

# Enterprise Development Glossary

---

## Pass Task 3.1

**1. Business Logic Layer.** In 3-tier architecture, the business logic layer, also known as middleware layer, is responsible for modeling all the logics related to business requirements into one or many class libraries or business objects. Its purpose is to separate all logics related to business rules into one place to promote separate of concern in software design and improve adaptability, extensibility and maintenance quality of software. By putting all business logic into a separated layer, the software system will be able to adapt to a change in business rule easily. Developer can easily pin point the place where new business rule should implement. In addition, the change can be applied without affecting other parts of the system such as the presentation screen or other business rules. This layer uses the DAL layer to access data, then process the data according to business rules and requirements and then forward that result to the application layer. The application layer uses business logic layer to handle data processing requested by the users. By extracting all business rules on to a separated layer, it promotes the reusability of the system as well as support a clear separation of concern in code quality.

**2. Business Objects.** They are essential citizen lives in Business Logic Layer. Each of these objects represents a set of business rules and requirements that link or associate in a clear scope in the domain business model. Business object is effectively the most important part of a software system that map real world processes into software procedures. Business objects can interact with each other to provide a full functionality to upper level of the software system architecture. Business objects can be stateless or stateful. Stateless business object does not maintain state of request or session, whereas stateful business objects hold information about the session until a timeout expired or exclusively destroyed by the external requesting processes.