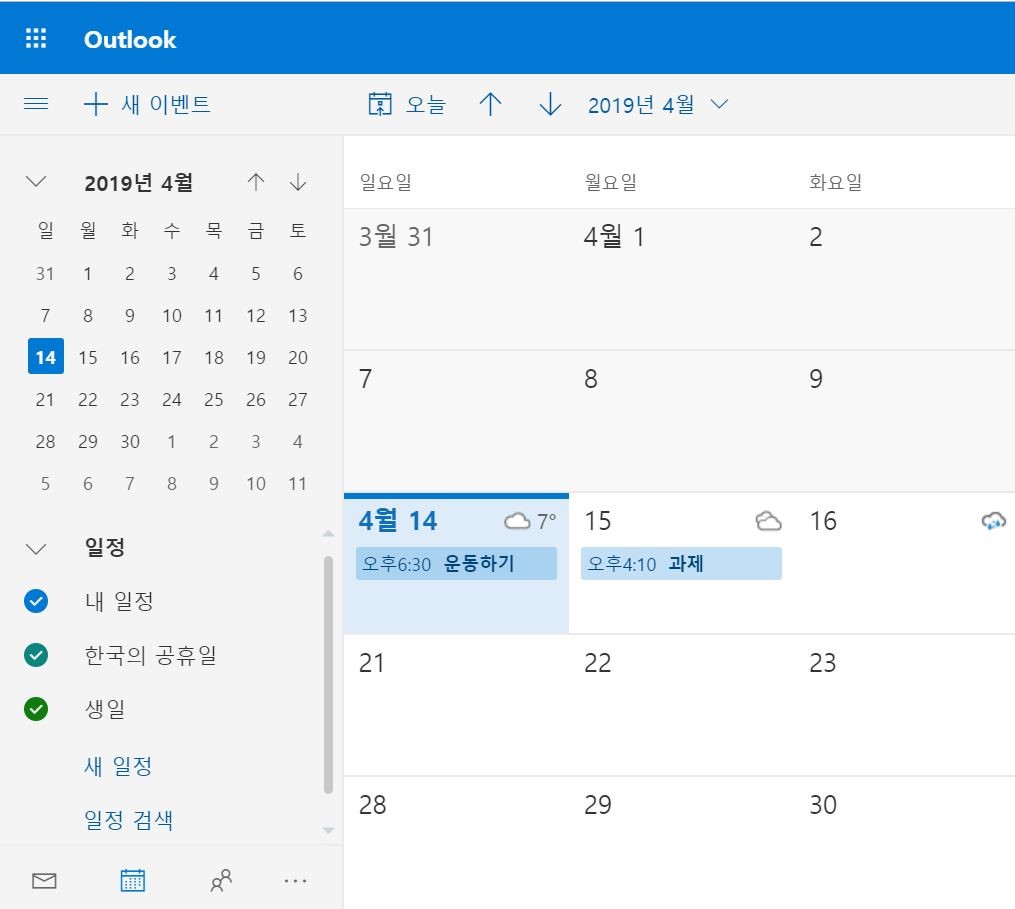
*Merge Calendar*

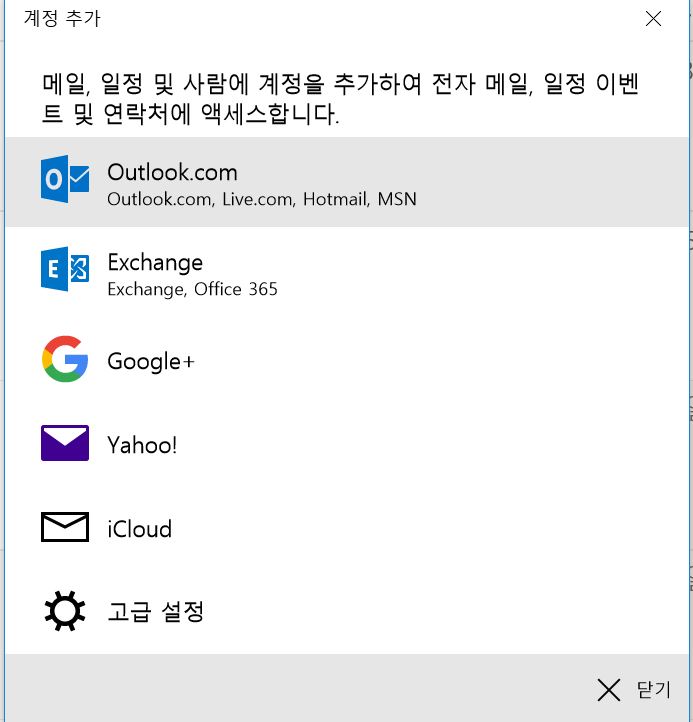
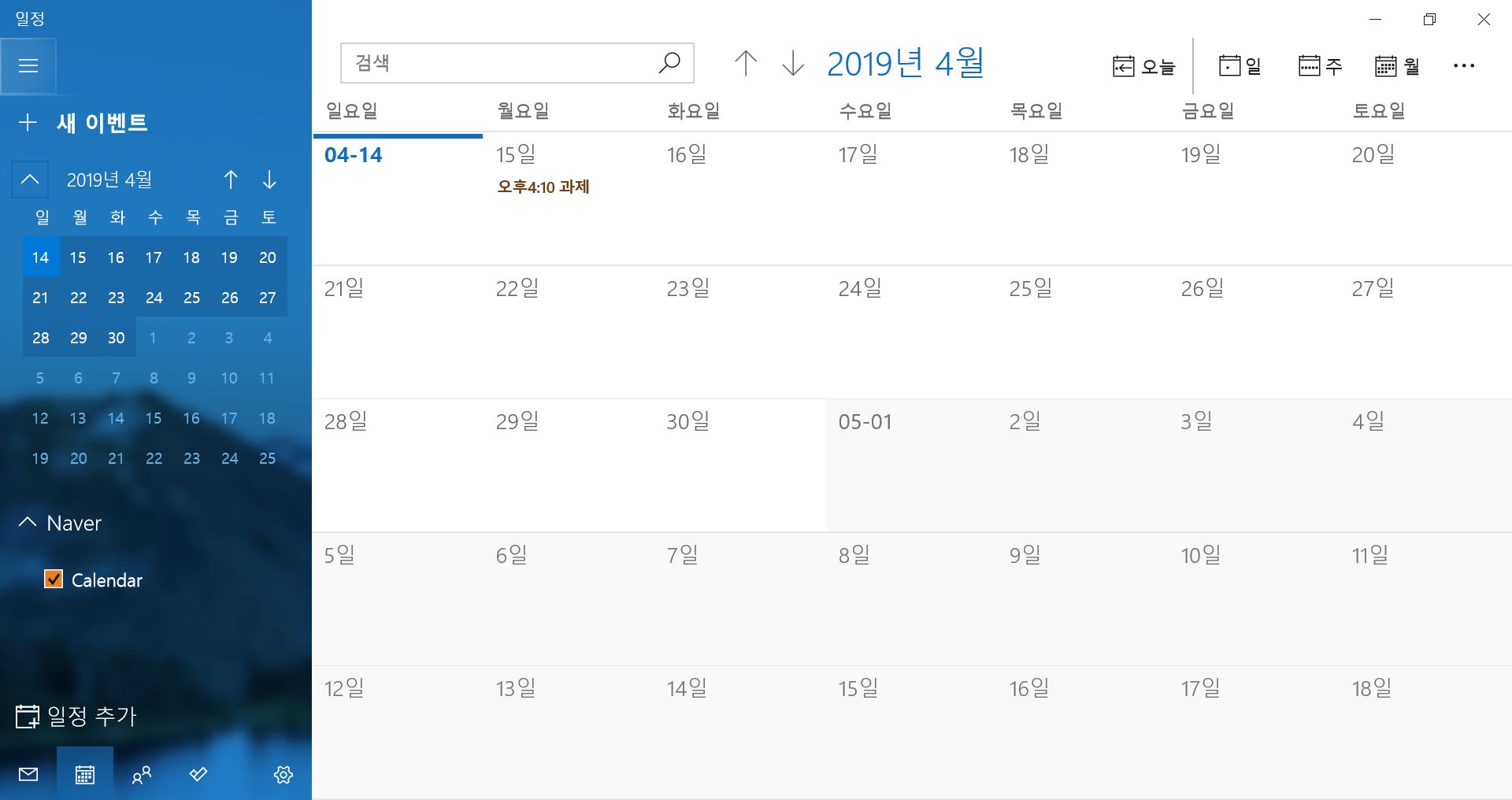
*이영훈 (201610674, 컴퓨터공학부)*

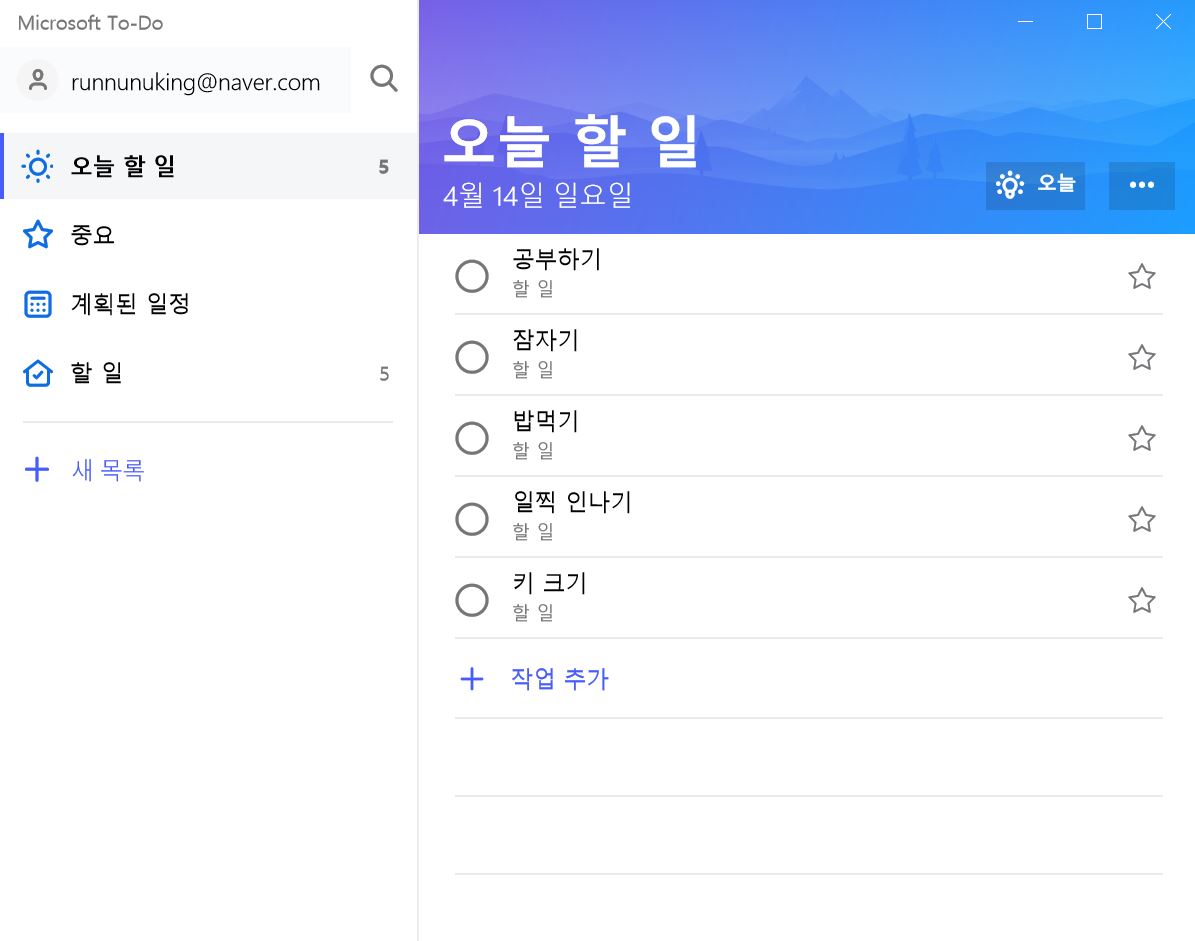
*정종범 (201614906, 컴퓨터공학부)*

1. **프로그램 개요**
   * 서비스(앱, 웹, 프로그램) 별로 분산되어 있는 일정을 하나의 서비스로 통합한 일정 관리 프로그램
   * 일정과 To-Do List를 날짜별로 통합해서 효율적인 스케줄 관리
   * 학습에 도움이 되는 일정(국가기술 자격시험 일정, 학사 일정 등)을 API로 일정 등록
   * 개별 서비스들의 장점들을 유지하며 하나의 서비스로 이용하고 싶어 개발하게 됨
   * 학생들이 사용하기 편리한 직관적인 일정관리 프로그램을 목표로 설정
   * 다양한 일정과 할 일 목록을 통합한다는 의미에서 Merge Calendar로 이름 정함
2. **관련 프로그램/서비스 조사 결과**
   * MS Office Outlook

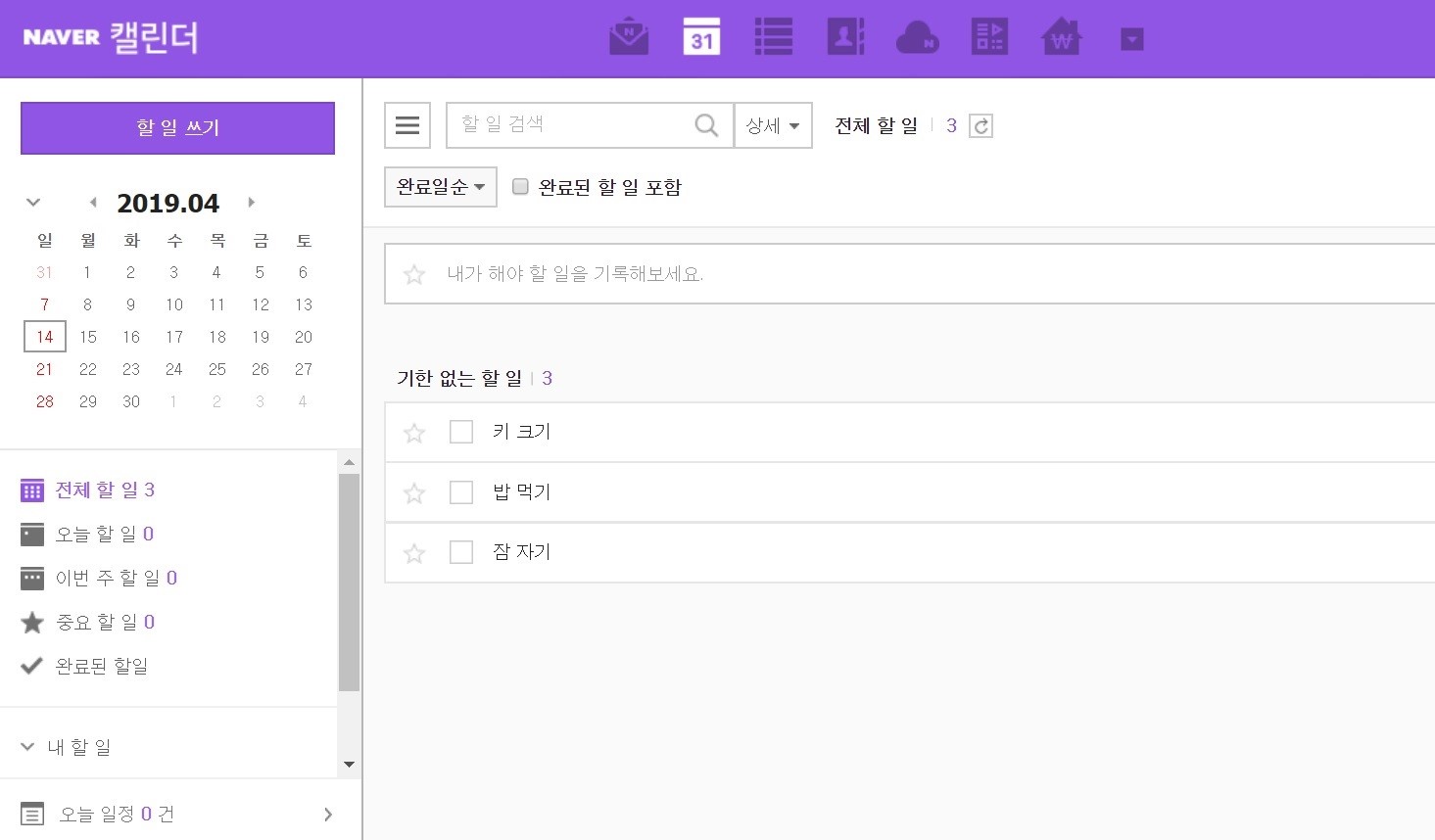
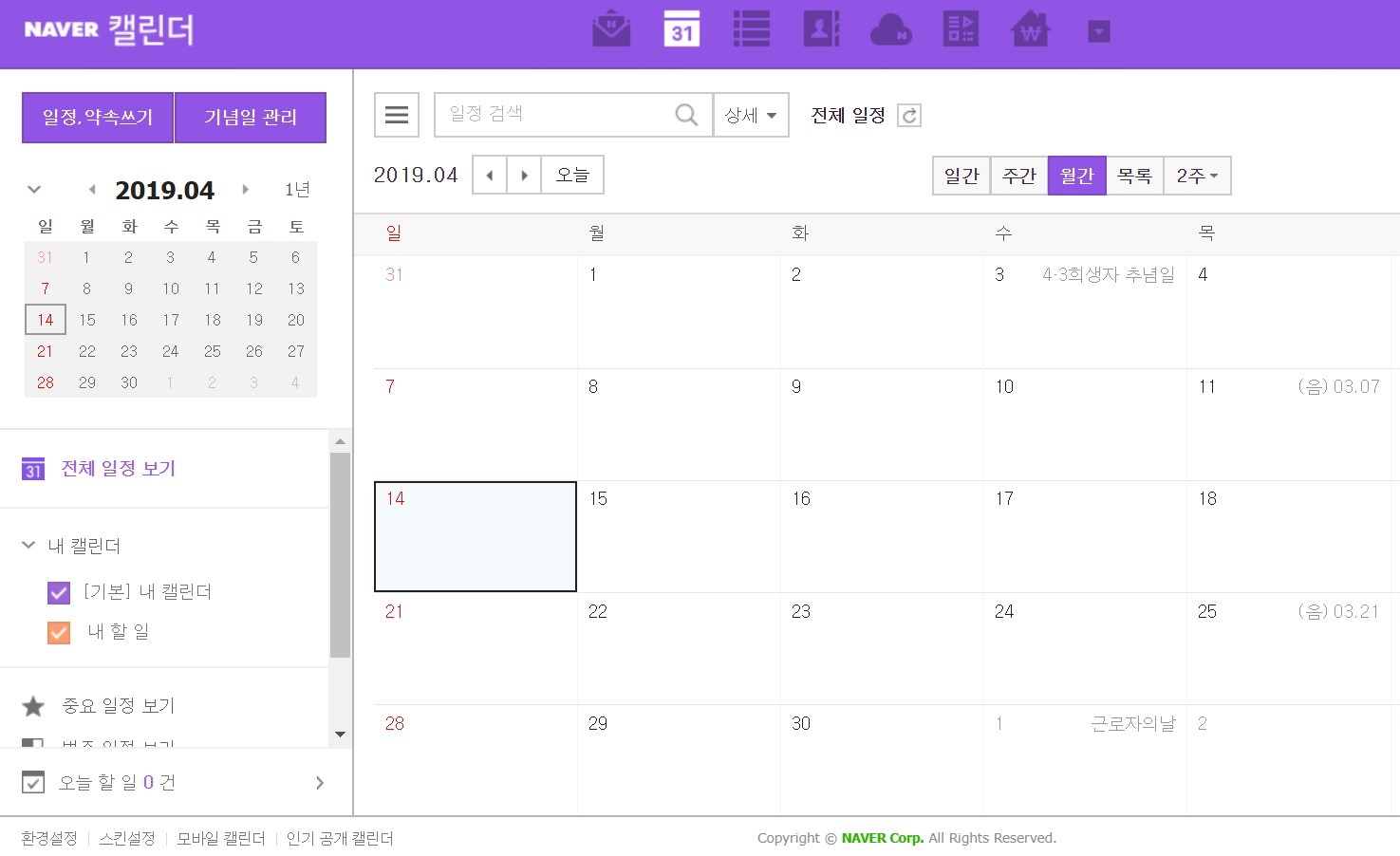


* + 1. MS 계정 연동 가능하나, 할 일 목록은 따로 존재하지 않음
    2. 타 캘린더 서비스와 통합 불가능(MS사의 Skype 통화 가능)
  + MS 일정, To-Do, Task 어플리케이션

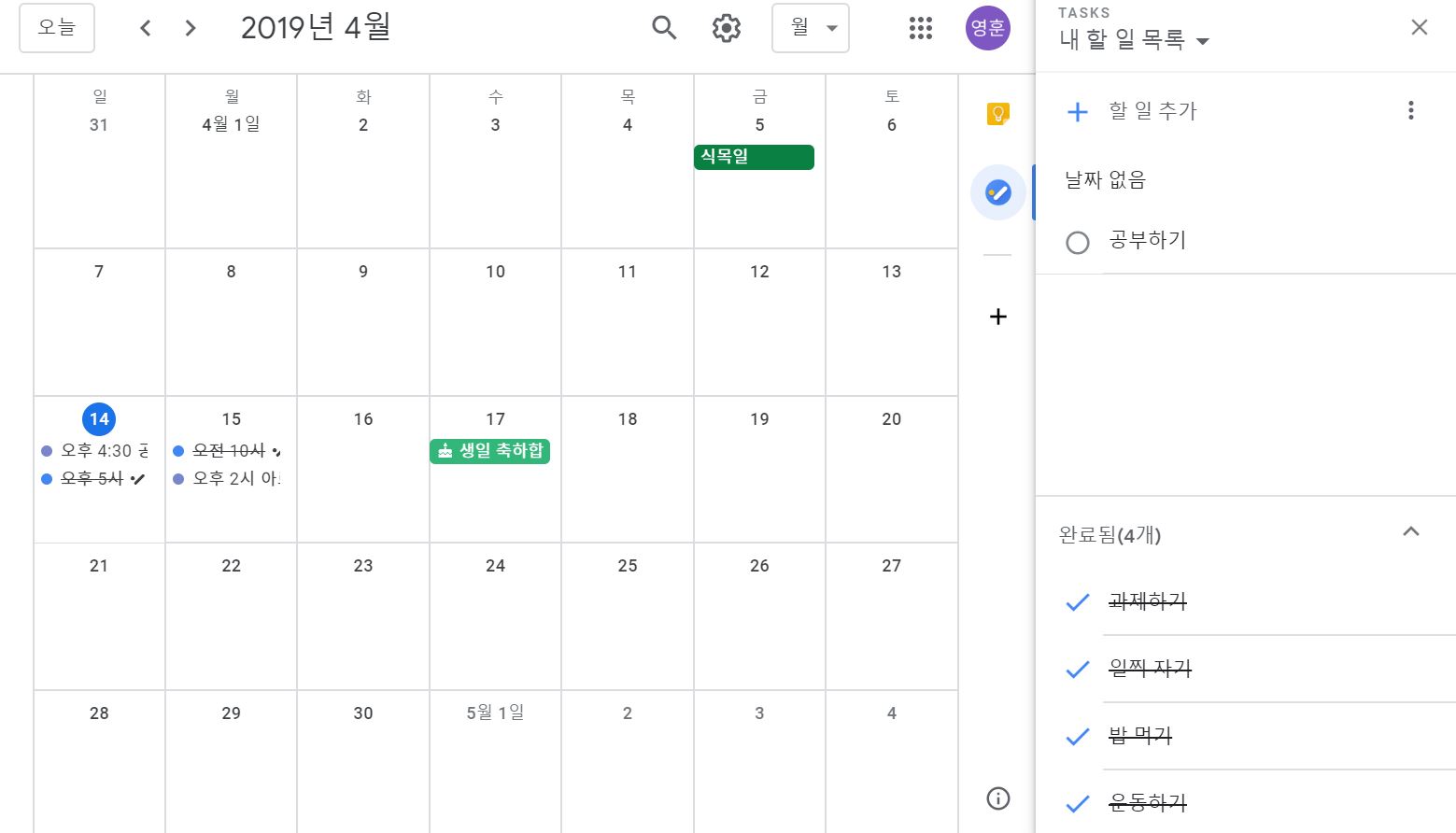
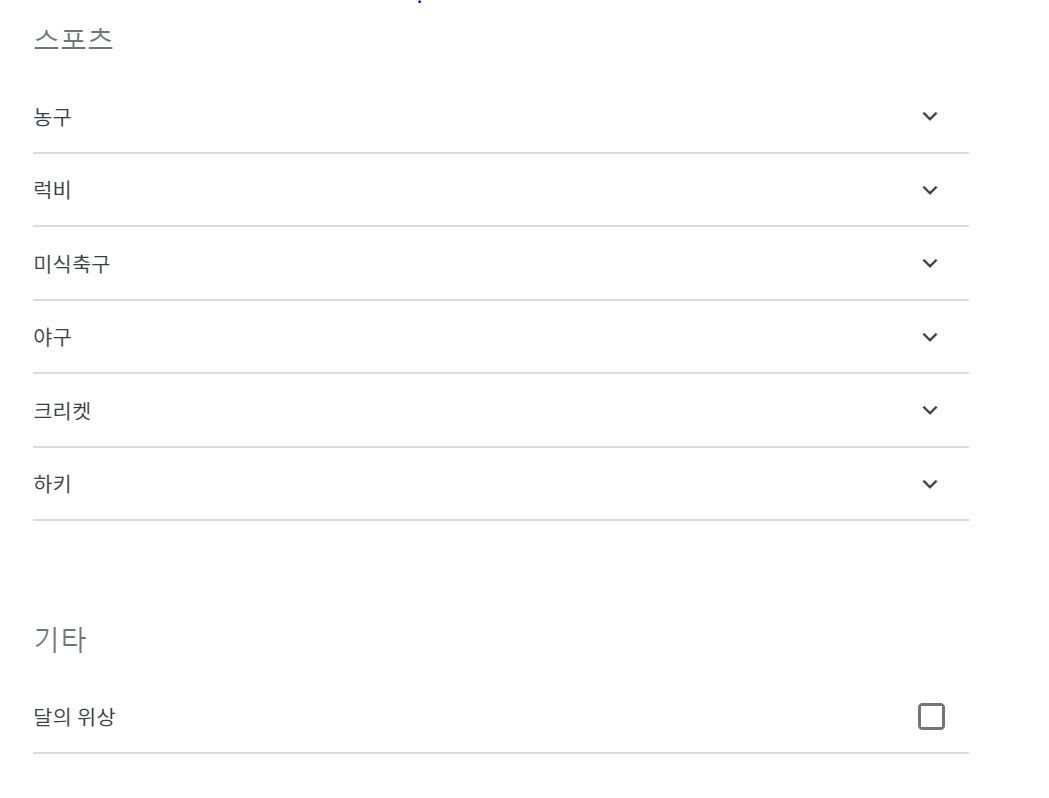
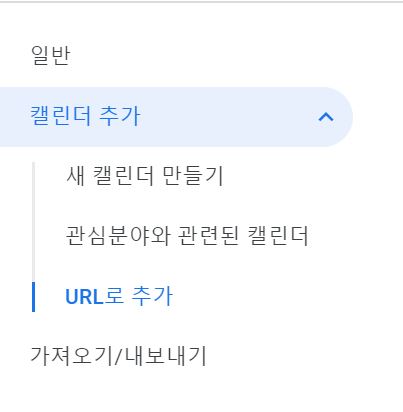
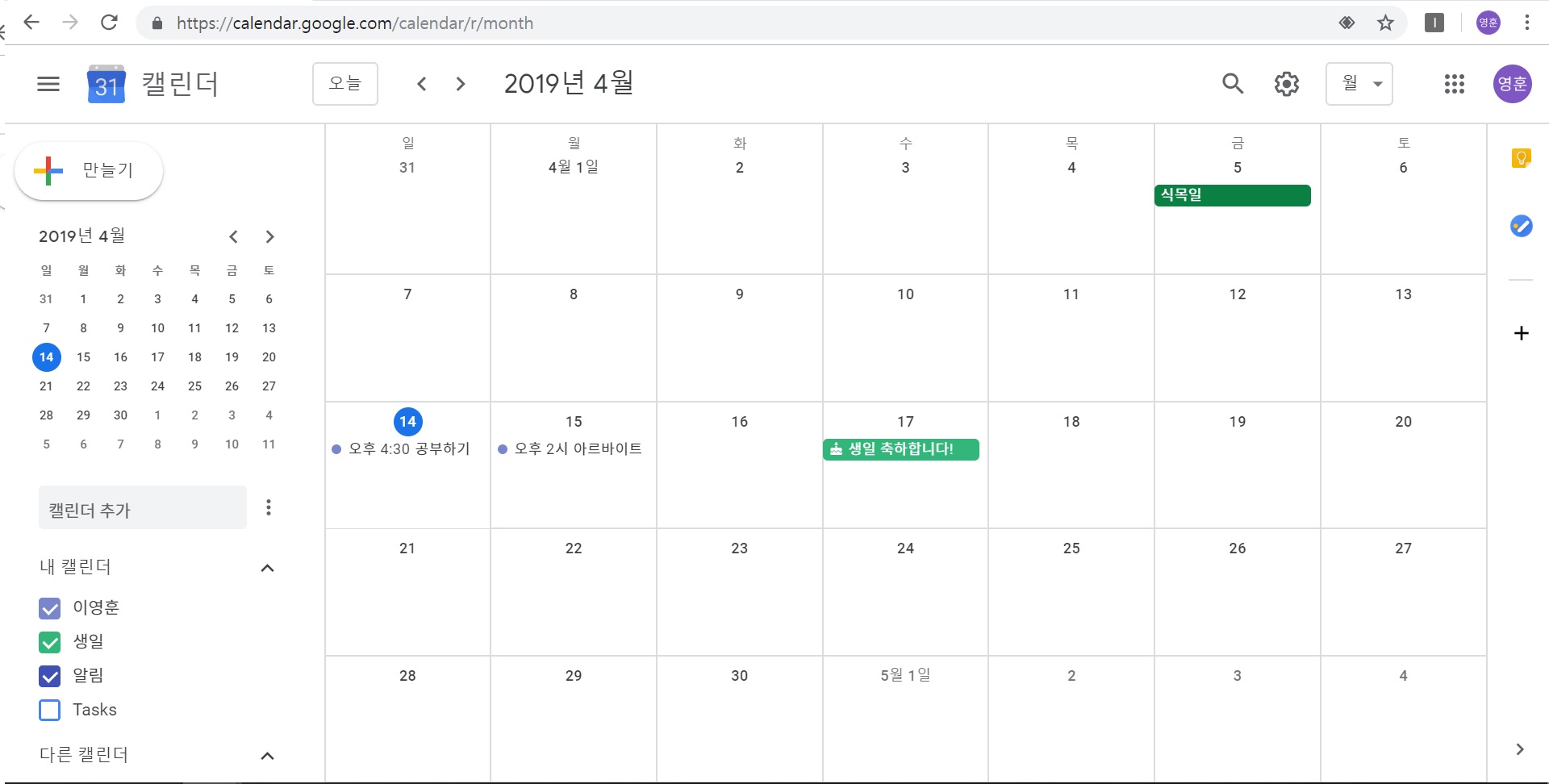




* + 1. Outlook과 마찬가지로 MS 계정 연동 가능
    2. 타 캘린더 서비스와 연동 가능
    3. To-Do나 Task 어플리케이션 설치 시 이용 가능하나 별도의 어플리케이션으로 유기적인 연결이 부족
  + Naver 캘린더

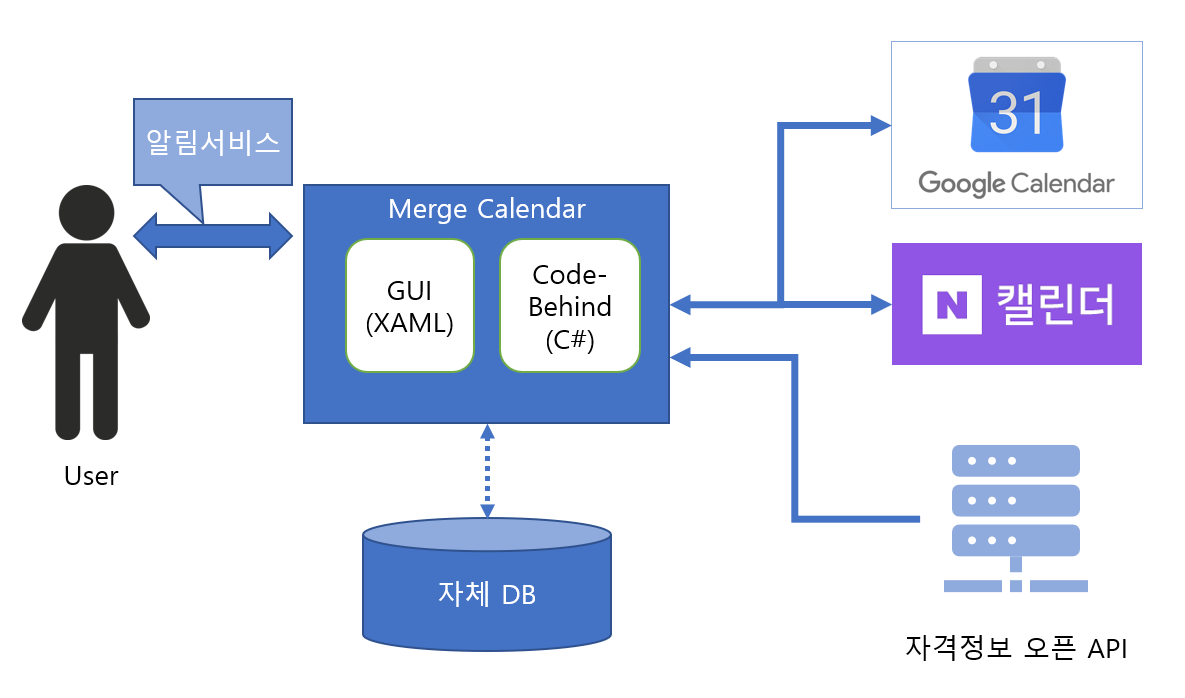


* + 1. 웹 프로그램으로 일정과 To-Do List 기능이 존재
    2. 할 일 목록을 동시에 볼 수 없으며, 타 캘린더와 연동이 되지 않음
  + Google 캘린더, Tasks, Keep (웹, 모바일 앱)

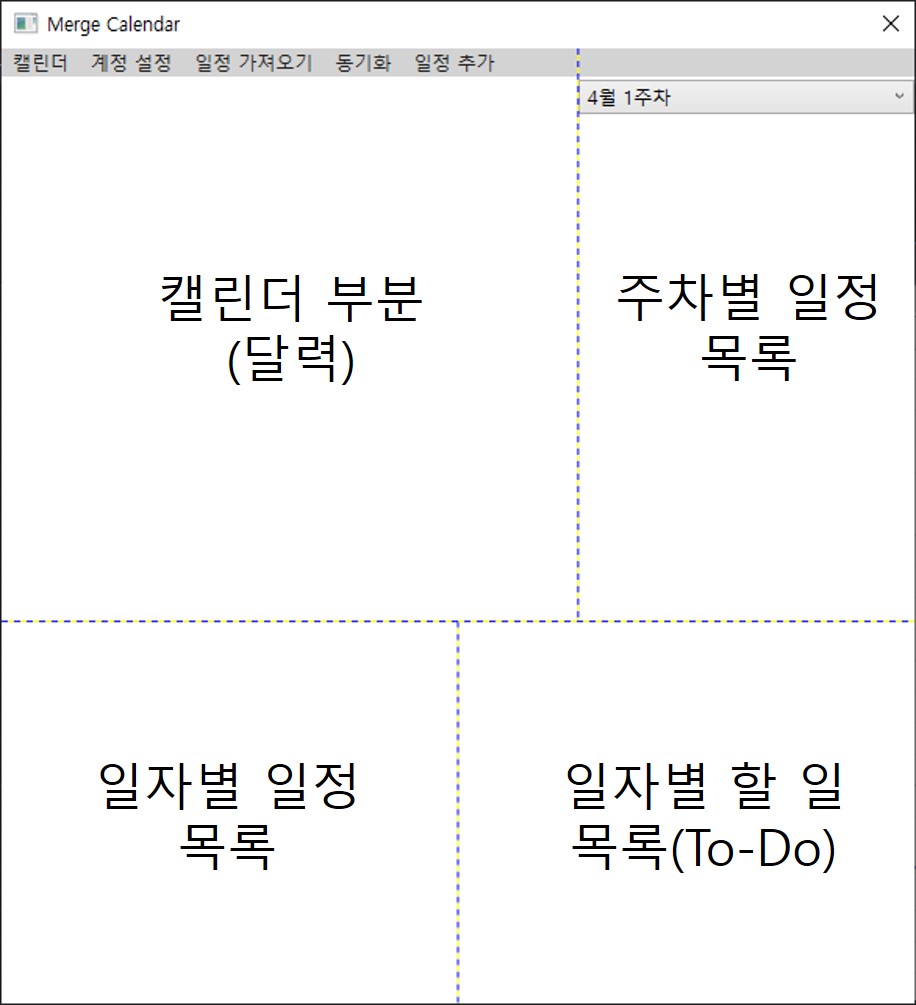


* + 1. 일정 관리, 타 캘린더와 연동가능, Tasks와 Keep을 날개에서 사용 가능
    2. Tasks에 할 일 등록시 Calendar에 자동 등록, 알림 기능 사용 가능
    3. ‘관심분야와 관련된 캘린더’기능이 있지만 극히 제한적 (ex. 스포츠, 달의 위상, 국가별 공휴일)
  + To-Do List와 캘린더를 유기적으로 연결해주는 서비스는 구글이 유일함
  + 그러나 관심분야의 주요 일정(Merge Calendar의 경우 학업, 취업 일정)을 가져오는 기능이 제한적
  + 모두 클라우드 기반 서비스로 오프라인 상태에서는 접속 불가능
  + 대부분의 서비스(Google, Naver, Outlook)이 Open-API를 개방해 놓은 상태로 개발이 용이하며, 사용자 인증으로 OAuth를 이용

1. **프로그램 구조**
   * 프로그램 구조 요약도

****

* + 1. 프로그램 자체 데이터베이스에 일정과 할 일 정보를 저장
    2. 타 캘린더 서비스 연결 시 별도의 계정 인증 과정이 필요
    3. 사용자가 타 캘린더의 일정을 API로 가져와 데이터베이스에 별도 저장 및 표시
    4. 프로그램에서 등록한 일정을 외부 서비스에 자동 추가
    5. 시험 일정(기사, 산업기사 등)은 자격정보 오픈 API를 통해 주기적으로 자동 등록
    6. 사용자는 Notification 서비스를 통해 정해진 시간에 일정을 알림 받을 수 있음
  + 메인 화면 레이아웃



* + 1. 레이아웃은 그리드 패널을 주로 사용하여 구성
    2. 메뉴를 이용하여 기능 선택 가능하도록 함
    3. 캘린더 부분에 직관적인 사용자 정의 캘린더 컨트롤 개발 후 이용할 예정
    4. 주차별 일정 목록은 선택된 주차의 일정을 스택 형태로 보여줌
    5. 일자별 일정과 할 일은 캘린더의 선택된 부분에 따라 내용이 변경되도록 개발

1. **개발 일정**
   * 역할 분담
     1. 디자이너 역할 : 이영훈 (GUI / XAML 위주)
        1. 전반적인 프로그램 레이아웃 구성
        2. 메뉴, 할 일 목록, 인증 폼 구성
        3. 사용자 정의 컨트롤(달력) 개발
        4. 알림(Notification) 기능 구현
     2. 개발자 역할 : 정종범 (Code-Behind / C# 위주)
        1. API 일정 불러오기, 등록하기 숙지 및 구현
        2. OAuth로 계정 인증 구현
        3. 데이터베이스 이용 방법 숙지 및 구현(LINQ)
        4. Timer를 이용한 주기적인 타 캘린더 동기화
        5. 알림(Notification) 기능 구현
   * 향후 개발 일정 (학교 수업 주차에 따름)
     1. 9주차
        1. 디자이너 : Panel 구성, 메뉴 제작
        2. 개발자 : API 활용 신청, 활용 방법 숙지
     2. 10주차
        1. 디자이너 : 달력 사용자 정의 컨트롤 개발, 인증 폼 구성
        2. 개발자 : DB 이용방법, OAuth를 이용해 계정 인증 구현
     3. 11주차
        1. 디자이너 : 할 일 목록에 체크박스, 우선순위 변경 기능 추가
        2. 개발자 : DB 저장 및 불러오기 구현, Timer 이용
     4. 12주차 ~ 13주차 : Notification 등 추가적인 구현 / 디버깅
     5. 14주차 : 발표 준비