Collection platform

Client: Mnr Clauwaert. Students: Stan Vandamme, Ryan Quivy

Plan of action

For the type of collection, we were free to choose. Stan and I chose "Games" as this was the example set from the client.

Requirements:

- Simple UI
- Account based
- Use of keyboard shortcuts
- Reverse eBay (2deHands)
- Having a collection and wishlist separate
- Usable on phone (PWA)
- Offline functionality
    - For when somewhere without internet or for less technical people
- Sharing your wishlist for other people to see in case of gifts
- One entry for the same game in multiple zones released (EU, USA, JP, ...)
    - Alternate titles
- Easy setup for client
    - Docker file
- Games have a state

Planning:

- Create the database setup
- Start creating the backend
- Start creating the frontend with dummy data
- Integrate them
- Make changes as requirements change after meetings
    - A semantic search feature
    - Toggle to switch between regular and semantic search
    - Toggle to show similarity score of semantic search
    - In use case of a less technical person, no account required to view others' wishlist/collection
- Create appropriate error messages
- Create Docker file
- Create seeders for Docker file for demo purposes
- Create needed readme and other technical explanation files
- Create video of how the application works

Communication

Weeks before 1st project week:

- Figuring out technical aspects
- Checking if they are okay with the client
- Change where needed
- Start working on the database setup
- Basic setup of all projects

> Every time we had a meeting with the client we would show our diagrams and written down aspects of the project.
> The client would give feedback on it and say what he wanted differently.
> We made sure to change where needed.

1st project week:

- Start on backend and frontend
- During meeting explaining what we have done, what we are going to do, and make changes where needed based on client's feedback
    - Semantic search
    - Removing of viewing everyone's collection/wishlist
        - Only via follow
    - Asking about UI and applying feedback

> During this meeting we showed the progress.
> Client was overall happy, but wanted something extra like the semantic search.
> We wrote it down and made sure to implement when possible.
> All other requests the client previously made were applied.

Weeks between 1st and 2nd project week:

- We talked with the client weekly about progress and possible additions
- Applied feedback, added additions

> When we talked we told about our progress every week.
> Client would give feedback (positive or negative).
> We took that into account and tried to apply it before the next meeting.
> Generally, all went very smoothly.

2nd project week:

- Applying final touches
- Getting more feedback
- Asked about final deliverables

> Client was happy with the final (99% done) result of the application.
> Wanted 2 small extra features: a toggle to see the score of the semantic search per entry and paging.
> Applying them to the project.
> Very happy with the Docker file for easy setup.
> For deliverables, the final presentation and a zip in email with source code is enough (due to client being a lecturer).

Reflection

Ryan Quivy

For me, this project was more of a recap of what we have seen the past years. Not that this is a bad thing, I see it as a refreshing of the already known skills. Some of the things we did were new and quite interesting like the semantic search. This was the first time hearing of it, after some research and implementation it was nice to see how it worked.

> From this, I learned that the size of the input query weighs very heavily on how the result is. A 1-word query like "ultimate" will be less accurate than something like "ultimate world" (as an example for the games Final Fantasy).
> Also, numbers in titles are sorted descending unless a number is given in the input.

The fun thing about this project was the ability to choose what technologies we could use. Since Stan wanted to do frontend and I backend, we both chose for ourselves and discussed it.
I chose Deno for backend since it is relatively new and we had just seen a bit about it in class. Stan chose Vue.js since he liked working in it from previous classes.

The planning and deadlines were always reached on time, we did not have to rush anything about this project which was nice because other projects were taking up a lot of time. We tried to work on school most days (at least half a day) when possible so we could easily give and receive feedback from each other. As said before, Stan worked on the frontend and I on the backend so we could easily keep working even though the other could not. Working with Stan was a joy, he works well and fast and has great technical knowledge and a good work drive. We never really got into heavy discussions about the project, mainly because I think we are more or less on the same technical level and if needed helped each other.

Communication with everyone was really simple and never a problem. Stan and I saw each other almost every day, if not there is Discord. Talking with the client was at least weekly because we saw him every Wednesday in the course "Service Architecture & Deployment". We never really talked to any of the lecturers for help, because we both like (I think) to solve things on our own.

If I were to do this project again I would maybe choose different technologies where I am not so familiar with, this would have given me an opportunity to learn something new. For instance, a different programming language, or more advanced techniques in already known languages.

Stan Vandamme

This project went very smoothly overall. The team size was appropriate, and both Ryan and I had the expertise needed to complete our respective parts. When dividing the work, simply asked each other, “Can I do this part?” This ensured that we both worked on areas we were interested in. Ryan took on the backend and semantic search, while I focused on the frontend development and Docker setup.

To maintain progress, we met every Thursday on campus and worked together for at least half a day. At the end of each session, we decided whether the remaining tasks for the day could be completed remotely. I found I was slightly more productive on campus compared to working from home, so I chose to do most of my work there during project week. Ryan, on the other hand, preferred to work remotely when possible, as avoiding the commute saved him a lot of time. Since we worked on separate parts of the project the small amount of communication that was needed could easily be done over Discord.

Communication within the team and with the client went smoothly. Ryan and I could easily discuss our progress and challenges in person or via Discord. We also maintained regular communication with the client, meeting at least once a week during our Wednesday class. For any additional or urgent questions, we used email and Teams.

We required help from the teachers only once, when I encountered an issue with the Docker file for the backend. I reached out to Mr. De Smet during one of the SAD lab sessions, and he was able to resolve the problem. It definitely would have taken us significantly longer to solve this problem on our own, so I thank Mr. De Smet for his help.

Throughout the project, my skills in Vue.js improved significantly. I also gained valuable experience working with Docker Compose and Docker files. Using Docker in a real project environment helped me appreciate how much time and effort it can save. Once the Docker setup was complete, I could easily redeploy the entire stack or wipe and reseed the database with a single command. This streamlined both development and testing.

Looking back, one improvement I would make is to prioritize creating the Docker and database seeding scripts earlier. Having these tools available from the start would have simplified testing during development.

Analysis

First, we created multiple text files where we wrote down our understanding of the project and requirements. Then we created a database schema to have something to show the client and get feedback. After some questions about UI and certain features, we created a clickable wireframe. Other than these two, we did not make any other schemas.

Final Deliverable

- A git (lab or hub) repository that the client can access
    - https://gitlab.ti.howest.be/ti/2024-2025/s5/project-iv/projecten/project-01
    - https://github.com/BreadstickoSaurus/CollectionPlatform
        - When using the nested projects please check if you are on the main branch, you could be on a certain commit branch (dont know how to fix)
- A zip file with the source code, emailed to the client
- The final presentation
- A video showing the application (https://youtu.be/3aohpzpuWOc)
- A guide on how to deploy (docker repo)