Apr 20th, 1:00 PM - 3:00 PM

KeepUp

Zachary Cloutier
Sacred Heart University

Nicolas A. Treglia
Sacred Heart University

Cloutier, Zachary and Treglia, Nicolas A., "KeepUp" (2018). *Academic Festival*. 157.
https://digitalcommons.sacredheart.edu/acadfest/2018/all/157

KeepUp: Implementation of a Cloud-based Residential Activity Tracker System

Zachary Cloutier, Nicolas A. Treglia, and D. Cenk Erdil
{cloutierz, treglian76}@mail.sacredheart.edu and erdild@sacredheart.edu
School of Computing
Sacred Heart University
Fairfield, CT 06825

Abstract

With the advent of the Internet of Things (IoT), sensors and actuators start to take place in our everyday life, such as tracking activity of individuals in residential settings. One example in taking advantage of IoT in relationship with sensors and actuators is to expand upon the classical notion of setting a simple alarm clock to wake up in the morning. With a plethora of sensors and actuators installed in integrated chips, we can now implement a system that tracks an individual from the moment of responding to a wake up call, to the point of exiting the residence. In this setting, individual triggers with associated alarms are set on each sensor; where each alarm is either starts or get turned off due to triggering relevant sensors. One example might be a simple motion sensor in the bathroom, as the next step in two minutes after the wake up call. Thus, an individual either has to complete a certain pattern of sensor triggers among a list of acceptable patterns, or have to turn off each alarm that is associated with a sensor that did not get triggered. This way, a select number of routines, such as morning, medicine, sleep, etc., can be completed and also verified externally by the system.

I. Introduction

The KeepUp project is a system in which users are able to turn off an alarm clock sound only by activating motion sensors in predetermined rooms of a household. KeepUp will provide its users with a more effective way of waking up in the morning. The KeepUp iOS app is an alarm system that offers the ability to set a morning routine the night before. Depending on the type of routine scheduled by the user, the alarm sound from the app itself will only be turned off once the user triggers sensors from different areas of a living space. There will be different sensors spread out across the user's location that can only be turned off by physical input when the user enters the pre-designated area. Thus, setting a routine the night before will encourage the user to get out of his or her bed in the morning to stop the alarm sound.

The rest of this paper is organized as follows. Section II provides details of our implementation. Section III gives an overview of related work. Section IV summarizes our contributions and lists next steps.

II. Implementation

Our project utilizes Arduino IDE [1], WiFi modules [2], Passive Infrared Sensors [3], Google Firebase [4], and Xcode [5]. We use the Arduino IDE to construct code into the NodeMcu microcontroller since it is fully compatible with Arduino coding. With basic setups, the NodeMcu can be easily connected to a WiFi network being that it has a built-in ESP8266 WiFi module. Thus, the connection to a WiFi network through the NodeMcu offers a smooth and simplistic way of incorporating WiFi into our IoT project. As well as the NodeMcu microcontroller, we implemented Passive Infrared Sensors in order to collect data from non-physical human interaction. These types of sensors track human heat as it is in proximity. PIR sensors also give us the option to adjust how far its heat tracking will reach. For example, if we were to increase the range, the possibility of putting the sensors in different locations of a living space is an option for our users (i.e., ceilings). In addition, our project also makes use of Google Firebase. Firebase provides us with the tools necessary to create a strong communication through a WiFi connected

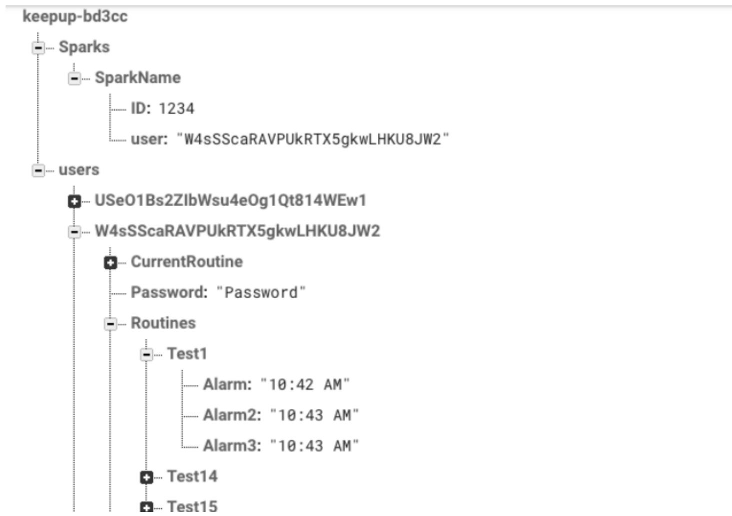

Figure 1. Google Firebase management interface. Firebase stores all user accounts, customizations, sensor information, set alarms, and so on.

NodeMcu and a cloud database that houses our system's data. Also, Firebase is completely adaptive with our iOS app since it is built with Apple's application development software, Xcode. With that said, Xcode allows us to create a unity between our NodeMcus and our databases over WiFi communications.

The final goal of this project is to release a fully functional mobile application that can be downloaded from common app stores, that enable their users to create custom routines for their daily schedules. Users will be able to choose what time they would like to physically be in a different room. When a custom alarm routine activates at preselected time and location, an alarm will ring on the user device until a particular sensor is triggered at a predefined location. Our mobile application is in communication with the cloud-based database, Google Firebase. As Figure 1 shows, a Firebase server instance will be storing user account information, which includes first and last names, an email addresses, KeepUp passwords, and so on.

A. Authentication and Access Control

The purpose of requiring a login for our application is to grant the user the ability to save previously made routines, so that they will be able to use them again in the future when needed. Firebase server will also be in communication with our WiFi modules (sensors) in order to retrieve the value of the motion sensors. The PIR motion sensors will send their value to Firebase once the alarm starts. When a change in the sensor value is determined,the alarm will then be turned off. From there, depending on what the user decides, following alarm times will proceed in accordance with the particular user routine.

B. Initial Alarm

The first alarm in a routine will not be tied to any sensors. Therefore the initial alarm will turn off through a personal interaction with the device itself. The second alarm is the first alarm that can be connected to a sensor in a different area of a living space. From the second alarm and on, the same logic applies; each selected alarm time will be connected to a sensor located in a room of choice.

C. Registering Sensors

Additionally, our mobile application will require users to pre-register each sensor module. Doing so will connect the sensors to a WiFi network and will associate sensor information with the corresponding user account, so they may begin using them for their routines. As far as the registering process of the

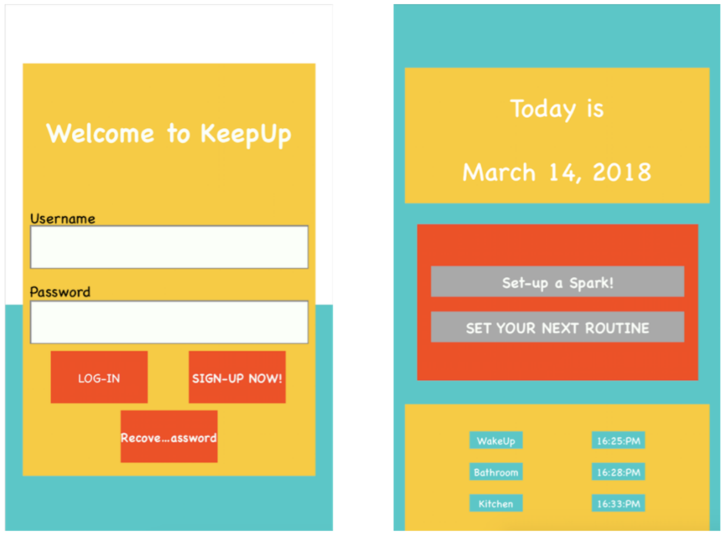

Figure 2. Login screen for KeepUp app, after downloading and installing it from the app store (left); and user home screen after logging in (right).

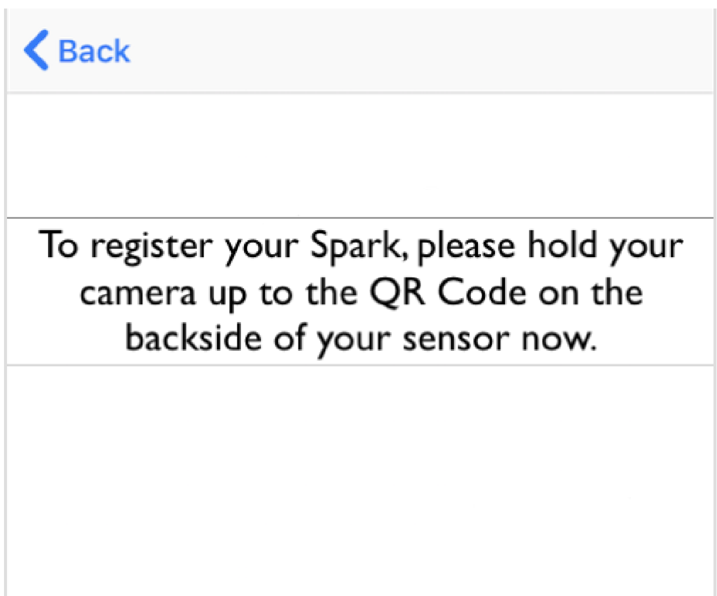

Figure 3. Mobile interface for registering sensors on KeepUp app.

sensors, we incorporate a unique identification number in combination with a Quick Response Code (QR Code) tied to each specific sensor.

D. Hardware Design

In terms of hardware design, we are using PIR sensors running on 3.3 volts on a NodeMcu [2] WiFi module. We also plan to create custom housing for these sensors using a 3D printer, that will fit both the NodeMcu and the PIR sensor, as well as the power supply. Given that our users will have the option to place sensors anywhere they would like, we will incorporate a power supply attached to the sensors. Therefore, users may place the sensors anywhere without having to designate any outlet for a sensor. In

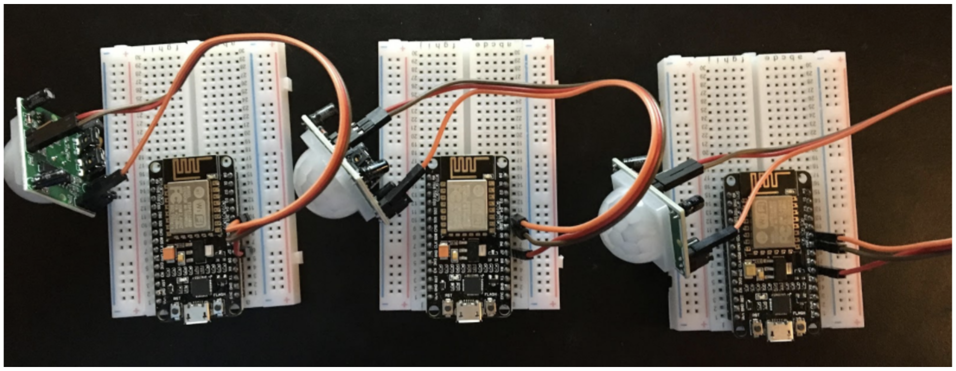

Figure 4. NodeMcu hardware that KeepUp system utilizes - three NodeMcus with three PIR sensors connected to each.

addition, the power supply will have a built-in power switch that allows the user to turn an entire sensor off at their preference.

III. Related Work

Clocky [6] is a similar product currently on the market today that also provides its users with an active way of waking up in the morning. It is an alarm clock with two wheels on each side of the clock itself, and rolls around a user's living space while its alarm rings. Clocky's movements are randomized so that the user will not be able to predict where the Clocky will before waking up. When the alarm sounds, Clocky does not perform any movement. Though if the user presses the snooze button on the initial alarm sound, then Clocky will begin to roll around after the predetermined time passes that the user sets.

Compared to KeepUp, Clocky is more of a simple alarm clock. KeepUp's alarm system lets its user customize and save different alarm routines for future use. Moreover, KeepUp provides its users with a user-friendly app that controls the alarm clock system. In addition, KeepUp utilizes infrared motion sensors whereas Clocky only has buttons for user input.

IV. Summary and Future Work

The goal of this project is to conduct research in order to create an active way people can wake up in the morning and begin their routines. We plan to construct an alarm ecosystem with the aid of an Arduino platform, so it will have a smooth connectivity to the sensors as well as our other entities. The device will allow the user to create, edit, and delete morning routines that can also be saved for different days of the week through an iOS application. We will use NodeMcus that will frequently communicate with PIR sensors in order to detect when the user triggers the sensors to turn off a step in the routine.

In the future, we plan to implement this system in other areas an effort to reach other various types of users. For example, we would like to tailor our system in order to fit the needs of the elderly as well. KeepUp could provide family members of the elderly with a sense of assurance that their loved one is living the way they need to be without physically being there. In other words, KeepUp may allow these types of users to ensure that an elderly family member is taking medication when necessary.

References

[1] "The Arduino IDE," Retrieved from https://www.arduino.cc/en/Main/Software, on September 17th, 2017.
[2] "NodeMcu, An Open-Source Firmware and Development Kit," http://www.nodemcu.com, on November 10th, 2017.
[3] "Adafruit PIR Motion Sensor," Retrieved from https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor, on December 19th, 2017.
[4] "Google Firebase Backend-as-a-Service," https://firebase.google.com, on September 31st, 2017.
[5] "Apple Xcode," https://developer.apple.com/xcode, on September 17th, 2017.
[6] "Nanda Home Clocky," https://nandahome.com/products/clocky, on September 23rd, 2017.