Char5. 서버 프로그램 구현

개발 환경 구축

응용 소프트웨어 개발을 위해 개발 프로젝트를 이해하고 소프트웨어 및 하드웨어 장비를 구축하는 것을 의미

하드웨어 환경

사용자와의 인터페이스 역할을 하는 **클라이언트** 그리고 클라이언트와 통신하여 서비스를 제공하는 **서버**로 나뉜다. Client Server

클라이언트	PC, 스마트폰 등		
서버	웹서버 Web Server	클라이언트로부터 직접 요청 받아 처리하는 서버 저용량 정적파일 등을 제공	Apache HTTP Server Microsoft Internet Information Service Google Web Server
	웹 애플리케이션 서버 WAS	사용자에게 동적 서비스를 제공하기 위해 웹서버로부터 요청받아 데이터 가공 작업을 수행하거나 웹서버와 데이터베이스 서버 또는 웹 서버와 파일 서버 사이에서 인터페이스 역할을 수행하는 서버	Apache Tomcat IBM WebSphere Oracle WebLogic
	데이터베이스 서버 DB Server	데이터베이스와 이를 관리하는 DBMS를 운영하는 서버	MySQL Server Oracle Server Microsoft SQL Server
	파일 서버 File Server	데이터베이스에 저장하기에 비효율적이거나 서비스 제공을 목적으로 유지하는 파일들을 저장하는 서버	AWS S3

HTTP/HTTPS 지원	**브라우저로부터 요청 받아 응답할 때 사용되는 프로토콜**
통신 기록 **Communication Log**	**처리한 요청들을 로그 파일로 기록하는 기능**
정적 파일 관리 **Managing Static Files**	**HTML, CSS, 이미지 등의 정적 파일들을 저장하고 관리하는 기능**
대역폭 제한 **Bandwidth Throttling**	**네트워크 트래픽의 포화를 방지하기 위해 응답 속도를 제한하는 기능**
가상 호스팅 **Virtual Hosting**	**하나의 서버로 여러 개의 도메인 이름을 연결하는 기능**
인증 **Authentication**	**사용자가 합법적인 사용자인지를 확인하는 기능**

소프트웨어 환경

클라이언트와 서버 운영을 위한 시스템 소프트웨어와 개발에 사용되는 개발 소프트웨어로 구성

시스템 소프트웨어		운영체제OS, 웹 서버 및 WAS운용을 위한 서버 프로그램, DBMS	
개발 소프트웨어	요구사항 관리 도구	요구사항 수집과 분석, 추적 등을 편리하게 도와주는 소프트웨어	JIRA, IBM DOORS, inteGREAT, Reqtify, Trello
	설계/모델링 도구	UML을 지원하며 개발의 전 과정에서 설계 및 모델링을 도와주는 소프트웨어	DB Designer, PlantUML, ArgoUML
	구현 도구	개발 언어를 통해 애플리케이션의 실제 구현을 지원하는 소프트웨어	Eclipse, Intellij IDEA, Visual Studio, Netbeans, Node.js
	빌드 도구	구현 도구를 통해 작성된 소스의 빌드 및 배포, 라이브러리 관리를 지원하는 소프트웨어	Ant, Grandle, Maven, Jenkins
	테스트 도구	모듈들이 요구사항에 적합하게 구현되었는지 테스트하는 소프트웨어	CppUnit, JUnit, HttpUnit, NUnit, Spring Test
	형상 관리 도구	산출물들을 버전별로 관리하여 품질 향상을 지원하는 소프트웨어	GIT, CVS, Subversion, Mercurial

개발언어 선정기준

1. **적정성**

   개발하려는 소프트웨어의 목적에 적합해야 한다.

2. **효율성**

   코드의 작성 및 구현이 효율적이어야 한다.

3. **이식성**

   다양한 시스템 및 환경에 적용이 가능해야 한다.

4. **친밀성**

   개발 언어에 대한 개발자들의 이해도와 활용도가 높아야 한다.

5. **범용성**

   다른 개발 사례가 존재하고 여러 분야에서 활용되고 있어야 한다.

팬인(Fan-In)

어떤 모듈을 제어(호출)하는 모듈의 수

팬아웃(Fan-Out)

어떤 모듈에 의해 제어(호출)되는 모듈의 수

모듈 Module

모듈화를 통해 분리된 시스템의 각 기능들

모듈의 기능적 독립성

소프트웨어를 구성하는 각 모듈의 기능이 서로 독립됨

독립성이 높은 모듈은

1. 모듈을 수정하더라도 다른 모듈들에게는 거의 영향을 미치지 않으며
2. 오류가 발생해도 쉽게 발견하고 해결 가능

모듈의 독립성은 응집도 Cohesion 와 비례

결합도 Coupling, 모듈의 크기와 반비례

품질은 응집도 Cohesion 와 비례

결합도 Coupling 와 반비례

재사용되는 대상은 외부 모듈과의 결합도는 낮고 응집도는 높아야 한다.

결합도 coupling

모듈 간의 상호 의존하는 정도 또는 두 모듈 사이의 연관관계

(결합도 약함) 자 스(민) 제 외 공 용 (강함)

자료 결합도 Data Coupling	**모듈 간의 인터페이스가 자료 요소로만 구성 될 때의 결합도**
스탬프 결합도 Stamp Coupling	**모듈간의 인터페이스로 배열이나 레코드 등의 자료 구조가 전달될 때의 결합도**
제어 결합도 Control Coupling	**어떤 모듈이 다른 모듈 내부의 논리적인 흐름을 제어하기 위해 제어 신호를 이용하여 통신하거나 제어 요소를 전달하는 결합도**
외부 결합도 External Coupling	**어떤 모듈에서 선언한 데이터를 외부의 다른 모듈에서 참조할 때의 결합도**
공통 결합도 Common Coupling	**공유되는 공통 데이터 영역을 여러 모듈이 사용할 때의 결합도**
내용 결합도 Content Coupling	**한 모듈이 다른 모듈의 내부 기능 및 그 내부 자료를 직접 참조하거나 수정할 때의 결합도**

응집도 cohesion

정보 은닉 개념을 확장한 것으로,

명령어나 호출문 등 모듈의 내부 요소들의 서로 관련되어 있는 정도

(응집도 강함) 기 차 교환 절차 시 논 유 (약함)

기능적 응집도 Functional Cohesion	**모듈 내부의 모든 기능 요소들이 단일 문제와 연관되어 수행될 경우의 응집도**
순차적 응집도 Sequential Cohesion	**모듈 내 하나의 활동으로부터 나온 출력 데이터를 그 다음 활동의 입력 데이터로 사용할 경우의 응집도**
교환적 응집도 Communicational Cohesion	**동일한 입력과 출력을 사용하여 서로 다른 기능을 수행하는 구성 요소들이 모였을 경우의 응집도**
절차적 응집도 Procedural Cohesion	**모듈이 다수의 관련 기능을 가질 때 모듈 안의 구성 요소들이 그 기능을 순차적으로 수행할 경우의 응집도**
시간적 응집도 Temporal Cohesion	**특정 시간에 처리되는 몇 개의 기능을 모아 하나의 모듈로 작성할 경우의 응집도**
논리적 응집도 Logical Cohesion	**유사한 성격을 갖거나 특정 형태로 분류되는 처리 요소들로 하나의 모듈이 형성되는 경우의 응집도**
우연적 응집도 Coincidental Cohesion	**모듈 내보의 각 구성 요소들이 서로 관련 없는 요소로만 구성된 경우의 응집도**

공통 모듈의 명세 기법

정확성 Correctness	시스템 구현시 해당 기능이 필요하다는 것을 알 수 있도록 정확히 작성한다.
명확성 Clarity	해당 기능을 이해할 때 중의적으로 해석되지 않도록 명확하게 작성한다.
완전성 Completeness	시스템 구현을 위해 필요한 모든 것을 기술한다.
일관성 Consistency	공통 기능들 간 상호 충돌이 발생하지 않도록 작성한다.
추적성 Traceability	기능에 대한 요구사항의 출처, 관련 시스템 등의 관계를 파악할 수 있도록 작성한다.

재사용 Reuse

배용과 개발시간을 절약하기 위해 이미 개발된 기능들을 파악하고 재구성하여 새로운 시스템 또는 기능 개발에 사용하기 적합하도록 최적화 시키는 작업

재사용 규모에 따른 분류

- 함수와 객체: 클래스나 메소드 단위의 소스코드를 재사용한다.
- 컴포넌트 자체에 대한 수정없이 인터페이스를 통해 통신하는 방식으로 재사용
- 공통된 기능을 제공하는 애플리케이션을 공유하는 방식으로 재사용

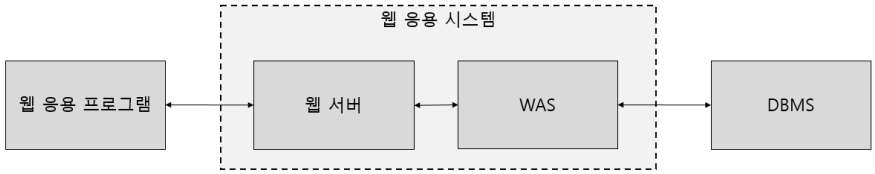
DBMS 접속

사용자가 데이터를 사용하기 위해 응용 시스템을 이용하여 DBMS로부터 전달 받은 결과를 사용자에게 전달하는 매개체 역할을 수행

웹 응용 시스템

웹 서버와 웹 애플리케이션 서버로 구성되며 서비스 규모가 작은 경우 웹 서버와 웹 애플리케이션 서버를 통합하여 하나의 서버만을 운용할 수 있다.

DBMS 접속 기술



JDBC	Java DataBase Connectivity Java언어로 다양한 종류의 데이터베이스에 접속하고 SQL문을 수행할 때 사용되는 표준 API
ODBC	Open DataBase Connectivity 데이터베이스에 접근하기 위한 표준 개방형 API로 개발 언어에 관계없이 사용할 수 있음
MyBatis	JDBC코드를 단순화하여 사용할 수 있는 SQL Mapping 기반 오픈 소스 접속 프레임워크 SQL을 거의 그대로 사용할 수 있어 SQL 친화적인 국내 환경에 적합하여 많이 사용됨

	정적 SQL (Static SQL)	동적 SQL (Dynamic SQL)
SQL 구성	커서를 통한 정적 처리	문자열 변수에 담아 동적 처리
개발 패턴	커서의 범위 안에서 반복문을 활용해 SQL 작성	NVL 함수 없이 로직을 통해 SQL 작성
실행 속도	빠름	느림
사전 검사	가능	불가능

서버 개발 프레임워크

서버 프로그램 개발 시 다양한 네트워크 설정, 요청 및 응답 처리, 아키텍처 모델 구현 등을 손쉽게 처리할 수 있도록 클래스나 인터페이스를 제공하는 소프트웨어

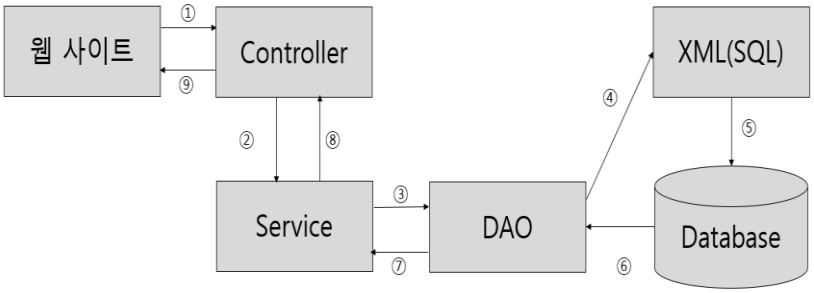
프레임워크	특징
spring	JAVA를 기반으로 만들어진 프레임워크로, 전자정부 표준 프레임워크의 기반 기술로 사용되고 있음
Node.js	JavaScript를 기반으로 만들어진 프레임워크로, 비동기 입·출력 처리와 이벤트 위주의 높은 처리 성능을 갖고 있어 실시간으로 입·출력이 빈번한 애플리케이션에 적합함
Django	Python을 기반으로 만들어진 프레임워크로, 컴포넌트의 재사용과 플러그인화를 강조하여 신속한 개발이 가능하도록 지원함
Codeigniter	PHP를 기반으로 만들어진 프레임워크로, 인터페이스가 간편하며 서버 자원을 적게 사용함
Ruby on Rails	Ruby를 기반으로 만들어진 프레임워크로, 테스트를 위한 웹 서버를 지원하며 데이터베이스 작업을 단순화, 자동화시켜 개발 코드의 길이가 짧아 신속한 개발이 가능함

프레임워크의 특징

모듈화 (Modularity)	프레임워크는 캡슐화를 통해 모듈화를 강화하고 설계 및 구현의 변경에 따른 영향을 최소화함으로서 소프트웨어의 품질을 향상시킴
재사용성 (Reusability)	프레임워크는 재사용 가능한 모듈들을 제공함으로써 개발자의 생산성을 향상시킴
확장성 (Extensibility)	프레임워크는 다형성(Polymorphism)을 통한 인터페이스 확장이 가능하여 다양한 형태와 기능을 가진 애플리케이션 개발이 가능함
제어의 역흐름 (Inversion of Control)	개발자가 관리하고 통제해야 하는 객체들의 제어 권한을 프레임워크에 넘김으로써 생산성을 향상시킴

서버 프로그램 개발 과정

DTO (Data Transfer Object)/ VO(Value Object) 구현	• 데이터 교환을 위해 사용할 객체를 만드는 과정 • 변수 및 객체를 송 · 수신할 데이터의 자료형(Data Type)에 알맞게 생성함 • 알고리즘 등의 로직은 구현하지 않고, 변수와 데이터를 저장하고 반환하는 메소드만 구현함
SQL 구현	• 데이터의 삽입, 변경, 삭제 등의 작업을 수행할 SQL문을 생성하는 과정 • SQL문은 소스 코드 내에 직접 입력하거나, 별도의 XML 파일로 저장하여 관리함 • XML 파일로 SQL문을 관리하는 경우 중복되는 SQL문을 최소화할 수 있고, 유지보수가 간편해짐
DAO (Data Access Object) 구현	데이터베이스에 접근하고, SQL을 활용하여 데이터를 실제로 조작하는 코드를 구현하는 과정
Service 구현	사용자의 요청에 응답하기 위한 로직을 구현하는 과정
Controller 구현	사용자의 요청에 적절한 서비스를 호출하여, 그 결과를 사용자에게 반환하는 코드를 구현하는 과정



① 웹 사이트로부터 사용자의 요청이 Controller에 전달

② Controller는 해당 요청에 맞는 Service를 호출

③ Service는 수행을 위한 데이터를 DAO에 요청

④ ~ ⑥ DAO는 XML을 통해 Database로부터 service가 요청한 데이터를 가져옵니다.

⑦ 가져온 데이터를 service에 반환

⑧ Service의 수행 결과를 Controller에 반환

⑨ Controller의 수행 결과를 웹 사이트에 반환

* DTO/VO는 ①, ⑤, ⑨를 제외한 데이터 교환 전 과정에서 요청과 응답 시 사용됩니다.

배치 프로그램 Batch Program

사용자의 상호작용 없이 여러 작업들을 미리 정해진 일련의 순서에 따라 일괄적으로 처리하는 것

1. 정기배치 2. 이벤트성 배치 3. On-Demand 배치

- 배치 프로그램의 필수 요소

대용량 데이터	대량의 데이터를 가져오거나, 전달하거나, 계산하는 등의 처리가 가능해야 한다.
자동화	심각한 오류가 발생하는 상황을 제외하고 사용자의 개입 없이 수행되어야 한다.
견고성	잘못된 데이터나 데이터 중복 등의 상황으로 중단되는 일 없이 수행되어야 한다.
안정성/신뢰성	오류가 발생하면 오류의 발생 위치, 시간 등을 추적할 수 있어야 한다.
성능	다른 응용 프로그램의 수행을 방해하지 않아야 하고, 지정된 시간 내에 처리가 완료되어야 한다.

배치 스케줄러 Batch Scheduler

일괄 처리Batch Processing작업이 설정된 주기에 맞춰 자동으로 수행되도록 지원해주는 도구

스프링 배치 (Spring Batch)	Sprinf Source사와 Accenture사가 2007년 공동 개발한 오픈소스 프레임워크	Job	수행할 작업 정의
		Job Launcher	실행을 위한 인터페이스
		Step	Job처리를 위한 제어 정보
		Job Repository	Step의 제어 정보를 포함하여 작업 실행을 위한 모든 정보 저장
Quartz	스프링 프레임워크로 개발되는 응용 프로그램들의 일괄처리를 위한 다양한 기능을 제공하는 오픈소스 라이브러리	Scheduler	실행 환경 관리
		Job	수행할 작업 정의
		JobDetail	Job의 상세 정보
		Trigger	Job의 실행 스케줄 정의
Cron	리눅스의 스케줄러 도구로 crontab 명령어를 통해 작업을 예약할 수 있음	-e	편집기를 호출하여 작업 추가 및 수정
		-l	작업 목록 출력
		-r	작업 삭제