# SOFTWARE ARCHITECTURE

## Architecture Style: Layered

This is the most commonly used architecture pattern—used in most enterprise-level applications. It’s also often referred to as the **n-tier** architecture pattern, assuming n number of tiers or layers. This is the de-facto pattern for all JAVA EE applications, too. The layered architecture pattern closely matches the traditional IT communication and organizational structures found in most companies, making it a natural choice for most business application development efforts.

## What does a Layered Architecture entail?

We identify distinct classes of services that can be arranged hierarchically in a series of horizontal layers. Each layer performs a specific role within the application. Although the layered architecture doesn’t specify the number and types of layers that must exist in the pattern, most small applications go with three layers, whereas larger and more complex business layers can contain up to four or even five layers. This application will consist of three layers—database, business, and presentation.

### Roles of each layer:

#### database layer:

Simple Databases expanding up to SANs (Storage Area Networks)

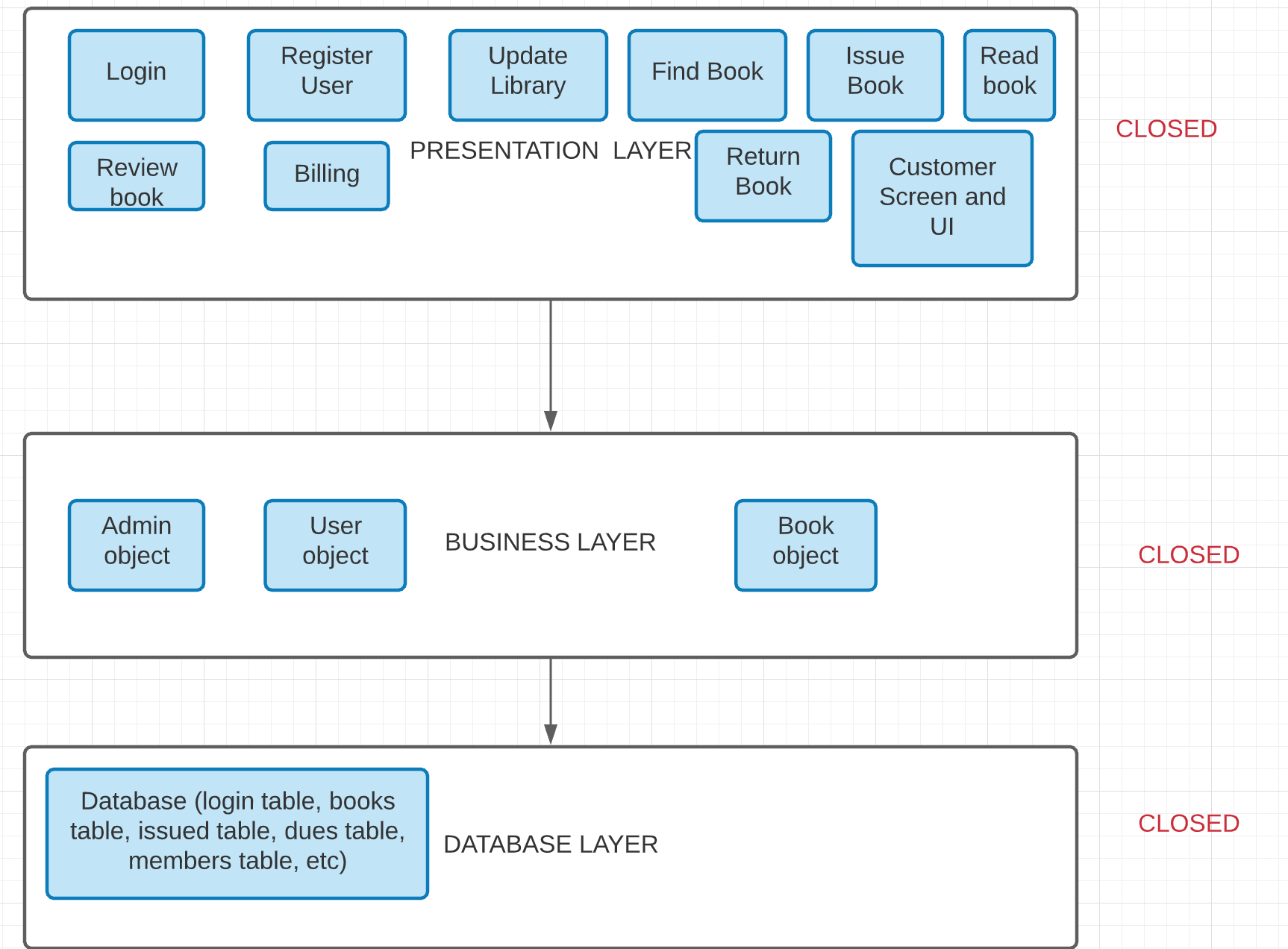
#### BUSINESS LAYER:

Business logic is the programming that manages communication between an end user interface and a database.

#### presentation layer:

Presentation of web pages, end user interacting APIs, and functions delivered.

# software architecture diagram



Each layer depends on (calls) the services from the layer below.