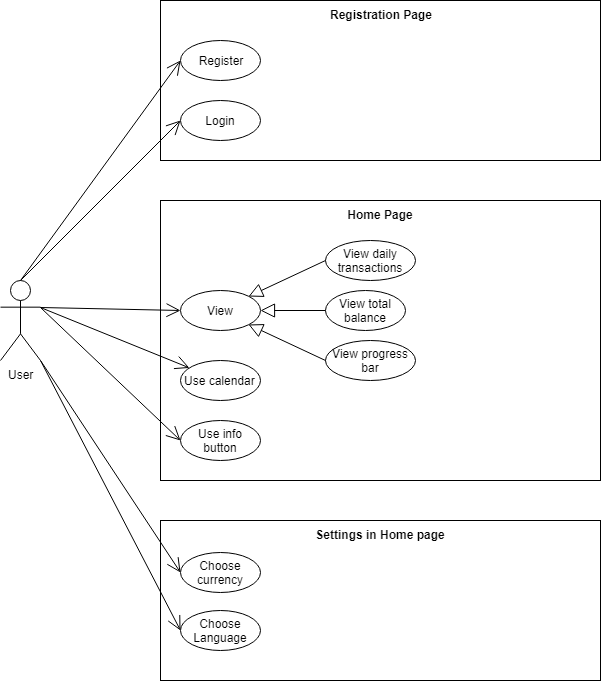
**Analysis**

**Modeling System Requirements (Use-Case Glossary)**

|  |  |  |
| --- | --- | --- |
| **Use-Case Glossary** | | |
| **Use-Case Name** | **Use-Case Description** | **Participating Actors** |
| Register | The user creates an account by registering on the system | * User (primary business and primary system actor) |
| Login | The user can login into his/her account | * User (primary business and primary system actor) |
| Choose currency | The user chooses the currency that he/she uses | * User (primary business and primary system actor) |
| Choose language | The user can chooses the language he/she wants | * User (primary business and primary system actor) |
| Add wallet | The user can create (add) a wallet (e.g. Food wallet, Technology wallet, Clothes wallet) by entering wallet name and balance (budget) | * User (primary business and primary system actor) |
| Add item | The user can create (add) an item (e.g. chicken sandwich, Samsung phone, t-shirt) | * User (primary business and primary system actor) |
| Add transaction | The user can document a transaction by selecting a wallet, an item, and enter the price of the item; a transaction is when the user buys something in real life he documents it on the system to track his/her spendings | * User (primary business and primary system actor) |
| View daily transactions | The user can view his/her daily transactions by showing the item purchased, its price, and from to wallet the item belongs to on the home page | * User (external receiver) |
| View total balance | The user can view on the home page the total balance, which is the sum of all the balances of the wallets, in other words; the user can see how much money he still has from his total balance, that he did not yet spend this month | * User (external receiver) |
| Use calendar | The user can use the calendar on the system and select any day to see all transactions made that day | * User (primary business and primary system actor) |
| Use info button | The user can read instructions on how to use the app by selecting/pressing the info button found on the home page; this requirement ensures a user-friendly experience | * User (primary business and primary system actor) |
| Access statistics of expenses | The user can view statistics (pie chart and histogram) about his spending on the statistics page | * User (primary business and primary system actor) |
| Compare expenses | The user can compare his spending by days, weeks, or months using the histogram.  The user can compare his daily spendings per item using the pie chart (in percentage form) | * User (primary business and primary system actor) |
| Delete | The user can delete wallets, items, and transactions | * User (primary business and primary system actor) |
| Edit wallet | The user can edit the name and balance of a wallet at any time | * User (primary business and primary system actor) |
| Edit item | The user can edit the name of an item at any time | * User (primary business and primary system actor) |
| View progress bar | The user can view a progress bar on the home page, that indicates how much he has spent from his/her total balance | * User (external receiver) |
| Receive reminder notifications | The user receives daily notifications from the system reminding him/her to document transactions in order to manage his money | * User (external receiver) |
| Receive balance limit notification | The user receives a notification when he reaches the wallet’s balance limit(he/she cannot spend anymore from this wallet) | * User (external receiver) |
| Receive action made notification | The user receives a notification each time he/she adds a wallet or an item, or makes a transaction | * User (external receiver) |
| Receive daily tips | The user receives daily tips on how to spend his money wisely | * User (external receiver) |

(You can add as much as you need)

**Modeling System Requirements (Use-Case Model Diagram)**

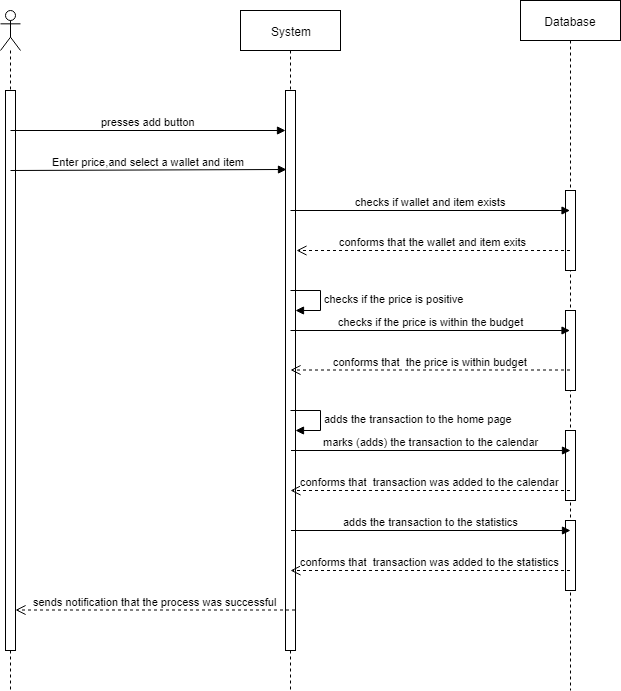


**Modeling System Requirements (Use-Case Narrative)**

(Write the Use-Case Narrative for ONE major Use-Case)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use-Case Name:** | Add Transaction | **Use-Case Type:** | | Business Requirement |
| **Priority:** | High | | | |
| **Source:** | Requirement-MMS-R6.00 | | | |
| **Primary Business Actor:** | User | | | |
| **Other Participating Actors:** | None | | | |
| **Other Interested Stakeholders:** | None | | | |
| **Description:** | The use case describes the event of when a person purchases an item in real life; he/she can now document that transaction on the system. The action is done by pressing the add button, and then chooses to add a transaction. The user must select a wallet and item that belongs to the transaction, and enter the price. Therefore, the system subtracts the price from the wallet and adds the transaction to the home page. | | | |
| **Precondition:** | The person adding a transaction must be registered on the system. | | | |
| **Trigger:** | This use case is initiated when the user presses the add button. | | | |
| **Typical Course of Events:** | **Actor Action** | | **System Response** | |
| **Step 1:** The user presses the add button, chooses to add a transaction. In addition, the user next selects a wallet, an item, and enters the price (amount paid) of the corresponding transaction. | | **Step 1:** The system takes the input, and checks if the selected wallet and item exists.  **Step 2:** The system checks if the price is a positive number.  **Step 3:** The system checks if the price is within the balance (budget) of the wallet.  **Step 4:** The system subtracts the price from the current wallet’s balance.  **Step 5:** The system adds the transaction to the home page, and displays its information (item name, price, its corresponding wallet).  **Step 6:** The system marks (adds) the transaction to the day it was made on, on the calendar.  **Step 7:** The system includes the transaction to the statistics (pie chart and histogram), and then recalculate the statistics with this new addition.  **Step 8:** The system sends a notification to confirm that the addition process was successful. | |
| **Alternate Courses:** | **Alt-Step 1:** If the wallet or item does not exist, the system asks the user to add (create) that wallet and item then retry the transaction.  **Alt-Step 2:** If the price is negative, the system notifies the user that the price cannot be negative, and asks him/her to retry again.  **Alt-Step 3:** If the price if greater than the current wallet’s balance, the system notifies the user that the transaction addition failed because the price is greater than the balance. | | | |
| **Conclusion:** | The use case concludes when the user receives a confirmation that the addition was successful. | | | |
| **Postcondition:** | The transaction has been added and documented. | | | |

**Modeling System Requirements (Sequence Diagram)**



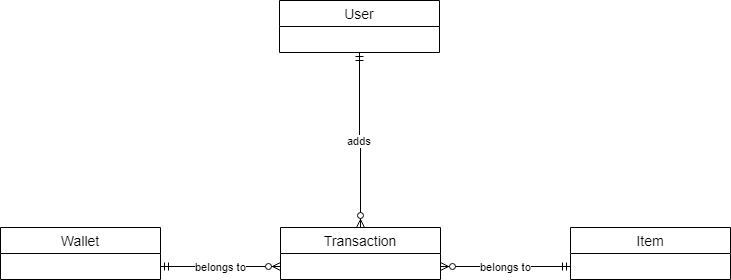
**Phase 3:**

**Data**

**Modeling**

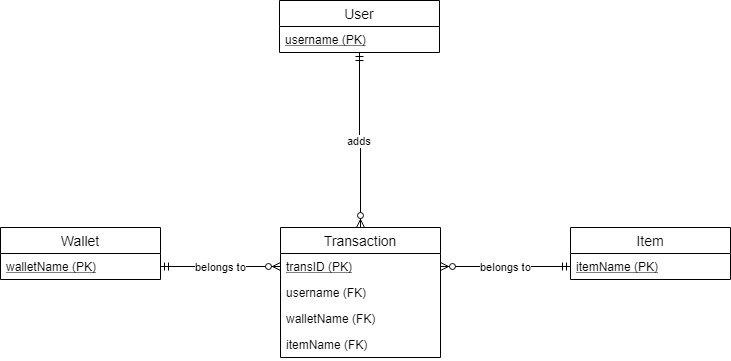
**Context Data Model**

(It shows the entities and the entities relationships without any additional information)



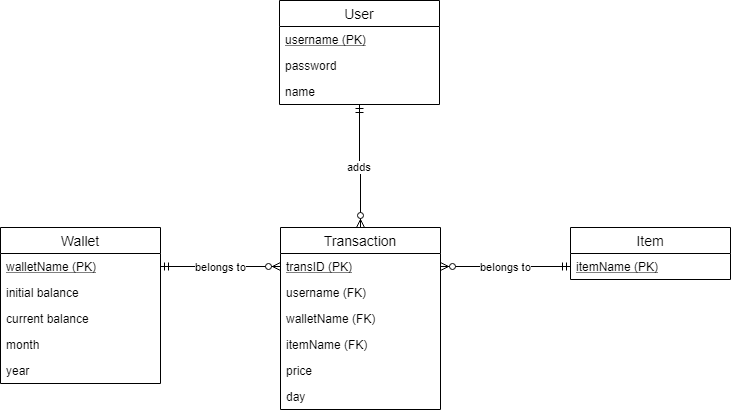
**Key-Based Data Model**

(It shows the entities and the primary keys)



**Fully-Attributes Data Model**

(It shows the entities, the relationships, the primary keys, and all attributes including the foreign keys)



**Phase 4:**

**Process Modeling**

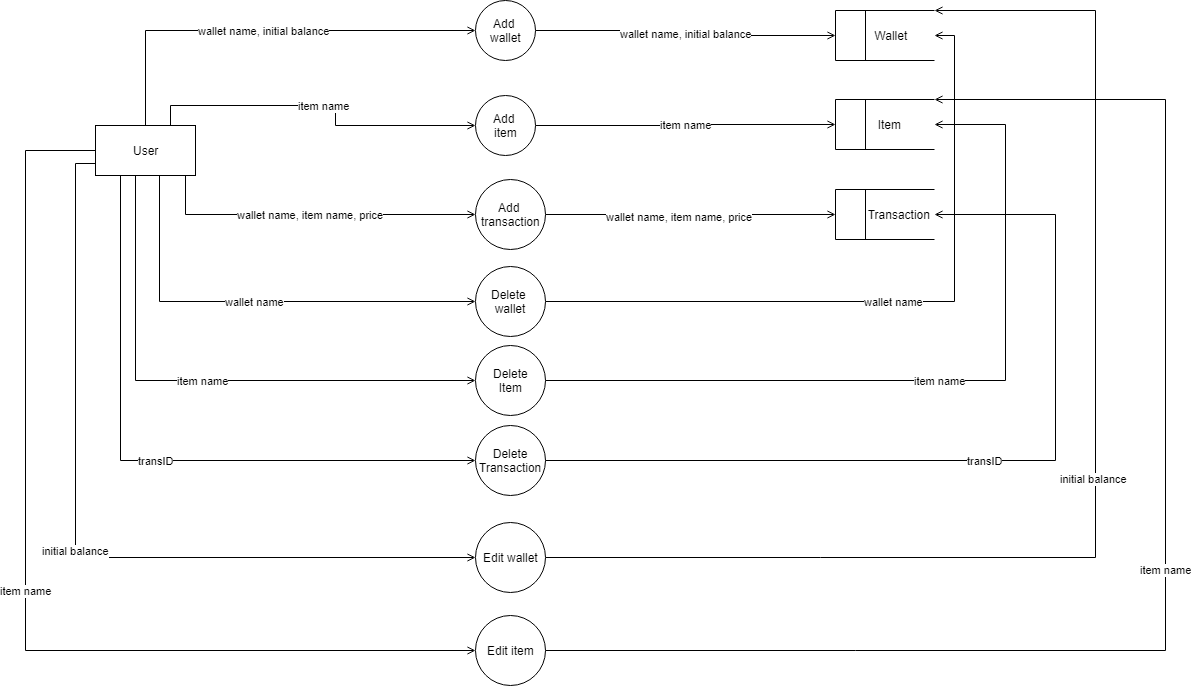
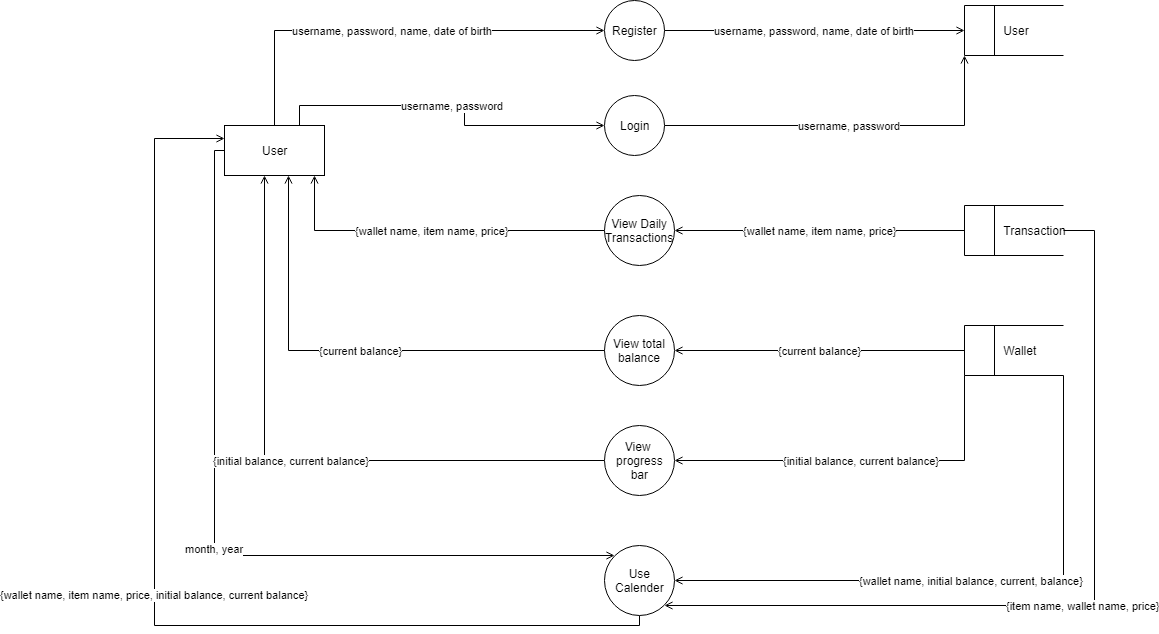
**Event Response (Use Case List)**

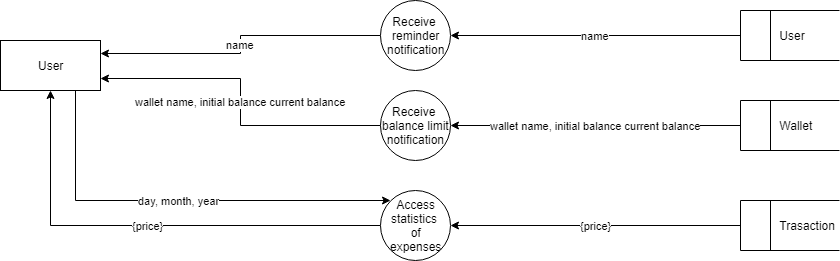
(It shows the business events to which the system must provide a response)

|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | **Event (Use-Case)** | **Trigger (Input(s))** | **Responses (Output(s))** |
| User | Register | username, password, name, date of birth | The system creates a new account in the User database |
|  | Login | Username, password | The system confirms and validates the username and password |
|  | View Daily Transactions | wallet name, item name, price | The system displays all daily transactions, which the user can view |
|  | View total balance | current balance | The system generates the total balance by adding all current balances, which the user can view |
|  | View progress bar | Initial balance, current balance | The generates a progress bar from the balances which the user can view |
|  | Use calendar | month, year, day, wallet name, item name, price, initial balance, current balance | The system logically selects all transactions and wallet the belongs to a specific date, which the user can view |
|  | Add wallet | wallet name, initial balance | The system creates a new wallet and adds it to the wallet database |
|  | Add item | Item name | The system creates a new item and adds it to the item database |
|  | Add transaction | wallet name, item name, price | The system creates a new transaction and adds it to the transaction database |
|  | Delete wallet | wallet name | The system deletes the wallet that the user selected |
|  | Delete item | Item name | The system deletes the item that the user selected |
|  | Delete transaction | transID | The system deletes the transaction that the user selected |
|  | Edit wallet | Initial balance | The system updates the wallet’s initial balance according the value entered by the user |
|  | Edit item | Item name | The system updates the item name of an item in the item database |
|  | Receive reminder notification | name | The system send the user a reminder notification that has the users name in it, in order to remind him to use the app |
|  | Receive balance limit notification | wallet name, initial balance, current balance | The system send the user a notification when the user reaches the balance limit of a wallet |
|  | Access statistics if expenses | day, month, year, price | The system generates a histogram comparing the daily amount spend (prices) |

(You can add as much as you need)

**Full System Data Flow Diagram**



****