Department of Economics and Business
CENTRAL EUROPEAN UNIVERSITY

A DOCUMENT PROCESSING SOLUTION FOR DIGITAL FACTORING AUTOMATION|

PROJECT SUMMARY

CAPSTONE PROJECT

AUTHOR

Dominik Gulácsy – 2003374

MSc in Business Analytics

07/06/2021

Most of the services and processes that surround our lives got heavily digitalized and automated during the past two decades. Nowadays we park on the streets with our phones, shop online, take online courses etc. Even the banking industry is shifted towards a digital environment and user experience, offering smart solutions like mobile banking or online credit application. Despite all of these advancements the remains of a world more dependent on manual tasks carried out by humans resurface more often than we might think. Factoring is a typical example of that. As digitalization and automation are especially slow in the financial services industry it should not come as a surprise that digital factoring in Hungary is still a new thing. As a matter of fact, my project is related to the initial steps of creating a fully digital and semi-automated factoring solution at a Hungarian startup company. The project is concerned about the processing and validation of a certain document type called the trial balance which is a bookkeeping worksheet displaying the state and change of a firm's accounts. These papers are submitted as part of the application process for factoring. The problem is that these documents are all PDF files which was originally designed to present data and information for a human audience and not to be parsed by machine code.

The final goal of this capstone project was to bridge the gap between the widespread use of PDF documents and their poor processability concerning this particular document type by implementing a solution that can be added to the firm's SaaS service offering. After learning about the company's strategy, business needs and technical requirements, fundamentally, I had to do three main things in terms of the project. Firstly, I designed a solution architecture by breaking up the problem into two modules, determining the responsibilities of the modules, and figuring out how the operation of these modules should be coordinated. Secondly, I built the first module, the Trial Balance Reader, which reads in documents and turns them into standardized and validated dataframes. Thirdly, I

built the Trial Balance Transformer that takes the clean dataframe and then extracts and calculates useful features from the data.

During the design of this solution many software engineering perspectives needed to be considered. For instance, the coordination and communication between the modules had to be dealt with in a way that it could be easily integrated into the production system. Due to this, I implemented the modules as webservices having a single functionality. These webservices communicated via web requests and responses with a JSON in their body containing all necessary input and metadata that is needed to do the job. Additionally, I also needed to design the delivered solution to be run on GCP (Google Cloud Platform) with the help of Google Cloud Functions while the relevant PDF documents were accessed from a Google storage bucket. This setup made sure that the solution was scalable, resilient and its performance could be continuously monitored with Google's Cloud Monitoring service.

Looking at the Trial Balance Reader module, it consists of many key steps that I had to figure out to standardize and validate the data in the PDF documents. First of all, I had to find a package that could map the objects in the PDF to a dataframe structure. After some testing, I decided to use the Camelot package for this. Then I developed a header extraction method that could handle the changing column names across documents and map them to standard categories. On top of this, I also had to deal with the conversion of numeric-looking cells to actual numeric cells. This was hard since formatting was non-uniform, but eventually, I managed to come up with a solution. Next, I worked on filtering the columns and rows that were relevant and checking whether cells were mapped correctly to the corresponding columns by looking at their data type.

In contrast to the Trial Balance Reader, the Trial Balance Transformer happened to be a much lighter module regarding complexity. It was because the input

coming from the Trial Balance Reader was pretty much clean. The main thing that I had to do here was to turn the input table into a series of relevant feature variables. I did it by looking at the account id variable of rows identifying their content with the trial balance numbering guide presented in Hungarian accounting and bookkeeping laws. After this, I wrote the part that calculates financial ratios from the identified variables and returns all these features in a single-row format to an orchestrator script that later uploads it into Google's Metabase data warehousing service or adds it to a local CSV file.

All in all, the whole trial balance document processing solution helped the factoring company to make a very important first step toward converting this subtask into a fully digitalized and automated process. The cleaned, standardized, validated feature data can be used for many types of further projects. For example, better understanding the profile of applicants by developing a dashboard application or trying to train a model which can learn the ruleset applied by financial experts to decide whether an application should be accepted or rejected. This project and subsequent ones mentioned are all aligned with the long-term strategy of the company which is to improve such convenience features of the company's SaaS product as speed, integrability and minimal required user interaction. Consequently, they contribute to the company's growth.

To conclude, I learned a lot during this project. First and foremost, I learned how important it is to fully understand the business context of the analytics projects and how to ask questions that shed light on the most notable pain points that need to be addressed by the end result. I also realized the significance of being able to communicate arising issues and difficulties to internal stakeholders to adjust their expectations accordingly. From a technical point of view, I learned a lot about document processing and working with unstructured data in general. Finally, I also gained some experience on how to pack a data engineering solution into a cloud-based service.