0719TUE

​

1차 발표 피드백

meatPlatter

​

배지 선정: Grodan (단위 면적 당 흙의 5배 효과)

​

센서 연결:

아두이노 & 직접/간접 연결?

​

AI 모델 입력값: 센서값?

​

실내 디자인을 목표로. 인테리어에 집중하기 보다 완전 제어형으로 본격적으로 키우고 싶은 사람을 위함.

​

면적, 비용 최소화를 위해 분리형. 기존 제품과의 호환성도 생각.

​

교수님:

왜 아쿠아포닉스? 1줄 요약(발표 시간 줄이는 방법)

: 지금까지 배운 기술 다 보여주기와 제품 가치가 충분하다고 생각.

​

발표 시간 1,2분 오차는 이해 가능.

​

대본에는 핵심적인 키워드만(몸, 시선 왔다갔다 하지 않게)

​

통계자료 활용한다는 건 주장 뒷받침 잘 할 수 있게 하기 위함.

반려동물 시장 커진다는 얘기에서 물고기에 대한 얘기 했어야 함.

​

엑셀로 정리 후 출처 표시해서 깔끔하게(캡처해서 줄 긋기 X)

​

결과물?

가지도 키우고, 센서 이용 제어. 디자인은 미정.

​

2달 기간 내 가지가 없다가 생장하는 모습을 볼 수 있나?

y data가 나올 정도로 성장이 가능하지는 않다. 센서 data x값을 통해서 예측하는 방향으로.

​

y data, accuracy가 안 나오는데 어떻게 예측?

​

y data가 나와야 유의미한 이유가 있다.

​

상추가 빨리 자라기 때문에 y data가 나와서 빨리 학습을 할 수 있다.

​

생물이 자라는 걸 어떻게 시뮬레이션?

​

좀 더 고민해봐야 할 듯.

​

​

​

SEED

​

스마트 픽킹로봇(AMR)

​

AMR은 첨단기술이지만 AGV 보다 저렴한 솔루션.

​

사용자 추적 기술:

OpenCV의 Tracking 기술- MeanShift

​

최대목표: GUI 환경에서 원하는 지점으로 이동, 카메라로 인식 가능한 코드들로 부터 위치정보 호출.

​

목표 시장: 물류 산업 쪽. 대규모에서 쿠팡이나 아마존은 선반을 들어서 옮기는 방식이고, 저희는 소규모 시장 목표.

​

시뮬레이션 상태에서는 가상의 맵 구현, 실제 RC카에서는 실시간으로 맵을 구성하는 방식.

​

꼬부기로만 하기 보다 RC카로 먼저 해보며 훼손의 위험성과 바로 시도해보기 어려운 면 보완 위해 로봇 2개 선택.

​

교수님:

바코드인식 관련은 오픈소스 활용하면 금방 끝날 것 같다.

구현하고자 하는 목표가 확실해야 한다.

​

면접에서 구체적으로 한 게 뭔지 질문함. 다운 받은 오픈소스에서 어떤 걸 수정했는지 반드시 물어본다.

​

​

​

해보조

​

결과물에 대해 너무 추상적이다. 구체적으로 어떻게 보여줄지 더 생각해봐야 함.

시뮬레이션을 어떻게 보여줄지 정해져야 정확한 예산도 나온다.

​

HW/SW architecture 잘못됨. 예시 올려주셨던 거 다시 보기.

​

​

​

dragon

​

Sign digits language Translation

​

오픈소스 딥러닝 기술 중 손동작 예측 관련 많았음.

​

YOLO의 One-stage 방식으로 정확도는 낮지만 속도는 빠름.

​

개발환경:

HW: PC, USB cam, Arduino, led.

SW: ubuntu 18.04, python 3.8, opencv, 플랫폼: Tenserflow or Keras or PyTorch, 가상개발환경: colaboratory or jupyter lab.

ref.: Github, 유튜부, arduino lab…

​

최소 목표: 숫자 수어 표현 인식-> LED.

최대 목표: 수어.

​

수어랑 제스쳐 구별 기능? 나중에 dataset에 추가하면 됨.

​

공부가 목표.

​

교수님:

미니프로젝트(1~2주 단위) 수준.

오픈소스 실행되게 구현하는 것도 중요하긴 함. 추가적으로 한 일이 중요.

왜 하는지에 대한 이유가 필요.

사회적으로 어떤 문제가 있어서 이걸 만들었다는 이유는 있지만 공부해보고 싶다고 해서 만으론 안된다.

공부만으로만 생각하면 프로젝트의 의미가 없다.

기업에서 발표하는 거라 생각하고 주제를 재구성 해야 함.

실패한 것에 대한 원인과 해결과정으로 어필하면 됨.

어떤 부분에 대한 문제를 해결하고 싶은지에 대해 생각해보면 될 듯.

사람들이 납득했다면 질문이 안 나온다.

​

​

​

​

2시반까지 각자 맡은 부분 하고 다시 모이기.

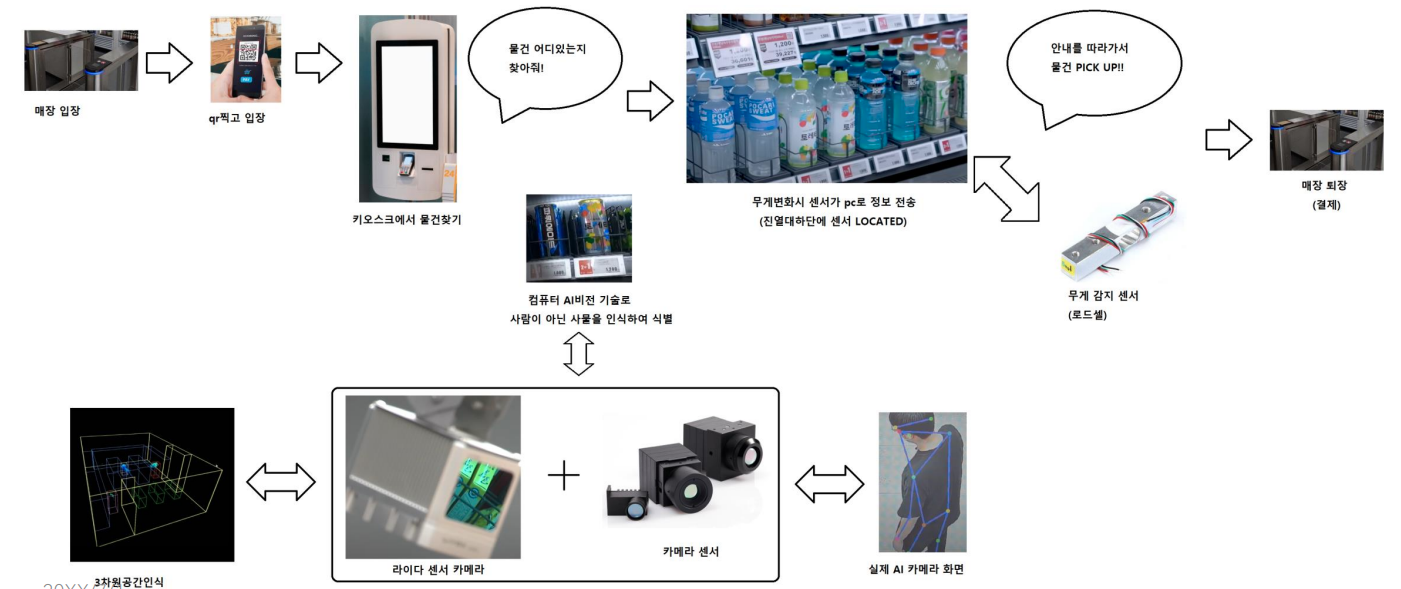
​

HW/SW architecture 수정하기

​

HW architecture

​



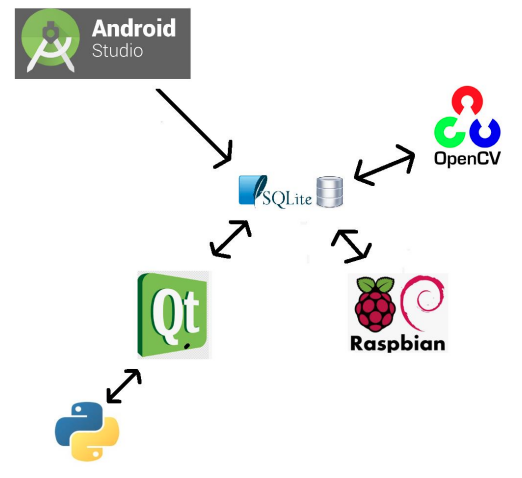
: 수정 전

​

​

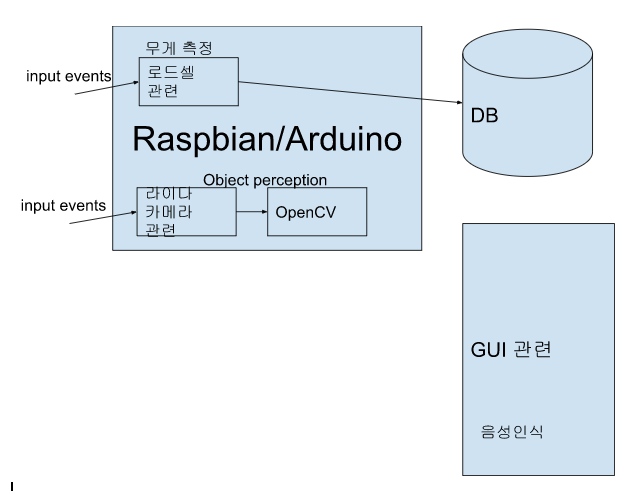
​

SW architecture



: 수정 전.

​



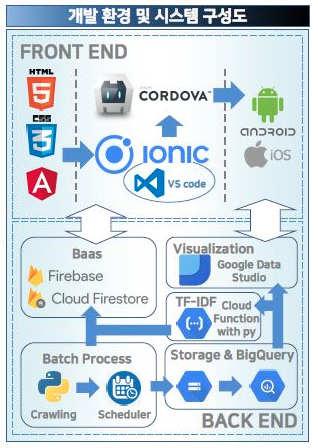
: 수정 후(진행 중)

​

​

참고 자료:

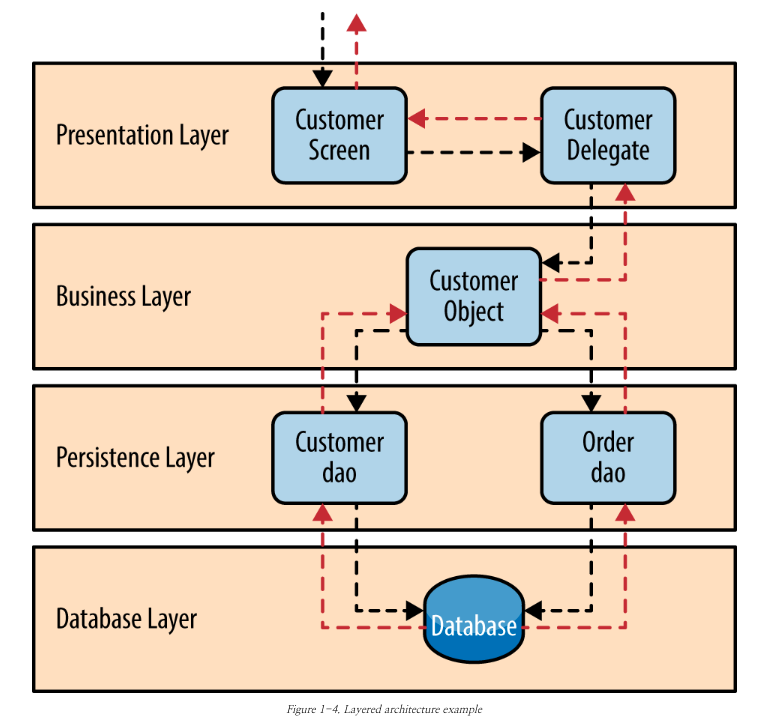
​



<https://camo.githubusercontent.com/928cce4c82a243050c36655587d851f85341723733ba90ecfc12200fad43992b/68747470733a2f2f696d6775722e636f6d2f4d4c34666867532e706e67>

​

​



[1. Layered Architecture - Software Architecture Patterns [Book]](https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html)

Presentation Layer에는 사용자 인터페이스나 브라우저 통신 등을 처리한다. Business Layer에서는 실제적인 서비스를 구현하는 비즈니스 로직이 있고, Persistence Layer는 데이터베이스에 객체를 매핑시키는 역할을 하고, Database Layer에서는 이를 처리한다.

이 구조의 핵심은 수행하는 동작을 분리시키고, 하위 레이어에만 의존하도록 하는 것이다. 예를 들면 Presentation Layer는 하위 레이어인 Business Layer에만 의존하여 동작한다. 그 아래 계층인 Persistence Layer에서 무슨 처리를 하는지 알 수 없으며, 변화가 생기더라도 문제가 없다.

이 구조의 장점은 우선 각 계층이 확실하게 분리되어 있기 때문에 테스트 하기가 쉽다는 점이다. 각 계층을 유연하게 변경할 수 있다. 예를 들면 가장 상위 계층인 Presentation Layer를 변경하면 시스템 전체를 수정하지 않고도 웹에서 동작하던 것을 앱에서 동작하게 할 수도 있다.

하지만 한 기능을 수행하기 위해서 여러 계층을 거쳐서 동작하기 때문에 성능이 낮고, 가장 하위 계층인 데이터 베이스에 의존적이라는 단점이 있다.

<https://velog.io/@pond1029/software-architecture>

​

​

[[번역] 10가지 소프트웨어 아키텍처 패턴 요약](https://mingrammer.com/translation-10-common-software-architectural-patterns-in-a-nutshell/)

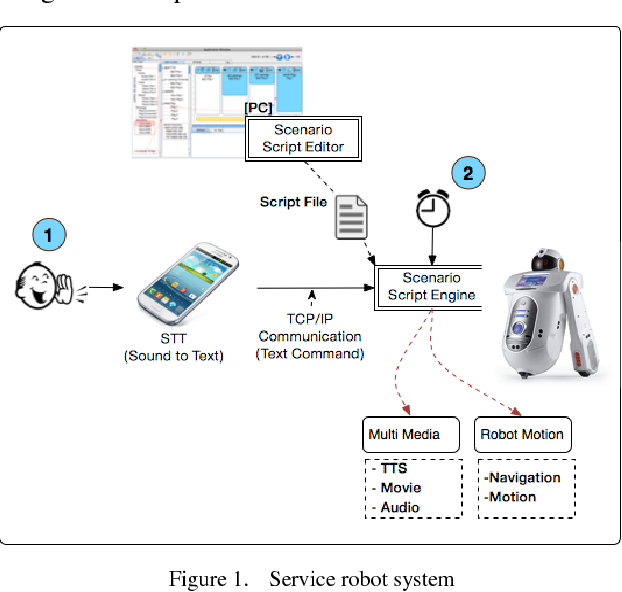
​

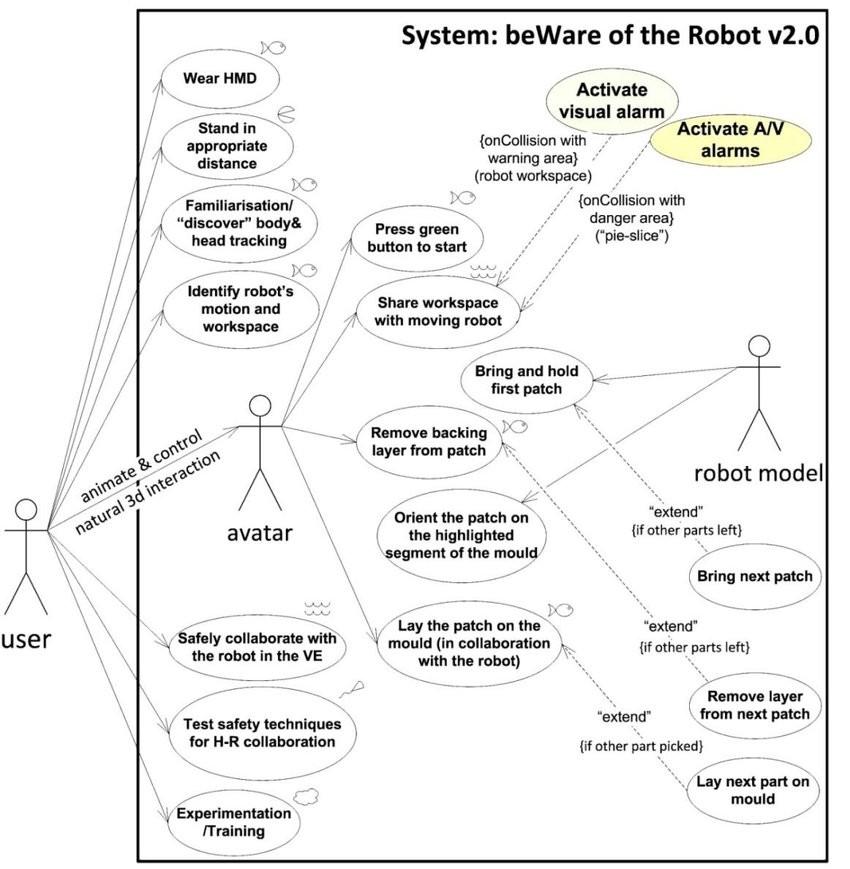
[GUI 프로그래밍]

미니프로젝트\_1\_주제\_및\_일정

​

- robot service scenario & use case diagram

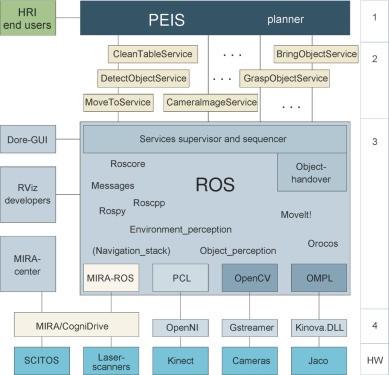


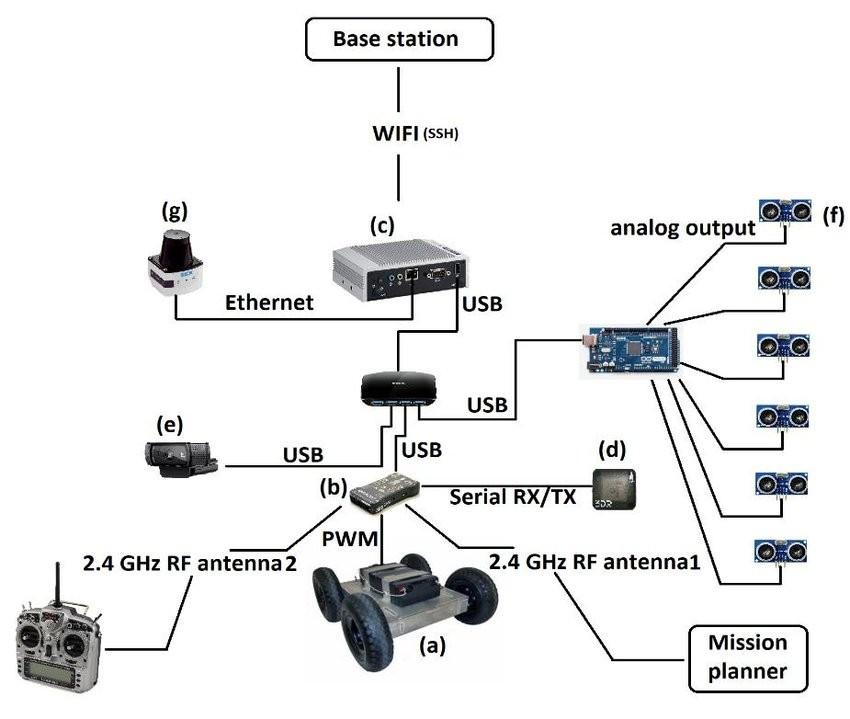


<https://www.semanticscholar.org/paper/Development-of-Robot-Scenario-Script-Language-and-Choi-Ha/56786fa39232a58db6823eb9248addb502d89b1b>

<https://www.researchgate.net/figure/Use-Case-Diagram-of-the-application-beWare-of-the-Robot-v20_fig2_316741832>

- robot sw & hw architecture





<https://www.sciencedirect.com/science/article/pii/S209580991630042X>

<https://www.researchgate.net/figure/Hardware-architecture-of-the-mobile-robot_fig1_325312866>

​