Roger That Project

Software Engineering

Requirements Document

Denis Garabajiu *(S4142551)*
Constantin Cainarean *(S4142152)*
Valeria Mavceanscaia *(S3673952)*
Gasan Rzaev *(S3553213)*
Wesley Haverkort *(S3811808)*

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. As a solution to this question RogerThat team proposed the following : Organizing financial information of an individual.

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. This data is going to be gathered using the APIs of these resources, and our goal is to make a profile with structured financial data for each user. We should also make sure that a user has 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 Actors

- Adult : An adult with problematic financial situation. Is giving permission to the application to use his financial data.
- Financial Advisor : A financial advisor. Is offering professional help to the person that is in unfavorable financial situation.
- Governmental financial institutions : By accessing to statistical data, the Government can analyze on which categories do people with financial problems spend the most. Based on this information the Government can help them by passing different regulations, tax deductions and giving new allowances.
- Company's employee : A staff member of Roger That company. Is willing to have a database with sorted financial information of an user, in order to be able to detect anomalies.

3 User Stories

User

- As a user I would like to add more personal data to the platform. However, the data from the sources remain unchanged
- As a 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, tax office, municipality and others.
- As a user, I would like my spendings to be categorizes so I could see in real time on what I spend the most and optimize my budget.
- As a user I would like to export all the data collected in one document, in order to be able to send this data to professionals that can help me.
- As a user I want to be able to login through a website so only I will be able to access my data.
- As a user I want to be able to give someone else the possibility to access my data.
- As a user I would like to have all my input data structured and categorized such that it is easy to gain an insight in my expenditures.
- As a user I would like that only the relevant and appropriate data is collected from the given sources.
- As a staff member I would like to be able to view if this dataset likely includes a ZZP 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

Financial Advisor

- As a financial advisor I would like to be able to receive collected data from the sources in order to help my client to help them fix the financial issues.
- As a financial advisor I want to be able to set a budget for my client for different categories and get notified whenever my client does not respect it.
- As a financial advisor I want to have the data properly structured so that it is easy for me to gain an insight in my client's problem.
- As a financial advisor I want to have enough information about my client's status such that I can create the best tax scheme to manage his expenses.

Governmental financial institutions

- As a governmental financial institution I want to access statistical data(an overview of all users) so I can have a general idea on what do they spend most money on. Based on this information the government will take action.

4 Project Architecture

Our team decided to split this project into 3 components. First component is the web application, it will be used by regular people so it must have an easy to understand UI/UX, later it can be converted to an Android/iOS app. For the frontend technology we decided to use Angular. The second component will be the backend - the core of our project. All complex data analysis, database modifications and API will be done here. We chose JAVA as the backend language because it provides great tools for working with the db(Quarkus Panache ORM), all of the team members are familiar with it and there are a big variety of libraries that can be used in the development. The last but not least, as the database we will use MySQL.

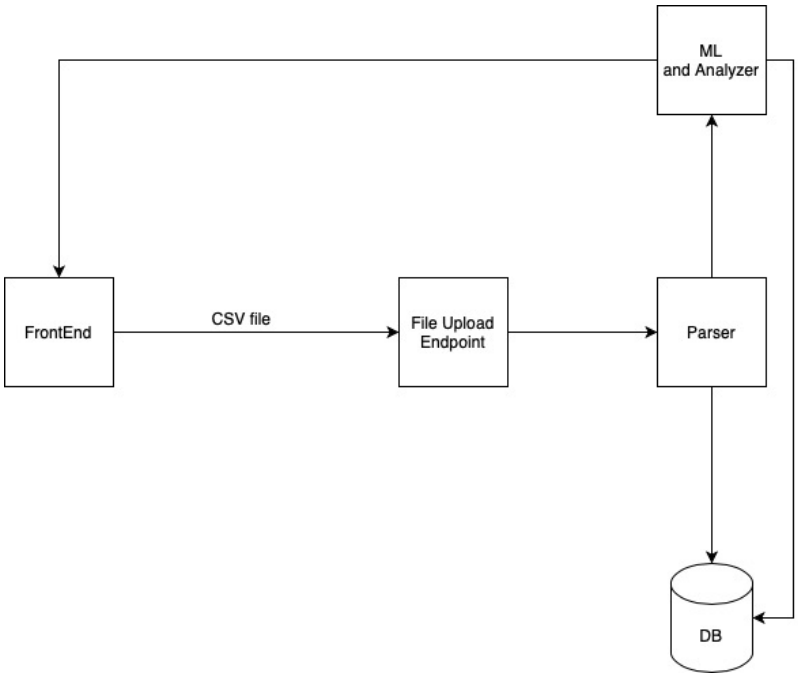
5 Functional Requirements

- To be able to collect the information from different sources and store them in a single database.
- To be able to "normalize" the data inside the database. The data should be structurized, in order to be easily accessible, maintanable and readable.
- To be able to retrieve the information 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 detect anomalies in user's financial transactions (possibly using SEMI-technologies ML model).

6 Might be Useful

- It would be easier to use if the interface was in dutch, because of the company being Dutch. (English wouldn't be a problem)
- The interface should be easy to understand, people will be more pleased and more exited to go aboard with the program.
- An English documentation of the program would be easy to maintain by all the developers and in future by staff members.
- To have the structured data categorized for the user, in our case the categories will be specified with colors: red,green,yellow.

7 Design and Architecture



- The User uploads the file and it ends up at File Upload Endpoint, where several checks will be performed before passing the data to the parser
- If the File has passed all checks, the parser will be called. Parser will parse the data and will add them to the DB related to the user X. Parser will call ML And Analyzer.
- ML and analyzer is the main part of the project. It will take all available transactions for the user X and will try to analyze them based on several categories. At the beginning our ML model will be quite dumb, but with time it will get better, the users will help us improve it by manually categorizing the transactions.
- The overview of user's financial status will be saved into the DB for futher use

8 Non-functional Requirements

- We must ensure that all the collected personal data from the user will be properly secured.
- The website should be easy to use and the structured data should be easy to understand by the user.
- The code has to be readable and clean such that the RogerThat team can use it and update it.
- The project has to have the needed documentation containing all the relevant information about the project.

9 Customer Meetings

Date	Discussed topic
12/02/2020	Introduction Meeting and decided to come with a debriefing next time
26/02/2020	Analysis of the debriefing and some functional requirements and general ideas discusse

10 Change Log

Date	Changes
26/02/2020	Denis created the template
26/02/2020	Denis wrote the introduction and added actors
28/02/2020	Constantin and Gasan worked on the Actors and User Stories
01/03/2020	Valeria added some more user stories, and made some changes in the introduction
01/03/2020	Wesley made the Might Be Useful section
01/03/2020	Constantin and Denis wrote the Project Architecture section
01/03/2020	Denis added "employee" actor and the user stories for them
01/03/2020	Denis and Constantin worked on Functional Requirements section
02/03/2020	Valeria completed the Might be useful section
02/03/2020	Valeria added the Non-functional Requirements section
04/03/2020	Denis has adapted the document following the tips from the TA