ARCHITECTURE & DESIGN PATTERNS

Discussion

Presented By Truong Tran

SCS Solutions

CONTENT

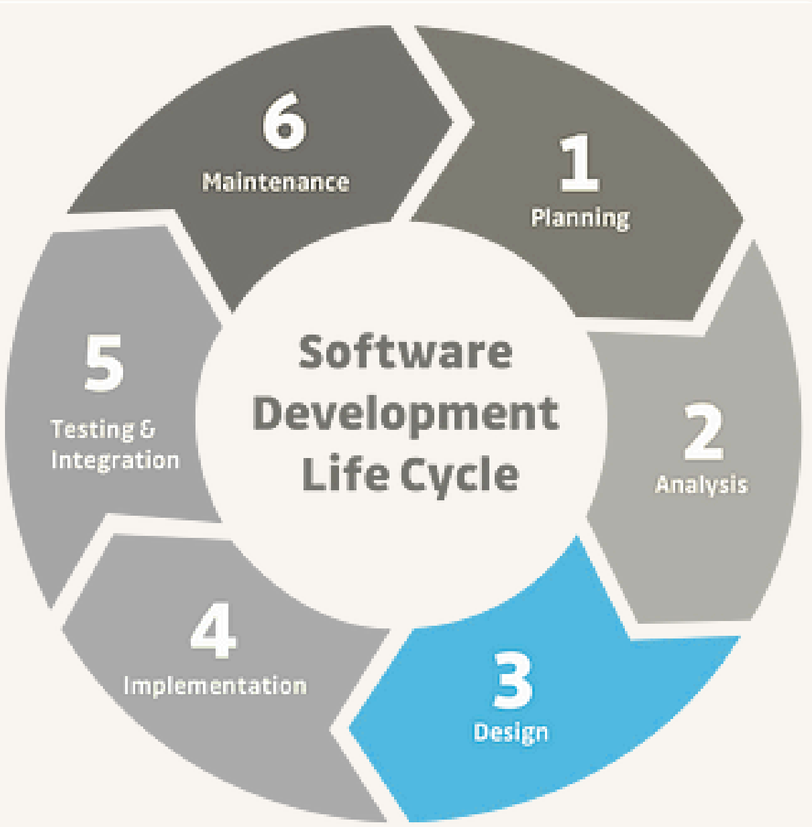
1. Overview
2. Architecture
3. Design Patterns
4. Demo
5. Discussion

OVERVIEW

Complexity: Large systems, connecting many components.

High requirements: Need to scale quickly, operate stably.

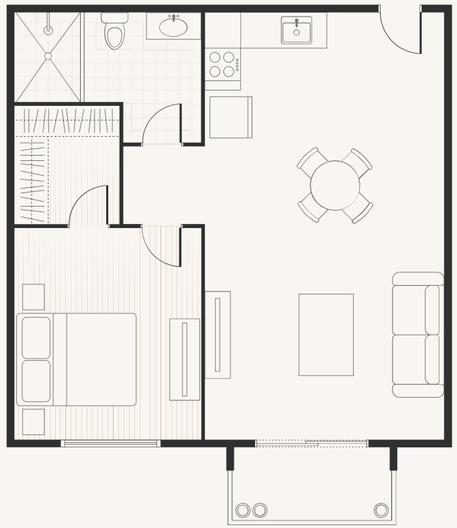
Challenge: How to develop quickly without creating a product that is difficult to maintain?



SYSTEM DESIGN

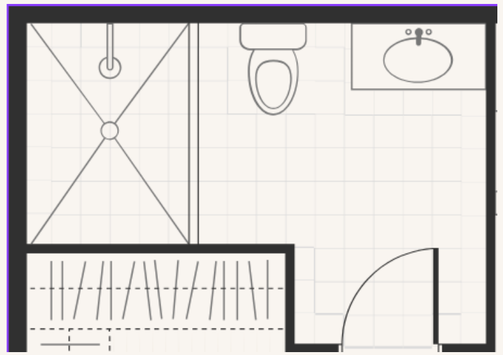
ARCHITECTURE (HLD)

Software architecture is the fundamental structure of a software system, encompassing its components, their relationships, and the principles guiding its design and evolution. It's essentially the blueprint that dictates how a software system will be built and how its various parts will interact to fulfill its intended purpose.



DESIGN PATTERNS (LLD)

Software design patterns are reusable, general solutions to commonly occurring problems in software design. They are not concrete implementations or code, but rather blueprints or templates that can be adapted to solve a specific problem in different contexts.



Resort Building

- A 5-storey main hotel building.
- 10 separate bungalows with private pools.
- A central restaurant.
- A children's play area.
- The system must be environmentally friendly, safe, and able to accommodate 500 guests at a time

ARCHITECTURES

Monolithic: Entire application in one deployment unit

Microservices: Split into independent services

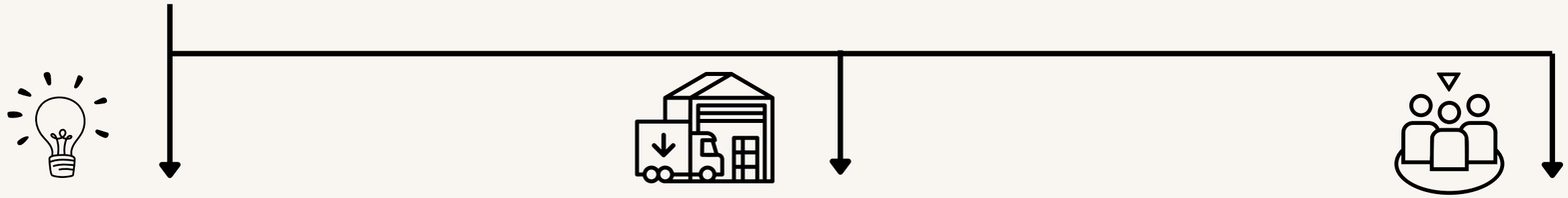
Component-based: Split into independent reusable components

Service-Oriented (SOA): Based on reusable services

Event-Driven: Based on creating, detecting, and reacting to events

Layered: Organized into layers (presentation, business, data)

DESIGN PATTERNS



Creational

- Factory Method
- Abstract Factory
- Builder
- Prototype
- Singleton

Structural

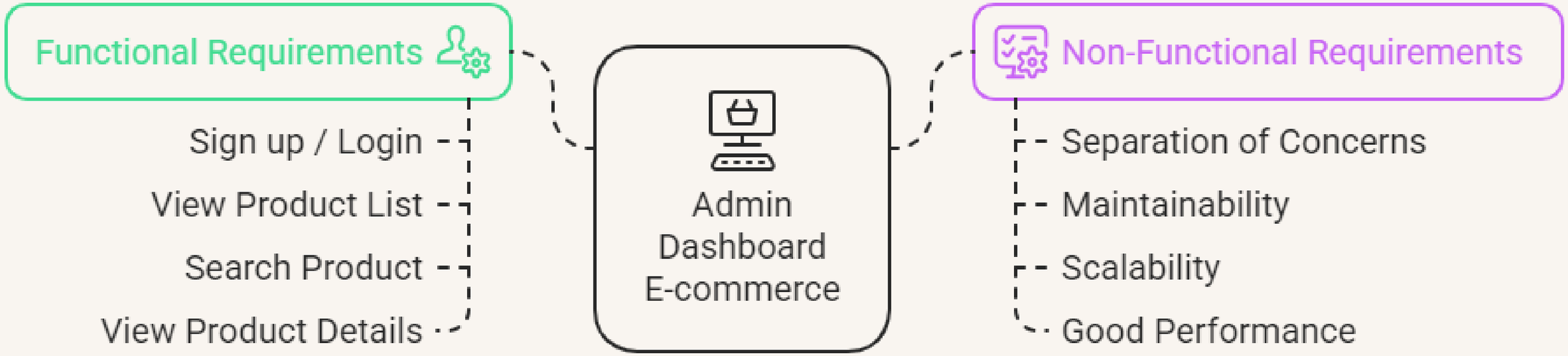
- Adapter
- Bridge
- Composite
- Decorator
- Facade
- Flyweight
- Proxy

Behavioral

- Chain of Responsibility
- Command
- Interpreter
- Iterator
- Mediator
- Memento
- Observer
- State
- Strategy
- Visitor
- Template Method

DEMO

Design an Simple Admin Dashboard E-commerce



DEMO

System Design

- Technology
  - BE: ASP.NET Core
  - FE: Angular
  - DB: SQL
  - Deploy: AWS, Azure
- Communication: FE and BE communicate with each other via HTTP protocol, using JSON format.

Architectures

- BE:
  - Clean or Onion
  - Microservices (Future)
- FE:
  - Component-based
  - Modular/Featured
  - MVC

Design Patterns

- BE:
  - Repository
  - Unit of Work
  - Mediator
- FE:
  - Singleton
  - Reactive Programming
  - Smart / Dumb Components
  - Lazy Loading Module

DISCUSSION

The goal of discussion should not be victory, but progress.

Truong Tran

By ChatGPT