

PROJECT ABSTRACT

Personal Safety SOS App with Live Location Sharing

A personal-safety Android app: press-and-hold SOS, live location broadcast to a trusted circle, per-contact notification channels, check-in timers and safe places. Delivered built-to-order with Kotlin app source, alert pipeline, report, PPT and viva kit.

Title

Personal Safety SOS App with Live Location Sharing — an emergency UX built around one interaction: hold SOS, and your trusted circle gets your live location automatically.

Introduction / Background

In an emergency, nobody opens an app, navigates menus and types a message. Safety apps fail when they ask for fine motor control and attention under stress — the correct interface is one big button and everything else automatic. This project builds that: a press-and-hold SOS that broadcasts live location to a trusted circle every 15 seconds, notifies each contact through their configured channel (SMS, push, phone call), opens an audio channel, and keeps running with the screen locked through a foreground service. A check-in timer covers the quieter case: set it when heading out, and a missed check-in raises the same alert with the last known location automatically.

Problem Statement

Most safety apps are contact lists with a panic button that needs the screen on, the app foregrounded and several taps — exactly what fails under stress. Location sharing in maps apps is always-on (a privacy cost) and requires contacts to use the same app. The project addresses this with an event-driven design: nothing is shared until SOS or a missed check-in; the alert survives screen lock via a foreground service; and each contact gets their own channel (SMS/call/push) so they need no app installed.

Objectives

- Implement press-and-hold SOS (2 seconds, configurable) raisable from a lock-screen widget.
- Broadcast live location every 15 seconds to the trusted circle for the alert duration.
- Dispatch per-contact notification channels: SMS, push notification, phone call, or combinations.
- Run the alert as a foreground service with a persistent notification, surviving screen lock.
- Provide check-in timers (15 min–12 h) that auto-raise alerts with the last known location on miss.
- Show per-contact delivery confirmations on the active-alert screen.
- Support saved safe places with arrival detection that can auto-resolve a timer.

Proposed Methodology

The alert is modelled as a session: SOS hold starts a foreground service that captures the location, creates the alert record in Room, and fans out notifications per contact channel via FCM and the SMS Manager API. A location loop posts fixes every 15 seconds to the trusted circle's live-tracking view. The check-in timer runs as a scheduled alarm with a grace-window reminder; expiry fires the same pipeline with the last known fix. Cancellation requires a confirmation step and notifies the circle.

System Architecture / Working

The Kotlin app has the SOS UI (hold-to-confirm button, alert screen, circle management), the alert service (foreground service, location loop, channel dispatch), and the data layer (Room for contacts, timers, alert history). FCM carries push; SMS Manager is the carrier fallback; the live-tracking view renders on Google Maps SDK for contacts opening the alert link. The demo documents staged alert drills so the flow is testable without a real emergency.

Software Requirements

- Kotlin (Android app)
- Foreground Service (lock-screen-resilient alerts)
- Fused Location Provider (location fixes)
- Firebase Cloud Messaging (push notifications)
- SMS Manager API (fallback SMS alerts)
- Room database (contacts, timers, alert history)
- Google Maps SDK (live tracking view)
- Material 3 UI components

Expected Outcomes

- A working SOS flow: hold → alert → live location to the circle with delivery confirmations.
- Check-in timers that escalate a miss into a full alert automatically.
- A documented test plan of staged drills covering each channel.
- A complete report, PPT and viva Q&A; (foreground services, location APIs, FCM, SMS fallbacks).

Applications

Student safety (especially hostel commutes), solo travellers, field workers and elderly care; the alert-pipeline pattern adapts to lone-worker industrial safety.

Future Scope

Automatic crash/fall detection via sensors, direct emergency-service dialling options, wearable companion trigger, and iOS counterpart app.