

SS22053 – Architecting a Backend System using Modern Methodologies and Frameworks

Presented by Leow Ken Hing Bryan

Supervised by Assoc Prof Ringo Ho, Co-supervisor Mr Pham Tan Phat

Motivation

- Architecting a modern backend system for an application has always been a technological challenge.
- Issues of development speed come to mind in the short-term, while that of maintainability, security, and scalability come to mind in the long-term.**
- We implement our systems with these non-functional ISO25010 requirements in mind and provide a qualitative analysis of the trade-offs.

Methodology

- Databases:** relational databases are good for maintainability and performance, especially for highly normalised data like questionnaire scores.
- Language:** Java's verbosity makes it difficult initially, but readability improves over time with exposure. Mature MVC and logging frameworks like Spring Boot and SLF4J respectively allow for a robust codebase with long term maintainability.
- API Paradigm:** a JSON-like structure is perfect for data visualisation with little overhead as opposed to RPC. A mutable endpoint like GraphQL is also not needed for our well-defined screens.
- Authentication:** JWTs allow for stateless session management, authentication, and authorisation.

Findings

Changing requirements e.g. in questionnaires and scoring mechanics necessitate writing clean and robust code. GOF design patterns such as the strategy design pattern are crucial in ensuring project maintainability.

Requirement priority is also often project-specific. Localisation, for example, is extremely crucial for our application, when it is often overlooked in commercial applications.

Future Work

Deploying onto the cloud, such as AWS

