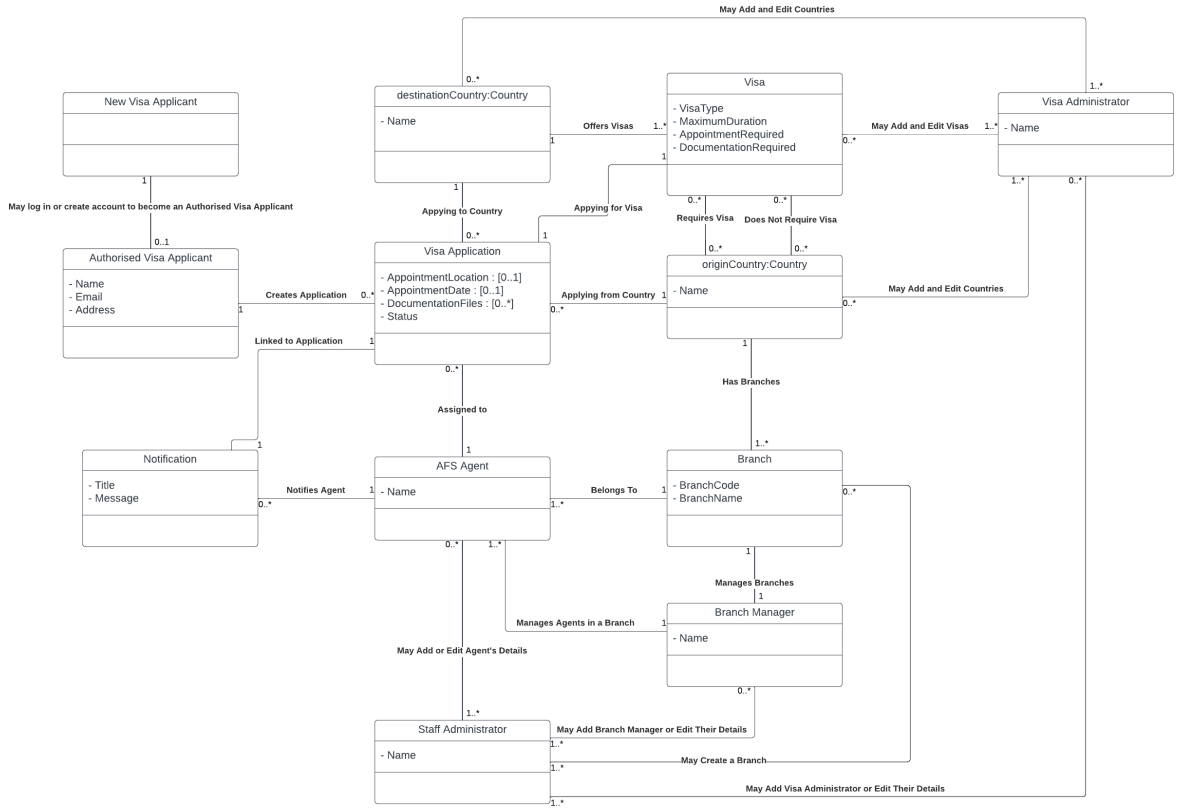
Advanced Foreign Services System Architecture

Domain

Element	Description
New Visa Applicant	May only use the system to determine what visa they need, or log in/create an account to become an authorised visa applicant.
Authorised Visa Applicant	May apply for visa to a destination country, from their origin country. May need to provide documentation or book an appointment depending on the visa they're applying for.
AFS Agent	Can review visa applications and reject them, request for extra information, or send the application to the relevant embassy.
Destination Country	The country for which the visa application is applying. May have many different Visas on offer.
Visa	A type of visa that a destination country offers. May not be required for certain origin countries.
Origin Country	The country from which the applicant creating the application originates.
Visa Application	Created by an authorised visa applicant, from an origin country to a destination country. Must be reviewed by an AFS Agent
Branch	Branches are where applicants can book appointments if required. AFS agents will belong to a branch.
Branch Manager	Each branch will have one branch manager who may view statistics and reports for a branch, and assign/re-assign applicants to agents.
Staff Administrator	Can create new branches, add staff members, and move them between branches.
Visa Administrator	Can add new countries into the system and can manually edit visa rules for a country.
Notification	Notifications are sent to AFS agents when a application is assigned or re-assigned to them, and includes a link to the application.

Domain Model

[Link to Domain Model]



Notes

Destination Country and Origin Country will be separate instances of Country. In the context of our scope, they will need to be differentiated as when a country is selected as a destination country will offer the visa, depending on the applicant's origin country. The origin country will link to the branches in that country, and the agent reviewing an application must be a part of a branch in the country of the applicant's origin.

Non-Functional Requirements Prioritization

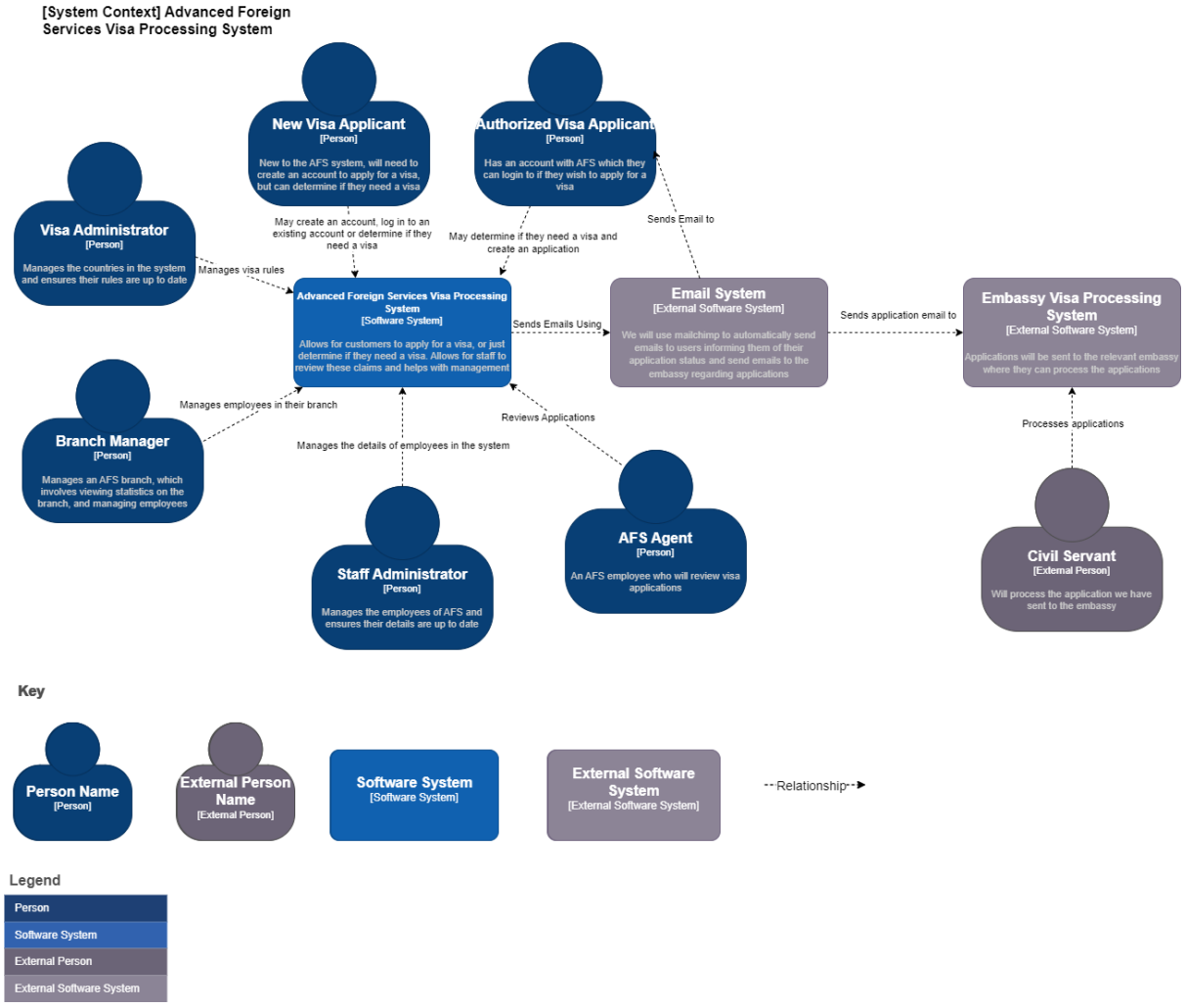
It is often the case that the prioritization of non-functional requirements is mostly ad-hoc and nonfunctional requirements are mostly ignored or neglected (Gupta et al., 2017), despite the fact non-functional requirements must be prioritized separately but in accordance with the functionality of the proposed system (Chopra et al., 2016). In light of this, I will outline the 5 most important NFRs for the AFS system that should be prioritized in the architecture where applicable.

1. Security – Due to the sensitive nature of the data handled in the system, ensuring this data is kept secure is vital to the system. Failing to do so could result in legal action and mistrust in the company.

2. Usability, Internationalisation, Accessiblity – there are many usability NFRs outlined due to the diversity of the system's users; it is imperative that they all feel comfortable using the system, or they won't use it again.
3. Data Integrity – Losing data could result in applications mid-processing being lost, meaning we would have to reimburse the applicant, and hurting AFS' reputation.
4. Scalability – Due to the seasonal nature of visa applications, we must be able to scale the system up and down depending on the amount of traffic expected at one time.
5. Availability – Customers are often not patient with information systems; if they are unable to access the system they may immediately switch to a competitor's service.

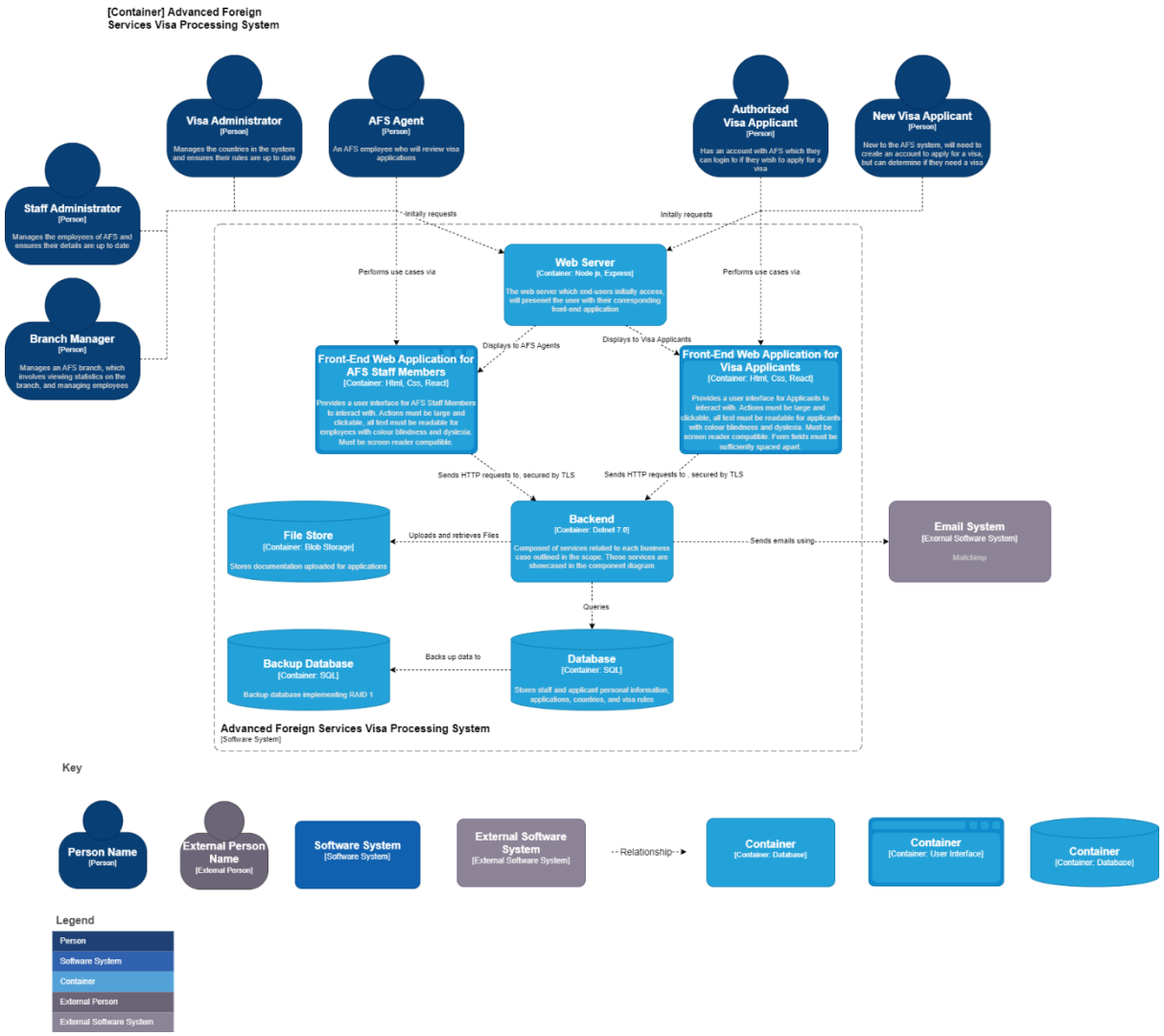
System Context

Link to System Context Diagram



Container Model

[Link to Container Model Diagram]



Container Diagram Notes

- A different front-end application is offered to applicants and AFS Agents, improving the security as users not working for AFS won't be able to attempt a staff login.
- The backend is composed using a service-oriented architecture, assisting with the scalability, and reliability.
- A backup database is implemented to ensure the data integrity of the system. This database will adhere to RAID 1 as this ensures if one drive fails, the data can still be recovered (Shooman & Shooman, 2012).

References

Chopra, R. K., Gupta, V., & Chauhan, D. S. (2016). Experimentation on accuracy of non functional requirement prioritization approaches for different complexity projects. *Perspectives in Science, Volume 8*, 79-82. https://doi.org/10.1016/j.pisc.2016.04.001

Gupta, V., Lohia, S., Çetinkaya, D., & Kim, H.-j. (2017). Non-functional Requirement Prioritization Approach. *International Journal of Software Engineering and Its Applications, 11 (1)*, 61-66. https://doi.org/10.14257/ijseia.2017.11.1.06

Shooman, A. M., & Shooman, M. L. (2012). A Comparison of RAID Storage Schemes: Reliability and Efficiency. *2012 Proceedings Annual Reliability and Maintainability Symposium*. IEEE. https://doi.org/10.1109/RAMS12136.2012