

# 설계 명세서



## Homework Alert System(HAS)



## Software Engineering 10 조

최가람, 라종휘, 김종혁, 배수민, 이기웅

## 내용

1. Preface.....	5
1.1. Document Structure .....	5
1.2 Readership.....	6
1.3. Version of the Document .....	6
2. Introduction.....	8
2.1. 배경 및 필요성 .....	8
2.2. Homework Alert System (HAS).....	9
2.3. 기대 효과.....	10
2.4. Constraints .....	11
3. Glossary.....	12
4. System Architecture.....	13
4.1. System Architecture Overview .....	13
4.2. Subsystem1. Prof. Log-in Approval system.....	14
4.3. Subsystem2. Information Administrator.....	15
4.4. Subsystem3. Set H/W Alarm.....	16
4.5. Subsystem4. Send H/W Alarm.....	16
4.6. 전체 네트워크 구조.....	17
4.7. 데이터베이스 구조 .....	18
5. User Activity .....	19
5.1. Class Diagram .....	19
5.2. Scenario.....	20
5.3. Activity Models.....	21
6. Prof. Log-in Approval Subsystem.....	25
6.1. Class Diagram .....	25

6.2. Sequence Diagram .....	25
6.3. State Diagram .....	26
7. Information Administrator Subsystem .....	27
7.1. Class Diagram .....	27
7.2. Sequence Diagram .....	27
7.3. State Diagram .....	28
8. Set H/W Alarm Subsystem.....	29
8.1. Class Diagram .....	29
8.2. Sequence Diagram .....	29
8.3. State Diagram .....	30
9. Send H/W Alarm Subsystem .....	31
9.1. Class Diagram .....	31
9.2. Sequence Diagram .....	31
9.3. State Diagram .....	32
10. Test.....	33
10.1. Test Image .....	33
11. Development Environment.....	34
11.1. Operating System.....	34
11.2. 코딩 환경.....	34
11.3. Programming Language .....	35
11.4. Application Framework .....	36
12. Index.....	37
12.1 Figure Index .....	37
13.2 Diagram Index.....	38
13.3 Table Index .....	39

13. Appendices.....	40
References.....	41

# 1. Preface

---

Preface에서는 본 문서의 대한 개략적인 부분과 각 부분의 역할에 대하여 설명한다.  
독자가 누구인지를 밝히고, 각 챕터의 역할을 소개한다.

## 1.1. Document Structure

이 문서에서는 총 10 개의 부분으로 구성되어 있다. 각 부분에 대한 설명은 다음과 같다.

### A. Preface

-Preface 는 문서의 구조 및 각 부분의 의미, 대상 독자 등을 간략히 설명한다.

### B. Introduction

-User 관점에서의 시스템을 간단히 서술한다.

### C. Glossary

-문서에 나오는 생소할 수 있는 용어들의 간략한 설명이 붙어있는 part 이다.

### D. System Architecture

-System Architecture 에서는 전체 시스템에 대한 개요를 자세히 서술한다. 시스템의 구조를 diagram 을 통해 설명하고 각 서브시스템들의 기능과 구조를 설명한다.

### E. User Activity

-HAS(Homework Alert System)에서 user 가 사용할 수 있는 인터페이스에 대한 기능을 Class Diagram, Sequence Diagram, State Diagram 을 통해 서술한다.

### F. Prof. Log-in Approval Subsystem

-첫번째 Sub-system 인 Prof. Log-in Approval Subsystem 에 대해 서술한다.

### G. Information Administrator Subsystem

-두번째 Sub-system 인 Information Administrator Subsystem 에 대해 서술한다.

### H. Set H/W Alarm Subsystem

-세번째 Sub-system 인 Set H/W Alarm Subsystem 에 대해 서술한다.

### I. Send H/W Alarm Subsystem

-네번째 Sub-system 인 Send H/W Alarm Subsystem 에 대해 서술한다.

### J. Database Design

-시스템 구동 중 시스템의 정보들을 저장하고 관리하는 데이터베이스의 기능과 구현에 대해 자세히 설명한다.

#### K. Test

-System 의 기능과 성능을 시험하기 위한 테스트 모듈에 대해 설명한다.

#### L. Development Environment

-시스템개발중에 사용한 개발환경에 대해 설명한다.

#### M. Index

## 1.2 Readership

System End-User, Client Engineers, System Architects, Software Developers

설계 명세서의 잠재적 독자 대상은 위의 4 분류와 같다. 본 시스템을 구매하고 사용하는 user, 그리고 각 분야에서 개발에 힘쓰는 developer 들이 그 대상자이다.

## 1.3. Version of the Document

### A. A. Version format

버전 번호는 major.minor[.maintenance]로 구성되며, 문서의 버전은 0.1 부터 시작한다.

### B. Version management policy

요구사항 명세서를 수정할 때 마다 버전을 업데이트한다. 다만 변경 간의 간격이 1 시간 이내 일 때 에는 버전 번호를 업데이트 하지 않고 하나의 업데이트로 간주한다. 이미 완성된 파트를 변경할 때에는 minor number 를 변경하며, 새로운 부분을 추가하거나 문서의 구성이 예전에 비해 괄목할 변화가 있을 경우 major number 를 변경한다. 이미 작성한 부분에 대해서 오타 수정하거나, 문서의 구조를 변경할 경우 maintenance number 를 추가하거나 변경한다.

### C. Version update history

Version	DATE	Explanation
0.0	2015-05-05	문서의 초안과 구조 작성
1.0	2015-05-05	Preface, Glossary, Index 작성
1.0	2015-05-05	Introduction 작성 및 문서 skeleton 구축
2.0	2015-05-11	System Architecture 작성
2.1	2015-05-16	System Architecture 수정
2.1	2015-05-16	각 서브시스템 구상
2.1.1	2015-05-18	System Architecture 부분 오타 수정
3.0	2015-05-18	Diagrams 작성
3.1	2015-05-20	Diagrams 수정 및 추가
3.2	2015-05-20	System Architecture block diagram 추가
4.0	2015-05-23	User activity 작성
4.0	2015-05-23	각 서브시스템 서술 부분 작성
5.0	2015-05-25	서브시스템 추가 작성 및 수정
5.0	2015-05-25	Structure model, behavioral model 작성
5.1	2015-05-25	Scenario, Sequence diagram 작성
5.1	2015-05-27	Class diagram 작성
6.0	2015-05-27	각 diagram 취합 및 정리
6.1	2015-05-27	문서 index 수정
6.1.1	2015-05-28	오타 수정
7.0	2015-05-28	Database Design 작성
7.0	2015-06-02	신규 diagram 및 표 추가
7.0	2015-06-02	Test 작성
8.0	2015-06-04	Appendices, Reference 작성
8.0.1	2015-06-04	문서 정리 및 오타 확인

## 2. Introduction

이 섹션에서는 시스템의 필요성에 대해서 설명한다. 시스템에 포함된 서브시스템들의 간략한 소개와 이들의 상호작용에 대해서 설명한다.

### 2.1. 배경 및 필요성

2016 년 I-campus 를 이용하는 유저들을 대상으로 진행한 설문 조사 결과에 따르면, 매 학기, 상당수의 학생이 교수님이 I-campus 에 과제를 늦게 업로드 해주어서 과제를 정해진 시간 내에 제출하지 못하여 불이익을 받은 경험을 가지고 있는 것으로 조사되었다. 본 설문 조사를 통해 I-campus 를 일주일에 5 번 이상 접속함에도 불구하고 갑작스런 과제 업로드 때문에 과제를 미리 준비하지 못하는 학생들도 있다는 것을 알게 되었다. 이는 학생들이 수업 material 을 준비하고, 과제를 수행함에 있어서 매우 안좋은 영향을 주고 있다. 또한 대부분의 학생들에게 성적은 중요하고 민감한 부분이기 때문에 이러한 문제는 간과할 수 없다고 판단하였다.

Have you ever miss to hand out homework  
since late referring to icampus?

■ Yes ■ No



average number of visits for a week

■ 0 ■ 1~2 ■ 3~4 ■ more than 5



그림 1. I-campus 를 이용하는 학생들의 설문결과(google 설문지)



## 2.2. Homework Alert System (HAS)

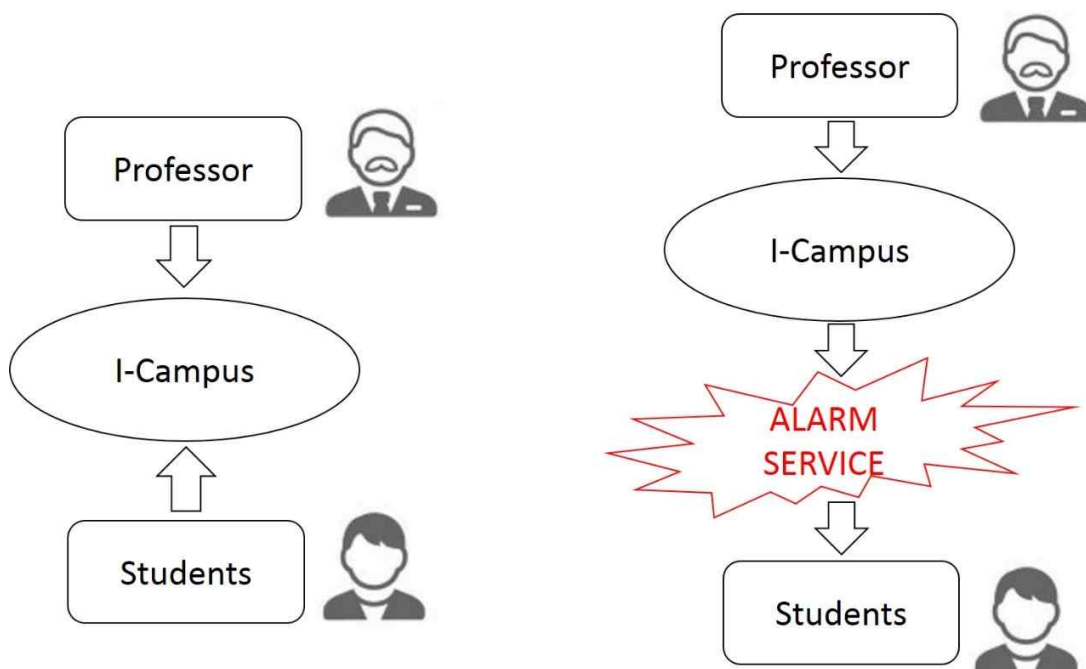
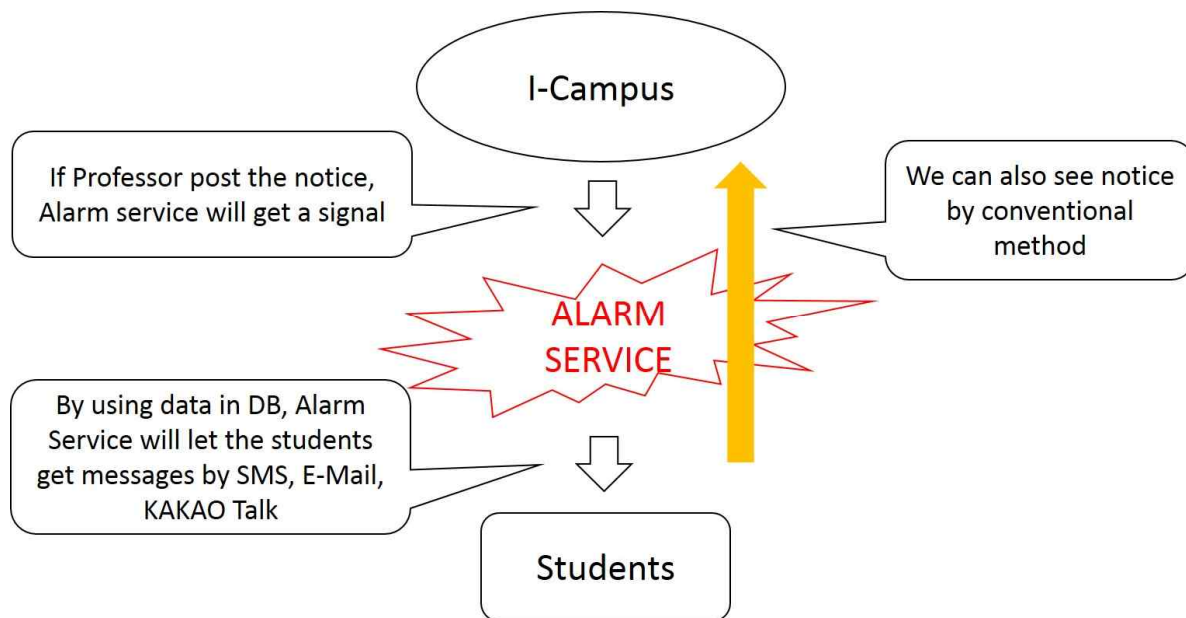


그림 2. HAS(Homework Alert System)의 전체 구조

HAS(Homework Alert System)은 교수님과 학생들의 자료 및 정보 공유를 통하여 교수님이 과제, 프로젝트를 I-Campus에 올릴 시에 즉시 HAS 통하여 스마트폰으로 메시지를 보내 주어 학생들이 공지사항을 늦게 확인 하지 않고, 빠르게 확인할 수 있게 해주는 시스템이다. <그림 2>는 HAS의 구성요소, I-Campus, 학생 그리고 교수가 서로간의 Interface를 통하여 신호를 송신 및 수신하는 diagram이다. HAS에 교수님이 학과목을 생성하고, 그 안에 학생들이 본인의 정보를 입력하면 입력된 정보들을 통하여 HAS는 학생들이 가지고 있는 스마트폰으로 즉시 메시지를 전송해 주어 공지사항을 확인할 수 있게 해준다. HAS는 교수님으로부터 접근 신호를 받으면 DB에 교수님의 정보가 일치하는지를 확인하고, DB에서 교수님 접근 권한이 있다는 정보를 받는다. 이후 HAS는 교수님이 과제를 부여할 course의 정보를 요구한다. 교수님이 course정보를 입력하면 HAS는 DB로부터 course의 정보 유무를 확인하고 course내에 학생들의 정보 유무를 확인한다. DB에서 학생들의 정보가 존재하는지 확인이 되면 HAS는 교수님께 학생에게 전송할 과제 정보를 요구한다. 마지막으로 교수님이 과제 정보를 HAS에 업로드 하면 HAS는 DB로부터 과제 Notice를 전송할 학생 정보를 받고 학생에게 과제가 생겼다는 신호와 과제에 대한 간략한 정보를 전송한다.

\* I-Campus는 교수의 정보와 학생들의 정보를 가지고 있는 System이며, 학생들과 교수님 간의 활발한 인터페이스가 이루어 지는 공간이다.

\* HAS는 교수님이 강의중인 학과목을 입력 받고 그 과목을 수강중인 학생들의 정보를 입력 받는다. 그리고 그 정보를 가지고 과제가 업로드 시 HAS에 입력한 학생들에게 메시지를 전송한다.

## 2.3. 기대 효과

이 System 이 성균관대학교에 구축이 된다면 학생들의 과제에 대한 심적불안도를 낮출 수 있을 것이다. 대부분의 학생들은 스마트폰을 이용하기 때문에, 스마트폰을 가지고 시공간의 제약없이 정보를 편리하고 편안하게 확인할 수 있게 될 것이다. 또한 매번 I-Campus 에 접속하여 정보를 확인해야 하는 번거로움을 없앨 수 있기 때문에 효율적이다. 교수님의 입장에서는 올리는 즉시 학생들이 더 쉽고 확실하게 확인할 수 있게 하기 때문에 교수님의 입장에서 매우 편리한 측면을 가지는 시스템이다.

## 2.4. Constraints

### A. KAKAO TALK APP 필요

본 시스템은 카카오톡과의 연동을 기본전제로 생각하고 process 를 진행한다. 따라서 학생 입장의 user 는 KAKAOTALK APP 사용권한이 없다면 본 시스템은 이용할 수 없다. 따라서 본 시스템을 이용하기 위해서는 단말기 혹은 개인 PC 에 KAKAO TALK 을 사용할 수 있는 환경이 구축 되어야한다. 또한 단말기에 시스템으로부터 메시지가 도착하기 위해서는 사용자의 카카오톡에 시스템과 연동된 카카오 옐로 아이디가 친구목록에 등록되어야 한다.

### B. 지속적 시스템 연동 필요

본 시스템은 앞서 A 에서 언급했듯 카카오톡과의 연동을 기본전제로 생각한다. 즉 카카오톡 APP 이 업데이트를 하거나 새로운 format 을 적용한다면 이는 maintenance 로 새롭게 설정을 적용시켜 주어야 한다. 즉 언제 생길지 모르는 잠재적 error 에 대해 지속적인 maintenance 가 필요하다.

### C. 개인정보 사용 동의 필요

본 시스템은 학생들의 정보를 데이터베이스로부터 호출하여, 학생들에게 메시지를 전송하는 시스템이다. 따라서 학생들의 정보가 데이터베이스에 저장되어 있어야 한다. 시스템의 데이터베이스에 저장되는 정보가 학생들의 개인정보이기 때문에 학생들에게서 개인정보 사용 동의를 사전에 요구한다. I-Campus 로부터 각 강의 별 수강하는 학생들의 데이터가 잘 저장되어 있지 않으면 시스템의 메시지 전송 과정에서 문제가 발생한다.

### D. 데이터베이스의 정보 지속적 관리 필요

본 시스템의 데이터베이스에는 학생들은 개인정보가 저장된다. 때문에 악의적인 사용자가 데이터베이스에 접근하여 개인정보를 유출할 위험이 항상 존재한다. 따라서 본 시스템을 운영하는 동안은 시스템 데이터베이스에 접근권한을 가진 사람 외의 이용자가 시스템에 접근하는지 여부를 지속적으로 확인하여야 한다.

# 3. Glossary

이 섹션에서는 문서에 사용된 기술용어들의 정의를 설명하여 독자들의 문서 이해도를 높이도록 한다.

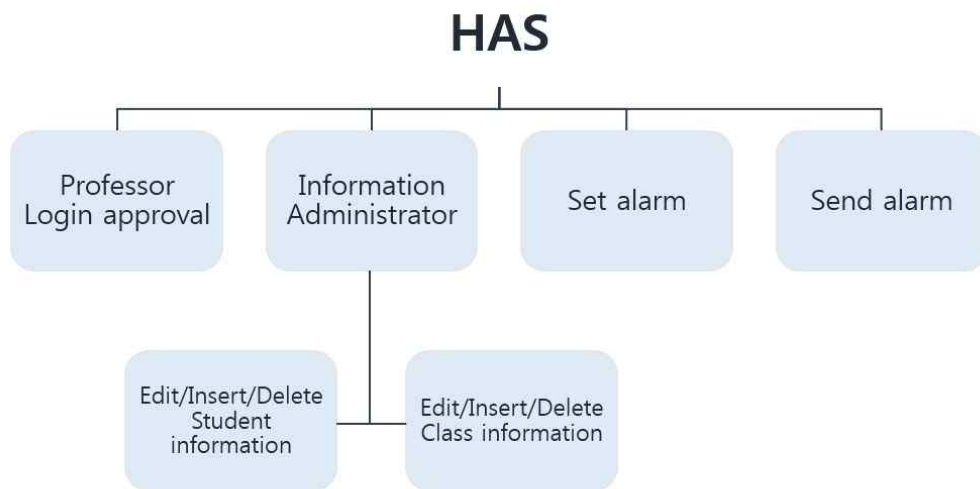
표 1. Glossary

용어	정의
App (앱)	Application(응용 소프트웨어)의 준말로 운영체제에서 실행되는 모든 소프트웨어 또는 사용자가 직접 사용하게 되는 소프트웨어를 의미
Java	썬 마이크로시스템즈의 제임스 고슬링(James Gosling)과 다른 연구원들이 개발한 객체 지향 적 프로그래밍 언어이며 플랫폼 독립적이라는 특성 때문에 애플리케이션 개발 등에 폭넓게 활용되고 있다.
모바일 디바이스	주머니에 쏙 들어갈만한 크기의 컴퓨터 장치로, 일반적으로 터치 입력이 가능한 화면이나 소형 자판을 가짐
URL	네트워크 상에서 자원이 어디 있는지를 알려주기 위한 규약이다. 웹 사이트 주소뿐만 아니라 컴퓨터 네트워크상의 자원을 모두 나타낼 수 있다. 그 주소에 접속하려면 해당 URL 에 맞는 프로토콜을 알아야 하고, 그와 동일한 프로토콜로 접속해야 한다.
User Interface	사람(사용자)과 사물 또는 시스템, 특히 기계, 컴퓨터 프로그램 등 사이에서 의사소통을 할 수 있도록 일시적 또는 영구적인 접근을 목적으로 만들어진 물리적, 가상적 매개체를 뜻한다.
Waterfall Model	소프트웨어 개발 과정인 Specification, Design, Implementation, Evolution 단계를 차례대로 수행해 나가는 소프트웨어 개발 프로세스 모델.
Wi-Fi	와이파이 얼라이언스(Wi-Fi Alliance)의 상표명으로, IEEE 802.11 기반의 무선랜 연결과 장치 간 연결 등을 지원하는 일련의 기술을 의미
C#	<a href="#">C++</a> 와 <a href="#">자바</a> 에서 강한 영향을 받았다. 이름의 유래는 두 가지에서 따왔다고 한다. <a href="#">#</a> 첫번째 의미는 음악에서 유래한 것으로, 도(C)에 반음(#)을 올린 것.
HAS	Homework Alert System 으로 학생들에게 메시지를 보내기 위한 시스템
DB	Data Base 의 약자로 데이터를 관리 하기 위한 시스템
SQL	Structure Query Language 의 약자로 데이터 정의어(DDL)와 데이터 조작어(DML)를 포함한 데이터베이스용 질의언어(query language)의 일종이다

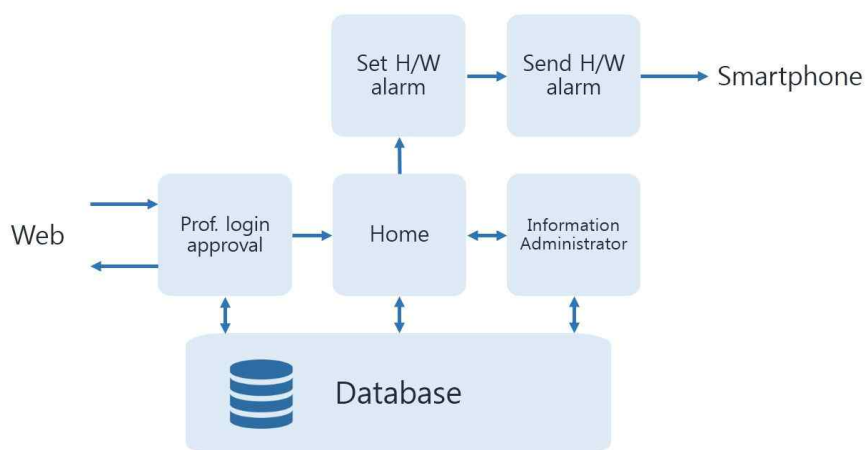
# 4. System Architecture

이 섹션에서는 시스템에 대한 전반적인 개요 및 기능에 대해 서술한다. 시스템의 구조를 여러 Diagram 으로 표현하며 이에 대한 자세한 설명도 첨부한다.

## 4.1. System Architecture Overview



다이어그램 1. System Architecture Overview



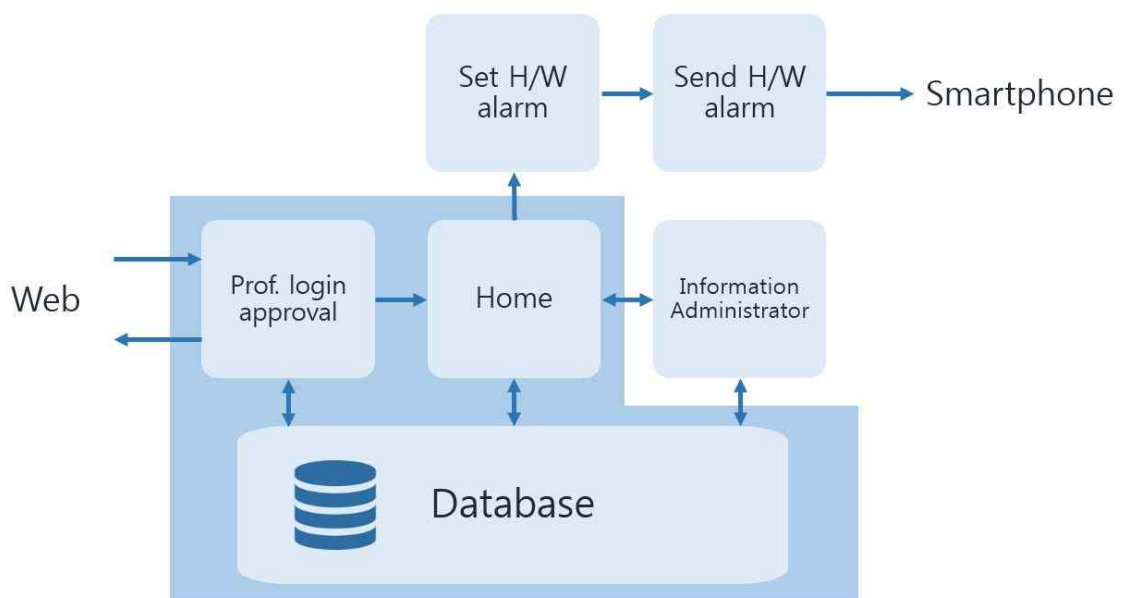
다이어그램 2. System Application Overview

HAS 는 Web-based system 으로서 교수님이 HAS 에 과제를 생성하면 학생들의 모바일 기기로 알림을 전해주는 system 이다. HAS 가 학생들에게 알림을 보내는 데에 필요한 데이터는 Database 에 저장된다.

## 4.2. Subsystem1. Prof. Log-in Approval system



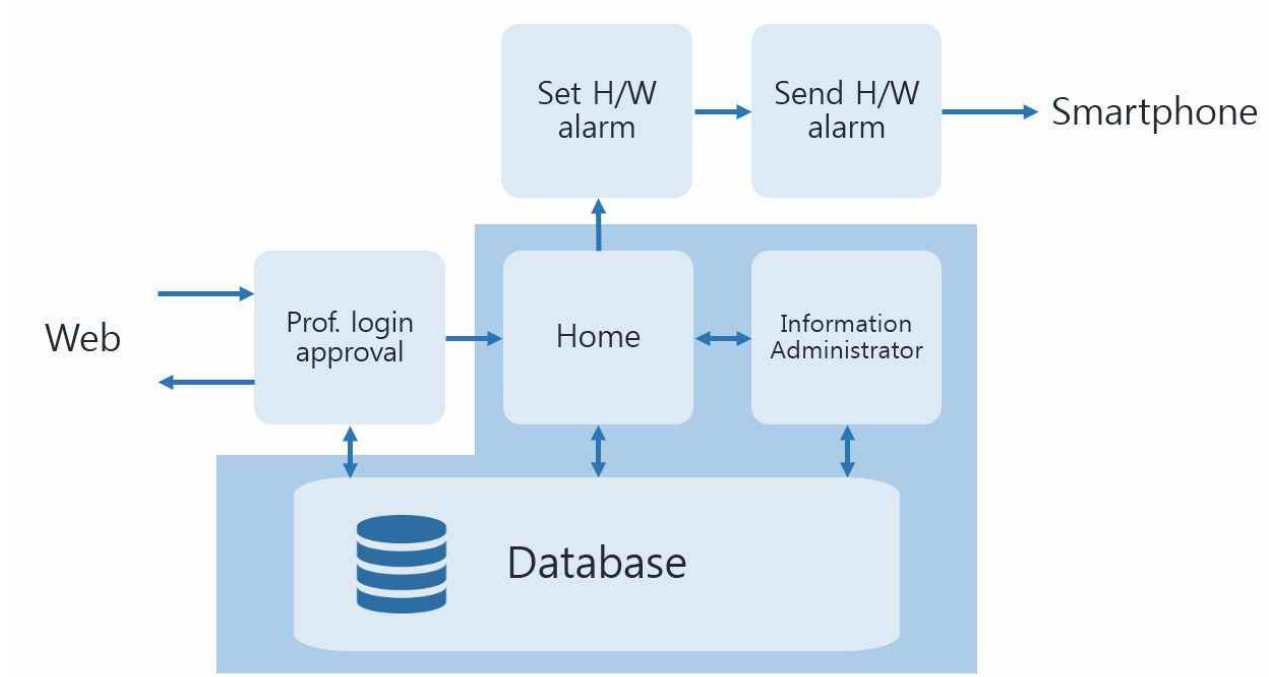
그림 3. System Application Overview



다이어그램 3. Prof. Log-in approval subsystem Overview

HAS의 첫번째 subsystem인 Prof. Log-in Approval system은 시스템을 시작했을 때 처음에 인증을 담당하는 시스템이다. 로그인 정보는 데이터베이스에서 관리되며 본 시스템에서 입력된 정보와 비교한다.

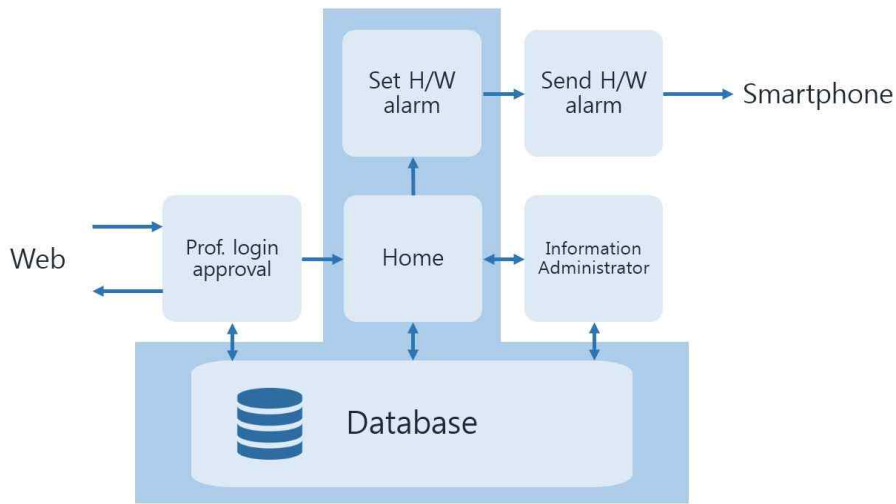
### 4.3. Subsystem2. Information Administrator



다이어그램 4. Information Administrator Overview

HAS의 두번째 subsystem인 Information Administrator은 로그인 인증 이후, 이용자인 교수님의 계정에 할당된 Class, Student 정보를 관리하는 시스템이다. 저장된 내용들을 확인하는 기능과 내용들을 수정하는 기능을 포함한다.

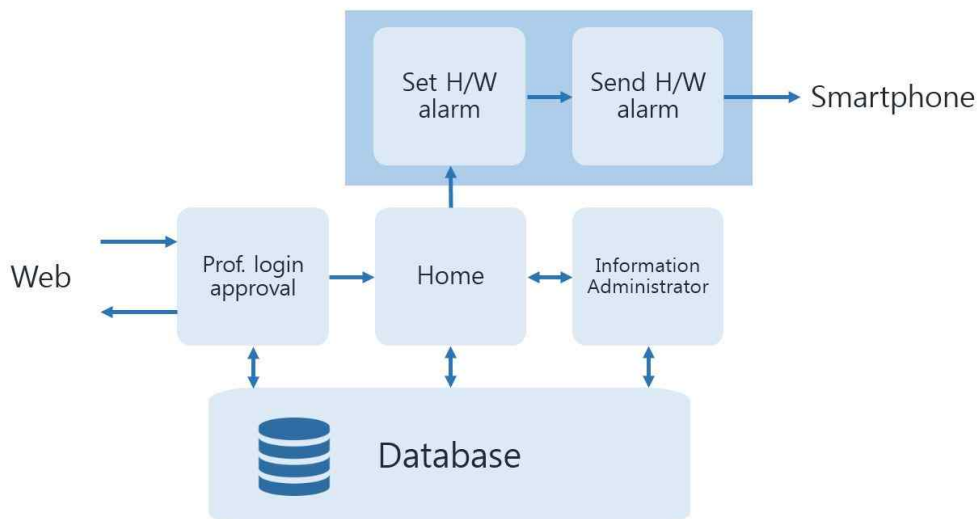
#### 4.4. Subsystem3. Set H/W Alarm



다이어그램 5. Set H/W Alarm Overview

HAS 의 세번째 subsystem 인 Set H/W Alarm 은 보낼 알림의 내용을 작성하는 서브시스템이다. 본 서브시스템안에서는 과제의 간략한 내용과 due date 등을 포함하고 있으며 이를 형식에 맞게 저장해주면 발송할 메시지를 자동으로 생성한다. 추가로 보내야할 대상인 학생들의 정보 역시 본 subsystem 에서 할당한다.

#### 4.5. Subsystem4. Send H/W Alarm



다이어그램 6. Send H/W Alarm Overview

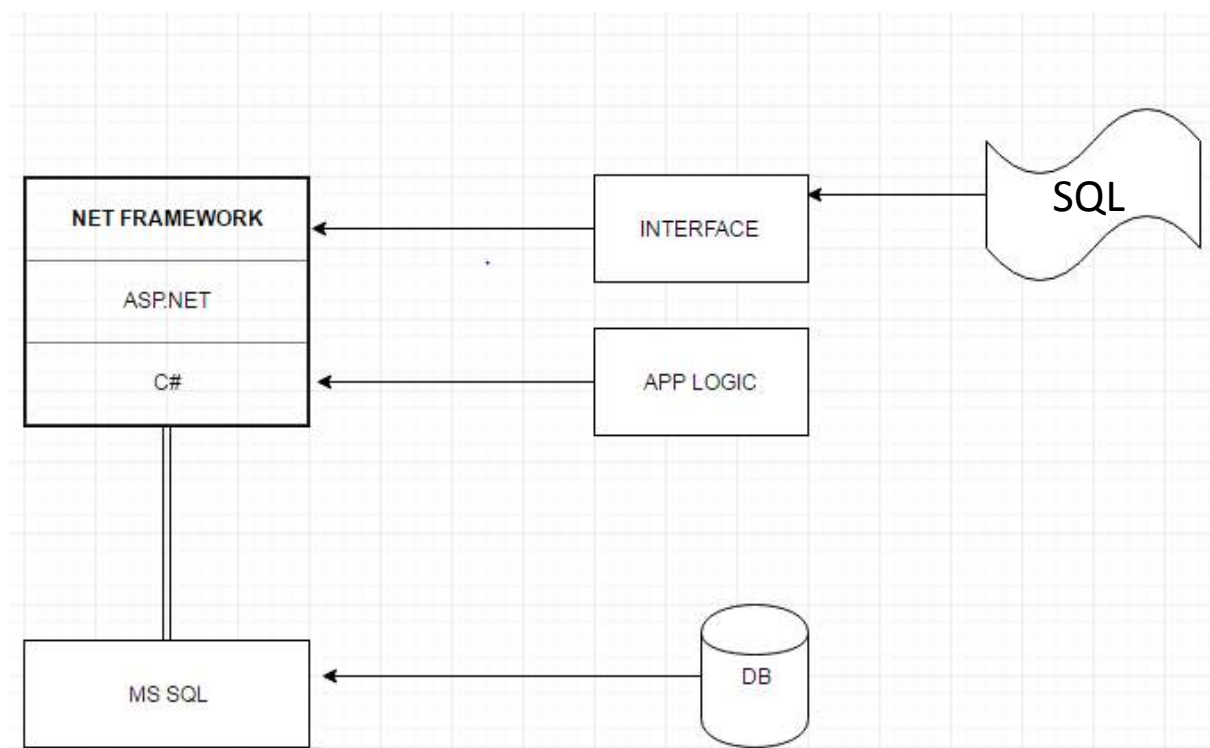


HAS 의 네번째 subsystem 인 Send H/W Alarm 은 이전 단계에서 저장된 메시지를 최종적으로 지정된 학생들의 카카오톡 아이디로 발송한다. 메시지를 보낼 때는 본 시스템은 카카오톡 옐로 아이디를 사용한다.

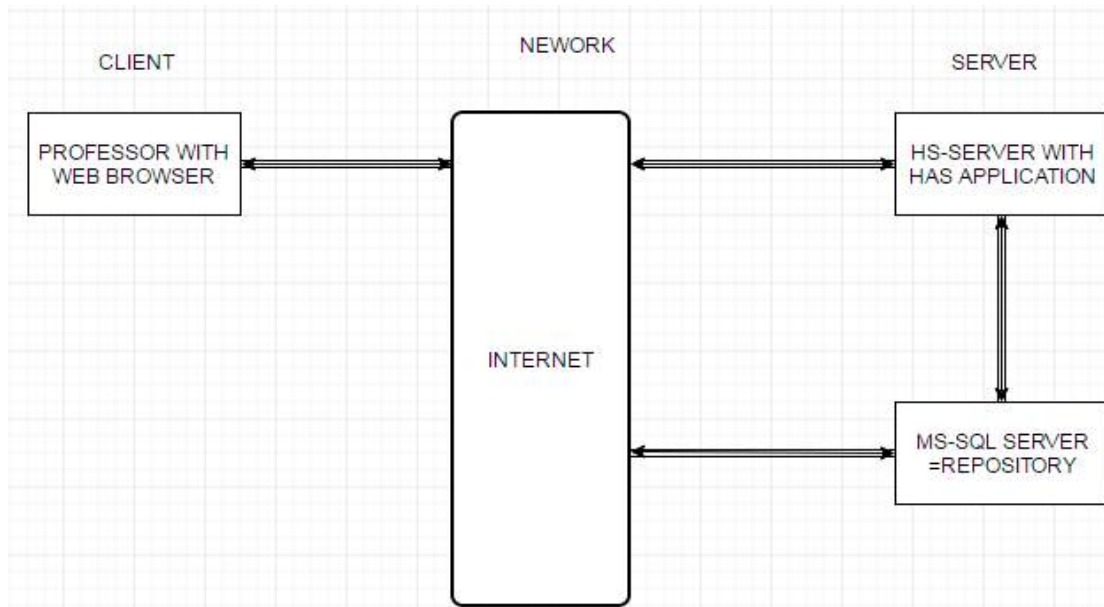


그림 4. 카카오톡 옐로아이디

## 4.6. 전체 네트워크 구조

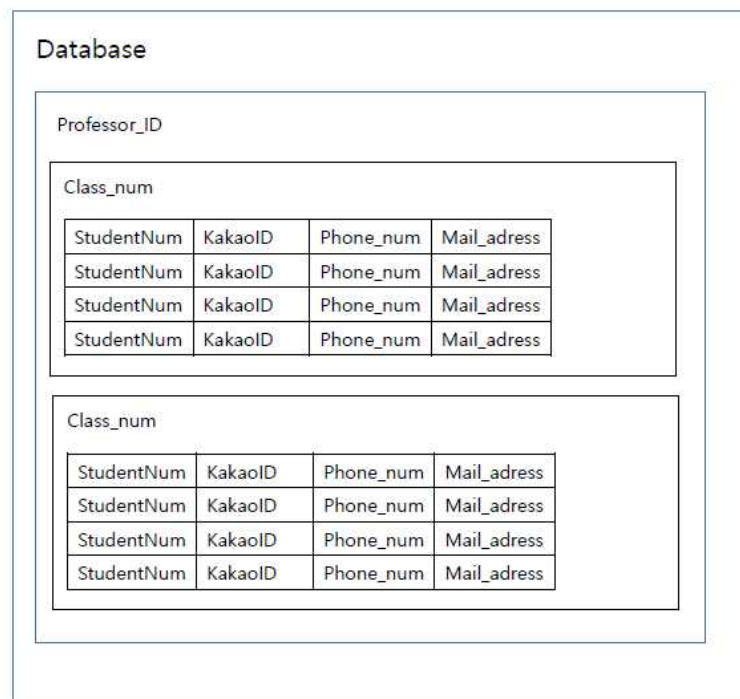


다이어그램 7. NETWORK 접속 구조도



다이어그램 8. NETWORK 접속 구조도

## 4.7. 데이터베이스 구조



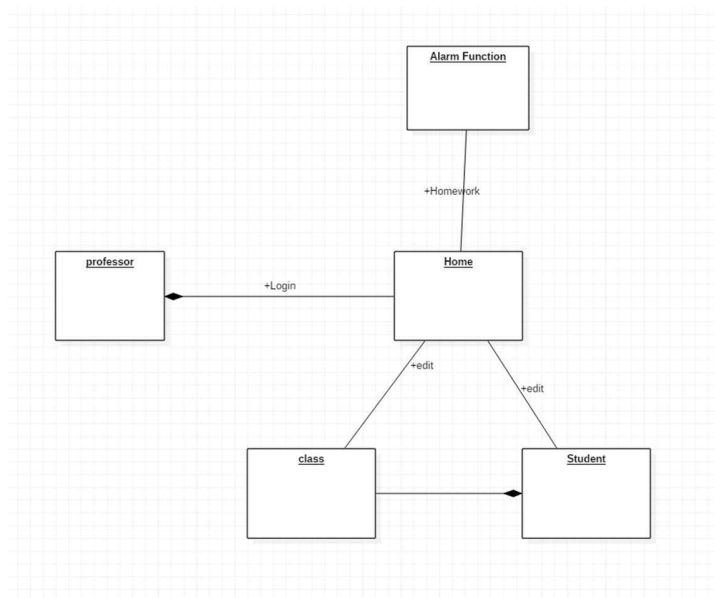
다이어그램 9. Database 구조도

Database 에는 각 Professor 별로 Class\_num 가 할당되어 있고, 각 Class 별로, 수강학생의 학번, 카카오 ID, 휴대폰 번호, 메일주소가 저장되어 있다.

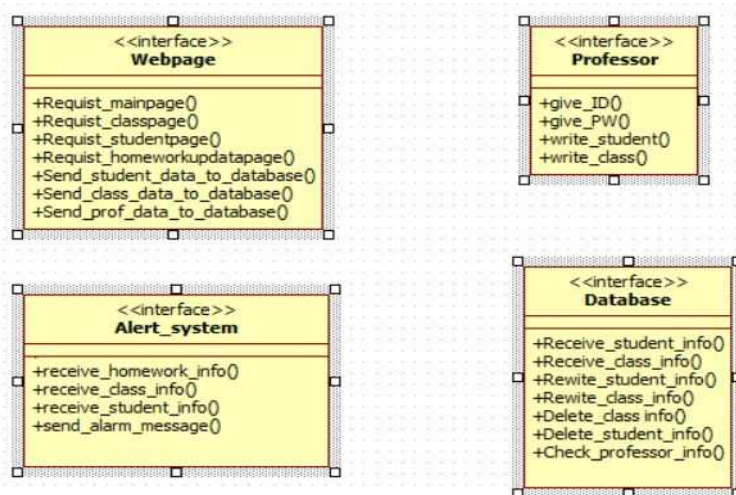
# 5. User Activity

User Activity 섹션은 HAS 시스템으로부터 User 까지 직접 제공되는 인터페이스들에 대해 설명하는 섹션이다.

## 5.1. Class Diagram

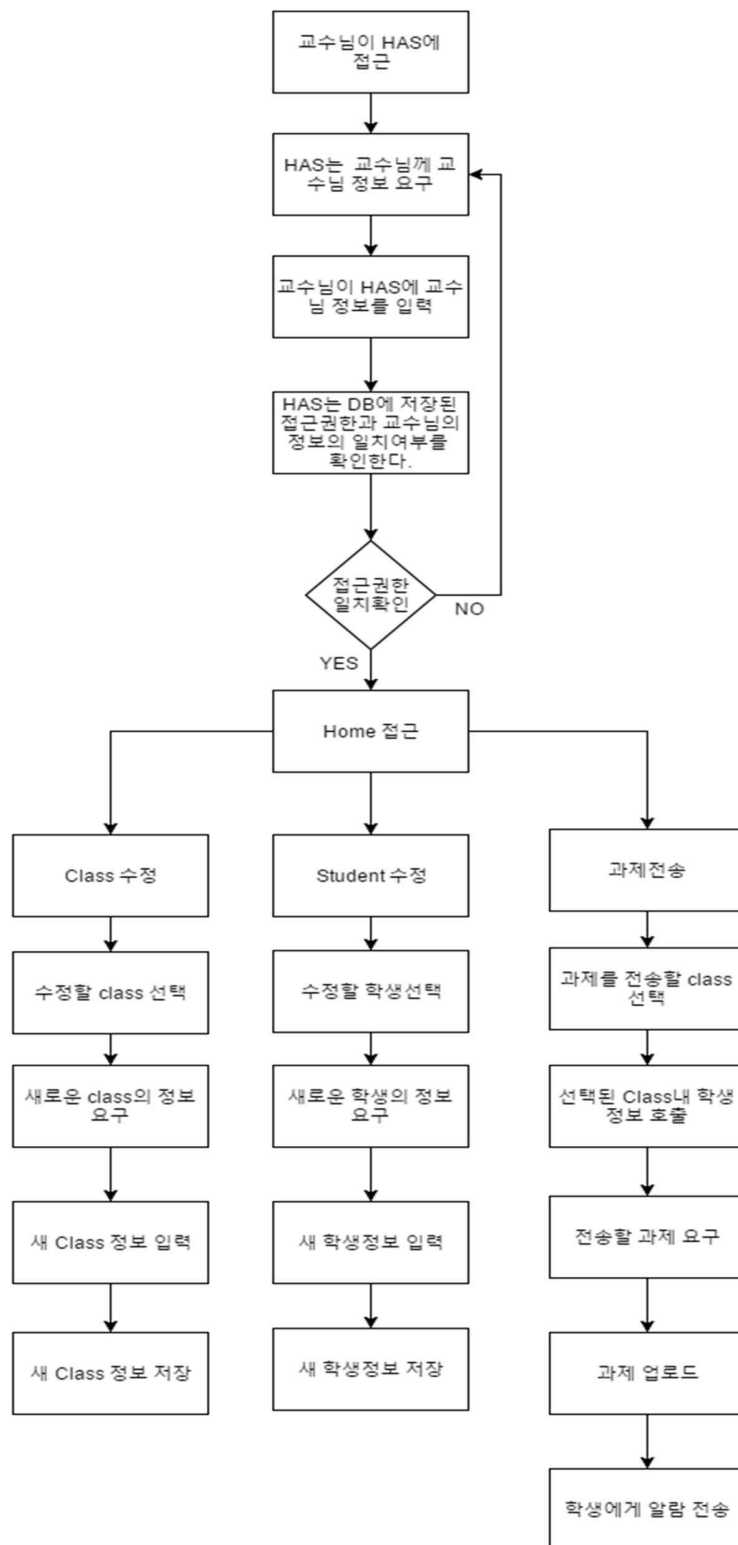


다이어그램 10. HAS class diagram (1)



다이어그램 11. HAS class diagram (2)

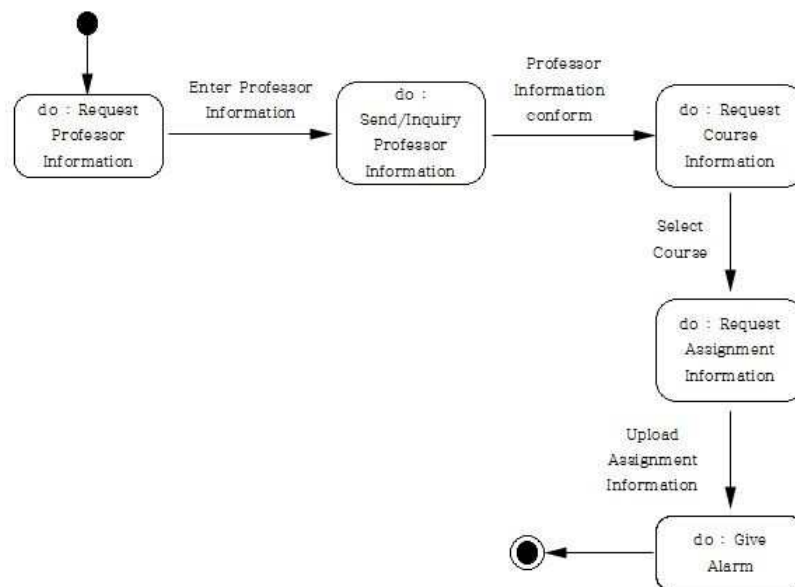
## 5.2. Scenario



다이아그램 12. HAS Scnario

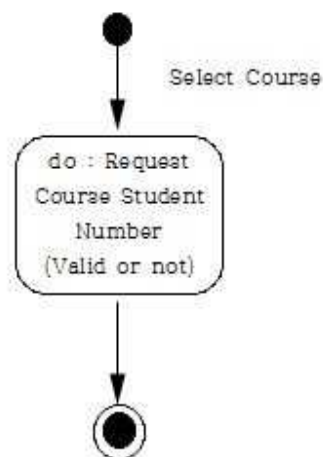
## 5.3. Activity Models

### 5.3.1. Professor activity 모델



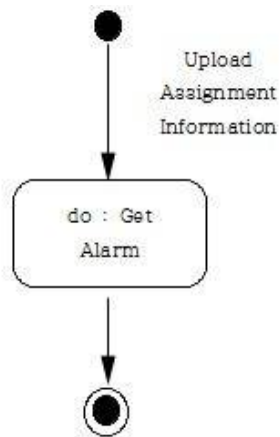
다이어그램 13. Professor activity 모델

### 5.3.2. Course activity 모델



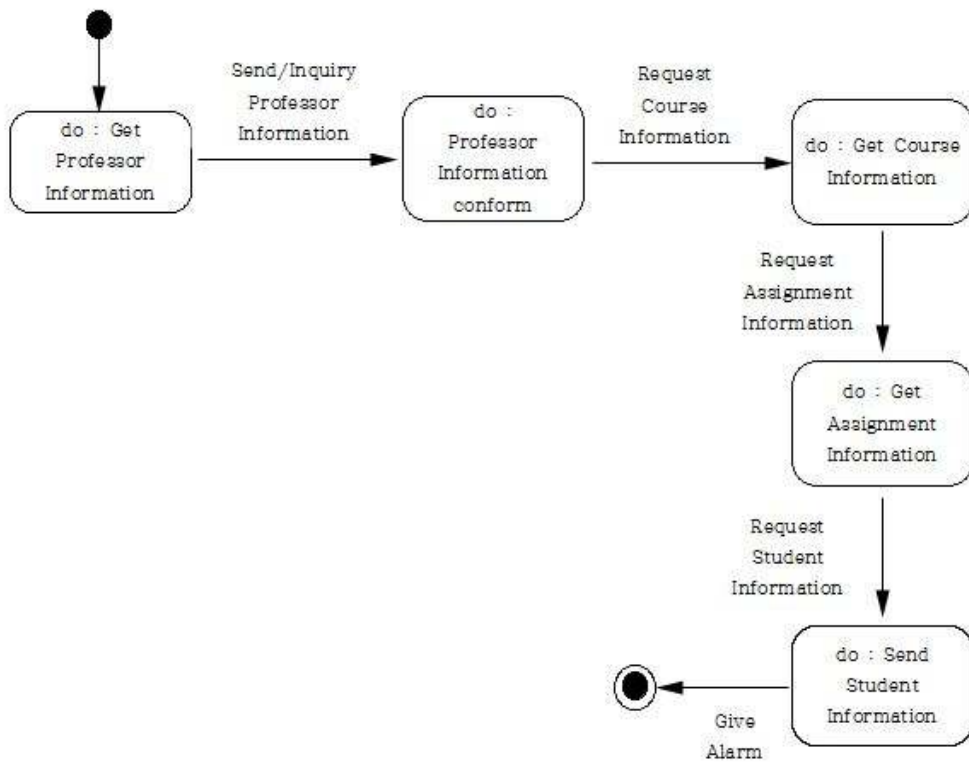
다이어그램 14. Course activity 모델

### 5.3.3. Student activity 모델



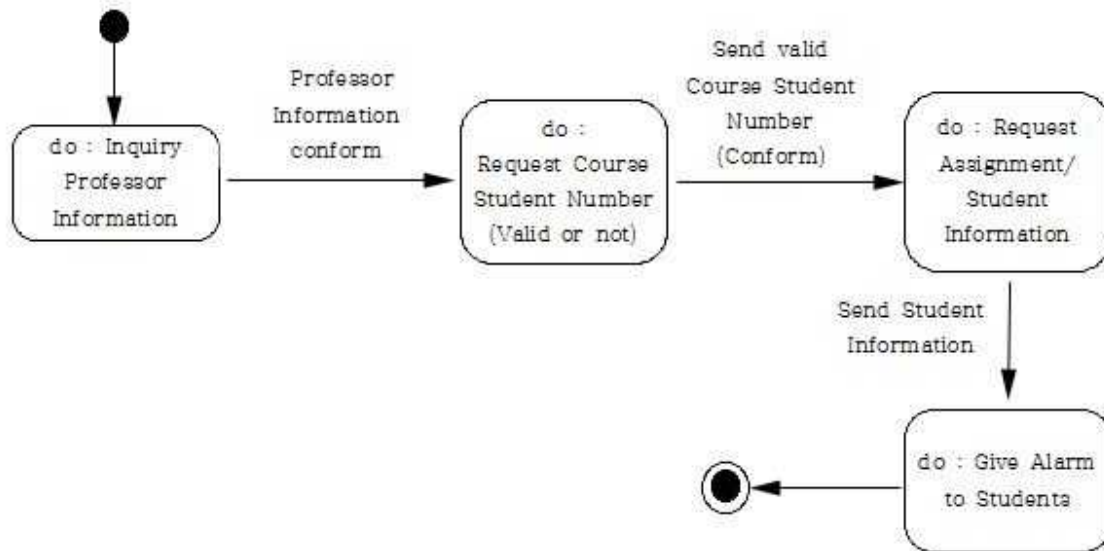
다이어그램 15. Student activity 모델

### 5.3.4. HAS activity 모델



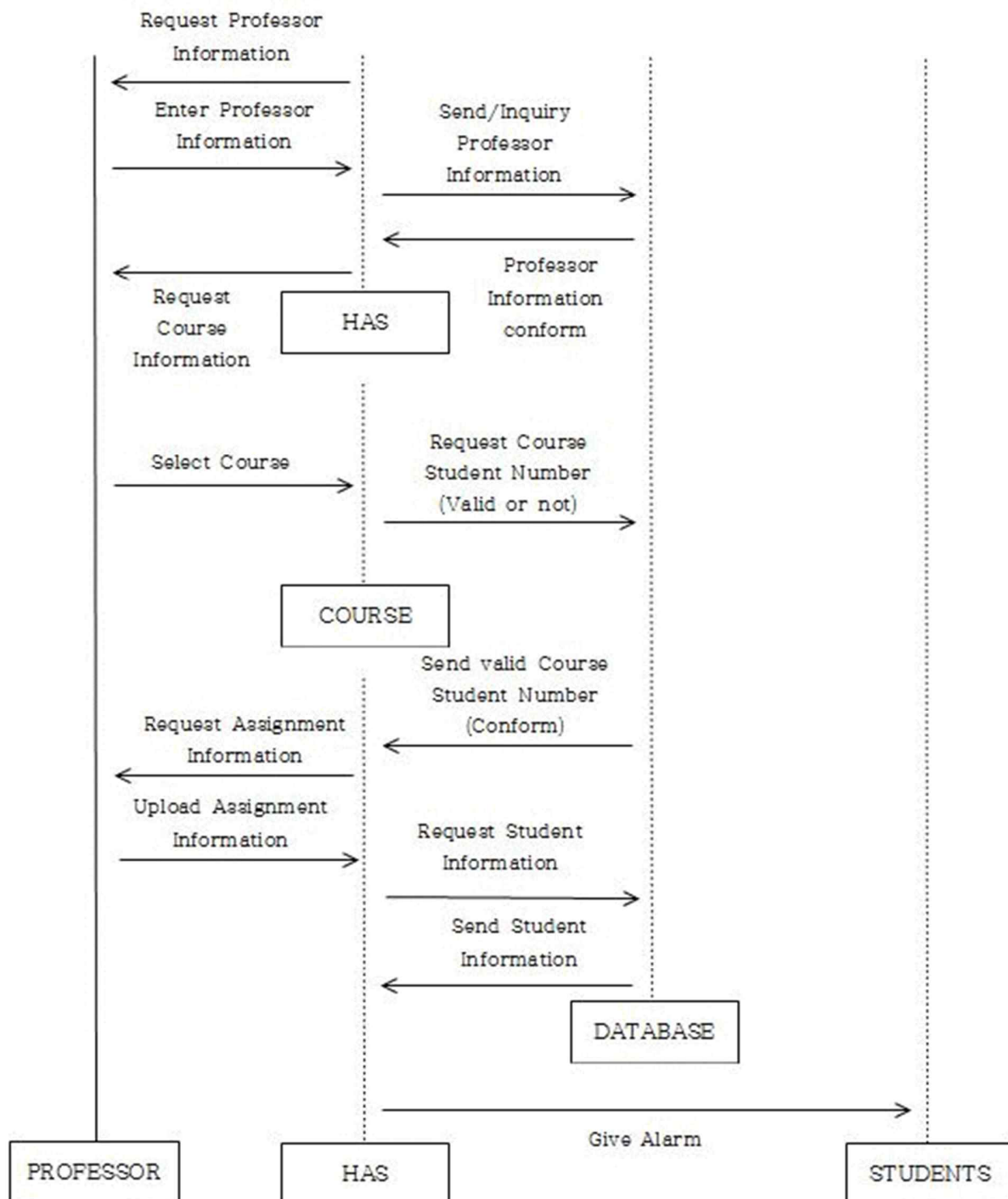
다이어그램 16. HAS activity 모델

### 5.3.5. Database activity 모델



다이어그램 17. Database activity 모델

## 5.4. Overall Sequence Diagram

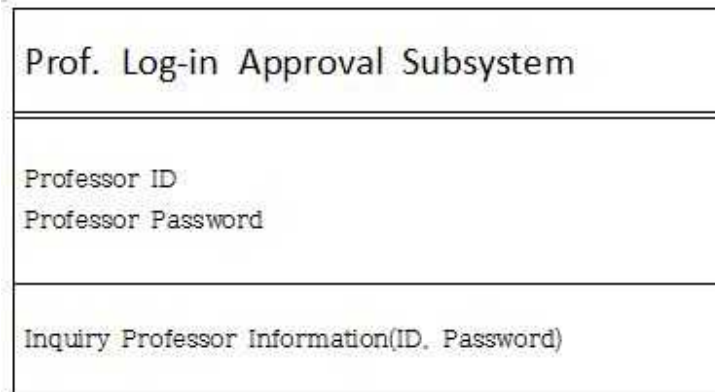


다이어그램 18. Overall Sequence Diagram



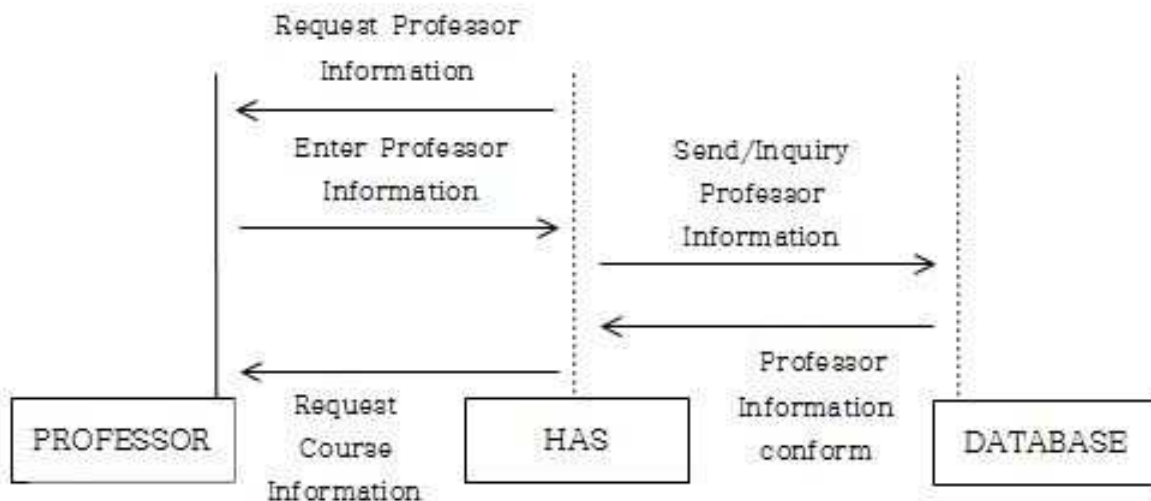
## 6. Prof. Log-in Approval Subsystem

### 6.1. Class Diagram



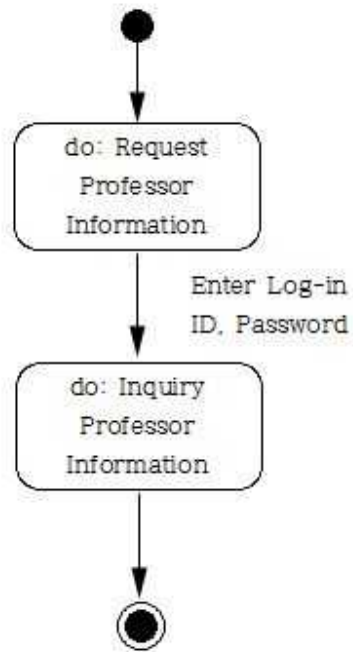
다이어그램 19. Log-in Approval sub-system Class Diagram

### 6.2. Sequence Diagram



다이어그램 20. Log-in Approval sub-system Sequence Diagram

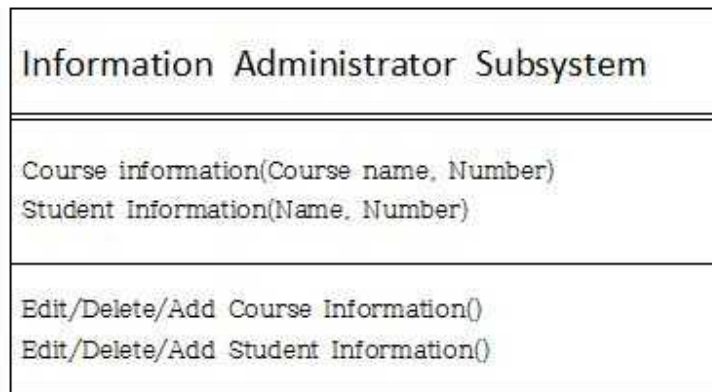
### 6.3. State Diagram



다이어그램 21. Log-in Approval sub-system State Diagram

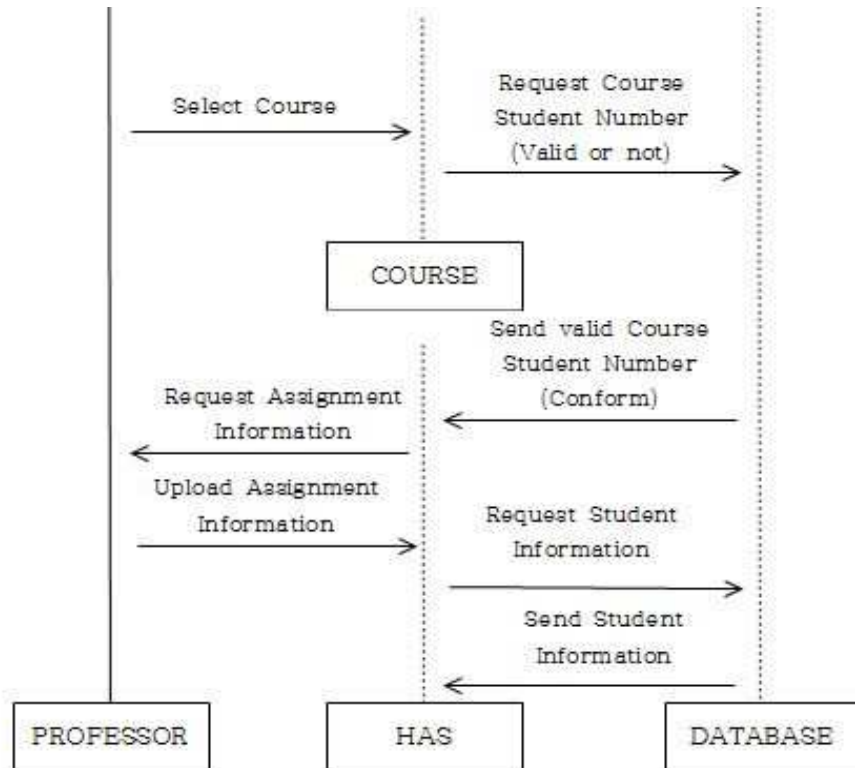
## 7. Information Administrator Subsystem

### 7.1. Class Diagram



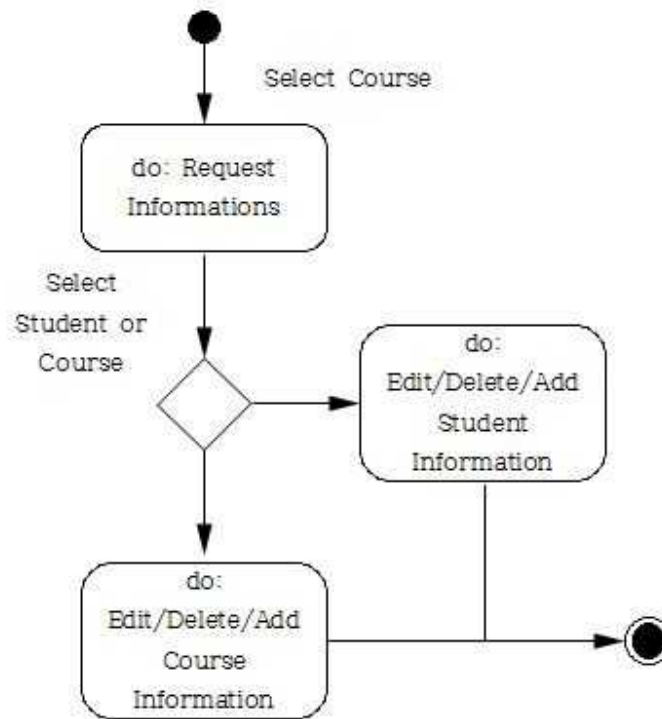
다이어그램 22. Information Administrator sub-system Class Diagram

### 7.2. Sequence Diagram



다이어그램 23. . Information Administrator sub-system Sequence Diagram

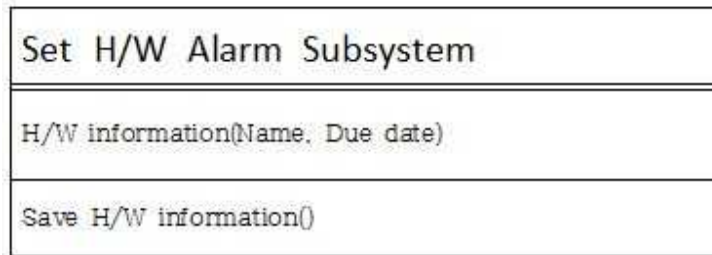
### 7.3. State Diagram



다이어그램 24. . Information Administrator sub-system State Diagram

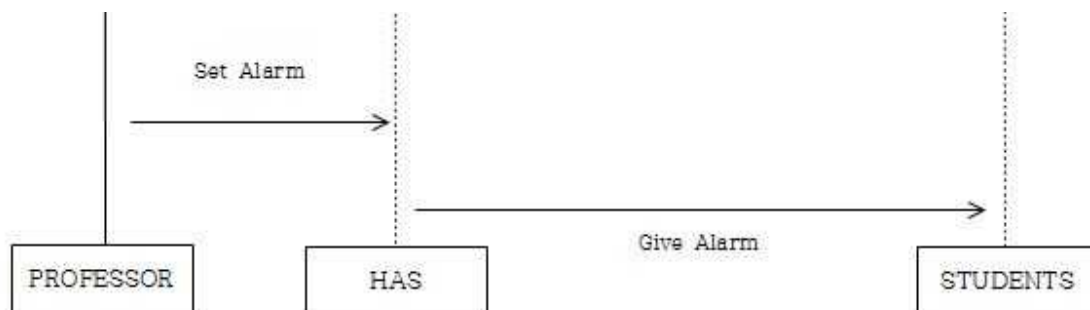
# 8. Set H/W Alarm Subsystem

## 8.1. Class Diagram



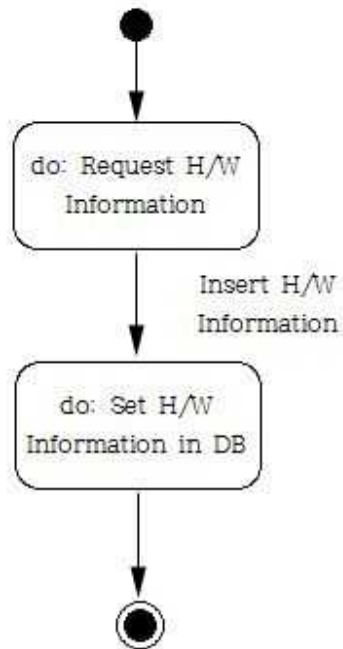
다이어그램 25. Set H/W Alarm sub-system Class Diagram

## 8.2. Sequence Diagram



다이어그램 26. Set H/W Alarm sub-system Class Diagram

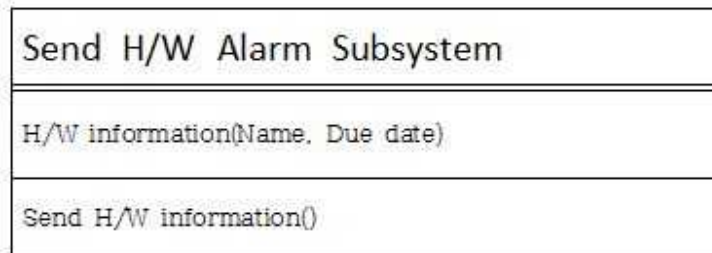
### 8.3. State Diagram



다이어그램 27. Set H/W Alarm sub-system State Diagram

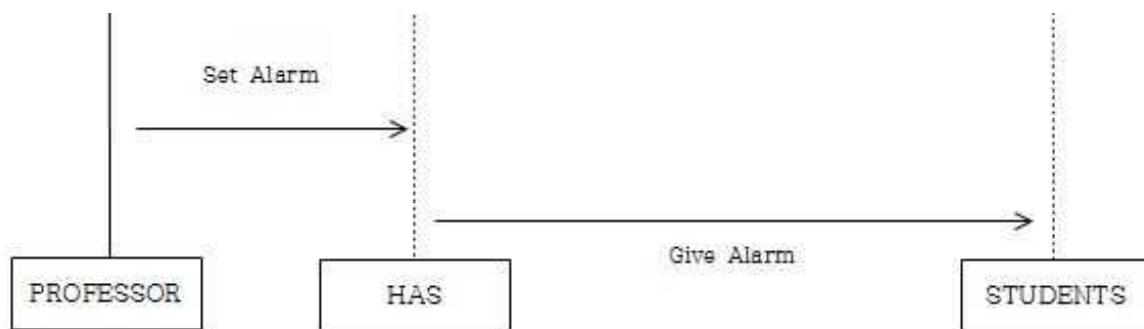
# 9. Send H/W Alarm Subsystem

## 9.1. Class Diagram



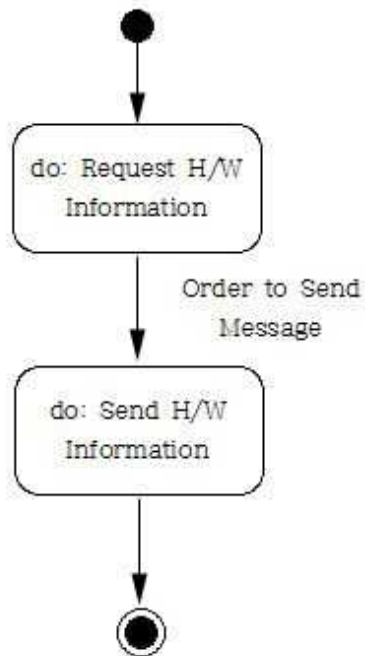
다이어그램 28. Send H/W Alarm sub-system Class Diagram

## 9.2. Sequence Diagram



다이어그램 29. Send H/W Alarm sub-system Class Diagram

### 9.3. State Diagram



다이어그램 30. Send H/W Alarm sub-system State Diagram



# 10. Test

Test 섹션에서는 개발 구현 과정에서 시도한 시험들에 대해 서술한다. 또한 System 의 기능과 성능을 시험하기 위한 테스트 모듈에 대해 설명한다.

## 10.1. Test Image

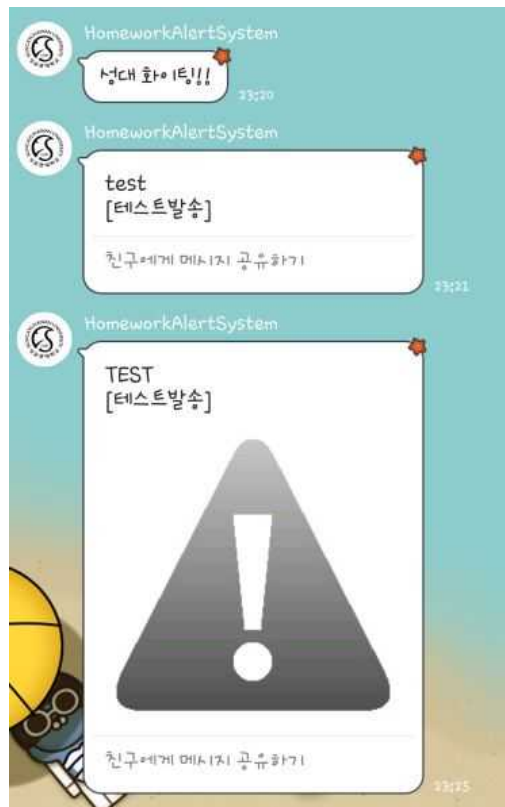


그림 5. Test of HAS

Test 는 승인 받은 카카오톡 옐로아이디로부터 학생들의 개인계정으로 test 메시지를 보내는 일련의 과정으로 진행하였다. 위의 이미지와 같이 다음과 같은 3 가지 형식의 메시지를 학생의 개인 카카오톡 계정으로 보내는 Test 를 진행하였는데, 결과는 위의 이미지와 같이 모두 완벽히 전송되었다.

\* 3 가지형식

1. 일반 메시지
2. 공유 가능 메시지
3. 이미지 포함 메시지

# 11. Development Environment

Development Environment 섹션에서는 구현 과정에서 이용한 개발 Tool 에 대해 설명한다.

## 11.1. Operating System



그림 6. Window 7

우선적으로 개발 OS 환경의 경우 대부분의 학생이 사용하는 Window7 을 선택하였다. 이는 범용성면에서도 상당히 큰 장점을 가져가며 대부분의 학생이 개발에 참여함에 불편함이 없게 만들어주었다.

## 11.2. 코딩 환경



그림 7. Window 7

또한 코딩 환경의 경우 모든 학생들이 이용해본 경험이 있는 Visual Studio 를 사용하였다. 역시 Window7 의 경우와 마찬가지로 범용성면에서 강점을 가졌고, 또한 학생들의 참여율도 올릴 수 있다는 강점도 가지고 있다.

## 11.3. Programming Language

### A. C#

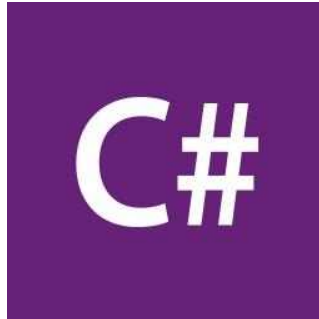


그림 8. C#

마이크로소프트에서 개발한 객체지향 프로그래밍 언어이다 C++와 자바에서 강한 영향을 받았으며, 이름의 유래는 #의 의미로써, 이는 음악에서 유래한 것으로, 도(C)에 반음(#)을 올린 것이라고 한다. 언어적으로는 JAVA와 유사하지만, 특정 플랫폼에 한정되어 있지는 않다. 여러가지 기능이 새롭게 추가되고 있으며, 점유율 역시 보편적인 편이다. [6]

### B. JAVA



그림 9. Java

**자바**(Java)는 썬 마이크로시스템즈의 제임스 고슬링(James Gosling)과 다른 연구원들이 개발한 객체 지향적 프로그래밍 언어이며, 썬 마이크로시스템즈에서 무료로 제공하고 있다. 1991 년 그린 프로젝트(Green Project)라는 이름으로 시작해 1995 년에 발표했다. 처음에는 가전제품 내에 탑재해 동작하는 프로그램을 위해 개발했지만 현재 웹 애플리케이션 개발에 가장 많이 사용하는 언어 가운데 하나이고, 모바일 기기용 소프트웨어 개발에도 널리 사용하고 있다. 현재 버전 9 까지 출시했다. 자바의 개발자들은 유닉스 기반의 배경을 가지고 있었기 때문에 문법적인 특성은 파스칼이 아닌 C++의 조상인 C 언어와 비슷하다. 자바를 다른 컴파일언어와 구분 짓는 가장 큰 특징은 컴파일 된 코드가 플랫폼 독립적이라는 점이다. 자바 컴파일러는 자바 언어로 작성된 프로그램을 바이트코드라는 특수한 바이너리 형태로 변환한다. 바이트코드를 실행하기 위해서는 JVM(자바 가상 머신, Java Virtual Machine)이라는 특수한 가상 머신이 필요한데, 이 가상 머신은 자바 바이트코드를 어느 플랫폼에서나 동일한 형태로 실행시킨다. [7]

## C. SQL



그림 10. SQL Server

SQL(Structured Query Language)는 관계형 데이터베이스 관리 시스템(RDBMS)의 데이터를 관리하기 위해 설계된 특수 목적의 프로그래밍 언어이다. 관계형 데이터베이스 관리 시스템에서 자료의 검색과 관리, 데이터베이스 스키마 생성과 수정, 데이터베이스 객체 접근 조정 관리를 위해 고안되었다. SQL은 데이터베이스로부터 정보를 얻거나 갱신하기 위한 표준 대화식 프로그래밍 언어이다. 많은 수의 데이터베이스 관련 프로그램들이 SQL을 표준으로 채택하고 있다.[9]

## 11.4. Application Framework

### A. ASP.net



그림 11. ASP.net

**ASP.NET**은 마이크로소프트사가 개발하여 판매하는 웹 애플리케이션 프레임워크이며 프로그래머들이 동적인 웹 사이트, 웹 애플리케이션, 웹 서비스를 만들 수 있게 도와 준다. 2002년 1월에 닷넷 프레임워크 버전 1.0과 함께 처음 출시되었으며 마이크로소프트의 액티브 서버 페이지 (ASP) 기술의 뒤를 잇는다. ASP.NET은 공통 언어 런타임 (CLR)을 기반으로 작성되며 프로그래머들이 닷넷 언어가 사용된 ASP.NET 코드를 기록할 수 있게 도와 준다. ASP.NETSOAP 확장 프레임워크는 ASP.NET 구성 요소가 SOAP 메시지를 처리할 수 있게 도와 준다. [8]

# 12. Index

---

## 12.1 Figure Index

### 1. Preface

### 2. Introduction

그림 1. I-campus 를 이용하는 학생들의 설문결과(google 설문지)

그림 2. HAS 시스템의 전체 구조

### 3. Glossary

### 4. System Architecture

그림 3. System Application Overview

그림 4. 카카오톡 옐로아이디

### 5. User Activity

### 6. Prof. Log-in Approval Subsystem

### 7. Information Administrator Subsystem

### 8. Set H/W Alarm Subsystem

### 9. Send H/W Alarm Subsystem

### 10 Test

그림 5. Test of HAS

### 11. Development Environment

그림 6. Window7

그림 7. Visual Studio

그림 8. C#

그림 9. Java

그림 10. SQL Server

그림 11. Asp.net

## 13.2 Diagram Index

### 1. Preface

### 2. Introduction

### 3. Glossary

### 4. System Architecture

다이어그램 1. System Architecture Overview

다이어그램 2. System Application Overview

다이어그램 3. Prof. Log-in approval subsystem Overview

다이어그램 4. Information Administrator Overview

다이어그램 5. Set H/W Alarm Overview

다이어그램 6. Send H/W Alarm Overview

다이어그램 7. NETWORK 접속 구조도

다이어그램 8. NETWORK 접속 구조도

다이어그램 9. Database 구조도

### 5. User Activity

다이어그램 10. HAS class diagram (1)

다이어그램 11. HAS class diagram (2)

다이어그램 12. Professor activity 모델

다이어그램 13. Professor activity 모델

다이어그램 14. Course activity 모델

다이어그램 15. Student activity 모델

다이어그램 16. HAS activity 모델

다이어그램 17. Database activity 모델

다이어그램 18. Overall Sequence Diagram

### 6. Prof. Log-in Approval Subsystem

다이어그램 19. Log-in Approval sub-system Class Diagram

다이어그램 20. Log-in Approval sub-system Sequence Diagram

다이어그램 21. Log-in Approval sub-system State Diagram

### 7. Information Administrator Subsystem

다이어그램 22. . Information Administrator sub-system Class Diagram

다이어그램 23. . Information Administrator sub-system Sequence Diagram

다이어그램 24. . Information Administrator sub-system State Diagram

### 8. Set H/W Alarm Subsystem

다이어그램 25. Set H/W Alarm sub-system Class Diagram

다이어그램 26. Set H/W Alarm sub-system Sequence Diagram

다이어그램 27. Set H/W Alarm sub-system State Diagram

9. Send H/W Alarm Subsystem

다이어그램 28. Send H/W Alarm sub-system Class Diagram

다이어그램 29. Send H/W Alarm sub-system Sequence Diagram

다이어그램 30. Send H/W Alarm sub-system State Diagram

10. Test

11. Development Environment

## 13.3 Table Index

1. Preface

2. Introduction

3. Glossary

표 1. Glossary

4. System Architecture

5. User Activity

6. Prof. Log-in Approval Subsystem

표 2. Log-in Approval sub-system Class Diagram

7. Information Administrator Subsystem

표 3. Information Administrator sub-system Class Diagram

8. Set H/W Alarm Subsystem

표 4. Set H/W Alarm sub-system Class Diagram

9. Send H/W Alarm Subsystem

표 5. Send H/W Alarm sub-system Class Diagram

10. Test

11. Development Environment

# 13. Appendices

이 섹션에서는 개발되는 시스템에 대해 구체적인 정보들을 제공한다. 개발에 적용된 프로세스 등에 대해 설명한다.

## 13.1. 적용 개발 프로세스

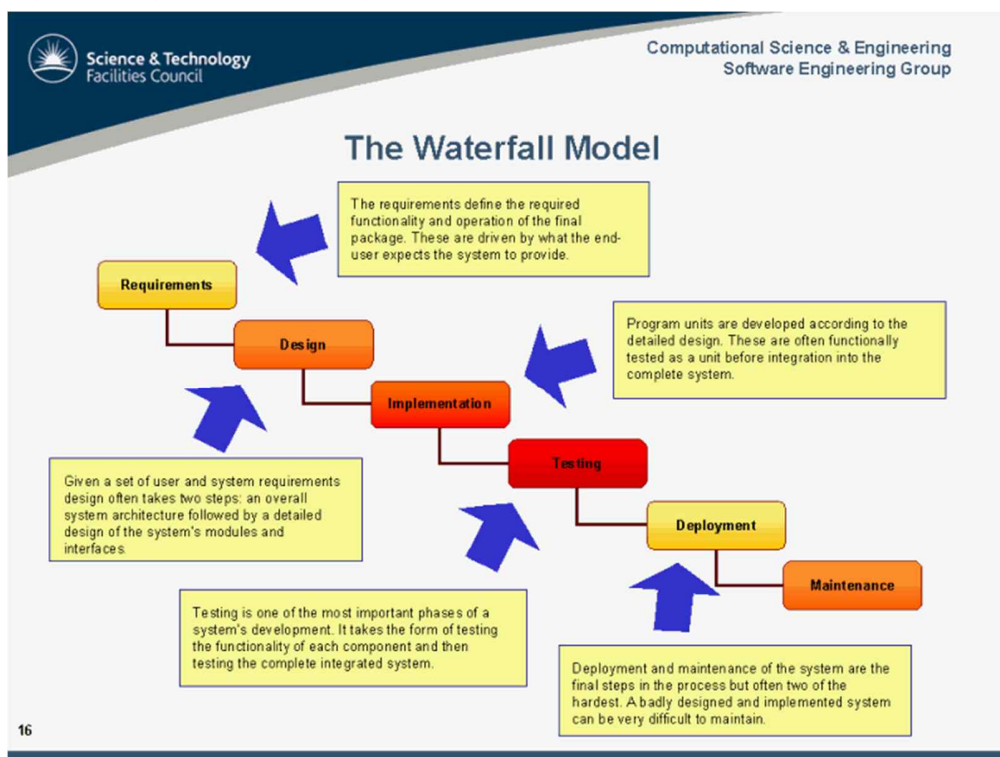


그림 11. Waterfall model

시스템의 개발 모델은 Waterfall model으로 한다. HAS 시스템의 특성상 유지 및 보수가 중요한 부분이기 에 이를 좀 더 용이하게 하기 위하여 문서가 자세히 작성할 필요가 있었습니다. 그리고 HAS 시스템은 sequential 구조로써 절차적으로 이루어 져야 하는 부분이기 때문에 Waterfall model을 선택하게 되었다.

Waterfall model은 계획 기반 모델로써 소프트웨어 프로세스의 각 단계를 나누고 구별한 모델로, 모든 각 각의 단계에서는 산출물이 명확하고, 또한 문서가 자세하고 정확히 작성된다. 그리고 병렬적 개발이 가능 하기 때문에 일반적으로 많이 쓰이는 모델이며, 대규모 프로젝트에서 많이 쓰인다. [1]



# References

- [1] "Waterfall",  
<http://www.softeng.rl.ac.uk/st/archive/SoftEng/SESP/Presentations/SoftwareEngineeringforCSED/sld016.htm>
- [2] "Database", <https://en.wikipedia.org/wiki/Database>
- [3] "sql", <https://azure.microsoft.com/ko-kr/services/sql>
- [4] "옐로우아이디", <https://yellowid.kakao.com/login>
- [5] "인터페이스",  
<https://ko.wikipedia.org/wiki/%EC%9D%B8%ED%84%B0%ED%8E%98%EC%9D%B4%EC%8A%A4>
- [6] "C#", <https://namu.wiki/w/C%2>
- [7] "JAVA",  
[https://ko.wikipedia.org/wiki/%EC%9E%90%EB%B0%94\\_\(%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D\\_%EC%96%B8%EC%96%B4\)](https://ko.wikipedia.org/wiki/%EC%9E%90%EB%B0%94_(%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D_%EC%96%B8%EC%96%B4))
- [8] "ASP.net", <https://ko.wikipedia.org/wiki/ASP.NET>
- [9] "SQL" , <https://ko.wikipedia.org/wiki/SQL>