Locate Me

### Android & iOS App

Technical Documentation

# 1. Project Overview

The “Locate Me” is a cross-platform mobile application built with Google’s new UI framework Flutter. The purpose of the application is to notify the emergency contact in an emergency situation with the least time and easy way.

# 

# 2. App Key Features

* Works sensor-based
* Saves emergency contact for sending SMS
* Sends SMS
* Can automatically trigger the sending of SMS and flashes light.

# 3. Used IDE & Tools

For the cross-platform application, here we use the Flutter framework of Google. The Flutter framework uses the Dart programming language for its development.

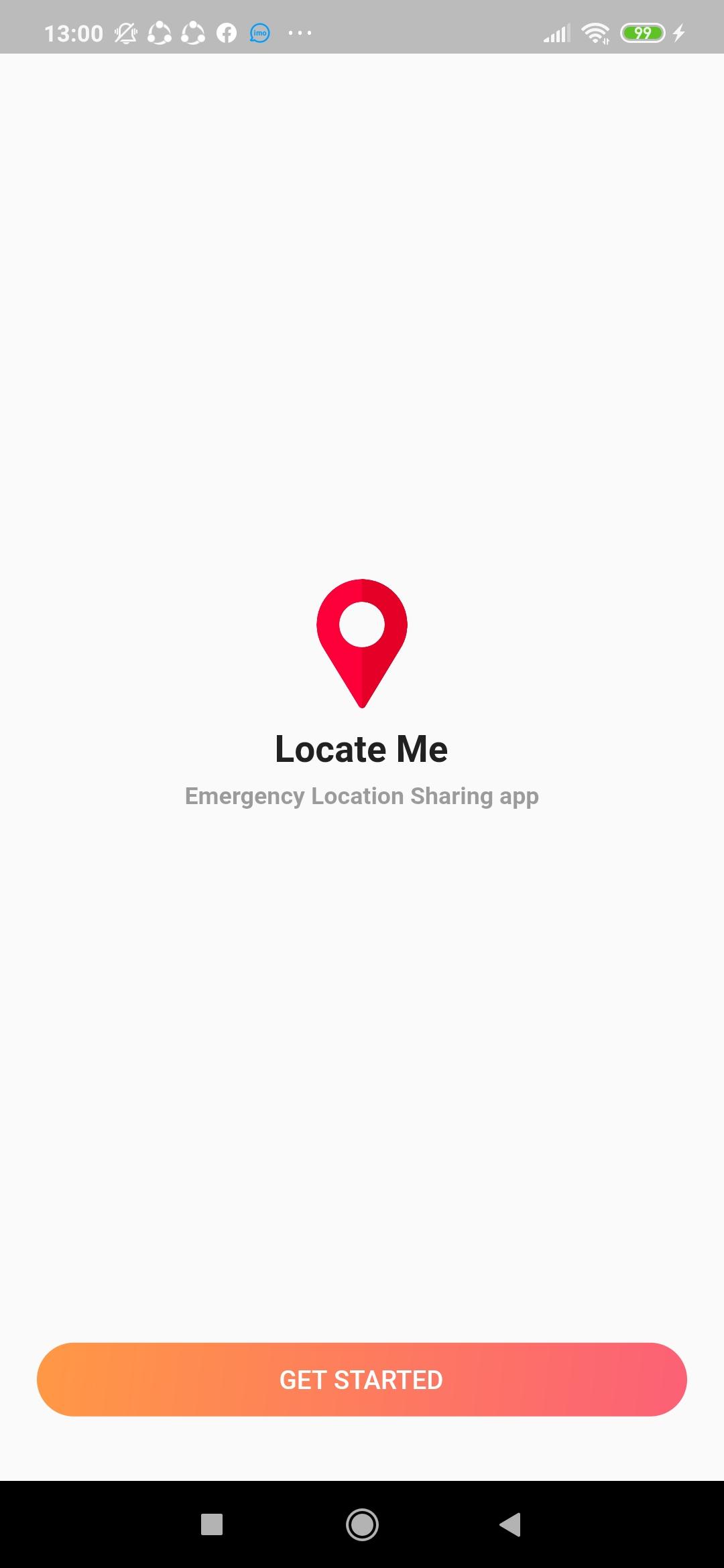
We have used Android Studio as a development IDE for developing the whole project. and only for iOS builds, we have used Xcode.

# 4. Implementation & Integration:

##### Install & launch app

##### Splash Screen

Tap the **GET STARTED** button to launch the app.



Development Process:

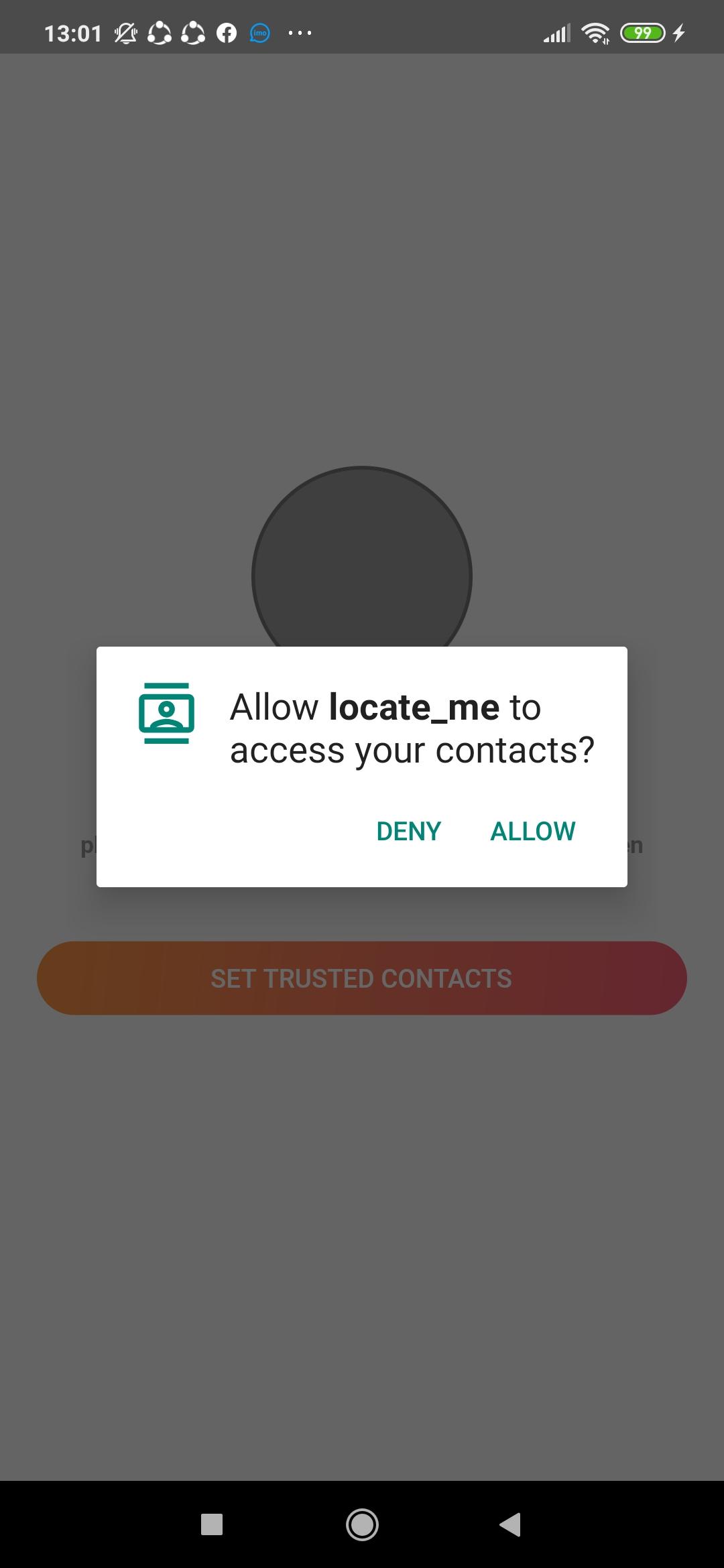
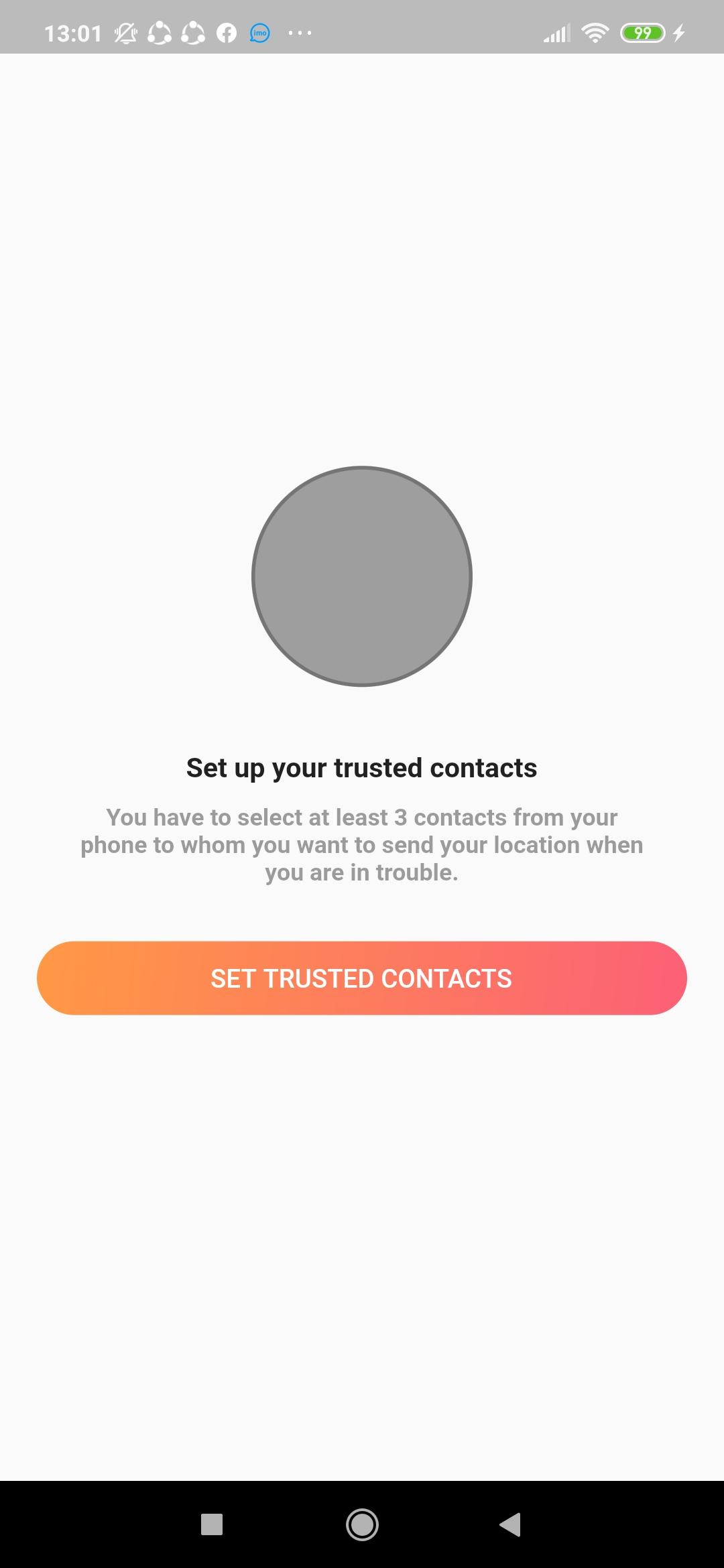
Splash screen class in main.dart file controls this screen.

firstUseFunc() -> this function checks whether a user is using this app for the first time or not.

‌ ‌ **GET STARTED** button uses roundedRectButton widget from round\_gradient\_button.dart file with given parameter('fromGetStartedButton').

##### Set up your Trusted contacts .

Shows a brief tutorial about selecting a minimum of 3 trusted contacts from phone’s contact list to whom the user wants to send location in emergency. Tap SET TRUSTED CONTACTS button. A dialog will appear prompting the user to give access to the phone contact list.



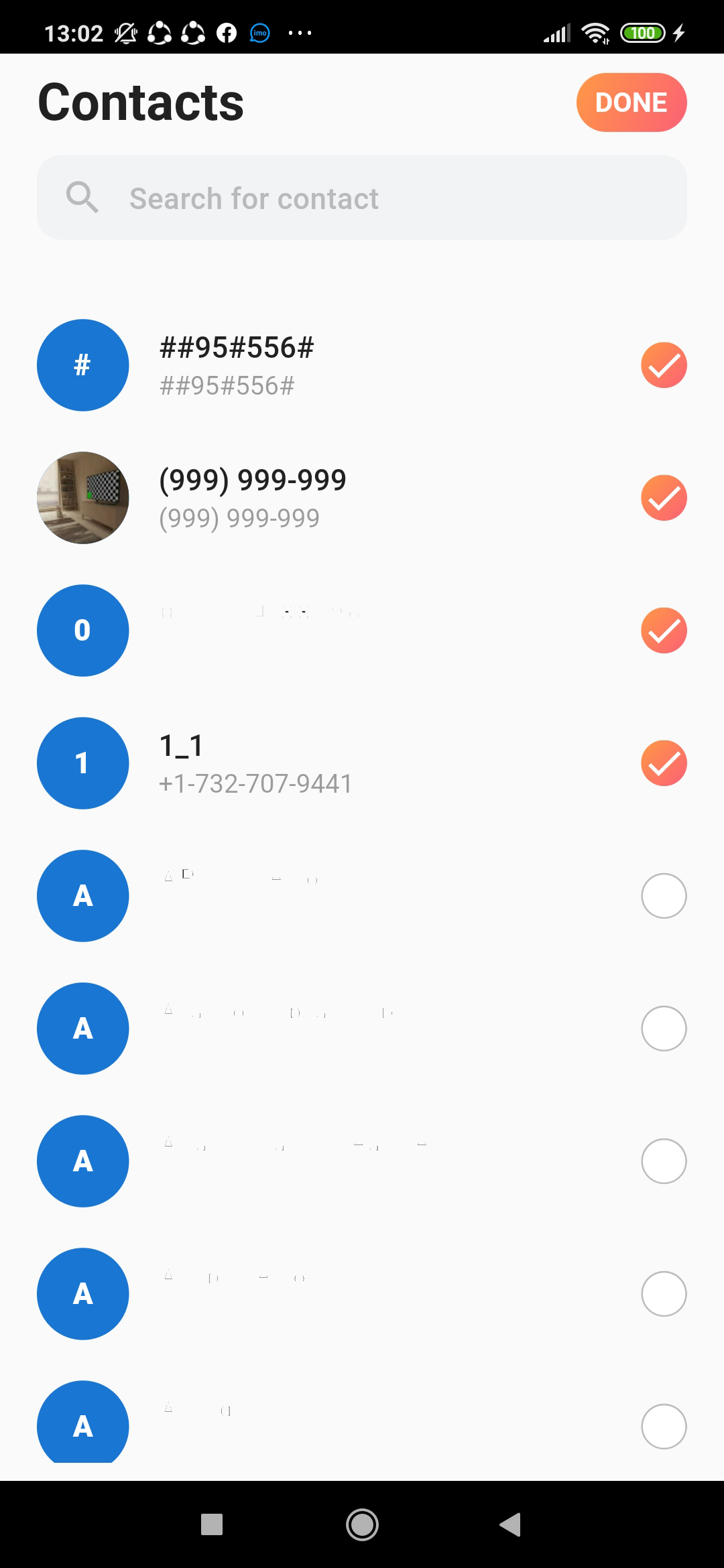
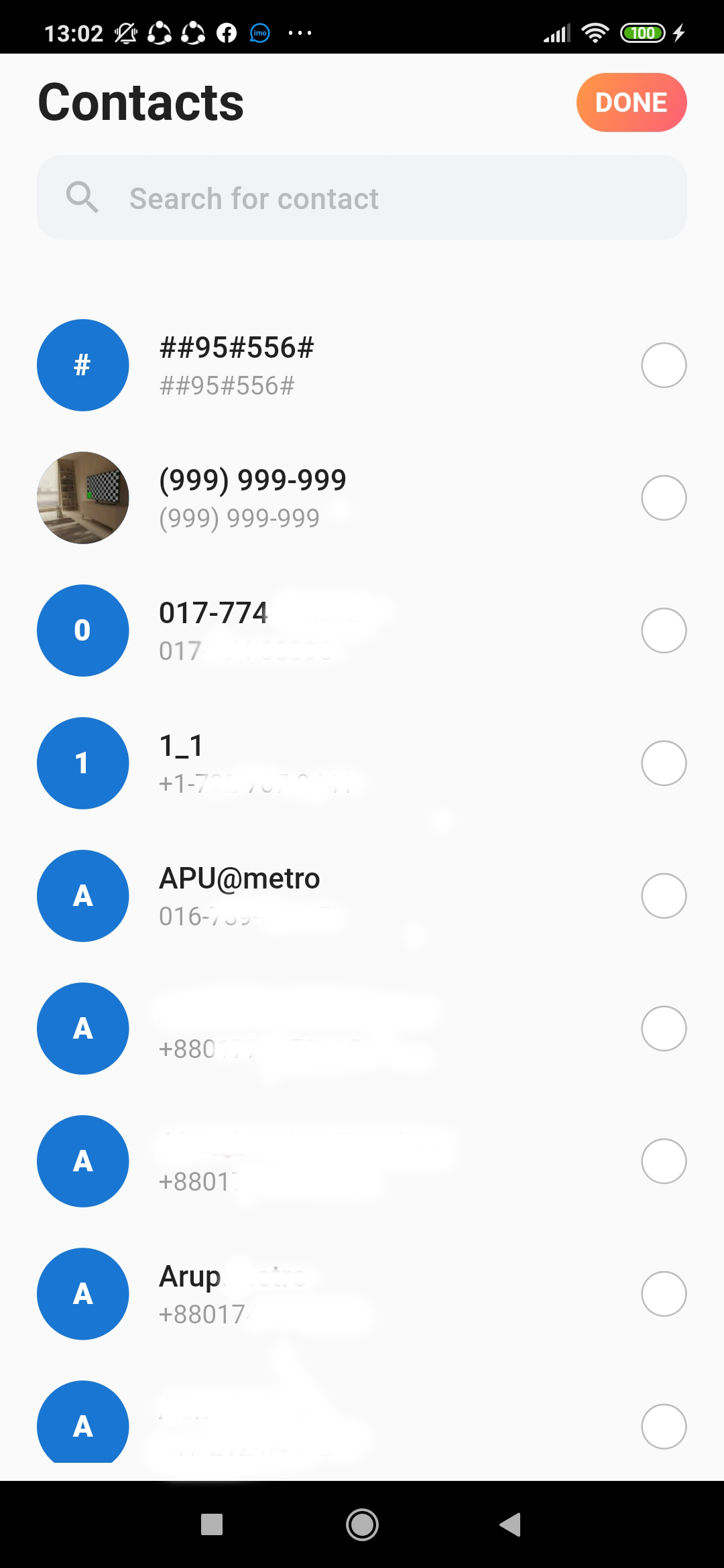
Development Process:

The SetTrustedContacts class controls this screen.

**‌ ‌** SET TRUSTED CONTACTS ‌ button uses roundedRectButton widget from round\_gradient\_button.dart file with given parameter(‘fromSetTrustedContactsButton’).

##### Contacts

Shows all available contacts from the phone's contact list allowing the user to search and select from available contacts.At least 3 contacts must be chosen.



Development Process:

The ContactList class controls this screen.

getContactsPermission() -> this function request for permission to access device contacts.

\_buildListTile() -> this function builds single contact info UI.

refreshContacts() -> this function refresh or get all contacts and populate all contacts in the UI.

\_populateContacts() -> this function populates all contacts in the UI.

getSingleContact() -> this function gets single contact info from local database.

getSearchBarUI() -> this widget search contacts from available contact list.

CustomContact class -> this class model to check/select contact from contact list to set as trusted contacts.

doneRoundedRectButton() -> this widget adds all selected contacts to trusted contacts and navigates users to SelectedTrustedContacts class.

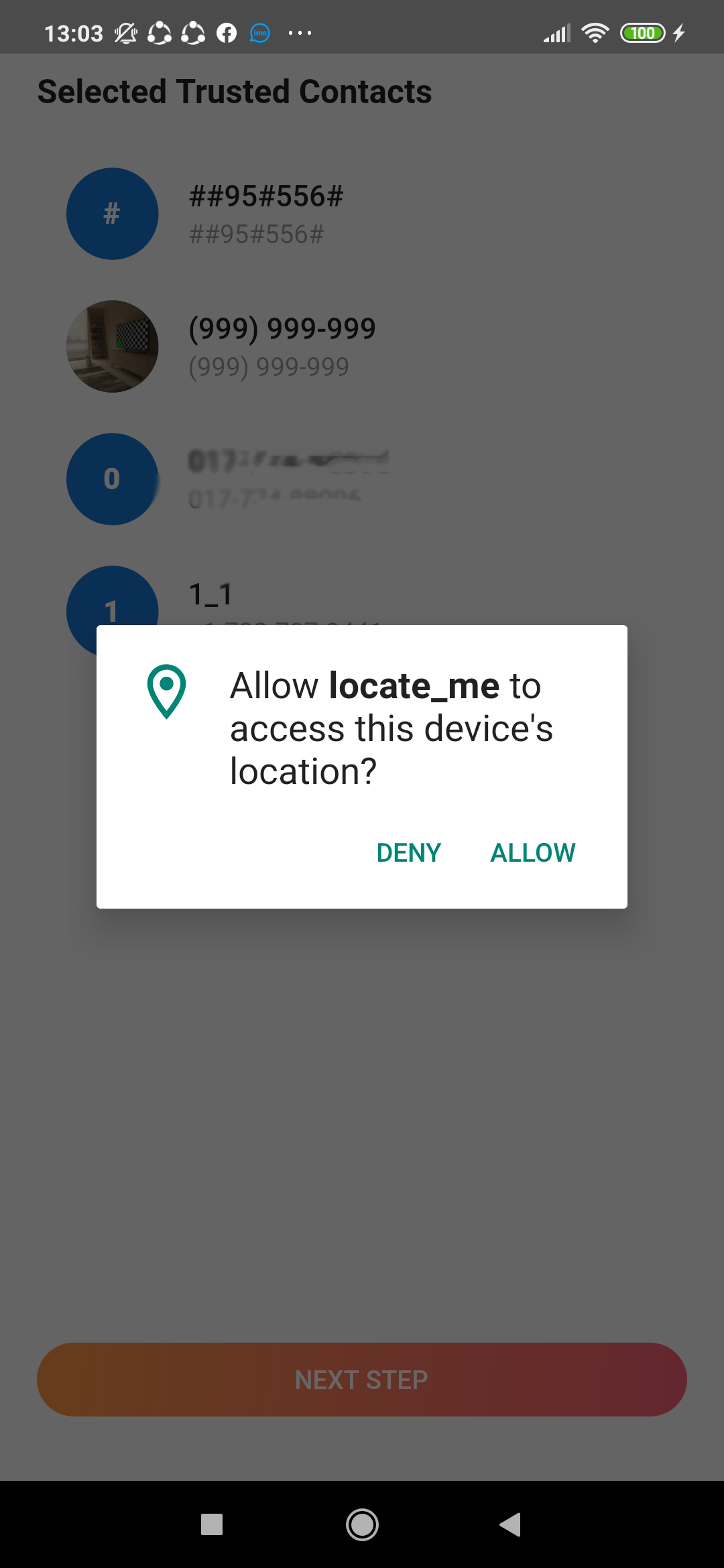
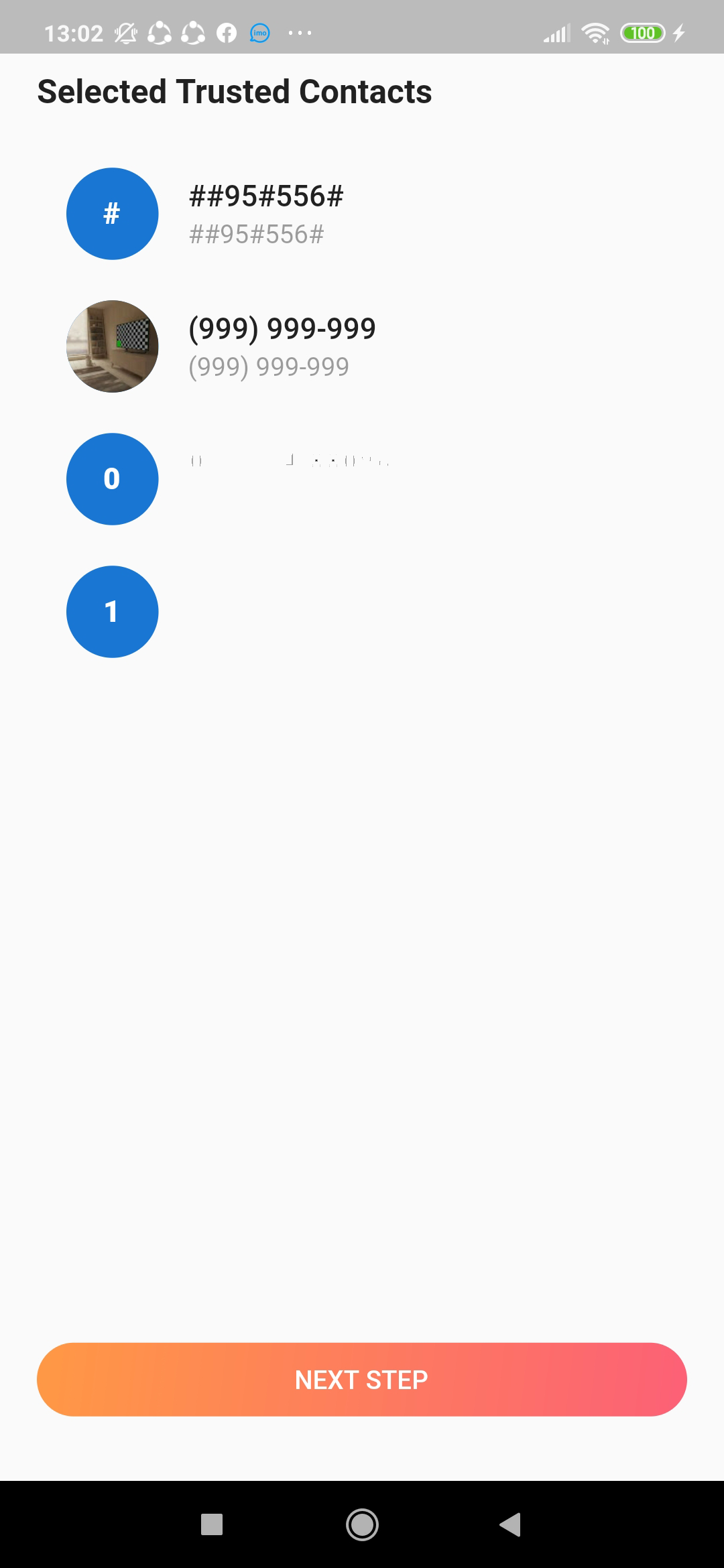
contactsLeftDialog() -> this function shows dialog to show the minimum number of contacts needed to be selected to set as trusted contacts.

ButtonCheck class -> Checks and selects contacts that are set as trusted contacts.

checkIfContains() -> checks if contact given as parameter to this function is already selected or not.

##### Selected Trusted Contacts

Shows all the selected trusted contacts user chosen.This page will be the first screen from the second time the user enters the app. On tap of NEXT STEP button, a dialog will appear prompting the user to give access to the device’s location.



Development Process:

The SelectedTrustedContacts class controls this screen.

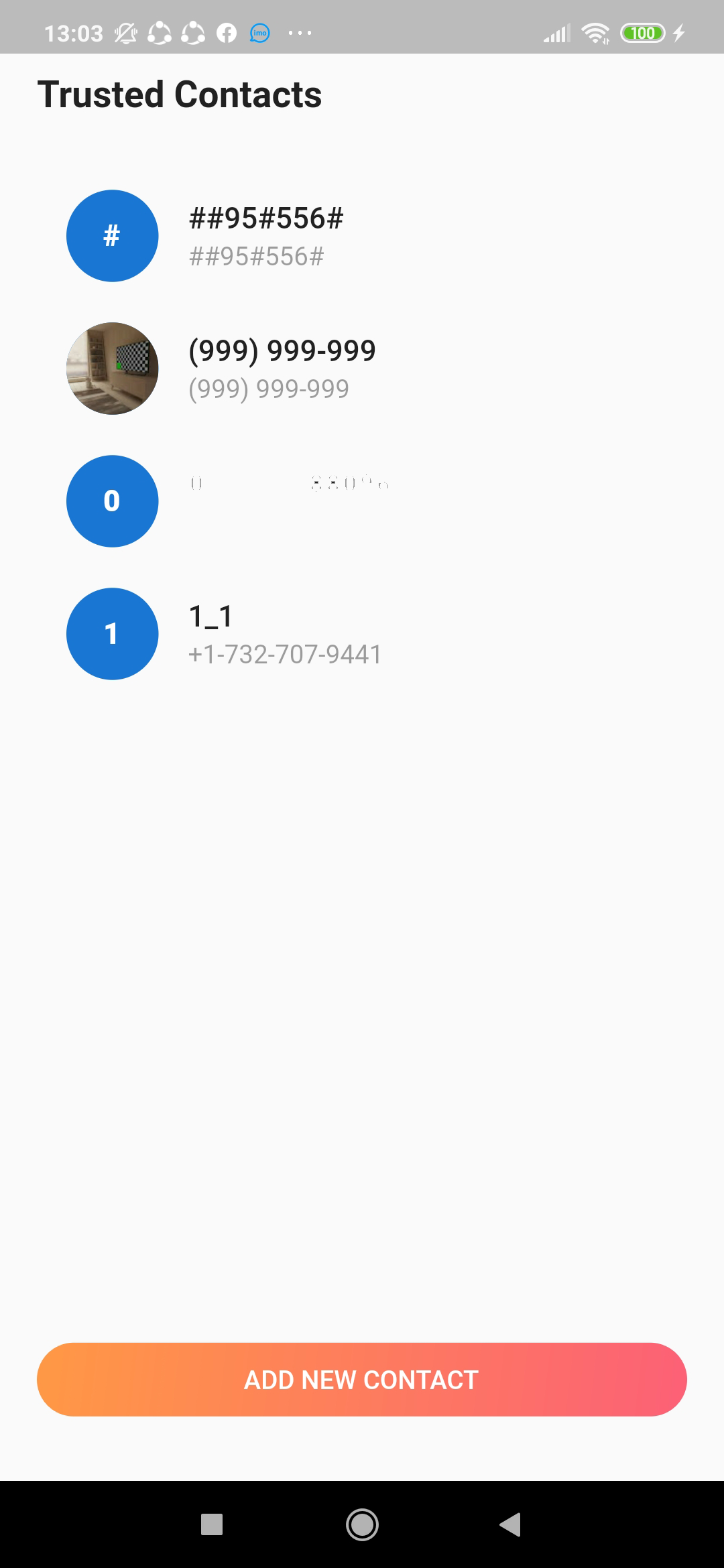
**NEXT STEP** ‌ button uses roundedRectButton widget from round\_gradient\_button.dart file with given parameter('fromSelectedTrustedContacts') request location permission from user.

getLocationPermission() -> this function from round\_gradient\_button.dart file request user to access device’s location.

TurnOnLocationAccess class -> from turn\_on\_location\_access.dart file is a UI class that takes permission from the user to enable location access to share their exact location.

##### Trusted Contacts

Shows all contacts set as trusted contacts to send the device's location in emergency. **ADD NEW CONTACT** button allows the user to add more contacts on trusted contacts list.



Development Process:

The TrustedContacts class controls this screen.

didChangeAppLifecycleState() -> called when the system puts the app in the background or returns the app to the foreground with its certain properties.

allContacts() -> function get all contacts from local database

startTimer() -> Countdown timer or stopwatch (given 10 seconds cycle)

startServiceInPlatform() -> puts the app in background

getDialogStatus() -> change dialog status and show myAlertDialog.

yAlertDialog() -> shows dialog notifying user that he/she is about to send message about his/her whereabouts with the option to cancel it

MyCustomDialog -> class is the UI of dialog that shows user that he/she is about to send message about his/her whereabouts with the option to cancel it

sendSmsToContact() -> function from alertDialog.dart file checks/gets location permission from user and sends sms to trusted contacts with location info.

##### Location

Shows the live location of the user using google map.



Development Process:

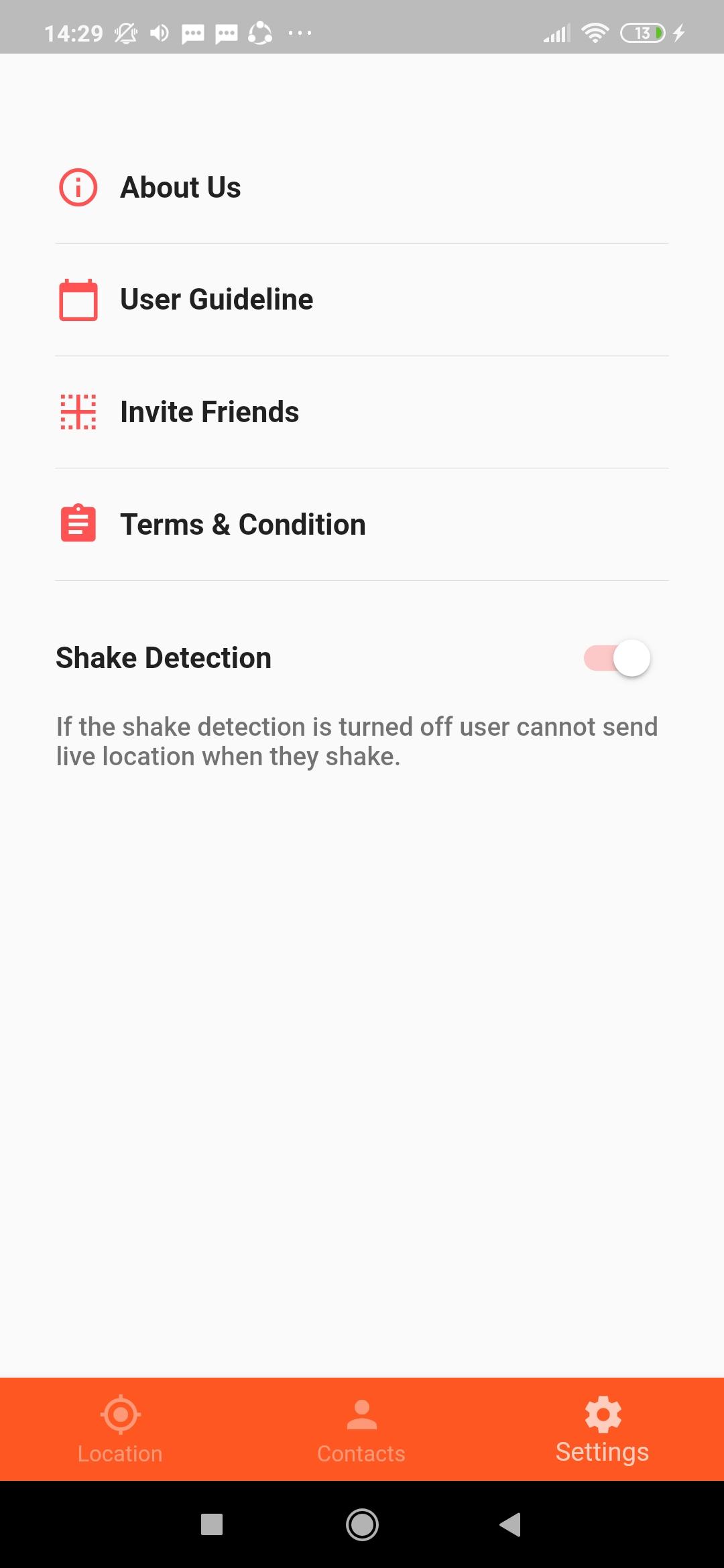
The MapSample class mainly controls this screen and the TrustedContacts class also partially(small segment)) controls this screen.

currentLocation() -> function gets/checks location access permission from the user and gets their current location.

##### Settings

The shake detection setting is the most important settings option here. Users can control the App functionality with this. If the shake detection is disabled the app will not awake and not prompt to automatically send the SMS on shaking.

The User Guideline settings option describes the use of this Application. There are also the Invite Friend option to share this app with others and some basic options such as About Us, Terms & conditions.



Development Process:

Segment of TrustedContacts class mainly controls this screen.

settings()-> is the main widget that controls this screen.

# 5. Platform

Targeted platforms are Android & iOS.

The target API level is 28 and the minimum supported API level is 16 for the Android platform.

# 6. Conclusion

The “Locate Me” app is developed in a functional way and it serves its job very well. Moreover, it can be used in real life and it helps people in an emergency situation.