MrPenn

REQUIREMENTS SPECIFICATION DOCUMENT

Revision 1 – 12/02/2020

Abbo Giulio Antonio

Contents

[1 Introduction 3](#_Toc32436353)

[1.1 Purpose 3](#_Toc32436354)

[1.2 Scope 3](#_Toc32436355)

[1.3 Overview 3](#_Toc32436356)

[1.3.1 Context – Product perspective 3](#_Toc32436357)

[1.3.2 Functions 3](#_Toc32436358)

[1.3.3 User characteristics 3](#_Toc32436359)

[1.3.4 Assumptions and dependencies – Limitations 3](#_Toc32436360)

[1.4 Definitions 3](#_Toc32436361)

[1.5 Acronyms and abbreviations 3](#_Toc32436362)

[1.6 Revision history 3](#_Toc32436363)

[2 References 4](#_Toc32436364)

[3 System requirements 5](#_Toc32436365)

[3.1 Functional requirements 5](#_Toc32436366)

[3.2 Interface requirements 5](#_Toc32436367)

[3.2.1 User interfaces requirements 5](#_Toc32436368)

[3.2.2 Hardware interfaces requirements 5](#_Toc32436369)

[3.2.3 Software interfaces requirements 5](#_Toc32436370)

[3.2.4 Communication interfaces requirements 5](#_Toc32436371)

[3.3 Software system attributes 5](#_Toc32436372)

[3.3.1 Reliability 5](#_Toc32436373)

[3.3.2 Availability 5](#_Toc32436374)

[3.3.3 Security 5](#_Toc32436375)

[3.3.4 Portability 5](#_Toc32436376)

[3.3.5 Maintainability 5](#_Toc32436377)

[3.4 Physical characteristics 5](#_Toc32436378)

[3.4.1 Physical requirements 5](#_Toc32436379)

[3.4.2 Adaptability requirements 5](#_Toc32436380)

[3.4.3 Environmental requirements 5](#_Toc32436381)

[3.5 Policies and regulations 5](#_Toc32436382)

[3.6 Packaging, handling, shipping and transportation 5](#_Toc32436383)

# 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.

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; modular approach to the statistics; regular transactions.

### Goals

## Overview

### Context – Product perspective

Describe the world, block diagram if exists a major system

### Functions

Requirements

### User characteristics

### Assumptions and dependencies – Limitations

## Definitions

## Acronyms and abbreviations

## Revision history

# References

# System requirements

## Functional requirements

Use case, sequence diagrams, mapping. Modes and states

## Interface requirements

### User interfaces requirements

### Hardware interfaces requirements

### Software interfaces requirements

Logical database requirements

### Communication interfaces requirements

## Software system attributes

### Reliability

### Availability

### Security

### Portability

### Maintainability

## Physical characteristics

### Physical requirements

### Adaptability requirements

### Environmental requirements

## Policies and regulations

## Packaging, handling, shipping and transportation