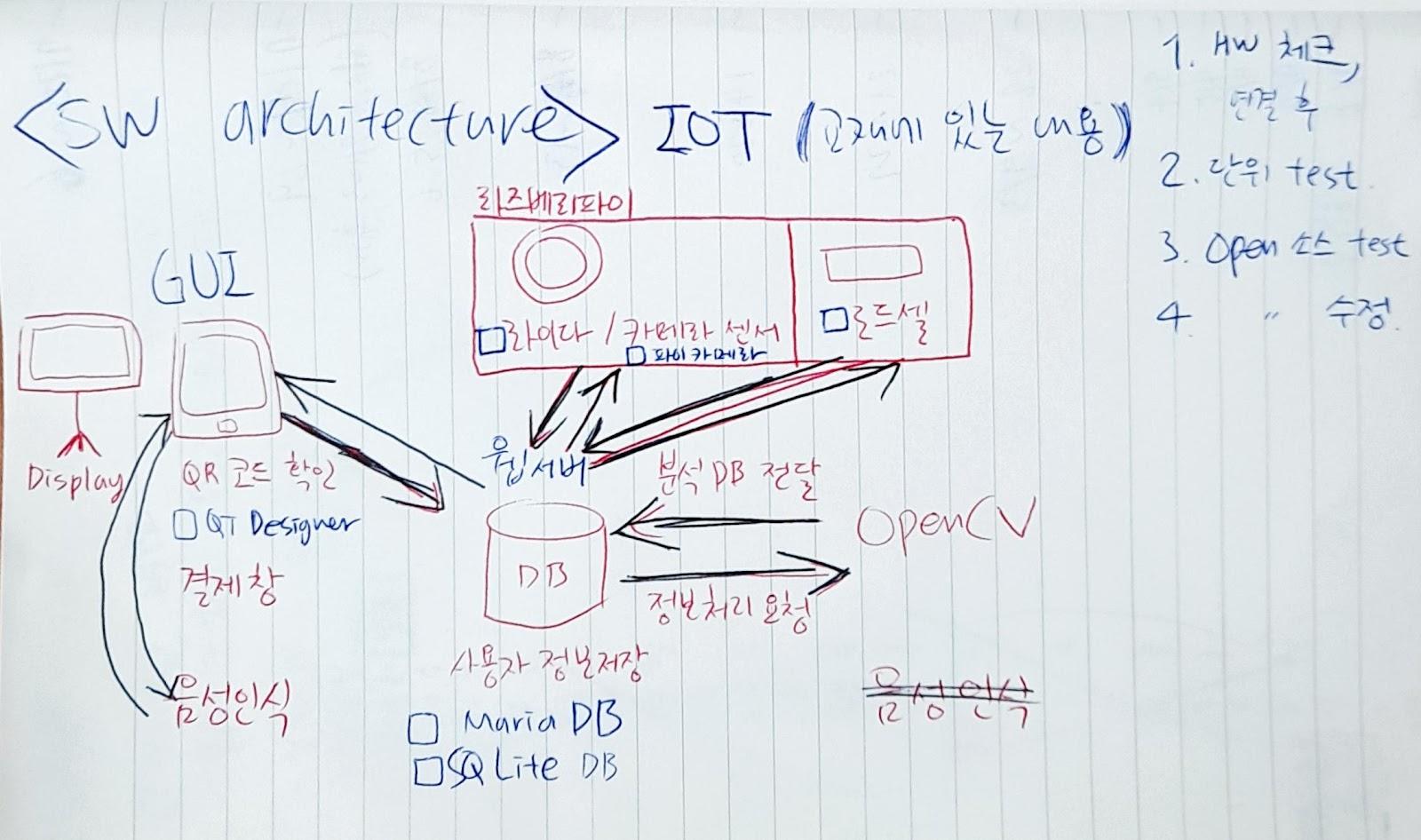
<SW architecture>

​

로드셀

[아두이노, 라즈베리파이에서 로드셀 사용하기](https://m.blog.naver.com/simjk98/221756769519)

​



​

​

비용

서버비:

메인 PC에 DB에 자료 저장.

웹서버 까는 건 본격적으로 서비스 하거나, 장기간 유지해야 될 때.

지금 우리가 하는 거는 프로젝트이므로 웹서버를 따로 만들어야 할 필요 없음.

​

reasonable: 사람 인건비 + 바코드 찍는 기계값 + 편리성에 대한 값 > SW 이용료(1달 이용료)

​

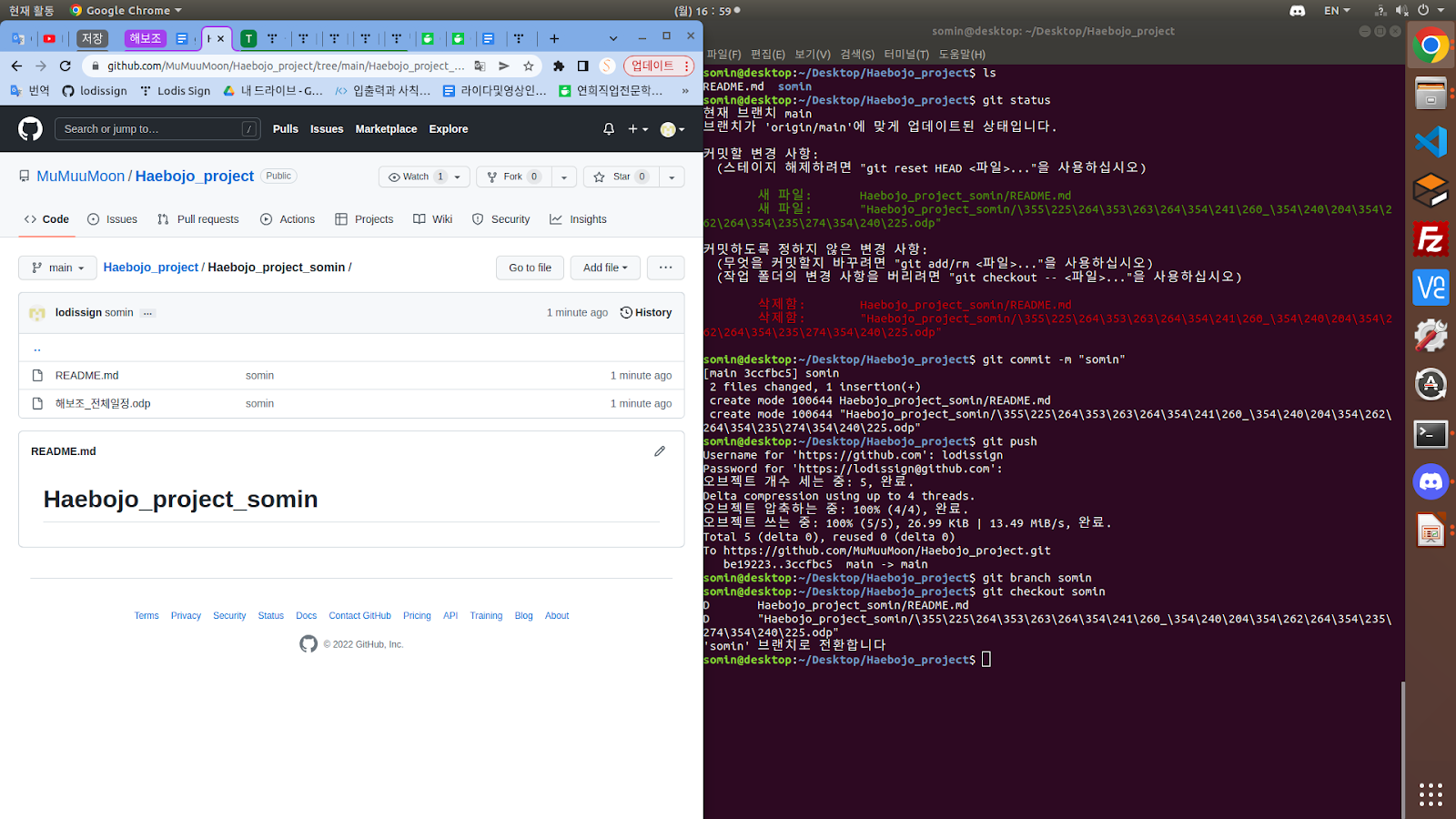
Display

​

​

​

git push, branch 생성.



​

​

하드웨어 시나리오 참고하여 전체 일정 수정하기

​

소프트웨어아키텍쳐를 완성하기 전에 글로써 옮기고 그려보고자함

.사람 입장

.입장한 사람이 qr코드로 체크인

.qr코드로 보낸 정보를 pc에서 받아서 정보저장

.정보저장을 DB(sqlite)에 하여 업데이트가 가능하게 만듦

.입장과 동시에 라이다 카메라 센서로 입장한 사람을 지속적으로 추적

.최소목표는 opencv로 비교하면서 단순히 픽셀값이 많이 변하는 곳에 대해서만 변화를 감지해서 네모로 표시

.최대목표는 machine-learning 으로 사람인지 동물인지를 학습하여 사람만 추적하게 만드는 기술

.입장한 사람이 물건근처로 간다

.입장한 사람이 물건을 집는다

.물건을 집은 곳의 무게를 로드쉘이라는 무게센서가 변화를 감지

.물건이 빠져나간곳에 무게의 차이를 재서 몇개인지 알아냄

.무게를 감지한 센서가 iot통신으로 메인pc로 정보전송

.몇그람빠진 것에대해 해당 위치에 가격까지 알고있음(메인pc에서할일)

.메인컴퓨터에 어디위치한 물건이 빠져나갔다는것 감지

.물건이 빠져나간곳에 대한 위치를 pc로 전송

.사람이 위치한 곳에 정보또한 전부 전송

.우영님<a와b가입장하여 로드쉘이 반응한 위치정보에 가깝게 있는

사람의 위치를 비교하여 해당 객체의 db값을 업데이트>

.태훈님<물건이 빠져나간곳에 대한 위치와 모든 사람들의 위치정보를 비교>

.두 위치가 일치하는 곳에 대한 사람의 정보를 따로 저장

.메인pc에서 빠져나간곳에 대한 가격정보와 위치정보를 저장

.메인pc에서 해당위치에 있었던 사람정보를 알고

.메인pc에서 해당위치에 빈만큼의 물건의 수량과 가격을 곱한값을 알고

.해당 위치에 있었던 객체(사람)의 정보를 불러와서

.해당 위치에 있었던 사람의 DB 컬럼중에서 구해진 가격을 업데이트(플러스)

.매장을 나갈때 센서가 해당 사람의 정보(ID)를 알아내어(이게 가능한 이유는 추적중)

.매장을 나간사람의 DB를 조사하여 만약 가격 컬럼에 가격이 업데이트되어있다면

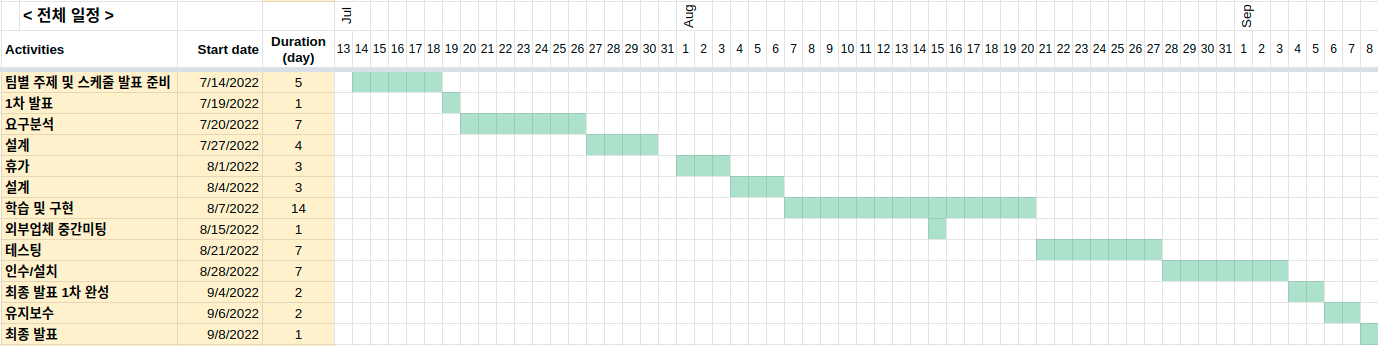
.그 업데이트 된 가격의 인트값을 알아내어

.인트값을 가격으로 환산하여

.qr로 찍었던 사람의 개인정보(ID PW)를 토대로

.매장에서 사람이 나갈때 그 사용자의 별다른 승인없이 결제를 대신 함

​



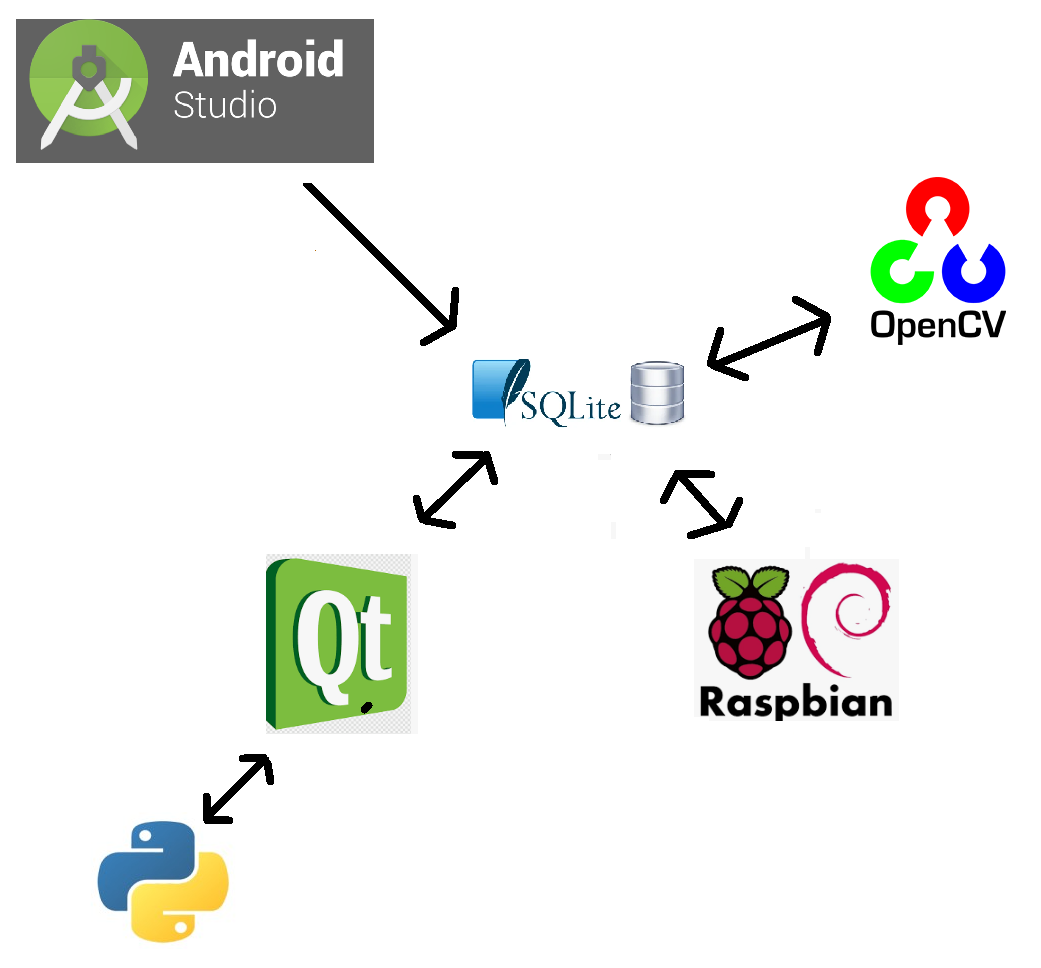
​

​

​

​

SW architecture 깔끔하게 수정하기



​

개인 일정 짜기

Raspbian

라이다/카메라 조사(open 소스 등)

[TFmini Plus LiDAR 아두이노에서 라이다(라이더)를 사용해보자! : 네이버 블로그](https://blog.naver.com/roboholic84/221740028135)

​

​

소민>

​

요구분석:

필요한 라이다 센서값 조사.

어떤 라이다 쓸 건지 정하기.

​

설계:

HW 연결 방법 조사

(라이다 카메라 센서값

라즈비안 or 메인 컴 둘 중

어디로 전송할지)

​

학습 및 구현:

단위 test([응용 프로그램에서 테스트 가능한 가장 작은 소프트웨어를 실행하여 예상대로 동작하는지 확인](https://tecoble.techcourse.co.kr/post/2021-05-25-unit-test-vs-integration-test-vs-acceptance-test/))

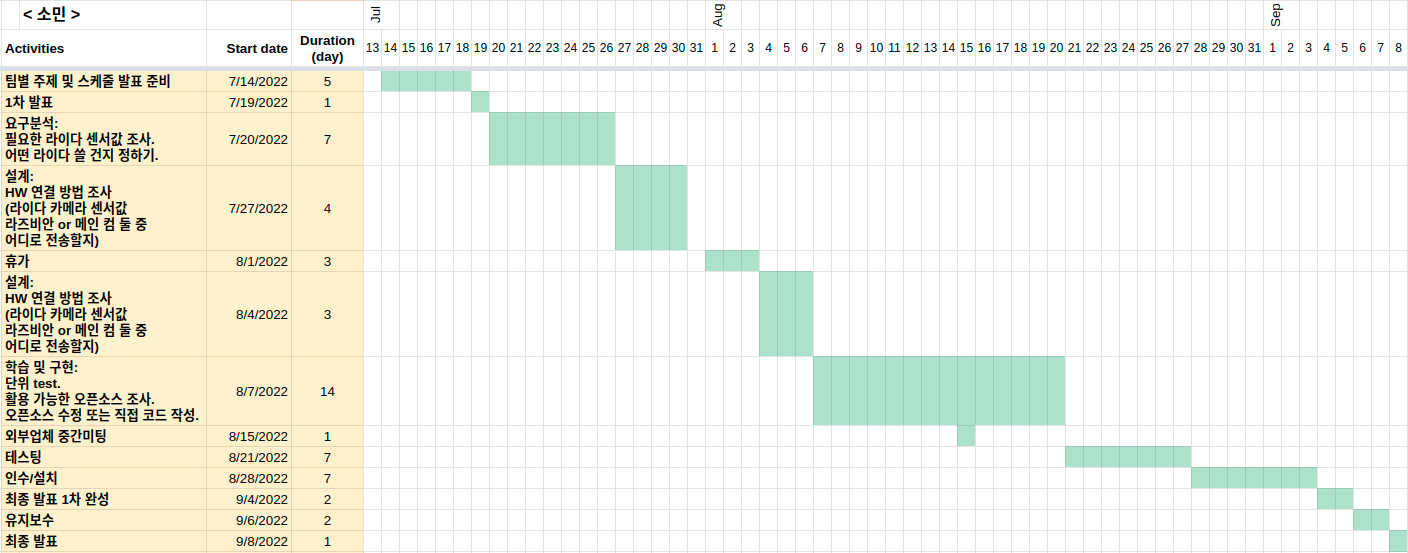
활용 가능한 오픈소스 조사.

오픈소스 수정 또는 직접 코드 작성.

​

테스팅

​



​

​

​

태훈님 >

​

7/14 ~ 7/17 - 주제선정 및 자료조사

7/18 - 자료조사 및 PPT 준비

7/19 - PPT 발표 및 일정 수정 보안

7월 20일 ~ 7월 22일 qr체크인하고 pc와 통신방법 조사

7월 25일 ~ 7월 29일 DB 설계5단계

8월 1일 ~ 8월 3일 휴가

8월 4일 ~ 8월 5일 DB 구축한것 이상없는지 테스트

8월 8일 ~ 8월 12일 DB 구축한것을 토대로 정보를 받아와 연동하는 코드 구현

8월 15일 ~ 8월 21일 연동된 시스템과 db사이에 정보가 잘 오고가는지 테스트

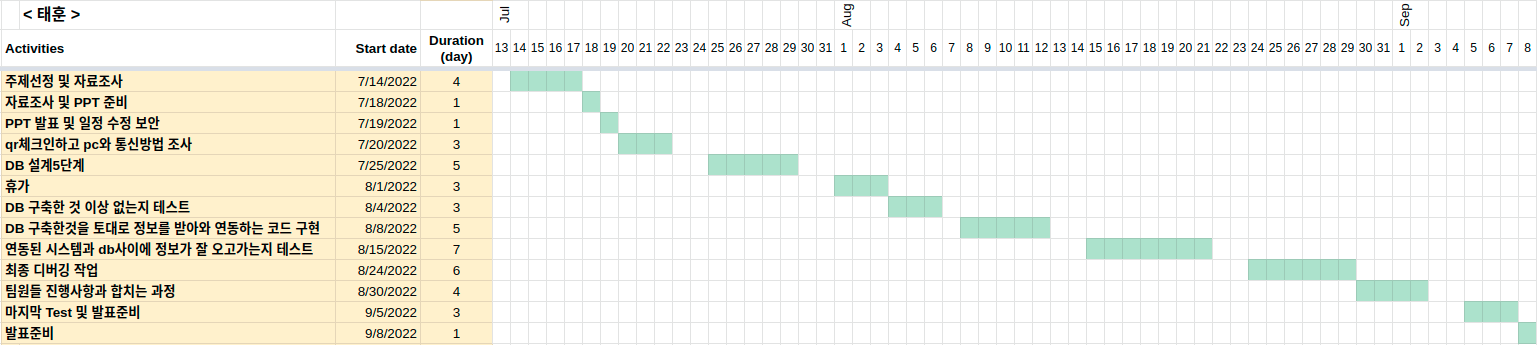
8월 24일 ~ 8월 29일 최종 디버깅 작업

8/30 ~ 9/2 - 팀원들 진행사항과 합치는 과정

9/5 ~ 9/7 - 마지막 Test 및 발표준비

9/8 - 발표준비

​



​

우영 >

​

7/14 ~ 7/17 - 주제선정 및 자료조사

7/18 - 자료조사 및 PPT 준비

7/19 - PPT 발표 및 일정 수정 보안

7/20 ~ 7/22 - 라즈베리파이와 아두이노보드 연결

7/25 ~ 7/29 - 아두이노와 라이다센서 연결하는 소스코드 분석 및

아두이노와 라이다센서 연결

8/1 ~ 8/3 - 휴가

8/4 ~ 8/12 - 아두이노와 로드쉘 연결하는 소스코드 분석 및

아두이노와 로드쉘 연결

8/15 ~ 8/21 - IOT 통신으로 아두이노 연결 방법 및 연결

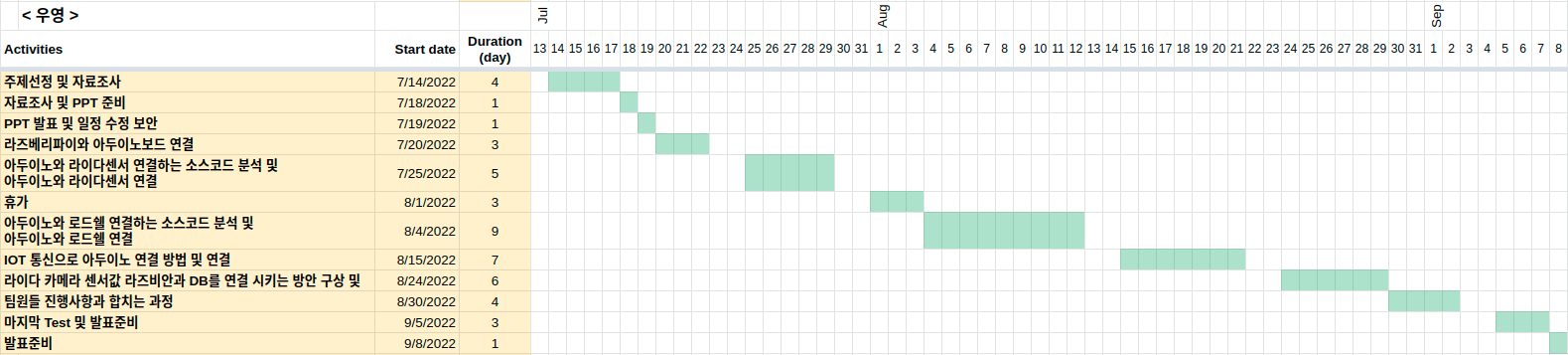
8/24 ~ 8/29 - 라이다 카메라 센서값 라즈비안과 DB를 연결 시키는 방안 구상 및

8/30 ~ 9/2 - 팀원들 진행사항과 합치는 과정

9/5 ~ 9/7 - 마지막 Test 및 발표준비

9/8 - 발표준비

​



​

재경>

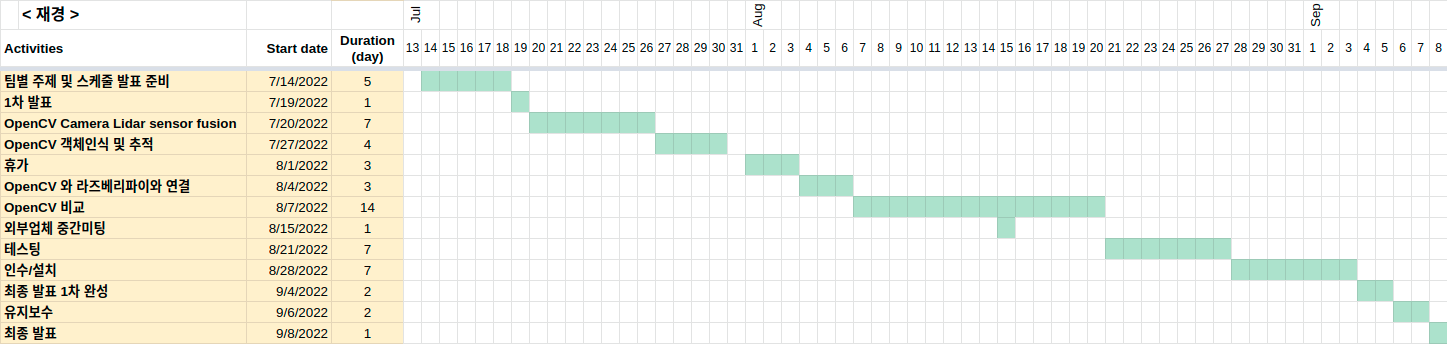
OpenCV Camera Lidar sensor fusion

OpenCV 객체인식 및 추적

OpenCV 와 라즈베리파이와 연결

OpenCV 비교

​



​

​

​

프레젠테이션 준비하기.

- 1교시 제출 및 발표 항목

1. 주제 선정 (문제 정의(needs or wants) & 시장(기술 가치) 조사 & 기술(상용, 오픈소스) 조사)

2. 개발 계획 (어떻게 해결해 갈지)

3. 역할 분담

4. 일정 (개발 계획 및 역할 기반)

\* 팀 일정 1일 단위

\* 개인 일정 1일 단위

​

- 1교시 발표관련

\* 각 조당 10분 발표(시간 엄수)

\* 피드백 (동료 5분, 교수 5분)