Mid Presentation:
Health Buzz
A health reminder

By Team 2:
Yang Hyeonseo
Song UGyeong
Jang Donghae
Jasmine Abtahi

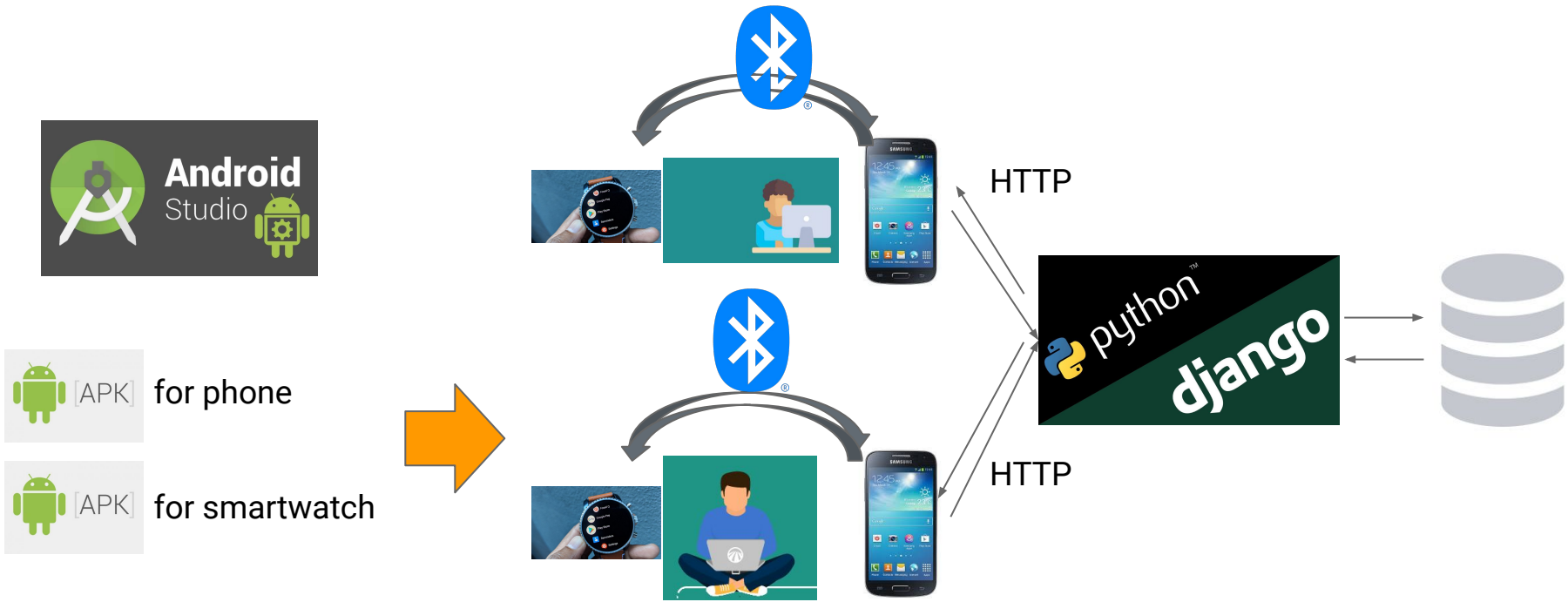
Refined project idea and scope

- People have hard time studying and working on their seat. While doing burdensome work, they sometimes forget to have refreshing time. For these people's health, stretching and drinking water is important.
- **Target user**
  - People who sit in front of their desk for long time need reminder to stretch (customized interval)
  - People who forget to drink enough water for health(8 cups)
- **Key approach**
  - People need extra effort to remind these actions in daily life.
  - By using sensor data, we build model to detect cup grabbing and inactive state and remind user automatically
- **Changes in the project scope**
  - We get rid of body pose recognition part
  - We will provide different service by checking user's smartwatch availability

Use cases & demonstration of one or two key functions

Live demo.

Health Buzz’s overall architecture



The system architecture overview

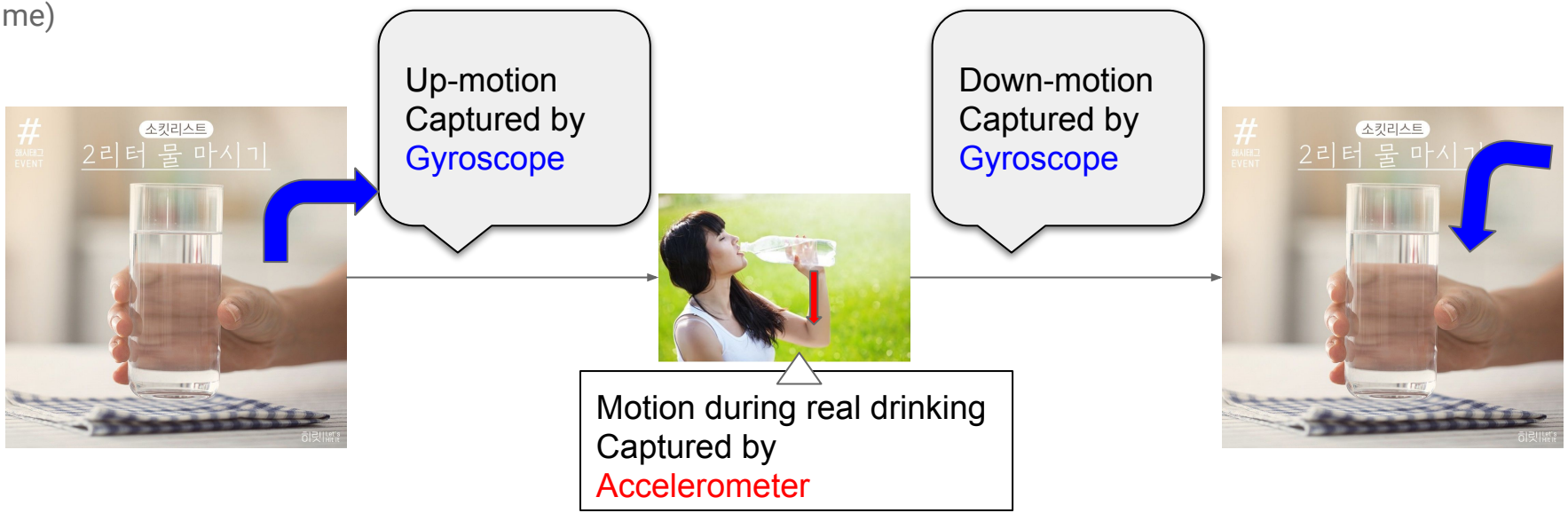
1. There are some Android components: WelcomeActivity, DashboardActivity, StretchingDetailActivity, WaterDetailActivity, DebugActivity, SettingsActivity, InferenceActivity, SensorService, and BootReceiver.
2. Sensor monitoring and detection logic are called from the foreground SensorService. SensorService updates the detection result and notify UI via MutableLiveDatas which resides in a singleton object named RealtimeModel.
    a. Service loads the AI model to detect movement states from the sensor data collected, and infer from the model.
    b. Then it shows notification.
3. Activities show and encourage the user to be in health, and starts our SensorService.
4. BootReceiver starts our SensorService automatically on boot.

Key technical challenges & solutions

1. The app has to monitor the sensor even when the user is not using the app foreground.
    a. Solved by creating a service
2. The service needs to be started automatically when the phone booting is finished.
    a. Solved by creating a BroadcastReceiver
3. The UI needs to be updated in real time based on settings and the service's detection result.
    a. Solved by using LiveData and Observable
4. We have to infer whether the user is moving or sitting
    a. Solved by using Weka library
5. The smartwatch and the phone should communicate with each other.
    a. The app will listen as a server and the smartwatch will send its data via bluetooth. (RFCOMM)
6. Latency, Power, Accuracy
    a. Latency: Fine enough
    b. Power consumption: Low for now (two stage not needed for accelerometer yet)
    c. Accuracy: 98% for moving motion.

Our first try for water drinking motion

We need two models: First one is Gyroscope(**rotation**) (Classify **up-motion**, **down-motion** and **other**).
Second one is Accelerometer(**acceleration->force**)(Classifying **real drinking** and **not**).
We will consider these three series motions as water drinking. (each motion should be lasting for some time)



Project schedule - current plan

- **What were done**
  - Build skeleton app for frontend
  - Gather sensor data from smartphone and watch
  - Develop heuristic/ML model for detecting user status(moving, drinking water)
  - Implement app's background service part for giving notification

- **What are to be done**
  - Implement bluetooth communication between smartphone and watch
  - Improve model accuracy on detecting water drinking by using DL model and feature engineering
  - Develop proper backend
  - Add community feature like ranking or competition
  - Improve UI/UX
  - Deploy server
  - Testing

Project schedule – Things that are done

Feature	Who did
0. Build a skeleton	Everyone
1. Get friendly with smartwatch and phone sensors	Everyone
2. Add additional app component(service, notification, broadcast)	Hyeonseo
3. Study ML/DL model	Everyone (Donghae lead)
4. Collect data from sensors and Train model and deployment	Donghae & UGyeong
5. Improve UI	Jasmine
6. Detail page design	UGyeong

Project schedule – Things to do

Feature	Assignee
7. Bluetooth communication	Jasmine & Hyeonseo
8. Backend for user account and data history	UGyeong
9. Improve model accuracy by DL model design and feature engineering	Donghae & UGyeong
10. Improve UI & UX	Jasmine & Donghae
11. Deploy server	Hyeonseo
12. Community feature	Hyeonseo
13. Testing	Everyone

Final deliverable and success criteria

o Our Final App will be like

Using sensors of smart phone and watch, without setting fixed time people can get alarmed for their stretching and water drinking at appropriate time.

People can also record and share their health history about stretching and water drinking.

o Our Success Criteria are

If our model for moving gets upper 99% acc, one for water drinking gets upper 95% acc(19 of 20) and other functions(data recording and competing with others) are implemented, we will judge our project succeeds.

Potential threats to the project and discussion

- Accuracy for water drinking detection model could be low -> need to deal with wrong prediction case
- Inference time of DL model -> optimize model or use other light model
- Energy consumption of DL model -> two stage method for sensors
- Load on backend -> Currently not many features use backend request, deploying server
- No business model in current project plan -> Recommend stretching Youtube content

Thank you!