

# 소프트웨어공학 7조

## PROJECT1

컴퓨터소프트웨어학부 2021089216 이현석

컴퓨터소프트웨어학부 2020056480 정민규

컴퓨터소프트웨어학부 2017029989 우현빈

컴퓨터소프트웨어학부 2021034184 다브완 라나

컴퓨터소프트웨어학부 2019004102 서문하선

미래자동차공학과 2020005387 임규빈

# Real Time Assistant Service

## Application Proposal

### 1. Introduction

'Real Time Assistant Service Application(가칭)'은 캘린더와 연동하여 사용자의 일정을 관리하고 수행할 수 있도록 도와주는 application이다. 많은 수업과 과제, 동아리 등의 일정을 소화하면서 효율적으로 시간을 관리하기까지 수많은 시행착오를 겪는데, AI의 도움을 받아 이러한 시행착오를 줄이고 보다 효율적인 일정관리 및 도움을 받고자 본 application을 계획하였다. 인공지능 API를 사용하여 등록된 일정에 적합한 정보 및 팁을 일정에 맞추어 제공하며, 일정의 달성률을 바탕으로 피드백을 제공하고, 사용자가 등록한 일정과 task의 우선순위를 고려하여 일정을 추천해주는 서비스를 제작할 예정이다. 이러한 application의 개발을 통해 개인의 생산성을 높이고, 일상을 챙길 수 있는 시간을 확보하여 사용자의 삶의 질을 높일 수 있을 것이다.

### 2. Objectives

위 프로젝트를 통해 AI의 도움을 받아 사용자가 등록된 일정을 원활히 수행하도록 하는 데 초점을 맞춘 application을 제작하며, 다음과 같은 세부 목적을 가진다.

- 바쁜 일상에서 놓칠 수 있는 부분을 AI를 사용하여 팁 및 피드백 제공
- 일정 달성에 도움을 줄 수 있는 정보 및 피드백 제공
- Task의 우선순위를 고려하여 일정 추천 및 수립
- 등록된 사용자 간의 일정 공유를 통한 공유 일정 수립
- 사용하기 편리한 인터페이스 제작

### 3. Target Customers

'Real Time Assistant Service Application(가칭)'은 deadline이 있는 task와 고정된 일정이 많은 현대인들을 대상으로 한다. 또한 일정 공유도 가능하므로 공유된 일정이 많은 사람에게도 적합하다. 위 application이 주된 타겟으로 하는 사용자들은 다음과 같다.

- 바쁜 일정으로 일상생활을 챙기는 데 어려움이 있는 직장인
- 많은 과제와 수업을 효율적으로 챙기는 데 도움을 받고 싶은 학생
- 집중력 있게 정해진 일정을 소화하는 데 도움을 받고 싶은 사용자

- 본인이 일하는 방식에 맞게 일정을 추천받고 싶은 사용자

## 4. Features

본 application의 feature을 Main Feature, Design Feature, Security Feature, ETC로 나누어 정립하였다.



[Figma를 통한 feature 브레인스토밍]

### 4.1. Main Features(Calendar & AI)

Main Features에는 application이 기본적으로 수행할 일정 관리 및 피드백 기능을 선정하였다.

#### a. Task Management

Task의 deadline과 우선순위를 지정하고, task를 생성, 수정, 삭제한다.

#### b. Calendar Integration

Google Calendar, Outlook 등 사용자의 다른 Calendar와 일정 및 Task를 동기화하고 캘린더에 표시한다.

#### c. Reminders and Notifications

예정된 일정과 마감시한이 다가온 Task에 대한 알림을 보낸다. 일정 변동이 있을 경우 알림을 띄운다.

#### d. Customizable Categories

일정 및 Task에 라벨을 붙여 어떤 종류의 일정인지 식별해 맞춤 서비스를 제공한다. 예를 들어 '소프트웨어공학 팀프로젝트 과제' Task에 work 라벨이 붙어있으면 모여서 회의하기 좋은 장소를 추천해주고, personal 라벨이 붙어있으면 들으면서 작업하기 좋은 팝송을 추천해준다. 캘린더에서 라벨을 쉽게 확인할 수 있도록 라벨마다 다른 색깔로 표시한다.

**e. AI-Powered Insights**

일정의 달성률에 대한 정보를 사용자 feedback을 통해 적극적으로 받고, AI로 사용자의 일정 패턴을 분석해 사용자의 생산성을 높이고 여유시간을 확보할 수 있는 피드백을 제공한다.

**f. AI-Powered Task Scheduling and Time Estimation**

Task의 deadline, 우선순위 및 소요시간을 지정하여 AI로부터 task를 수행할 최적의 시간대를 제안받는다.

## 4.2. Design features

디자인 측면에서 application이 가져야 할 기능을 정립하였다. 사용자가 불편함 없이 사용할 수 있도록 편안한 컬러를 사용하고, 가시성이 높은 디자인을 추구한다.

**a. Dark Mode and Themes**

사용자 재량으로 변경이 가능한 다크모드 제공

**b. Data Visualization**

Calendar에 월별/일별로 일정 및 task를 표시한다. 또한 등록된 task를 리스트 view 및 그리드 view 등으로 보여준다.

## 4.3. Security features

보안 관련 기능을 제공한다. 기본적으로 로그인을 통해 본인 계정의 캘린더를 관리하며, 허용된 사람 외 타인이 접근할 수 없도록 한다.

**a. Community and Support**

일정 관리 모범사례나 팁을 공유할 수 있는 사용자 커뮤니티를 구성한다. 또한 application 관리자에게 메일을 넣는 등 관리자가 사용자 피드백을 받고 도움을 줄 수 있는 다양한 고객지원서비스를 제공한다.

**b. Collaboration Features**

Task나 일정을 가족, 동료 등 등록된 사용자와 공유하는 기능을 통해 가족 일정, 약속, 협업 계획 수립에 도움을 제공한다.

#### 4.4. ETC features

##### a. Data Backup and Sync

사고에 대비하여 데이터 복구를 위해 클라우드에 자동으로 백업한다. 또한 여러 디바이스에서 사용할 경우 동기화될 수 있도록 한다.

##### b. Offline Mode

오프라인 상태더라도 사용자가 계획을 수정할 수 있도록 한다. 인터넷이 활성화되면 오프라인에서의 수정사항이 반영되도록 한다.

#### 5. Tools & Resources to use

안드로이드, IOS 등에 호환이 가능한 application을 제작할 예정이므로 멀티플랫폼 어플리케이션 제작 툴이 필요하다. 또한 백업 및 커뮤니티 구성을 위한 서버 및 클라우드 툴, 인공지능 서비스 제공을 위한 인공지능 api 및 구글캘린더 api, 디자인 툴을 리스트업하였다. 세부적으로 어떤 툴을 사용할 지는 프로젝트를 진행하면서 각 툴의 장단점을 파악해 정할 예정이다.

프로젝트 관리 및 코드 레포 관리를 위한 툴들을 선정하였다. 협업툴 Figma를 사용하여 프로젝트의 진행도를 체크하고 회의를 하며, git을 사용해 레포지토리를 관리할 예정이다.

Application tools	Server	Cloud tools
AppMaster Appery.io Android Studio	Java / Spring MySql	AWS Google Cloud * 비용 발생
API	Design tools	Code repository
ChatGPT API 구글 캘린더 API Bing AI * 비용 발생	Adobe Illustrator Adobe Sensei * 비용 발생	Git (형상관리) figma (협업) ERD cloud (ERD 다이어그램) StarUML (UML 다이어그램) draw.io (기타 다이어그램)

## 6. Challenge

본 application의 기능을 검토하며 아래와 같은 다양한 challenge가 발생할 것을 예상하였다. 아래와 같이 미리 고민해봄으로써 수정이 필요하다면 어느 부분을 어느 정도로 수정할지에 대해 도움을 받을 수 있다.

### 6.1. 직관적인 인터페이스 구성 및 데이터 시각화

Main feature 구현에 있어서 직관적인 인터페이스 설계와 효과적으로 기능이 작동할 수 있도록 정교한 알고리즘 설계가 필요함.

### 6.2. 캘린더 API의 사용

Google Calendar와 어떻게 연동을 할지, 또한 직접 다른 Calendar 기능을 만들어서 원하는 항목에만 Input을 받을지 Tradeoff 고려와 설계가 필요함.

### 6.3. Task의 라벨에 따른 AI 학습 데이터 준비 및 원활한 상호작용 구현

Task 카테고리가 다르다면 그에 대한 AI의 반응도 다르게 해야 하며, 적합한 답변을 할 수 있도록 해야함.

### 6.4. 보안 및 동기화

커뮤니티 등의 기능에서, 다른 사용자가 허락 없이는 내 Calendar를 볼 수 없도록 해야함. 두 개의 디바이스가 오프라인 상태에서 서로 다른 스케줄을 구축했다면 서버와 동기화하는 과정에서 데이터가 일관되지 않는 문제가 발생함. 이를 해결키 위한 고민 필요함. 오프라인에서 작성/수정된 스케줄 데이터가 서버에 저장된 스케줄과 비교될 때 이 방법을 어떻게 구현할지 고민이 필요함.