ExperimentNo.:3

Date:23-01-2026

Experiment Title:

Design and Develop a Multi-Sensor Android Prototype – *Smart Safety Companion*

Aim:

To design and develop a multi-sensor Android prototype named *Smart Safety Companion* that detects emergency situations using phone sensors, captures the user's current location and direction, and generates a meaningful emergency alert message by applying Software Engineering design and implementation principles.

Problem Description:

In real-world safety scenarios such as accidents or personal emergencies, immediate detection and alert generation play a crucial role in ensuring timely assistance. The objective of this experiment is to develop a functional Android application that detects a sudden phone shake as an emergency event, identifies the user's geographical location, determines the phone's direction, and generates an emergency alert message. The application focuses on converting clearly defined requirements into a working prototype using built-in mobile sensors rather than developing a full-scale commercial application. This experiment simulates practical safety systems by integrating sensor data processing, event detection, and alert generation.

Given Conditions / Constraints:

- The application must use **Accelerometer Sensor**, **Location Sensor**, and **Orientation Sensor**
- Shake detection must be calculated using Shake Value = $|X| + |Y| + |Z|$
- Emergency event must be triggered if ShakeValue > 25
- Direction must be displayed as North, East, South, or West based on heading ranges
- Application interface must include:
    - One TextBox for emergency contact number

  - Two Buttons (Start Monitoring and Stop Monitoring)
  - Multiple Labels for sensor values and status
- Alert message must include location, direction, and current time
- Cooldown period of 10 seconds must be enforced after an alert
- SMS sending is optional, but alert display is mandatory

Approach / Methodology:

The experiment was carried out using a structured development approach. Initially, functional requirements and constraints were analyzed to identify mandatory sensors and interface components. The user interface was designed to allow easy interaction and clear display of sensor data. Sensor values were continuously monitored once the Start Monitoring button was pressed. Emergency detection logic was implemented using accelerometer readings and a predefined threshold. Upon detecting an emergency event, location, direction, and time details were captured and combined to generate an alert message. A cooldown mechanism was added to prevent repeated alerts, and the Stop Monitoring button was used to halt sensor processing.

Planning and Estimation:

The development process was planned in sequential stages:

- User interface design
- Sensor integration
- Shake detection logic implementation
- Location and direction mapping
- Alert message generation
- Cooldown mechanism implementation
- Testing and validation

Each stage was implemented incrementally to ensure correctness and ease of debugging. Since the application uses built-in sensors and standard Android components, no additional hardware or cost estimation was required.

Scheduling / Design Considerations:

The application workflow was designed to ensure smooth monitoring and accurate event handling. Sensor monitoring begins only when explicitly enabled by the user. Emergency detection is performed continuously during monitoring, while the cooldown mechanism ensures controlled alert generation. Direction mapping was simplified into four cardinal directions to improve readability. The Stop Monitoring feature ensures immediate termination of sensor activity, enhancing user control and system reliability.

Feasibility Analysis:

TechnicalFeasibility:
The application uses standard Android sensors and components, making it technically feasible on most smartphones.

EconomicFeasibility:
No additional hardware or paid services are required, making the solution cost-effective.

OperationalFeasibility:
The application is simple to use, with minimal user input and clear visual feedback.

ScheduleFeasibility:
The prototype can be developed and tested within the allocated laboratory time.

Observation:

During the experiment, it was observed that combining multiple sensor inputs enables accurate detection of emergency situations. Shake threshold selection plays a crucial role in avoiding false alerts, and the cooldown mechanism significantly improves system reliability. Real-time display of sensor values helps in validating system behavior during testing.

Result:

A functional Android prototype named *Smart Safety Companion* was successfully developed. The application accurately detects phone shakes, captures location and direction, generates emergency alert messages, and enforces a cooldown period as specified.

Conclusion:

This experiment demonstrates the effective use of Software Engineering principles in developing a real-world safety-oriented mobile application. By integrating multiple sensors, applying structured planning, and implementing validation mechanisms, a reliable and practical prototype was achieved. The experiment highlights the importance of requirement analysis, logical design, and controlled event handling in mobile application development.