|  | [ 2월 1주차 프로젝트 수행일지] | | | |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| 프로젝트 타이틀 | | **프라이빗 주차 공간 & 관리 시스템** | | | |
| 프로젝트 팀명 | | 타요타요(1조) | **프로젝트 팀원** | (빅데이터) 김준호 정수연 (IOT) 배민지  (클라우드) 김도현 구경선 송경진 이재원 | |

※ 프로젝트 팀장은 해당 일지 작성 후 매 주 금요일 퇴실 전 구글드라이브에 업로드해주세요

| 1. 프로젝트 수행 계획 및 현황 | | |
| --- | --- | --- |
| 이름 | **금주 프로젝트 수행 내용** | **차주 프로젝트 수행 계획** |
| 공통 | 1. 각 전공별 공정 진행 및 특이사항 공유  2. Web UI, 데이터 구체 기획  3. Web 개발(IOT : 1명 / 클라우드 : 2명 : 총 3명) | 1. 각 전공별 공정 진행 및 특이사항 공유  2. Web개발  3. 1차 프로토타입 시연 및 보완 : 2. 7.(화) ~ |
| (빅데이터) | 1. 공공데이터 분석 및 시각화 마무리  2. 기존 OCR 모듈 작동 확인, 모델 직접 개발 | 1. Web 개발 지원  2. AI 모델 개발 마무리 |
| (IoT) | 1. 제품 구조물 제작 완료  2. Test용 차량 이미지 촬영  3. Web DB 개발 | 1. IoT 장비 전체 마무리  2. 주차장 이미지 촬영  3. Web 기능 개발 |
| (클라우드) | 1. 1차 AI모델 서버 배포 및 동작 확인 (GPU 인스턴스 새로 생성 및 VPC 별도 구성  Region별 지원 인스턴스 타입 상이)   * S3: 최신 수신 이미지 input 설정 (완) * 모델에 경로 설정(완) * 출력 결과 .txt파일 ->버킷 반환(완) * 모니터링 : 이미지 생성(수신)   시에만 실행(진행)  2.Web DB<-> RDS 연동(완)  3. 추출번호<-> 등록번호 비교 람다 코드 작성(완)   * RDS 속 회원 데이터와 S3 버킷 데이터와 비교하는 코드 | 1. Web 배포 (~ : 1차 시연 前)  2. SNS 알림 기능 활성화   * 데이터베이스 연동 후 SNS 문자 전송 코드 마무리 예정(~: 1차 시연)   3. AI 모델 서버 파일 모니터링 -> run Test |

| 2. 강사님 피드백 | |
| --- | --- |
| 빅데이터 문경미 강사님 | 1. 모델 1 (번호판 추출)과 모델 2 (숫자 추출)의 연결성  - 번호판 이미지로만 모델2 학습 진행했어야함  - 모델 1로 번호판 범위 제외하고 블라 처리 고려해볼 것  - 모델 2만 사용하는 방안 추천  2. 번호판 뒤 4자리로만 검증하는 것은 다른 차도 정답으로 인식 가능  - 산술적 확률이 0.01%이나 실제 문제 발생할 수 있으므로 보완사항에 해당  3. 좌표 조정 후 텍스트 변환 처리  - Yolo 내 좌표값 처리 기능 함께 찾아주시기로 함  - 4자리 텍스트화하는 기능 구현 우선적으로 수행할 것 |
| IoT  김구수 강사님 | 1. 주차공간 테이블과 사용자 테이블간의 외래키 연결의 어려움  - 사용자가 주차공간의 FK를 가지는 것이 아니라, 주차공간이 사용자의 FK를 가져야 함  2. 외래키 사용의 어려움  - related\_name을 설정하지 않으면 테이블명\_set 으로 이용해야함  - 사용자가 소유주차공간을 구분할 수 있게 주차공간 이름을 추가하는 것이 편리함  3. 현재 제작된 테이블 확인  - 문제 없음 |
| 클라우드  이우성 강사님 | 1. 이미지 -> 버킷(AI:EC2 마운트포인트) 수신시 run.py 실행 방법에 대한 질의   * python code내에서 dir에 img 이미지 수신여부 판단하고 수신시 AI모델 run : 직접 하드코딩 * 코드를 직접 ec2 안에서 run하여 demon 형식으로 동작하게 할 것(background 사용) * 또는 컨테이너(doker)이용 run할 수 있는 환경구성해놓고 작동 * 전자 방안의 경우 ubuntu(OS)내서 demon 상시 run 시켜야함(crontab? : 0초마다 실행) * incron(S3:마운트포인트-> IN\_CREATE -> YOLO.py )이용하여 모니터링 하는건 ubuntu (OS)자체에 대한 권한/기능이고 AWS(클라우드)시스템에서는 연동 지원 안함(불가능) .   2. 부가설명: Container는 개발환경과 실행환경이 다른 것을 보완해줌(패키지 버전 충돌 문제 해결)   * EC2 생성마다 실행환경을 그대로 진행할 수 있기 때문에 사용하는 것임. * 여러개의 컨테이너 자동화 시키는 것은 쿠버네티스를 사용해야함. * 한개는 Docker |