

웹 기반 프로젝트 협업 관리도구

김형근, 임우철, 최환용, 서영조
 경상대학교 컴퓨터학과
 e-mail: lminggvick@naver.com

Web-based Project Collaboration Management Tool

Hyeong-Geun Kim, Woo-Cher Leem, Hwan-Yong Choi, Young-Jo Seo
 Dept of Computer Science, Gyung-Sang University

요 약

학생 때 규모가 작은 프로젝트에서 직장에서의 규모가 큰 프로젝트까지 다른 사람들과 협업하며 프로젝트를 진행하는 것이 당연하다. 효율적으로 프로젝트를 수행하기 위한 간단하면서도 편리한 UI를 제공하는 웹 기반 프로젝트 협업 관리 도구를 제작하였다. 본 프로젝트는 소프트웨어 공학에서 사용되는 디자인 패턴 중 하나인 MVC 2 Model을 사용하였으며, 단순 서비스 제공을 목적으로 하는 것뿐만 아니라 시스템 성능과 보다 유연한 서비스를 제공하기 위해서 Maven, 자동화 스크립트, Logback, 단위 테스트, 데이터 6단계 정규화 등을 사용하는 환경을 구축하여 구현하였다.

1. 서론

학생부터 직장인까지 다양하고 많은 프로젝트를 경험한다. 효율적으로 프로젝트를 수행하기 위해서는 협업이 매우 중요하다. 현재 상용화되고 있는 협업도구들은 영어로 구성되어 있으며 기능을 사용하기 위해서는 학습이 필요해 처음 접했을 때 대부분 사용에 힘들어한다. 간단하면서도 편리한 UI를 제공하는 협업도구의 필요성을 느꼈다.

프로젝트 생성, 수정 및 삭제를 간단한 UI로 설계하여 편리성을 높였고 카드 형태로 관리 가능하도록 하였다. 드래그 앤 드롭 방법으로 카드와 리스트를 이동하여 변동이 많은 일정을 관리하기 편리하다. 각 카드마다 일을 처리할 담당자, 체크리스트, 파일 등을 업로드 할 수 있고 프로젝트 계획과 진행 상황, 세부적인 일정들을 간트 차트를 이용해 수정하고 나타낼 수 있다.

채팅서버를 구현하여 다른 응용 프로그램 없이 프로젝트 관리와 동시에 협업 팀원들과 실시간으로 회의가 가능하다. 이미지를 첨부하여 캠퍼스안에 해당 이미지를 보여줌으로써 그리기 도구를 이용해서 부가설명을 할 수 있다.

기존 서비스들에서 잘 이루어지지 않았던 데이터 간의 연동을 개선하여 Board 메뉴의 카드와 간트 차트 메뉴 간 이중 작업을 방지하였다.

2. 사용된 기술

2.1 JSP & Servlet

JSP (JavaServer Page) 는 HTML 내에 자바코드를 삽입할 수 있도록 설계된 언어이다. 자바코드를 이용하여 웹 서버에서 특정 역할을 수행하여 그 결과를 웹 브라우저에 돌려주는 방식으로 작동한다.

Servlet은 자바를 이용해 웹 서버 성능을 향상시키기 위한 자바 클래스이다. JSP 에 비해, Servlet 은 자바 코드 안에 HTML을 포함하고 있다는 차이가 있다.

2.2 MVC 2 Model

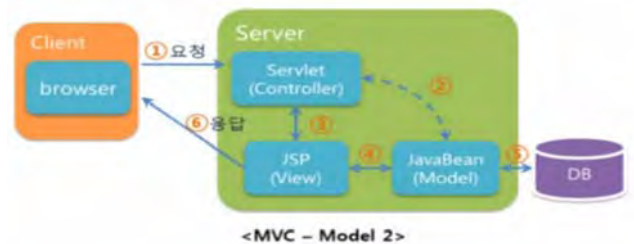


그림 1. MVC 2 Model 구조 및 행위 모델

소프트웨어 공학에서 사용되는 디자인 패턴 중 하나인 ‘MVC’ 는 Model-View-Controller 3가지를 칭하는 뜻이다. Model 은 애플리케이션의 정보(데이터, 객체)를 나타내며, View 는 사용자 인터페이스 요소를 나타내며, Controller 는 데이터와 비즈니스 로직 사이에서 특정 동작을 수행한다.

웹브라우저에서 들어온 요청을 처리하는 데 있어서, ‘jsp - jsp’ 와 같은 형식을 거치는 것이 Model 1 방식이다. 한편, Model 2 방식은 ‘jsp - servlet - jsp’ 와 같은 과정으로 처리된다. 즉, 요청이 들어온 jsp에서 그 요청을 servlet을 거쳐 처리가 되고 그에 대한 결과를 jsp 에 돌려주는 형식이다. Model 1은 프로젝트 규모가 커짐에 따라 가독성이 떨어지며 유지보수가 힘들다는 단점이 존재한다. 무엇보다, 테스트 작업을 위해서 브라우저를 통해서만 진행해야 한다. 이 단점들을 Model 2 방식으로써 해결할 수 있다.

결과를 웹브라우저에 돌려주는 방식으로는 단순히 요청을 처리하여 결과 페이지로 이동만 시키는 redirecting 과 요청에 맞는 처리를 한 후 그 결과를 jsp 페이지에 넘겨주는 forwarding 방식을 사용하였다.

2.3 Maven Build Tool

Maven 은 자바 프로젝트의 라이프사이클을 관리하는데 사용하는 도구이다. 컴파일과 빌드를 동시에 수행하여 테스트 작업을 수행하고, 어플리케이션에 사용된 다양한 라이브러리를 의존성(Dependency) 관계를 통해서 유연하게 관리할 수 있도록 한다.

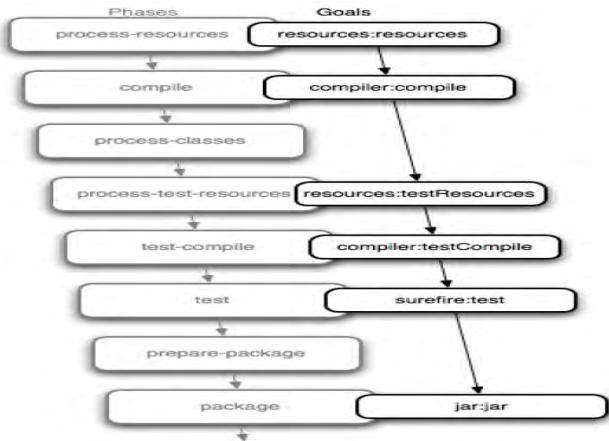


그림 2. Maven Lifecycle & Goals

Maven 활용하여, 공동의 프로젝트를 효율적으로 관리하기 위해 pom.xml 설정을 통해 디렉토리 구조를 설계하고 의존성 관계를 이용한 라이브러리 관리하였다. 또한, Maven에서 제공되는 다양한 플러그 인을 사용하였다.

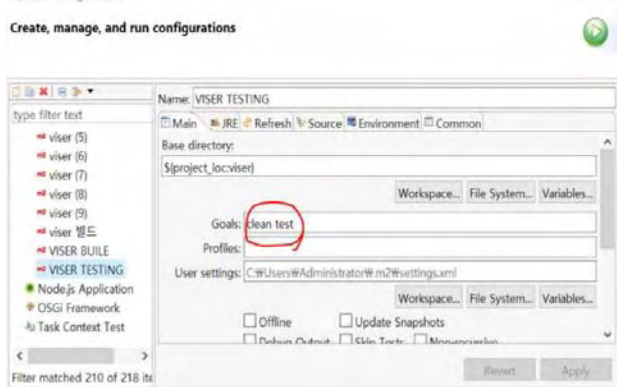


그림 3. Maven Goals 활용 예

2.4 fabric 자동화 스크립트

스크립트란 프로그래밍 언어로서 응용 소프트웨어를 제어하는 언어의 일종이다. fabric은 SSH(Secure Shell)를 통해 애플리케이션을 배포 하거나 시스템 관리자 일을 돕는다. 이는 Python에서 제공되는 library 이다.

애플리케이션의 버전 업이나 유지보수 등의 작업 이후에 다시 서버로 배포하는 과정에서 발생하는 비용을 줄이기 위해서, python의 fabric을 이용하면 자동화 스크립트를 생성해 유닉스 환경에서 명령어 하나로 서버에 배포할 수 있다.

```

1 from fabric.api import *
2
3
4 env.hosts = ["ec2-52-78-74-182.ap-northeast-1.compute.amazonaws.com"]
5 env.user = "ubuntu"
6 env.key_filename = "C:/cygwin64/home/ping2/.ssh/awspwd.pem"
7 env.port = 22
8
9
10 def build():
11     sudo('cd /home/ubuntu/viser')
12     run('mvn clean package')
13
14 def applyChanges():
15     sudo('cd /home/ubuntu/viser')
16     run('git pull')
17
18 def start():
19     sudo('/etc/init.d/tomcat8 start') # tomcat8 start
20
21 def stop():
22     sudo('/etc/init.d/tomcat8 stop') # tomcat8 stop
23
24 def delete():
25     sudo('rm -rf /var/lib/tomcat8/webapps/viser') # directory delete
26
27 def copy():
28     sudo('cp /home/ubuntu/viser/target/viser.war /var/lib/tomcat8/webapps/') # file copy
29
30
31 def deploy():
32     execute(applyChanges)
33     execute(build)
34     execute(stop)
35     execute(delete)
36     execute(copy)
37     execute(start)
    
```

그림 4. fabric 자동화 스크립트

2.5 Javascript, jQuery

2.5.1 UI

HTML 클라이언트 측 조작을 단순화 해 주는 것이 jQuery 이다. UI 중 카드의 드래그 앤 드롭, 각종 메뉴 팝업 등의 시각효과를 위해 사용하였다.

2.5.2 클라이언트

캔버스 스케치, 채팅 기능, 회원가입 시의 비밀번호 유효성 검사 등 클라이언트 측 기능들은 javascript 와 jQuery 를 이용해 구현하였다.

2.5.3 Ajax

비동기적으로 Json 형식의 데이터를 주고받을 수 있도록 하여 페이지의 이동이나, 새로 고침 없이 바로 데이터를 화면에 갱신시킬 수 있도록 하기 위해 Ajax를 이용하였다.

2.5.4 Drag & Drop

특정 보드 안에 리스트, 카드를 마우스로 간편하게 옮길 수 있도록 유연한 프로젝트 관리가 될 수 있도록 하였다. 이들은 각각 순서 값을 가지는데 이 때 위치 변화가 발생하는데 이는 순서 갱신 알고리즘을 통해서 순서 값들이 변경되어 DB 에 갱신되도록 설계 하였다.

2.5.5 웹 소켓 기반 채팅

TCP 접속으로 전 이중 통신 채널을 제공하는 웹 소켓을 이용해 비동기적으로 데이터를 실시간으로 동기화 할 수 있도록 채팅 기능을 구현했다. 웹 소켓 서버는 자바코드로 하나를 두고 클라이언트측은 javascript를 통해 문자 채팅이 나 그림 그리기, 데이터 변동을 감지하여 웹 소켓 서버로 JSON 데이터를 전송한다. 웹 소켓 서버는 클라이언트가 보낸 JSON 형식의 데이터를 parsing 하여 프로젝트 명에 등록되어있는 session 들에게 다시 전달해준다.

3. 구성

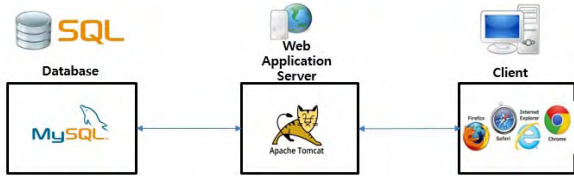


그림 5. viser 시스템 구성도

Fig 5은 본 웹 애플리케이션 서비스의 시스템 구성도이다. 별도의 설치과정 필요 없이 서비스하기 위해서 아마존 웹 서비스로 부터 도메인을 할당 받고 해당 도메인 서버에 Tomcat 서버를 구축하여 데이터 관리를 위한 MySQL DB를 사용하였다. 클라이언트 측에서 요청이 들어오면 Fig 1과 같은 방식으로 응답을 되돌려준다.

3.1 Web Application 배포 비용 절약 방법

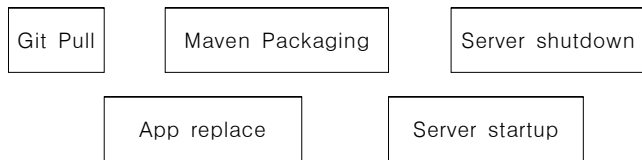


그림 6. 애플리케이션 서버 배포 과정

배포 과정은 Fig 6과 동일하다. 변경 이슈가 발생할 때마다 위 과정을 수행하는 것은 시간이 오래 걸릴 뿐더러 효율적인 서비스를 제공할 수 없기 때문에, 앞서 언급한 Fabric 자동화 스크립트를 이용하면 유닉스 환경에서 './fabric.sh' 명령어 하나로 위 과정을 한 번에 해결함으로써 비용을 절감할 수 있다.

3.2 시스템 성능을 고려한 효율적인 테스트

3.2.1 사용한 디버그 방식

본 애플리케이션 특성 상, 상호 의존성 있는 데이터를 통신하므로 프로젝트 규모가 커짐에 따라 성능 이슈가 발생할 가능성이 매우 높다. 따라서 만약 본 프로젝트 변경 사항에 관한 작업을 수행 중일 때 디버그 과정을 'logback' 라이브러리를 통해 수행함으로써 시스템 성능 저하를 줄였다. 또한, 비즈니스 기능 별로 Junit 기반 테스트 소스를 이용해 단위 테스트 소스를 별도로 구축하여 테스트를 진행함으로써 실제 서버를 구동하여 해당 기능을 테스트 하는 것이 아니므로, 보다 빠른 시일에 애플리케이션 서비스를 제공할 수 있고, 시스템을 안정화 할 수 있도록 프로젝트를 구성하였다.

3.2.2 체계적인 Model (객체) 간 의존성 관리

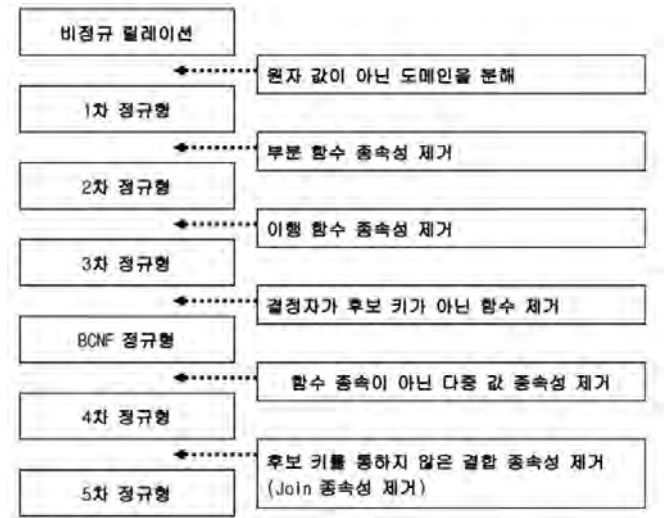


그림 7. 데이터베이스 6단계 정규화 과정

본 애플리케이션은 프로젝트, 보드, 리스트, 카드 등 다양한 의존 관계를 가지는 객체들로 구성된다. 이러한 객체 간의 관계를 효율적으로 관리하지 않으면 데이터를 관리하고 통신하는 과정에서 시스템이 불안정해 질 수 밖에 없다. 이에 따라, Fig 7과 같이 데이터베이스 정규화 과정을 통해서 상호 연관 데이터를 분석, 스키마를 설계하고, 수정, 삭제의 경우 상호 연관성을 가지는 데이터에 대해서는 'Delete, Update Cascade' 외래 키 제약 조건을 설정하여 보다 효율적인 데이터 관리가 될 수 있도록 구성했다.

4. 알고리즘



그림 8. 리스트와 카드 예

본 웹 애플리케이션에 적용된 기술인 카드 / 리스트의 드래그 앤 드롭은 순서 갱신 알고리즘을 적용하였다. 이 알고리즘은 다음 3가지 경우의 수가 존재한다.

4.1 리스트의 순서만을 갱신

한 보드에 대하여 test1 과 test2, test3 이라는 리스트가

존재한다고 할 때, 리스트는 왼쪽부터 0, 1, 2의 순서 값을 가진다. 이때, test3 리스트를 test1 위치로 옮기면 각 리스트의 순서는 test1 이 0에서 1로, test2 는 1에서 2로, test3 은 2에서 0으로 갱신된다.

즉, 리스트의 경우 이동되어진 리스트의 순서 뒤로 기존에 있던 리스트들의 순서 값이 1씩 증가한다.

4.2 같은 리스트 내에서 카드 순서를 갱신

카드의 경우는 가장 첫 번째부터 0 아래로 내려갈수록 1씩 증가되는 순서를 가진다. test1 에 있는 aa 카드를 bb 위치로 이동할 경우 기존에 순서가 0이었던 bb 가 1, cc 가 2로 순서가 1씩 증가된다.

4.3 다른 리스트로의 카드 순서를 갱신

test1 에 있는 cc 카드를 test2 의 22 위치로 이동할 경우에는 cc 아래에 있던 aa 의 순서가 1씩 감소되고 이동되는 22 위치에 기존에 있었던 22 의 순서가 1씩 증가한다.

5. 결론 및 향후과제

본 애플리케이션은 프로젝트의 중요성이 커짐에 따라 많은 프로젝트 과정을 진행하는 현 대학생들로 하여금 높은 접근성을 가지게끔 하려는 목적을 가지고 유연한 애플리케이션 배포와 시스템 성능 차원을 고려한 애플리케이션이다. 따라서 사용자들로 하여금 높은 접근성을 가지며 프로젝트에서 중요한 ‘협업’을 자연스럽게 가이드하고, 프로젝트를 효율적으로 관리 할 수 있는 기능들을 제공한다.

따라서, 현존하는 도구들과는 달리 실시간 채팅과 이미지 공유 및 스케치 등 차별성을 띤 협업 도구로서 쉽게 사용할 수 있을 것이다.

이를 시작으로, 보다 다양한 기능을 제공할 수 있도록 보완하고, 앞으로는 Spring 기반 애플리케이션으로 전환하여 보다 성능과 서비스 차원을 높일 것이다. 차후에는 현존하는 협업 도구들과 빗댈 수 있고, 회사에서도 사용할 수 있는 웹 서비스로 발전시켜 나갈 계획이다.

참 고 문 헌

[1] 박재성, 로드 북 “자바 웹 프로그래밍 Next Step : 하나씩 벗겨가는 양파껍질 학습법”

[2] 박재성, 한빛미디어 “자바 세상의 빌드를 이끄는 메이븐”