A colorful letters on a black background

AI-generated content may be incorrect.

Lakeridge Community

Health Centre

RFP Response

February 5, 2025

Our Mission

Promoting, protecting and improving the lifelong health of everyone

in the greater Oshawa area through care, reliance, precision and health equity.

Who We Are

The BLOR Company has been turning data management into effective, secure, reliable, and optimized solutions for leading organizations in various industries.

We do a lot more than build and deploy a database in the strict sense: we are the architects who lead you to data-driven success. The basis of our competitive advantage is a commitment to excellence, innovative solutions, and collaborative relationships with clients. Our team is composed of highly professional database experts with extensive experience.

One common driving force unites them: to create solutions that truly matter in patient care.

Our portfolio contains numerous successful projects ranging from small clinics to large networks of hospitals, which serve as a testament to our capability in translating complicated requirements into sophisticated, efficient database systems.

BLOR renders the intricacies of data understandable, informative, and thus adoptable for creative decisions. We manage to do so by focusing on security, ease of user interaction, and transparency while providing results within the pre-accorded timeline.

Where You Are Now

At LCHR, the existing information system is not meeting your needs. But before we take anything apart, we want to understand your needs and what has been working for you, so that we can build upon that success.

Below is a mapping of your current flow of data across your organization, today:

A diagram of a diagram

AI-generated content may be incorrect.

Data only flows one way at LCHR. We can change that.

It starts by analyzing the context in which your data interacts. If we imagine the LCHR as a data-hub, here’s what that looks like:

A diagram of a system

AI-generated content may be incorrect.

# Functional Requirements

* The System must send Notifications to physicians
* The System must check the room availability.
* The System must update the room availability.
* The system should check, confirm, and update room availability based on the patient throughput (the patient being assigned and discharging from the room)
* The System must assign room to patients.
* The System must generate and send room utilization report to hospital administrator
* The System must request specific lab tests.
* The System must store patient data with their assigned room.
* The System should collect patient information
* The System must store gathered information in the database
* The System must validate the patient ID
* The System must flag the potential duplicates
* The System must confirm verification and store it in the database
* The System must determine room availability
* The System must Assign Available Room
* The System must Record Room Assignment
* The System must Create New Patient Record
* The System must Update Existing Patient Record
* The System must Communicate Room Assignment
* The System must verify the patient payment information
* The System must receive the payment.
* The System must process the payment information.
* The System must generate a payment confirmation receipt and send an email with payment confirmation details to the patient.
* The System must calculate the medical cost.
* The System must validate the medical cost data.
* The System must generate the bill.
* The System must send payment information to accounting system.
* The System must send receipt information to the accounting system.
* The System must collect all transactions related to patients from central database.
* The System must validate the transaction data.
* The System must create and manage invoices.
* The System must generate a revenue report based on transaction information.
* The System must send revenue report to hospital administrator
* The System must receive the patient’s condition summary.
* The System must request the lab test information.
* The System must conduct the lab test.
* The System must send the lab test reports to the room.
* The System must give prescribed medication to a patient.
* The System must check and update the medicine availability/inventory.
* The System must send reports to the room utilization system.
* The System must accept prescriptions from physician.
* The System must send a medicine report to the room.
* The System must review the treatment.
* The System must generate the medical report.
* The System must send the medical report to the physician system.
* The System must gather the physician information.
* The System must validate the physician information.
* The System should check the physician’s availability.
* The System should update the physician’s status.
* The System must assign physician to a patient.
* The System must receive the physician’s data.
* The System must request the patient’s condition.
* The System must send the patient’s condition information to the pharmacy system.
* The System must Store Patient Data to database
* The System must allow retrieval of data from the database
* The System must keep track of patient data (patients' admission, readmission, discharge, etc.)
* The System will keep track of the doctor’s availability
* The System should data backup and recovery: the system should automatically backup data to a secure location on a predetermined schedule.
* Support the updating and maintenance of patient prescriptions.
* Support the scheduling and tracking of physician appointments.
* Provide an accessible interface and data store for laboratory results.
* Real-time updating and display of patient information.
* Real-time updating and display of physician details and patients.
* Real-time updating and display of room utilization.
* Real-time financial tracking with breakdowns by cost centre.
* Automatically generate daily & weekly reports, as required (Referring Physician Reports, Revenue Reports, Room Utilization Reports, and others)
* Automatically generate patient billing information.

# Non-Functional Requirements

Performance:

* Real-Time Responsiveness - provide information in real-time for critical operations like patient registration and inquiries.
* Query Performance: queries should execute efficiently and return results quickly.
* Transaction Throughput: able to handle a large volume of transactions concurrently, especially during peak hours in a hospital.

Scalability:

* Scalability to 200 Beds: accommodate the planned expansion to 200 beds and the associated increase in patients, physicians, and data volume.
* Scalability for Future Growth: scalable to handle future growth beyond the initial 200 beds as the city and region grow.
* The system architecture should support easy scaling, allowing for future expansion to accommodate additional facilities or departments.
* Horizontal Scalability: Adding more servers (risk of integration problems).
* Vertical Scalability: Upgrading existing servers (risk of losing data).

Availability and Reliability:

* High Availability: ensure continuous operation and minimal downtime.
* Reliability: reliable and stable, minimizing errors and data loss.
* Data Backup and Recovery: protect against data loss due to system failures or disasters.
* Backups will be automated weekly and stored securely with a rollback option for up to three months in case of system errors.
* The system will be backed up every four months.
* The system should remain operational 24/7 to support continuous hospital operations.
* The system interface should comply with accessibility standards (e.g., WCAG 2.1) to support users with disabilities.
* The system must be compatible with future Windows OS updates and maintain support for legacy Windows versions as long as they are commonly used in hospitals.
* ability to run on different platforms or environments

Security:

* Data Security: protect sensitive patient data from unauthorized access, modification, or disclosure. Implement appropriate security measures (access control, encryption, audit trails).
* Role-Based Access Control: ensure that users only have access to the data and functionalities necessary for their roles (physicians, nurses, administrators, etc.).

Usability:

* User-Friendly Interface: intuitive interface for all user groups (physicians, nurses, administrative staff) with varying levels of technical skills.
* Easy to Learn and Use: minimize training time and maximize user adoption.
* Efficient Workflow Support: streamline hospital workflows and processes.
* Provide comprehensive training materials and user manuals for staff to ensure effective system usage

Maintainability:

* Easy to Maintain: designed to be easily maintained and updated.
* Modularity: facilitate easier maintenance and future enhancements.
* Well-Documented: database schema, code, user manuals to aid in maintenance and support.

Data Integrity and Consistency:

* Data Accuracy: minimize data entry errors through validation rules and data constraints.

# 3NF Relational Schema

|  |  |
| --- | --- |
| PERSONS | (**PERSON\_ID, FNAME, LNAME, ADDRESS\_ID 🡪 ADDRESSES**, **PHONE**, EMAIL) |
| ADDRESSES | (**ADDRESS\_ID, STREET\_NUM, STREET\_NAME,** LINETWO, **CITY, POSTAL\_CODE 🡪 POSTAL\_CODES**) |
| PROVINCES | (**PROVINCE\_CODE, PROVINCE\_NAME**) |
| COUNTRIES | (**COUNTRY\_CODE, COUNTRY\_NAME**) |
| POSTAL\_CODES | (**POSTAL\_CODE, PROVINCE\_CODE 🡪 PROVINCES**) |
| COUNTRY\_PROVINCES | (**PROVINCE\_CODE 🡪 PROVINCES, COUNTRY\_CODE 🡪 COUNTRIES**) |
| ROOMS | (**ROOM\_NUM, ROOM\_TYPE**) |
| BEDS | (**BED\_ID**, **ROOM\_NUM, EXTENSION**) |
| PATIENTS | (**PATIENT\_ID, PERSON\_ID, HCN, SEX, ROOM\_NUM 🡪 ROOMS, BED, FINANCIAL\_STATUS**) |
| PHYSICIANS | (**PHYSICIAN\_ID, PERSON\_ID, STAFF\_ID, SPECIALTY**) |
| STAFF | (**STAFF\_ID, PERSON\_ID**) |
| VENDOR | (**VENDOR\_ID, ADDRESS\_ID 🡪 ADDRESSES, PHONE**, EMAIL) |
| VENDOR\_REPS | (**VENDOR\_ID, PERSON\_ID 🡪 PERSONS**) |
| STAYS | (**STAY\_ID**, **PATIENT\_ID 🡪 PATIENTS, BED, DATE\_ADMITTED,** DATE\_DISCHARGED**)** |
| PERSCRIPTIONS | (**PERSCRIPTION\_ID, PHYSICIAN\_ID 🡪 PHYSICIANS, PATIENT\_ID 🡪 PATIENTS, DRUG\_SKU, DOSE, QTY**) |
| TREATMENTS | (**TREAMENT\_ID, PATIENT\_ID 🡪 PATIENTS, PHYSICIAN\_ID 🡪 PHYSICIANS, DETAILS**) |
| TREATMENT\_PERSCRIPTIONS | (**TREATMENT\_ID, PERSCRIPTION\_ID**) |
| APPOINTMENTS | (**PHYSICIAN\_ID, PATIENT\_ID, DATE\_TIME**, NOTES) |
| APPLIED\_TREATMENTS | (**APPOINTMENT\_ID, TREAMENT\_ID**) |
| LABORATORY\_RESULTS | (**RESULTS\_ID, PATIENT\_ID 🡪 PATIENTS, PHYSICIAN\_ID 🡪 PHYSICIANS,** |
| COST\_CENTRES | (**COST\_CENTRE, NAME**) |
| BILLING\_CODE | (**BILLING\_CODE**, **DESCRIPTION, COST\_CENTRE, COST**) |
| INVOICES | (**INVOICE\_NUM**, **PATIENT\_ID 🡪 PATIENTS, INVOICE\_DATE, PAID)** |
| INVOICE\_ITEMS | (**INVOICE\_NUM, BILLING\_CODE**) |

# Entity-Relationship Diagram (ERD)

A computer diagram of a computer

AI-generated content may be incorrect.

# Next Deliverable

Text