Realization

Report

Internship JUVO | IT Factory

Van Eynde Michiel 3-APP

Academic Year 2021-2022

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

LID VAN ASSOCIATIE KU LEUVEN

THOMAS MORE

TABLE OF CONTENTS

1 INTRODUCTION

The main goal of the assignment we received was to make the recruitment task easier and more maintainable. Before every resume was in a separate document and there was no central storage. Now with the newly created CV-application a user or an applicant can upload their CV. After the upload a text recognizer reads out the text and forms it into a nice overview, in which the user can correct the data if needed.

You can find a nice video on my portfolio, in which I will go over to complete application to show what has been created in a visual manner.

2 DESIGN CV-APPLICATION

2.1 Layout

Because the project was already been created, we hadn't to design the global layout. If we had created a new page, we just had to make sure it was in the same style, but that came down to just using the right classes.

Programming logic was something different, here we had to refactor a few times, in both back-end as in front-end. An example here: at first an integer was used as the id-key, but for security and making the app less predicable we chose to change it to UUID's to be more secure.

2.2 Mobile

The application will be mostly used on a computer but being a consultant company, it has to be possible to use the application on a mobile device. It will work but the downside is we use heavy equipment to extract all data, so on pc it will take around 10-30 seconds. If it is used on mobile, timings will be much longer, so there is still room for improvement.

2.3 Text recognition

During the internship we optimized how the extractors and convertors worked. We have expanded out text-helpers so more certifications, educations and work experiences. After we could get more data out of the resumes, it was time to make the extractors better. During this process we also implemented Google Vison API. With this API it is possible to extract text from an image to use later in the conversion to our model. After all data is extracted in both ways (API and our normal NLP[1]) we combine it to make one model. For now we only check how long every section is. So if the Vision API finds 2 educations and our the NLP only 1 we pick out the one with the most.

2.4 Other cases

Our main focus was to improve the data recognition, but this was not the only thing we did. My goal was to update in front-end to show errors and how everything was ordered/placed. Even the generated resume was tweaked a little bit so everything was nice to look at. In back-end we added a role management system and how an administrator could change them. Also we made sure an admin could add new skills and made sure they had a category.

2.5 Link to demo

Here you can find the link to a small demo. During this demonstration you can find a small introduction on how the application works and what we have achieved during our internship

Link: https://youtu.be/4rrv1Bb2p3k

1: Natural language processing

3 PERSONAL FAVORITE

The way how it shows when a field is not valid after extraction is one of the things I am the most proud of. During the scope of this ticket I had to use a lot of new techniques. For example I had to use a service to show the form, so I could validate it before it was ever rendered to the screen. Using this method it was possible to use the output (valid or not) three files higher to created a border around the component that had a false input.

During this ticket I had to think a lot on what would be the best way to get this done. I've had a few set backs but I used it as a way to clear my mind and a path that wouldn't work, so that in my next attempt I could know what not to do.