Roger That Project

Software Engineering

Requirements Document

Monday 14th June, 2021

Authors:
Gasan Rzaev
(s3553213)
Valeria Mavceanscaia
(s3673952)
Denis Garabajiu
(s4142551)
Constantin Cainarean
(s4142152)

Lecturer:
A. Capiluppi
Teaching Assistant
D. Plamadeala

Contents

1 Introduction

An actual economical and social problem in the Netherlands is a relatively big number of people, especially young adults (18 - 35 y.o) with debts or bad financial situations. A lot of people do not know about some income schemes they can use, also how to deal with taxes or how to organize expenses. People with debts are not able to see the whole picture of their money flows, due to the fact that they have to pay many stakeholders, money are withdrawn automatically from their bank account, new payment obligations are made and so on. The first step to create clarity in the financial situation of the user is to make an overview of the current situation.

This however raises other difficulties such as: the data that is needed to work with is not located in a single source, it can be gathered from many governmental websites, banks etc. Our goal is to make a profile with structured financial data for each user. We should also make sure that an user is able to upload a file with his own bank transactions, so that bases on this document an overview is generated. User should have access to his personal file, can add / update / delete some of the personal data, and also the possibility to share this information in a comfortable format with an expert in the field. All the gathered information should be very well structured such that it is easy for the user, as well as for the financial advisor to read and to understand it.

2 Overview of the System

The Roger That project consists of three components.

- *Web Application.* Will be used by the user's, so it must have an easy to understand UI/UX[1], later it can be converted to an Android/iOS app. For the frontend technology we decided to use Angular Framework
- *BackEnd.*The core of the project. All complex data analysis, database modifications and API[2] will be done here. JAVA is the backend language, because it provides great tools for working with the Databases such as Quarkus Panache ORM [3].
- *Database.* The database will hold all the necessary information for the web application to work properly (e.g. user's login credentials, financial information provided, user's overview). MySQL will be used.

3 Target Users

The application of the RogerThat project is designed as an app that provides an overview of someone's current financial situation, therefore the RogerThat app is primarily targeting the adults that are currently in a problematic financial situation and are unable to manage their own funds. The financial overview might favour the people with debts in order to have a better understanding of how they manage their resources and request help from financial advisors.

However, the RogerThat application might be useful with the users that do not have any financial problems, but are interested to have an insight of their income and spendings.

Last but not least, staff members of the Roger That Company are willing to have a database with sorted financial information of the user's of their app, which might be helpful in the future when adding new functionality, or updating the existing one, so that the whole user information is not lost.

4 User Stories

Currently we are having two types of users which are using the application, however as the system grows in the future, new user types and stories might be added.

The format of a user story is as follows : First comes the user type, which is followed by a short text with what the user wants the system to be able to do, and finally is optionally followed by a short reason why the user wants this to be possible.

The user stories are categorized in two sections, each for one of the user type.

User

- As an User, I would like to add all my financial transactions as a CSV[4] file format, that will later be used to generate the overview of my financial situation.
- As an User, I want to be able to observe my financial status (on a scale from red to green) based on the data collected from the bank transactions.
- As an User, I would like to export all the data collected in one document, preferably in PDF format, in order to be able to send this data to professionals that can help me.
- As an User, I want to be able to login through the website securely so only I will be able to access my personal data.
- As an User, I want to be able to give someone else (e.g. Financial Advisor) the possibility to access my data, so that I can receive financial help.
- As an User, I would like to have all my input data structured and categorized such that it is easy to gain an insight of my expenses and incomes.

Staff member of RogerThat

- As a staff member, I would like to be able to view if this dataset likely includes a ZZP[5] account (personal business) (yes/no).

- As a staff member, I would like to be able to view if this dataset shows recurring payments for X,Y,Z (e.g. BTW(VAT))
- As a staff member, I would like to be able to view if in X months we don't see the expected recurring payments.
- As a staff member, I would like to be able to view if in X months we do see similar payments to a "deurwaarder" (yes/no).
- As a staff member, I would like to be able to view if in X months we see an increase of cash withdrawals and a decrease in grocery shopping payments (anomaly yes/no).
- As a staff member, I would like to be able to estimate that an amount X has not been paid and is indebted to parties X, Y, Z

5 Functional Requirements

From the above mentioned user stories we are able to extract the following functional requirements of RogerThat project :

5.1 Critical functional requirements

- The user's account should be encrypted, such that no can view the user's login credentials in the database
- To be able to login to the app securely so that only the user might access the information he provides.
- To be able to collect the information from the CSV file with bank transactions uploaded by the user, and store the information in a single database.
- To be able to categorize the financial data inside the database. The data should be structured in categories, in order to be easily accessible, maintainable and readable by the user.
- To be able to generate an overview from user's financial transactions and conclude on the current status of the user (e.g. worst - red, best - green).

5.2 Useful functional requirements

- To be able to retrieve the generated financial overview for a specific user from the database, in a comfortable format (ex. PDF), that will ease the possibility of sharing the financial information with an advisor.
- To be able to collect bank transactions from multiple banks such as: ING, ABN, RaboBank etc.
- To be able to download a PDF of the generated overview of the User's collected and structured information.

6 Non-functional Requirements

- We must ensure that all the collected personal data from the user will be properly secured and the login credentials encrypted inside the data base.

- Since the project is focused on a Web application, the website should be easy to use and available on most of the browsers.
- The interface of the application must be user friendly and intuitive, so that an user of any age is able to use the app.
- The code, and the database, has to be readable and clean such that the RogerThat team can use it later and implement future developments.

7 Customer Meetings

Date	Discussed topic
12/02/2021	Introduction Meeting and decision to come with a debriefing on the next meeting.
26/02/2021	Analysis of the debriefing and some functional requirements and general ideas discussed.
30/04/2021	Presenting the updated main idea on the workflow of the app. Discussed on the already available functionality. Questions on future implementations of the Data Analyzer (ML or Rule-based) Received some advices and some documentation that might help further.

8 Change Log

Date	Changes
26/02/2021	Denis created the template
26/02/2021	Denis wrote the introduction and added actors.
28/02/2021	Constantin and Gasan : worked on the Actors and User Stories.
01/03/2021	Valeria : added some user stories and changes in the introduction
01/03/2021	Wesley made the Might Be Useful section
01/03/2021	Constantin and Denis : wrote the Project Architecture section
01/03/2021	Denis : added "staff member" actor and the user stories for them.
01/03/2021	Denis and Constantin : worked on Functional Requirements section
02/03/2021	Valeria : completed the Might be useful section
02/03/2021	Valeria : added the Non-functional Requirements section
04/03/2021	Denis : has adapted the document following received tips and feedback.
29/03/2021	Constantin : added Design And Architecture Section
22/04/2021	Denis : has divided the document in two parts :Requirements and Architecture
24/04/2021	Denis : updated the whole Requirements document, based on received feedback.
25/04/2021	Denis : Added some user stories, functional and non-functional requirements.
03/05/2021	Denis : Added "Overview of the System" section
31/05/2021	Valeria : Fixed layout
01/06/2021	Valeria : Added the "Critical functional requirements" and the "Useful functional requirements" sections
01/06/2020	Valeria : Added the "Glossary" section

9 Glossary

Keywords and possibly ambiguous or unknown terms are defined below:

1. **UI/UX** - User interface/ User experience
2. **API** - Application programming interface that defines interactions between multiple software applications or mixed hardware-software intermediaries.
3. **Quarkus Panache ORM** - is the JPA implementation and offers you the full breadth of an Object Relational Mapper.
4. **CSV** - A comma-separated values file that contains delimited text file that uses a comma to separate values.
5. **ZZP** - An entrepreneur without any staff who works for a number of different customers.