Clément Caron – Etienne Goury – Rapahel Veron – Capucine Boudin
CDOF2

Projet Software Engineering

Table des matières

Preface

In this report, we will examine an application that differs from the one previously presented in our class. Initially, our intention was to create a website named VintageVault, dedicated to the sale of vintage items. We have actually developed this website, and a demo video showing its features accompanies this report.

However, the feedback we received following the presentation was unexpected. While we were aware that our backend was relatively limited, consisting of only a few tables on SQL Developer, we also faced criticism regarding our frontend. Specifically, concerns were raised about our choice of visualization using Wix, with suggestions that it was not the optimal option. These reactions left us feeling disappointed and unsure about how to enhance our project further.

Fortunately, employing Scrum methods provided us with an approach to address these challenges. We were able to retrace our steps and reevaluate our application effectively. It became evident that in order to strengthen our backend, we needed to simplify our frontend. The VintageVault website had an extensive array of functions that demanded implementation, which led us to the realization that prioritizing quality over quantity was crucial.

Consequently, this report will focus on the second application we developed. Through following the Scrum methodologies, we were able to adapt and refine our project successfully. By simplifying our frontend and refining our backend, we aimed to create a more cohesive and improved application.

Introduction

This project serves as a valuable learning experience in the development of an API using the Scrum methodology. Throughout this report, we will provide a comprehensive overview of each step involved in the development process and demonstrate how the implementation of Scrum facilitated the accomplishment of our tasks. The primary objective is to gain professionalism in using essential tools that are widely employed in companies. By doing this project, we had the opportunity to work with Jira, a popular project management tool. Its utilization allowed us to effectively manage our tasks, track progress, and collaborate within our development team. Moreover, the adoption of Scrum methodology enabled us to understand its significance and application in professional settings, where it is widely used to deliver high-quality results. Through this project, we aimed to acquire practical skills and insights into software development. The combination of utilizing Jira and implementing Scrum methodology has proven invaluable in achieving our project goals while simulating a professional work environment.

In the following sections, we will delve into the details of the development process, highlighting the specific steps undertaken, challenges encountered, and the lessons learned throughout this API development journey.

Scrum Methodology

Definition of the role

The first step of every project including Scrum methodology is the definition of each other's roles. Therefore, we attributed the role of Scrum Master to Raphael, Product Owner to Clément and development team to Etienne and Capucine.

Definition of the project

With the team now assembled, we collectively brainstormed ideas for the project we wanted to work on. As mentioned in the preface, our initial project did not meet our expectations. Consequently, we wanted to develop something that would be truly beneficial and meaningful. As students, we understand the importance of effective learning techniques. Thus, the idea arose to create an API that could assist us in the revision process. This led to the idea of our flashcard creation application. A flashcard is a card typically used in learning and studying. It consists of two sides: one side contains a question while the other side provides the corresponding answer.

Product Backlog

The primary objective of this app is to facilitate the creation and management of flashcards. To ensure a simple user experience, our focus is on developing a user-friendly interface that eases the organization and manipulation of flashcards. One essential feature we aim to implement is the ability for users to categorize flashcards based on subjects. This categorization allows for efficient organization and easy retrieval of flashcards based on specific topics. Additionally, users should have the option to delete flashcards when they are no longer needed or relevant to their learning goals. In line with effective learning principles, we want to respect the concept of active recall. To achieve this, our app will present flashcards in a manner that encourages users to engage with their memory. Specifically, users will first encounter the question of the flashcard, and only upon their approval or request, will the corresponding answer be revealed. This approach promotes active engagement and reinforces the learning process. An important aspect of the app is providing users with insights into their progress. Users will be able to track their learning journey, including details such as how many times they have reviewed a particular flashcard and which cards they have already mastered. This progress tracking feature allows users to monitor their learning achievements, identify areas that require further attention, and track their overall progress over time.

Sprint Planning

Once the product backlog was established, we proceeded to assign tasks to each of the desired features. To accomplish this, we used Jira because it is very valuable in organizing and distributing tasks efficiently among team members. Jira enabled us to allocate tasks based on individual roles and responsibilities within the team. In addition to task assignment, Jira allowed us to prioritize tasks by assigning points that reflected their relative importance. This prioritization helped us effectively manage our time and resources by focusing on high-priority tasks first. Furthermore, Jira enabled us to define the goals and timelines for each task. This feature allowed us to set realistic deadlines and track the progress of individual tasks throughout the project's lifecycle. By establishing clear goals and timeframes, we maintained focus and ensured that the project stayed on track.

Sprint

For the sprint, we set a deadline of one week to complete the project. To ensure effective progress and coordination, we conducted daily Scrum meetings throughout this two-week period. The daily Scrum meetings served as opportunities for team members to share updates on their assigned tasks and discuss their progress. This allowed us to identify any potential issues early on and make necessary adjustments to respect the two-week timeframe. By regularly sharing updates, we formed a collaborative environment that promoted transparency and accountability.

API Development

Creation of the environment

To develop our application, we made the decision to utilize the Webstorm IDE due to its ability to easily split the project into distinct frontend and backend components. This division allowed us to efficiently manage and coordinate the development process. For the frontend, we used Angular, Typescript, and HTML. Angular provided a robust framework for building dynamic and responsive user interfaces, while Typescript enhanced the development experience with its strong typing and additional features. HTML was used for structuring the web pages and presenting the content to users. On the backend side, we used Node.js, Typescript, and SQL for developing the server-side logic and managing data. Node.js provided a scalable and efficient runtime environment, while Typescript facilitated the development process by bringing static typing and modern language features to the server-side code. SQL was used for database management, and we integrated it with our Webstorm project. To accomplish this, we used ElephantSQL as our database service, allowing us to store and retrieve data easily.

Creation of the Product Backlog

Let's review the previously described product backlog and the corresponding components we applied:

Product Backlog:

1. User-friendly interface.
2. Categorize flashcards based on subjects and topics.
3. Delete flashcards when they are no longer needed or relevant to learning goals.
4. Respect the concept of active recall.
5. Provide users with insights into their progress.

Implemented Components:

1. We designed a simple interface with a pleasing color scheme, predominantly using purple tones. The interface consists of minimal buttons, each clearly labeled with their respective roles. To enhance usability, we incorporated a single table that collects and displays all the flashcards.
2. Users can easily enter the name of the flashcard and specify multiple questions and corresponding answers within the same category.
3. We included a user-friendly icon, such as a trashcan, to indicate the option to delete flashcards. This intuitive visual cue simplifies the process of removing flashcards that are no longer needed.
4. To respect the concept of active recall, we implemented an icon that enables users to study the flashcards. Upon clicking the icon, the questions are displayed first. Then, we

provide a "show answer" button, allowing users to check their response and verify if they were correct.

5. To provide users with insights into their progress, we incorporated two values. The first value increments by one each time the user studies a flashcard, indicating the number of times it has been reviewed. The second value informs the user whether the flashcard is new or has already been encountered, enabling them to track their learning journey effectively.

Sprint Planning and Task Estimation

In Jira, we used the concept of epics to organize our product backlogs. Each product backlog was divided into specific tickets that outlined the concrete steps required to achieve our goals. For each ticket, we assigned responsible team members and assigned point values. We used a scale where the easiest tasks were worth 1 point, while the most challenging ones were assigned 4 points.

In terms of scheduling, we prioritized the first two product backlogs (referred to as product backlog 1 and 2) to be completed initially. The remaining backlogs (3, 4, and 5) were planned to be executed simultaneously at a later stage.

Sprint Execution

With everyone now assigned to their respective tasks, we were ready to dive into the work at hand. However, one significant challenge that emerged was establishing a connection between the backend and the frontend components. It's worth noting that the individuals assigned to backend development might not necessarily possess a deep understanding of frontend technologies and vice versa. This disparity between backend and frontend developers created a potential bottleneck in achieving a cohesive system.

Testing and Quality Assurance

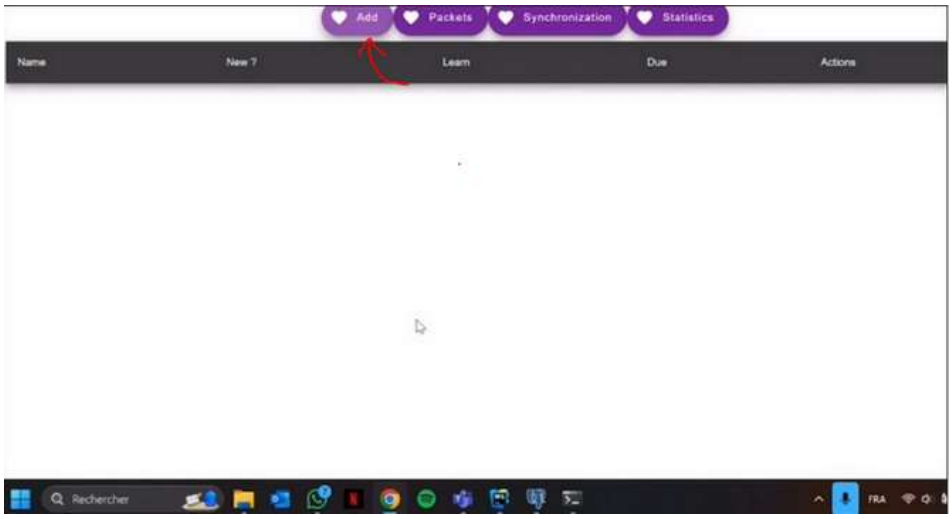
Ensuring smooth collaboration between the backend and frontend teams was very important each time we shared our work. The seamless integration of both domains and the stability of our API were our priority. To accomplish this, we executed a series of steps, including program execution and comprehensive testing, to verify the functionality of the website and the accurate incorporation of updates into our database. Thus, we tested each and every one of our product backlogs as well as diving into PostgreSQL to see if the new manipulations were well implemented. By simulating user interactions and meticulously examining user flows, we gained valuable insights into the website's behavior under different scenarios. This comprehensive approach allowed us to identify any potential issues, bugs, or performance bottlenecks that required immediate attention.

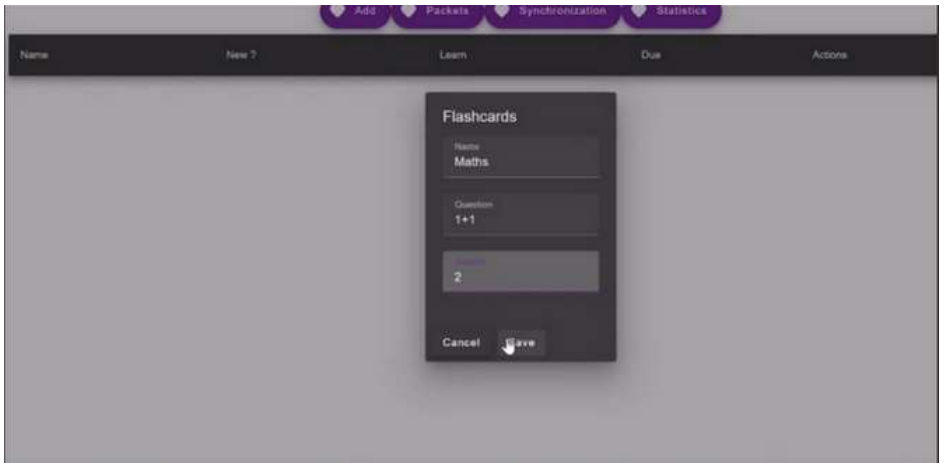
Sprint Review and Feedback Incorporation

During our sprint review, we identified certain areas of the code that were not up to par. This issue arose because we primarily worked individually on the code, and only later conducted joint testing. Consequently, we didn't give sufficient attention to the quality of the code itself. Upon further examination, we discovered that certain sections of the code were not fully implemented in TypeScript. Specifically, some of the requests and responses were not properly typed. To rectify this, we made sure to type all variables in accordance with TypeScript standards, ensuring consistent language usage throughout our entire project.

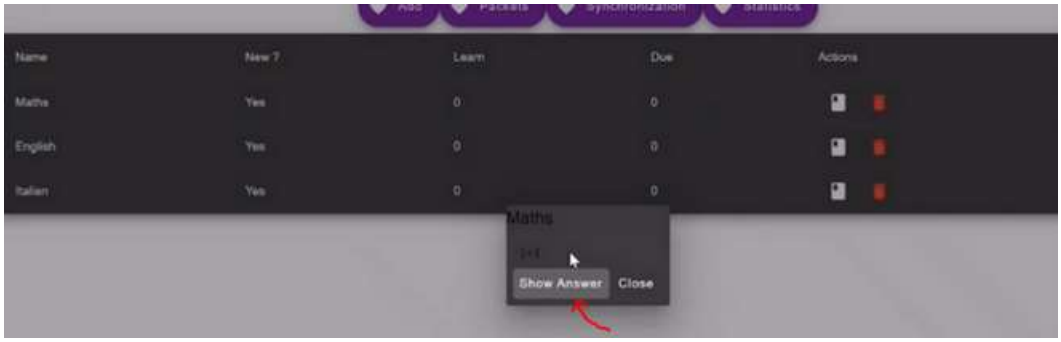
API functionalities

Create a flashcard

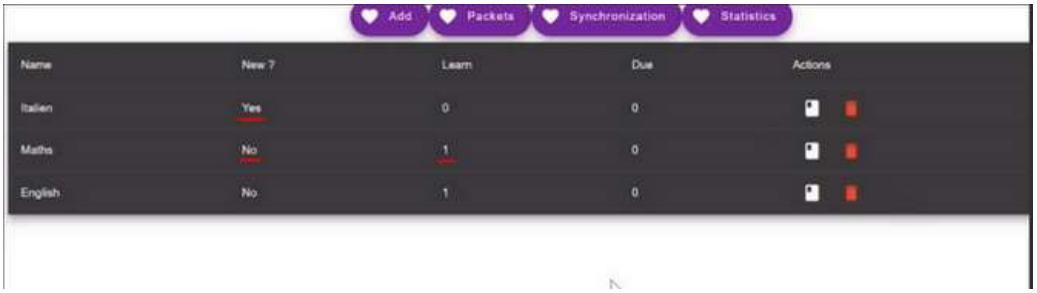




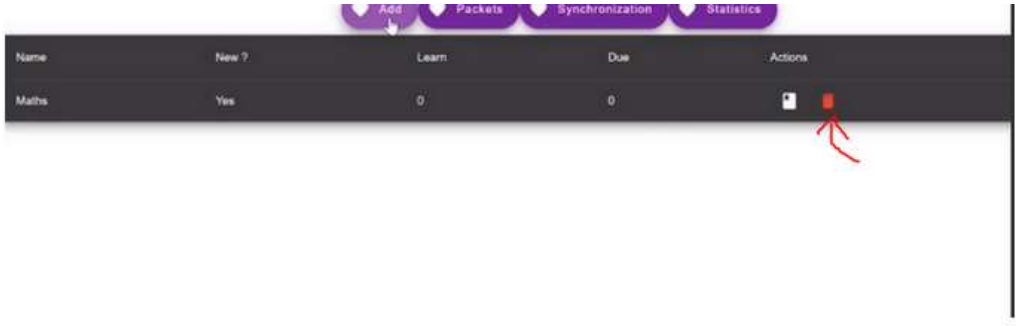
Revise a flashcard



See the progress



Delete a flashcard



Conclusion

In conclusion, our application successfully addresses the key elements outlined in the product backlog. We have prioritized user-friendliness by providing a simple and intuitive interface with a cohesive color scheme and minimal buttons, ensuring a smooth and enjoyable user experience. The organization and categorization of flashcards based on subjects and topics have been implemented effectively, allowing users to easily manage and locate specific flashcards as per their learning needs. Furthermore, the ability to delete flashcards when they are no longer relevant ensures that users can maintain a streamlined collection tailored to their learning goals. By respecting the concept of active recall, we have designed the flashcard study process to encourage active engagement and reinforce learning. Providing users with insights into their progress is a key feature of our application. The tracking of the number of times a flashcard has been studied allows users to gauge their review frequency, while the indication of whether a flashcard is new or familiar helps them focus on areas that require more attention. The successful implementation of the Scrum methodology, along with the completion of the product backlog, ensured a well-structured and efficient development process. The integration of Scrum's steps, such as Sprint Planning, Daily Scrums, Sprint Reviews, and Sprint Retrospectives, enabled us to consistently deliver value and enhance our teamwork throughout the project.