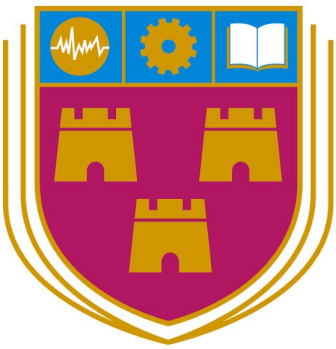
Institiúid Teicneolaíochta Cheatharlach



INSTITUTE *of* TECHNOLOGY CARLOW

At the Heart of South Leinster

LifeBuoy Monitor
Technical Document/
User Manual

BSc (Hons) in Software Development

Name: Garry Byrne

Student ID: C00120055

Year: 4th Year

Supervisor:Dr. Oisin Cawley

Due date: 18-04-2018

Table Of Contents

User Manual

System Requirements

- Android Device

Installation Instructions

- Install apk

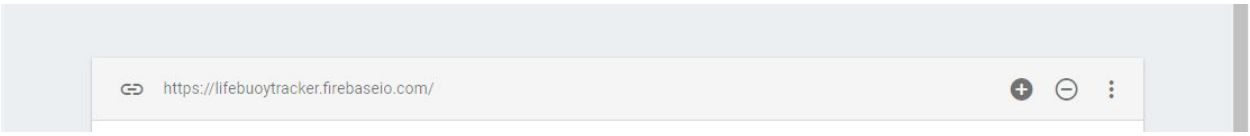
System usage

App

- Register a user account(must be same as email used to register the device)
- Log In
- Devices displayed on map.
- Click Devices to see list of all your active devices.
- Click into wanted device.
- Update Location/Reset Alarm of selected lifebuoy
- View new location of lifebuoy on map
- Logout

Database

- Create a firebase account.
- Create a project within firebase.
- Click database in the right hand menu.
- Click Real Time database from the two options provided.
- Copy the link below.



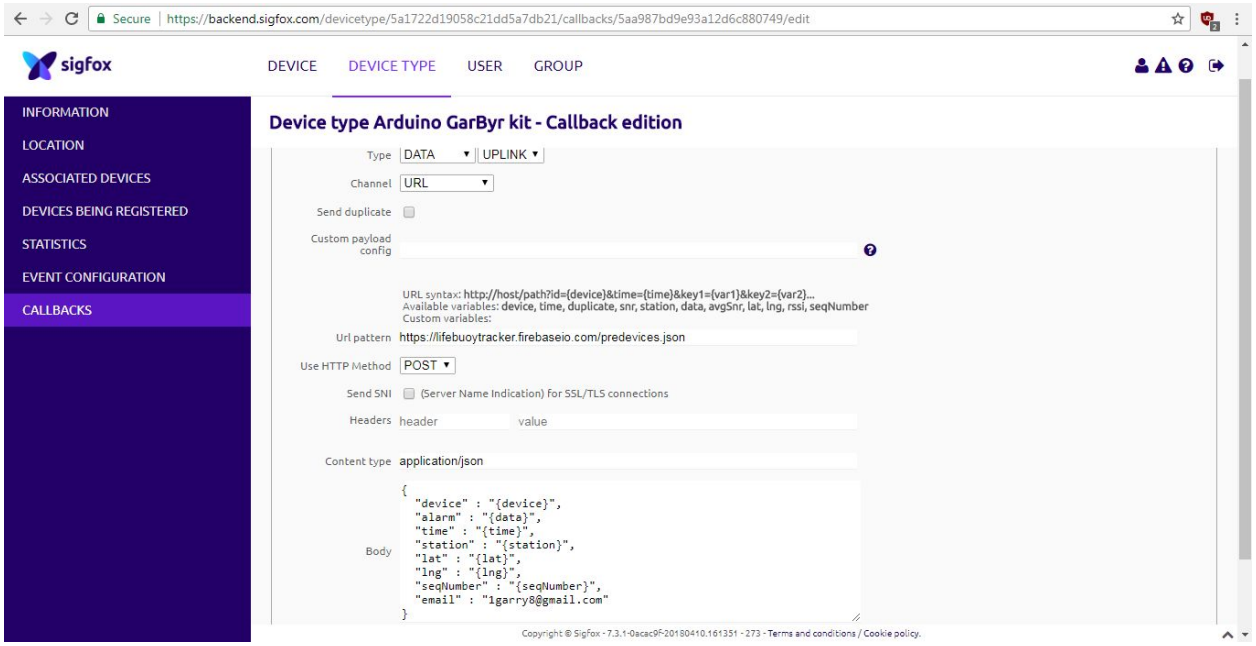
- You will need this link for setup of custom callback with the Sigfox backend.

Device

- Follow these instructions http://makers.sigfox.com/getting-started/#register
- If you have multiple devices follow these instructions after step 1

Custom Callback

- Log into the sigfox backend https://backend.sigfox.com/auth/login
- Click Device at the top
- Click Device Type
- Click on callbacks in the right hand menu
- Click new in the top right hand corner
- Click on custom callback
- Fill in your details like the image below


Secure | https://backend.sigfox.com/devicetype/5a1722d19058c21dd5a7db21/callbacks/5aa987bd9e93a12d6c880749/edit
sigfox
DEVICE
DEVICE TYPE
USER
GROUP
INFORMATION
LOCATION
ASSOCIATED DEVICES
DEVICES BEING REGISTERED
STATISTICS
EVENT CONFIGURATION
CALLBACKS
Device type Arduino GarByr kit - Callback edition
Type DATA UPLINK
Channel URL
Send duplicate
Custom payload config
URL syntax: http://host/path?id={device}&time={time}&key1={var1}&key2={var2}...
Available variables: device, time, duplicate, snr, station, data, avgSnr, lat, lng, rssi, seqNumber
Custom variables:
Url pattern https://lifebuoytracker.firebaseio.com/predevices.json
Use HTTP Method POST
Send SNI (Server Name Indication) for SSL/TLS connections
Headers header value
Content type application/json
Body
{
"device" : "{device}",
"alarm" : "{data}",
"time" : "{time}",
"station" : "{station}",
"lat" : "{lat}",
"lng" : "{lng}",
"seqNumber" : "{seqNumber}",
"email" : "1garry8@gmail.com"
}
Copyright © Sigfox - 7.3.1-0acac9f-20180410.161351 - 273 - Terms and conditions / Cookie policy.

- Click okay.

Technical Document(Code)

Arduino Code

```
#include <Wire.h>
#include <SPI.h>
#include <SigFox.h>
#include <ArduinoLowPower.h>
#include <Adafruit_LIS3DH.h>
#include <Adafruit_Sensor.h>


double averageX, averageY, averageZ;
int trigger = 0;
int num = 0;
int t = 0;

Adafruit_LIS3DH lis = Adafruit_LIS3DH();

void setup() {
#ifndef ESP8266
  while (!Serial);     // will pause Zero, Leonardo, etc until serial console opens
#endif

  Serial.begin(9600);



  if (! lis.begin(0x18)) {   // change this to 0x19 for alternative i2c address
    Serial.println("Couldnt start");
    while (1);
  }
  Serial.println("LIS3DH found!");

  lis.setRange(LIS3DH_RANGE_4_G);   // 2, 4, 8 or 16 G!
  sensors_event_t event;
  lis.getEvent(&event);
```

```
  int measurementRuns = 50;

sendString(t);// initial setup message


  for(int i = 0; i < measurementRuns; i++)// code to tget average in x y z planes ie taking any
sway into account.
  {
     averageX = averageX + event.acceleration.x;
     Serial.print(i);Serial.print("       ");Serial.println(averageX);
     averageY = averageY + event.acceleration.y;
     Serial.print(i);Serial.print("       ");Serial.println(averageY);
     averageZ = averageZ + event.acceleration.z;
     Serial.print(i);Serial.print("       ");Serial.println(averageZ);
     delay(200);
  }

  // getting the average
  averageX = averageX/measurementRuns;
  averageY = averageY/measurementRuns;
  averageZ = averageZ/measurementRuns;
  }


void loop()
{
  sensors_event_t event;
  lis.getEvent(&event);

  //waiting for movemement to occur

  if(event.acceleration.x > (averageX + 5) || event.acceleration.x < (averageX - 5) ||
event.acceleration.y > (averageY + 5) || event.acceleration.y < (averageY - 5) ||
event.acceleration.z > (averageZ + 10) || event.acceleration.z < (averageZ - 10))
    {
      if(trigger == 0)
        {
          t = 1;
          sendString(t);
        }
    }
}
```

```
void loop1()
{

}



void sendString(t)
{
  if(trigger == 0)
  {
    SigFox.begin(); // send alarm message
    Serial.println("sent");
    delay(100);
    SigFox.beginPacket();
    SigFox.write(t);

    int ret = SigFox.endPacket();

    if (ret > 0) {
      Serial.println("No transmission");
    } else {
      Serial.println("Transmission ok");
    }

    SigFox.end();
    trigger++;
    loop1();
  }
}
```

Cloud Code

```
var functions = require('firebase-functions');
var admin = require('firebase-admin');
admin.initializeApp(functions.config().firebase);

exports.PushWarning = functions.database.ref('/devices/{alarm}')
    .onUpdate((snapshot) => {
```

```
    // Grab the current value of what was written to the Realtime
Database.

var test = snapshot.after.val();

        if(test.alarm === "01000000")
    {
        var email = test.email;
        var from = "Lifebuoy Monitor";
        var reply = "hello";
        var reply1 = "garry";

admin.database().ref('/pushtokens').orderByChild('uid').once('value').then
((alltokens) => {
            var rawtokens = alltokens.val();
        var tokens = [];
        processtokens(rawtokens).then((processedtokens) => {

            for (var token of processedtokens) {
                if(token.email == email)
                {
                    tokens.push(token.devtoken);
                }

            }

        var payload = {

                "notification":{
                    "title":"LifeBuoy Alarm" ,
                    "body":"Device " + test.device,
                    "sound":"default",
                    "color":"#1B70A6"
                    }
        }
```

```
            return admin.messaging().sendToDevice(tokens,
payload).then((response) => {
                console.log('Pushed notifications');
            }).catch((err) => {
                console.log(err);
            })
        })
    })

    }
    else{
        console.log("No alarm");
    }

      return true;
    });

    function processtokens(rawtokens) {
    var promise = new Promise((resolve, reject) => {
         var processedtokens = []
    for (var token in rawtokens) {
        processedtokens.push(rawtokens[token]);
    }
    resolve(processedtokens);
    })
    return promise;

}
}
```

App code

Home.html

```
<ion-header>
   <ion-navbar>
       <ion-title>

           Map
       </ion-title>
   </ion-navbar>
</ion-header>
<ion-content>
   <div style="width:100%; height:90%" #map=""
id="map"></div></ion-content>
```

Home.module.ts

```
import { NgModule } from '@angular/core';
import { IonicPageModule } from 'ionic-angular';
import { HomePage } from './home';

@NgModule({
 declarations: [
   //HomePage,
 ],
 imports: [
   IonicPageModule.forChild(HomePage),
 ],
})
export class HomePageModule {}
```

Home.ts

```
import { Component, ViewChild, ElementRef, OnInit } from '@angular/core';
import { NavController } from 'ionic-angular';
import { RegisterPage } from '../register/register';
import { AngularFireAuth } from 'angularfire2/auth';
import { Geolocation } from '@ionic-native/geolocation';
import { User } from '../../model/user.model';
import { LoginPage } from '../login/login';
import { DeviceProvider } from '../../providers/device/device';
import { Platform } from 'ionic-angular';
import {GoogleMaps, GoogleMap, GoogleMapsEvent, GoogleMapOptions,
CameraPosition, MarkerOptions, Marker} from '@ionic-native/google-maps';
import { Observable } from 'rxjs/Observable';
import { Device } from '../../model/device.model';
import { TabsPage } from '../tabs/tabs';
import firebase from 'firebase';
import { AngularFireDatabase } from 'angularfire2/database';

declare var google;

declare var FCMPlugin;
var details;
var me$;
var markers = [];
var user;
var found = false;
var number = 0;;
var deviceEntries;
var map;

@Component({
 selector: 'home-page',
 templateUrl: 'home.html'
})
export class HomePage implements OnInit {
```

```
firestore = firebase.database().ref('/pushtokens');


nowUser : string;
user = {} as User;
deviceList$ : Observable<Device[]>;
me$ : string ;
cord: any;
lat: any;
long: any;
dID: any;
lats : string;
latlng : any;

 @ViewChild('map') mapElement: ElementRef;
map: any;
 constructor( public geolocation: Geolocation,
  public navCtrl: NavController,
  private afAuth : AngularFireAuth,
  private platform: Platform,
  public afd: AngularFireDatabase,
  private deviceProvider: DeviceProvider,


) {

  this.tokensetup().then((token) => {
    this.storetoken(token);
  })
}

ngOnInit(){
  this.afAuth.authState.subscribe((res)=>{

   this.me$ = this.afAuth.auth.currentUser.email;
      });
```

```
      this.loadMap();
  }

  ionViewWillEnter()
  {
    this.loadMap()
  }


ionViewDidLoad(){


  FCMPlugin.onNotification(function(data){
    if(data.wasTapped){
      //Notification was received on device tray and tapped by the user.
      alert( JSON.stringify(data) );
    }else{
      //Notification was received in foreground. Maybe the user needs to be
notified.
      alert( JSON.stringify(data) );
    }
    });

FCMPlugin.onTokenRefresh(function(token){
    alert( token );
});
  }

  tokensetup() {
    var promise = new Promise((resolve, reject) => {
      FCMPlugin.getToken(function(token){
    resolve(token);
      }, (err) => {
        reject(err);
});
```

```
    })
    return promise;
  }

  storetoken(t)
  {
    deviceEntries = firebase.database().ref('/pushtokens');

      deviceEntries.orderByChild("uid").on("value" ,function(data)
          {

            data.forEach(function(snap)
            {
              var result = snap.val();
              if(result.uid == firebase.auth().currentUser.uid)
              {
                found = true;
              }

            })
            runAfterQuery(found);

          })
  function runAfterQuery(f)
{
  if(f == false)
  {

    firebase.database().ref('pushtokens/' +
firebase.auth().currentUser.uid).set({
      uid:firebase.auth().currentUser.uid,
      devtoken: t,
      email: me$
    });

  }
  else{
```

```
firebase.database().ref("pushtokens/"+firebase.auth().currentUser.uid).upd
ate({ devtoken: t });

 }



}
}

  loadMap(){
   this.afAuth.authState.subscribe((res)=>{

     me$ = this.afAuth.auth.currentUser.email;
        });
     // array does not exist, is not an array, or is empty
     var array1 = [];
   this.platform.ready().then(()=>{
     this.geolocation.getCurrentPosition().then((resp) => {
       let unique_array = [];

     let latLng = new google.maps.LatLng(resp.coords.latitude,
resp.coords.longitude);
     let mapOptions =
     {
     center: latLng,
     zoom: 15,
     mapTypeId: google.maps.MapTypeId.ROADMAP
     }

     map = new google.maps.Map(this.mapElement.nativeElement, mapOptions);

     user = firebase.database().ref('/devices');

       //this.countryRef..subscribe(cord=>{
         user.orderByChild("device").on("value" ,function(data)
```

```
        {

        data.forEach(function(snap) {
          var key = snap.val();
          var lat = key.lat;
          var lng = key.lng;
          var dID = key.device;
          var alrm = key.alarm;

          if(key.email == me$)
          {
            details = {
              lat: lat,
              lng: lng,
              dID: dID,
              alrm: alrm
            };


            if(details.alrm === "00000000")
              {
                let latLng = new google.maps.LatLng(details.lat,
details.lng);
                 var marker = new google.maps.Marker({
                   position: latLng,
                   map: map,
                   icon:
'http://maps.google.com/mapfiles/ms/icons/green-dot.png'
                 });
                let content = `<h4>${details.dID}</h4>`;

                let infoWindow = new google.maps.InfoWindow({
                 content: content});
                 google.maps.event.addListener(marker, 'click', () => {
                 infoWindow.open(map, marker);
                 });
```

```
                marker.setMap(map);
              }
            else{
                let latLng = new google.maps.LatLng(details.lat,
details.lng);
                var marker = new google.maps.Marker({
                  position: latLng,
                  map: map,
                  icon:
'http://maps.google.com/mapfiles/ms/icons/red-dot.png'
                });

              let content = `<h4>${details.dID}</h4>`;
              let infoWindow = new google.maps.InfoWindow({
                content: content});
                google.maps.event.addListener(marker, 'click', () => {
                infoWindow.open(map, marker);
                });

              marker.setMap(map);
            }
            number++;
          }
        });

      });

        }

)
}
)

 }
```

```
  addInfoWindow(marker, content){
   let infoWindow = new google.maps.InfoWindow({
     content: content
   });

   google.maps.event.addListener(marker, 'click', () => {
     infoWindow.open(this.map, marker);
   });

 }
 }
```

DeviceList.html

```
<ion-header>

       <ion-navbar>
           <ion-title>Device Owned </ion-title>
       </ion-navbar>
       <ion-searchbar (ionInput)="getItems($event)"></ion-searchbar>

   </ion-header>
   <ion-content padding>
<ion-list>
       <ng-container *ngFor="let d of countryList">
               <ion-item detail-push navPush="EditDevicePage"
[navParams]="{d:d}" *ngIf="d.email ===  me$"> Device ID: {{d.device}}

               </ion-item>

           </ng-container>

</ion-list>
   </ion-content>
```

DeviceList.module.ts

```
import { NgModule } from '@angular/core';
import { IonicPageModule } from 'ionic-angular';
import { DeviceListPage } from './device-list';

@NgModule({
 declarations: [
   //DeviceListPage,
 ],
 imports: [
   IonicPageModule.forChild(DeviceListPage),
 ],
})
export class DeviceListPageModule {}
```

DeviceList.ts

```
import { Component, OnInit } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { Observable } from 'rxjs/observable';
import { Device } from '../../model/device.model';
import { DeviceProvider } from '../../providers/device/device';
import { AngularFireDatabase } from 'angularfire2/database';
import { AngularFireAuth } from 'angularfire2/auth';
import firebase from 'firebase';


@Component({
 selector: 'page-device-list',
 templateUrl: 'device-list.html',
})
export class DeviceListPage implements OnInit{
```

```
 deviceList$ : Observable<Device[]>;

 me$ : string ;//= '1garry8@gmail.com';

 public countryList:Array<any>;
public loadedCountryList:Array<any>;


 constructor(
   public navCtrl: NavController,
   public navParams: NavParams,
   private deviceProvider: DeviceProvider,
   private afDatabase : AngularFireDatabase,
   private afAuth : AngularFireAuth
 ) {
   this.initializeItems();
 }
  initializeItems(){
   this.countryList = this.loadedCountryList;
 }


 getItems(ev: any) {
   // Reset items back to all of the items
   this.initializeItems();

   // set q to the value of the searchbar
   var q = ev.target.value;
   console.log("q", q);
    // if the value is an empty string don't filter the items
   if (!q) {
     return;
   }
    this.countryList = this.countryList.filter((v) => {
     if(v.device && q) {
       if (v.device.toLowerCase().indexOf(q.toLowerCase()) > -1) {
         return true;
```

```
        }
        return false;
      }
    });
     console.log(q, this.countryList.length);
  }

ngOnInit(){
 this.deviceList$ = this.deviceProvider
 .getDevice()
 .snapshotChanges()
 .map(
   changes => {
     return changes.map(c=>({
       key:c.payload.key,
       ...c.payload.val(),
     }));
   });


   this.deviceList$.subscribe(cord=>{
     let countries = [];
     cord.forEach(function(snap) {


       countries.push(snap);
       return false;




     });
     console.log(countries);
     this.countryList = countries;
     this.loadedCountryList = countries;
   });
```

```
    //gets the current user email
    this.me$ = this.afAuth.auth.currentUser.email;
    console.log(this.me$);
    console.log(this.deviceList$);

}
}
```

Editdevice.html

```
<ion-header>

    <ion-navbar>
        <ion-title>Device Info Page</ion-title>
    </ion-navbar>

</ion-header>


<ion-content padding>
    <ion-card>
        Your current Latitude : {{currentLat}}
        <br>Your current Longtitude : {{currentLong}}

        <br> Device ID : {{device.device}}
        <br> Device Latitude : {{device.lat}}
        <br> Device Longtitude :{{device.lng}}
        <br> Alarm :{{device.alarm}}

        <br>

        <button ion-button outline item-end icon-left
(click)="update()">Update Position</button>
        <button ion-button outline item-end icon-left
(click)="resetAlarm()">Reset Alarm</button>
```

```
    </ion-card>
</ion-content>
```

Edit.module.ts

```
import { NgModule } from '@angular/core';
import { IonicPageModule } from 'ionic-angular';
import { EditDevicePage } from './edit-device';

@NgModule({
 declarations: [
   EditDevicePage,
 ],
 imports: [
   IonicPageModule.forChild(EditDevicePage),
 ],
})
export class EditDevicePageModule {}
```

Editdevice.ts

```
import { Component, OnInit } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';
import { Device } from '../../model/device.model';
import { DeviceProvider } from '../../providers/device/device';
import { Geolocation } from '@ionic-native/geolocation';
import { HomePage } from '../home/home';
import { TabsPage } from '../tabs/tabs';

@IonicPage()
@Component({
 selector: 'page-edit-device',
 templateUrl: 'edit-device.html',
})
```

```
export class EditDevicePage implements OnInit {

 device: Device;

 // these need to be set to the current gps cords like before
 currentLat : any;
 currentLong : any;

 constructor(
   public navCtrl: NavController,
   public navParams: NavParams,
   private deviceProvider: DeviceProvider,
   private geolocation: Geolocation,
 ) { }

 ionViewWillLoad() {
  this.device = this.navParams.get('d');
 }

 ngOnInit() {

   this.geolocation.getCurrentPosition().then((resp) => {
     resp.coords.latitude
     resp.coords.longitude
   }).catch((error) => {
     console.log('Error getting location', error);
   });

   let watch = this.geolocation.watchPosition();
   watch.subscribe((data) => {
    // data can be a set of coordinates, or an error (if an error
occurred).
     //data.coords.latitude
     //sets the value of the update variables
  this.currentLat = data.coords.latitude;
  this.currentLong = data.coords.longitude;
  console.log(this.currentLong);
```

```
    });
  }

  update() {

    // this is just setting them to change
    this.device.lat = this.currentLat;
    this.device.lng = this.currentLong;

    //function in th provider
    //updating this instance of device
    this.deviceProvider.updateDevice(this.device).then(()=>{
      //put in a toast message here to let them know it was updated
    }).then(()=>{
      //navigate back to the map maybe#
      //this.navCtrl.setRoot(this.navCtrl.getActive().component);
    })
  }

   resetAlarm()
  {
    this.device.alarm = "00000000";
    //function in th provider
    //updating this instance of device
    this.deviceProvider.updateDevice(this.device).then(()=>{
      //put in a toast message here to let them know it was updated
    }).then(()=>{
      //navigate back to the map maybe
    })
  }

}
```

Login.html

```
<ion-content padding>
   <ion-card-header>
       LOGIN
   </ion-card-header>
   <ion-card-content>
       <ion-list no-lines>
           <ion-item>
               <ion-label floating>Email</ion-label>
               <ion-input type="email"
[(ngModel)]="user.email"></ion-input>
           </ion-item>
           <ion-item>
               <ion-label floating>Password</ion-label>
               <ion-input type="password"
[(ngModel)]="user.password"></ion-input>
           </ion-item>
           </ion-list>
           </ion-card-content>

           <button ion-button full
(click)="loginUser(user)">Login</button>

   <button ion-button full (click)="registerPage()">Register</button>


</ion-content>
```

Login.module.ts

```
import { NgModule } from '@angular/core';
import { IonicPageModule } from 'ionic-angular';
import { LoginPage } from './login';

@NgModule({
 declarations: [
   //LoginPage,
 ],
 imports: [
   IonicPageModule.forChild(LoginPage),
 ],
})
export class LoginPageModule {}
```

Login.ts

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, ToastController } from
'ionic-angular';
import { RegisterPage } from '../register/register';
import { User } from '../../model/user.model';
import { AngularFireAuth } from 'angularfire2/auth';
import { TabsPage } from '../tabs/tabs';

@IonicPage()
@Component({
 selector: 'page-login',
 templateUrl: 'login.html',
})
export class LoginPage {

  user = {} as User;
 errorMessage;
 loggedIn;
```

```
  constructor(public navCtrl: NavController, public navParams: NavParams,
private afAuth:AngularFireAuth, public toastCtrl:ToastController ) {
 }
 ngOnInit() {
   this.loggedIn = this.afAuth.authState.subscribe((data)=>{
     if(data) {
       // User is signed in.
       console.log('is user');
         this.navCtrl.setRoot(TabsPage);

     }else{
       console.log('is not user')
       // No user is signed in.
     }
    });
  }


 //remove this
 ionViewDidLoad() {
   console.log('ionViewDidLoad LoginPage');
 }

 //this can be removed and use a nav . push in th html if you want
 registerPage() {
   this.navCtrl.push(RegisterPage);
 }


 //fix to the login
 loginUser(user: User) {
   this.afAuth.auth.signInWithEmailAndPassword(user.email.valueOf(),
user.password.valueOf())
   .then( () => {
     console.log( 'Success');
     console.log('Found You');
```

```
        this.navCtrl.setRoot(TabsPage);

      //success
    }, (err) => {
      // Do something with error
      console.log(err.message, 'failed');
      console.log('Nobodys here');
      let toast = this.toastCtrl.create({
        message: `Woops!!!!!!!!!!!
        Something has gone wrong, Try entering the information again?`,
        duration: 6000,
        position: 'bottom',
        showCloseButton: true,
        dismissOnPageChange: true,
      });
      toast.present();
    })
  }

}
```

Register.html

```
<ion-header>

    <ion-navbar>
        <ion-title>Register</ion-title>
    </ion-navbar>

</ion-header>


<ion-content padding>
    <ion-item>
        <ion-label color="primary" floating>Email Address</ion-label>
        <ion-input clearInput type="email" [(ngModel)]="user.email">
</ion-input>
    </ion-item>

    <ion-item>
        <ion-label color="primary" floating>Password</ion-label>
        <ion-input clearInput type="password"
[(ngModel)]="user.password"></ion-input>
    </ion-item>

    <button ion-button full (click)="registerUser(User)">Register
Account</button>

</ion-content>
```

Register.ts

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams } from 'ionic-angular';

import { User } from '../../model/user.model';
import { AngularFireAuth } from 'angularfire2/auth';
import { TabsPage } from '../tabs/tabs';

@IonicPage()
@Component({
 selector: 'page-register',
 templateUrl: 'register.html',
})
export class RegisterPage {

 user = {} as User;

 constructor(
   public navCtrl: NavController,
   public navParams: NavParams,
   private afAuth : AngularFireAuth,
 ) { }

 ionViewDidLoad() {
   console.log('ionViewDidLoad RegisterPage');
 }
   registerUser(){

   this.afAuth.auth.createUserWithEmailAndPassword(this.user.email,
this.user.password)
   .then(()=> this.navCtrl.setRoot(TabsPage));

  }
```

```
}
```