



Posture Fixer

Jia Lee,
Cheng Peter Qian,
Lécuyer Cédric



Plan

1. Introduction
2. Architecture
3. Sensor Placement and Posture Detection
4. Hardware Setup
5. Development Environment
6. Communication
7. Functionality of Application
8. Scenario
9. Improvements

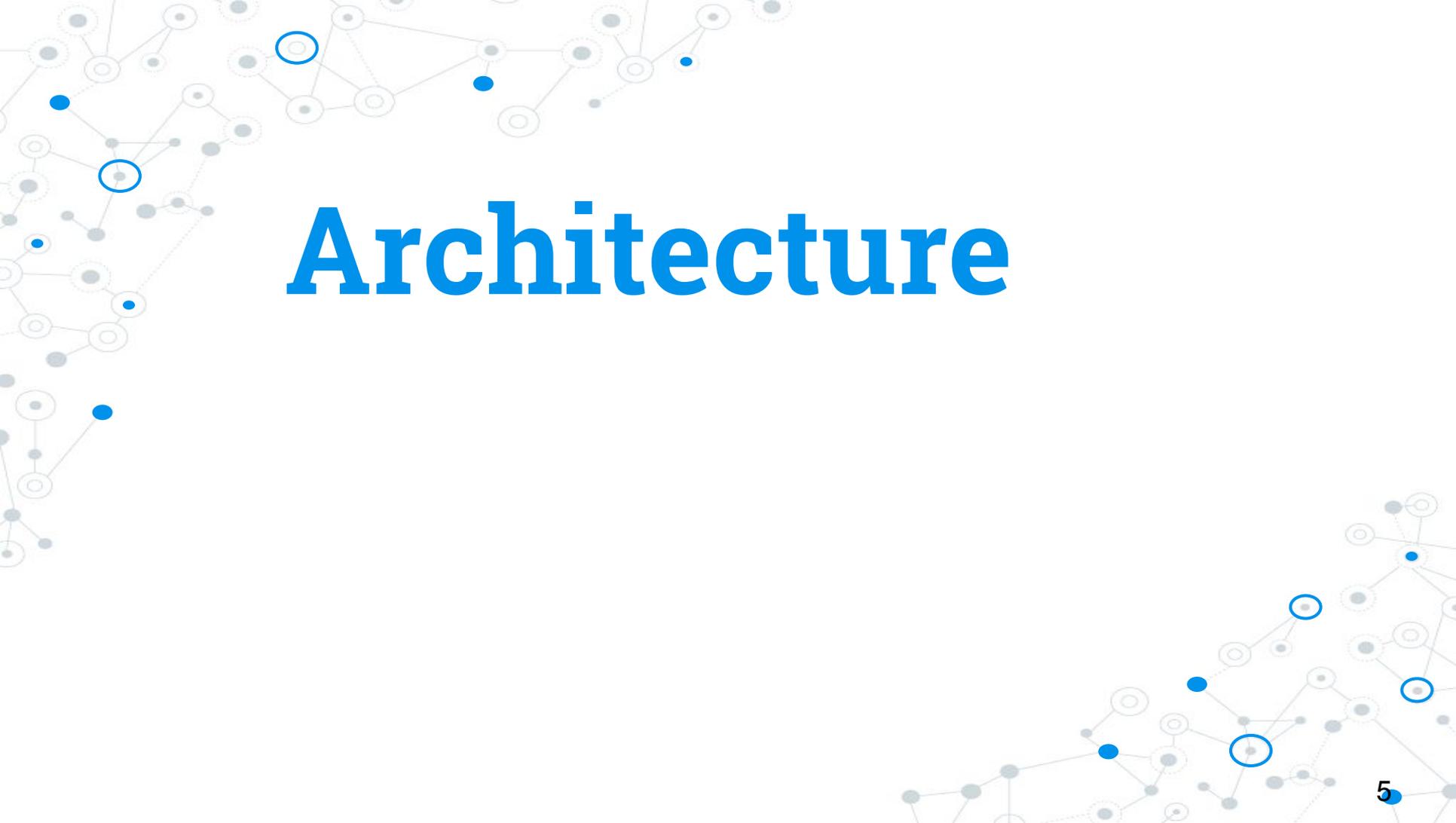


Introduction

Introduction

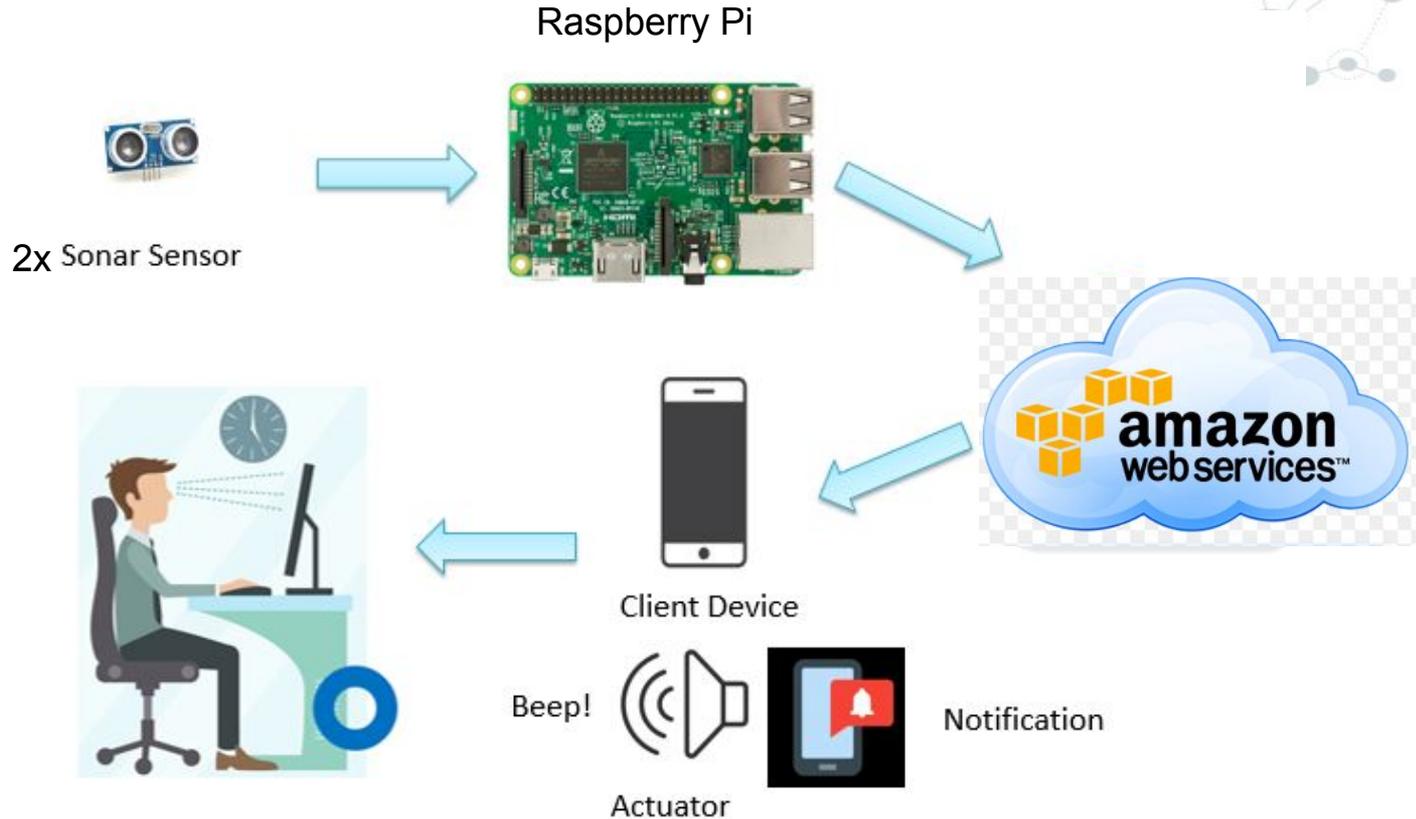


- ◎ Many people sit in a bad posture for a long time.
- ◎ It makes people's spine tense much.
- ◎ So, we will make "Posture Fixer", that can detect if the user is sitting in a bad posture, and if so, send an alert to his smartphone.



Architecture

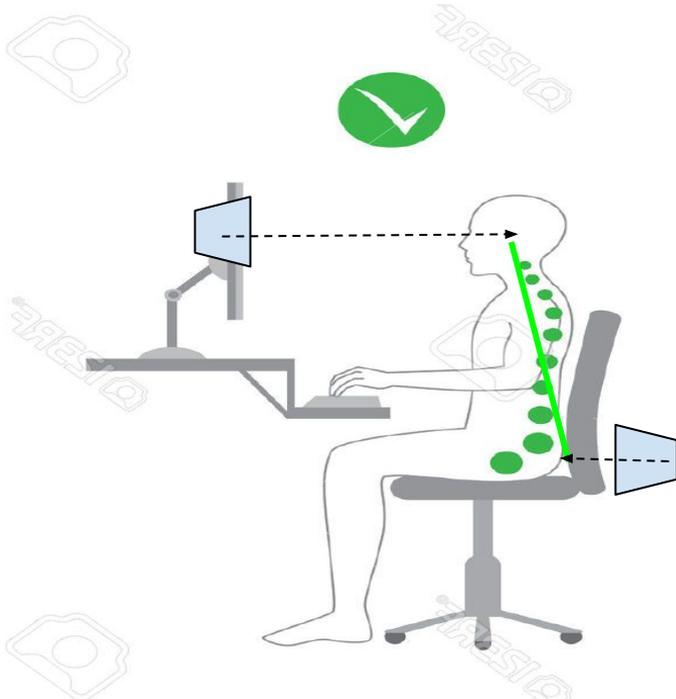
Architecture





Sensor Placement and Posture Detection

Sensor Placement Diagram

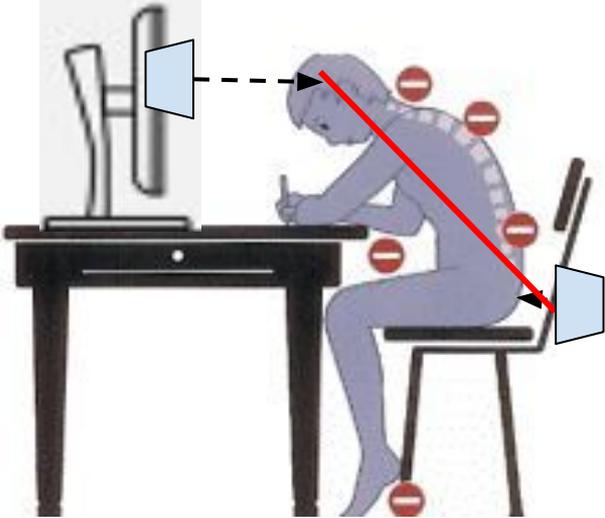


 Sensor

 Detection range

 Good inclination

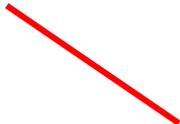
Sensor Placement Diagram



Sensor



Detection range



Bad inclination

Three Posture States

Good posture:



Three Posture States

Bad Posture 1:



Three Posture States

Bad Posture 2:

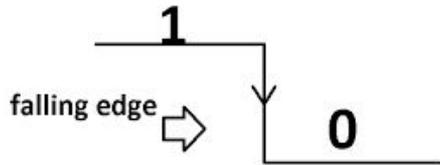
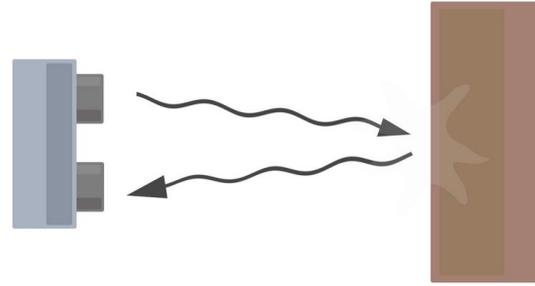




Hardware Setup

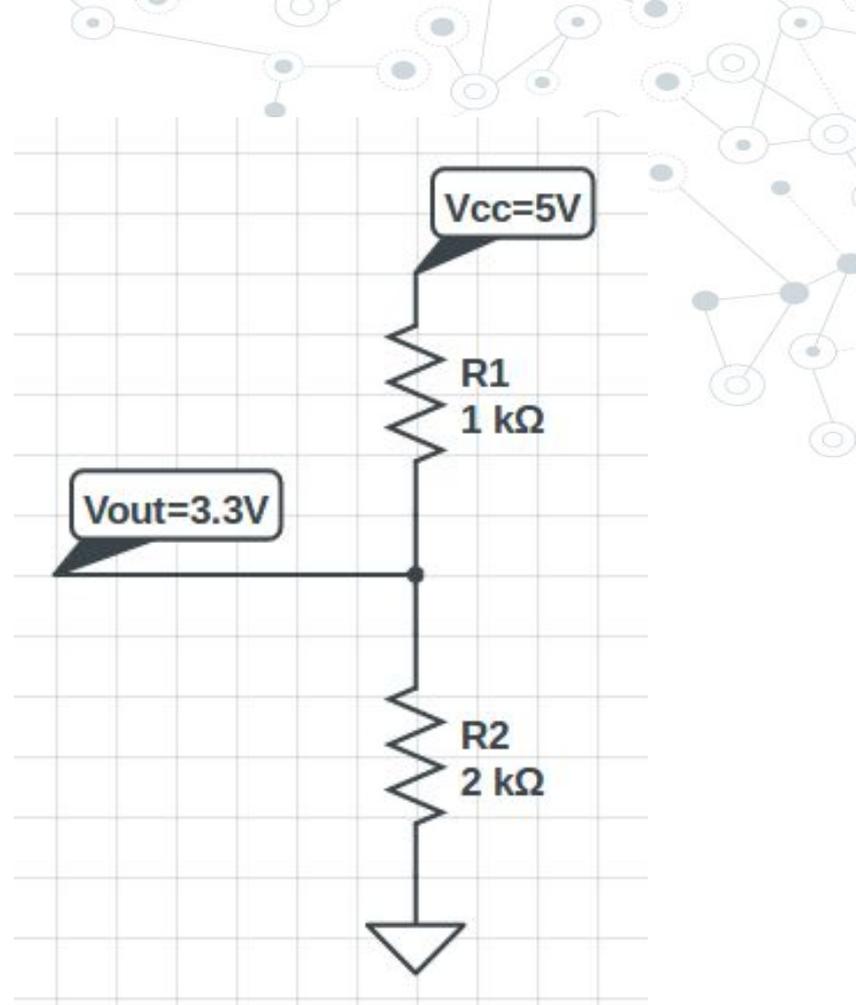
Sonar Sensor

- ⦿ Vcc: 5V (power rail)
- ⦿ GND: Ground
- ⦿ Trigger: input that tells sonar sensor to send a wave signal on a falling edge
- ⦿ Echo: output that goes high when sensor receives sent wave



Voltage Step Down

- ◎ Sonar sensor requires 5V to power
- ◎ But RPi input pins can only handle 3V3
- ◎ Use a simple voltage divider



Calculating Distance

- ◎ Sonar sensors measure times
- ◎ Time taken is twice the distance to object
- ◎ Convert the time to a distance
- ◎ Speed of sound in air is about 34000 cm/s

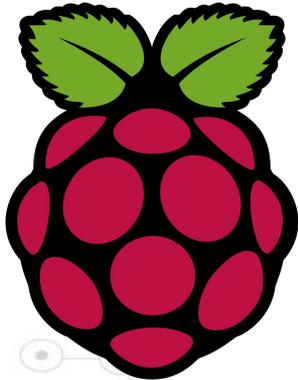
$$\text{distance} = \frac{\text{speed of sound} \times \text{time taken}}{2}$$



Development Environment

Raspberry Pi

- ◎ Python 2.7
- ◎ AWS IoT Python SDK => Link between Cloud and Raspberry Pi



Cloud

- ◎ Lambda function
 - Insert data from raspberry messages to database
- ◎ DynamoDB
 - Collect every sensor data



DynamoDB

Application - Development Environment

- ◎ We use Andorid Studio 3.0 for developing application.
- ◎ Samsung Galaxy A5 2015
- ◎ API 19(Kit Kat)



A background network diagram consisting of interconnected nodes and lines. The nodes are represented by circles of varying sizes and colors, including light gray, dark gray, and blue. Some nodes are highlighted with a blue outline. The lines connecting the nodes are thin and light gray, forming a complex web-like structure. The overall aesthetic is clean and modern, typical of a professional presentation.

Communication

Communication - Sensors -> Cloud

- ① Communication through MQTT (publish/subscribe model)
- ① Send every data sensor to Cloud



distanceSensors/distance

Dec 15, 2017 8:02:40 PM +0900

```
{
  "Top": "34.023",
  "Bottom": "9.457",
  "userID": "1",
  "dateTime": "20171215200239"
}
```

Communication - Smartphone App <-> Cloud

- ① Use AWS Mobile Android SDK
- ① App access directly to DynamoDB through request :
 - One to get the most recent data inserted
 - One to get every data in a time interval



Functionality of Application

Functionality of App

- ⦿ Execute Posture Detection program.
- ⦿ User can always see the status of the program.
- ⦿ Notify to user if user's posture is bad during running the program.
- ⦿ User can see his posture through graph when the detection program is stopped.
- ⦿ We suggest Spine exercise.

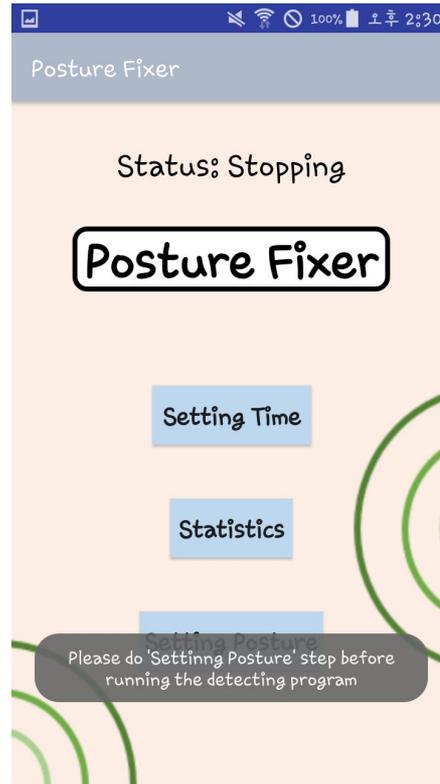


Scenario

Scenario (1)

- ◎ User have to do “Setting Posture” step at first.
So, if user didn't do that, user can't use our application.
- ◎ Scenario 1 : User press the button “Setting Time” or “Statistics” at first.
=> It shows the message “Please do ‘Setting Posture’ step”.

Scenario (1)



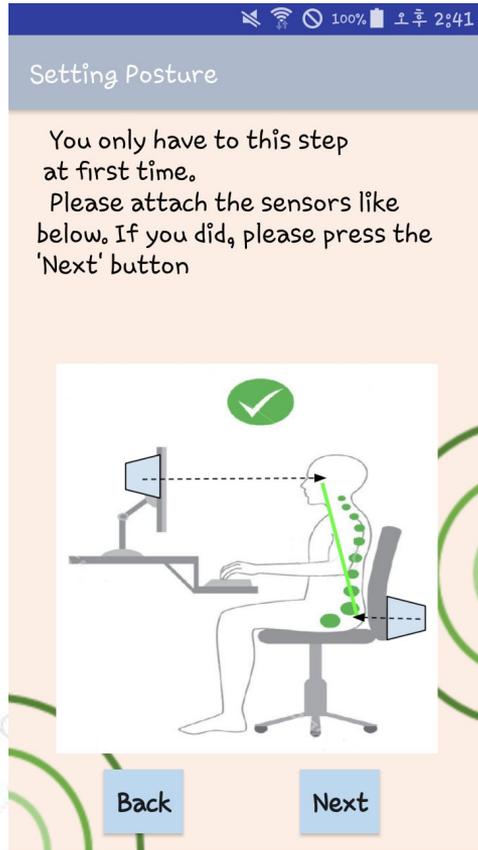
Scenario (2)

- ① User have to attach two sonar sensors, so we provide the guideline how to attach them.
- ① User also be requested to measure some distances that is needed to decide the posture.

Scenario (2)

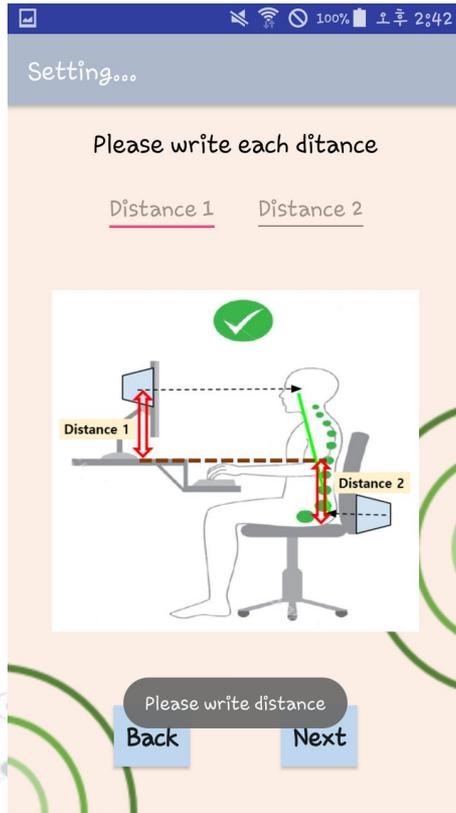
- Scenario 2 : User press the button “Setting Posture”.
=> User can see the image of how to attach the sensors. If user attach it, user press “Next” button.
=> Next page request to measure some distances, it shows the image of guideline.

Scenario (2)



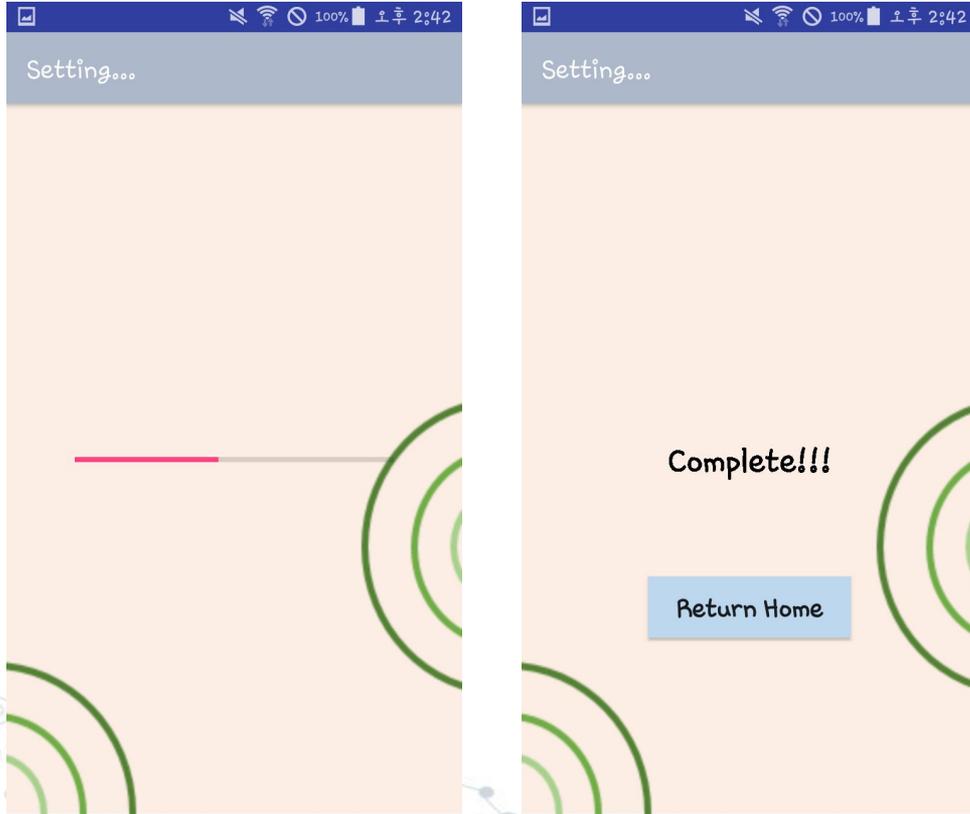
- ⇒ Next page request to measure some distances, it shows the image of guideline.

Scenario (2)



- ⇒ If user click “Next” button with blank distance, it shows error message and request the distances again.
- ⇒ If not, user can see the loading page.

Scenario (2)



- ⇒ After finish the loading, user can use our detection program.

Scenario (3)

- ⦿ After finish the setting posture part, user can execute our detection program.
- ⦿ User press “Setting Time” button.
 - => Press “Specific Time” button.
 - => Press “Next” button after setting the start time.
 - => User setting the finish time.
 - => If user set the finish time earlier, it shows error and request again.

Scenario (3)

Setting Start Time

Setting Start Time

오전	1	54
오후	2	55
	3	56

Back Next



Setting Finish Time

Setting Finish Time

오전	1	52
오후	2	53
	3	54

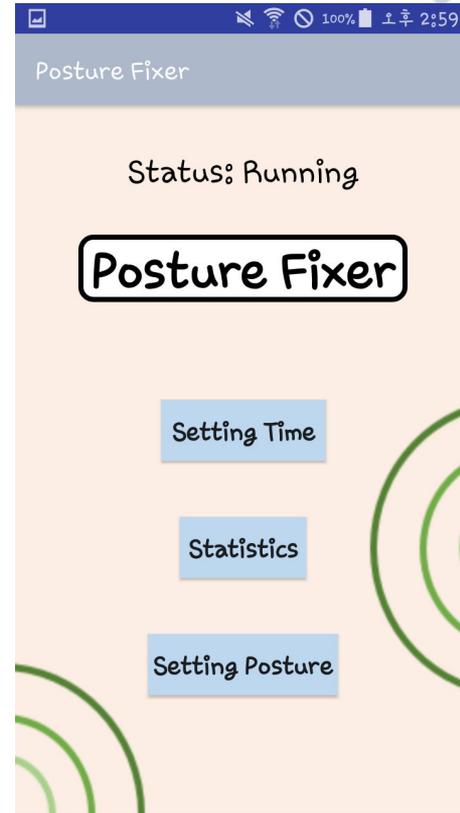
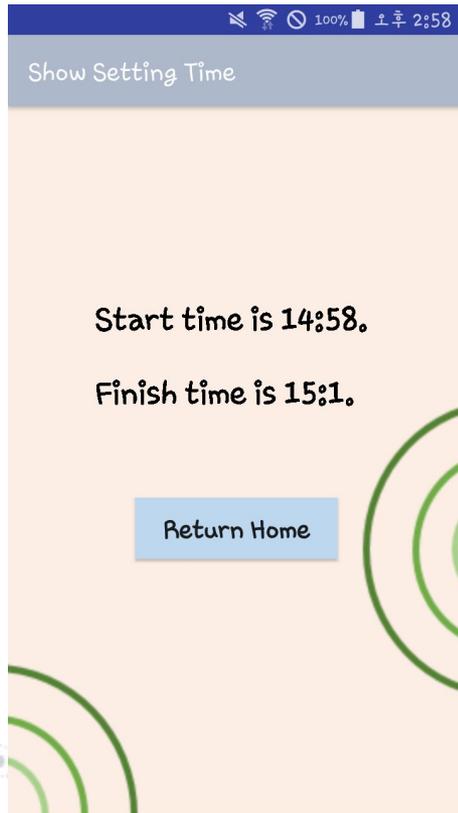
Back Finish

Finish time is less than Start time

Scenario (3)

- ⇒ User can press “Next”, if user set the finish time correctly.
- ⇒ It shows the setting time result.
- ⇒ The “Status” is changed from “Stopping” to “Running”

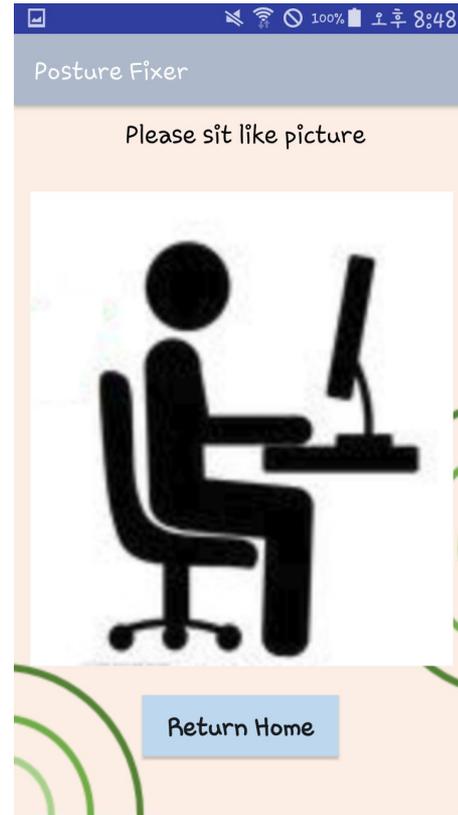
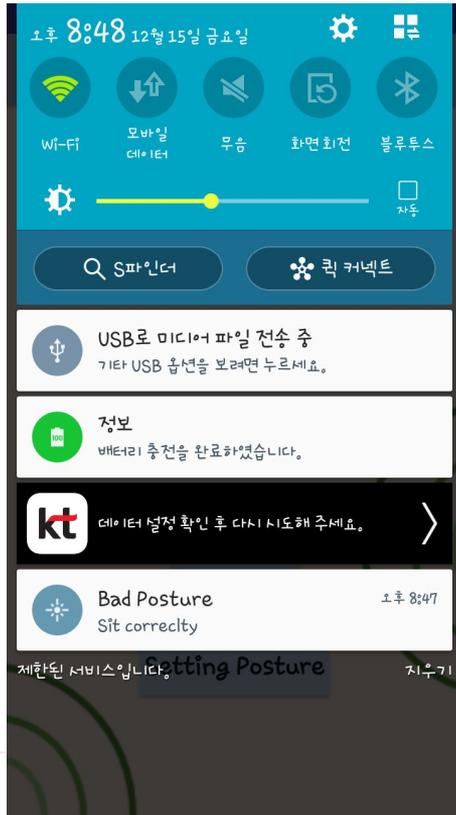
Scenario (3)



Scenario (4)

- ① User receives notification(sound/notification) when his/her posture is bad during the program.
- ① If user click the pop-up, user can see the image of good posture.

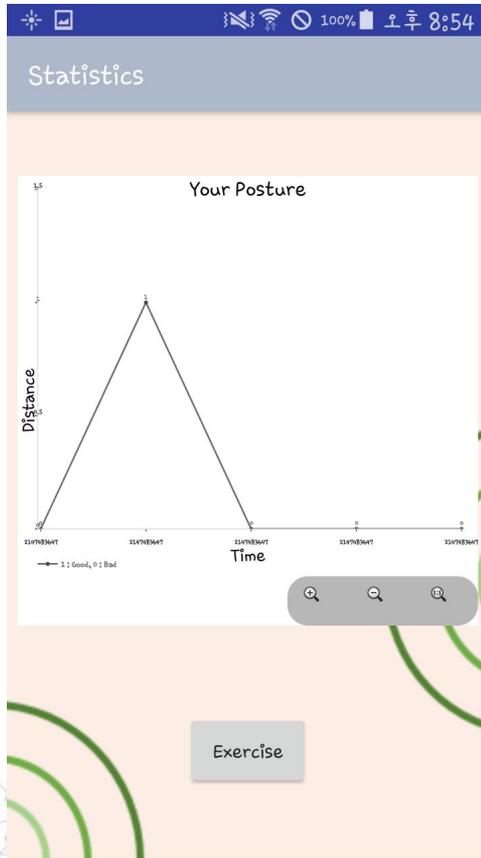
Scenario (4)



Scenario (5)

- ⦿ User can see the result of the detection program through graph after finish time.
- ⦿ User can see the graph
 - => User press “Exercise” Button at the page
 - => User can see image of spine exercise
 - => Status change to “Stopping”

Scenario (5)



The screenshot shows a mobile application interface for a 'Posture Fixer'. At the top, there is a status bar with icons for signal strength, Wi-Fi, and battery, along with the time 3:01. Below the status bar is a header labeled 'Posture Fixer'. The main content area features a grid of exercises under the heading 'SPINAL FLEXIBILITY & STRETCHING EXERCISES'. The grid is organized into columns for different positions: LYING, PRONE, HANDS & KNEES, and NECK FLEX. Each exercise includes a small illustration of a person performing the exercise and a brief description of the action and the muscles it targets. At the bottom of the screen, there is a button labeled 'Return Home'.

The screenshot shows a mobile application interface for a 'Posture Fixer'. At the top, there is a status bar with icons for signal strength, Wi-Fi, and battery, along with the time 3:01. Below the status bar is a header labeled 'Posture Fixer'. The main content area features a large text message that says 'Status: Stopping' and a large button labeled 'Posture Fixer'. Below the button, there are three smaller buttons labeled 'Setting Time', 'Statistics', and 'Setting Posture'. At the bottom of the screen, there is a button labeled 'Return Home'.



Improvements

Sensors

- ◎ Sonar sensor, they rely a lot to the surface :
 - Sensors should be parallel to surface
 - Surface should be smooth
- ◎ User can't touch the back of the chair
- ◎ User can't move his computer/chair
- ◎ Use more precise sensors
 - voltage of current output instead of binary output
 - shorter range sensors
- ◎ Use smaller sensors
 - more user friendly

Hardware

- ◎ Create a more compact and portable prototype
 - uses too many wires
 - long setup time
- ◎ Inclination calculation implies too much user's constraint

Application

- ① We proposed two options for setting time, specific time and just press start or end button. For now, we implemented only specific time option.
- ① More efficient deciding posture algorithm should be developed.

Communication

- ◎ Handle communication between smartphone and raspberry directly (Bluetooth)

Cloud

- ◎ Handle user account, using Cognito

Appendices

- ◎ Proposal presentation:

https://docs.google.com/presentation/d/1mPka3o9Z_TKMSviaAPKNeDv78ohkpxW6ljdrkyaTXtU/edit?ts=59e6e68d#slide=id.g270fc8a2a8_0_42

- ◎ Requirement presentation:

https://docs.google.com/presentation/d/1do6esaTsl_NTU8wtmG-OQR-oYSvhouKGIJo0PmHZSfA/edit?usp=sharing

- ◎ Design presentation:

https://docs.google.com/presentation/d/1QEi9DpJgX1ahcsWslhRTsRzQiPK9iy36jz_aqkT6jZl/edit?usp=sharing

Code: <https://github.com/cedriclec/postureFixer>