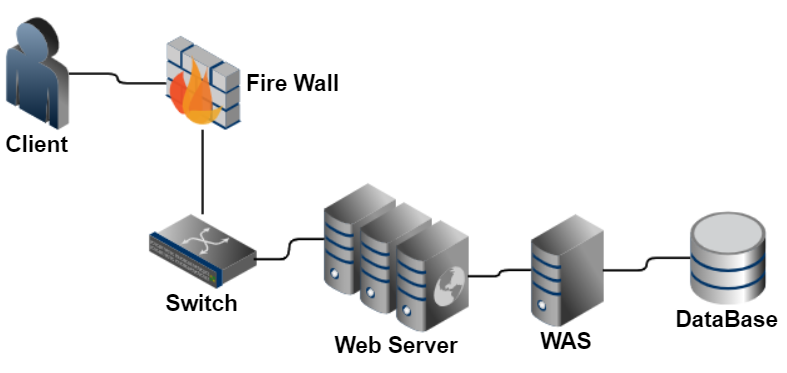
**TERM PROJECT 1**

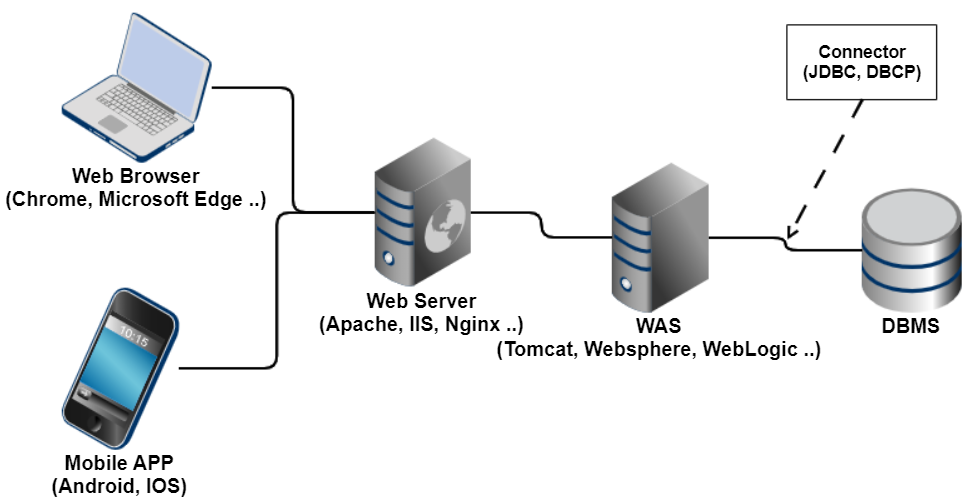
|  |  |
| --- | --- |
| **사 원 명** | **윤형오** |
| **입 사 일** | **2020.02.06** |
| **담 당 자** | **윤용성 부장** |
| **수습 기간** | **2020. 02. 06 ~ 2020. 05. 05** |

**목차**

1. **WEB 시스템 구성도**
   1. **H/W 구성도**
   2. **S/W 구성도**
   3. **H/W, S/W의 역할과 관련 소프트웨어**
   4. **N/W 구성도**
   5. **N/W의 역할과 관련 소프트웨어**
2. **WEB 시스템 기술요소 설명**
   1. **클라이언트측 기술요소**
   2. **서버측 기술요소**
3. **WEB 시스템 구조와 논리**
   1. **WEB 시스템 구조**
   2. **WEB 시스템 논리**
4. **WEB 시스템 개발 방법론**
   1. **정보공학 개발론**
   2. **객체지향 개발론**
   3. **CBD 개발론**
   4. **애자일 개발론**
5. **WEB 시스템 구성도**
   1. **H/W 구성도**



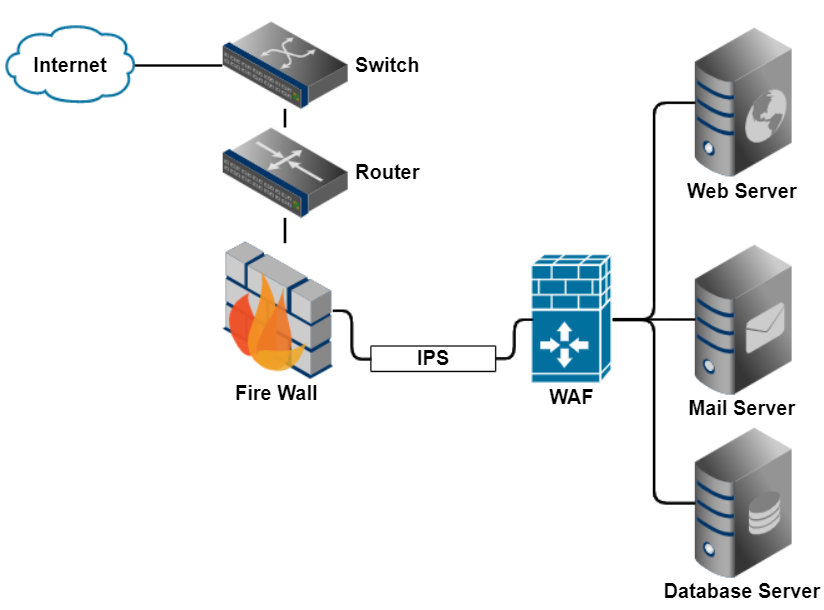
* 1. **S/W 구성도**



* 1. **H/W, S/W의 역할과 관련 소프트웨어**

|  |  |  |
| --- | --- | --- |
| 명칭 | 역할 | 종류/소프트웨어 |
| Client | 서버가 제공하는 서비스를 요청하고, 서버로부터 반환되는 응답을 사용자에게 적절한 방식으로 표현하는 기능을 가진 프로그램이나 시스템 | Chrome, Microsoft Edge, Internet Explorer, Firefox, Android, IOS 등 |
| Web Server | 클라이언트로부터 HTTP 요청을 받아 HTML문서나 이미지, 파일 등의 정적인 오브젝트를 전송해주는 서버  클라이언트가 정적인 컨텐츠 이외의 요청을 할 경우 WAS로 요청을 넘겨주고 WAS로부터 받은 결과를 클라이언트에 응답한다. | Apache, IIS, Nginx, Google Web Server, Zeus 등 |
| WAS | 데이터 베이스 조회나 로직 처리 등 동적인 컨텐츠를 제공하는 서버  WAS도 자체적으로 Web Server 기능을 내장하고 있어 정적인 컨텐츠를 처리할 수 있다. 하지만 자원 이용의 효율성 및 장애 극복, 배포 및 유지보수의 편의성을 위해 Web Server와 WAS를 분리해서 사용한다. | Tomcat, Websphere, WebLogic, JBOSS, Resin, Jeus 등 |
| DB | 다수의 사용자가 공유하고 사용할 목적으로 데이터들을 통합하여 관리하는 데이터 집합체를 의미 | 계층형 DB, 네트워크형 DB, 관계형 DB, NoSQL |
| DBMS | 사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해 주고 데이터베이스를 관리해주는 소프트웨어  데이터베이스를 구축하는 틀을 제공하고, 효율적으로 데이터를 검색하고 저장하는 기능을 제공한다. 다른 응용 프로그램들이 데이터베이스에 접근할 수 있는 인터페이스와 장애 복구 기능 및 사용자 권한에 따른 보안성 유지 기능 등을 제공한다. | Oracle, MySQL, MS-SQL, MariaDB, MongoDB, sqlite 등 |
| Connector | 데이터베이스에 접근하여 데이터 조회, 삽입, 수정, 삭제를 가능하게 하는 응용 프로그램 인터페이스 | JDBC, ODBC |

* 1. **N/W 구성도**



* 1. **N/W의 역할과 관련 소프트웨어**

|  |  |
| --- | --- |
| 명칭 | 역할 |
| Switch | 같은 네트워크 내부에서 데이터 전송을 수행하는 기기  MAC 주소를 사용하여 같은 네트워크의 LAN포트간 데이터 전송 |
| Router | 서로 다른 네트워크 사이의 데이터 전송을 수행하는 기기  IP주소를 사용하여 네트워크 간의 데이터 전송 |
| Fire Wall | 외부 네트워크로부터 내부 시스템을 보호하기 위해 네트워크 계층에서 IP와 Port 정보를 이용해 접근을 제어하는 보안 솔루션  네트워크 계층에서 암호화된 데이터를 복호화 할 수 없어 웹 공격을 예방하는데 한계 존재 |
| WAF  (Web Application Fire Wall) | 웹 어플리케이션 보안에 특화되어 개발된 솔루션  어플리케이션 계층에서 HTTP 프로토콜을 문자열 비교 방식으로 감시하여 침입 시도를 탐지/차단 |
| IPS  (침입 방지 시스템) | 악의적인 공격에 대한 공격탐지를 하고 설정해 놓은 규칙에 기반해 즉각적으로 대응하는 시스템 |
| IDS  (침입 탐지 시스템) | 방화벽의 부족한 점을 채우기 위해 외부 침입에 대한 정보를 수집/분석하여 침입을 탐지하고 관리자에게 알려 대응하는 시스템 |

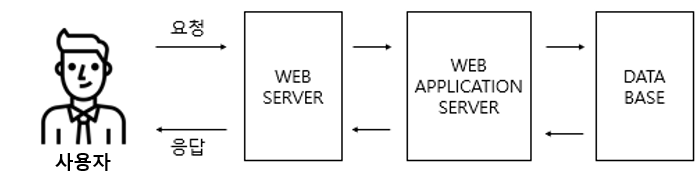
1. **WEB 시스템 기술요소 설명**
   1. **클라이언트측 기술요소**

|  |  |
| --- | --- |
| 명칭 | 역할 |
| HTML | ‘Hypertext Markup Language’의 약자로, 웹 페이지를 만드는 가장 기초가 되는 마크업 언어이며 내용과 구조를 담당해 웹 페이지의 큰 뼈대를 구성한다. |
| Javascript | HTML과 CSS로 만들어진 웹 페이지를 동적으로 변경해줘 풍부하게 해주는 작고 가벼운 언어, 웹 브라우저에서 실질적인 프로그래밍을 담당한다. |
| CSS | ‘Cascading Style Sheets’의 약자로 HTML의 스타일을 지정하고 꾸미는 것을 담당, HTML은 정보를, CSS는 디자인을 관리해 HTML을 수정하지 않아도 CSS 파일 하나만 수정해 CSS를 적용한 HTML의 디자인을 변경할 수 있어 유지보수와 가독성이 좋아지는 장점이 있다. |
| Plugin | 현대적인 웹 페이지들은 개발의 편리성, 효율성을 위해 각종 HTML, CSS, Javascript 기반의 플러그인 프로그램을 통해 개발이 이루어진다.  Ex) Javascript 기반의 JQuery, HTML과 CSS 기반의 Bootstrap 등 |

* 1. **서버측 기술요소**

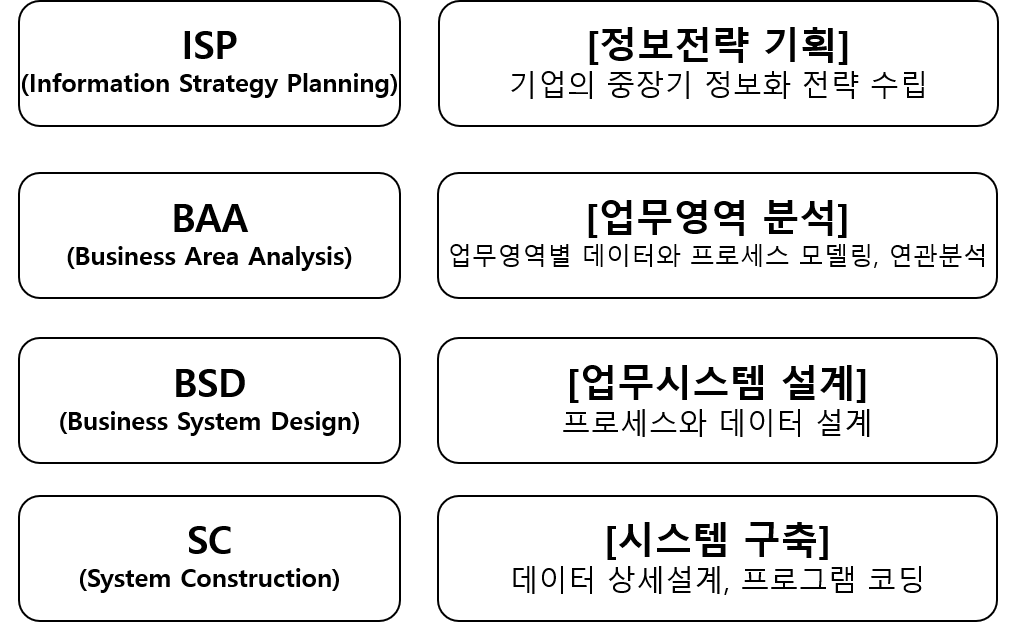
|  |  |
| --- | --- |
| 명칭 | 역할 |
| JSP | HTML에 Java 코드를 넣어 동적 웹 페이지를 쉽게 구축할 수 있도록 만드는 스크립트 언어 |
| PHP | 거의 모든 운영체제에서 구현 가능하고 웹 개발에 필요한 수 많은 로직들이 함수의 형태로 미리 제공된다. 처리 속도 역시 매우 빠르고 코드 작성이 쉽고 문법이 간단해 빠른 생산성과 저렴한 유지비용이 장점 |
| ASP | ‘Active Server Page’의 약자, 마이크로소프트사에서 개발한 동적으로 서버에서 작동하는 페이지 PHP와 다르게 윈도우 운영체제에서만 작동한다는 단점이 있다. |
| ASP.NET | 마이크로소프트사가 개발한 프로그래머들이 동적인 웹 사이트, 웹 어플리케이션, 웹 서비스를 만들 수 있게 도와주는 개발 기술 |

1. **WEB 시스템 구조와 논리**
   1. **WEB 시스템 구조**

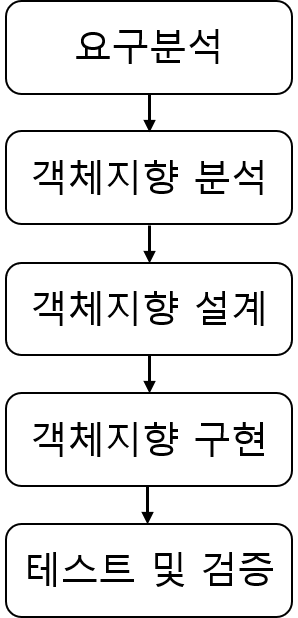


* 1. **WEB 시스템 논리**
     1. 클라이언트가 웹 서버에 페이지 요청
     2. 웹 서버는 요청을 받아들여 요청한 정적인 페이지를 클라이언트에게 응답
     3. 이때, 클라이언트가 동적인 페이지를 요청할 경우 웹 서버는 WAS로 요청을 전달하고 WAS는 데이터베이스와 로직을 처리
     4. WAS는 처리한 결과를 다시 웹 서버로 전달
     5. 웹 서버는 전달받은 결과를 클라이언트에 응답

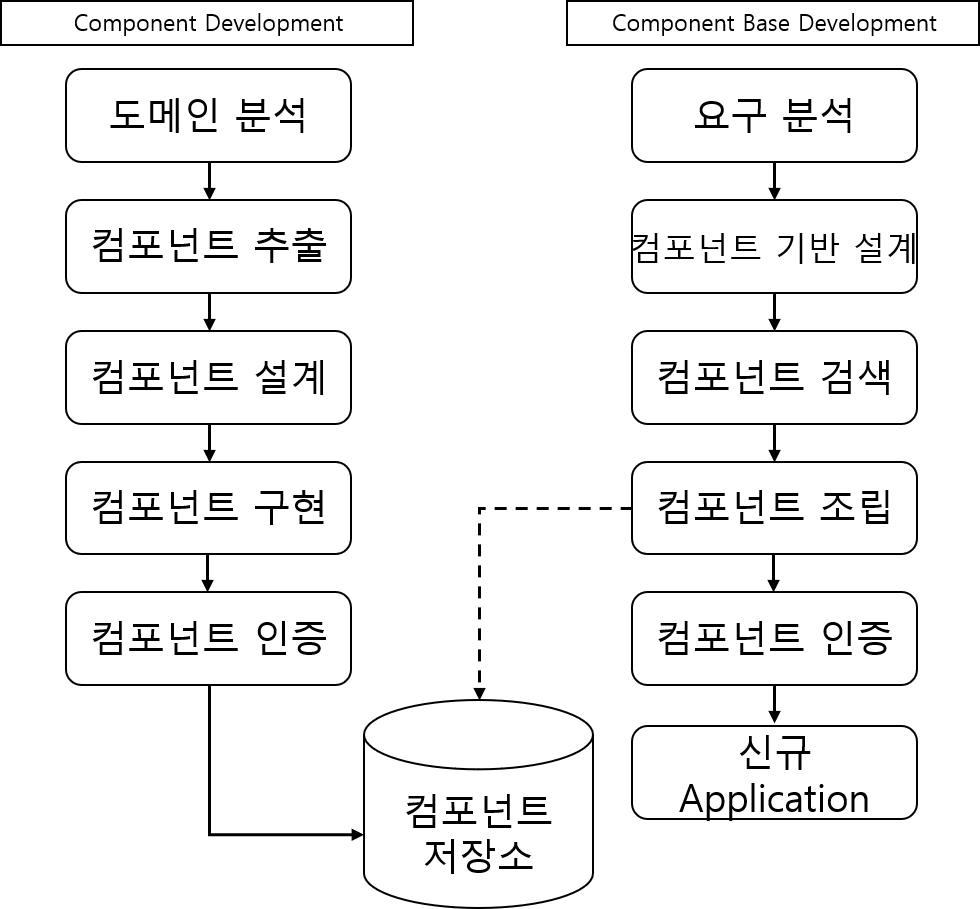
1. **WEB 시스템 개발 방법론**
   1. **정보공학 개발 방법론**
      * + 정보공학 개발 방법론은 1980년대 중반에 등장한 방법론으로 기업의 전사적인 관점에서 출발해 데이터 중심으로 시스템을 구축하는 방법론이다.
        + 정보공학 개발론 단계별 수행 내용



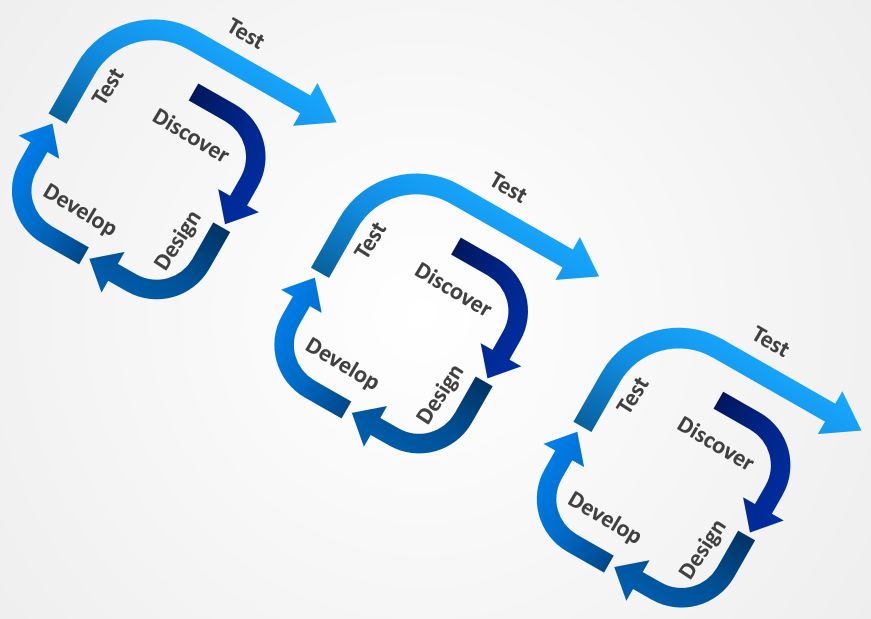
* ISP 단계: 기업의 경영전략을 뒷받침할 수 있는 정보화 전략을 수립하기 위해 현행 업무 프로세스와 시스템을 분석하고 미래 아키텍처와 전략계획을 수립
* BAA 단계: 기업의 업무 현황을 분석해서 개념 수준의 데이터와 프로세스를 설계하는 업무분석 단계
* BSD 단계: 실질적으로 시스템을 설계하는 단계, ERD, 분할 다이어그램, 액션 다이어그램, 의존 다이어그램 등을 사용해 프로세스를 설계한다.
* SC 단계: 물리적 데이터 베이스를 설계하고 BSD 단계에서 작성한 산출물을 바탕으로 프로그램 개발
  + - * 장점: 일관성 있고 통일된 정보 시스템 구축 가능, 시스템의 장기적인 진화, 발전 허용, 데이터 중심으로 업무 절차 및 환경변화에 유연
      * 단점: 정보 공학 효과를 위해 장기간의 시간 필요, 기능의 유지보수와 재사용성 낮음
  1. **객체 지향 방법론**
     + - 요구분석, 설계, 구현, 테스트 및 검증 등의 소프트웨어 생명주기에 객체 지향 개념을 접목시켜 일관된 모델을 가지고 소프트웨어를 개발하는 방법론



* + - * 객체지향 개발 방법론은 전 과정을 객체 중심으로 진행한다. 데이터를 저장하는 테이블도 데이터 객체로 설계해서 관계형 DB를 사용할 경우 객체를 관계형 테이블로 변환해주는 과정이 필요하다. 이 때문에 객체지향 방법론은 실 세계를 정확하게 반영하며, 높은 재사용성과 안전성을 갖고 있다는 장점이 있지만, 관계형 테이블로 전환해주는 매핑 과정이 필수적으로 따라오기 때문에 활발하게 사용되지는 않는다.
  1. **CBD(Component Based Development) 방법론**
     + - 재사용 가능한 컴포넌트의 개발 또는 상용 컴포넌트를 조합해 어플리케이션 개발 생산성과 품질을 높이고 시스템 유지보수 비용을 최소화할 수 있는 개발 방법론



* + - * 컴포넌트를 개발하는 Component Development 단계와 컴포넌트를 이용해서 개발하는 Component Base Development 단계로 나뉜다.
      * 적절한 상황에서 사용한다면 재사용성이 우수하고 컴포넌트 조립을 통한 생산성을 높일 수 있다는 장점이 있지만, 높은 설계와 개발 수준을 요구해 활용하기 힘들다는 한계가 있다.
  1. **애자일 방법론**
     + - 기존 방법론들이 절차를 너무 중시해 변화에 대응하기 어렵다는 단점을 개선하기 위해 나온 방법론으로, 절차보다 사람, 문서보다는 소프트웨어, 철저한 계획보다 고객의 요구 변화에 따른 신속한 대응을 중요하게 생각하는 개발 방법론



* + - * 고객과 팀원 및 개발자의 끊임없는 협의와 교류를 통해 최대한 고객의 요구사항에 맞춰 모든 변화에 빠르고 유연하게 대응하는 것이 목표
      * 프로젝트를 수행하는데 있어 한정된 시간과 한정된 자원으로 끊임없이 고객의 니즈를 만족시킨다는 것은 불가능하다. 프로젝트 규모가 커질수록 들어가는 비용은 점차 불어나고, 고객은 자신의 정확한 요구사항이 무엇인지도 모르고 제시할 수 있으며 구현이 불가능한 요구도 생길 것이다. 이러한 여러 이유들 때문에 애자일 방법론은 실제사용하기에 무리가 있는 단점들을 보유하고 있다.