National College of Ireland

BSHc in Computing

2017/2018

Sarah Yun Tiong

x14111187

x14111187@student.ncirl.ie

Supervisor: Thibaut Lust, Eamon Nolan

GPS Safety Tracking Device

Technical Report

Table of Contents

Executive Summary 5

1 Introduction 6

1.1 Background 6

1.2 Aims 7

1.3 Technologies 7

1.4 Structure 8

2 System 9

2.1 Requirements 9

2.1.1 Functional requirements 9

2.1.2 Data requirements 17

2.1.3 User requirements 17

2.1.4 Environmental requirements 17

2.1.5 Usability requirements 18

2.1.6 Performance/Response time requirement 18

2.1.7 Availability requirement 18

2.1.8 Recover requirement 18

2.1.9 Robustness requirement 18

2.1.10 Security requirement 18

2.1.11 Reliability requirement 18

2.1.12 Portability requirement 18

2.1.13 Extendibility requirement 19

2.1.14 Reusability requirement 19

2.1.15 Resource utilization requirement 19

2.2 Design and Architecture 19

2.2.1 Class diagram 19

2.2.2 Sequence Diagram 20

2.3 Implementation 21

2.3.1 Verify the device by checking QR Code 21

2.3.2 Subscribe to MQTT's topic 21

2.3.3 Real-time Location Map View 22

2.3.4 Compute distance between current location & safety location 22
2.3.5 Send on/off order to Firebase if device exceeds safety zone 22
2.4 Graphical User Interface (GUI) Layout 23
2.5 Testing 25
2.5.1 Unit Testing 25
2.5.2 Survey 26
2.6 Evaluation 28
2.6.1 Evaluate with the system 28
2.6.2 Evaluate with an end-user 28
3 Conclusions 29
3.1 Advantages 29
3.2 Disadvantages 29
3.3 Opportunities 29
3.4 Limits 29
4 Further development or research 30
4.1 Advance portability 30
4.2 Adjunction of the GSM service 30
5 References 31
6 Appendix 32
6.1 Project Plan 32

Declaration Cover Sheet for Project Submission

SECTION 1

Name: • **Sarah Yun Tiong**
Student ID: • **x14111187**
Supervisor: • **Thibaut Lust** • **Eamon Nolan**

SECTION 2 Confirmation of Authorship

The acceptance of your work is subject to your signature on the following declaration:

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature:_____Sarah Yun Tiong_____ Date:_____11st May 2018____

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

Complete the sections above and attach it to the front of one of the copies of your assignment.

Executive Summary

The scope of this particular project is to develop a device which will allow real time location tracking through a mobile application using GPS technology. Its aim is to make people's life easier through achieving the 'attach-and-locate' any belongings in seconds.

The location tracking is achieved by obtaining the satellites' data using GPS technology. This is accomplished by a connection of a RaspberryPi and a GPS Module.

The corresponding mobile app will constantly update the registered devices' current location, distance between the device and the defined safety-zone, as well as the alarm status; It not only allows user to define custom 'safety zone' and trigger alarm when the devices enter/leave the area; and also allows remote LED/buzzer controlling through the mobile application for a safety purpose.

1 Introduction

1.1 Background

History

Boston Globe technology writer Hiawatha Bray recalls the moment that inspired him to write his new book, "*You Are Here: From the Compass to GPS, the History and Future of How We Find Ourselves*" (Brownell, 2014). During the 19th century, only landline callers' location was allowed to accurately track by the emergency responders such as 911 although the cellphone usage was increasing significantly at that time. FCC then determine that it's also necessary to provide emergency responders an accurate location of all cellphone users too when they were calling 911 due to a safety manner.

"We were about to enter a world in which…everybody had a cellphone, and that would also mean that we would know where everybody was." (Brownell, 2014). Human beings evolved from primitive navigation tools such as compass, to high-technical instant digital mapping achieved by GPS tracking. GPS tracking is the surveillance of location through use of the Global Positioning System (GPS). This satellite-based navigation system which made up of at least 24 satellites can accurately pinpoint longitude, latitude, ground speed, and course direction of an entity (Rouse, 2014).

Of course individual whereabouts tracking is covered by privacy protection but now widely exposed to public usage in the 21st century, not limited to web browsing, social media or a new innovated personal safety-tracking purpose.

Tracking device

In generally speaking, a tracking device is a system capable of observing coordinates of an object on its move and supply a timely ordered sequence of location data for further processing (Y. Ali and Beram Jasim, 2015).

Nowadays tracking devices are widely developed for expensive vulnerable assets which do not have their own source of power but somehow need to be tracked. This small-size device comes in several forms, helps to easily locate and secure your vehicles, luggage, laptops, employees, children, elderlies, or even pets. The beauty of this new technology is that you will be able to access to every movement of your equipment by simply attach the device on them. Likewise you can login to the corresponding software application and watch your tracker or trackers in real-time or, even the route history. Any of your family or friends is allowed to download the application, and if you share the login details they too can watch your tracker location on their phone or desktop, provides supreme protection.

1.2 Aims

The objective of this project is to make people's life easier through achieving the 'attach-and-locate' any belongings in seconds.

This device can be simply placed in your bag, luggage, vehicles and etc. Ensure you know where your car or fleet of vans are at all times, and also giving parents the peace of mind by keep track on their children's day on a school trip or at an amusement park without being too instructive.

Using your Smartphone you will always know where your loved ones are. Or, conversely you are also able to alert the particular device if needed.

Features of the device include:

- **Real-time tracking**
  The system shall allow real time location tracking and provides detailed information including the longitude & latitude which could be viewed on a map.

- **Define safety zone**
  The system shall allow to track living creatures such as human beings and pets in a safety manner. The master will be able to define a safety area which will send alerts when the particular STrack moves or goes outside a predefined area/zone.

- **Remote alarming**
  The system shall allow remote LED & buzzer controlling so the master is able to alert surroundings of STrack and enhance the ease of STrack to be found in dark. This function is controlled by the master's mobile application.

1.3 Technologies

Raspbian

Raspbian is a Debian-based official operating system provided by RaspberryPi. It comes with plenty of software to support RaspberryPi's programming.

Cloud9

Cloud9 is a cloud-based integrated development environment (IDE) that allows to write, run, and debug the code within a browser. This is used to implement the code to subscribe to the Raspberrypi and publish the received data to specific AWS MQTT's topic, to be used later for the Android Application.

Android Studio

The corresponding Mobile App for STrack is going to be Android-based. The development of the application will be done in the globally platform provided by Google.

Python

Python is an easy-to-use, clear and powerful functional programming language. The first part of the project focused on extracting the GPS module's data from the RaspberryPi, which requires the capability to process data swiftly in an environment with minimal resource requirements. Its readability without those mysterious syntax makes it a great choice for development.

Java

Java as the most-popular platform independent programming language, is widely used in Android application development. Its object-oriented attribute suits the design of the program.

Adafruit Ultimate GPS Breakout

Built around the MTK3339 chipset, Adafruit shop has come out with a no-nonsense, low-power while high-quality GPS module that can track up to 22 satellite on 66 channels, along with a built in antenna and high-sensitivity receiver. (Adafruit.com, 2018)

1.4 Structure

In section 2, the System's requirements as well as its design will be described. The GUI of the mobile application, unit-testing as well as the result of the survey will be included as well.

In section 3, the conclusion of the report is illustrated, along with the advantage, disadvantage and concerns of the product.

In section 4, the future development of the project is described.

Section 5 describes the references used in this report.

In section 6, all the appendix is described.

2 System

2.1 Requirements

2.1.1 Functional requirements

STrack (RaspberryPi):

- Allows GPS tracking.
- Switch on/off LED/buzzer.
- Display latest location information on the LCD.

STrack's Mobile application:

- Register a STrack device.
- Update the registered STrack's alias.
- Delete a registered device.
- View real time information of the registered STrack. (E.g. Latitude & Longitude, distance)
- Connect to MQTT for a selected STrack.
- Map view for a selected STrack.
- Define safety zone for a selected STrack.
- Remote LED/Buzzer control for a selected STrack.

2.1.1.1 Use Case Diagram

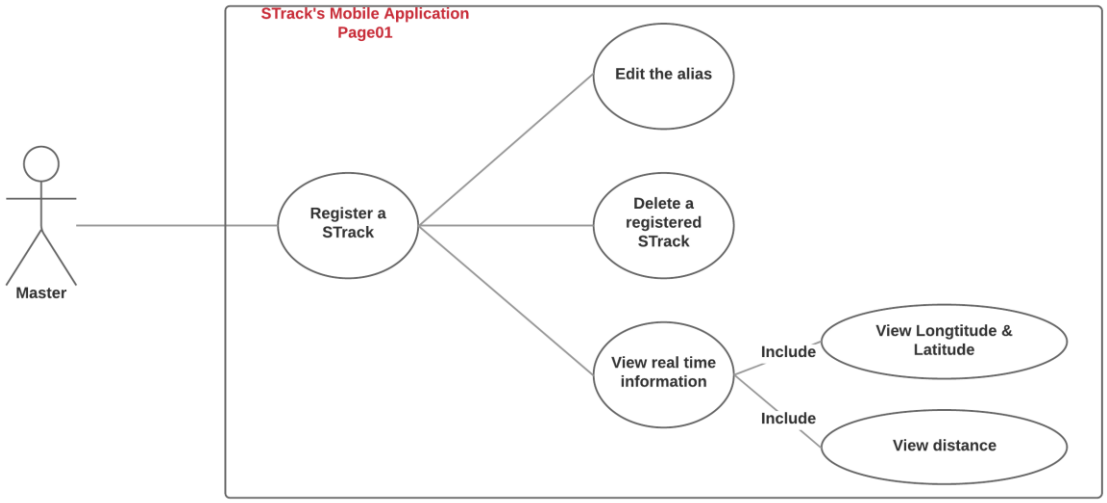
The following use case diagrams demonstrate the features and functions provided by STrack and STrack's mobile application:

STrack:

Presents functionalities such as GPS tracking, switch on/off its LED or buzzer, and display the latest location information on the LCD.

STrack's mobile application:

Including the device's registration, update and delete operation, display real time information, map-view, define safety-zone, and remote LED/Buzzer controlling.



2.1.1.1.1 Requirement 1 <Register a device>

Description & Priority

Upon opening the STrack mobile application, this will be the main screen shown to users. It allows them to register a device. The users will then be taken to differing pages reflecting this decision. The user won't be able to access any device if none is registered.

Use Case

- **Scope**

  The scope of this use case is for the user to register a device.

- **Description**

  This use case describes how the user register a device.

- **Flow Description**

  Precondition

  The User had entered the mobile application.

  Activation

  This use case starts when a User enter the application.

Main flow

1. The System provides a float button at the bottom-right corner.
2. The User presses the button.
3. The System navigates the User to the 'Scan QR Code' Page.

4. The User scans the QR Code which comes with the device purchased.
5. The System identify the QR Code. (See E1 – Invalid QR Code)
6. The System approved the QR Code provided.
7. The User presses the 'Confirm' button.
8. The System navigates the User to the 'Enter Device Details' Page.
9. The User enters the required fields.
10. The User presses the 'Save' button.
11. The System saves the information to the database and list-view.
12. The System navigates the User back to the main screen.

Alternate flow

-

Exceptional flow

E1: < Invalid QR Code >

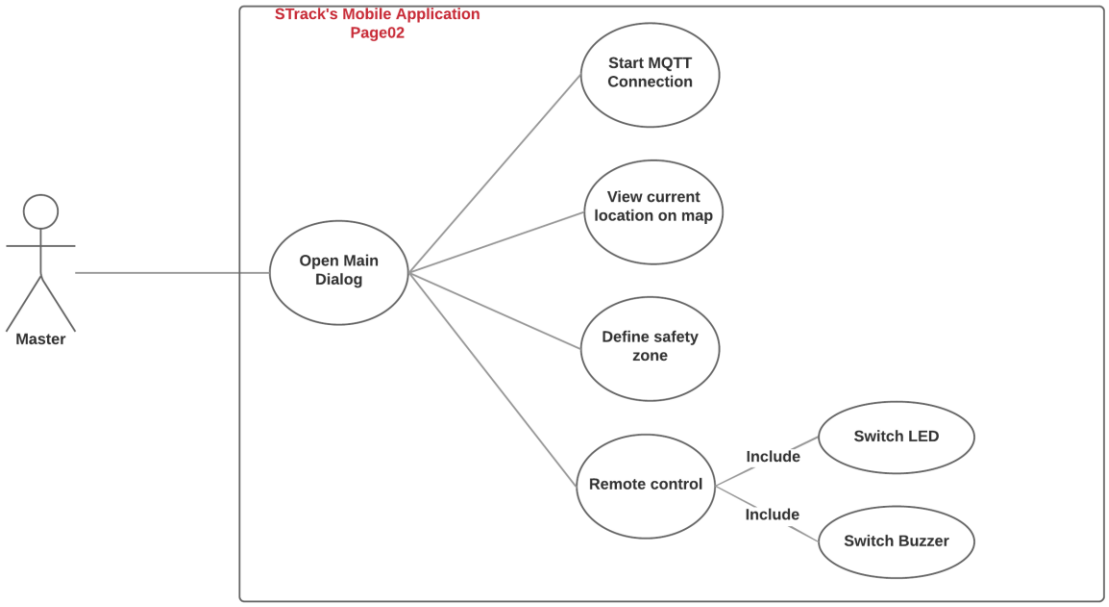
1. The User had scanned an invalid QR Code.
2. The System navigate back to the previous screen.
3. The use case continues at position 1 of the main flow.

Termination

The System navigates the user back to the main screen once saved.

Post condition

The system goes into a wait state.



2.1.1.1.2 Requirement 2 <Open the main dialog>

Description & Priority

Once one or more devices are registered to the account, all the information of the devices will be shown in the list-view of the main screen. In order to access further features regarding a particular device, the user could press any list-item which represents a particular device.

Once a list-item is pressed, a pop-up-dialog (Main Dialog) will show up. The user is able to navigate to others activities regarding the particular device through pressing the buttons on the dialog.

Use Case

- **Scope**

  The scope of this use case is for the user to open a list item's Main Dialog.

- **Description**

  This use case describes how the User opens a list item's Main Dialog.

- **Flow Description**

  Precondition

  The User had successfully registered one or more devices.

  Activation

  This use case starts when a User pressed on a list item.

Main flow

1. The System provides a list view for registered devices.
2. The User pressed a list item.
3. The System opens the Main Dialog.
4. The User pressed 'Click to Connect'. (See A1 – Click to Connect)
5. The User pressed 'Map View'. (See A2 – Map View)
6. The User pressed 'Define Safety Zone'. (See A3 – Define Safety Zone)
7. The User pressed 'Remote Controls'. (See A4 – Remote Controls)
8. The User pressed 'Exit Button'. (See A5 – Exit Button)

Alternate flow

A1: < Click to Connect >

1. The System navigates the User to the page 'Click to connect'.

A2: < Map View >

2. The System navigates the User to the page 'Map View'.

A3: < Define Safety Zone >

3. The System navigates the User to the page 'Define Safety Zone'.

A4: < Remote Controls >

4. The System navigates the User to the page ‘Remote Controls’.

A5: < Exit Button >

5. The System navigates the User to the page ‘Exit button’.

Exceptional flow

-

Termination

The User pressed any button on the Main Dialog.

Post condition

The system goes into a wait state.

2.1.1.1.3 Requirement 3 <Start MQTT Connection>

Description & Priority

Once the Main Dialog is opened, there are several options available for the User. When the User pressed the ‘Click to Connect’ button, the User will enters the connection page.

After the ‘Status’ changed from ‘Disconnect’ to ‘Connected’, the User could press the subscribe button to start listening to the device’s real-time information.

Use Case

- **Scope**

The scope of this use case is for the user to start the MQTT connection.

- **Description**

This use case describes how the user start the MQTT connection.

Flow Description

Precondition

The User had opened the Main Dialog.

Activation

This use case starts when a User selects the ‘Click to Connect’ button in the Main Dialog.

Main flow

1. cc
2. The System changes the ‘Status’ from ‘disconnected’ to ‘connected’.
3. The User presses the ‘Subscribe’ button.
4. The System starts to subscribe to the MQTT message.

Alternate flow

-

Exceptional flow

-

Termination

The User pressed the back button at the bottom-right corner.

Post condition

The system goes into a wait state.

2.1.1.1.4 Requirement 4 <View device current location on Map>

Description & Priority

Once the Main Dialog is opened, there are several options available for the User. When the User pressed the 'Map View' button, the User will enters the map view page, where the User is able to view the device's latest location via Google Map.

Use Case

- **Scope**

The scope of this use case is for the user to view the latest location of a device in Google Map.

- **Description**

This use case describes how the user check the latest location of a device in a map.

Flow Description

Precondition

The User had started the MQTT Connection.

Activation

This use case starts when a User selects the 'Map View' button in the Main Dialog.

Main flow

5. The User selects the 'Map View' button in the Main Dialog.
6. The System display the device's latest location via Google Map in real-time.

Alternate flow

-

Exceptional flow

-

Termination

The User pressed the back button at the bottom-right corner.

Post condition

The system goes into a wait state.

2.1.1.1.5 Requirement 5 <Define safety-zone>

Description & Priority

Once the Main Dialog is opened, there are several options available for the User. When the User pressed the 'Define Safety Zone' button, the User will enters the safety zone page, where the User is able to select a safety-zone via Google Map.

Use Case

- **Scope**

The scope of this use case is for the user to define a safety-zone.

- **Description**

This use case describes how the User defines a safety-zone.

Flow Description

Precondition

The User had a registered device.

Activation

This use case starts when a User selects the 'Define Safety Zone' button in the Main Dialog.

Main flow

1. The User selects the 'Define Safety Zone' button in the Main Dialog.
2. The System display a Google Map.
3. The User selects a 'Safety-zone' on the Map.
4. The System push the Safety-zone data to Firebase.
5. The System navigates back to the main screen.

Alternate flow

-

Exceptional flow

-

Termination

The User goes back to the main screen.

Post condition

The system goes into a wait state.

2.1.1.1.6 Requirement 6 <Remote LED/Buzzer control>

Description & Priority

Once the Main Dialog is opened, there are several options available for the User. When the User pressed the 'Remote Controls' button, the User will enters the remote control page, where the User is able to switch on/off the LED/Buzzer.

Use Case

- **Scope**

The scope of this use case is for the user to control the LED/Buzzer of a device.

- **Description**

This use case describes how the user controls the LED/Buzzer of a device.

Flow Description

Precondition

The User had a registered device.

Activation

This use case starts when a User selects the 'Remote Controls' button in the Main Dialog.

Main flow

1. The User selects the 'Remote Controls' button in the Main Dialog.
2. The User turn-on the toggle button of the sound/light.
3. The System changes the Boolean value of the sound/light from '0' to '1'.
4. The System notifies the change to the device.
5. The Device changes the Boolean value of the sound/light from '0' to '1'.
6. The sound/light on the device turned on.

Alternate flow

-

Exceptional flow

-

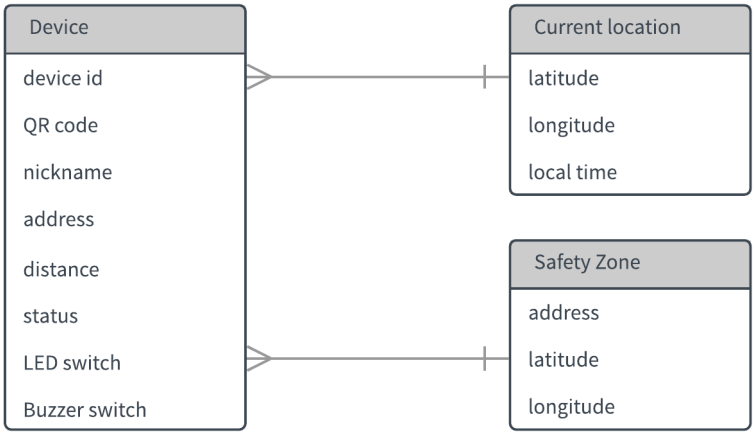
Termination

The User turn-off the toggle button.

Post condition

The system goes into a wait state.

2.1.2 Data requirements



2.1.3 User requirements

People nowadays worries and seek for an easy, low-cost solution for their expensive vulnerable assets that need tracking but do not have their own source of power. This objective could be achieve by allowing the users to attach a small, portable tracking device to anything and manage multiple assets' real-time location in one single mobile application.

- **Real-time tracking**
  The system shall allow real time location tracking and provides detailed information including the longitude & latitude which could be viewed on a map.

- **Define safety zone**
  The system shall allow to track living creatures such as human beings and pets in a safety manner. The master will be able to define a safety area which will send alerts when the particular Strack moves or goes outside a predefined area/zone.

- **Remote alarming**
  The system shall allow remote LED & buzzer controlling so the master is able to alert surroundings of STrack and enhance the ease of STrack to be found in dark. This function is controlled by the master's mobile application.

2.1.4 Environmental requirements

A Wi-Fi-enable smartphone is needed in order to access STrack's Mobile Application.

2.1.5 Usability requirements

The system shall be friendly for the end-user with easy-to-learn interface, which should be achieved by a concise User Interface.

2.1.6 Performance/Response time requirement

The response time of STrack plays a vital role as a requirement. It's very important for both the master to receive real-time update on his/her mobile application and the device to send the alert immediately. Response time for the alarm system should not exceeds 5 seconds. The alert and real-time update could be affected by the Wi-Fi signal, which should be aware.

2.1.7 Availability requirement

STrack as an electronic device, is run by battery, which means it'll stop running once the battery is empty. It is essential to choose a long-lasting battery for it to last longer. The corresponding mobile application will displays the last recorded information and update once the system's back.

On the other hand, the mobile app must run with Wi-Fi.

2.1.8 Recover requirement

All the data received will be hosted on a cloud database – Firebase with a unique id, which means if the device's corrupted in some unexpected circumstances, all the past data could still be recovered.

2.1.9 Robustness requirement

As a portable device which could be carried by children or pets STrack is required to be firm, water-proof and temperature-proof.

2.1.10 Security requirement

Every devices comes with a unique QR Code that the user could only register the device by scanning the right code to access the information provided by the registered device.

2.1.11 Reliability requirement

As a location tracking device it is essential for STrack to provide very-accurate location information on Google Map. It is expected for the user to not to obtain wrong information.

2.1.12 Portability requirement

Portability is considered as the most important requirement regards to STrack. It is essential for the user to feel comfortable and ease to carry STrack with them all the time, and the master to be able to access the corresponding software application in anywhere, anytime, therefor designed as a mobile application.

2.1.13 Extendibility requirement

As a new-talent device within the IOT industry, STrack is also having possibility to be improved. The most essential micro-controller needed in this device is continuing to be developed smaller. The features of the alarming system on the device, as well as the controls from the mobile application could always be modified.

2.1.14 Reusability requirement

Reusability is another main focus of STrack, as there'll be a chance that several masters might want to access the same information provided by one particular device. E.g. the mum & dad.

As a daily-use device it is essential for STrack to be friendly to use.

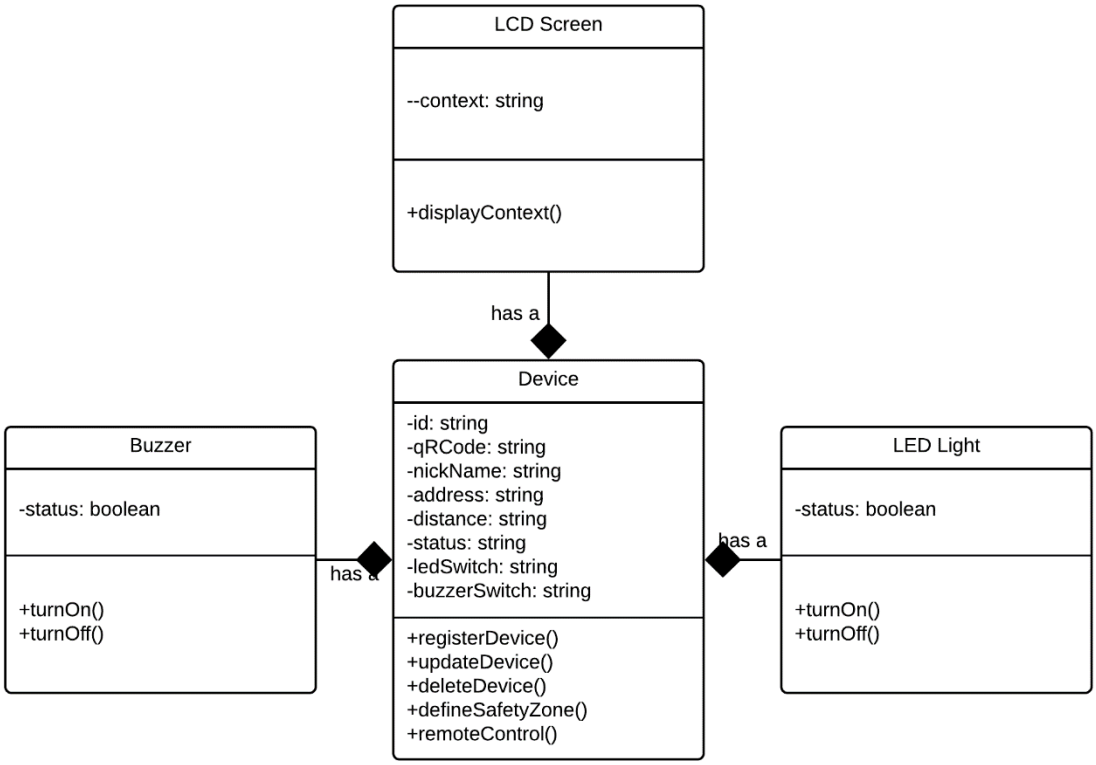
2.1.15 Resource utilization requirement

In order to access all the functionalities provided by STrack user is required to have a Wi-Fi enabled mobile and Google Map's permission.

2.2 Design and Architecture

2.2.1 Class diagram

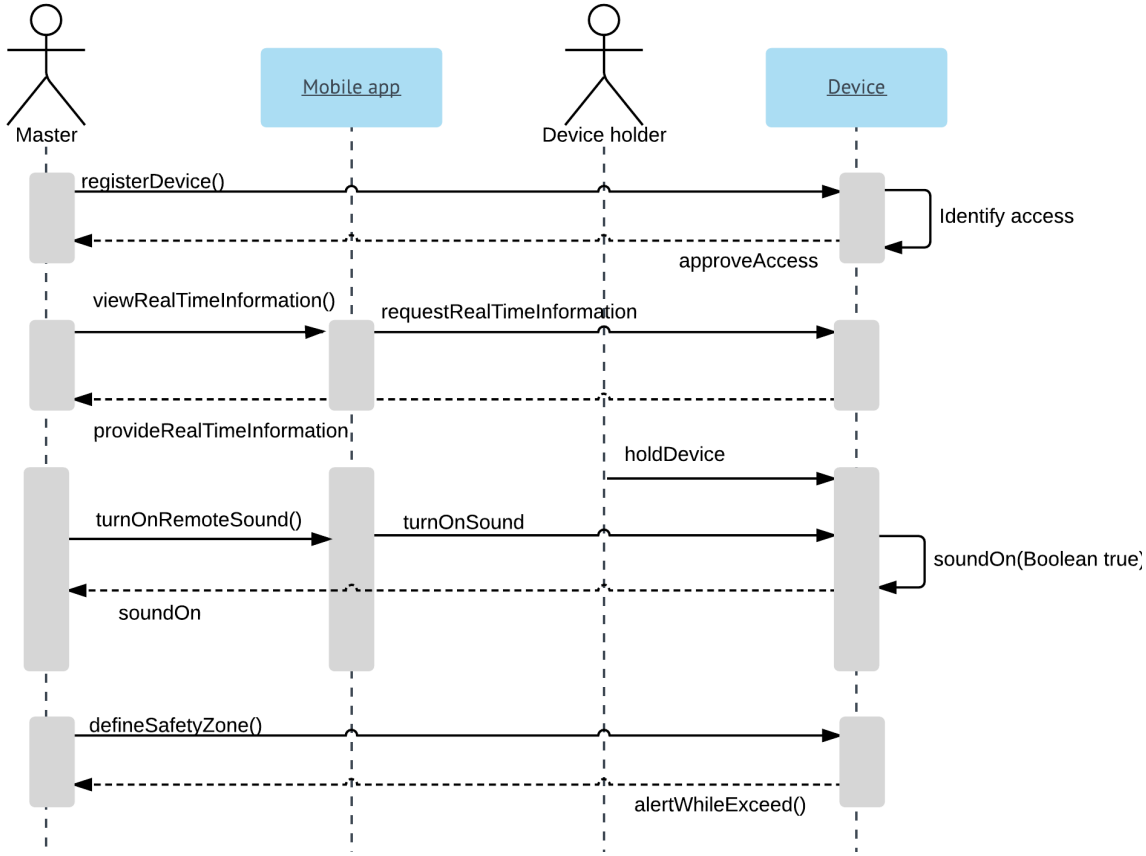
The following class diagram details a high level view of the relationships within the STrack system. A device has a LED light, buzzer, and a LCD screen.



2.2.2 Sequence Diagram

The following sequence diagram outlines the interaction a MasterUser and a DeviceHolder has with the system. The diagram presumes that a MasterUSer has purchased a device prior to the flow beginning. The flow starts when a MasterUser registers a device, where the device will authenticate and approve the request.

A DeviceHolder has the device attached. Now the MasterUser will view the information regarding the device on a mobile app. A MasterUser is allowed to remote controls the LED and Buzzer of the particular device, and also define a safety zone so that the device will send alert when it exceeds the area.



2.3 Implementation

2.3.1 Verify the device by checking QR Code

```
//Function - Check if the qrCode is valid
private void checkQRCode(String qrCode){
    Boolean valid = false;

    //For every list item:
    for(int i=0; i<qrCodeList.size(); i++){
        Log.d(TAG,  msg: "Checking: '"+qrCode +"', check with '"+qrCodeList.get(i)+"'.");
        //Process to register if valid
        if(qrCode.equals(qrCodeList.get(i))){
            valid = true;
            break;
        }else if(!qrCode.equals(qrCodeList.get(i))){ //Stop if not valid.
            valid = false;
        }
    }

    if(valid==true){
        Toast.makeText( context: this,  text: "Code valid.", Toast.LENGTH_LONG).show();
    }else if(valid==false){
        Toast.makeText( context: this,  text: "Code not valid.", Toast.LENGTH_LONG).show();
        finish();
    }
}
```

After the user success to scan a QR Code on the registration, the App will check and verify if the QR Code is valid or not, and results in successful registration or registration denying.

2.3.2 Subscribe to MQTT's topic

```
//On Subscribe click
View.OnClickListener subscribeClick = (v) -> {

        //Subscribe to the specific MQTT topic
        final String topic = "pi/observations/DeviceID";

        try {
            mqttManager.subscribeToTopic(topic, AWSIotMqttQos.QOS0,
                    (topic, data) -> {
                        runOnUiThread(() -> {
                            try {
                                mqttMessage = new String(data,  charsetName: "UTF-8");

                                //Extract the specefic keys from the json object retrieved from MQTT.
                                try {
                                    JSONObject reader = new JSONObject(mqttMessage);
                                    JSONObject gps_data  = reader.getJSONObject("gps_data");
                                    latitude = gps_data.getString( name: "Latitude");
                                    longtitude = gps_data.getString( name: "Longtitude");
                                    utc_time = gps_data.getString( name: "UTC Time");

                                    //Update the particular device's current location
                                    myRef = FirebaseDatabase.getInstance().getReference( s: "Device/" +deviceId+ "/Current location/Latitude");
                                    myRef.setValue(latitude);
                                    myRef = FirebaseDatabase.getInstance().getReference( s: "Device/" +deviceId+ "/Current location/Longitude");
                                    myRef.setValue(longtitude);
                                    myRef = FirebaseDatabase.getInstance().getReference( s: "Device/" +deviceId+ "/Current location/Local Time");
                                    myRef.setValue(utc_time);
                                    //Update the particular device's address
                                    myRef = FirebaseDatabase.getInstance().getReference( s: "Device/" +deviceId+ "/address");
                                    myRef.setValue(latitude+ ", " +longtitude);

                                } catch (JSONException e) {
                                    e.printStackTrace();
                                }

                                tvLastMessage.setText("Updated: " +latitude+ " " +longtitude+ " " +utc_time);

                            } catch (UnsupportedEncodingException e) {
                                Log.e(LOG_TAG,  msg: "Message encoding error.", e);
                            }
                        });
                    });
        } catch (Exception e) {
            Log.e(LOG_TAG,  msg: "Subscription error.", e);
        }
};
```

The real-time location information is sent to the specific AWS's topic from Cloud9. Once the user click the subscribe button on the Android Application, the App will subscribe to the topic, fetch the data and store them to the relating device's table in Firebase.

2.3.3 Real-time Location Map View

```
@Override
public void onChildChanged(DataSnapshot dataSnapshot, String s) {
    //Check if the safety_zone's latitude & longitude is updated before processing:
    if (dataSnapshot.child("Current location/Latitude").exists() && dataSnapshot.child("Current location/Longitude").exists()) {
        current_latitude = Double.parseDouble(dataSnapshot.child("Current location/Latitude").getValue(String.class));
        current_longitude = Double.parseDouble(dataSnapshot.child("Current location/Longitude").getValue(String.class));

        //Update the marker's location
        LatLng device_updated_location = new LatLng(current_latitude, current_longitude);
        myMarker.setPosition(device_updated_location);
        mMap.moveCamera(CameraUpdateFactory.newLatLng(device_updated_location));
        mMap.animateCamera(CameraUpdateFactory.zoomTo( v: 17.0f));
    }
}
```

Once the App perceived new location information updated from Firebase, it'll fetch the latest latitude & longitude and reset the marker's location on the Map to provide real-time marking on Google Map.

2.3.4 Compute distance between current location & safety location

```
//Function - Compute and update the latest distance.
private void computeLatestDistance(DataSnapshot dataSnapshot){
    //Check if the safety_zone is defined already before processing:
    if (dataSnapshot.child("SafetyZone").exists()) {

        //Grab the current location
        double current_latitude = Double.parseDouble(dataSnapshot.child("Current location/Latitude").getValue(String.class));
        double current_longitude = Double.parseDouble(dataSnapshot.child("Current location/Longitude").getValue(String.class));

        //Check if the safety_zone's latitude & longitude is updated before processing:
        if (dataSnapshot.child("SafetyZone/s_latitude").exists()) {
            safety_latitude = dataSnapshot.child("SafetyZone/s_latitude").getValue(Double.class);
        }
        if(dataSnapshot.child("SafetyZone/s_longitude").exists()) {
            safety_longitude = dataSnapshot.child("SafetyZone/s_longitude").getValue(Double.class);
        }

        //Compute the latest distance
        float[] new_d = new float[3];
        Location.distanceBetween(current_latitude, current_longitude, safety_latitude, safety_longitude, new_d);

        //Push the distance data to the Firebase AGAIN.* //("Device/" +deviceId+ "/distance");
        dataSnapshot.child("distance").getRef().setValue(new_d[0] + "m");

        //Trigger the sensor's status based on latest distance.
        triggerSensorStatus(dataSnapshot, new_d[0]);

    }
}
```

The function 'Compute Latest Distance' is used in the onStart() method. Once the App perceived new location information updated from the Firebase, it'll compares the latest latitude & longitude of the device's current location & device's defined safety location for distance in meters.

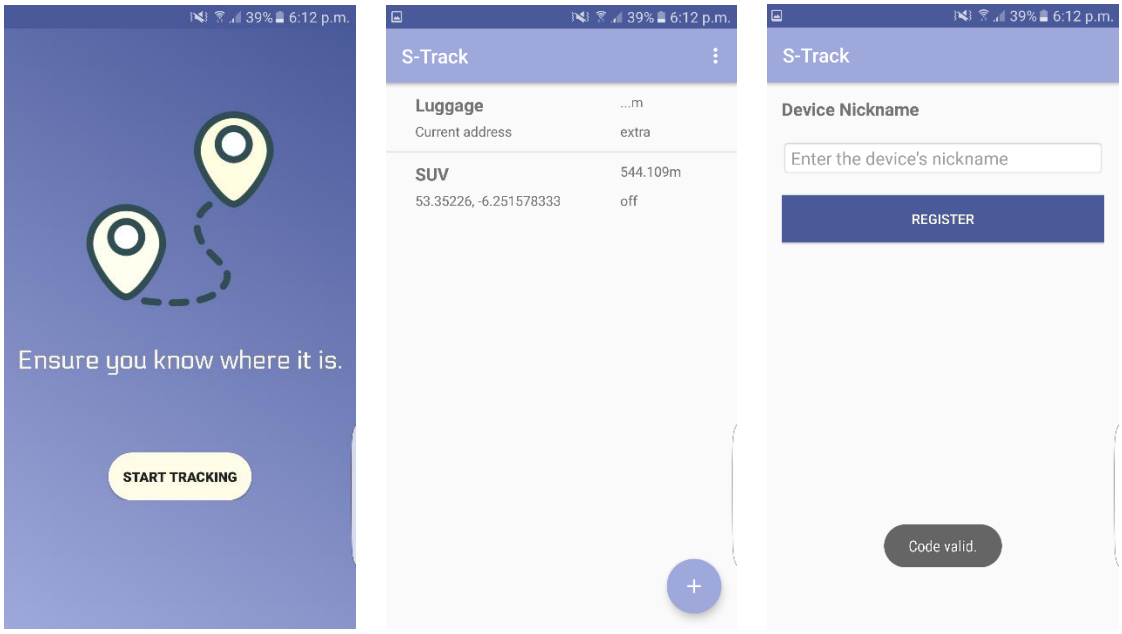
2.3.5 Send on/off order to Firebase if device exceeds safety zone

```
//Function - Trigger the sensor's status based on latest distance.
private void triggerSensorStatus(DataSnapshot dataSnapshot, float latestDistance){
    //If distance < 50m, trigger sensor status to 'on'.
    if(latestDistance > 50){
        dataSnapshot.child("extra").getRef().setValue("on");
    }else{
        dataSnapshot.child("extra").getRef().setValue("off");
    }
}
```

The function 'Trigger Sensor Status' will send on/off order to Firebase if the device exceeds the defined safety zone. Here the safety area is set as 50 meters, which could be changed according to needs.

2.4 Graphical User Interface (GUI) Layout

The following describes the mock-ups of the STrack's mobile application:



1st screen

Once the user starts the Mobile App, an animated Splash Screen is displayed.

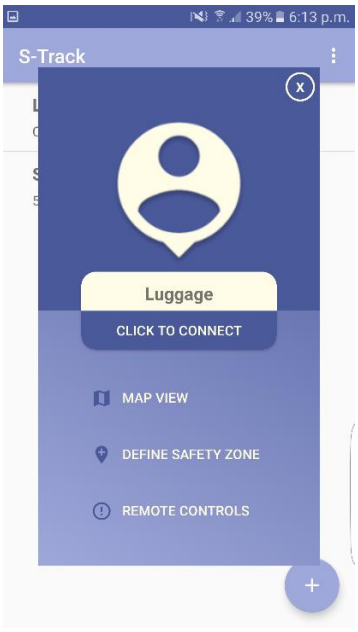
2nd screen

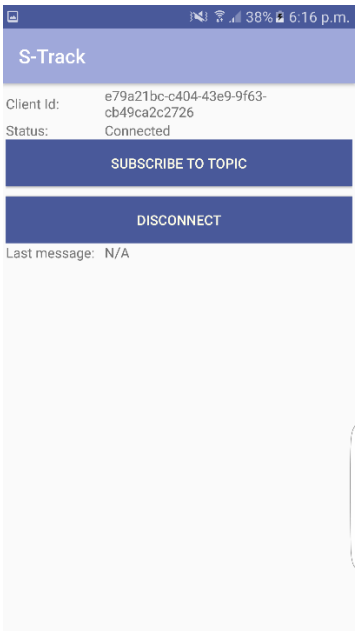
Once the user enter the App, he/she will be navigated to this page. Here the user will be able to access the registered devices' list in a list view. The user will be able to obtain the information of the particular device including the device name, current location, distance, and alarm status.

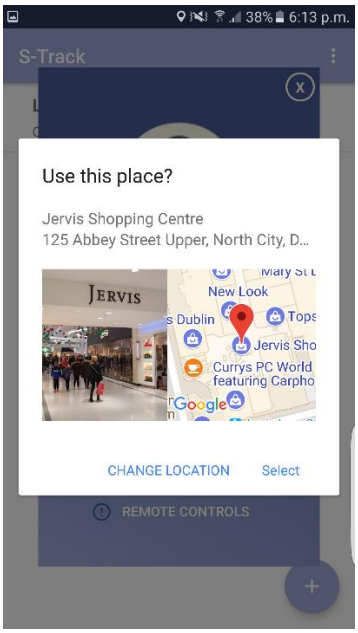
If the user have no device registered yet, the list will be empty. By pressing the '+' button, the user will be navigated to QR Code scanning page to verify and register a device.

3rd screen

Once the user success to scan the QR code, he/she will be navigated to this page. This allows the user to enter a nickname for the particular registered device. Once the user save the details entered, he/she will be navigated back to the main screen.







4th screen

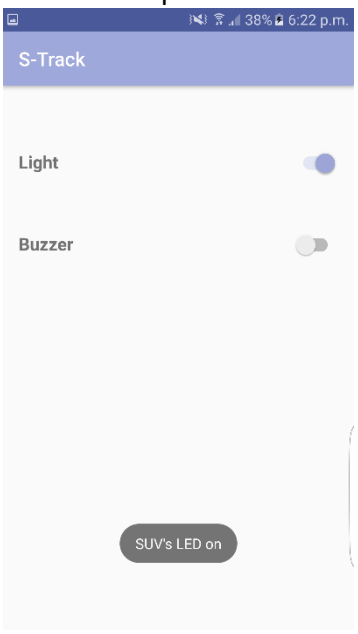
This screen shows the options provided once a list item is clicked.

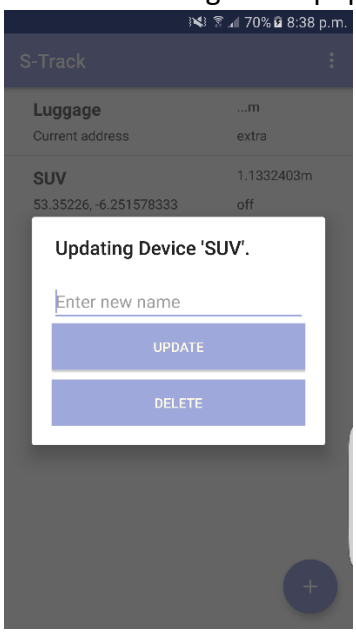
5th screen

Once the user choose the option 'Click to connect', he/she will be navigated to this page. Here the user will be able to start the connection to specific AWS MQTT topic.

6th screen

Once the user choose the option 'Define safety zone', he/she will be navigated to a Google Map Place Picker page where he/she will be able to define a safety zone for the device. Once a place is selected a dialog is popped to confirm selection.





7th screen

Once the user choose the option 'Remote controls', he/she will be navigated to this page. Here the user will be able to remote controlling the LED & buzzer of the device. The features will be activated through turning on the toggle button.

8th screen

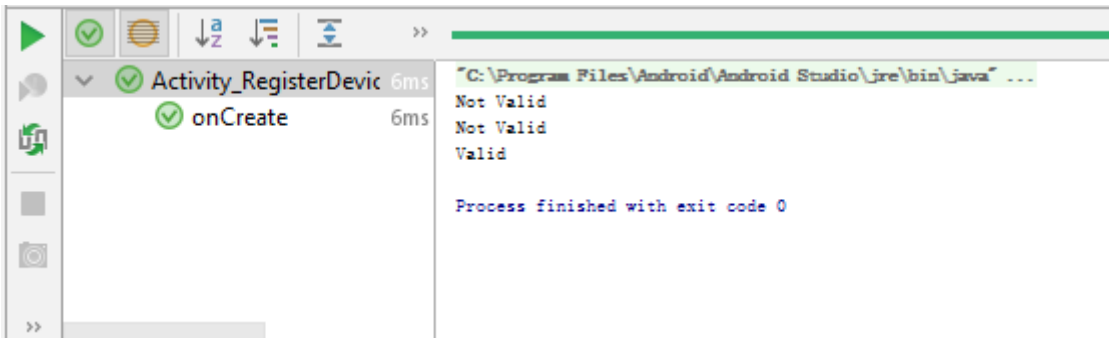
Once the user performs a long-list-item-click, a pop-up dialog is shown. Here the user is allowed to update the particular device's name, or delete the registered device.

2.5 Testing

2.5.1 Unit Testing

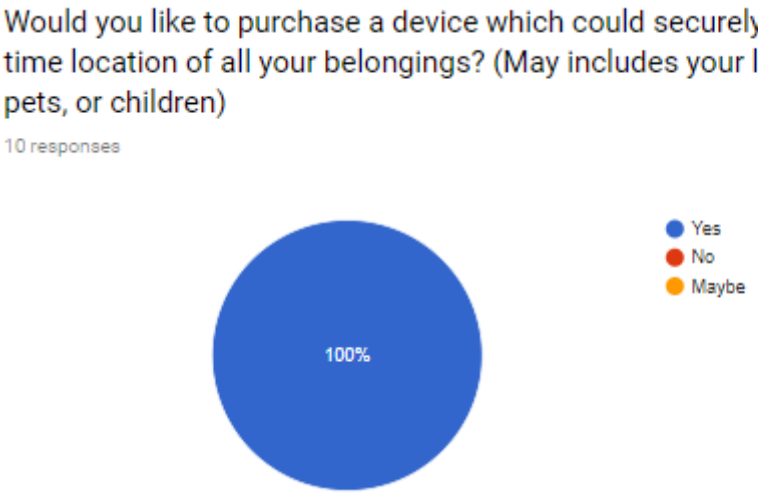
```
public class Activity_RegisterDeviceTest {

    List<String> qrCodeList;

    @Test
    public void onCreate() {
        //Initiate qrCodeList with 3 codes.
        qrCodeList = new ArrayList<>();
        qrCodeList.add("STrackCode1");
        qrCodeList.add("STrackCode2");
        qrCodeList.add("STrackCode3");

        //Define a scanned result code and check validation.
        String qrCodeResult = "STrackCode3";
        checkQRCode(qrCodeResult);
    }

    //Function - Check if the qrCode is valid
    private void checkQRCode(String qrCode){

        //For every list item:
        for(int i=0; i<qrCodeList.size(); i++){
            //Process to register if valid
            if(qrCode.equals(qrCodeList.get(i))){
                System.out.println("Valid");
                break;
            }else if(!qrCode.equals(qrCodeList.get(i))){
                System.out.println("Not Valid");
            }
        }

    }

    @Test
    public void registerClick() {
    }
}
```

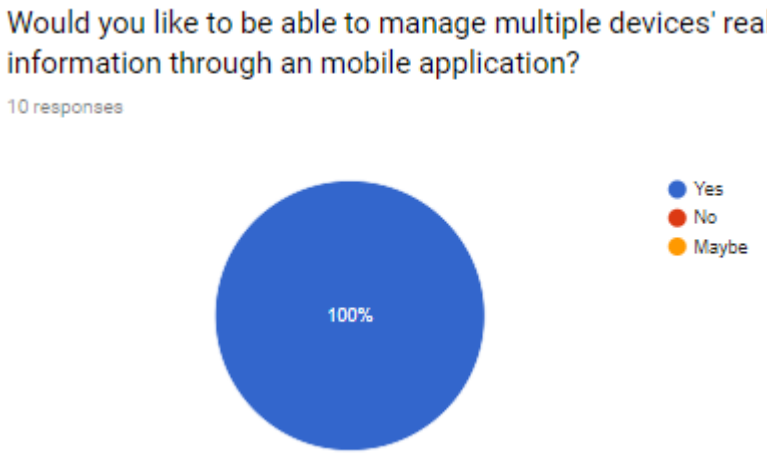
Here the QR code's verifying function is tested. Where a 'qrCodeList' with 3 valid codes is pre-defined, while the scanned result is defined as 'qrCodeResult'.



The test provides positive result where it compares the 'qrCodeResult' with every items in the 'qrCodeList', and only outputs 'valid' when the two items matched.

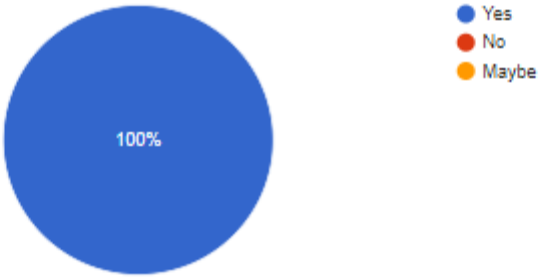
2.5.2 Survey





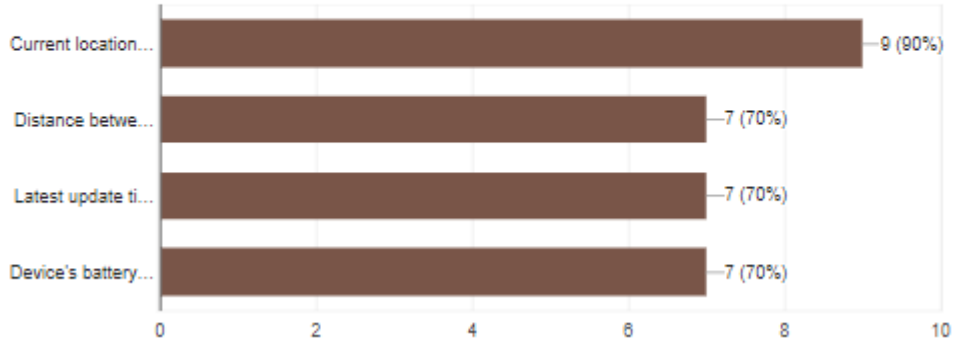
Would you think having 'anytime' access to your belongings makes you feel safer?

10 responses



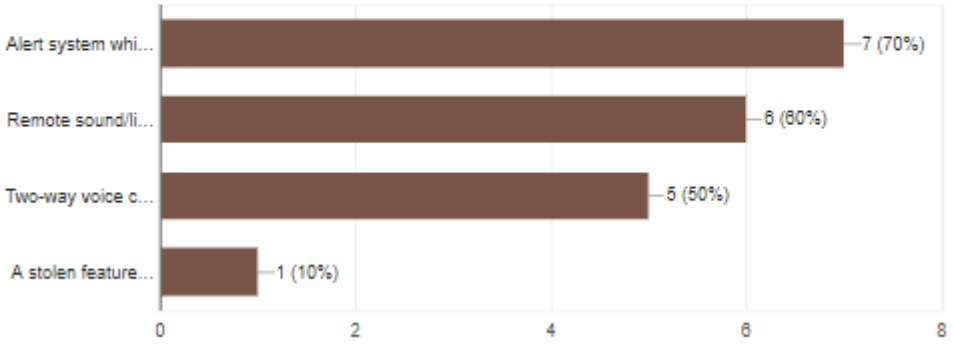
What real-time information would you like to have access from the mobile app?

10 responses



What additional functionalities would you like to see on the device with its mobile app, instead of real-time tracking?

10 responses



2.6 Evaluation

2.6.1 Evaluate with the system

Accuracy - The deviation of the tracked location shouldn't larger than a particular distance.

Constant update - The GPS data received should be updated in a 10 second base.

Data storage - All the data caught should be stored in the cloud database.

2.6.2 Evaluate with an end-user

As STrack is consider as a multi-purpose device, here we'll focus on one of its functioning area – pet tracking. By attaching the device on your pet cat, you'll be able to keep track on your cat's latest location when you are in your office. Once the cat leave the safety zone you'll get an alarm.

3 Conclusions

The scope of this particular project is to develop a device which will allows real time location tracking through a mobile application using GPS technology. Its aim is to make people's life easier through achieving the 'attach-and-locate' any belongings in seconds.

3.1 Advantages

Providing real-time location tracking is a huge inspection regards to a safety manner.

3.2 Disadvantages

There were be some security concern despite the QR Code identification provided. Hackers might be able to access your device's location data which might cause problem.

3.3 Opportunities

There's opportunity for the device to provide further safety-functionalities such as embedding it with the elderly's home-tracking system which could track the elderly's life style without invasion of individual's privacy.

3.4 Limits

There would be a huge limitation on the device as it need to be small. While nowadays microcontroller is getting smaller and smaller which might solve this problem.

4 Further development or research

As of today, the system is continuing to grow and evolve. Practically re-inventing features of the application allow the project to evolve according to user needs and maintain growth.

4.1 Advance portability

In the age of Internet-of-things, vendors work hard to constantly ameliorate smaller while stronger chips. Make use of itty-bitty hardware instead of credit-card size Raspberrypi and GrovePi would not only increase the portability of the device but also allow the device to be indiscernible, which would strengthen its utility in the safety-manner.

4.2 Adjunction of the GSM service

GSM stands for Global System for Mobile Communication. It determines an object's position through triangulation from base stations when GPS catches location information through satellites. Base stations are capable of providing locations in areas like tunnel and dense areas, which allows it to remedy the inaccuracy of GPS in certain scenarios. Combining the GSM and GPS will consummate the project with no doubt.

5 References

Brownell, B. (2014). *The 2,000-Year History of GPS Tracking*. [online] Mother Jones. Available at: http://www.motherjones.com/politics/2014/04/you-are-here-book-hiawatha-bray-gps-navigation/3/ [Accessed 14 Nov. 2017].

Rouse, M. (2014). *What is GPS tracking?*. [online] WhatIs.com. Available at: http://whatis.techtarget.com/definition/GPS-tracking [Accessed 12 Nov. 2017].

Y. Ali, A. and Beram Jasim, I. (2015). *Tracking Mobile Emitter Using TDOA and FDOA Techniques*. [ebook] Iraq, p.2. Available at: http://www.ijeert.org/pdf/v3-i9/7.pdf [Accessed 15 Nov. 2017].

Adafruit.com. (2018). *Adafruit Ultimate GPS Breakout - 66 channel w/10 Hz updates*. [online] Available at: https://www.adafruit.com/product/746 [Accessed 6 Dec. 2018].

Differencebetween.info. (2018). *Difference between GSM and GPS | GSM vs GPS*. [online] Available at: http://www.differencebetween.info/difference-between-gsm-and-gps [Accessed 3 Mar. 2018].

6 Appendix

6.1 Project Plan

Task Name	Duration	Start	Finish
Final Project Development	**165 days**	**Mon 18/09/17**	**Fri 04/05/18**
Scope	**30 days**	**Mon 18/09/**	**Fri 27/10/17**
Determine project scope	11 days	Mon 18/09/17	Mon 02/10/17
Define preliminary resources	19 days	Tue 03/10/17	Fri 27/10/17
Scope complete	0 days	Fri 27/10/17	Fri 27/10/17
Analysis/Software Requirements	**21 days**	**Mon 30/10/17**	**Mon 27/11/17**
Conduct needs analysis	5 days	Mon 30/10/17	Fri 03/11/17
Draft preliminary software specifications	3 days	Mon 06/11/17	Wed 08/11/17
Develop preliminary budget	2 days	Thu 09/11/17	Fri 10/11/17
Review software specifications	4 hrs	Mon 13/11/17	Mon 13/11/17
Secure required resources	9.5 days	Mon 13/11/17	Fri 24/11/17
Analysis complete	1 day	Mon 27/11/	Mon 27/11/17
Design	**7 days**	**Tue 28/11/1**	**Wed 06/12/17**
Review preliminary software specifications	1 day	Tue 28/11/17	Tue 28/11/17
Develop functional specifications	1 day	Wed 29/11/17	Wed 29/11/17
Develop prototype based on functional specifications	2 days	Thu 30/11/17	Fri 01/12/17
Review functional specifications	2 days	Mon 04/12/17	Tue 05/12/17
Incorporate feedback into functional specifications	1 day	Wed 06/12/17	Wed 06/12/17
Design complete	0 days	Wed 06/12/	Wed 06/12/17
Development	**33 days**	**Mon 18/12/**	**Wed 31/01/18**
Review functional specifications	1 day	Mon 18/12/17	Mon 18/12/17
Identify modular/tiered design parameters	1 day	Tue 19/12/17	Tue 19/12/17
Assign development staff	1 day	Wed 20/12/17	Wed 20/12/17
Develop code	15 days	Thu 21/12/1	Wed 10/01/18
Developer testing (primary debugging)	15 days	Thu 11/01/18	Wed 31/01/18
Development complete	0 days	Wed 31/01/18	Wed 31/01/18
Testing	**67 days**	**Thu 01/02/1**	**Fri 04/05/18**
Develop unit test plans using product specifications	4 days	Thu 01/02/18	Tue 06/02/18
Develop integration test plans using product specifications	4 days	Thu 01/02/18	Tue 06/02/18
Unit Testing	**15 days**	**Wed 07/02/**	**Tue 27/02/18**
Review modular code	5 days	Wed 07/02/18	Tue 13/02/18
Test component modules to product specifications	2 days	Wed 14/02/18	Thu 15/02/18
Identify anomalies to product specifications	3 days	Fri 16/02/18	Tue 20/02/18
Modify code	3 days	Wed 21/02/	Fri 23/02/18
Re-test modified code	2 days	Mon 26/02/18	Tue 27/02/18
Unit testing complete	0 days	Tue 27/02/18	Tue 27/02/18
Integration Testing	**12 days**	**Wed 28/02/**	**Thu 15/03/18**
Test module integration	5 days	Wed 28/02/18	Tue 06/03/18
Identify anomalies to specifications	2 days	Wed 07/03/18	Thu 08/03/18
Modify code	3 days	Fri 09/03/18	Tue 13/03/18
Re-test modified code	2 days	Wed 14/03/18	Thu 15/03/18
Integration testing complete	0 days	Thu 15/03/18	Thu 15/03/18