Group Members

Michelle Lam, Michelle Tan, Isabel Ketner, Jacqueline Lam

Overview

Project Title WAP (Wellness and Productivity)

Abstract

Having to stay at home everyday has increased the importance of self-awareness about one's mental and physical health. In response to that, we have decided to create WAP, which is a device that aims to improve an individual's overall wellbeing through personalized wellness tracks and guided exercises. With the incorporation of speech processing, computer vision, and gesture recognition, WAP provides a much more seamless experience for users to attain their goals.

High-level project description

Objectives

1. To guide people through steps they can take to improve or maintain their work from home wellness.
2. Increase interactivity between friends, co-workers, family, etc. in a simple and joyful manner.
3. Incorporate personalized wellness plans to fit the needs of each user.

Target features (MVP)

1. (Addresses obj. 1 and 3) **Customizable checklist**: User being able to check off a list of guided exercises that they will be reminded of and need to complete. Alternatively could just pick a premade wellness plan.
    a. Add description of which tasks would help improve mental, physical, productivity, etc.
2. (Addresses obj. 1) **Guided exercises:** We would need to create a feature that reminds the individual when it is time to do their task and can also help verify if they are doing them correctly

    First Quarter Goals

    a. Taking deep breaths (i.e. meditation)
        i. Visualization (circle that expands when inhaling and circle that contracts when exhaling)

   b. Stretching poses- pose determination
      i. Include animation of different poses
   c. Drinking water - IMU
      i. Can use tinyML to detect certain gestures

   Possible Second Quarter Goals

   a. Cognitive exercises to stimulate the BRAIN
   b. Posture correction
   c. PRODUCTIVITY RELATED STUFF
      i. Detecting how focused someone is based on eye tracking
      ii. Detecting tiredness through eye blinking rate and reminder to not look at screen for too long (screen wellness)
   d. Taking deep breaths (i.e. meditation)
      i. Webcam
      ii. Could incorporate sound using mic with speech processing

3. (Addresses obj. 2) **Words of encouragement:** An inspiring quote or supportive words from another user or even from oneself can be displayed daily every time they open up the app or wherever we have our project
4. (Addresses obj. 1 and 3) **Simon says:** Using voice commands to retrieve health stats or setting reminders

<u>Stretch goal features</u>

1. (Addresses obj. 1 and 3) **Feels:** User can input how they are feeling each day and we can suggest options to make them feel better
2. (Addresses obj. 1 and 3) **Achievements:** User can be rewarded badges for completing tasks to incentivise using the app
3. (Addresses obj. 2) **Music messages**: Send another friend a small animation to go with song so they feel less alone (An addition to “words of encouragement”)

<u>Use cases or scenarios</u>

1. High school/college students
2. Working professionals
3. Children (elementary/middle school)

   Possibilities

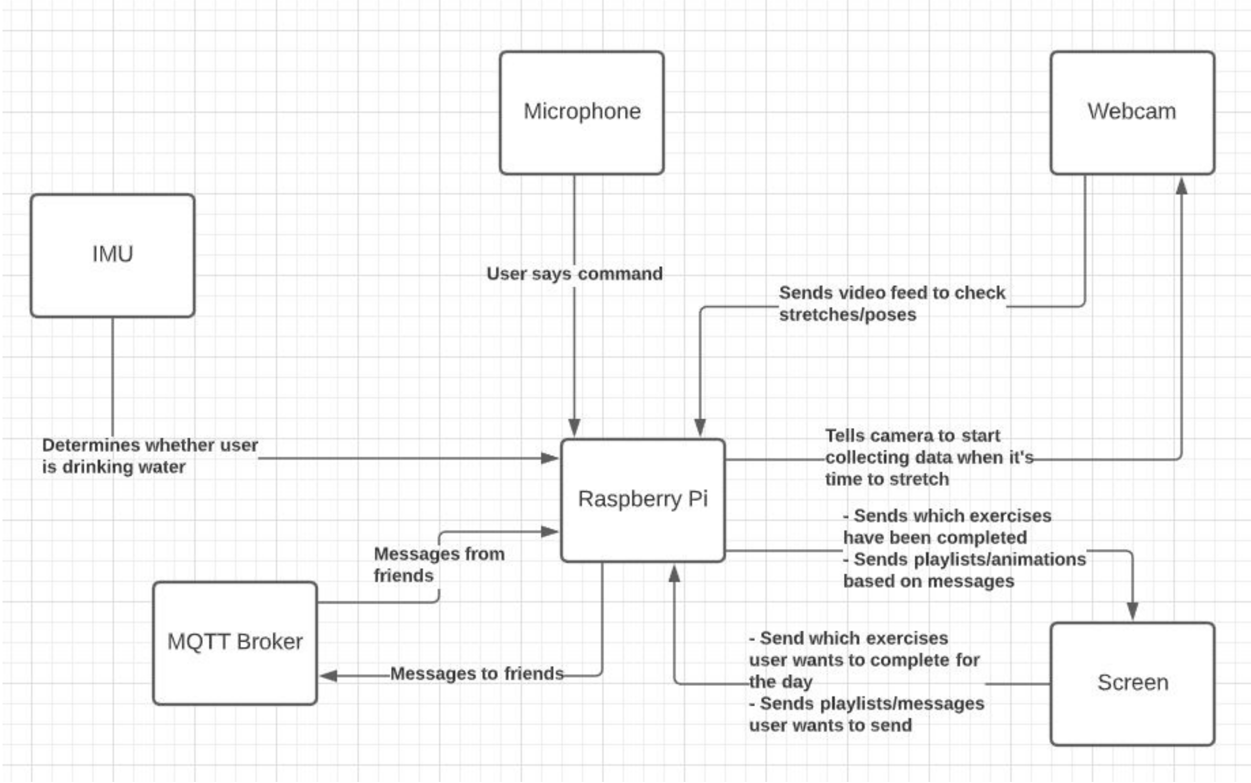
   a. Limited features
   b. Child-safe mode (likely a second quarter if have time)

Requirements

Specification	Requirements (Minimum)	Goals (Want)
Ages	Young adults - adult	All ages
Power adapter Portable power bank	N/A	Flexibility to use laptop or separate device
Laptop	Need webcam + microphone	External devices for all

High-level system description

High-level architecture and block diagram


Microphone
Webcam
IMU
User says command
Sends video feed to check stretches/poses
Determines whether user is drinking water
Raspberry Pi
Tells camera to start collecting data when it's time to stretch
- Sends which exercises have been completed
- Sends playlists/animations based on messages
Messages from friends
MQTT Broker
Messages to friends
- Send which exercises user wants to complete for the day
- Sends playlists/messages user wants to send
Screen

- Microphone, webcam, screen is apart of a laptop
    - Implementing it like this theoretically could transfer over to an external mic, webcam, screen to make this device its own separate entity

- Maybe integrate more of the features into the wearable thing

Hardware and software components and how they would interface/interact

Hardware / software component	**Requirement fulfilled + Action**	**People interface with it by…**	**Considerations (is it connected to other parts?)**
BerryIMU Accelerometer	Gesture recognition with IMU - Tracks when bottle is picked up	Wearable for the user (watch? bracelet? ring?)	Detects if user is drinking water and sends info to RPi
Laptop Webcam Camera	Pose determination with webcam - Detect stretch breaks - Detect posture	Visual of an expanding circle Following poses	RPi tells camera to start collecting data Webcam sends video feed to RPi
Laptop mic	Voice commands with mic/speaker - Alexa-inspired features - Play songs or voice memos from friends	Speak into mic on laptop or headphones	Mic commands sent to RPi
Laptop screen	Visualization module - Display encouraging messages - Display time of continuous productivity	Clicking buttons on the screen	Sends desired exercises, goals, and messages to RPi Receives messages and completed exercises
RPi (2+) MQTT (preferred) TCP/UDP	Communication with the Internet - Send music messages over broker	Send and receive messages over app (or terminal for testing)	Send and receive messages with another RPi
Web app	Good user experience	App on laptop	Takes in user

	- Easy pairing with RPis - User-friendly design		checklist and sends info to RPi

High-level task lists/timeline for design, implementation, and test

Features to do concurrently

Features	Before Midpoint Presentation	After Midpoint Presentation, but before end of Fall quarter	Assignee
Get MQTT between two people in different locations	-Sending any message remotely -Sending animation remotely	Trying to incorporate with webapp	Michelle Lam
Training model with IMU to pick up water bottle	Can use tinyML Foundational model	Watch wearable (physical design of the wearable)	Michelle Tan
Voice command using raspberry pi or can use laptop/phone mic	Recognizes one command Create list of commands: health stats, sending message, turn on reminder	Takes command that interacts with a HW feature and executes it	Michelle Tan
Recognizing pose using webcam and counting how long the pose is held by the person	May need to train model using webcam https://www.learnopencv.com/deep-learning-based-human-pose-estimation-using-opencv-cpp-python/	Have final list of poses ready and can identify all of them	Jackie

Creating customized checklist in webapp	Create sample list of exercises and create “health” tracks Start implementing on chosen platform	Uses what was checked off to remind someone + make scalable with database/aws	Isabel

After Achieving Functionality of Features

- Create mockup/prototype of what we want app to look like and its flow
  - Can do B&W wireframe/prototype in Figma for general idea/flow

Risk items

- Compatibility between individual features and main web app
- File size for sending items over MQTT
- Integrating RPi and IMU as a wearable for the wrist