**디지털 핵심 실무인재 양성사업(K-Digital Training)**

**3조 3차 프로토타입**

**2023년 02월 07일**

## 서버

[클라우드](https://drive.google.com/open?id=13U5ewqhC10owJwYV1M8ocWgNUpUEP0M1) 참조

## 데이터베이스

#### 주차가능·금지구역 데이터베이스

| 세부 기능 | 주차가능·금지구역 위치 데이터 수집 및 전처리 | | |
| --- | --- | --- | --- |
| 작성자 | **박민정** | | |
| 세부 기능 정의  (기능 시나리오) | 1. 주차가능구역, 주차금지구역에 대한 위치 데이터 수집  2. 강남구에 해당하는 지점만 선별, 위도/경도로 변환, 인코딩 변환 | | |
| 설계 | - 위도/경도가 아닌 주소만 있는 데이터에 대해 TMap의 Geocoding api로 위도/경도로 변환  - utf-8로 저장하여 인코딩 변환 | | |
| 구축 / 구현 | 주차금지구역   * 지하철역 서울교통공사\_1\_8호선 역사 좌표(위경도) 정보 <https://www.data.go.kr/data/15099316/fileData.do?recommendDataYn=Y> 국가철도공단\_수도권9호선\_역위치 <https://www.data.go.kr/data/15041335/fileData.do> * 버스정류장 국토교통부\_전국 버스정류장 위치정보 <https://www.data.go.kr/data/15067528/fileData.do> * 택시승강장 서울특별시\_택시승차대 현황 <https://www.data.go.kr/data/15086412/fileData.do> * 어린이보호구역 어린이보호구역 위치도 <https://data.seoul.go.kr/dataList/OA-20448/S/1/datasetView.do> * 지상소화전 서울시 소방재난본부\_소방용수시설 <https://data.seoul.go.kr/dataList/OA-21306/S/1/datasetView.do>   주차가능구역   * 전동킥보드 전용 주차구역 서울특별시 강남구\_전동킥보드 주차구역 <https://www.data.go.kr/data/15107840/fileData.do> | | |
|
| 평가 |  | | |

| 세부 기능 | 주차가능·금지구역 위치 데이터 수집 및 전처리 | | | |
| --- | --- | --- | --- | --- |
| 작성자 |  | | | |
| 세부 기능 정의  (기능 시나리오) |  | | | |
| 설계 | **데이터** |  | | |
| **IoI** |  | | |
| **인프라** |  | | |
| **UI** |  | | |
| 구축 / 구현 |  | | | |
|
| 평가 |  | | | |

#### 회원별 점수 데이터베이스

| 세부 기능 | 회원 정보 데이터베이스 구성 | | | |
| --- | --- | --- | --- | --- |
| 작성자 |  | | | |
| 세부 기능 정의  (기능 시나리오) |  | | | |
| 설계 | **데이터** |  | | |
| **IoI** |  | | |
| **인프라** |  | | |
| **UI** |  | | |
| 구축 / 구현 |  | | | |
|
| 평가 |  | | | |

| 세부 기능 | 점수 데이터베이스 구성 | | | |
| --- | --- | --- | --- | --- |
| 작성자 |  | | | |
| 세부 기능 정의  (기능 시나리오) |  | | | |
| 설계 | **데이터** |  | | |
| **IoI** |  | | |
| **인프라** |  | | |
| **UI** |  | | |
| 구축 / 구현 |  | | | |
|
| 평가 |  | | | |

## 주차 점수

#### 이미지 분석 모델

| 세부 기능 | 이미지 데이터 수집 | | | |
| --- | --- | --- | --- | --- |
| 작성자 | **박민정** | | | |
| 세부 기능 정의  (기능 시나리오) | 1. 데이터 수집  2. 이미지 레이블 변환 | | | |
| 설계 | **데이터** | 1. 데이터 수집   * 자전거도로, 점자블럭, 횡단보도 레이블을 포함하고 있는 이미지 데이터 수집   2. 이미지 레이블 변환   * 레이블 파일을 YOLO의 형식에 맞춰 txt 파일로 변환 | | |
| **IoI** |  | | |
| **인프라** |  | | |
| **UI** |  | | |
| 구축 / 구현 | 1. 데이터 수집  [인도보행 영상](https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=189) 중 Surface Masking 데이터셋   * 자전거도로, 점자블럭, 횡단보도 레이블이 포함된 5,009개의 이미지 발췌  |  | 자전거도로 | 점자블럭 | 횡단보도 | | --- | --- | --- | --- | | 원 데이터의 레이블 | - 대분류가 bike\_lane | - 대분류가 braille\_guide\_blocks | - 대분류가 alley, 소분류가 crosswalk  - 대분류가 roadway, 소분류가 crosswalk | | 이미지 개수 | 1434 | 1610 | 3490 | | bounding box 개수 | 1698 | 2446 | 4930 |   2. 이미지 레이블 변환   |  | 기존 | 변환 후 | | --- | --- | --- | | 파일 확장자 | xml | txt | | 한 파일 당 이미지 | 30 ~ 120개 | 1개 | | 모양 | 다각형(polygon) | 사각형 | | 형식 | <polygon 대분류, 잘림여부, 위치, 객체 순서>  <attribute>소분류</attribute>  </polygon> | 클래스 아이디, 가운데x좌표, 가운데y좌표, x길이, y너비 | | 예시 | <polygon label="braille\_guide\_blocks" occluded="0" points="897.80,70.70;904.20,141.10;783.90,144.60;772.10,247.40;907.20,236.50;910.20,297.90;1052.60,289.20;1065.90,360.70;1224.70,349.00;1250.80,429.00;1415.60,416.10;1339.80,267.20;1198.10,276.70;1124.10,88.80;1010.70,94.00;1004.40,64.20" z\_order="4">  <attribute name="attribute">normal</attribute>  </polygon> | 2 0.5697135 0.2283333 0.3351562 0.3377778 | | | | |
|
| 평가 |  | | | |

| 세부 기능 | 모델 생성 | | | |
| --- | --- | --- | --- | --- |
| 작성자 | **박민정** | | | |
| 세부 기능 정의  (기능 시나리오) | 1. 모델 기본 설정 및 테스트  2. 모델 학습 | | | |
| 설계 | **데이터** | YOLO 이미지 처리 모델 생성 | | |
| **IoI** |  | | |
| **인프라** |  | | |
| **UI** |  | | |
| 구축 / 구현 | 1. 모델 기본 설정 및 테스트   * 마스크 쓴 사람과 쓰지 않은 사람의 데이터 활용 * YOLOv4-Tiny 모델을 재학습 * <https://github.com/AlexeyAB/darknet>   2. 모델 학습   * max\_iter 10000으로 설정하여 학습 | | | |
|
| 평가 |  | | | |

| 세부 기능 | 이미지 데이터 추가 수집 및 모델 생성 | | | |
| --- | --- | --- | --- | --- |
| 작성자 | **박민정** | | | |
| 세부 기능 정의  (기능 시나리오) | 1. 이미지 데이터 추가 수집  2. 이미지 데이터 레이블 변경  3. 주차금지구역 판별 모델 구조 변경  4. 모델 학습 | | | |
| 설계 | **데이터** | 1. 이미지 데이터 추가 수집   * 전동킥보드 이미지 데이터 수집 * 자동차 등 교통수단의 바퀴 이미지 데이터 수집   2. 이미지 데이터 레이블 변경   * 1차 프로토타입에서 수집한 이미지에 대해 sidewalk(보도) 레이블 추가 * 추가 수집한 이미지에 대해 레이블 파일을 YOLO의 형식에 맞춰 txt 파일로 변환   3. 주차금지구역 판별 모델 구조 변경   * surface 모델: 도로 노면에서 도보, 점자블럭, 자전거도로, 횡단보도 탐지 * kickboard 모델: 전동킥보드와 바퀴 탐지   4. 모델 학습 | | |
| **IoI** |  | | |
| **인프라** | AWS GPU 탑재 EC2 INSTANCE를 통하여 모델 이미지 학습 | | |
| **UI** |  | | |
| 구축 / 구현 |  | | | |
|
| 평가 | 모델 학습 결과   | 노면 | 킥보드 | | --- | --- | |  |  | |  |  | | | | |

#### 주차 이미지 판독

| 세부 기능 | 주차 이미지 촬영 및 서버 전송 | | | |
| --- | --- | --- | --- | --- |
| 작성자 | **조정희** | | | |
| 세부 기능 정의  (기능 시나리오) | 사용자가 촬영 버튼을 누르면 주차금지 예상 구역 사진을 찍을 수 있게 되고, 전송 버튼을 누르면 데이터가 백 앤드 서버로 전송된다 | | | |
| 설계 | **데이터** |  | | |
| **IoI** |  | | |
| **인프라** | 웹에 카메라 앱을 연동시켜 촬영 후 Django가 Image Data 수신 | | |
| **UI** | 카메라 앱 연동 버튼, 전송 버튼 구현 | | |
| 구축 / 구현 | -Django에서 url설정 후 view와 연결시켜 request요청시 view를 실행시킨다.  -Client가 url을 get으로 들어올시 interface HTML을 볼 수 있게 view를 설정한다.  -HTML 문서 Input태그에 camera속성을 추가한뒤 Form method=post,action=”백앤드 서버url”로 감싼다  -Client가 촬영후 전송시 image를 post로 보내게 하고 post로 받을 시 action하는 함수를 따로 구성한다. | | | |
|
| 평가 |  | | | |

| 세부 기능 | 주차 이미지를 이미지 분석 모델에 전송 | | | |
| --- | --- | --- | --- | --- |
| 작성자 | **조정희** | | | |
| 세부 기능 정의  (기능 시나리오) | Image Data를 받은 Django가 Model Container로 데이터 전달 및 분석결과 수신 | | | |
| 설계 | **데이터** | 생성한 모델에서 넘겨준 데이터 형식에 맞게 분석 및 전달하도록 Input과 output 코드를 수정 | | |
| **IoI** |  | | |
| **인프라** | Django에서 post로 받은 image데이터를 Model에 전송하고 분석결과를 수신할 수 있게 API를 구성한다 | | |
| **UI** |  | | |
| 구축 / 구현 | -구현된 모델에 fastapi를 통해 전달받을 url과 method 형식, dataform 및 다시 보낼 response 코드를 작성한다.,  -Django에서 Post로 image data request 받을시 url과 형식에 맞게 모델API에 재 request를 요청한다.  -받은 response를 해석한다. | | | |
|
| 평가 |  | | | |

| 세부 기능 | 주차 이미지 분석 | | | |
| --- | --- | --- | --- | --- |
| 작성자 |  | | | |
| 세부 기능 정의  (기능 시나리오) | 카메라 사진을 보냈을때 모델을 통해 금지구역이 있는지 판별하고 인터페이스에 사진 데이터와 판별 데이터를 나열한다. | | | |
| 설계 | **데이터** | -모델에 opencv base64를 통해 사진 데이터 uri 형성기능 추가  -모델 출력값 딕셔너리 형태로 변환 | | |
| **IoI** |  | | |
| **인프라** | 인프라 구성도:  kubernetes를 통한 svc deployment 배포 구조  Service 구성도:  Client(Request) => web server(NGINX) => BACKEND server(Django) => Model Server(fast api) 구조 설계 | | |
| **UI** | 홈 - 주차 가능 여부 점수 확인 버튼 | | |
| 구축 / 구현 | -Django에서 주차 가능 여부 점수 확인 버튼 생성 후 input type=Camera 속성으로 카메라 연동 => click시 view의 fileupload 실행  -Model을 Fast api package를 통해 api를 생성하여 사진 데이터 request 받을시 분석 데이터 및 uri 데이터 response하게 형성  -Django에서 Model api에 사진 데이터를 request 해 분석값 Response | | | |
|
| 평가 |  | | | |

| 세부 기능 | 주차 이미지 판독 점수 환산 | | | |
| --- | --- | --- | --- | --- |
| 작성자 | **박민정** | | | |
| 세부 기능 정의  (기능 시나리오) |  | | | |
| 설계 | **데이터** |  | | |
| **IoI** |  | | |
| **인프라** |  | | |
| **UI** |  | | |
| 구축 / 구현 |  | | | |
|
| 평가 |  | | | |

#### 주차 위치 판독

| 세부 기능 | 현재 위치 GPS 정보 서버 전송 | | | |
| --- | --- | --- | --- | --- |
| 작성자 | **조정희, 박민정** | | | |
| 세부 기능 정의  (기능 시나리오) | 1. 사용자의 현 위치에 대한 위도, 경도 정보를 서버에 전송  2. 사용자의 현 위치가 강남구인지 파악 | | | |
| 설계 | **데이터** | 현 위치에 대한 행정동 코드로 DB에 있는 강남구 행정동 코드와 일치하는지 비교 | | |
| **IoI** | 핸드폰 GPS 데이터 서버 전달 | | |
| **인프라** | Navigator geolocation 함수로 현 위치 위도/경도 불러옴 | | |
| **UI** | 위도/경도 정보를 input의 value로 설정하여 post | | |
| 구축 / 구현 | 1. 사용자의 현 위치에 대한 위도, 경도 정보를 서버에 전송   * Navigator geolocation 함수로 위도, 경도를 불러옴 * 불러온 내용 input(type=”hidden”)의 value로 설정 * 사용자가 전송 버튼 누르면 post   2. 사용자의 현 위치가 강남구인지 파악   * TMap의 ReverseGeocoding api로 현재 위치의 위도, 경도에 해당하는 행정동 코드를 가져옴 * pymysql로 DB 서버의 dong 테이블의 dongCode와 일치하는 값이 있는지 비교함 | | | |
|
| 평가 |  | | | |

| 세부 기능 | 현 위치와 주차금지구역 지점 사이의 거리 계산 | | | |
| --- | --- | --- | --- | --- |
| 작성자 | **조정희, 박민정** | | | |
| 세부 기능 정의  (기능 시나리오) | 1. 현 위치 행정동의 주차가능구역/금지구역 지점 데이터 불러옴  2. 현 위치와 각 지점에 대한 거리 계산 | | | |
| 설계 | **데이터** | RDS MariaDB server에 전처리 된 데이터 업로드 | | |
| **IoI** |  | | |
| **인프라** | Django Server Service Deployment 구축 및 배포 | | |
| **UI** |  | | |
| 구축 / 구현 | 1. 주차가능구역/금지구역 지점 데이터 불러옴   * pymysql로 DB 서버의 parkinzone 테이블과 cautionzone 테이블에서 현 위치의 행정동 코드와 dongCode가 일치하는 값 불러옴   2. 현 위치와 각 지점에 대한 거리 계산   * haversine 함수를 활용하여, 위도/경도를 바탕으로 현 위치와 각 지점의 거리를 계산함 | | | |
|
| 평가 |  | | | |

| 세부 기능 | 위치 판독 점수 환산 | | | |
| --- | --- | --- | --- | --- |
| 작성자 | **박민정** | | | |
| 세부 기능 정의  (기능 시나리오) |  | | | |
| 설계 | **데이터** |  | | |
| **IoI** |  | | |
| **인프라** |  | | |
| **UI** |  | | |
| 구축 / 구현 |  | | | |
|
| 평가 |  | | | |

#### 주차 기울기 판독

| 세부 기능 | 기울기 센서 구현 | | | |
| --- | --- | --- | --- | --- |
| 작성자 |  | | | |
| 세부 기능 정의  (기능 시나리오) |  | | | |
| 설계 | **데이터** |  | | |
| **IoI** |  | | |
| **인프라** |  | | |
| **UI** |  | | |
| 구축 / 구현 |  | | | |
|
| 평가 |  | | | |

| 세부 기능 | 기울기 센서 데이터 서버 전송 | | | |
| --- | --- | --- | --- | --- |
| 작성자 |  | | | |
| 세부 기능 정의  (기능 시나리오) |  | | | |
| 설계 | **데이터** |  | | |
| **IoI** |  | | |
| **인프라** |  | | |
| **UI** |  | | |
| 구축 / 구현 |  | | | |
|
| 평가 |  | | | |

| 세부 기능 | 기울기 판독 점수 환산 | | | |
| --- | --- | --- | --- | --- |
| 작성자 |  | | | |
| 세부 기능 정의  (기능 시나리오) |  | | | |
| 설계 | **데이터** |  | | |
| **IoI** |  | | |
| **인프라** |  | | |
| **UI** |  | | |
| 구축 / 구현 |  | | | |
|
| 평가 |  | | | |

#### 결과 도출

| 세부 기능 | 주차 점수 계산 | | | |
| --- | --- | --- | --- | --- |
| 작성자 |  | | | |
| 세부 기능 정의  (기능 시나리오) |  | | | |
| 설계 | **데이터** |  | | |
| **IoI** |  | | |
| **인프라** |  | | |
| **UI** |  | | |
| 구축 / 구현 |  | | | |
|
| 평가 |  | | | |

| 세부 기능 | 주차 점수에 영향을 준 요인 안내 | | | |
| --- | --- | --- | --- | --- |
| 작성자 | **조정희** | | | |
| 세부 기능 정의  (기능 시나리오) | 현 위치 위도 경도 데이터를 통해 criteria 범위 안의 주차 금지구역 데이터를 map으로 보여준다. | | | |
| 설계 | **데이터** | Maria DB에 주차 금지구역 데이터를 업로드한다. | | |
| **IoI** | 현 위치의 GPS 데이터 (위도,경도)값을 넘겨준다. | | |
| **인프라** | 인프라 구성도:  Kubernetes를 사용한 Ingress, service, deployment ,pod 구조  service 구성도:  Client(Request)-> Web Server(Nginx) -> Backend Server(Django),DB Server(RDS Maria DB)->Model Server(Fastapi) => Client(Response) | | |
| **UI** | 주차 금지구역 표시된 지도, 주차 가능 점수 표시 , 현재 근처 주차 금지구역 종류 및 거리 표시 | | |
| 구축 / 구현 | (Backend) Context 구성  -criteria내에 distance가 포함되는 데이터 list (Gps)  -uri(카메라)  -length kinds(카메라)    (Front) Map api를 통해 주차금지 구역 좌표를 통해 지도 좌표 표시를  img src={{uri}}를 통해 사진 이미지를  length kinds를 통해 주차금지구역 안내를 Front에 나열한다. . | | | |
|
| 평가 | https://www.pzone.site/test/fileupload | | | |

## 주차가능구역 경로 안내

#### 주차 가능구역 경로 안내

| 세부 기능 | 현재 위치 GPS 정보 서버 전송 | | | |
| --- | --- | --- | --- | --- |
| 작성자 | **조정희, 박민정** | | | |
| 세부 기능 정의  (기능 시나리오) | 1. 사용자의 현 위치에 대한 위도, 경도 정보를 서버에 전송 | | | |
| 설계 | **데이터** |  | | |
| **IoI** | 핸드폰 GPS 데이터 서버 전달 | | |
| **인프라** | Navigator geolocation 함수로 현 위치 위도/경도 불러옴 | | |
| **UI** | 위도/경도 정보를 input의 value로 설정하여 post | | |
| 구축 / 구현 | 1. 사용자의 현 위치에 대한 위도/경도 정보를 서버에 전송   * Navigator geolocation 함수로 위도, 경도를 불러옴 * 불러온 내용 input(type=”hidden”)의 value로 설정 * 사용자가 전송 버튼 누르면 post | | | |
|
| 평가 |  | | | |

| 세부 기능 | 현 위치에서 가장 가까운 주차가능구역 안내 | | | |
| --- | --- | --- | --- | --- |
| 작성자 | **조정희, 박민정** | | | |
| 세부 기능 정의  (기능 시나리오) | 1. 사용자의 현 위치와 예상 소요 시간이 제일 짧은 지점과의 경로 반환  2. 화면에 표시 | | | |
| 설계 | **데이터** | 위도 경도로 변환한 주차장 데이터를 DB 서버에 업로드 | | |
| **IoI** | 핸드폰 GPS 데이터 서버 전달 | | |
| **인프라** | Client(Request)->Web Server -> Back End Server+DB server ->Client(Response) | | |
| **UI** | 주차장 안내 버튼, 지도 및 가까운 주차장 좌표 안내, 빠른 길찾기 안내, 걸리는 시간 및 거리 안내 | | |
| 구축 / 구현 | (Back) 받아온 위도 경도 데이터와 비교하여 minTime 구함 (T-map api 사용)  (Front) minTime Data를 지도 api(T-map api) 사용하여 최적의 route,걸린 시간, 거리 interface에 나열  1. 사용자의 현 위치와 예상 소요 시간이 제일 짧은 지점과의 경로 반환   * pymysql로 DB에 저장된 주차가능구역 정보 불러옴 * TMap의 routes/pedestrian api로 주차가능구역 지점별 도보 경로 가져옴 * 예상 소요 시간이 제일 짧은 지점에 대해 경로 정보(point, line) 반환   2. 화면에 표시   * TMap 지도에 경로 표시 * 예상 소요 시간과 예상 거리 표시 | | | |
|
| 평가 | https://www.pzone.site/map/geolocation | | | |

## 

## 6.3 데이터베이스에 업로드

| 세부 기능 | 데이터 베이스에 업로드 | | |
| --- | --- | --- | --- |
| 작성자 | **조정희** | | |
| 세부 기능 정의  ( 기능 시나리오 ) | 주차 금지구역, 주차 가능 구역, 동 code CSV파일을 현 DB서버에 업로드 한다. | | |
| 설계( 데이터 ) | 주차 금지 구역의 dong code와 주차 가능 구역의 dong code를 동 code csv파일과 일치 시켜놓는다. | | |
|
| 설계( IoT ) | x | | |
|
| 설계( 인프라 ) | 업데이트된 DB server로부터 pymysql을 사용해 데이터를 불러온다. | | |
|
| 설계( UI ) | x | | |
| 구축( 구현 ) | - maria DB서버(RDS)에 접속한다.  - Table을 총 금지구역, 가능구역, 동code 세 개를 생성한다.  - column 을 csv column과 일치시켜 생성하는데 id는 PK dongcode는 FK(동code의 column에 연결)로 생성한다.  - csv 파일을 업로드 하는데 한글 값이 들어가므로 column 값과 csv 인코딩 값을 utfmb4로 일치시켜준다. | | |
|
| 평가 | https://drive.google.com/file/d/11pQLyX1J0cZ3YKq2Wr9BGqXy\_8X2zDmz/view?usp=share\_link | | |

## 서버 배포

| 세부 기능 | 서버 배포 | | |
| --- | --- | --- | --- |
| 작성자 | **조정희** | | |
| 세부 기능 정의  ( 기능 시나리오 ) | Clients가 구현한 서비스를 이용할 수 있게 서버에 배포한다. | | |
| 설계( 데이터 ) | x | | |
|
| 설계( IoT ) | x | | |
|
| 설계( 인프라 ) | - Client가 Application Load balancer 주소로 들어 왔을 때  Ingress에서 지정한 path를 통해 안내 서비스를 이용할 수 있게 설정한다, | | |
|
| 설계( UI ) | 웹에서 ALB Domain name에 Django의 설정한 url path를 입력할 시 서비스를 이용할 수 있게 한다. | | |
| 구축( 구현 ) | -local에서 Django runserver를 통해 서비스 작동 상태를 확인한다.  -Git push를 통해 repository에 업로드 한다.  -업로드 된 내용을 pull 및 세부 내용을 적어 DockerFile을 설정한다.  -Docker 서버에서 임시 배포를 통해 서비스가 작동 되는지 확인한다.  -Docker 서버에서 ECR push를 통하여 이미지를 repository에 업로드 한다.  -EKS control plane 서버에서 각 web server , backend server.,model server의 service 및 deployment를 배포한다.  -ingress.yaml에서 path를 수정하여 안내 서비스 path를 생성및configure한다. | | |
|
| 결과 및 평가 | 2차 프로토타입 시스템 구성도 참조 | | |