

Project Proposal

동아리 스케줄 공유를 통한 공동공간 예약 및 조회 시스템 WePlan

7 조 박종범, 정유환, 김동령, 이승주

Table of Contents

<u>1</u>	<u>MOTIVATION AND BACKGROUND</u>	<u>3</u>
1.1	MOTIVATION	3
1.2	IMPORTANCE	3
<u>2</u>	<u>PROJECT PROBLEM STATEMENT</u>	<u>3</u>
2.1	PROBLEM SPECIFICATION	3
2.2	IMPACT	3
<u>3</u>	<u>SCOPE AND OBJECTIVE</u>	<u>4</u>
3.1	IN THE SCOPE	4
3.2	OUT OF SCOPE	4
3.3	OBJECTIVE	4
<u>4</u>	<u>RELATED WORK/TRENDS</u>	<u>4</u>
4.1	EXISTING SOLUTIONS	4
4.2	CURRENT TREND	5
4.3	DRAWBACKS / WEAKNESS IN THE CURRENT TREND OR EXISTING SOLUTIONS	5
<u>5</u>	<u>PROPOSED IDEA (OUR APPROACH)</u>	<u>6</u>
<u>6</u>	<u>MAIN FEATURES OF THE SYSTEM-TO-BE</u>	<u>6</u>
6.1	MAIN FEATURES	6
6.2	HIGH LEVEL REQUIREMENTS	6
<u>7</u>	<u>TECHNICAL APPROACHES</u>	<u>6</u>
7.1	COMMON	6
7.2	FRONTEND: FLUTTER + DART FOR MOBILE APP DEVELOPMENT	7
7.3	BACKEND	7
<u>8</u>	<u>CONCERNS</u>	<u>7</u>

9	CHALLENGES	7
10	PROJECT PLAN	8
11	REFERENCES	8

Figures

Figure 1-1 카포 합주시간 예약	3
Figure 4-1 IoT Space - 회의실 예약 시스템	4
Figure 4-2 IoT Space - 회의실 예약 시스템	4
Figure 4-3 Slack	5
Figure 4-4 Notion	5
Figure 4-5 Notion	5
Figure 10-1 Project Plan Gantt Chart	8

1 Motivation and Background

1.1 Motivation

평소 학교에서 동아리 활동을 진행하면서, 동아리방 및 교내 시설 등 동아리 구성원 간의 공용 공간을 관리하는 시스템의 부재로 불편함을 겪었던 경험이 많았다.

예를 들어, 밴드 동아리의 경우, 동아리방에서 같은 공연을 준비하는 팀끼리 연습을 하게 되는데, 해당 시간에는 동아리방을 해당 팀이 독점적으로 이용해야 하기 때문에 미리 공연부장에게 예약을 받고, 매주 해당하는 시간표를 공연부장이 직접 공지하는 방식으로 운영되었다. (Figure 1)

이와 비슷한 예로, 테니스 동아리의 경우 매주마다 구글 폼을 통해 교내 테니스장 예약 신청을 받은 뒤, 동아리 관리자가 해당 주차의 시간표를 공지해주는 방식으로 테니스장을 예약하여 이용하는 방식으로 예약 및 관리가 되고 있다.

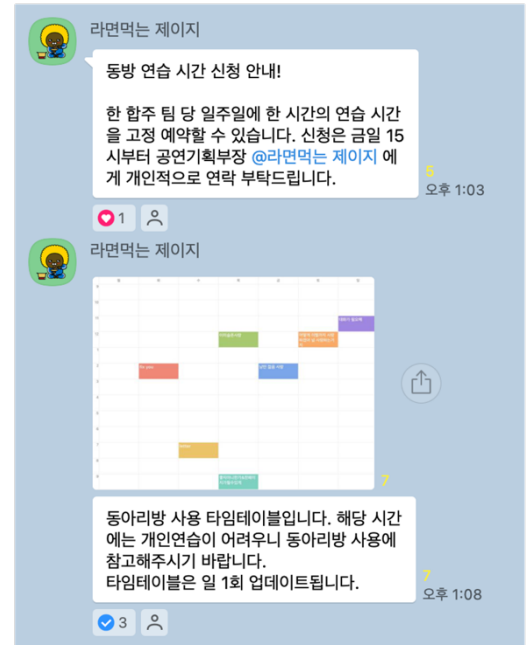


Figure 1-1 카포 합주시간 예약

1.2 Importance

위의 두 예시 모두 공통적으로 예약을 진행할 때에 해당 시간에 예약이 가능한지는 관리자만이 알고 있다. 이로 인하여 이용자 입장에서는 매번 관리자와의 연락을 통해 확인 및 승인을 거쳐야 하는 불편함이 있고, 관리자 역시 이를 확인 및 정리하여 이용자에게 공지해야 하는 번거로운 과정을 거쳐야 한다는 단점이 있다. 이러한 불편함을 겪었던 과거 경험과 더불어, 해당 공간 관리자인 밴드 동아리 회장으로부터 동아리 공용 공간을 예약하는 시스템을 제작할 수 있을 지에 대한 제안을 받았던 경험이 있어 이번 기회를 통해 본 프로젝트를 기획하게 되었다.

2 Project problem statement

2.1 Problem Specification

앞서 언급한 밴드 동아리의 예시에서는 매주 팀원들이 공연부장에게 예약하고자 하는 시간에 동아리 방을 사용할 수 있는지 확인하는 과정, 동아리 방을 사용할 것을 요청해야 하는 과정이 필요하고, 공연부장 역시 동아리 방 이용 시간표를 매번 제작해야 하는 번거로움이 존재한다. 테니스 동아리 역시 마찬가지로 관리자가 부원들이 제출한 폼을 직접 확인하고, 먼저 예약한 사람을 우선순위로 두어 이를 바탕으로 직접 시간표를 제작하여 이를 공지해야 하는 불편함이 존재한다.

따라서 이번 프로젝트를 통해 해결하고자 하는 문제는 구성원들 간의 스케줄을 공유하여 공용 공간 이용시간 공유의 어려움을 해결하고, 해당 스케줄에 공간 사용을 예약하는 과정과 이를 공지하는 번거로움을 해소하고자 한다.

2.2 Impact

해당 시스템을 통해서 같은 공용 공간을 이용하고자 하는 구성원들 간에 시간을 조율하여 스케줄을 공유하고 예약하는 과정이 간소화될 것을 기대할 수 있고, 예약된 일정을 공유하여 해당 시간에는 타 구성원들에게 사용이 불가능함을 고지하기 더욱 수월할 것이다. 이에 따라 동아리 구성원은 동아리 활동의 본질에 더욱 집중할 수 있고, 동아리 관리자 역시 매주마다 예약을 받아 이를 공지해야 하는 수고를 덜 수 있다. 또한, 공용공간을 더 밀도 있게 적극 활용할 수 있어 이용 효율이 좋아질 것이다.

3 Scope and Objective

3.1 In the Scope

- 시간표 어플인 에브리타임과 같이 동아리 채널의 사용 현황 타임 테이블을 제공한다.
- 원하는 시간에 이용하고자 하는 공간을 예약한다.
- 각 공간에 대한 예약 신청을 설정에 따라 관리자가 승인하거나 자동 승인하여 타임 테이블에 적용한다.
- 각 회원마다 자신의 예약 현황을 확인할 수 있는 페이지를 제공한다.

3.2 Out of Scope

- 회원들 간의 단순 소통기능 (ex. 채팅, 게시물 등)은 이번 구현에서 제외하기로 한다.
- 회원들 간의 스케줄의 공유로 구현을 제한하고, 실제 공용공간 시간이용의 제한은 자율적으로 실시한다.
- 스케줄에 대한 정보는 이름, 시간, 이용자, 장소 등 해당 스케줄에 대해 필수적인 정보들만으로 제한하며 기타 부가적인 정보를 (ex. 사진, 영상, 음악 등) 첨부하는 기능은 제외한다.

3.3 Objective

- 해소하고자 하는 문제: 회원들 간 스케줄 예약 및 공유의 어려움, 임원들의 스케줄 공지의 불편함
- 해결 방법: 같은 동아리의 회원들 간에 스케줄 예약 및 공유를 편리하게 제공하는 앱 개발

4 Related work/trends

4.1 Existing Solutions

- IoT Space - 회의실 예약 시스템

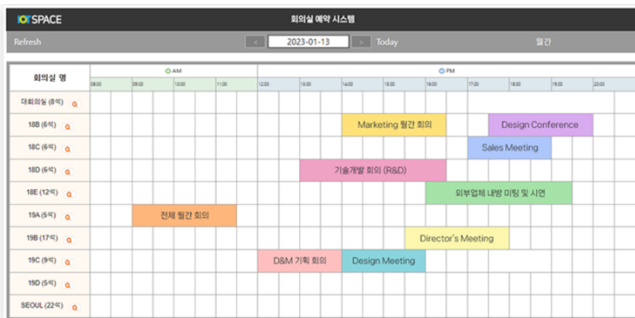


Figure 4-1 IoT Space - 회의실 예약 시스템



Figure 4-2 IoT Space - 회의실 예약 시스템

해당 서비스는 특히 공유 오피스를 대상으로, 회의실 예약과 각 회의실의 예약 정보 및 일정(회의명, 예약시간, 예약자 명)등을 효율적으로 등록 및 조회하는 서비스이다. 서비스 내에서 각 회의실의 예약 현황 및 일정을 조회할 수 있고, 회의실 입구에 별도의 디스플레이 기기를 부착하여 해당 회의실의 예약 및 현황을 확인할 수 있도록 제공하고 있다. 또한, 별도의 앱이나 프로그램 설치 없이 데스크탑 및 모바일 기기로 해당 서비스에 접속할 수 있는 접근성을 확보하고 있다.

- Slack

슬랙은 클라우드 컴퓨팅 기반 인스턴트 메신저 및 프로젝트 관리용 협업 툴이다. 현재 웹 앱과 안드로이드/iOS 모바일 앱 형태로 제공이 되고 있다. 슬랙은 메신저 기능 중심의 업무용 협업 툴로서 프로젝트 단위로 구성한다. 메세징, 채널, 음성 및 영상 통화, 파일 공유 등의 기능을 제공하고 있다. 또한, 개방성을 중점으로 슬랙 API를 제공하여 사용자가 필요한 어플리케이션을 만들어 연동할 수 있도록 하고 있다.

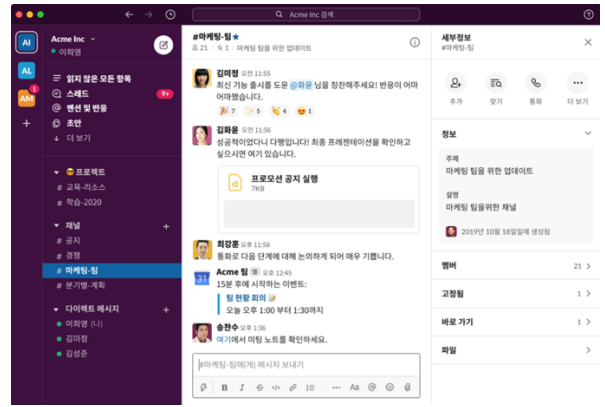


Figure 4-3 Slack

- Notion

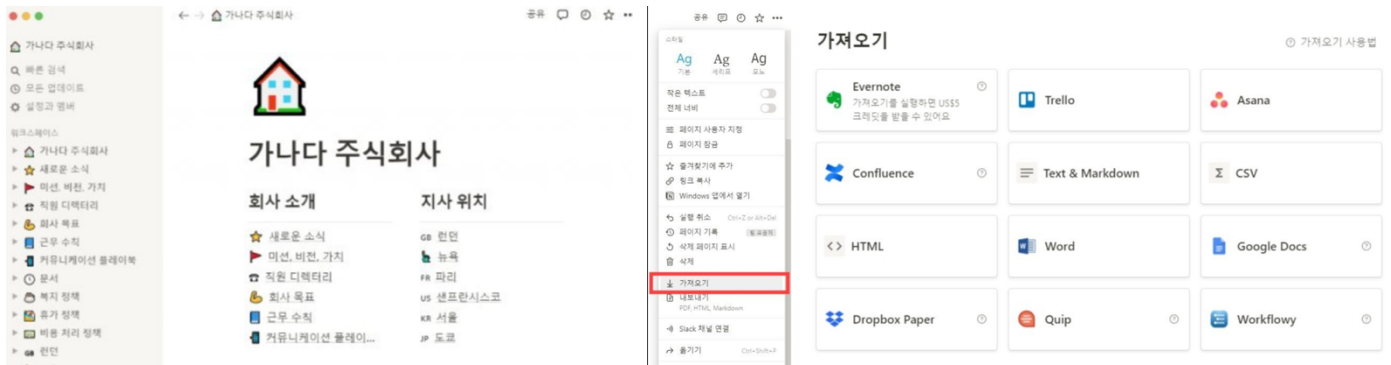


Figure 4-5 Notion

Figure 4-4 Notion

노션은 개인적인 문서 및 노트 작업을 중심으로, 이를 다른 사용자와 공유 및 관리할 수 있는 협업 기능을 제공하고 있는 서비스이다. 워크스페이스 단위로 프로젝트를 나누고 문서를 페이지 단위로 작성하는 구조이며, 페이지안에 이미지, 영상, 링크, 파일, 코드, 외부 API 등 다양한 형태의 데이터를 삽입할 수 있으며 페이지안에 페이지도 삽입할 수 있다. 개인의 문서 작성을 중심으로 다양한 기능을 임베드하여 일정 관리도 수행할 수 있으며, 페이지 단위로 다른 사용자를 초대하여 협업도 진행할 수 있다.

4.2 Current Trend

현재 일정 관리, 예약 시스템은 사용자들이 어디서든 접근할 수 있고 편리하게 서비스를 이용할 수 있도록 대부분 모바일 환경에서 이용할 수 있도록 하고 있고, 업데이트 사항을 실시간으로 확인할 수 있는 유비쿼터스한 환경을 제공하고 있다. 또한, 단순한 일정 등록 및 공유 기능을 넘어서 협업에 필요한 다양한 기능들을 함께 올인원으로 제공하고 있는 추세이다.

4.3 Drawbacks / Weakness in the current trend or existing solutions

해당 프로젝트에서 제공하고자 하는 서비스는 '대학교 커뮤니티 내에서 동아리 별 일정 관리 및 동아리 공간 예약 시스템' 이다. 그러나 앞서 제시한 세 서비스는 학교 커뮤니티 뿐만 아니라 다양한 프로젝트를 일반적으로 포괄할 수 있는 '올인원 협업툴'로 제공되고 있기 때문에, 프로그램의 규모자체가 매우 크다. 따라서, 현재 상용화된 서비스들은 다양한 기기에서 접속할 수 있는 접근성은 보장하고 있으나, 보다 빠르고 간편한 사용성을 제공하기에는 어려운 환경이다. 이에 따라, WePlan은 교내 동아리 행사 일정, 동아리 별 공용 공간 예약을 조회할 수 있는 타임 테이블을 제공하고 동아리 공간 사용 일정을 빠르게 추가하고 삭제할 수 있는 두 기능을 중심으로 하여, 대학생 커뮤니티에 적합한 가볍고 직관적인 서비스를 제공하고자 한다.

5 Proposed Idea (Our Approach)

각 '동아리 채널'의 예약 일정을 타임 테이블로 시각화하여 보다 직관적인 UI로 사용자가 쉽게 확인할 수 있도록 한다.

타임테이블을 참고하여 각 사용자가 예약 현황을 확인하여 간편하게 예약/취소를 할 수 있으면서도, 각 공간의 관리자가 사용을 승인하는 구조로 중복 예약이 발생하지 않도록 한다.

6 Main features of the system-to-be

6.1 Main Features

- 타임 테이블 시각화
 - 각 동아리의 공용 공간 예약 일정을 타임 테이블 형식으로 시각화하여 예약/공실 여부를 쉽게 확인할 수 있으며 예약하는 팀마다 색을 다르게 하는 등 사용자가 쉽게 확인할 수 있는 직관적인 UI를 제공한다.
- 타임 테이블 시간 예약
 - 사용자는 각 동아리의 타임 테이블을 참고하여 예약 현황을 빠르게 조회하고 예약/취소를 할 수 있다.
- 관리자 승인
 - 각 동아리의 관리자가 사용자의 예약을 승인하거나 시간에 따라 자동승인 될 수 있도록 하여 중복 예약을 방지한다.
- 동아리 채널 개설
 - 동아리의 관리자가 해당 서비스를 시작하기 위한 과정을 구현하여 시작하기 수월하도록 한다.

6.2 High Level Requirements

- 동아리 채널의 타임 테이블에 대한 실시간 업데이트가 필요하다. 타임 테이블 시각화 시, 예약/취소는 실시간으로 시스템에 업데이트되어, 다른 사용자가 동시에 예약 현황을 실시간으로 확인할 수 있도록 해야 한다. 이로써 사용자들에게 최신 정보를 제공한다. 또, 중복된 예약으로 충돌이 생기지 않도록 해야 한다.
- 관리자 권한 및 인증 프로세스가 필요하다. 각 공간의 관리자에게 관리 권한을 부여할 때, 별도의 관리자 인증 및 권한 부여 프로세스를 구현해야 한다. 이를 통해 관리자와 일반 사용자의 역할 및 권한이 구분되며 예약 관리를 효율적으로 처리할 수 있다.
- 타임 테이블 시각화를 할 때 예약/취소가 다른 사용자에게도 실시간으로 보일 수 있도록 가능한 빠른 업데이트가 필요하다. 공간에 대한 예약 승인은 각 동아리 채널의 관리자가 담당하는데 이 관리자에게 관리 권한을 줄 때 다른 사용자와는 다른 인증 과정이 필요하다.
- 예약 신청 결과가 관리자에 의해 결정되었을 때 사용자가 알림을 받을 수 있는 형태로 제공되어야 한다.

7 Technical approaches

7.1 Common

클라이언트와 서버 간의 통신을 위해 REST API를 설계한다.

Git, Github을 이용하여 코드 변경 사항을 추적하고 협업하는 툴로서 사용한다.

Notion의 Kanban Board를 이용하여 팀원 간에 진행상황을 공유하고, Gantt Chart를 이용하여 프로젝트의 전체 일정을 관리한다. 그리고, 기타 회의록 등의 문서 및 팀원 간 연락처, 일정 등을 공유한다.

7.2 Frontend: Flutter + Dart for Mobile App Development

Flutter라는 프론트엔드 모바일앱 라이브러리를 이용하여 모바일 앱을 개발한다. Flutter를 이용하면, Android와 iOS, Web, macOS, Windows specific한 Native Code를 최소화하여 플랫폼에 관계없이 동일한 코드로 앱을 제작할 수 있다는 유용함이 있어 선택하게 되었다. 또, 다른 Cross-Platform 프론트엔드 개발 라이브러리인 React Native도 고려해보았으나, 단기간에 빠른 완성물을 만들어 내기 위해서는 초기 제공되는 리소스가 풍부한 Flutter가 더 적합하다고 판단하였다.

7.3 Backend

AWS EC2를 이용해 백엔드 애플리케이션 서버를 구축하며 AWS RDS를 이용하여 데이터베이스 서버를 구축한다. 백엔드 개발로 단순 I/O 처리를 많이 할 경우 성능이 더 좋은 Node.js를 이용할 수도 있지만 Java 기반의 Spring이 멀티 스레드 기반으로 안정적인 서버를 운영할 수 있고 팀원들의 숙련도를 고려하였을 때 더 나은 방법이라고 생각하였다. 여기에 더해 자동 구성 기능, 의존성 관리, 내장 톰캣을 제공하는 Spring Boot를 사용한다. DB 접근 기술로는 Spring Data JPA, QueryDsl을 이용하여 sql 쿼리를 작성한다.

8 Concerns

- 관리자 권한 위임: 각 동아리 채널의 관리자가 변경되는 경우 어떠한 방식으로 관리자 권한을 위임할 것인지에 대한 지정이 필요하다. 각 관리자 계정에 대하여 계정 하나를 지정해 놓고 자치적으로 양수자에게 양도하는 방식으로 진행할 것인지, 아니면 관리자 권한을 인계하는 기능을 사용할 것인지에 대한 선택이 필요할 것이다. 전자의 경우, 계정이 외부로 유출되는 등의 보안 이슈가 있을 것이고, 후자의 경우 해당 기능을 위한 별도의 구현이 필요할 것이다.
- 동일 사용자의 다중 예약 시도 가능성: 동일 사용자가 무분별한 다중 예약을 시도하는 경우, 해당 요청이 서버에 부하를 일으킬 가능성이 존재한다. 따라서 한 계정 당 일정 기간 내에 최대 예약 신청 횟수를 제한하거나, 예약 간 제한시간을 두는 등의 추가적인 구현이 필요할 것이다.
- 사용자 정보 수집: 사용자의 개인정보 수집의 범위를 어디까지 두어야 하는지에 대한 면밀한 검토가 필요하다. 학교 구성원 여부에 대한 판별 및 타임 테이블에 보여주기 위한 최소한의 범위 내에서 사용자 정보를 수집해야 할 것이며, 또 타임 테이블에는 이름, 소속 등의 정보를 어디까지 노출할 것인지에 대한 논의가 필요할 것이다. 마지막으로, 이들을 외부에 유출시키지 않기 위해 추가적인 보안 조치가 필요할 것이다.
- 예약 마감 시간: 인가제로 예약이 진행되는 만큼, 신청하는 사용 기간이 신청하는 시점 바로 직후인 경우 관리자의 예약 승인이 곧바로 이루어지기 어렵다. 따라서, 직전에 신청되는 예약을 동아리 구성원 내에서 자율적으로 처리하도록 할 것인지, 아니면 일정 시간 이후의 예약은 자동으로 취소되도록 할 것인지에 대하여 고찰이 필요하다.

9 Challenges

이 프로젝트에서 가장 큰 도전은 다양한 사용자가 불편을 겪는 상황을 정확하게 이해하고, 이를 시스템에 반영하는 것이다. 각각의 동아리와 그룹이 서로 다른 운영 방식과 요구 사항을 가지고 있기 때문에, 이들 모두를 만족시키는 범용적인 시스템을 만드는 것은 쉽지 않다. 사용자 인터뷰나 설문조사 등을 통해 실제 사용자의 필요성을 파악하고, 이를 기반으로 기능 목록을 작성하는 과정이 필요하다.

또한, 여러 사용자가 동시에 같은 시간대를 예약하려고 할 때 발생할 수 있는 동시성 문제 역시 중요한 과제이며, 데이터의 일관성을 유지하면서 충돌 없이 예약 처리를 해야 하기 때문에 성능이 떨어지지 않으면서 효율적인 동시성 관리 방법을 찾아야 한다.

사용자 인터페이스 디자인 역시 중요한 과제이다. 사용자가 쉽게 예약할 수 있도록 직관적이고 친숙한 인터페이스를 제공해야 한다.

다음으로, 사용자 알림 기능은 이 시스템에서 중요한 부분을 차지하게 될 것이다. 사용자가 예약을 하거나, 예약이 승인되거나 거절되었을 때 즉시 알림을 받아야 한다. 이러한 알림 기능은 사용자 경험을 향상시키는 데 크게 기여할 수 있다. 사용자는 시스템에 대해 지속적으로 인지하고 상호작용할 수 있으며, 필요한 정보를 얻어서 효율적으로 자신의 일정을 관리할 수 있다. 하지만 이러한 알림 기능 구현은 단순하지 않기 때문에 추가적인 기술적 도전 과제이다.

마지막으로 보안 문제도 중요하다. 사용자 개인 정보와 예약 정보 등 민감 정보를 안전하게 보호하는 기능이 필요하며, 권한 있는 사용자만 특정 작업(예: 예약 승인)을 수행할 수 있어야 한다.

이런 복잡한 요소들에 대응하기 위해서는 충분한 계획과 설계 단계가 필수적이다. 기술적 지식과 경험 외에도 심층적인 사용자 연구 및 테스트가 요구된다.

10 Project Plan

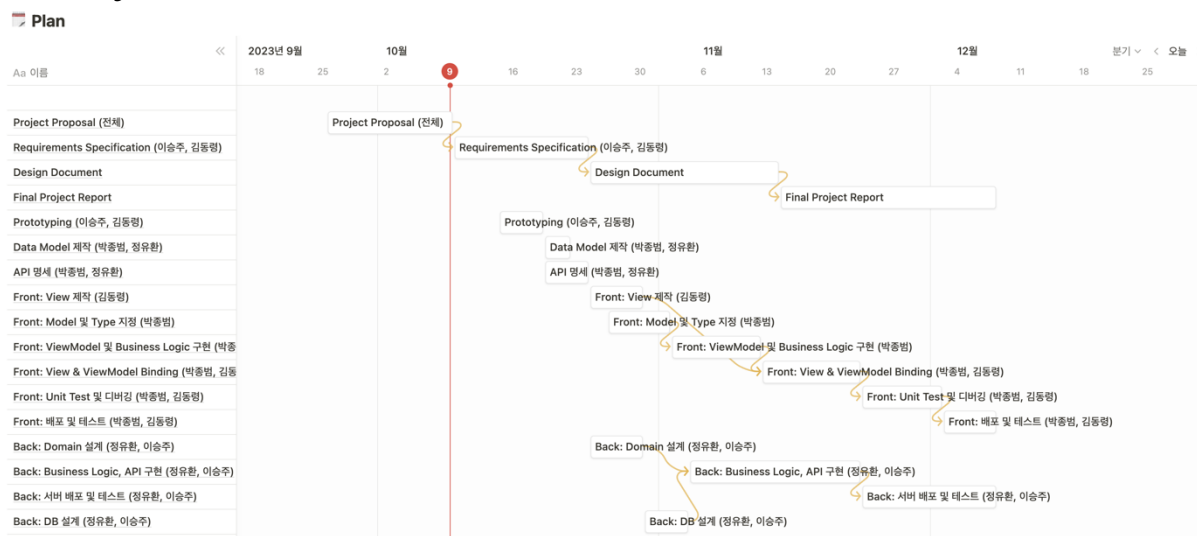


Figure 10-1 Project Plan Gantt Chart

11 References

- 조르테 간단등록군(<https://torokun.jorte.com/ko/index.html>)
- 스페이스 클라우드 (<https://www.spacecloud.kr>)
- 에브리 타임 (<https://everytime.kr/>)
- IOT SPACE - 회의실 예약 시스템(<https://iotspace.co.kr/meetingroombooking>)
- Slack (<https://mokeya.tistory.com/142>)
- 업무의 연속성 유지를 돕는 협업툴 '노션' - 동아일보
(<https://www.donga.com/news/It/article/all/20220513/113388460/1>)