Architectural Validation Report

Kieran Foy SAAD Assessment

Scope

Use case 4: Applying for a visa

Scope Justification

The current scope of my system that I will be exploring and demonstrating with a proof of concept includes use case 4 - applying for a visa. This subset of my scope covers the main bulk of the system and what I also believe is the main use case of the system which is, applying for visas utilising the system that I am creating.

By implementing this into my proof of concept I should be able to aptly demonstrate and validate the architecture of my system. Furthermore I can demonstrate the several non-functional requirements that my system accounts for.

Demonstrated & Validated NFRS & Functional Requirements

- Maintainability - code structured logically
- Scalability - code is loosely coupled allowing for new feature implementations
- Usability - The site layout is obvious and takes very few clicks to navigate.
- Localisation - The site provides language options for those defined in my scope.
- Accessibility - The site colour scheme is easy to read and provides dark mode alternatives for those who don’t like to use a high brightness.

Validation Technique and Justification

Proof of concept prototype

I have chosen to use a prototype system developed in C# using the .NET Blazor framework this is a very lightweight and simple to setup technology stack of which I can easily demonstrate the architecture of my software system.

I believe using a proof of concept prototype is the most appropriate method of validating my system architecture as you can clearly see the actual entities implemented into their classes and objects and how they interact on a code level, this can be directly referenced back to my C4 code diagram and data models.

Any changes necessary upon building the prototype will be very apparent as the program will either not run, be too complicated or run slowly. Any of these issues encountered at this stage can be resolved and I will outline the changes made to account for this in the video demonstration.

Furthermore I can show the flexibility of my code structure through the demonstration of a unit test. I intend to use this as an example of how easy the implemented code structure is to test in parts and how this would not be possible if my code was based on concretions.

Video showcasing the prototype to review the architecture

Please find attached in the ZIP the video “Architectural Validation Demonstration” showcasing the review of the system architecture and how the functional and non-functional requirements within my subset of the scope have been met.

A discussion on improvements made due to the validation

I originally included the Factory builder design pattern however I believe this provided no extra utility to my system and instead added unnecessary complexity. This is because there were far too many subclasses for every different type of visa and when the system grows this would become even more challenging to handle.

Furthermore it did not make sense to have separate classes for each type of visa regardless as I determined that they in fact did not have differing functionality depending on the type.

The solution I implemented was to keep VisaType as a general class which is used by VisaApplication. The VisaType class itself is just an enum containing the different types such as Work, Travel, Business, etc. Based on this my system can handle an increasing number of different types regardless and it makes more sense in this context as they don’t need any differing functionality or even functionality at all in my current implementation of the system.

When developing the proof of concept prototype I also included Logger as a proof of how event sourcing would be incorporated into my system. I felt this was a valuable demonstration of how useful it could be when dealing with sensitive data. From the demonstration in the video one can see how a trail of the VisaApplication would appear and how this would be useful for auditing purposes.

Validation outcome

I believe that with this proof of concept prototype I have been able to effectively validate my system architecture. The produced artefact demonstrates the main functionality of the system of which is applying for a visa so that requirement has been met. This can be measured using the user acceptance criteria and the unit test.

The unit test I included has a direct link to the user acceptance tests in the user stories section and you can see how it is used to validate that the functional requirements for this subset of my scope has been met. I feel the unit test was extremely useful in proving that my system architecture and code structure in general is loosely coupled and allows for testing without needing to change any of the code. (Kovač, 2023)

The highest priority non-functional requirements that I outlined have also been met as I have shown that the system is Maintainable in the way that code that performs a single action can easily be found in its own service as opposed to a monolithic approach to code structure which is not maintainable.

Also due to the loose coupling using a repository pattern with interfaces detailing a services' functions and then a class implementing those functions, I have a scalable system that can be added on to as and when needed without having an effect on the rest of the system as concretions would result in.

The site is accessible and supports localisation via the language and dark mode implementations to appeal to a wider audience and account for those who struggle with their non-native language.

Finally I believe that I managed to achieve a relatively straightforward system requiring very few clicks to navigate and that this should promote a good user experience overall.

In conclusion my architecture was not perfect however during the validation I made several changes for the good that I believe significantly improved the final iteration of the prototype and leading into the full development of the web application this would be a very useful tool.

References

Kovač, R. (2023, April 23). *Unit Testing in .NET 7 with xUnit | by Roko Kovač*. Medium. Retrieved December 22, 2023, from https://medium.com/@kova98/unit-testing-net-7-code-with-xunit-a0cfbca75599