**Polygot Persistence**

**Contents**

[**1.** **Business Need** 4](#_Toc194449073)

[1.1 Business Problem 4](#_Toc194449074)

[1.1.1 Opportunity 4](#_Toc194449075)

[1.1.2 Context of the Problem 5](#_Toc194449076)

[1.2 Objectives 5](#_Toc194449077)

[1.2.1 Functional Objectives 5](#_Toc194449078)

[1.2.2 Non-Functional Objectives 6](#_Toc194449079)

[**2.** **Justification of Polyglot Persistence and NoSQL** 6](#_Toc194449080)

[2.1 Hybrid Approach 6](#_Toc194449081)

[2.1.1 Advantages of Using SQL for Structured Data 7](#_Toc194449082)

[2.1.2 Advantages of Using NoSQL for Unstructured Data 7](#_Toc194449083)

[**3.** **Data Management Plan** 8](#_Toc194449084)

[3.1 Types of Data Managed 8](#_Toc194449085)

[3.1.1 Structured Data 8](#_Toc194449086)

[3.1.2 Unstructured Data 10](#_Toc194449087)

[3.2 Data Integrity, Security and Compliance 11](#_Toc194449088)

[**4.** **Cost Plan** 11](#_Toc194449089)

[4.1 Setup Costs 11](#_Toc194449090)

[4.2 Operational Costs 12](#_Toc194449091)

[4.3 Budget Estimation 12](#_Toc194449092)

[4.3.1 Initial Setup (Month 1) 12](#_Toc194449093)

[4.3.2 Ongoing Monthly Operational Costs (Subsequent Months) 12](#_Toc194449094)

[**5.** **System Design** 13](#_Toc194449095)

[5.1 High-Level Architecture Diagram 13](#_Toc194449096)

[5.2 Entity-Relationship Diagram (ERD) 14](#_Toc194449097)

[**6.** **Implementation** 15](#_Toc194449098)

[6.1 SQL 15](#_Toc194449099)

[6.2 NoSQL 18](#_Toc194449100)

[6.3 Integration of the Databases 20](#_Toc194449101)

[6.4 Final Output of MongoDB and SQL 22](#_Toc194449102)

[**References** 25](#_Toc194449103)

**Table of Figures**

[Figure 1 Architecture Diagram 13](#_Toc194442801)

[Figure 2 ERD 14](#_Toc194442802)

# **Business Need**

## Business Problem

Organ transplantation exists as a life-preserving medical procedure that needs precise data management for optimal donor-recipient matching and organ preservation while upholding healthcare compliance (Vladan Borović et al., 2018). A robust data management system for transplant lists is essential because worldwide organ transplant needs demand systems that properly assist patients and doctors alongside organ procurement organizations (OPOs) (World Health Organization, 2023).

Traditional centralized databases face major problems:

* A fragmented data architecture.
* Delayed query processes.
* Inefficient organ donor matching (Rozony et al., 2024).

More than 100,000 individuals in the United States require organ transplants which underscores the necessity for a well-designed transplant system according to Rosenthal et al. (2025). According to Nimmo et al. (2022), the decline in transplant activity reached 16% because of logistical inefficiencies which negatively affected patient survival rates in 2020.

Organ transplant processes can benefit from structure of a polygot persistence database management system, which connects SQL with structured data and NoSQL with unstructured data simultaneously. The combination of SQL and NoSQL databases allows for scalable and secure data processing that suits the specific needs of organ transplant networks.

### Opportunity

A Polyglot Persistence application provides the following opportunities to address important system problems:

**Centralized Data Management:** The use of SQL and NoSQL databases allows for efficient data retrieval and immediate data updates (Tabatabaei Hosseini et al., 2024).

**Real-Time Organ Matching:** Automated systems, which match medical candidates using precise medical norms, enhance system operational efficiency.

**Scalability and Performance:** According to the National Academies of Sciences et al. (2022),

a robust system fulfils its mission for rising transplant requests while maintaining performance excellence.

**Regulatory Compliance:** Healthcare regulations can be fulfilled by implementing secure encryption techniques alongside strict data accessibility controls (Tsapepas et al., 2023).

### Context of the Problem

The organ transplant industry performs an essential function in modern healthcare while encountering important obstacles including conventional data handling processes, slow communication methods and non-compliance situations (Tabatabaei Hosseini et al., 2024). Specifically, developing a well-built database framework will lead to better transplant team coordination together with patient medical care results.

The following are the stakeholders of our project:

**Patients:** Rely on the system for timely organ matching and reduced waiting times (Cho et al., 2018).

**Doctors:** Require real-time access to patient records, medical histories and transplant status.

**OPOs:** Responsible for identifying donors, managing logistics and ensuring proper allocation.

**Regulatory Body:** Responsible for audit control, supervising data security protocols, managing access control and monitoring transplant records.

## Objectives

The system works to accomplish different functional and non-functional objectives to enhance transplant operations:

### Functional Objectives

**Comprehensive Patient and Donor Data Management:** Patient records will be maintained through the system by storing exact patient histories coupled with eligibility status and waiting list data (Ali et al., 2024). Enhanced matching techniques lead to a decrease in graft failure alongside an increase in survival rates.

**Real-Time Organ Matching and Availability Updates:** Through real-time tracking systems, OPOs gain the ability to efficiently match donations with receiving patients. NoSQL-based real-time data processing integration enhances resource distribution performance (Sousa, 2023).

**Regulatory Compliance and Audit Tracking:** The system upholds regulations from GDPR and national health regulations to provide complete data privacy. The audit log system will sustain regulatory compliance through tracing activities (Tsapepas et al., 2023).

**Secure Role-Based Access Control (RBAC):** User access is limited by roles so authorised figures encompassing patients, doctors and OPO staff members can view restricted data (Sangeetha et al., 2024).

### Non-Functional Objectives

**Scalability and Performance Optimization:** The system needs to efficiently manage extensive data sets and processing with rapid response capabilities. The implementation of polyglot persistence results in better scalability performance than conventional monolithic system architectures (Khine & Wang, 2019).

**Fault Tolerance and High Availability:** The system must protect continuous operation with automatic backup capabilities together with failover protections and duplicate storage replication. Existing research by Olawade et al. (2024) reveals how transportation inefficiencies and logistics problems lead to missed transplant opportunities.

**User-Friendly Interface and Multi-Device Accessibility:** Users will benefit from a well-designed interface system, which supports desktop computers as well as mobile devices. Madanian et al. (2023) show that satisfactory system design leads to enhanced patient involvement and broader system utilization.

# **Justification of Polyglot Persistence and NoSQL**

## Hybrid Approach

The database system must adopt a hybrid database model which combines SQL and NoSQL because this approach enables the system to handle multiple types of data while guaranteeing performance along with scalability and reliability. SQL databases provide patient record storage through their structured data system, which combines strict consistency rules. Quick and flexible processing of organ availability updates requires NoSQL databases because they work best with unstructured and semi-structured data types.

The requirement for accurate and consistent data exists for healthcare applications because healthcare regulations and patient safety standards apply (McGraw & Mandl, 2021). Healthcare systems generate massive volumes of high-speed data, but traditional RDBMS prove inefficient for handling such amounts of data (Rehman et al., 2021). The transplant database implements polyglot persistence which ensures structural data compliance and utilizes NoSQL for real-time database updates as well as handling extensive datasets.

### Advantages of Using SQL for Structured Data

The structured and relational model of data storage exists within SQL databases. In the transplant system:

* The pre-defined relationships in SQL databases enable accurate maintenance of patient records together with medical histories and transplant status data (Przytarski et al., 2021).
* Healthcare reliability depends on SQL databases to function properly as the foundation of secure processes.
* The advanced querying abilities of SQL enable medical staff to generate reports along with carrying out compliance audit procedures and making decisions about transplant procedures management.

### Advantages of Using NoSQL for Unstructured Data

NoSQL databases emerged as critical structures for data storage because of their capability to handle big volumes of semi-structured data with real-time processing and scalability needs (Azeroual et al., 2024). In the transplant system:

* NoSQL databases operate by delivering immediate updates including organ availability updates to healthcare providers.
* NoSQL databases dynamically store a wide range of data types, which helps accommodate dynamic alterations in transplant tracking and regulatory logging needs.
* NoSQL offers an efficient method for handling transplant-related information through its distributed design.

# **Data Management Plan**

A robust data management plan is critical for ensuring efficient data storage, retrieval, integrity and security in a database management system. This system integrates both SQL and NoSQL in a polyglot persistence approach to handle structured and unstructured data effectively.

## Types of Data Managed

### Structured Data

Structured data follows a fixed schema and is critical for maintaining data consistency and integrity. The following structured datasets will be managed using Microsoft SQL:

**Table 1** Patient\_ID will act as our primary Key. This table tracks patients in need of organ transplantation and makes sure that every patient has a Unique ID that includes vital medical information.

|  |
| --- |
| **Patients Table** |
| Patient\_ID |
| Patient\_Name |
| Patient\_Age |
| Patient\_Sex |
| Patient\_PhoneNumber |
| Patient\_Email |
| Patient\_BloodGroup |
| Patient\_OrganName |

**Table 2** Doctor\_ ID will be our primary key. This table has information on the doctors, their hospital and their area of expertise. This connects physicians with transplant recipients and patients.

|  |
| --- |
| **Doctors Table** |
| Doctor\_ID |
| Doctor\_Name |
| Doctor\_Age |
| Doctor\_Sex |
| Doctor\_Specialization |
| Doctor\_HospitalName |

**Table 3** OPO\_ID is primary key. This table manages the organizations responsible for organ collection and distribution. It monitors registered donors and available organs.

|  |
| --- |
| **OPO’s Table** |
| OPO\_ID |
| OPO\_Name |
| OPO\_Location |
| OPO\_ContactInfo |
| OPO\_OrgansAvailable |
| OPO\_DonorsRegistered |

**Table 4** Organ\_ID is the primary key. OPO ID and Donor ID are the foreign keys. This table will keep track of every organ that is available for transplant, along with its status, expiration date and the OPO in charge of it.

|  |
| --- |
| **Organs Table** |
| Organ\_ID |
| Organ\_Name |
| OPO\_ID |
| Organ\_Lifespan |
| Donor\_ID |

**Table 5** Donor\_ID is primary key. This table will maintain a record of donors, including their information and their reason of death. It will also help in confirming whether organs are compatible with patients.

|  |
| --- |
| **Donors Table** |
| Donor\_ID |
| Donor\_Name |
| Donor\_Age |
| Donor\_Sex |
| Donor\_BloodGroup |
| Donor\_DonorStatus |
| Donor\_CauseOfDeath |
| Donor\_OrganDonated |

**Table 6** This is going to be our linking table, it will connect patients, doctors and organs. Transplant\_ID is primary key. Patient\_ID, Doctor\_ID and Organ\_ID are foreign keys.

|  |
| --- |
| **Transplant Table** |
| Transplant\_ID |
| Patient\_ID |
| Doctor\_ID |
| Organ\_ID |

Table 7 This table acts as a central authority table that oversees access and audits. Regu\_MemberID is the primary key.

|  |
| --- |
| **Regulatory Body Table** |
| Reg\_MemberID |
| Reg\_MemberName |
| Reg\_MemberContactEmail |

### Unstructured Data

Unstructured data is highly dynamic, real-time and schemaless. The system uses MongoDB to manage:

Table 8 This table records the real-time location, estimated arrival time and status of organs during transit.

|  |
| --- |
| **Organ Transport Tracking** |
| Current Location |
| Estimated Arrival Time |
| Transit Status |
| Organ Use Status |

**Table 9** This table stores doctors' notes, updates and the entry timestamp.

|  |
| --- |
| **Doctor-Patient Communication** |
| Update Notes |
| Timestamp |
| Status |

**Table 10** This table logs user actions with a unique LogID, user identifier and type of action.

|  |
| --- |
| **Audit Logs** |
| LogID |
| UserID |
| Action |
| Timestamp |

## Data Integrity, Security and Compliance

**Data Integrity and Validation:** Incorrect data entry will be avoided by enforcing data entry rules at the database level.To ensure that relationships between tables remain consistent, foreign key constraints will be used in SQL.Redundant transplant records will be prevented using data deduplication techniques.

**Security Measures:** Only authorised personnel can access specific datasets due to the Role-based access control (RBAC).Patient and transplant data can be accessed by doctors.OPOs cannot access full patient medical history but can manage organ logistics.

**Regulatory Compliance:** Adhering to the GDPR, the database management system will protect patient data rights and mandate data encryption.

# **Cost Plan**

## Setup Costs

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Estimated Cost** | **Source** |
| **Database Hosting** | A dedicated server for database management. | Starting at £2.79 per month | <https://hostinguk.net/web-hosting> |
| **Cloud Storage** | Secure object storage for data backups and archival. | £0.023 per GB per month | <https://cloud.google.com/storage/pricing> |
| **Development Costs** | Expenses related to software development. | Depends on scope and developer rates. | N/A |

## Operational Costs

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Description** | **Estimated Cost** | **Source** |
| **Maintenance** | Regular system updates and backups. | Included in hosting fees or additional based on service agreements. | N/A |
| **Scaling** | Adjusting resources to accommodate increased demand. | Additional storage at £0.023 per GB per month. | <https://cloud.google.com/storage/pricing> |
| **Security Monitoring** | Implementing measures to protect data integrity. | Involve additional services or in-house resources. | N/A |

## Budget Estimation

To provide a comprehensive budget estimation, consider the following scenario:

### Initial Setup (Month 1)

* **Database Hosting:** £2.79​
* **Cloud Storage:** Assuming 100 GB usage: 100 GB x £0.023/GB = £2.30​
* **Development Costs:** Assuming 100 hours at £50/hour = £5,000​
* **Total Initial Setup Cost:** £2.79 + £2.30 + £5,000 = **£5,005.09**​

### Ongoing Monthly Operational Costs (Subsequent Months)

* **Database Hosting:** £2.79​
* **Cloud Storage:** 100 GB x £0.023/GB = £2.30​
* **Maintenance:** Included in hosting or additional based on agreements.​
* **Scaling:** Additional storage as needed. Assuming that there is a 50 GB increase: 50 GB x £0.028/GB = £1.4
* **Security Monitoring:** Assuming that there is a £100 for third-party services.​
* **Total Monthly Operational Cost:** £2.79 + £2.30 + £1.4 + £100 = **£106.49**

# **System Design**

## High-Level Architecture Diagram

The overall structure of the transplant database management system is represented by a high-level architecture diagram. It outlines how different components interact:

* **Frontend Interface:** Web-based or desktop application for users.
* **Security Layer:** Implements authentication, authorisation and data encryption.
* **Backend API:** Handles business logic, processes request and communicates with databases.
* **SQL Database:** Stores structured data.
* **NoSQL Database:** Manages unstructured data.

A diagram of a software application

AI-generated content may be incorrect.

Figure 1 Architecture Diagram

## Entity-Relationship Diagram (ERD)

The ERD visually represents the relationships between entities in the system.

A diagram of a network

AI-generated content may be incorrect.

Figure 2 ERD

Key entities include:

**Patient:** A patient undergoes transplants, linking them to the required organ. They also receive medical updates.

**Transplant**: Each transplant involves organs, ensuring proper tracking.

**Doctor**: A doctor performs transplants, linking to each transplant. They also send updates to patients.

**Organ:** An organ is matched with recipients.

**Donor:** A donor provides organs that are matched with patients.

**OPO:** OPOs manage the collection and distribution of organs.

**Audit Log:** Each transplant is logged for compliance and tracking.

**Organ Transport:** Tracks the organ on its way to the patient for the transplant.

**Regulatory Body:** The body interacts with the audit logs to perform audits on the transplant.

**Doctor Patient Communication:** The doctor sends updates to the patient.

# **Implementation**

The Polyglot Persistence integrates SQL and NoSQL databases ensuring data consistency and scalability. Transactional integrity and regulatory compliance is ensured by SQL meanwhile, NoSQL handles unstructured data offering flexibility and rapid data retrieval.

## SQL

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer screen

AI-generated content may be incorrect.

A computer screen shot of text

AI-generated content may be incorrect.

A computer screen shot of a computer screen

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

## NoSQL

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Integration of the Databases

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen shot of a computer code

AI-generated content may be incorrect.

A computer screen shot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

## Final Output of MongoDB and SQL

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# **References**

Ali, H., Shroff, A., Soliman, K., Molnar, M. Z., Sharif, A., Burke, B., Shroff, S., Briggs, D., & Krishnan, N. (2024). Improved survival prediction for kidney transplant outcomes using artificial intelligence-based models: development of the UK Deceased Donor Kidney Transplant Outcome Prediction (UK-DTOP) Tool. *Renal Failure*, *46*(2). https://doi.org/10.1080/0886022x.2024.2373273

Azeroual, A., Benayad Nsiri, Rachid, & Taoufiq Belhoussine Drissi. (2024). Big Data in the Revolution of Medical Data: A Review. *Statistics Optimization & Information Computing*, *13*(1), 378–395. https://doi.org/10.19139/soic-2310-5070-2054

Cho, S., Mohan, S., Husain, S. A., & Natarajan, K. (2018). Expanding transplant outcomes research opportunities through the use of a common data model. *American Journal of Transplantation*, *18*(6), 1321–1327. https://doi.org/10.1111/ajt.14892

Khine, P. P., & Wang, Z. (2019). A Review of Polyglot Persistence in the Big Data World. *Information*, *10*(4), 141. https://doi.org/10.3390/info10040141

Madanian, S., Nakarada-Kordic, I., Reay, S., & Chetty, T. (2023). Patients’ perspectives on digital health tools. *PEC Innovation*, *2*(2), 100171–100171. https://doi.org/10.1016/j.pecinn.2023.100171

McGraw, D., & Mandl, K. D. (2021). Privacy protections to encourage use of health-relevant digital data in a learning health system. *NPJ Digital Medicine*, *4*(1), 1–11. https://doi.org/10.1038/s41746-020-00362-8

National Academies of Sciences, E., Division, H. and M., Services, B. on H. C., Policy, B. on H. S., Committee on A Fairer and More Equitable, C.-E., Hackmann, M., English, R. A., & Kizer, K. W. (2022). The U.S. Organ Transplantation System and Opportunities for Improvement. In *www.ncbi.nlm.nih.gov*. National Academies Press (US). https://www.ncbi.nlm.nih.gov/books/NBK580025/

Nimmo, A., Gardiner, D., Ushiro-Lumb, I., Ravanan, R., & Forsythe, J. L. R. (2022). The Global Impact of COVID-19 on Solid Organ Transplantation: Two Years Into a Pandemic. *Transplantation*, *106*(7), 1312–1329. https://doi.org/10.1097/tp.0000000000004151

Olawade, D. B., Marinze, S., Qureshi, N., Weerasinghe, K., & Teke, J. (2024). Transforming organ donation and transplantation: Strategies for increasing donor participation and system efficiency. *European Journal of Internal Medicine*. https://doi.org/10.1016/j.ejim.2024.11.010

Przytarski, D., Stach, C., Gritti, C., & Mitschang, B. (2021). Query Processing in Blockchain Systems: Current State and Future Challenges. *Future Internet*, *14*(1), 1. https://doi.org/10.3390/fi14010001

Rehman, A., Naz, S., & Razzak, I. (2021). Leveraging big data analytics in healthcare enhancement: trends, challenges and opportunities. *Multimedia Systems*, *28*(4), 1339–1371. https://doi.org/10.1007/s00530-020-00736-8

Rosenthal, B. M., Hansen, M., & White, J. (2025, February 26). Organ Transplant System “in Chaos” as Waiting Lists Are Ignored. *The New York Times*. https://www.nytimes.com/interactive/2025/02/26/us/organ-transplants-waiting-list-skipped-patients.html

Rozony, F. Z., Aktar, M. N. A., Ashrafuzzaman, M., & Islam, A. (2024). A SYSTEMATIC REVIEW OF BIG DATA INTEGRATION CHALLENGES AND SOLUTIONS FOR HETEROGENEOUS DATA SOURCES. *ACADEMIC JOURNAL on BUSINESS ADMINISTRATION, INNOVATION & SUSTAINABILITY*, *4*(4), 1–18. https://doi.org/10.69593/ajbais.v4i04.111

Sangeetha, S. K. B., Selvarathi, C., Mathivanan, S. K., Cho, J., & Easwaramoorthy, S. V. (2024). Secure Healthcare Access Control System (SHACS) for Anomaly Detection and Enhanced Security in Cloud-Based Healthcare Applications. *IEEE Access*, *12*, 164543–164559. https://doi.org/10.1109/access.2024.3492024

Sousa, R. (2023). Big Data and real-time knowledge discovery in healthcare institutions. *Uminho.pt*. https://hdl.handle.net/1822/89214

Tabatabaei Hosseini, S. A., Kazemzadeh, R., Foster, B. J., Arpali, E., & Süsal, C. (2024). New Tools for Data Harmonization and Their Potential Applications in Organ Transplantation. *Transplantation*, 10.1097/TP.0000000000005048. https://doi.org/10.1097/TP.0000000000005048

Tsapepas, D. S., King, K., Husain, S. A., Yu, M. E., Hippen, B. E., Schold, J. D., & Mohan, S. (2023). UNOS Decisions Impact Data Integrity of the OPTN Data Registry. *Transplantation*. https://doi.org/10.1097/tp.0000000000004792

Vladan Borović, Saša Borović, Drąsutė, V., & Dejan Rančić. (2018). SECURE ORGAN TRANSPLANT INFORMATION SYSTEM. *Facta Universitatis Series Automatic Control and Robotics*, *17*(1), 1–1. https://doi.org/10.22190/fuacr1801001b

World Health Organization. (2023). *Transplantation*. Www.who.int. https://www.who.int/health-topics/transplantation#tab=tab\_1