

요구사항 명세서

Homework Alert System(HAS)

10 조

최가람, 라종휘, 김종혁, 배수민, 이기웅

내용

1. Preface.....	4
1.1. Document Structure.....	4
1.2 Readership.....	4
1.3. Version of the Document.....	5
2. Introduction	7
2.1. 배경 및 필요성	7
2.2. Homework Alert System (HAS)	8
2.3. 기대 효과	9
3. Glossary.....	10
4. User Requirements Definition.....	11
4.1. Functional Requirements	11
4.2. Non-functional Requirements	11
5. System Architecture	15
5.1. 전체 시스템 구조	15
5.2. HAS 접속 시스템	16
5.3. 전체 네트워크 구조.....	17
5.4. 데이터베이스 구조	18
6. System Requirements Specification	19
6.1. 기능적 요구사항.....	19
6.2. 비기능적 요구사항	20
6.3. 시나리오.....	23
7. System Models	26
7.1. Context models.....	26
7.2. Interaction models.....	28

8. System Evolution	39
8.1. Hypothesis of System.....	39
8.2. Evolution of System	39
9. Index.....	40
9.1 Figure Index.....	40
9.2 Diagram Index	41
9.3 Table Index.....	42
10. Appendices	43
10.1. 데이터베이스 요구사항	43
10.2. 사용자 시스템 요구사항	43
References.....	45

1. Preface

Preface 에서는 본 문서의 대한 개략적인 부분과 각 부분의 역할에 대하여서 설명한다. 그리고 독자가 누구인지를 밝히고, 각 챕터의 역할을 소개한다.

1.1. Document Structure

이 문서에서는 총 10 개의 부분으로 구성되어 있다. Preface, introduction, Glossary, User Requirements Definition, System Architecture, System Requirements Specification, System Models, System Evolution, Appendices, Index 으로 구성되어 있다. 각 부분의 역할은 다음과 같다.

1.2 Readership

본 문서는 크게 User requirements 와 System requirements 두 부분으로 구성되어 있다. User requirements 는 사용자의 관점에서 요구사항을 간략히 명세한 것이고, System requirements 는 시스템이 제공하는 기능을 더욱 자세하게 설명한 것이다. 이는 I-campus 관리자 및 시스템 개발자를 위한 설명이다.

A. User requirements readership

본 문서의 User requirements 는 고객들의 주요 요구사항을 개발자와 관리자가 개발 도중 참고하기 위해 작성한 문서이다. User requirements 는 고객들의 요구사항을 정리한 것 이므로, User requirements 의 주요 독자는 고객층이 아닌 I-campus managers, 개발자이다.

B. System requirements readership

System requirements 의 주요 독자는 해당 시스템을 개발할 개발자들이다. 그러하기 때문에 system requirements 에서는 해당 시스템의 특정 기능이 어떻게 구현될 지에 대한 자세한 내용이 설명되어 있다. Software developer 외에도 client engineers, system architects, 등 경우에 따라서는 System end-users 까지 독자가 될 수 있다.

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	2015-04-22	문서의 초안과 구조 작성
1.0	2015-04-28	Preface, Glossary, Requirements 작성
1.0	2015-04-28	Introduction 작성 및 문서 skeleton 구축
2.0	2015-04-30	System Architecture, Requirement 작성
2.1	2015-05-05	System Architecture 수정
2.1	2015-05-05	Functional, non-functional 부분 추가
2.1.1	2015-05-05	Requirement 부분 오타 수정
3.0	2015-05-11	System model, context model 작성
3.1	2015-05-16	Modeling 부분 수정
3.2	2015-05-16	Requirements 신규 부분 추가
4.0	2015-05-18	User case 작성
4.0	2015-05-18	Structure model, behavioral model 작성
5.0	2015-05-20	Scenario, Sequence diagram 작성
5.0	2015-05-20	Class diagram 작성
5.1	2015-05-23	각 diagram 취합 및 정리
5.1	2015-05-23	문서 index 작성
6.0	2015-05-25	System evolution 구상
6.0	2015-05-25	System architecture 작성
6.0	2015-05-25	신규 diagram 추가
7.0	2015-05-27	System evolution 작성 및 수정
7.0	2015-05-27	Readership 작성 및 추가
7.0	2015-05-27	예시 문서와 비교 및 확인
8.0	2015-05-28	Appendices, Reference 작성
8.0.1	2015-05-28	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 설문 조사한 결과에 따르면, 매 학기 동안 교수님이 I-campus 에 과제를 늦게 업로드 해주어서 과제를 정해진 시간 내에 제출하지 못하여 불이익을 받은 경험을 가지고 있다고 조사되었다. 일부 학생들은 I-campus 에 일주일에 5 번 이상을 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 학생들이 성적에 상당히 민감하기 때문에 이러한 문제는 간과할 수 없다.

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)

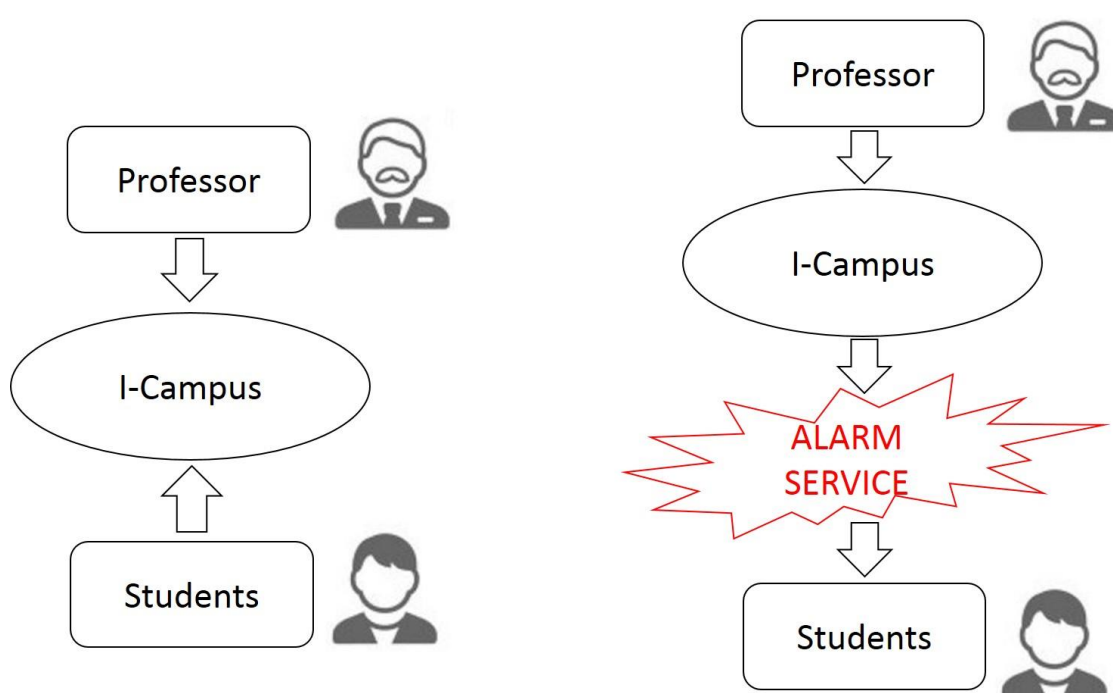
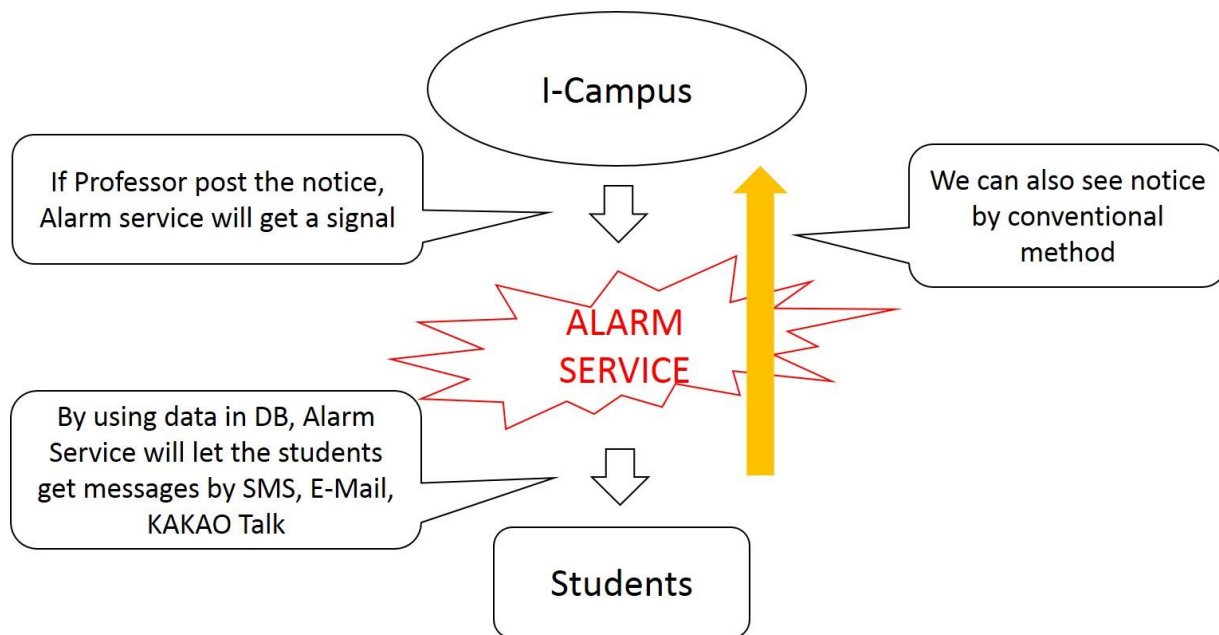


그림 2. HAS 시스템의 전체 구조

HAS(Homework Alert System)은 교수님과 학생들의 자료 및 정보 공유를 통하여 교수님이 과제, 프로젝트, 숙제를 I-Campus에 올릴 시에 즉시 HAS 통하여서 스마트폰으로 메시지를 보내 주어 학생들이 공지사항을 늦게 확인 하지 않고, 빠르게 확인할 수 있는 시스템이다. <그림 2>는 HAS 시스템의 구성요소와 I-Campus, 학생, 교수가 서로간의 Interface를 통하여 송신과 수신하는 시스템이다. HAS에 교수님이 학과목을 생성하고, 그 안에 학생들이 본인의 정보를 입력한다. 이 정보들을 통하여 HAS는 학생들이 가지고 있는 스마트폰으로 즉시 메시지를 전송해 주어 공지사항을 확인할 수 있다. HAS는 교수님으로부터 접근 신호를 받으면 DB에 교수님의 정보가 일치하는지를 확인한다. DB에서 교수님 접근 권한이 있다는 정보를 받으면 HAS는 교수님께 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. 교수님이 과제 정보를 HAS에 업로드 하면 HAS는 DB로부터 과제를 전송할 학생 정보를 받고 학생에게 과제가 생겼다는 신호와 과제에 대한 정보를 전송한다.

* I-Campus는 교수의 정보와 학생들의 정보를 가지고 있는 System이다. 그리하여 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 I-Campus와 연동하여 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 있다. 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 입장에서도 매우 편리하다.

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#	C++ 와 자바 에서 강한 영향을 받았다. 이름의 유래는 두 가지에서 따왔다고 한다. # 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. User Requirements Definition

이 섹션에서는 USER 에게 어떤 서비스를 제공하는지에 대해 설명한다. 이러한 Functional System Requirement 이외에도 Nonfunctional system requirement 에 대해서도 설명한다. 이를 위해 자연어, 다이어그램 또는 유저가 이해할 수 있는 다른 개념들을 사용할 것이다. 이렇게 설명된 내용은 HAS 시스템에 충실히 반영된다.

4.1. Functional Requirements

HAS 시스템은 즉각적인 정보제공을 목적으로 한다. HAS 에서는 교수님께서 I-Campus 에 공지사항을 업로드 시 모바일 디바이스(스마트폰)로 확인할 수 있는 기능을 제공한다.

A. 공지사항 확인 기능

I-Campus 에 과목에 대한 공지사항이 올라온 경우 HAS 가 전송 시켜준 메시지를 통하여 즉시 확인(Check)이 가능하다.

B. 학생에게 유용한 기능

학생들이 시간과 환경에 구애 받지 않고, 언제 어디서든 스마트폰을 이용하여 정보 확인이 가능하다.

C. 학생에게 편안함을 제공함

학생들에게 갑작스런 공지사항에 압박감을 주지 않고, HAS 를 통하여서 학생들에게 정보를 메시지로 전달될 때에 학생들은 심적 편안함을 느낄 수 있다.

4.2. Non-functional Requirements

A. Product Requirements

A.1. Usability requirements

HAS 는 학생들이 I-Campus 에 매번 접속해야 하는 번거로움을 덜어주기 위해 만들어진 시스템이다. 사용자는 학생과 교수님으로 나뉘어지는데, 학생의 경우 문자를 수동적으로 받는 입장이기 때문에 본인의 정보를 제공하는 것 이외에는 학생이 시스템에 능동적으로 참여하는 경우는 존재하지 않는다. 즉, 개발자는 능동적으로 과제를 활성화 하고 메시지 송신의 주체가 되는 사용자인 교수님을 위해 사용하기 좋은 User Interface 를 구축한다.



그림 8. User Interface (Initial Log-in page)

A.2. Dependability requirements

- Availability, Reliability

HAS 시스템의 구동면에서는 메시지를 보내지 못하는 오류가 발생해도, 기존 과제 확인 방법과 같이 직접 I-CAMPUS 게시물을 확인하는 방법으로 과제 확인이 가능하다. 하지만 시스템의 문제가 생겼음을 학생이 인지하지 못하면 I-CAMPUS 의 게시글을 확인하지 않을 가능성이 높다. 따라서 관리자의 지속적인 확인을 통해 시스템의 이상이 생겼을 경우, 학생측에 공지를 해주어 기존 방법으로 과제를 확인할 수 있게 해주어야 한다.

- Maintainability

HAS 시스템의 경우 교수님과 학생들의 정보를 관리하는 시스템과 메시지를 보내는 시스템이 분리되어 있기 때문에, 관리자의 측면에서 에러 발생시 시스템 분할 관리에 용이하다. 즉 한쪽의 에러가 다른 Sub-System 에 상대적으로 적은 영향을 끼치기 때문에, 관리자의 유지보수 활동 측면에서 장점을 가진다.

A.4. Security requirements

HAS 는 학생의 개인정보를 바탕으로, 메시지를 전송하기 때문에 데이터베이스에 저장된 정보는 보안이 필요하다. HAS 에 있는 DB 에 악의적인 방법으로 접속하여 학생들의 개인적인 정보를 빼앗아 갈 수 있다. 따라서, 학생들과 교수님들 그리고 관리자를 제외한 사람들이 이 서버에 접속하지 않을 수 있도록 ID 와 PW 를 통해 system 접근을 제한하고, 주기적인 데이터베이스 접근 로그를 확인하여, 학생들의 개인정보가 유출되었는지의 여부를 확인해야 한다.

A.5. Performance requirements

HAS 의 경우 performance 측면에서 강한 요구사항은 없지만, 많은 사용자가 동시접속하여 동시에 메시지 전송을 요구할 경우, 대기시간이 길어질 수 있다. 이러한 요구사항에 따라 본 시스템은 다음과 같은 기준을 만족시키도록 설계한다.

(기준: 1000 개의 메시지 queue 를 5 분안에 처리한다.)

B. Organizational Requirements

B.1. Environmental Requirements

- Developer

개발자의 경우 주요 코드들이 모두 JAVA 와 C#으로 이루어지기 때문에 각 언어를 컴파일할 수 있는 PC 환경이 필요하다.

- Students

학생들의 경우 notification 을 받기 위해 카카오톡이 설치된 모바일기기가 필요하다.

- Professor

교수님들의 경우 HAS 에 접속하기 위한 개인 PC 가 필요하다.

B.2. Operational Requirements

시스템을 사용하는 교수님들과 학생들이 장소와 시간에 구애 받지 않고 모두 용이하게 접속하기 위해 모든 시스템은 Web-Based 로 구현된다.

B.3. Development Requirements

HAS 와 I-Campus 사이에서 데이터를 주고 받을 수 있는 표준화된 환경이 필요하다. 그리고 홈페이지를 구축해야 하기 때문에 HTML 이 필요하며, DB 구축 및 C#을 기반한 환경이 필요하다.

C. External Requirements

C.1. Business Requirements

카톡 엘로아아디는 고객과 커뮤니케이션을 할 수 있는 비즈니스용 카카오톡 아이디이다. 우리가 만들 HAS 를 여기에 연계시켜 메시지를 받을 수 있도록 승인이 필요하다.



그림 13. 카카오톡 엘로아아디

C.2. Safety Requirements

데이터 수집 및 데이터를 저장함에 있어서 전송 및 받아들이는 과정에서 스니핑과 같은 중간에 데이터가 다른 제 3 자에게 빠져나가는 것이 있어서는 안된다. 휴대용 디바이스와 WIFI 그리고 HAS 를 사용하는 중간에 Blocking 하는 행위가 개입되지 않게 하고 통신은 모두 암호화 하여 전송 수신 할 수 있도록 설계한다. 또한 이러한 외부적 해킹에 대비하여 개인정보는 최소한으로 수집한다.

C.3. Legal Requirements

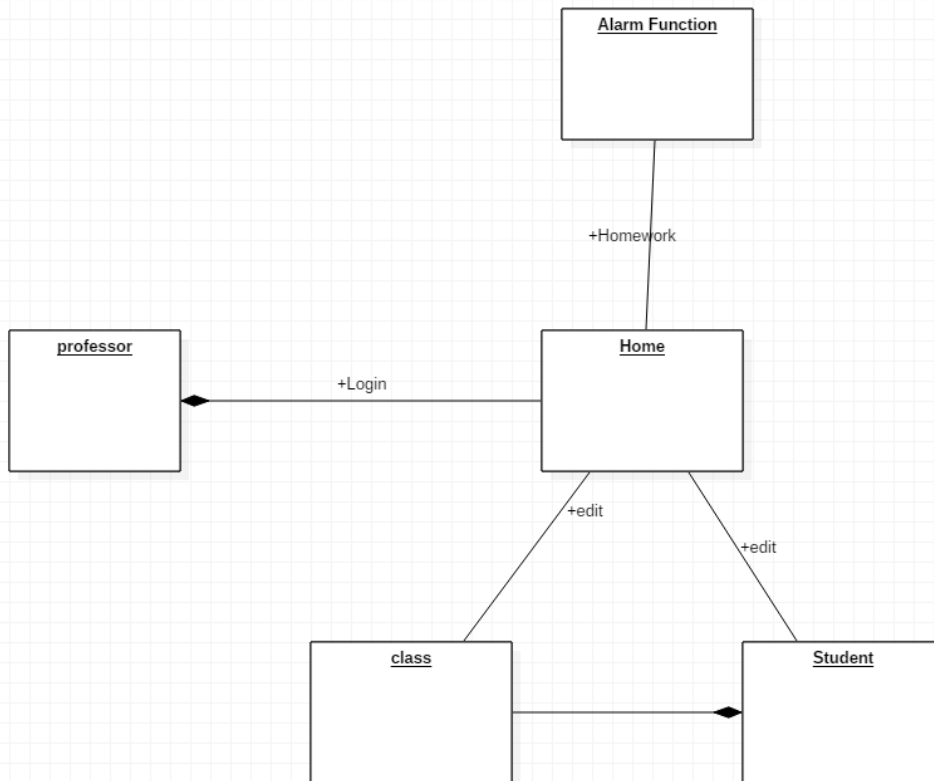
HAS 의 데이터 베이스에는 교수님의 ID, Password 와 학생들의 카카오톡 아이디, 학생정보들이 모두 포함되어 있기 때문에 그들의 정보를 이용하기위해 서비스 이용 동의를 받을 필요가 있다.

5. System Architecture

System Architecture에서는 HAS 시스템의 전체적인 구조 개요를 설명한다.

5.1. 전체 시스템 구조

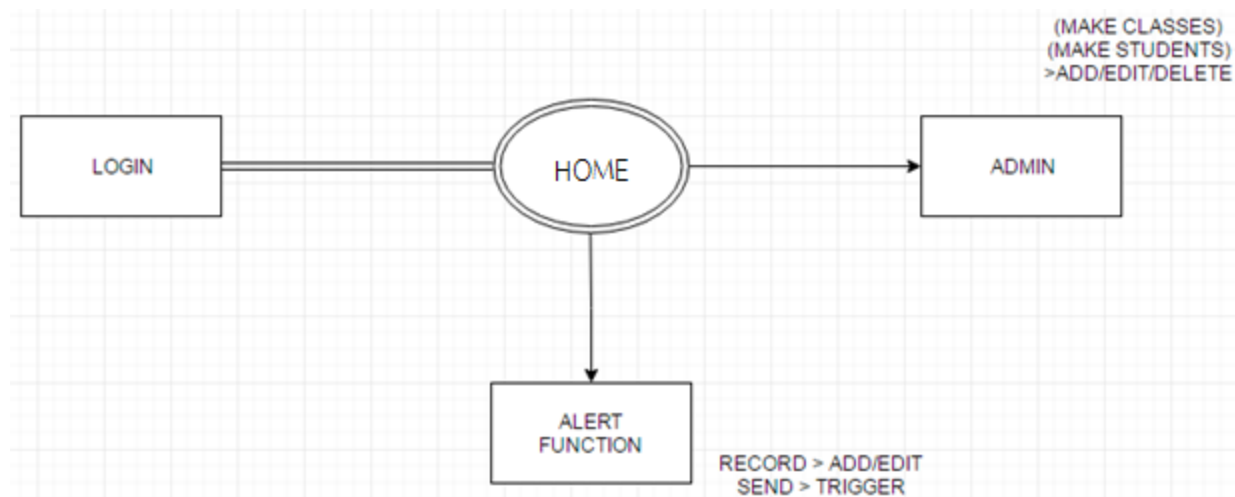
HAS 시스템에 포함된 전체적인 구조를 표현하면 다음과 같다.



다이어그램 1. HAS 시스템의 전체 구조도

교수님께서 과제를 생성하여 HAS System에 업로드 시 HAS는 Database로부터, 해당 강의를 수강하는 학생들의 정보를 받아, 학생들에게 과제의 정보를 전송한다.

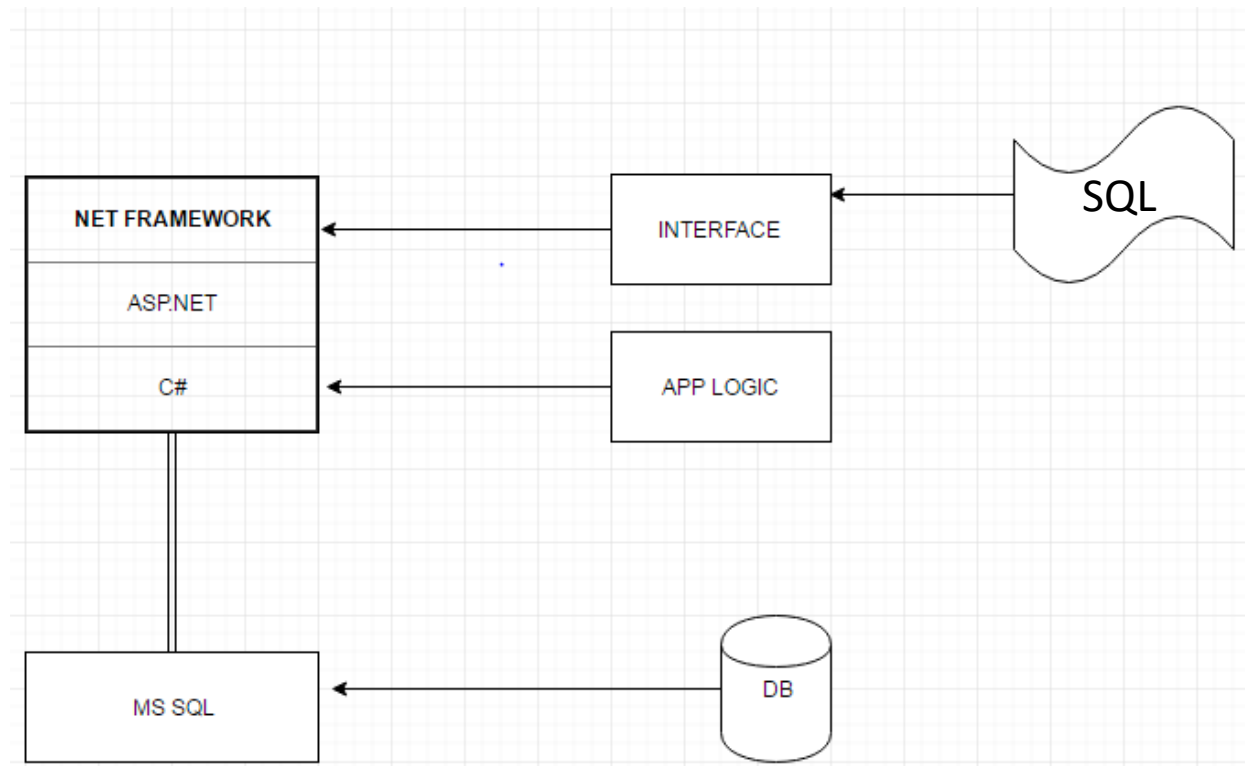
5.2. HAS 접속 시스템



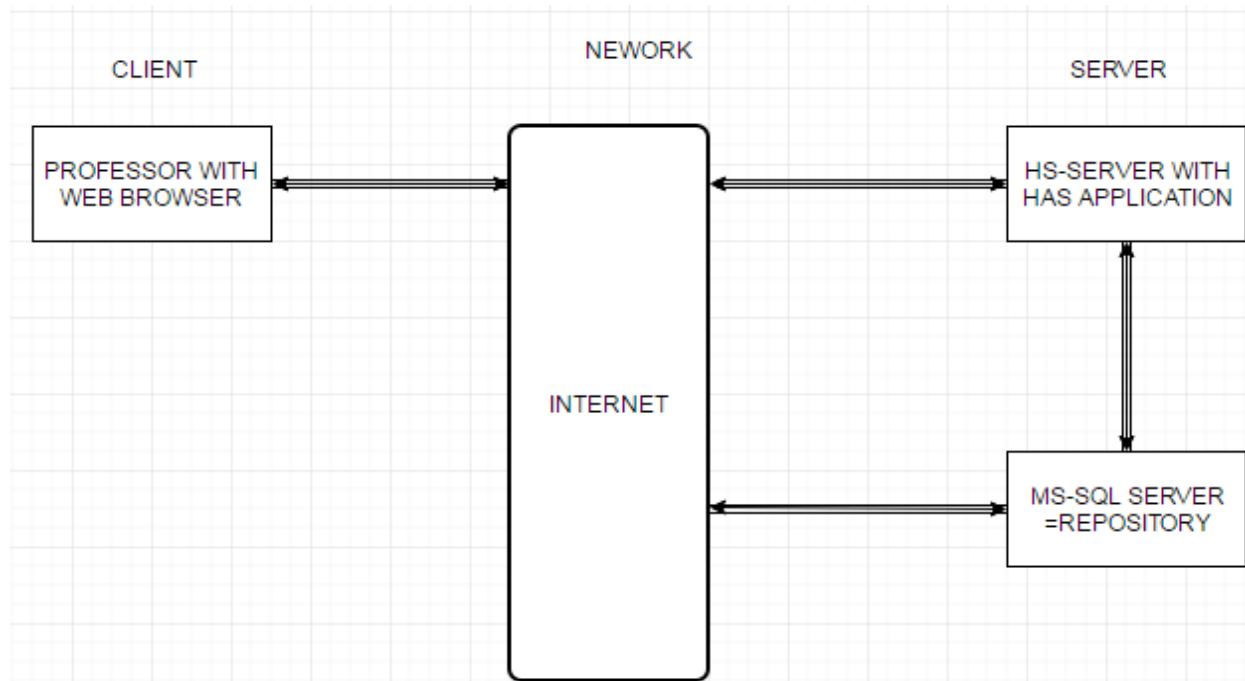
다이어그램 2. HAS 접속 구조도

교수님이 HAS 에 접속할 때, HAS 는 교수님의 ID, PW 를 요구하고, 해당 사용자가 접근 가능한 사용자인지, ADMIN 을 통해 확인한다.

5.3. 전체 네트워크 구조

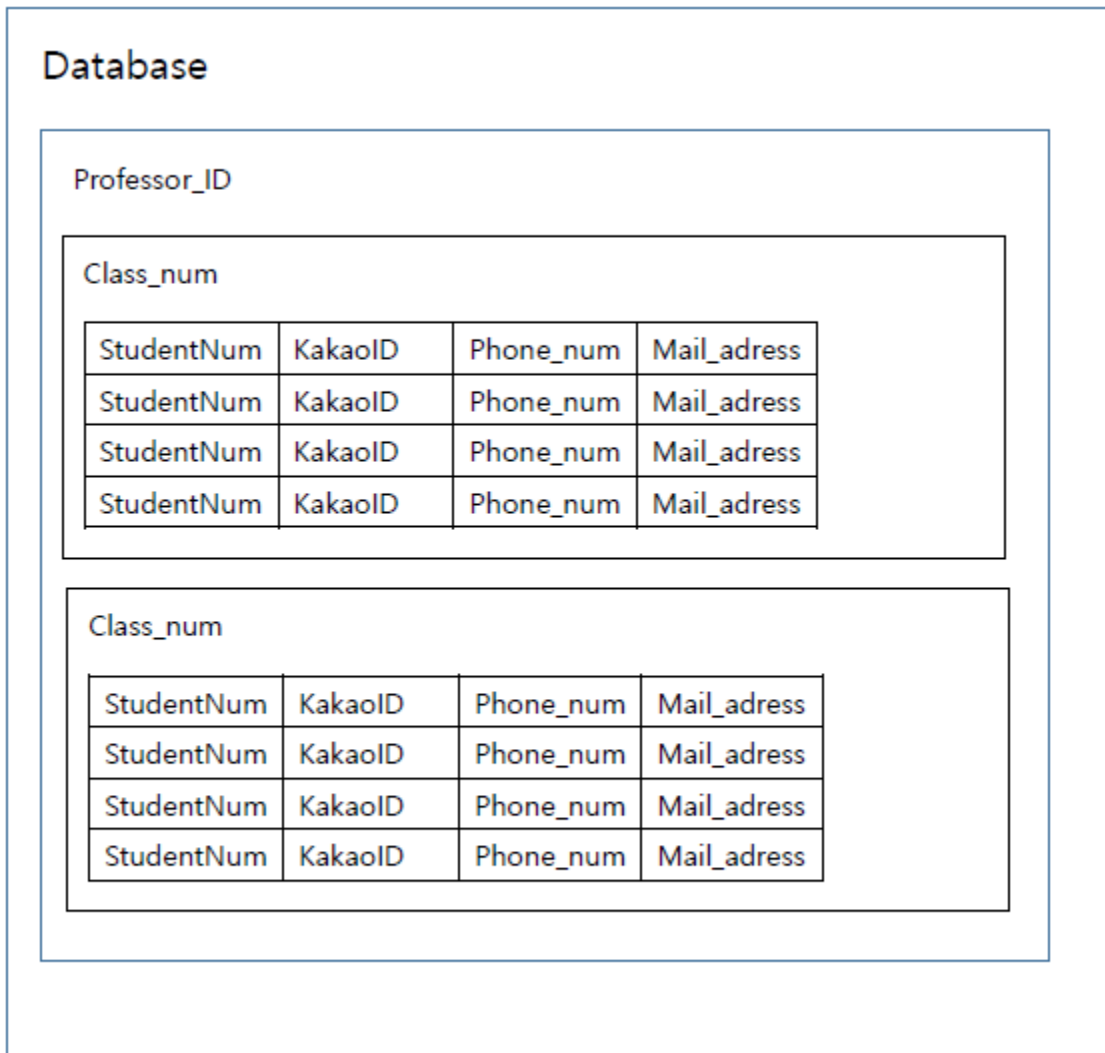


다이어그램 3. NETWORK 접속 구조도



다이어그램 4. NETWORK 접속 구조도

5.4. 데이터베이스 구조



Database에는 각 Professor 별로 Class_num가 할당되어 있고, 각 Class 별로, 수강학생의 학번, 카카오톡 ID, 휴대폰 번호, 메일주소가 저장되어 있다.

6. System Requirements Specification

이 섹션에서는 기능적 요구사항, 비기능적 요구사항에 대해 설명하고 사용자가 사용하는 시나리오를 묘사한다.

6.1. 기능적 요구사항

HAS 시스템은 다음과 같은 기능을 제공해야 한다.

A. 학생들에게 메시지 전송 기능

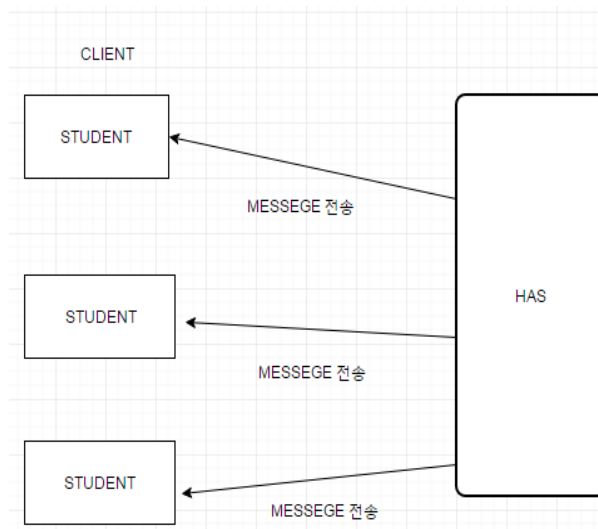


그림 14. HAS 를 통해서 메시지 전달하는 시스템

표 2. 사용자 외출 감지 후 자동 전력 제어 기능 요구사항

설명	교수님이 HAS 를 통해 과제를 전송할 시, 해당 수업에 등록된 학생들에게 메시지를 전송한다.
입력	HAS 를 통한 과제 정보 입력
출력	카카오톡 메시지
처리	과제에 대한 정보를 HAS 를 통해 카카오 옐로아이디로 전송, 옐로아이디를 통해 학생들에게 메시지 전송
조건	과제가 업로드 될 시 즉시 학생들의 휴대폰으로 메시지 전달되기 위해 학생의 스마트폰의 데이터통신이 가능해야 하며, 인터페이스가 잘 유지되어야 한다.

6.2. 비기능적 요구사항

Non-functional requirement 는 크게 Product requirements, Organizational requirements, External requirements 로 나뉜다.

A. Product requirements

A.1. Usability requirements

HAS 의 대상 고객층은 대학교에 있는 교수님들과, I-campus 관리자들, 강의를 듣는 학생들이다. 기존 시스템에 익숙해진 사용자들이 새로운 시스템을 사용하기 위해서는 별도의 매뉴얼 없이 시스템을 사용할 수 있도록 자세하고 쉽게 작성되어야 한다. 따라서 이 시스템은 사용자가 직관적으로 알아보기 쉬운 UI 를 제공하여야 한다.

A.2. Performance requirements

HAS 는 학생이 교수님이 내주신 과제를 빠르게 확인하는 것이 목적이다. HAS 는 네트워크를 이용하여, 교수님이 HAS 를 통해 과제를 업로드하는 즉시 학생이 과제를 확인할 수 있기 때문에, performance requirement 를 만족한다.

A.3. Space requirements

HAS 는 서버, 인터넷통신, 휴대용 모바일 디바이스로 구성된다. 실질적으로 사용자가 HAS 시스템을 사용하기 위해 필요로 되는 디바이스는 인터넷 접속이 가능한 디바이스다. 스마트폰의 보급으로 모바일 디바이스로도 쉽게 인터넷 접속이 가능하고, 메시지를 받는 수단 또한, 스마트폰이기 때문에, HAS 시스템은 시공간에 제약 없이 언제든지 사용할 수 있다.

A.4. Dependability requirements

HAS 시스템의 경우 교수님과 학생들의 정보를 관리하는 시스템과 메시지를 보내는 시스템이 분리되어 있기 때문에, 관리자의 측면에서 에러 발생시 시스템 분할 관리에 용이하다. 즉 한쪽의 에러가 다른 Sub-System 에 상대적으로 적은 영향을 끼치기 때문에, 관리자의 유지보수 활동 측면에서 장점을 가진다.

A.5. Security requirements

HAS 시스템은 데이터에 접근가능한 사용자를 교수님으로 제한하여, 사용자의 정보를 요청하고, 사용자의 정보가 입력된 정보와 일치할 경우에만 접근을 허용하는 구조로 되어있다. 또한, 데이터베이스를 수동으로 편집, 수정이 가능하도록 설계되어 있어, 시스템의 Security requirement 가 확보된다.

B. Organizational requirements

B.1. Environmental requirements

HAS 를 사용하기 위해서는 HAS 에 접근하는 디바이스와 HAS 를 통해 과제의 정보를 받는 데이터통신이 가능해야 한다. 스마트폰의 보급으로 네트워크 접속에 공간적 제약을 받지 않기 때문에, HAS 는 Environmental requirements 를 만족한다.

또한 과제 정보를 보내는 external system 인 카카오톡은 대부분의 스마트폰 유저에게 보급되어 있고, 디바이스의 종류에 관계없이 쉽게 설치할 수 있어 Environmental requirements 를 만족한다.

B.2. Operational requirements

시스템을 사용하는 교수님들과 학생들이 장소와 시간에 구애 받지 않고 모두 용이하게 접속하기 위해 모든 시스템은 Web-Based 로 구현된다. 따라서, HAS 는 Low level 의 Operational environment 를 요구한다.

B.3. Development requirements

HAS 와 I-Campus 사이에서 데이터를 주고 받을 수 있는 표준화된 환경이 필요하다. 그리고 홈페이지를 구축해야 하기 때문에 HTML 이 필요하며, DB 구축 및 C#을 기반한 환경이 필요하다.

C. External requirements

C.1. Legal requirements

학생의 개인정보를 데이터베이스에 저장하기 때문에, 학생들의 개인정보 사용 동의가 있어야 한다.

C.2. Safety requirements

HAS 는 1000 개의 메시지 queue 를 5 분안에 처리하는 기준으로 만들기 때문에, 예상치 못한 대량의 데이터가 들어와도 순차적으로 분할하여 데이터를 처리하여, system 의 safety 가 높다.

6.3. 시나리오

A. 시스템 가동 시나리오

HAS 는 접속하는 사용자의 ID 와 PW 를 요구한다. 사용자가 HAS 에 등록된 사용자라는 인증과정이 완료되면, HAS 는 사용자에게 할당된 Class 별로 데이터를 수정할 수 있는 버튼과 과제를 작성하여, 학생들에게 보낼 수 있는 UI 를 제공한다.

B. 시스템 접근 시나리오

B.1. Normal flow

사용자가 처음 홈페이지에 들어가게 되면 아이디와 비밀번호를 입력받는 화면이 출력되고, 정해진 아이디와 비밀번호를 입력하게 되면 수강과목과 학번, 이름을 입력 받을 수 있다.

B.2. Possible errors

사용자가 HAS 에서 비밀번호를 잘못 입력하였다. HAS 는 안에 접속 되어 지지 않는다. 그리고 아이디와 비밀번호를 입력 받는 화면으로 돌아간다.

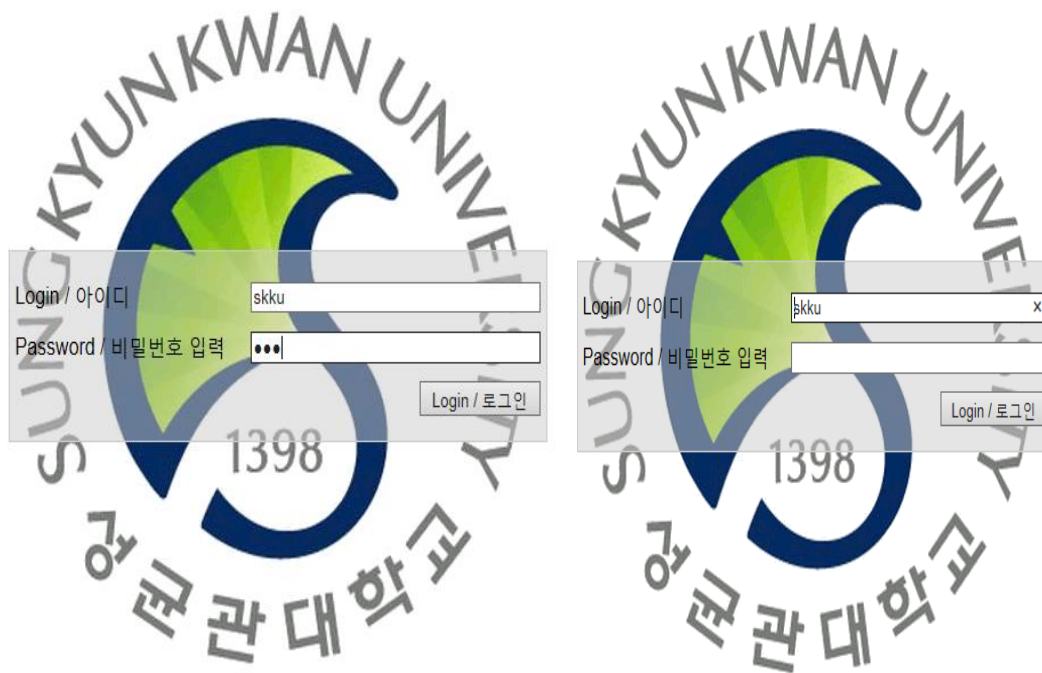


그림 15. 비밀번호를 잘못입력 하기 전과 비밀번호를 잘못입력 한 후

B.3. System state on completion

사용자가 올바른 아이디와 비밀번호를 입력하게 되면 HAS 시스템에 접속되어 진다.

C. 수업 정보 수정 시나리오

C.1. Normal flow

교수님이 과목 번호를 수정하고자 한다. HAS 시스템에 접속하여 Edit Icon 을 클릭하여 수정 및 업데이트를 할 수 있다.

ID	Class Number
1	ICE3041-41
3	ICE0000-00

Edit

id: 1

classnumber: ICE3041-41

[Update](#) [Cancel](#)

그림 16. 과목번호를 바꿀 수 있다.

C.2. Possible error

과목 번호를 바꿀 번호를 잘못 입력하였더라면, Cancel 을 통해 과목 번호를 바꾸는 것을 취소 할 수 있다.

C.3. System state on completion

과목번호 수정이 완료되면, 수정된 과목번호가 Home 화면에 표시된다.

D. 과제 전공 시나리오

D.1. Normal flow

사용자는 HAS 에 접근이 성공하면, Home 화면에는 사용자에게 할당된 과목의 목록이 과목번호로 나열된다. 사용자는 과제를 부여하려는 과목번호를 클릭하여 해당 과목의 페이지에 접속한 뒤, 과제 전송 icon 을 클릭하여 과제를 전송할 수 있다.

D.2. System state on completion

해당 과목에 할당된 학생들의 스마트폰으로 HAS 의 카카오 옐로아이디에서 메시지를 전송한다.

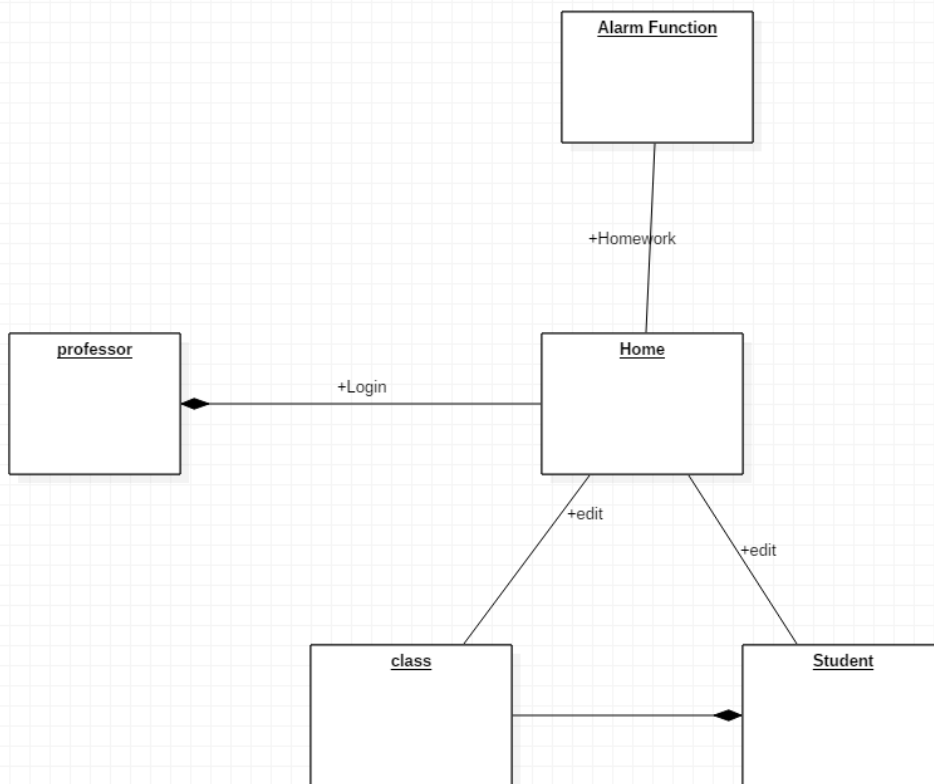
7. System Models

이 섹션에서는 시스템 컴포넌트와 시스템 환경 간의 관계를 다이어그램으로 표현한다.

7.1. Context models

A. Context model

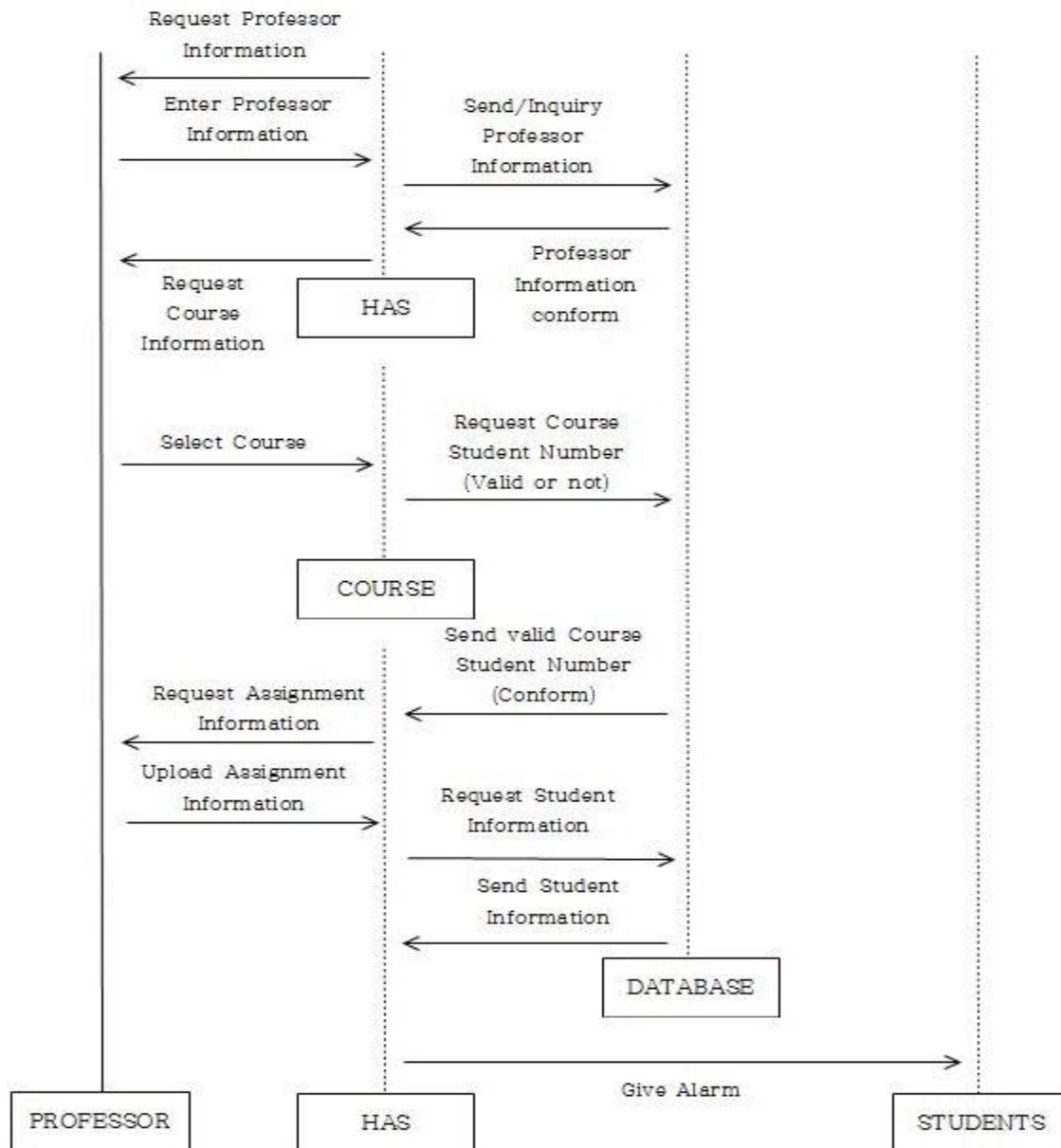
HAS 시스템은 다음과 같은 시스템 관계를 갖고 있다.



다이어그램 5. HAS 시스템 Context

B. Process diagram

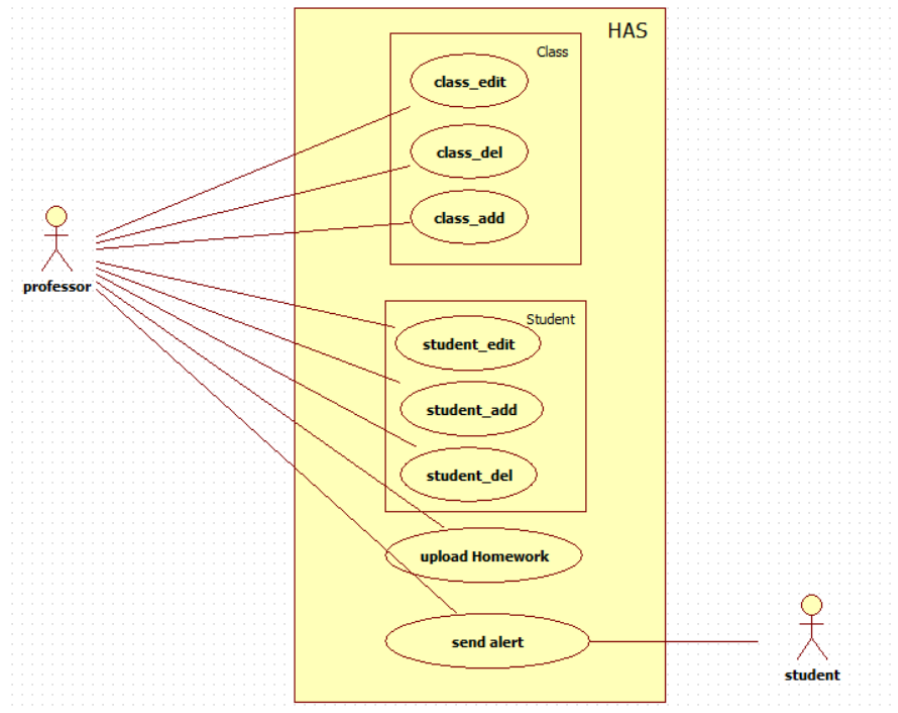
시스템의 전반적인 프로세스를 나타내기 위한 것으로 HAS 시스템은 다음과 같은 프로세스를 갖고 있다.



7.2. Interaction models

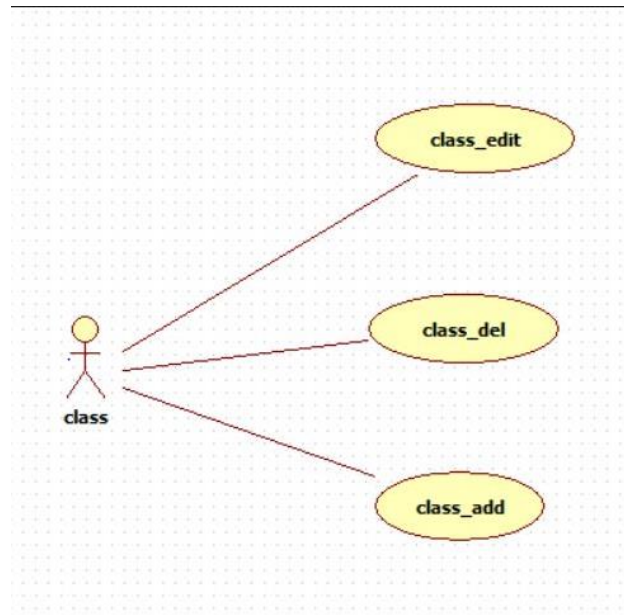
A. Use case modeling

A.1. 시스템 전체 Actor 와 use case



다이어그램 7. 시스템 전체 Actor 와 use case

A.2. class edit 상태 확인



다이어그램 8. Class edit 상태 확인 use case

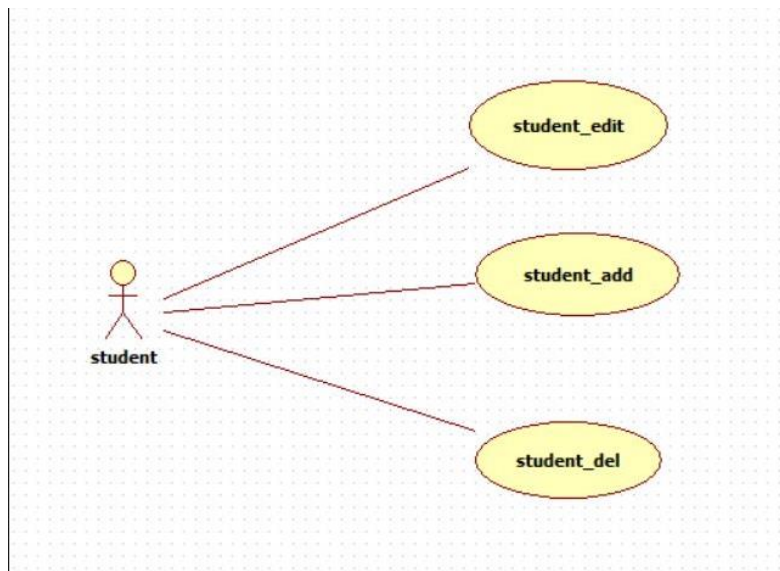
표 1. Class edit 상태 확인 use case 설명

system	Homework Alarm System
Use case	Class_edit
Actor	Professor
Description	Professor want to edit class which he will send message about homework. Then professor access to HAS and can edit Class data in Database.
Stimulus	Class data which will be fixed
Respond	change class data.

system	Homework Alarm System
Use case	Class_add
Actor	Professor
Description	Professor want to add New class which he will send message about homework. Then professor access to HAS and can edit Class data in Database.
Stimulus	New Class data
Respond	add New class and generate class data.

system	Homework Alarm System
Use case	class_del
Actor	Professor
Description	Professor want to delete class which he doesn't need anymore. Then professor access to HAS and can erase Class data in Database.
Stimulus	class data which actor want to delete
Respond	delete class data.

A.3 student edit 상태 확인



다이어그램 8. Student edit 상태 확인 use case

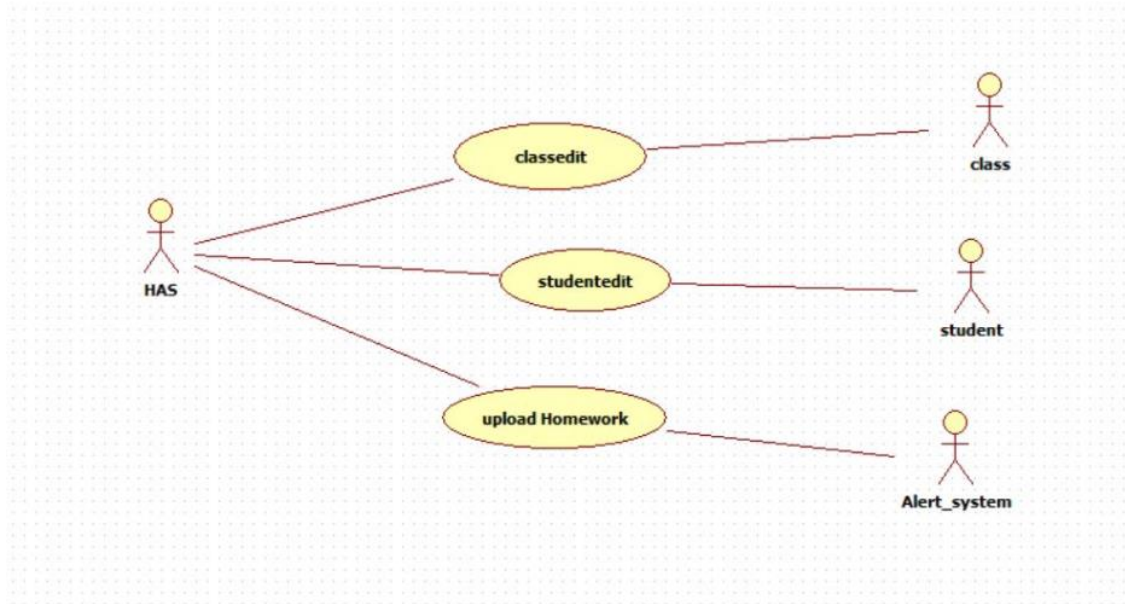
표 2. student edit 상태 확인 use case 설명

system	Homework Alarm System
Use case	student_edit
Actor	Professor
Description	Professor want to edit student which he have to send message of homework. Then, professor access to HAS and can change student's data
Stimulus	student's data which will be changed
Respond	change student's data

system	Homework Alarm System
Use case	student_add
Actor	Professor
Description	Professor want to add new student information which he have to send message of homework. Then, professor access to HAS and can add new student's data
Stimulus	student's data which will be added
Respond	register new student's data in database

system	Homework Alarm System
Use case	student_del
Actor	Professor
Description	Professor want to delete student's data which is added mistakenly. Then, professor access to HAS and can delete student's data
Stimulus	student's data which will be deleted
Respond	delete student's data

A.4 HAS edit 상태 확인



다이어그램 9. HAS edit 상태 확인 use case

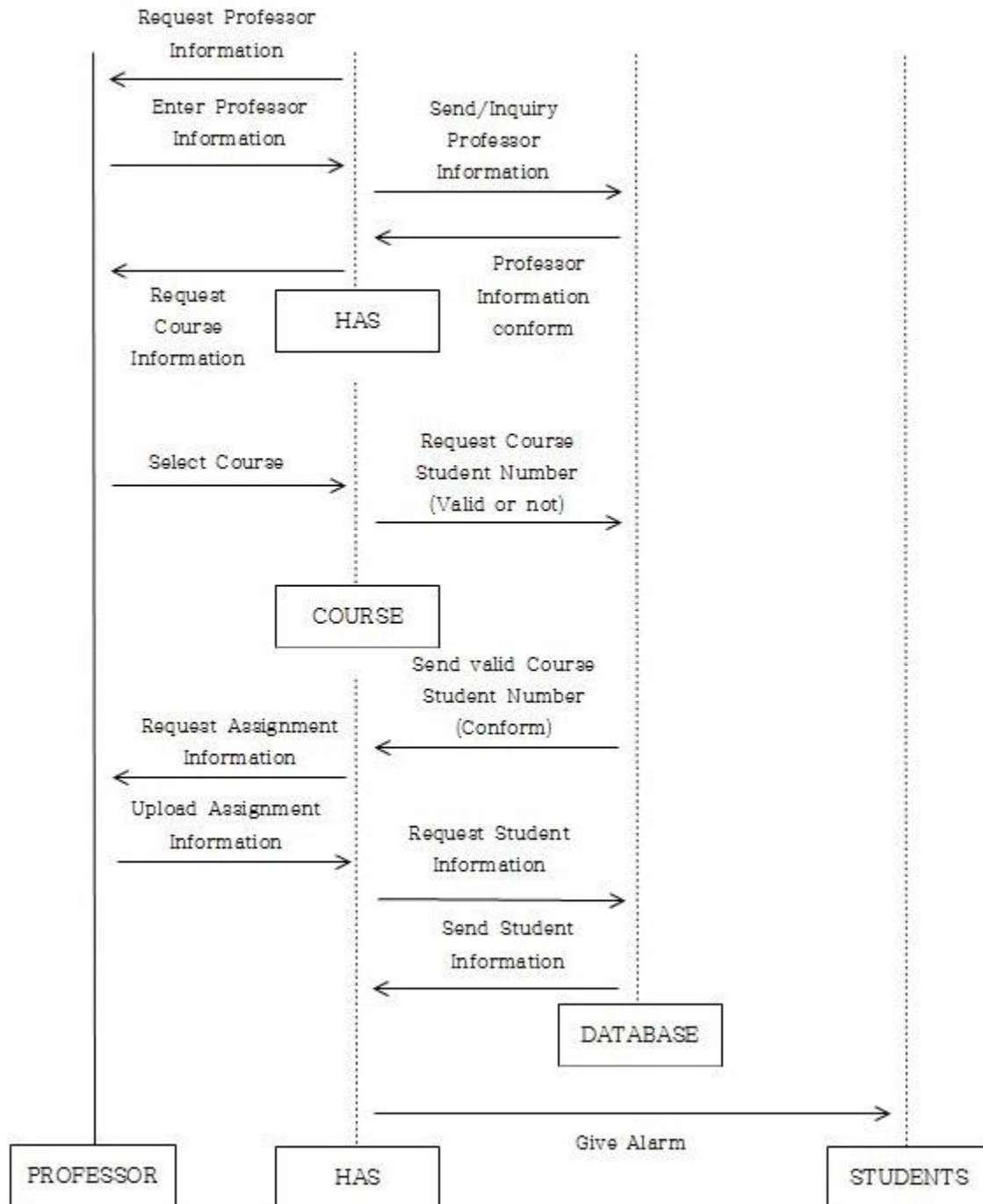
표 3. HAS edit 상태 확인 use case 설명

system	Homework Alarm System
Use case	upload_homework
Actor	professor
Description	Before professor alert student that homework is emitted, Professor can descript information of Homework by HAS
Stimulus	information of descript information
Respond	

system	Homework Alarm System
Use case	send_Alert
Actor	professor
Description	When professor want to notice student that homework is emitted, professor can send message to them by using HAS
Stimulus	signal generated when professor push alert button
Respond	massage will be sent to student

B. Sequence Diagram

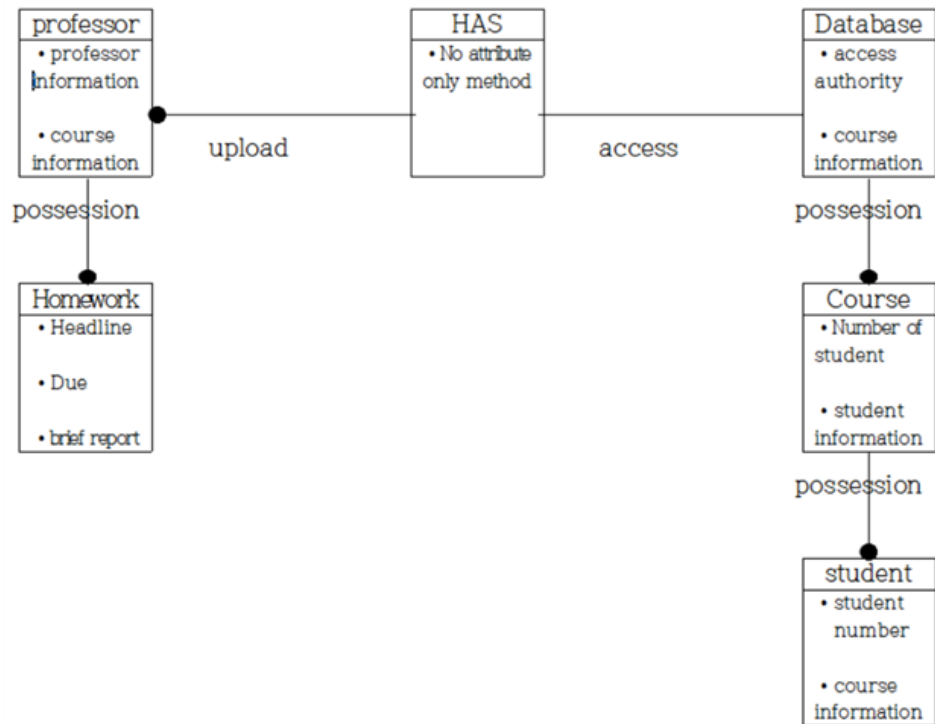
B.1. HAS 등록 시퀀스 다이어그램



다이어그램 10. HAS 등록 sequence diagram

C. Structural models

C.1. 서버 클래스 다이어그램

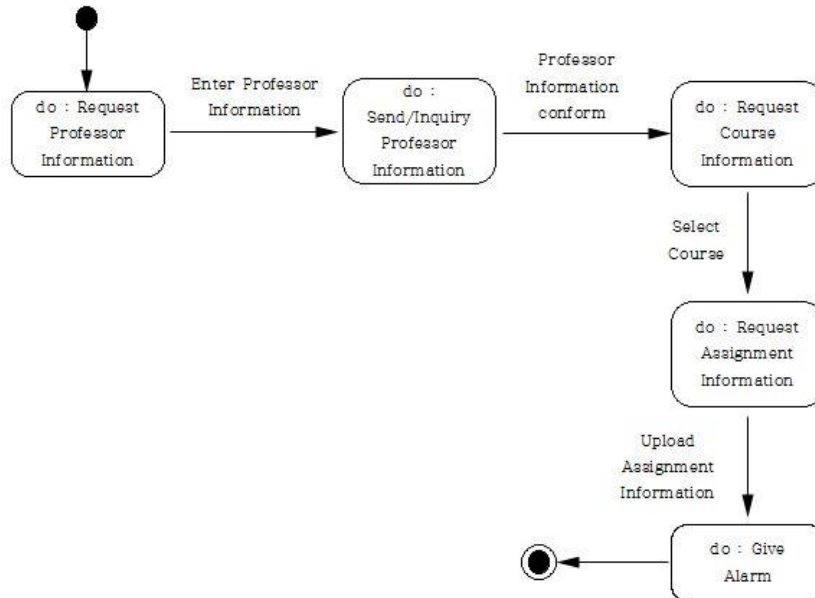


다이어그램 11. HAS class diagram

D. Behavioral models

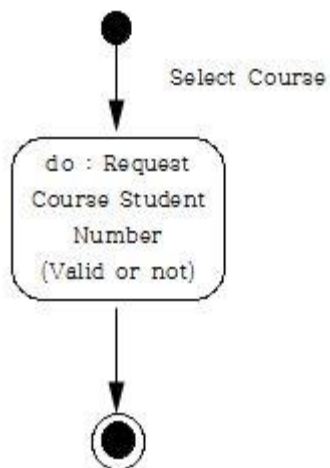
D.1. Data-driven modeling

1) Professor activity 모델



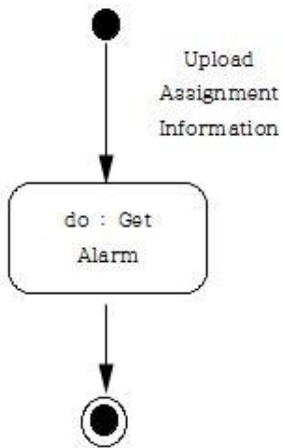
다이어그램 12. Professor activity 모델

2) Course activity 모델



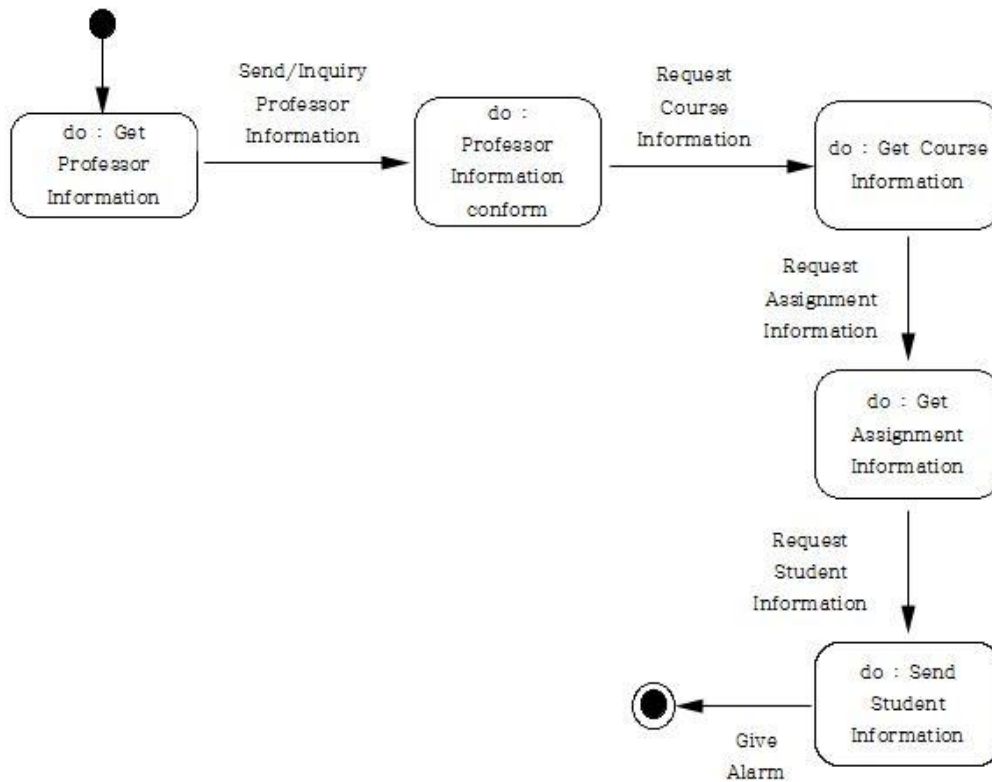
다이어그램 13. Course activity 모델

3) Student activity 모델



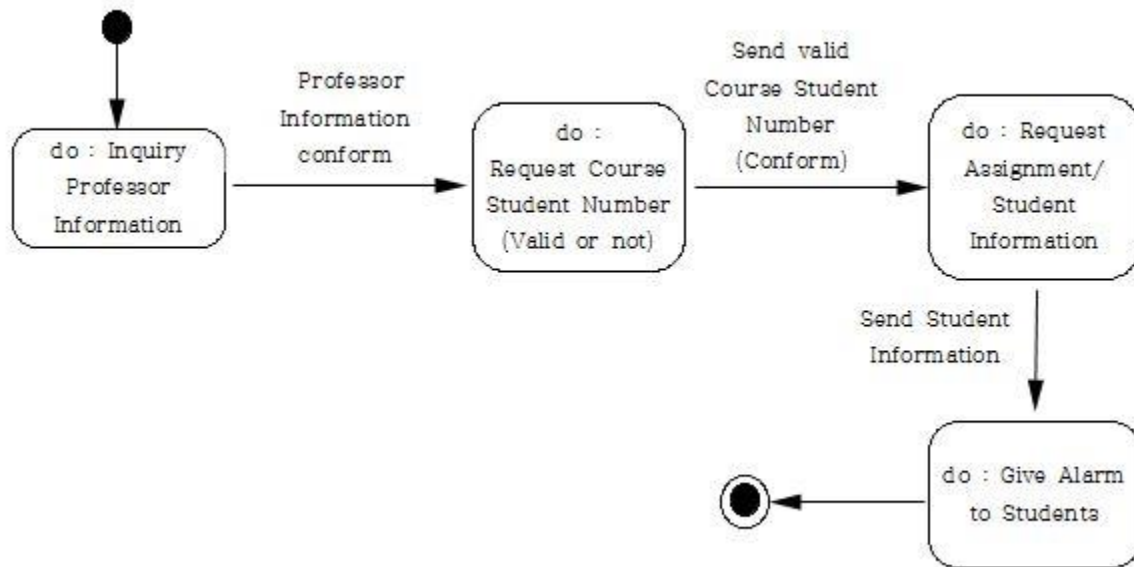
다이어그램 14. Student activity 모델

4) HAS activity 모델



다이어그램 15. HAS activity 모델

5) Database activity 모델



다이어그램 16. Database activity 모델

8. System Evolution

이 섹션에서는 시스템이 기반하는 전반적인 가정들에 대해 설명한다. 그리고 시스템에서 일어날 수 있는 가능한 모든 변경에 대해서도 설명한다.

8.1. Hypothesis of System

A. 교내 데이터베이스 참조

HAS 는 교수님이 강의하는 학수번호와, 해당 학수번호의 강의를 수강하는 학생들의 데이터를 시스템의 repository 에서 호출하여, 메시지를 전달하는 시스템이다 따라서, I-campus 에 저장된 해당 교수님의 과목별 학수번호와, 해당 강의에 등록된 학생정보를 받아와야 system 이 기능을 발휘할 수 있다.

8.2. Evolution of System

A. 카카오톡 어플리케이션이 사용이 불가능한 경우

카카오톡 시스템이 예기치 못한 오류로 인해 사용이 불가능한 경우를 대비해, Database 에 학생들의 카카오톡 ID 뿐만 아니라, 개개인의 메일주소를 저장한다. alert_system 이후에 연동되는 외부 시스템을 MSmail 로 옮겨, 학생들 개개인의 메일로 과제의 정보를 송신한다.

B. I-Campus 의 Interface 변경

I-campus 의 interface 가 변경되면, HAS 에서 학생들의 정보를 Database 로 가져올 때에 정보가 저장된 주소를 찾지 못하여, 문제가 발생할 수 있다. 따라서, HAS 는 학생들의 정보를 수동으로 입력이 가능하게 설계하고, I-campus 의 변동사항이 있는지를 수시로 확인하여야 한다.

9. Index

9.1 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. User Requirements Definition

그림 3. 공지사항 및 과제 확인 기능

그림 4. 유용한 기능

그림 5. 심리적 편안함 제공

그림 6. 숙제 및, 과제 그리고 프로젝트 공지 제공

그림 7. 학생들의 디바이스를 이용하여 정보 획득

그림 8. 사용하기 복잡한 UI(좌)와 직관적인 UI(우)

그림 9. 부피가 큰 space(좌)와 부피가 작은 space(우)

그림 10. 와이파이와 휴대용 디바이스의 인터페이스

그림 11. 다양한 통신 환경의 조성

그림 12. 인터넷 환경

그림 13. 엘로아이디 승인

5. System Architecture

6. System Requirements Specification

7. System Models

8. System Evolution

9. Index

10. Appendices

그림 14. HAS 를 통해서 메시지 전달하는 시스템

그림 15. 비밀번호를 잘못 입력하기 전과 비밀번호를 잘못 입력 한 후

그림 16. 과목번호를 바꿀 수 있다.

9.2 Diagram Index

1. Preface

2. Introduction

3. Glossary

4. User Requirement Definition

5. System Architecture

다이어그램 1. HAS 시스템의 전체 구조도

다이어그램 2. HAS 접속 구조도

다이어그램 3. NETWORK 접속 구조도

다이어그램 4. NETWORK 접속 구조도

6. System Requirements Specification

7. System Models

다이어그램 5. HAS 시스템 Context

다이어그램 6. HAS 시스템 프로세스

다이어그램 7. 시스템 전체 Actor 와 use case

다이어그램 8. Class edit 상태 확인 use case

다이어그램 9. Structure edit 상태 확인 use case

다이어그램 10. HAS edit 상태 확인

다이어그램 11. HAS 등록 Sequence diagram

다이어그램 12. HAS class Sequence diagram

다이어그램 13. Professor activity Sequence

- 다이어그램 14. Course activity Sequence diagram
- 다이어그램 15. Student activity Sequence diagram
- 다이어그램 16. HAS activity Sequence diagram
- 다이어그램 17. Database activity Sequence diagram

8. System Evolution

9. Index

10. Appendices

9.3 Table Index

1. Preface

2. Introduction

3. Glossary

표 1. Glossary

4. User Requirements Definition

5. System Architecture

6. System Requirements Specification

표 2. 사용자 외출 감지 후 자동 전력 제어 기능 요구사항

7. System Models

표 3. Class edit 상태 확인 use case 설명

표 4. Student edit 상태 확인 use case 설명

표 5. HAS edit 상태 확인 use case 설명

8. System Evolution

9. Index

10. Appendices

10. Appendices

이 섹션에서는 개발되는 시스템에 대해 더 자세하고 구체적인 정보들을 제공한다.
데이터베이스, 시스템 요구사항, 사용된 기술, 개발에 적용된 프로세스 등에 대해 설명한다.

10.1. 데이터베이스 요구사항

HAS 시스템에서는 학생들과 교수님들의 정보를 HAS 시스템 내에 DB 에 저장하기 위해서 데이터베이스를 사용한다.

10.2. 사용자 시스템 요구사항

사용자 시스템 요구사항으로는 특별한 것은 없다. HAS 시스템에 본인의 개인정보를 잘 업그레이드를 할 필요가 있으며, 어디에서든 메시지를 받아 볼 수 있도록 모바일 디바이스가 필요하다.

10.3. 적용 개발 프로세스

Waterfall diagram Tutorial



다이어그램 17. 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.

References

[1] "Waterfall",

[https://www.google.co.kr/search?q=waterfall+diagram&biw=1649&bih=877&tbm=isch&imgil=YIAzsFWH-](https://www.google.co.kr/search?q=waterfall+diagram&biw=1649&bih=877&tbm=isch&imgil=YIAzsFWH-LL_BM%253A%253BXsqPaoxh8YmuqM%253Bhttp%25253A%25252F%25252Fwww.presentation-process.com%25252Fwaterfall-diagram-in-powerpoint.html&source=iu&pf=m&fir=YIAzsFWH-LL_BM%253A%252CXsqPaoxh8YmuqM%252C_&usg=__FRuG0Hfx6qnQVrkUzhUX1xLm9Wc%3D&ved=0ahUKEwja8-Xa2fnMAhXIH5QKHVzhCaUQyjcIOQ&ei=-_JHV5qKDMi_0ATcwqeoCg#imgsrc=YIAzsFWH-LL_BM%3A)

[LL_BM%253A%253BXsqPaoxh8YmuqM%253Bhttp%25253A%25252F%25252Fwww.presentation-process.com%25252Fwaterfall-diagram-in-powerpoint.html&source=iu&pf=m&fir=YIAzsFWH-](https://www.google.co.kr/search?q=waterfall+diagram&biw=1649&bih=877&tbm=isch&imgil=YIAzsFWH-LL_BM%253A%253BXsqPaoxh8YmuqM%253Bhttp%25253A%25252F%25252Fwww.presentation-process.com%25252Fwaterfall-diagram-in-powerpoint.html&source=iu&pf=m&fir=YIAzsFWH-LL_BM%253A%252CXsqPaoxh8YmuqM%252C_&usg=__FRuG0Hfx6qnQVrkUzhUX1xLm9Wc%3D&ved=0ahUKEwja8-Xa2fnMAhXIH5QKHVzhCaUQyjcIOQ&ei=-_JHV5qKDMi_0ATcwqeoCg#imgsrc=YIAzsFWH-LL_BM%3A)

[LL_BM%253A%252CXsqPaoxh8YmuqM%252C_&usg=__FRuG0Hfx6qnQVrkUzhUX1xLm9Wc%3D&ved=0ahUKEwja8-Xa2fnMAhXIH5QKHVzhCaUQyjcIOQ&ei=-_JHV5qKDMi_0ATcwqeoCg#imgsrc=YIAzsFWH-LL_BM%3A](https://www.google.co.kr/search?q=waterfall+diagram&biw=1649&bih=877&tbm=isch&imgil=YIAzsFWH-LL_BM%253A%253BXsqPaoxh8YmuqM%253Bhttp%25253A%25252F%25252Fwww.presentation-process.com%25252Fwaterfall-diagram-in-powerpoint.html&source=iu&pf=m&fir=YIAzsFWH-LL_BM%253A%252CXsqPaoxh8YmuqM%252C_&usg=__FRuG0Hfx6qnQVrkUzhUX1xLm9Wc%3D&ved=0ahUKEwja8-Xa2fnMAhXIH5QKHVzhCaUQyjcIOQ&ei=-_JHV5qKDMi_0ATcwqeoCg#imgsrc=YIAzsFWH-LL_BM%3A)

(2) "Database", <https://en.wikipedia.org/wiki/Database>

(3) "sql", <https://azure.microsoft.com/ko-kr/services/sql>

(4) "엘로우아이디", <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](https://ko.wikipedia.org/wiki/%EC%9D%B8%ED%84%B0%ED%8E%98%EC%9D%B4%EC%8A%A4)

(6) "Wi-Fi",

<https://ko.wikipedia.org/wiki/%EC%99%80%EC%9D%B4%ED%8C%8C%EC%9D%B4>

(7) "모바일 장치",

https://ko.wikipedia.org/wiki/%EB%AA%A8%EB%B0%94%EC%9D%BC_%EC%9E%A5%EC%B9%98