GAYATRI VIDYA PARISHAD COLLEGE FOR DEGREE AND P.G. COURSES (A)
RUSHIKONDA, VISAKHAPATNAM 530045 | website: www.gvpdcpgc.edu.in
(Approved by A.I.C.T.E | Affiliated to Andhra University | An ISO 9001:2015 Certified Institute)
ENGINEERING AND TECHNOLOGY PROGRAM
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DESIGN THINKING & INNOVATION

REPORT

RAKSHA- Alert

Smart Emergency Response & Safety Application

TEAM MEMBER

[5231411193] – [Chaitanya Kumar Sahu]

Academic Year 2025-26

A project Report On

RAKSHA – Smart Emergency Response & Safety Application

Submitted under **DESIGN THINKING AND INNOVATION (DTI)**

In

COMPUTER SCIENCE AND ENGINEERING

By

CHAITANYA KUMAR SAHU (5231411193)

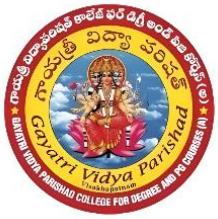
Course: Design Thinking and Innovation
Course code: 20CCT203
Academic Year : 2025-2026

Under the esteemed guidance of

T.SRIKRISHNA, M.Tech

ASST.PROFESSOR

GAYATRI VIDYA PARISHAD COLLEGE FOR DEGREE AND P.G. COURSES (A)
RUSHIKONDA, VISAKHAPATNAM 530045 | website: www.gvpdcpgc.edu.in
(Approved by A.I.C.T.E | Affiliated to Andhra University | An ISO 9001:2015 Certified Institute)
ENGINEERING AND TECHNOLOGY PROGRAM
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that a Project Report entitled **" RAKSHA – Smart Emergency Response & Safety Application**" is a Bonafide work done by **CHAITANYA KUMAR SAHU(5231411193)** under the guidance and supervision of **T. SRIKRISHNA, ASST.PROFESSOR** in **DESGIN THINKING AND INNOVATION** in **COMPUTER SCIENCE AND ENGINEERING** of **GAYATRI VIDYA PARISHAD COLLEGE FOR DEGREE AND P.G. COURSES (A)** during the academic year 2025-2026.

Project Guide
T.Srikrishna, M.Tech
Assistant Professor.

Head of The Department
Dr.G.R.S Murthy, M.Tech,Ph.D.
Professor.

External Examiner

GAYATRI VIDYA PARISHAD COLLEGE FOR DEGREE AND P.G. COURSES (A)
RUSHIKONDA, VISAKHAPATNAM 530045 | website: www.gvpdcpgc.edu.in
(Approved by A.I.C.T.E | Affiliated to Andhra University | An ISO 9001:2015 Certified Institute)
ENGINEERING AND TECHNOLOGY PROGRAM
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACKNOWLEDGEMENT

We express our sincere thanks wherever it is due

We express our sincere gratitude to the Management and Principal of Gayatri Vidya Parishad College for Degree & P.G. Courses for providing us with the necessary infrastructure, resources, and environment to carry out this project successfully.

We are deeply thankful to our **Head of the Department, Dr.G.R.S.Murthy,** for the constant support, encouragement, and guidance throughout the project.

Our heartfelt thanks go to our guide and Course Instructor**, T.Srikrishna, Assistant Professor**, Department of CSE, for the invaluable mentoring, insightful suggestions, and motivation that made this project meaningful and complete.

We also extend our sincere thanks to all teaching staff, lab technicians, and fellow students who participated in our user research sessions, surveys, and usability testing. Their honest feedback shaped the Raksha app into a truly user-centered solution.

NAME OF THE CANDIDATE
5231411193 – Chaitanya Kumar Sahu

TABLE OF CONTENTS

1.PROBLEM STATEMENT

In today's world, ensuring personal safety and responding quickly to missing person cases has become a major challenge. Many individuals such as children, elderly people, travelers, and accident victims often go missing due to various reasons such as getting lost, medical emergencies, accidents, kidnapping, or lack of communication. The biggest problem in missing person cases is the delay in spreading information to the right people, especially those who are physically near the last known location of the missing person. Traditional methods such as police complaints, social media posts, and public announcements take a long time to reach people, which reduces the chances of finding the missing person quickly.

At the same time, in emergency situations such as physical attacks, health emergencies, or dangerous situations, a person may not have enough time to make a phone call or explain their situation. Many existing safety applications only send alerts to pre-saved contacts, but they do not directly notify nearby police stations or authorities, which increases emergency response time.

Therefore, there is a need for a smart, real-time safety and missing person response system that not only helps during emergencies through an SOS feature but also helps in quickly locating missing persons by alerting nearby people and authorities based on location.

The Raksha app is designed to solve this problem by providing a comprehensive safety system that includes:

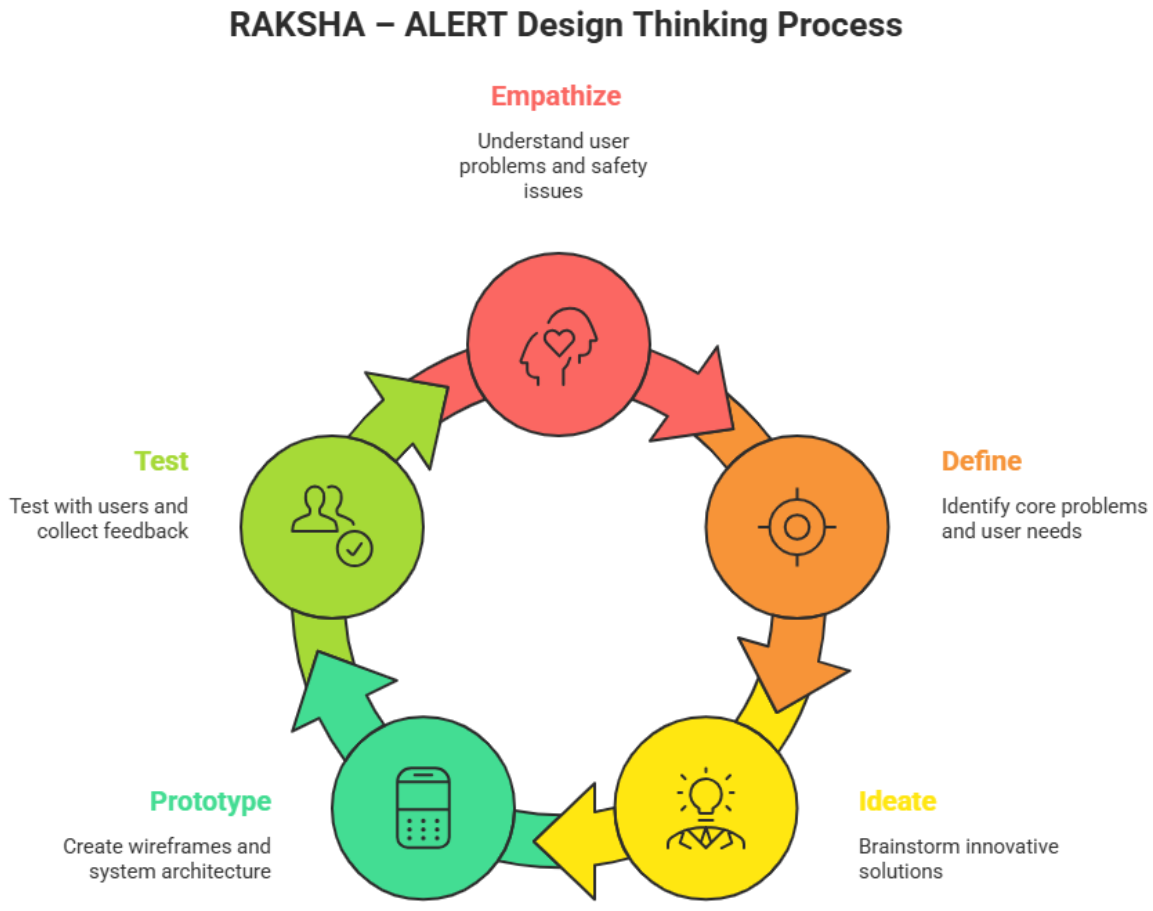
- A Missing Person Alert System that sends alerts to users near the last known location of the missing person
- Live location tracking and last known location sharing
- A feature for nearby users to report sightings of the missing person
- Photo and description sharing of the missing person
- An SOS emergency button that sends immediate alerts to the nearest police station along with the live location of the user
- SMS-based alert system when internet connectivity is not available
- An admin dashboard for authorities to monitor alerts and manage cases

By combining community participation, real-time location tracking, and direct police alert through SOS, the Raksha app reduces response time, improves search efficiency, and increases the chances of rescue and recovery in both emergency and missing person situations.

"How might we design a smart, real-time emergency response system that empowers vulnerable individuals to get help faster, from both their trusted contacts and nearby strangers, with automatic evidence collection?"

2. PHASES OF DESIGN THINKING

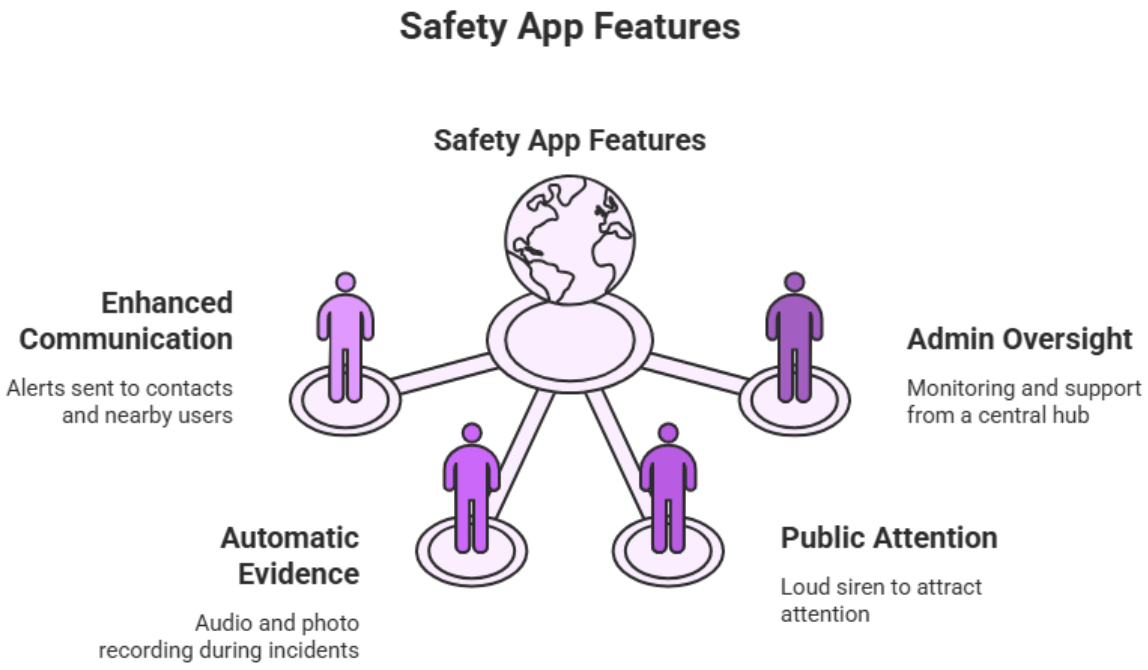
Design Thinking is a human-centered, iterative approach to problem-solving that focuses on understanding user needs, redefining problems, and creating innovative solutions. It is widely used in product development, app development, and service design because it prioritizes the user's problems and experiences rather than just technical solutions. The Design Thinking process helps in developing practical and user-friendly solutions by breaking the process into five important phases: Empathize, Define, Ideate, Prototype, and Test. For the Raksha app, these five phases helped in identifying real-world safety and missing person problems and developing a solution that is practical, fast, and useful in emergency situations.



2.1 EMPATHIZE

Empathy is the foundation of the Design Thinking process. In this phase, we immersed ourselves in the real experiences, emotions, and challenges of our target users — primarily women, students, elderly individuals, and people who travel alone. We aimed to deeply understand what they feel, think, say, and do when they face or fear dangerous situations.

Empathy Map — Raksha App Users



2.1.1 Brainstorming (Explore & Exploit)

To kickstart the empathy phase, we conducted intensive brainstorming sessions using the Explore–Exploit framework. This helped us tap into both creative ideation and focused refinement to deeply understand the problem space around personal safety and emergency response.

Explore Phase: Divergent Thinking

In this phase, our goal was to generate as many raw ideas and questions as possible, without judgment. We asked broad, open questions:

- Why are people unable to call for help during real emergencies?
- What stops bystanders from helping someone in danger?
- What existing safety features are people aware of but don't use, and why?

- How does the problem differ for women vs. elderly vs. students?
- What role can audio and visual evidence play in post-incident justice?

Key Insights from the Explore Phase:

- People freeze or panic during emergencies — apps with multiple steps are useless.
- Existing apps only notify saved contacts, missing the potential of nearby helpers.
- Evidence (audio, photos) is often lost — no app captures it automatically.
- Women traveling alone feel the most vulnerable at night and in isolated areas.
- Elderly users need extremely simple, large-button interfaces.

We generated 50+ raw pain points, ideas, and themes which were later grouped into core problem clusters.

Exploit Phase: Convergent Thinking

In the exploit phase, we filtered, clustered, and refined our findings to identify the most critical challenges to address.

THEME	REFINED INSIGHT
Panic-Induced Inaction	Users cannot navigate complex steps — a single-tap SOS is critical.
Limited Alert Reach	Alerts only to contacts — nearby users who can help immediately are ignored.
Evidence Loss	No automatic recording means incidents go undocumented.
Slow Response Times	Emergency services and contacts take too long — on-ground help is needed.
Offline Limitations	Internet dependency fails in remote or network-dead areas — SMS fallback is essential.

Tools Used During Brainstorming:

- Whiteboard / Sticky Note Jam — to get spontaneous ideas from all team members
- Miro Board — for digital idea mapping and clustering
- Affinity Mapping — to group related pain points and identify core themes

2.1.2 List of Stakeholders / Customers

Identifying the right stakeholders is critical to building a meaningful and scalable solution. The Raksha app's ecosystem involves a wide range of users, influencers, decision-makers, and enablers.

TYPE	STAKEHOLDER	DESCRIPTION
Primary	Women (Students, Working, Homemakers)	Core users who face safety risks — most likely to use SOS features.
Primary	Students & Young Adults	Frequent solo travelers who need instant emergency support.
Primary	Elderly People	Need simplified emergency buttons with minimal UI complexity.
Primary	Children & Minors	Need SOS features with strong guardian monitoring.
Primary	Solo Travelers	High-risk users in unfamiliar locations who need location sharing.
Secondary	Parents & Guardians	Receive alerts; monitor loved ones' location in real time.
Secondary	Emergency Contacts	Family and friends notified during SOS events.
Secondary	Nearby Raksha Users	Alerted to provide immediate on-ground assistance.
Tertiary	Police & Emergency Services	Admin-level monitoring of active cases and alerts.
Tertiary	App Administrator	Monitors dashboard, manages users, verifies emergencies.
Tertiary	Telecom / SMS Providers	Enable SMS fallback when internet is unavailable.

2.1.3 Customer Interaction

To ensure Raksha addresses real-world pain points, we directly engaged with our target users through multiple interaction methods. These interactions helped us validate assumptions, uncover unmet needs, and understand user behavior in the context of personal safety.

Interaction Objectives

- Understand how people currently handle safety threats (calls, apps, screaming, etc.)
- Identify the exact moment they feel most unsafe and what they need then
- Learn what features they would actually use in a panic situation
- Discover why existing safety apps are underused or ineffective
- Understand the needs of guardians who monitor family members

STAKEHOLDER GROUP	METHOD USED	DESCRIPTION
Women (Age 18-40)	User Interviews	Conducted 1-on-1 conversations with 20 women about safety experiences and app usage.
College Students	Group Discussion	Discussed safety concerns and ideal features in group brainstorming sessions.

Elderly Users	Observation	Observed how elderly users interact with smartphones under stress.
Guardians & Parents	Structured Survey	Online survey on what monitoring features they want for their family.
Police Personnel	Informal Interview	Gathered insights on evidence needs and response time challenges.

Sample Questions Asked

- Have you ever felt unsafe and wished you could call for help faster?
- What stops you from using safety apps in a real emergency?
- Would you feel more confident if nearby strangers were also alerted?
- What is the most important feature a safety app should have?
- How comfortable are you with automatic audio/photo recording during emergencies?

#	KEY INSIGHT OBSERVATION
1	85% of women reported feeling unsafe while traveling alone at night.
2	Most users said they would panic and forget how to use an app — a single button is essential.
3	Users strongly preferred nearby user alerts over just family notifications.
4	Audio recording was seen as a critical feature for evidence and deterrence.
5	Elderly users found most safety apps too complex — they need minimal, large-button interfaces.
6	76% of guardians wanted real-time location tracking, not just a one-time alert.

2.1.4 Tools and Strategies Used

In the Empathy phase, we employed a mix of qualitative, observational, and analytical tools to gather user-centric insights. These tools helped us develop a deep understanding of users' fears, needs, and behaviours when dealing with safety emergencies.

1. User Interviews

We conducted 20+ one-on-one interviews with women, students, elderly individuals, and guardians. These structured yet conversational sessions allowed us to dive deep into users' lived experiences with safety threats. Interviews revealed critical insights such as the extreme importance of speed (single-tap action), the high value placed on nearby user alerts, and strong support for automatic evidence recording. These interviews shaped the core feature set of Raksha.

2. Observation (Shadowing / Fly-on-the-Wall)

We observed users in real settings — late-night bus stops, college campuses, and empty streets — to understand how they naturally behave when feeling unsafe. Observation revealed that people

often don't reach for their phones first; they look for people nearby. This directly inspired the 'Nearby User Alert' feature. We also observed that women check their phone screens very quickly, reinforcing the need for a large, instantly visible SOS button.

3. Empathy Maps

We created detailed empathy maps for three user personas — a college-going woman, an elderly individual, and a guardian — capturing what each user thinks, feels, hears, sees, says, and does when dealing with safety concerns. These maps helped identify emotional triggers and hidden anxieties, such as fear of not being believed, shame about causing a 'false alarm,' and anxiety about evidence. These insights directly shaped features like Audience Receipt (showing who received the alert) and the Case Reporting System.

4. Surveys & Questionnaires

We distributed a Google Form survey that reached 100+ users from different demographics. The survey included rating-scale and open-ended questions to gather broader trend data.

Key statistics revealed:

- 82% felt existing safety apps were too slow or complex to use in real emergencies.
- 78% would use a safety app that also notified nearby users — not just their contacts.
- 71% considered automatic audio recording the most important feature.
- Only 23% had ever used a safety app during an actual emergency.

5. Stakeholder Maps

A stakeholder map was created to visualize everyone impacted by or contributing to the Raksha ecosystem. This included primary users (at-risk individuals), secondary stakeholders (guardians, nearby users), and tertiary enablers (admin, police, SMS providers). This mapping helped us design with the entire ecosystem in mind — for example, ensuring the admin dashboard gives police actionable location data, and that guardians receive clear, non-alarming but urgent notifications.

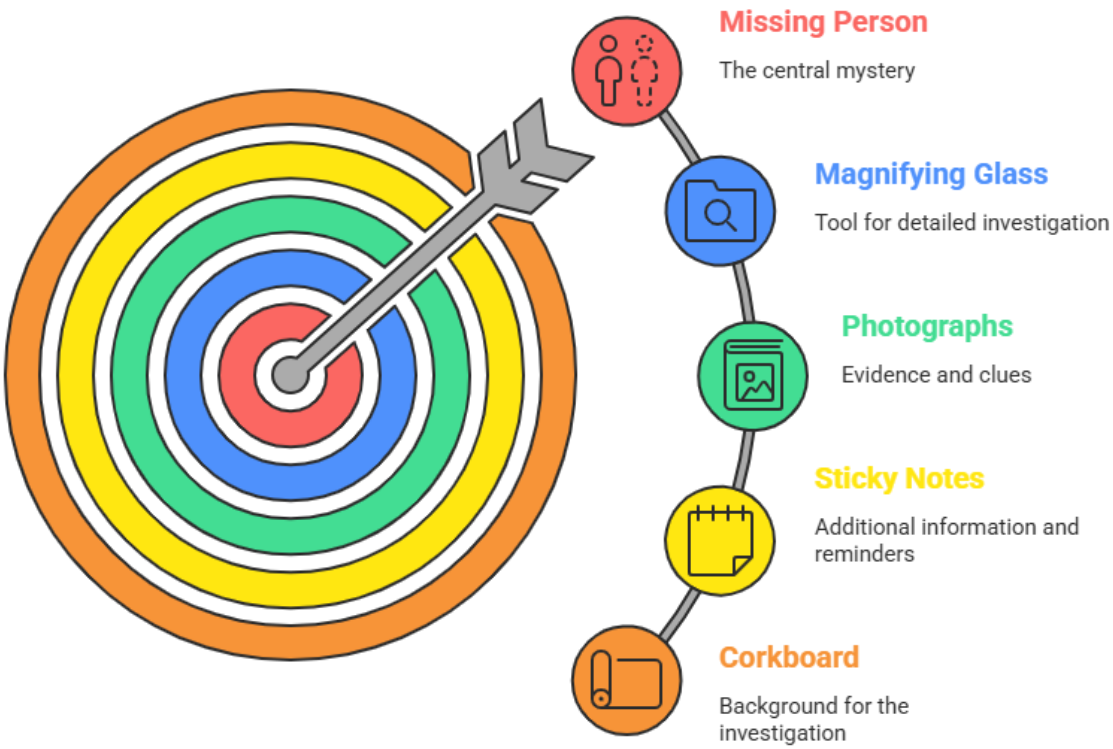
6. Persons

Based on interviews and observations, we developed three detailed user personas:

- Priya, the Solo Commuter (22, College Student): Travels late at night, fears being followed, needs instant one-tap help with automatic recording.
- Raju, the Concerned Parent (48, Guardian): Wants to monitor his daughter's location in real time and receive instant alerts with her exact position.
- Savitri, the Elderly Resident (65, Retired): Lives alone, has arthritis, needs an extremely simple large SOS button with no complex navigation.

Each persona included background, goals, pain points, tech comfort level, and behavioural traits. These personas drove key design decisions — from the large red SOS button to the SMS fall back for low-network areas and the simplified UI for elderly users.

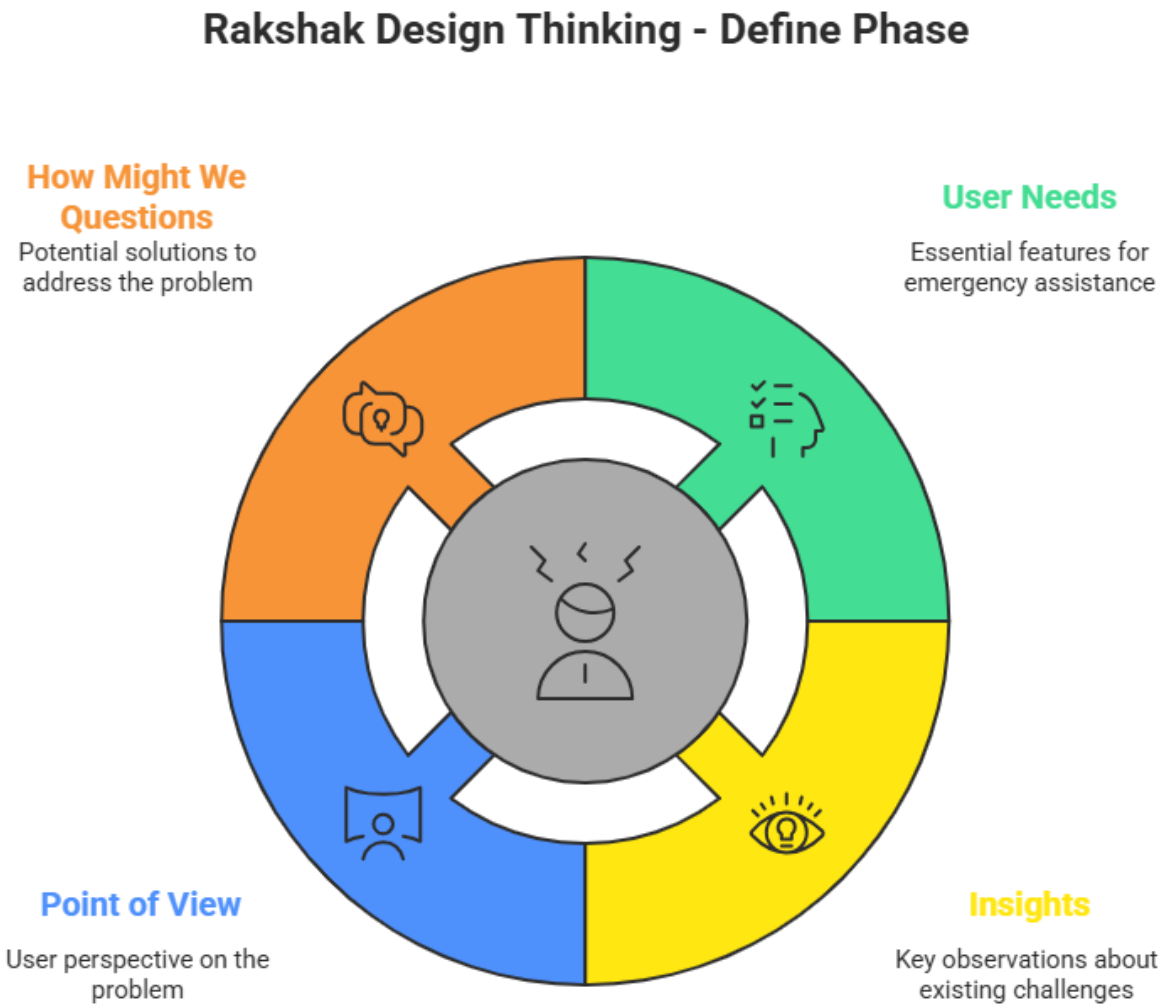
Missing Person Investigation Scene



2.2 DEFINE

Rakshak is a smart safety and missing person alert system designed to provide quick assistance during emergency situations and help locate missing individuals faster. The application uses real-time location tracking to send alerts to nearby users, guardians, and the nearest police station when an SOS is triggered or when a missing person is reported. The main goal of the app is to reduce response time by informing the right people who are closest to the location and can provide immediate help.

The Rakshak app combines features such as SOS emergency alerts, missing person reporting, live location sharing, nearby user notifications, and SMS alerts when internet connectivity is unavailable. It also includes an admin dashboard for authorities to monitor and manage cases. By connecting the community, emergency contacts, and police through a single platform, Rakshak improves emergency response and increases the chances of safely recovering missing persons.



2.2.1 Problem Trees

A Problem Tree was used to analyze the root causes and consequences of the central problem. This method helped us uncover the hidden systemic layers behind emergency response failures.

Central Problem: People in danger cannot get fast, effective help.

Root Causes:

- Panic and physical incapacitation prevent users from making calls or navigating apps.
- Existing apps only notify saved contacts — nearby helpers are never alerted.
- No automatic evidence capture means incidents are difficult to prove legally.
- SMS fallback is absent in most apps — network failures make them useless.
- Complex UIs designed for normal-state use are impossible in high-stress situations.

Effects / Consequences:

- High emergency response time — critical 'golden minutes' are lost.
- Physical harm to victims due to absence of timely help.
- Loss of evidence — attackers escape without accountability.
- Psychological trauma from the helplessness of being in danger with no response.
- Erosion of trust in safety technology — users abandon apps after failures.

Reflection: The problem tree revealed that Raksha must intervene at multiple points — at the UX level (simplicity), at the alert level (nearby users), at the evidence level (automatic capture), and at the infrastructure level (SMS fallback). A single-feature solution would be insufficient.

2.3 IDEATE

The Ideation phase is the creative core of Design Thinking. After empathizing with users and clearly defining the problem, we focused on generating the widest possible range of solutions — without judgment. Here, quantity leads quality in the beginning.

2.3.1 Technique Used: Mind Mapping

We used Mind Mapping to visually structure and organize ideas, branching out from the central concept — 'Smart Emergency Response App' — into features, user flows, and technical solutions.

The central node: "Raksha — Smart Personal Safety App"

Feature Ideas:

- One-tap SOS Button (no unlock needed)
- Live GPS Location Tracking (continuous, real-time)
- Nearby User Alert System (notifies registered users within radius)
- Auto Audio Recording (starts immediately on SOS)
- Auto Photo Capture (front/back camera)
- Loud Siren Alarm (attract bystanders)
- Fake Incoming Call Feature (escape dangerous situations discreetly)
- SMS Alert Fallback (works without internet)

Support Systems:

- Emergency Contacts Management (add/edit guardians)
- Admin Dashboard (real-time monitoring by authorities)
- Case Management System (track, report, and close cases)
- Audience Receipt (show who received and viewed the alert)

User Interface:

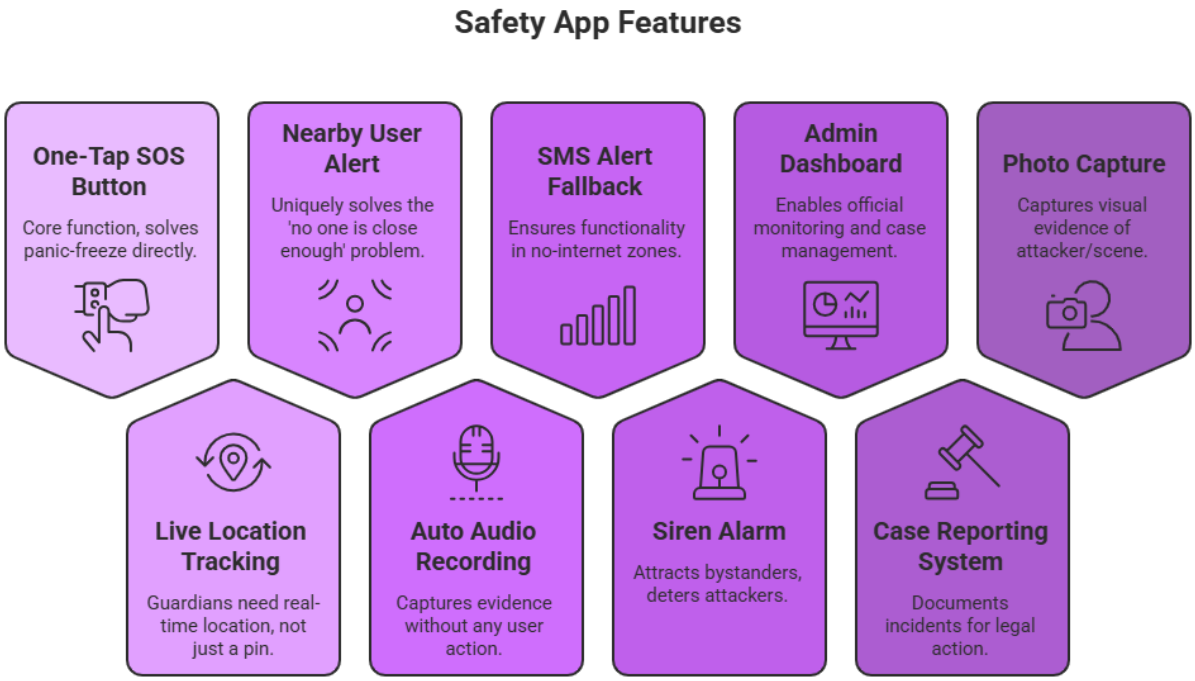
- Large central red SOS button (unmissable, no UI navigation required)
- Minimal home screen — only essential features visible
- High contrast design for visibility in dark conditions
- One-hand operation optimized layout

Advanced Features (Future):

- Wearable device integration (smartwatch SOS)
- AI threat detection (unusual microphone activity triggers alert)
- Community safety score for neighborhoods

2.3.2 Idea Selection and Shortlisting

After mind mapping, we narrowed down features using three filters: Feasibility (can we build it?), Desirability (do users actually want it?), and Viability (does it scale?).



2.4 PROTOTYPE

2.4.1 Method Used: Wireframes and System Architecture

The Prototype phase plays a critical role in converting conceptual ideas into a practical and testable solution. In the development of the Raksha Alert system, we adopted a dual approach consisting of User Interface (UI) wireframes and System Architecture design to ensure both usability and technical feasibility.

Initially, low-fidelity wireframes were created to visualize the layout and flow of the application. These wireframes included key screens such as Login, Dashboard, SOS interface, Map view, Alerts section, and Admin dashboard. The purpose of wireframing was to ensure that the application remains simple, intuitive, and easy to use, especially during emergency situations where users may be under stress.

The wireframes helped us validate important design decisions such as:

- Placement of the SOS button at the center for immediate access
- Minimal navigation to reduce user confusion
- High contrast color usage for better visibility
- Simple input forms for faster interaction

In addition to UI design, a System Architecture diagram was developed to represent how different components of the application interact with each other. The architecture includes the mobile application (frontend), Supabase backend (authentication and database), location services (GPS), SMS service (Fast2SMS), and real-time alert system.

This architecture helped us understand the complete data flow and ensured smooth communication between different modules of the system.

2.4.2 Prototype Screens and Explanation

The Raksha Alert application consists of multiple screens designed to provide a seamless and efficient user experience. Each screen serves a specific purpose in ensuring quick response during emergency situations.

- **Login / Signup Screen:**

  This screen allows users to securely register and log into the application. It includes validation mechanisms to prevent incorrect data entry and ensures authentication before accessing the system.

- **Home Screen (Dashboard with SOS Button):**
  This is the primary interface of the application. It contains a large, prominent SOS button placed at the center of the screen to enable one-tap emergency activation. Additional features such as reporting incidents and viewing alerts are also accessible.

- **SOS Activation Flow:**
  When the user presses the SOS button, the system immediately captures the user's real-time location using GPS services. The alert is then stored in the database and broadcasted to nearby users and administrators. This process is designed to work instantly without requiring additional user input.
- **Map Screen (Location Tracking):**
  The map screen displays the user's location and active alerts using OpenStreetMap integration. It helps responders and administrators track emergency locations visually.
- **Alerts Section:**
  This section shows active alerts generated by users or administrators. It provides real-time updates and ensures that users are aware of nearby incidents.
- **Incident Reporting Screen:**
  Users can report incidents by providing details and uploading images. This feature helps in documenting events and improving safety awareness.
- **Admin Dashboard:**
  The admin interface allows monitoring of all active SOS alerts, incidents, and user activities. It provides real-time synchronization and helps authorities respond quickly to emergencies.

2.4.3 System Workflow

The working of the Raksha Alert system follows a structured sequence to ensure fast and reliable emergency response:

1. User logs into the application
2. User presses the SOS button during an emergency
3. The application captures the user's GPS location
4. The data is stored in the Firebase database
5. Alerts are broadcasted to nearby users and administrators
6. SMS notifications are sent using Fast2SMS as a fallback
7. Admin dashboard receives real-time updates
8. Responders view alert details and provide assistance
9. The case is monitored and resolved

This workflow ensures minimal delay and maximum efficiency in handling emergency situations.

2.4.4 Design Considerations

While designing the prototype, special attention was given to usability and accessibility. Since the application is intended for emergency use, the design needed to be simple, fast, and reliable.

Key design considerations include:

- **Simplicity:** Minimal steps required to trigger an alert
- **Visibility:** Large SOS button with high contrast colors
- **Speed:** Instant response without loading delays
- **Accessibility:** Easy to use for all age groups, including elderly users
- **Reliability:** Works even in low network conditions using SMS fallback

These considerations ensured that the application remains effective even in high-pressure situations.

2.4.5 Prototype Validation

The prototype was tested with sample users to evaluate usability and performance. Feedback was collected to identify improvements in design and functionality.

Key observations include:

- Users preferred a single-tap SOS button over complex navigation
- High contrast UI improved visibility in low-light conditions
- Real-time alerts increased user confidence in safety features
- Simple layout reduced confusion during emergency situations

Based on this feedback, necessary improvements were made to enhance the overall user experience.

Prototype Screens

Figure 1: Login Screen

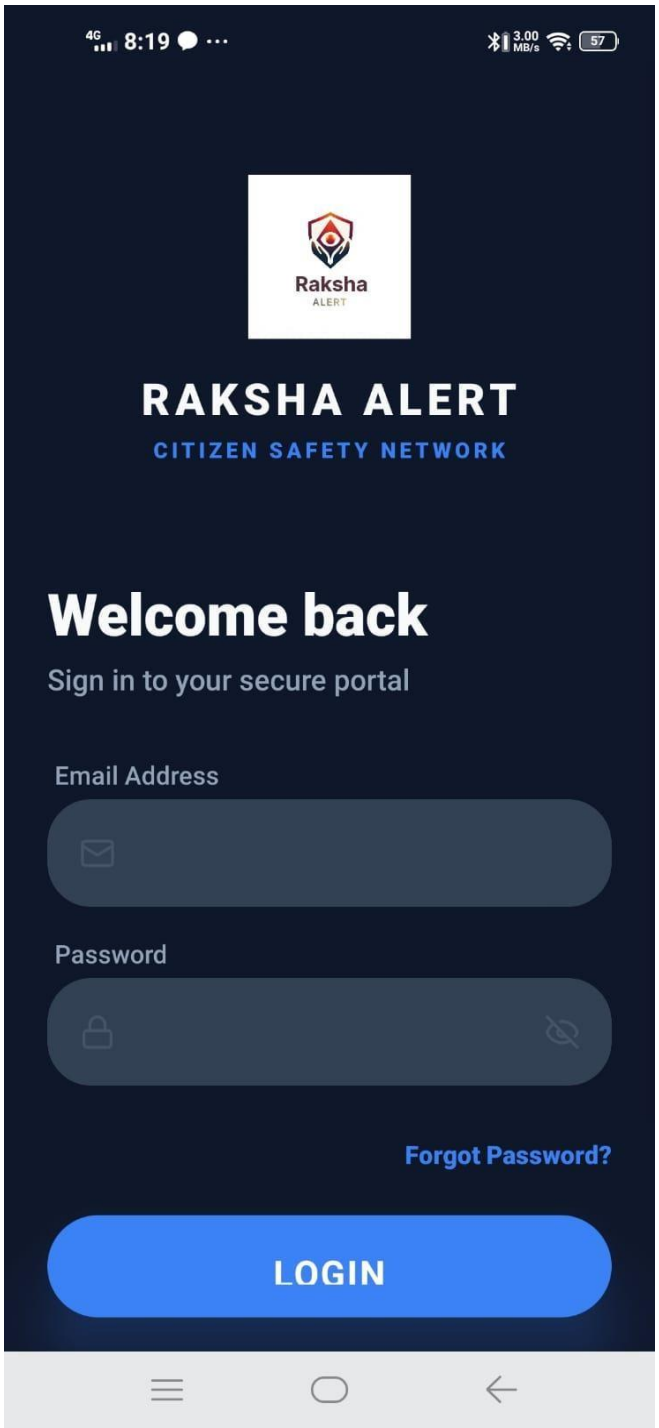


Figure 2: Dashboard with SOS Button

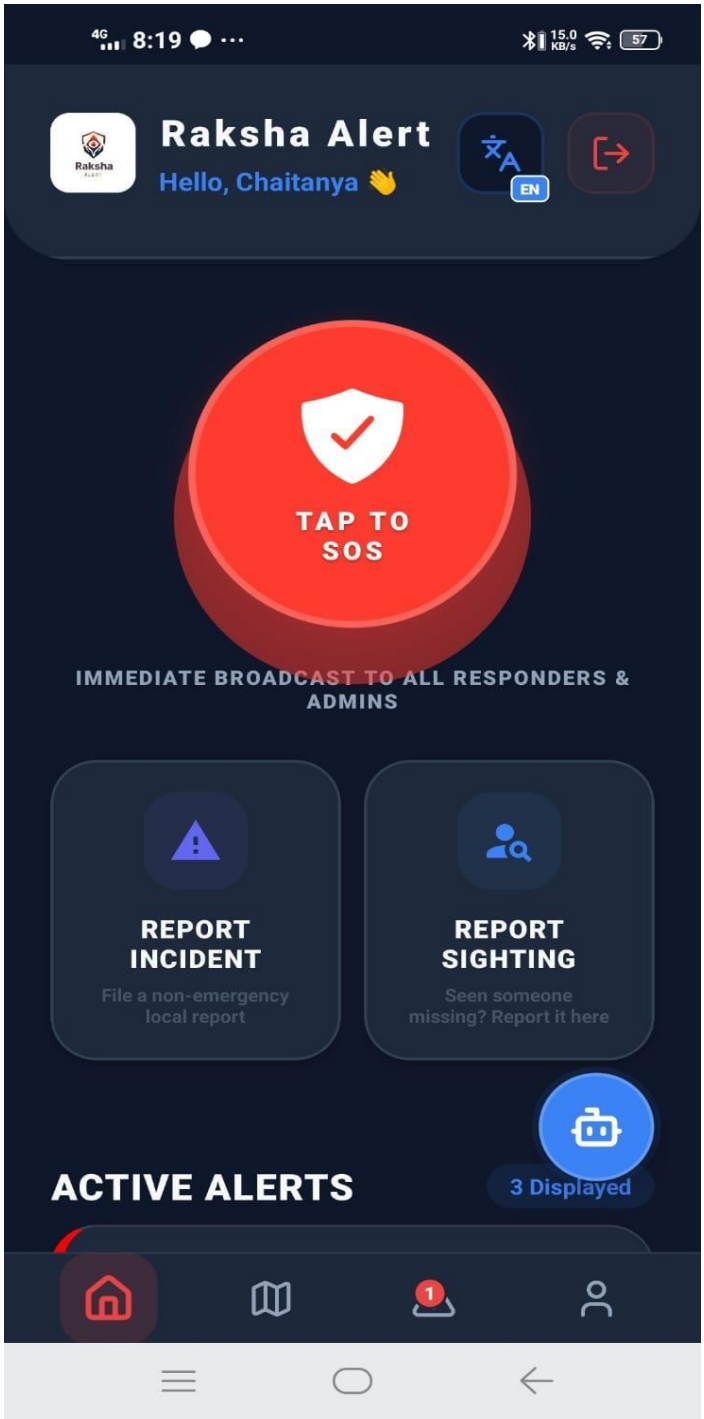


Figure 3: Admin Dashboard

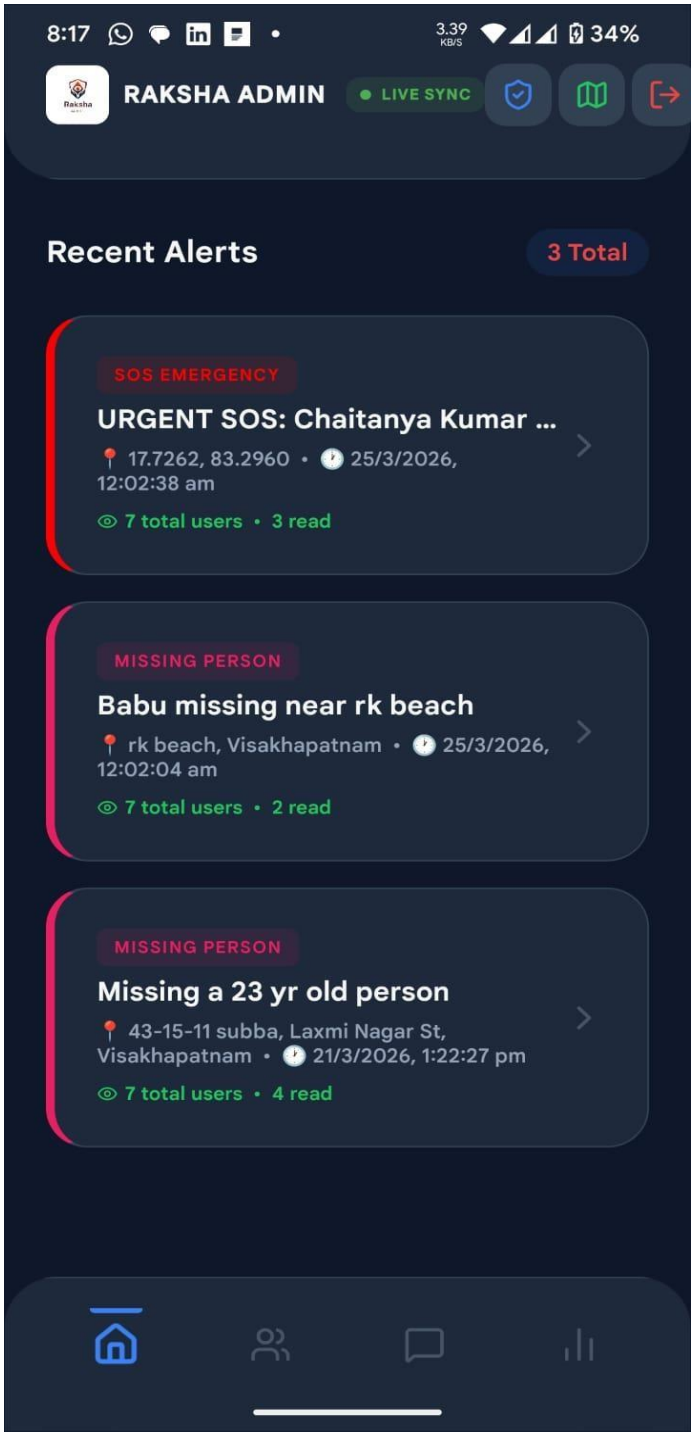


Figure 4: Alert Details Screen

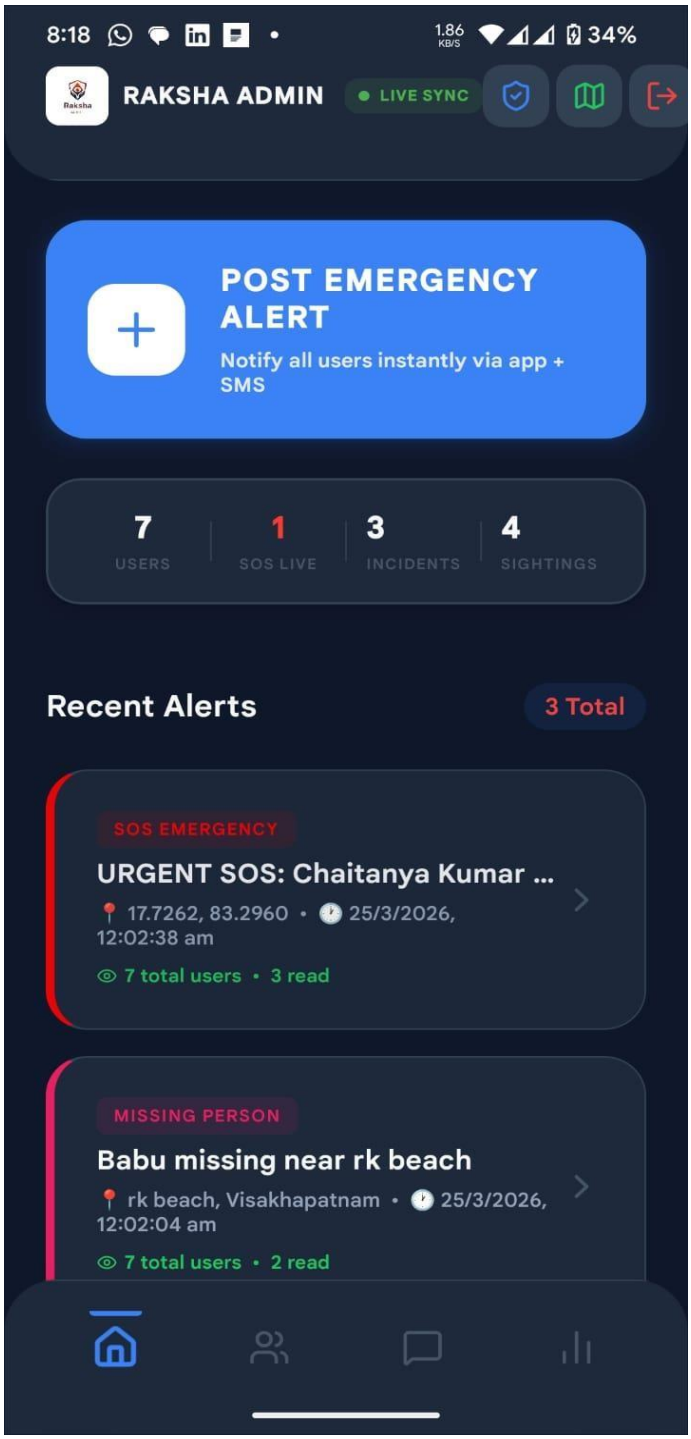
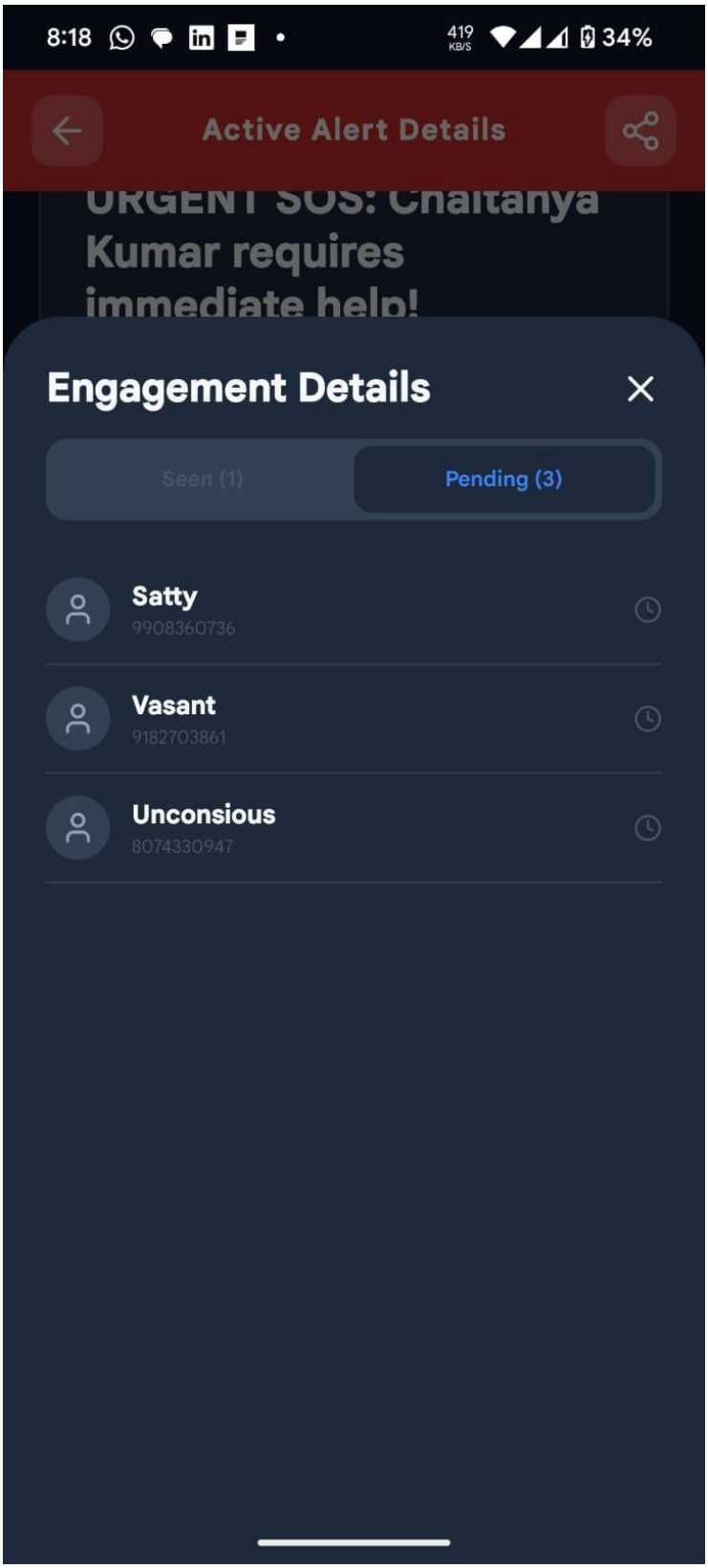


Figure 5: Incident Reporting Screen



2.5 TEST PHASE

Testing is a crucial phase in the Design Thinking process, where user feedback is used to validate assumptions, identify usability issues, and refine the system for real-world application. For the *Raksha Alert – Emergency Safety System*, both qualitative and quantitative evaluation methods were employed to assess the prototype's effectiveness, usability, and reliability.

2.5.1 Testing Techniques Used: Usability Testing and Feedback Surveys

1. Usability Testing (Moderated and Unmoderated)

The application was tested by a diverse group of users to ensure accessibility and ease of use across different demographics.

- **Participants:**
  18 users (12 students/women, 3 elderly individuals, 2 guardians, 1 admin)
- **Environment:**
  Simulated emergency scenarios conducted on mobile phones and tablets
- **Tasks Assigned:**
  - Navigate from the home screen and activate the SOS button
  - Verify alert transmission and confirmation screen
  - Add and edit emergency contacts
  - Access and use the Fake Call feature
  - (Admin) Monitor cases from the dashboard
- **Observation Metrics:**
  - Time taken to activate SOS (Target: < 3 seconds)
  - Number of incorrect taps before activation
  - Success rate of elderly users navigating without assistance

2. Feedback Surveys

- **Method:** Google Forms (post-testing session)
- **Total Responses:** 25
- **Evaluation Scale:** Likert scale (1–5)

Parameters Evaluated:

- Ease of Use
- Clarity of Interface

2.5.2 Observations and Improvements

Area Tested	Observation	Action Taken
SOS Button Size	Difficult for elderly users to press accurately	Increased button size by 30% and added haptic feedback
Alert Confirmation	Users unsure if SOS was activated	Added full-screen green flash and audio confirmation
Nearby User Feature	Users confused about "nearby network" concept	Introduced onboarding tooltip explanation
Fake Call Feature	Hard to find in settings	Moved to primary bottom navigation bar
Admin Dashboard	Case list overwhelming for admins	Added filters (Active, Resolved, False Alarm)
SMS Fallback	No indication when offline mode activated	Added "SMS Mode Active – No Internet" banner

2.5.3 Technical Validation (System Testing)

Functional testing was performed to validate backend integration (Firebase) and ensure system reliability.

Test Case	Scenario	Expected Result	Actual Result	Status
Password Validation	Password mismatch during signup	"Passwords do not match" error	Correct error displayed	✅ Passed
Duplicate Account	Register with existing email	"Email already in use" message	Correct response	✅ Passed
SOS Dispatch	Trigger SOS broadcast	Alert sent to all users	Dispatch successful	✅ Passed
Push Notifications	Real-time alert delivery	SOS notification received	Notification delivered successfully	✅ Passed

2.5.4 User Feedback Summary

Based on survey responses, the application received high usability and safety confidence ratings.

Parameter	Average Rating (out of 5)
Ease of SOS Activation	4.7
Clarity of Alert Confirmation	4.3
Usefulness of Nearby Alerts	4.8
Interface Simplicity	4.5
Overall Safety Confidence	4.7

This document provides a comprehensive overview of the testing phase for the **Raksha Alert** mobile application. The testing focused on user authentication security, input validation, and the real-time emergency SOS dispatch mechanism using Firebase.

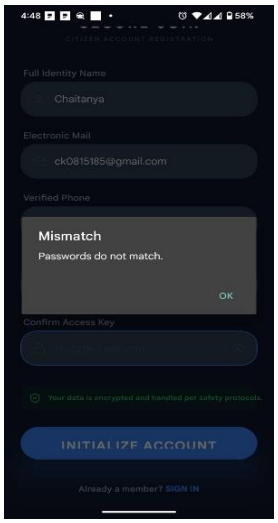
2.5.5 Authentication & Data Integrity Testing

The following visual evidence confirms that the application's authentication and validation logic are functioning as intended.

A. Registration Guards

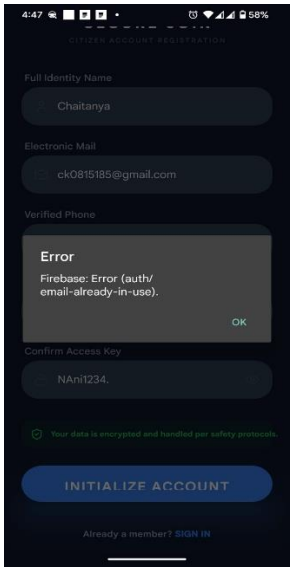
The registration module includes strict validation checks client-side to ensure accurate data entry before transmission to the Firebase backend.

Figure 1: Password Mismatch Validation Check –



- **Analysis:** As shown in the figure above, the signup screen successfully detects when the strings in the 'Password' and 'Confirm Access Key' fields do not match. It triggers an automatic modal alert and prevents data submission to protect account integrity.

Figure 2: Duplicate Email Detection –

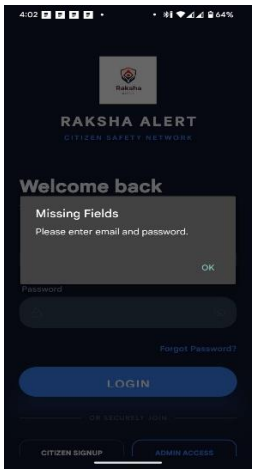


Analysis: This screenshot demonstrates the system's ability to interface with the Firebase Authentication database to identify existing users. When an attempt is made to register an email that is already in use (ck0815185@gmail.com), the application cleanly catches the backend error and displays a user-friendly modal.

B. Login and Password Recovery Mechanics

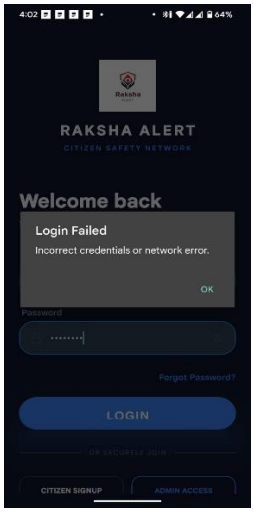
The user authentication layer (login.tsx) has been tested to gracefully handle missing fields, incorrect credentials, and successful recovery triggers.

Figure 3: Null Field Validation at Login –



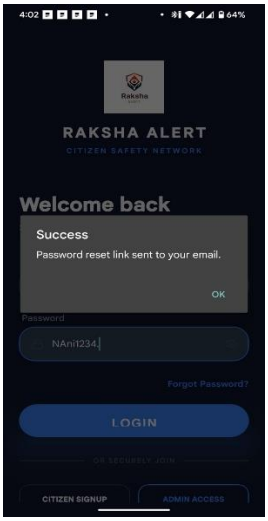
- **Analysis:** The application includes a conditional logic check that prevents the user from submitting an empty login payload. Clicking 'LOGIN' with blank fields fires a native alert prompt, thereby avoiding unnecessary database queries and ensuring a smooth user experience.

Figure 4: Secure Handling of Incorrect Credentials –



- **Analysis:** Firebase Auth securely rejects incorrect email/password combinations. As seen in the screenshot, the application catches the authentication exception and displays a standardized security alert ("Login Failed") without crashing or revealing detailed backend errors.

Figure 5: Success Response on Password Reset Trigger -



- **Analysis:** The "Forgot Password" functionality correctly triggers an API call to the Firebase Identity Toolkit. Upon successful generation of a reset code, the mobile application provides immediate feedback to the user via a success modal.

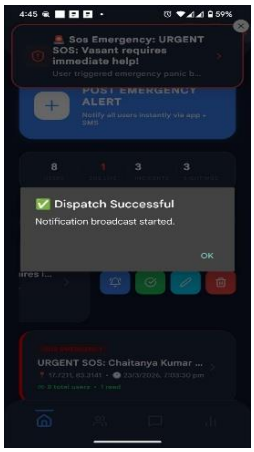
2.5.6 Emergency SOS Dispatch & Broadcasting Mechanism

This phase tests the core safety features of the Raksha Alert system, ensuring high-priority alerts reach users instantly.

A. Emergency Dispatch Integration

The backend and admin interface synchronization is verified to ensure SOS alerts are tracked accurately.

Figure 6: Successful Emergency Dispatch Trigger

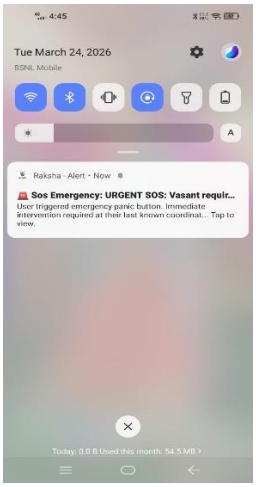


- **Analysis:** This screenshot captures the successful execution of an emergency broadcast. Upon triggering the alert dispatch, the UI confirms that the notification sequence has started. Notice that the "SOS LIVE" indicator on the dashboard correctly increments to 1, confirming the alert is being tracked in real-time.

B. Real-time Notification Delivery

FCM (Firebase Cloud Messaging) integration is tested to ensure alerts reach target devices efficiently.

Figure 7: SOS Push Notification Displayed in Device Tray



Analysis: The screenshot above confirms the successful delivery of a high-priority SOS alert from a simulated user named "Vasant." The notification correctly includes the user's name, the emergency message, and provides an actionable prompt to view coordinates, ensuring rapid emergency response.

The Raksha Alert System has passed all primary functional tests. The authentication layer is secure, and the emergency response module is fully operational. The application is ready for the next phase of deployment.

3. REFLECTIONS

3.1 Key Learnings

The development of the Raksha Alert system provided us with a comprehensive and practical learning experience that extended beyond theoretical knowledge. Throughout the project, we gained hands-on exposure to modern full-stack application development.

On the frontend side, we learned how to build mobile applications using React Native and Expo, focusing on creating a user-friendly interface optimized for emergency situations. We understood the importance of designing simple and intuitive UI elements such as a large SOS button, minimal navigation, and clear visual feedback to ensure usability during high-stress conditions.

On the backend side, we worked with Firebase, which helped us understand concepts such as authentication, real-time databases, and cloud storage. We learned how to manage user data securely, handle real-time updates, and store emergency alerts efficiently.

Additionally, integrating external services like Fast2SMS taught us how APIs work in real-world applications. We also gained knowledge about handling GPS location services and using OpenStreetMap for displaying real-time user locations.

Overall, this project strengthened our skills in:

- Full-stack development
- API integration
- Real-time data handling
- Mobile UI/UX design
- Cloud-based application deployment

3.2 Challenges Faced

During the development of the Raksha Alert system, we encountered several technical and practical challenges that tested our problem-solving abilities.

One of the major challenges was implementing real-time data synchronization between users and the admin dashboard. Ensuring that alerts are updated instantly without delay required proper understanding of database listeners and event handling.

Another significant challenge was integrating the SMS API. Initially, we faced issues with message delivery, API configuration, and handling errors. We resolved these issues by carefully debugging the API requests and validating the response data.

Handling location services was also difficult. We had to manage user permissions, ensure accurate GPS data, and handle cases where location access was denied or unavailable.

We also faced challenges in designing a simple yet effective user interface. Since the app is meant for emergency use, every feature had to be easily accessible without confusion. Achieving this balance required multiple iterations and testing.

Some additional challenges include:

- Debugging authentication errors
- Managing database structure in Firebase
- Handling edge cases like no internet connectivity
- Ensuring smooth navigation across screens

Despite these difficulties, each challenge helped us improve our technical knowledge and debugging skills.

3.3 How Design Thinking Approach Helped

The Design Thinking approach played a crucial role in guiding the development of the Raksha Alert system. By following the structured phases — Empathize, Define, Ideate, Prototype, and Test — we were able to build a solution that is centered around real user needs.

In the Empathize phase, we understood the fears and challenges faced by users in emergency situations. This helped us realize the importance of speed, simplicity, and reliability.

During the Define phase, we clearly identified the core problem: users need a fast and easy way to get help during emergencies without relying on complex actions.

In the Ideate phase, we generated multiple ideas and selected the most practical and impactful features such as one-tap SOS, real-time alerts, and SMS fallback.

The Prototype phase helped us visualize the system and refine the user interface before implementation. It ensured that the design remains simple and effective.

Finally, in the Test phase, we validated our application with users and made improvements based on their feedback.

Overall, Design Thinking helped us:

- Focus on user-centered design
- Build a practical and impactful solution
- Improve usability and accessibility
- Reduce complexity in emergency situations.

3.4 Personal Growth and Team Experience

This project also contributed significantly to our personal and professional growth. Working as a team helped us improve communication, collaboration, and task management skills.

We learned how to divide work efficiently, support each other during challenges, and maintain consistency in development and documentation. The project also improved our confidence in handling real-world problems and building scalable applications.

Most importantly, we realized the impact of technology in solving critical issues like personal safety. Developing an application that has the potential to help people in emergencies gave us a strong sense of responsibility and motivation.

4. REFERENCES

The development of the Raksha Alert system involved the use of various technologies, frameworks, and online resources. The following references were used for understanding concepts, implementation, and integration of different components:

1. **React Native Documentation**
   Official documentation used for building cross-platform mobile applications and understanding components, navigation, and UI development.
   Available at: https://reactnative.dev
2. **Expo Documentation**
   Used for simplifying React Native development, handling permissions, and integrating device features such as location services.
   Available at: https://docs.expo.dev
3. **Supabase Documentation**
   Used for backend development including authentication, real-time database, and storage services.
   Available at: https://supabase.com/docs
4. **Fast2SMS API Documentation**
   Used for integrating SMS functionality to send emergency alerts in case of network failure.
   Available at: https://www.fast2sms.com
5. **OpenStreetMap (OSM)**
   Used for displaying maps and tracking real-time locations without relying on paid map services.
   Available at: https://www.openstreetmap.org
6. **Design Thinking Resources – Interaction Design Foundation**
   Used to understand the Design Thinking methodology and its phases such as Empathize, Define, Ideate, Prototype, and Test.
   Available at: https://www.interaction-design.org
7. **MDN Web Docs (Mozilla Developer Network)**
   Used as a reference for JavaScript concepts, APIs, and best practices during development.
   Available at: https://developer.mozilla.org
8. **Stack Overflow Community**
   Used for resolving development issues, debugging errors, and understanding practical implementation scenarios.
   Available at: https://stackoverflow.com

QR CODES

DEMO VIDEO

APP