

Final Report

- Ajou Univ LIfe Manager(AULIM) -



Team 8

수학과 201621136 이재협

경영학과 201621352 이창엽

소프트웨어학과 201621763 김진수

소프트웨어학과 201720733 신승현

소프트웨어학과 201820773 양준길

INDEX

1. Project Overview -----	6
1.1 Problem Statement and Motivation -----	
1.2 Proposed Idea and Solution -----	
2. Scope and Objective -----	9
3. User Requirements -----	9
4. System Requirements -----	11
4.1 Functional Requirements -----	
4.2 Non-Functional Requirements -----	
5. Requirements Specification -----	15
5.1 Use Case 1: 시간표와 활동 스트림 연동	
5.2 Use Case 2 : 개별 활동 추가	
5.3 Use Case 3 : 학습 계획표 추천받기	
5.4 Use Case 4 : 그룹 공강시간 조회	
5.5 Use Case 5 : 수업 시작 알림	
5.6 Use Case 6 : 메모 작성	

6. Requirements Analysis ----- 21

6.1 Behavioral Analysis -----

6.1.0 Sub - Use Case : 로그인

6.1.1 Use Case 1 : 시간표/활동 스트림 불러오기

6.1.2 Use Case 2 : 개별활동 추가

6.1.3 Use Case 3 : 학습 계획표 추천

6.1.4 Use Case 4 : 그룹 형성 및 공통 공강 시간 추출하기

6.1.5 Use Case 5 : 실시간 수업 알림 받기

6.1.6 Use Case 6 : 메모 작성하기

6.2 Structural Analysis -----

6.2.1 Use Case 1: 시간표/활동 스트림 불러오기

6.2.2 Use Case 2: 개별 활동 추가

6.2.3 Use Case 3: 학습 계획표 추천

6.2.4 Use Case 4: 그룹 형성 및 공통 공강 시간 추출하기

6.2.5 Use Case 5: 실시간 수업 알림 받기

6.2.6 Use Case 6: 메모 작성하기

7. Architectural Design -----	31
7.1 Logical View / Architecture -----	
8. Design Modeling -----	32
8.1. Use-case 01. 시간표와 활동스트림 연동	
8.2. Use-case 02. 개별 활동 추가 및 주당 학습시간 설정	
8.3. Use-case 03. 학습 계획표 추천	
8.4. Use-case 04. 그룹 공강시간 조회	
8.5. Use-case 05. 수업시간 알림 설정하기	
8.6. Use-case 06. 메모 작성	
8.7. Combined Design Class Diagram	
9. Implementation -----	38
9.1 Implementation Specific -----	42
9.1.1 System Overview	
9.1.2 AULIM Server(Web-node js)	
9.1.3 AULIM Server(node js-MySQL)	
9.1.4 Crawler	
9.2 Message Format	
9.3 System/ Data Dependencies & Requirements -----	
10. UI Design -----	48

11. Test Plan & Result -----	55
11.1 Test Scope & Objectives -----	55
11.2 Test Plan -----	55
11.2.1. Software to be Tested	
11.2.2 Testing Strategy	
11.2.2.1 Unit testing	
11.3 Test Procedures -----	56
11.3.1 Unit Test Cases	
11.4 Integration / System Test Case -----	57
11.4.1. T_001	
11.4.2. T_002	
11.4.3. T_003	
11.4.4. T_004	
11.5 Test Results -----	59
11.6 Requirements Satisfaction -----	60
11.6.1 User Requirements	
11.6.2 System Requirements	
11.6.2.1 Functional Requirements	
11.6.2.2 Non-Functional Requirements	

Appendix

A. Glossary -----	69
B. Domain/Design Class Description -----	69
C. Project Status -----	70
C.1 Planned vs Actual	
C.2 Team members' contribution	
D. Change History -----	73
E. References -----	74
F. List of Figures -----	75

1. Project Overview

1.1 Problem Statement and Motivation/Background

1.1.1 Problem Statement

대학 생활에 있어서 학과 공부는 필수적이다. 정해진 시간에 수업을 듣고 과제를 제출하며, 퀴즈나 중간/기말고사를 준비한다. 그 과정에서 아래와 같은 어려움을 겪고 있다.

먼저, 공지사항을 미처 확인하지 못하는 경우이다. PC를 사용하면서, 아주 Bb에 올라온 공지사항을 확인하기 위해서는 사용자가 직접 해당 과목을 찾아 확인을 해야 한다.

두번째로, 실시간 강의를 깜빡 잊는 경우이다. 최근 코로나19로 인해서 대면 수업 대신 실시간 수업으로 진행하는 과목이 많다. 하지만 실시간 강의를 수업을 깜빡 잊고 지각 혹은 결석을 해본 학생들이 있을 것이다.

다음으로, 동영상 강의를 학생이 스스로 시간을 내어 학습해야 함에 따라 효율적인 시간 관리를 위한 학습 계획 수립이 필요하다. 그러나, 대부분의 학생들은 시간 관리에 어려움을 겪고 있다.

마지막으로, 다른 사람과 스터디 등의 이유로 공통된 공강시간을 알고 싶을 때 불편함이 있다. 인원 수가 늘어날수록, 사용자가 직접 공강시간을 비교하기에는 매우 번거롭다.

따라서, 우리는 공지사항 및 실시간 강의 알림, 학습 계획표 추천, 공강시간 추출 등의 기능을 모두 통합하는 서비스 제공하여 위의 문제들을 해결하고자 한다. 추가적으로, 강의 수강 중 핵심 내용을 메모하는 기능도 추가할 예정이다. 이러한 기능들을 통해서 아주대학교 학생들의 학교 생활에 밀접한 도움이 되기를 바란다.

1.1.2 Motivation / Background

대부분의 사람들은 20대의 절반을 대학 생활을 하면서 보낸다. 정해진 시간에 수업을 수강하고, 주어진 과제를 수행하고, 심지어는 동아리나 스터디 그룹을 통해 취미 활동이나 자기 계발을 한다. 그러나, 아주대 학생들이 대학 생활과 관련된 어플리케이션은 ‘에브리타임’과 ‘Blackboard’ 어플리케이션을 제외하고는 대중적으로 사용하고 있는

서비스들이 부족한 것이 현실이다. 에브리타임 어플은 시간표 공유, 강의 평가, 게시판 등의 기능 제공하며 이 중에서 가장 유용한 것은 시간표 공유 서비스이다. 반면, Bb는 동영상 강의 수강, 공지사항 및 강의 노트 확인 등의 기능을 하고 있다. 따라서, 사용자들은 두 개의 어플리케이션을 번갈아가며 사용해야 하는 번거로움이 있다.

코로나19로 인해 많은 과목들이 실시간/동영상으로 수업을 진행하고 있다. 그 결과 실시간 수업에 대한 공지사항을 확인하지 못하거나 수업을 깜빡하고 결석하는 경우도 많다. 동영상 강의 또한 따로 시간을 내어 수강해야 하는데, 이를 계획적으로 동영상을 시청하며 자발적으로 공부하기에는 많은 노력을 필요로 한다. 이외에도, 식사, 팀프로젝트 등의 이유로 타인과 시간표를 공유할 때에도 가능한 시간을 찾는데 번거로움이 있다. 특히 사람이 많아질 경우 모두가 비는 시간을 하나하나 비교하는 것은 매우 불편하다.

위와 같이, 대학 생활을 하는데 있어 에브리타임과 Bb 어플리케이션의 주된 기능을 통합하는 것과 더불어 공지사항 확인 및 실시간/동영상 수업 여부에 따른 알람, 학습 계획 추천 서비스 등의 서비스가 꼭 필요하다고 판단된다. 대학 생활에 보다 밀접한 도움을 줄 수 있는 서비스를 제공하여 많은 대학생들에게 편리함과 대학 생활의 질을 향상시키고자 이러한 프로젝트 주제를 선정하게 되었다.

1.1.3 Related Work and Trends

1) 에브리타임과 Bb의 독립성에 따른 불편함

모든 대학생들을 대상으로 서비스하는 에브리타임에는 Bb 공지사항, 강의노트 등을 열람할 수 있는 기능이 없다. 기존의 해결책으로는 시간표의 강의마다 메모, 할 일 등을 추가할 수 있는 기능을 제공하고 포털사이트 링크를 제공한다. 그러나, 메모나 할 일 추가를 위해서는 세부 내용이나 기한을 직접 설정해야 하는 번거로움이 있다.

2) 알람 기능의 부재

등록된 수업의 실시간/동영상 여부와 알림을 설정할 수 없으나 이에 대한 해결책은 아직 제시되지 않았다.

3) 다른 사용자와의 공강시간 공유 기능 부재

다른 사용자와 공통되는 공강 시간을 알아내기 어렵다. 이 기능은 스터디 그룹, 동아리 활동을 위해서는 꼭 필요하다. 친구 등록이 된 사용자의 시간표를 “열람”하거나 시간표를 사진으로 저장하여 카카오톡 단톡방에 올려 공강시간을 직접 비교는 가능하지만 여전히 불편하다.

4) 시간표 계획 추천 기능 부재

- 캘린더의 과제 마감 기한을 확인할 수 있는 기능은 존재하지만, 학습 시간 계획표를 추천해주는 기능은 존재하지 않는다.

1.2 Proposed Idea and Solution

1) 에브리타임과 Bb 어플리케이션의 기능을 통합.

- 학생마다 자신의 시간표를 구성할 수 있다.
- 다른 서비스로 이동할 필요없이 각 과목마다 공지사항, 과제들을 크롤링 기술을 이용하여 사용자에게 알려준다.

2) 실시간 강의 알람.

- 등록한 강의시간이 실시간 강의인 경우, 해당 강의시간에 맞추어 알림을 보내주어 memorization의 역할을 한다.

3) 공강시간 추출 알고리즘 구현

- 특정 사용자들의 공통된 공강시간이 표시해주어 편리함을 제공한다.

4) 추천 공부 계획표 제시

- 과목마다 매주 필요한 학습 시간을 미리 설정.

- 사용자의 시간표와 과목마다 설정된 학습시간을 맞추어 학습 계획표를 추천해주어 이를 그대로 사용하거나 일부 수정할 수 있다.
- 통합된 서비스와 부가 기능을 제공하여 사용자들의 학교 생활에 있어 간편함을 제공한다.

2. Scope and Objective

1) In the scope

- 개인 시간표 및 실시간/동영상 여부 저장
- 실시간 수업 시간에 맞춘 알람
- 학습 계획표 추천 서비스 제공
- 다른 사용자와 공통되는 공강 시간 추출 서비스

2) Out of scope

- App 개발에 대한 경험 부족으로 인해 Web 서비스로 구현 예정
- 다른 대학교가 아닌 아주대학교 학생으로 한정된 서비스 제공

3) Object of our project

- Providing Convenient University-Life Service (대학생활에 유용한 서비스를 제공)

3. User Requirements

No	USER_RS_001
Title	사용자는 필수적으로 현재 수강 신청된 과목을 확인할 수 있어야 한다.
description	-

No	USER_RS_002
Title	사용자는 필수적으로 아주Bb 활동스트림에 업로드 된 최근 공지사항을 제공받아야 한다.
description	-

No	USER_RS_003
Title	사용자는 필수적으로 수강 신청 과목 이외 활동을 추가할 수 있어야 한다.
description	수강 신청 과목 이외에도 스터디/동아리/자격증 공부 등의 활동을 하는 경우가 많다. 따라서 사용자가 직접 추가할 수 있어야 한다.

No	USER_RS_004
Title	사용자는 필수적으로 수강 신청 과목과 추가된 활동을 시간표 형식으로 제공받아야 한다.
description	-

No	USER_RS_005
Title	사용자가 필수적으로 수강 중이거나 추가 활동에 대해 매주 몇 시간을 공부할 지 설정할 수 있어야 한다.
description	사용자는 수업을 듣는 것 뿐만 아니라 따로 공부할 시간이 필요할 수 있다.

No	USER_RS_006
Title	사용자는 필수적으로 학습 추천 계획표를 확인 및 저장할 수 있어야 한다.
description	사용자의 시간표와 1주일 당 학습 시간에 맞춰서 학습 추천 계획표가 생성된다.

No	USER_RS_007
Title	사용자는 필수적으로 그룹을 만들어 다른 사용자를 초대할 수 있어야 한다.
description	-

No	USER_RS_008
Title	사용자는 필수적으로 같은 그룹원들과 공통되는 공강시간을 제공받아야 한다.
description	사용자는 이를 스터디/동아리/팀프로젝트 등에 사용할 수 있다.

No	USER_RS_009
Title	사용자는 필수적으로 각 과목의 실시간 강의 알림 여부를 설정하고 수업 시간에 맞추어 소리 혹은 푸시 알림을 받을 수 있어야 한다.
description	-

No	USER_RS_010
Title	사용자는 수업 내용을 과목별, 날짜별로 메모 및 열람할 수 있어야 한다.
description	-

4. System Requirements

1) Functional Requirements

No	SYS_RS_001	Related requirements	USER_RS_001 USER_RS_002
Title	AULIM 로그인은 필수적으로 사용자의 아주Bb 아이디/비밀번호를 이용해야 한다.		
description		시간표/활동 스트림을 불러오기 위해서는 아주Bb 사이트 접근이 필요하고, 따라서 아주Bb 아이디/비밀번호가 필요하다. AULIM에 따로 회원 가입을 만들더라도, 아주Bb 비밀번호가 바뀌면 시간표/활동 스트림을 불러오지 못한다. 따라서, AULIM 개별 회원 가입 없이, 아주Bb 아이디/비밀번호로 로그인 할 수 있도록 하는 것이 합리적이다.	

No	SYS_RS_002	Related requirements	USER_RS_001 USER_RS_002

Title	시스템은 필수적으로 사용자가 로그인 할 때, 사용자의 시간표와 활동 스트림을 업데이트해야 한다.
description	수강 정정/포기로 인해 시간표/활동 스트림은 변동될 수 있다. 그러나, 모든 사용자의 정보를 몇 시간마다 혹은 매일 확인하는 것은 불필요하다. 사용자가 로그인할 때에만 그 사용자의 시간표/활동 스트림을 업데이트 하는 것으로 충분하다.

No	SYS_RS_003	Related requirements	USER_RS_003
Title	시스템은 사용자가 수강 신청 과목 이외 활동을 추가 및 저장할 수 있도록 해야한다.		
description	수강 신청 과목 이외의 스터디/동아리/자격증 공부 등에 활용될 수 있다.		

No	SYS_RS_004	Related requirements	USER_RS_004
Title	시스템은 필수적으로 사용자의 수강 신청 과목, 추가된 활동을 사용자가 알아볼 수 있는 시간표의 형태로 한 화면에 출력해야 한다.		
description	-		

No	SYS_RS_005	Related requirements	USER_RS_005
Title	시스템은 필수적으로 사용자가 과목 및 활동별로 필요한 희망 학습 시간을 기록 및 저장할 수 있도록 해야 한다.		
description	-		

No	SYS_RS_006	Related requirements	USER_RS_005
Title	시스템은 필수적으로 사용자가 입력한 학습 시간과 시간표를 기반으로 “학습 추천 계획표”를 제공해야 한다.		
description	-		

No	SYS_RS_007	Related requirements	USER_RS_006
Title	시스템은 필수적으로 사용자가 추천 학습 계획표의 사용 여부를 선택할 수 있고, 새로운 계획표를 추천 받을 수 있어야 한다.		
description	사용자가 제공 받은 학습 계획표가 불만족스럽다면, 새로운 학습 추천 계획표를 제공받을 수 있다.		

No	SYS_RS_008	Related requirements	USER_RS_007
Title	시스템은 필수적으로 사용자가 그룹을 형성하여 다른 사용자를 초대할 수 있도록 해야 한다.		
description	-		

No	SYS_RS_009	Related requirements	USER_RS_008
Title	시스템은 필수적으로 사용자가 가입되어 있는 그룹의 그룹원과 공통된 공강시간을 제공해야 한다.		
description	-		

No	SYS_RS_010	Related requirements	USER_RS_009
Title	시스템은 필수적으로 사용자가 수강 과목의 실시간 알람 여부를 Yes/No 형태로 설정할 수 있도록 해야 한다.		
description	-		

No	SYS_RS_011	Related requirements	USER_RS_009
Title	실시간 알림을 “Yes”로 선택했다면 수업 시작 10분전 시스템은 사용자에게 알람을 보내주어야 한다.		
description	사용자가 수업 시간을 잊어버릴 것을 대비하여, 실시간 알림을 희망하도록 설정했다면 수업 시간에 맞추어 알람을 보내준다.		

No	SYS_RS_012	Related requirements	USER_RS_010
Title	강의가 시작될 때 자동적으로 해당 과목, 날짜에 대한 메모입력칸이 생성되어야 한다.		
description	메모장 실행버튼과 취소 버튼중 실행 버튼이 클릭되었을 때 메모 제목을 해당 강의명, 날짜로 생성해야 하며, 새로운 메모를 적을 수 있는 페이지 혹은 창이 생성되어야 한다.		

No	SYS_RS_013	Related requirements	USER_RS_010
Title	시스템은 사용자가 수업을 듣고 있을 때, 그 과목에 대한 메모를 작성 및 저장할 수 있어야 한다.		
description	사용자가 수업 내용을 기록하여 추후에 읽을 수 있도록 한다.		

No	SYS_RS_014	Related requirements	USER_RS_010
Title	시스템은 사용자가 이전에 작성했던 메모를 불러올 수 있어야 한다.		
description	사용자가 이전에 기록한 수업 내용을 변경할 수 있도록 한다.		

2) Non - Functional Requirements

No.	NF_RS_001	Related Requirement	USER_RS_001~005 SYS_RS_001~009
Title	시스템 이동 시 Time Delay가 1초 이내로 한다.		
Description	Time Delay가 길어지면 사용자들은 불편함을 겪는다.		

No.	NF_RS_002	Related Requirement	USER_RS_001 SYS_RS_001~002
Title	사용자가 시간표 및 활동스트림 불러오기를 요청할 때 30초 이내로 완료한다.		

Description	기본적인 정보를 불러오는 데에 30초 이상의 긴 시간이 걸릴 경우 사용자 입장에서 불편함을 느낄 수 있다.		
-------------	---	--	--

No.	NF_RS_003	Related Requirement	USER_RS_001 SYS_RS_002
Title	활동 스트림 정보를 5분마다 업데이트 한다.		
Description	아주 BlackBoard에서 새로운 활동 스트림이 발생하면, AULIM은 이를 최대한 빠르게 업데이트 해야 한다. 그러나, Bb 활동스트림 발생 즉시 업데이트하기에는 어려움이 있으므로 업데이트 주기를 5분으로 한다.		

No.	NF_RS_004	Related Requirement	USER_RS_004 SYS_RS_007
Title	한 과목 당 저장할 수 있는 메모의 용량은 최소 350KB 이상 이어야 한다.		
Description	메모 하나의 최대 용량을 10KB 했을 때, 350KB 정도면 사용자가 한 학기동안 사용하기에 충분하다.		

No.	NF_RS_005	Related Requirement	USER_RS_001
Title	다른 사용자에게 자신의 시간표가 보이지 않아야 한다.		
Description	사용자의 시간표 또한 일종의 개인 정보이므로 Private 하게 유지해야 한다.		

No.	NF_RS_006	Related Requirement	USER_RS_001~004
Title	DB에 저장되는 data들을 1시간마다 백업한다.		
Description	의도치 않게 종료되는 등의 문제 발생 시 database의 안정성을 보장한다.		

5. Requirements Specification

5.1 Usecase 1: 시간표와 활동 스트림 연동

Use Case Number	UC_001
Use Case Name	시간표와 활동 스트림 연동
Primary Actor	사용자(아주대학생)
Stakeholders and Interests	사용자(아주대학생)의 편의성 증가
Pre-conditions	1. 사용자가 웹페이지에 로그인하고 있어야 한다. 2.
Post-conditions	개인 시간표와 활동 스트림 페이지가 업데이트된다.
Main Success Scenario	1. 사용자가 개인시간표와 활동스트림 업데이트 버튼을 누른다. 2. 로딩 화면이 띄워지며 시간표와 활동스트림이 업데이트 되기를 기다린다. 3. 시간표와 활동스트림 페이지가 최신 상태로 업데이트된다.
Extensions(Alternative Flows)	*3-1 . 시간표와 활동스트림 정보를 얻는데에 실패하여 경고문이 표시된다. 4 . 화면이 새로고침되며 기존의 버튼을 누르기 전으로 돌아온다.
Open Issues	정보 가져오기 실패

5.2 Usecase 2: 개별 활동 추가

Use Case Number	UC_002
Use Case Name	학습계획표 추천받기
Primary Actor	사용자(아주대학생)
Stakeholders and Interests	사용자(아주대학생)의 편의성 증가

Pre-conditions	<ol style="list-style-type: none"> 1. 사용자가 웹페이지에 로그인하고 있어야 한다. 2. 사용자가 강의 외에 추가로 등록한 활동이 1개이상 존재해야 한다.
Post-conditions	개인 시간표가 추가로 등록한 활동 시간이 분배된 상태로 업데이트된다.
Main Success Scenario	<ol style="list-style-type: none"> 1. 사용자가 학습시간 추천 버튼을 누른다. 2. 로딩 화면이 띄워지며 시간표가 업데이트 되기를 기다린다. 3. 구현된 알고리즘이 적용되어 학습시간이 분배된 시간표가 화면에 등장한다. 4. 사용자가 등장한 시간표를 선택한다. 5. 개인시간표가 활동 시간이 분배된 시간표로 업데이트된다.
Extensions(Alternative Flows)	<ul style="list-style-type: none"> *4. 사용자가 분배된 시간표를 마음에 들어하지 않아 취소한다. *5. 다음 시간표를 추천받을지 선택받는다. *6. 추천받기를 선택하면 2번으로 돌아간다. **6. 거부를 선택하면 초기화면으로 돌아간다.
Open Issues	사용자가 추천된 시간표를 수정하거나 거부하기를 희망함.

5.3 Usecase 3: 학습 계획표 추천받기

Use Case Number	UC_003
Use Case Name	학습계획표 추천받기
Primary Actor	사용자(아주대학생)
Stakeholders and Interests	사용자(아주대학생)의 편의성 증가
Pre-conditions	<ol style="list-style-type: none"> 1. 사용자가 웹페이지에 로그인하고 있어야 한다. 2. 사용자가 강의 외에 추가로 등록한 활동이 1개이상 존재해야 한다.

Post-conditions	개인 시간표가 추가로 등록한 활동 시간이 분배된 상태로 업데이트된다.
Main Success Scenario	<ol style="list-style-type: none"> 1. 사용자가 학습시간 추천 버튼을 누른다. 2. 로딩 화면이 띄워지며 시간표가 업데이트 되기를 기다린다. 3. 구현된 알고리즘이 적용되어 학습시간이 분배된 시간표가 화면에 등장한다. 4. 사용자가 등장한 시간표를 선택한다. 5. 개인시간표가 활동 시간이 분배된 시간표로 업데이트된다.
Extensions(Alternative Flows)	<ul style="list-style-type: none"> *4. 사용자가 분배된 시간표를 마음에 들어하지 않아 취소한다. *5. 다음 시간표를 추천받을지 선택받는다. *6. 추천받기를 선택하면 2번으로 돌아간다. **6. 거부를 선택하면 초기화면으로 돌아간다.
Open Issues	사용자가 추천된 시간표를 수정하거나 거부하기를 희망함.

5.4 Usecase 4: 그룹 공강시간 조회

Use Case Number	UC_004
Use Case Name	그룹 공강 시간 조회
Primary Actor	사용자(아주대학생)
Stakeholders and Interests	사용자(아주대학생)의 편의성 증가
Pre-conditions	<ol style="list-style-type: none"> 1. 사용자가 웹페이지에 로그인하고 있어야 한다. 2. 사용자가 1개 이상의 그룹에 참여하고 있어야 한다.
Post-conditions	공강시간이 시간표에 표시된다.
Main Success Scenario	<ol style="list-style-type: none"> 1. 사용자가 공통으로 빈 시간을 조회하고 싶은 그룹의 공강시간 조회 버튼을 누른다. 2. 로딩 화면이 띄워지며 공강시간이 조회되기를 기다린다. 3. 시간표에 공강시간이 표시된 상태로 업데이트된다.

Extensions(Alternative Flows)	*3 . 공강시간을 얻는데에 실패하여 경고문이 표시된다. 4 . 화면이 새로고침되며 기존의 버튼을 누르기 전으로 돌아온다.
Open Issues	공통된 공강시간이 존재하지 않음

5.5 Use Case 5: 수업 시작 알림

Use Case Number	UC_005
Use Case Name	수업 시작 알림받기
Primary Actor	사용자(아주대학생)
Stakeholders and Interests	사용자(아주대학생)의 편의성 증가
Pre-conditions	사용자가 웹페이지에 로그인하고 있어야 한다. 작성된 시간표가 있어야 한다. 수업의 알림여부가 선택되어 있어야 한다.
Post-conditions	해당 수업 시작시간 10분전에 새로운 팝업으로 사용자에게 수업시작 10분전 임을 알려준다..
Main Success Scenario	1. 서버시간이 수업시작 10분전이 되면 수업시작 10분전이라는 알림이 등장한다. 2. 사용자가 메모장 실행과 닫기 버튼 중 메모장 실행을 선택한다. 3. 해당 수업의 메모장이 실행된다.
Extensions(Alternative Flows)	2* 사용자가 확인을 선택한다. 3. 초기화면으로 돌아간다.
Open Issues	사용자의 Local Time과 Server Time이 맞지 않아 제 시간에 알림이 울리지 않음

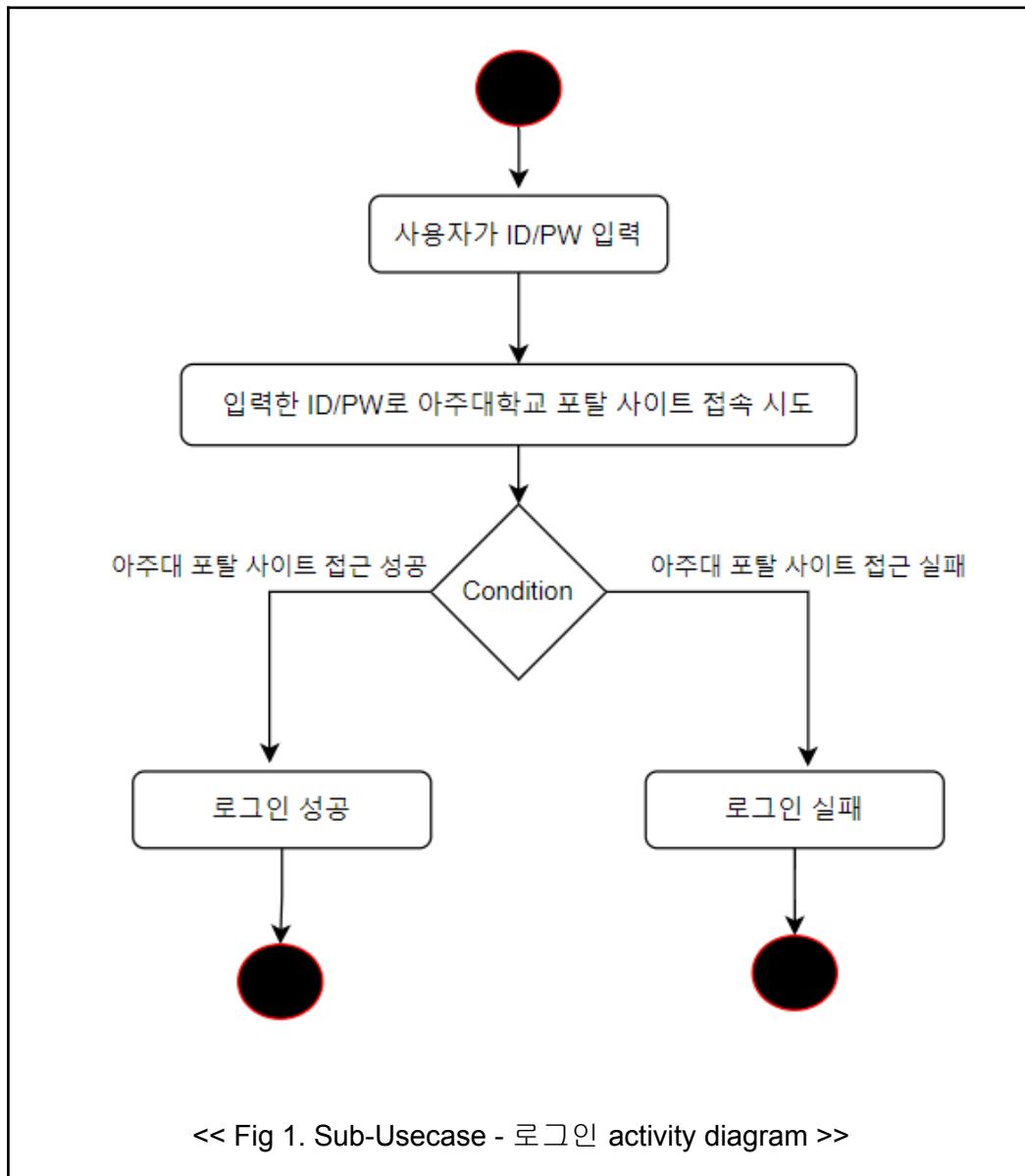
5.6 Use Case 6: 메모 작성

Use Case Number	UC_006
Use Case Name	수업에 대한 메모 작성하기
Primary Actor	사용자(아주대학생)
Stakeholders and Interests	사용자(아주대학생)의 편의성 증가
Pre-conditions	사용자가 웹페이지에 로그인하고 있어야 한다.
Post-conditions	해당 수업과 날짜에 메모가 1개 추가된다.
Main Success Scenario	<ol style="list-style-type: none"> 1. 상단 바에서 메모 페이지를 선택한다. 2. 메모 페이지의 강의 목록에서 메모할 강의를 선택한다. 3. 날짜 목록에서 새 문서 작성을 선택한다. 4. 텍스트 상자에 메모할 내용을 입력한다. 5. 저장 버튼을 선택한다.
Extensions(Alternative Flows)	<p>*3-1. 이전의 메모를 수정하기위해 날짜 목록에서 이전 메모를 선택한다.</p> <p>*3-2. 수정할 내용으로 변경한다.</p>
Open Issues	사용자가 일정 한도 이상의 텍스트를 저장하려고 함

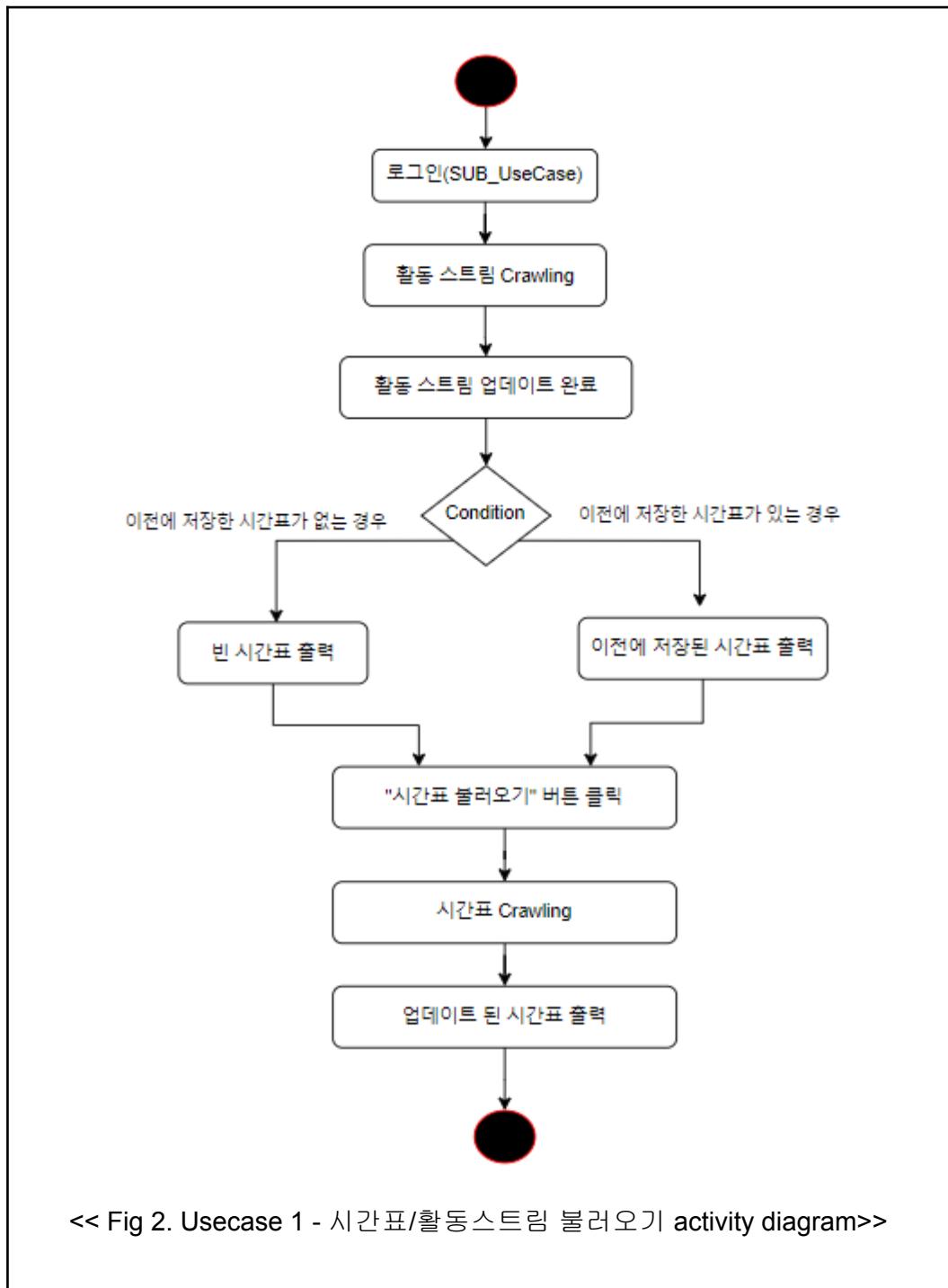
6. Requirements Analysis

6.1 Behavioral Analysis

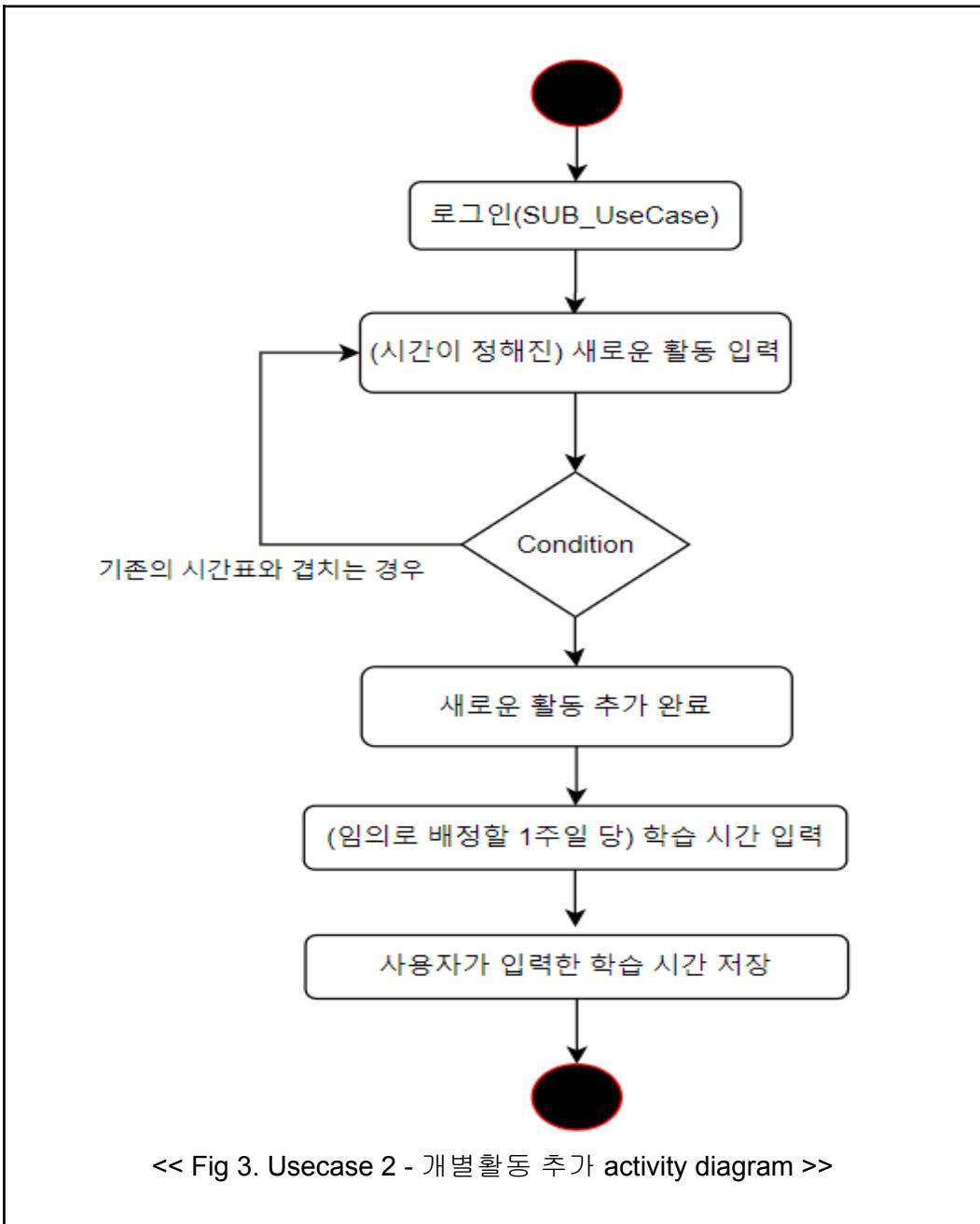
6.1.0 Sub-Usecase : 로그인



6.1.1 Usecase 1 : 시간표/활동 스트림 불러오기

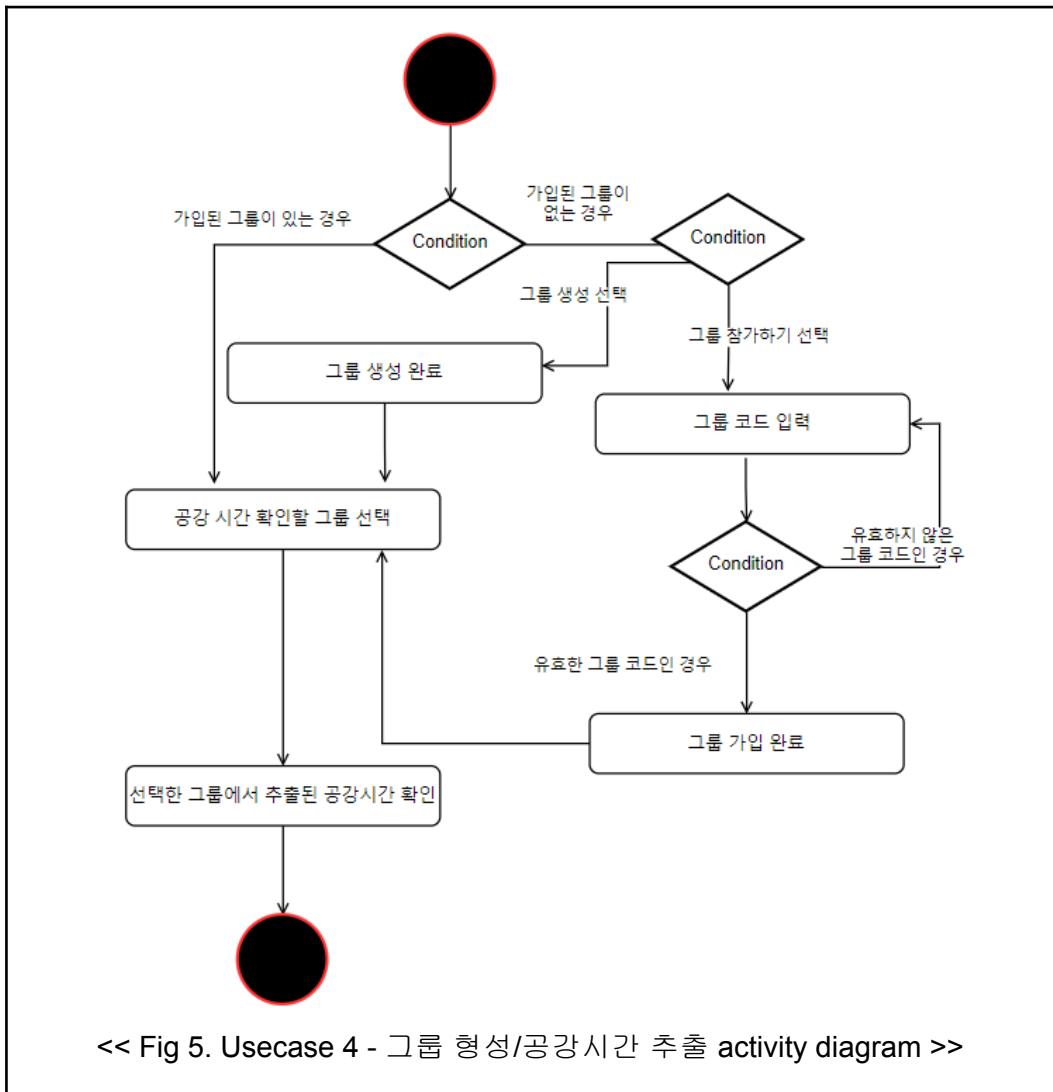


6.1.2 Usecase 2 : 개별 활동 추가

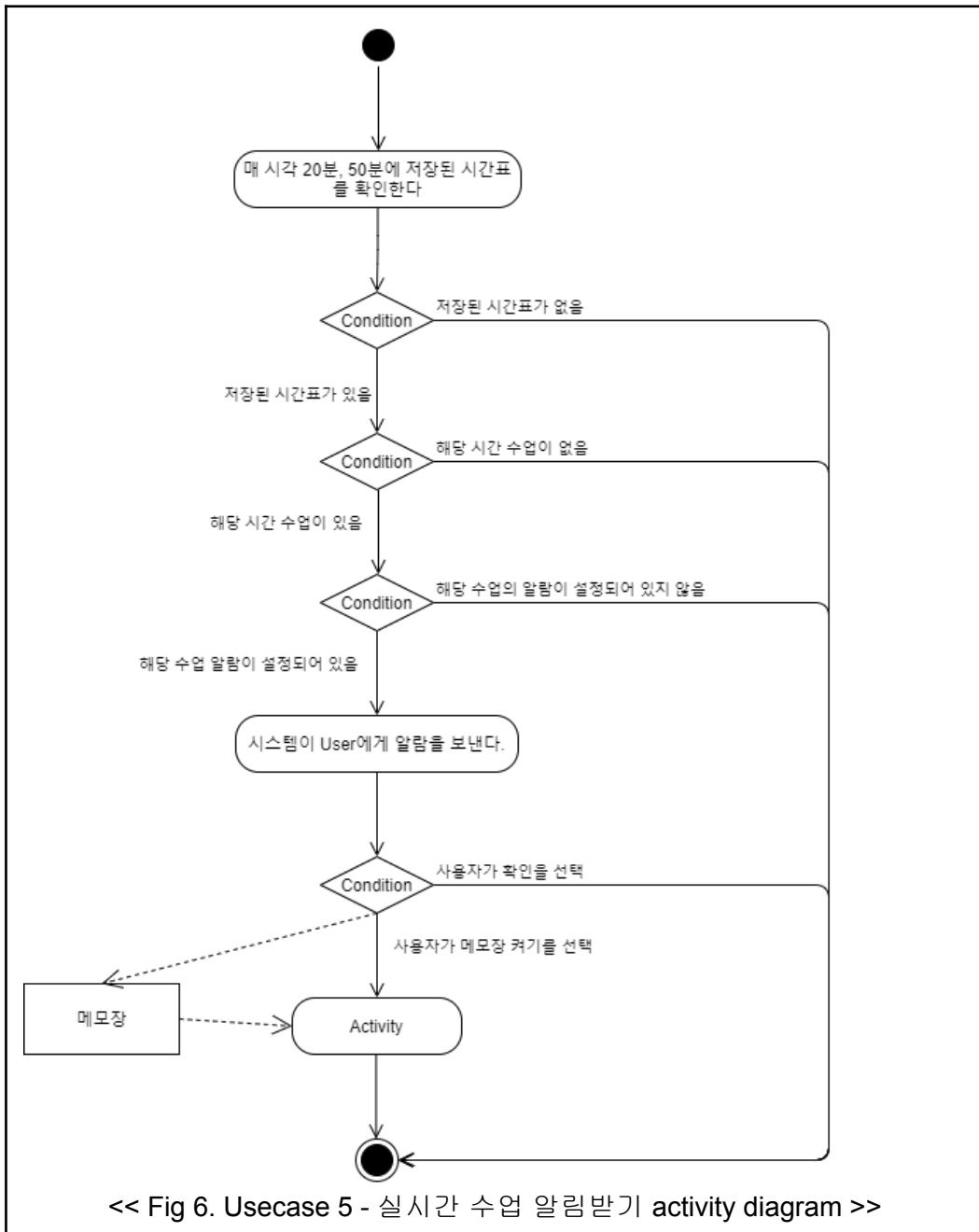


6.1.3 Usecase 3 : 학습 계획표 추천 ``` graph TD Start(()) --> Login[로그인(SUB_UseCase)] Login --> Add[개별 활동 추가(UC_002)] Add --> Click["'추천 받기' 버튼 클릭"] Click --> Decision{Condition} Decision -- "시간표, 추가한 개별 활동을 모두 고려한 추천 계획표가 존재" --> Output[추천 계획표 출력] Output --> SecondDecision{Condition} SecondDecision --> Store[추천 계획표 저장] SecondDecision --> End1((())) Decision -- "시간표, 추가한 개별 활동을 모두 고려한 추천 계획표가 존재X" --> End2((())) ``` << Fig 4. Usecase 3 - 학습 계획표 추천 activity diagram >> 25

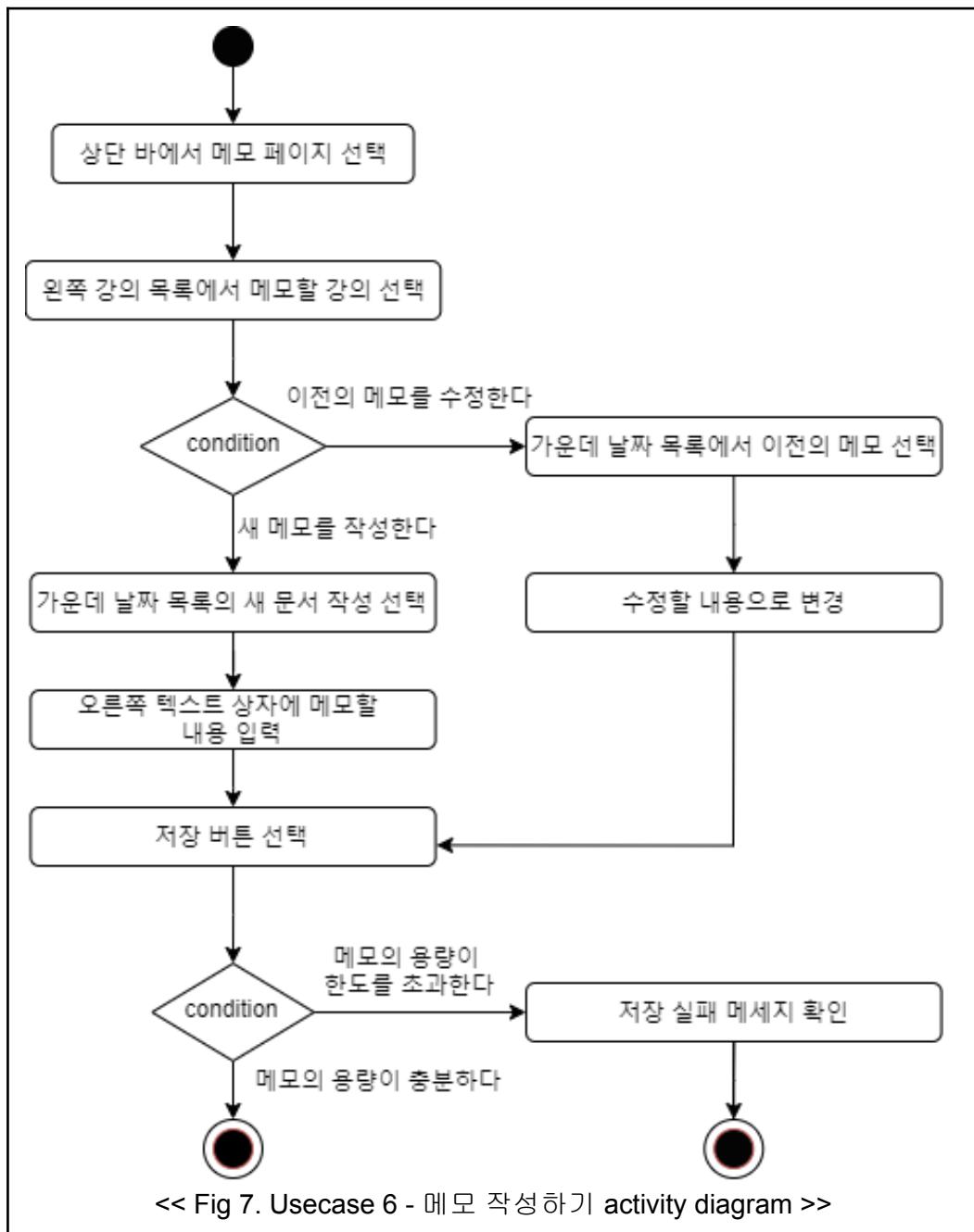
6.1.4 Usecase 4 : 그룹 형성 및 공강 시간 추출하기



6.1.5 Usecase 5 : 실시간 수업 알림 받기

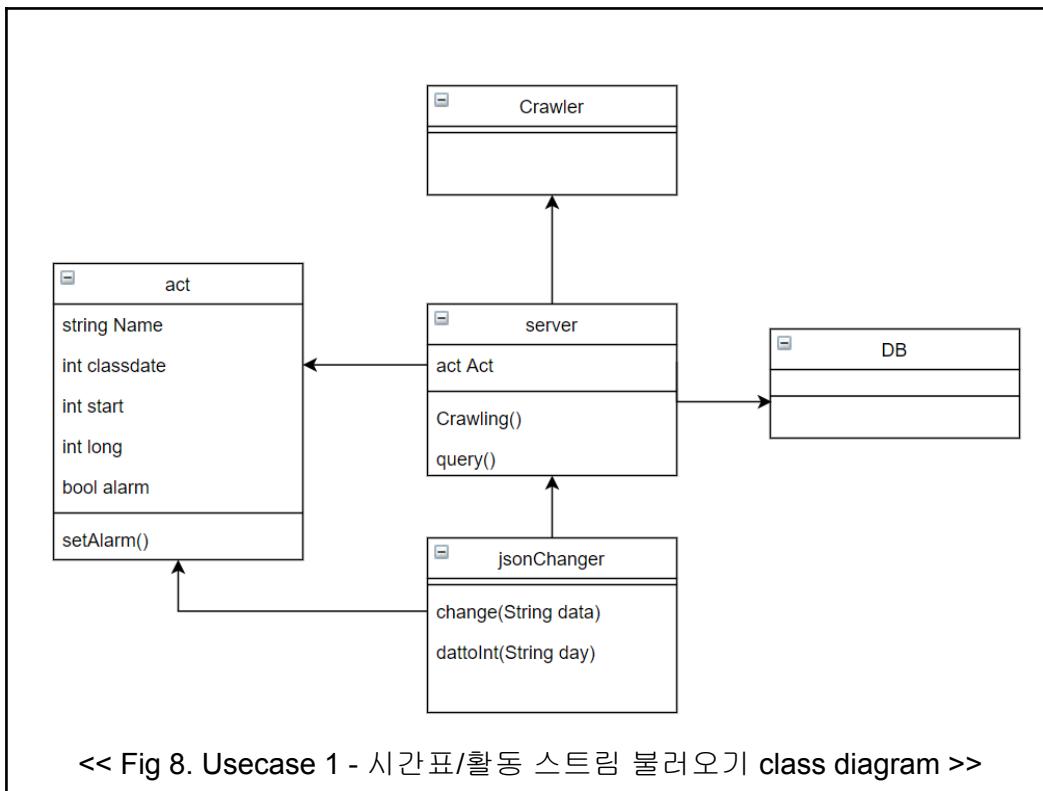


6.1.6 Usecase 6 : 메모 작성하기

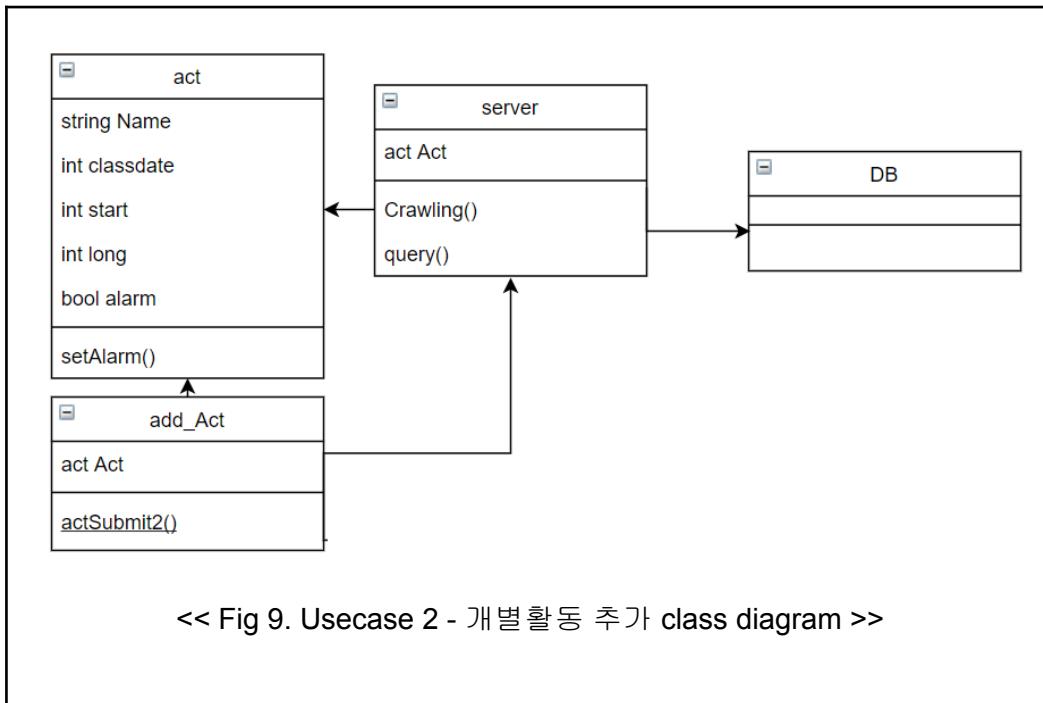


6.2 Structural Analysis

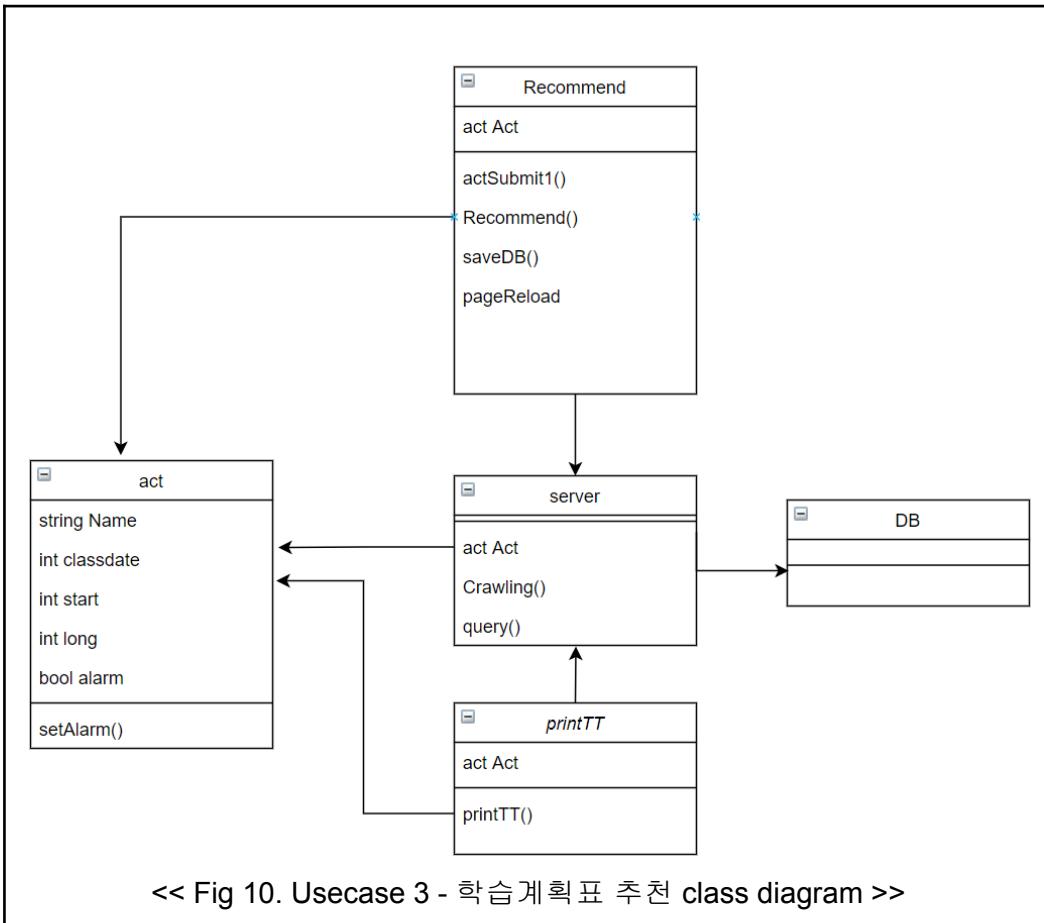
6.2.1 Usecase 1 : 시간표/활동 스트림 불러오기



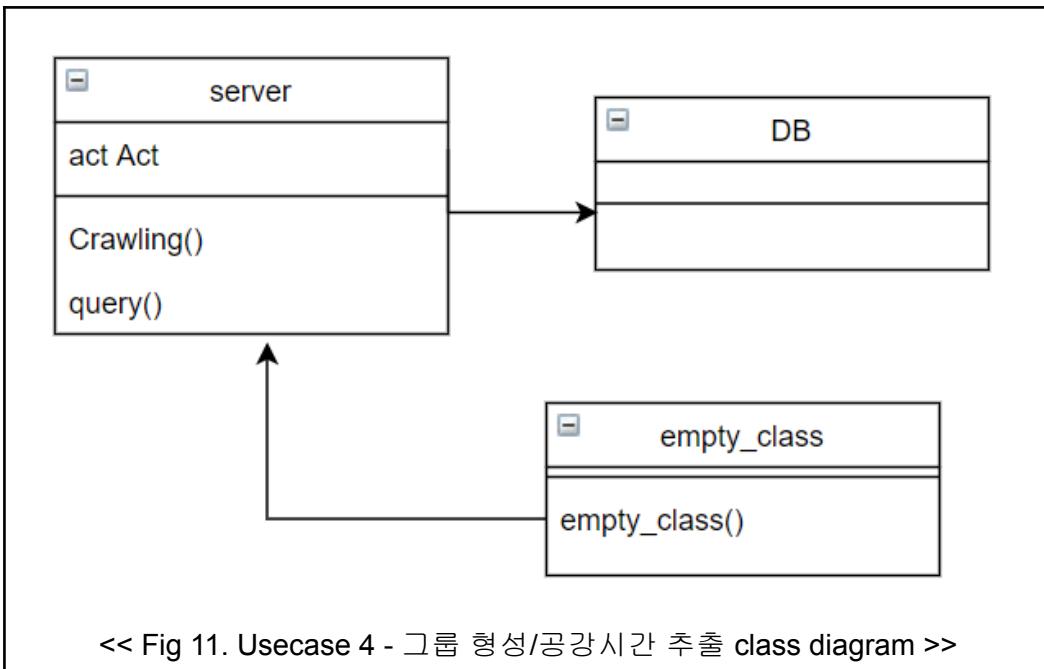
6.2.2 Usecase 2 : 개별 활동 추가



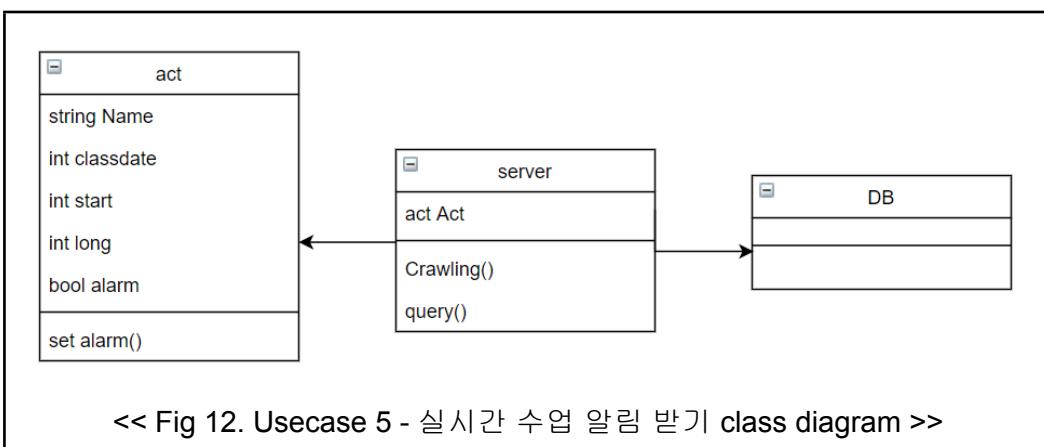
6.2.3 Usecase 3 : 학습 계획표 추천



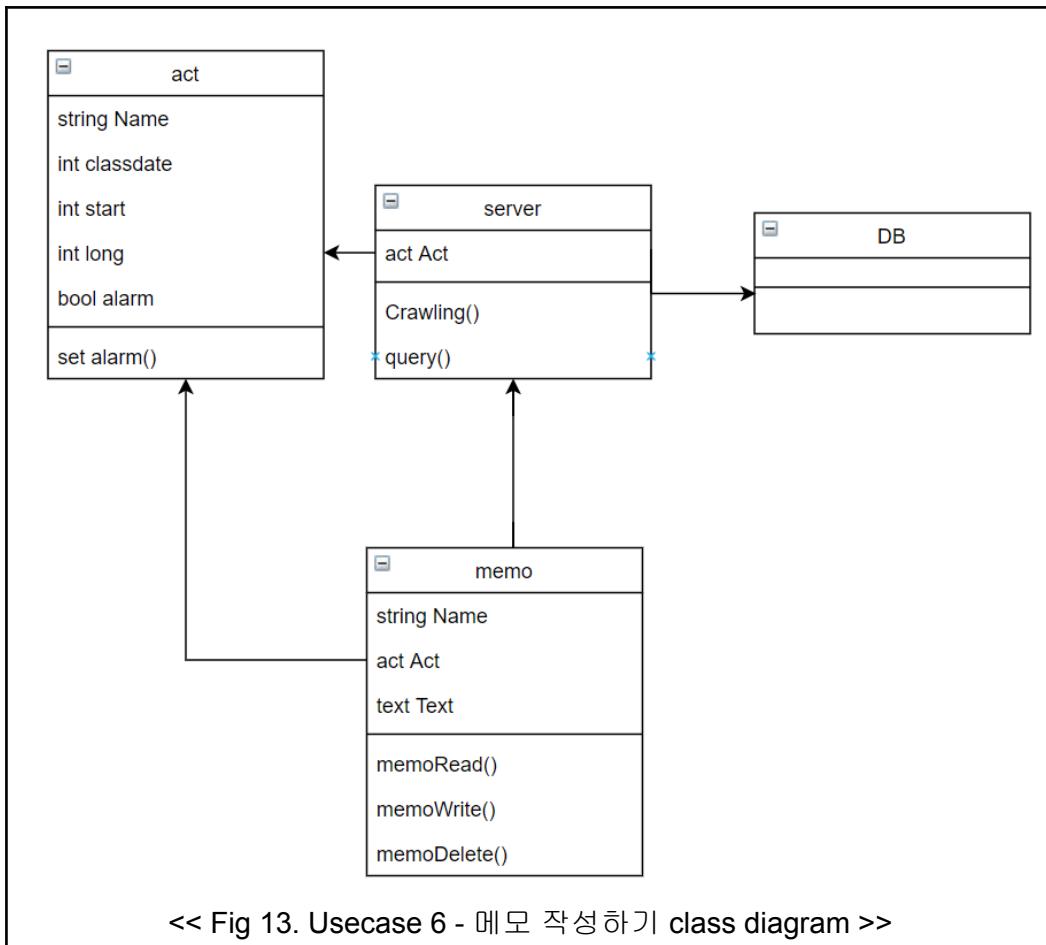
6.2.4 Usecase 4 : 그룹 형성 및 공통 공강 시간 추출하기



6.2.5 Usecase 5 : 실시간 수업 알림 받기



6.2.6 Usecase 6 : 메모 작성하기



7. Architectural Design

7.1 Logical View / Architecture

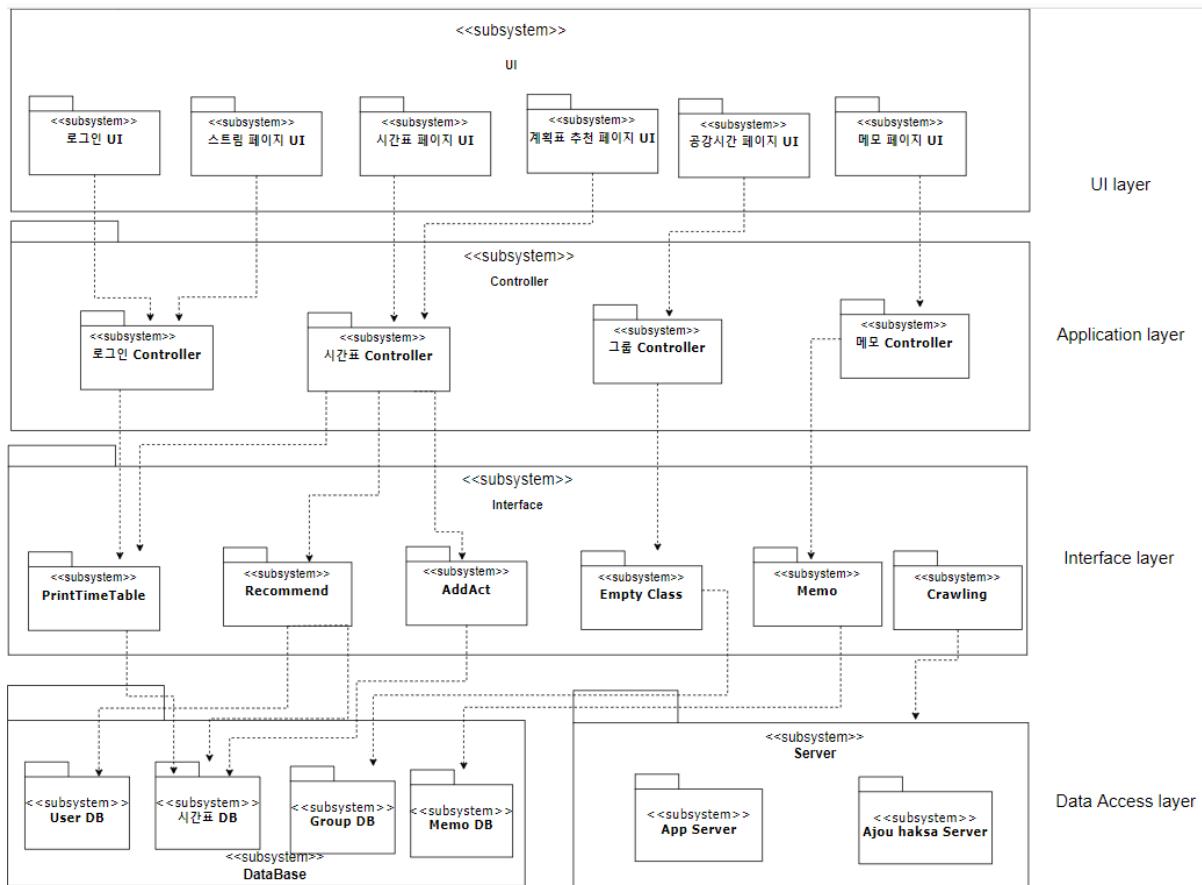
7.1.1 UI Layer

7.1.2 Application Layer

7.1.3 Interface Layer

7.1.4 Data Access Layer

7.1.5 Persistence Layer Description

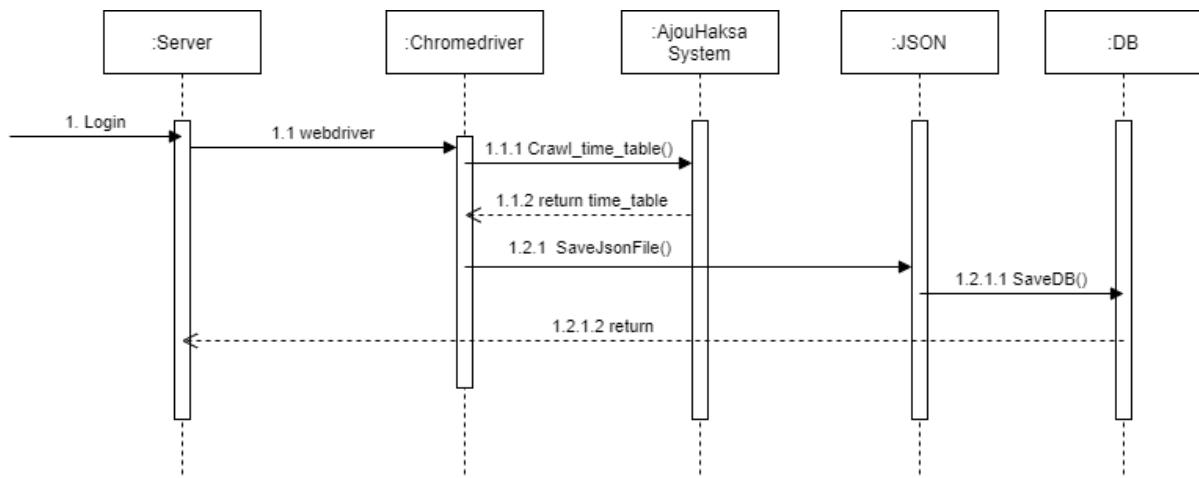


<< Fig 14. Logical view / Architecture >>

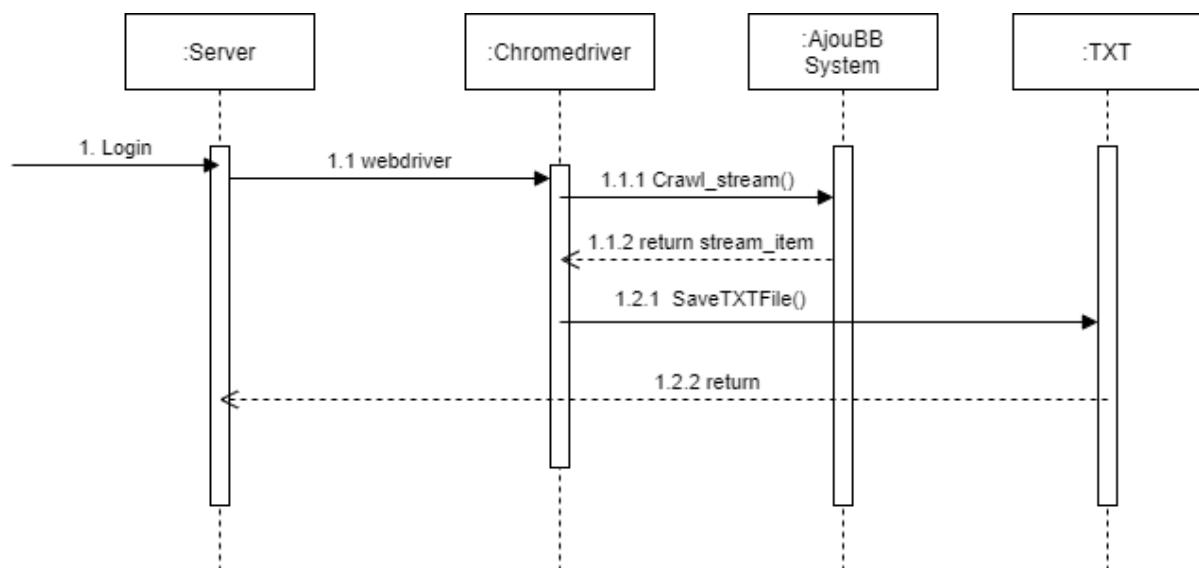
8. Design Modeling

8.1. Use Case 01 - 시간표와 활동스트림 연동

8.1.1. Design Sequence Diagram

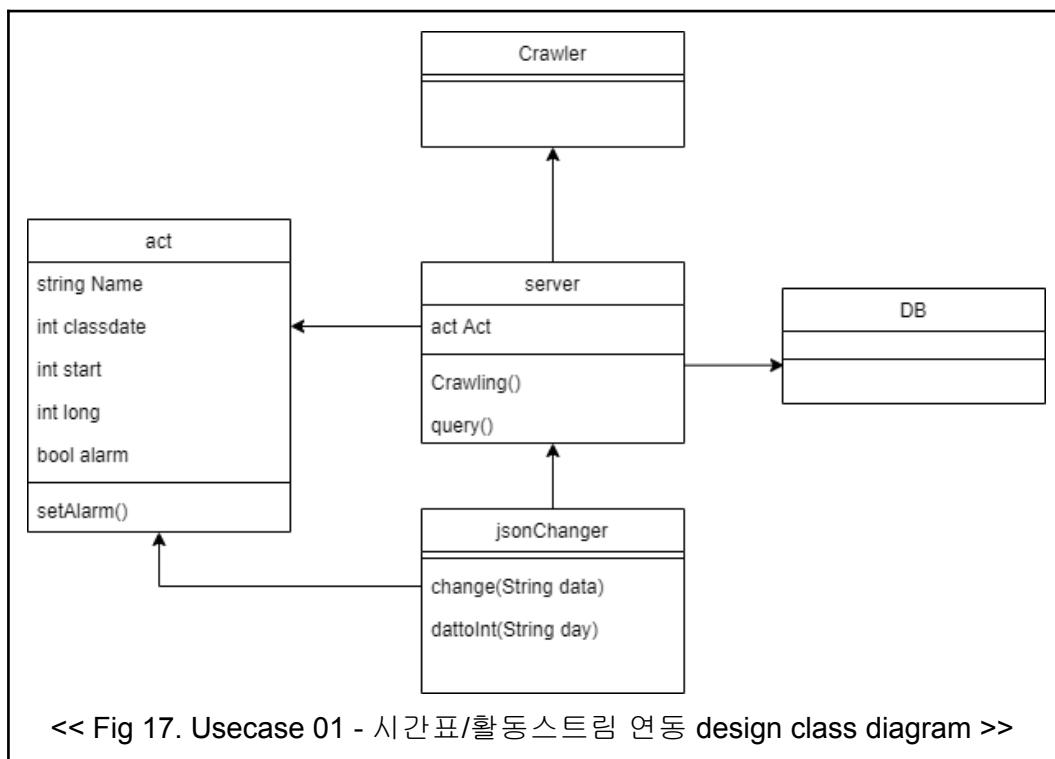


<< Fig 15. Usecase 01 - 시간표 연동 sequence diagram >>



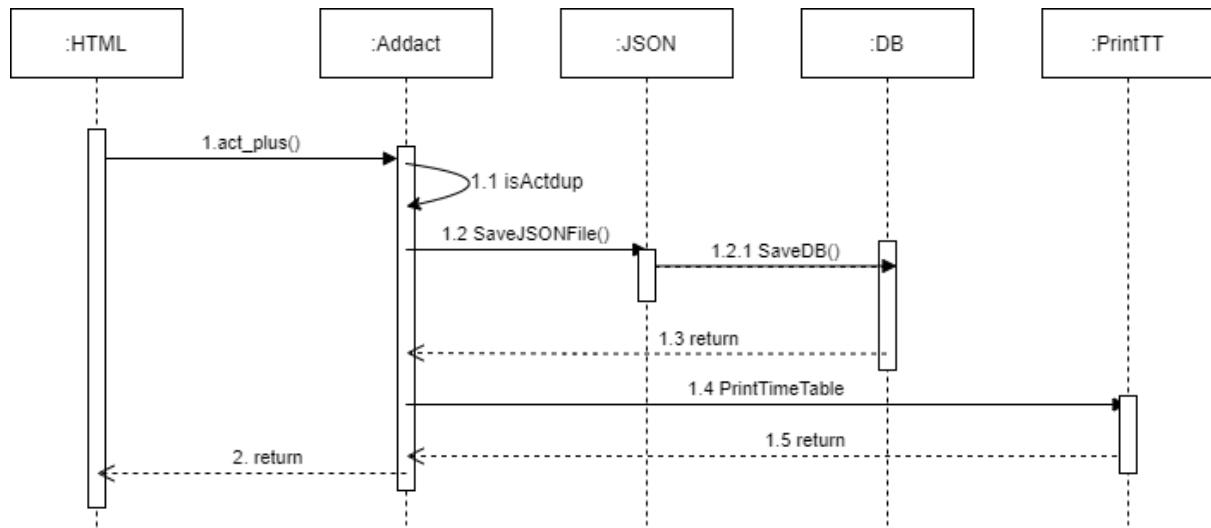
<< Fig 16. Usecase 01 - 활동스트림 연동 sequence diagram >>

8.1.2. Design Class Diagram

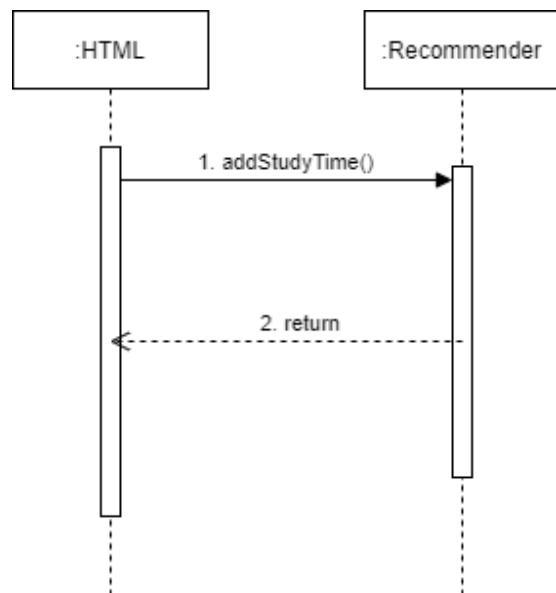


8.2. Use Case 02 - 개별 활동 추가 및 주당 학습시간 설정

8.2.1. Design Sequence Diagram

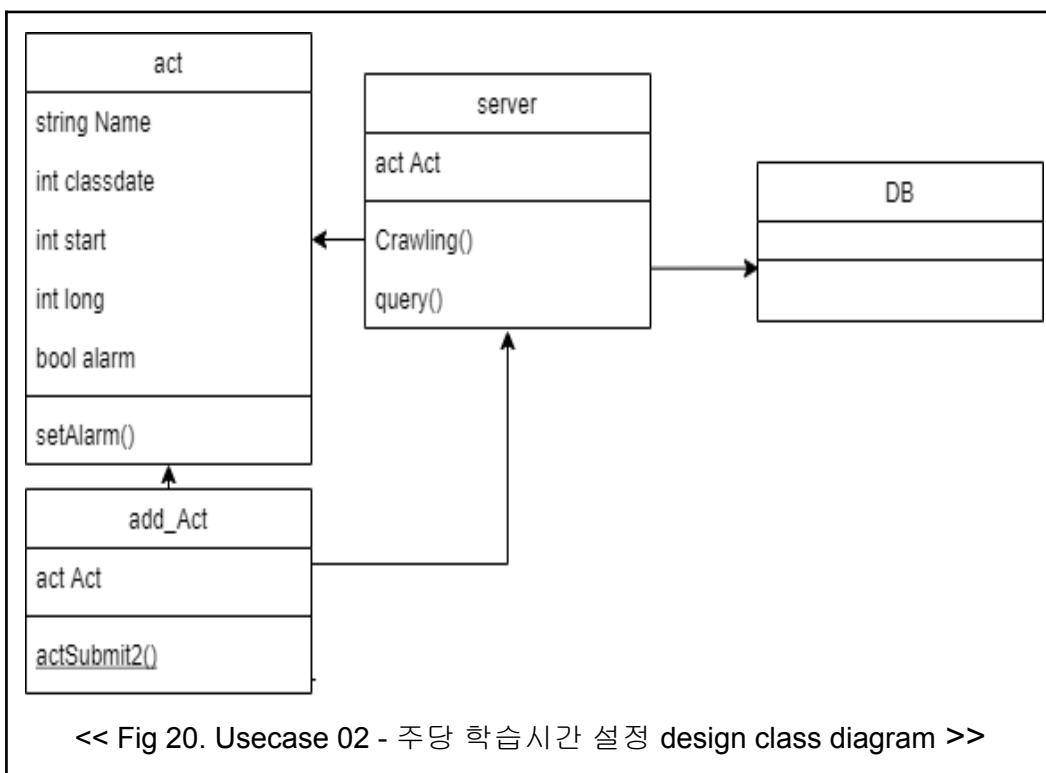


<< Fig 18. Usecase 02 - 개별활동 추가 sequence diagram >>



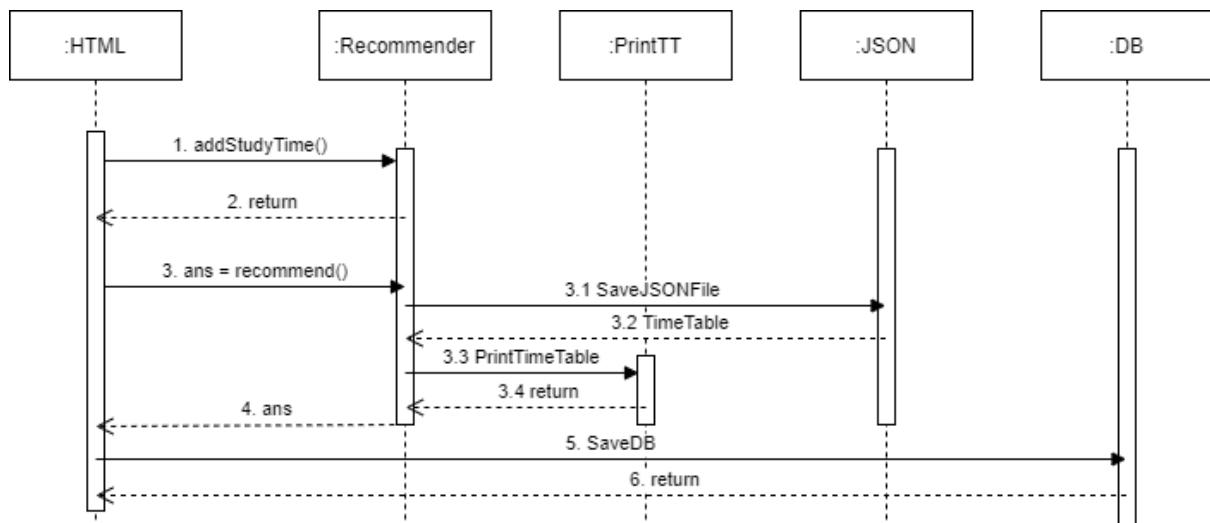
<< Fig 19. Usecase 02 - 주당 학습시간 설정 sequence diagram >>

8.2.2. Design Class Diagram



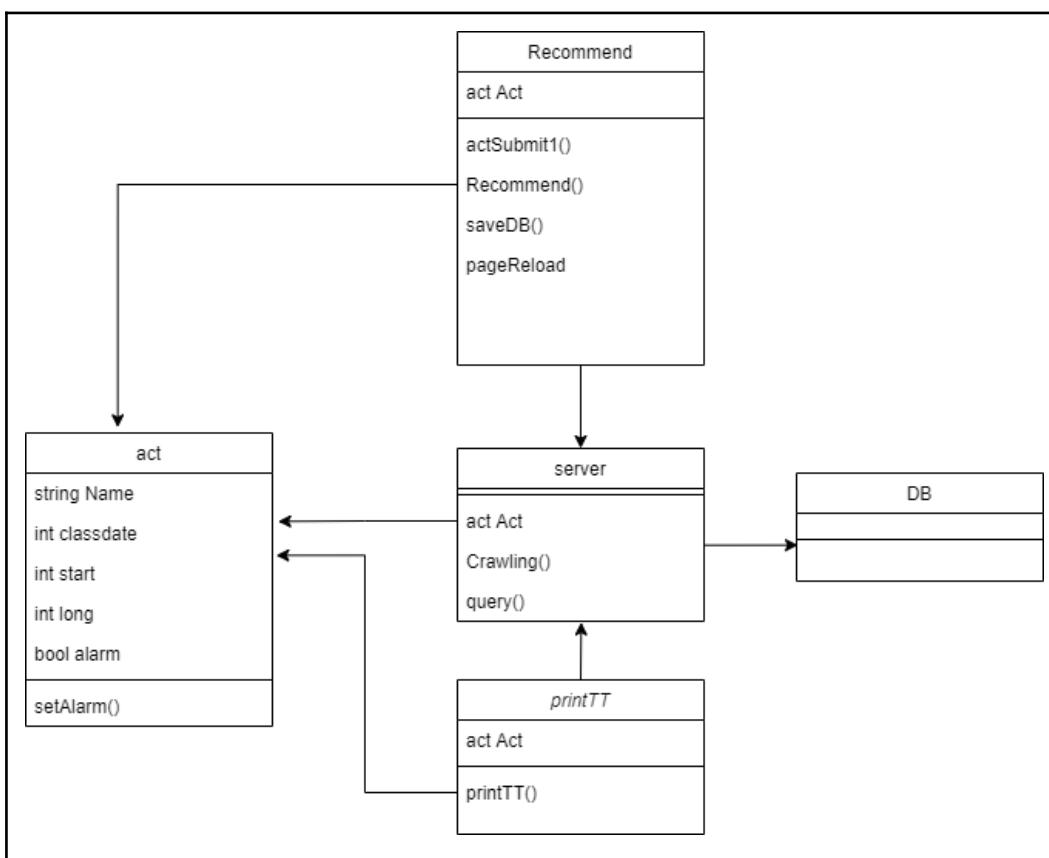
8.3. Use Case 03 - 학습 계획표 추천

8.3.1. Design Sequence Diagram



<< Fig 21. Usecase 03 - 학습계획표 추천 sequence diagram >>

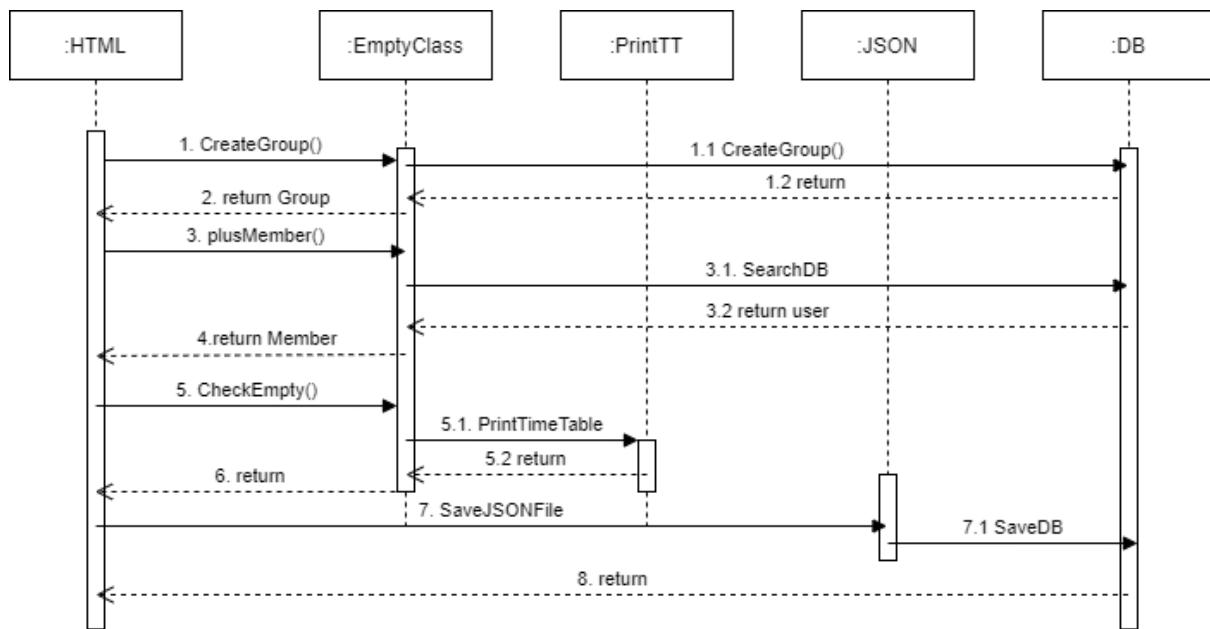
8.3.2. Design Class Diagram



<< Fig 22. Usecase 03 - 학습계획표 추천 design class diagram >>

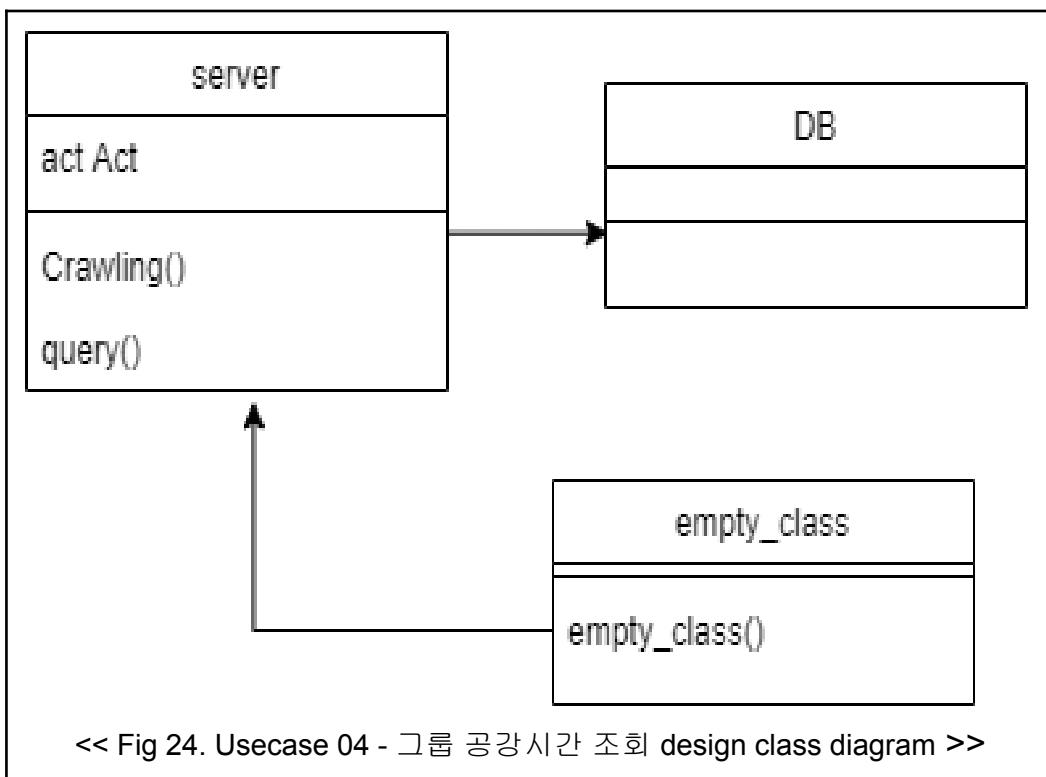
8.4. Use Case 04 - 그룹 공강시간 조회

8.4.1. Design Sequence Diagram



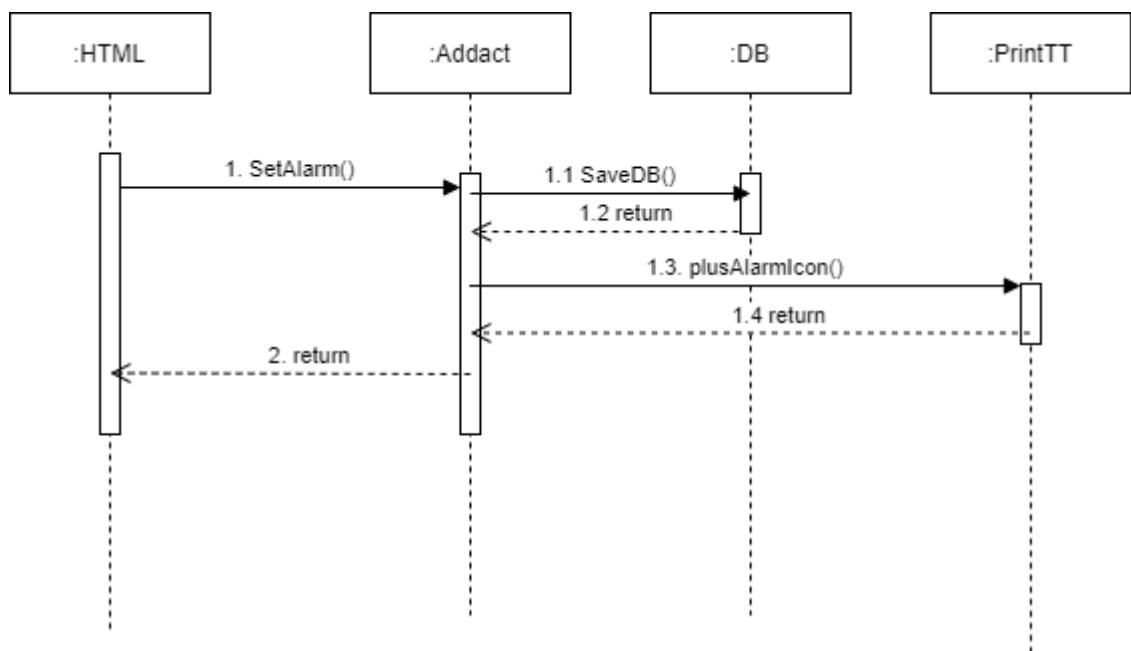
<< Fig 23. Usecase 04 - 그룹 공강시간 조회 sequence diagram >>

8.4.2. Design Class Diagram



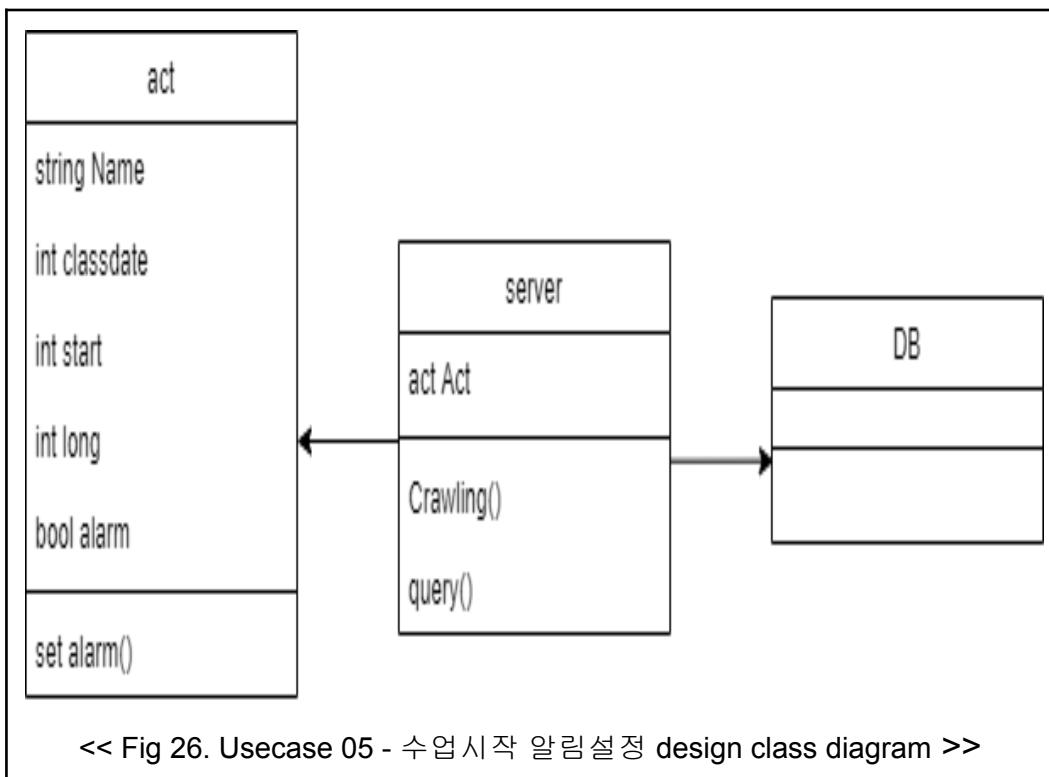
8.5. Use Case 05 - 수업 시작 알림 설정하기

8.5.1. Design Sequence Diagram



<< Fig 25. Usecase 05 - 수업 시작 알림설정 sequence diagram >>

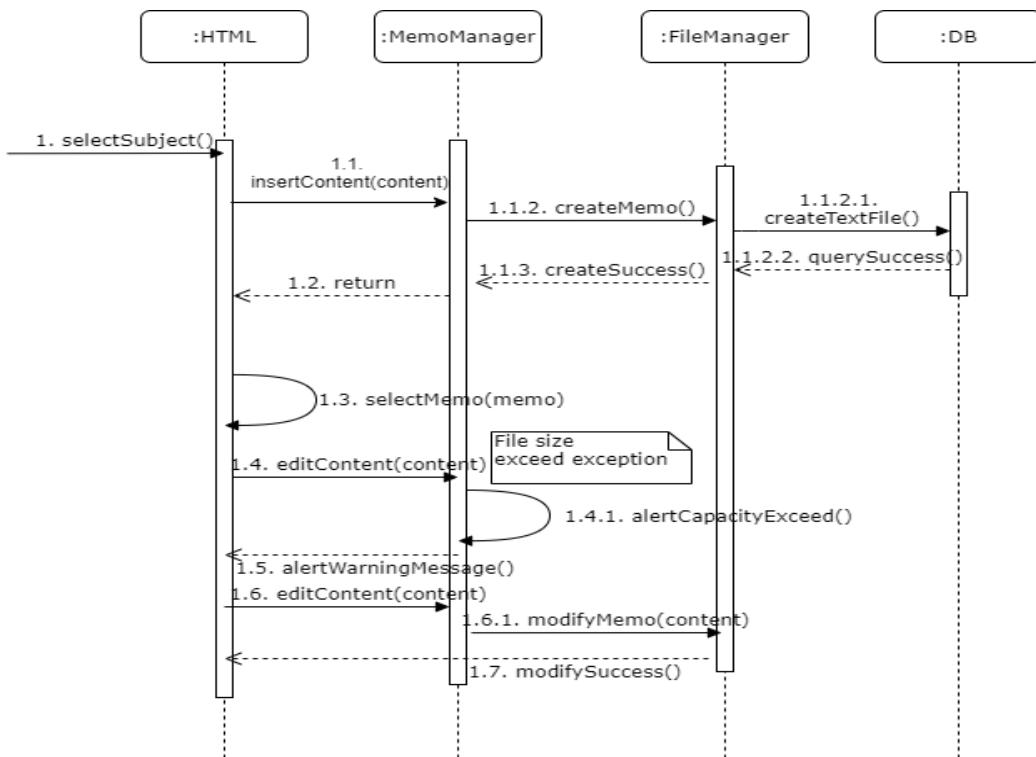
8.5.2. Design Class Diagram



<< Fig 26. Usecase 05 - 수업 시작 알림설정 design class diagram >>

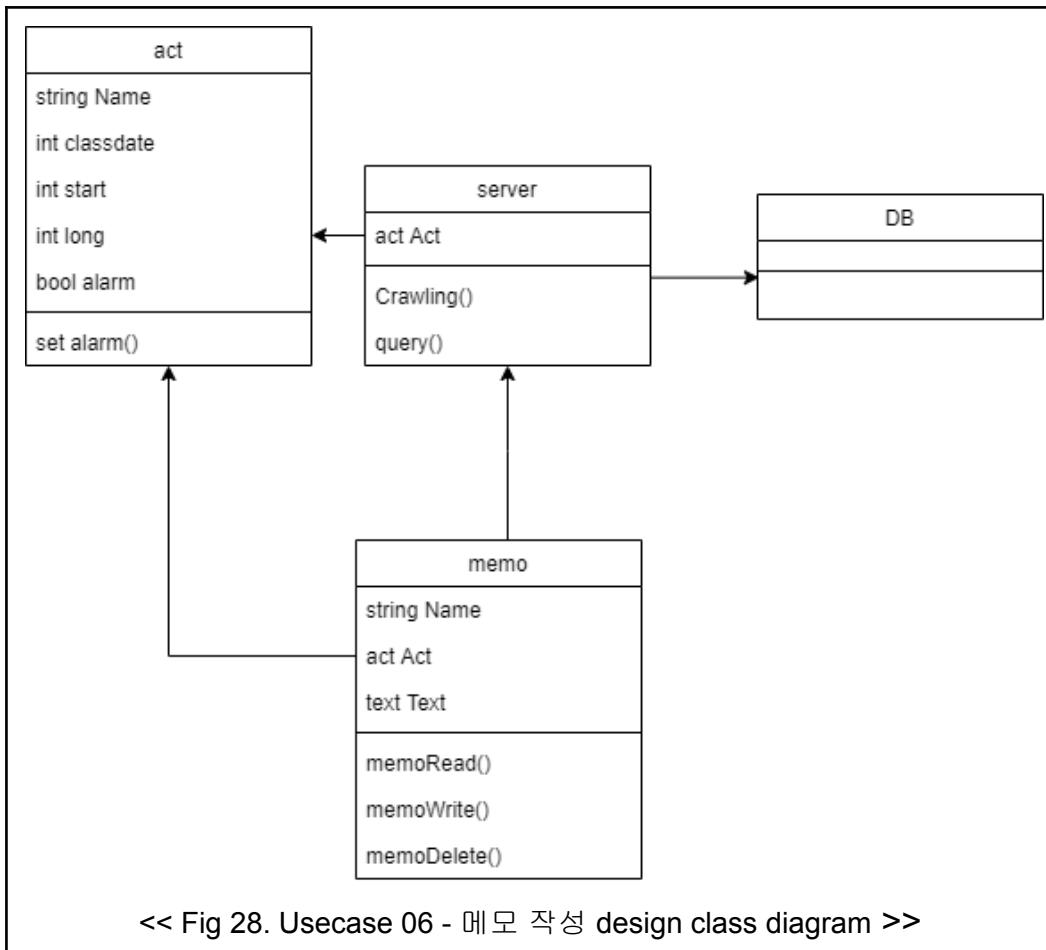
8.6. Use Case 06 - 메모 작성

8.6.1. Design Sequence Diagram

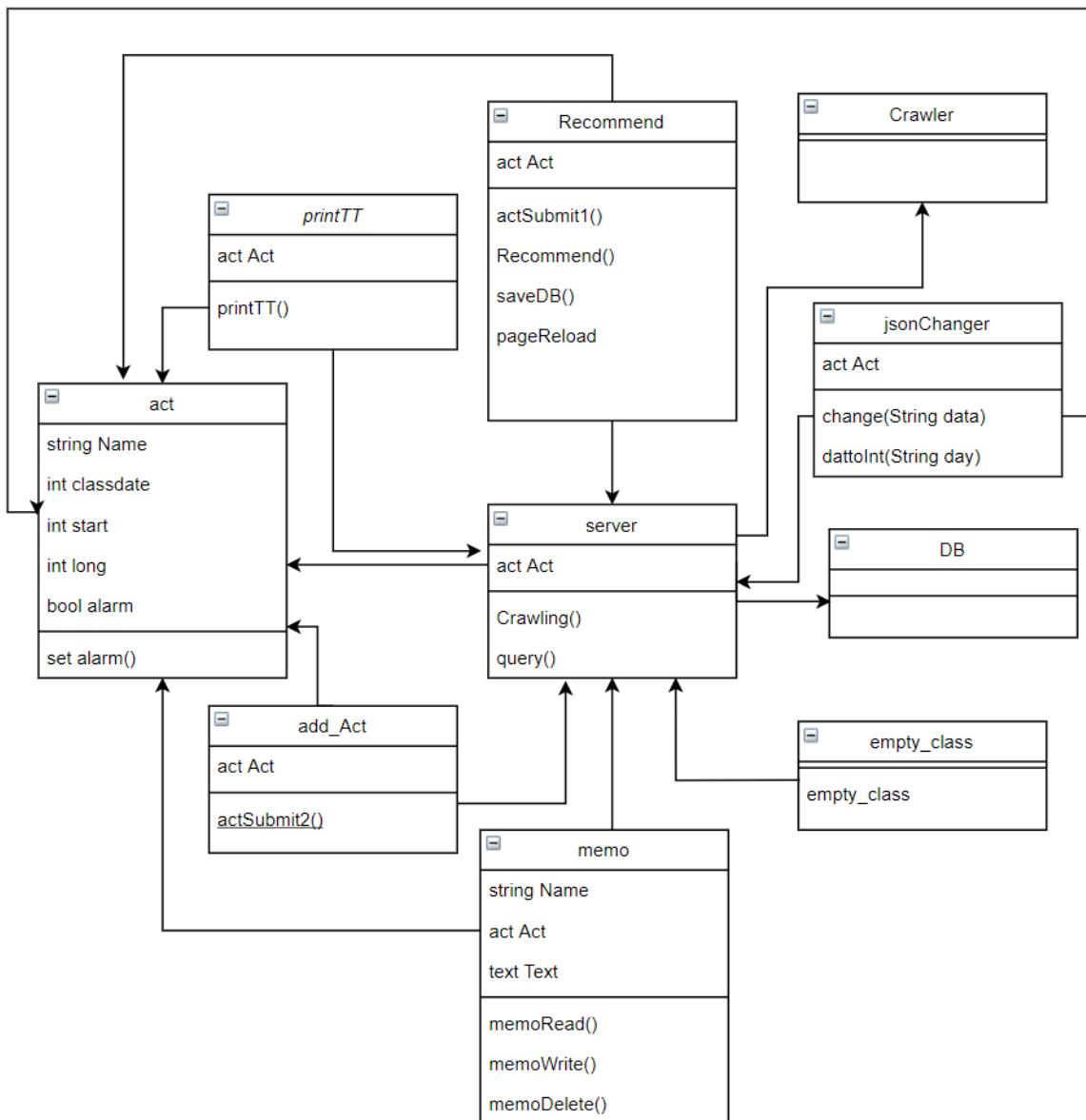


<< Fig 27. Usecase 06 - 메모 작성 sequence diagram >>

8.6.2. Design Class Diagram



8.7 Combined Design Class Diagram

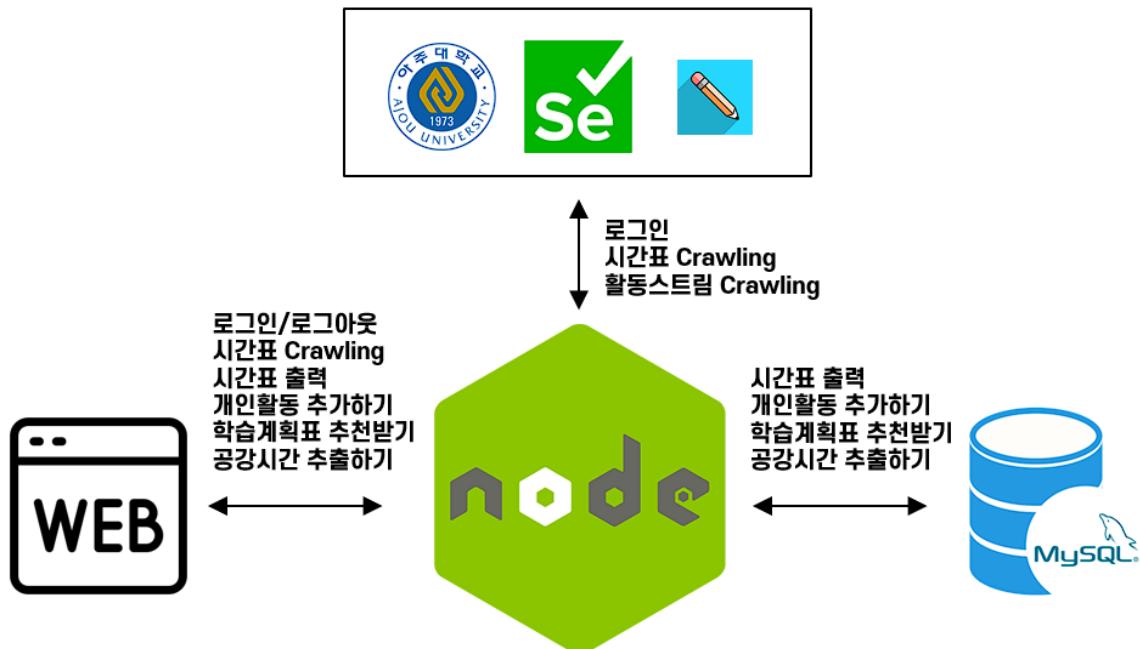


<< Fig 29. combined design class diagram >>

9. Implementation

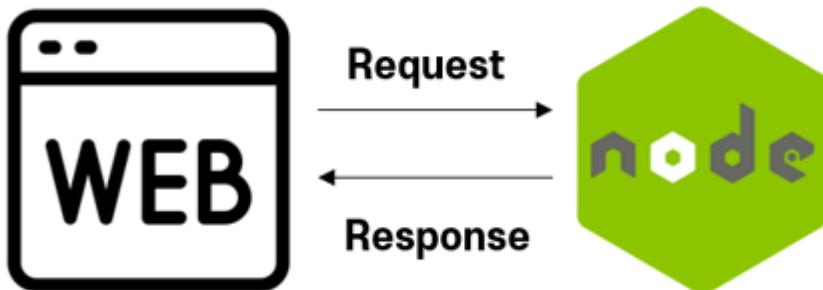
9.1. Implementation Specific

9.1.1. System Overview



<< Fig 30. System overview >>

9.1.2 AULIM Server(Web-node js)



<< Fig 31. Web/AULIM Server Implementation >>

위 그림은 AULIM Server의 구조이다. 사용자는 web page를 통해 AULIM의 기능을 수행할 수 있다. Web page와 node간의 통신은 로그인/로그아웃, 시간표 크롤링, 시간표 출력, 개인활동 추가하기, 학습계획표 추천받기, 공강시간 추출하기에서 사용된다.

9.1.2.1 로그인/로그아웃

로그인 버튼을 통해 사용자가 입력한 아이디와 비밀번호의 정보를 서버로 전송받고 세션에 저장하여 로그인을 진행하며 로그아웃 버튼을 누를 시 세션을 삭제한다.

9.1.2.2 시간표 출력

`printTable()` - 서버에서 시간표 정보를 받아 Web page에 출력한다.

9.1.2.3 개인활동 추가하기

`addSubmit2()` - 사용자가 Web Page에서 입력한 추가활동 정보를 서버로 전송한다.

9.1.2.4 학습계획표 추천받기

`addSubmit1()` - 사용자가 Web Page에서 추천 계획표를 받기 위해 입력한 추가 활동정보를 임시로 저장해둔다.

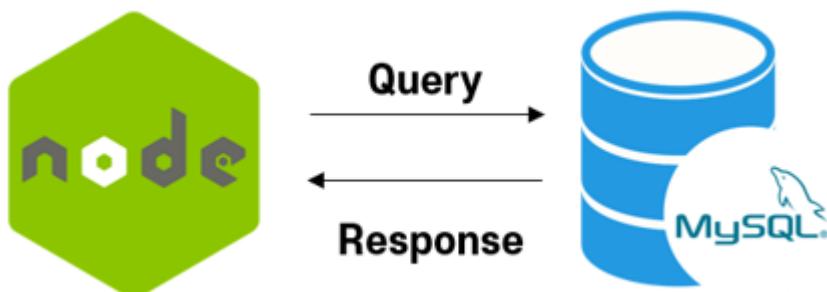
`Recommend()` - `addSubmit1` 함수로 저장해둔 추가 활동정보를 기준 시간표의 빈공간에 입력하여 추천계획표를 출력해주는 함수이다.

`pageReload()` - 사용자가 `Recommend()`로 추천받은 계획표가 마음에 들지 않아 다시 추천받기 위해서 초기화 시키기 위한 함수이다. 새로고침을 통해 입력해둔 추가 활동정보를 초기화 시킨다.

9.1.2.5 공강시간 추출하기

`empty_class` - 사용자가 속한 그룹원들 간의 시간표를 비교하여 그룹원들이 모두 공강인 시간을 찾아 출력해주는 함수이다.

9.1.3 AULIM Server(node js-Mysql)



<< Fig 32. AULIM Server/MySQL Implementation >>

사용자가 요청한 데이터를 `node.js`와 `Mysql`을 통해 DB에서 찾고 전송한다. 시간표 출력, 개인활동 추가하기, 학습계획표 추천받기, 공강시간 추출하기에서 사용된다.

9.1.3.1 시간표 출력

`printTable()` - MySql이 서버로부터 Query문을 받아 DB에서 사용자의 시간표정보를 찾아 서버로 전송한다.

9.1.3.2 개인활동 추가하기

`addSubmit2()` - MySql이 서버로부터 Query문을 받아 DB에 새로운 활동을 추가한다.

9.1.3.3 학습계획표 추천받기

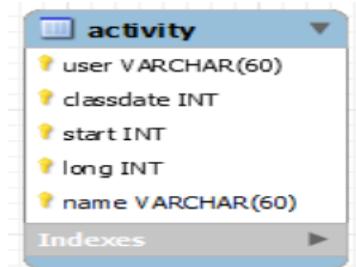
`saveDB()` - MySql이 서버로부터 Query문을 받아 Recommend() 함수로 추천받은 시간표를 DB에 저장한다.

9.1.3.4 공강시간 추출하기

`empty_class` - MySql이 서버로부터 사용자가 속한 그룹들에 대한 query문을 받아 DB에서 사용자가 속한 그룹원들의 시간표를 서버로 전송한다.

[Table]

기존에 고려했던 Table들 중 현재 구현된 기능에 사용되는 Table인 activity Table만 사용하는 것으로 결정하였다.



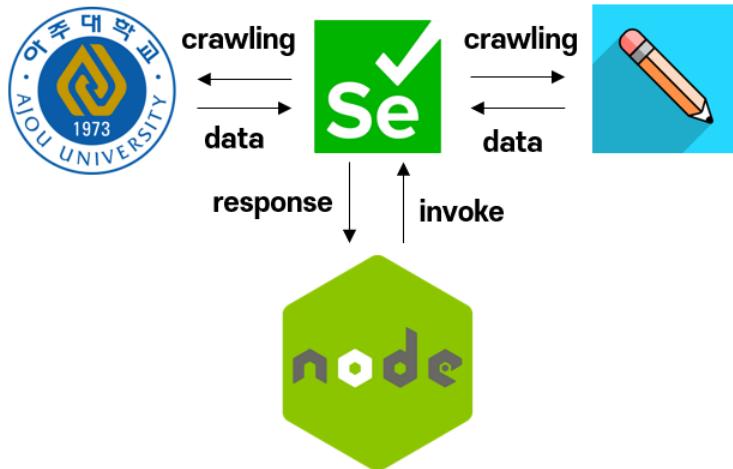
<< Fig 33. Table implementation >>

[Table: activity]

Attribute	Type	Description
user	VARCHAR(60)	활동 주체 데이터
classdate	INT	활동요일 데이터 AULIM 시스템에서 사용하기 쉽도록 정수 변환
start	INT	활동 시작 시간 데이터 AULIM 서비스에서 사용하기 쉽도록 정수 변환
long	INT	활동 지속 시간 데이터 AULIM 서비스에서 사용하기 쉽도록 정수 변환

name	VARCHAR(60)	활동 이름 데이터
------	-------------	-----------

9.1.4 Crawler



<< Fig 34. Crawler implementation >>

9.1.4.1 로그인

9.1.4.2 시간표 크롤링

시간표 가져오기 버튼을 누르면 서버 저장소의 파이썬 스크립트를 내부적으로 실행하여 현재 세션의 유저 정보를 바탕으로 Selenium 라이브러리를 활용해 외부 시스템인 아주 학사 서비스에 로그인 한 뒤 시간표 페이지까지 접근하여 시간표 정보를 JSON 형태로 서버로 전달한다.

9.1.4.3 활동스트림 크롤링

사용자가 로그인하면 9.1.4.2. 에서와 마찬가지로 Selenium 라이브러리를 활용해 외부 시스템인 아주BB 서비스에 로그인 한 뒤 활동스트림 정보를 얻을 수 있는 페이지까지 접근 한 뒤 활동스트림 정보를 서버 저장소에 텍스트 형태로 저장한다.

9.2. Message Format

AULIM 시스템과 아주대학교 학사서비스 시스템 간의 통신에서 문자를 크롤링하여 JSON 파일 형태로 만들어 전송하며 JSON의 format은 아래 표와 같다.

Metho d	Path	Message Format(JSON)	Description
crawl_ time_ t able	/data	<pre>{ "activities": { "name": "소프트웨어 공학", "classdate": 2, "start": 7, "long": 3 } }</pre> <p><<Fig 35. Message format implementation example>></p>	아주대학교 학사서비스에서 시간표 정보를 담은 파일이다. name은 과목명이다. classdate는 수업요일, start는 수업시작 시간, long은 수업길이를 의미하며 AULIM 시스템과 연동되도록 int형으로 변환되어 저장된다.

node js와 DB간의 통신은 query 문을 문자열의 형태로 전송한다.

9.3. System/Data Dependencies & Requirements

AULIM Server

- Node js 14.17.0
- Mysql community 8.0.24
- npm packages (cookie parser, ejs, express, express-session, jquery, mysql, nodemon, python-shell, socketio, websocketio)

Web Driver

- Chrome driver version 89 / 91

10. UI Design

Use Case 01 : 시간표와 활동스트림 연동

1. ID와 PW를 입력해 로그인을 한다.

The screenshot shows the AULIM application interface. At the top, there is a navigation bar with links for Stream, Schedule, Public Time, Plan Selection, Memo, Logout, and Schedule Import. Below the navigation bar is a title '시간표' (Schedule). To the right of the title is a login form with fields for 'ID' (wnsr0721) and 'PW' (redacted), and buttons for 'Login' and 'Register'. A red box highlights the 'ID' and 'PW' input fields. On the left side of the main area is a weekly schedule grid from Monday to Sunday, with time slots from 09:00 to 14:00. On the right side, there is a sidebar titled '새로운 활동 추가' (Add New Activity) with dropdown menus for '활동명' (Activity Name), '날짜' (Date) set to '월요일' (Monday) at '9시' (9 AM) and '0분' (0 minutes), and a '길이' (Length) of '1' hour, followed by a 'submit' button.

<< Fig 36. ID/PW 입력 >>

2. 로그인이 성공하면 자동으로 활동스트림을 불러온다.

The screenshot shows the AULIM application interface after login. The navigation bar and title are the same as in Fig 36. The schedule grid now shows some occupied time slots. The sidebar on the right is still visible, showing the '새로운 활동 추가' (Add New Activity) form with the date and time set to '9시' (9 AM) and '0분' (0 minutes).

<< Fig 37. 로그인 후 배너 모습 >>

추가

2021. 5. 19. 15:17 SCE333_ 데이터베이스(F054-2) 마감: 과제4 - 수정 21. 6. 15. 23:59
 2021. 6. 1. 15:33 SCE433_ 컴퓨터그래픽스(F018-1) 마감: Final Project (20%) 21. 6. 20. 23:59
 2021. 5. 27. 17:55 SCE337_ 소프트웨어공학(F048-1) 오늘 마감: Final Document assignment -updated 6/5 21. 6. 7.
 2021. 6. 4. 11:43 SCE337_ 소프트웨어공학(F048-1) 오늘 마감: Final presentation 21. 6. 7. 23:59
 2021. 6. 4. 11:47 SCE337_ 소프트웨어공학(F048-1) 오늘 마감: Final demo code 21. 6. 7. 23:59
 2021. 6. 4. 11:47 SCE337_ 소프트웨어공학(F048-1) 추가됨: Final demo code 21. 6. 7. 23:59
 2021. 6. 4. 11:43 SCE337_ 소프트웨어공학(F048-1) 추가됨: Final presentation 21. 6. 7. 23:59
 2021. 6. 4. 11:01 SCE337_ 소프트웨어공학(F048-1) 추가됨: Code inspection form 21. 6. 4. 23:59
 2021. 6. 2. 7:44 JAP101_ 일본어1(X286-3) 추가됨: 14주차 이야기가 있는 드로잉 21. 6. 4. 23:59
 2021. 6. 2. 7:44 JAP101_ 일본어1(X286-3) 추가됨: 음독과제8: 18과회화본문 21. 6. 7. 23:59
 2021. 6. 1. 15:33 SCE433_ 컴퓨터그래픽스(F018-1) 추가됨: Final Project (20%) 21. 6. 20. 23:59
 2021. 6. 1. 10:58 SCE322_ 컴퓨터통신(F011-1) 추가됨: 과제 6 21. 6. 7. 23:59
 2021. 6. 1. 9:22 SCE3315_ 기계학습(F056-2) 추가됨: 수업요약(14주차) 21. 6. 3. 23:59

<< Fig 38. 스트림 페이지 이동 모습 >>

3. ‘시간표 불러오기’ 버튼을 누르면 시간표를 불러온다.

시간표

교시/요일	월	화	수	목	금	토
09:00						
09:30		드로잉 연습		컴퓨터통신	드로잉 연습	
10:00						
10:30						
11:00	컴퓨터그래픽스	컴퓨터통신		컴퓨터그래픽스		
11:30						
12:00						
12:30	데이터베이스	소프트웨어공학	데이터베이스		소프트웨어공학	
13:00						
13:30						
14:00			일본어1		일본어1	
14:30						
15:00						
15:30						
16:00						
16:30						
17:00			기계학습		기계학습	
17:30						
~~~						

새로운 활동 추가

활동명:

날짜:

월요일  9시  0분

길이:  submit

&lt;&lt; Fig 39. 시간표 불러오기 버튼을 누른 후의 모습 &gt;&gt;

## Use Case 02 : 개별 활동 추가 & 주당 학습시간 설정

1. ‘시간표’ 메뉴의 ‘새로운 활동 추가’ 박스에 개별 활동을 추가한다.

1.1 상단바 위 ‘시간표’ 메뉴 클릭



<< Fig 40. 시간표 메뉴의 모습 >>

1.2 개별 활동 추가

A screenshot of a 'New Activity Addition' form. The form has a light gray background and a thin gray border. Inside, there is a title '새로운 활동 추가' in bold black font. Below it is a field labeled '활동명:' with a text input box containing '수학1'. Next is a section labeled '날짜:' with three dropdown menus: '월요일' (Monday), '14시' (14:00), and '0분' (0 minutes). Below that is a section labeled '길이:' with a dropdown menu showing '3'. At the bottom is a 'submit' button.

<< Fig 41. 개별 활동을 입력한 모습 >>

2. 시간표에 개별활동이 추가되었는지 확인한다.

AULIM

스트림 시간표 공강시간 계획표 추천 강의 메모 Logout 시간표 가져오기

## 시간표

교시/요일	월	화	수	목	금	토
09:00						
09:30		드로잉 연습		컴퓨터통신	드로잉 연습	
10:00						
10:30						
11:00	컴퓨터그래픽스	컴퓨터통신		컴퓨터그래픽스		
11:30						
12:00						
12:30	데이터베이스	소프트웨어공학	데이터베이스		소프트웨어공학	
13:00						
13:30						
14:00			일본어1		일본어1	
14:30	수학1					
15:00						
15:30						
16:00						
16:30						
17:00			기계학습		기계학습	
17:30						
18:00						

새로운 활동 추가

활동 명: 수학1

날짜:

월요일 ▼ 14시 ▼ 0분 ▼

길이: 3 ▼

submit

<< Fig 42. 개별 활동 추가 버튼을 누른 뒤의 모습 >>

3. ‘계획표 추천’ 메뉴의 ‘할당 시간 설정’ 박스에 학습 시간을 설정한다.

할당 시간 설정

활동 명: 토익공부

시간: 4 ▼

button

추가할 학습 활동:

추천 받기 저장 취소

<< Fig 43. 학습 시간을 입력한 모습 >>

4. ‘button’ 을 눌러 학습시간이 추가되는지 확인한다.

할당 시간 설정

활동명: 토익공부

시간: 4  
button

추가할 학습 활동:  
토익공부 4시간  
추천받기 저장 취소

<< Fig 44. 입력한 학습시간을 추가한 모습 >>

### Use Case 03 : 학습 계획표 추천

1. ‘추천받기’ 버튼을 눌러 추가한 학습시간이 시간표에 추가되는지 확인한다.

교시/요일	월	화	수	목	금	토
09:00						
09:30						
10:00						
10:30						
11:00	컴퓨터그래픽스	컴퓨터통신				
11:30						
12:00						
12:30	데이터베이스	소프트웨어공학	데이터베이스			
13:00						
13:30						
14:00			일본어1			
14:30	수학1					
15:00						
15:30						
16:00						
16:30						
17:00			기계학습			
17:30						
18:00						
18:30						
19:00						
19:30						
20:00						
20:30						
21:00						
21:30						
22:00						
22:30	토익공부		토익공부			
23:00						
23:30						
24:00						

<< Fig 45. 추천받기 버튼을 눌러 추천 학습 시간표가 출력된 모습 >>

2. 추천된 학습 시간표를 선택하려면 저장 버튼을, 거부하려면 취소 버튼을 누른다.  
- 취소 버튼 선택 시

## time_table

- 기본 시간표 보기
- 추가일정 포함한 시간표 보기

교시/요일	월	화	수	목	금	토
09:00						
09:30		드로잉 연습		컴퓨터통신	드로잉 연습	
10:00						
10:30						
11:00	컴퓨터그래픽스	컴퓨터통신		컴퓨터그래픽스		
11:30						
12:00						
12:30	데이터베이스	소프트웨어공학	데이터베이스		소프트웨어공학	
13:00						
13:30						
14:00			일본어1		일본어1	
14:30	수학1					
15:00						
15:30						
16:00						
16:30						
17:00		기계학습			기계학습	
17:30						
18:00						
18:30						
19:00						
19:30						
20:00						
20:30						
21:00						
21:30						
22:00						
22:30						
23:00						
23:30						

<< Fig 46. 취소 버튼을 눌러 이전 시간표로 되돌아간 모습 >>

## Use Case 04 : 그룹 공강시간 조회

1. ‘공강시간’ 메뉴의 그룹 박스 안에 있는 ‘공강 시간 확인’ 버튼을 누른다.

### 1.1 ‘공강시간’ 메뉴 클릭



<<Fig 47. 공강시간 메뉴의 모습>>

1.2 그룹 박스에 있는 ‘공강시간 확인’ 버튼을 누른다.

A screenshot of the AULIM application showing the 'time_table' section. At the top, there is a navigation bar with 'AULIM' logo, '스트림' (Stream), '시간표' (Schedule), '공강시간' (Lecture Time) (highlighted), and '계획표 추천' (Recommended Schedule). Below the navigation bar, there is a heading 'time_table'. On the left, there is a sidebar titled 'My Group' with buttons for '그룹 생성' (Create Group), '그룹 삭제' (Delete Group), '그룹 1' (Group 1), '멤버추가' (Add Member), and a button labeled '공강시간 확인' (Check Lecture Time) which is highlighted with a red box. To the right of the sidebar is a table showing a lecture schedule for Tuesday. The table has columns for '교시/요일' (Class/Day), '월' (Monday), and '화' (Tuesday). The schedule shows times from 09:00 to 12:30. The rows for each hour are empty.

교시/요일	월	화
09:00		
09:30		
10:00		
10:30		
11:00		
11:30		
12:00		
12:30		

<< Fig 48. 공강시간 확인 버튼의 모습 >>

3. 해당 그룹의 공강시간이 시간표에 보이는 걸 확인한다.



<< Fig 49. 공강시간이 계산되어 출력된 모습 >>

## 11. Test Plan & Result

### 11.1. Test Scope & Objectives

#### Objectives

- 메인 서버에서 사용자의 개인 시간표 정보와 활동 스트림 정보를 가져와 출력한다.
- 입력한 학습 시간을 바탕으로 적절한 학습 계획표를 추천한다.

#### Scope

- 아주 학사 서비스와 아주 BB 서비스에 접근해야 한다.
- 학습 계획표 추천 알고리즘이 구현되어야 한다.
- 시간표 정보를 데이터베이스에 저장하여 관리해야 한다.

### 11.2. Test Plan

#### Software to be Tested

- Nodemon api를 통해 자동으로 서버를 재시작하여 초기 환경에서 테스트가 가능하도록 한다.

#### Testing Strategy

- Unit testing
  - 서버

Nodemon api를 통해 소스코드의 변경이 있을 때마다 자동으로 로컬 서버를 재시작해 초기 환경부터 테스트를 진행하며 서버 측의 콘솔 터미널로 진행 상황을 파악한다.

- 데이터베이스

상황에 따라 변화되는 데이터를 MySQL의 테이블에 저장 혹은 삭제하여 최신 데이터 상태를 업데이트하고 결과를 조회한다.

- 클라이언트

클라이언트 측의 콘솔 터미널을 활용하여 진행 상황을 파악한다.

### 11.3. Test Procedure

#### Unit Test Cases

- 서버

S_UT_01 : 사용자가 로그인을 시도 했을 때 아이디와 비밀번호 정보를 서버로 전송할 수 있는가.

S_UT_02 : 전달받은 사용자 정보를 통해 아주 학사 및 아주 BB 서비스에 정상적으로 로그인 할 수 있는가.

S_UT_03 : 시간표와 활동 스트림 정보를 JSON 형태로 받아올 수 있는가.

S_UT_04 : 사용자가 입력한 학습 시간을 서버로 전달할 수 있는가.

S_UT_05 : 학습 시간 추천 알고리즘을 통해 계산된 추천 계획표를 클라이언트 측으로 전송할 수 있는가.

S_UT_06 : 그룹원의 시간표 데이터를 종합하여 공강 시간 계산 알고리즘을 통해 계산된 공강 시간표를 클라이언트 측으로 전송할 수 있는가.

- 데이터베이스

D_UT_01 : 크롤링을 통해 가져온 시간표 정보를 테이블에 튜플로 추가하여 저장할 수 있는가.

D_UT_02 : 사용자가 개인활동을 추가했을 때 해당 정보를 테이블에 추가하여 저장할 수 있는가.

D_UT_03 : 사용자가 추천받은 학습 계획을 포함한 시간표 정보를 테이블에 추가하여 저장할 수 있는가.

- 클라이언트

C_UT_01 : 로그인 했을 때 배너의 로그인 입력창이 사라지는가.

C_UT_02 : 시간표를 출력할 수 있는가.

C_UT_03 : 개인 활동 및 학습 시간을 입력 받을 수 있는가.

C_UT_04 : 추천 받은 학습 계획표의 적용 여부를 선택할 수 있는가.

#### 11.4 Integration / System Test Cases

Test Case ID : T_001	
Test Case Name : 사용자 로그인 및 시간표/활동 스트림 가져오기	
For Use case : 시간표/활동 스트림 불러오기	
Related Requirement IDs : SYS_RS_001, SYS_RS_002, USER_RS_001, USER_RS_002	
Precondition : None	
Steps	Expected Results
1. 사용자가 로그인을 시도한다.	로그인에 성공한 화면이 배너에 표시된다.
2. 사용자가 시간표 불러오기 버튼을 누른다.	페이지가 새로고침 되며 가져온 시간표가 메인 페이지에 출력된다.
3. 사용자가 활동 스트림 페이지를 선택한다.	최신 상태의 활동 스트림 정보가 출력된다.

Test Case ID : T_002	
Test Case Name : 개별 활동 추가하기 및 주당 희망 학습 시간 설정.	
For Use case : 개별활동 및 주당 희망학습시간 입력	
Related Requirement IDs : USER_RS_003,USER_RS_004,USER_RS_005,SYS_RS_003	
Precondition : 시간표 불러오기가 완료된 상태여야함.	
Steps	Expected Results
1. 사용자가 개별 활동명과 시간을 입력한다.	화면에 과목명과 희망시간이 보여진다.

2. 활동 추가 버튼을 누른다.	추가된 활동이 포함된 시간표가 출력된다.
3. 사용자가 학습 계획표 페이지로 이동한다.	학습계획표 페이지가 로드되고 보여진다.
4. 사용자가 활동명과 학습 시간을 입력하고 추가 버튼을 누른다.	입력한 아래부분에 추가될 활동명과 시간이 출력된다.
5. 사용자가 저장 버튼을 누른다.	서버에 활동명과 학습시간이 저장된다.

Test Case ID : T_003	
Test Case Name : 학습 계획표 추천 받기	
For Use case : 학습 계획표 추천 받기	
Related Requirement IDs : SYS_RS_005, SYS_RS_006, SYS_RS_007, USER_RS_005, USER_RS_005, USER_RS_006	
Precondition : 학습 활동 입력을 완료한 상태	
Steps	Expected Results
1. 사용자가 출력된 학습 추천 계획표를 확인한다.	화면에 학습 추천 시간표가 출력된다.
2. 사용자가 취소 버튼을 누른다.	화면에 추천 되기 이전의 원본 시간표가 출력된다.
3. 사용자가 다시 학습 활동을 추가한다.	화면에 이전과 다른 학습 추천 시간표가 출력된다.
4. 사용자가 저장 버튼을 누른다.	메인 페이지에 사용자의 시간표에 추천된 학습 활동이 추가된 시간표가 출력된다.

Test Case ID : T_004
----------------------

Test Case Name : 공강시간 확인하기	
For Use case : 공통 공강 시간 추출하기	
Related Requirement IDs : SYS_RS_008, SYS_RS_009, USER_RS_007, USER_RS_008	
Precondition : 그룹이 형성되어 있는 상태	
Steps	Expected Results
1. 사용자가 공강 시간 확인 페이지로 이동한다.	화면 왼쪽에 그룹원 내용과 가운데에 공강 시간이 표시될 빈 시간표가 출력된다.
2. 사용자가 공강 시간 확인하기 버튼을 누른다.	화면에 그룹원들이 공통적으로 비어있는 시간표가 색칠되어 출력된다.

## 11.5 Test Results

Test Case Number	Test Case Description	Test Case Test Result	Comments
1	사용자 로그인 및 시간표/활동 스트림 가져오기	Success	
2	개별 활동 추가하기 및 주당 희망 학습 시간 설정.	Success	
3	학습 계획표 추천 받기	Success	
4	공강시간 확인하기	Success	

Total Tests	4	
Tests Passed	4	100%
Tests Failed	0	
Tests Pending	0	
Test Coverage	4 / 4	100%

Test Quality	4 / 4	100%
--------------	-------	------

## 11.6 Requirements Satisfaction

### 11.6.1 User Requirements

No	USER_RS_001
Title	사용자는 필수적으로 현재 수강 신청된 과목을 확인할 수 있어야 한다.
Verifiable Unit Test	S_UT_01, S_UT_02, D_UT_01, C_UT_02
Verifiable Integrated Test	T_001

No	USER_RS_002
Title	사용자는 필수적으로 아주Bb 활동스트림에 업로드 된 최근 공지사항을 제공받아야 한다.
Verifiable Unit Test	S_UT_01, S_UT_02, S_UT_03
Verifiable Integrated Test	T_001

No	USER_RS_003
Title	사용자는 필수적으로 수강 신청 과목 이외 활동을 추가할 수 있어야 한다.
Verifiable Unit Test	D_UT_02
Verifiable Integrated Test	T_002

No	USER_RS_004
Title	사용자는 필수적으로 수강 신청 과목과 추가된 활동을 시간표 형식으로 제공받아야 한다.

Verifiable Unit Test	S_UT_01, S_UT_02, D_UT_01, D_UT_02 , C_UT_02
Verifiable Integrated Test	T_001, T_002

No	USER_RS_005
Title	사용자가 필수적으로 수강 중이거나 추가 활동에 대해 매주 몇 시간을 공부할 지 설정할 수 있어야 한다.
Verifiable Unit Test	S_UT_04, C_UT_03
Verifiable Integrated Test	T_002

No	USER_RS_006
Title	사용자는 필수적으로 학습 추천 계획표를 확인 및 저장할 수 있어야 한다.
Verifiable Unit Test	S_UT_05, D_UT_03, C_UT_04
Verifiable Integrated Test	T_003

No	USER_RS_007
Title	사용자는 필수적으로 그룹을 만들어 다른 사용자를 초대할 수 있어야 한다.
Verifiable Unit Test	없음
Verifiable Integrated Test	없음

No	USER_RS_008
Title	사용자는 필수적으로 같은 그룹원들과 공통되는 공강시간을 제공받아야 한다.

Verifiable Unit Test	S_UT_06, C_UT_02
Verifiable Integrated Test	T_004

No	USER_RS_009
Title	사용자는 필수적으로 각 과목의 실시간 강의 알림 여부를 설정하고 수업 시간에 맞추어 소리 혹은 푸시 알림을 받을 수 있어야 한다.
Verifiable Unit Test	없음
Verifiable Integrated Test	없음

No	USER_RS_010
Title	사용자는 수업 내용을 과목별, 날짜별로 메모 및 열람할 수 있어야 한다.
Verifiable Unit Test	없음
Verifiable Integrated Test	없음

## 11.6.2 System Requirements

### 11.6.2.1 Functional Requirements

No	SYS_RS_001	Related requirements	USER_RS_001 USER_RS_002
Title	AULIM 로그인은 필수적으로 사용자의 아주Bb 아이디/비밀번호를 이용해야 한다.		

Verifiable Unit Test	S_UT_01, S_UT_02, C_UT_01
Verifiable Integrated Test	T_001

No	SYS_RS_002	Related requirements	USER_RS_001 USER_RS_002
Title	시스템은 필수적으로 사용자가 로그인 할 때, 사용자의 시간표와 활동 스트림을 업데이트해야 한다.		
Verifiable Unit Test	S_UT_02, S_UT_03, D_UT_01, C_UT_02		
Verifiable Integrated Test	T_001		

No	SYS_RS_003	Related requirements	USER_RS_003
Title	시스템은 사용자가 수강 신청 과목 이외 활동을 추가 및 저장할 수 있도록 해야한다.		
Verifiable Unit Test	S_UT_04, D_UT_02, C_UT_03		
Verifiable Integrated Test	T_002		

No	SYS_RS_004	Related requirements	USER_RS_004
Title	시스템은 필수적으로 사용자의 수강 신청 과목, 추가된 활동을 사용자가 알아볼 수 있는 시간표의 형태로 한 화면에 출력해야 한다.		
Verifiable Unit Test	S_UT_04, S_UT_05, D_UT_02, D_UT_03, C_UT_02, C_UT_03		
Verifiable Integrated Test	T_002		

No	SYS_RS_005	Related requirements	USER_RS_005
Title	시스템은 필수적으로 사용자가 과목 및 활동별로 필요한 희망 학습 시간을 기록 및 저장할 수 있도록 해야 한다.		
Verifiable Unit Test	S_UT_03, S_UT_04, D_UT_03, C_UT_03		
Verifiable Integrated Test	T_002, T_003		

No	SYS_RS_006	Related requirements	USER_RS_005
Title	시스템은 필수적으로 사용자가 입력한 학습 시간과 시간표를 기반으로 “학습 추천 계획표”를 제공해야 한다.		
Verifiable Unit Test	S_UT_04, S_UT_05, D_UT_03		
Verifiable Integrated Test	T_003		

No	SYS_RS_007	Related requirements	USER_RS_006
Title	시스템은 필수적으로 사용자가 추천 학습 계획표의 사용 여부를 선택할 수 있고, 새로운 계획표를 추천 받을 수 있어야 한다.		
Verifiable Unit Test	S_UT_05, D_UT_03, C_UT_04		
Verifiable Integrated Test	T_003		

No	SYS_RS_008	Related requirements	USER_RS_007
Title	시스템은 필수적으로 사용자가 그룹을 형성하여 다른 사용자를 초대할 수 있도록 해야 한다.		
Verifiable Unit Test	없음		
Verifiable Integrated Test	없음		

No	SYS_RS_009	Related requirements	USER_RS_008
Title	시스템은 필수적으로 사용자가 가입되어 있는 그룹의 그룹원과 공통된 공강시간을 제공해야 한다.		
Verifiable Unit Test	S_UT_06, C_UT_02		
Verifiable Integrated Test	T_004		

No	SYS_RS_010	Related requirements	USER_RS_009
Title	시스템은 필수적으로 사용자가 수강 과목의 실시간 알람 여부를 Yes/No 형태로 설정할 수 있도록 해야 한다.		
Verifiable Unit Test	없음		
Verifiable Integrated Test	없음		

No	SYS_RS_011	Related requirements	USER_RS_009
Title	실시간 알림을 “Yes”로 선택했다면 수업 시작 10분전 시스템은 사용자에게 알람을 보내주어야 한다.		
Verifiable Unit Test	없음		
Verifiable Integrated Test	없음		

No	SYS_RS_012	Related requirements	USER_RS_010
Title	강의가 시작될 때 자동적으로 해당 과목, 날짜에 대한 메모입력칸이 생성되어야 한다.		

Verifiable Unit Test	없음
Verifiable Integrated Test	없음

No	SYS_RS_013	Related requirements	USER_RS_010
Title	시스템은 사용자가 수업을 듣고 있을 때, 그 과목에 대한 메모를 작성 및 저장할 수 있어야 한다.		
Verifiable Unit Test	없음		
Verifiable Integrated Test	없음		

No	SYS_RS_014	Related requirements	USER_RS_010
Title	시스템은 사용자가 이전에 작성했던 메모를 불러올 수 있어야 한다.		
Verifiable Unit Test	없음		
Verifiable Integrated Test	없음		

#### 11.6.2.2 Non Functional Requirements

No.	NF_RS_001	Related Requirement	USER_RS_001~005 SYS_RS_001~009
Title	시스템 이동 시 Time Delay가 1초 이내로 한다.		
Verifiable Unit Test	S_UT_01, S_UT_02, S_UT_04, S_UT_05, S_UT_06		
Verifiable Integrated Test	T_001, T_002, T_003, T_004		

No.	NF_RS_002	Related Requirement	USER_RS_001 SYS_RS_002
Title	사용자가 시간표 및 활동스트림 불러오기를 요청할 때 30초 이내로 완료한다.		
Verifiable Unit Test	S_UT_03, D_UT_01		
Verifiable Integrated Test	T_001		

No.	NF_RS_003	Related Requirement	USER_RS_010 SYS_RS_012 ~ 014
Title	한 과목 당 저장할 수 있는 메모의 용량은 최소 350KB 이상 이어야 한다.		
Verifiable Unit Test	없음		
Verifiable Integrated Test	없음		

No.	NF_RS_004	Related Requirement	USER_RS_001
Title	다른 사용자에게 자신의 시간표가 보이지 않아야 한다.		
Verifiable Unit Test	없음		

Verifiable Integrated Test	없음
----------------------------------	----

No.	NF_RS_005	Related Requirement	USER_RS_001~004
Title	DB에 저장되는 data들을 1시간마다 백업한다.		
Verifiable Unit Test	D_UT_01, D_UT_02, D_UT_03		
Verifiable Integrated Test	T_001, T_002, T_003, T_004		

## Appendix

### A. Glossary

Web Crawling	프로그램을 이용해 조직적, 자동화된 방법으로 웹을 탐색하는 기술.
상단바	웹 페이지 상단에 위치한 배너. ex)
 <i>&lt;&lt;Fig 50. AULIM 배너의 예시&gt;&gt;</i>	
학습계획표	학습 시간이 적절히 분배된 상태의 시간표
활동 스트림	- 블랙보드의 기능 중 하나. - 공지사항 추가, 강의자료 업로드, 과제제출 기한 등 확인이 가능.
쿼리(Query)	웹서버에서 특정한 정보를 데이터베이스에 요청하는 언어
Local Time	클라이언트 환경에 설정되어 있는 시간
Server Time	사용하고 있는 Web Server에서 사용하고 있는 시간

### B. Domain/Design class Description

Activity		
Type	Data	Description
char	name	활동의 이름
int	classdate	활동이 있는 요일
int	start	활동을 시작하는 시간
int	long	활동 시간의 길이
char	user	활동하는 사람의 이름
Type	Operation	Description
void	printTable()	user의 활동들을 시간표에 출력한다.
void	addAct(Act)	시간표에 활동정보를 추가한다.

Memo		
Type	Data	Description
String	userId	사용자를 식별할 수 있는 ID이다.
Date	writeDate	메모를 작성한 날짜에 대한 항목이다.
Date	modifyDate	수정된 메모 날짜에 대한 항목이다.
String	fileName	과목명과 날짜가 들어간 항목이다.

User		
Type	Data	Description
String	userid	크롤링에 사용되는 유저의 ID이다.
String	userpw	유저의 PW이다.

Stream		
Type	Data	Description
String	date	활동스트림의 날짜이다.
String	time	활동스트림의 시간이다.
String	name	활동스트림 과목의 이름이다.
String	content	활동스트림 내용이다.

## C. Project Status

### C.1 Planned vs. Actual

구분	Done	Doing	Progress
UI Design	<ul style="list-style-type: none"> <li>- 기본 HTML 레이아웃 및 CSS 구현</li> <li>- 각 페이지에 대한 세부 구성요소 구현</li> <li>- CSS를 활용한 시각효과 구현</li> <li>- 반응형 Web 구현</li> </ul>		100%
Server 구성	<ul style="list-style-type: none"> <li>- 배너에 포함된 각 기능 페이지에 대한 라우팅</li> <li>- 사용자의 요청에 대한 메인 기능 함수 구현</li> </ul>	<ul style="list-style-type: none"> <li>- 실시간 알림 기능을 위한 시간 업데이트</li> </ul>	70%
Crawling 기능	<ul style="list-style-type: none"> <li>- 학사서비스에서 시간표 정보 Crawling</li> <li>- BlackBoard 활동스트림 Crawling</li> </ul>		100%
시간표 관리	<ul style="list-style-type: none"> <li>- (시간이 정해진) 수강신청 이외 개별 활동 추가</li> <li>- (임의로 배정될) 주당 학습 시간 추가</li> </ul>		100%
학습 시간 추천	<ul style="list-style-type: none"> <li>- 학습 계획 추천 알고리즘 구현</li> <li>- 입력한 주당 학습시간에 맞는 추천 시간표 출력</li> </ul>		100%
공강 시간 추출	<ul style="list-style-type: none"> <li>- (그룹원들의 시간표를 얻었다는 전제) 추출 알고리즘 구현</li> </ul>	<ul style="list-style-type: none"> <li>- 그룹 형성하기</li> <li>- 추출된 공강 시간을 시간표 형태로 저장 하기</li> </ul>	30%
Memo 기능	<ul style="list-style-type: none"> <li>- 메모 저장 method 구현</li> <li>- 수강과목 목록 출력 구현</li> </ul>	<ul style="list-style-type: none"> <li>- 삭제, 추가 method 구현</li> <li>- 실시간 수업 시 메모 창띄우기 구현</li> </ul>	30%
알람 기능		<ul style="list-style-type: none"> <li>- 시간표에 대한 알림 함수 구현</li> </ul>	0%
Database	<ul style="list-style-type: none"> <li>- Node.js와 Mysql 연동</li> <li>- USER TABLE, MEMO TABLE 작성</li> <li>- 시간표 정보 전송 및 불러오기 구현</li> <li>- Trigger 작성</li> </ul>	<ul style="list-style-type: none"> <li>- 그룹원들의 시간표 불러오기</li> <li>- 시간표 정보 삭제하기</li> </ul>	75%
전체			70%

## C.2 Team Members' Contribution

Name	Contributed	Ratio
김진수	1.2 Propoosed Idea and Solution 4.2 Non-Functional Requirements 5.2 Use Case 2 5.3 Use Case 3 6.1 Behavioral Analysis 8.1 Design Modeling 11.2 Test Plan 11.6 Requirements Satisfaction	20%
신승현	2. Scope and Objective 4.1 Functional Requirements 5.4 Use Case 4 6.1 Structural Analysis 8.1 Design Modeling 11.1 Test Scope & Objectives 11.2 Test Plan 11.5 Test Results 11.6 Requirements Satisfaction	20%
양준길	3. User Requirements 4.1 Functional Requirements 5.5 Use Case 5 6.1 Structural Analysis 8.1 Design Modeling 10. UI Design 11.3 Test Procedures 11.4 Integration / System Test Case	20%
이재협	1.1. Problem Statement and Motivation 4.2 Non-Functional Requirements 5.6 Use Case 6 7. Architectural Design 9.1 Implementation Specific 11.2 Test Plan 11.4 Integration / System Test Case 11.6 Requirements Satisfaction	20%
이창엽	3. User Requirements 5.1 Use Case 1 6.1 Behavioral Analysis 8.1 Design Modeling 9.2 Message Format 9.3 System/Data Dependencies & Requirements 11.3 Test Procedures 11.4 Integration / System Test Case 11.6 Requirements Satisfaction	20%

## D. Change History

Part	Reason	Date	Author
1. Project Overview	Original		
2. Scope and Objective	Original		
3. User Requirement	- User Requirement 중요도 순서대로 번호 변경	2021.05.09	이재협
4. System Requirements	- System Requirement 14번 추가	2021.05.09	이재협
5. Requirements Specification	- Use Case 순서 변경 및 번호 설정	2021.05.31	신승현
6. Requirements Analysis	- SYS_RS_002 Analysis 수정	2021.05.10	김진수
7. Architectural Design	- Layer 간의 상관 관계 표시	2021.05.29	이창엽
8. Design Modeling	- UseCase2 Sequence Diagram 수정	2021.06.01	김진수
9. Implementation	- 9.1.4 Crawler 설명 추가	2021.06.04	양준길
10. UI Design	Original		
11. Test Plan & Result	- USER_RS_003 Verifiable Unit Test 수정	2021.06.06	신승현

## E. References

인하대git 프로젝트 웹강요정[웹사이트].(2021.04.08).

URL:<https://github.com/challenger71498/OnlineLectureFairy>

아주대학교BlackBoard[웹사이트].(2021.04.08).URL:<https://eclass2.ajou.ac.kr/ultra/course>

에브리타임[웹사이트]. (2021.04.08). URL: <https://everytime.kr/>

Selenium with Python[웹사이트]. (2021.04.09).

URL: <https://selenium-python.readthedocs.io/>

웹 프로그래밍 튜토리얼[웹사이트]. (2021.03.31)

URL: <https://poiemaweb.com/>

동적 자바스크립트 구성[웹사이트]. (2021.04.03)

URL: <https://nahosung.tistory.com/28>

유튜브(생활코딩)[웹사이트].(2021.03.30).

URL:<https://www.youtube.com/channel/UCvc8kv-i5fvFTJBFAk6n1SA>

draw.io[웹사이트].(2021.06.07).

URL:<https://diagrams.net>

## F. List of Figures

- <<Fig 1. Sub-Usecase 로그인 activity diagram >>
- <<Fig 2. Usecase 1 - 시간표/활동스트림 불러오기 activity diagram>>
- <<Fig 3. Usecase 2 - 개별활동 추가 activity diagram>>
- <<Fig 4. Usecase 3 - 학습 계획표 추천 activity diagram>>
- <<Fig 5. Usecase 4 - 그룹 형성/공강시간 추출 activity diagram>>
- <<Fig 6. Usecase 5 - 실시간 수업 알림받기 activity diagram>>
- <<Fig 7. Usecase 6 - 메모 작성하기 activity diagram>>
- <<Fig 8. Usecase 1 - 시간표/활동 스트림 불러오기 class diagram>>
- <<Fig 9. Usecase 2 - 개별활동 추가 class diagram>>
- <<Fig 10. Usecase 3 - 학습계획표 추천 class diagram>>
- <<Fig 11. Usecase 4 - 그룹 형성/공강시간 추출 class diagram>>
- <<Fig 12. Usecase 5 - 실시간 수업 알림 받기 class diagram>>
- <<Fig 13. Usecase 6 - 메모 작성하기 class diagram>>
- <<Fig 14. Logical view / Architecture>>
- <<Fig 15. Usecase 01 - 시간표 연동 sequence diagram>>
- <<Fig 16. Usecase 01 - 활동스트림 연동 sequence diagram>>
- <<Fig 17. Usecase 01 - 시간표/활동스트림 연동 design class diagram>>
- <<Fig 18. Usecase 02 - 개별활동 추가 sequence diagram >>
- <<Fig 19. Usecase 02 - 주당 학습시간 설정 sequence diagram >>
- <<Fig 20. Usecase 02 - 주당 학습시간 설정 design class diagram >>
- <<Fig 21. Usecase 03 - 학습계획표 추천 sequence diagram >>
- <<Fig 22. Usecase 03 - 학습계획표 추천 design class diagram >>
- <<Fig 23. Usecase 04 - 그룹 공강시간 조회 sequence diagram >>
- <<Fig 24. Usecase 04 - 그룹 공강시간 조회 design class diagram >>
- <<Fig 25. Usecase 05 - 수업시작 알림설정 sequence diagram >>
- <<Fig 26. Usecase 05 - 수업시작 알림설정 design class diagram >>
- <<Fig 27. Usecase 06 - 메모 작성 sequence diagram >>
- <<Fig 28. Usecase 06 - 메모 작성 design class diagram >>
- <<Fig 29. combined design class diagram >>
- <<Fig 30. System overview>>
- <<Fig 31. Web/AULIM Server Implementation>>
- <<Fig 32. AULIM Server/MySQL Implementation>>
- <<Fig 33. Table implementation>>
- <<Fig 34. Crawler implementation>>
- <<Fig 35. Message format implementation example>>
- <<Fig 36. ID/PW 입력>>
- <<Fig 37. 로그인 후 배너 모습>>
- <<Fig 38. 스트림 페이지 이동 모습>>

<<Fig 39. 시간표 불러오기 버튼을 누른 후의 모습>>

<<Fig 40. 시간표 메뉴의 모습>>

<<Fig 41. 개별 활동을 입력한 모습>>

<<Fig 42. 개별 활동 추가 버튼을 누른 뒤의 모습>>

<<Fig 43. 학습 시간을 입력한 모습>>

<<Fig 44. 입력한 학습시간을 추가한 모습>>

<<Fig 45. 추천받기 버튼을 눌러 추천 학습 시간표가 출력된 모습>>

<<Fig 46. 취소 버튼을 눌러 이전 시간표로 되돌아간 모습>>

<<Fig 47. 공강시간 메뉴의 모습>>

<<Fig 48. 공강시간 확인 버튼의 모습>>

<<Fig 49. 공강시간이 계산되어 출력된 모습>>

<<Fig 50. AULIM 배너의 예시>>