

RECEDE: A SOCIAL DISTANCING

SYSTEM USING ESP32 MICROCONTROLLER AND BLUETOOTH LOW ENERGY FOR PEER-TO-PEER WARNING SIGNAL

An Undergraduate Thesis

Presented to the Faculty of the

College of Information and Communications Technology

West Visayas State University

La Paz, Iloilo City

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science In Information Technology

by

Catherine G. Duero

Jason P. Esperela

Jellie Marie J. Jover

John Ray T. Godin

Albert S. Parreño

November 2022

Disclaimer

This software project and its corresponding documentation titled “*Recede: A Social Distancing System using ESP32 Microcontroller and Bluetooth for Peer to Peer Warning System*” is submitted to the College of Information and Communications Technology, West Visayas State University, in partial fulfillment of the requirements for the degree, Bachelor of Science in Information Technology. It is the product of our own work, except where indicated text.

We hereby grant the College of Information and Communications Technology

permission to freely use, publish in local or international journal/conferences, reproduce, or distribute publicly the paper and electronic copies of this software project and its corresponding documentation in whole or in part, provided that we are acknowledged.

Catherine G. Duero

Jason P. Esperela

Jellie Marie J. Jover

John Ray T. Godin

Albert S. Parreño

TABLE OF CONTENTS

Title Page

Cover Page 1

Title Page 2

Disclaimer 3

Table of Contents 4

User Manual 5

Recede User Manual 5

Dashboard Guide 6

Arduino IDE Guide 7

Troubleshooting Guide 10

FAQs 10

Contact Details 11

RECEDE

Recede was created to help mitigate the spread of Covid-19 virus. Using an ESP32 microcontroller and its Bluetooth Low Energy capability, the researchers created a system that could aid in social distancing and contact tracing of individuals who may have contracted the disease.

**USER GUIDE**

****

**Let’s Get Started!**

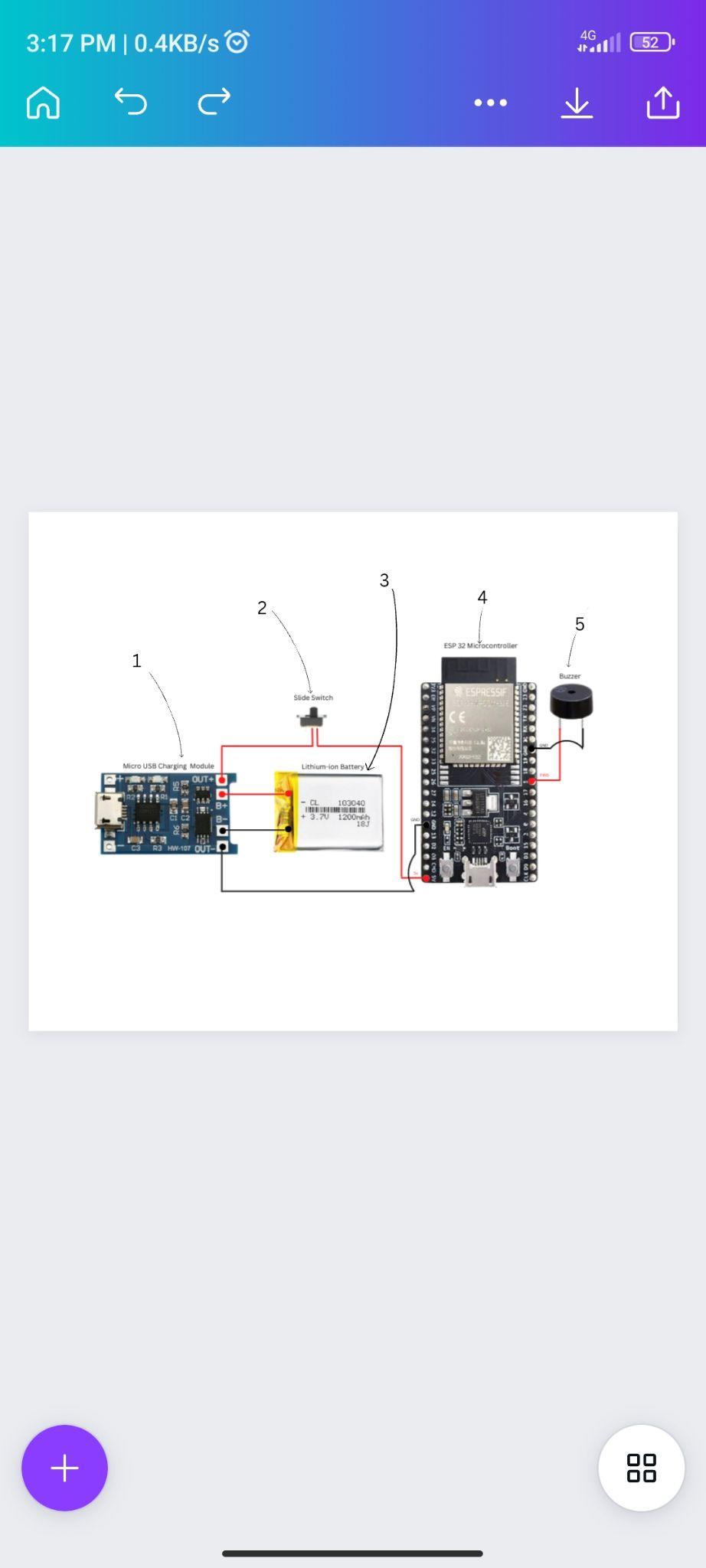
Welcome to RECEDE! A social distancing system that utilizes different tools that could mitigate the spread of COVID-19 infection and other related diseases.



**Initiating the system**

Slide the toggle switch to turn on the Recede, if you see the red light indicator then it is successfully on. Next, wear it on your neck or carry the recede in your pocket or like a keychain. Then, you’re ready to go. If you hear the buzzer emit a pulsing sound, observe proper social distancing. If not, you will hear a continuous sound which indicates that you have been logged into the database that is subject for contact tracing if any positive individual is detected in your area. 

**Schematic Diagram**

****

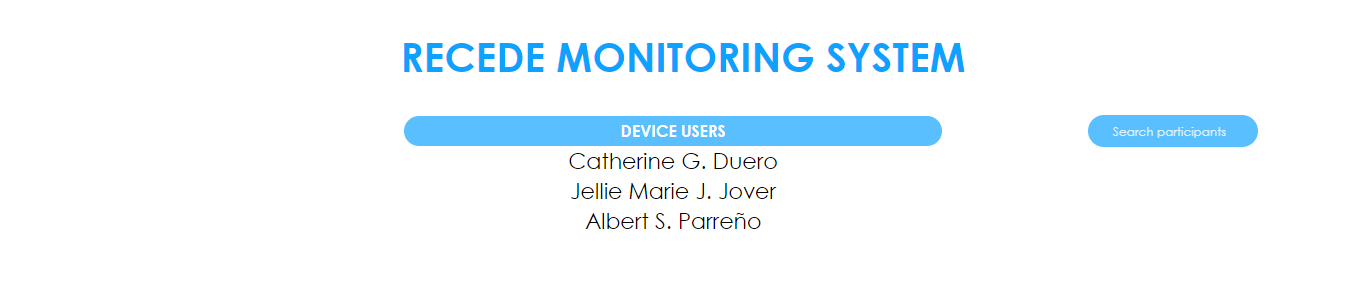
1. Micro USB Charging Module - it is connected to the lithium-ion battery to provide power if the battery runs out.
2. Slide Switch - used to turn on and off the Recede.
3. Lithium-ion Battery - it is the power source of the device. It has 3.7 volts with 1200mAh capacity.
4. ESP32 Microcontroller - it is the main component of the device. It is used to program the device like setting the allowed distance, configure the information per device, and assign to which Wi-Fi it should be connected.
5. Buzzer - it emits sound whenever the social distancing is violated.



**DASHBOARD GUIDE**



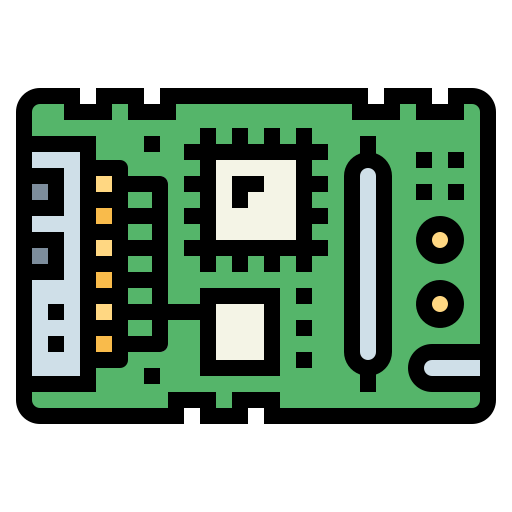
**Searching for a User**



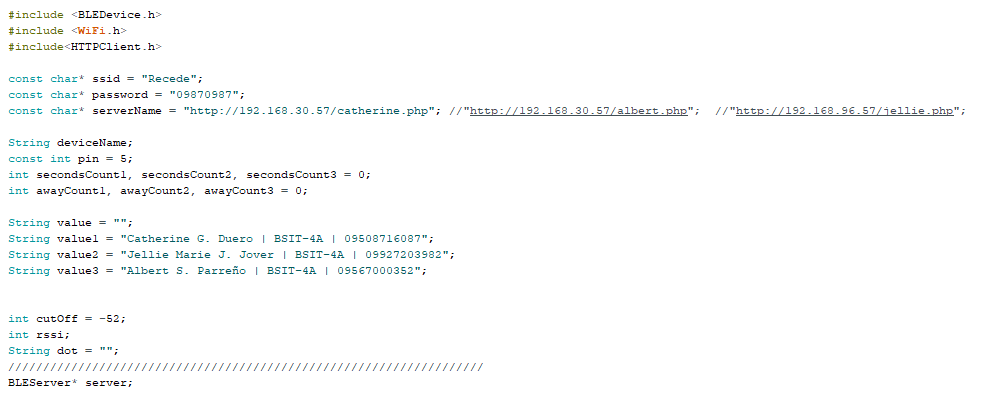
In the upper right corner of the dashboard, you can click the search bar to search for participants that use Recede. This will come in handy if there are a lot of participants if this was implemented inside the campus of West Visayas State University. If you press a participant under device users column it will redirect you to a separate page.



This will be the landing page if you perform the previous action where it will provide you with lists of individuals who came close contact with Albert. This provides the name, year and section, cell phone number, and the date and time of contact. Also this page will let you filter the date of contact in the upper right corner of the page.



**ARDUINO IDE GUIDE**



~ The “n” below stands as a placeholder for a number.

- one number corresponds to one device

1. const char\* “ssid”

- Data type: String

- This is where we declare the name of the Wi-Fi that we are using in our system.

1. const char\* “password”

- Data type: String

- This is where we declare the password of the Wi-Fi that we are using in the system.

1. “serverName”

- Data type: String

- This is where we declare the IP address of the Wi-Fi we are connected to and the name of the .php file of the respondent in the database table

1. int “secondsCount(n)” and “awayCount(n)”

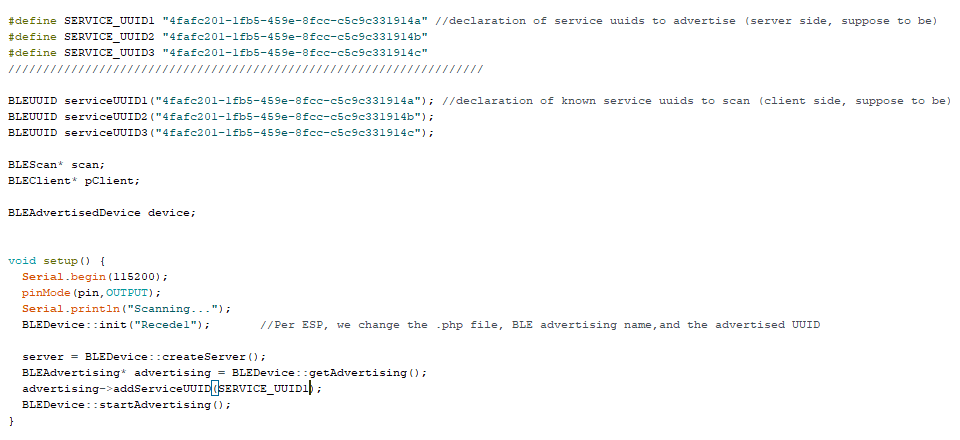
- Data type: integer

- These are the placeholder for the number of seconds (counting of the seconds) for the device if it is below one meter or over one meter.

1. String “value(n)”

- Data type: String

- This is where we declare the details of the user that holds a certain Recede module. One string of details equates to one user.



1. #define SERVICE\_UUID(n)

- Data type: String

- This is the declaration of the UUID that is being used or advertised by the device, this is a server-side code declaration.

- We only use one UUID (unique) for each device.

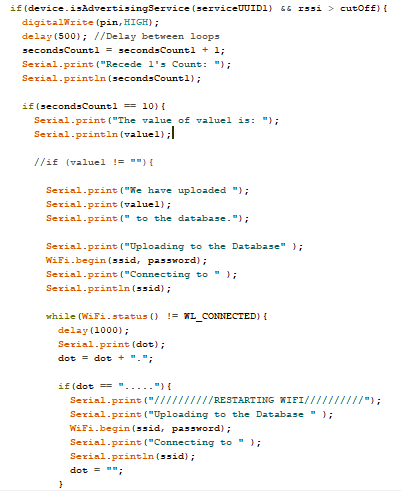
1. BLEUUID serviceUUID(n)

- Data type: String

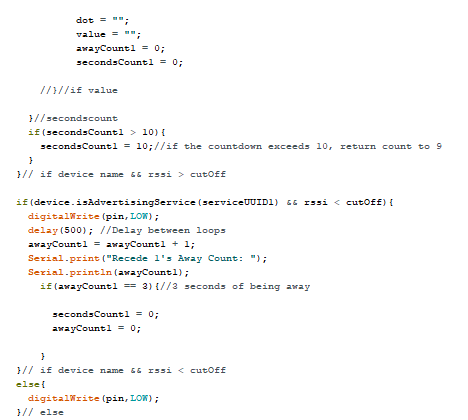
- This is a declaration of the Bluetooth Low Energy Universal Unique Identifier (BLEUUID) of another device involved in the Recede system. One declared BLEUUID corresponds to one Recede device. This is a client side declaration.

1. BLEDevice::init(“bluetoothDeviceName”)

- Declared within the parameter is the device’s name that will be shown when scanned.







**TROUBLESHOOTING**

**GUIDE**

* Device does not turn on.

- Check if the slide switch is turned on. If the device still does not glow its red indicator, charge the device for 20 minutes and turn it on again.

* Device does not emit a warning noise.

- Open the device and click the EN button. There could just be a bug in the starting of the device.

- If it still doesn’t sound, contact the system admin for troubleshooting.

* Device got stuck in a steady buzzing state.

- Open the case and click the EN button to restart the device.



**FAQs**

* How long does it take for my device to be fully charged?

- Devices are completely operable at 20 minutes of charging using a 5 watts charging adapter.

* Am I allowed to modify the code of the device to update or modify my details?

- No. The system admin is the sole manager and modifier of the codes within the device.

* Am I allowed to open the case of the device?

- Yes. You are allowed to open the case of the device if you were to reset/restart the device’s code.

* Will other people who are not involved in the system who hold an ESP32 be able to read/get my details?

- No. Your devices have unique identifiers embedded on each of them so no external devices are able to intrude with the system’s operation.

