Jaden Patrick

Dr. Damon Gray

Management Information Systems

30th November 2025

Governance Memo

The Help Desk Information System I help built for this project uses synthetic ticket data, but I designed it the same way a real company would when dealing with sensitive information. Even though we aren't storing actual customer details, the system still follows strong privacy, security, and governance practices so it could function in a real environment without putting the organization at risk. This memo breaks down the policies, safeguards, and technical steps I used to protect the data, manage user access, prevent system failures, and prepare the platform to scale into PostgreSQL later on.

A big part of keeping a system trustworthy is protecting user privacy. Even though my dataset is fully fake, I structured the database as if it were holding live information. To keep things safe, the system doesn't collect any personal data—no emails, phone numbers, addresses, or financial info. It only stores what's necessary for ticket tracking: priority, status, timestamps, and randomly generated names. If this system ever handled real submissions, it would follow strict privacy rules: collect only what's needed, mask sensitive details in reports, encrypt data in transit and at rest, and rotate credentials regularly. These steps keep privacy risks low and help build user confidence in the system.

Another major part of governance is controlling who can access what. That's where role-based access control (RBAC) comes in. In a real help desk, different users need different levels of access. A regular user should only be able to submit tickets and view their own cases. Support agents should be able to update tickets assigned to them, but not delete records or change system settings. Supervisors need a wider view so they can monitor performance and reassign work. System administrators get full access because they maintain the database, backups, and technical setup. This layered RBAC structure prevents unauthorized actions, reduces mistakes, and makes everything more accountable.

Protecting data also means having a solid backup plan. Even small systems can lose data if the database corrupts or hardware fails. For this project's SQLite database, a realistic backup strategy would include an automatic daily copy of the tickets.db file, a full weekly snapshot stored separately, and keeping backups for 30–60 days. Each backup should be tested to make sure it actually restores correctly. If something ever goes wrong, recovery is simple: stop the app, replace the damaged database with the latest backup, restart it, and confirm everything loads. In a real company, backups would be stored securely in the cloud and encrypted. As the system grows, change control becomes more important. Any updates to the schema or the Streamlit app should be documented and tested in a safe environment before being deployed. Updating during low-traffic hours helps avoid downtime, and using Git makes it easy to track changes and roll back if needed. This prevents broken dashboards, mismatched tables, or crashes that could impact users.

Logging and auditing also play a big role in security and troubleshooting. The system should track when tickets are created, updated, closed, or reopened—and record which agent did it. It should also log failed login attempts or system errors. Keeping these logs separate from the main database prevents tampering and helps resolve disputes or spot misuse. Eventually, this help desk system may need to scale past what SQLite can handle. That's why I included a clear migration path to PostgreSQL. The process is simple: export the SQLite tables as CSVs, rebuild the same tables in PostgreSQL, import the data with bulk-loading tools, update the Streamlit app to use a Postgres driver like psycopg2 or SQLAlchemy, and thoroughly test the dashboards. PostgreSQL offers better indexing, concurrency, and security, which makes it the right long-term upgrade for heavier traffic or more advanced reporting.

Overall, the governance framework behind this system makes it private, secure, reliable, and ready to grow. With strong privacy rules, clean access control, dependable backups, organized change management, proper logging, and a solid plan for scaling, the Help Desk Information System matches real industry standards and is built to support both current use and future expansion.