

산학프로젝트 결과 보고서

학사 일정 기반 다이어리 앱 개발

지도교수 고 경 민

소프트웨어공학부 컴퓨터공학전공

2161047 박 예 원

2161010 이 소 현

2161024 임 세 빈

2161012 최 민 서

이 보고서를 산학프로젝트 결과 보고서로 제출함.

2024년 12월

차 례

I. 서 론

II. 과제 설계

1. 수행 과제 정보
2. 결과물 형태 및 결과물 명
3. 과제 수행 배경
4. 과제 목표 및 필요성
5. 팀 구성 설명

III. 과제 수행결과

1. 과제 내용
2. 과제수행 과정
3. 도출결과 내용
4. 기대 효과 및 활용 방안

V. 결론

1. 결론
2. 과제수행 후기

I. 서론

현대 사회에서 시간 관리는 개인의 효율적인 삶과 목표 달성을 위한 필수적인 요소로 자리 잡고 있다. 특히 대학생들은 학업, 대외 활동, 취업 준비, 자격증 공부 등 다양한 활동을 동시에 계획하고 실행해야 하며, 이를 체계적으로 관리하지 못하면 시간 낭비와 스트레스, 중요한 일정의 누락 같은 문제가 발생하기 쉽다. 특히 스스로 체계적인 시간 관리를 하지 못하는 학생들은 이러한 문제에 더욱 취약하다. 하지만 복잡한 일정을 효과적으로 관리하기란 쉽지 않으며, 이를 돕는 도구의 필요성이 점점 커지고 있다.

현재 시장에 나와 있는 일정 관리 애플리케이션은 사용자들의 다양한 요구를 충분히 반영하지 못하고 있으며, 학사 일정과 개인 일정을 통합적으로 관리하는 기능이 부족하다. 대다수의 애플리케이션은 학사 일정을 자동으로 연동하지 못하며, 이를 확인하더라도 개인 일정을 추가하거나 수정하는 데 한계가 있다. 학생들은 학사 일정을 학교 홈페이지에서 수동으로 확인한 후 별도의 일정 관리 애플리케이션에 다시 입력해야 하는 번거로운 과정을 거쳐야 한다. 특히 인턴십, 휴학, 아르바이트 등으로 학사 일정을 숙지하지 못한 학생들에게 이러한 접근성과 통합 관리의 한계는 더욱 크게 느껴진다.

대학생들이 학업과 개인 생활을 조화롭게 관리하도록 돕기 위해, 학사 일정과 개인 일정을 통합적으로 관리할 수 있는 애플리케이션이 요구된다. 본 프로젝트에서는 학교 홈페이지에서 제공하는 학사 일정을 자동으로 연동하고, 사용자가 개인 일정을 관리할 수 있는 다이어리 애플리케이션을 개발하고자 한다. 일정을 추가, 삭제, 수정, 검색하여 관리할 수 있으며, 캘린더 기반의 관리 기능과 알림 서비스, 사용자 맞춤형 설정(테마 변경, 알림 옵션 등)을 통해 사용자 편의성을 극대화하고자 한다.

이 애플리케이션은 학사 일정을 잘 알지 못하거나 바쁜 일상 속에서 효율적으로 시간을 관리하고자 하는 학생들에게 실질적인 도움을 제공할 것이다. 학사 일정 자동 연동 기능을 통해 학교 홈페이지 방문 없이도 중요한 일정을 확인할 수 있으며, 개인 일정과의 통합 관리를 통해 보다 체계적인 시간 관리가 가능하다. 이를 통해 학생들은 학업과 취업 준비, 개인 생활 사이의 균형을 유지하며 시간 관리 능력을 향상시킬 수 있다.

본 프로젝트는 단순한 일정 관리 도구를 넘어 대학생들의 실질적인 시간 관리 문제를 해결하고, 자기 주도적인 시간 관리 습관을 형성하며, 학업과 목표 달성을 돕는 도구로 자리 잡고자 한다. 이러한 애플리케이션을 통해 학생들의 일정 관리 어려움을 해소하고, 보다 효율적이고 통합적인 관리 환경을 제공하는 것을 목표로 한다.

II. 과제 개요

1. 수행 과제명

학사 일정 기반 다이어리 앱 개발

1-1. 수행 과제 정보

본 과제는 학사 일정을 기반으로 개인의 학업 및 생활 관리를 지원하는 다이어리 애플리케이션 개발을 목표로 한다. 본 애플리케이션은 학사 일정과 연동되며, 사용자 맞춤형 일정 관리, 알림 기능, 캘린더 관리 등 다양한 기능을 제공하여 사용자의 시간 관리와 학업 성취를 돕는 것을 목적으로 한다.

2. 결과물 형태 및 결과물명

결과물명 - Dear Diary (애플리케이션)

결과물의 형태 - 직접 앱을 배포할 수 없어 해당 보고서로 작성하여 제출

3. 과제 수행 배경

현대 사회에서 학생들은 학사 일정과 개인 일정을 동시에 관리해야 하는 환경에 놓여 있다. 그러나 기존의 다이어리 애플리케이션은 일반적인 일정 관리 기능만을 제공하며, 학사 일정과 연동되지 않아 사용자가 직접 정보를 입력해야 하는 불편함이 존재한다. 이는 학생들이 시간 관리에 추가적인 부담을 주며, 일정 누락과 비효율적인 시간 사용으로 이어질 수 있다.

특히, 학업 계획을 체계적으로 세우고 실천하려는 학생들에게 있어 일정 관리 도구는 필수적이다. 하지만 현재 사용 가능한 많은 도구들이 직관성이 부족하거나, 학업 중심의 일정 관리 기능이 미흡하여 충분한 도움을 주지 못한다. 이러한 문제를 해결하기 위해 학사 일정과 연동된 다이어리 애플리케이션의 개발하는 것을 과제로 선정하였다.

4. 과제 목표 및 필요성

본 과제의 주요 목표는 학사 일정 데이터를 통합하여 사용자가 학사 일정과 개인 일정을 효율적으로 확인하고 관리할 수 있는 다이어리 애플리케이션을 개발하는 것이다. 이를 통해 사용자가 별도의 입력 없이도 학사 일정을 손쉽게 확인하고, 개인 일정과 통합 관리할 수 있도록 지원한다. 또한, 중요한 일정을 자동으로 알림으로 제공하여 사용자들이 일정 누락을 방지하고 시간을 효율적으로 활용할 수 있도록 돕는다.

본 애플리케이션은 학업과 개인 생활 일정을 체계적으로 관리함으로써 학업 성취와 균형 잡힌 삶을 지원한다. 기존의 일정 관리 애플리케이션이 가진 한계를 극복하고, 사용자에게 더 높은 편리함과 시간 관리 효율성을 제공한다.

특히, 학사 일정과 개인 일정의 통합 관리는 사용자의 학습 몰입도를 높이고, 학업 성취도 향상뿐만 아니라 균형 잡힌 생활을 유지하는 데 기여할 것으로 기대된다.

본 과제는 학사 일정과 개인 일정을 통합하여 관리해 사용자의 시간 관리 효율성을 높이고, 학습 몰입도를 증가시켜 학업과 생활의 균형을 유지하도록 돕는 도구로서의 가치를 지닐 것이다.

5. 추진방법

본 과제는 IPP 현장실습 프로그램에 참여 중인 팀원들이 과반수를 차지하고 있어, 팀원 간의 역할을 분담하여 각 구성원이 프로젝트의 주요 기능 구현에 고르게 참여하도록 진행되었다. 매주 1회 온라인 회의를 통해 진행 상황을 점검하고 피드백을 공유하며, 필요시 추가 회의를 통해 문제를 해결하였다.

먼저, 요구분석을 통해 학사 일정 기반 다이어리 애플리케이션의 주요 요구사항을 정의하고, 필수 기능을 도출하였다. 요구분석 단계에서 도출된 내용을 기반으로 UI 설계서를 작성하였으며, 이 설계서를 바탕으로 실제 화면을 구성하며 개발을 진행하였다.

개발 과정에서는 프론트엔드와 백엔드로 나누지 않고, 각 기능별로 역할을 분담하여 개발을 진행하였다. 학사 일정 연동, 사용자 맞춤형 일정 관리, 알림 서비스, 캘린더 등 주요 기능을 구현하는 데 초점을 맞추었으며, 모든 화면과 기능이 사용자 친화적이고 직관적으로 동작할 수 있도록 설계와 구현 간의 일치성을 지속적으로 검토하고 개선하였다.

기능 구현 후, 테스트를 통해 시스템의 오류를 점검하고 수정하였다. 주요 기능이 예상대로 작동하는지 검증하였으며, 전체 시스템의 완성도를 높이기 위해 반복적인 피드백과 수정을 진행하였다.

최종적으로 개발 과정과 결과를 보고서로 작성하여 제출하였다. 보고서에는 요구분석, 설계, 개발, 테스트 과정과 함께 제안된 시스템의 효과와 개선 사항에 대한 평가가 포함되었으며, 이를 통해 프로젝트의 결과물과 그 실질적인 가치를 체계적으로 정리하였다.

6. 팀 구성(팀명) 설명

관심 분야와 만들고자 하는 프로젝트 주제가 같은 팀원들이 모여 만들어진 평범무난팀.

조원 및 역할

학과	이름	학번	역할
컴퓨터공학과	박예원	2161047	달력 선택, 알림, 지원 및 도움말
컴퓨터공학과	이소현	2161010	캘린더 관리, 일정 검색, 상하단 톨바
컴퓨터공학과	임세빈	2161024	일정 관리(전시/추가/수정/삭제/검색)
컴퓨터공학과	최민서	2161012	달력 UI, 학사일정 연동, 테마 설정

보고서는 각자 개발한 부분을 작성해오되, 최종 작성은 이소현이 담당함.

III. 과제 수행결과

1. 과제 내용

본 프로젝트는 학사 일정 기반 다이어리 애플리케이션을 개발하여 학생들의 학업 및 개인 일정을 체계적으로 관리할 수 있도록 지원하는 것을 목표로 한다. 이를 위해 다음과 같은 주요 내용을 포함한다.

- 1) 학사 일정 자동 연동
- 2) 달력 기능 제공 (연간, 월간, 주간, 일간)
- 3) 일정 관리 기능 제공 (추가, 편집, 삭제, 검색)
- 3) 캘린더 관리 기능 제공 (불러오기, 추가, 편집, 삭제)
- 4) 설정 기능 제공 (테마, 알림, 지원)

2. 과제수행 과정

1) 계획

팀 구성과 프로젝트 주제 선정을 통해 초기 단계에서 프로젝트의 방향성을 설정하였다.

팀장과 팀원을 선정한 뒤, 각자의 아이디어를 공유하며 프로젝트 방향을 논의하였다. 학사 일정 관리와 개인 일정 통합의 필요성이 제기되었으며, 이를 기반으로 프로젝트 주제를 확정하였다. 이후 프로젝트의 방향성과 주요기능, 필요성에 대해 논의하며 Android Studio, Firebase등의 기술 스택을 조사하여 프로젝트 구현에 필요한 기술 요구 사항을 정의하였다. 기능 요구사항과 설계 방향을 논의하고, 주요 기능을 구체화하여 프로젝트 전반의 요구사항과 설계 방안을 수립하였다. 이러한 과정을 통해 세부 계획서를 작성하고, 팀원 간 역할을 명확히 분담하여 각자 맡은 부분을 중심으로 프로젝트를 준비하였다.

2) 요구사항 정의 및 설계

앱의 주요 UI/UX 설계를 진행하였다. 주요 화면 배치와 사용자 경험을 고려하여 화면 설계서를 작성하였으며, 학사 일정 데이터와 사용자 일정 데이터를 효과적으로 관리하기 위해 자료 조사와 데이터 흐름을 설계하였다. 이 과정에서 프로젝트의 주요 화면과 기능 요소를 명확히 규정하고, 구체적인 요구사항을 정의한 화면 설계서가 작성되었다.

Page Title	학사 일정 기반 다이어리 앱 개발	Screen ID	UI-DIARY-001	Date	2024.10.09
Screen Path	Home	Team Name	궁법무난	Author	이소현

Description
<p>프로그램 시작 시 연간 달력 표시.</p> <p>각 월을 클릭하면 해당 월의 월간 달력으로 전환됨.</p> <p>오늘 날짜는 연결된 월간 달력에서 강조된 색상으로 표시.</p> <p>오늘 날짜로 돌아가는 버튼 추가. 월간 달력의 오늘 날짜로 이동.</p> <p>이벤트 확인 버튼 추가. 현재 날짜의 이벤트 확인 및 추가 가능. 선택한 날짜, 주간/월간 일정을 확인하는 기능도 제공.</p> <p>달력 선택 버튼 추가. 연간 달력에서 월간/주간/월간 달력으로 이동 가능.</p> <p>검색 버튼 추가. 일정 검색 팝업 창이 열리며, 제목, 날짜, 태그 등을 기준으로 일정 검색 가능.</p> <p>캘린더 버튼 추가. 모든 캘린더 목록을 확인 가능. 새로운 캘린더 추가, 기존 캘린더 수정 및 삭제 가능.</p> <p>설정 버튼 추가. 앱의 전반적인 설정을 관리. 테마 변경, 알림 설정, 이벤트 편집, 캘린더 설정, 학교 일정 불러오기, 도움말 등 설정 가능.</p> <p>학교 학사 일정을 불러오는 기능 추가. 학사 일정은 연간 달력에서 특정 색상으로 표시되며, 개인 일정과 통합하여 볼 수 있음.</p> <p>스크롤을 통해 달력 확대/축소 및 이동. 연간, 월간, 주간 달력 뷰를 전환 가능.</p>
Check Point
<p>월간 달력과의 연동 및 데이터 정확성이 확보되어야 함.</p> <p>월간 달력에 표시된 이벤트는 연간 달력에 표시 되지 않음.</p> <p>현재 위치(연간/월간/주간/일간)를 추적할 수 있어야 함.</p> <p>학교 학사 일정 API 또는 데이터 소스의 연동이 필요함.</p>

그림 1 메인 화면(연간달력) UI 설계서

Page Title	학사 일정 기반 다이어리 앱 개발	Screen ID	UI-DIARY-002	Date	2024.10.09
Screen Path	Home > 월간 달력	Team Name	궁법무난	Author	이소현

Description
<p>연간 달력에서 특정 월을 클릭하면 해당 월의 월간 달력으로 이동.</p> <p>월간 달력에서 해당 월의 일별로 추가된 이벤트를 확인 가능. 클릭 시 세부 이벤트 정보가 하단에 표시.</p> <p>오늘 날짜는 강조된 색상으로 달력에 표시.</p> <p>달력에서 날짜를 선택하면 선택한 날짜를 강조된 색상으로 달력에 표시.</p> <p>오늘 날짜로 돌아가는 버튼 추가.</p> <p>날짜를 더듬 클릭하면 클릭한 날짜에 새로운 이벤트 추가 가능. 팝업창 생성.</p> <p>날짜를 길게 클릭하면 클릭한 날짜의 일간 달력으로 이동.</p> <p>날짜 선택 후 하단의 이벤트 버튼을 클릭시 선택한 날짜의 이벤트 확인 가능. 이벤트 수정 및 삭제 가능.</p> <p>달력 선택 버튼으로 연간/주간/월간 달력으로 이동 가능.</p> <p>검색, 캘린더, 설정 버튼 사용 가능.</p>
Check Point
<p>연간 달력, 월간 달력과의 연동 및 데이터 정확성이 확보되어야 함.</p> <p>현재 위치(연간/월간/주간/일간)를 추적할 수 있어야 함.</p> <p>학교 학사 일정 API 또는 데이터 소스의 연동이 필요함.</p>

그림 2 월간 달력 UI 설계서




그림 3 주간 달력 UI 설계서



그림 4 일간 달력 UI 설계서

Page Title	학사 일정 기반 다이어리 앱 개발	Screen ID	UI-DIARY-005	Date	2024.10.09
Screen Path	이벤트 추가 팝업	Team Name	꿈발무난	Author	이소현



The image shows a mobile app screen for adding a new event. It has a purple header with the text '다이어리'. Below the header, there's a section titled '새로운 이벤트' (New Event) with a '제목' (Title) input field. Underneath, there's a list of options: '시작 일자 > 종료 일자' (Start date > End date), '반복 없음' (No repeat), '캘린더' (Calendar), '알림' (Notification), '위치' (Location), 'URL', and '메모' (Memo). At the bottom right, there's a '저장' (Save) button.

Description
<p>새로운 이벤트 추가 팝업.</p> <p>제목과 시작/종료 일시는 필수 입력 항목으로 설정.</p> <p>캘린더 선택기를 통해 시작 및 종료 날짜 설정 가능.</p> <p>매일, 매주, 매월 등 일정 반복 옵션 선택 가능.</p> <p>일정 알림 시간 설정 가능.</p> <p>이벤트에 대한 위치 및 관련 URL 입력 가능.</p> <p>저장 버튼 클릭 시 이벤트가 캘린더에 반영됨.</p>
Check Point
<p>현재 위치(연간/월간/주간/일간)를 추적할 수 있어야 함.</p> <p>학교 학사 일정 API 또는 데이터 소스와의 연동이 필요함.</p>

그림 5 일정 페이지 UI 설계서

Page Title	학사 일정 기반 다이어리 앱 개발	Screen ID	UI-DIARY-006	Date	2024.10.09
Screen Path	Home > 캘린더	Team Name	꿈발무난	Author	이소현



The image shows a mobile app screen for the calendar. It has a purple header with the text '다이어리'. Below the header, there's a section titled '캘린더' (Calendar) with three input fields: '캘린더' (Calendar), '학교' (School), and '대한민국 공휴일' (Korea's Public Holidays). At the bottom right, there's a button labeled '→ 캘린더 추가' (Add Calendar).

Description
<p>캘린더 관리 페이지.</p> <p>하단의 캘린더 버튼을 누르면 이동하는 페이지.</p> <p>사용자가 관리 중인 캘린더 목록 확인 가능.</p> <p>이름과 색상을 지정하여 새로운 캘린더 추가 가능.</p> <p>기존 캘린더 수정 및 삭제 가능.</p> <p>캘린더 불러오기 가능.</p> <p>학교 학사 일정 불러오기 가능. 학교 웹사이트 기타 외부 소스에서 학사 일정을 불러오는 기능. 학사 일정은 사용자 일정과 함께 연간 달력에서 통합되어 표시.</p>
Check Point
<p>현재 위치(연간/월간/주간/일간)를 추적할 수 있어야 함.</p> <p>학교 학사 일정 API 또는 데이터 소스와의 연동이 필요함.</p> <p>외부에서 캘린더 불러오기가 가능해야 함.</p> <p>캘린더 추가 시 색상과 제목을 필수로 입력해야 함.</p>

그림 6 캘린더 페이지 UI 설계서

Page Title	학사 일정 기반 다이어리 앱 개발	Screen ID	UI-DIARY-007	Date	2024.10.09
Screen Path	Home > 설정	Team Name	꿈빛무늬	Author	이소현



Description
<p>설정 관리 페이지.</p> <p>하단의 설정 버튼을 누르면 이동하는 페이지.</p> <p>앱의 테마를 사용자가 정의할 수 있는 옵션 제공.</p> <p>이벤트 알림 시간을 설정할 수 있는 기능 제공.</p> <p>기본 이벤트 설정(기본 알림 시간, 기본 캘린더 등)을 설정할 수 있는 기능 제공.</p> <p>달력 관련된 기능을 설정 가능.</p> <p>메인 화면 달력을 연간/월간/주간 중 선택하여 설정할 수 있는 기능 제공.</p> <p>주간/월간 달력에서 원하는 달력을 선택하여 볼 수 있는 기능 제공.</p>
Check Point
<p>앱의 테마를 설정하기 위한 기능을 구현해야 함.</p> <p>알림을 받을 수 있는 환경이어야 함.</p> <p>주간/월간 달력 페이지가 2개 이상 구현 되어 있을 경우 달력 편집 설정에 포함 시킴.</p> <p>학교 학사 일정 API 또는 데이터 소스와의 연동이 필요함.</p> <p>외부에서 캘린더 불러오기가 가능해야 함.</p> <p>각 학교마다 학사일정에서 지원하는 부분이 학교에서 없을 경우 학사 일정을 캘린더에 불러올 수 있는 시스템 분리.</p> <p>학사 일정 파일 구조 완성됨.</p> <p>구독 추가할 수 있는 시스템이 필요.</p> <p>불러오는 것까지는 가능.</p>

그림 7 설정 페이지 UI 설계서

위와 같이 UI 설계서를 작성하고 UI 설계서를 기반으로 개발을 수행하였다.

3) 개발 환경 구축

- 안드로이드 스튜디오(Android Studio) 설치
- SQLite 안드로이드 스튜디오 내부에서 사용

4) 기능 개발

(1) 연간, 월간, 주간, 일간 달력 기능

본 프로젝트에서는 연간, 월간, 주간, 일간 달력을 통합적으로 제공하여 사용자가 직관적으로 일정 관리와 조회를 수행할 수 있도록 설계되었다. 주요 목표는 달력 간 데이터 연속성을 유지하며, 선택된 날짜와 관련된 이벤트 데이터를 실시간으로 동기화하여 사용자에게 일관되게 제공하는 것이다. 이를 위해 CalendarView, RecyclerView, SharedPreferences, Intent, Calendar 객체를 활용하여 날짜 및 이벤트 데이터를 저장, 전달, 동기화하는 구조를 설계하였다.

① 연간 달력 기능



그림 8 메인 화면 (연간 달력)

사용자가 연간 달력에서 각 월의 주요 이벤트 데이터를 표시하고, 특정 월 선택 시 해당 월로 이동한다.

연간 달력은 애플리케이션의 메인 화면으로, 한 해의 전체 날짜를 한눈에 볼 수 있으며 사용자 선택이 가능한 UI를 제공한다. Calendar 객체를 활용하여 현재 연도를 기준으로 각 월의 첫날과 마지막 날을 동적으로 계산하였으며, 이를 기반으로 각 월의 주요 이벤트 데이터를 표시하였다. 사용자가 특정 월을 클릭하면 MonthlyCalendarActivity로 전환되어 해당 월의 세부적인 이벤트 데이터를 확인할 수 있으며, 달력 간의 데이터 연속성을 유지하기 위해 선택된 월의 데이터를 Intent를 통해 전달하였다.

② 월간 달력 기능



그림 9 월간 달력 화면

사용자가 월간 달력에서 날짜를 선택하고 이벤트 데이터를 확인할 수 있는 인터페이스를 제공한다.

월간 달력에서는 사용자가 특정 날짜를 선택하여 이벤트 데이터를 조회할 수 있는 인터페이스를 제공한다. CalendarView를 통해 선택된 날짜(연도, 월, 일)를 기반으로 SharedPreferences에서 저장된 이벤트 데이터를 조회하도록 구현하였다. 해당 데이터는 TextView를 통해 화면에 표시되며, 조회된 데이터가 없을 경우 기본 메시지인 *‘이벤트 없음’*을 출력하여 사용자 경험을 개선하였다.

이 과정에서 SharedPreferences는 키-값 구조를 활용하여 특정 날짜를 키로, 이벤트 내용을 값으로 저장 및 조회할 수 있도록 설계되었다. 기본적으로 오늘 날짜가 초기화되며, 선택한 날짜는 “YYYYMMDD” 형식의 문자열 키로 변환되어 이벤트 조회에 활용되었다. 또한, onResume() 메서드를 활용하여 데이터 변경 시 인터페이스에 즉시 반영되도록 설계하여 동적인 데이터 업데이트가 가능하게 하였다.

③ 주간 달력 기능



그림 10 주간 달력 화면

월간 달력에서 선택된 날짜를 기반으로 주간 범위를 계산하여 해당 범위 내 이벤트 데이터를 표시한다.

월간 달력에서 사용자가 날짜를 선택한 후 주간 보기 버튼을 클릭하면, 선택된 날짜 데이터가 Intent를 통해 WeeklyCalendarActivity로 전달된다. WeeklyCalendarActivity에서는 전달받은 날짜 데이터를 활용하여 해당 날짜가 포함된 주간의 시작일(일요일)과 종료일(토요일)을 동적으로 계산한다. 이를 위해 Calendar객체를 활용하였으며, 계산된 범위는 “YYYY-MM-DD ~ YYYY-MM-DD”형식으로 변환되어 화면에 표시되었다.

계산된 범위는 “YYYY-MM-DD ~ YYYY-MM-DD”형식으로 변환되어 화면에 표시되며, 주간 범위 내의 모든 날짜에 대해 SharedPreferences에 저장된 이벤트 데이터를 조회하여 리스트 형태로 출력하도록 구현하였다. 이 과정에서 월간 달력에서 추가된 이벤트 데이터가 주간 달력에서도 동일하게 반영되도록 SharedPreferences와 Intent를 활용한 데이터 연동 구조를 설계하여 두 화면 간의 데이터 일관성을 유지하였다.

④ 일간 달력 기능

일간 달력은 특정 날짜의 상세 정보를 시간 순서대로 조회할 수 있는 인터페이스를 제공한다. 사용자가 일간 보기로 전환하면 해당 날짜의 이벤트 데이터가 RecyclerView를 통해 시간 순서대로 정렬되어 화면에 표시된다. 사용자는 이벤트를 추가하거나 편집, 삭제할 수 있는 인터페이스를 제공받는다. 이벤트 데이터는 SharedPreferences에 저장되며, 데이터 변경 사항은 즉시 반영되어 다른 화면에서도 동기화가 유지되도록 설계하였다.

⑤ 데이터 연동 및 동기화

달력 간 데이터 연동은 사용자가 SharedPreferences와 Intent를 중심으로 설계되었으며, 사용자가 한 화면에서 저장한 데이터를 다른 화면에서도 실시간으로 확인할 수 있도록 구현되었다. 이 과정에서 발생한 날짜 데이터의 정확성 문제, 이벤트 데이터 누락 문제, 데이터 연속성 문제는 Calendar 객체의 날짜 계산 로직 및 SharedPreferences 데이터 로드 로직을 최적화함으로써 해결하였다. 또한, Intent로 전달되는 데이터의 오류를 방지하기 위해 디버깅 로그 출력을 통해 문제를 분석하고 수정하였다.

(2) 테마 설정

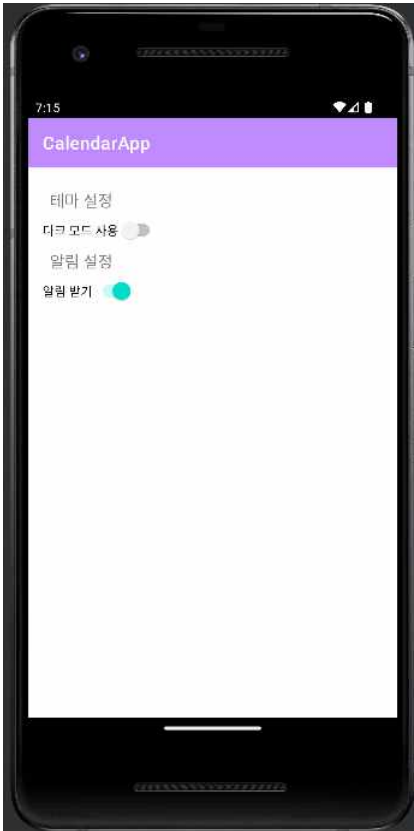


그림 11 테마 설정

설정 화면에서 애플리케이션의 테마를 사용자가 설정을 할 수 있는 옵션을 제공한다.

사용자 맞춤형 경험을 제공하기 위해 앱의 테마 설정 기능을 추가로 구현하였다.

- 색상 테마 설정 : 원하는 색상 테마를 선택할 수 있도록 UI를 설계하였으며, SharedPreferences를 활용하여 설정된 테마가 앱 재실행 후에도 유지되도록 하였다.
- 다크 모드 지원 : 다크 모드를 포함한 다양한 테마 옵션을 제공하여 사용자가 자신의 환경에 맞는 설정을 선택할 수 있도록 하였다. 테마 변경 시 UI가 즉시 반영되도록 설계하였으며, 앱 전반의 시각적 일관성을 유지하였다.

(3) 학사 일정 연동 (ICS 구독)



그림 12 학사일정 구독 기능

캘린더 화면에서 학사 일정 ICS 파일을 구독하여 달력에 학사 일정을 표시하는 기능을 제공한다.

학사일정을 달력에 연동하여 효과적으로 관리하기 위해 ICS 파일 구독 기능을 추가하였다.

- ICS 파일 파싱 및 동기화 : 학사일정 데이터는 ICS 파일 형식으로 제공되며, 앱에서 해당 파일을 구독하여 일정 데이터를 동기화할 수 있다. ICS 파일을 파싱하여 일정 데이터를 읽어들이는 기능을 구현하였고, 각 일정은 날짜와 이벤트 세부 정보를 포함하여 앱의 SharedPreferences에 저장되었다.
- 일정 데이터 통합 : 저장된 학사일정 데이터는 연간, 월간, 주간, 일간 달력의 모든 뷰에서 확인할 수 있으며, 사용자가 각 뷰에서 학사 일정을 손쉽게 확인할 수 있도록 설계하였다.
- 자동 업데이트 : ICS 구독 데이터는 일정 업데이트 시 자동으로 동기화되며, 최신 학사 일정을 실시간으로 제공한다.

❖ 수행 결과

본 기능의 설계 및 구현 결과, 연간, 월간, 주간, 일간 달력 간 데이터 연동을 성공적으로 구현하였으며, 사용자에게 직관적이고 일관된 인터페이스를 제공하였다. 사용자는 학사 일정과 개인 일정을 통합적으로 관리하며, 특정 날짜나 주간에 대한 상세 정보를 손쉽게 조회하고 효율적으로 일정을 관리할 수 있게 되었다. 또한, 테마 설정과 ICS 구독 기능을 통해 사용자 편의성을 높이고, 앱의 활용성을 극대화하였다.

(4) 상하단 툴바 기능

상단 툴바(Toolbar)와 하단 네비게이션(BottomNavigationView)을 사용하여 네비게이션 기능을 설계하고 구현하였다. 이를 통해 사용자는 직관적이고 효율적인 방식으로 애플리케이션 내 주요 화면과 기능에 접근할 수 있다.



그림 13 상하단 툴바 기능 추가

메인화면으로, 상하단의 버튼(아이콘)을 클릭하여 해당하는 기능으로 이동할 수 있다.

① 상단 툴바

Toolbar는 상단에 위치하며, 사용자가 자주 사용하는 액션 아이콘과 메뉴를 포함하고 있다. Toolbar는 일정을 검색하는 검색 버튼(아이콘)과 캘린더를 현재 날짜로 이동하는 오늘로 이동 버튼(아이콘)으로 구성되어있다. 메뉴 항목을 menu 리소스 파일을 통해 정의하고, XML 파일을 사용하여 각 메뉴 항목의 아이콘과 이벤트를 지정하였다.

setSupportActionBar() 메서드를 활용하여 현재 액티비티에 연결하였으며, onCreateOptionsMenu()를 사용하여 Toolbar에 메뉴를 inflate하였다.

사용자가 메뉴 항목을 클릭할 경우 onOptionsItemSelected()메서드를 활용하여 해당 Activity로 이동하는 액션이 구현되었다.

② 하단 툴바

BottomNavigationView는 네 가지 주요 탭을 제공하도록 설계되었다. 캘린더 메인화면으로 이동하는 버튼(아이콘)과 이벤트 목록으로 이동하는 이벤트 버튼(아이콘), 달력 선택이 가능한 버튼(아이콘), 설정 화

면으로 이동하는 설정 버튼(아이콘)으로 하단 네비게이션을 구성하였다. 메뉴 항목은 menu 리소스 파일을 통해 정의하고, 각 항목에 아이콘과 텍스트를 지정하여 직관적인 인터페이스를 구현하였다.

NavController와 `setOnItemSelectedListener()` 메서드를 사용하여 각 탭 클릭 시 Fragment 또는 Activity 화면으로 전환되도록 동작을 처리하였다.

④ Dependency 추가 및 라이브러리 사용

본 설계에서는 Material Design Components 의존성을 추가하였으며, navigation 사용을 위한 의존성 또한 추가하여 사용하였다. 코드 내에서 `com.google.android.materialandroidx.navigation` 라이브러리를 사용하였다. 사용자의 직관적인 네비게이션을 돕는 동시에, 상하단의 네비게이션 구조를 통해 화면의 주요 기능에 빠르게 접근할 수 있도록 하였다.

```
dependencies {  
    implementation 'androidx.navigation:navigation-fragment:2.7.5'  
    implementation 'androidx.navigation:navigation-ui:2.7.5'  
    implementation 'com.google.android.material:material:1.9.0'  
}
```

위와 같이 의존성을 추가하여 사용하였으며, 버전에 따라 오류가 발생할 수 있다. Android Studio 버전에 따라 ActionBar 오류가 발생할 수 있으며, Manifest에서 NoActionBar 처리 후 Style에서 새 정의가 필요하다.

❖ 수행 결과

본 상하단 툴바 기능은 다이어리 애플리케이션을 효율적이고 편리하게 사용할 수 있도록 도와준다. 설계된 Toolbar와 BottomNavigationView는 사용자의 직관적이고 신속한 네비게이션을 지원하며, 애플리케이션의 전반적인 품질과 사용성을 향상시킬 수 있다.

(5) 캘린더 관리 기능

본 프로젝트에서는 캘린더 관리 기능을 Fragment와 Navigation Component를 활용하여 구현하였다. 사용자가 관리 중인 캘린더 목록을 확인하고, 항목 추가, 수정, 삭제와 같은 관리 작업을 효율적으로 수행할 수 있도록 설계되었다. 데이터의 저장 및 동기화를 위해 SQLite 데이터베이스를 활용하였으며, 화면 간의 전환은 Navigation Component를 통해 구현하였다.

① 캘린더 메인 화면

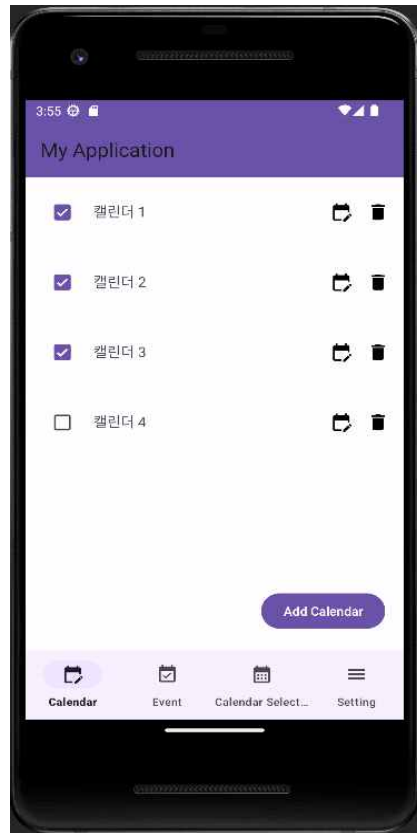
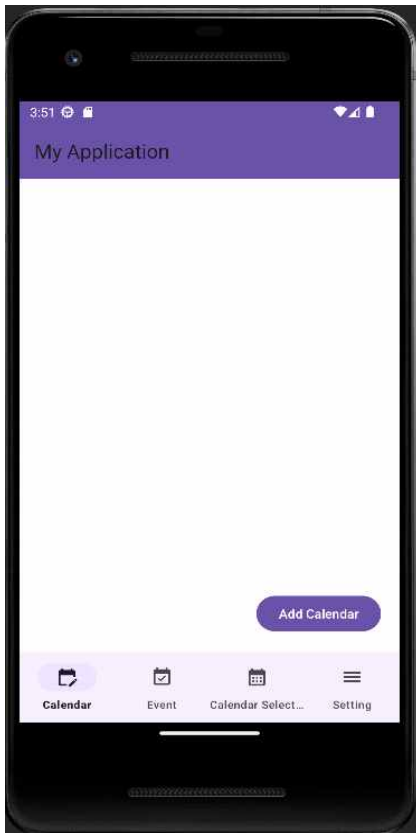


그림 14 캘린더 메인 화면

그림 15 캘린더 추가된 목록

캘린더 메인 화면에서 사용자가 관리 중인 캘린더 목록을 확인할 수 있도록 표시한다.

캘린더 메인 화면에서는 사용자가 관리 중인 캘린더 목록을 확인할 수 있도록 구성하였다. 캘린더 목록은 SQLite 데이터베이스에서 캘린더 데이터를 조회하고 RecyclerView를 활용하여 캘린더 목록을 표시하였다. 각 항목은 캘린더 이름, 카테고리(계정) 정보를 포함하고 수정 아이콘, 삭제 아이콘, 체크박스 아이콘을 함께 표시한다. 각 항목은 추가된 순서대로 정렬되며, 사용자가 각 아이콘을 통해 조작할 수 있다.

캘린더 메인 화면의 하단에서는 새로운 캘린더를 추가하기 위한 버튼을 배치하여 사용자가 새로운 캘린더를 생성할 수 있도록 하였다. 사용자가 추가 버튼을 클릭할 경우 Navigation Component를 활용하여 캘린더 추가를 위한 Fragment로 전환된다.

캘린더 메인 화면의 레이아웃 구성에서는 FragmentContainerView를 활용하여 네비게이션 전환을 설정하였다. navGraph를 정의하여 목록을 표시하는 Fragment를 초기 진입 화면으로 설정하였으며, 각 기능별 화면 전환을 효율적으로 관리하였다. 이러한 구성은 화면 전환 시 navigation에 설정된 경로로 fragment가 직관적으로 처리될 수 있도록 설계되었다.

② 캘린더 추가 기능

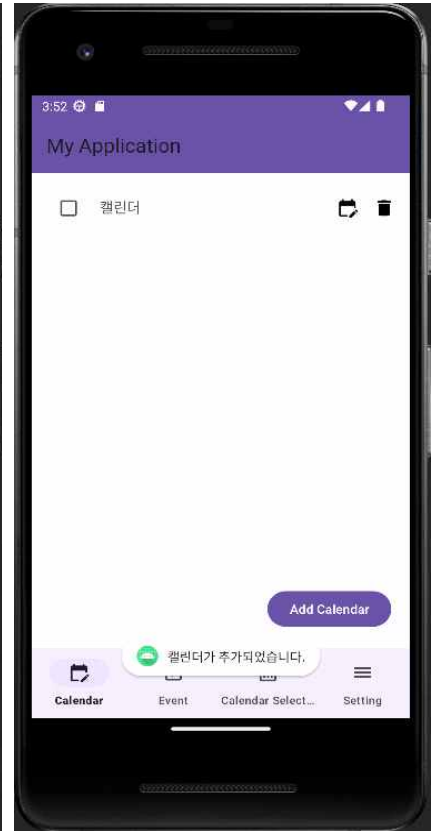
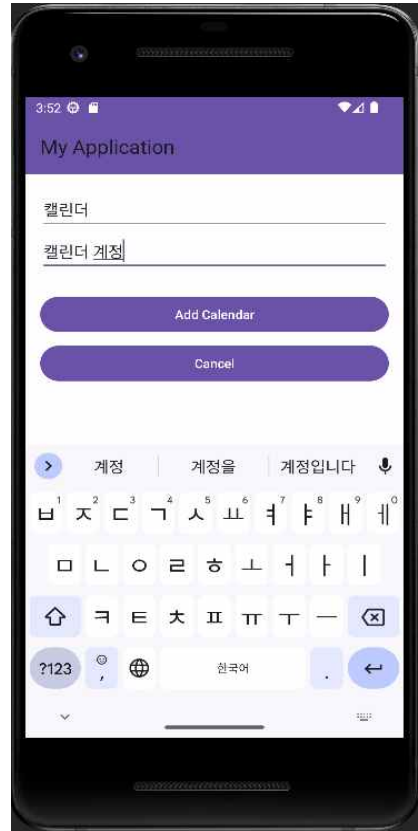
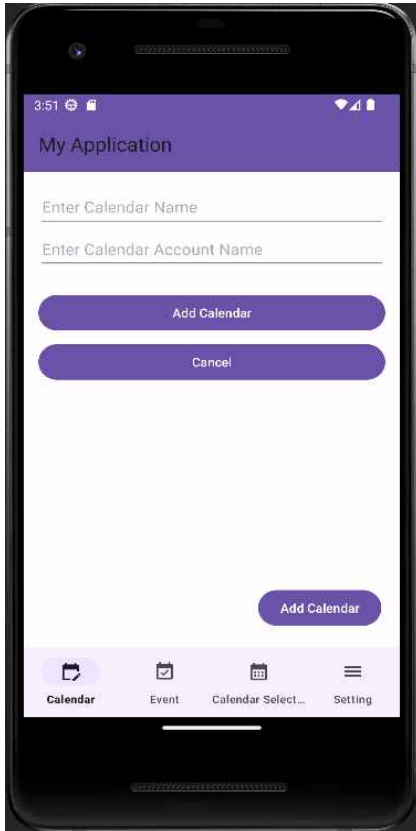


그림 16 캘린더 추가 Fragment 그림 17 캘린더 입력 그림 18 캘린더 추가 완료
캘린더 추가를 위한 Fragment를 사용하여 캘린더를 입력하고 캘린더 메인화면에 추가가 가능하다.

캘린더 추가 기능은 사용자가 새로운 캘린더를 생성할 수 있도록 설계되었다. 캘린더 추가는 별도의 캘린더 추가를 위한 Fragment를 통해 이루어지며, 해당 화면에서는 사용자가 캘린더 이름과 카테고리(계정) 정보를 입력할 수 있는 EditText 필드와 추가/취소 버튼으로 UI로 구성되어 있다.

Fragment에서 사용자가 정보를 입력한 후 추가 버튼을 누르면 SQLite 데이터베이스에 새로운 캘린더가 저장된다. 이후 Navigation Component를 통해 캘린더 메인 화면으로 돌아오며, RecyclerView가 즉시 갱신되어 추가된 항목이 목록에 표시된다.

반면, 취소 버튼을 클릭하면 데이터베이스에 변경 없이 Navigation Component를 통해 캘린더 메인 화면으로 복귀하도록 설계하였다.

③ 캘린더 수정 기능

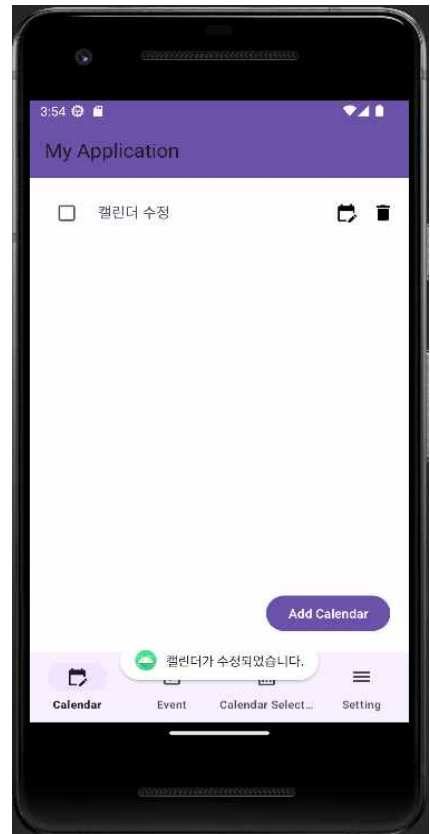
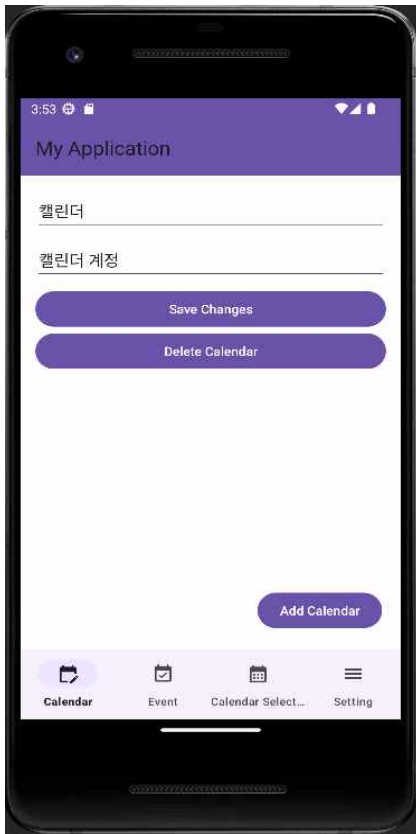


그림 19 기존 캘린더 불러오기

그림 20 캘린더 수정

그림 21 캘린더 수정 완료

캘린더 메인화면에서 수정 아이콘 클릭 시 기존 캘린더 데이터를 불러와 편집할 수 있으며, 수정된 데이터가 목록에 반영된다.

캘린더 수정 기능은 기존 캘린더 항목을 편집할 수 있도록 설계되었다. 캘린더 수정은 메인 화면의 목록에서 각 항목에 표시된 수정 아이콘을 클릭하여 진행된다. 별도의 캘린더 수정을 위한 Fragment로 이동하며, 수정 화면에서는 데이터베이스에서 기존 데이터를 조회하여 사용자에게 표시하고 편집할 수 있도록 설계하였다.

Fragment에서는 사용자가 정보를 수정한 후 수정 버튼을 누르면 SQLite 데이터베이스에 변경사항이 저장된다. Navigation Component를 통해 캘린더 메인 화면으로 돌아오며, RecyclerView가 즉시 갱신되어 수정한 데이터가 목록에 반영되어 표시된다.

삭제 버튼도 함께 제공이 되며 삭제 버튼 클릭 시 해당 항목은 SQLite 데이터베이스에서 항목을 삭제된다. Navigation Component를 통해 캘린더 메인 화면으로 복귀하며, RecyclerView가 갱신되어 삭제한 항목이 목록에 보이지 않게 된다.

반면, 취소 버튼을 클릭하면 데이터베이스에 변경 없이 Navigation Component를 통해 캘린더 메인 화면으로 복귀하도록 설계하였다.

수정 화면은 사용자가 수정하지 않고 복귀하거나, 잘못된 데이터를 입력할 경우를 고려하여 데이터의 유효성을 검증하고 예외 상황을 처리할 수 있는 구조로 설계되었다.

④ 캘린더 삭제 기능

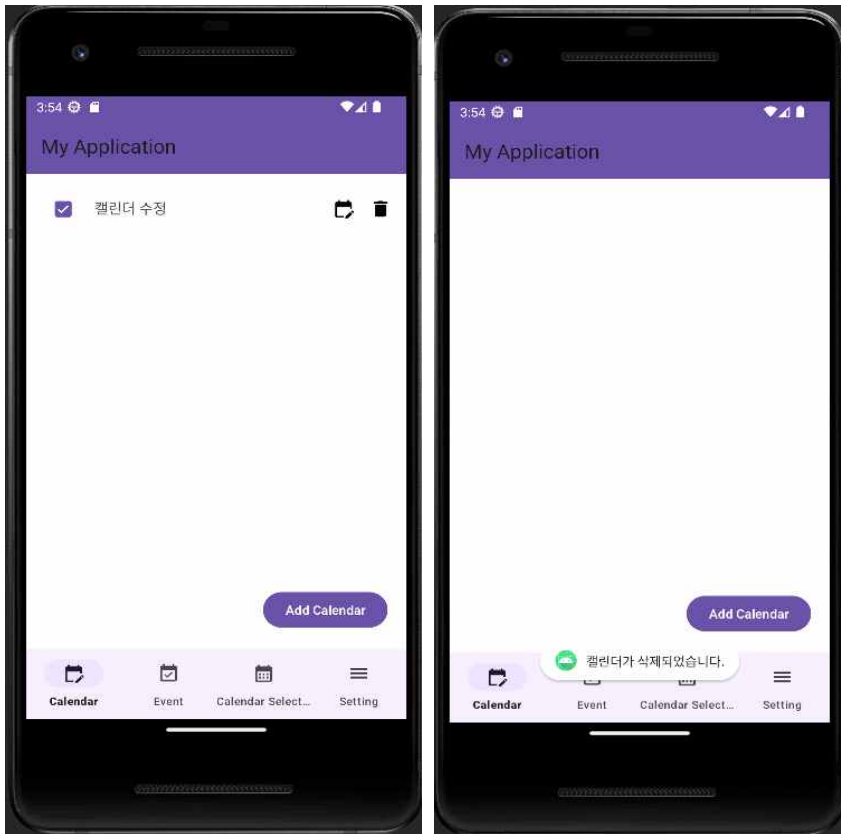


그림 22 체크박스 사용 가능

그림 23 목록의 캘린더 삭제

캘린더 메인화면에서 삭제 아이콘을 사용하거나, 수정 화면에서 삭제 버튼을 활용하여 캘린더 삭제가 가능하다.

캘린더 삭제 기능은 두 가지 방식으로 제공된다.

첫 번째로, 캘린더 메인 화면에서 각 항목의 삭제 아이콘을 클릭하여 삭제할 수 있으며, 이 경우에 SQLite 데이터베이스에 항목이 삭제되고, RecyclerView에 반영되어 삭제된 항목이 더이상 표시되지 않는다.

두 번째로는 수정 화면에서 삭제 버튼을 통해 삭제하는 방식으로, 삭제 시 동일한 데이터베이스 제거 및 목록 갱신 절차를 따른다.

⑤ 설계 및 구현 방식

캘린더 관리 기능은 Fragment를 활용하여 캘린더 추가, 수정, 삭제 기능을 각각의 화면으로 분리하였으며, Navigation Component를 통해 화면 간 전환과 데이터 흐름을 관리하였다.

데이터베이스는 SQLite와 DatabaseHepler를 사용하여 데이터 저장 및 관리를 수행하였으며, 데이터 조작은 Adapter를 활용하여 구현하였다. 이전 화면으로 돌아올 때 데이터를 동기화하고 데이터 변경 사항을 즉시 반영하여, 데이터 일관성을 유지하면서도 실시간 갱신이 가능하도록 설계되었다.

목록 화면은 RecyclerView를 활용하여 사용자가 데이터를 직관적으로 확인하고 조작할 수 있는 인터페

이스를 제공하였다. 또한, 각 Fragment는 재사용성을 고려하여 설계되었으며, 프로젝트의 확장성을 높이는 데 기여하였다.

❖ 수행 결과

본 캘린더 관리 기능은 사용자가 캘린더를 효율적으로 관리할 수 있도록 직관적인 사용자 경험을 제공한다. Fragment 기반의 구조와 Navigation Component를 활용하여 화면 전환의 효율성을 높였으며, SQLite 데이터베이스와의 동기화는 데이터 관리의 신뢰성을 보장하였다. 이를 통해 향후 확장성을 고려한 설계로 추가 기능 구현 시에도 용이하며, 사용자 경험이 최적화된 캘린더 관리 시스템을 제공할 수 있다.

(6) 일정 관리

① 일정 검색

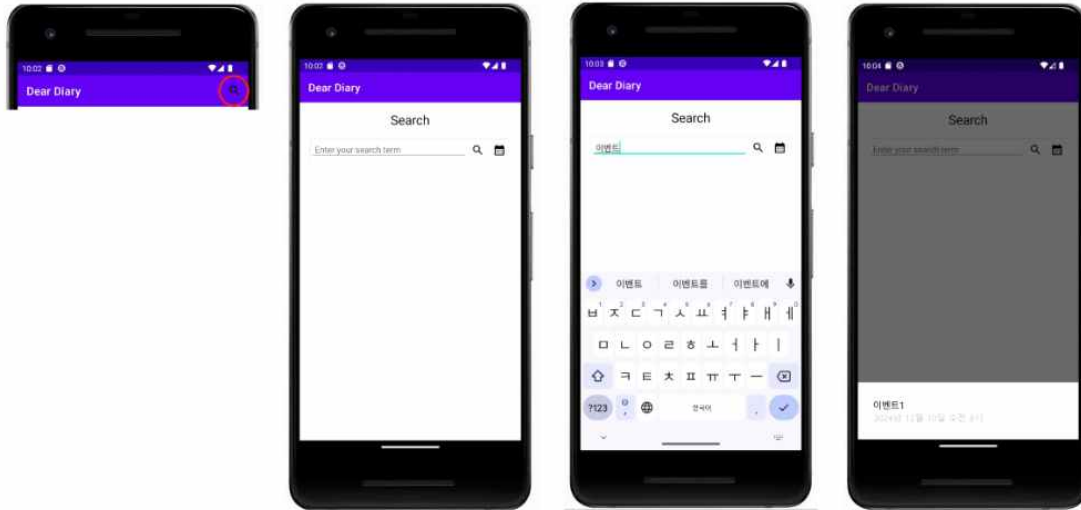


그림 24 일정 검색 기능

상단 Toolbar의 검색 아이콘을 클릭하면 추가된 일정을 검색하여 확인할 수 있다.

일정 검색 기능은 상단 Toolbar의 검색 아이콘 클릭 시 별도의 검색 화면으로 이동하여 수행할 수 있도록 설계되었다. 검색 화면은 텍스트 입력 상자, 검색 버튼, 캘린더 버튼으로 구성되며, 이를 통해 사용자는 검색 키워드를 입력하여 검색을 수행할 수 있다.

① 검색 화면 구성

검색 입력 부분은 사용자가 검색어를 입력할 수 있는 공간으로, 일정 제목이나 일부 키워드를 입력하면 검색이 가능하다. 텍스트박스과 검색 버튼(아이콘), 캘린더 버튼(아이콘)으로 이루어져 있으며, 사용자의 가독성과 입력 편의를 위해 한 줄로 배치하였으며, 이 구성을 위해 LinearLayout을 사용하여 텍스트박스와 버튼을 감싸도록 설계하였다.

② 키워드 기반 검색

사용자가 텍스트 상자에 키워드를 입력한 후 검색 버튼을 클릭하면, SQLite 데이터베이스의 일정 테이블에서 입력된 키워드를 포함하는 데이터를 조회한다. 이때, SQL의 LIKE연산자를 활용하여 검색어가 포함된 모든 일정 데이터를 검색할 수 있도록 구현되었습니다. 예를 들어, '회의'를 입력하면 '팀 회의', '회의 준비'와 같이 '회의'를 포함하는 일정이 모두 결과로 반환됩니다.

③ 캘린더 기반 검색

캘린더 버튼을 클릭하면 사용 가능한 캘린더 목록을 불러오고, 캘린더 선택 시 선택한 캘린더의 일정을 RecyclerView로 검색 입력 부분 하단에 리스트로 표시하는 옵션이 제공된다. 선택한 캘린더에 따라 해당 일정을 필터링할 수 있다.

④ 검색 결과 표시

검색된 일정은 BottomFragment를 통해 하단에 동적으로 표시되며, 검색된 일정의 제목, 날짜, 시간과 같

은 주요 정보를 포함한다. 검색 결과는 RecyclerView를 활용하여 목록 형태로 표시되며, 검색 입력 부분 하단에 배치되어 사용자가 직관적으로 검색 결과를 확인할 수 있다. RecyclerView는 데이터베이스 검색 결과를 기반으로 동적으로 업데이트되며, Fragment 간의 데이터 연동을 통해 검색 결과가 실시간으로 반영되도록 설계되었다.

❖ 수행 결과

본 일정 검색 기능은 사용자가 효율적이고 직관적으로 특정 일정을 찾을 수 있도록 지원하며, 검색 결과는 화면 하단에 즉각적으로 표시되어 사용자 경험을 향상시킨다. BottomFragment를 활용한 결과 표시 방식은 화면 구성의 일관성을 유지하며, 필요한 정보에 빠르게 접근할 수 있도록 지원한다.

현재 첨부된 이미지는 코드 통합 중 도출된 예제이며, 캘린더 검색과 RecyclerView가 아직 포함되지 않은 상태다. 검색된 일정을 목록 형태로 정보를 제공하며, 각 일정 항목을 클릭 시 상세 정보 확인 및 수정이 가능하도록 지원할 예정이다.

② 선택한 날짜의 일정 목록 화면

달력에서 선택된 날짜의 날짜정보와 해당 날짜에 등록된 이벤트(일정)를 확인할 수 있고 새 이벤트를 등록하거나 기존 이벤트의 수정 및 삭제 기능을 제공한다.



그림 25 선택된 날짜의 이벤트 목록화면

- 선택된 날짜 정보 확인: 달력에서 선택된 날짜 데이터를 전달받아 화면 최상단에 '20--년 -월 -일 -요일' 형식으로 날짜정보가 출력된다.
- 이벤트 목록 확인: 날짜정보 하단에는 ListView형태의 이벤트 목록이 표시된다. 현재 날짜와 일치하는 이벤트 데이터를 DB에서 Select쿼리를 사용하여 검색한 후, 검색된 이벤트의 ID에 해당하는 데이터를 가져와 객체 리스트에 추가한다. 추가된 객체 리스트의 데이터는 Adapter를 통해 이벤트 목록에 표시된다. 이벤트 목록은 시작시간 순으로 정렬된다. 목록의 항목에는 이벤트명과 시간을 표시하여 사용자에게 간략화된 이벤트 정보를 제공한다. 시간은 시작시간과 종료시간으로 구성하여 사용자가 쉽게 확인할 수 있도록 한다.
- 새 이벤트 등록창 실행: 화면 우측 하단에 이벤트 등록 버튼(+)이 표시된다. 버튼을 클릭 시 새 이벤트를 등록할 수 있는 다이얼로그 창이 실행된다.

③ 새 일정 등록 화면

새 이벤트 등록 버튼을 클릭 시 표시되는 화면(다이얼로그 창)이다. 이벤트의 상세정보가 표시되며 새 이벤트의 정보 입력 및 등록 기능을 제공한다.

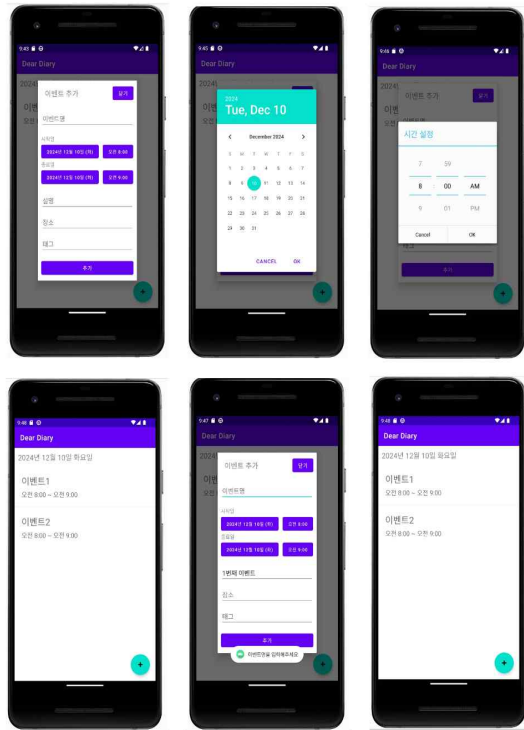


그림 26 새 이벤트 등록 기능

- 이벤트 정보 입력 : 이벤트명, 기간, 그 외 이벤트 정보를 입력할 수 있다. 초기 실행 시 시작일과 종료일의 기본값이 선택한 날짜로 설정 되어 있으며 시작시간과 종료시간의 기본값은 오전 8시와 오전 9시로 지정된다. 날짜와 시간 버튼을 클릭 시 각각 DatePicker와 TimePicker를 통해 날짜와 시간을 설정할 수 있다. 설정된 날짜는 '20--년 -월 -일 (요일)' 형식으로 표시되며, 시간은 12시간제를 사용하여 '오전 -시 -분' 또는 '오후 -시 -분' 형식으로 표시된다.
- 새 이벤트 등록: 화면 하단의 등록 버튼이 표시된다. 등록 버튼 클릭 시 입력된 이벤트 데이터를 DB에 전달하여 이벤트 테이블에 추가한다. 이벤트명과 날짜, 시간은 필수 입력 항목이다. 값이 입력되지 않거나 공백이 입력된 상태에서 등록 버튼을 클릭 시 필수 항목 입력 경고문이 ToastMessage로 출력되며, 이벤트 등록이 실행되지 않는다.

④ 기존 일정 수정 화면

목록에서 기존 이벤트를 클릭 시 실행되는 화면(다이얼로그 창)이다. 기존 이벤트의 정보 수정 기능을 제공한다.

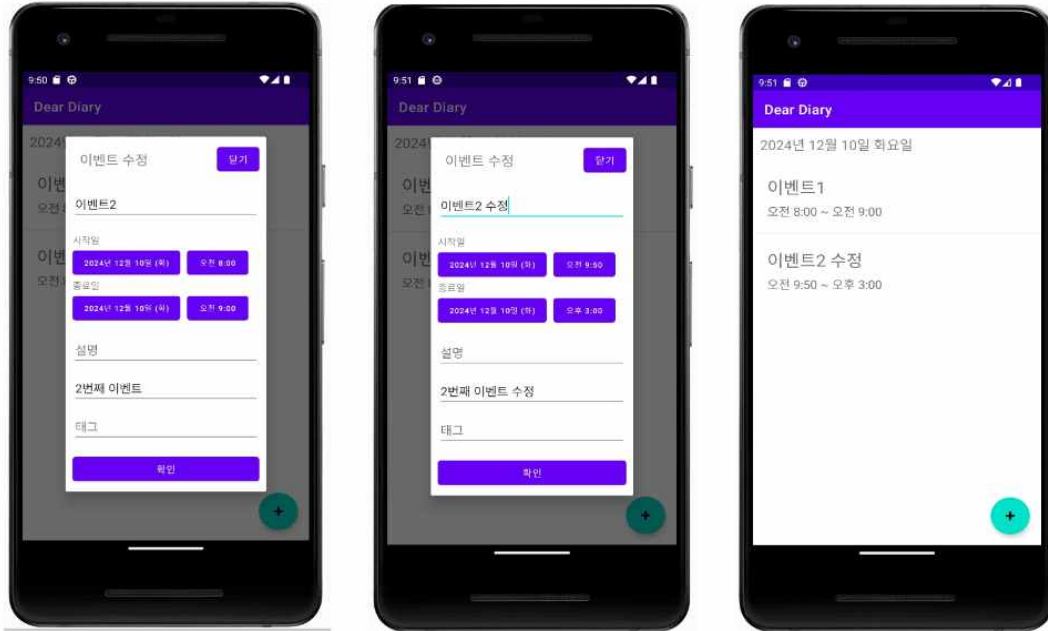


그림 27 이벤트 수정 기능

- 등록 이벤트 정보 확인: 목록에서 선택된 이벤트의 이벤트명, 날짜, 시간을 비롯한 기존 데이터가 기본 값으로 설정되어 표시된다.
 - 이벤트 정보 수정: 새 이벤트 등록화면과 동일한 방식으로 입력을 통해 이벤트 정보를 수정할 수 있다.
 - 수정된 이벤트 등록: 화면 하단에 수정 버튼이 표시된다. 수정 버튼을 클릭 시 DB로부터 기존 이벤트의 ID를 검색한 후 update문을 통해 수정된 데이터로 이벤트 정보를 갱신한다.
- 이벤트명의 입력을 비우거나 공백이 입력된 상태에서 수정 버튼을 클릭 시 필수 항목 입력 경고문이 ToastMessage로 출력되며, 이벤트 수정이 실행되지 않는다.

⑤ 기존 일정 삭제 화면

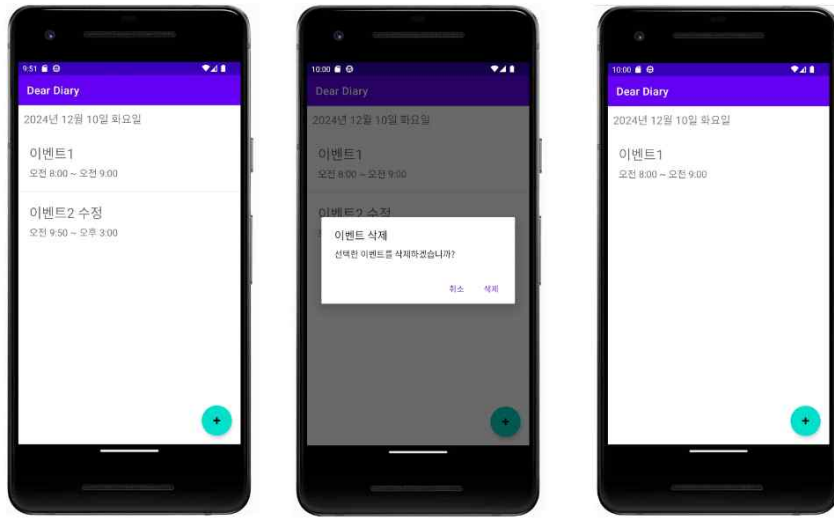


그림 28 이벤트 삭제 기능

- 이벤트 삭제: 이벤트 목록에서 특정 이벤트 항목을 길게 클릭 시, DB에서 클릭한 이벤트의 ID에 해당하는 이벤트를 검색한다. 이후 Delete쿼리를 사용해 검색된 이벤트를 DB에서 삭제하여 항목이 목록에서 제거된다.

⑥ 일정 데이터베이스 설계

이벤트 데이터는 일정 정보를 효율적으로 처리하고 관리하기 위해 데이터베이스(DB)를 활용하였다. DB를 통해 대용량의 일정을 체계적으로 저장하고, 사용자가 필요할 때 빠르고 정확하게 데이터를 조회할 수 있도록 한다. 또한, 데이터의 일관성 및 안정성을 확보하고, 시스템 성능을 최적화하여 사용자 경험을 향상시키는 것을 목표로 한다.



그림 29 이벤트 테이블 다이어그램

DB에 사용된 이벤트 테이블 구조이다. 이벤트 ID를 Primary Key로 하여 이벤트가 추가될 때마다 ID값이 자동으로 증가된다. 이벤트명을 나타내는 event_title과 날짜, 시간을 나타내는 start_date, end_date, start_time, end_time은 NOT NULL 제약조건을 두어 일정 관리에 있어서 필수적인 항목에는 값 누락을 방지하고자 하였다. 설명(description), 장소(place), tag(태그)에는 NULL이 허용된다.

❖ 수행 결과

일자별 이벤트 정보를 보다 체계적으로 관리할 수 있게 되어, 각 날짜에 해당하는 일정과 관련된 사항들을 명확하게 추적하고 기록할 수 있는 시스템이 구축되었다. 이 시스템은 이벤트 등록, 수정, 삭제 기능을 통해 사용자에게 더 높은 편리함과 유연성을 제공한다. 사용자는 각 날짜에 대한 일정을 쉽게 등록하고, 필요에 따라 일정을 수정하거나 삭제할 수 있으며, 이러한 모든 변경 사항은 실시간으로 반영되어 관리의 정확성을 더욱 강화한다. 이를 통해 이벤트와 관련된 모든 사항을 효율적으로 처리할 수 있게 되어, 일정 관리의 신속성과 정확성을 크게 향상시킬 수 있다.

(7) 알람 설정 및 캘린더 전환 기능

본 애플리케이션은 사용자가 알람과 캘린더 기능을 직관적으로 사용할 수 있도록 다양한 버튼과 화면 전환 기능을 제공한다. 또한 설정 화면을 통해 사용자 편의성을 강화하고, 문제 해결을 돕는 기능을 포함하고 있다.

① 알람 설정 기능

본 애플리케이션은 사용자가 알람 설정을 손쉽게 수행할 수 있도록 네 가지 주요 단계를 포함한 알람 설정 기능을 구현하였다. 각 단계별 세부 사항은 다음과 같다.



그림 30 설정 화면

- 날짜 선택: 사용자가 알람을 설정할 날짜를 선택할 수 있는 인터페이스를 제공한다.
- 시간 선택: 알람이 울릴 특정 시간을 사용자가 지정할 수 있도록 한다.
- 알람 설정: 선택된 날짜와 시간을 기반으로 AlarmManager를 활용하여 알람을 정확히 설정할 수 있도록 구현하였다.
- 알람 취소: 기존에 설정된 알람을 해제하는 기능을 제공한다.

레이아웃은 ConstraintLayout을 활용하여 화면 크기에 상관없이 버튼과 레이블이 일관되게 배치되도록 설계되었다. 사용자가 날짜와 시간을 선택한 후 "알람 설정" 버튼을 클릭하면 AlarmManager가 알람을 설정하며, "알람 취소" 버튼을 통해 설정된 알람을 손쉽게 해제할 수 있도록 구현하였다.

② 캘린더 화면 및 보기 전환 기능

본 애플리케이션은 다양한 캘린더 보기 모드를 제공하여 사용자의 필요에 따라 일간, 주간, 연간 보기를 전환할 수 있도록 설계되었다.

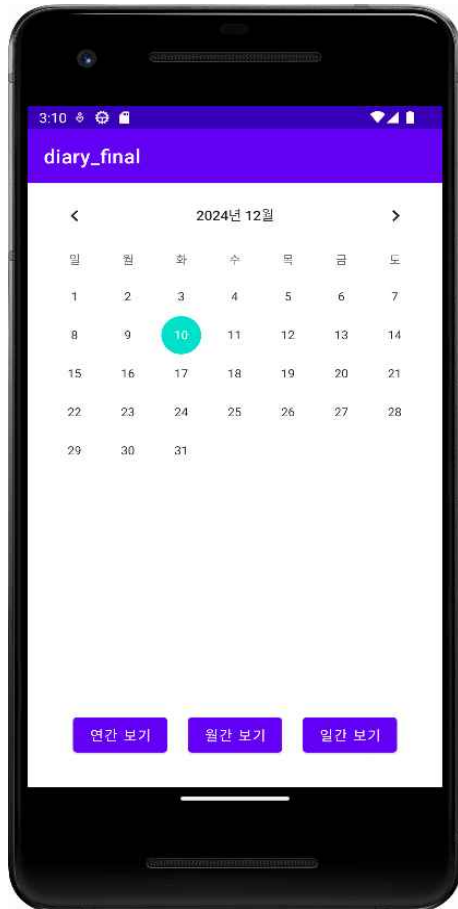


그림 31 캘린더 화면 이동

- 캘린더 화면 이동 : 메인 화면에서 캘린더 보기 버튼을 클릭하면 캘린더 화면으로 전환된다.
- 보기 전환 : 캘린더 화면 하단의 버튼을 통해 YearlyViewActivity, WeeklyViewActivity, DailyViewActivity로 각각 연결되며, 연간, 주간, 일간 보기를 손쉽게 전환할 수 있다.

캘린더 관련 인터페이스는 XML 레이아웃 파일을 기반으로 직관적인 UI를 제공하도록 설계되었다. 버튼 및 전환 기능은 사용자가 자연스럽게 탐색할 수 있는 환경을 조성하였다.

③ 지원 및 도움말 기능



그림 32 지원 및 도움말 화면



그림 33 문의 사항



그림 34 답변 확인 가능

지원 및 도움말 화면은 사용자가 애플리케이션을 효과적으로 활용할 수 있도록 설계되었으며, 세 가지 주요 기능을 포함한다.

- 문의 사항: 애플리케이션 사용 중 발생한 문제나 요청을 직접 게시할 수 있는 인터페이스를 제공한다. 이를 통해 관리자가 실시간으로 답변을 제공하며, 사용자와 원활하게 소통할 수 있다. 피드백을 효율적으로 수집하여 애플리케이션 품질을 향상시킬 수 있다.

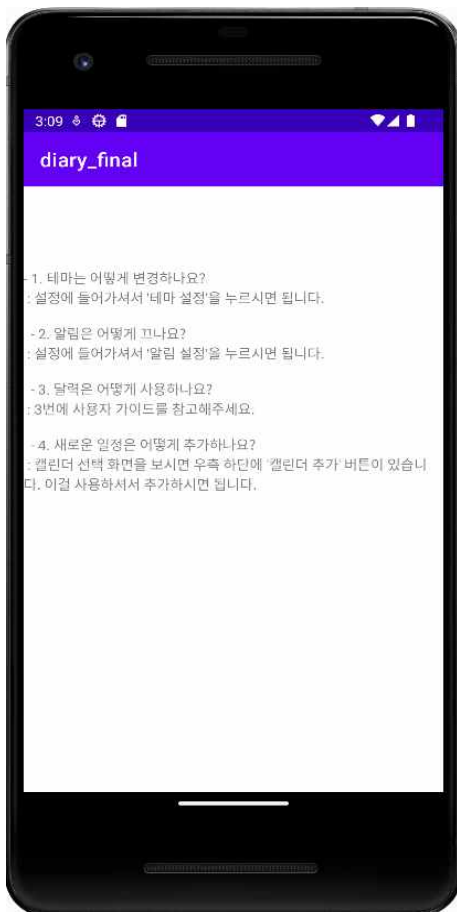


그림 35 자주 묻는 질문

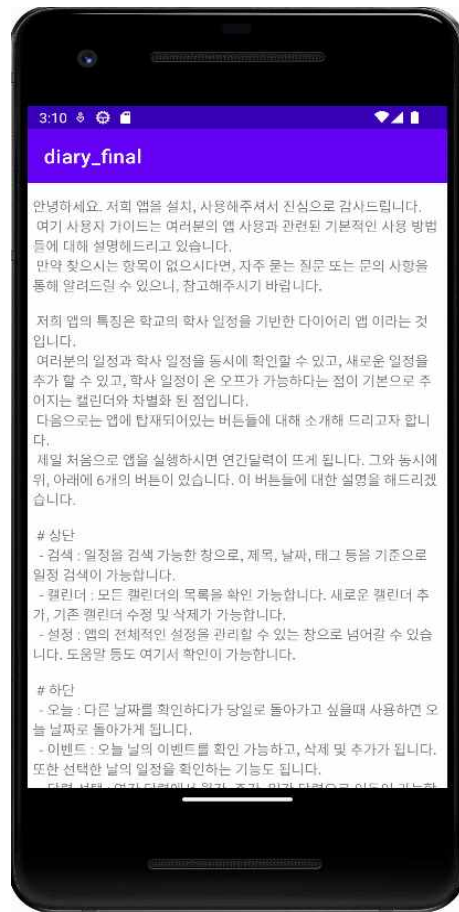


그림 36 사용자 가이드

- 자주 묻는 질문(FAQ): 자주 발생하는 문제 및 궁금증에 대한 답변을 미리 제공하여 사용자가 원하는 정보를 신속하게 찾을 수 있도록 한다. 이를 통해 반복적인 질문을 줄이고 사용자 편의성을 증대시킬 수 있다.

- 사용자 가이드: 애플리케이션의 주요 기능과 설정 방법에 대한 상세한 설명을 제공하여, 초보 사용자도 애플리케이션을 쉽게 익힐 수 있도록 지원한다.

설정 화면은 XML 레이아웃 파일을 활용하여 버튼과 레이블을 직관적으로 배치하였으며, 필수적인 정보를 명확히 전달하도록 설계되었다.

❖ 수행 결과

본 애플리케이션은 알람 설정, 캘린더 전환, 설정 화면 기능을 통합적으로 제공하여 사용자 친화적인 인터페이스를 구현하였다. 각 기능은 AlarmManager와 Activity 전환을 활용해 설계되었으며, 설정 화면을 통해 사용자가 발생 가능한 문제를 해결하고 애플리케이션을 효과적으로 사용할 수 있는 환경을 조성하였다. 이를 통해 사용자 요구를 충족하고 앱의 활용성을 극대화하였다.

5) 테스트

본 프로젝트에서는 오류 사항을 점검하고 성능을 개선하는 작업을 수행하였다. 현재 예정된 일정보다 지연되어 테스트 단계에 있으며, 코드 통합과 통합 테스트 절차를 진행하고 있다.

3. 도출결과 내용

본 애플리케이션은 학교 일정과 개인 다이어리를 통합 관리할 수 있는 기능을 제공하며, 이를 통해 사용자는 학교 일정과 개인 일정 관리를 효율적으로 수행할 수 있다. 학사 일정은 애플리케이션에서 자동으로 연동되며, 실시간 업데이트를 통해 사용자는 최신 학사 일정을 확인하고 관리할 수 있다. 이를 통해 사용자는 학사 일정을 확인하고, 필요한 정보를 쉽게 파악할 수 있다.

또한, 전체 1년 달력을 기본으로 하여 연간, 월간, 주간, 일간 보기 형태의 달력 기능을 제공하여 사용자가 원하는 형식으로 일정을 확인할 수 있도록 설계되었다. 이러한 보기 형태 간 데이터 연동이 원활히 이루어져, 사용자는 특정 날짜나 주간의 세부 정보를 쉽게 조회할 수 있다.

개인 일정의 경우 사용자가 직접 추가하고 수정할 수 있으며, 일정 추가 시에는 일정의 이름, 시작 시간, 종료 시간, 반복 여부 등 세부적인 설정이 가능하여 사용자 맞춤형 일정 관리를 지원한다. 일정 검색도 지원하는데, 검색은 키워드와 날짜를 기반으로 수행되며, 일부 단어만 포함된 일정도 검색 결과에 되어 검색 결과에서 확인할 수 있다.

캘린더 기능을 통해 사용자가 학교 일정과 개인 일정을 한눈에 확인할 수 있도록 설계되었다. 사용자가 다양한 캘린더를 불러오고, 추가, 편집, 삭제할 수 있도록 지원한다.

설정 기능을 통해 사용자는 애플리케이션의 다양한 옵션을 사용자 맞춤형으로 설정할 수 있다. 알림 설정, 캘린더 전환, 테마 변경 등의 설정 기능을 통합적으로 제공하며, 이를 통해 사용자 요구를 충족하고 애플리케이션의 활용성을 극대화하였다.

본 애플리케이션의 구현 결과, 학사 일정과 개인 일정 간 통합 관리가 가능해졌으며, 일정 관리의 효율성과 정확성을 향상시키는 데 기여하였다. 사용자는 직관적인 인터페이스를 통해 일정을 체계적으로 관리할 수 있으며, 캘린더 관리 및 설정 기능은 사용자 편의성과 애플리케이션의 활용성을 더욱 높였다.

4. 기대 효과 및 활용 방안

1) 기대효과

이 프로젝트를 통해 개발된 애플리케이션은 학교 일정과 개인 일정을 통합적으로 관리할 수 있는 도구를 제공한다. 이를 통해 사용자들이 얻을 수 있는 기대효과는 다음과 같다.

① 시간 관리 향상

학사 일정과 개인 일정을 통합적으로 관리하여 시간 활용의 효율성을 극대화할 수 있다. 사용자는 별도의 노력 없이도 일정을 체계적으로 관리하고 계획을 실천할 수 있다. 한 곳에서 일정을 관리하여 중복과 누락을 방지할 수 있다.

② 학업 성취도 증대

학습 목표 설정 및 추적 기능을 통해 학생들이 체계적으로 학업 계획을 수립하고 실천할 수 있도록 돕는다. 일정 누락과 같은 문제를 방지하여 학업 성과를 향상시킬 수 있다.

③ 스트레스 감소 및 생활 균형 지원

학사 일정 관리의 번거로움을 해소하고, 사용자에게 적절한 알림과 가이드를 제공함으로써 스트레스를 줄인다. 학업과 개인 생활의 균형을 유지할 수 있도록 지원한다.

이러한 기대효과를 통해 본 과제는 단순히 일정 관리의 편리성을 넘어, 사용자의 시간 관리 능력을 향상시키고 업무와 학업의 효율성을 증대시키며, 전반적인 삶의 질을 향상시키는 데 기여할 것으로 기대된다.

2) 활용 방안

개발된 학사 일정 기반 다이어리 애플리케이션은 사용자의 개인 일정 관리를 효율적으로 지원하며, 다양한 방식으로 활용될 수 있다. 다음은 주요 활용 방안을 제시한다.

① 시간 관리 교육 도구로 활용 가능

학습 목표 설정 기능을 추가하여 사용자가 개인화된 학습 계획을 수립하고 달성 상황을 점검할 수 있는 기회를 제공한다. 이를 통해 학생들의 자기 주도적 학습을 촉진하며, 스스로 계획을 세우고 시간 관리를 실천할 수 있게 된다. 또한, 학업뿐만 아니라 운동, 취미, 사회활동 등 개인 일정도 함께 관리할 수 있는 기능을 추가하여 학생들이 학업과 일상생활의 균형을 유지하는 데 도움을 줄 수 있다. 일정 충돌 시 알림 기능을 통해 사용자에게 효율적인 시간 계획 수립을 돕고, 시간 관리의 중요성을 교육하는 도구로 활용 가능하다.

② 교육기관과의 연계 활용

교육기관과 협업을 통해 시간 관리 교육 도구로 활용할 수 있다. 교육 기관에서 제공하는 학사 일정을 애플리케이션에 통합하여, 학생들이 앱을 통해 학사 일정을 즉시 확인하도록 지원할 수 있다. 또한, 교사는 학사 일정, 과제, 학습 목표 등을 제시할 수 있으며, 학생들은 이를 애플리케이션을 통해 확인하고 관리할 수 있다. 학부모는 학생들의 학습 목표 달성 상황을 확인하고, 일정 관리에 도움을 줄 수 있다. 학사 일정뿐만 아니라 학교 행사, 동아리 활동 일정, 시험, 과제 마감일, 휴일 등의 정보도 통합하여 관리함으로써, 학생들의 시간 관리와 전반적인 교육 활동을 효율적으로 지원할 수 있다.

③ 확장 가능성

건강 관리, 재정 관리, 개인 프로젝트 관리 등 다양한 일정 관리 기능과의 연계를 통해 더 광범위한 활용이 가능하다. 이러한 확장을 통해, 사용자에게 다양한 관리 기능을 제공하고 전반적인 삶의 질 향상에 기여할 수 있을 것으로 예상 된다.

이러한 활용 방안을 통해 본 과제는 학사 일정과 개인 일정 통합 관리뿐만 아니라, 시간 관리 교육 도구, 건강 관리, 재정 관리 등의 분야에 활용될 수 있는 가능성을 가지고 있다. 이를 통해 사용자는 보다 효율적인 시간 관리와 삶의 질 향상을 기대할 수 있을 것이다.

V. 결론

1. 결론

다이어리 애플리케이션은 이미 시장에 다양하게 존재하며, 많은 사용자가 자신에게 익숙한 애플리케이션을 계속 사용하는 경향이 있다. 이에 따라 새로운 애플리케이션을 선택하게 하려면 기존 앱에서 제공하지 않는 차별화된 기능이 필수적이다. 우리는 대학생들을 주요 타깃으로 삼아 학업, 대외 활동, 취업 준비, 자격증 공부 등 다양한 활동을 동시에 계획하고 수행할 수 있도록 돕는 학사 일정과 개인 일정을 통합하여 관리하는 기능을 도입하였다. 이 기능은 기존 다이어리 애플리케이션에서 제공하지 않는 차별화된 요소로, 사용자들의 관심과 이용을 이끌어낼 것으로 기대된다.

본 과제는 학사 일정과 개인 일정을 통합적으로 관리할 수 있는 다이어리 애플리케이션 Dear Diary를 개발함으로써 사용자들의 시간 관리와 학업 성취를 효과적으로 지원하는 것을 목표로 삼았다. 개발 과정에서는 학사 일정 연동, 사용자 맞춤형 일정 관리, 알림 서비스, 캘린더 기능 등 주요 요구사항을 충족시키기 위해 지속적으로 설계와 기능을 개선하였으며, 사용자 친화적이고 직관적인 인터페이스 구현에 주력하였다.

프로젝트를 통해 대학생들이 시간 관리에서 겪는 어려움을 실질적으로 해결할 수 있는 도구를 제공하고자 하였으며, 이를 통해 학업과 개인 생활의 균형을 유지하고 학업 몰입도를 높이는 데 기여할 것으로 기대된다. 특히 학사 일정을 자동으로 확인하고, 별도의 입력 없이 개인 일정과 통합적으로 관리할 수 있는 기능은 기존 다이어리 애플리케이션의 한계를 효과적으로 극복하였다.

개발 과정은 요구분석, 설계, 구현, 테스트의 체계적인 단계를 거쳤으며, 팀원 간의 역할 분담 및 협력을 통해 모든 기능을 효율적으로 구현하였다. 반복적인 피드백과 개선 과정을 통해 프로젝트의 완성도를 높였으며, 최종 보고서 작성과 제출로 프로젝트의 결과물을 체계적으로 정리하였다.

Dear Diary는 단순한 일정 관리 도구를 넘어, 대학생들이 시간 관리 습관을 형성하고 체계적인 학업 계획을 통해 목표를 달성하도록 돕는 데 실질적인 가치를 제공한다. 본 과제는 향후 유사한 일정 관리 시스템 개발의 기초 자료로 활용될 수 있기를 기대하며, 이를 기반으로 사용자들에게 더욱 발전된 시간 관리 솔루션을 제공할 수 있는 가능성을 열어두고자 한다.

2. 과제수행 후기

이번 프로젝트를 통해 이전보다 더 많은 지식을 습득할 수 있었다. 데이터 연동, 데이터 저장 및 전달, 데이터베이스 활용, Fragment, Navigation, 의존성 등 새로운 기술들을 접하고 배울 수 있는 좋은 기회였다. 사용자에게 유용한 기능을 제공하기 위해 끊임없이 고민하고 개선하는 과정을 통해 문제 해결 능력과 협업의 중요성을 깨달았다. 또한, 이번 프로젝트는 처음으로 대면이 아닌 온라인으로만 진행된 팀 프로젝트였는데, 이를 통해 온라인 협업의 장단점을 뚜렷하게 경험할 수 있었다. 무엇보다 개인 일정보다 팀 과제를 우선시하고, 업무를 미루지 않는 것이 얼마나 중요한지 깨닫게 된 시간이었다.

다른 팀원 후기 생략