International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Bike Theft Protection App: Make Your Bike Unique with Bike Theft Protection App

Sanket Jadhav[a], *Shantanu Chougule*[a], *Shrinivas Kangralkar*[a], *Hrishikesh Kadam*[a], *Nilesh Ghorpade*[b]

Student,Sanjay Ghodawat Polytechnic, Atigre,India
Head of Department(C.S.E),Sanjay Ghodawat Polytechnic,Atigre,India

ABSTRACT

Bike Theft Protection App is an Arduino based project. Through this application user can track his/her vehicle live from anywhere from the world. User also gets notified when his/her vehicle falls or moves. Module built on bike uses an Arduino microcontroller for controlling all the connected modules. GSM Sim900a used for wireless communication between the bike and to the Thingspeak server. For obtaining GPS coordinates, we have used NEO 6M GPS module. For fall detection, we have used MPU 6050 3-axis Gyroscope sensor. We this system, chances of getting bike stolen will decrease and process of retrieving stolen bike also becomes easier.

Keywords:Internet of Things (IOT), Arduino, GSM, Gyroscope,cross-platform mobile app.

Introduction

Motorbikes have now become an integral part of Indian culture today. Due to the large population, overcrowded streets, and poor infrastructure, motorbikes have now become a preference for most Indians trying to commute to work or to run errands. As motorbikes are getting more and more affordable most Indian families have at least two motorbikes for their daily purpose. Electric vehicles are also in trend today. There are nice for running errands or even commuting to work.

As motorbike industry went boom, bike theft has also went rampant over the years. Such cases can get serious as much to hurt the owner of the motorbikes. The chances of getting stolen motorbikes are very slim and can be very frustrating. According to statistics done by the Delhi police, more than four vehicles are stolen every hour with motorbikes being the main target. Motorbikes are frequently used for street crimes such as snatching and robberies. The criminal uses stolen bikes for such purpose so that even if the whole crime gets recorded in the CCTV footage, the criminal cannot be traced.

Cars have better security system than bikes such as GPS tracking, send real time alerts and ability to shutdown the engine remotely. Through this project, we propose to adapt car security to motorbikes. With such security system user is able to get real time alerts, thus preventing the motorbike for being stolen. This security system also provides ability to track and recover the vehicle with help of a Global Positioning System (GPS) technology.

The proposed system consists of a cross-platform mobile app and a hardware unit. The hardware unit consists of Arduino UNO as a microcontroller, GSM Sim900a module for GPRS data transmission, MPU 6050 3-axis gyroscope + accelerometer, GPS module for reading sensor values. The main battery of the vehicle provides a power supply to the hardware unit.

The mobile app is built using React-native developed by Facebook. It helps build mobile apps faster and the feel of the app is close to a native app. For storing data we are going to use Thingspeak Server.

Our aim through this project is to make motorbikes especially electric motorbikes safer. This project is scalable and can be applied not to electric vehicles but to any other vehicles such as cars, trucks, bicycles, etc.

Literature Review

Our main motivation for this project is the number of increasing bike thefts in various major cities in India and less recovery rate. In Nashik, 8 in 10 stolen bikes are not recovered in 7 years. Police registered 4,632 cases of vehicle theft and managed to detect only 971 cases, which is mere 20.96, in the last seven years. Similar situation can be found in other major cities like Kolkata. Highest number of bike theft is found in capital Delhi. Statistics shows that there is more need for project such as ours to be implemented at large scale. If not implemented, bike security issue will never be resolved.

- ***Intelligent Anti-Theft and Tracking System for AutomobilesMontaser N. Ramadan, Mohammad A. Al-Khedher, Senior Member, IACSIT, and Sharaf A. Al-Kheder***

An efficient automotive security system is implemented for anti-theft using an embedded system occupied with a Global Positioning System (GPS) and a Global System of Mobile (GSM). The client interacts through this system with vehicles and determines their current locations and status using Google Earth. The user can track the position of targeted vehicles on Google Earth. Using GPS locator, the target current location is determinhttps://www.overleaf.com/project/8ed and sent, along with various parameters received by vehicle's data port, via Short Message Service (SMS) through GSM networks to a GSM modem that is connected to PC or laptop. The GPS coordinates are corrected using a discrete Kalman filter. To secure the vehicle, the user of a group of users can turn off any vehicle of the fleet if any intruders try to run it by blocking the gas feeding line. This system is very safe and efficient to report emergency situations as crash reporting or engine failure.

- ***Two-Wheeler Vehicle Security System***

Proposed security system also focuses on adapting car security to bike. The system uses same components such as GPS module for communication, Global Positioning System (GPS) technology for tracking of vehicle. All the alerts are provided to user through SMS.

- ***Anti-Theft Protection of Vehicle Using GPS Tracker & Android Apps***

Proposed system uses an android app and similar setup for bike security. It includes ability to remotely shut off the power supply to the engine, GPS tracking and also fingerprint scanner as an extra additional layer of security.

Objective and Scope of project

- ***Objective of Project***
    - To develop an affordable Vehicle tracking system.
    - The GPS coordinates of the bike's position will be stored in Thingspeak Server.
    - To prevent the bike from being stolen.
    - To make the recovery of stolen vehicles easier.
- ***Scope of project***

Systems like this are being proposed for a very long time. Asides from cars, such systems were never implemented on bikes. Due to this, bikes have become easier targets for thieves to steal. Through this project security features of the car can be implemented to bikes like GPS tracking and fall detection. Such a security system can be placed on any kind of bike as the main power supply comes from the batteryof the vehicle. This project can also be used for bicycles by providing power through an external battery.

Methodology

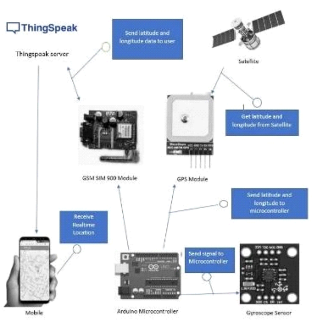


Fig. 1 – System Architecture

The above figure shows the system architecture of the project. The project is divided into two parts i.e., software and hardware. The hardware consists of Microcontroller (Arduino UNO), GSM Sim900a module, GPS module and Gyroscope sensor. Arduino UNO serves as the main controller of all hardware component's operations. GSM module acts as an intermediatory between vehicle and owner providing wireless communication. It is responsible for sending data to the database. GPS module is used to obtain vehicles GPS coordinates. Gyroscope sensor is used for obtaining gyroscopic data.

- ***Module 1: Global Positioning System***

Fig. 2 - Global Positioning System

Global Positioning System is responsible for fetching real-time GPS coordinates of the bike. There are 24 GPS satellites revolving around the Earth. GPS module uses these satellites to calculate latitude and longitude, and as well as the altitude of the vehicle. The GPS receiver gets a signal from each GPS satellite. The satellites transmit the exact time the signals are sent. By subtracting the time the signal was transmitted from the time it was received, the GPS can tell how far it is from each satellite. The GPS receiver also knows the exact position in the sky of the satellites, at the moment they sent their signals. So given the travel time of the GPS signals from three satellites and their exact position in the sky, the GPS receiver can determine your position in three dimensions – east, north and altitude.
The module we are going to use is GPS NEO6MV2 NEO-6M GPS. The operation voltage is 3.3V. This module has a built-in voltage regulator and it is sometimes referred to as GY-GPS6MV2. It is based on a u-blox NEO-6M GPS module with an onboard backup battery and built-in EEPROM.

Fig. 3 – MPU 6050 Gyroscope Sensor

- ***Module 2: Gyroscope Sensor***

A gyroscope module is used to obtain rotational y-axis data of the bike. Basically to detect whether the bike has fallen or not. The gyroscope sensor we are going to use is MPU-6050. The MPU-6050 3-Axis Accelerometer and Gyro Sensor module use MPU-6050 which is a little piece of motion processing tech. The operation voltage of the device is about 3.3V to 5V. The MPU6050 devices combine a 3-axis gyroscope and a 3-axis accelerometer on the same silicon together with an onboard Digital Motion Processor (DMP) capable of processing complex 9-axis MotionFusion algorithms.

- ***Module 3: GSM Sim900a***

GSM Sim900a is used for wireless communication between the module mounted on the bike and Thingspeak server. GSM Sim900a uses cellular data to connect to the internet. GSM Sim900a is capable of making HTTP requests. Through write API from ThingsSpeak server we can send data to the server using HTTP POST request.

Fig. 4 – ThingsSpeak Server

- ***Module 4: Data storage module***

Thingspeak server is used for storing data received from the vehicle. GSM module is used for connection between the vehicle and the database. ThingsSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to ThingsSpeak from your devices, create instant visualization of live data, and send alerts.

Fig.5 - React Native

- ***Module 5: Mobile App***

React Native is used for developing the mobile app. Due to its native functionality, faster development speed, it is suitable for application such as ours which has relatively less functionalities.

React Native is a mobile app development platform developed by Facebook for its internal app development. It was open sourced in March 2015 for iOS mobile app development. By September, an Android version was released too. Later, it provided support for Windows and Tizen platforms as well.It provides a slick, smooth and responsive user interface, while significantly reducing load time thus suitable for our purpose.

- **Flowchart**

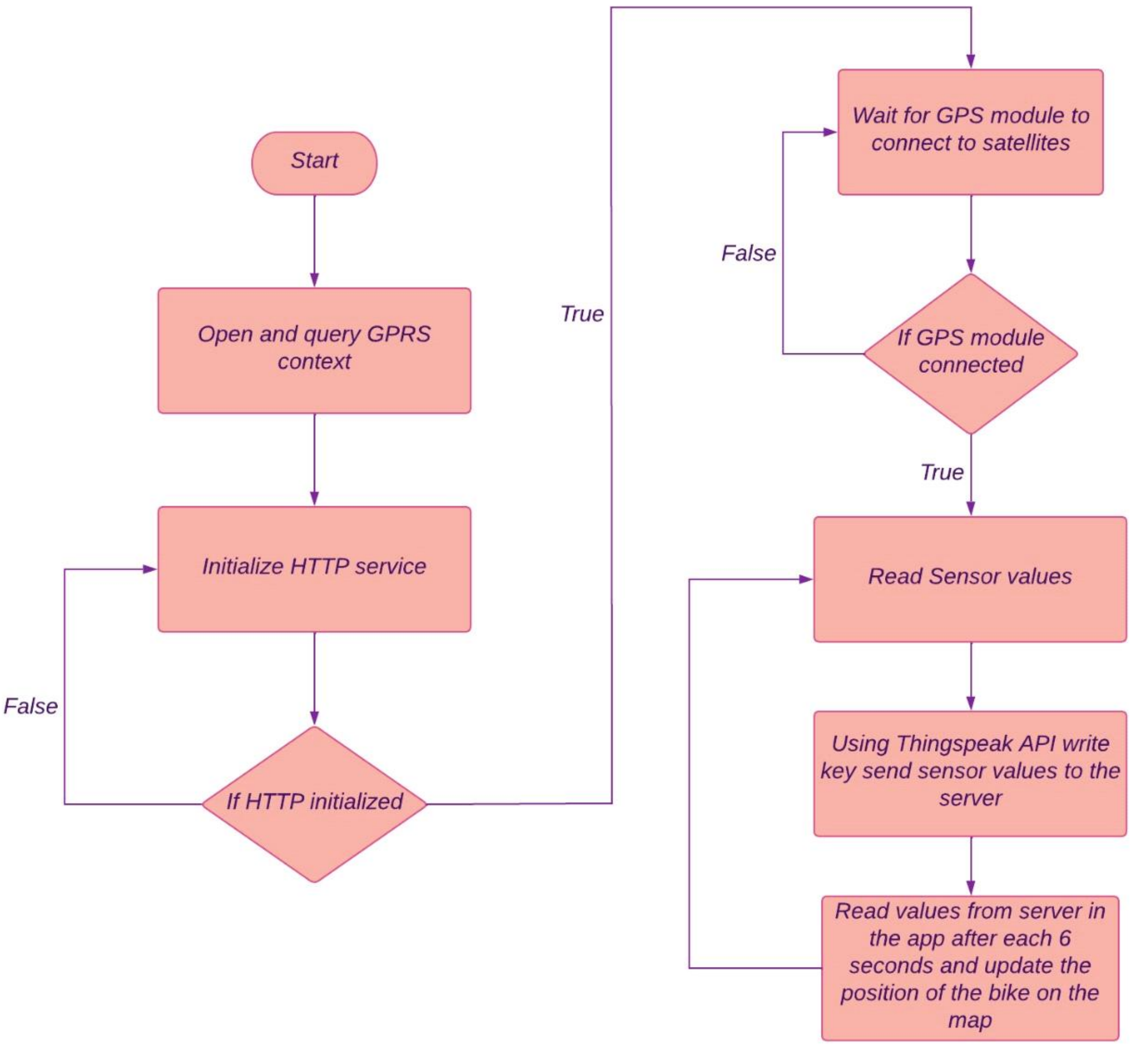


Fig. 1 Flowchart

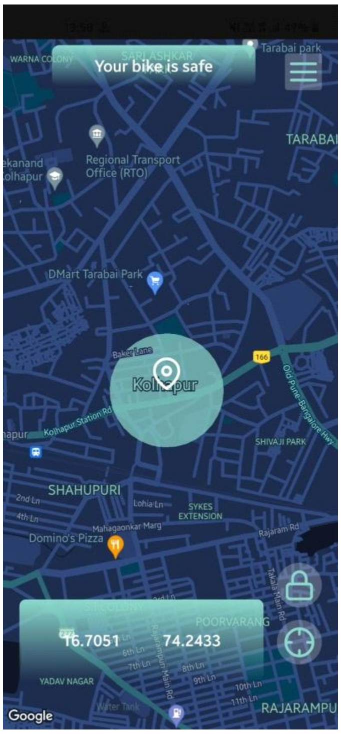
- **System Configuration**
  - ***Hardware Requirement***
    - Arduino UNO
    - GMS SIM900A Module
    - MPU6050 3-Axis Accelerometer and Gyroscope
    - NEO 6M GPS
    - Electric Bike
  - ***Software Requirement***
    - Operating System- Windows 10
    - Languages - C, Javascript, React Native Framework
    - Arduino IDE

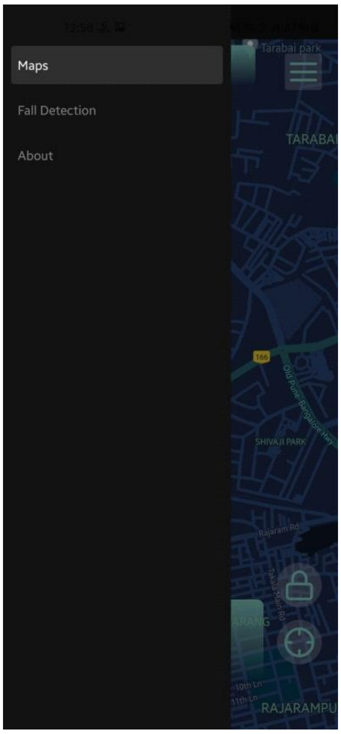
- ***Database Requirement***

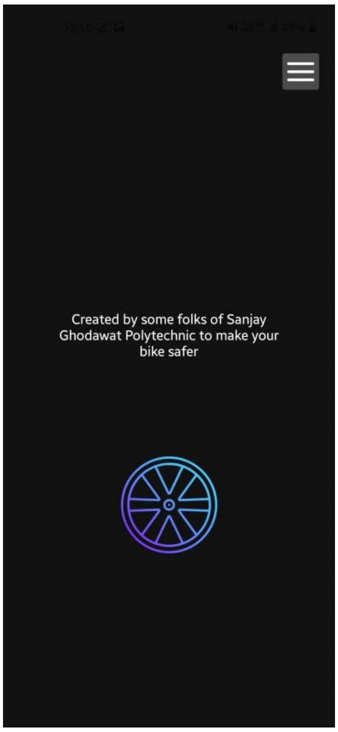
  Thingspeak Server

- **Result of Project**

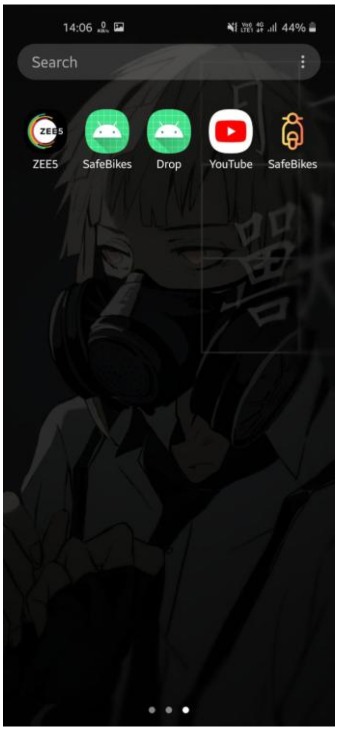
  7.1 Snapshots of Project

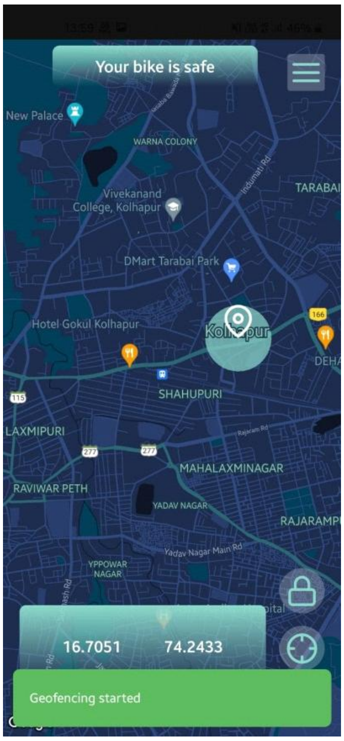


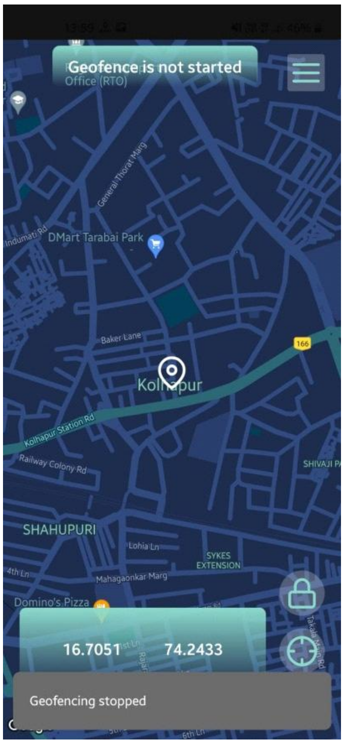




Snapshot 7.1.1 User Interface Snapshot 7.1.2 Navigation options Snapshots 7.1.3 About Us







Snapshots 7.1.4 Live Location Snapshots 7.1.5 Geo-fencing Started Snapshots 7.1.6 App closed

- *Code of Module*

```
import * as React from 'react';
import { useRef, useState, useEffect } from 'react';
import MapView, { Marker, PROVIDER_GOOGLE, AnimatedRegion, Circle } from 'react-native-maps';
import { StyleSheet, Alert, Text, View, Dimensions, TouchableOpacity, Button, TouchableWithoutFeedback,
  TouchableHighlight } from 'react-native';
import { SafeAreaView } from 'react-native-safe-area-context';
import { Feather } from '@expo/vector-icons';
import { BlurView } from 'expo-blur';
import { darkStyle, silverStyle, aubergineStyle } from './../assets/MapStyles';
import styles from './../assets/Styles';
import { LinearGradient } from 'expo-linear-gradient';
import { ALERT_TYPE, Dialog, Root, Toast } from 'react-native-alert-notification';
import AsyncStorage from '@react-native-async-storage/async-storage';
import FloatingButton from './../components/FloatingButtons';
import FlashMessage, { showMessage, hideMessage } from 'react-native-flash-message';
import * as Notifications from 'expo-notifications';
import * as Permissions from 'expo-permissions';
import Constants from 'expo-constants';
import { Audio } from 'expo-av';

const screen = Dimensions.get('window');
const ASPECT_RATIO = screen.width / screen.height;
const LATITUDNAL_DELTA = 0.02;
const LONGITUDNAL_DELTA = LATITUDNAL_DELTA * ASPECT_RATIO;

const sendNotif = (seconds) => {
const shedulingOptions = {
content: {
title: 'Warning',
body: 'Your bike has gone out of the geofenced area',
sound: true,
priority: Notifications.AndroidNotificationPriority.HIGH,
color: 'blue',
},
trigger: {
seconds: seconds
}
}
Notifications.sheduleNotificationAsync(
shedulingOptions,
);
}

const askPermissions = async  () => {
const { state } = await Permissions.askAsync(Permissions.NOTIFICATIONS);
}

const handleNotification = () => {
alert("You have got you notif");
}

// maps screen component
export default function MapsScreen(props) {

const thingspeakReadURL =
  "https://api.thingspeak.com/channels/1597049/feeds.json?api_key=YOE7IDHF4EG3ET01&results=1";

// for sound
const [sound, setSound] = useState();
```

```
// for playing sound
const [soundPlayed, setSoundPlayed] = useState(false);

async function playSound() {
const { sound } = await Audio.Sound.createAsync(
require('../assets/alarmTone.wav')
);
setSound(sound);
await sound.playAsync();
}

useEffect(() => {
return sound
? () => {
sound.unloadAsync();
}
: undefined;
}, [sound]);

const [ state, setState ] = useState({
curLoc: {
latitude: 16.7050,
longitude: 74.2433,
},
coordinate: new AnimatedRegion({
latitude: 16.7050,
longitude: 74.2433,
latitudeDelta: LATITUDNAL_DELTA,
longitudeDelta: LONGITUDNAL_DELTA,
}),
dataFromApi: {},
initialMapRegion: {},
});

// for geofence status of the component
const [ geoFenceState, setGeoFenceState ] = useState({
hasStarted: false,
loc: {},
isSafe: true,
});

// to set GeoFence status
const setGeoStatusLocal = async(geoStatus) => {
try {
updateGeoState(geoStatus);
const stringGeoStatus = JSON.stringify(geoStatus);
await AsyncStorage.setItem('geoStatus', stringGeoStatus);
} catch(e) {
alert("Failed to geofence the bike");
}
}

// to get GeoFence status from local storage
const getGeoStatusLocal = async() => {
try {
var geoStatus = await AsyncStorage.getItem('geoStatus');
geoStatus = JSON.parse(geoStatus);

// if geoStatus is not equal to null then update the state
if (geoStatus !== null) {
updateGeoState(geoStatus);
}
} catch(e) {
```

```
alert("Falied to retrieve data");
}
}

const updateState = (data) => setState((state) => ({ ...state, ...data }));

const updateGeoState = (data) => setGeoFenceState((geoFenceState) => ({ ...geoFenceState, ...data }));

const { curLoc, coordinate, dataFromApi, initialMapRegion } = state;
const { hasStarted, loc, isSafe } = geoFenceState;

// ref for map and marker
const mapRef = useRef(null);
const markerRef = useRef(null);

// to move marker as coordinates gets updated
const animate = (latitude, longitude) => {
const newCoordinates = { latitude, longitude };
if (Platform.OS == 'android') {
if (markerRef.current) {
markerRef.current.animateMarkerToCoordinate(newCoordinates, 7000);
}
} else {
coordinate.timing(newCoordinates).start();
}
}

// to get data from the thingspeak server
const getLocationFromApi = async () => {
const response = await fetch(thingspeakReadURL);
const json = await response.json();
const jsonObject = new Object(json.feeds);
const latitude_ = Number(jsonObject[0].field1);
const longitude_ = Number(jsonObject[0].field2);
animate(latitude_, longitude_);
updateState({
curLoc: {
latitude: latitude_,
longitude: longitude_
},
dataFromApi: jsonObject,
coordinate: new AnimatedRegion({
latitude: latitude_,
longitude: longitude_,
latitudeDelta: LATITUDNAL_DELTA,
longitudeDelta: LONGITUDNAL_DELTA
}),
initialMapRegion: {
latitude: latitude_,
longitude: longitude_,
latitudeDelta: LATITUDNAL_DELTA,
longitudeDelta: LONGITUDNAL_DELTA
}
});
}

// To calculate distance between two coordinates in meters
function measure(lat1, lon1, lat2, lon2) {
var R = 6378.137;
var dLat = lat2 * Math.PI / 180 - lat1 * Math.PI / 180;
var dLon = lon2 * Math.PI / 180 - lon1 * Math.PI / 180;
var a = Math.sin(dLat/2) * Math.sin(dLat/2) +
Math.cos(lat1 * Math.PI / 180) * Math.cos(lat2 * Math.PI / 180) *
```

```
Math.sin(dLon/2) * Math.sin(dLon/2);
var c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
var d = R * c;
return d * 1000;
}

// to get live data form api when the maps page is rendered for the first time
// get the geofence status from local storage
useEffect(() => {
getLocationFromApi();
getGeoStatusLocal();
askPermissions

const listener = Notifications.addNotificationReceivedListener(handleNotification);
return () => listener.remove();
}, []);

// to get live location after the page is rendered after each 5 seconds
useEffect(() => {
const interval = setInterval(() => {
getLocationFromApi();
}, 5000);
}, []); // [] means only one time at the start of the application

// to check wheter the bike has left the geofenced area or not
useEffect(() => {
// if geofence has started
if (hasStarted) {
// need to update the current location here
var dist = measure(curLoc['latitude'], curLoc['longitude'], loc['latitude'], loc['longitude']);
if (dist > 250) {
updateGeoState({
isSafe: false,
});
sendNotif(5);
// method to play the full sound until it is completed
}
else if (dist < 250) {  // doesn't update the state needlessly
if (isSafe == false) {
updateGeoState({
isSafe: true,
})
}
}
}
}
, [curLoc]);

// to start geofencing
const toggleGeoFencing = () => {
if (hasStarted) {
setGeoStatusLocal({
hasStarted: false,
loc: {}
});
showMessage({
message: 'Geofencing stopped',
type: 'default',
autoHide: true,
duration: 2000,
floating: true,
});
} else {
```

```
setGeoStatusLocal({
hasStarted: true,
loc: curLoc
});
showMessage({
message: 'Geofencing started',
type: 'success',
autoHide: true,
duration: 2000,
floating: true,
});
}
}

const animateToCurrentLocation = () => {
if (mapRef.current) {
mapRef.current.animateToRegion(initialMapRegion, 2000);
}
}

function userMessage() {
if (isSafe && hasStarted) {
return(
<Text style={{color: '#fff', fontSize: 18, fontWeight: 'bold'}}>Your bike is safe</Text>
);
} else if (hasStarted && !isSafe) {
return <Text style={{color: '#F32013', fontSize: 18, fontWeight: 'bold'}}>Your bike is not safe</Text>;
} else {
return <Text style={{color: '#fff', fontSize: 18, fontWeight: 'bold'}}>Geofence is not started</Text>;
}
}

return (
<SafeAreaView style={styles.container}>
<MapView
style={customStyle.map}
provider={PROVIDER_GOOGLE}
showUserLocation
initialRegion={initialMapRegion}
showCompass={false}
ref={mapRef}
mapType="terrain"
customMapStyle={aubergineStyle}
>
<Marker
coordinate={curLoc}
ref={markerRef}
>
<Feather name="map-pin" size={40} color="#fff" />
</Marker>
{hasStarted?
<Circle
radius={250}
fillColor={"rgba(137, 214, 200, 0.7)"}
center={loc}
strokeWidth={0}
/>: null
}
</MapView>

<LinearGradient style={customStyle.bikeStatus} colors={['#89d6c8', 'transparent']} end={ {x : 0, y : 1} }>
{
userMessage()
```

```
}
</LinearGradient>

{/*Side menu and cross floating button*/}
<FloatingButton
data={props.navigation.toggleDrawer}
tint={"light"}
color={"#89d6c8"}
name={"menu"}
style={styles.sideMenuButton}
/>

<FloatingButton
data={animateToCurrentLocation}
tint={"light"}
color={"#89d6c8"}
name={"crosshair"}
style={styles.crosshairButton}
/>

<FloatingButton
data={toggleGeoFencing}
tint={"light"}
color={"#89d6c8"}
name={"lock"}
style={styles.lockButton}
/>

<LinearGradient style={styles.liveCoordinates} colors={['#89d6c8', 'transparent']} end={ {x : 0, y : 1} }>
<Text style={{color: '#fff', fontSize: 18, fontWeight: 'bold'}}>
<Text>
{typeof dataFromApi[0] == 'object' ?  JSON.stringify(dataFromApi[0].field1).replace(/^"(.*)"$/, '$1'): 'Fetching'}
</Text>
</Text>
<Text style={{color: '#fff', fontSize: 18, fontWeight: 'bold'}}>
<Text>
{typeof dataFromApi[0] == 'object' ?  JSON.stringify(dataFromApi[0].field2).replace(/^"(.*)"$/, '$1'): 'Fetching'}
</Text>
</Text>
</LinearGradient>
<FlashMessage position="bottom" style={customStyle.messageStyle}/>
</SafeAreaView>
);
}

const customStyle = StyleSheet.create({
map: {
width: Dimensions.get('window').width,
height: Dimensions.get('window').height,
},
messageStyle: {
height: 70,
justifyContent: 'center',
},
bikeStatus: {
height: 50,
width: 250,
position: 'absolute',
top: 50,
left: 60,
justifyContent: 'center',
alignItems: 'center',
borderRadius: 9,
```

```
}
  });
```

9.Advantages and Disadvantages of Proposed System

9.1 ***Advantages***

- This proposed system can effectively decrease bike theft cases.
- The proposed system can also increase recovery rate of stolen bikes.
- Bike can be tracked from anywhere from the world.

9.2 ***Disadvantages***

- Transitioning to new technology and processes.
- Managing public reaction and customer acceptance of the new GPS tracked bike.
- Managing and storing vast quantities of GPS coordinates and Gyroscopic data.
- Ensuring the security of metering data.

Conclusion

The proposed system ensures your bike's security. Such systems have been proposed for a long time but are never implemented on large scale or have reached to people. As India is the second most populous country in the world with no surprise India is also one of the largest users of motorbikes with record 37 million motorcycles/mopeds. Bike security is one of the major concerns in India and such system will help solve this problem.

References

1. Ibrahim et. al. (2012). Microcontroller Based Anti-Theft Security System Using GSM Networks with Text Message as Feedback[Online]. Available :http://www.ijerd.com/paper/vol2-issue10/D02HYPERLINK "http://www.ijerd.com/paper/vol2-issue10/D02" 101822.pdf
2. Patilet. al. (2013). Design and Implementation of Automotive Security System using ARMProcessor[Online].Available: http://www.ijesit.com/Volume%202/Issue%201/IJESIT2HYPERLINK "http://www.ijesit.com/Volume%202/Issue%201/IJESIT2" 01301_68.pdf
3. Prashantkumar et. al. (2013). Two Wheeler Vehicle Security System [Online]. Available : http://www.ijeset.com/media/0002/6N13-IJESET060312 7-v6-iss3-324-334.pdf
4. Dalizon, A. (May 17, 2015). Car, MV thefts rise due to PNP's 'genuine' crime reporting program.[Online] Available: http://www.journal.com.ph/news/nation/car-mv-thefts-riHYPERLINK "http://www.journal.com.ph/news/nation/car-mv-thefts-ri" se-due-to-pnp-s-genuine-crime-reporting-program [8] McDono, C. The Changing Face of Vehicle Theft. [Online] Available http://www.policechiefmagazine.org/magazine/index.cf m?fuseaction=display_arch&article_id=2420&issue_id= 72011