

4월 5일차 중간 발표

B팀

김준서
최지현
임승현
정종현

INDEX

1. P.E, Map,
Scale of Robot
2. Actor Controller
3. Back-end(API)
4. Front-end

1

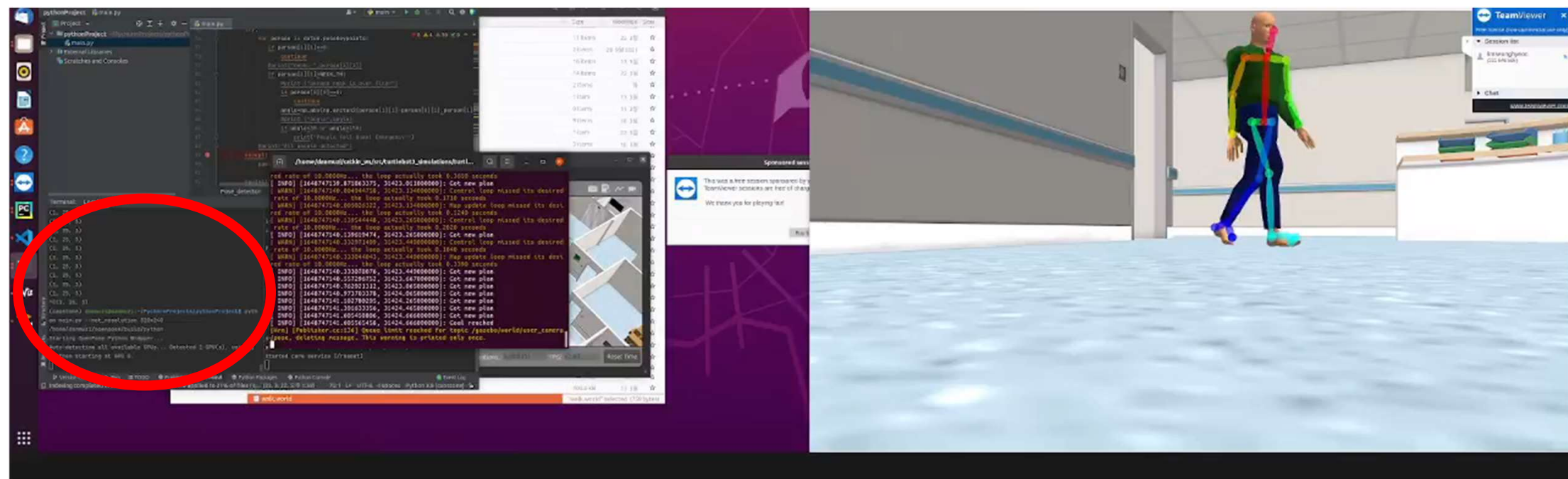
P.E, Map, Scale of Robot

1 Pose estimation, Map, Scale of Robot

- Openpose로 사람 넘어짐 감지

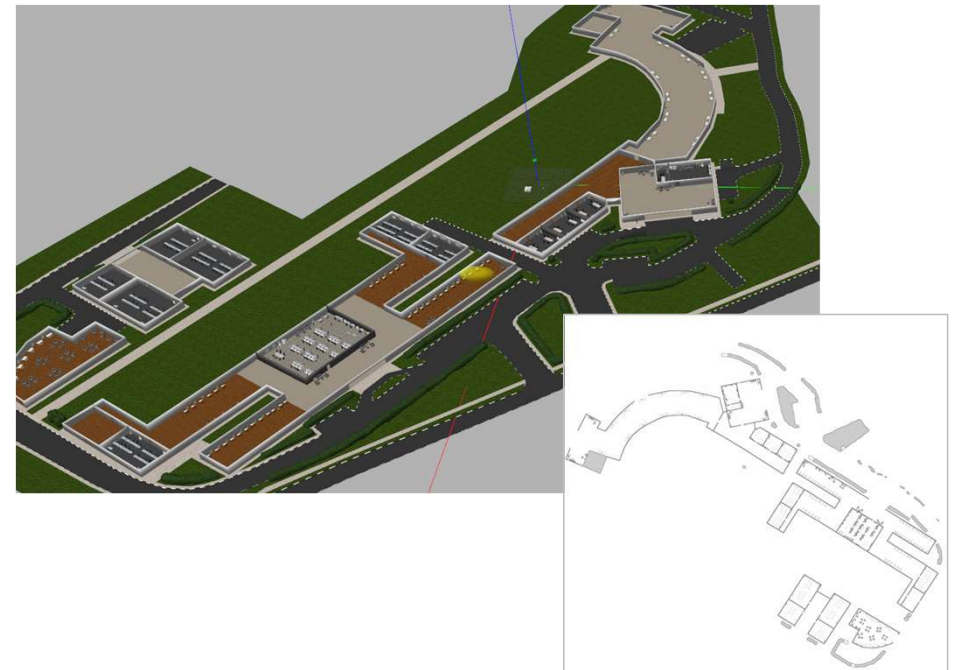
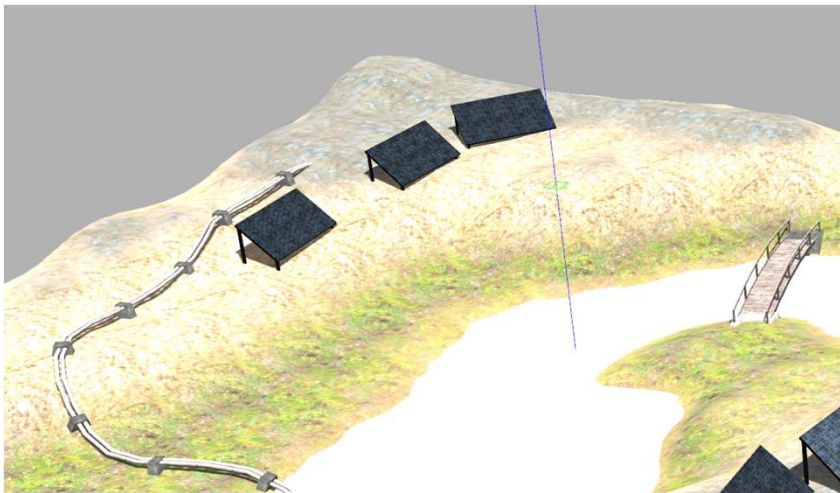
넘어짐 감지 조건:

$$y_{head} \leq floor_threshold \text{ and } abs(slope \text{ of lines connecting head and hip points}) \leq slope_threshold$$



1 Pose estimation, Map, Scale of Robot

● 실외 맵 교체



기존 맵의 문제점: 3d 구조로 되어있다.

시뮬레이션 맵이 2d로 되어 있다. -> 맵을 기반으로 주행을 하는데 Localization에 문제가 생긴다.
따라서 이러한 이슈를 해결하기 위해 맵을 2d 맵으로 바꾸었다.

1 Pose estimation, Map, Scale of Robot

● Issue: 로봇 스케일 조정



1. 문제 인식: 로봇이 맵과 사람에 비해서 상대적으로 작아서 카메라에 사람이 잘 안 담기는 문제가 발생
2. 로봇 사이즈를 시뮬레이터 내에서 키우는 것을 시도
3. Gazebo .world 파일을 수정해서 로봇을 구성하는 센서 바디, 타이어를 각각 수정
4. Challenge: 외형적인 것을 바꾸는 것은 쉬웠으나, 로봇의 질량, 관성 모멘트 같이 역학적인 측면까지 고려해야 되는 이유에 따라 예상보다 진척이 느려짐
5. 해결 방안: 결국에는 이동하면서 사람을 잘 찍는 것이 중요 -> 카메라만 위에다가 달 예정

6. Question: 따로 물리적인 지지대 없이 카메라만 공중에 띄워두는가에 대해서

1 Pose estimation, Map, Scale of Robot

● 앞으로 진행할 task

1. Emergency detection을 하면 로봇의 카메라로 사진을 찍어옴.

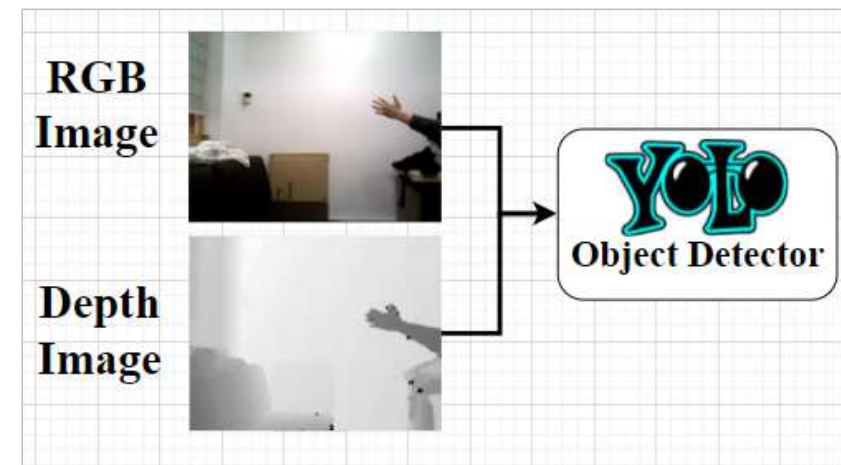
2. 로봇에 추가 카메라 부착

:전 방향으로 object detection 가능하도록 설정

3. 어르신을 따라가기 위한 object detection

: yolo의 object detection한 이미지와 depth image 정보를 합쳐 로봇이 어르신의 위치를 쫓아갈 수 있도록 함.

: 사람이 여러명 있으면 issue가 발생할 수 있음.



2

Actor Controller

Motivation

To implement the situation that a user take a walk with robot in simulation, we should have control of an actor.

Actor Controller

But, basic function doesn't serve the control of actor, it only serve scripts with some point and time.

We develop actor controller plugin on the gazebo.

Its movement is implemented by changing position of an actor.

2 Actor Controller

- Example



2 Actor Controller

● Difficulty

LiDAR can't recognize an actor.

By adding collision, LiDAR can recognize an actor, but actor can't move.



Depth Camera can recognize an actor

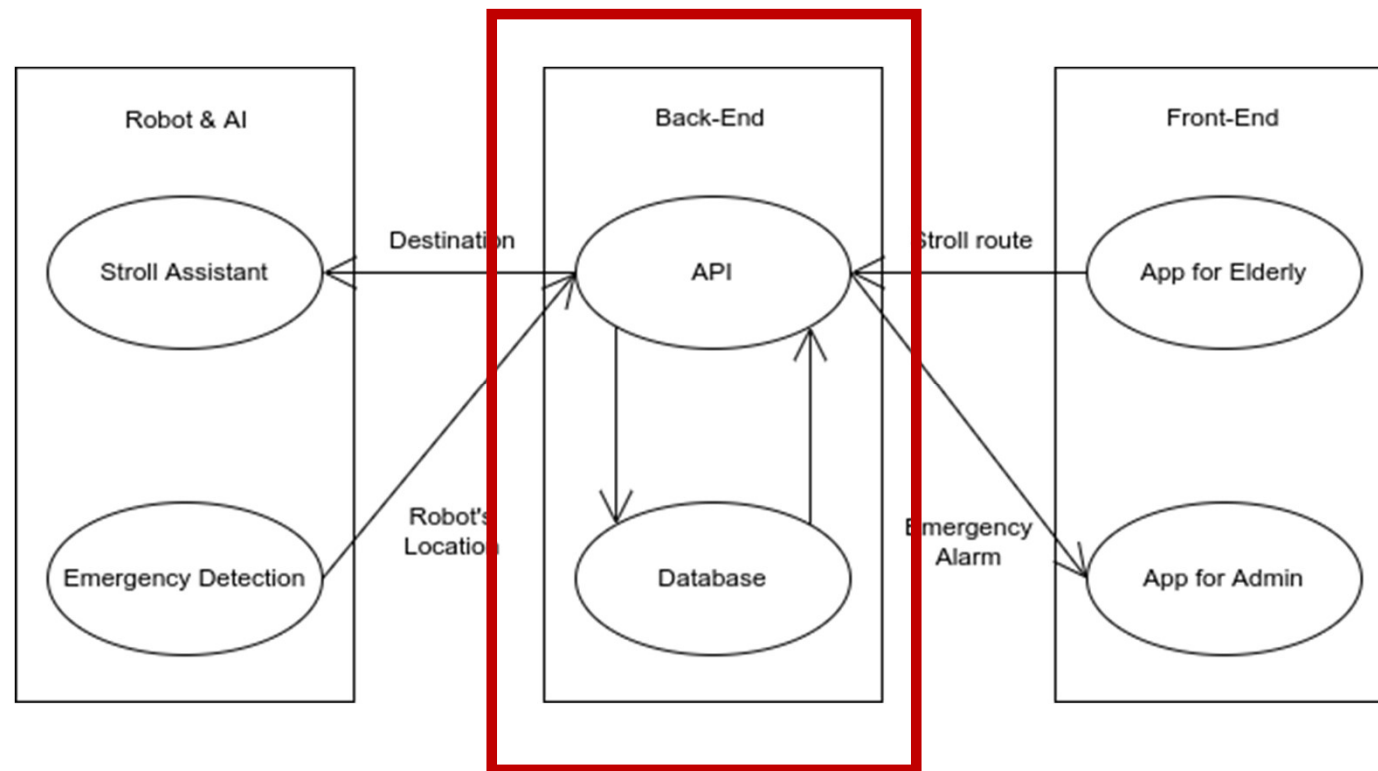
By using depth camera, we will keep a distance from the user.

3

Back-end(API)

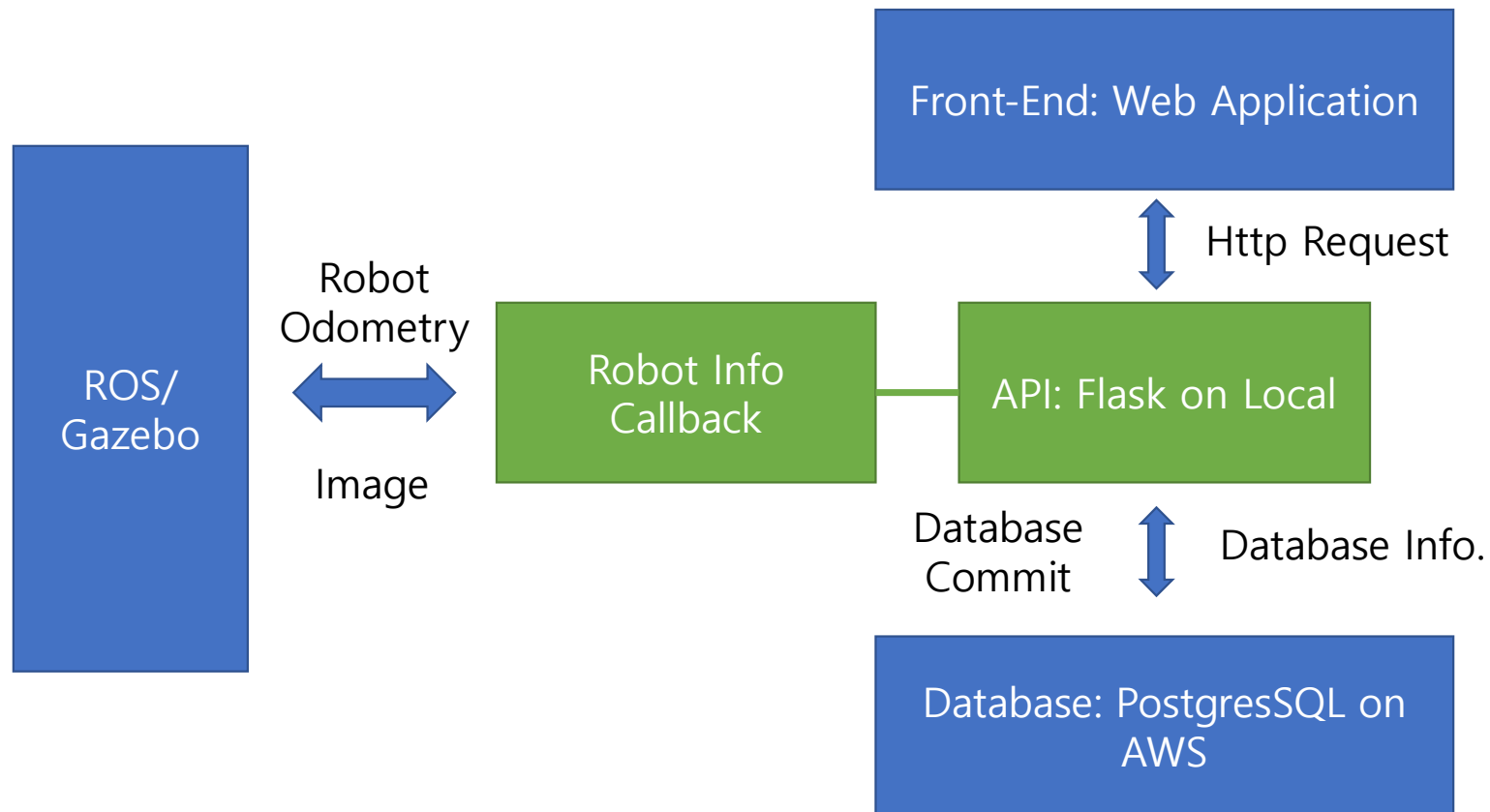
3 Back-end(API)

Progress



3 Back-end(API)

● Progress

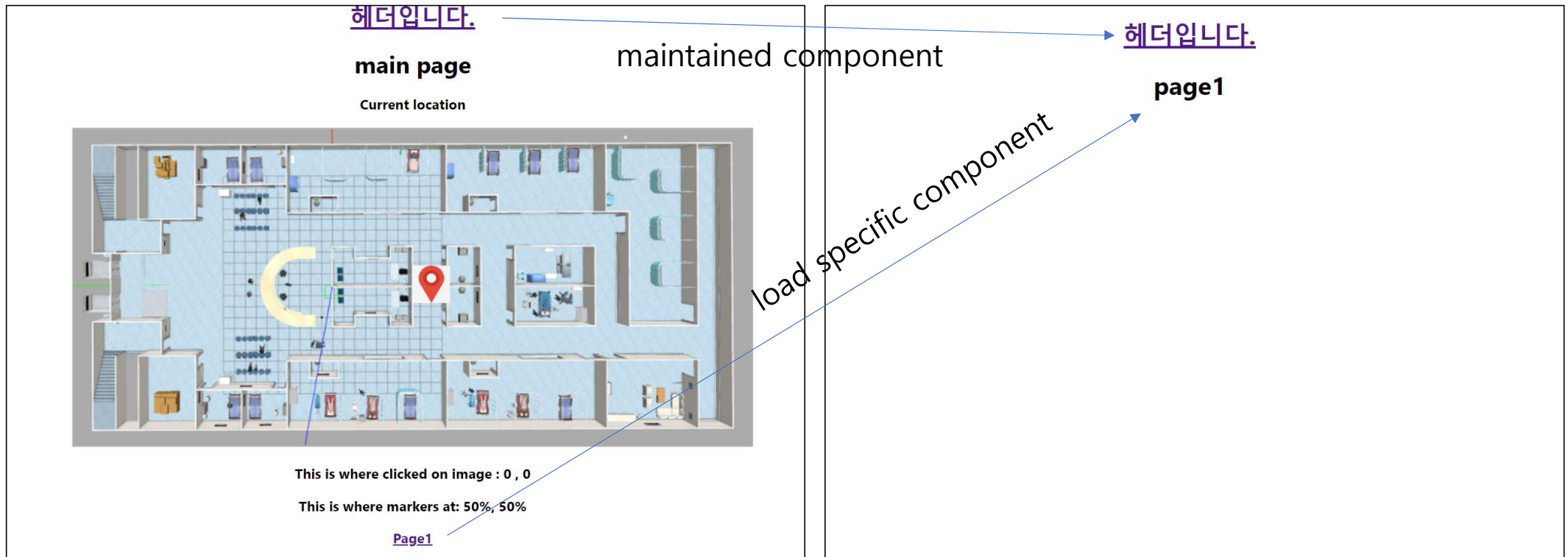


4

Front-end

4 Front-end

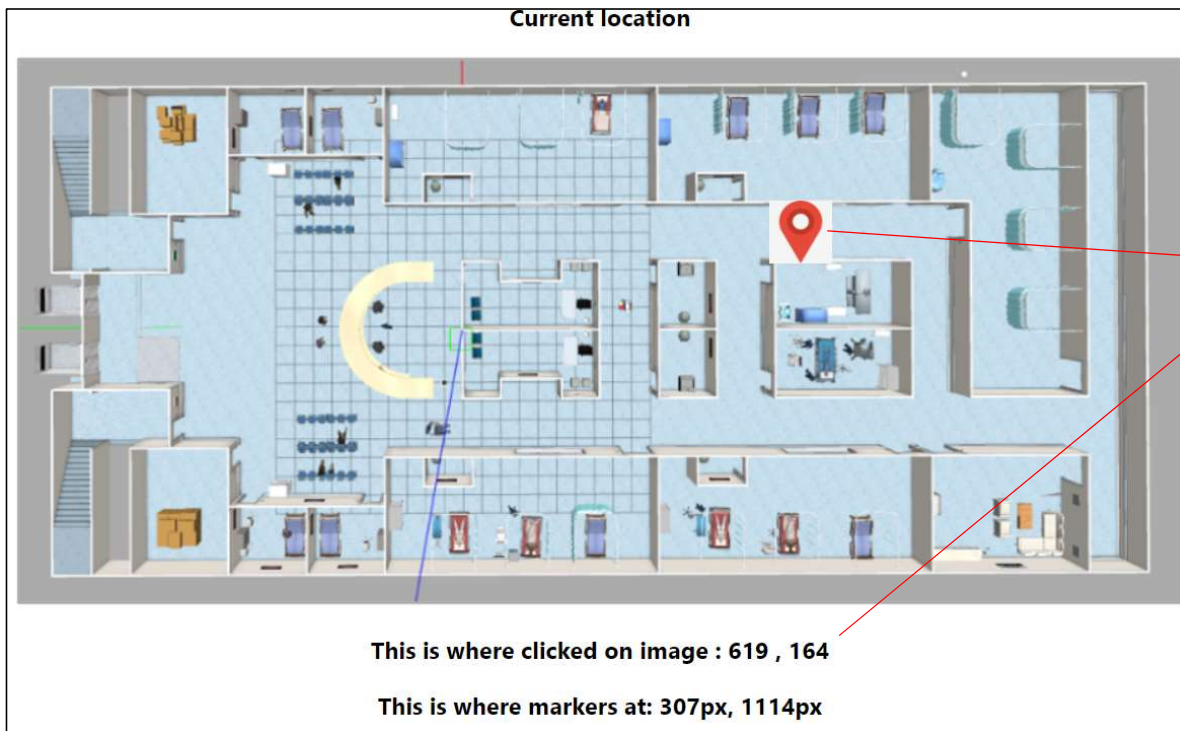
Page routing



further task: apply page layout and routing

4 Front-end

● Location marker



when clicked

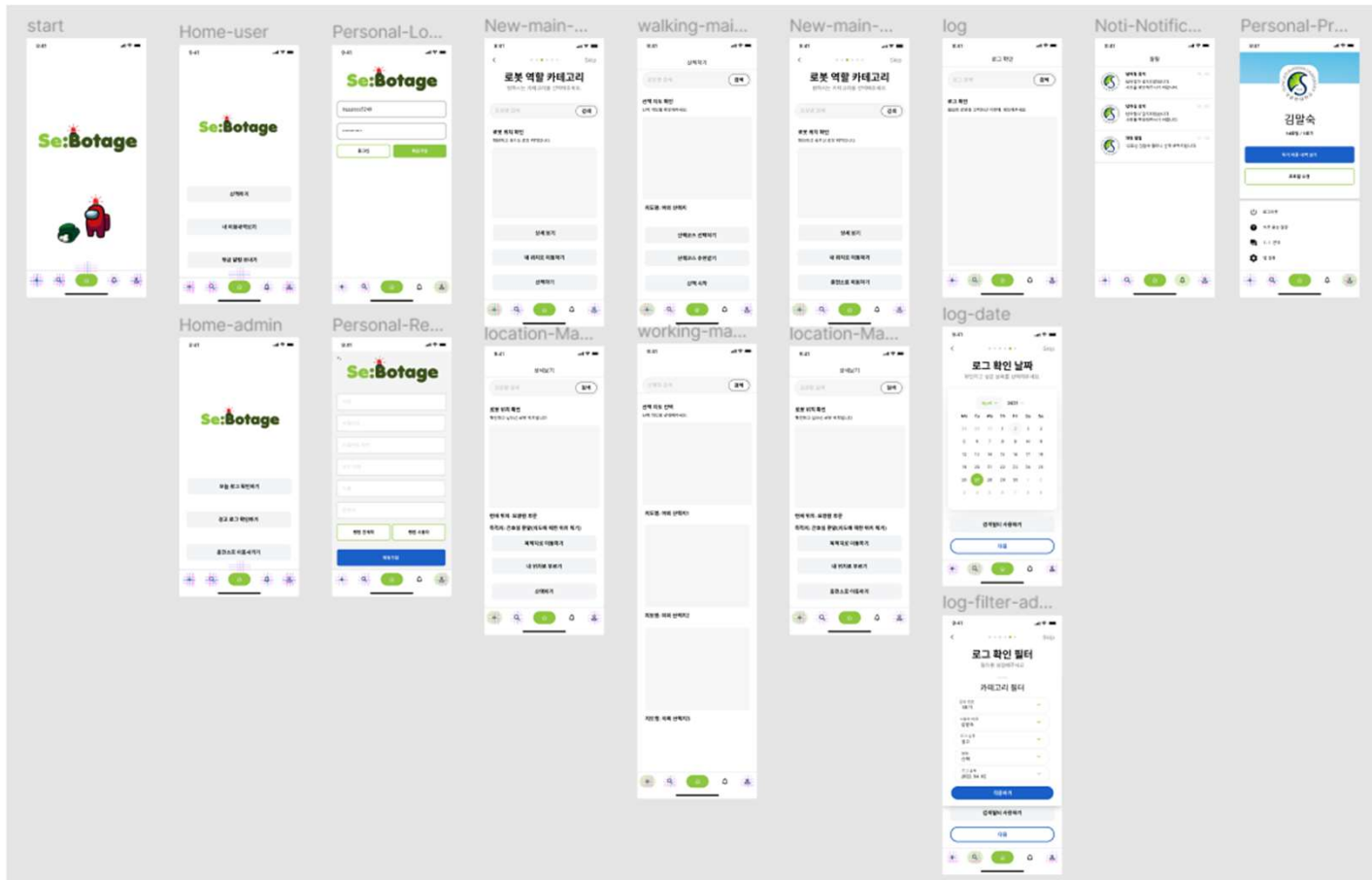
locate marker
get the pixel information

further task

change pixel information
to location information in ROS for robot

4 Front-end

App-Design



요양원 앱 디자인

먼저 Figma를 사용해서 앱 디자인을 구성

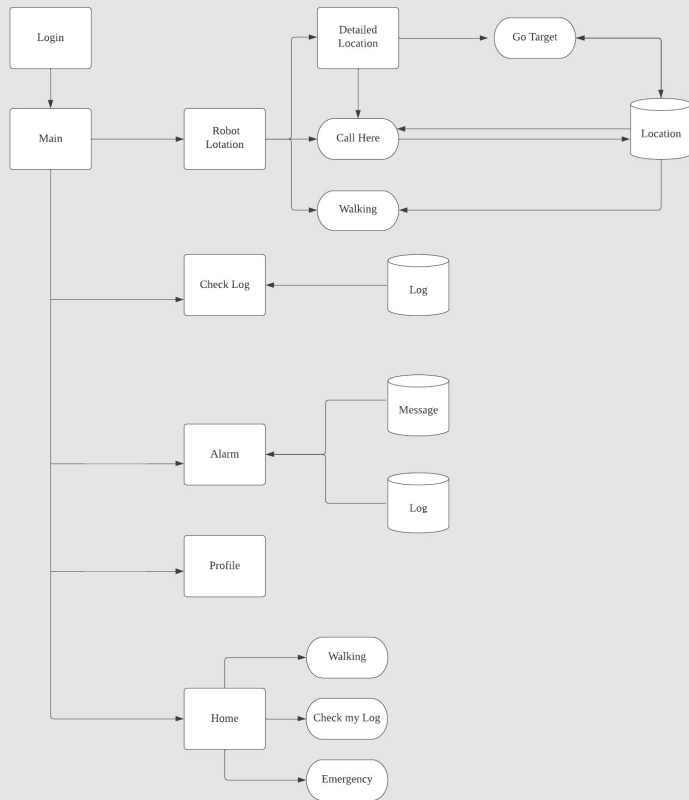
admin 모드, user 모드

두 모드에 다르게 각 상황에 맞는 기능과, 디자인을 기획해서 제작함.

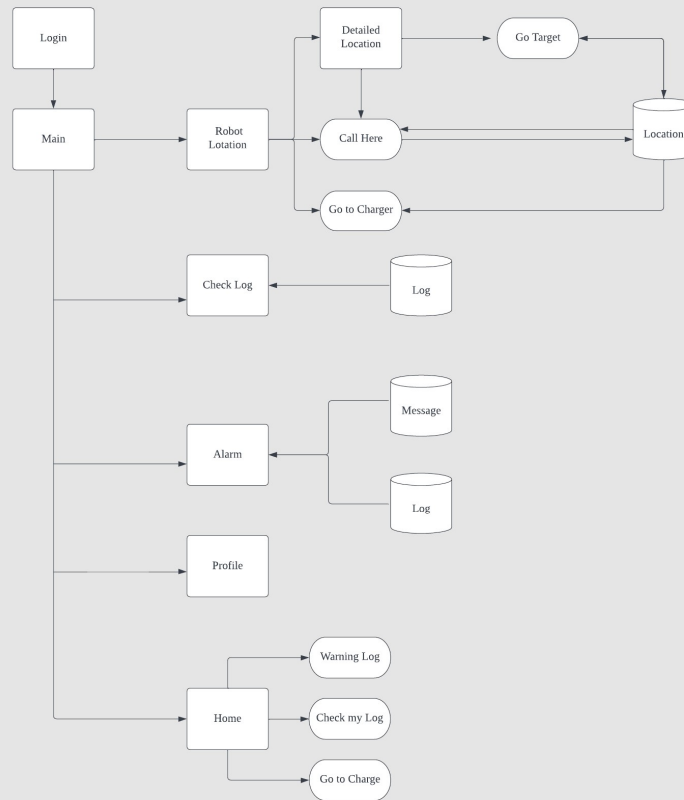
4 Front-end

Flow Chart

요양원 앱 플로우차트 - 사용자



요양원 앱 플로우차트 - 관리자



Flow Chart

요양원 맵 DB를 통해서 위치와 산책로에 대한 값을 상호적으로 주고 받으며 산책을 가능하게 만들 예정

사용자 앱과 관리자 앱의 같은 DB를 사용하되, 접근 범위 등을 조절하여 앱 내 기능을 변경할 예정

세부적인 Flow chart 결과는 추후에 수정 예정

5 Goal

Progress

	Week6	Week7	Week8	Week9	Week 10 presentation	Week11	Week12	Week13	Week14	Week 15 final presentation
Robotics	Setting on Gazebo	Add user model Edit robot model	Edit robot model	Set Navigation Stack parameter			Check navigation function	Test and trouble shooting		
DL		Train model	Train model 2							
Frontend		Build app for user		Connect with robot		Emergency alert message	Build app for admin			
Backend		Design Database	Login feature	Connect robot with app Set API for emergency		Set API for command log				
Document	Proposal document							Final document		