Sprint 3 Individual Reflection Essay

Name: Yi Zheng Tang

Student ID: G00425075

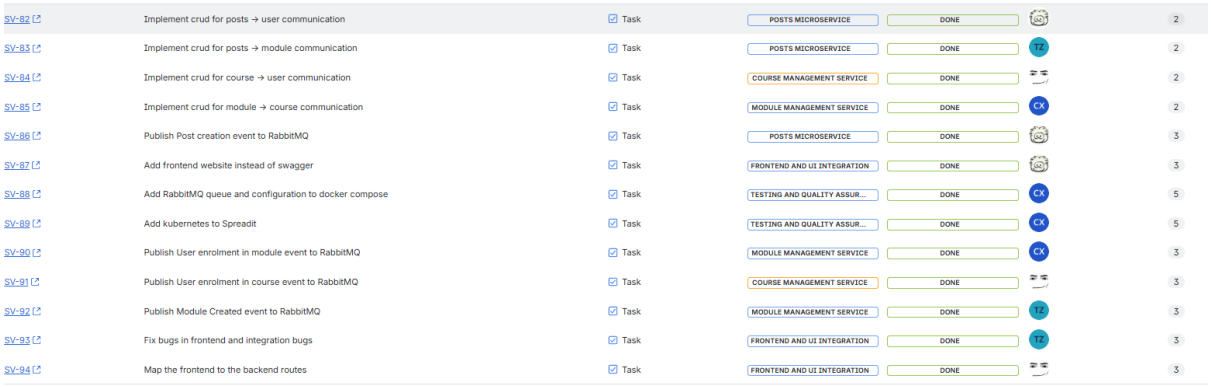
For sprint 3, we focused on containerization, cross-microservices communication, and quality assurance throughout the sprint. We did a great progress in building out the core microservices infrastructure, we successfully connected the microservices together with event-driven communication. Towards the end of the sprint we also migrated our full stack development project, mapped all routes to our microservices, makes it the frontend of our project. Due to project deadlines and submissions from other modules, we had to start the sprint late, but with good teamwork and communication, we managed to deliver most of the planned features within one day.

The features we completed in this sprint:

1. Containerized Post and Course microservices.
2. Migrated Post microservices from MongoDB to PostgreSQL
3. Integrated CI pipelines across all microservices.
4. Established Kubernetes, RabbitMQ

Below are the tickets we have for the sprint:

SV-54	Use Docker containerization + compose on posts microservice	Task	POSTS MICROSERVICE	DONE		5
SV-55	Migrate Posts micro service to Postgres	Task	POSTS MICROSERVICE	DONE		2
SV-56	Add pytests for posts mircroservice	Task	TESTING AND QUALITY ASSUR...	DONE		8
SV-57	Connect Course to User	Task	COURSE MANAGEMENT SERVICE	DONE	TZ	3
SV-58	Connect Posts to User	Task	POSTS MICROSERVICE	DONE	CX	3
SV-59	Connect Posts to Module	Task	POSTS MICROSERVICE	DONE		3
SV-44	Connect module to course	Task	MODULE MANAGEMENT SERVICE	DONE	CX	3
SV-65	Use Docker containerization + compose on Course microservice	Task	COURSE MANAGEMENT SERVICE	DONE	TZ	5
SV-67	Integrate CI Pipeline to each microservice	Task	CICD AND DEVOPS	DONE	TZ	3
SV-69	Add missing Patch endpoint for Course Microservice	Task	COURSE MANAGEMENT SERVICE	DONE		1
SV-70	Add more comprehensive tests for Course Microservice	Task	COURSE MANAGEMENT SERVICE	DONE		2
SV-81	Updated the Posts microservice endpoints status codes	Task	POSTS MICROSERVICE	DONE		2



SV-82	Implement crud for posts → user communication	Task	POSTS MICROSERVICE	DONE		2
SV-83	Implement crud for posts → module communication	Task	POSTS MICROSERVICE	DONE	TZ	2
SV-84	Implement crud for course → user communication	Task	COURSE MANAGEMENT SERVICE	DONE		2
SV-85	Implement crud for module → course communication	Task	MODULE MANAGEMENT SERVICE	DONE	CX	2
SV-86	Publish Post creation event to RabbitMQ	Task	POSTS MICROSERVICE	DONE		3
SV-87	Add frontend website instead of swagger	Task	FRONTEND AND UI INTEGRATION	DONE		3
SV-88	Add RabbitMQ queue and configuration to docker compose	Task	TESTING AND QUALITY ASSUR...	DONE	CX	5
SV-89	Add kubernetes to Spreadit	Task	TESTING AND QUALITY ASSUR...	DONE	CX	5
SV-90	Publish User enrolment in module event to RabbitMQ	Task	MODULE MANAGEMENT SERVICE	DONE	CX	3
SV-91	Publish User enrolment in course event to RabbitMQ	Task	COURSE MANAGEMENT SERVICE	DONE		3
SV-92	Publish Module Created event to RabbitMQ	Task	MODULE MANAGEMENT SERVICE	DONE	TZ	3
SV-93	Fix bugs in frontend and integration bugs	Task	FRONTEND AND UI INTEGRATION	DONE	TZ	3
SV-94	Map the frontend to the backend routes	Task	FRONTEND AND UI INTEGRATION	DONE		3

My task for this sprint is to ensure test coverage, and quality assurance. To achieve that:

1. I developed comprehensive pytest as we are writing new endpoints and routes.
2. Ensure the coverage maintains above 85% across all services.
3. Ensure test reliability and maintainability for future development.

I also did some backend development, by adding missing endpoints to microservices to support full CRUD functionality, assisted in connecting microservices by implementing data retrieval logic. I also helped maintain consistent API endpoint patterns across microservices.

The biggest problem we faced is time management, for the past two weeks our sprint was heavily delayed due to other module's project deadline and submission. We have to start our CICD sprint nearly towards the end. For the frontend implementation, we faced migration complexity as the routing and endpoint mapping is a long and complex process.

But with good teamwork and good communication, we decided to come in together during Friday, and we managed to get everything done within that day.

For the next sprint, we have a roughly plan of what we should do, we will finish up all the microservices communication, while also conduct end-to-end testing to identify and resolve any remaining bugs during the integration process. Replace frontend local storage with direct PostgreSQL database connections, and validate data persistence across the full stack. And lastly is to integrate external API as required by the project rubric.