A Practical Approach to Handling a Project Using Agile Principles

Ramya Peri
Technical Product Manager, Product Management, Greater Munich Metropolitan Area, Germany
Email: ramyaperi2@gmail.com, linkedin.com/in/ramya-peri

Managing any project efficiently requires careful planning, thoughtful architecture, and effective implementation. This article will explore a practical approach to building a student management system, utilizing agile principles for a more iterative and adaptable development process. The project's example will serve as a reference throughout the article.

Example: Building a Student Management System

Requirements Gathering and Use Cases:

The first step in any project is to gather requirements and brainstorm use cases. For a student management system, it is essential to understand the existing processes within the university or college. This might involve interviewing key stakeholders, such as clerks, the Head of Department (HOD), and principals, to comprehend the workflow and approval processes.

To visualize and document these processes, a process mapping diagram with swimlanes can be used. This will help identify bottlenecks, improvement areas, and a clear understanding of how the system should function.

Next, to capture the user needs effectively, create user stories. User stories should be concise, describing a single functionality from the user's perspective, regardless of their role. Keeping them small makes it easier to manage and prioritize tasks throughout the development process.

The most basic user stories indispensable to the system's functionality should be included in the first release. Prioritize them over other features to ensure a minimum viable product (MVP) is delivered early on. This approach aligns with the Agile development philosophy of incremental progress and continuous improvement.

Architecture:

Having a well-defined architecture is crucial to ensure scalability, maintainability, and efficiency of the student management system. The architecture of the given case study is depicted in Fig. 1. In this example, we will consider the following components for the architecture:

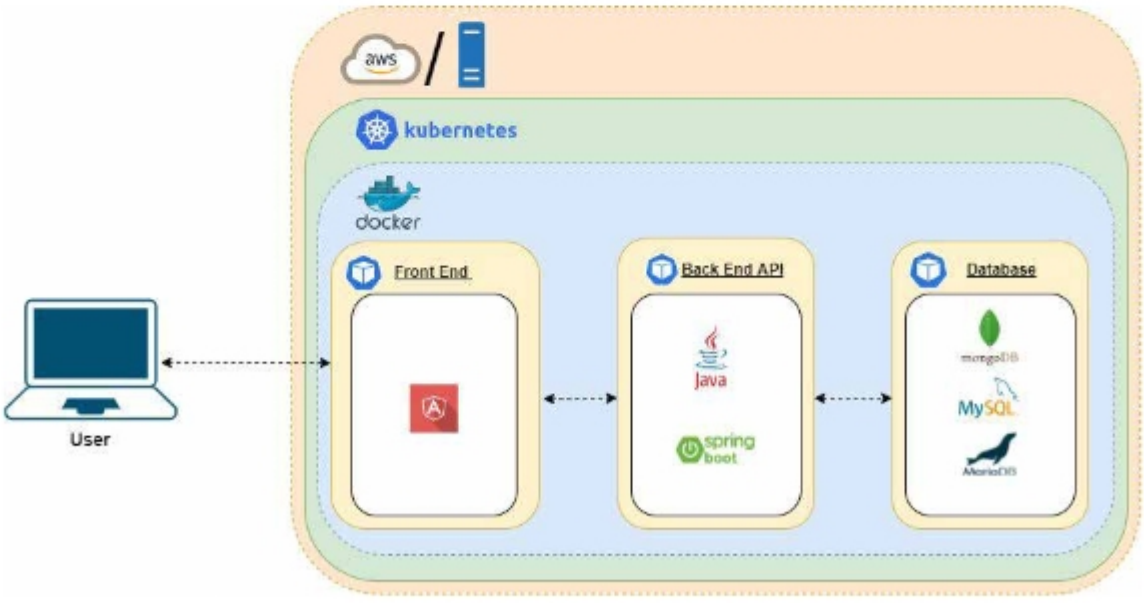

Fig. 1

https://doi.org/10.5281/zenodo.8250073

Frontend: Angular - Angular is a popular frontend framework based on TypeScript and CSS, providing a robust and modular structure for building responsive user interfaces.

Backend: Java Microservices - Microservices architecture allows breaking down the application into smaller, independent services, enabling easier development, testing, and deployment.

Database: A relational database like MySQL can be used for record tracking. For actual storage of student data in JSON format, MongoDB, a NoSQL database, can be utilized for its flexibility and scalability.

Deployment: Docker - Docker containers provide an isolated and consistent environment for deploying applications, making it easier to manage and scale the system efficiently.

The choice between on-premise or cloud deployment should also be based on budget, scalability requirements, and the organization's policies.

Implementation:

Before diving into coding, assessing if any existing solutions can be readily integrated to fulfill specific requirements is crucial. For example, integrating Google Sign-In or Azure Sign-In could be viable options for student authentication. These services offer secure and user-friendly authentication methods at minimal costs, saving development time and resources.

Adopting an agile approach should divide the implementation phase into short iterations or sprints, where specific user stories are developed and tested within each sprint. This iterative process enables the team to receive feedback from stakeholders regularly and make necessary adjustments throughout the project's lifecycle.

Conclusion:

Handling a student management system project requires a systematic and agile approach to ensure completion. The team gains clarity on the project's scope by gathering requirements, mapping processes, and creating user stories. The chosen architecture ensures a scalable and efficient system, including Angular for the front end, Java microservices for the backend, and MySQL and MongoDB for databases. Leveraging Docker for deployment enables consistency and easy management.

Moreover, exploring existing solutions for certain functionalities saves time and resources during implementation. Following an agile methodology, the team can adapt to changing requirements, receive timely feedback, and deliver an effective student management system that meets user needs and expectations. Building a student management system exemplifies how a practical approach combined with agile principles can lead to successful project outcomes.

About the Authors

Ramya Peri is an accomplished product management leader with a solid academic foundation, having completed her undergraduate studies at Andhra University. With an impressive 15-year tenure in the IT industry in India and beyond, Ramya has honed her skills and expertise to become a recognized authority in her field. Throughout her career, Ramya has been instrumental in handling projects for multinational clients, garnering exceptional feedback for her technical acumen and seamless team integration. Her ability to cultivate positive working relationships with diverse team members has driven her success. Notably, Ramya demonstrates exceptional coordination and issue-resolution capabilities between various technical infrastructure and application teams, ensuring project timely and effective implementation. Additionally, her solution-driven leadership abilities shine through in her adeptness at organizing employee engagement activities in collaboration with Human Resources. As a testament to her contributions and accomplishments, Ramya Peri continues to leave a lasting impact in the IT industry and beyond, solidifying her reputation as a proficient and respected professional.