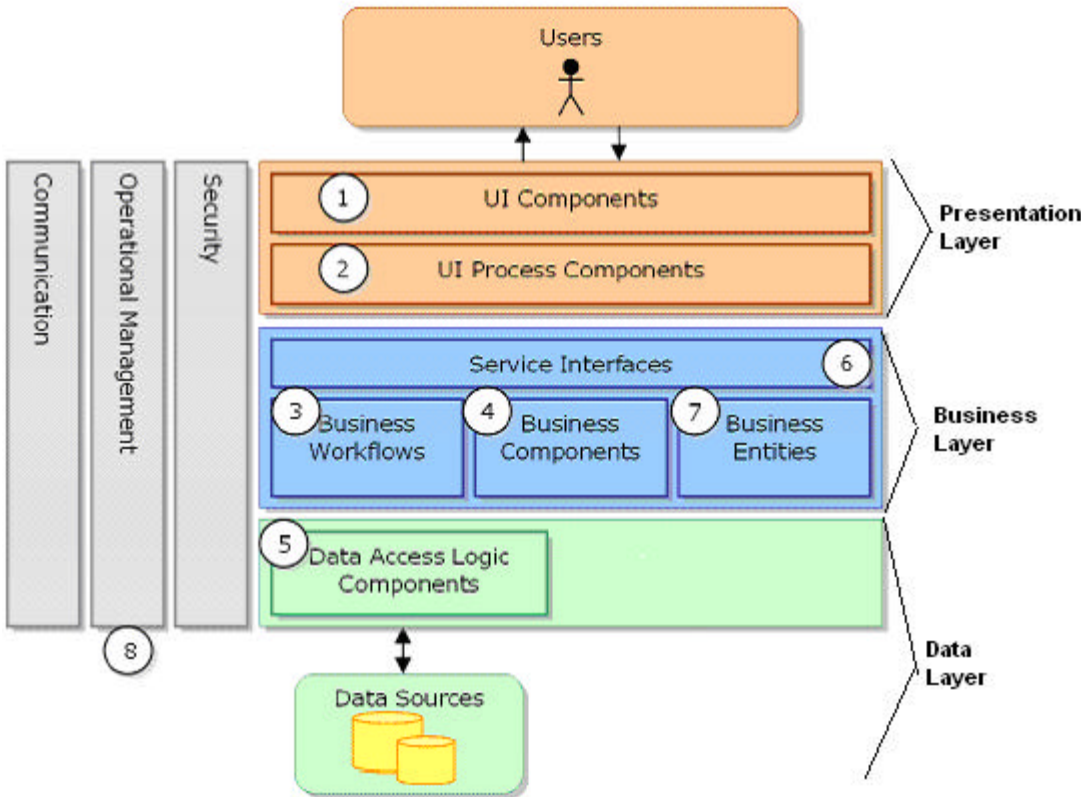


Design /Architecture: .NET Applications

The application design is based on a distributed architecture using the Microsoft® .NET Framework.

The POS application architecture is divided into Presentation layer, Business layer and Data layer. The application layers and components are graphically represented as below.



The component types identified in the POS application design are explained below:

1. User interface (UI) components

User interface components define a way for the users to interact with the application.

For example, the browser-based interface lets users to view products details and accept orders. User interfaces are implemented using Microsoft ASP.NET web forms and controls.

2. User process components

The application follows a predictable process on user interactions with the system.

For example, the procedure for viewing product data that has the user select a category from a list of available product categories and then select an individual product in the chosen category to view its details.

Similarly, when the user makes a purchase, the interaction follows a predictable process of gathering data from the user, in which the user first supplies details of the products to be purchased, then provides payment details, and then enters delivery details.

To help synchronize and orchestrate these user interactions, it can be useful to drive the process using separate user process components. This way the process flow and state management logic is not hard-coded in the user interface elements themselves, and the same basic user interaction “engine” can be reused by multiple user interfaces.

3. Business workflows

A business process may be performed after a user process collects the required data.

For example, after the product, payment, and delivery details are submitted to the application, the process of taking payment and arranging delivery can begin. Many business processes involve multiple steps that must be performed in the correct order and orchestrated.

For example, the system would need to calculate the total value of the order, validate the credit card details, process the credit card payment, and arrange delivery of the goods. This process could take an indeterminate amount of time to complete, so the required tasks and the data required to perform them would have to be managed.

Business workflows define and coordinate long running, multi-step business processes.

4. Business components

The application will require components that implement business rules and perform business tasks.

For example, in the application, it would need to implement the functionality that calculates the total price of the goods ordered and calculates the discount percentage and amount applicable.

Business components implement these business logics of the application.

5. Data access logic components

The application uses the data access logic components to access a data source during a business process.

For example, the application needs to retrieve product data from a database to display product details to the user, and it needs to insert order details into the database when a user enters an order.

The logic to access data is abstracted in a separate layer of data access logic components. Doing so centralizes data access functionality and makes it easier to configure and maintain.

6. Service interfaces

Service interfaces are created to expose business logic as a service that supports the communication exchanges its different processes require.

For example, the credit card authorization service must expose a service interface that describes the functionality offered by the service and the required interfaces for calling it.

7. Business entity components

Application requires data to be passed between components because of its component based distributed architecture.

For example, in the application a list of products must be passed from the data access logic components to the user interface components so that the product list can be displayed to the users.

The business entities that are used internally in the application are usually data structures, Extensible Markup Language (XML) streams or custom object-oriented classes.

8. Components for security, operational management, and communication

The application will use components to perform exception management, to authorize users to perform certain tasks, and to communicate with other services and applications.