

Scenario Test Check Sheet		Project Name		Smart City(Pole)	
		Pole No.		1	
		Inspection Date		0000-00-00	
		Inspector			
No.	Test Item	Result		Remarks	
		Yes	No		
#1 관리자/직원/사용자 웹 서비스 시나리오 (RU10)					
1	공통) 디크모드/라이트모드			관리자(Admin)와 사용자(Member)용 웹 인터페이스를 통해 계정 관리, 신고 접수 및 처리, 데이터 조회, 영상 열람, 문의 접수 기능을 제공한다.	
2	사용자) 회원 가입 및 로그인 기능				
3	사용자) 회원정보 수정				
4	사용자) 실시간 영상 조회 (ESP32-CAM 연동)				
5	사용자) 문의 접수				
6	사용자) 실시간 채팅 상담 및 카카오톡 문의				
7	관리자) 직원 등록 및 삭제				
8	직원) 로그인 기능				
9	직원) 가로등 등록 및 삭제				
10	직원) 가로등 조회				
11	직원) 자가진단 센서를 통한 고장난 가로등 조회				
12	직원) 도로용 CCTV 확인 및 불법 자동차 단속				
13	직원) 인도용 CCTV확인 및 인도위 오토바이 단속				
14	직원) 문의 내역 확인 및 답변 기능				
#2 불법 주정차 감지 및 신고 시나리오 (RU04, RU08)					
1	ESP32-CAM이 정지된 차량을 촬영			ESP32-CAM으로 차량을 촬영하고, YOLOv8으로 번호판을 탐지한 뒤, Google OCR로 문자를 인식하고, 일정 횟수 이상 발견 시 시청 API를 통해 자동 신고한다.	
2	YOLOv8이 번호판 탐지				
3	Google OCR이 번호판 문자를 인식				
4	Flask 서버가 DB에 저장				
5	동일 번호판이 일정 횟수 이상 발견되면 시청 API로 신고				
#3 오토바이 인도 주행 감지 및 인도 분석 (RU04, RU09)					
1	ESP32-CAM이 인도 영상 촬영			ESP32-CAM으로 인도를 촬영하고, YOLOv8으로 인도 위 오토바이를 탐지하여 영상 저장 후, DB에 기록된다.	
2	YOLOv8이 오토바이 탐지				
3	Flask 서버가 영상을 저장하고 위치 데이터를 DB에 기록				
#4 SOS 비상 신고 시나리오 (RU05)					
1	SOS 버튼 누름			비상 상황 발생 시 사용자가 SOS 버튼을 누르면 LED 경고, 부자 작동, 실시간 영상 전송, 경찰/소방 API를 통한 자동 신고가 수행된다.	
2	LED 점멸 * 부자 작동				
3	ESP32-CAM이 실시간 영상 전송				
4	Flask 서버가 영상 수신				
5	경찰 API를 통해 신고 전송				
6	관리자 웹에서 신고 내역 확인				
#5 가로등 센서 모니터링 및 고장 감지 (RU03, RU06, RU07)					
1	센서 값 실시간 수집 (조도, 기온기, 온습도)			조도, 기온기, 온습도 센서를 통해 가로등 상태를 실시간으로 모니터링하고, 고장이 의심될 경우 고장난 가로등에 등록된다.	
2	메인 조도센서 값이 기준값에 도달 시 LED제어 (400보다 작을때 : 켜짐, 200보다 작을때 : 더 밝게 켜짐)				
3	기준치를 벗어나는 경우 LED 고장 판단				
4	가로등 기온기가 기온어져 tilt_angle 값이 0 -> 1로 변경시 고장 판단				
5	Flask 서버로 전송 및 DB 저장				
6	웹 대시보드에서 이상 상태 시각화				
#6 중앙제어 앱 및 웹 기반 LED 제어 기능 확인 : 유지보수 (RU03, RU10)					
1	앱에서 [LED ON], [LED OFF], [AUTO_MODE] 버튼 클릭 시 명령 전송 확인			사용자가 MIT App Inventor로 만든 중앙제어 앱 또는 웹사이트에서 가로등 제어 버튼을 클릭하여 LED를 ON, OFF하거나 자동 모드(AUTO_MODE)로 전환하는 기능을 점검한다.	
2	웹사이트에서도 동일한 LED 제어 기능 테스트				
3	Flask 서버에서 제어 명령 수신 및 아두이노로 전송				
4	실제 가로등의 LED 상태가 명령대로 변경되는지 확인				
5	웹과 앱에서 제어할 때 Restful API Check 값이 0->1변경 확인				
6	DB sensor 테이블에서 insepction 컬럼값이 1로 저장(가로등 수리/점검으로 판단)				
#7 CI/CD 자동 배포 및 유지관리 (RU11)					
1	GitHub에 코드 푸시			GitHub Actions를 통해 코드 변경 시 자동으로 Flask 백엔드 및 MySQL 데이터베이스를 업데이트 하고, Ubuntu 서버에 배포가 이루어진다.	
2	GitHub Actions 워크플로우 실행 확인				
3	발드 및 배포 로그 확인				
4	서버 접속 후 업데이트된 기능 확인				
#8 비기능 요구사항 만족 테스트 (RU12)					
1	센서 데이터 저장 및 처리 성능 (최적화 여부)			보안, 안정성, 확장성, 가용성 등 시스템 운영에 필수적인 비기능 요구사항을 점검한다.	
2	가로등 수 증가 시 성능 저하 여부 확인				
3	서버 장애 발생 시 자동 복구 여부 (Git 자동 배포 재시작 테스트)				
#9 시스템 아키텍처 기반 통합 점검 (RU13)					
1	프론트엔드 ↔ 백엔드 ↔ DB 흐름 점검			시스템 아키텍처 구성 요소 간의 연결 상태 및 통합 흐름이 설계대로 작동하는지 전체적으로 점검한다.	
2	YOLOv8 ↔ OCR ↔ Flask 연동 점검				
3	하드웨어(ESP32, 센서) ↔ 서버 연동				
4	모든 외부 API(Google Vision API, 임시 경찰청, 임시 소방서) 호출 및 응답 확인				
※ Remarks					
-					