Francois Theron 11-03-2019

Assignment description:

https://github.com/chamatheapp/chama-backend-assignment-course-signup

Candidate Profile:
https://www.linkedin.com/in/francois-theron-38b70788/

Applicant's result:

My project: github.com/FTheron/LearningHub

	Riaan	Andrei	Comments
Coding Skills			
• Writing testable code	✅	✅	RR: Uses interfaces where appropriate, business rules are encapsulated in single place easy to test.
• Test infrastructure	❌	❌	RR: Not present, the folder was created, but no tests implemented, wonder if the candidate has experience with unit test frameworks or any sort of testing patterns.
• Exception handling and logging	❌	❌	RR: No guards within public methods, no exception handling or logging of control flow or errors. Logger added to services but not used anywhere. ✅ the validators is a very nice touch on the input parameters, so error responses are returned instead of exceptions.
• Asynchronous code (when and how (not) to use async/await)	⚠️	⚠️	RR: Awaits all tasks, not delaying execution till the utmost end of the call stack, could potentially indicate that the candidate doesn't understand how tasks work. Return types are tasks but there are no asynchronous flows within the method i.e GetCourseDetail

• Asynchrony through messaging (events, commands)	⚠	⚠	RR: able to publish an event, onto a ServiceBus queue, a simple consumer was built, but doesn't take advantage of .NET core WebHost, so no DI within the consumer, not a problem but could have been cleaner. Directly using the repository within the consumer, shows a lack of understanding of the Application Services within a domain model
• Knowledge of Azure infrastructure and storage technologies platform (WebJobs, Azure Functions,SQL, Table Storage, Cosmos DB, ...)	⚠	⚠	RR: Service Bus used, no other azure infrastructure used. Used an abstraction on top of Entity Core, good as a business developer, but doesn't showcase that the candidate has knowledge of how it works or how to set it up without the helper package.
Software Engineering Skills			
• Code organisation (modularity, dependencies between modules, etc)	✅	⚠	RR: a clean separation between business, application and infrastructure components. AC: it seems the project is spread across different modules with no clear indication on what purpose they serve (touches on both layered architecture and DDD) but none seem implemented in a clear way.
• Namings	✅	✅	

• Domain model design (OOP, DDD concepts, etc)			RR: understands encapsulation, polymorphism but doesn't apply it everywhere, StudentDomain is newed up everywhere, why not abstract it and inject it with DI. DDD knowledge seems to come from the DotNetCore NuGet packages used not the candidates own knowledge would inquire if the candidate understands why things are separated on the specific boundaries, and how to decompose the domain to get to the different parts implemented in the assignment. AC: DDD concepts and layered architecture are both used without a clear scope (both anemic and rich models, both services and domain objects, repositories and units of work used everywhere) data access suffers the most here - there are repositories and units of work being used and passed around to other classes but then there's no tracking to whether the instance is being manipulated (commited) or not, there's only an assumption of that
• Handling concurrency (avoid exceeding the course capacity)			RR: implemented through the DB abstraction, checks the current DB count before continuing, good attempt but if there are many concurrent requests to enroll into a single course then multiple could end up succeeding while the capacity should not allow it. AC: out-of-process concurrency not being handled
Overall Feeling			

▪ Assignment completion	⚠	⚠	RR: implement parts 1 and 2, part 3 is partially complete as the assignment explicitly asks not to implement the aggregation on every request. Would discuss how the candidate proposes to solve this without calculating on each request. Candidate implemented an InMemoryCache which is potentially a solution, but how would he handle a distributed computing model, where you have little control which node/instance of your application would handle that request. It is not a scalable solution.
▪ Overall code quality (edge cases, usage of tools)	⚠	⚠	RR: Overall good assignment, but depended heavily on a framework DotNetCore. Which is not bad per se, would have to discuss on whether the candidate understands why you do some things versus others.