

We implemented an api-gateway microservice in conjunction with Keycloak to route access through its authenticator. Swagger was also implemented and routed through for documentation purposes; however, it is not accessible currently through the gateway – only by accessing the microservice directly.

The major challenge we faced was getting the project to clone properly onto Simon's IDE. None of the modules were detected. We surmise this was due to the project being created in Groovy and not Kotlin.

Due to time constraints, Enhanced Inter-Service Communication with Kafka and Schema Registry and Circuit Breaker Pattern with Resilience4J were not implemented.

We learned much about dockerization, Postman testing and authentication.

[x] We setup Spring Cloud Gateway as the entry point to route requests to our microservices.

[x] Integrated Keycloak to manage authentication:

[x] Configured the API Gateway to use Keycloak for securing all microservices.

[x] Ensured that only authenticated users can access specific services (e.g., Booking and Approval services).

[o] Role-based access should be enforced through Keycloak ▪ Research & Implement → Only staff can access **ApprovalService**.

[o] Provides accessible documentation for all REST APIs through the API Gateway.

[x] Integrates **Swagger** into each microservice to document available endpoints and methods.

