Project definition: Deployment and design of a question and answers platform

J. Alarcón. Sánchez, *Student Member*, & J.P. Mosquera Marín, *Student Member, Universidad Distrital Francisco José de Caldas*

Abstract -- **Since the beginning of the internet, forums have played a significant role in connecting users and facilitating discussions. Today, forums remain a vital part of the web ecosystem, providing a unique space for individuals to connect and exchange ideas on a wide range of topics. In this paper, we strive to show the process implied in modeling and deploying one of these apps to the internet covering its highlights such as its use and importance in the current time, its main key features and functionality, planned web deployment, and its development model and decisions made in its base version.**

In summary, this paper details the development of a backend for a Q&A forum, focusing on the use of contemporary tools and best practices in software engineering. We aim to create a backend that is not only robust and scalable but also maintainable and efficient. Through this project, we demonstrate how these tools and practices can be integrated to address common challenges in backend development, ultimately contributing to the creation of high-quality web applications.

I. INTRODUCTION

In this digital age, question-and-answer platforms have become crucial. They offer fast and precise responses to various inquiries and embrace individuals to share their expertise with a larger community. These platforms promote learning, troubleshooting, and knowledge dissemination, rendering them an invaluable asset for those in search of information.

This paper focuses on presenting a project that addresses these needs by developing an application that serves as a central point for discovering, acquiring, and managing digital applications and products. This application offers a comprehensive experience for users, from exploring and discovering new content to managing and updating their digital library.

With the increasing demand for reliable, scalable, and maintainable web applications, the development of a robust backend infrastructure becomes critical. This paper presents the design and implementation of the backend for a Q&A forum, leveraging modern software development practices and tools to achieve these objectives.

The objective of this paper is to showcase a project that fulfills the requirement of creating a forum application, with a key focus on building an application that acts as a hub for members to connect and exchange information. Such an application provides a set of tools like posting questions for the whole platform to answer, being able to answer and vote to questions, managing and controlling the users' accounts, moderate the content in the platform.

Our project is centered on utilizing Python, a versatile and powerful programming language, to build the backend, its extensive libraries and frameworks that streamline development and ensure high performance. Visual Studio, a comprehensive Integrated Development Environment (IDE), is employed to facilitate coding, debugging, and project management, enhancing the efficiency of our development process.

II. METHODS AND MATERIALS

During the development process of this application, we conducted research and proposed methods and fundamental points to ensure that the project-maintained software modeling standards. These standards are commonly followed in this type of project.
These methods and ideas are:

Model Design:
- Utilizing UML and design patterns can greatly benefit the development of well-structured and efficient app models. Making use of a previously made design and illustrating in UML will speed up our first steps in the app design in terms like class definition and the whole code connections. By taking advantage of this we can define and create a better and more optimal code from the start allowing faster and more steady progress.

Database Design:
- An Entity-Relationship (ER) diagram is indispensable for defining a database structure's key aspects in an application. It identifies entities, attributes, and the relationships between them, allowing us developers to ensure the correct use of data, fast recollection, ease of access to the user. In our case we decided to implement it for: user accounts management, save posts both answers and questions, admin records and changes, and on.

User Stories:
- User stories are indispensable, concise descriptions of a feature or functionality, written from the user's needs. They help us understand their needs and design solutions focused on their specific requirements. User stories form part of the main objectives, guiding development towards what should be prioritized based on user needs. They have let us see the needs like question creation, replying with answers, the direct management of pots, user account management, etc.

Process Diagrams:

[*] Universidad Distrital Francisco José de Caldas

- By utilizing process diagrams, intricate procedures become more manageable, and we like developers can communicate and collaborate with greater ease. Making these diagrams offers insights into the app's functions step by step and user use flow, allowing for streamlined optimization, more concrete and functional classes, and finally, improved final code.

Design and Architecture:
- Design Patterns: Applied software design patterns such as Singleton, Factory, Observer, and Strategy to ensure a modular, maintainable, and scalable architecture.
- System Architecture: Designed a layered architecture separating concerns into presentation, business logic, and data access layers.
- API Design: Created RESTful API endpoints using FastAPI to handle various operations such as user management, question and answer management, and voting.

Code Quality and Style Enforcement:
- Pylint: Integrated Pylint for static code analysis to enforce coding standards and identify potential errors.
- Black Formatter: Employed Black for consistent code formatting, ensuring the codebase remains uniform and readable.

Backend Implementation:
- FastAPI: Chose FastAPI for building the backend due to its high performance, ease of use, and automatic interactive documentation capabilities.
- Database Management: Implemented database models and operations using SQLAlchemy, providing an ORM layer to simplify database interactions.

Containerization and Deployment:
-Docker: Containerized the application using Docker, allowing for consistent deployment across different environments. Created Dockerfiles and Docker Compose configurations to manage the application's containers.

By integrating these methods and materials, our project demonstrates a comprehensive approach to building a high-quality backend for a Q&A forum. This approach ensures that the backend is not only functional and performant but also maintainable and scalable, addressing the diverse needs of modern web applications.

III. EXPERIMENTS & RESULTS

To ensure the backend of our Q&A forum met the required standards of functionality, performance, and scalability, we conducted a series of experiments. Through careful analysis, we have been able to gather valuable insight that will inform our future work. These experiments were designed to answer specific research questions to test hypotheses, the forward are such possible experiments:

Versions Testing:
Beta versions refer to software that is not yet fully developed, yet they are released to a select group of users to test in this case of a project us 2 developers and provide feedback on. These versions are used to see if there is any kind of error or feature that needs improving for the app's sake. These experiments search to help us understand better the user and the possible things we may have missed while coding. Beta testing is an important step in the software development process and helps ensure that the final product meets the needs and expectations of its users.

Database Scalability and Recurrence:
We aim to determine if the previously designed ER model is good enough so that it can be altered according to the new needs and requirements set by the app's future. In hopes of doing this, we must test if the database can maintain a constant increase of users and calls to the stored data.

Functional Testing:
The primary objective of functional testing was to verify that all the features of the Q&A forum backend worked as intended. We performed extensive tests on various endpoints, including user authentication, question submission, answer posting, and voting mechanisms, as well as real-time updates. Utilizing tools such as Pylint for automated testing, we confirmed that all core functionalities were operational and error-free

Scalability Testing:
Assessing the scalability of the backend involved deploying the application in a containerized environment using Docker. By scaling the number of containers, we tested the system's ability to handle increased load. Using Docker Compose for multi-container deployments and Kubernetes for more complex orchestration, we found that the backend scaled effectively. Performance and reliability were maintained with additional containers, and Kubernetes orchestration provided seamless scaling and improved resource management.

Integration Testing:
Integration testing aimed to ensure that all components of the system worked together seamlessly. We verified the interactions between FastAPI endpoints, the database managed by SQLAlchemy, and real-time communication handled by WebSockets. and Docker Compose to simulate a production-like environment, we confirmed that data flowed smoothly between the API, database, and real-time features, with all components interacting correctly.

In conclusion, the results of our experiments demonstrate that the backend for the Q&A forum is robust, scalable, and performs well under various conditions. The integration of modern development tools and practices, such as FastAPI, Docker, and CI/CD pipelines, played a significant role in the project's success. The backend is well-prepared to support a dynamic and engaging Q&A forum, as evidenced by the positive outcomes from functional, performance, scalability, integration, and user acceptance tests.

IV. CONCLUSION

The development and testing of the backend for our Q&A forum project have yielded several key insights and outcomes, demonstrating the effectiveness of our approach and the tools used.Firstly, leveraging Python and FastAPI for backend development proved to be an excellent choice. FastAPI's ability to provide high performance, ease of use, and automatic interactive documentation greatly facilitated the development process.

Our approach to ensuring code quality and consistency through Pylint and Black was highly effective. Pylint helped us maintain high coding standards by identifying potential issues early, while Black ensured that our codebase remained clean and uniform, improving readability and collaboration.

The use of Docker for containerization significantly streamlined the deployment process. Containerizing the application ensured that it could run consistently across different environments, simplifying development, testing, and production deployment. The scalability testing demonstrated that our backend could effectively handle increased load by scaling horizontally, especially when orchestrated with Kubernetes.

In conclusion, the backend for our Q&A forum is well-designed, robust, and scalable, capable of supporting a dynamic and engaging platform for knowledge exchange. The use of modern development practices and tools, including FastAPI, and Docker, contributed significantly to the project's success. This project not only demonstrates the practical application of software design patterns and best practices in backend development but also highlights the importance of thorough testing and user feedback in creating high-quality web applications.

REFERENCES

[1] C. A. Sierra (2024, March 19). Software Modeling [Online]. Available: https://github.com/EngAndres/ud-public/tree/main/courses/software-Modeling

[2] *Design patterns* (no date) *Refactoring.Guru*. Available at: https://refactoring.guru/design-patterns (Accessed: 01 June 2024).