

Sheshield: Intelligent Protection for Every Step

Sarojini Yarramsetti¹, Darshan A², D Chandana³, Harshavardhan S⁴, Monika H A⁵

¹Professor, Department of Computer Science and Engineering – Data Science,
AMC Engineering College, Bangalore, Karnataka, India.

^{2,3,4,5} Final Year Students, Department of Computer Science and Engineering – Data Science,
AMC Engineering College, Bangalore, Karnataka, India.

DOI: <https://doi.org/10.51583/IJLTEMAS.2026.150900128>

Received: 20 September 2026; Accepted: 25 September 2026; Published: 19 October 2026

ABSTRACT

Personal safety systems need to provide rapid assistance while keeping emergency actions simple for the user. SheShield, titled “Intelligent Protection for Every Step,” is a women-safety solution that combines secure user access, map-based safe movement support, an assistance chatbot, an emergency SOS mechanism that can be started from an on-screen button or by shaking the phone (Shake-to-SOS), live-location sharing, and emergency contacts. A wearable safety pendant built around an Arduino microcontroller is described as a future enhancement, and an Arduino Uno push-button trigger has been assembled on a breadboard as a proof-of-concept. A key feature is a two-minute no-response mechanism: after an emergency is initiated and not cancelled, the system can monitor the user's response and, when no response is received within the defined interval, share the user's live location with the top three emergency contacts. The architecture is organized around modular safety, location, alert, and communication functions so that the system can be extended with additional sensors and services. This paper presents the problem, objectives, related work, system architecture, data and interface design, functional workflow, cancellation, network handling and logging, implementation approach, testing considerations, security, privacy and ethics, usability, limitations, and future enhancements of SheShield.

Keywords: Women safety, SOS, Shake-to-SOS, live location, GPS, emergency alert, safe navigation, chatbot, wearable safety, Arduino, location sharing.

INTRODUCTION

Personal safety is a critical concern when people travel, commute, or move through unfamiliar environments. Technology can support prevention and emergency response by reducing the number of steps required to request help, sharing relevant location information, and connecting the user with trusted contacts. The need for safer public spaces and practical safety interventions has also been highlighted by international organizations [1], [2].

Existing mobile safety applications commonly provide SOS alerts, GPS tracking, emergency contacts, and location sharing. Research has also explored integrated systems that combine real-time tracking with emergency communication and wearable devices [3]–[6]. However, users may still face difficulties when an emergency prevents them from interacting continuously with a phone. SheShield addresses this concern through a simple emergency workflow, two ways of triggering it (an SOS button and Shake-to-SOS), and a defined no-response escalation mechanism. The current prototype is a web-based application. An Arduino-based hardware trigger has been assembled at bench level (Fig. 13), and the wearable pendant built around it is a planned future enhancement (Section XIV).

Problem Statement

During an unsafe situation, a user may have limited time, limited ability to operate a smartphone, or

difficulty communicating their exact location. A safety solution should therefore provide a fast trigger, automatically capture the current location, notify trusted contacts, and continue the emergency workflow even when the user cannot respond immediately. The proposed problem is to design a unified safety platform that combines these functions in a simple and accessible interface.

Objectives

- Provide a secure login mechanism so that safety functions are available to an authenticated user.
- Provide an SOS facility for quickly initiating an emergency alert, started either from the on-screen SOS control or by shaking the phone (Shake-to-SOS).
- Allow an accidental activation to be cancelled before the alert is dispatched.
- Capture and share the user's live location with predefined emergency contacts.
- Support safe movement using map-based location information.
- Provide a chatbot interface for basic assistance and emergency guidance.
- Use a two-minute no-response mechanism to increase the chance of alert delivery when the user cannot respond.
- Document how the system handles data, network failure, logging, privacy, and ethical concerns.
- Describe a wearable safety pendant, built around an Arduino microcontroller, and its integration with the phone, as a future physical emergency trigger
- Design the system in modular components so future sensors, authorities, and services can be integrated.

Related Work

Previous research demonstrates that mobile and wearable technologies can improve emergency communication. A mobile women-safety application combines registration, emergency contacts, GPS tracking, SOS, and real-time location sharing [3]. Another system, SAKHI, explores SOS activation, location information, video evidence, and cloud-based storage [4]. Recent work has investigated compact IoT-based safety devices integrating GPS, communication, and a physical panic button [5]. Integrated women-security systems have also considered safe-route navigation and instant reporting [6].

SheShield builds on these ideas by combining mobile safety functions with a defined no-response escalation workflow and a Shake-to-SOS trigger. The aim is not to replace emergency services, but to reduce the delay between recognizing danger and notifying trusted people.

System	SOS	GPS	EC	VE	WT	SR	NR	CB
[3] Mobile app	✓	✓	✓	-	-	-	-	-
[4] SAKHI	✓	✓	-	✓	-	-	-	-
[5] IoT device	✓	✓	-	-	✓	-	-	-
[6] Integrated sys.	✓	-	-	-	-	✓	-	-
SheShield (proposed)	✓	✓	✓	-	✓	✓	✓	✓

Fig. 1. Feature comparison of SheShield with related work [3]–[6]

SOS: SOS alert; GPS: GPS/live location; EC: emergency contacts; VE: video evidence; WT: wearable trigger; SR: safe-route navigation; NR: no-response escalation; CB: chatbot assistance. The WT entry for SheShield refers to the planned Arduino-based pendant (Section XIV) and is not part of the current prototype.

Comparison with the reviewed systems. As Fig. 1 shows, all four reviewed systems support SOS, and [3]–[5] also provide GPS or live location. Each then covers a different subset of the remaining features: [3] adds emergency contacts, [4] adds video evidence, [5] depends on a dedicated device as its trigger, and

[6] adds safe-route navigation. None of them, as summarized in Fig. 1, defines a no-response escalation step or a chatbot. SheShield therefore differs by combining SOS, emergency contacts, safe-route navigation, no-response escalation, and chatbot assistance in one platform, and by offering Shake-to-SOS, which needs no dedicated hardware. Shake-based triggering is a common mobile interaction and is used here as an additional trigger rather than as a novel contribution. Conversely, SheShield does not currently provide video evidence, which [4] does, or direct law-enforcement reporting, which is considered in [6]; these gaps are noted in Sections XIII and XIV.

Proposed System

SheShield is organized as a user-facing safety platform connected to location and communication services. The user first authenticates through the secure login interface. After authentication, the main interface provides access to maps, the assistance chatbot, emergency contacts, and SOS. When an emergency is triggered, the system obtains the current location and begins the alert process. The architecture is illustrated in Fig. 2.



Fig. 2. Proposed SheShield system architecture

The interface layer hosts login, maps, chatbot, contacts, and SOS controls. The safety and location layer obtains the position, evaluates risk, ranks routes, and runs the countdown and no-response logic. The alert and communication layer delivers notifications to the emergency contacts over the network or, when no connection is available, through an offline SMS path. The optional wearable layer shown in Fig. 2 is a planned Arduino-based extension and is not part of the current prototype.

Functional Modules



Fig. 3. SheShield functional module map

Secure Login

The login module protects access to the application and its safety settings. Authentication prevents unauthorized users from modifying emergency contacts or safety preferences.

SOS and Emergency Alert

The SOS function provides a direct emergency trigger. Once activated, the system initiates the emergency workflow and prepares the user's current location for sharing with trusted contacts. Shake-to-SOS. In addition to the on-screen SOS control, the SOS can be activated by shaking the phone. A shake raises the same emergency event as the SOS button and so enters the same workflow, including the countdown and cancellation described in Section VII. Because shaking can occur accidentally, for example while walking or travelling, the countdown and cancel option are the safeguard against false alerts.

Live Location Sharing

Location information is used to communicate where the user is during an emergency. The design supports live location sharing so trusted contacts can follow changes in the user's position when location updates are available. Location handling. The position is requested with the user's permission and is captured when an emergency starts. Sharing is intended to last only for the duration of an emergency workflow, and the system should clearly show when sharing is active. If permission is denied or the position cannot be obtained, the alert cannot carry an accurate location; this dependency is recorded as a limitation (Section XIII).

Two-Minute No-Response Mechanism

After an emergency event, a two-minute response interval is started. If the user does not provide the expected response during this interval, the system can escalate the alert by sharing the user's live location with the top three emergency contacts. The interval is fixed at two minutes in the current design, and a response before it expires closes the workflow without escalation.

Safe Movement and Maps

The map module supports location awareness and safer movement. It can display the user's position and provide map-based assistance. Future versions can incorporate configurable safe-route criteria and

community safety information.

Emergency Chatbot

The chatbot provides quick access to basic assistance and guidance. It can present emergency instructions, help the user navigate available safety options, and direct the user toward the appropriate emergency action. It offers guidance only: it directs the user to the SOS control or the listed helplines rather than acting on the user's behalf, and it is not a substitute for professional, medical, or legal advice.

Safety Pendant (Planned, Arduino-Based)

The wearable pendant is a planned future extension and is not implemented in the current prototype. It is intended to be built around an Arduino microcontroller that reads a physical button and sends the trigger to the alert service. A breadboard proof-of-concept of this trigger is shown in Fig. 13. The pendant is retained in the module map (Fig. 3) and the architecture (Fig. 2) to show where a physical trigger would connect to the alert service. Its integration with the phone application and a wearable hardware prototype are described under Future Enhancements (Section XIV).

Emergency Workflow

The emergency workflow is designed to minimize user interaction. A typical sequence begins when the user presses SOS or shakes the phone. The system obtains the current location and notifies the configured emergency contacts. A response timer is then monitored. If the user does not respond within two minutes, the live location is sent to the top three emergency contacts as an escalation step.

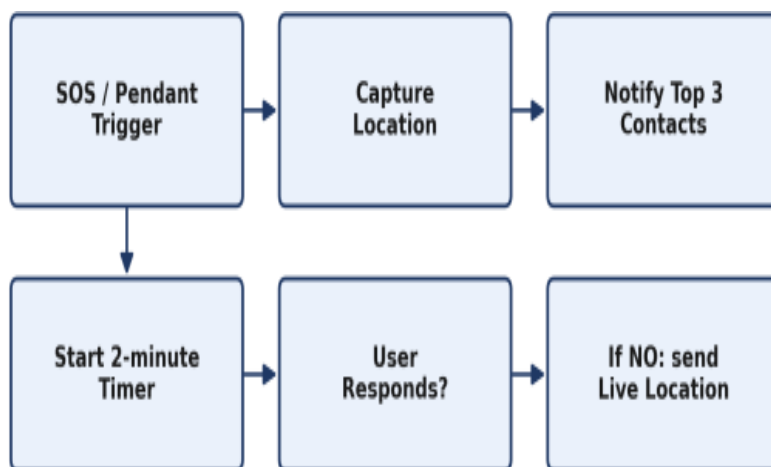


Fig. 4. Emergency response workflow

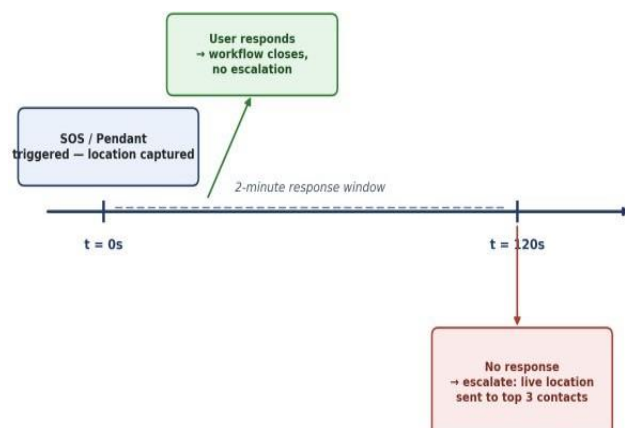


Fig. 5. Two-minute no-response escalation timeline

In Figs. 4 and 5 the pendant trigger (to be built around an Arduino) is a planned extension; in the current software prototype the trigger is the SOS button or Shake-to-SOS, and an Arduino push-button trigger exists at bench-prototype stage (Fig. 13). Fig. 12 shows the complete flow including cancellation.

Trigger and Cancellation

Either trigger starts a short countdown before the alert is dispatched, which gives the user an opportunity to cancel an accidental activation (Section XII.E). If the user cancels during the countdown, no alert is sent and the cancellation is recorded. If the countdown ends without cancellation, the system captures the location and notifies the emergency contacts.

Escalation

After the initial notification, the two-minute response interval runs. A response within the interval closes the workflow. With no response, the live location is sent to the top three emergency contacts, meaning the three highest-ranked entries in the user's emergency-contact list.

Network Issues and Offline Behaviour

Alerts and location are transmitted over the network. When no connection is available, the system uses an offline SMS path to send the location (Section XII.E). The SMS path still depends on cellular coverage and device permissions, so it reduces but does not remove the dependency on connectivity. If no channel is available, delivery cannot be guaranteed (Section XIII), and the interface should make the dispatch status visible to the user.

Event Logging

Each stage of the emergency workflow is designed to be recorded as a timestamped event: trigger (button or shake), countdown start, cancellation, location capture, contact notification, timer start, user response, and escalation. These records support the real-time activity feed (Fig. 8), troubleshooting, and post-incident review, and are subject to the retention and access rules in Section X.

Implementation Approach

The prototype is organized as a modular software system. The user interface contains authentication, maps, chatbot, emergency contacts, and SOS controls. A location service provides the current position. The alert layer manages emergency notification and the no-response timer. An Arduino-based wearable layer, planned as a future enhancement, could be implemented independently and connected to the alert service through an appropriate communication interface.

Hardware integration (Arduino). The hardware trigger uses an Arduino Uno board (Fig. 13). A push button is connected to a digital input pin, and the Arduino firmware reads the pin and treats a debounced press as a trigger event. The event is then passed to the alert service, which raises the same SOS event as the on-screen button or a shake. The communication link can be USB serial on the bench, or a Bluetooth or cellular module in a wearable form. This link is a design choice and has not yet been evaluated.

For production deployment, authentication credentials should be securely stored, sensitive location data should be protected in transit and at rest, and access to emergency-contact information should be restricted to authorized users. Location permissions should be requested transparently, and the system should clearly communicate when location sharing is active.

Data Model

The data used by the platform can be described by the logical entities in Table I. This is a logical

description; the storage technology and field formats are implementation choices.

Table I Logical Data Entities

Entity	Purpose and main content
User account	Identity and hashed credential used by the login module.
Emergency contact	Contact name and phone number owned by a user, with a rank; the top three ranked contacts receive the escalation.
SOS event	One emergency: trigger source (button or shake), start time, and state (countdown, cancelled, dispatched, responded, escalated).
Location record	Position and timestamp attached to an SOS event, and the channel used to send it.
Event log entry	Timestamped record of each workflow stage (Section VII-D).
Risk and route data	Historical and demonstration data used by the risk score and the safe-route ranking.

Service Interfaces

The user interface communicates with the services through the operations in Table II, described at a logical level.

Table II Logical Service Operations

Service	Operations
Authentication	Log in and log out; session check.
Contacts	Add, edit, remove, list, and rank emergency contacts.
SOS	Start (button or shake), cancel during countdown, record user response.
Location	Submit the current position for an active SOS event.
Alert and notification	Send the alert and live location to contacts over the network, or by offline SMS.
Route and risk	Compute a safety-ranked route; return the live risk score.
Companion	Submit a query; return guidance as text, with optional audio playback.
Hardware trigger (planned)	Receive a button event from the Arduino unit and start an SOS event.

Testing and Evaluation

Testing should verify both normal and emergency scenarios. The following functional test cases are recommended:

- Login with valid and invalid credentials.
- Add, edit, and remove emergency contacts.
- Trigger SOS with the on-screen control and verify that the emergency workflow starts.
- Shake the phone and verify that the same workflow starts; verify that ordinary movement does not start it unintentionally.
- Cancel during the countdown and verify that no alert is dispatched.
- Verify that the current location is captured and shared when permission is available.
- Verify that the two-minute timer starts after the emergency event.
- Provide a response before two minutes and verify that escalation is not unnecessarily triggered.
- Provide no response and verify that the top three emergency contacts receive the live-location alert.
- Test map access and location updates under different connectivity conditions, including the offline SMS path.
- Verify that workflow events are recorded in the log.
- Test chatbot responses for common safety queries.

Pendant trigger and reconnection tests, including Arduino button debouncing, press-to-alert latency, and link reliability, are deferred to the future hardware work (Section XIV). The cases added for Shake-to-SOS, cancellation, offline delivery, and logging are recommended tests; this paper reports no measurements for them.

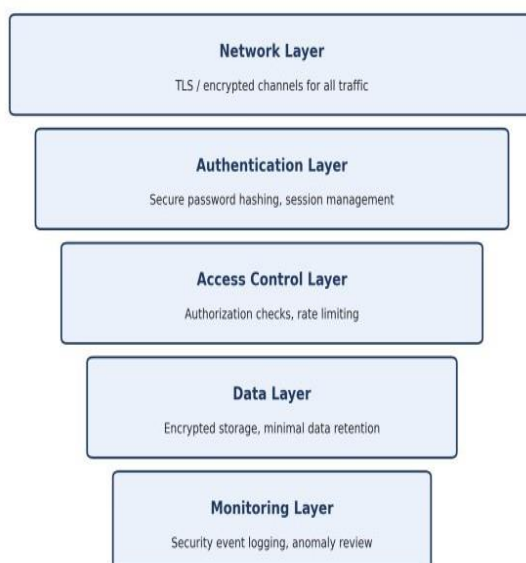


Fig. 6. Layered security architecture for SheShield

Security and Privacy Considerations

Because the system handles authentication and location information, privacy and security are essential. Passwords should never be stored in plain text; authentication data should use secure password hashing. Communication should use encrypted channels. Emergency contacts and location records should be accessible only to authorized users.

The application should minimize unnecessary data retention and provide clear controls for location permissions. A production system should also include rate limiting, secure session management, input validation, logging of security events, and protection against unauthorized account access. These are design requirements for production; the prototype has not undergone a security audit.

Privacy and Ethical Considerations

Consent and control. Location is shared only with contacts the user has chosen, and only for the duration of an emergency workflow. The user should be able to review and change contacts and see when sharing is active. Users should also tell the people they name as contacts, since those contacts will receive location data.

Data minimization and law. Only the contacts, location, and event records needed for the emergency workflow should be kept, with a defined retention period. Handling of personal data in India falls under the Digital Personal Data Protection Act, 2023 [7]; a deployed system needs a consent notice, purpose limitation, and a compliance review.

Misuse. Live location sharing could be misused for surveillance. Tying sharing to an emergency event, and restricting contact management to the authenticated user, limits this risk but does not remove it.

False alerts. Accidental shakes or button presses can send unnecessary alerts and burden contacts. The countdown and cancel option reduce this, and false-alert rates should be measured in field testing.

Fairness. Risk scores and hotspot overlays are derived from historical data, which may reflect reporting bias and may unfairly label neighbourhoods. They should be presented as indicative guidance, not as a definitive judgement of an area. The chatbot should likewise be presented as guidance only.

Advantages

- Fast emergency activation through the SOS control and Shake-to-SOS.
- A cancel window reduces the impact of accidental activation.
- Automatic location sharing reduces the need to explain the user's position during an emergency.
- The two-minute no-response mechanism provides an additional escalation path.
- Maps, chatbot, contacts, and SOS are integrated in one platform.
- The modular architecture can be extended with sensors, authorities, or additional emergency services, including an Arduino-based hardware trigger.

RESULTS AND DISCUSSION

The SheShield prototype was implemented as a web-based application and was exercised against the functional test cases listed in Section IX. This section presents the output of each implemented module and discusses what the results indicate about the feasibility of the proposed design.

Home Interface

The landing interface provides the entry point to the platform and exposes the main modules through a persistent side navigation bar: AI Dashboard, Smart Map, AI Companion, Analytics, Account Access, and Emergency SOS. A live status indicator confirms that the system is online, so the user can see at a glance that safety services are active before setting out. Fig. 7 shows this interface.

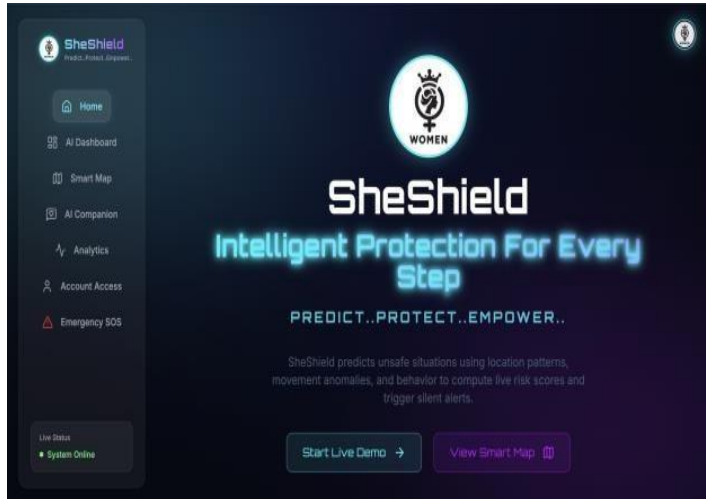


Fig. 7. SheShield home interface with module navigation and live status

AI Risk Prediction Dashboard

The dashboard computes a live risk score from location and movement data and presents it alongside the factors that produced it. In the output shown, the engine reported a score of 35 in the safe band, with behavior anomaly stable and no route deviation. The threat-vector panel breaks the score down into area safety, time-of-day risk, and movement pattern, and a real-time activity feed records the tracked events. Fig. 8 shows the dashboard output.

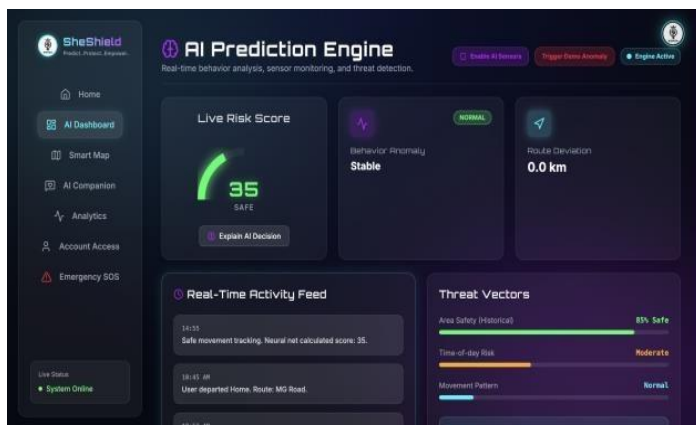


Fig. 8. AI prediction dashboard showing live risk score and threat vectors

Smart Map and Safe Route Navigation

The map module accepts a start point and destination and computes candidate routes ranked by safety rather than distance alone. The output distinguishes a recommended safe path, an alternative path, and a high-risk path, and overlays crime hotspots together with police stations and safe havens. Each route is labelled with a status such as safe or caution and the reason for that rating, which allows the user to make an informed choice. Fig. 9 shows a computed route between two locations in Bengaluru.

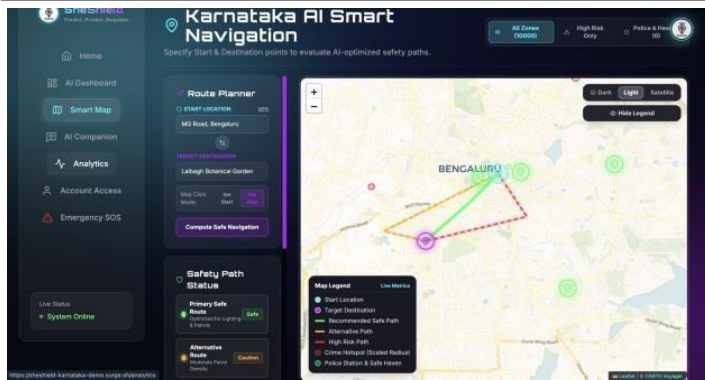


Fig. 9. Smart map output with ranked safe, alternative, and high-risk routes

AI Safety Companion

The chatbot module provides guidance through a conversational interface and offers quick-access prompts for common distress situations, such as feeling unsafe while walking alone or suspecting that one is being followed. Responses can also be played back as audio, which supports use when the user cannot read the screen. Fig. 10 shows the companion interface.



Fig. 10. AI safety companion with guided distress prompts

Emergency SOS and Escalation

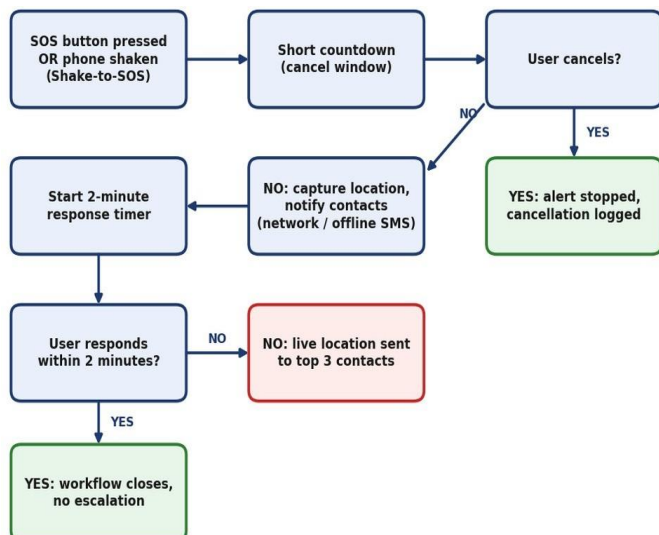
The SOS screen implements the emergency workflow described in Section VII. The central trigger starts a short countdown before dispatching the alert, which gives the user an opportunity to cancel an accidental activation. Location can be transmitted over the network or, when no connection is available, through an offline SMS path. The screen also exposes a loud siren, a hands-free voice trigger, the registered escalation contacts, and official helpline numbers. Fig. 11 shows the emergency output.



Fig. 11. Emergency SOS screen with escalation contacts and helplines

Shake-to-SOS Trigger

Shake-to-SOS is the feature added in this revision of the prototype. Shaking the phone activates the SOS, giving a second way to start the emergency workflow without locating and pressing the on-screen control. Fig. 12 summarizes how either trigger proceeds through the countdown, cancellation, and the two-minute escalation. This paper reports no measurements of shake-detection accuracy, sensitivity, or false-trigger



rate; these remain to be evaluated (Section XIV).

Fig. 12. Shake-to-SOS and SOS button flow with cancellation and two-minute escalation

Hardware Trigger Prototype (Arduino)

To examine the physical-trigger concept behind the planned safety pendant, a bench prototype was assembled around an Arduino Uno microcontroller board, as shown in Fig. 13. A tactile push button is mounted on a mini breadboard and connected to the board's header pins with jumper wires, and the board is powered over a USB cable. The Arduino reads the button as a digital input. A press is intended to be forwarded to the alert service, where it raises the same SOS event as the on-screen button or a shake and so passes through the same countdown, cancellation, and two-minute escalation (Fig. 12).

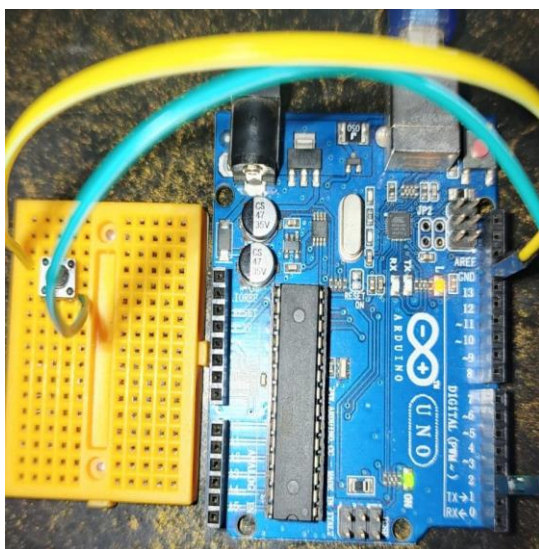


Fig. 13. Arduino Uno bench prototype of the hardware trigger: a push button on a mini breadboard wired to the board and powered over USB.

The Arduino suits an early prototype because it is inexpensive, widely available, and simple to program. This paper reports no measurements of the hardware trigger, such as press-to-alert latency, debounce behavior, or the reliability of the link between the Arduino and the application; these remain to be evaluated (Section XIV). The setup is a proof-of-concept on a breadboard and not the wearable pendant itself, which still requires an enclosure, battery power, and a wireless link.

DISCUSSION

The outputs confirm that the individual safety functions can be combined into a single workflow without adding user steps during an emergency: one trigger, either the on-screen SOS control or a shake of the phone, is sufficient to raise the alert, attach location, and begin escalation. The two-minute no-response mechanism remains the component that most clearly differentiates the prototype from the systems reviewed in Fig. 1, since it continues the workflow when the user is unable to interact with the device at all. The Arduino bench setup (Fig. 13) shows that a physical trigger can be built from commodity components and is designed to use the same SOS event, but its integration with the alert service has not yet been evaluated.

Two practical observations follow from the tests. First, the reliability of the workflow depends heavily on factors outside the application, particularly location permission state and network availability; the offline SMS path reduces but does not remove this dependency.

Second, the risk scores and route ratings produced by the prediction and map modules are derived from historical and demonstration data, so they indicate that the scoring pipeline works end to end rather than establishing predictive accuracy. These results should therefore be read as functional verification of the prototype, not as a measurement of real-world emergency performance; the controlled field testing described in Section XIV is still required.

Usability Considerations

The interface is designed for low effort in a stressful situation: a persistent navigation bar with a live status indicator (Fig. 7), guided prompts and audio playback in the companion (Fig. 10), a single central SOS control with a cancel window, and a siren and helpline numbers on the same screen (Fig. 11).

Two trigger routes, the button and a shake, allow the user to choose whichever is easier at the moment. No usability study with participants has been conducted, so these points are design rationale, not measured results; a study is listed under Future Enhancements.

LIMITATIONS

The effectiveness of live location and notification depends on device permissions, network connectivity, GPS availability, and the reliability of external communication services. The two-minute interval is a design choice and may need to be configurable.

The wearable pendant is a future extension that is not implemented; the Arduino push-button setup (Fig. 13) is only a breadboard proof-of-concept, and a wearable form requires hardware, communication, power-management, and reliability testing before real-world deployment. The prototype should therefore be treated as a safety-support system, not as a guaranteed substitute for police, medical, or other emergency services.

Further limitations are as follows. Shake detection may miss a deliberate shake or respond to an accidental one, and its behavior has not been measured. The Arduino trigger has not been measured either, and its link to the alert service is not yet evaluated.

The system does not provide video evidence or direct reporting to law-enforcement agencies. Risk scores and route ratings use historical and demonstration data and are not validated for predictive accuracy.

Escalation depends on the contacts noticing and acting on the alert. The prototype has not had field testing, a usability study, or a security audit, and the chatbot gives general guidance only.

Future Enhancements

- Design and build the wearable safety pendant around an Arduino microcontroller as a hardware prototype, including its integration with the phone application (pairing, reconnection, and raising the same SOS event), power management, and an enclosure; then test it for reliability.
- Develop a low-power Arduino-based Bluetooth or cellular safety pendant with reliable offline/poor-network behavior.
- Evaluate the Arduino button trigger: debouncing, press-to-alert latency, and link reliability.
- Evaluate Shake-to-SOS: detection accuracy, sensitivity settings, and false-trigger rate.
- Make the countdown and the two-minute escalation interval configurable.
- Integrate verified emergency-service or local-authority communication channels where legally and technically feasible.
- Add configurable safe-route scoring using lighting, traffic, crowd, and verified incident information.
- Improve the reliability of voice-activated emergency triggering for hands-free use.
- Add multilingual chatbot support and accessibility features.
- Use anomaly or risk detection carefully, with user consent and privacy-preserving design.
- Conduct usability studies and controlled field testing to measure response time, reliability, false alerts, and user experience.

CONCLUSION

She Shield presents a unified approach to personal safety by combining secure access, SOS with an on-screen and a Shake-to-SOS trigger, cancellation, live location sharing, map-based support, a chatbot, and emergency contacts. Its main contribution is the integration of these functions with a two-minute no-response escalation mechanism that can automatically share live location with the user's top three emergency contacts. The proposed architecture is modular and can be extended as additional hardware, including the planned Arduino-based wearable pendant, communication services, and safety data become available. Future development should focus on rigorous security, privacy, reliability, usability, and real-world evaluation before deployment.

REFERENCES

1. World Health Organization, "Violence against women," Fact sheet, WHO, Geneva, Switzerland, 2026. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/violence-against-women>
2. UN Women, Safe Cities and Safe Public Spaces for Women and Girls Global Initiative: Global Results Report 2017–2020. New York, NY, USA: UN Women, 2021.
3. D. S. Prashanth, G. Patel, and B. Bharathi, "Research and development of a mobile based women safety application with real-time database and data-stream network," in Proc. Int. Conf. Circuit, Power and Computing Technologies (ICCPCT), 2017, pp. 1–5, doi: 10.1109/ICCPCT.2017.8074261.
5. A. V. Agarwal, V. Singh, A. Kamboj, A. Sirohi, and A. Mehto, "Development of a women safety

- smartphone application—SAKHI,” in Proc. 3rd Int. Conf. Secure Cyber Computing and Communication (ICSCCC), 2023, pp. 212–217, doi: 10.1109/ICSCCC58608.2023.10176701.
6. Kushal Agrawal, Aviral Srivastava, Kanishk Sharma, “An emergency alert and monitoring system using compact IoT-based real-time women and child safety,” in Proc. 10th Int. Conf. Smart Structures and Systems (ICSSS), 2025, [pages], doi: 10.1109/ICSSS66939.2025.11346389.
 7. Laxmi Raj Gupta, Krupashankari S Sandya, Anisha, “Integrated women's security system with safe route navigation and instant law enforcement reporting,” in [Proc. conference name], 2024, [pages, doi].
 8. Government of India, The Digital Personal Data Protection Act, 2023, Act No. 22 of 2023, 2023.