**Delloite Techno Utsav**

Round 2.3: Working Prototype

Team CGC\_Symbiotes

Kartik Chawla

Bhavya Jain

Manik Nagpal

Theme Alignment: Internet of Things

Industry Alignment: Life Sciences and Health Care

Patient Surveillance

**Executable Code**

#include <iostream>

#include <string>

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include "twilio.hh" //The class library required to send the distress message

void SOS();

void random();

void random()

{

std::mt19937 eng(rd()); // seed the generator

std::uniform\_int\_distribution<> distr(0, 190); // define the range

for(int n=0; n<40; ++n)

std::cout << distr(eng) << ' '; // generate numbers

} //This code is to mimic the input from arduino hardware

int main()

{

int i;

while(Arduino!=NULL) //The program will continuously run until the machine is switched off

{

arduino.getvalue(i); //To scan the heartbeat pulse using arduino board and sensor

if((i>60)||(i<100)) //Condition for normal pulse rate

{

cout<<"Condition=normal";

}

else if(i<60) //Condition for batycardia

{

cout<<"Batycardia";

SOS();

}

else if(i>100) //Condition for tachycardia

{

cout<<"Tachycardia";

SOS();

}

}

return 0;

}

void SOS()

{

arduino.getvalue(i);

send message to (int m); //m= mobile number of medical staff

}

int main(int argc, char \* argv[])

{

int cmd;

std::string account\_sid;

std::string auth\_token;

std::string message;

std::string from\_number;

std::string to\_number;

std::string picture\_url;

bool verbose = false;

opterr = 0;

while ((cmd = getopt(argc, argv, "a:s:m:f:t:p:vh?")) != -1) {

switch (cmd) {

case '?':

case 'h':

printf("Twilio C++ Example Help:\n");

printf("-a: Account\t\t"

"(ex: -a \"ACXXXXX\")\n");

printf("-s: Auth Token\t\t"

"(ex: -s \"your\_token\")\n");

printf("-f: From Number\t\t"

"(ex: -f \"+18005551212\")\n");

printf("-t: To Number\t\t"

"(ex: -t \"+18005551212\")\n");

printf("-m: Message to send\t"

"(ex: -m \"Hello, Twilio!\")\n");

printf("-p: (Opt.) URL to Image\t"

"(ex: -p \"Hello, Twilio!\")\n");

printf("-v: Verbose Mode\n");

printf("-h: This help dialog\n");

return 0;

case 'a':

account\_sid = optarg;

break;

case 's':

auth\_token = optarg;

break;

case 'm':

message = optarg;

break;

case 'f':

from\_number = optarg;

break;

case 't':

to\_number = optarg;

break;

case 'p':

picture\_url = optarg;

break;

case 'v':

verbose = true;

break;

default:

abort();

}

}

if ( account\_sid.empty() || auth\_token.empty() || from\_number.empty()

|| to\_number.empty() || message.empty() ) {

std::cout<< "You didn't include all necessary inputs!\n"

"Call using -h for help.\n" << std::endl;

return -1;

}

// Instantiate a twilio object and call send\_message

std::string response;

twilio::Twilio \*twilio = new twilio::Twilio(account\_sid, auth\_token);

bool message\_success = twilio->send\_message(

to\_number,

from\_number,

message,

response,

picture\_url,

verbose

);

delete twilio;

// Report success or failure

if (!message\_success) {

if (verbose) {

std::cout << "Message send failed." << std::endl;

if (!response.empty()) {

std::cout << "Response:" << std::endl

<< response << std::endl;

}

}

return -1;

} else if (verbose and message\_success) {

std::cout << "SMS sent successfully!" << std::endl;

std::cout << "Response:" << std::endl << response

<< std::endl;

}

return 0;

}