**2020년 1학기 인터넷 응용**

**X-Project 최종 보고서**

Smart Study Room

- **한국외국어대학교 정보통신공학과 3팀 -**

**윤준영(201502166) 이영상(201502464) 박재원(201601333) 김가영(201800528) 서우빈(201801841)**

# 서론

본 프로젝트는 머신러닝을 통해 카메라를 활용한 객체 인식, 그리고 장고를 이용한 열람실 좌석 사용에 불편함을 덜어줄 좌석 예약 시스템을 합친 스마트 열람실 서비스이다. 현재까지 예약을 하여 사용하는 열람실, 예약 없이 자유롭게 쓰는 열람실 두 종류가 있었고, 두 경우 모두 단점들이 존재했다. 이 프로젝트는 그 단점들을 최대한 보완하고자 고안하게 되었다. 사용자는 웹앱을 통해 열람실의 좌석 현황을 볼 수 있고, 빈자리를 예약하여 사용을 할 수 있다. 또한 예약 없이 사용할 경우, 열람실에 설치된 카메라가 머신러닝을 통해 객체 인식을 하여 자리에 사람을 감지하면, 값을 장고에게 보내 해당 자리를 이용 중인 좌석으로 바꿔준다.

# 목차

1. 용어 및 약어
2. 프로젝트 개요 3. 제품 소개
   1. 제품 개요
   2. 제품 필요성
   3. 제품 사용법
3. 구성도 및 설계
   1. 전체 시스템 구상도
   2. 서버-사용자 구상도
   3. 설치장소-서버 구상도
   4. 시퀀스 다이어그램
4. 프로젝트 실행
5. 프로젝트 수행 방법 6-1. 요구분석
   1. 설계
   2. 구현
   3. 테스트
   4. 유지보수
6. 각 분야 상세 설명
   1. 웹캠
   2. 장고(서버)
   3. 소켓
7. 문제점 및 해결방안
8. 기대효과 및 활용분야
9. 프로젝트 세부 일정
10. 팀원 개별 담당 업무
11. 느낀점

# 1. 용어 및 약어

|  |  |
| --- | --- |
| 용어 및 약어 | 풀이 |
| 장고 (Django) | Django란 보안이 우수하고 유지보수가 편리한 웹사이트를 신속하게 개발하도록 도움을 주는 파이썬 웹 프레임워크이다. 훌륭한 개발자에 의해 만들어진 이 프렘워크는 웹 개발을 하는데 많은 도움을 주기 때문에 새롭게 웹 개발을 시작할 필요 없이 그저 프레임워크를 활용하여 앱 개발에만 집중할 수 있게 도와준다. |
| 딥러닝 (Deep  Learning) | 딥러닝은 여러 비선형 변환기법의 조합을  통해 높은 수준의 추상화(abstractions, 다량의 데이터나 복잡한 자료들 속에서 핵심적인 내용 또는 기능을 요약하는 작업)를 시도하는 기계 학습 알고리즘의 집합으로 정의되며, 큰 틀에서 사람의 사고방식을 컴퓨터에게 가르치는 기계학습의 한 분야라고 이야기할 수 있다.  어떠한 데이터가 있을 때 이를 컴퓨터가 알아 들을 수 있는 형태(예를 들어 이미지의 경우는 픽셀정보를 열벡터로 표현하는 등)로 표현하고 이를 학습에 적용하기 위해 많은 연구가 진행되고 있으며, 이러한 노력의  결과로 deep neural networks, convolutional deep neural networks, deep belief networks,와 같은 다양한 딥 러닝 |

|  |  |
| --- | --- |
|  | 기법들이 컴퓨터비젼, 음성인식, 자연어 처리, 음성/신호 처리 등의 분야에 적용되어 최첨단의 결과들을 보여주고 있다. |
| 소켓 | 소켓은 소프트웨어로 작성된 추상적인 개념의 통신 접속점이라고 할 수 있는데 네트워크 운용 프로그램은 소켓을 통하여 통신망으로 데이터를 수신하게 된다. 소켓은 응용 프로그램에서 TCP/IP를 이용하는 창구 역할을 하며 응용 프로그램과 소켓 사이의 인터페이스 역할을 하고 있다. 소켓은 두 프로그램이 네트워크를 통해 서로 통신을 수행할 수 있도록 양쪽에 생성되는 링크의 단자이다. 두 소켓이 연결되면 서로 다른 프로세스끼리 데이터를 전달할 수 있다. 결국 소켓이 구현됨으로써 네트워크 및 전송 계층의 캡슐화가 가능해진다. |
| cvlib | cvlib은 파이썬의 간단하고 사용하기 쉬운 오픈 소스 컴퓨터 비전 라이브러리이다. 영상처리 라이브러리인 OpenCV와 딥러닝, 머신러닝 라이브러리인 텐서플로우를 기반으로 하는 cvlib은 크게 세가지의 기능을 제공하는데, 안면감지, 성별감지, 그리고 물체감지이다.  비교적 접근성이 좋고 단순한 라이브러리인 OpenCV와 텐서플로우의 장점을 그대로 이어받아 cvlib은 단순하고, 접근성이 좋으며 확장성이 큰 라이브러리이다. |

|  |  |
| --- | --- |
| Ngrok | Localhost에 있는 서버를 웹 상에 띄워주는 tunnel로 만든 project를 테스트하기위한 서버로 사용. |

# 2. 프로젝트 개요

시험기간에 학교나 공공 도서관의 열람실을 방문해 본 적 있는가? 야심 차게 공부하려 책가방을 들고 열람실에 갔지만 자리가 없어 하루의 시작을 망친 경험이 많을 것이다. 지금 이 시간에 열람실에 가면 자리가 있을지 직접 가지 않으면 알 수 없기 때문에 생긴 문제점이다. 본 팀은 이에 착안하여 이번 프로젝트를 통해 django web app을 이용하여 인터넷으로 실시간 열람실 이용 현황을 확인하고, 원하는 시간에 자리 예약을 할 수 있으며, 이용 시간이 만료되기 전이라도 이용자와 짐이 좌석에 없으면 자동으로 이용이 취소되는 시스템을 구현하고자 한다. 본 프로젝트를 통해 보다 효율적으로 열람실을 이용할 수 있게 함을 목적으로 한다.

본 팀이 제작한 smart studyroom은 학교 열람실뿐만 아니라 이용자의 입/퇴실 여부가 중요하고 예약 서비스가 필요한 어떤 열람실에서도 활용할 수 있다. 현재 공공 도서관은 입실과 예약 시스템은 합리적으로 관리되고 있지만 퇴실은 이용자가 퇴실 시 직접 입력하거나 관리인이 수시로 직접 확인하여 퇴실 처리를 하기 때문에 매우 비효율적이다. 하지만 본 팀이 제작한 제품은 이용자가 자리를 비우면 자동으로 퇴실 처리를 해주기 때문에 차별점을 가진다.

# 3. 제품 소개

## 3.1. 제품 개요

SMART STUDY ROOM은 열람실을 편하게 사용할 수 있게 만들어 주는 제품이다. 이 제품은 총 4가지의 중심 서비스를 제공한다. 첫 번째로 현재 열람실에 있는 CCTV를 통해 사용자 현황을 제공받아서 웹서버에서 확인할 수 있게 하는 서비스와 두 번째로 웹서버에서 예약이 가능하게 하는 서비스와 마지막 세 번째로 예약을 하지 않아도 열람실 자리에 앉으면 자동으로 사용 중으로 바꿔주는 서비스를 제공한다. 마지막으로 예약 여부와 상관없이 사용 중이던 자리에서 나오면 자동으로 퇴실을 시켜주는 서비스를 제공한다.

## 3.2. 제품 필요성

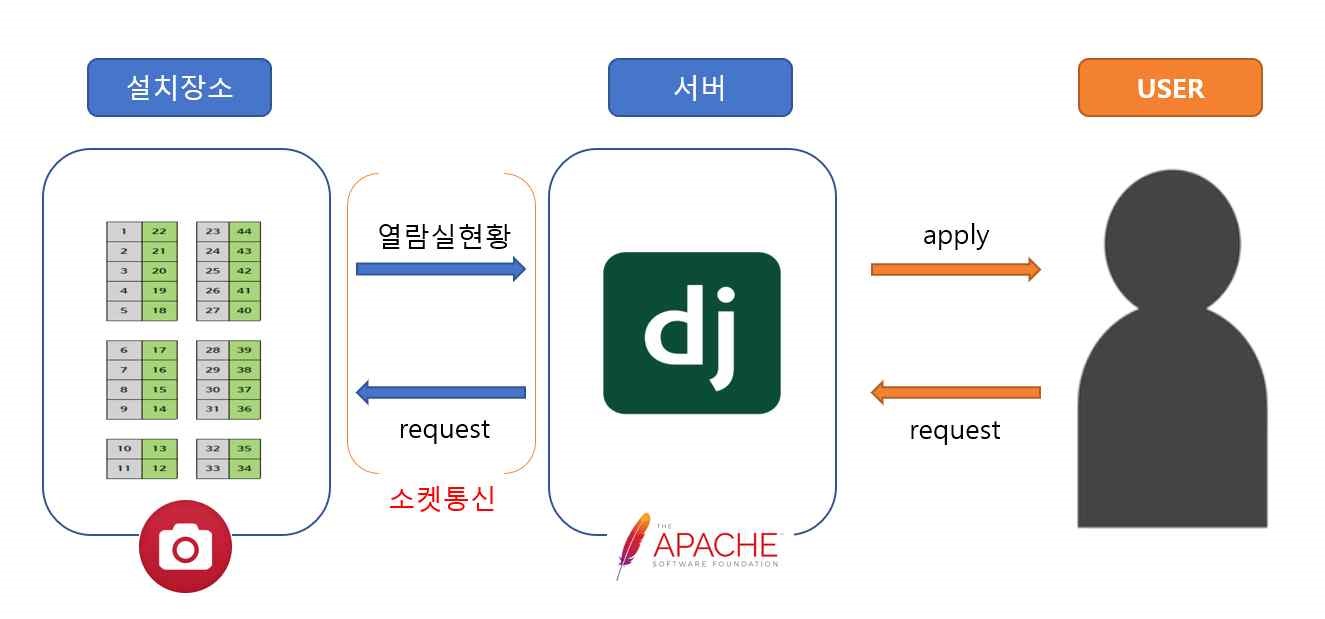


기존의 열람실 예약 시스템의 불편함을 생각해보면 첫 번째로 예약을 할 수 없는 열람실들은 사용 현황을 한눈에 보기 어렵다는 것이 있다. 예약제가 아닌 열람실들은 열람실로 가기 전까진 사용할 수 있는 좌석이 남아있는지, 파악하기가 어렵다. 굳이 예약을 하지 않아도, 사용 현황을 쉽게 파악할 수 있게 제품을 만들었다. 두 번째로 예약 시스템이 갖추어진 열람실에서 항상 예약을 해야만 자리를 사용할 수 있는 불편함이 있다. 그렇기에 예약을 따로 하지 않아도, 열람실 자리에 앉기만 해도 사용 중이라고 표시되게 하였다. 마지막으로 예약을 했을 경우, 항상 퇴실을 클릭해야만 퇴실이 가능한 불편함을 없앴다. 책상 위에 물건이 없고, 사람이 더 이상 사용하지 않는 좌석은 자동으로 퇴실로 바꿔주는 서비스를 제공한다.

## 3.3. 제품 사용법

SMART STUDY ROOM의 사용방법은 간단하다. 예약을 사용할 사람들은 웹서버에서 사용현황을 확인하고, 로그인을 한 후 빈자리를 예약하면 된다. 만약 예약을 하지 않고 사용할 사람들은 웹서버에서 사용현황을 확인하고 사용할 수 있는 좌석이 남아있다면 열람실로 가서 그 좌석을 사용하면 된다. 이 경우 좌석에 착석하여 사용하기 시작하면 자동으로 그 좌석은 사용중인 좌석으로 바뀌게 되어 쉽게 사용할 수 있다. 사용이 끝난 뒤에는 좌석에 짐을 치우고 좌석을 떠나기만 하면 자동으로 퇴실로 바꾸어 준다.

## **4. 구성도 및 설계**

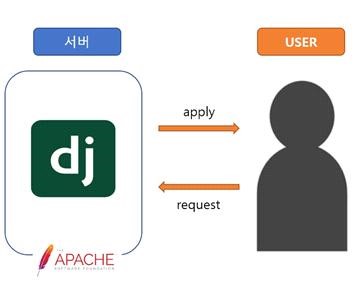


4.1. 전체 시스템 구성도

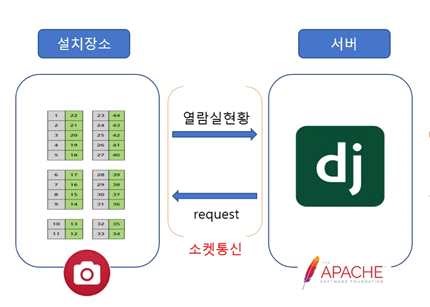
사용자가 예약 현황을 확인하려 할 때

서버는 항상 열려있고, 사용자가 연결을 요청한다. 둘이 연결이 되면, 서버는 웹캠에 연결을 요청한다. 웹캠과 서버가 연결이 되면 웹캠에서 열람실 좌석 실시간 영상으로 사용 현황을 파악한 뒤 서버에 전송해 준다. 서버에서 사용 현황을 받고, 장고를 통해 웹에 사용 현황 및 예약 현황을 띄워준다. 사용자가 에약을 하려 할 때 서버는 항상 열려있고, 사용자가 연결을 요청한다. 사용자와 서버가 연결이 되면, 서버는 웹캠에 연결을 요청한다. 웹캠과 서버가 연결이 되면 웹캠에서 열람실 좌석 실시간 영상으로 사용 현황을 파악한 뒤 서버에 전송해 준다. 서버에서 사용 현황을 받고, 사용자의 예약 요청을 추가한 뒤 장고를 통해 웹에 사용자의 예약이 포함된 사용 현황 및 예약 현황을 띄워주게 된다.

4.2. 서버-사용자

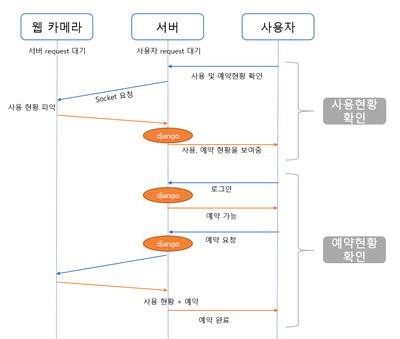


서버는 항상 열린 채로 둬야 한다. 사용자는 웹서버를 통해 서버와 통신할 수 있는데, 사용자가 처음 웹서버에 접속하면 로그인 버튼과 스터디 룸 선택 버튼을 볼 수 있다. 이때 스터디 룸 종류 중에서 원하는 스터디룸을 선택하면 좌석 사용 현황을 볼 수 있다. 로그인을 한 뒤 스터디룸을 선택하면 사용현황을 보여줄 뿐만 아니라 예약도 가능하게 화면이 바뀐다. 사용자의 입장에서 예약 서비스와 좌석 사용 현황 서비스는 모두 여기서 얻을 수 있다.



### 4.3. 설치장소-서버

서버는 항상 열린 채로 둬야 한다. 사용자가 웹에 접속하면, 서버에서 열람실 사용 현황 정보를 얻기 위해 설치장소(열람실)의 웹캠에 소켓통신으로 요청을 한다. 웹캠이 통신을 수락하면 서버가 클라이언트가 되어 정보를 요청한다. 웹캠은 지속적으로 좌석의 물체 존재 여부를 확인하여 데이터에 저장하고 서버가 연결되면 그 데이터에 저장된 정보가 보내지게 된다. 서버에서 현재 사용 현황에 예약 현황을 합해서 장고로 웹서버에 띄워준다.



4.4. 시퀀스 다이어그램

전체 흐름도를 위와 같이 시퀀스 다이어그램으로 나타낸 것이다.

# 5.프로젝트 실행

5.1 프로그램 시연 동영상입니다.

https://www.youtube.com/watch?v=9q2IDIQrM7A&feature=share

## 5.2 서버

|  |  |
| --- | --- |
| 웹 시작 페이지이다. |  |

|  |  |
| --- | --- |
| 정보통신공학과 스터디룸을 클릭 하면 좌석 현황을 보여준다.  로그인을 하지 않으면 예약 버튼 이 나오지 않는다. |  |

|  |  |
| --- | --- |
| 회원가입 후 로그인을 하겠다. |  |

|  |  |
| --- | --- |
| 로그인 후 좌석 현황을 누르면 예약 버튼이 생긴다. 위에 로그인한 유저를 보여준다. |  |

|  |  |
| --- | --- |
| 2번 자리를 예약하였다. |  |

## 5.3 웹캠

|  |  |
| --- | --- |
| 웹캠 부분이다.  책상과 의자는 좌석 사용여부에 영향이 미치지 않는다.  예약석에 사람이 앉으면 사용 중으로 표시된다. |  |
| 사람이 자리를 비워도 물건이 남아있으면 사용중인 자리로 판 단한다. |  |
| 사람과 물건이 좌석에서 모두 치워지면 퇴실로 판단한다. |  |

|  |  |
| --- | --- |
| 또한 사람이 예약을 한 후 5분 동안 자리에 오지 않으면 퇴실 로 바꾼다. |  |

# 6. 프로젝트 수행 방법

## 6.1. 요구분석

* 사용자가 웹에서 좌석 현황을 확인할 수 있어야 하며 로그인 후 좌석 예약이 가능할 수 있게 장고와 웹을 연동시켜야 한다.
* 웹캠이 열람실 좌석 영상으로 열람실 좌석 현황을 체크하고, 서버가 요청할 때 마다 현황을 서버로 전송해줘야 한다.
* 장고에서 웹캠의 좌석 현황을 받았으면 예약자와 사용자를 동시에 확인하여 웹에 띄워줌
* 본 프로젝트는 웹캠과 서버를 필요로 함.

## 6.2. 설계

* 웹캠은 항상 영상을 찍으며 좌석 현황을 체크하고 있고, 서버의 요청이 있을 때 마다 좌석 현황을 서버로 보내준다.
* 서버에서 예약 현황과 좌석 현황을 합하여 사용현황을 사용자에게 웹으로 보여준다.
* 서버에서 사용자가 로그인을 하면 예약을 할 수 있고, 예약이 끝난 뒤 사용 현황을 볼 수 있다.
* 사용자가 자리를 떠나면 웹캠에서 확인 후 자동 퇴실된다.
* 서버와 웹캠 사이의 통신은 소켓통신으로 한다.
* 웹캠에서 각 좌석마다 좌표값을 부여하여 사람 사용 여부를 판단함. - 웹캠에서 사용자의 퇴실 여부를 판별하기 위해 각 좌석에 사용자의 물건 여부를 판단한다.

## 6.3. 구현

* python을 이용하여 웹캠과 장고, 둘을 연결하는 소켓을 구현 - 장고가 실행되는 서버는 아파치를 사용함.
* 장고에서 지원해주는 데이터 베이스를 사용.
* 소켓연결을 할 때 아파치를 클라이언트로, 웹캠을 서버로 하여 클라이언트 의 요청에 따라 서버인 웹캠이 정보를 보내줌.
* 웹캠에서는 사물과 사람을 인식하는 open cv를 사용, cvlib이라는 library로 구현.
* 웹캠에서 각 좌석마다 좌표값을 주고, 좌석과 사물, 사람의 거리를 계산하여 좌석을 사람이 사용하는지 판단하여 사용 현황을 구현.

## 6.4. 테스트

* 서버, 웹캠, 소켓통신 부분을 개별적으로 테스트.
* 서버와 웹캠의 통신을 소켓으로 만들어, 서로 정보 공유가 원활히 되는지 확인.
* 사용자가 서버에 접속하여 서비스를 사용할 때 문제가 없는지 테스트.

6.5. 유지보수

* 서버와 웹캠의 통신 연결의 오류를 없애야 함.
* 장고 웹에 화면이 잘 올라오는지 확인.
* 웹캠의 좌석 및 사물과 사람인식이 잘 되는지 계속하여 테스트 후 좌표 수정.

# 7. 각 분야 상세 설명

## 7.1. 웹캠

|  |
| --- |
| import socket import threading import cvlib as cv  from cvlib.object\_detection import draw\_bbox import cv2 import math  data = "0"  constant\_object = ['chair', 'dining table'] #의자와 책상은 제외하기 위해 추가함  def detect(): global data  region=[[130,40,150,80,0],[130,120,150,90,0],[70,180,230,200,0],[350,30,200,90,0],[350,130,100,75,0],[35  0,200,200,200,0]] #좌석 좌표 지정(이 좌표는 좌석의 왼쪽 위 꼭짓점의 좌표이다.) def onMouse(event,x,y,flags,param): if event == cv2.EVENT\_LBUTTONDOWN:  print(event,x,y, ) def seat\_use(bbox, label, region): #사용 현황 함수 str\_buffer = ['0','0','0','0','0','0'] #시작 좌석은 0 for i in range(len(region)): count = 0  midle\_seat = (region[i][0]+region[i][2]/2,region[i][1]+region[i][3]/2) #좌석 좌표의 중앙 좌표 for j in range(len(bbox)):  midle\_object=(bbox[j][2]-(bbox[j][2]-bbox[j][0])/2,bbox[j][3]-(bbox[j][3]bbox[j][1])/2)  #물건의 중앙 좌표  x\_dis=midle\_object[0]-midle\_seat[0] #물건과 좌석 사이의 x좌표 거리를 측정 y\_dis=midle\_object[1]-midle\_seat[1] #물건과 좌석 사이의 y좌표 거리를 측정 max\_dis=max(region[i][2]/2,region[i][3]/2) #물건과 좌표사이의 거리를 측정하여 사용  dis = math.sqrt((x\_dis\*x\_dis)+(y\_dis\*y\_dis)) #사용중인 좌석의 범위를 설정 if dis <= max\_dis and label[j] not in constant\_object: #물건이 좌석 범위안에 들어왔고, 그 물건이 책상과 의자가 아닐 때  count = 1  if (region[i][4]>5): #좌석 범위안에 들어온지 5번이 넘을 때 사용중으로 판단 str\_buffer[i] = '1' else:  str\_buffer[i] = '0' if count == 1: if region[i][4]<10: |

|  |
| --- |
| region[i][4] += 1  elif count==0: #잠시 물건이 좌석 범위에서 벗어나면 완전히 0으로 초기화 시켜주지 않고 -1을하여 오차를 허용해줌 if region[i][4]>0: region[i][4] -= 1 print(i,region[i][4]) return ''.join(str\_buffer)  webcam = cv2.VideoCapture(1) #웹캠 사용 코드 if not webcam.isOpened():  print("Could not open webcam") exit() while webcam.isOpened():  status, frame = webcam.read() if not status: break cv2.waitKey(1000)  bbox, label, conf = cv.detect\_common\_objects(frame) data = seat\_use(bbox, label, region)  print(data)  """ for i in range(len(bbox)): print(bbox[i], label[i], conf[i]) """  out = draw\_bbox(frame, bbox, label, conf, write\_conf=True) cv2.imshow("Cam", out)  cv2.setMouseCallback("Cam",onMouse) if cv2.waitKey(1) & 0xFF == ord('q'):  break    webcam.release() cv2.destroyAllWindows()  def send\_data(): #소켓으로 사용 현황을 전송해주는 함수 global data msg = data  client\_socket.send(msg.encode('utf-8')) client\_socket.close()  webcam\_thread = threading.Thread(target=detect) #스레드를 사용하여 여러 서버 접속을 가능하게함 webcam\_thread.start()  server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) server\_socket.bind(('192.168.0.11', 10007)) server\_socket.listen(0)  while True:  client\_socket, addr = server\_socket.accept() |

send\_thread = threading.Thread(target=send\_data) send\_thread.start()

## 7.2. 소켓

|  |
| --- |
| import socket import threading  def transmit(ip): #room model에 저장되어있는 ip 주소 받아옴 sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  sock.connect((ip, 10007)) #지정해놓은 포트번호와 스터디룸의 ip주소로 connect data = sock.recv(65535) #좌석현황 정보를 받아옴  d = data.decode('utf-8') print(data.decode('utf-8')) sock.close() return d |

7.3. 장고(모델)

|  |
| --- |
| from django.db import models  from django.contrib.auth.models import User import uuid  from django.urls import reverse  class Room(models.Model):  name = models.CharField(max\_length=200) #스터디룸 이름 location = models.CharField(max\_length=200) #스터디룸 위치 row = models.IntegerField(null=True) #가지고있는 좌석의 row수 column = models.IntegerField(null=True)#가지고있는 좌석의 column ip = models.CharField(max\_length=200)#해당 스터디룸의 ip  def get\_absolute\_url(self): return reverse('room-detail', args=[str(self.id)])  #language = models.ForeignKey('Language', on\_delete=models.SET\_NULL, null=True) #genre = models.ManyToManyField(Genre, help\_text='Select a genre for this book')  class Seat(models.Model):  seat\_id = models.UUIDField(primary\_key=True, default=uuid.uuid4) seat\_num = models.IntegerField(null=True)#좌석 번호 |

|  |
| --- |
| reserved\_time = models.TimeField(null=True,blank=True) #예약한 시간  room = models.ForeignKey(Room, on\_delete=models.SET\_NULL, null=True, blank=True) SEAT\_STATUS = (  ('u', 'Use'),  ('r', 'Reserve'),  ('a', 'Available'),  )  status = models.CharField( max\_length=1, choices=SEAT\_STATUS, blank=True, default='a',  help\_text='Present Sit Status', ) |

# 문제점 및 해결방안

1: 라즈베리 파이의 ultrasonic 센서를 이용하려 하였으나, 가격과 현실성에서 비효율적이라고 판단, 웹캠을 사용

1. 예약자가 10분 이상 자리를 비웠을 경우 예약을 취소하려 했으나 현실성이없다고 판단, 이 서비스는 제공하지 않음
2. 예약을 하고 난 뒤 바로 자리에 앉지 않으면 퇴실했다고 간주하여 예약을 취소해 버림, 예약 후 5분동안 기다려줌
3. 웹캠에서 사용 여부를 인식할 때 인식이 정확하게 되지 않은 채로 사용 여부를 바로 서버로 전송해 버림, 사용 여부 인식을 웹 영상에서 5번 이상 연속으로 사용 중이 뜰 때만 전송.

# 기대효과 및 활용분야

## 9.1. 기대효과

· 직접 열람실에 가지 않아도 인터넷으로 좌석 이용 현황을 확인할 수 있다.

· 예약 시스템과 CCTV를 이용한 자동 입/퇴실 시스템으로 효율적인 열람실 이용이 가능하다.

· 학교 열람실뿐만 아니라 유사한 시스템으로 운영 되는 다른 분야에서도 활용 가능하다. 9.2. 활용분야

· 좌석의 입/퇴실과 예약이 필요한 곳이라면 기존에 설치되어 있는 CCTV를 활용하여 간단하게 활용할 수 있다.

(활용 예시 1) 공공 도서관 열람실 기존의 공공 도서관 열람실은 예약과 입/퇴실 서비스가 제공되고 있다. 하지만 입/퇴실 시 이용자가 직접 입/퇴실 여부를 입력해야 하는 번거로움이 있다. 또한 퇴실 시 퇴실 처리를 하지 않는 일이 빈번하여 관리인이 수시로 확인하며 퇴실 처리를 해야 하는 수고가 있고, 그 사이 좌석을 이용하지 못하는 불편함이 존재한다. 하지만 CCTV를 이용해 자동으로 입/퇴실을 처리하는 본 제품은 위 문제점들을 해결하는 좋은 방안이 될 수 있다.

(활용 예시 2) 독서실 사설 독서실은 지정 좌석을 긴 기간동안 이용하는 시스템도 있지만, 열람실처럼 운영되는 곳도 적지 않다. 이런 경우, 본 제품을 활용하면 이용자는 좌석 현황을 미리 확인하고 원하는 시간에 예약할 수 있고, 관리자도 큰 수고 없이 좌석을 효율적으로 관리하여 수익 창출에 도움을 줄 수 있다.

# 프로젝트 세부 일정







# 팀원 개별 담당 업무

|  |  |
| --- | --- |
| 김가영 | 소켓통신 및 물체 감지 코드 개발 |
| 서우빈 | 전체 총괄 |
| 윤준영 | 프론트 엔드 및 보고서 |
| 이영상 | 프론트 엔드 및 장고 |
| 박재원 | 장고 및 보고서 |

# 느낀점

|  |  |
| --- | --- |
| 김가영 | 진정한 프로젝트에 대해 배울 수 있던 시간이 였습니다. 아이디어를 구상하고, 피드백을 받고 제안서를 작성하며 좋은 아이디어를 내는 법과 다같이 의견을 모으는 법이 얼마나 어려운 일이 였는지를 알게 되었습니다. 프로젝트를 진행할 때에도 단지 한 부분만을 제작하는 과제와는 달리 장고와 웹캠과 소켓 등 여러 부분을 만들어서 조합하는 것이 신선하면서도 매우 어려웠습니다. 하지만 결국 서로 통신이 되고, 데이터를 주고받으면서 웹에 화면을 띄우는 것을 보고 이 프로젝트가 주는 교훈이 “노력과 결과는 비례한다.”라는 것을 알게 되었습니다. 여러모로 저에게 뜻 깊은 프로젝트가 되었습니다. |
| 서우빈 | 장고와 웹에 대하여, 소켓 통신에 대하여, 오픈cv를 사용하는 법 등 종합적으로 많은 부분들을 배울 수 있었던 시간이 였습니다. 머신 러닝 등 새로 배우게 된 부분들로 인하여 배움이 늘어났고, 소켓 통신과 장고 등 원래 알고 있던 부분들도 직접 실습하고 실행시키면서 더욱 확실하게 배우게 되었습니다. 프로젝트를 할 때 아이디어를 짜고, 계획을 세우고, 각자의 분야를 종합하여 완성을 시키는 그 모든 경험들로 한층 더 성장한 기분을 느낄 수 있었습니다. |
| 윤준영 | 처음 제안서를 냈을 때, 교수님의 조언을 들으면서 많은 생각을 하게 되었습니다. 우선 저희는 이 프로젝트를 진행함에 있어 비용도 고려하지 못 했고, 여러가지 변수를 생각하며 이 정도 생각이면 충분히 보완한 거다 싶어 발표를 진행했고, 또한 프로젝트를 위해 필요한 소프트웨어 하드웨어를 알아보면서 우리가 생각하는 것 보다 더 많은 정보가 필요 하다는 걸 깨달았습니다. 이번 프로젝트를 통해 앞으로 |

|  |  |
| --- | --- |
|  | 다른 프로젝트를 진행할 때 어떻게 공부를 할 지, 어떻게 아이디어를 낼 지 더 깊이 진행할 수 있을 것 같습니다. |
| 이영상 | 프로젝트를 시작하기에 앞서 아는 것도 많이 없고 Django에 대한 이해도 잘 하지 못하였기에 걱정이 많이 되었다. 아직 내가 많이 부족하구나라는 생각이 가장 컸던 것 같다. 코드를 한줄 한줄 이해하는 것도 오래 걸리고 현재 코드 상황에 맞는 코드를 짜는 것도 쉽지 않았다. 그래서인지 프로젝트 수행에 조금 소극적으로 참여하게 되었던 것 같다. 하지만 이번 프로젝트를 진행하면서 모르는 부분을 새로 알아가고 부족했던 부분을 채워가면서 많은 것들을 얻은 것 같다. |
| 박재원 | 캡스톤 강의를 들으며 입학하고 처음으로 팀 프로젝트를 수행했는데, 저번 pygame 팀 프로젝트 때도 그랬지만 걱정을 많이 했습니다. 하지만 좋은 팀원을 만나 모르는 부분을 같이 공부해 가며 프로젝트를 완성해 가다 보니 성취감도 크고 더 재밌었던 것 같습니다. 처음 django를 배울 때는 몰랐는데 프로젝트를 수행해 갈수록 응용이 무궁무진하고 개발자의 역량에 따라 많은 것을 할 수 있다고 느꼈습니다. Django로 app을 제작하는 것도, 서로 맡은 부분을 합치는 것도, cam과 django를 연동하는 것도 처음엔 막막하고 어려웠지만 서로 아는 것을 공유하고 공부하면서 하나씩 해 나갈 때 마다 성취감을 느꼈습니다. 제안서에서 교수님께서 피드백 주신 부분을 완벽히 구현했다고 자신하긴 어렵지만 덕분에 더 열심히 할 수 있었고 완성된 작품에 자부심을 느낄 수 있었습니다. |