**Requirements Analysis Document**

Purpose

The results of the requirements elaboration and the analysis activities are documented in the Requirements Analysis Document (RAD). This document completely describes the system in terms of functional and nonfunctional requirements and serves as a contractual basis between the client and the developers.

Table of Contents

1. Introduction (Matthias)

1.1 Purpose of the system

1.2 Scope of the system

1.3 Objectives and success criteria of the project

1.4 Definitions, acronyms, and abbreviations

2. Proposed system

2.1 Overview (Yvonne)

2.2 Functional requirements (Dario)

2.3 Nonfunctional requirements (Dario)

2.4 System models (Bendix + Valentin)

3. Glossary

**1. Introduction**

Content: The Introduction provides a brief overview of the function of the system and the reasons for its development, its scope, and references to the development context. The introduction also includes the objectives and success criteria of the project.

* 1. **Purpose of the system**

The systems purpose lies in the organization of private end users’ finances. The system helps structure and categorize expenditures and gives direct feedback through the budgets that the user can set for certain categories of payments.

* 1. **Scope of the system**

The system allows the user to:

* View his balance and transactions
* Create custom categories for expenditures
* Analyse expenditures through automatic categorization
* Create budgets automatically based off transactions
* Create custom budgets
* Receive notifications when the budget limit is reached
  1. **Objectives and success criteria of the project**

The project is successful when a proof of concept has been built. The MVP for the project is an application, that shows how the basic functionalities of the app are supposed to work. Within the project, the application is not built completely, but rather a mock-application to show the use cases is built.

* 1. **Definitions, acronyms, and abbreviations**

MVP: Minimal viable product

**2. Proposed system**

This section documents the requirements elicitation and the analysis model of the new system.

**2.1 Overview** -The overview presents a functional overview of the system.

After starting the app for the first time, the user is taken to the start screen where he can either register or log in. When registering, the user must enter his standard personal data (name, last name, email, etc.), connect the app to his bank account and set his monthly income as the total budget. Using the monthly saving rate the user can set how high the total expenses per month may be. On the main page the user can see his expanses organized in different categories. The user can set individual budgets for all categories and sees on the main page as well how much money he has left to spend for each category.

**2.2 Functional requirements**

Functional requirements describe the high-level functionality of the system. This section list all functional requirements and additionally presents the dependencies between them.

Functional Requirements (specific for the product):

* Register and Login are necessary (including financial account details)
* App has access to the account balances
* App has real time overview of all transactions made
* App categorizes the customers expenses
* Provides selection of a monthly overall budget or specific for each category
* Within a category the app displays the date of the last spending, the budget and the free amount
* App checks your financial budget when paying
* It reminds you with a push notification if you are exceeding your preselected budget

**2.3 Nonfunctional requirements**

Nonfunctional requirements describe user-level requirements that are not directly related to functionality. This includes usability, reliability, performance, supportability, implementation, interface, operational, packaging, and legal requirements. The section list all these non-functional requirements and additionally presents the dependencies between them.

Non-Functional requirements (unspecific for the product):

* Usability:
  + Easy to understand
  + User become familiar with the app’s functions in a short time
* Reliability:
  + The system should be reachable 24/7
* Performance:
  + Short loading time (e.g. for the starting screen)
* Responsiveness:
  + Fast response to user inputs
  + Save state when app gets interrupted
* Compatibility and Availability:
  + Android
  + Googles Play Store
* Scalability:
  + App can handle high amount of user data without delay
* Interface:
  + The user interface should allow for intuitive interactions
  + Minimalistic
  + Responsive design (screen adaption)
* security:
  + System blocks login after three failed attempts
  + Fixes and security updates
  + Data protection is a priority

**2.4 System models**

The System models include scenarios, use cases, and the analysis object model for the system. This section should contain the complete functional specification, including mock-ups, paper-based prototypes or storyboards illustrating the user interface of the system and navigational paths representing the sequence of screens.

**2.4.1 Scenarios**

**2.4.2 Use case model**

**2.4.3 Analysis object model**

**2.4.4 User interface mockups**

**3. Glossary**

A glossary of important terms used in the project and in the system model ensures consistency in the specification and a common understanding of terms used by the client.