



캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	<i>K-Shareplace</i>
팀 명	펼기아
문서 제목	결과보고서

Version	2.0
Date	2024-05-19

팀원	안수현(팀장)
	정일형
	정의석
	이재훈

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	펼기아	
	Confidential Restricted	Version 2.3	2024-05-23

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인 I 수강 학생 중 프로젝트 “K-Shareplace”를 수행하는 팀 “펼기아”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “펼기아”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	중간보고서-K-Shareplace.doc
원안작성자	안수현, 정일형, 정의석, 이재훈
수정작업자	안수현, 정일형, 정의석, 이재훈

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2024-05-16	정일형	1.0	최초 작성	전체 초안 작성
2024-05-19	이재훈	2.0	추가	AI 관련 기술 스택
2024-05-20	정의석	2.1	추가	서버 및 데이터베이스 기술 스택
2024-05-20	안수현	2.2	추가	앱 관련 기술 스택
2024-05-23	정일형	2.3	추가	DB 및 시스템 아키텍처

본 양식은 컴퓨터공학부 캡스톤 디자인 I 과목의 프로젝트 결과보고서 작성을 위한 기본 양식입니다. 문서의 필수 항목을 제시하는 것이니 폰트, 문단 구조 등의 디자인 부분은 자유롭게 설정하기 바랍니다. 양식 내에 붉은 색으로 기술한 부분은 지우고 작성하기 바랍니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

목 차

1 개요	4
1.1 프로젝트 개요	4
1.2 추진 배경 및 필요성	5
2 개발 내용 및 결과물	8
2.1 목표	8
2.2 연구/개발 내용 및 결과물	10
2.2.1 연구/개발 내용	10
2.2.2 시스템 기능 요구사항	14
2.2.3 시스템 구조 및 설계도	16
2.2.4 활용/개발된 기술	18
2.2.5 결과물 목록	22
2.3 기대효과 및 활용방안	33
3 자기평가	35
4 참고 문헌	35
5 부록	37
5.1 사용자 매뉴얼	37
5.2 운영자 매뉴얼	38
5.3 배포 가이드	43
5.4 테스트 케이스	44

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	펼기아	
	Confidential Restricted	Version 2.3	2024-05-23

1 개요

1.1 프로젝트 개요

강의실과 동아리 및 소모임 방을 전환한 공유공간을 활용하여 교내의 공간부족 문제를 해결하고자 하여 나온 아이디어인 **K-SharePlace** 프로젝트는 앱을 중심으로 웹과 ai 기술을 접목하여 학부생들이 느끼는 공간부족 현상에 대한 해결책을 제시하고자 한다.

공간부족에 대한 갈등이 해결되기 위해서는 학부생들이 스터디, 멘토-멘티 활동 등 각종 모임활동을 하고자 할때 쉽게 장소를 제공 받아야한다. 학부생들이 장소를 쉽게 제공 받을 수 있는 가장 실용적이고 효율적인 방법은 단과대학별로 공간 대여에 관한 시스템을 확립하여 모임활동 장소를 제공하는 것이다..

사용자가 시스템에 대한 접근성이 뛰어나야 시스템의 편리함 및 성능이 극대화되기 때문에 접근성이 뛰어난 스마트폰 앱을 통해 시스템을 구현하기로 결정하였다. **K-SharePlace**를 이러한 점을 고려하여 학부생들이 앱을 통해 강의실과 공유공간의 대여 가능여부를 쉽게 판단하고 예약할 수 있도록 계획하였으며, 관리자 웹을 통해 시스템이 필요한 단과대학에서 이 서비스를 적용하여 해당 단과대학에 알맞는 정보를 등록할 수 있으며 앱을 관리하기 용이하도록 최대한 간결하게 동작시키며 직관적인 **UI/UX** 및 컬러를 적용하고자 하였다.

또한, ai 기술은 사용자가 공간 이용 후 제출한 강의실과 공유공간의 반납 사진에 대한 정리정돈 여부를 수치화하여 반납사진과 함께 관리자에게 전송되어 관리자가 직접적인 공간 관리에 대한 부담을 덜어준다. 강의실과 공유공간 외에 교내의 여러 공용 공간에 대한 혼잡도를 분석해주는 모델을 통해 학생들에게 밀집 정보를 제공하여 전체적 인구 분산 효과를 기대하고 있다.

먼저 앱은 **Flutter**를 사용하여 데이터베이스에 등록된 정보를 바탕으로 동적으로 사용자가 예약할 수 있는 페이지를 만들며 사용자는 등록되어있는 공유공간 및 강의실을 예약하고 관리자에게 문의를 보내거나 알림을 받을 수 있다. 또한 강의실과 공유공간외에 예약이 불가능한 자율주행스튜디오, 무한상상실등의 혼잡도를 **AI**를 이용하여 공용공간의 혼잡도를 분석후 안내하여 교내에서 자습할 공간에 대한 정보를 받아 보다 스스로 쉽게 자습할 공간을 찾아갈 수 있다.

앱을 관리할 수 있는 웹은 **React**로 구현하였으며 관리자가 관리자계정으로 로그인하여 강의실과 공유공간에 대한 정보를 입력하면 데이터베이스에 정보가 등록되도록 했으며 누구나 쉽게 앱을 관리할 수 있도록 직관적이며 간결하게 웹을 구성하였다. 관리자는 사용자 전체 혹은 개인에게 알림을 보낼 수 있으며 사용자의 문의에 대한 답변을 보내 줄 수 있다. 강의실과 공유공간의 예약내역을 확인 할 수 있으며 필요에 의해서 강의실을 학부에서 사용해야하는 경우 강의실 예약 내역을 직접 삭제 할 수 있고 새로운 강의실 예약을 생성할 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

서버는 **Express**로 구성되어있으며 앱과 데이터베이스 그리고 웹과 데이터베이스 사이에서 데이터를 주고 받을때 필요한 모든 **api**를 구현해 놓았고 데이터베이스는 **Firestore**를 사용했다.

AI는 라즈베리파이를 활용하여 **restful api**를 이용하여 **flask**서버에 이미지를 인코딩하여 1분에 한개의 이미지를 보내주고 디코딩된 이미지를 활용하여 혼잡도 모델을 통하여 피어베이스에 혼잡도에 관한 데이터와 라즈베리 파이가 작동중인지의 여부에 대한 데이터를 저장한다. 웹은 설치된 라즈베리 파이의 작동여부를 통해 관리자의 라즈베리파이 유지/보수를 용이하게 하며 혼잡도 모델을 통해서 생성된 데이터는 앱의 사용자가 활용할 공간을 찾는 데 도움을 준다.

추가로 강의실 및 공유공간의 수월한 관리를 위하여 **Flutter** 앱에서 공간에 대한 반납시 뒷정리 상태를 체크하기 위해 이미지와 함께 요청을 넣으면 **ai**모델을 사용하여 판별 후 어느정도로 뒷정리가 되어있는지 데이터를 추출하여 피어베이스에 저장 및 페널티를 부여하여 관리자가 직접 강의실 및 공유공간에 방문하여 상태를 확인하지 않아도 된다.

1.2 추진 배경 및 필요성

현재 교내의 대다수 단과대학에서는 공간부족 문제로 어려움을 겪고 있다. 이는 학생들의 학습의 질 저하 및 스터디와 모임활동과 같은 다양한 활동들을 제한하며 학생들의 불편함을 야기하는 원인이 되고 있다. 새로운 부지 확보가 지속적으로 무산되는 어려운 상황 속에서 국민대학교의 고질적인 공간부족 문제는 해소되고 있지 않다. 학생들은 도서관, 식당, 공유공간에서 공간부족 문제를 가장 크게 실감하고 있으며, 본 학부인 소프트웨어융합대학을 예로 들었을 때 단과대학 내 공유공간 부족 문제로 인하여 단과대학이 소유한 두개의 공유공간을 8개의 동아리 중 4개의 동아리만이 공유하며 사용하는 것이 현재 상태이다. 이 또한 단과대학 학생회가 중재한 동아리 대표자 회의를 통해 타결된 사항으로, 동아리 8개를 감당할 수 있는 공간이 없는 환경과 공간 공유에 대한 시스템 부재로 인해 위와 같은 내용으로 결정되었다. 또한 2025년의 교육 개정안에 따르면 단과대학 모집 정원을 축소하고 자유전공화를 통해 융합형 인재 양성 및 전공에 대한 자유도를 부여할 것으로 계획 되어있다. 2025년부터 자유전공화 정책이 진행된다면 소프트웨어융합대학에 대한 자유전공 지원률이 가장 높을 것으로 예상될 뿐만 아니라 타 단과대학도 자유전공으로 인한 관심분야의 심층적 연구를 위해 단과대학 소모임 및 동아리 지원률이 높아질 것으로 예상된다. 각 단과대학은 한정된 공간을 부여받아 그 안에서 관리하게 되어있으며 이렇게 된다면 동아리 방 및 학업목적으로 부여된 공간에 대한 학생들 간의 갈등과 단과대학 차원에서의 문제 또한 심화될 것으로 예상된다.

 국민대학교 컴퓨터공학부 캠프스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

기존에 존재하던 동아리 및 소모임이 할당 받은 공간을 독점한다면, 새로운 동아리를 생성하는 것에 대한 장벽에 마주할 수 있으며 이는 다양한 분야의 동아리가 생성되는 것을 막는다. 다양한 분야의 동아리가 생성되지 못한다면 학부생이 원하는 분야의 동아리가 없어 해당 분야에 대한 경험을 제한하는 문제를 야기하며 단과대학의 발전 정체로 이어질 수 밖에 없다.

대학교에서의 모임활동은 동기 혹은 선후배간의 친목을 나눌 공간이자 자신의 분야를 확장, 발전 시킬수 있는 공간이다. 이러한 공간 부족문제들은 끊임없이 제기될 것이며 후에 어떠한 정책에 뒷받침 될 수 있는 시스템을 제시해야한다. 따라서 이러한 문제들을 해결하고 학부생들과 학부의 문제 해결 발판을 마련하고자 공간 부족 문제 해결에 관한 프로젝트를 시작하게 되었다.

교내 공간 문제, 시급한 대책 마련이 필요할때

✎ 이승준 기자,원한빈 기자,황혜운 수습 기자,양새람 수습 기자 | ⓒ 승인 2023.01.02 13:26 | □ 호수 989 | 🗨 댓글 0



캠퍼스 공간 부족 문제가 지속적으로 제기되고 있다. 시험기간마다 도서관에는 공부할 자리가 부족하고, 점심시간에는 식사 공간이 부족하다. 또 특정 단과대가 사용하는 건물의 경우 활동 공간 부족으로 학생들이 불편을 겪고 있다. 지난 2학기에는 복지관 동아리방 공간을 둘러싼 논란이 일기도 했다. 이러한 상황에서 우리학교가 공간 확장을 위해 어떤 노력을 해왔는지, 학생들은 어떤 점에서 불편함을 느끼는지 <국민대신문>이 취재했다.

평창캠퍼스 추진과 고려대학교 보건과학대학 부지 매입 시도

우리학교는 2007년 교지 확장 목적으로 평창동 부지를 매입한 바 있다. 하지만 매입 당시 부지 일부의 비오톱(biotope, 도심에 존재하는 특정 생물의 서식 공간) 1등급 지정이나 교통 혼잡 우려 등의 문제 때문에 개발을 추진하지 못했다. 이후 2014년 평창동 캠퍼스 이전 건으로 종로구청과 상호 업무협약을 체결하면서 종로구 아트밸리 조성 사업의 일환으로 캠퍼스를 이전한다는 계획이 수립됐다. 그리고 2015년 서울시가 평창동에 제3캠퍼스 조성이 가능해진 도시계획시설 변경안을 최종 통과시키면서 평창캠퍼스 신설에 대한 기대감이 높아졌다. 완공됐다면 예체능 계열 1300여 명의 학부생과 대학원생이 해당 건물을 활용할 예정이었지만 최종적으로 무산됐다.

우리학교는 평창동 일대에 조성 예정이었던 제3캠퍼스가 무산된 이후로 고려대학교 보건과학대학(이하

<공간 부족 관련 기사1>



국민대학교
컴퓨터공학부
캡스톤 디자인 I

결과보고서

프로젝트 명

K-SharePlace

팀 명

펼기아

Confidential Restricted

Version 2.3

2024-05-23



▲<국민대신문> 550호 4면 발취

<공간 부족 관련 기사2>

2 개발 내용 및 결과물

2.1 목표

우리 팀은 정책과 시스템을 동시에 제안한다. 제안하는 정책으로는 누구나 상시 확인 가능하고 예약 가능한 강의실의 개방화, 동아리방 및 소모임방을 공유공간으로 전환하여 더 많은 동아리들의 공간여건 보장 및 활동 기회 제공을 목표로 한다. 이를 뒷받침할 시스템으로 K-SharePlace를 설계했으며 AI를 활용하여 사용자들의 관리,예약, 인구분산 등의 편의성을 돕는다. 이로 인해 공간 부족문제에 대한 갈등을 해소하고 공간 부족문제가 제기 되었을 때 좋은 선례가 될 수 있는 프로젝트가 되는 것이 목표이다.

먼저 현재 강의실을 대여하기 위해서는 과 사무실에 강의실 예약 신청서를 받아 대표자가 대여한 강의실에 참석할 학생들의 서명을 모두 받아 다시 과 사무실에 제출하여 신청하는 방식을 사용하고 있다. 하지만 양식을 오프라인으로 직접 받은 후 작성해서 다시 오프라인으로 제출해야하는 과정은 매우 번거로우며 사용자들의 사용을 감소시킬 수 있다.

K-SharePlace의 이용하는 경우 직접 과 사무실에 들리지 않고 쉽게 신청서를 받아 예약하고 서명할 수 있기 때문에 대표자가 직접 움직이며 서명을 받아야하는 부담을 덜 수 있다.

 국민대학교 컴퓨터공학부 캠퍼스 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

강의실을 대여할 수 있다는 것을 아는 학생의 수가 많지 않으며 안다고 해도 빌리는 방법에 대해 구체적으로 모르기 때문에 빌리는 것이 어려울 것으로 보인다. 앱을 통해서 예약을 한다면 강의실 예약이 비어있는 시간을 한눈에 파악할 수 있으며 직접 과 사무실을 방문하지 않아도 되어 현재보다 많은 학생들이 빈 강의실을 활용할 것이다.

관리자의 경우 강의실 내부의 물건 도난등 관리에 어려움이 있어 강의실 대여에 관한 심사가 까다로울 수 밖에 없다. 신청서에는 대표자에게 책임소재가 있는 것을 명확히 밝힘과 동시에 공간 이용 후 사용 완료 사진을 찍어서 제출하기 때문에 문제가 발생할 경우 사진을 찍어서 관리자의 공간 관리에 대한 부담을 최소화 할 수 있다. 문제가 발생했을 경우 대표자에 책임소재가 있음이 밝혀져있으며 페널티를 부여하여 이용을 제한하기 때문에 사용자들도 대여를 위해서는 반납에 더욱 신경을 쓸 수 밖에 없다. 또한 관리자 웹은 확장성 좋은 관리를 할 수 있도록 만들었기 때문에 강의실, 공유공간, 혼잡도를 위한 카메라 설치 등 데이터 등이 직관적이기에 모든 데이터에 대한 추가, 수정, 삭제가 쉽다.

공유공간의 경우 동아리 소속원들이 친목을 다지거나 작은 규모로 모여 회의를 해야 하는 경우 공유공간을 예약하여 동아리에 소속된 인원이라면 누구든 공유공간을 사용할 수 있도록 하여 기존 8개의 동아리 중 4개의 동아리만 사용하던 동아리 방을 , 공유공간이라는 개념을 통해 추후 동아리의 개수가 늘어나더라도 예약제를 통해 공유함으로써 공간의 독점을 막고 새로운 동아리들의 유입과 발전을 도모할 수 있다.

위와 같은 영향을 통해 본교의 공간 부족 문제를 없앨 뿐만 아니라 본교 내부에서의 학부생 활동을 촉진 시켜 동기, 선후배 간의 교류에 도움이 될 것이다. 이는 학부생들의 발전에 도움을 줄 해결책이자 나아가 학교의 발전에도 기여할 수 있을 것으로 보인다.

 국민대학교 컴퓨터공학부 캠퍼스 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

2.2 연구/개발 내용 및 결과물

2.2.1 연구/개발 내용

본 프로젝트는 계획 단계에서 소프트웨어융합대학 내에서 사용할 플랫폼을 기획하였으나 공간 부족 문제가 소프트웨어융합대학을 포함한 여러 단과대학에서 해결되지 않는 고질적인 문제점이라는 것을 생각하여 대학교 전체의 공간부족 문제를 해결하고자 하였다.

대학교 전체의 문제를 해결하기 위해서 각 단과대학별로 남는 공간을 잘 활용하고 현재 가용 가능한 공간의 활용도를 극대화 시킬 수 있는 시스템을 구축하고자 했다.

이를 위해서는 학부생들이 쉽게 접근 할 수 있는 수단으로 시스템을 개발해야 한다고 생각하여 ‘캠퍼스 공간 최적화를 위한 스마트 예약 플랫폼’이라는 슬로건과 함께 ‘K-SharePlace’ 어플을 개발하기로 하였다.

어플의 확장성을 극대화하기 위해서는 앱이 각 단과대학에 쉽게 적용되어야 하며 관리가 쉬워야한다는 점을 고려하여 앱을 관리 할 수 있는 관리자 웹을 따로 만드는 것으로 기획하였다.

앱은 예정대로 안드로이드와 ios 쪽에서 확장성이 높고 개발시간과 일관성 유지가 가능한 Flutter로 유지하였으며 완성도 높은 앱을 기획하기 위해 가장 먼저 많은 학생들이 사용하는 운영체제인 ios로 개발하기로 결정하였다. Flutter는 라이브러리가 많고 문법 자체도 간결했기 때문에 앱 디자인의 원안을 유지하는 것에 큰 도움이 되었다.

웹은 React를 사용하였는데 높은 호환성과 다양한 라이브러리 그리고 import를 통해 재사용 가능한 컴포넌트 등 반복적인 기능이 많은 프로젝트와 적합하다고 판단되어 선택하게 되었다. 마침 backend도 Node.js를 사용하여 javascript를 주 문법으로 사용하기에 프론트와 백엔드 사이에서 서로의 코드를 이해하고 상의하기 편하다고 생각되어 선택하게 되었다.

서버도 마찬가지로 frontend와 코드리뷰가 편하다는 점과 비동기 I/O 작업을 처리해 여러 요청을 동시에 처리할 수 있으며 실시간 어플리케이션에 높은 성능을 제공하기 때문에 그대로 유지하기로 하였으며 사용자에게 알림을 보내거나 공지사항을 올려야하는 시스템 특성상 데이터베이스로는 Firebase를 채택하였다.

자율주행스튜디오, 무한상상실등의 교내 공용시설의 혼잡도를 라즈베리파이와 카메라를 이용하여 제공하기로 하였으며 기존에는 혼잡도를 검출하기위해 객체탐지 모델을 사용하여 빈책상수, 사람수를 탐지하여 사람이 몇명이 있는지, 빈 책상이 몇개인지 탐지하여 임계치를 설정하면 이를 바탕으로 혼잡도를 조사하는 방식이었으나 pretrain 되어있는 detr모델을 사용하는 만큼 성능 면에서는 매우 뛰어났으나 모델의 용량이 무겁고, 화질이 좋은 사진에서는 좋은 성능을 보이지만 화질이 떨어질 경우 성능이 안좋은 모습을 보여 밀집도를 판단하는 모델을 density map을 추출하는 CSRNet을 이용하여 혼잡도를 구하는 방식으로 변경하였다.

라즈베리파이 카메라에서의 실시간 영상과 인공지능 모델을 사용하기 위한 서버가 필요하다고 판단되어 flask서버를 추가하였다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

먼저 전체적인 구조로 앱과 웹은 서버 **api**를 통해 데이터베이스에 데이터를 주고받으며 데이터베이스의 데이터를 공유한다. 앱의 초기 상태는 강의실, 공유공간, 카메라 등 모든 데이터가 비워져 있는 상태로 관리자가 관리자 웹에서 데이터를 생성하면 앱은 동적으로 관련 페이지를 생성하도록 개발하였다. 이는 초기 기획에서 단과대학 관리자가 공유공간의 사진이나 필요한 도안을 올리는 방식을 개선하여 관리자가 공간에 대한 전반적인 관리가 용이하게 만들었다.

[앱]

앱에서는 피그마 디자인에 맞추어서 크게 공간대여, 예약 내역, 마이페이지 탭으로 페이지를 구성했다. 이 후에는 혼잡도 확인 기능이 추가되어 총 4개로 구성되었으며 UX를 고려하여 메인페이지에 공간대여, 강의실 대여 버튼을 상단 라벨로 올려서 페이지를 최소화 하였다. 이와 마찬가지로 예약 내역 탭에서도 상단 라벨을 이용하여 데이터들을 분류하고 페이지를 최소화 할 수 있도록 탭 **status**에 따라 **api**를 다르게 호출 하였다.

공유공간대여에서 예약 버튼을 눌렀을때는 공유공간 예약 화면으로 진입하고 강의실 대여에서 예약 버튼을 눌렀을 때는 강의실 예약 화면으로 진입할 수 있도록 **IS_TAP** 상태에 따라 다른 **api**를 호출하며 버튼을 누른 **listview builder**의 **roomname, time, date, table** 등의 정보들을 다음 페이지로 담아 전송했다.

공간 대여 페이지에 진입할때마다 서버로 관리자가 등록한 공간 정보들을 받아올 수 있는 **api**를 호출하여 공간들을 리스트업 하게 구현했다.

예약 페이지에서는 캘린더 라이브러리를 사용하고 시간선택 버튼이나 테이블 선택 버튼은 **listview builder**를 통해 여러개의 버튼을 이어붙여서 구현했다. 이에 따라 버튼이 눌렀는지에 대한 상태, 마감되었는지에 대한 상태를 확인할 수 있는 배열들을 생성해서 사용해야 했으며 테이블 선택 제한에 따른 알고리즘 등이 사용되었다. 모든 내용들을 선택하지 않으면 예약하기 버튼이 동작하지 않도록 설계했으며 강의실 대여 페이지에서는 테이블이 존재하지않고 제출 양식과 함께 따로 승인받아야 하므로 공유공간대여와는 다르게 설계되었다. 강의실 신청 양식은 실제 소프트웨어융합대학 강의실 신청 양식에 의거하여 설계되었으며 여러개의 텍스트 필드와 **signature** 라이브러리를 통해 서명데이터를 저장하고 이를 인코딩하여 서버에 전송할 수 있도록 구현했다. 참여자 명단은 따로 **Participant** 리스트에 정보를 넣어 서명과 함께 추가적으로 전달했다.

혼잡도 페이지는 공간대여페이지와 비슷하게 페이지에 진입할때 **AI-SERVER**로 혼잡도 정보 **api**를 호출하여 위치 별 관리자가 등록한 공간의 혼잡도를 받아와서 UI에 보여줄 수 있도록 구현했다.

예약 내역 페이지에서는 공간대여 예약페이지에서 예약이 완료되었을 때의 페이지에서 정보들을 서버로 요청하여 저장하고 해당 데이터들을 **starttime - endtime**을 지정하여 원하는 기간으로 **api**를 호출하여 예약 예정내역과 예약 완료 내역을 불러올 수 있도록 구현했다. **status**를 세가지로 나누어서 저장했으며 입장하기 전이라면 **previous**, 반납 완료되었다면 **done**, 사용중이라면 **using**

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

으로 상태를 나누었다. 해당 페이지에서는 페이지에 접근했을 때와 입장하기 버튼을 눌렀을 때 현재 시간을 불러와서 입장할 수 있는 상태인지, 입장을 완료했다면 그 시간부터 타이머가 동작하여 남은 시간을 출력할 수 있도록 구현했다. 그 남은 시간내에 반납하지 않는다면 페널티가 부과되며 만약 반납하기를 누른다면 **dialog**에 예시 사진 설명을 보여주고 반납하는 해당 테이블의 반납 예시 사진을 카메라 위에 투명도 **0.3**으로 구현하여 사용자가 카메라로 테이블을 찍어 잘 반납할 수 있도록 구현했다. 해당 테이블들의 예시 사진은 관리자가 공간 등록 당시에 업로드한다. 사진을 찍어 제출하기를 누른다면 **AI-SERVER**로 해당 이미지가 인코딩되어 전송되고 전달받은 **AI-SERVER**에서 디코딩하여 **DB**에 저장, 모델을 통해 청결도를 파악하고 유저의 **DB** 정보에 페널티를 부과한다. 성공적으로 반납되었다면 로컬 알림을 통해 반납되었음을 사용자에게 알리며 반납 과정이 마무리된다.

마이페이지에서는 사용자의 회원정보 수정, 프로필 사진을 넣을 수 있도록 구현을 해두었지만, 해당 기능들은 크게 기능들이 주요하지 않다고 판단되어 준비되지 않았다는 **dialog**를 출력되게 했다. 이에 추가적으로 문의사항 등록을 통해 **date**, **title**, **content**의 내용들을 서버로 문의할 수 있도록 구현했고 이용안내, 로그아웃 기능을 마이페이지에 구현해두었다. 로그아웃 버튼을 누른다면 **signin** 페이지로 이동되며 **status**를 바꾼다.

[웹]

웹 개발 초기에는 데이터베이스인 파이어베이스의 인증키를 **.env** 파일로 만들어 파이어베이스와 직접 연동하여 데이터를 가져오는 형식을 취했으나 서버를 통해서 데이터베이스에 접근하는 방식이 보안 측면에서 뛰어나며 복잡한 요청처리도 서버의 **api**를 통해 쉽게 구현할 수 있다는 점 등 여러 장점이 있다고 판단되어 웹도 서버를 통해 파이어베이스와 연동하기로 하였다.

먼저 웹에 로그인시 관리자 인증 **api**를 통해 관리자 이메일로 로그인하는지 확인 한 후 관리자 이메일과 아이디가 일치하는 경우 파이어베이스 **Authentication**을 이용하여 비밀번호가 일치하는지 판단하고 로그인을 승인해준다. 로그인이 승인 된 경우 **localStorage**에 로그인된 이메일을 저장하여 이후 **api**를 통해 서버에 요청을 보낼때 중요한 정보에 대한 수정이나 삭제하는 기능인 경우 관리자인지 판단하는 함수를 통해 이중으로 보안에 신경쓸 수 있다.

강의실과 공유공간 등 앱에 들어갈 데이터는 기획 단계에서 앱 관리자가 추가하기로 하였지만 관리자 웹을 통한 사진 및 데이터 등록이 관리가 간편하다고 생각되어 공간관리 페이지를 따로 구현하였다. 강의실 추가의 경우 단과대학, 강의실 이름, 사용가능 시간, 사용가능 인원과 사진을 입력하면 관리자는 강의실을 등록하기 전에 등록하려는 사진을 **UI**로 직접 확인할 수 있으며 등록하기 버튼을 누르면 이벤트핸들러 함수를 통해 **base64**로 인코딩되어 **firestore**에 저장되며 이는 어플리케이션 페이지에 동적으로 반영된다.

공유공간의 경우 강의실과 다르게 테이블별로 예약이 가능하기 때문에 등록하는 과정에서 테이블 수를 관리자가 입력하는 경우 테이블 등록하는 팝업에 테이블 수만큼 테이블에 대한 정보를 입력하는 칸이 동적으로 생성되며 관리자가 추가로 공유공간 사진과 공유공간 도안 사진을

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

첨부하면 이벤트핸들러 함수는 사진 두개를 인코딩, 테이블 정보는 테이블별로 json 데이터로 변환하고 나머지 데이터는 모두 firestore에 저장하여 강의실과 마찬가지로 어플리케이션에 동적으로 페이지가 생성되게 처리한다.

저장된 데이터는 다시 웹에서 불러와 목록으로 나열되어 관리자가 필요한 경우 삭제할 수 있도록 강의실 및 공유공간 삭제 api를 통해 서버에 요청을 보낼 수 있도록 하였다. 공유공간, 강의실, 카메라 위치 등 앱에 제공해야할 초기 데이터들은 모두 관리자가 먼저 등록할 수 있도록 수정, 삭제, 등록, 조회 api와 연결된 상태이다.

강의실 예약 조회는 승인처리가 된 강의실 예약은 status를 false로 지정하며 관리자가 신청서를 확인한 후 승인버튼을 누르면 status를 true로 바꿔서 firestore에 보내도록 이벤트핸들러 함수에 api 요청을 넣어놓았다. status가 true로 바뀐 강의실 예약 내역은 승인된 강의실 목록으로 옮겨지며, 승인된 강의실 예약내역의 신청서에서는 승인 버튼이 사라지며 신청서 조회만 가능하도록 해놓았다.

긴급 일정이 생긴경우 강의실 예약을 취소해야 할 수 있으므로 강의실 예약 내역 삭제와 앱 사용자들이 예약할 수 없도록 관리자가 강의실을 예약을 직접 추가할 수 있는 이벤트핸들러 함수도 api 요청과 함께 구현하여 관리자의 요청에 따라 앱에 바로 반영된다.

[서버]

서버 api는 사용자, 강의실 대여, 공유공간 대여로 크게 세가지로 분류하여 구성하려고 하였으나 관리자를 위한 웹을 만들게 되며 알림 및 문의 등 부가적인 기능도 구현해야하기 때문에 사용자, 강의실 대여, 공유공간 대여, 알림, 문의 api를 완성하였다.

또한 관리자 웹을 추가하게 되며 기존 서버를 앱 서버와 웹 서버로 구별하여 api를 구성했다. 관리자 api는 사용자, 강의실, 공유공간, 문의, 알림, 카메라로 총 6개로 분류하여 api를 완성하였다.

[DB]

사용자에게 요청을 받았을 때 uid 값으로 사용자가 등록되어있는지 여부에 대해 먼저 판단한다. 해당 사용자의 정보들을 컬렉션을 통해 받아오고 해당 정보들을 사용하여 컬렉션으로 진입한다. 모든 단과대학 소속의 학생들이 사용할 수 있는 카메라나 공용으로 사용되는 user 컬렉션 등에는 공용 컬렉션으로 빼두어서 모두가 접근 가능하게 구성했고 단과대학 별의 관리가 필요한 강의실이나 공유공간 같은 경우에는 사용자의 정보에서 받아와서 단과대학 별 컬렉션에 진입(소속 강의실 및 공유공간)할 수 있도록 구현하였다. 구성은 컬렉션 - 문서 - 컬렉션 - 문서 순으로 되어있기 때문에 사용자의 예약 흐름에 맞게 구성이 진행되었다.

데이터의 양을 줄이기 위해 예약되지 않은 날짜는 데이터베이스에 올라가지 않게 했고 사용자가 예약 정보들을 요청했을 때 예약된 존재하는 날짜들만 전송하게하여 데이터의 과대한 전송을 줄였다.

 국민대학교 컴퓨터공학부 캠프톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

또한 테이블 도안, 인원 수 등의 다양한 정보들을 참조에 저장함으로써 사용자가 언제든지 정보들을 쉽게 가져올 수 있도록 구현했고 테이블 리스트와 함께 사용중인지 아닌지 구분하기 위한 **status**를 사용했다.

또한 **status**에 따라 **previous, using, done**으로 구분하여 사용자가 이용 예정내역을 요청하는지, 이용 완료 내역을 요청하는지 구분하여 응답이 가능하게 구현했다.

[AI]

인공지능 모델은 공용공간의 혼잡도, 반납시의 관리상태 측정시에 사용되며 이를 수동하기위해 **flask**로 서버를 개발하여 **api**를 구현하였다.

공용공간의 혼잡도는 자율주행스튜디오, 무한상실, 해동 등 공용공간의 혼잡도를 **csrnet**을 이용하여 측정하여 데이터베이스 저장 및 혼잡도를 알려주는 **api**를 만들었다.

반납시의 관리상태측정은 사용자가 사용후 정리정돈을 하고 반납하는지를 점검하기 위해서 사용전 기본 이미지와 반납 이미지를 인공지능 모델을 통해 비교하여 측정하며 점수가 낮게 나왔다고 판단되면 데이터베이스의 유저 정보에 페널티를 주고, 반납시의 이미지를 데이터베이스에 저장하는 **api**를 구현하였다.

2.2.2 시스템 기능 요구사항

[앱]

- 회원가입/로그인/로그아웃: 완료
- 공유공간 대여가능 공간 조회: 완료
- 강의실 대여가능 공간 조회: 완료
- 강의실 예약 마감 조회: 완료
- 공유공간 예약 페이지 구현: 완료
- 공유공간 예약 마감 조회: 완료
- 강의실 예약 페이지 구현: 완료
- 강의실 예약 신청서 페이지 기능 구현: 완료
- 예약 신청 기능: 완료
- 예약 예정 내역 불러오기 기능: 완료
- 예약 완료 내역 불러오기 기능: 완료
- 입장, 사용, 반납 상태 저장 기능: 완료
- 남은 시간 확인 기능: 완료
- 카메라로 공간 반납 기능 구현: 완료
- 혼잡도 UI 출력 기능: 완료
- 마이페이지 문의하기 기능: 완료

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	K-SharePlace
	팀 명	필기아
	Confidential Restricted	Version 2.3 2024-05-23

마이페이지 이용안내 기능 : 완료
 마이페이지 로그아웃 기능 : 완료
 알림 기능 : 완료

[웹]

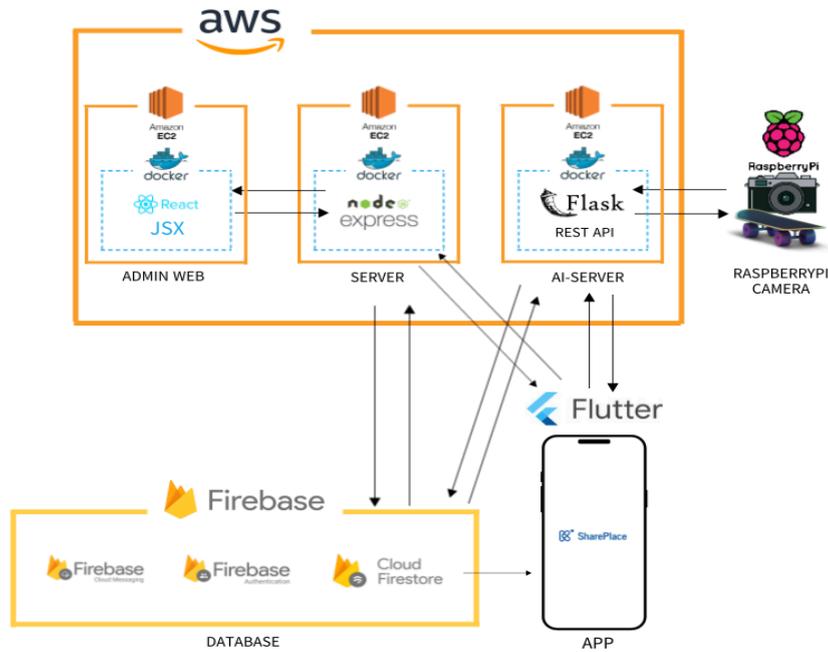
로그인/로그아웃 : 완료
 앱 이용자 회원 조회 : 완료
 회원 개인 알림 전송 : 완료
 예약 확인용 달력 : 완료
 강의실 예약 조회 : 완료
 강의실 예약 삭제 : 완료
 강의실 신청서 조회 : 완료
 강의실 신청서 승인 : 완료
 관리자 강의실 예약 : 완료
 강의실 반납사진 조회 : 완료
 공유공간 예약 조회 : 완료
 공유공간 반납사진 조회 : 완료
 문의 조회 : 완료
 문의 답변 : 완료
 알림 조회 : 완료
 알림 등록 : 완료
 강의실 등록 : 완료

[AI]

혼잡도 추정 모델 : 완료
 반납사진 정리정돈 상태 비교 모델 : 완료

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

2.2.3 시스템 구조 및 설계도



<K-SharePlace 시스템 아키텍처>

기획 단계에서 Flutter 앱을 서버나 데이터베이스를 거치지 않고 React 웹과 연동하여 직접 UI에 대해서 관리하는 방법을 생각해 보았지만, Flutter 앱을 React 웹에서 직접 관리하는 방식은 불가능하다고 판단되며 서버를 활용하여 Flutter에 이용될 데이터를 데이터베이스에서 관리하는 방식은 express를 통해 여러개의 요청을 한번에 처리하는 등 효율성이나 보안성에서 뛰어나다고 판단되어 이런 방식을 취하게 되었다.

앱 사용자의 회원가입 및 로그인 데이터부터 앱 사용시 발생하는 모든 데이터들은 서버의 api 요청에 의해 데이터베이스에 저장되며 웹 관리자가 생성하는 데이터들도 모두 서버의 api 요청에 의해 데이터베이스에 저장된다. 저장된 데이터는 앱과 웹에서 다시 사용할 수 있으며 데이터베이스를 공유하기 때문에 관리자는 데이터베이스에 직접 접근하지 않고 웹 클라이언트 요청에 의해 추가, 수정, 삭제하며 앱을 관리할 수 있다.

AI는 별도의 서버인 Flask를 이용하여 데이터베이스에서 필요한 사진을 전달받거나 모델을 통해서 판단된 score와 라즈베리파이의 작동여부를 전달하게 된다.

모든 데이터들은 서버를 통해서 데이터베이스로 전달되며 데이터베이스는 단과대학 값을 입력

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

받고 단과대학별 데이터를 컬렉션별로 저장하기 때문에 확장성이 뛰어나다.

원활한 버전관리 및 실행을 위해 **docker**를 사용하여 어떤 환경에서도 똑같은 품질의 서비스를 제공할 수 있도록 하였으며 **docker**와 **AWS ec2**를 활용하여 각 프레임워크를 시스템화하여 배포하였다.

<DB 구성>

데이터베이스는 파이어베이스를 사용하며 **Firestore**를 이용하여 웹과 앱은 각종 데이터를 공유한다.

컬렉션은 **Login**, {단과대학}_**classroom**, {단과대학}_**classroom_queue**, {단과대학}_**Inquiry**, {단과대학}_**Club**, **Users**, **Camera**, **notification**으로 구성되어있다.

먼저 **Login** 컬렉션 내부에는 **count**라는 문서가 있으며 필드에는 앱 클라이언트로부터 사용자 로그인 이벤트에 대한 총합을 받아 방문자 수에 대한 데이터를 구현한다.

Camera 컬렉션 내부에는 카메라 위치 정보인 문서 이름 그리고 필드에는 관리자가 등록한 건물 위치와 카메라 작동여부인 **state**와 혼잡도 정보인 **info**가 존재한다. 건물 위치에 대한 정보는 관리자가 등록한 정보이며 **state**와 **info**는 **flask** 서버를 통해 라즈베리파이로 촬영한 영상에 대한 AI모델 **score**를 나타낸다.

{**faculty**} 데이터 값은 관리자가 설정한 단과대학이며 **_classroom**은 미승인된 강의실 예약이며 **_classroom_queue**는 승인된 강의실 예약 내역을 나타낸다 각 컬렉션에는 관리자가 등록된 강의실이 문서로 존재하며 필드에는 강의실 이미지와 사용가능 시간과 사용가능 인원이 나타난다. 내부의 또 다른 컬렉션에 날짜별 예약 내역이 존재하며 문서로는 예약 시간 필드에는 대표자의 정보, 배열로 생성된 여러명의 참여자의 정보, 강의실 예약 신청서에 들어갈 서명 이미지 및 각종 데이터가 존재한다. 승인된 강의실은 **boolAgree**값이 **true**이며 미승인 강의실은 **false**로 지정되어있다. 관리자가 승인하여 **boolAgree**값이 **true**로 변경되면 해당 예약 내역은 **_classroom**으로 옮겨지게 되어있다.

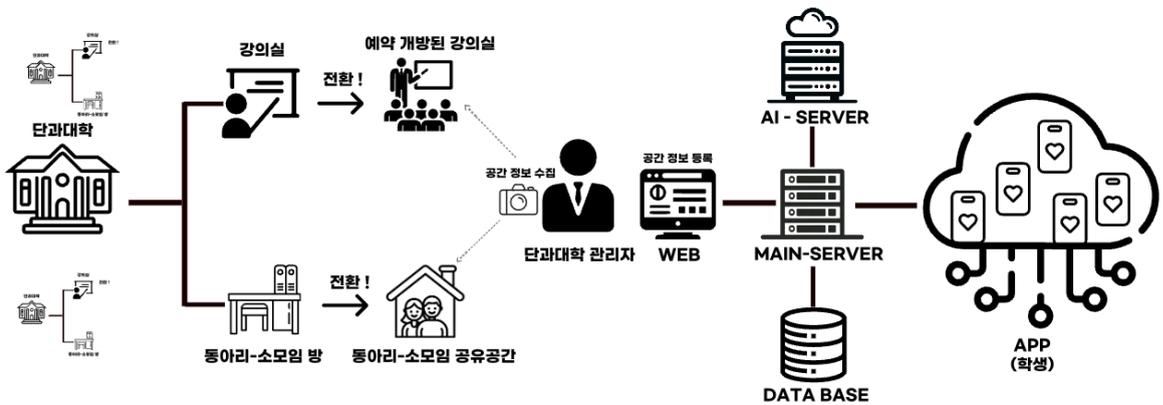
_Club은 관리자가 공유공간을 등록하면 생기는 컬렉션이며 마찬가지로 {**faculty**}에는 관리자가 입력한 단과대학이 들어가며 문서에는 공유공간의 정확한 위치가 나타난다. 공유공간을 최대한 활용하기 위해 모든 테이블을 분리하였으며 각 테이블별 좌석수와 테이블 번호가 **list**형태로 저장되어있다. 강의실 예약내역과 마찬가지로 아래에 있는 컬렉션은 예약날짜를 의미하며 문서는 한시간 단위로 생성된다. 필드에는 각 테이블별 예약 여부가 **bool**값으로 나타나며 사용자가 사용중인지 사용전인지 사용완료한 상태인지 나타내는 **status**값을 포함하고 있다.

notification은 알림과 관련된 데이터를 저장하는 컬렉션으로 **firebase**에서 제공하는 **Fcm** 토큰값을 문서로 사용한다. 하위 컬렉션은 관리자가 공지사항을 올린 날짜로 구성되어있으며 필드는 관리자의 알림 제목과 알림 내용이 각각 **title**과 **body**로 저장된다. **firebase**에서 제공하는 **Fcm** 토큰값을 통해 사용자 개인 혹은 전체에게 효율적으로 알림을 보낼 수 있게된다.

문의 관련 데이터를 포함하는 **_Inquiry**는 사용자 정보에서 {**faculty**}를 가져와 단과대학을

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

나타낸다. 문서로는 사용자의 고유번호인 학번이 나타나도록하여 분류하였고 아래의 새로운 컬렉션에는 문의를 보낸 날짜 필드에는 문의내용, 문의 날짜 및 시간 등 사용자의 정보와 관리자의 답변 시간, 답변, 답변여부는 빈 문자열로 생성된다. 관리자가 답변을 생성하는 경우 response에 답변이 저장되며 response_date에 사용자가 답변하기를 누른 시간 그리고 response_status는 true로 변경되어 저장된다.



<K-SharePlace 시스템 흐름도>

시스템 Flow를 통해 전체적인 흐름을 분석해본다면, 메인 서버인 express를 통해 웹과 앱은 서로 데이터베이스를 공유하며 데이터를 동적으로 다루며 별도의 AI 서버인 Flask를 통해 모델링한 데이터를 앱이 제공 받는다.

단과대학의 관리자는 관리자 웹을 통해 학부생에게 제공할 공간만 등록한다면 공간 활용 시스템을 손쉽게 해당 단과대학에 적용시킬 수 있다.

2.2.4 활용/개발된 기술

[ai 기술]

혼잡도 측정을 위하여 CSRNet을 사용하였다. CSRNet은 높은 품질의 density map을 출력하며 군중 장면에서 사람 수를 추정하는데 초점을 맞춘 딥러닝 모델이며, 군중 밀도추정을위해 위해 설계된 모델이다.

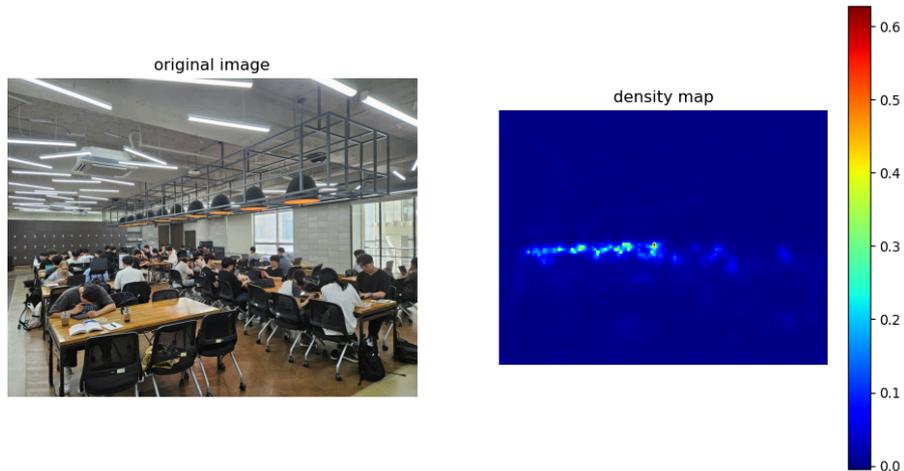
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

자율주행 스튜디오, 무한상상실과 같은 공간의 혼잡도를 구하기 위해 라즈베리파이 카메라를 이용해 실시간 영상을 flask 서버로 보내며 이를 CSRNet모델의 입력으로 사용하여 사람수 및 밀도를 측정하여 혼잡도를 출력으로 받아 사용한다.

카메라에서 혼잡도를 측정할때 정해진 위치에서 라즈베리파이 카메라의 이미지를 CSRNet 모델의 입력으로 주고 출력으로 나온 점수를 여러 테스트를 거쳐 설정한 임계치인 70점 이상시 '매우 혼잡', 70-55점에서는 '혼잡', 55-40 상태에서는 '보통', 40점 이하일때는 '여유'의 상태를 부여하는 방식으로 혼잡도의 측정이 이루어진다.

아래는 자율주행스튜디오에서의 상황별 점수이다.

core: 71.35

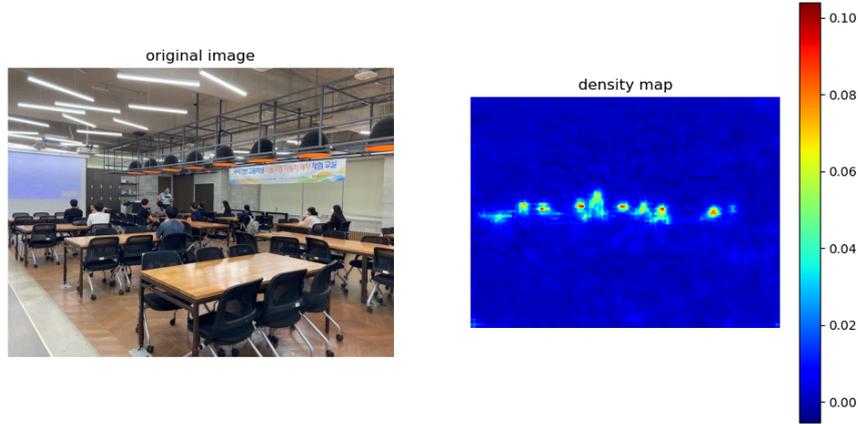


<혼잡도 점수 및 density map>

점수는 72.35점 이며 이는 '매우 혼잡' 상태에 해당한다.

 국민대학교 컴퓨터공학부 캠프스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

core: 34.22



<혼잡도 점수 및 density map>

점수는 34.22점 이며 이는 '여유' 상태에 해당한다.

위의 결과물을 보면 혼잡한 상태에서는 높은 점수가, 여유로운 상태에서는 낮은 점수가 나온것을 볼수 있고 당시 **sota** 모델을 적용한만큼 혼잡도의 수치를 보았을때 좋은 성능을 보이는것을 알 수있다. 또한 새로운 공용공간이 추가 될 경우 라즈베리파이의 카메라를 설치해야 하는데 이때 공간의 크기, 좌석수, 수용인원등을 고려하여 임계치를 쉽게 설정 할 수 있다는 장점이 있다.

이와같이 혼잡도를 구하기 위해서 공용공간에 설치되어있는 라즈베리파이에서 카메라를 이용하여 실시간으로 배포한 **aws**의 주소로 이미지 송신을 진행하며 **aws**의 **flask**서버에서는 실시간으로 이미지를 받아 오며, 각 공용공간에서의 혼잡도를 측정하여 데이터 베이스에 저장하는 **flask api**개발하였다.

반납시의 청소 및 정리정돈의 상태를 측정을 위해서는 사용전 기본이미지와 사용후의 반납 이미지를 비교하는 방식으로 이루어진다. 두개의 이미지 비교를 위해서 **classification**에 사용되며 **backbone**모델도 많이 사용되는 **image net** 으로 **pre-train**되어있는 **vision transformer model**, **inception v2**, **resnet34**의 세가지 모델을 이미지 비교를 위해 사용하였다. 직접 학습을 시키는것 보다 **image net**으로 학습되어있는 모델은 일반화 성능이 좋기 때문에 동아리방이 추가될때마다 학습을 따로 시킬 필요가 없어서 앱의 특성상 확장성이 좋아서 단과대 별로 동아리방과 강의실이 계속 추가 될 수 있고, 추가될때마다 학습을 다시 시키는것이 비효율적인 반면 이미 일반화 성능이 좋은 모델을 이용하면 어떤 책상에서 비교하더라도 좋은 성능을 보인다는 장점이 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

classification 모델을 이용하여 이미지를 비교하기 위해서는 기본 이미지, 반납 이미지를 각 모델에 입력으로 주었고 출력으로 나온 임베딩 벡터의 **cosine similarity**를 구하여 사용하였다. 각 모델에서의 **cosine similarity** 점수를 여러 상황, 상태에서 비교를 하여 각 모델별 임계치를 설정하였으며 세가지의 모델을 사용한 이유로는 vit 모델에서는 쓰레기, 옷등이 추가되었을때 매우 낮은 점수가 나오고, **resnet34** 모델에서는 이미지의 각도가 달라질때 비교적 낮은 점수를 주는 것으로 보아 하나의 모델을 사용하여 임계치를 설정하는것이 아닌 앙상블 기법을 적용하여 세가지의 모델의 출력값중 2가지 이상에서 임계치를 넘을경우 청소가 되었다고 판단하며 2개 이상에서 넘지 못했을 경우에는 청소가 되지 않았다고 판단하는 알고리즘을 구현하였다.

이처럼 모델만을 사용하는것이 아닌 앙상블기법 까지 적용하여 모델의 성능을 극대화 하였다.

아래의 이미지는 전 후 사진의 비교 및 각 모델의 점수 이다.



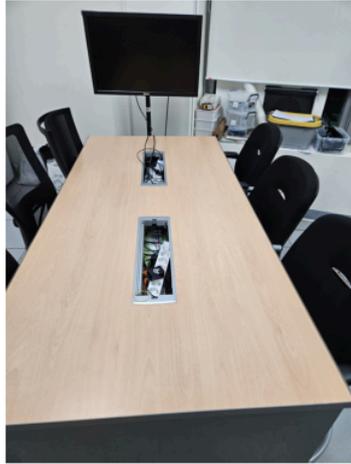
<비교하는 이미지 및 각 모델의 점수>

반납시 찍은 이미지가 기존의 이미지와 유사한 경우 높은 점수를 받는것을 알 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	펼기아	
	Confidential Restricted	Version 2.3	2024-05-23

vit: 0.902

Before Image



resnet: 0.928

After Image



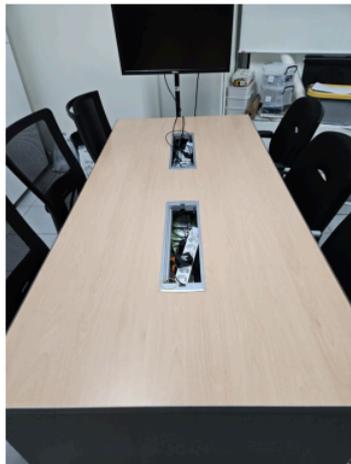
inc: 0.878

<비교하는 이미지 및 각 모델의 점수>

반납시 찍은 이미지가 각도가 많이 달라도 높은 점수를 받는것을 알 수 있다.

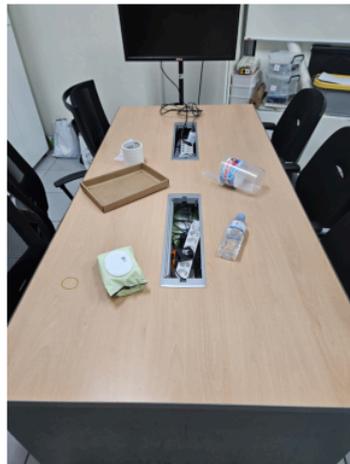
vit: 0.574

Before Image



resnet: 0.682

After Image



inc: 0.601

<비교하는 이미지 및 각 모델의 점수>

유사한 구도에서 이미지를 찍어도 정리가 안되어 있으면 낮은 점수를 받는것을 알 수 있다.

 국민대학교 컴퓨터공학부 캠프스톤 디자인 I	결과보고서	
	프로젝트 명	K-SharePlace
	팀 명	필기아
	Confidential Restricted	Version 2.3

각 모델의 임계치로는 vit : 0.7, resnet : 0.8, inception : 0.75 이며 위의 이미지를 비교했을때 반납시 정리가 잘 되어있고 이미지를 잘 찍은 경우, 정리는 잘 되었지만 이미지의 각도가 조금 다른 경우에는 모두 임계치를 여유롭게 넘는 것을 알 수 있으며 반납시 이미지를 각도에 맞게 잘 찍었지만 쓰레기를 치우지 않은 경우에는 세 모델 전부 임계치를 넘지 못하는 모습을 보여준다. 이처럼 모델의 결과는 매우 좋은 모습을 보인다. 하지만 불을 끄고 어두운 상태에서 반납 이미지를 찍는 것과 같이 기본 이미지와 반납 이미지의 밝기 차이가 큰 경우 성능 부분에서 아쉬움을 보인다.

이러한 방법으로 책상의 기본 이미지, 반납시의 이미지를 각 모델의 입력으로 주며 사용후 반납상태를 측정하며 반납 후 깨끗하게 사용하지 않았다고 판단되면 데이터베이스에서 유저에게 페널티를 부여하는 flask api를 개발하였다.

<앱>

2.2.5 결과물 목록

<관리자 웹 >

K-SharePlace 관리자 웹

관리자 아이디

비밀번호

[Login](#)



<관리자 웹 로그인 화면: 관리자 로그인 가능>

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23



<관리자 웹 home 화면: 전체 예약 및 방문자 수 요약 및 그래프 나열>

회원관리

학번 또는 이름 검색

이름	학번	단과대	메일	연락처	알림보내기
정의석	20203134	소프트웨어융합대학	wjddmltjr07@gmail.com	010-5662-8175	알림
안수현	20191444	소프트웨어융합대학	saker123@google.com	0105731838	알림
안수순	20191621		saker123@naver.com	01057317909	알림
이재훈	20213053	소프트웨어융합대학	jhoon5061@gmail.com	00000	알림
Admin	00000000	Admin	react@kookmin.ac.kr	Admin	알림
안수현	20195959	소프트웨어융합대학	saker123@kookmin.com	09192322	알림

<관리자 웹 회원관리 페이지 화면: 회원 조회 및 알림 보내기 기능>

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

SharePlace 로그아웃

home < > 2024년 5월 > >

회원 관리	월	화	수	목	금	토	일
예약 확인	29일	30일	1일	2일	3일	4일	5일
예약 관리							
문의 관리	6일	7일	8일	9일	10일	11일	12일
알림 관리	13일	14일	15일	16일	17일	18일	19일
공간 관리							
혼잡도 관리	20일	21일	22일	23일	24일	25일	26일
	27일	28일	29일	30일	31일	1일	2일

<관리자 웹 예약확인 페이지 화면: 특정 날짜 클릭시 해당 날짜의 예약 내역 표시>

SharePlace 로그아웃

home

승인 강의실 예약내역 *강의실 예약 추가하기							
날짜	대표자 이름	대표자 학번	강의실 이름	예약 시간	신청서	반납 사진 확인	삭제하기
2024-05-17	안수현	20195959	미래관 231호	10시 ~ 12시	<input type="button" value="신청서 보기"/>	<input type="button" value="반납 사진"/>	<input type="button" value="삭제하기"/>
2024-05-17	안수현	20195959	미래관 231호	10시 ~ 12시	<input type="button" value="신청서 보기"/>	<input type="button" value="반납 사진"/>	<input type="button" value="삭제하기"/>
2024-05-17	안수현	20195959	미래관 231호	12시 ~ 14시	<input type="button" value="신청서 보기"/>	<input type="button" value="반납 사진"/>	<input type="button" value="삭제하기"/>
2024-05-17	안수현	20195959	미래관 231호	12시 ~ 14시	<input type="button" value="신청서 보기"/>	<input type="button" value="반납 사진"/>	<input type="button" value="삭제하기"/>

미승인 강의실 예약내역						
날짜	대표자 이름	대표자 학번	강의실 이름	예약 시간	신청서	
2024-05-24	안수현	20191444	미래관 231호	18시 ~ 21시	<input type="button" value="신청서 보기"/>	
2024-05-24	안수현	20191444	미래관 231호	18시 ~ 21시	<input type="button" value="신청서 보기"/>	
2024-05-24	안수현	20191444	미래관 231호	18시 ~ 21시	<input type="button" value="신청서 보기"/>	

<관리자 웹 예약관리 페이지 화면: 미승인 예약에 대한 승인 및 반납사진 확인, 예약추가,삭제>

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

소프트웨어융합대학 강의실 신청서

구분	미래관 231호	
사용일시	2024-05-17 10시 ~ 12시	
사용목적	확인	
대표자 / 책임자	학번	20195959
	성명	안수현
	연락처	09192322
	E-Mail	saker123@kookmin.com
서명		
학번	2019221	
소속	소용	
성명	안수현	
서명		

<관리자 웹 예약관리 페이지-신청서>

공지사항

등록날짜	공지분류	공지내용	공지 관리
2024. 5. 21.	긴급	강의실 세미나로 금일 18시 이후 예약은 취소하겠습니다	삭제
2024. 5. 21.	일반	금일 12시부터 캡스톤 관련 전시가 있을 예정이니 많은 참여 부탁드립니다	삭제
2024. 5. 21.	긴급	5월 31일 캡스톤 시상식 일정으로 611호 강의실 예약은 일관 취소하겠습니다	삭제

<관리자 웹 알림관리 페이지 화면: 공지사항 등록 및 삭제>

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23



home

회원 관리

예약 확인

예약 관리

문의 관리

알림 관리

공간 관리

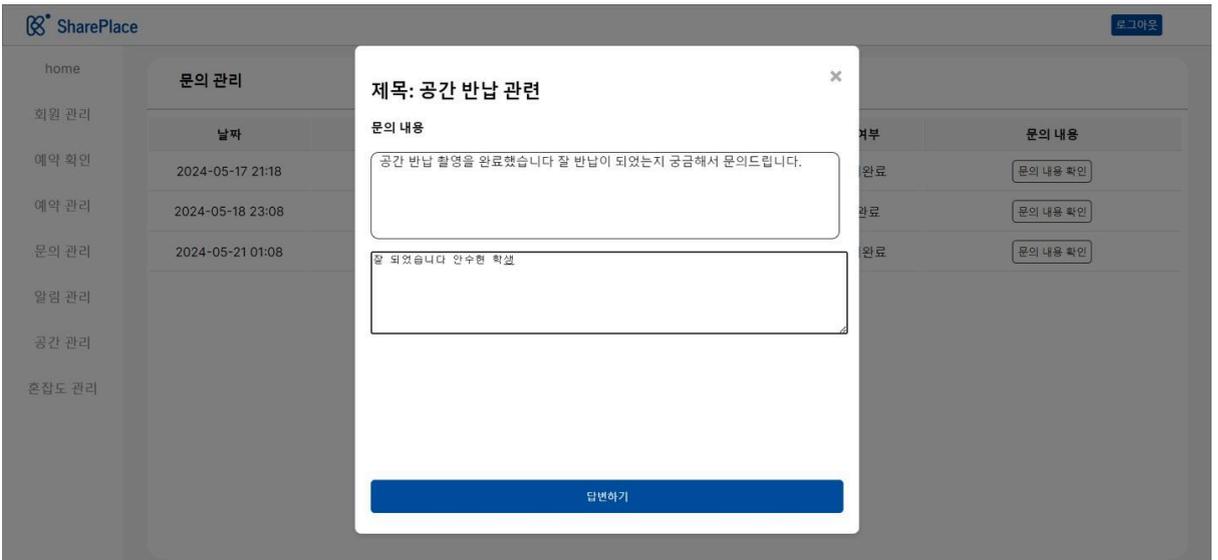
혼잡도 관리

로그아웃

문의 관리

날짜	이름	학번	단과대학	답변 여부	문의 내용
2024-05-17 21:18	안수현	20191444	소프트웨어융합대학	답변 미완료	문의 내용 확인
2024-05-18 23:08	안수현	20191444	소프트웨어융합대학	답변 완료	문의 내용 확인
2024-05-21 01:08	안수현	20191444	소프트웨어융합대학	답변 미완료	문의 내용 확인

<관리자 웹 문의관리 페이지 화면: 회원 문의 조회 및 내용확인 후 답변>



home

회원 관리

예약 확인

예약 관리

문의 관리

알림 관리

공간 관리

혼잡도 관리

로그아웃

문의 관리

날짜	이름	학번	단과대학	답변 여부	문의 내용
2024-05-17 21:18	안수현	20191444	소프트웨어융합대학	답변 미완료	문의 내용 확인
2024-05-18 23:08	안수현	20191444	소프트웨어융합대학	답변 완료	문의 내용 확인
2024-05-21 01:08	안수현	20191444	소프트웨어융합대학	답변 미완료	문의 내용 확인

제목: 공간 반납 관련

문의 내용

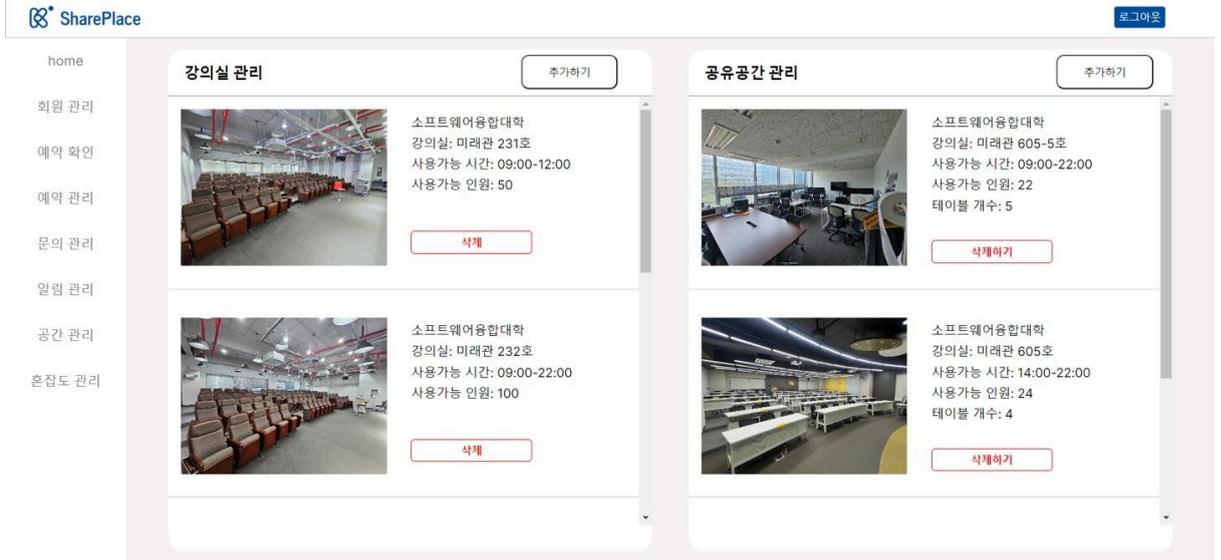
공간 반납 촬영을 완료했습니다 잘 반납이 되었는지 궁금해서 문의드립니다.

잘 되었습니다 안수현 학생

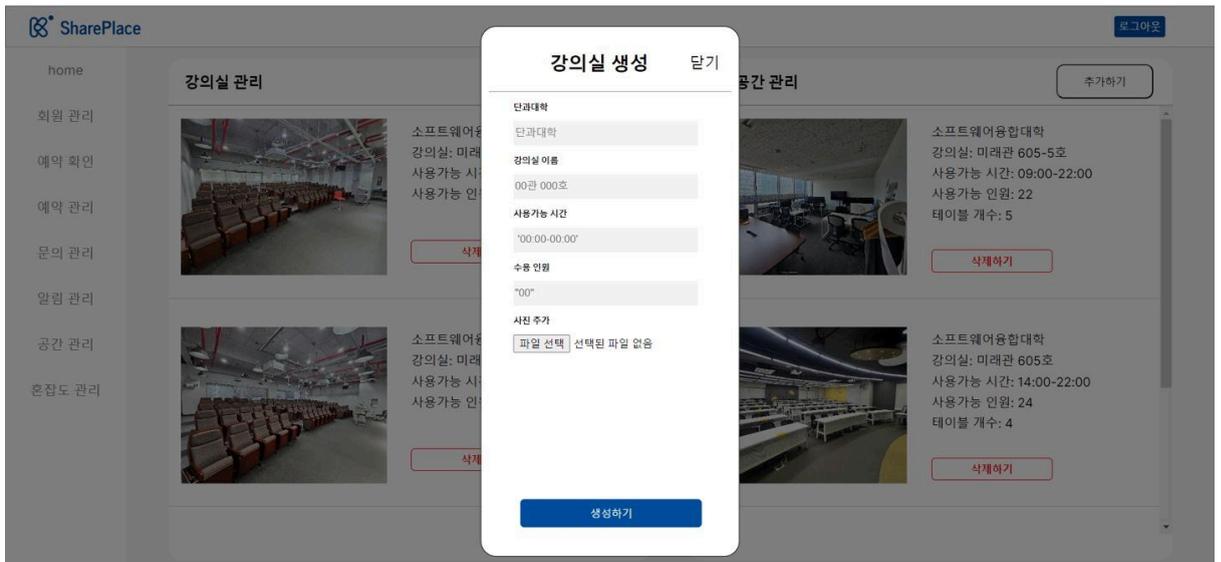
답변하기

<관리자 웹 문의관리 페이지 화면-문의 답변>

 국민대학교 컴퓨터공학부 캠퍼스 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23



<관리자 웹 공간관리 페이지 화면: 강의실 및 공유공간 등록, 삭제>



<관리자 웹 공간관리 페이지-강의실 등록>

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

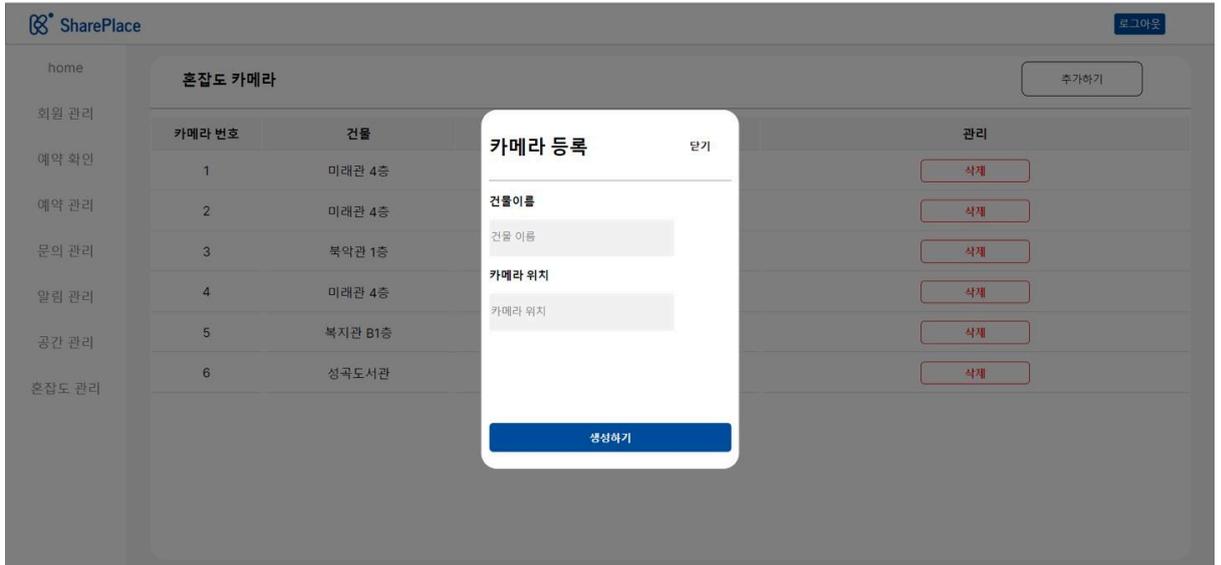


<관리자 웹 공간관리 페이지-공유공간 등록>



<관리자 웹 혼잡도관리 페이지 화면: 카메라 실시간 작동 여부 조회, 위치 등록 및 삭제>

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	K-SharePlace
	팀 명	필기아
	Confidential Restricted	Version 2.3



<관리자 웹 혼잡도관리 페이지-카메라 등록>

<앱>



국민대학교
컴퓨터공학부
캡스톤 디자인 I

결과보고서

프로젝트 명

K-SharePlace

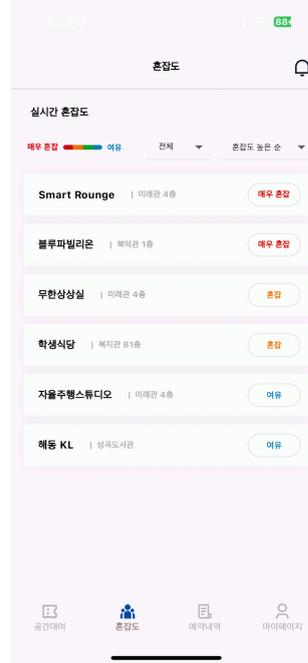
팀 명

필기아

Confidential Restricted

Version 2.3

2024-05-23





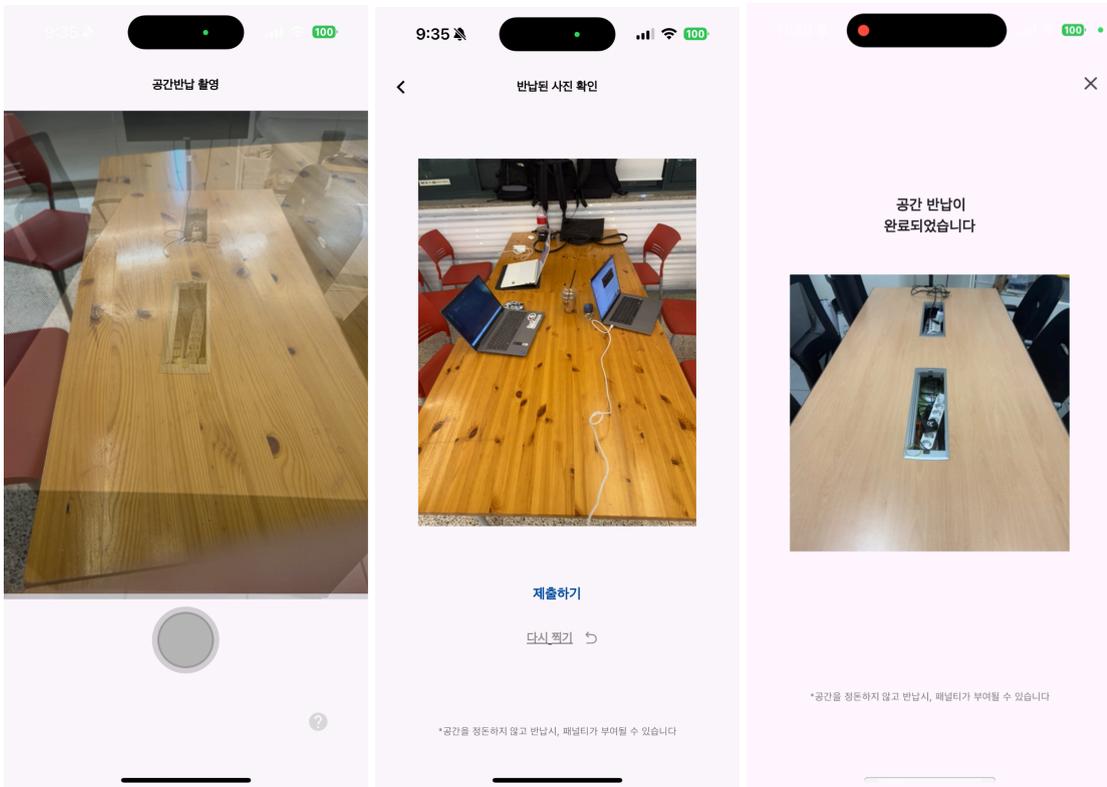
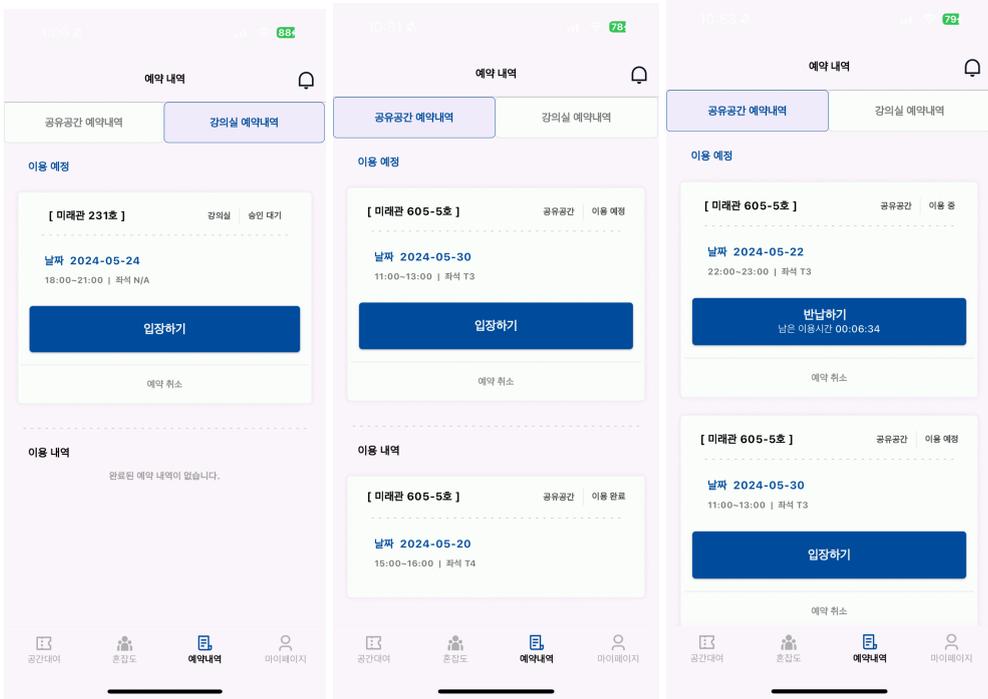
국민대학교
컴퓨터공학부
캡스톤 디자인 I

결과보고서

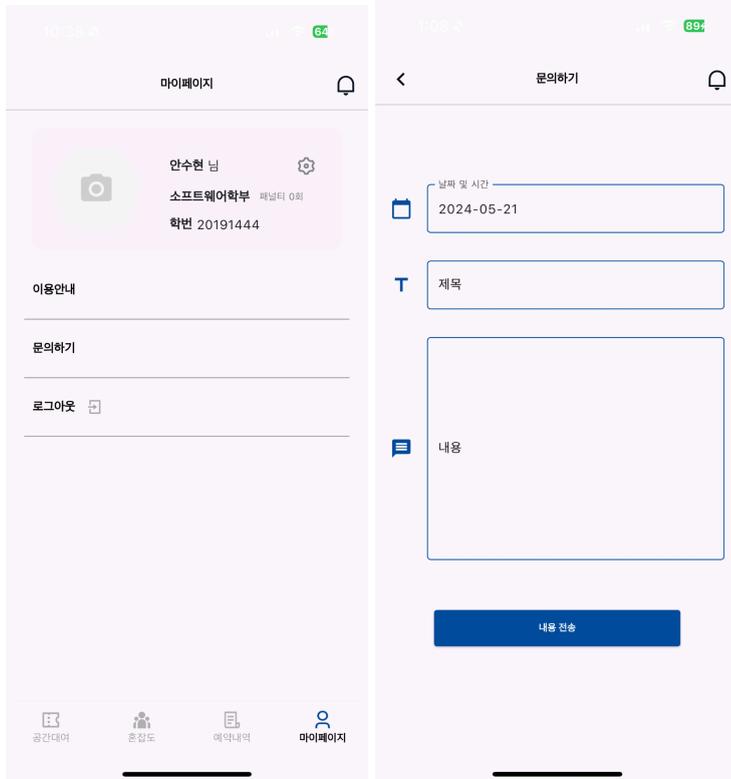
프로젝트 명	K-SharePlace	
팀 명	필기아	
Confidential Restricted	Version 2.3	2024-05-23



 국민대학교 컴퓨터공학부 캠퍼스 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23



 국민대학교 컴퓨터공학부 캠퍼스 디자인 I	결과보고서	
	프로젝트 명	K-SharePlace
	팀 명	필기아
	Confidential Restricted	Version 2.3



2.3 기대효과 및 활용방안

K-SharePlace는 앱을 사용하는 학부생들뿐만 아니라 해당 시스템을 사용하는 단과대학의 관리자, 그리고 단과대학 자체에 긍정적인 효과를 기대할 수 있다.

K-SharePlace(이하 시스템)를 기획하고 개발하며 초점을 맞췄던 가장 궁극적인 목표는 고질적으로 해결되지 않던 공간 부족 문제에 대한 해결이다.

시스템을 사용함으로써 단과대학의 공간부족 문제가 해결 될 경우 학부생들은 각종 집단 활동이 필요하여 강의실을 빌려야하는 경우 과 사무실을 들러 양식을 받아 모든 참여자의 서명을 받고 제출해야했던 이전 과정보다 훨씬 간편하게 강의실을 대여할 수 있으며 교내 공간 내에 공용 공간들에 혼잡도 확인을 제공함으로써 인구 밀집도를 피할 수 있는 정보 제공이 가능하다. 학부생들은 학교 내부의 공간들을 최대한 활용하여 학교 생활을 할 수 있게 되며 이것은 공간 부족에 대한 갈등을 완화할 수 있을 것이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	펼기아	
	Confidential Restricted	Version 2.3	2024-05-23

또한 공유공간에 대한 확장도 다양한 효과를 기대해볼 수 있다. 학술 동아리는 학부에서 미처 배우지 못하거나 자신이 더 배우고싶은 분야에 대한 욕구를 해소할 수 있는 장소이자 학부 수업 외에 여러 프로젝트를 진행하며 자신의 포트폴리오를 만들고 사회로 진출하기 전 경쟁력을 갖추 수 있는 수단으로도 사용되곤 한다. 또한 동아리 내부에서의 소속감은 동아리 구성원 사이의 유대감을 형성하여 서로를 이끌어주는 긍정적인 효과를 기대할 수 있다.

동아리 방은 각종 동아리 활동을 할 장소를 제공하는 것 뿐만 아니라 동아리 구성원 간의 친목을 형성할 기회도 제공한다.

기존 8개의 동아리 중 4개의 동아리만 사용하던 동아리 방을 공유공간 시스템을 통해 동아리에 소속된 누구든 소속원들과 함께 동아리 활동을 진행할 수 있으며 현재 4개의 동아리만 독점하던 불평등을 해소할 수 있을 것으로 예상된다.

기존 동아리와 신설 동아리 간의 불평등이 해소된다면 새로운 학술 동아리 및 다양한 동아리가 만들어질 발판이 될 것이며 이는 학부생들이 더욱 다양한 분야에 대해서 공부하고 경험 할 기회를 제공하며 자신이 원하는 분야의 학술 동아리가 없거나 인원 초과로 인해 학술 동아리에 참여하지 못하는 사례도 적어질 것으로 예상된다.

관리자의 기대효과도 존재한다. 시스템을 설계하는 과정에서 앱의 관리자를 단과대학 내부의 과 사무실 직원이 직접 관리하는 것을 가정하고 개발하였다. 따라서 앱의 초기상태는 강의실과 공유공간 그리고 카메라 위치에 대한 정보가 들어있지 않다. 관리자는 웹을 통하여 앱의 데이터를 등록하고 수정, 삭제 가능한데 굉장히 직관적이고 간결하게 만들어 놓았기에 코딩에 대한 전문적인 지식이 없는 직원도 앱을 용이하게 관리할 수 있다.

또한 웹을 통해서 앱을 가입한 모든 학생들의 연락처 등 개인정보를 볼 수 있으며 필요한 경우 공지사항을 올리거나 개인 메시지를 보내는 등 강의실 예약을 긴급한 상황으로 인해 취소해야 할 경우 모두 웹으로 처리가 가능하다. 이는 시스템을 단과대학별로 도입하기 쉽게 만들어 공간 부족 문제 해결에 대한 시스템이 필요한 어느 단과대학이든 쉽게 적용시킬 수 있어 소프트웨어융합대학 뿐만 아니라 다양한 단과대학의 공간 부족 문제를 해결 할 수 있는 기대효과가 있다.

공간 부족의 문제는 미래에도 다시 언급되고 더욱 심화될 문제이다. 계속해서 소수의 동아리만 공간을 독점하게 된다면 소프트웨어융합대학 내의 동아리 및 소모임의 발전은 기대할 수 없다. 이미 동아리방 부족 문제로 교수와 학생들이 논의했을 당시 공유공간으로 전환할만한 시스템과 정책 등이 뒷받침 되지 않아서 수포로 돌아갔던 이력이 있다.

만약 이 프로젝트가 어느 한계에 의해 바로 실사용되지 못하더라도 K-SharePlace는 이러한 문제들이 다시 논의되고 심화되었을 때 선례와 기반이 될 수 있는 프로젝트가 되는 것을 목표로 하고있다. 당장 실사용하는 데에는 어려움이 있겠지만 그러한 문제들에 도움이 될 수 있는 이력을 남기고 방안을 제시했다는 것 자체만으로도 캡스톤 프로젝트의 방향성을 충족시켰다고 볼 수 있다. 팀 프로젝트의 최종 목표는 단과대학별로 쉽게 적용시킬 수 있는 앱/웹 시스템을 개발하여 국민대학교 내부의 공간 부족 문제를 해결하는 것이다.

 국민대학교 컴퓨터공학부 캠퍼스 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

특정 단과대학에서 해당 정책과 **K-SharePlace** 시스템이 필요한 경우, 관리자는 관리자 계정 받아 관리자 웹에 로그인 할 수 있다. 관리자는 해당 단과대학에서 학부생들이 사용할 수 있는 공간에 대해 조사한 후 공간관리 페이지를 이용하여 강의실 및 공유공간에 대한 데이터를 등록한다. 관리자가 최초로 공간을 등록할때 단과대학에 대한 정보가 저장되며 단과대학별로 데이터베이스에 저장되기 때문에 데이터의 충돌과 섞이는 상황이 연출되지 않는다.

관리자를 통해 생성된 데이터를 통해 앱에는 강의실 및 공유공간에 대한 페이지가 동적으로 생성된다. 이후 해당 단과대학의 학부생들이 앱에서 회원가입을 진행하면 강의실 및 공유공간을 예약할 수 있다.

알림, 문의, 공지사항 등 부가적인 기능으로 관리자와 학부생들은 소통할 수 있으며 필요한 경우 관리자는 단과대학 주변의 자율 공간에 라즈베리파이를 설치하여 학부생들에게 자습할 공간에 대한 혼잡도 정보를 제공해줄 수 있다.

위와 같이 시스템 적용이 굉장히 간편하며 관리자와 사용자 모두 윈윈할 수 있다. 학교 측 관리자는 쉽게 관리가 가능하며 공간 부족에 대한 목소리에 대응할 수 있으며 학생들은 쉬운 예약과 정보제공을 통해 공간 부족에 대한 갈증 해소가 가능하다.

3 자기평가

안수현 : 프로젝트의 전반적인 흐름이나 설계를 이끌었는데 데이터베이스 관리에 대한 능숙하지 못한 부분들이 있음에 아쉬움이 남는다. 나머지 앱 디자인 및 개발은 생각했던 것보다 좋은 성과를 내어 만족스러웠고 풀스택 개발을 맡으면서 예약관련 프로젝트의 완성도를 높이는 것이 쉬운 일이 아니라는 것을 깨달았고 최선을 다했기에 후회가 없다. 좋은 경험이 된 프로젝트였다고 생각한다.

정일형 : 초기에 계획했듯이 어느 단과대학이든 시스템을 쉽게 적용하여 대학교 내부의 전체적인 공간 부족 문제를 해결하고자 하였다. 관리자 웹은 간결한 디자인과 쉬운 사용방식을 통해 확장성을 늘려 어느 단과대학에서든 쉽게 사용 가능하게 하였으며 앱은 강의실 대여에 필요한 신청서와 서명 그리고 각종 예약에 대한 기능을 구현하여 앱 사용자가 손쉽게 공간 대여에 접근할 수 있도록 하였다. 추가로 AI 모델은 혼잡도와 정리정돈 여부를 판단하며 사용자와 관리자에게 편의를 제공한다.

위와 같은 기능들은 실제로 적용될 경우 부족한 공간이 만든 불편함을 해소할 것으로 예상된다.

이재훈 : 앱에서 인공지능 모델을 적용하기위해 ai모델 및 ai 서버를 맡아서 하였는데 모델의 성능과 기능적으로는 잘 구현되어서 만족스러웠지만 인공지능을 적용하는데 있어서 개발 초기에 생각했던 모델과 알고리즘이 아닌 기능에 맞는 모델을 찾고 적용시키는 부분에서 인공지능의 식견이 넓지못해 적용할 모델 및 알고리즘을 찾는데 오래 걸렸던 점이 아쉬웠다.

 국민대학교 컴퓨터공학부 캠프톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

프로젝트를 하면서 초기에 구상했던것처럼 흘러가지 않아서 바뀐것이 많지만 많은것들을 경험할수 있어서 좋은 경험이 될것 같다.

4 참고 문헌

번호	종류	제목	출처	발행년도	저자	기타
1	웹페이지	Cloud Firestore docs				
2	웹페이지	Firebase cloud messaging docs				
3	웹페이지	모달 : 부트스트랩				
4	기사	교내 공간 문제, 시급한 대책 마련이 필요한 때	국민대학교 신문사	2023	이승준 기자	
5	웹페이지	[React]이미지 업로드, 프리뷰				
6	기사	예나 지금이나 존재하는 공간부족 문제	국민대학교 신문사	2012	국민대 학교 신문방 송사	
7	논문	CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes	cvpr	2018	Yuhong Li, Xiaofan Zhang , Demin Chen	

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

8	웹 페이지	Image Similarity Comparison using VGG16 Deep Learning Model	Image Similarity Comparison using VGG16 Deep Learning Model by Roman Medium			
9	웹 페이지	[AWS] EC2 인스턴스에 Docker 컨테이너 배포하기	[AWS] EC2 인스턴스에 Docker 컨테이너 배포하기 (tistory.com)			

5 부록

5.1 사용자 매뉴얼

안내사항으로 서버는 이미 배포되어있어서 따로 설치가 필요 없지만 aws의 서버 실행시에 사용자가 사용할 수 있다. 배포된 서버가 아닌 로컬 서버 작동 방법은 아래에 설명해두었다.

먼저 앱 사용자의 앱 사용 매뉴얼은 다음과 같다.

/2024 capstone/frontend/flutter 경로에 Flutter 3.22.0 , Dart 3.4.0 버전의 SDK 설치

1. git clone <https://github.com/kookmin-sw/capstone-2024-26.git>

2. cd frontend

3. flutter pub get 명령어 실행

4. ctrl(command)+ shift + p

[에뮬레이터로 실행]

5. Flutter : Launch emulator에서 IOS Simulator(IOS 17.5) 선택

6. main.dart 파일에서 flutter run 명령어 실행

[디바이스로 실행]

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	K-SharePlace
	팀 명	필기아
	Confidential Restricted	Version 2.3 2024-05-23

5. Flutter : Select Device에서 연결된 기기(IOS) 선택

6. main.dart 파일에서 flutter run 명령어 실행

다음은 웹 사용자(관리자)의 실행 가이드이다. 따로 로컬에서 설치하지 않는 경우 이는 aws에서 웹 및 서버가 켜져 있을 시 url 13.209.30.139:3000로 접속 하여 실행이 가능하다.

다음은 웹 사용자(관리자)의 실행 가이드인데 로컬에서 설치하여 사용하는 상황을 보여주고 있다.

```
git clone https://github.com/kookmin-sw/capstone-2024-26.git
```

다음 명령어로 깃허브에서 주소를 클론하여 가져온다.

다음은 도커 이미지를 빌드하는 과정이며 도커가 설치되어 있어야한다.

```
cd react_admin
```

```
docker build -t [your docker image name] .
```

```
docker run -p 3000:3000 [your docker image name]
```

이후 localhost:3000으로 접속하여 사용이 가능하다.

5.2 운영자 매뉴얼

1. 시스템 구성환경

앱 프레임워크 : flutter , Dart 버전 : 3.22.0, 3.4.0 SDK 설치

웹 프레임워크 : reactjs, react 버전 : 18.2.0, node 버전 : 20.10.0

MAIN_SERVER : nodejs, 버전 : 20.10.0

AI_SERVER : flask, 버전 : 3.0.3

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

라즈베리파이 : 라즈베리파이 4b 및 카메라

2. 설치 및 실행 가이드

프로젝트를 사용하기 위해서는 깃허브의 주소를 클론하여 가져온다.

```
git clone https://github.com/kookmin-sw/capstone-2024-26.git
```

(실행 전 주소를 각자의 환경에 맞는 주소로 변경)

다음은 flutter 설치 및 실행 과정이다.

1. git clone https://github.com/kookmin-sw/capstone-2024-26.git

2. cd frontend

3. flutter pub get 명령어 실행

4. ctrl(command)+ shift + p

[에뮬레이터로 실행]

5. Flutter : Launch emulator에서 IOS Simulator(IOS 17.5) 선택

6. main.dart 파일에서 flutter run 명령어 실행

[디바이스로 실행]

5. Flutter : Select Device에서 연결된 기기(IOS) 선택

6. main.dart 파일에서 flutter run 명령어 실행

다음은 react 및 express 실행을 위한 과정이며 node 20.10.0 버전이 설치 되어 있어야한다.

```
cd react_admin
```

설치 : npm install

(실행 전 주소를 각자의 환경에 맞는 주소로 변경)

실행 : npm start

다음은 express 설치 및 실행하는 과정이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	K-SharePlace
	팀 명	필기아
	Confidential Restricted	Version 2.3 2024-05-23

cd backend

설치 : npm install

(비밀키값을 받아 .env에 등록)

실행 : npm start

다음은 flask 설치 및 실행 방법이다.

cd ai

설치 : pip install -r requirements.txt

(비밀키값을 받아 .env에 등록)

실행 : python app.py

다음은 라즈베리파이 및 카메라를 설정하는 과정이다.

ai/faspi_flask 의 코드를 복사하여 라즈베리파이에서 실행한다.

운영자는 파이어베이스에 프로젝트를 생성하여 비밀키를 기입하고 이를 등록해야

사용이 가능하다.

.env 파일의 의 구성은 다음과 같다.

```

FLUTTER_APP_apikey=YOUR_FLUTTER_APP_apikey

FLUTTER_APP_authDomain=YOUR_FLUTTER_APP_authDomain

FLUTTER_APP_projectId=YOUR_FLUTTER_APP_projectId

FLUTTER_APP_storageBucket=YOUR_FLUTTER_APP_storageBucket

FLUTTER_APP_messagingSenderId=YOUR_FLUTTER_APP_messagingSenderId

FLUTTER_APP_appId=YOUR_FLUTTER_APP_appId

FLUTTER_APP_measurementId=YOUR_FLUTTER_APP_measurementId

TYPE=YOUR_TYPE

PROJECT_ID=YOUR_PROJECT_ID

```

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서	
	프로젝트 명	K-SharePlace
	팀 명	필기아
	Confidential Restricted	Version 2.3 2024-05-23

```

PRIVATE_KEY_ID=YOUR_PRIVATE_KEY_ID

PRIVATE_KEY=YOUR_PRIVATE_KEY

CLIENT_EMAIL=YOUR_CLIENT_EMAIL

CLIENT_ID=YOUR_CLIENT_ID

AUTH_URI=YOUR_AUTH_URI

TOKEN_URI=YOUR_TOKEN_URI

AUTH_PROVIDER_X509_CERT_URL=YOUR_AUTH_PROVIDER_X509_CERT_URL

CLIENT_X509_CERT_URL=YOUR_CLIENT_X509_CERT_URL

UNIVERSE_DOMAIN=YOUR_UNIVERSE_DOMAIN

```

5.3 배포 가이드

react, express, flask 3개의 서버를 도커로 aws ec2에 배포하였다.

첫번째로 react 도커 aws 배포 방법이다.

우선 aws ec2에서 프로젝트 및 도커를 설치하고 다음 명령어를 실행한다.

```
cd react_admin
```

```
빌드 : docker build -t [your docker image name] .
```

```
실행 : docker run -p 3000:3000 [your docker image name]
```

두번째로 express 도커 aws 배포 방법이다.

aws ec2에서 프로젝트 및 도커를 설치하고 다음 명령어를 수행한다.

```
cd backend
```

```
.env키 등록
```

```
빌드 : docker build -t [your docker image name] .
```

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

실행 : docker run -p 3000:3000 [your docker image name]

세번째로 flask 도커 aws 배포 방법이다.
 gpu(cuda 12.1)가 있는 aws ec2인스턴스를 생성하고
 aws ec2에서 프로젝트 및 도커를 설치한다.
 마지막으로 다음 명령어를 수행한다.
 cd ai
 .env키 등록

빌드 : docker build -t [your docker image name] .

실행 : docker run -p 5000:5000 [your docker image name]

위의 방법과 같이 수행하면 aws에 배포한 도커로 실행이 가능해진다.

5.4 테스트 케이스

대분류	소분류	기능	테스트 방법	기대 결과	테스트 결과
웹	웹	현재 열린 파일을 저장한다.	상단 메뉴바에서 [파일] > [저장] 버튼을 누르면, 1) 파일이 저장된 적이 없으면, 파일 저장 다이얼로그가 열리고 원하는 파일명을 입력 후, 저장 버튼을 클릭한다. 2) 기존에 저장된 적이 있으면, 별도의 액션이 일어나지 않는다.	지정된 위치에 해당 내용의 파일이 저장된다.	성공
웹	로그인	관리자 로그인	생성된 웹페이지에 이메일과 비밀번호를 입력한다	관리자 이메일인지 확인한 후 파이어베이스를 통하여 비밀번호가 맞는지 확인 후 로그인 실행.	성공
웹	Home	예약 확인	로그인 후 가장 먼저 실행되는 페이지	1) 승인 강의실 예약 내역이 뜨고 클릭시 강의실 예약	



				<p>내역 페이지로 이동</p> <p>2) 미승인 강의실 예약 내역이 뜨고 클릭시 강의실 예약 내역 페이지로 이동</p> <p>3) 어제 앱에 로그인 한 사용자 수를 띄워준다</p> <p>4) 사용자가 앱에서 관리자에게 요청한 문의의 개수를 띄워주며 클릭시 문의관리 페이지로 넘어간다</p> <p>5) 어제를 포함한 총 일주일의 앱 사용자를 그래프로 변환하여 띄워준다</p> <p>6) 위에서 나열한 기능들을 그래프로 변환하여 보기 좋게 나열한다</p>	성공
웹	예약확인	예약 확인	좌측 사이드바에서 예약확인 메뉴를 클릭한다	달력의 날짜를 클릭하면 해당 날짜에 잡혀있는 강의실과 공유공간 예약 내역이 테이블로 나열된다	성공
웹	예약관리 / 강의실	예약 승인 예약 생성 예약 조회 예약 삭제 반납사진 조회	<p>좌측 사이드바에서 예약관리 메뉴의 드롭다운 메뉴 강의실을 클릭</p> <p>미승인 강의실 예약 내역 목록에서 신청서 보기 버튼을 눌러 신청서를 확인,</p>	<p>미승인 강의실 예약 신청서 내부의 승인 버튼을 누르면 바뀐 승인 값이 저장되며 승인 강의실 예약 내역으로 승급</p> <p>신청서 내부에는 신청자의 개인정보 및</p>	



			강의실 예약을 승인할 경우 신청서 최하단의 승인 버튼을 눌러 승인처리 승인된 강의실 예약 중 삭제해야할 내역이 있는경우 우측의 삭제 버튼을 눌러 해당 예약 내역을 삭제 강의실 예약을 생성해야하는 경우 상단의 강의실 예약 추가하기 버튼을 눌러 강의실 이름, 날짜, 사용목적, 사용시간을 입력하고 예약 추가 버튼을 클릭하여 강의실 예약 생성 반납사진 버튼을 클릭하여 사용자가 반납한 강의실 사진 확인	서명, 강의실 이용목적이 나열됨 승인된 강의실 예약 내역에서도 신청서를 볼 수 있지만 승인 버튼이 사라짐 승인된 강의실 예약을 삭제하면 데이터베이스에서 예약 내역이 삭제됨 강의실 예약을 추가하면 새로운 강의실 예약이 데이터 베이스에 추가됨	성공
웹	예약관리 / 공유공간	예약 조회 반납사진 조회	좌측 사이드바에서 예약관리 메뉴의 드롭다운 메뉴인 공유공간 메뉴를 클릭	공유공간 예약 내역이 조회되며 사용자가 이용 후 제출한 반납사진이 테이블로 표시됨	성공
웹	문의관리	문의 조회 문의 답변	좌측 사이드바에서 문의관리 메뉴를 클릭 문의내용 확인 버튼을 눌러 문의내용 확인 후 답변 입력 후 답변하기 버튼 클릭	앱 사용자의 정보, 문의를 보낸 날짜, 답변 여부를 테이블로 표시 답변을 입력한 후 '답변하기' 버튼을 클릭하면 답변 여부가 답변 미완료에서 답변 완료로 변경되며 관리자가 작성한 답변은 데이터베이스에 저장됨	성공
웹	알림 관리	알림 조회 알림 생성	좌측 사이드바에서 문의관리 메뉴를 클릭 우측 상단의 +버튼을 클릭한 후 공지사항 분류를 선택하고 공지내용을 작성한 후 등록하기 버튼 클릭	관리자가 공지사항을 등록한 날짜 등 공지사항에 관한 정보가 나옴 추가 버튼을 눌러 공지사항을 등록한	성공



			필요없거나 잘못 작성한 공지사항에 대해서는 우측의 삭제 버튼을 클릭하여 삭제	경우, 관련 데이터가 데이터베이스에 저장	
				삭제 버튼을 눌러 삭제한 경우 테이블에서 공지사항이 삭제됨과 동시에 데이터베이스에서도 삭제됨	
웹	공간관리 / 강의실	강의실 조회 강의실 등록 강의실 삭제	좌측 사이드바에서 공간관리 메뉴를 클릭 강의실 관리 탭의 추가하기 버튼을 누른 뒤 단과대학, 강의실 이름, 사용가능한 테이블, 수용인원을 입력 파일선택 버튼을 클릭하여 강의실 사진을 첨부한 뒤 생성하기 버튼을 클릭 사용불가하게 된 강의실은 삭제 버튼을 통해 삭제함	관리자가 입력한 정보들과 인코딩된 강의실 사진은 데이터베이스에 저장되어 이후 앱에서 사용됨 생성된 강의실에 대한 정보를 파이어베이스에서 불러와 사진을 디코딩하여 강의실 관리 탭에 리스트로 나열됨	성공
웹	공간 관리 / 공유공간	공유공간 조회 공유공간 삭제	좌측 사이드바에서 공간관리 메뉴를 클릭 공유공간 관리 탭에서 추가하기 버튼을 누른 후 단과대학, 강의실 이름, 사용가능한 테이블, 사용가능한 시간 등을 입력하고 공유공간 사진과 도안 사진을 등록한다 사용가능한 테이블을 적는 칸 옆의 입력 버튼을 누르고 생성되는 입력란에 각 테이블의 좌석수를 입력한 후 생성하기 버튼을 클릭 사용불가하게 된 공유공간이나 테이블이 있는 경우 삭제 버튼을 통해 삭제함	추가하기 버튼을 누른 뒤 각종 데이터를 입력한 후 사용 가능한 테이블 입력란 옆 입력 버튼을 누르면 동적으로 각 테이블에 대한 좌석 수 입력 칸이 생성됨 공유공간 사진과 도안을 첨부하면 각각 관리자가 확인 할 수 있는 미리보기 사진이 UI에 띄워짐 생성하기를 누르면 각종 데이터는 데이터베이스에 저장되며 테이블과 좌석에 대한 정보는	성공



				<p>배열로 저장되어 따로 데이터베이스에 저장됨</p> <p>추가한 두 사진은 각각 인코딩되어 데이터베이스에 저장됨</p> <p>데이터베이스에 저장된 인코딩된 사진은 공유공간 조회시 디코딩되어 데이터베이스에 어떤 사진이 들어갔는지 관리자가 확인할 수 있도록 표시됨</p>	
웹	혼잡도관리	카메라 위치 등록 위치 수정 정보 삭제 등록 조회	<p>좌측 사이드바에서 혼잡도 관리 메뉴를 클릭</p> <p>우측 상단의 추가하기 버튼을 클릭한 후 필요한 데이터를 입력한 후 생성하기 버튼을 누름</p> <p>테이블에 생성된 수정하기 버튼을 통해 수정된 정보를 입력한 후</p>	<p>페이지가 랜더링 될때 데이터베이스에서 받아온 데이터를 통해 테이블이 구성되어 현재 등록된 카메라의 위치정보와 현재 작동 여부를 알려줌</p> <p>추가하기 버튼을 눌러서 새로운 카메라 위치 정보를 등록하면 데이터베이스에 위치정보가 등록됨</p> <p>수정, 삭제버튼을 누른 경우도 데이터베이스에 반영되어 이후 앱이 사용할 데이터가 됨</p>	성공
앱	사용자	회원가입	회원가입 버튼을 눌러 내용들을 기입하고 가입을 완료한다.	데이터 베이스에 정상적으로 사용자 정보들이 등록된다	성공
앱	사용자	로그인	이메일과 패스워드를 기입하여 로그인 버튼을 눌러 메인화면으로 진입한다. 또는 구글로 로그인 버튼을 눌러 구글 이메일로 연동하여 로그인이 가능하다.	데이터베이스에 있는 사용자 정보들과 기입한 값들을 비교하여 유저 정보가 있다면 정상적으로 정보들을 제공한다.	성공



앱	공간대여	등록된 공간 불러오기	백엔드에 구현된 관리자가 등록된 공간 정보들을 받아오는 api를 통해 공간 정보들을 가져온다.	listview빌더에 배열 순서에 따라 강의실 정보들을 ui에 출력하고 리스트 별로 다른 값들을 위젯화하여 동적으로 제공할 수 있도록 한다.	성공
앱	공간대여	예약하기	백엔드로 해당 공간에 대한 정보를 요청하여 받아오고 예약정보들을 통해 사용자의 ui에 출력하고 마감표시를 한다. 버튼들을 클릭하여 값을 가지고 예약하기를 누른다면 공유공간 예약이 완료된다. 강의실같은 경우에는 테이블이 없고 강의실 품이 존재하였고 시간과 날짜를 선택하여 품 작성하기를 누르면 관련 정보들을 기입하는 창과 서명데이터들을 서버로 전송할 수 있게 구현하였다.	예약 구성에서 페이지를 이동할때마다 다음 페이지로 값을 전달해야했고 해당 버튼들을 통해 예약 알고리즘의 제한 등을 구현할 수 있었다. 또한 서명 데이터들을 참여자 배열에 담아 인코딩하여 서버로 보내는 데에 성공했고 정상적으로 데이터베이스에 반영하는 데에 성공했다	성공
앱	혼잡도	혼잡도 공간별로 불러오기	혼잡도 페이지에 진입한다. 혼잡도 높은순 ,낮은 순에 따라 정렬하는지 확인하고 공간 별로 정보 확인이 가능한지 테스트한다.	혼잡도 정보 api를 ai 서버와 연결하는 데에 성공했고 이를 listview빌더에 동적으로 보여지게 하는데에 성공했다	성공
앱	예약내역	공간 별 예약내역 불러오기	예약 내역 페이지에 진입하여 공유공간, 강의실 버튼을 눌러 예약내역을 확인한다. 예약된 시간에 입장하기 버튼을 눌러 타이머가 제대로 동작하는지 확인하고 예약시간이 아니라면 입장할 수 없는 알림이 뜨는 것을 확인한다. 반납버튼을 눌렀을 때 카메라 촬영페이지로 이동할 수 있는지 확인하고 촬영 사진이 제출하기를 눌렀을때 정상적으로 완료 화면이 구성되는지 확인한다.	예약페이지에 오늘 날짜부터 기간을 설정하고 공간 반납 status에 따라 이용예정내역과 이용완료내역을 확인하는데에 성공했다. 현재 시간을 받아와서 예약 시간과 다르다면 입장할 수 없게 만드는 데에 성공했고 가능 시간에 입장하게 하여 해당시간부터 타이머로 남은 시간을 표시할 수 있도록 결과물을 도출하였다. 반납하기 버튼을 눌렀을 때 카메라 라이브러리를 통해 사진을 촬영하고 ai	성공



				서버의 제출 api를 호출하는 데에 성공하였다.	
앱	마이페이지	문의하기, 이용안내, 로그아웃	문의하기 버튼을 눌러 제목 내용을 입력한 후 정상적으로 전송되는지 메시지를 확인한다. 이용안내버튼을 눌러 정상적으로 출력되는지 확인한다. 로그아웃 버튼을 눌러 로그아웃 했을 때 정상적으로 로그인 페이지로 이동되는지 확인한다.	문의하기 내용들을 서버 api를 호출하여 보내는 데에 성공했고 이용안내 알림 dialog 띄우기, 로그아웃 했을 때 signin페이지로 네비게이트 하는 데까지 성공했다.	성공
앱	알림	알림페이지 확인	예약이 완료되었을 때 해당 공간 정보에 대한 알림이 오는지 확인한다. 또한 반납이 완료되었을 때 해당 강의실에 대해 페널티가 부과되지 않고 성공적으로 반납되었는지 알림 메시지를 확인할 수 있다.	알림 라이브러리와 ios 권한 설정을 통해 알림 구현을 완료했으며 초기화 후 해당 액션들이 보여졌을 때 각 상황에 따라 메시지를 보여질 수 있도록 결과를 도출하였다.	성공
서버	회원 관리	1) 회원가입 2) 로그인 3) 로그아웃 4) 프로필 수정 5) 프로필 조회	POSTMAN TOOL을 이용하여 1) 회원 정보를 입력한다. 2) 이메일과 패스워드를 입력한다. 3) 로그아웃을 요청으로 보낸다. 4) 수정한 회원 정보를 입력한다. 5) 해당 프로필 uid를 GET으로 요청한다.	1) 데이터베이스에 회원정보가 저장된다. 2) 로그인 성공 3) 로그아웃 성공 4) 데이터베이스에 수정된 회원 정보가 저장된다. 5) 회원 정보를 불러온다.	성공
서버	공유 공간 관리	1) 관리자가 등록한 예약 가능 공유공간 조회 2) 예약 날짜를 받았을 때 가능 시간대	POSTMAN TOOL을 이용하여 1) 사용자 UID를 포함하여 GET요청을 보낸다. 2) 사용자 userId, roomName, date, startTime, endTime, tableNumber을 req.body에 포함하여 POST 요청을 보낸다. 3) 사용자 userId, roomName, date를 포함하여 POST요청을 보낸다. 4) GET 요청을 보낸다. 5) GET 요청을 보낸다.	1) 관리자가 등록한 예약 가능 공유 공간을 반환한다. 2) 사용자가 해당 시간대에 예약할 수 있다. 3) 해당 날짜에 가능한 시간대를 반환한다. 4) 사용자의 예약 내역을 반환한다.	성공



		<ul style="list-style-type: none"> 보여주기 4) 사용자별 예약 내역 반환 5) 사용자별 이용 예정 내역 반환 6) 예약 취소 7) 반납 8) 사용자가 반납을 잘했는지 판단 	<ul style="list-style-type: none"> 6) 사용자의 userId, roomName, date, startTime, endTime, tableNumber를 req.body에 포함하여 POST요청을 보낸다. 7) 사용자의 userId, roomName, date, startTime, endTime, tableNumber를 req.body에 포함하여 POST요청을 보낸다. 8) cron-scheduler를 이용하여 백라운드로 돌린다. 	<ul style="list-style-type: none"> 5) 사용자의 이용 예정 내역을 반환한다. 6) 해당 예약 내역을 취소한다. 7) 해당 공유공간을 반납한다. 8) 모든 사용자가 반납을 잘했는지 판단하고 반납을 제대로 안했을 시 패널티를 적용한다. 	
서버	강의실 관리	<ul style="list-style-type: none"> 1) 관리자 등록한 예약 가능 강의실 조회 2) 예약 3) 날짜 받았을 때 가능 시간대 보여주기 4) 사용자별 예약 내역 조회 5) 사용자별 이용 예정 	<p>POSTMAN TOOL을 이용하여</p> <ul style="list-style-type: none"> 1) POST요청을 보낸다. 2) 사용자의 userId, roomName, date, startTime, endTime, usingPurpose, studentId, participants, numberOfPeople, signature를 req.body에 포함하여 POST요청을 보낸다. 3) 사용자의 userId, roomName, date를 req.body에 포함하여 POST요청을 보낸다. 4) GET요청을 보낸다. 5) GET요청을 보낸다. 6) 사용자의 userId, roomName, date, startTime, endTime를 req.body에 포함하여 POST요청을 보낸다. 	<ul style="list-style-type: none"> 1) 관리자가 등록된 예약 가능 강의실 정보를 반환한다. 2) 사용자가 해당 시간대에 예약할 수 있다. 3) 해당 날짜에 가능한 시간대를 반환한다. 4) 사용자의 예약 내역을 반환한다. 5) 사용자의 이용 예정 내역을 반환한다. 6) 해당 강의실을 반납한다. 7) 해당 예약 내역을 취소한다. 	성공



		<ul style="list-style-type: none"> 내역 조회 6) 반납 7) 예약 취소 	<ul style="list-style-type: none"> 7) 사용자의 userId, roomName, date, startTime, endTime를 req.body에 포함하여 POST요청을 보낸다. 		
서버	문의	<ul style="list-style-type: none"> 1) 문의 등록 2) 사용자 별 전체 문의 내역 조회 	<p>POSTMAN TOOL을 이용하여</p> <ul style="list-style-type: none"> 1) 사용자의 userId, date, title, content를 req.body에 포함하여 POST요청을 보낸다. 2) parameter에 userId, startDate, endDate를 입력하여 GET요청을 보낸다. 	<ul style="list-style-type: none"> 1) 사용자의 문의 내용이 등록된다. 2) 사용자의 전체 문의 내역이 반환된다. 	성공
서버	알림	<ul style="list-style-type: none"> 1) 모든 사용자에게 공유 공간 입실 10분 전에 알림 보내기 2) 모든 사용자에게 강의실 입실 10분 전에 알림 보내기 	<ul style="list-style-type: none"> 1) console.log를 입력하여 백그라운드가 돌아가는지 확인한다. 2) console.log를 입력하여 백그라운드가 돌아가는지 확인한다. 	<ul style="list-style-type: none"> 1) 해당 사용자에게 공유 공간 입실 알림이 보내진다. 2) 해당 사용자에게 강의실 입실 알림이 보내진다. 	성공
서버	관리자 회원 관리	<ul style="list-style-type: none"> 1) 관리자 인지 확인 2) 관리자 로그인 3) 방문자 수 조회 4) 회원 정보 삭제 	<p>POSTMAN TOOL을 이용하여</p> <ul style="list-style-type: none"> 1) email 정보를 받아 관리자인지 확인한다. 2) email, password를 req.body에 포함하여 POST요청을 보낸다. 3) GET요청을 보낸다. 4) 특정 사용자 userId를 parameter에 포함하여 GET요청을 보낸다. 	<ul style="list-style-type: none"> 1) 관리자인지 확인된다. 2) 로그인이 성공한다. 3) 현재 날짜에 방문한 사용자 수가 반환된다. 4) 회원 정보가 삭제된다. 5) 회원이 등록된다. 	성공



		<ul style="list-style-type: none"> 5) 회원 등록 6) 프로필 수정 7) 모든 사용자 프로필 조회 8) 특정 사용자 프로필 조회 	<ul style="list-style-type: none"> 5) useremail, password, name, studentId, faculty, department, club, phone, agreeForm을 req.body에 포함하여 POST요청을 보낸다. 6) 특정 사용자 userId를 parameter로 포함하고 password, name, studentId, faculty, department, club, phone, agreeForm을 req.body에 포함하여 POST요청을 보낸다. 7) GET요청을 보낸다. 8) 특정 사용자 userId를 parameter에 포함하여 GET요청을 보낸다. 	<ul style="list-style-type: none"> 6) 프로필이 수정된다. 7) 모든 사용자의 프로필이 반환된다. 8) 특정 사용자 프로필이 반환된다. 	
서버	카메라	<ul style="list-style-type: none"> 1) 카메라의 설치 정보 조회 2) 카메라 세팅 설정 3) 카메라 세팅 삭제 	<p>POSTMAN TOOL을 이용하여</p> <ul style="list-style-type: none"> 1) GET요청을 보낸다. 2) locationName, location을 req.body에 포함하여 POST요청을 보낸다. 3) locationName을 parameter에 포함하여 DELETE 요청을 보낸다. 	<ul style="list-style-type: none"> 1) 카메라의 설치 정보가 반환된다. 2) 카메라 세팅이 설정된다. 3) 카메라 세팅이 삭제된다. 	성공
서버	관리자 공유 공간 관리	<ul style="list-style-type: none"> 1) 관리자 동아리방 설정 생성 2) 동아리방 정보 조회 3) 동아리방 정보 삭제 4) 특정 기간의 	<p>POSTMAN TOOL을 이용하여</p> <ul style="list-style-type: none"> 1) faculty, roomName, available_Table, tableList, available_People, available_Time, clubRoomImage, clubRooDesignImage를 req.body에 포함하여 POST요청을 보낸다. 2) faculty를 parameter에 포함하여 GET요청을 보낸다. 3) faculty, roomName을 req.body에 포함하여 DELETE 요청을 보낸다. 	<ul style="list-style-type: none"> 1) 동아리방이 생성된다. 2) 생성된 동아리방 정보들이 반환된다. 3) 동아리방을 삭제한다. 4) 모든 예약 내역이 반환된다. 	성공



		모든 예약 내역 반환	4) faculty, startDate, endDate를 parameter에 포함하여 GET요청을 보낸다.		
서버	관리자 문의 관리	1) 문의하기 답변 2) 전체 문의 내역 조회	POSTMAN TOOL을 이용하여 1) studentId, date, time, response를 req.body에 포함하여 POST요청을 보낸다. 2) faculty, startDate, endDate를 parameter에 포함하여 GET요청을 보낸다.	1) 문의하기 답변이 성공한다. 2) 전체 문의 내역들이 반환된다.	성공
서버	관리자 알림 관리	1) 모든 사용자에게 알림 보내기 2) 개별 사용자에게 알림 보내기	POSTMAN TOOL을 이용하여 1) title, body를 req.body에 포함하여 POST요청을 보낸다. 2) studentId, title, body를 req.body에 포함하여 POST요청을 보낸다.	1) 모든 사용자의 디바이스에 알림이 보내진다. 2) 개별 사용자의 디바이스에 알림이 보내진다.	성공
서버	관리자 강의실 관리	1) 강의실 생성 설정 2) 생성된 강의실 정보 조회 3) 강의실 정보 삭제 4) 강의실 예약 승인 5) 특정 기간에 승인된 모든 예약 내역 조회	POSTMAN TOOL을 이용하여 1) faculty, roomName, available_People, available_Time, conferenceImage를 req.body에 포함하여 POST요청을 보낸다. 2) faculty, roomName을 req.body에 포함하여 GET 요청을 보낸다. 3) faculty, roomName을 req.body에 포함하여 DELETE 요청을 보낸다. 4) studentId, roomName, date, startTime, endTime을 req.body에 포함하여 POST요청을 보낸다. 5) faculty, startDate, endDate를 parameter에	1) 강의실이 생성된다. 2) 생성된 강의실 정보가 반환된다. 3) 강의실 정보가 삭제된다. 4) 강의실 예약이 승인된다. 5) 특정 기간에 승인된 모든 예약 내역이 반환된다. 6) 특정 기간에 아직 승인되지 않은 예약 내역이 반환된다. 7) 예약 내역이 삭제된다.	성공



		6) 특정기간에 아직 승인되지 않은 모든 예약 내역 조회 7) 예약 내역 삭제 8) 예약	포함하여 GET요청을 보낸다. 6) faculty, startDate, endDate를 parameter에 포함하여 GET요청을 보낸다. 7) studentId, roomName, date, startTime, endTime를 req.body에 포함하여 DELETE 요청을 보낸다. 8) faculty, roomName, date, startTime, endTime, usingPurpose를 req.body에 포함하여 POST요청을 보낸다.	8) 예약이 성공된다.	
라즈베리파이 flask	실시간 카메라 이미지 송신	라즈베리파이에서 실시간으로 카메라 영상을 flask서버로 송신	라즈베리파이에서 http 통신을 위해 주소를 배포한 주소로 설정 후 app.py를 실행	실시간으로 aws의 서버에 이미지 송신	성공
flask	실시간 이미지 받는 api	라즈베리파이에서 실시간으로 보낸 영상을 수신하여 사용가능한 이미지로 변경	서버 테스트 툴을 이용하여 flask서버에 인코딩된이미지, 카메라위치정보를 json post요청을 보낸다.	실시간으로 aws의 서버에 이미지가 수신되어 모델의 입력으로 사용하여 혼잡도를 측정할수 있으며 저장을 가능하게 하여 반납후 파이어베이스에 저장가능함	성공
ai	혼잡도 측정	수신한 이미지를 입력으로 하여 인공지능 모델을 통해 혼잡도 측정	정의되어있는 모델에 이미지를 입력으로 넣으면 score가 출력된다.	자율주행 스튜디오, 무한상상실, 해동과 같은 공유공간의 혼잡도를 실시간으로 알려주어 가보지 않아도 혼잡도를 미리 알 수 있음	성공
flask	혼잡도 api	혼잡도정보 저장 및 제공하는 api	서버 테스트 툴을 이용하여 공용공간의 이미지 및 정보를 json으로 post요청을 보낸다.	혼잡도의 정보를 데이터베이스에 저장 및 데이터베이스에 저장되어있는 혼잡도 정보를 앱에서 실시간으로 확인이 가능해짐	성공

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	K-SharePlace	
	팀 명	필기아	
	Confidential Restricted	Version 2.3	2024-05-23

ai	청소상태 측정	인공지능 모델을 이용하여 반납시 사용상태를 체크하는 기능	정의되어있는 모델에 반납전 이미지, 반납후의 이미지를 입력으로 넣으면 score가 출력된다.	반납시에 책상들에 맞게 이미지를 입력으로 받아 청소되어있는 책상과 책상의 청소 및 관리상태를 판단하여 편리한 관리가 가능해짐	성공
flask	반납이미지 저장 및 페널티 부여 api	반납시의 이미지를 통해 관리 상태에 따라 유저에 페널티 부여 및 반납사진을 저장	서버 테스트 틀을 이용하여 반납전 이미지, 반납후의 이미지, user등의 정보로 post요청을 보낸다.	측정 결과를 바탕으로 데이터베이스에서 user에게 페널티 부여 및 반납사진 저장	성공