Contents

[Plan of Approach 1](#_Toc156837311)

[Communication with the Client 2](#_Toc156837312)

[Reflection 3](#_Toc156837313)

[Ben Cattoor 3](#_Toc156837314)

[Wout Depreitere 4](#_Toc156837315)

[Yentl De Brabander 4](#_Toc156837316)

[Analyse 5](https://studenthowest-my.sharepoint.com/personal/yentl_de_brabander_student_howest_be/Documents/Bijlagen/Academic%20year%202023-2024/SE5/Software_Engineering_Project/DPTechnics/FinalDocumentation/Research/FinalDoc.docx#_Toc156837317)

[Finale deliverable 6](#_Toc156837318)

[Summary 6](#_Toc156837319)

[Flow 6](#_Toc156837320)

[Research 6](#_Toc156837321)

## Plan of Approach

**Week 1-2: Project Setup and Research**

* Define detailed project requirements based on client specifications.
* Research and choose appropriate web scraping libraries and tools.
* Set up the development environment and version control system.

**Week 3-4: The basics**

* Develop the web scraper for one target platform as a prototype.
* Implement JSON schema validation for the scraped data.

**Week 5-6: Data management**

* Create JSON schemas for each target platform (bakeries, butchers, chip shops).
* Develop the translator module to map scraped data into Croissant system's schema.
* Integrate the scraper and translator modules into the Croissant platform.

**Week 7-8: AI integration**

* Integrate AI to make the web scrapers robust to change.
* Integrate AI to write translators, so different forms of data have the same final JSON schema.

**Week 9+: Testing and Refinement**

* Conduct extensive testing to identify and fix bugs and issues.
* Perform user acceptance testing (UAT) with the client to ensure the system meets their requirements.
* Refine the system based on client feedback and conduct final testing.

## Communication with the Client

During out time with this project we were in constant contact with DPTechnics. The first half of the project we communicated by mail, in the last half of the project DPTechnics made a discord server so we could communicate easier. We often updated them on our progress and received feedback on what we could do more and what our priorities should be.

**Week 1 (18 - 24/09):**

* Presentation about all the projects
* Get assigned to a client

**Week 2 (25/09 - 01/10):**

* Meet the client on premise
* Get an explanation of the assignment

**Week 3 (02 - 08/10):**

* We tested out several packages for making web scrapers
* We made a basic web scraper with selenium in python for Bakkersonline

**Week 4 (09 – 15/10):**

* We added more basic web scrapers for all platforms
* Tested out whether these scrapers are consistent with all websites for the platforms

**Week 5 (16 – 22/10):**

* Sit together with DPTechnics to explore possibilities and new research directions
* Make the scripts headless
* Refactor the scripts so it gets all product data, not just superficial data

**Week 6 (23 – 29/10):**

* Compare script performances between the “Online Scraper” and the “Unipage Scraper”
* Try different web drivers and caching to try and improve performance

**Herfstvakantie (30/10 - 5/11)**

**Week 7 (06 – 12/11) (Projectweek):**

* Debugging

**Week 8 (13 - 19/11):**

* Sit together to discuss possible improvements with DPTechnics in a feedback session

**Week 9 (20 – 26/11):**

* Further debug and complete scripts

**Week 10 (27/11 - 03/12):**

* Debugging

**Week 11 (04 - 10/12):**

* Make a translator

**Vakantie (11 – 17/12)**

**Week 12 (18 - 24) (Projectweek):**

* Finish all scripts
* Clean up the scripts
* Make a bash load balancer system that takes a csv file as input that they can run periodically
* Make useful documentation and write down our notes and research

## Reflection

### Ben Cattoor

This was a really interesting project to work on. This felt like a tiny internship, which is really nice considering that we have a real internship in the next semester. This was a good experience and warmup to work in a team setting, while actually delivering results for something meaningful in the real world, rather than just another school project. This made me really motivated to deliver a good project, and made me learn a whole lot about new subjects I would otherwise never have spent my time on.

Embarking on this project proved exceptionally intriguing. Punctuality was a constant, with clear and effective communication regarding deadlines. Our collaborative sessions unfolded every Wednesday, predominantly at Vives due to space constraints at Howest. Physical meetings with Jonas from DPTechnics occurred at Howest for logistical simplicity. During dedicated project weeks, our commitment spanned the entire week, striking a balance between remote work and on-campus presence. The cohesive synergy within our team was remarkable, contributing to a highly effective and harmonious work environment. These collective efforts culminated in the successful execution of the project's goals and objectives, underscoring our ability to navigate challenges collaboratively and deliver results with precision.

We worked closely together and had no real roles, except for Wout, who was most often in contact with DPTechnics since he was the communication manager. We did split ourselves up to have a pretty balanced workload each and so we could progress steadily on most issues. I feel like everyone was capable of completing their own tasks easily, so no comment there. The only little problem was that timing could be a little awkward from time to time due to having to work on multiple school projects at the same time, and not everyone having the same projects. This was not a big problem since every member of the group just completed what they had to do if they were not done with it at the end of the day.

The communication with DPTechnics was really smooth. We started off with communicating with emails, but switched to a discord server somewhere in the middle of the semester. We had contact with them basically every week, whether it was feedback from them or a status update from us, there were no real problems with communication. One thing i would have liked to happen is that the discord server was made right from the start. While emailing goes a long way it can be a little awkward to send a mail for a little thing, while the discord server just made it feel a lot easier to send updates and progression.

If I had to do this project again I would have liked the first feedback session with DPTechnics to have been a little earlier. Although we were making good progress during the early stages of development, this feedback really opened up our minds to possibilities and possible solutions. Again I would not say we had wasted the first few weeks of our time, but from that point on our time was spent in a much more meaningful way than before.

### Wout Depreitere

### Yentl De Brabander

A diagram of a computer

Description automatically generated

## Analyse

<https://lucid.app/lucidchart/f0cd16d9-814c-45a5-9689-f1752fac8542/edit?invitationId=inv_da0de440-3cc9-4418-ae2f-2639e49129d3>

## Finale deliverable

### Summary

Our assignment was to research and test out different solutions on how bakeries and butcher shops could easily change platforms to DPTechnics. We worked with a practical approach, making the system, trying out different ways to make it, and documenting our findings in the process.

We came up with a good solution that works. There is a bash loadbalancer that DPTechnics can run on their linux server. This bash script can read a csv in a format they specified and gets the data accordingly by running the appropriate scraper.

### Flow

1. **ScrapeRunner.sh → ScrapeController.sh:**
   * Initiates the process and passes each entry to the ScrapeController.
2. **ScrapeController.sh → PythonScrapeController.py / JavascriptScrapeController.js:**
   * Retrieves data using either Python or JavaScript-based controllers.
3. **PythonScrapeController.py / JavascriptScrapeController.js → OnlineScraper.py / UnipageScraper.py / OnlineScraper.js / UnipageScraper.js:**
   * Executes the appropriate scraper based on the engine specified.
4. **OnlineScraper.py / UnipageScraper.py / OnlineScraper.js / UnipageScraper.js → raw/temp.json:**
   * Outputs the scraped data to the raw/temp.json file.
5. **raw/temp.json → JsonTranslator.py:**
   * Reads the data from the raw/temp.json file.
6. **JsonTranslator.py → json\_out/yyyy\_mm\_dd/:**
   * Translates the data and outputs it to a directory within json\_out, with the directory name as the current date.

### Research

As mentioned before, this was an open ended research based assignment. These are some of the things we found out during the development of the system.

* **AI Integration:**

Although it would be possible to integrate the use of openAI, it would require a homemade service to be built. Another important factor is that AI is still not completely consistent and that this brings a certain risk.

* **Possible Expansions:**

We built our system to be easily expanded upon. Our time was limited for this project and there are certain to be more packages out there that would fit or even improve our current system. It is possible to just add them to the current framework.

* **Recommendations:**

There are certain packages we have tested and recommended to stay away from. While they seem to have considerable benefits, these are just bait for what would be required to actually make them useful.