STDISCM - Problem Set 4 Distributed Fault Tolerance

by Lord John Benedict Pinpin
and Brett Harley Mider

Implementation

- Separate service for each feature (one each for the front-end, authentication, courses, enrollment, viewing and uploading student grades)
- Each service is hosted in its own Docker container on different ports
  - The centralized database was also hosted on its own Docker container, which was shared between the services.
- Coded in Java, used Javalin for the front-end and MVC, PostgreSQL for database, JWT for authentication and session management

Fault Tolerance

- Each program/service was given its own Docker container and runs with its own services. They would be able function on their own even without the presence of the other services. This means that if one of the programs failed, the other wouldn’t fail immediately alongside it.
- For example, if the courses service failed, the auth service would still be running. If we built more services, then those services failing wouldn’t affect the others either.
- The dockerized services can also be replicated, then load balanced whenever certain services fail to make sure uptime isn’t compromised.
- The dockerized services can also self-check and perhaps even restart by itself without requiring an admin to fix it, unlike perhaps an entire system crashing due to just one part failing. This also avoids having to troubleshoot for one small issue.