Logo, company name

Description automatically generated

Notifyneighborhood.com is a smartphone panic button app that is a great idea given the increasing rate of theft in neighborhoods. The app would allow users to easily notify 10 neighbors when they feel unsafe and in need of help. The app would also provide a map of the respondents’ movement towards the user’s house, so that they can be assured that help is on the way.

To create the app, we would need to develop a user-friendly interface, set up a secure network connection, and create a notification system that would alert the neighbors. Additionally, we would need to ensure the security of the app user’s data and make sure the app is compliant with all relevant laws. Furthermore, we would monetize the app, either through in-app purchases or paid subscriptions.

Overall, creating this smartphone panic button app is a great idea that could provide peace of mind to people in dangerous or uncertain situations.

With the right resources and dedication, we can create a helpful and secure app that can make a real difference.

java

Copy code

import android.Manifest;

import android.content.Context;

import android.content.pm.PackageManager;

import android.location.Location;

import android.location.LocationListener;

import android.location.LocationManager;

import android.os.Bundle;

import android.support.v4.app.ActivityCompat;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

import java.util.ArrayList;

import java.util.List;

public class PanicButtonActivity extends AppCompatActivity {

private static final int REQUEST\_LOCATION\_PERMISSION = 1;

private List<String> neighbors;

private LocationManager locationManager;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_panic\_button);

neighbors = new ArrayList<>();

neighbors.add("Neighbor 1");

neighbors.add("Neighbor 2");

// Add more neighbors as needed

locationManager = (LocationManager) getSystemService(Context.LOCATION\_SERVICE);

Button panicButton = findViewById(R.id.panic\_button);

panicButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

sendPanicAlert();

}

});

}

private void sendPanicAlert() {

if (ActivityCompat.checkSelfPermission(this,

Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {

ActivityCompat.requestPermissions(this,

new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION},

REQUEST\_LOCATION\_PERMISSION);

} else {

locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER,

0, 0, locationListener);

}

// Notify neighbors

for (String neighbor : neighbors) {

// Send notification to neighbor

}

}

private final LocationListener locationListener = new LocationListener() {

@Override

public void onLocationChanged(Location location) {

// Handle location updates

double latitude = location.getLatitude();

double longitude = location.getLongitude();

// Display map with the movement of respondents towards the user's house

// Code for displaying map goes here

}

@Override

public void onStatusChanged(String provider, int status, Bundle extras) {

// Handle status changes

}

@Override

public void onProviderEnabled(String provider) {

// Handle provider enabled

}

@Override

public void onProviderDisabled(String provider) {

// Handle provider disabled

}

};

@Override

public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {

if (requestCode == REQUEST\_LOCATION\_PERMISSION) {

if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {

sendPanicAlert();

} else {

Toast.makeText(this, "Location permission denied", Toast.LENGTH\_SHORT).show();

}

}

}

}

Please note that this is just a basic code structure and you'll need to further customize it to meet your specific requirements. Additionally, you'll need to handle the notification sending logic and implement the map functionality separately

HTML CODE

html

Copy code

<!DOCTYPE html>

<html>

<head>

<title>Smartphone Panic Button App</title>

<style>

/\* CSS styles for the app \*/

/\* Add your styles here \*/

</style>

<script>

// JavaScript code for the app

// Add your code here

</script>

</head>

<body>

<h1>Smartphone Panic Button App</h1>

<button id="panicButton" onclick="notifyNeighbors()">Panic Button</button>

<div id="mapContainer">

<!-- The map will be displayed here -->

</div>

<script>

function notifyNeighbors() {

// Get user's location

navigator.geolocation.getCurrentPosition(function(position) {

var latitude = position.coords.latitude;

var longitude = position.coords.longitude;

// Send notification to neighbors

var message = "Help! I feel unsafe and need assistance. My location is: " + latitude + ", " + longitude;

var neighbors = ["neighbor1@example.com", "neighbor2@example.com", "neighbor3@example.com"]; // Replace with actual neighbor email addresses

for (var i = 0; i < neighbors.length; i++) {

sendEmail(neighbors[i], message);

}

// Show map with responders' movement

showMap(latitude, longitude);

});

}

function sendEmail(email, message) {

// Code to send email notifications

// Replace with your own email sending implementation

}

function showMap(latitude, longitude) {

// Code to display map with responders' movement

// Replace with your own map implementation

}

</script>

</body>

</html>