MrPenn

REQUIREMENTS SPECIFICATION DOCUMENT

Revision 1 – 15/02/2020

Abbo Giulio Antonio

Contents

[1 Introduction 3](#_Toc32692452)

[1.1 Purpose 3](#_Toc32692453)

[1.2 Scope 3](#_Toc32692454)

[1.2.1 Goals 3](#_Toc32692455)

[1.3 Overview 4](#_Toc32692456)

[1.3.1 Context – Product perspective 4](#_Toc32692457)

[1.3.2 Functions – Requirements 4](#_Toc32692458)

[1.3.3 User characteristics 4](#_Toc32692459)

[1.3.4 Assumptions and dependencies – Limitations 5](#_Toc32692460)

[1.4 Definitions 5](#_Toc32692461)

[1.5 Acronyms and abbreviations 5](#_Toc32692462)

[1.6 Revision history 5](#_Toc32692463)

[2 References 6](#_Toc32692464)

[3 System requirements 7](#_Toc32692465)

[3.1 Functional requirements 7](#_Toc32692466)

[3.2 Interface requirements 7](#_Toc32692467)

[3.2.1 User interfaces requirements 7](#_Toc32692468)

[3.2.2 Hardware interfaces requirements 7](#_Toc32692469)

[3.2.3 Software interfaces requirements 7](#_Toc32692470)

[3.2.4 Communication interfaces requirements 7](#_Toc32692471)

[3.3 Software system attributes 7](#_Toc32692472)

[3.3.1 Reliability 7](#_Toc32692473)

[3.3.2 Availability 7](#_Toc32692474)

[3.3.3 Security 7](#_Toc32692475)

[3.3.4 Portability 7](#_Toc32692476)

[3.3.5 Maintainability 7](#_Toc32692477)

[3.4 Physical characteristics 7](#_Toc32692478)

[3.5 Policies and regulations 8](#_Toc32692479)

[3.6 Packaging, handling, shipping and transportation 8](#_Toc32692480)

# Introduction

## Purpose

The purpose of the MrPenn system is to allow its users to store their money transactions, then they will have access to their personal balance and some personal statistics or graphs based on this data. All the data is personal and confidential, thus is not shared between users.

## Scope

The main function is the possibility to add new transactions; a transaction should contain an amount (positive), an origin and a destination (this will be called entities), one or more categories, a date and some optional notes.

A transaction can be returned, in this case, it will contain a flag that identifies this behaviour; when returned, it will contain a reference to the transaction that returned it.  
For example, Alice gives Bob some money and expects it back, this transaction is T1. Then transaction T2 represents the fact that Bob gave back the money. T1 and T2 should have the same categories, T1 will hold a field that states that had to be returned and a reference to T2. T2 is not to be returned.

The user can see a list of all the transaction that have to be returned but are not yet. He can open a precompiled form to insert the returning transaction.

All the previous transactions are accessible and can be modified and deleted.

The user can see the balance for some preferred entities he chooses. When an entity is chosen the user is asked for the initial balance, then all transactions that have it as a source (destination) are subtracted (added) to it. It can be useful to keep this partial data (the balances for each entity and category, even if not requested) and update on each insertion.

Some of the preferred entities can be grouped in a “total” balance. This is shown together with the other entities wherever they appear, but transaction can not have it as an origin or destination.

Each category can be set as “positive” or “negative”, indicating whether the transactions associated with it are incomes or outgoings.

Some of the statistics include a graph with preferred entities balances by month, the partial monthly balances for the preferred entities and categories, the partial up-to-this-month balances. Statistics should be easy to expand in the future.

Future possible function: maximum budget for each category; keep track of when the user has never spent so much in a category for a month; regular transactions; cryptography of the data.

### Goals

The system has the following goals:

* G1: The system allows users to privately store their transactions.
* G2: Each user can see, update and delete each of his transactions.
* G3: The user can request statistics based on his transactions.
  + Transaction that have not been returned.
  + Total balances for each entity.
  + Partial monthly balances for each entity and category.

## Overview

### Context – Product perspective

The system will need a front end and a back end. The front end is not secure, nor will be always updated: the system must maintain backward compatibility. The back end is always up to date and implements all the necessary security checks.

The data stored is private and not shared between the users.

### Functions – Requirements

* R1: The user is prompted to log in if necessary
* R2: The user can sign up to the system
* R3: The user can add a new transaction with:
  + Id
  + Positive amount
  + Origin and destination entities
  + Date
  + Optionally one or more categories
  + Optional notes
  + Whether must be returned
  + Optional returning id
* R4: The backend check that the new transaction is correct and stores it
* R5: For each transaction, the value is subtracted to the source entity
* R6: For each transaction, the value is added to the destination entity
* R7: For each transaction, the value is added to its categories
* R8: The data is only accessible by the user that created it
* R9: The user can see, update and delete each of his transactions
* R10: The user can specify preferred entities
* R11: The user can specify preferred categories
* R12: The user can add some of the preferred entities to a “total”, that is their sum
* R13: The “total” is shown with other entities in the statistics, but cannot exist in transactions
* R14: When an entity is set as preferred, the user is asked for its initial balance
* R15: Each category allows to specify if its value is positive or negative
* R16: The user can request the monthly partial balances for each entity
* R17: The user can request the monthly partial balances for each category
* R18: The user can request the up-to-this-month balances for each entity
* R19: The user can request the up-to-this-month balances for each category
* R20: The user can request the total balances for each entity
* R21: The user can see some statistics:
  + Transaction that have not been returned
  + Graph with preferred entities balances by month
  + Partial monthly balances for the current and previous month, for preferred entities
  + Partial monthly balances for the current and previous month, for preferred categories
  + Total balance for the preferred entities
  + Graph with the monthly spending for each category

### User characteristics

To use the system, the user has access to a device and to the MrPenn app.

Each user is an individual, if an individual manages to sign up with two accounts, these will be handled by the system as if they belonged to two distinct individuals.

The user is responsible for his actions, for example the system does not protect the data if he shares his password.

### Assumptions and dependencies – Limitations

* D1: The system relies on the abstraction it is built on for the security of the connection.
* D2: The user has a device and the MrPenn app.
* D3: The user is connected to the internet most of the time.
* D4: The external services behave correctly.

## Definitions

* System: the software to be.
* User: an individual that uses the app.
* Transaction: an exchange of money from a certain origin to a certain destination.
* (Personal) balance: the sum of all the transactions for an entity or category.
* Entity: the source or destination of the money in a transaction.
* Category: labels for the transactions.
* To return: a sum of money is to be returned if it must be given back eventually.
* Initial balance: for an entity, the initial sum of money, used to produce statistics.
* Total: a special entity that is the sum (in the context) of other selected entities.
* Positive or negative category: whether the transactions in this category are incomes or outgoings.

## Acronyms and abbreviations

* Gn: goal number n.
* Rn: requirement number n.
* Dn: domain assumption number n.
* Un: use case number n.
* UIn: user interface requirement number n.
* On: other requirement number n.

## Revision history

* Version 1: first version.

# References

The structure of this document is inspired on those proposed in IEEE 29148.

Material Design by Google.

GDPR regulations.

# System requirements

## Functional requirements

To provide the required functionalities, the following use case are kept into account.

* U1: User sign up.
* U2: User login.
* U2: New transaction.
* U3: Request transactions.
* U4: Request statistic data.

## Interface requirements

### User interfaces requirements

* UI1: The UI adheres for the most part to the Material guidelines, especially for usability.
* UI2: A transaction that must be returned allows to open a precompiled form to insert the returning transaction.
* UI3: When adding a transaction, the user can specify which transaction is returning.

### Hardware interfaces requirements

The system does not use directly hardware interfaces.

### Software interfaces requirements

The system does not have to use software interfaces directly.

For what concerns the data storage, the system will have for each user: the settings (preferred…), the transactions, some cached data.

### Communication interfaces requirements

The communication between front end and back end is secure.

## Software system attributes

### Reliability

The front end should never crash because of bugs in the code, it can crash because of underlying problems with the abstraction levels. The back end should have a high reliability.

### Availability

The back end should strive for at least a 99.9% availability.

### Security

The data must be secure: the front end is protected by the login; the communication must be secure. The back end must not trust the front end and implement additional security measures.

### Portability

* O1: The data must be portable.

### Maintainability

* O2: The back end is always backward-compatible (from the viewpoint of the front end).

## Physical characteristics

The system does not need to survive specific extreme environmental conditions.

## Policies and regulations

The data of the users is confidential and must not be available to the public or to other users.

* O3: The system will adhere to the principal GDPR guidelines.

## Packaging, handling, shipping and transportation

The back end will be accessible through an internet connection. The front end will be distributed with an APK or other suitable methods.