

Project Description

Kitchen Sentinel is a simple, user-friendly, and accessible Android application created by Macabenta, Morden, and Pilapil of Team MPM. It is designed to assist an IoT system, also named Kitchen Sentinel, that detects gas leaks, temperature changes, and motion in the kitchen. The system alerts users and caregivers to prevent accidents. The target population for this app includes elderly individuals who face challenges due to old age, especially those living alone. The application's intended task is to display the sensor readings from the IoT system and alert users when the sensors detect any issues.

Requirement Summary

| | | |
|---------------------------------|--------------|-------------------------------|
| Minimum Requirements | Processor | Dual-Core |
| | OS | Android 5.0 (Lollipop) |
| | RAM | 2 GB |
| | Connectivity | Wi-Fi or Mobile Data |
| Recommended Requirements | Processor | Quad-Core |
| | OS | Android 8.0 (Oreo) |
| | RAM | 4 GB |
| | Connectivity | Wi-Fi or Mobile Data |
| Other Requirements | Permissions | Notification Access, Internet |

Table 1. System Requirements

The application targets devices running at least Android 5.0, with a dual-core processor and 2 GB of RAM. It remains lightweight to support lower-end models. For optimal performance, Android 8.0, a quad-core processor, and 4 GB of RAM are recommended. Internet access and notification permissions are required.

User Scenario

Lola Maria, a 75-year-old who lives alone, enjoys cooking but has recently become forgetful. One morning, she accidentally left the gas stove on. Her daughter Ana, concerned for her safety, started looking for ways to prevent future kitchen accidents.

Ana discovered Kitchen Sentinel, a smart kitchen safety app made for elderly users. After setting it up in Maria's home, the system began detecting gas leaks, stove activity, and presence near the stove. Maria now gets simple alerts through lights and sounds, while Ana receives real-time updates through the app, which keeps Maria safe and independent in her own kitchen.

Kitchen Sentinel Mock-up/Prototype

Opening Application – Users will see the Kitchen Sentinel logo displayed prominently upon launching the app.

Dashboard – Users see a clean, intuitive layout showing three core detection functions: Gas, Fire/Heat, and Motion. Each is marked with a clear icon, and visual alerts appear beside any active threat. 'Activity Log' and 'Settings' buttons provide quick access to event history and customization.



Figure 1. Loading and Dashboard Screens

Activity Log – Users can view a chronological list of all past alerts and system events. Each entry includes a timestamp, making it easy for users or caregivers to review and verify specific incidents.

Settings – In the Settings page, users will be able to manage notifications, test the sensors, add or update a caregiver number, access and edit their emergency contact list, switch between light and dark mode, adjust the font size for better readability, and access the Help Center for support and frequently asked questions.

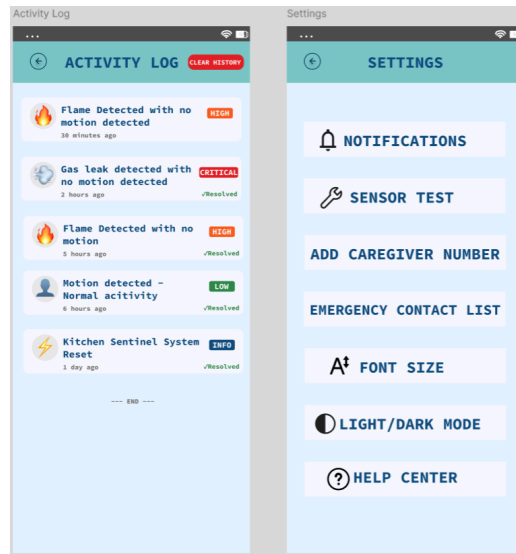


Figure 2. Activity Log and Settings Screen

Various Setting Options – Users, once settings are opened, can enable or disable notifications; perform a sensor test; add a number for the caregiver, update their view and update their emergency contact list, and adjust the font size to suit their reading comfort.

Activity Log when Cleared – When a user clears the Activity Log, all entries will be removed, and a message will appear indicating that the log is empty.

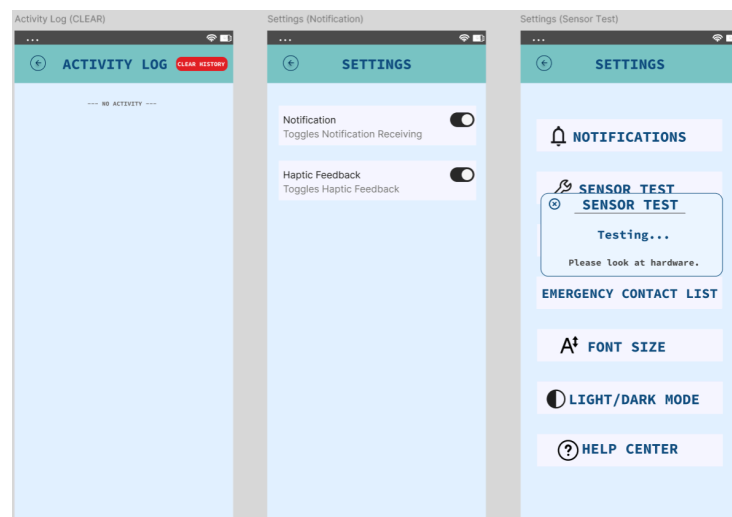


Figure 3. Activity Log (Clear) and Various Setting Options

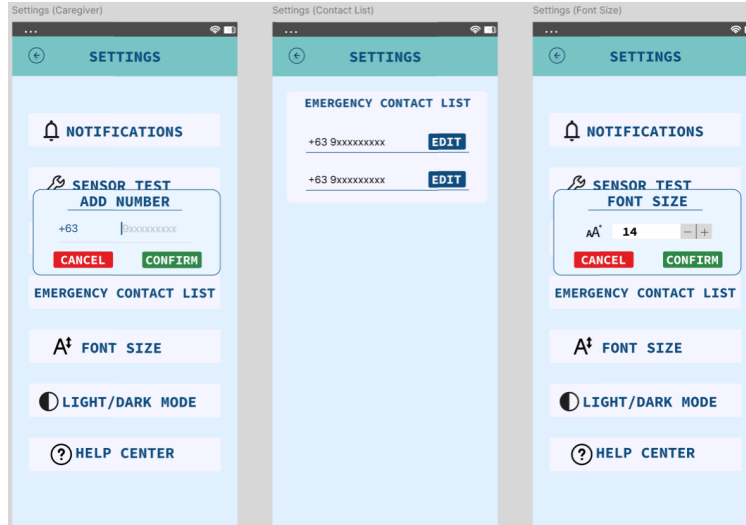


Figure 4. Various Setting Options

Prototype Description

The prototype is developed using React Native for the Android front end and Firebase for backend communication. It connects to an ESP32-based IoT system comprising:

- MQ-2 Gas Sensor
- Temperature Sensor
- PIR Motion Sensor

When an abnormal condition is detected (e.g., gas leak or high heat), the ESP32 transmits data to Firebase. The mobile app then displays updated readings and sends alerts through push notifications.

Kitchen Sentinel Figma Prototype

In addition to the functional prototype, a user interface design prototype was created using Figma. This visual prototype serves as the foundation for the application's user interface and overall interaction design. It can be accessed via the following link:

<https://www.figma.com/proto/zxHhrWdK6nB4wN1uAssqkv/Kitchen-Sentinel?node-id=0-1&t=XwbkP0AyIXhGDh6Y-1>

Rationale

The Kitchen Sentinel prototype was chosen for its alignment with the system's core goals: simplicity, low interaction, and real-time safety monitoring for elderly users. It effectively meets the identified user needs, particularly those with limited mobility, memory issues, or low technical literacy, by relying on passive hazard detection and clear audiovisual alerts rather than requiring complex interactions.

The system also provides remote notifications to caregivers, adding a critical safety layer for users living alone. The mobile app, built using React Native and Firebase, ensures real-time synchronization with the IoT device while offering a clean and accessible interface.

Advantages:

- Designed for elderly users with limited tech skills.
- Provides passive safety without needing constant user input.
- Real-time data syncing with Firebase.
- Simple UI for immediate understanding and feedback.
- Remote alerting allows caregiver involvement.

Disadvantages:

- Depends on stable internet connectivity for remote alerts.
- Lacks automatic gas shutoff or fire suppression.
- Indoor use only; limited sensor range may cause occasional false alerts.

Despite the limitations, the prototype was still able to meet the functional and usability requirements for the target population, making it a great and suitable solution.

Changes to the Requirements:

There were no major changes to the requirements and usability criteria of the project, only minor revisions were made to improve the overall design and user experience of the prototype, based on the 10 usability heuristics. In the application, we reduced the number of features and focused on the core functions to enhance clarity and usability, making it more elderly-friendly. Accessibility features such as dark mode and font size adjustments were added to improve visibility and comfort for elderly users. We also adopted a minimalist design approach to make the app easier to navigate. Additionally, we enhanced the help and documentation section by adding more information and FAQs, accessible through the Help button in the Settings page.

Initial Evaluation Plan

As school is still ongoing and the rest of the group members are busy with their academics and other commitments, conducting in-person testing is not practical at this time. Therefore, to ensure that the developers can still test the application, we have opted to use remote and online methods. Platforms such as Discord, Microsoft Teams or Messenger will be utilized, where developers can invite participants to join testing sessions and provide feedback.

To evaluate the project, the following techniques will be used:

1. **Usability Specifications** - Measuring usability goals (e.g., time to complete a task, user satisfaction) to evaluate the effectiveness and efficiency of the prototype.
2. **Heuristic Evaluation** - Using Nielsen's 10 Usability Heuristics to identify design issues in the prototype and ensure it adheres to established usability principles.
3. **Survey (System Usability Scale)** - Administering the SUS (System Usability Scale) questionnaire to gather user feedback and calculate a usability score out of 100.

Population

Due to time and resource constraints, the sample size of participants is limited to only 10 individuals, 5 elderly users and 5 able-bodied individuals who can be considered as caregivers. They will be asked to perform the core functionalities of the system, and we will measure the system's efficiency, effectiveness, and their overall satisfaction with the application.

| Developer / UI Designer | Member Task(s) |
|-------------------------|---|
| Mel John S. Macabenta | Ensures the prototype is functioning correctly |
| Jhouvann S. Morden | Assist participants with technical issues during the test (if any). |
| Marga G. Pilapil | Record observations and other metrics during the test. |

Table 1. Roles

Usability Specifications

The following are the key usability metrics the prototype aims to meet:

1. **Efficiency** – Time taken by users to complete specific tasks.
2. **Effectiveness** – Accuracy and completeness with which users achieve their goals.
3. **Satisfaction** – User comfort and positive attitudes towards using the system.

| Mean Score Range | Interpretation | Classification |
|------------------|-------------------------------|-------------------|
| 3.6 – 4.0 | Excellent / Highly Acceptable | Very Successful |
| 3.0 – 3.5 | Good / Acceptable | Successful |
| 2.0 – 2.9 | Fair / Moderately Acceptable | Needs Improvement |
| 1.0 – 1.9 | Poor / Not Acceptable | Unsuccessful |

Table 2. Usability Score Interpretation Table

| Average Time | Interpretation | Classification |
|-----------------------|-------------------------------|-------------------|
| 0.0 – 0.6 minutes | Excellent / Highly Acceptable | Very Successful |
| 0.7 – 1.5 minutes | Good / Acceptable | Successful |
| 1.6 – 2.5 minutes | Fair / Moderately Acceptable | Needs Improvement |
| More than 2.5 minutes | Poor / Not Acceptable | Unsuccessful |

Table 3. Time Interpretation Table

Prototype Tasks

The following are tasks to be performed by the participants in order to evaluate the effectiveness, efficiency, and overall satisfaction of the Kitchen Sentinel prototype. These tasks are based on the core features of the application

1. Hazard Detection and Alert Handling:

- Task 1: Open the app and view the current sensor readings (Gas, Temperature, Motion) on the Dashboard.
- Task 2: Identify which sensor triggered the alert using the icon or visual indicator.

2. Activity Log Review:

- Task 1: Navigate to the Activity Log section.
- Task 2: Review the most recent alert and identify its timestamp and type.

3. Settings:

- Task 1: Open the Settings page from the main Dashboard.
- Task 2: Enable Dark Mode and increase the font size for better readability.
- Task 3: Add or update an emergency contact or caregiver phone number.
- Task 4: Access the Help Center and locate information about handling gas alerts.

4. General Use and Navigation:

- Task 1: Launch the application and explore each section (Dashboard, Activity Log, Settings).
- Task 2: Interpret the overall kitchen safety status based on the sensor data.

Heuristic Evaluation

The heuristic evaluation for the Kitchen Sentinel will be conducted using the following Jakob Nielsen's 10 Usability Heuristics.

Visibility of System Status

The Dashboard should give clear real-time feedback by presenting current sensor data such as Gas, Temperature, Motion and alarm indicators. This ensures that users are always informed about the kitchen's safety condition.

Match Between System and the Real World

To guarantee that the interface speaks the users' language, the system makes use of commonly used terms and icons like "Gas" and "Activity Log." In order to accommodate elderly users', the design avoids the use of technical jargon.

User Control and Freedom

Users can easily navigate between app sections and perform actions like updating emergency contacts. Confirmation prompts for critical actions help users feel in control and prevent accidental data loss.

Consistency and Standards

All of the pages in the interface have the same layout, icons, and interaction. Users may rely on common mobile norms, which ensures easier navigation and lowers the learning curve.

Error Prevention

User error is reduced by preventive design decisions including contact number validation and toggle switches for settings. The streamlined settings structure also makes sure users don't misconfigure the system.

Recognition Rather Than Recall

The dashboard displays all relevant data, including alerts and sensor statuses. Users no longer need to commit actions or content locations to memory thanks to the usage of clear icons and labeled buttons.

Flexibility and Efficiency of Use

Features like dark mode and text size modifications are included in the program to accommodate users of all levels. These choices improve usability for older or visually impaired users and let users customize their experience.

Aesthetic and Minimalist Design

The application's UI is clear and straightforward. Users are able to concentrate on important information without being distracted by other features.

Help Users Recognize, Diagnose, and Recover from Errors

Users may resolve issues with the help of clear and detailed error messages and a Help Center. When there are alarms or issues with usage, these tools offer customers helpful guidance.

Help and Documentation

The Help Center, which offers solutions to commonly asked questions and fundamental troubleshooting techniques, offers easily accessible documentation for the program. For users who might want assistance using the app, this is essential.

Survey

A System Usability Scale (SUS) survey questionnaire will be distributed to the users. It consists of statements to be rated from 1 (Strongly Disagree) to 5 (Strongly Agree). The results will then be calculated and interpreted using the following steps:

Calculation:

- For odd-numbered questions: subtract 1 from the user's response.
- For even-numbered questions: subtract the response from 5.
- The total is then multiplied by 2.5 to yield a score out of 100.

| Score | Interpretation | Grade |
|---------------|---------------------|-------|
| 80.3 or above | Excellent Usability | A |
| 68 | Usability | C |
| Below 51 | Poor Usability | F |

Table 4. SUS Interpretation Table

Questions

Scale: 1 – Strongly Disagree, 5 – Strongly Agree

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.