

TESTING

PROJECT TITLE :

IoT Based Safety Gadget for Child Safety Monitoring and Notification

TEAM ID : PNT2022TMID38861

TEAM LEADER : D.Siva Sankar

TEAM MEMBER 1 : R.Gokul

TEAM MEMBER 2 : S.Jeevanantham

TEAM MEMBER 3 : N.Logesh Kumar

INTRODUCTION:

Testing is based on the behaviour of our application It is referred five parts of testing those are load testing, stress testing, spike testing, endurance testing, resilience testing.

LOAD TESTING:

Load test is depend up on the application speed our device is worked at more effective at all the time to performed in online space the load level testing result is minimum amount of load is uploaded and our application runs effectively it is depend up on our data speed and quality of the data .

STRESS TESTING:

Stress testing is referred to how much stress it will handle in all stages we upload the particular amount of stress its performed good it is advancement of crime rate security is having its alarming significance for school children, some applications exist to address issue and most of them internet connection which makes a solution expensive in this application we present a low cost solution and the stress test result is good .

SPIKE TESTING:

Spike testing is during the time how the system will behave or perform it , the monitoring system is performed in all time to our data connection and quality .if you want to connect your mobile to device set the particular time otherwise it will be off mode set the time to the app to connect a device example child's school timing to return to home timing our application is performed good .

ENDURANCE TESTING:

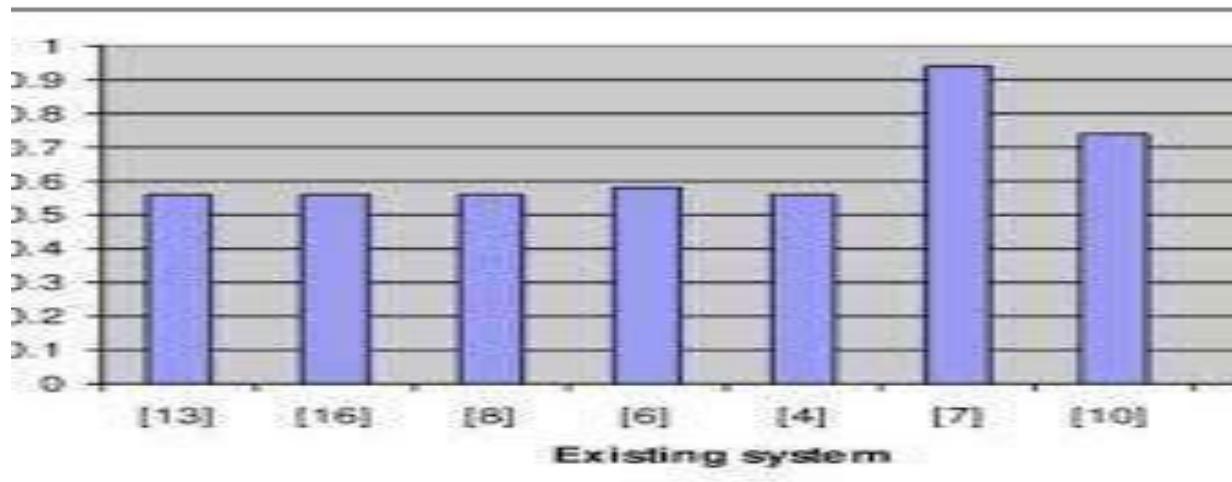
Endurance test will be approved to during the continuous stress how the application will serviced GPS provides the accurate data for tracking the children currently located and along with it also update the parents and ,GSM updates the sms to our parents mobile application this application is used to support child health care laevel and the notification is send to our parents or guide 's mobile , panic button is performed minimum level of stress to using good.

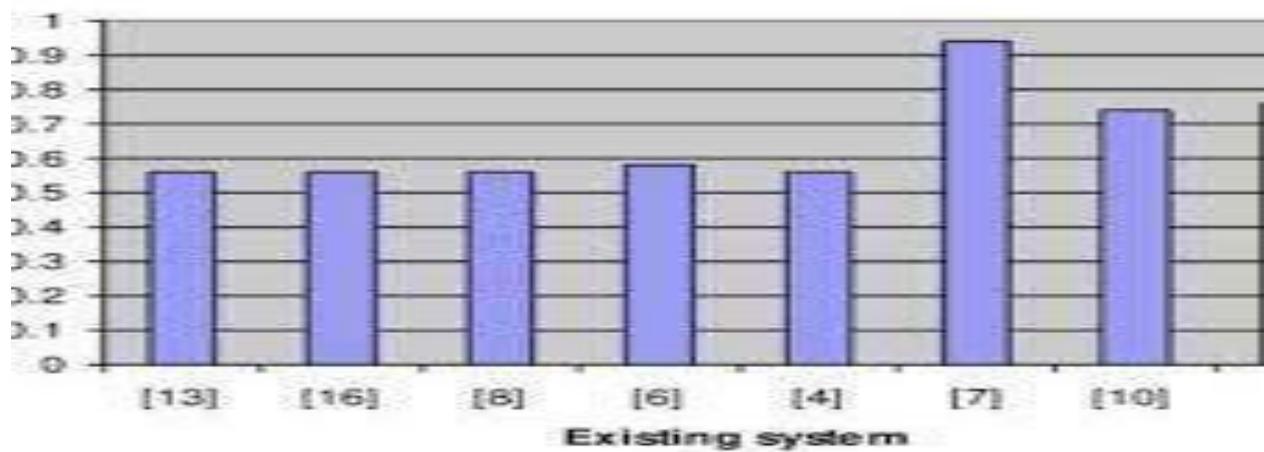
RESILENCE TESTING:

It is divided to two phase active and passive node ,active node whenever node is comming that node will pick the load if active node is down the passive is pick the load automatically the testing result is good

SAMPLE GRAPH:

Steady state





Arduino-1

Child safety device

Circuit design child safety monitor

child safety monitoring device

1 (Arduino Uno R3)

```

1 //Child safety device
2 char text1[] = "SGFVTG,054.7,t,034.4,M,005.5,N,010.2,K";
3 char text2[] = "SGPOGA,134658.00,5106.9792,N,11402.3003,W,2,09,1.0,1048.47,M,-16.27,M,08,AAAA*60";
4 int redbutton=3,greenbutton=2;
5 int redLED=5,greenLED=4;
6 int redstate=0,greenstate=0;
7 int buzzer=8;
8 int status;
9 int msg=9;
10 int greenCount=0;
11 #define count 3
12 void setup()
13 {
14     pinMode(redbutton,INPUT);
15     pinMode(greenbutton,INPUT);
16     pinMode(redLED,OUTPUT);
17     pinMode(greenLED,OUTPUT);
18     attachInterrupt(digitalPinToInterrupt(redbutton),Danger,RISING);
19     attachInterrupt(digitalPinToInterrupt(greenbutton),Safe,RISING);
20     Serial.begin(9600);
21     pinMode(msg,OUTPUT);
22     digitalWrite(greenLED,HIGH);
23 }
24 //SHOW THE STATUS OF CHILD
25 void loop()
26 {
27     if(status)
28     {
29         Serial

```

Circuit design child safety monitoring device

```
28 {
29     tone(buzzer,100,10);
30     digitalWrite(msg,HIGH);
31 }
32 else
33 {
34     digitalWrite(buzzer,LOW);
35     digitalWrite(msg,LOW);
36 }
37 //G/S
38 delay(200);
39 Serial.write(text1);
40 Serial.write("/");
41 delay(500);
42 Serial.write(text2);
43 delay(500);
44 }
45 void Danger()
{
46     digitalWrite(redLED,HIGH);
47     digitalWrite(greenLED,LOW);
48     status=1;
49 }
50
51
52 }
53 void Safe()
{
54     greencount=greencount+1;
55 }
```

Circuit design child safety monitoring device

```
40 Serial.write("/");
41 delay(500);
42 Serial.write(text2);
43 delay(500);
44 }
45 void Danger()
{
46     digitalWrite(redLED,HIGH);
47     digitalWrite(greenLED,LOW);
48     status=1;
49 }
50
51
52 }
53 void Safe()
{
54     greencount=greencount+1;
55
56     if(greencount==count)
57     {
58         digitalWrite(greenLED,HIGH);
59         digitalWrite(redLED,LOW);
60         status=0;
61         greencount=0;
62     }
63 }
64
65
66
67 }
```

Arudino-2

Circuit design child safety monitor

child safety monitoring device

```

1 #include <LiquidCrystal.h>
2
3 #define ADC_VREF_TYPE ((0<<REFS1) | (0<<REFS0) | (0<<ADLAR))
4
5 float stepADC = 0.0048828125;
6
7 LiquidCrystal lcd1(5, 4, 3, 13, 2, 6);
8 LiquidCrystal lcd2(12, 11, 10, 9, 8, 7);
9
10 char sSecventa[119];
11
12 //declarare global deoarece in cazul calculatiilor din ecranul 2
13 //memoria este deja scrisa si ar fi trebuit o alta initializare
14 //cu multe variabile, etc..
15 static char *aValori[2];
16 static char *token;
17 static char *aValori1[9];
18 static char *token1;
19 static char *aValori2[15];
20 static char *token2;
21
22 int contor = 0;
23 bool ecran = 0;
24 bool stareButoon = 0;
25 bool stareButoonAnterioara = 0;
26 //-----
27
28 unsigned int citesteADC(unsigned int adc_input)
29 {
30     Serial Monitor

```

Circuit design child safety monitor

child safety monitoring device

```

30     ADMUX = adc_input | ADC_VREF_TYPE;
31
32     //delay necesar pentru stabilirea ADC pt tens
33     //de intrare analogica
34     delayMicroseconds(10);
35
36     //start conversie
37     ADCSRA |= (1<<ADSC);
38     //Asteptare finalizare conversie
39     while ((ADCSRA & (1<<ADIF))==0){}
40     ADCSRA |= (1<<ADIF);
41
42     //returnam rezultat pe 16 bitia
43     return ADCW;
44 }
45
46 //cod recunoastere/segmentare secventa
47 void segmentareSecventa()
48 {
49     //segmentare secventa principală
50     token = strtok(sSecventa, "/");
51     static int increment = 0;
52     while (token != NULL)
53     {
54         aValori[increment++] = token;
55         token = strtok(NULL, "/");
56     }
57 }
58 Serial Monitor

```

Circuit design child safety monitor

child safety monitoring device

```

62 while (token1 != NULL)
63 {
64     aValori1[increment1++] = token1;
65
66     token1 = strtok(NULL, ",");
67 }
68
69 //segmentare seconta 2
70 token2 = strtok(aValori[1], ",");
71 static int increment2 = 0;
72 while (token2 != NULL)
73 {
74     aValori2[increment2++] = token2;
75
76     token2 = strtok(NULL, ",");
77 }
78
79 //incheiere segmentare seconte -----
80 }
81
82 void ecranulUnu()
83 {
84     //afisare viteza
85     static float iViteza = 0;
86     if (strcmp(aValori1[0], "$GPVTG") == 0)
87     {
88         iViteza = atof(aValori1[7]);
89     }
90
91     //setare cursori coloana 0, linia 1
92     lcd1.setCursor(0, 1);
93     lcd1.print("Viteza:");
94     lcd1.print(iViteza);
95     lcd1.print("km/h");
96
97     //afisare temp
98     static long int temp[4];
99     static int iSateliti = 0;
100    if (strcmp(aValori2[0], "$GPGLL") == 0)
101    {
102        //stocare sir
103        temp[0] = atof(aValori2[1]);
104        //stocare secunde
105        temp[3] = temp[0] % 100;
106        temp[0] = temp[0] / 100;
107        //stocare minute
108        temp[2] = temp[0] % 100;
109        temp[0] = temp[0] / 100;
110        //stocare ore
111        temp[1] = temp[0];
112
113        iSateliti = atoi(aValori2[7]);
114    }
115
116    Serial Monitor

```

Circuit design child safety monitor

child safety monitoring device

```

static float iViteza = 0;
if (strcmp(aValori1[0], "$GPVTG") == 0)
{
    iViteza = atof(aValori1[7]);
}
//setare cursori coloana 0, linia 1
lcd1.setCursor(0, 1);
lcd1.print("Viteza:");
lcd1.print(iViteza);
lcd1.print("km/h");

//afisare temp
static long int temp[4];
static int iSateliti = 0;
if (strcmp(aValori2[0], "$GPGLL") == 0)
{
    //stocare sir
    temp[0] = atof(aValori2[1]);
    //stocare secunde
    temp[3] = temp[0] % 100;
    temp[0] = temp[0] / 100;
    //stocare minute
    temp[2] = temp[0] % 100;
    temp[0] = temp[0] / 100;
    //stocare ore
    temp[1] = temp[0];
    iSateliti = atoi(aValori2[7]);
}

Serial Monitor

```

TINKER CAD child safety monitoring device

```

117 lcd1.print("Temp: ");
118 lcd1.print(temp[1]);
119 lcd1.print(":");
120 lcd1.print(temp[2]);
121 lcd1.print(".");
122 lcd1.print(temp[3]);
123
124
125 //setare cursor: coloana 0, linia 3
126 lcd2.setCursor(0,0);
127 lcd2.print("Nr. sati: ");
128 lcd2.print(iSatelitii);
129
130 //citire termometru de pe pinul analogic "0"
131 unsigned int sensorValue = citesteADC();
132 //calculare temperatura in functie de valorile din datasheetul termometrului
133 float fTemperatura = (stepADC*sensorValue-0.5)*100;
134
135 //setare cursor: coloana 0, linia 3
136 lcd2.setCursor(0, 1);
137 lcd2.print("Temp: ");
138 lcd2.print(fTemperatura, 2);
139 lcd2.print(" C");
140 //inchiere prelucrare ecran unu-----
141 }
142
143 //afisare/selectare date pe ecranul doi
144 void ecranulDoi()
145 {
146 Serial Monitor

```

TINKER CAD child safety monitoring device

```

147 //calculare, afisare latitudine
148 static float fLatitudine = 0;
149 static float fLongitudine = 0;
150 static float fAltitudine = 0;
151 if (strcmp(aValori2[0], "SGPOGA") == 0)
152 {
153     fLatitudine = atof(aValori2[4]);
154     fLongitudine = atof(aValori2[4]);
155     fAltitudine = atof(aValori2[9]) - atof(aValori2[11]);
156 }
157 //afisare date
158 //setare cursor: coloana 0, linia 1
159 lcd1.setCursor(0, 1);
160 lcd1.print("Lat: ");
161 lcd1.print((int)fLatitudine/100+((int)(fLatitudine)%100+(fLatitudine-(int)(fLatitudine))/60));
162 lcd1.print(aValori2[3]);
163 //setare cursor: coloana 0, linia 2
164 lcd2.setCursor(0, 0);
165 lcd2.print("Long: ");
166 lcd2.print((int)fLongitudine/100+((int)(fLongitudine)%100+(fLongitudine-(int)(fLongitudine))/60));
167 lcd2.print(aValori2[5]);
168 //setare cursor: coloana 0, linia 3
169 lcd2.setCursor(0, 1);
170 lcd2.print("Alt: ");
171 lcd2.print(fAltitudine);
172 lcd2.print("m");
173 //inchiere prelucrare ecran doi-----
174 }

```

Circuit design child safety monitoring device

```

177 //-----SETUP-----
178 //-----
179 void setup()
180 {
181   Serial.begin(9600);
182
183   lcd1.begin(2,16);
184   lcd2.begin(2,16);
185 }
186 //-----
187 //-----LOOP-----
188 //-----
189 void loop()
190 {
191   if(contor < 1)
192   {
193     Serial.readBytes(sSecventa, 119);
194     Serial.println(sSecventa);
195     segmentareSecventa();
196     contor++;
197   }
198
199   //citire stare buton
200   /*stareButon = (PIND & (1 << PIND2));
201
202   if(stareButon != stareButonAnteriora)
203   {
204     Serial.println("stareButon != stareButonAnteriora");
205     delay(200);
206     ecran = !ecran;
207
208     if( stareButon == 0 )
209     {
210       stareButon = !stareButon;
211     }
212   }
213   stareButonAnteriora = stareButon;
214
215 */
216
217   ecranulUnu();
218   delay(1000);
219   lcd1.clear();
220   lcd2.clear();
221
222   ecranulDoi();
223   delay(1000);
224   lcd1.clear();
225   lcd2.clear();
226
227 }

```

Serial Monitor

Circuit design child safety monitoring device

```

200 //citire stare buton
201 /*stareButon = (PIND & (1 << PIND2));
202
203 if(stareButon != stareButonAnteriora)
204 {
205   delay(200);
206   ecran = !ecran;
207
208   if( stareButon == 0 )
209   {
210     stareButon = !stareButon;
211   }
212 }
213 stareButonAnteriora = stareButon;
214
215 */
216
217   ecranulUnu();
218   delay(1000);
219   lcd1.clear();
220   lcd2.clear();
221
222   ecranulDoi();
223   delay(1000);
224   lcd1.clear();
225   lcd2.clear();
226
227 }

```

Serial Monitor

Arudino-3

Parents and friends Screen

Circuit design child safety monitoring device

```

1 //parents and friends screen
2 #include <LiquidCrystal.h>
3
4 LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
5 int status=8;
6 int data;
7 void setup()
8 {
9     pinMode(status,INPUT);
10    lcd.begin(16, 2);
11    lcd.setCursor(0,0);
12    lcd.print("Child status");
13    pinMode(8,INPUT);
14    Serial.begin(9600);
15 }
16
17 void loop()
18 {
19     data=digitalRead(status);
20     Serial.println(data);
21     if(data==HIGH)
22     {
23         lcd.setCursor(0,1);
24         lcd.print("Help me!");
25     }
26     if(data==LOW)
27     {
28         lcd.setCursor(0,1);
29         lcd.print("I'm safe!");
30     }
31 }
32
  
```

Serial Monitor

Circuit design child safety monitoring device

```

5 int status=8;
6 int data;
7 void setup()
8 {
9     pinMode(status,INPUT);
10    lcd.begin(16, 2);
11    lcd.setCursor(0,0);
12    lcd.print("Child status");
13    pinMode(8,INPUT);
14    Serial.begin(9600);
15 }
16
17 void loop()
18 {
19     data=digitalRead(status);
20     Serial.println(data);
21     if(data==HIGH)
22     {
23         lcd.setCursor(0,1);
24         lcd.print("Help me!");
25     }
26     if(data==LOW)
27     {
28         lcd.setCursor(0,1);
29         lcd.print("I'm safe!");
30     }
31 }
32
  
```

Serial Monitor

User Acceptance testing:

Circuit design child safety monitor

tinkercad.com/things/4OO8kACeMGc-child-safety-monitoring-device/editr

Inbox Login youtube wokwi cloud

child safety monitoring device

Simulator time: 00:00:01.126

Code Stop Simulation Send To

1 (Arduino Uno R3)

```

1 //Child safety device
2 char text1[] = "$GPVTG,054.7,T,034.4,M,005.5,N,010.2,K";
3 char text2[] = "$GPGGA,134658.00,5106.9792,N,11402.3003,W,2,09,1.
4 int redbutton=3,greenbutton=2;
5 int redLED=5,greenLED=4;
6 int redstate=0,greenstate=0;
7 int buzzer=8;
8 int status;
9 int msg=9;
10 int greencount=0;
11 #define count 3
12 void setup()
13 {
14   pinMode(redbutton,INPUT);
15   pinMode(greenbutton,INPUT);
16   pinMode(redLED,OUTPUT);
17 }

```

Serial Monitor

```
$GPVTG,054.7,T,034.4,M,005.5,N,010.2,K/$GPGGA,134658.00,5106.9792,N,11402.3003,W,2,09,1.
```

Send Clear