easyFinance Ltd.

**easyFinance
Software Requirements Specification
For the Backend**

Version 1.0

Revision History

Date	Version	Description	Author
10/16/2019	0.9	First draft	Nico Rahm
10/20/2019	1.0	Final	Nico Rahm

Table of Contents

Software Requirements Specification

1. Introduction

easyFinance is a HTML based finance management app using a relational SQL database as Backend. It is used to store all information about the payments the user did. In this document all requirements for the backend will be specified.

1.1 Purpose

The backend of easyFinance is used to store all information about the payments and transactions a user did. All information will be labeled with a timestamp and category to sort and get them into charts to visualize the actual finance status.

1.2 Scope

The backend includes the database and the calculations that will be needed to visualize the status and other graphs.

1.3 Definitions, Acronyms, and Abbreviations

Database: SQL database

1.4 References

This document refers to the SRS of the Frontend.

1.5 Overview

In the following all technical requirement will be specified. It is organized per

2. Overall Description

The easyFinance app is intended to make an easy and simple handling with one's income and spent money possible. One can manually and (therefore) individually enter each spending or transaction and assign it to categories which are created by the user himself. Equal to that, it is also possible to enter every income.

With that data, the app automatically creates graphs, tables, every kind of chart and even notifications – due to the user's wish.

In general, easyFinance gives the user an individual overview about his/her finance and optimizes the money-spending-behavior, showing the user where to save money.

3. Specific Requirements

3.1 Functionality

3.1.1 Database

The database will be saved on the clients device so all data belonging to the user remains in his own hand. The database contains all information about every payment the user did. It is possible to sort the payments by time, priority and amount.

3.1.2 Calculating Unit

The calculating Unit will read content from the Database and serve those to the Frontend for view single payments. Also it will calculate information for the grapes and visualization.

3.2   Usability

It is not allowed that any user has access to the backend, so this information will be precised in the SRS of the frontend.

3.3   Reliability

3.3.1   Availability

The backend and database needs to be available 99.0% of the year. This makes a maximum downtime of 3.65 days per year.

3.3.2   Mean Time Between Failures

The MTBF is specified to at least 720h.

3.3.3   Mean Time To Repair

The maximum MTTR is guaranteed to 24h.

3.4   Performance

3.4.1   Capacity

The Database should be abled to handle up to 4,000 entries with an max. age of three years.

3.4.2   Resources

The database should not use more than 100 MB in the non volatile memory and 20 MB in the RAM.
The calculating unit is not allowed to use more than 10 MB in the non volatile memory and RAM.

3.5   Supportability

3.5.1   Comments

The code needs to be well documented with comments per class and function as well an interfaces and key lines.

3.5.2   Documentation

There will be a documentation about all functions and their position in the whole code.

3.6   Design Constraints

3.6.1   Database

As database a relational SQL database will be used.

3.6.2   Language

As coding language JavaScript and TypeScipt will be used.

3.6.3   Architecture

The Backend will be only logical separated from the frontend. The HTML code from frontend and script code from the Backend will be implemented in the same files.

[The requirement description goes here.]

3.7   On-line User Documentation and Help System Requirements

See in the SRS of the Frontend,

3.8   Purchased Components

There will be no Purchased Components. All used frameworks and languages are open source based.

3.9 Interfaces

3.9.1 User Interfaces

See in the SRS of the Frontend.

3.9.2 Hardware Interfaces

For the Backend no dedicated hardware interface is needed.

3.9.3 Software Interfaces

There needs to be a SQL interface between the calculating unit and the SQL database. Also there must be an interface that enables the calculating Unit to transfer data to the frontend.

3.9.4 Communications Interfaces

There is no communication between any devices needed

3.10 Licensing Requirements

easyFinance is available under free License. It will be financed by ad.

3.11 Legal, Copyright, and Other Notices

easyFinance is copyright by easyFinance Ltd. in 2019