

REDIS를 활용한 학급경영 웹 애플리케이션 프레임워크의 설계

박준석 · 전석주

서울교육대학교 컴퓨터교육과

요 약

전통적인 학급경영시스템은 교사의 독단적인 의사에 의해 운영되는 경우가 일반적이다. 그러나 오늘날에는 자질을 갖춘 민주시민을 양성하기 위해 학생과 교사 모두 함께 자치 규범을 만들어 스스로 학급을 운영하도록 요구하고 있다. 그래서 기존의 학급 경영시스템은 이러한 다양성을 가진 오늘날의 학급에 더 이상 적합하지 않다. 본 연구에서는 REDIS를 활용한 학급경영 웹 애플리케이션 프레임워크를 설계한다. REDIS는 일종의 다양한 키 값을 저장하는 데이터 저장소인데 보통 공유 메모리를 가진 웹 애플리케이션을 개발하기 위한 솔루션을 제공한다. 개발될 웹 애플리케이션 프레임워크는 사용의 편의성과 접근성을 최대화하도록 설계되어진다. 이러한 학급경영시스템은 프레임워크에서 기본적으로 제공되는 다양한 템플릿 함수들을 사용하여 효과적으로 시스템의 확장성이 향상될 수 있다.

키워드 : REDIS, 웹 애플리케이션, 프레임워크, 학급경영, 데이터베이스

Design of Web Application Framework Using REDIS for Class Management

Joonseok Park · Seokju Chun

Dept. of Computer Education, Seoul National University of Education

ABSTRACT

It is a general tendency that a traditional class management system is operated by teachers' arbitrary decision. However, at the present day, it is needed that both students and teachers make an autonomy standard manual and manage the class by themselves in order to cultivate the qualified democratic citizen. Therefore existing class management systems do not meet present classes with diversity. In this paper, we design a new web application framework using REDIS(Remote Dictionary System) for class management. REDIS is a kind of data repository to store various key values and also generally provides a solution of developing web applications with shared

이 논문은 박준석의 2014년 석사학위 논문의 일부를 발췌 및 수정·보완한 논문임.

교신저자 : 전석주(서울교육대학교 컴퓨터교육과)

논문투고 : 2014-03-04

논문심사 : 2014-03-05

심사완료 : 2014-09-09

memories. We designed a web application framework to maximize both convenience of use and accessibility. The scalability of the class management system can be effectively enhanced using diverse template functions which are basically provided by the framework.

Keywords : REDIS, Web Application, Framework, Class Management, Database

1. 서론

우리나라는 21세기 정보화 사회로의 변화에 따라 국가 경쟁력의 바탕을 정보화에 두고 범국가적인 차원에서 강력하게 추진하고 있다. 본 논문에서는 이러한 정보화 시대에 맞추어 교육현장에서 보다 효과적으로 활용 가능한 소프트웨어 개발에 대해서 설계하고자 한다. 교육적으로 학생과 교사에게 유의미하고 정보화 인프라를 최대한 활용할 수 있는 방법을 고민해보니 학급경영에 도움을 주는 프로그램을 개발하여 활용하면 큰 효과를 거둘 수 있다고 생각했기 때문이다[5].

전통적인 학급경영이 교사가 주도적으로 이루어내는 것이라면, 지금의 학급경영은 일방적인 관리가 아니라 학생들과 함께 소통하며 학급 운영에 관련한 다양한 자치 규칙도 스스로 만들어가는 민주시민으로서의 자질을 갖추기 위한 좋은 무대가 될 수 있기 때문이다. 거기에 맞추어 학생과 교사, 그리고 학부모의 의사소통을 위한 네트워크를 더하게 된다면 보다 효과적으로 활용할 수 있다. 따라서 정보화 시대의 흐름에 맞춰 새로운 접근 방식으로 프로그램을 구상하여 보다 접근하기 쉽고 활용 영역이 많은 학급경영 프로그램을 활용할 수 있게 된다면 학급의 교사들에게 많은 도움을 줄 수 있다[1].

본 연구에서는 다양한 스마트 디바이스를 통하여 자유로운 접근성과 학급경영의 결과물의 활용도가 높은 학급 경영 프로그램을 설계하기 위해 다음과 같은 방향을 가지고 설계한다.

첫째, 웹기반으로 제작하여 운영체제와 상관없이 언제든지 사용할 수 있도록 설계한다.

둘째, 다양한 API를 제공하여 프로그램의 확장성 및 유연성을 가질 수 있도록 설계한다.

셋째, 프로그램의 기능적인 부분과 디자인 부분을 분리하여 설계한다.

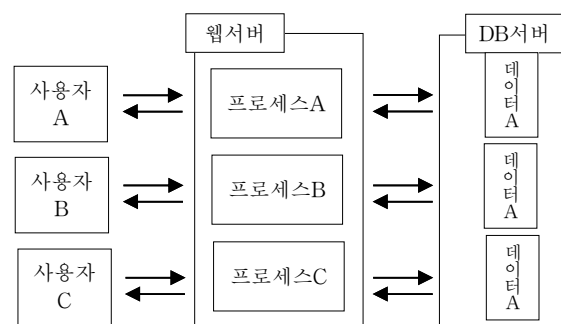
따라서 본 연구는 학급경영이 굉장히 넓은 범주를 통

합하고 있는 영역이므로 특정된 학급경영 프로그램을 구축하는 것보다 프로그램을 확장하여 개발할 수 있는 프레임워크를 설계하는 데 비중을 두고 있다.

2. 이론적 배경

2.1 REDIS의 개요

REDIS(Remote Dictionary System)는 다양한 형태의 키 값(key-value)을 저장할 수 있는 일종의 데이터 저장소 역할을 한다. 특히 REDIS는 데이터를 디스크가 아니라 시스템 메모리에 직접 적재하고 불러오는 방식으로 작동하고, 더하여 일반적인 데이터베이스 서버와 마찬가지로 네트워크 통신으로 데이터 입출력 작업을 할 수 있기 때문에 공유 메모리를 활용한 애플리케이션을 개발하기 위한 솔루션으로 활용되고 있으며, 특히나 프로세스 간 메모리 공유가 불가능한 웹 애플리케이션 영역에서 그 효과가 탁월하다.

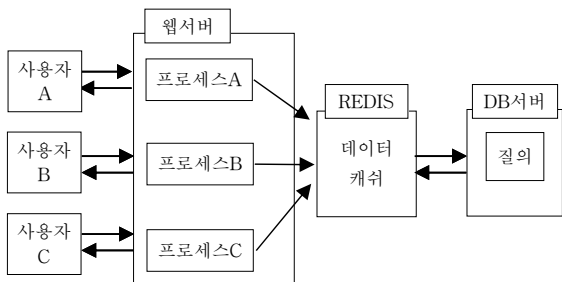


(Fig. 1) Web service data handling process

데이터베이스를 직접 핸들링하여 데이터를 저장하거나 가져오는 전통적인 방식의 가장 큰 문제점은 개별

사용자 간의 요청을 처리하는 프로세스 간의 메모리를 공유할 수 없으므로 요청이 있을 때마다 데이터베이스 서버를 통해 디스크로부터 읽기 때문에 속도가 느린 디스크로 잦은 입출력은 시스템에 많은 부하를 발생하게 된다(그림 1).

반면 REDIS와 같은 프로그램을 웹서버와 데이터베이스 사이에 위치하여 데이터베이스로부터 가져온 데이터를 시스템 메모리에 적재해두었다가 같은 데이터에 대한 요청이 발생하면 데이터베이스를 통하지 않고 REDIS를 통하여 시스템 메모리에서 데이터를 빠르게 가져올 수 있게 된다. 그러므로 중복된 디스크 입출력 횟수를 줄일 수 있어서 웹 애플리케이션 개발에 많이 활용되고 있다(그림 2).



(Fig. 2) Web service data handling process using REDIS

REDIS와 유사한 시스템으로는 아마존, 트위터, 위키 피디아와 같이 대량의 데이터를 다루는 사이트에서 사용하고 있는 memcached가 있으며, REDIS는 memcached를 개량하여 쉽고 효과적으로 사용할 수 있도록 개발된 공개소프트웨어이다. memcached가 저장할 수 있는 데이터 크기가 제한적이고 지원하는 데이터 형식이 문자열밖에 없다는 단점이 있는 반면, REDIS는 다양한 데이터 형식을 지원하고 저장할 수 있는 데이터의 크기도 크다는 장점이 있다[2][6][12].

2.2 프레임워크

소프트웨어 설계 분야에서 디자인 패턴으로 유명한 랄프 존슨(Ralph Johnson) 교수는 프레임워크를 소프트웨어의 구체적인 부분에 해당하는 설계와 구현을 재사용이 가능하도록 일련의 협업화된 형태로 클래스들을

제공하는 것이라고 정의하였다[3][4]. 프레임워크는 일반적인 라이브러리와 다르게 애플리케이션의 틀과 구조를 결정할 뿐만 아니라, 그 위에서 개발되는 개발자의 코드를 제어한다. 라이브러리가 단순히 특정 작업을 처리하는 메서드를 의미한다면 프레임워크는 다양한 작업을 위한 메서드 집합을 제공하고 이를 활용하여 애플리케이션을 구성하는 절차적인 부분까지도 포함된 영역이라고 할 수 있다.

대표적인 예가 스프링 프레임워크(Spring Framework)가 있다[14]. 이 프레임워크는 자바 플랫폼을 위한 오픈소스 애플리케이션 프레임워크로 복잡한 엔터프라이즈 애플리케이션을 보다 쉽게 개발하기 위해서 만들어졌다. 이 프레임워크는 특히 동적인 웹 사이트를 개발하기 위해 여러 가지 서비스를 제공하고 있고 개발과 유지 보수가 쉬우며 프레임워크의 신뢰가 높아 현재 우리나라 전자정부 표준 프레임워크의 기반 기술로 쓰이고 있다.

3. 학급경영 웹 애플리케이션 프레임워크 설계

3.1 학급경영 웹 애플리케이션 프레임워크의 설계 방향

효과적인 웹 애플리케이션의 설계를 위한 프로그램의 요구사항과 방향을 제시하면 다음과 같다.

가. 웹기반으로 제작하여 클라이언트의 종류와 상관없이 언제든지 접속할 수 있도록 설계한다.

인터넷을 통해 PC와 스마트폰을 통하여 시간과 장소에 관계없이 접속하여 학급경영 프로그램을 이용할 수 있어야 한다. 이것을 위해서 프로그램은 데이터를 직접 처리하는 서버 부분과 처리된 데이터를 사용자에게 보여주는 클라이언트 부분으로 나누어 개발하는 Client-Server 모델이 필요하다.

Client-Server 모델보다 효과적이고 개발이 용이한 방법은 웹 브라우저를 활용한 방법이다. Apache나 Internet Information Server[9], Nginx[10]와 같은 웹서버 소프트웨어에서 동적인 HTML을 작성하기 위한 프로그래밍 모듈(ASP.net, JSP 또는 PHP)을 탑재하여 서

버 프로그램을 준비하고, 클라이언트 프로그램은 기존의 웹 브라우저를 그대로 활용한다[11]. 이 경우 서버 프로그램만을 작성하면 되기 때문에 개인이 개발하고 적용하기에 용이한 장점이 있지만, 다양한 웹브라우저마다 지원되는 HTML 규격이 다르기 때문에 기기마다 출력되는 결과물에 차이가 발생할 수 있다.

나. 다양한 API를 제공하여 프로그램의 확장성 및 유연성을 가질 수 있도록 설계한다.

학급경영 방법은 교사의 학급 운영 철학이 다양한 것처럼 그 종류가 매우 다양할 수밖에 없다. 같은 학년이라고 해도 학교의 환경이나 문화에 따라 저마다 특징되는 점들이 있으며, 이 특징되는 점들로 인하여 학급경영의 방침은 늘 변할 수 있기 때문이다. 그러므로 학급경영 프로그램도 획일적인 기능을 제공하기보다는 사용자가 원하는 만큼을 수정할 수 있도록 그 틀을 제공하는 것이 훨씬 효과적이라는 생각한다. 공통으로 운영이 가능한 애플리케이션의 기능을 제외하고 언제나 추가 기능을 확장하고 생성할 수 있도록 다양한 형태의 API를 제공할 수 있어야 할 것이다. 약간의 프로그래밍 지식이 있다면 누구든지 쉽게 사용할 수 있도록 인터페이스를 최대한 단순하게 설계한다면 애플리케이션의 유지, 보수 및 개발을 쉽게 할 수 있다.

다. 프로그램의 기능적인 부분과 디자인 부분을 분리하여 설계한다.

애플리케이션은 내부 처리 부분과 출력 부분으로 나누어 볼 수 있다. 내부 처리 부분은 각종 데이터를 저장하고 불러오는 기능을 바탕으로 다양한 처리를 담당하는 부분이고, 출력 부분은 내부 처리 부분에서 처리된 결과를 사용자에게 어떻게 보여줄 것인지를 담당한다. 전통적인 프로그램 개발에서는 내부 처리 부분과 출력 부분을 특별히 구분하지 않고 처리된 결과를 바로바로 보여주는 방식이었다. 이러한 전통적인 설계 방법의 장점은 특별히 처리와 출력을 구분하지 않았기 때문에 프로그램 구현하기가 쉽다는 장점이 있지만, 그렇기 때문에 출력 방법을 수정하거나 다양화 할 수 없다는 단점을 가지고 있다. 이러한 템플릿 기반의 프로그램은 쉽게 찾아볼 수 있다. 당장 PC에서 사용하고 있는 운영체

제인 윈도우만 하더라도 테마라는 기능으로 이러한 기능을 구현하고 있다. 내용에는 변화가 없지만 사용자에게 보여주는 윈도우의 색깔이나 그림 등은 테마에 따라 다르고 심지어는 사용자가 테마를 직접 수정하여 사용할 수 있다.

라. 일반 PC에서 운영할 수 있도록 프레임워크를 경량화하여 설계한다.

일반적인 웹 애플리케이션이 네이티브 애플리케이션에 비해 부족한 부분은 스크립트 언어로 동작하기 때문에 기본적인 성능이 낮다는 문제와 스크립트가 종료하는 순간에 점유하고 있던 메모리도 모두 해제가 된다는 점이다. 즉, 네이티브 애플리케이션으로 제작된 프로그램은 데이터베이스에서 쿼리한 여러 가지 데이터를 메모리에 유지할 수 있어서 한 번의 쿼리로 원하는 결과를 반복하여 빠르게 제공할 수 있지만, 웹 애플리케이션은 페이지 로드가 끝나는 시점에서 인터프리터 인스턴스가 종료되기 때문에 데이터베이스에 쿼리하여 메모리에 저장해 둔 데이터가 사라지게 된다는 것이다. 그러므로 웹 애플리케이션에서는 같은 결과를 반복적으로 요청하게 되더라도 매번 데이터베이스에서 데이터를 가져와야 하는 문제가 발생한다[13].

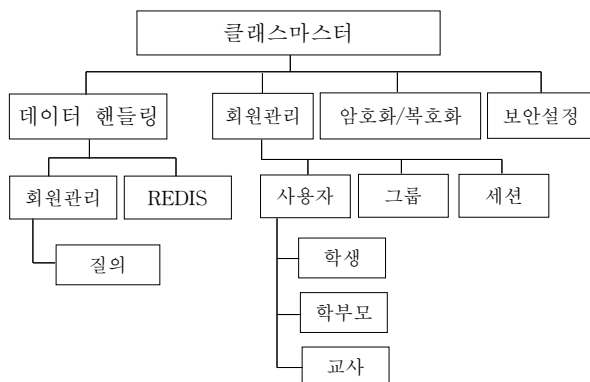
물론 데이터베이스에 저장된 데이터의 수가 많지 않을 경우에는 그다지 큰 차이가 없겠지만 축적된 데이터의 양이 증가하여 수십만 건의 데이터를 처리해야 하는 경우에는 그 차이가 어마어마하게 발생하게 된다. 이 문제를 해결하기 위해 고안된 방법이 Distributed Memory Object Caching System이다[7]. 이것은 데이터를 제어할 때 물리적으로 속도가 느린 디스크가 아니라 디스크에 비해 속도가 월등히 빠른 메모리에 적재하고 디스크는 보조적인 저장매체로 활용하는 것이다. 이에 해당하는 영역이 1)PHP 4.0.3부터 지원하는 Shared Memory, 2)유튜브, 아마존, 트위터에서 사용되고 있는 Memcached, 3)아직은 많이 알려지지 않았지만 Memcached보다 더 많은 기능을 제공하는 REDIS 등이 있다[12]. 이러한 Memory Object Caching System을 사용하면 웹 애플리케이션의 단점을 극복할 수 있을 것이다.

3.2 학급경영 웹 애플리케이션 프레임워크의 기본 역할

학급경영은 교사들의 다양한 철학과 깊은 관련성이 있기 때문에 정형화된 형식은 없다. 그러므로 각각의 교사들이 필요에 따라 학급경영 프로그램을 확장할 수 있어야 한다는 것은 이미 언급한 바가 있다. 그러므로 학급경영 프로그램의 기본 틀(여기에서는 프레임워크라고 칭한다)의 범위를 정하고 이에 해당하는 다양한 API를 구성해야 한다.

학급경영 애플리케이션 프레임워크의 기본 역할로 사용자(또는 개발자)에게 다양한 종류의 API를 제공하기 위한 프레임워크의 가장 상위 클래스인 클래스 마스터, 회원들이 활동에 따라 생성되는 다양한 결과물인 그림이나 동영상 및 텍스트 형태의 데이터를 관리하는 데이터 핸들링 클래스, 학급경영 애플리케이션의 주요한 기능으로 학생의 교육활동을 지원하고 학생 및 학부모, 교사 간의 커뮤니티를 형성해 주기 위한 회원 관리 클래스 및 학급경영을 통해 생성되는 데이터의 개인 정보를 보호하기 위한 암호화 등을 들 수 있다.

이러한 학급경영 애플리케이션 프레임워크의 기본 역할을 기준으로 애플리케이션 프레임워크 기본 구조를 그림으로 나타내면 (그림 3)과 같다.



(Fig. 3) Framework structure of web applications

3.2.1 프레임워크 API

학급경영 프로그램은 사용자 환경의 다양성을 지향

하고 있다. 그러므로 프레임워크를 통해 프로그램의 기반을 사용자에게 제공하여 개개인의 사용자의 목적에 맞는 프로그램의 수정이나, 애드온 프로그램 제작을 유도해야 한다. 그러므로 사용자에게 프레임워크에서 제공하는 다양한 기능을 손쉽게 이용할 수 있도록 단순하고 구조적인 API를 제공해야 한다. 사용자는 API를 제공하는 객체를 핸들링하는 것으로 애플리케이션에 포함된 하위 클래스의 모든 기능을 사용하여 원하는 애드온 프로그램을 제작할 수 있다.

3.2.2 클래스 마스터(ClassMaster)

클래스 마스터는 프레임워크의 가장 상위 클래스이다. 하부에는 프레임워크의 여러 가지 기능을 구현하고 있는 객체를 가지고 있으며, 이 객체의 주요 역할은 사용자(또는 개발자)에게 다양한 종류의 API를 제공한다.

3.2.3 데이터 핸들링

학급경영 애플리케이션을 운영하게 되면 앞으로 회원들이 활동에 따라 다양한 결과물이 생성된다. 이것은 그림이나 동영상이나 될 수 있고, 단순 텍스트 형태로도 생성될 수 있으며, 이러한 모든 데이터를 저장하고 필요시 적절한 데이터를 가져올 수 있어야 한다. 특히, 시스템 처리 능력을 향상하기 위해 적용한 REDIS를 활용할 경우 내부적으로 처리되는 복잡한 절차를 사용자가 직접 처리하지 않더라도 간단한 메서드를 사용하여 처리할 수 있도록 단순한 API를 제공할 수 있다.

3.2.4 회원 관리 및 암호화

학급경영 애플리케이션의 주요한 기능은 학생의 교육활동을 지원하고 학생 및 학부모, 교사 간의 커뮤니티를 형성해주는 것이다. 그러므로 회원관리 기능은 필수라고 할 수 있겠다. 회원은 크게 교사, 학생, 학부모, 일반에 해당하는 4개의 종류로 분류할 수 있고, 각각의 회원은 역할에 따라 적절한 권한이 부여되어 프로그램이 제공하는 기능의 차이가 있어야 한다.

뿐만 아니라 학급경영을 통해 생성되는 데이터는 개인정보와 밀접한 관련이 있는 내용이므로 암호화 기능

이 필요하다. 기본적으로 데이터베이스에 저장하는 데이터의 Primary Key는 SHA-1 해시 코드를 사용하여 데이터의 직접적인 외부 노출을 최소화한다.

3.3 학급경영 애플리케이션 프레임워크의 설계

3.3.1 클래스마스터(ClassMaster)

```
public class SingletonDemo {
    private static SingletonDemo instance = null;

    private SingletonDemo() { ... }
    public static synchronized SingletonDemo
    getInstance() {
        if (instance == null) {
            instance = new SingletonDemo();
        }
        return instance;
    }
}
```

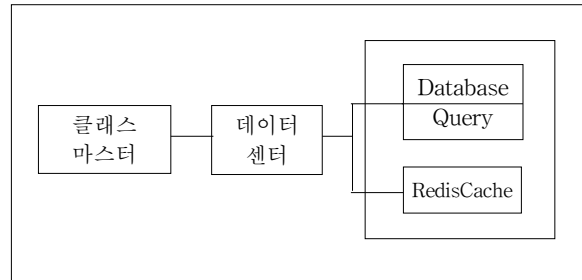
(Fig. 4) Example of Singleton pattern

클래스마스터는 프레임워크의 가장 상위 클래스이다. 하부에는 프레임워크의 여러 가지 기능을 구현하고 있는 객체를 가지고 있으며, 이 객체의 주요 역할은 사용자(또는 개발자)에게 다양한 종류의 API를 제공하기 위함이다. 그리고 프로그램의 최초 진입점이 포함된 클래스로서 이 클래스의 인스턴스를 생성하는 것이 프로그램을 실행하는 것과 같다고 간주할 수 있다. 그러므로 한 컴퓨터에는 하나의 인스턴스만 허용해야 하며, 프로그램의 모든 영역에서 참조할 수 있어야 하므로 Singleton Pattern으로 설계할 필요가 있다(그림 4).

3.3.2 데이터센터(DataCenter)

데이터센터는 학급경영 웹 애플리케이션을 운영하면서 생성되는 여러 가지 데이터를 핸들링하는 클래스이며 내부에 Database 객체와 RedisCache 객체를 가지고 있어서 데이터베이스와 REDIS를 직접적으로 컨트롤하는 역할을 한다(그림 5). 사용자는 데이터를 저장하거나

가져올 때에는 데이터센터 인스턴스를 통하여 처리하기 때문에 사용자는 Database 객체와 RedisCache 객체의 처리 내용을 숙지해야 할 필요가 없다.



(Fig. 5) Data center model

웹 애플리케이션의 최대 단점인 PHP 프로세스 간의 메모리 공유가 불가능하기 때문에 매번 접속할 때마다 디스크에서 데이터를 읽어와 처리를 한다는 것이다. 데이터베이스에 저장된 데이터의 양이 적을 경우에는 문제가 발생하지 않지만, 데이터의 양이 많을 경우 데이터를 불러오는 데 많은 시간이 소요되어 이용이 불편한 문제가 발생한다. 이런 문제점을 해결하기 위해서 웹 서비스 업체에서는 디스크의 속도를 높이기 위해 고가의 레이드 장비를 활용하기도 하지만 개인이 운영하기에는 경제적인 부담이 너무 크다.

REDIS는 이러한 문제를 해결하기 위해 나온 일종의 데이터 캐싱 서버이다. REDIS를 통해 저장되는 모든 데이터는 디스크에 기록되는 것이 아니라 서버의 메모리에 적재가 된다. 이것은 PHP 프로세스 간의 메모리 공유가 불가능했던 점을 해결해준다. 공유하려는 데이터를 REDIS에 기록해두고, 필요할 때마다 REDIS를 통해 가져오면 되기 때문이다. 하지만 REDIS는 메모리에 저장되기 때문에 급작스럽게 서버가 다운되었을 경우 메모리의 특징상 데이터 손실이 발생하게 된다. 물론 REDIS 자체적으로도 정기적으로 데이터를 디스크에 백업을 하지만 분명 데이터 손실을 가져올 수 있는 여지가 발생한다. 그러므로 데이터 무결성을 확보하기 위해 데이터가 생산될 경우 Database 객체를 통하여 디스크에 저장하고, RedisCache 객체는 REDIS 서버를 통해 데이터를 불러오는 역할을 하게 하여 처리한다.

기본적인 설계 방향은 애플리케이션에서 데이터가 필요하면 DataCenter는 RedisCache를 통해 데이터를

불러오고, 저장해야 할 데이터가 발생했을 때에는 DataCenter가 Database 객체를 이용하여 저장한다. 이것은 데이터의 무결성을 고려한 방법으로 불시에 서버에 문제가 발생하더라도 데이터의 손실을 최소화할 수 있다.

기본적으로 상위 클래스인 ClassMaster에서는 Database나 RedisCache를 직접 제어하지 않더라도 DataCenter를 통하여 데이터를 요청하면 메모리에 상주된 데이터를 우선적으로 보내주고, 메모리에 상주된 데이터가 존재하지 않는다면 데이터베이스에 쿼리하여 메모리에 상주시킨 후 보내주게 된다.

<표 1>은 데이터 센터의 메소드와 프로퍼티 목록을 나타낸다.

<Table 1> Datacenter Method/Property Lists

구분	제한자	이름	설명
메서드	public	DataCenter()	생성자
	private	init()	초기화 작업
	private	initMemberList()	사용자 리스트 생성
	public	resetMemberList()	사용자 리스트 캐쉬 초기화
	public	getMemberList()	전체 사용자 리스트 가져오기
	public	getMember()	사용자 정보 가져오기
	public	insertMember()	새 사용자 정보 저장
	public	deleteMember()	사용자 정보 삭제
	public	updateMember()	사용자 정보 갱신
	public	getUserDefine()	사용자 지정 변수 불러오기
public	setUserDefine()	사용자 지정 변수 저장	
프로퍼티	private	database	Database 인스턴스
	private	query	Query 인스턴스
	private	redis	RedisCache 인스턴스

3.3.3 회원관리(Membership)

학급경영 웹 애플리케이션을 이용하는 회원들을 관리하는 클래스이다. 이 클래스는 두 가지 기능을 하게 되는데 회원을 관리하는 기능과 세션을 통하여 로그인한 회원을 관리하는 역할을 한다. 그리고 회원의 여러 가지 정보를 변경하거나 저장하는 기능을 한다. 모든 회원은 User 클래스로 생성한다. 학부모나 학생, 교사의

경우는 추가 클래스인 Student, Parent, Teacher에 User 클래스의 사용자 고유 식별 번호로 추가 정보를 저장한다. 그러므로 User 클래스와 나머지 클래스는 1:N의 관계를 갖는다. 즉, User 객체는 Teacher이면서 Parent가 될 수 있으며, 하나의 User 객체가 복수의 Student 객체를 가질 수 있게 된다. 뿐만 아니라 클라이언트 PC에 저장된 세션 정보를 가지고 로그인 상태를 처리하는 시작점을 가지고 있다. 그래서 이 클래스 객체가 생성될 때 로그인 사용자를 처리할 수 있게 된다.

회원의 권한을 관리하기 위해서 모든 User 객체는 Group 객체를 가지게 된다. 그리고 사용자 고유 식별 번호 또는 그룹 고유 식별 번호를 기준으로 각 회원의 권한을 처리하게 된다. 사용자는 크게 4가지로 분류할 수 있다. User, Teacher, Student, Parent가 그것이다. 하지만 4가지 분류의 회원이 1:1 매칭이 되는 것은 아니기 때문에 클래스 상속관계로 표현하는 것에는 문제가 발생할 수 있다. 그래서 회원 정보는 기본적으로 종류에 상관없이 모두 User 클래스로 생성하고 추가적으로 필요한 부분에서 Teacher나 Student, Parent와 같은 추가적인 정보를 갖게 되는 형식으로 설계하였다.

예를 들면, 홍길동이라는 학생이 있는데 이 학생은 2012년도에 4학년 1반, 2013년도에는 5학년 2반이라고 가정해보자. 먼저 모든 사용자는 User 객체를 가져야 하므로 홍길동이라는 User 객체가 있을 것이다. 그리고 학생은 2012년도와 2013년도 두 개의 다른 종류의 학생이기 때문에 Student 객체를 2개 가질 수 있을 것이다. 이러한 방식으로 User 객체는 다른 객체와 1:N의 관계를 갖는다.

3.3.4 콘텐츠 보안(Permission)

여러 사용자로부터 애플리케이션이 제공하는 여러 콘텐츠 접근을 제한하는 역할을 하는 클래스이다. 기본적으로는 리눅스 시스템의 그것과 비슷하게 설계했다. 프로그램을 운영하면서 생성되는 모든 데이터에는 퍼미션과 소유자, 소유그룹 정보를 가지게 된다. 그리고 Membership 객체에서 가져온 로그인 사용자의 객체를 변수로 권한 비교를 한다. 먼저 사용자 고유 식별 번호를 기준으로 퍼미션을 체크하고, 해당사항이 없을 경우 그룹의 고유 식별 번호로 체크를 한다. 퍼미션은 3가지

대상에 대해 비교를 하는데, 첫 번째가 소유자 비교, 두 번째가 그룹 비교, 마지막으로 익명비교를 한다. 각각의 대상은 또 다시 3가지 부분에 대한 권한을 가지고 있는데, 읽기권한과 쓰기권한, 마지막으로 실행권한을 갖는다. 기본적인 작동 방법은 리눅스의 그것과 같은 방법으로 작동한다. 세부 권한에는 읽기, 쓰기, 실행이라는 3가지 종류가 있고, 생성자, 그룹, 기타 사용자에게 해당하는 3가지 종류를 대상으로 권한 부여를 하게 된다. 그리고 Permission 클래스는 애플리케이션에서 생성되는 모든 데이터에 부여할 수 있기 때문에 섬세한 권한 설정을 할 수 있다.

Permission 클래스에는 소유자에 해당하는 소유주 고유 번호, 소유 그룹 고유번호를 기본으로, 소유자에 대한 권한(owner), 그룹에 대한 권한(group), 기타 사용자에게 대한 권한(others)에 해당하는 5가지 멤버 변수를 가지고 있다. 그리고 세 가지 권한(읽기, 쓰기, 실행)에 대한 처리루틴을 가지고 있다. 함수에 인자로 전달 받은 값은 권한이 있는지 체크할 사용자의 User 객체를 전달받는다.

4. 결론 및 향후 연구 과제

본 논문에서는 정보화 시대에 맞추어 교육현장에서 보다 효과적으로 활용 가능한 학급경영 웹 애플리케이션 프레임워크에 대해서 설계를 해보았다. 교육적으로 학생과 교사에게 유의미하고 정보화 인프라를 최대한 활용할 수 있는 방법을 고민해보니 학급경영에 도움을 주는 프로그램을 개발하여 활용하면 큰 효과를 거둘 수 있다고 생각했기 때문이다.

전통적인 학급경영이 교사에 의해 일방적으로 이루어졌다면 오늘날의 학급경영은 교사와 학생이 함께 다양한 자치규칙을 만들고 더 나아가 교사와 학생뿐만 아니라 학부모들과도 자유로운 의사소통을 위한 네트워크가 필요하다. 이러한 네트워킹을 지원하도록 학급경영 시스템을 만들어서 활용하면 효과적이고 민주적인 학급경영이 될 것이다. 물론 개인이 모든 것을 개발하기에 시간과 기술이 부족하여 쉽지 않은 연구라고 생각한다. 뿐만 아니라 학교의 상황과 학급의 환경에 따라 가지각색으로 나타나는 학급경영에 도움을 주는 프로그램이라

고 하면 너무나도 커다란 일이 아닐 수 없다.

앞으로 학급경영에 직접적으로 필요한 기능을 좀 더 살펴보고 프레임워크를 보강할 필요가 있다. 특히 학급경영에 직접적으로 적용할 수 있는 성적과 관련한 부분이나 학생들의 자치활동을 지원해주는 기능들과 같은 영역이다. 학급경영이 포괄하는 영역이 광범위하기 때문에 앞으로도 꾸준히 연구되고 개발되어져야 한다. 이를 위해서 프레임워크가 어느 정도 틀이 갖춰지게 되면 moodle[8]과 같이 오픈 소스 프로젝트로 전환하여 학급경영 웹 애플리케이션에 관심이 있는 사람들에게 배포한 후, 맞춤형 학급경영 프로그램을 위해 애드온 제작을 위한 지식 공유 사이트를 운영하여 보다 많은 활성화가 이루어질 수 있도록 다양한 방법을 강구할 계획이다.

참고문헌

- [1] Chul Kim et al.(2002). A Design and Implementation of Class Management Support System for Effective Classroom Management. *Journal of The Korean Association of information Education*, 6(1), 68-78.
- [2] Patrick Galbraith(2009). *Developing Web Applications with Apache, MySQL, memcached, and Perl(1st Ed.)*. Wrox, 2009.
- [3] Ralph Johnson(1996). Evolving Frameworks: A Pattern Language for Developing Object-Oriented Frameworks. *In Proceedings of the Third Conference on Pattern Languages and Programming*, 3, 1996.
- [4] Ralph E. Johnson(1997). Frameworks=(components +patterns). *Communication of ACM*, 40(10), 39-42.
- [5] Burden, P. R.(2003). *Classroom management: Creating a successful learning community(2nd ed.)*. Hoboken, NJ: John Wiley, 2003.
- [6] Leon Shklar(2009). *Web Application Architecture: Principles, Protocols and Practices(2nd ed.)*. Wiley, April 27, 2009.
- [7] memcached—a distributed memory object caching system, <http://memcached.org>
- [8] Moodle—Open-source learning platform. <https://moodle.org>

- [9] The Official Microsoft IIS Site, <http://www.iis.net>
- [10] Nginx Community, <http://wiki.nginx.org>
- [11] PHP Reference Guidebook. <http://www.php.net/manual/en>
- [12] A PHP extension for Redis. <https://github.com/nicolasff/phpredis>
- [13] MySQL Reference Manual. <http://dev.mysql.com/doc/refman/5.7/en>
- [14] Spring Framework. <http://projects.spring.io/spring-framework>

저자소개



박 준 석

2004 광주교육대학교 전산교육과
(현 컴퓨터교육과) 졸업
2014 서울교육대학교 교육대학원
컴퓨터교육학과 졸업
2004~현재 경기도 동두천·양주
광송초등학교 교사
관심분야: 웹프로그램 개발, WBI,
학습자료 개발
e-mail: nostalg00@nate.com



전 석 주

2002 한국과학기술원 전산학박사
(공학박사)
2004~현재 서울교육대학교 컴퓨
터교육과 교수
관심분야: 컴퓨터교육, 프로그래
밍교육, 데이터마이닝
e-mail: chunsj@snue.ac.kr

