Architecture & Design

Kieran Foy SAAD Assessment

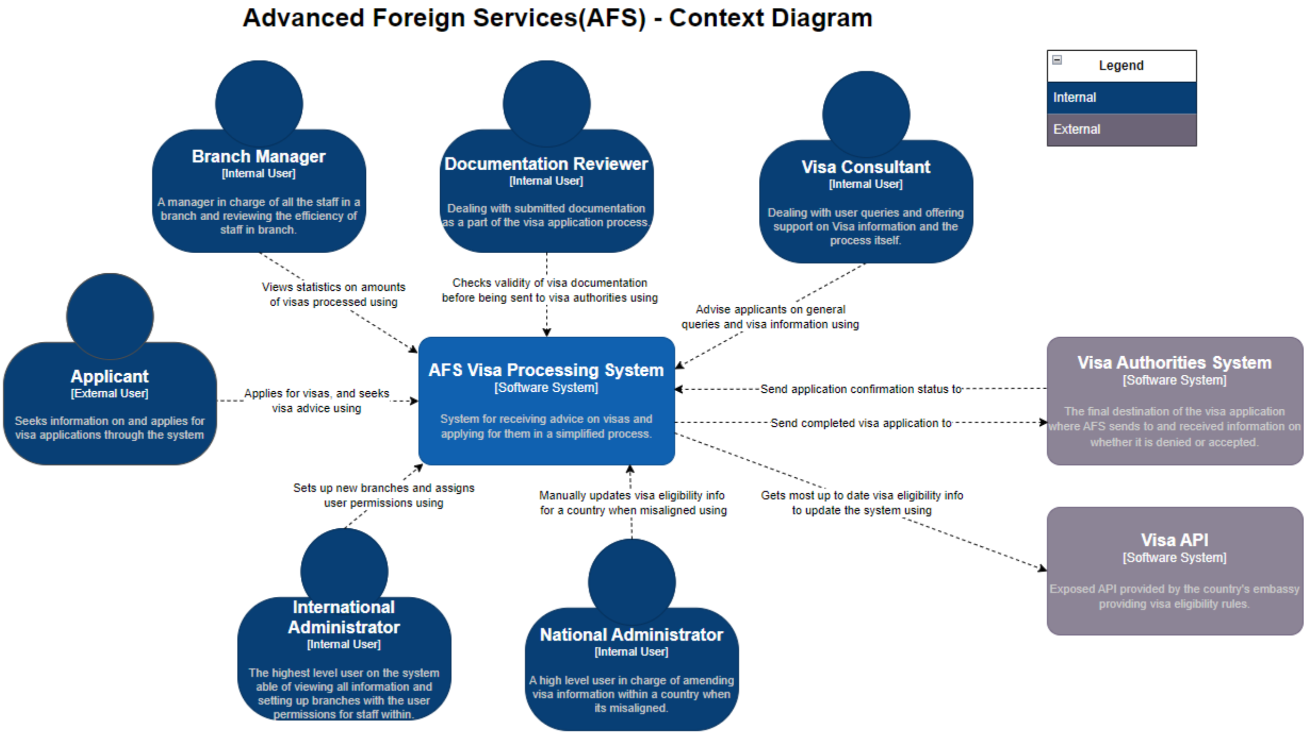
Architectural Style

I have chosen the Layered architectural style please see ADR and container diagram for further information.

C4 Model

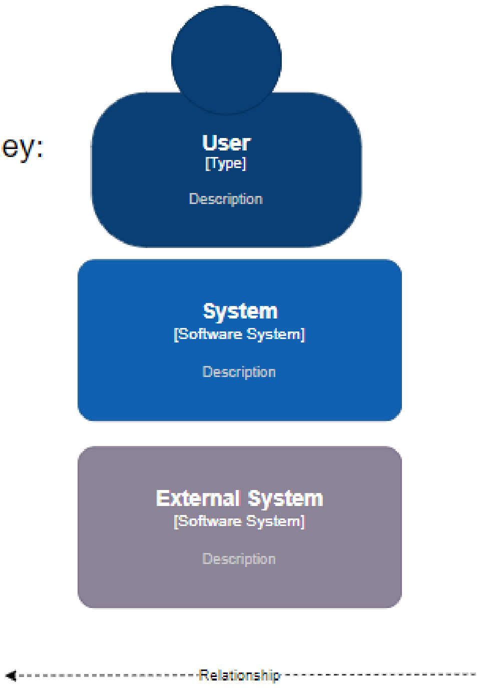
Level 1 - Context Diagram

C4 Context Diagram


Advanced Foreign Services(AFS) - Context Diagram
Legend
Internal
External
Branch Manager
[Internal User]
A manager in charge of all the staff in a branch and reviewing the efficiency of staff in branch.
Documentation Reviewer
[Internal User]
Dealing with submitted documentation as a part of the visa application process.
Visa Consultant
[Internal User]
Dealing with user queries and offering support on Visa information and the process itself.
Applicant
[External User]
Seeks information on and applies for visa applications through the system
Views statistics on amounts of visas processed using
Checks validity of visa documentation before being sent to visa authorities using
Advise applicants on general queries and visa information using
Applies for visas, and seeks visa advice using
AFS Visa Processing System
[Software System]
System for receiving advice on visas and applying for them in a simplified process.
Send application confirmation status to
Send completed visa application to
Visa Authorities System
[Software System]
The final destination of the visa application where AFS sends to and received information on whether it is denied or accepted.
Sets up new branches and assigns user permissions using
Manually updates visa eligibility info for a country when misaligned using
Gets most up to date visa eligibility info to update the system using
Visa API
[Software System]
Exposed API provided by the country's embassy providing visa eligibility rules.
International Administrator
[Internal User]
The highest level user on the system able of viewing all information and setting up branches with the user permissions for staff within.
National Administrator
[Internal User]
A high level user in charge of amending visa information within a country when its misaligned.

Key is detailed below

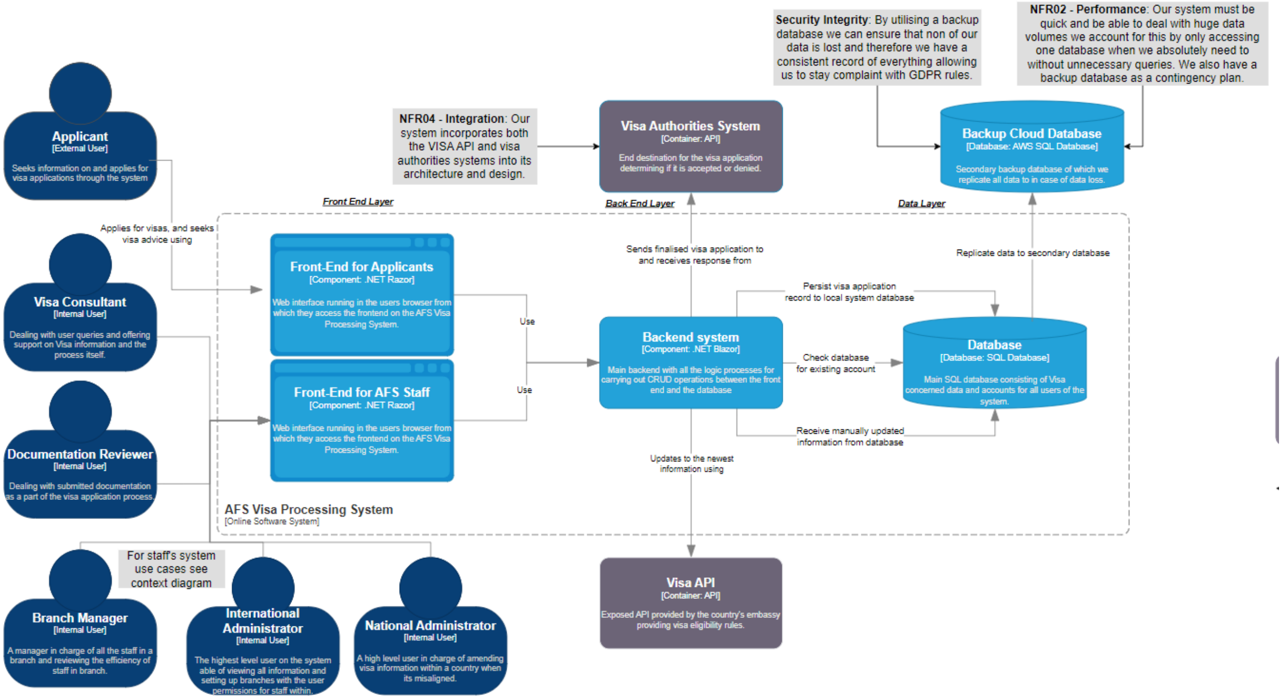
Key:



Level 2 - Container Diagram

[C4 Container Diagram]

Advanced Foreign Services(AFS) - Container Diagram


Security Integrity: By utilising a backup database we can ensure that non of our data is lost and therefore we have a consistent record of everything allowing us to stay complaint with GDPR rules.

NFR02 - Performance: Our system must be quick and be able to deal with huge data volumes we account for this by only accessing one database when we absolutely need to without unnecessary queries. We also have a backup database as a contingency plan.

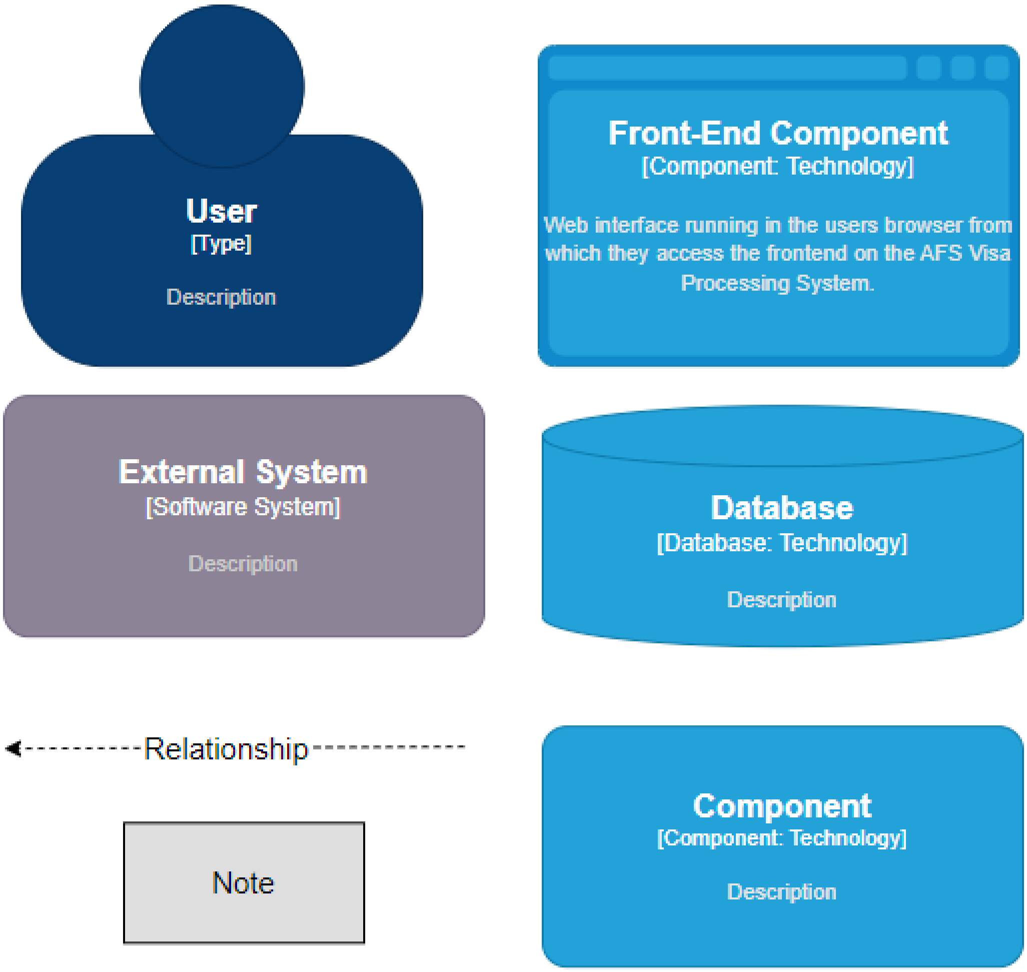
NFR04 - Integration: Our system incorporates both the VISA API and visa authorities systems into its architecture and design.

For staff's system use cases see context diagram

Applicant
[External User]
Seeks information on and applies for visa applications through the system
Applies for visas, and seeks visa advice using
Visa Consultant
[Internal User]
Dealing with user queries and offering support on Visa information and the process itself.
Documentation Reviewer
[Internal User]
Dealing with submitted documentation as a part of the visa application process.
Front End Layer
Front-End for Applicants
[Component: .NET Razor]
Web interface running in the users browser from which they access the frontend on the AFS Visa Processing System.
Front-End for AFS Staff
[Component: .NET Razor]
Web interface running in the users browser from which they access the frontend on the AFS Visa Processing System.
Use
Use
Back End Layer
Visa Authorities System
[Container: API]
End destination for the visa application determining if it is accepted or denied.
Sends finalised visa application to and receives response from
Backend system
[Component: .NET Blazor]
Main backend with all the logic processes for carrying out CRUD operations between the front end and the database
Persist visa application record to local system database
Check database for existing account
Receive manually updated information from database
Data Layer
Backup Cloud Database
[Database: AWS SQL Database]
Secondary backup database of which we replicate all data to in case of data loss.
Replicate data to secondary database
Database
[Database: SQL Database]
Main SQL database consisting of Visa concerned data and accounts for all users of the system.
Updates to the newest information using
AFS Visa Processing System
[Online Software System]
Visa API
[Container: API]
Exposed API provided by the country's embassy providing visa eligibility rules.
Branch Manager
[Internal User]
A manager in charge of all the staff in a branch and reviewing the efficiency of staff in branch.
International Administrator
[Internal User]
The highest level user on the system able of viewing all information and setting up branches with the user permissions for staff within.
National Administrator
[Internal User]
A high level user in charge of amending visa information within a country when its misaligned.

Container Diagram developed with the layered system architecture in mind by separating personeel and levels of access to data from front end all the way to the database layer, which only administrators will be able to access.

Key:



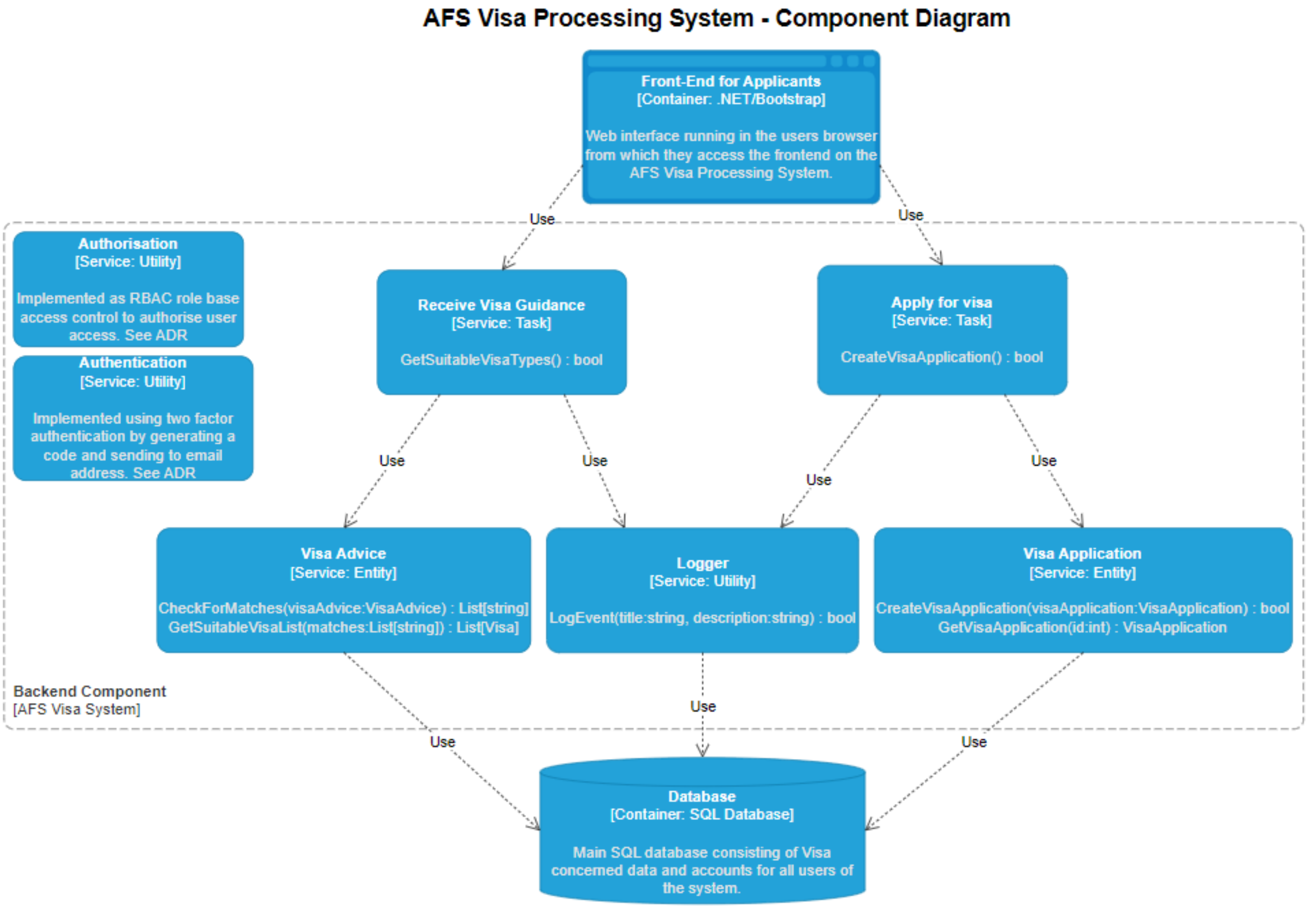
Level 3 - Component Diagram

C4 Component Diagram

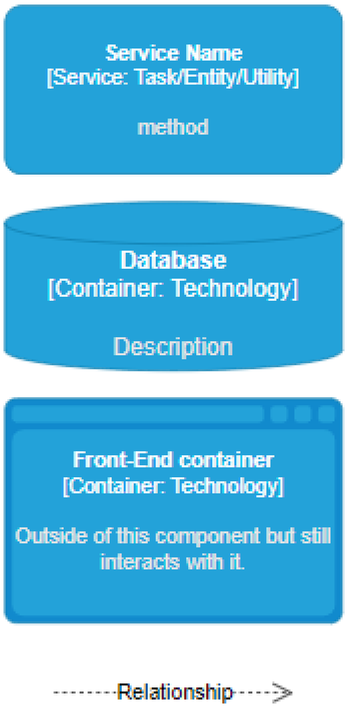
Scoped to:

- Use Case 2 - Receive guidance on the most relevant required Visa
- Use Case 4 - Submit a visa application

Structural Diagram


AFS Visa Processing System - Component Diagram
Front-End for Applicants
[Container: .NET/Bootstrap]
Web interface running in the users browser from which they access the frontend on the AFS Visa Processing System.
Use
Use
Authorisation
[Service: Utility]
Implemented as RBAC role base access control to authorise user access. See ADR
Authentication
[Service: Utility]
Implemented using two factor authentication by generating a code and sending to email address. See ADR
Receive Visa Guidance
[Service: Task]
GetSuitableVisaTypes() : bool
Apply for visa
[Service: Task]
CreateVisaApplication() : bool
Use
Use
Use
Use
Visa Advice
[Service: Entity]
CheckForMatches(visaAdvice:VisaAdvice) : List[string]
GetSuitableVisaList(matches:List[string]) : List[Visa]
Logger
[Service: Utility]
LogEvent(title:string, description:string) : bool
Visa Application
[Service: Entity]
CreateVisaApplication(visaApplication:VisaApplication) : bool
GetVisaApplication(id:int) : VisaApplication
Backend Component
[AFS Visa System]
Use
Use
Use
Database
[Container: SQL Database]
Main SQL database consisting of Visa concerned data and accounts for all users of the system.

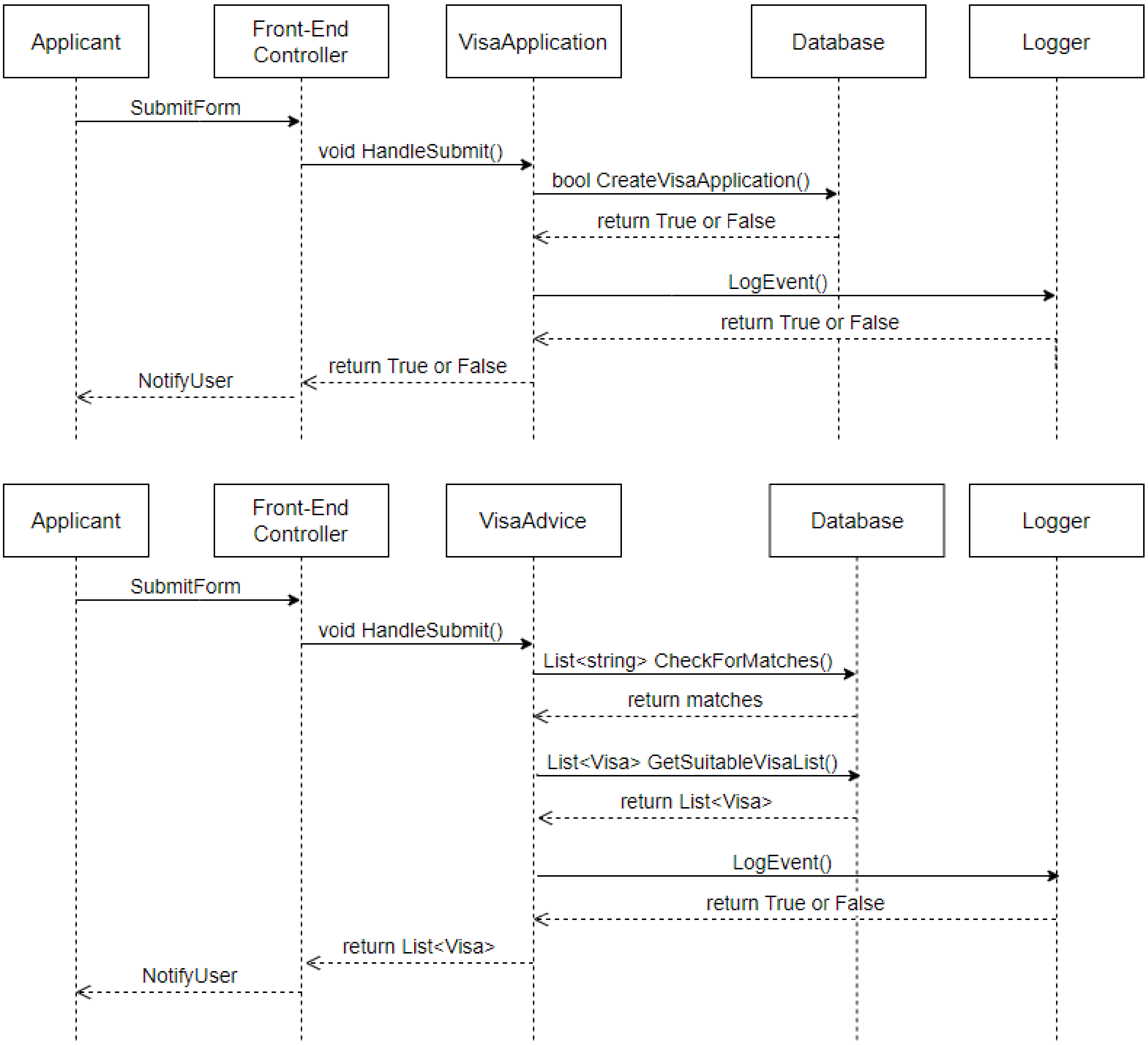
Key:



Behavioural Diagram

[Component Behavioural Diagram]

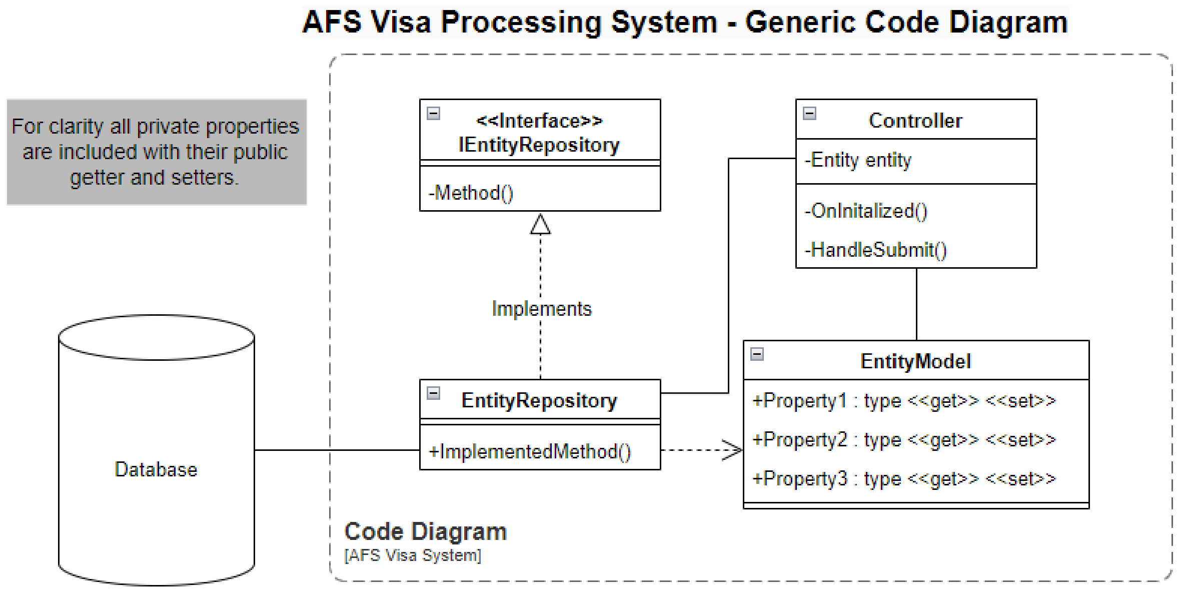
AFS Visa Processing System - Component Sequence Diagram



Level 4 - Code Diagram

[C4 Code Diagram]

Generic Diagram

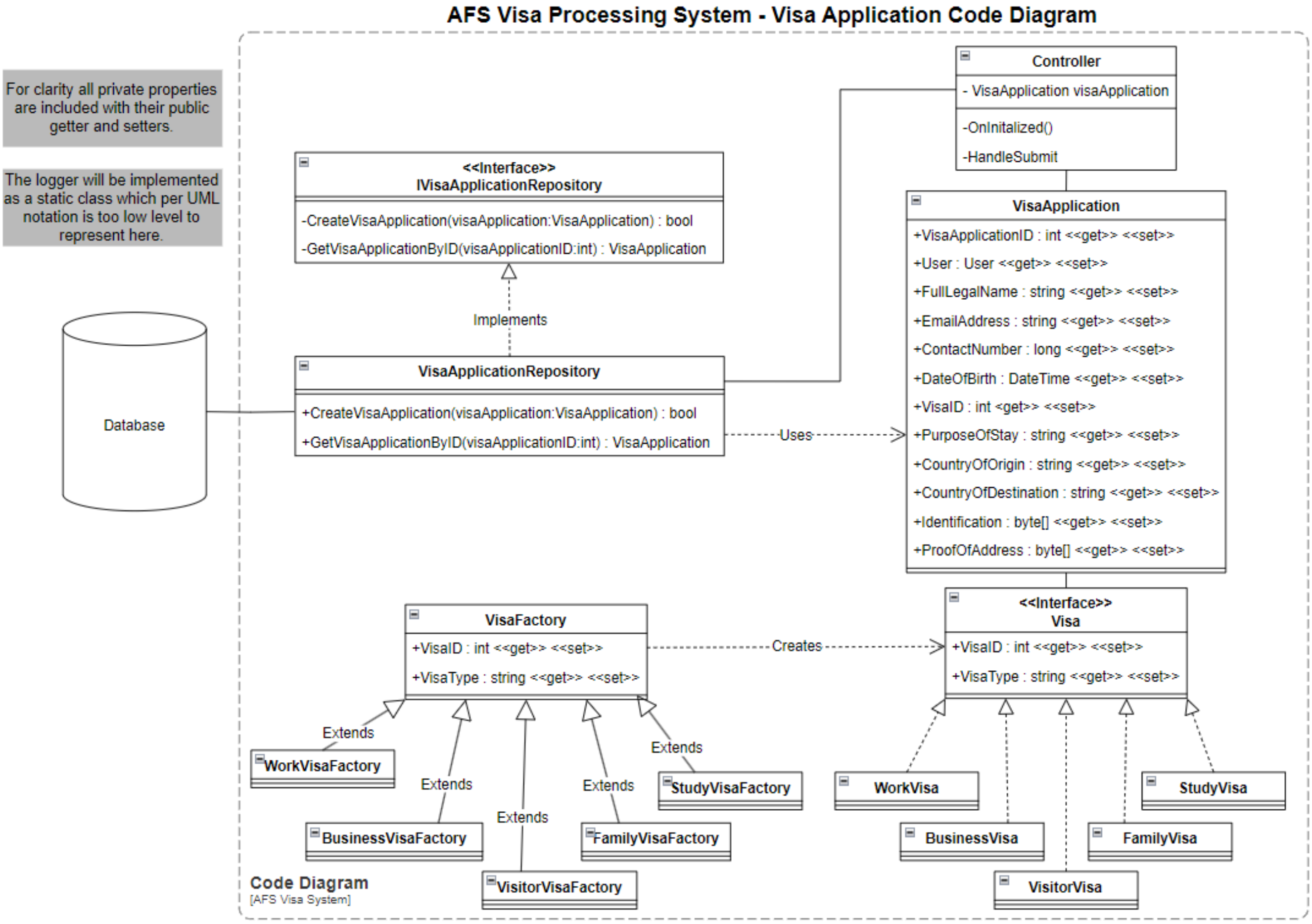

AFS Visa Processing System - Generic Code Diagram
For clarity all private properties are included with their public getter and setters.
<<Interface>>
IEntityRepository
-Method()
Implements
Controller
-Entity entity
-OnInitalized()
-HandleSubmit()
EntityModel
+Property1 : type <<get>> <<set>>
+Property2 : type <<get>> <<set>>
+Property3 : type <<get>> <<set>>
EntityRepository
+ImplementedMethod()
Database
Code Diagram
[AFS Visa System]

My code is structured using the **repository pattern** together with the use of dependency injection. Using this I can separate the concerns of services and their function and promote a maintainable and scalable code repository. (Brind, 2023)

I have chosen to use this pattern because I can have an interface stating all the functionality of an entity and then the actual implementation can be extended in whatever way possible, and can be changed and added upon as desired, I have achieved a loose coupling and if I need to change the code infrastructure in any way I can simply implement a repository that handles that such as changing the database from SQL to Oracle. (Pragimtech, 2020)

Specific Visa Application Code Diagram

**Please see Draw IO link for clearer image* C4 Code Diagram.


AFS Visa Processing System - Visa Application Code Diagram
For clarity all private properties are included with their public getter and setters.
The logger will be implemented as a static class which per UML notation is too low level to represent here.
Database
<<Interface>>
IVisaApplicationRepository
-CreateVisaApplication(visaApplication:VisaApplication) : bool
-GetVisaApplicationByID(visaApplicationID:int) : VisaApplication
Implements
VisaApplicationRepository
+CreateVisaApplication(visaApplication:VisaApplication) : bool
+GetVisaApplicationByID(visaApplicationID:int) : VisaApplication
Uses
Controller
- VisaApplication visaApplication
-OnInitalized()
-HandleSubmit
VisaApplication
+VisaApplicationID : int <<get>> <<set>>
+User : User <<get>> <<set>>
+FullLegalName : string <<get>> <<set>>
+EmailAddress : string <<get>> <<set>>
+ContactNumber : long <<get>> <<set>>
+DateOfBirth : DateTime <<get>> <<set>>
+VisaID : int <get>> <<set>>
+PurposeOfStay : string <<get>> <<set>>
+CountryOfOrigin : string <<get>> <<set>>
+CountryOfDestination : string <<get>> <<set>>
+Identification : byte[] <<get>> <<set>>
+ProofOfAddress : byte[] <<get>> <<set>>
VisaFactory
+VisaID : int <<get>> <<set>>
+VisaType : string <<get>> <<set>>
Creates
<<Interface>>
Visa
+VisaID : int <<get>> <<set>>
+VisaType : string <<get>> <<set>>
Extends
WorkVisaFactory
Extends
BusinessVisaFactory
Extends
VisitorVisaFactory
Extends
FamilyVisaFactory
Extends
StudyVisaFactory
WorkVisa
BusinessVisa
VisitorVisa
FamilyVisa
StudyVisa
Code Diagram
[AFS Visa System]

In the specific example of my code structure for the Visa Application service you can see that I have implemented a factory design pattern as a way of preparing my code for any necessary future implementation of new features. Or in this case new Visa Types that will inevitably need to be added in the future as the system grows.

Data Architecture

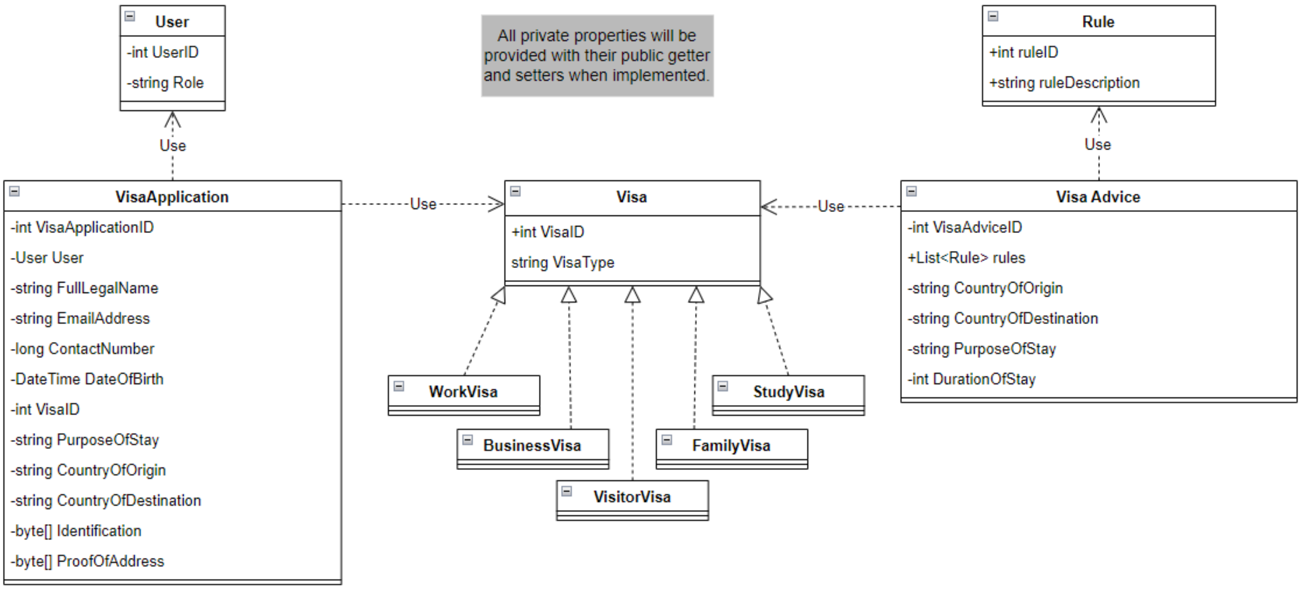
Data Model

The data model is presented as data only and so any functions have been omitted for brevity.

The data model is scoped to:

- Use Case 2 - Receive guidance on the most relevant required Visa
- Use Case 4 - Submit a visa application

AFS Visa Processing System - Data Model



Wireframes

Figma Wireframes

For the full set of wireframes please refer to the attached PDF in this zip file “**AFS Wireframes.pdf**”

Security

Permission Matrix

Permission ID	Business Task	Roles	Basic Permissions	Constraints
P-001	Receive guidance on the most relevant required Visa.	Applicant	Read	No constraints can be viewed without an account.
P-002	Submit a visa application.	Applicant	Create Update	Must be logged in. Must be completely filled in.