

# Requirements Specification

동아리 스케줄 공유를 통한 공동공간 예약 및 조회 시스템 WePlan

7 조 박종범, 정유환, 김동령, 이승주

# 목차

<b>Tables &amp; Figures</b> .....	<b>2</b>
<i>Tables</i> .....	2
<i>Figures</i> .....	3
<b>1. Project Overview</b> .....	<b>4</b>
1.1. <i>Problem definition and importance</i> .....	4
1.2. <i>Solution</i> .....	4
<b>2. Scope and Objective</b> .....	<b>4</b>
2.1. <i>Scope</i> .....	4
2.1.1. <i>In The Scope</i> .....	4
2.1.2. <i>Out of Scope</i> .....	4
2.2. <i>Objective</i> .....	4
<b>3. User Requirements</b> .....	<b>5</b>
<b>4. System Requirements</b> .....	<b>8</b>
<b>5. Domain Requirements</b> .....	<b>12</b>
<b>6. Supplementary Requirements</b> .....	<b>12</b>
6.1. <i>Interface requirements</i> .....	12
6.2. <i>Physical requirements</i> .....	12
6.3. <i>Design requirements</i> .....	12
6.4. <i>Implementation requirements</i> .....	12
6.5. <i>Additional NFRs</i> .....	12
<b>7. Requirements Analysis</b> .....	<b>13</b>
7.1. <i>Use Case: 로그인</i> .....	13
7.2. <i>Use Case: 유저 등록</i> .....	14
7.3. <i>Use Case: 동아리 채널 생성</i> .....	15
7.4. <i>Use Case: 스케줄 예약 요청</i> .....	17
7.5. <i>Use Case: 스케줄 상세정보 조회 및 삭제</i> .....	18
7.6. <i>Use Case: 관리자의 스케줄 승인/거절</i> .....	19
<b>8. Validation Criteria (Metrics and Measures)</b> .....	<b>21</b>
<b>9. Appendix: Project Status</b> .....	<b>22</b>

9.1. 미팅.....	22
9.2. Commit Logs.....	22
9.2.1. Server ( <a href="https://github.com/software7/weplan-server.git">https://github.com/software7/weplan-server.git</a> ).....	22
9.2.2. Client ( <a href="https://github.com/software7/weplan.git">https://github.com/software7/weplan.git</a> ).....	23

## Tables & Figures

### Tables

표 3-1.....	5
표 3-2.....	5
표 3-3.....	5
표 3-4.....	5
표 3-5.....	5
표 3-6.....	5
표 3-7.....	6
표 3-8.....	6
표 3-9.....	6
표 3-10.....	6
표 3-11.....	6
표 3-12.....	7
표 3-13.....	7
표 3-14.....	7
표 3-15.....	7
표 3-16.....	7
표 3-17.....	7
표 4-1.....	8
표 4-2.....	8
표 4-3.....	8
표 4-4.....	8
표 4-5.....	8
표 4-6.....	9
표 4-7.....	9
표 4-8.....	9
표 4-9.....	9
표 4-10.....	9
표 4-11.....	10
표 4-12.....	10
표 4-13.....	10
표 4-14.....	10
표 4-15.....	11
표 4-16.....	11
표 4-17.....	11
표 4-18.....	11
표 4-19.....	11

표 4-20.....	12
표 7-1.....	13
표 7-2.....	14
표 7-3.....	15
표 7-4.....	17
표 7-5.....	18
표 7-6.....	19
표 8-1.....	21
표 8-2.....	21
표 8-3.....	22
표 9-1.....	22

Figures

그림 7-1.....	13
그림 7-2.....	14
그림 7-3.....	15
그림 7-4.....	17
그림 7-5.....	18
그림 7-6.....	19
그림 7-7.....	21

# 1. Project Overview

## 1.1. Problem definition and importance

대학교에는 다양한 동아리가 있고, 각 동아리는 여러 활동을 진행한다. 동아리 활동의 원활한 진행을 위해 동아리 방과 같은 동아리 공용 공간의 이용 시간을 서로 조율하여 이용하는 것이 매우 중요하다.

현재 동아리 임원진들은 통상적으로 카카오톡과 같은 메신저의 채팅방을 통해 동아리 일정을 고지하고 있고, 이에 따라 회원들은 메신저를 통해 동아리 방 이용을 문의한다. 외부 메신저 앱에서 이용하는 동아리 공지 채팅방은 일정 공지를 편하게 확인할 수 있는 UI로 제공되지 않을 뿐만 아니라, 앱 내에서 사용자의 다른 다양한 채팅방과 똑같은 정보적 위계를 가진다. 이는 구조 상 사용자에게 중요한 공지가 제대로 확인되지도 못한 채 다른 메세지들에 의해 밀리기 쉽다.

이러한 현 상황을 개선하기 위해, 동아리 회장과 회원 간에 동아리 일정 관리 및 동아리 공간 이용 신청을 효율적으로 처리할 수 있는 독립적인 서비스가 필요하다. 이 서비스를 통해 교내 학우들이 동아리 활동의 본질에 집중할 수 있도록 하고, 동아리 문화의 활성화에 보탬이 되고자 한다.

## 1.2. Solution

앞서 제시한 문제들에 대해 서비스에서 핵심적으로 제공할 해결 방안을 다음과 같이 수립하였다.

- i. 회장-회원 간 동아리 공간 이용 신청 및 허가 과정이 간편하고 빠르게 일어나야 한다
- ii. 동아리원이 동아리 공간 예약 현황을 한 번에 조회할 수 있도록 하여, 개인의 동아리 활동에 참고가 될 수 있도록 해야 한다.
- iii. 이 모든 과정이 높은 접근성으로 보장되어야 한다.

이에 따라 대학교 각 동아리가 동아리 활동 일정 공유 및 조회, 동아리 방 이용 예약 신청 및 허가를 원활히 진행할 수 있는 모바일 앱 서비스 'WePlan'을 제시한다.

# 2. Scope and Objective

## 2.1. Scope

### 2.1.1. In The Scope

동아리 회장과 일반 동아리원의 구분을 위해 관리자 계정과 일반 회원 계정을 구분한다. 기본적으로 모든 사용자는 동아리의 이용 타임 테이블을 조회할 수 있으며, 일반 회원은 이를 바탕으로 빈 타임 슬롯에 이용 예약을 신청하거나, 이미 신청이 완료된 예약을 취소할 수 있도록 한다. 관리자는 예약 신청 현황을 따로 조회할 수 있으며, 각 예약 신청을 승인하거나 거절할 수 있다.

### 2.1.2. Out of Scope

제한된 개발 인력과 기간 상, 다음 항목들은 이번 서비스 구현에서 제외하도록 한다 - 기존 학사 시스템과의 연동, 채팅 및 게시판 등의 커뮤니티 기능, 각 사용자의 채널 가입 과정, 복수의 동아리 채널 가입.

## 2.2. Objective

동아리원은 모바일 앱을 통해 간편하고 빠르게 동아리 스케줄을 조회 및 신청할 수 있도록 하고, 동아리 회장은 동아리 스케줄 현황을 쉽게 관리할 수 있도록 한다. 즉, 이러한 일련의 과정들을 실시간으로 빠르게 출력되도록 하고, 해당 기능들을 모바일 앱의 높은 접근성을 기반으로 제공함으로써 전반적인 동아리 일정 공유 프로세스를 효율적으로 재구성하는 것을 목표로 한다.

### 3. User Requirements

표 3-1

No.	USER_RQ_001
Title	시스템을 이용하는 유저는 관리자와 게스트로 구분되어야 한다.
Description	본 동아리 채널 시스템은 학생들이 주도적으로 동아리 공간을 예약 및 사용할 수 있도록 하는 것을 목적으로 한다. 동아리 구성원은 크게 동아리 활동을 총괄하는 동아리장과 그 외 동아리원으로 나뉘어지므로, 동아리 채널 시스템 역시 이에 대응하여 각 동아리의 채널을 관리하는 관리자와 일반 회원인 게스트로 나뉘어야 한다.

표 3-2

No.	USER_RQ_001_01
Title	관리자 유저는 게스트 유저의 모든 기능 및 동작을 제공하여야 한다.
Description	게스트 유저와 같이 관리자 유저 역시 같은 동아리 공간을 공유하고 사용하는 학생이다. 만약 관리자 유저에게 게스트 유저와 동일한 기능에 접근할 수 있는 기능을 부여하지 않는다면, 관리자 유저는 게스트용 계정을 따로 만들어 두 개의 계정을 관리해야 할 수도 있으며, 유저는 이런 불편함을 겪으며 시스템을 사용하려고 하지 않을 것이다.

표 3-3

No.	USER_RQ_001_02
Title	관리자 유저는 게스트 유저는 접근할 수 없는 채널 관리 페이지에 대한 접근 권한을 갖는다.
Description	다른 사용자의 스케줄 예약을 승인 및 거절할 수 있는 권한을 불특정 다수의 동아리 게스트 유저에게 부여한다면 예약의 무분별한 승인 및 무단 취소 등 전체 시스템에 혼란을 가져올 가능성이 크다. 따라서 이러한 관리 권한은 한 명의 동아리 관리자 유저에게만 부여되어야 한다.

표 3-4

No.	USER_RQ_002
Title	시스템은 서비스를 이용하는 모든 유저에 대하여 유효한 인증 절차를 마련하여야 한다.
Description	본 동아리 채널 시스템은 보다 빠르고 편리한 동아리 공간 이용을 위하여 대학생들을 대상으로 하는 동아리 예약 및 관리 시스템이다. 동아리의 예약 관리는 오롯이 동아리 구성원들만의 몫이며, 본 시스템에 접근 가능한 유저는 대학교 학생 구성원 및 해당 동아리의 구성원이어야 한다. 그러므로 해당 요구사항이 충족되지 않는다면 중대한 문제가 발생할 것이다. 따라서 시스템은 인가된 (시스템에 등록된) 유저만이 시스템을 이용할 수 있도록 해야 한다.

표 3-5

No.	USER_RQ_003
Title	유저는 자동 로그인을 통해 간편하고 빠르게 시스템에 접속할 수 있어야 한다.
Description	동아리의 일정은 실시간으로 갱신되므로, 유저는 이를 시스템에 접근하여 주기적으로 확인하게 된다. 만약 시스템 접근 시마다 반복적으로 인증 절차를 거치게 된다면, 이는 사용자 편의성을 저해할 것이다. 해당 시스템의 유저의 사용 주기와 접근성을 고려하였을 때 시스템에 한 차례 인증을 거친 유저는 자동 로그인 시스템으로 로그인을 유지할 수 있도록 해야 한다.

표 3-6

No.	USER_RQ_004
-----	-------------

Title	유저의 인증 정보가 탈취되더라도 탈취자가 해당 정보로 인증할 수 없도록 해야 한다.
Description	악의적인 사용자가 다른 사용자의 정보를 탈취해 시스템의 인증 절차를 거쳐 시스템에 접근하게 된다면, 시스템은 중대한 위협에 직면할 수 있다. 따라서 시스템은 유저 정보가 탈취되더라도 실제 데이터를 확인할 수 없도록 정보를 암호화해 저장하고 있어야 한다.

**표 3-7**

No.	USER_RQ_005
Title	관리자 유저는 동아리 채널을 생성할 수 있어야 한다.
Description	스케줄은 유저의 예약 일정을 지칭하는 연속적인 시간의 단위이며, 채널은 동일한 공간 내에서 공유되는 스케줄의 집합을 공유하는 가상 공간이다. 시스템은 각 동아리마다 개별적으로 시간을 공유하고 관리할 수 있도록 관리자 유저에게 개별 채널을 생성하는 기능을 제공해 채널을 최초 생성하고, 게스트 유저들이 이에 접근할 수 있도록 하여야 한다.

**표 3-8**

No.	USER_RQ_006
Title	유저는 채널의 스케줄 예약 현황을 타임 테이블 형태로 조회할 수 있어야 한다.
Description	유저의 예약이 등록되면 이를 시각적으로 보여주는 수단이 필요하다. 따라서 시스템은 유저가 현재의 예약 현황들을 파악할 수 있도록 타임 테이블 형식을 사용하여 예약 정보를 제공하여야 한다.

**표 3-9**

No.	USER_RQ_006_01
Title	유저는 타임 테이블 상에서 각 스케줄을 쉽게 구분하고 이해할 수 있어야 한다.
Description	만약 각각의 스케줄 블록 간의 경계가 모호하거나, 각 스케줄 블록이 시각적으로 동일하거나, 스케줄 블록이 서로 유사한 색상으로 제공되는 경우, 유저는 예약을 확인하는 데 큰 어려움을 겪을 것이다. 이는 사용자 경험을 저해하는 요소로 작용함과 동시에, 보다 편리한 동아리 스케줄 예약 플랫폼을 제공하고자 하는 본 프로젝트의 취지와 반하는 사항이다. 따라서, 시스템은 타임 테이블 상의 스케줄들이 서로 구분이 가능하도록 보색 관계 등의 디자인을 이용하여 뚜렷한 구분 경계를 가지도록 하여야 한다.

**표 3-10**

No.	USER_RQ_007
Title	유저는 타임 테이블에서 각 스케줄의 상세한 정보를 추가적으로 확인할 수 있어야 한다.
Description	예약에 관한 상세한 정보를 제공하지 않는다면 유저는 누가, 어떤 목적으로 해당 시간을 예약한 것인지 파악할 수 없다. 예약에 관한 상세한 정보는 동아리 구성원 누구나가 파악할 수 있도록 상세한 정보를 제공하여야 한다.

**표 3-11**

No.	USER_RQ_008
Title	유저는 타임 테이블에서 시스템 접속 시점부터 한 달까지의 예약 현황을 조회할 수 있어야 한다.
Description	유저에게 단기간의 예약 현황만을 제공하는 것은 유저가 장기적인 계획을 수립하는 것을 어렵게 만든다. 유저가 충분한 정보를 기반으로 일정을 계획할 수 있도록 접속 시점부터 한 달까지의

	예약 현황들을 타임 테이블에서 확인할 수 있도록 하여야 한다.
--	------------------------------------

**표 3-12**

No.	USER_RQ_009
Title	유저는 채널에서 이용 가능한 시간 중 원하는 시간을 선택하여 스케줄을 예약할 수 있어야 한다.
Description	유저는 자신이 속한 채널에서 모바일 앱을 통해 시공간의 제약 없이, 이용 가능한 시간 중 본인이 원하는 시간을 자유롭게 선택하여 스케줄을 예약할 수 있어야 한다.

**표 3-13**

No.	USER_RQ_009_01
Title	유저가 이미 존재하는 스케줄 블록과 시간이 겹치는 스케줄 예약을 신청할 수 없어야 한다.
Description	이전에 이미 다른 게스트가 스케줄을 예약하여 승인 및 반영된 시간에 대해서는 예약이 불가능하게 하여 충돌을 방지해야 한다.

**표 3-14**

No.	USER_RQ_010
Title	관리자는 관리하고 있는 채널의 예약 현황을 별도로 확인할 수 있어야 한다.
Description	관리자는 별도로 게스트의 예약내용을 확인하고, 예약의 타당성 및 중요도 등을 예약 상세정보 조회를 통해 판단할 수 있어야 한다.

**표 3-15**

No.	USER_RQ_011
Title	관리자는 게스트의 예약을 승인 혹은 거절할 수 있어야 한다.
Description	시스템은 관리자가 예약을 한 눈에 조회하고, 이를 승인 및 거절하는데 필요한 모든 기능들을 제공해주어야 한다.

**표 3-16**

No.	USER_RQ_012
Title	게스트의 예약 신청이 승인 혹은 거절될 경우, 게스트는 이를 앱을 통해서 알림을 받아야 한다.
Description	예약의 승인 여부를 유저에게 직접적으로 알리지 않는다면 유저는 이를 확인하기 위해 주기적으로 시스템에 접속해야 하는 불편을 감수해야 한다. 사용자 경험 증진을 위해 시스템에서 이를 자동으로 게스트 유저에게 알려주어야 한다.

**표 3-17**

No.	USER_RQ_013
Title	유저는 타임 테이블의 변경 사항을 실시간으로 확인할 수 있어야 한다.
Description	모든 유저의 입장에서, 다수의 유저가 동시에 접근하는 타임 테이블은 스케줄 예약에 대한 관리자 승인, 게스트 유저의 스케줄 취소 등으로 인해 임의의 시간에 자주 변경될 것이다. 이러한 변경 사항들이 최대한 빠르게 타임 테이블에 실시간으로 업데이트되어야 모두가 똑같은 타임 테이블을 조회할 수 있게 되며, 예약 신청에 있어서의 충돌을 사전에 1차적으로 방지할 수 있다. 또한, 관리자 유저도 새로운 스케줄 예약을 이미 승인된 스케줄과의 상호 관계를 판단하여 승인 여부를 결정해야 하기 때문에, 해당 요구사항은 전반적인 서비스 품질을 향상시키는 데 있어 중요한 요소이다.

## 4. System Requirements

### 4.1 Functional Requirements

표 4-1

No.	SYS_RQ_FR_001	Related Requirement	USER_RQ_001 USER_RQ_002
Title	시스템은 회원가입 시 관리자 권한을 인증하기 위해 관리자 인증번호를 요청하고 이를 확인해야 한다.		
Description	사용자가 회원 가입을 진행할 때, 관리자 권한을 인증하기 위해 시스템은 관리자 인증번호를 요청하고 이를 확인해야 한다.		

표 4-2

No.	SYS_RQ_FR_002	Related Requirement	USER_RQ_002
Title	유저는 아이디, 비밀번호를 통해 로그인을 진행하며, 시스템은 성공적으로 인증이 되어야 로그인이 되도록 한다.		
Description	시스템이 관리하는 아이디가 중복되지 않도록 체크하며 비밀번호 유효성 검사를 진행하여 보안을 높일 수 있도록 한다		

표 4-3

No.	SYS_RQ_FR_003	Related Requirement	USER_RQ_003
Title	토큰 기반으로 이전에 수행된 로그인에 대해서는 최소 1달 간 로그인 상태를 유지한다.		
Description	한번 로그인한 유저에게 추가적으로 로그인을 요청하지 않도록 시스템은 사용자의 로그인을 유지할 있도록 한다.		

표 4-4

No.	SYS_RQ_FR_004	Related Requirement	USER_RQ_004
Title	비밀번호는 보안을 위해 해싱을 하여 저장한다.		
Description	사용자의 비밀번호는 보안을 최우선으로 고려하여, 해싱 알고리즘을 통해 변환된 형태로 저장된다. 이 방식은 원본 비밀번호를 직접적으로 알아낼 수 없게 하여 사용자의 정보를 보호한다		

표 4-5

No.	SYS_RQ_FR_005	Related Requirement	USER_RQ_005
Title	시스템은 채널을 생성할 수 있는 UI를 관리자 유저에게만 제공하도록 한다.		
Description	시스템은 관리자에게만 채널 생성 권한을 부여하여, 해당 권한을 가진 사용자만이 새로운 채널을 생성할 수 있는 사용자 인터페이스를 제공한다. 이를 통해 채널의 관리 및 통제를 강화할		

	수 있다.
--	-------

**표 4-6**

No.	SYS_RQ_FR_006	Related Requirement	USER_RQ_006
Title	시스템은 모든 종류의 유저에게 채널의 스케줄 예약 현황을 타임 테이블 형태로 제공하여야 한다.		
Description	시스템은 사용자가 현재 채널의 스케줄 현황에 관한 정보들을 확인할 수 있는 타임 테이블을 제공하여, 각 스케줄의 일시 및 간단한 대표 정보를 타임 테이블 안에 블록 형태로 나타나도록 한다.		

**표 4-7**

No.	SYS_RQ_FR_007	Related Requirement	USER_RQ_007
Title	시스템은 채널의 타임 테이블에서, 각 스케줄 블록의 스케줄 상세 정보를 확인할 수 있도록 하는 기능을 별도로 제공한다.		
Description	타임 테이블에서, 각 스케줄 예약에 대해 더욱 상세한 정보를 쉽게 확인할 수 있도록 하는 별도의 기능을 시스템이 제공해야 함을 명시한다. 이 기능은 예약된 시간, 예약자, 예약 목적 등 게스트 유저가 스케줄 예약 신청 시 기입한 정보들을 포함해야 한다.		

**표 4-8**

No.	SYS_RQ_FR_008	Related Requirement	USER_RQ_008
Title	채널의 타임 테이블은 접속 시점부터 최소 한달까지의 스케줄 정보를 출력해야 한다.		
Description	시스템은 사용자에게 한 동아리 채널의 최대 한달 간의 스케줄 정보를 제공하는 기능을 갖추고 있다. 이를 통해 사용자들은 중/장기적인 스케줄을 예약하고 관리할 수 있다.		

**표 4-9**

No.	SYS_RQ_FR_009	Related Requirement	USER_RQ_009
Title	사용자가 스케줄에 대한 예약을 신청할 수 있는 별도의 UI를 제공하여야 한다.		
Description	시스템은 사용자가 스케줄에 대한 예약을 신청할 수 있는 UI를 제공한다. 이를 통해 사용자는 쉽고 편리하게, 예약 가능한 시간 중 원하는 시간을 선택하여 신청할 수 있다.		

**표 4-10**

No.	SYS_RQ_FR_009_01	Related Requirement	USER_RQ_009_01
Title	시스템은 이미 예약이 이루어진 스케줄과 중복되는 새로운 스케줄을 게스트가 예약하는 것과, 그러한 새로운 스케줄 예약을 관리자가 승인하는 것이 발생하지 않도록 하여 스케줄 구성 시		

	충돌이 일어나지 않도록 한다.
Description	시스템은 중복 예약과 같은 충돌이 일어나지 않도록 경고 메시지 출력 등의 방법을 통해 기능적 장치를 마련하여 데이터 충돌을 사전에 방지한다. 또한, 이를 통해 관리자와 게스트의 서비스 이용에 있어 혼선이 빚어지지 않도록 한다.

**표 4-11**

No.	SYS_RQ_FR_010	Related Requirement	USER_RS_010
Title	시스템은 관리자가 해당 채널의 예약 신청 현황을 조회할 수 있는 UI를 별도로 제공하여야 한다.		
Description	시스템은 관리자가 해당 채널의 예약 현황을 쉽게 확인할 수 있는 별도의 UI를 제공한다. 이 UI는 관리자에게 각 스케줄 예약 신청의 각 시간, 이용 목적, 신청자 정보 등의 상세한 예약 정보를 조회하고 관리하는 기능을 제공하여, 효과적인 채널 운영을 지원한다.		

**표 4-12**

No.	SYS_RQ_FR_011	Related Requirement	USER_RQ_011
Title	시스템은 관리자가 스케줄 예약 요청에 대한 승인 혹은 거절할 수 있는 별도의 UI를 제공하여야 한다.		
Description	시스템은 스케줄 예약 요청에 대한 승인 또는 거절 권한을 관리자에게만 부여한다. 오직 관리자에게만 이러한 결정을 할 수 있는 사용자 인터페이스를 제공하며, 이를 통해 예약 요청의 관리와 통제가 가능하게 된다. 특정 스케줄 신청을 승인하여 시스템에 반영됐을 경우, 그 스케줄과 시간대가 겹치는 신청은 모두 자동으로 거절되도록 한다.		

**표 4-13**

No.	SYS_RQ_FR_012	Related Requirement	USER_RQ_012
Title	시스템은 관리자의 승인/거절 결과를 해당 예약을 신청한 게스트에게 알림을 발송하여 알린다.		
Description	시스템은 관리자의 승인 또는 거절 결과에 따라 게스트에게 알림을 발송하는 기능을 제공한다. 이를 통해 게스트는 신속하게 예약 상태를 확인하고 필요한 조치를 취할 수 있다.		

#### 4.2 Non-Function Requirements

**표 4-14**

No.	SYS_RQ_NFR_001	Related Requirement	
Title	시스템은 사용자 친화적이고 자연스러운 흐름의 UX를 제공해야 한다.		
Description	시스템은 사용자가 별다른 지체 없이 목표로 하는 기능을 쉽게 수행할 수 있도록 자연스러운 화면 설계 순서, 독립적인 기능 분리 등 사용자에게 편리한 이용 경험을 보장해야 한다.		

**표 4-15**

No.	SYS_RQ_NFR_002	Related Requirement	USER_RQ_006 USER_RQ_008
Title	한 채널의 스케줄을 표시할 때, 주 단위 뷰로, 사람들에게 익숙한 타임 테이블과 같은 형태로 제공할 수 있어야 한다.		
Description	시스템은 한 채널의 스케줄을 주 단위로 보여주는 기능을 제공한다. 이는 타임 테이블 형태로 표현되며, 각각의 예약 사항이 해당하는 시간대에 표시된다. 이를 통해 사용자들은 한 눈에 주간 스케줄을 파악하고, 예약 현황을 확인할 수 있다		

**표 4-16**

No.	SYS_RQ_NFR_003	Related Requirement	USER_RQ_006 USER_RQ_006_01 USER_RQ_008
Title	한 채널 내에서 스케줄마다 다른 색상으로 표기하여 구분할 수 있도록 제공한다.		
Description	시스템은 한 채널 내의 다양한 스케줄을 쉽게 구분할 수 있도록, 각각의 스케줄을 다른 색상으로 표기하는 기능을 제공한다. 이를 통해 사용자들은 각 스케줄이 무엇인지 명확하게 인식하고, 복잡한 일정 정보를 직관적으로 이해할 수 있다.		

**표 4-17**

No.	SYS_RQ_NFR_004	Related Requirement	USER_RQ_006 USER_RQ_007
Title	특정 스케줄 블록을 선택 시에 볼 수 있는 스케줄 상세 정보는 모달 형태로 출력되도록 한다.		
Description	시스템은 사용자가 특정 스케줄을 선택할 경우, 모달 형태로 해당 스케줄의 자세한 정보를 제공하는 기능을 갖추고 있다. 이 모달에는 예약자의 이름, 예약 시간, 예약 내용 등 필요한 상세 정보가 포함되어 있어 사용자들이 쉽게 스케줄에 대한 정보를 파악할 수 있다.		

**표 4-18**

No.	SYS_RQ_NFR_005	Related Requirement	USER_RQ_013
Title	시스템에서 정해진 시간(30초)이 지나면 자동으로 타임 테이블 업데이트를 제공한다.		
Description	시스템은 정해진 시간(30초)이 경과하면 자동으로 타임 테이블을 업데이트하는 기능을 제공한다. 이를 통해 사용자들은 항상 최신의 스케줄 정보를 확인할 수 있어, 일정 관리의 효율성을 높일 수 있다.		

**표 4-19**

No.	SYS_RQ_NFR_006	Related Requirement	
-----	----------------	---------------------	--

Title	앱은 즉각적으로 반응하여 사용자 경험을 저해하지 않아야 한다.
Description	사용자의 동작에 대한 응답 시간은 1초 이내여야 한다.

**표 4-20**

No.	SYS_RQ_NFR_007	Related Requirement	
Title	서버와의 통신속도는 느리지 않아야 한다.		
Description	정상적으로 네트워크에 연결된 상황 (Chrome Network Throttling Fast 3G 기준)에서 3초 이내에 데이터요청에 대한 응답이 이루어져야 한다.		

## 5. Domain Requirements

- 별도로 충족되어야 하는 외부 도메인 요구사항은 없다.

## 6. Supplementary Requirements

### 6.1. Interface requirements

- 안드로이드 / iOS 플랫폼에 관계없이 일관된 UI를 제공하여야 한다.
- 디자인 가이드라인과 색상 스킴을 포함한 모든 기능이 모든 플랫폼에서 동일하게 작동해야 한다.
- UI는 직관적이고 사용하기 쉬워야 하며, 모든 기능에 대한 사용 방법이 명확하게 표시되어 있어야 한다.

### 6.2. Physical requirements

- 앱의 응답은 즉각적이어야 한다.
- 서버와의 통신은 느리지 않아야 한다.

### 6.3. Design requirements

- client-server design
- 추후 확장성을 고려하여 유지보수가 용이하고 모듈화된 설계를 진행한다.
- 현재로서는 하나의 동아리에 대한 앱을 만들고 있지만, 추후에는 여러 개의 동아리를 한 앱에서 볼 수 있는 추가적인 구현을 할 수 있음을 고려하여 설계한다.

### 6.4. Implementation requirements

- Dart Flutter
- Java, Spring Boot, Spring Data JPA
- RESTful API
- DateTime ISO8601 준수
- RDBMS(MySQL, H2)

### 6.5. Additional NFRs

- 검증된 외부 패키지들을 사용하여야 하며, 보안 관련 최신 업데이트와 패치를 받을 수 있도록 한다.
- 해당 시스템에 외부 시스템이 접근하지 못하도록 하여 시스템 안정성을 보장한다.
- 사용자 데이터는 관련 법규를 준수하여 처리되어야 하며, 개인정보의 기밀성과 무결성을 보장한다.

## 7. Requirements Analysis

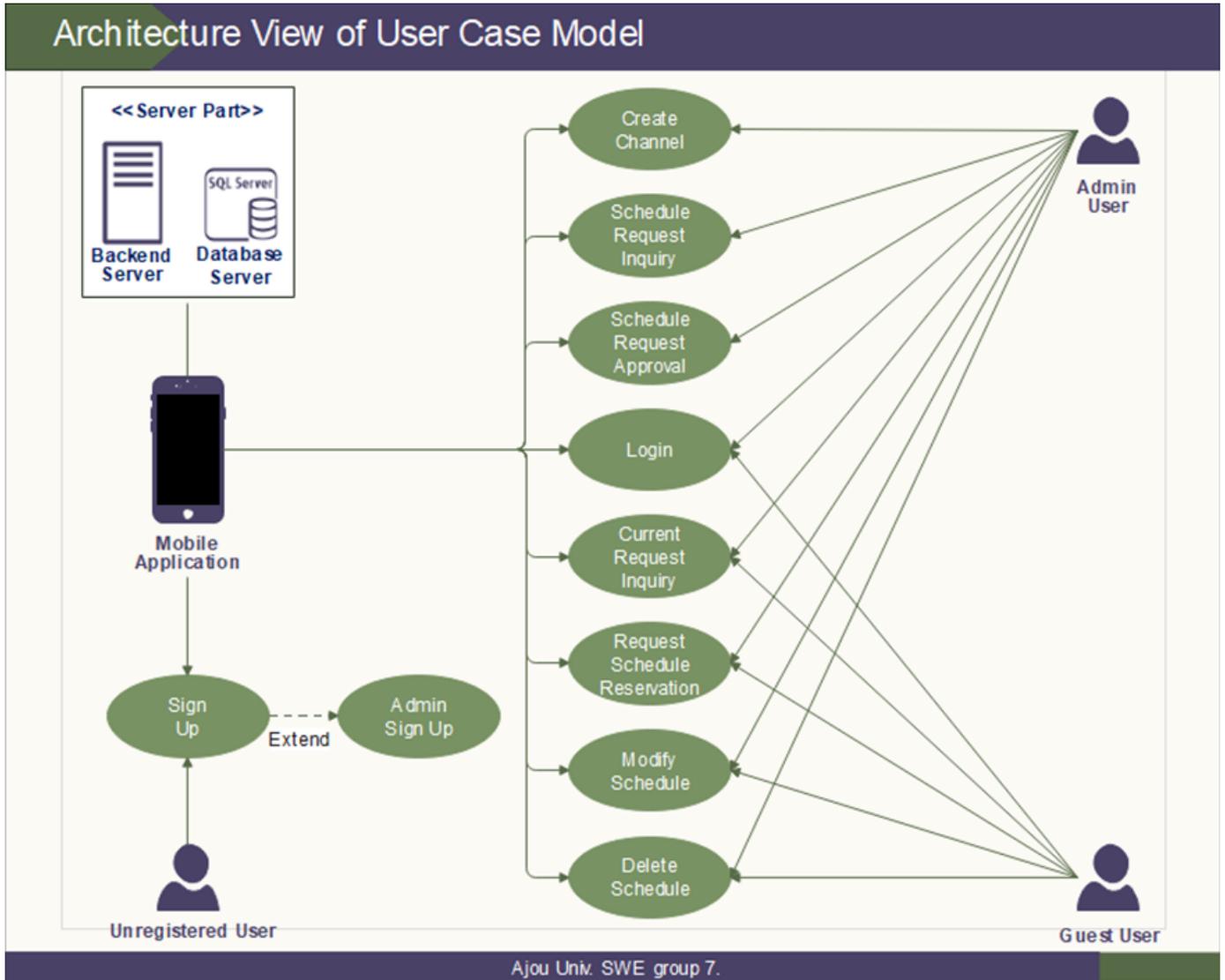


그림 7-1

### 7.1. Use Case: 로그인

표 7-1

Use Case	1
Case Name	로그인
Actors	유저(게스트 및 관리자), 미가입 유저, 모바일 애플리케이션, DB 서버
Description	시스템을 이용하고자 하는 유저는 이전에 등록한 유저 ID와 유저 패스워드를 사용하여 로그인해야 한다.
Pre-Conditions	유저는 회원가입을 통해 유저의 정보와 유저 ID, 유저 패스워드를 미리 등록해야 한다.
Post-Conditions	시스템을 이용할 수 있게 된다.
Primary Flow	<ol style="list-style-type: none"> <li>1. 유저는 로그인 창에 유저 ID와 유저 패스워드를 입력한다.</li> <li>2. 스토리지 서버는 모바일 애플리케이션으로부터 데이터를 수신한다.</li> <li>3. 스토리지 서버는 수신한 데이터와 저장된 데이터의 비교를 수행한다. <ol style="list-style-type: none"> <li>3.1. 만약 데이터가 일치하지 않으면, 스토리지 서버는 모바일 애플리케이션으로 불일치 데이터를 송신한다.</li> </ol> </li> </ol>

3.2. 일치되는 데이터가 있으면, 유저는 시스템을 이용할 수 있게 된다.

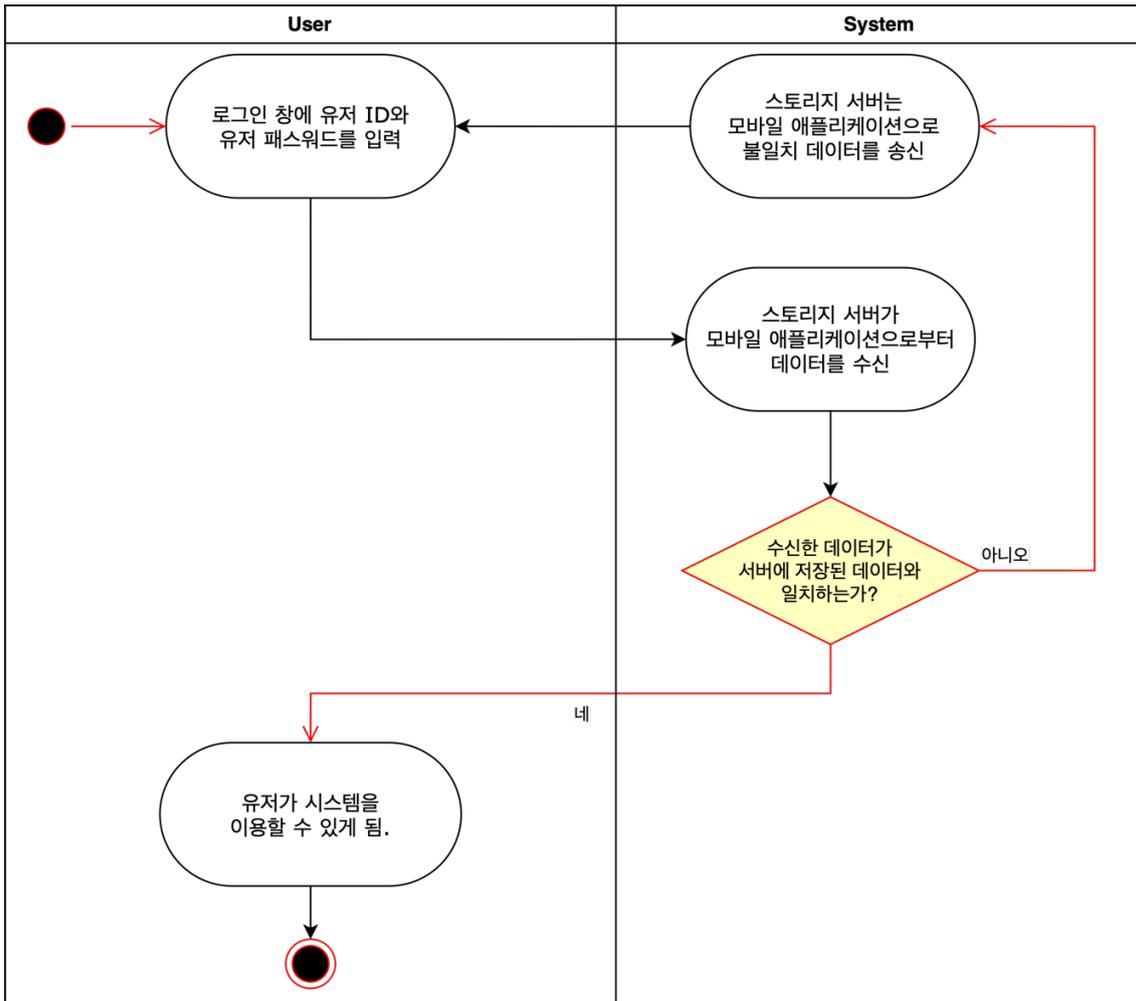


그림 7-2

7.2. Use Case: 유저 등록

표 7-2

Use Case	2
Case Name	유저 등록
Actors	사용자(게스트 및 관리자), 모바일 애플리케이션, 스토리지 서버
Description	유저는 시스템을 이용하기 위해 등록이 필요하다.
Pre-Conditions	유저는 애플리케이션을 사용하기 위한 디바이스를 소유해야 하며, 디바이스에 애플리케이션을 설치해야 한다.
Post-Conditions	유저 정보가 스토리지 서버에 등록되고 유저는 시스템을 이용할 수 있게 된다.
Primary Flow	<ol style="list-style-type: none"> <li>1. 유저가 애플리케이션을 실행한다.</li> <li>2. 유저가 로그인 페이지에서 회원가입 페이지로 이동한다.</li> <li>3. 유저는 회원가입 폼에 유저 ID, 유저 비밀번호, 이름, 전화번호 등의 정보를 입력한다.             <ol style="list-style-type: none"> <li>3.1. 만약 회원가입 폼에 입력되지 않은 정보가 있다면, 애플리케이션은 유저에게 경고 메시지를 전달한다.</li> <li>3.2. 등록 폼에 모든 필수 정보가 입력되면, 해당 정보는 스토리지 서버로 송신된다.                 <ol style="list-style-type: none"> <li>3.2.1. 만약 수신된 데이터가 아이디, 비밀번호, 전화번호의 생성 제약 조건을 만족하지 않으면, 스토리지 서버는 유저에게 경고 메시지를 전달한다.</li> <li>3.2.2. 만약 수신된 데이터가 이미 서버에 존재하면, 스토리지 서버는 유저에게 경고 메시지를</li> </ol> </li> </ol> </li> </ol>



Primary Flow	<ol style="list-style-type: none"><li>1. 관리자가 채널 리스트 보기 버튼을 눌러 리스트 메뉴를 연다.</li><li>1.1. 만약 현재 접속한 계정이 관리자 계정이 아니라면, 채널 추가 버튼을 활성화하지 않는다.</li><li>1.2. 관리자는 채널 추가 메뉴를 선택하고, 애플리케이션은 새 채널 추가 뷰를 보여준다.</li><li>1.3. 관리자는 채널 추가 폼에 추가하려는 채널의 이름과 장소 정보를 기입한다.<ol style="list-style-type: none"><li>1.3.1. 만약 채널 추가 폼에 입력되지 않은 정보가 있다면, 애플리케이션은 유저에게 경고 메시지를 전달한다.</li><li>1.3.2. 등록 폼에 모든 필수 정보가 입력되면, 해당 정보는 DB 서버로 송신된다.<ol style="list-style-type: none"><li>1.3.2.1. 만약 수신된 데이터와 일치하는 내용의 데이터가 이미 서버에 존재하면, DB 서버는 유저에게 경고 메시지를 전달한다.</li><li>1.3.2.2. 채널 정보를 DB 서버에 추가하여 새 채널을 추가하고, 서버는 애플리케이션을 통해 새 채널이 정상적으로 추가되었음을 보여주는 메시지를 전달한다.</li></ol></li></ol></li></ol>
--------------	---

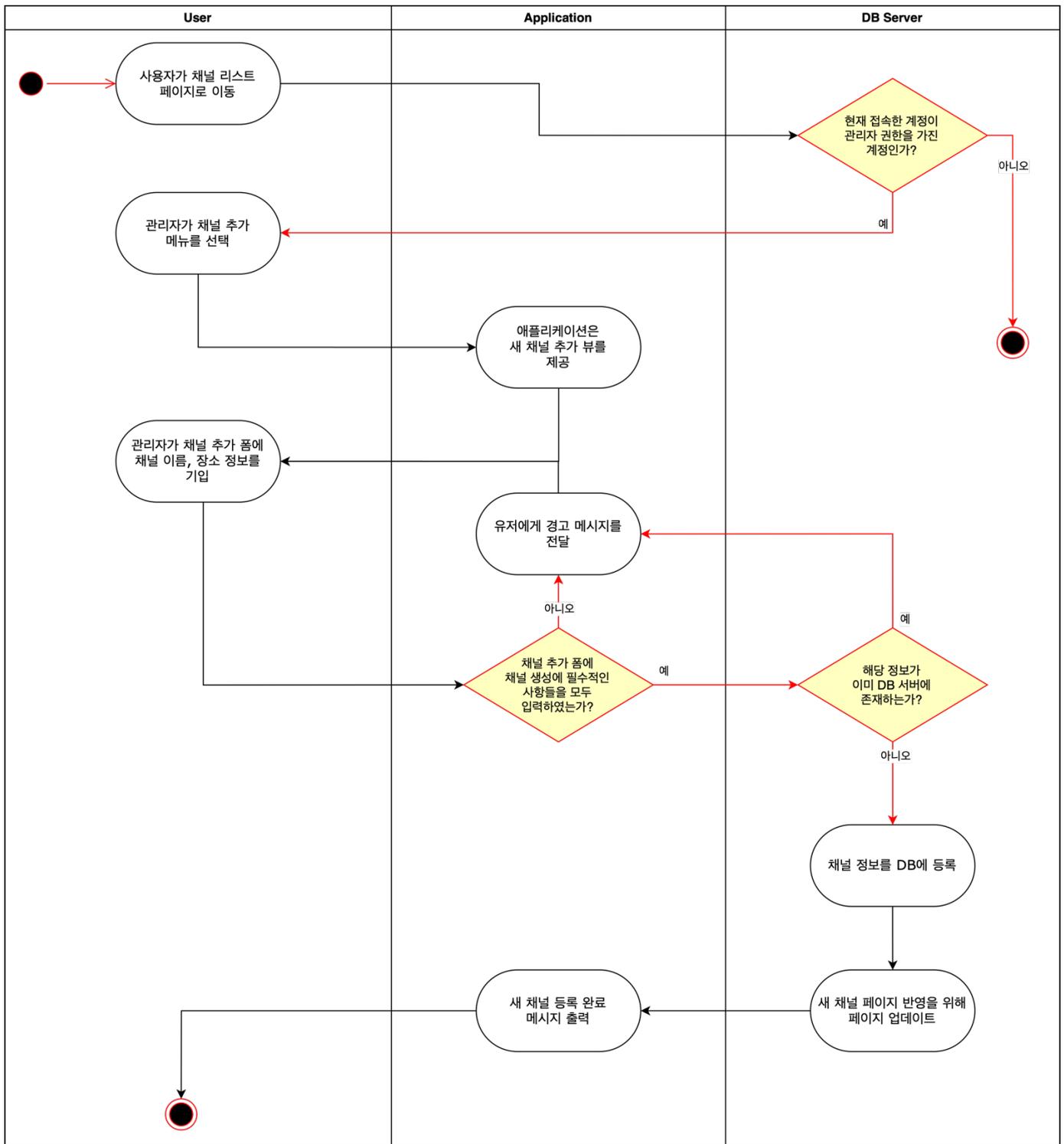


그림 7-4

7.4. Use Case: 스케줄 예약 요청

표 7-4

Use Case	4
Case Name	스케줄 예약 요청
Actors	유저(게스트), 모바일 애플리케이션, 백엔드 애플리케이션 서버, DB 서버
Description	게스트가 스케줄 예약을 요청한다.
Pre-Conditions	게스트가 현재 로그인 상태여야 한다.
Post-Conditions	게스트가 예약 요청한 스케줄이 관리자의 스케줄 승인 페이지로 이동한다
Primary Flow	1. 게스트가 동아리 채널의 타임 테이블 페이지로 이동한다.

2. 게스트가 타임 테이블 페이지에서 스케줄 예약 버튼을 누른다.
3. 애플리케이션은 스케줄 예약 페이지를 보여준다.
4. 게스트는 스케줄 예약 폼에서 예약일자와 예약시간을 선택하고, 예약명과 비고사항을 입력한다.
5. 게스트가 확인 및 제출 버튼을 누른다.
  - 5.1. 만약 비고사항을 제외한 예약일자, 예약시간이 선택되지 않았거나, 예약명이 입력되지 않았다면, 애플리케이션은 유저에게 경고 메시지를 전달한다.
  - 5.2. 모든 필수 정보가 정상적으로 입력되었다면, 해당 예약 데이터가 DB 서버에 저장되고, 관리자용 예약 관리 페이지에 해당 정보가 업데이트된다.
  - 5.3. 서버는 애플리케이션을 통해 새 채널이 정상적으로 추가되었음을 보여주는 메시지를 전달한다.

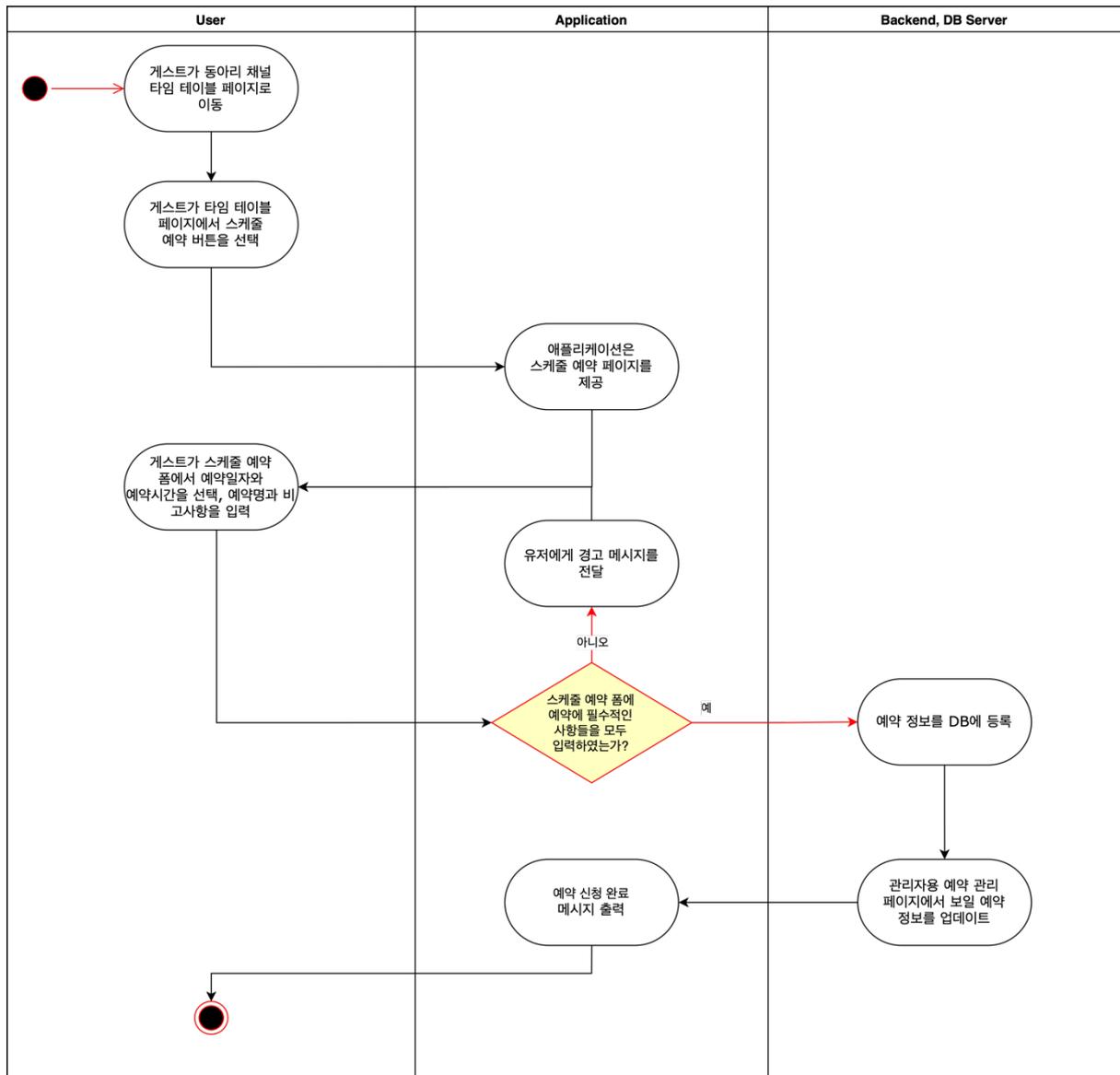


그림 7-5

7.5. Use Case: 스케줄 상세정보 조회 및 삭제

표 7-5

Use Case	5
Case Name	스케줄 상세정보 조회 및 삭제
Actors	유저(관리자, 게스트), 모바일 애플리케이션, 백엔드 애플리케이션 서버, DB 서버

Description	유저가 예약 상세정보를 조회하고, 게스트가 현재 예약된 스케줄을 삭제한다.
Pre-Conditions	유저가 현재 로그인 상태이고, 현재 조회 가능한 예약이 존재해야 한다.
Post-Conditions	예약 삭제 신청에 성공하면 게스트의 예약이 삭제된다.
Primary Flow	<ol style="list-style-type: none"> <li>1. 게스트가 동아리 채널의 타임 테이블 페이지로 이동한다.</li> <li>2. 게스트가 현재 타임 테이블상의 예약 블록을 선택한다.</li> <li>3. 애플리케이션은 스케줄 예약 상세 정보 페이지를 보여준다.</li> <li>4. 게스트는 스케줄 예약 상세 정보 페이지에서 예약 삭제 버튼을 선택한다. <ol style="list-style-type: none"> <li>4.1. 만약 서버에 등록된 예약자 정보와 현재 유저의 정보가 일치하지 않으면, 애플리케이션은 유저에게 경고 메시지를 보낸다.</li> <li>4.2. 만약 서버에 등록된 예약자 정보와 현재 유저의 정보가 일치하면, 해당 예약 데이터가 DB 서버에서 삭제된다. <ol style="list-style-type: none"> <li>4.2.1. 서버는 애플리케이션을 통해 예약 정보가 삭제되었음을 보여주는 메시지를 전달하고, 타임 테이블을 업데이트한다.</li> </ol> </li> </ol> </li> </ol>

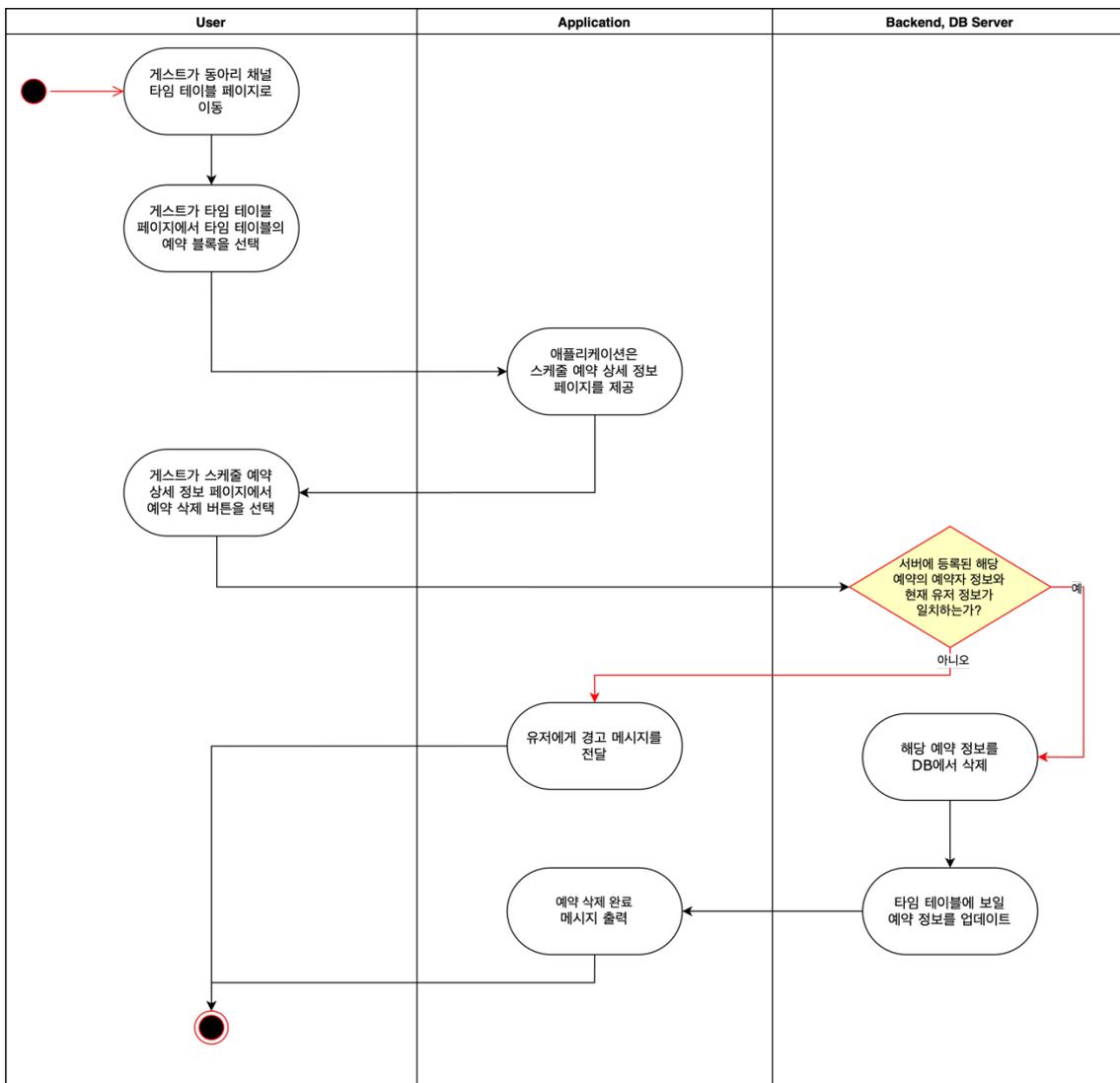


그림 7-6

7.6. Use Case: 관리자의 스케줄 승인/거절

표 7-6

Use Case	6
Case Name	관리자의 스케줄 승인/거절
Actors	유저(관리자), 모바일 애플리케이션, 백엔드 애플리케이션 서버, DB 서버

Description	관리자가 게스트가 요청한 스케줄을 승인하거나 거절할 수 있다
Pre-Conditions	유저는 관리자 권한이 부여된 계정으로 로그인을 성공해야 하며, 현재 승인을 대기 중인 예약이 존재해야 한다.
Post-Conditions	예약의 승인 여부에 따라 해당 예약 정보가 타임 테이블에 반영된다.
Primary Flow	<ol style="list-style-type: none"> <li>1. 사용자가 로그인을 한다.</li> <li>2. 접속한 계정의 권한을 확인한다. <ol style="list-style-type: none"> <li>2.1. 만약 현재 접속한 계정이 관리자 계정이 아닐 경우 <ol style="list-style-type: none"> <li>2.1.1. 채널 예약 관리 페이지 버튼을 활성화하지 않는다.</li> </ol> </li> <li>2.2. 만약 현재 접속한 계정이 관리자 계정일 경우 <ol style="list-style-type: none"> <li>2.2.1. 관리자는 채널 예약 관리 페이지에 접속하고, 애플리케이션은 채널 관리 페이지를 보여준다.</li> <li>2.2.2. 관리자는 게스트의 예약 요청 정보들을 확인하고, 해당 예약을 승인하거나 거절한다.</li> <li>2.2.3. 예약 요청의 승인 여부 <ol style="list-style-type: none"> <li>2.2.3.1. 만약 관리자가 예약 요청을 승인하면, 백엔드 애플리케이션 서버에서 게스트에게 승인 알림을 전송한다. <ol style="list-style-type: none"> <li>2.2.3.1.1. DB 서버는 해당 스케줄 정보를 저장한다.</li> <li>2.2.3.1.2. 백엔드 애플리케이션 서버에서 예약 정보를 업데이트한다.</li> </ol> </li> <li>2.2.3.2. 만약 관리자가 요청을 거절하면 모바일 애플리케이션 서버에서 게스트에게 거절 알림을 전송한다. <ol style="list-style-type: none"> <li>2.2.3.2.1. 백엔드 애플리케이션 서버에서 예약 정보를 업데이트한다.</li> </ol> </li> </ol> </li> </ol> </li> </ol> </li> </ol>

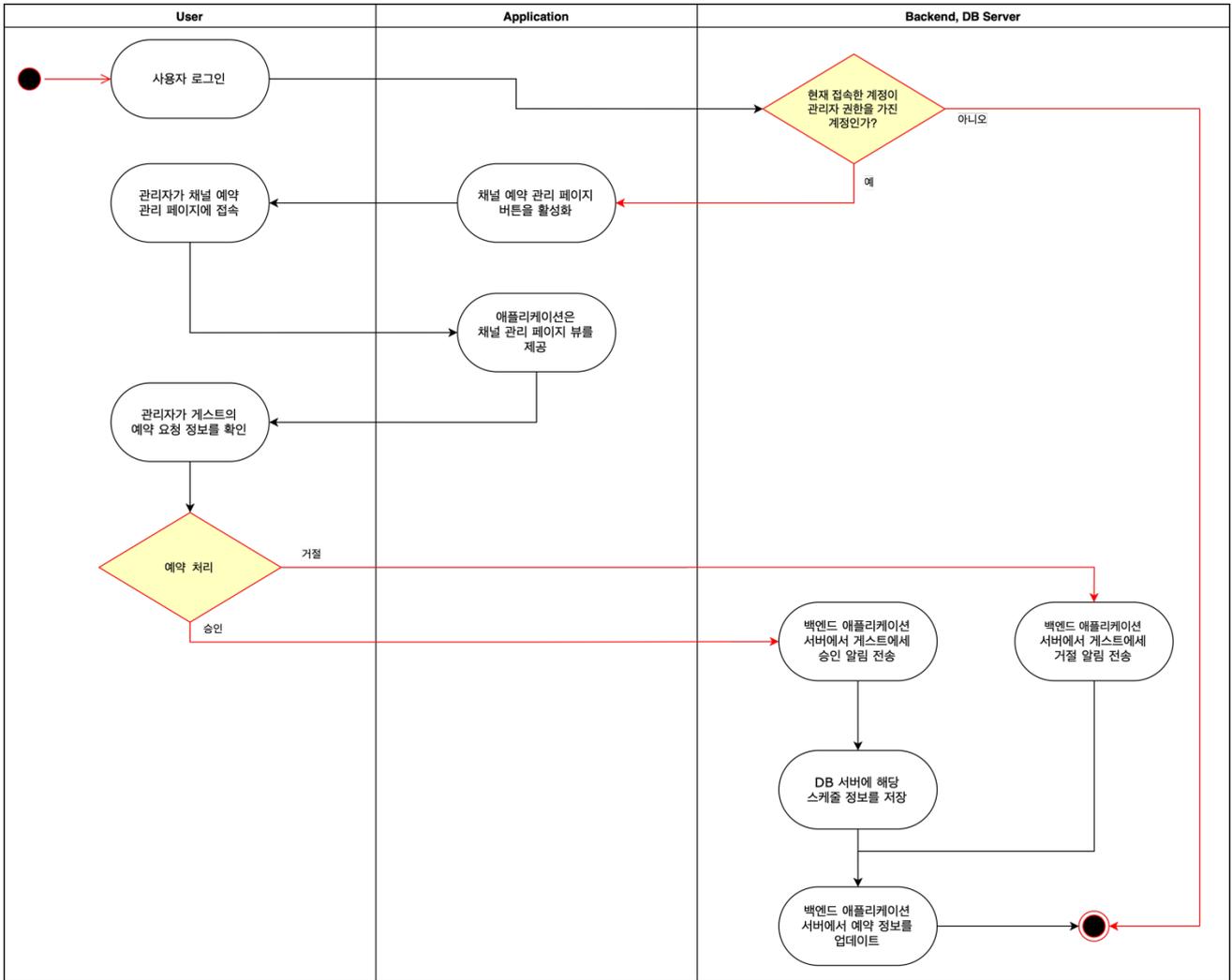


그림 7-7

## 8. Validation Criteria (Metrics and Measures)

표 8-1

No.	SYS_RQ_VA_001	Related Requirement	USER_RQ_009 SYS_RQ_NFR_001
Title	시스템은 사용자 친화적이고 자연스러운 흐름의 UI 및 UX를 제공해야 한다.		
Metrics	설문 조사 및 레퍼런스 조사		
Measure	예비 사용자 설문 조사 및 비슷한 서비스 레퍼런스 조사		

표 8-2

No.	SYS_RQ_VA_002	Related Requirement	SYS_RQ_NFR_006
Title	클라이언트 앱의 반응 시간은 1초 이내여야 한다		
Metrics	시간		

Measure	Dart Testing tool
---------	-------------------

**표 8-3**

No.	SYS_RQ_VA_003	Related Requirement	SYS_RQ_NFR_007
Title	정상적으로 네트워크에 연결된 상황 (Chrome Network Throttling Fast 3G 기준)에서 3초 이내에 백엔드 서버의 요청에 대한 응답을 받을 수 있어야 한다.		
Metrics	시간		
Measure	Java Testing tool, Dart Testing Tool		

## 9. Appendix: Project Status

### 9.1. 미팅

**표 9-1**

일자	구성원	진행내용
2023. 10. 09.	박종범, 이승주, 정유환	프로젝트 환경 세팅 및 Prototype 설계
2023. 10. 13.	김동령, 이승주	Prototype 구체화
2023. 10. 13.	박종범, 정유환	API 명세 제작
2023. 10. 15.	전체	Requirements Specification 설계
2023. 10. 19.	전체	Requirements Specification 구체화
2023. 10. 29.	전체	Requirements Specification 마무리 및 개발 인증관련 논의

### 9.2. Commit Logs

#### 9.2.1. Server ( <https://github.com/software7/weplan-server.git> )

- \* 9160f84 로그인 테스트 작성 및 API 문서화
- \* b1a90ec 회원가입 API 문서화
- \* 0dd3b17 스케줄 생성 로직 구현
- \* c5207eb 모든 채널 조회 로직 구현
- \* dc3babd 단일 채널 조회 로직 구현
- \* cb28e0c 채널 생성 로직 구현
- \* 12dcb31 Access, Refresh Token 관리 방식 변경
- \* 9739779 회원 로그인 구현
- \* f53e27f 회원가입 입력 값 검증 추가 및 도메인 필드 수정
- \* 87f00e0 회원 가입 로직 구현
- \* f530e3c 테스트 초기 설정
- \* e04c1a1 도메인 초기 모델 구현
- \* 41f61ff JWT 토큰 발급 로직 (AccessToken, RefreshToken)
- \* feead8b 공통 예외 처리 및 Jwt - ArgumentResolver 처리
- \* a84f5cd global config 세팅
- \* a75c29a build.gradle 및 yml 설정 추가
- \* d8b5fb9 초기 설정

## 9.2.2. Client ( <https://github.com/software7/weplan.git> )

- \* cc03335 feat: wrap materialapp with authservice provider
- \* 5b6660f feat: authService
- \* dc25703 remove: SignRequestType & add: dio error interceptor
- \* 0004347 feat: add login form & add validators for sign forms
- \* 920c074 remove: default test file
- \* 6cda60d chore: change linting rules
- \* 1d773b2 Create dart.yml
- \* 187454f test: signin header Authorization response
- \* 70953fd feat: sign api with header response & add error type
- \* 7aaac26 chore: exclude `\*.g.dart` from analyzing
- \* 8b26e03 rename: page to screens
- \* 175da21 feat: add basic models & api via retrofit
- \* 950bf86 feat: simple timetable & drawer view
- \* ca80653 chore: force code style & pre-commit
- \* d6f5a16 initial commit