Smart Rescue Robot

---

1. Introduction

This document defines the project requirements for the Smart Rescue Robot, outlining functional and non-functional requirements, use cases, and acceptance criteria. The robot is designed to assist in rescue operations by guiding people, detecting hazards, and providing emergency alerts.

2. Business Requirements

The Smart Rescue Robot aims to improve rescue missions by offering a reliable, autonomous system that detects obstacles, fires, and earthquakes. Its business objectives are to enhance safety, ensure quick emergency responses, and operate efficiently for extended periods on a single charge.

3. System Overview

The Smart Rescue Robot autonomously guides people through hazardous areas using sensors to detect obstacles, fires, and earthquakes. It follows predefined paths, provides alerts through sound and lights. The robot operates for 2 hours on a full charge and is easily programmable via USB.

---

4. Project Requirements:

1. Functional Requirements:

1. **Guiding People and Avoiding Obstacles:**
    - The robot must detect obstacles using the Ultrasonic Distance Sensor.
    - Upon detecting an obstacle, it must change its path to avoid collisions while continuing to guide people safely.
2. **Fire Sensing:**
    - Detect heat sources using an Infrared Sensor.
    - Emit an audible alert using a Buzzer or LED when detecting a heat source.
3. **Earthquake Sensing:**
    - Use a 3-axis Accelerometer to detect vibrations.
    - Trigger an audible alert if the vibrations exceed a certain threshold.
4. **Orientation:**
    - Use a 3-axis Compass to maintain the correct direction.
5. **Path Following:**
    - Follow predefined paths using Edge and Line Sensors.

6. **Light Sensing:**
    - Detect low-light areas using Light Sensors.
7. **Environment Interaction:**
    - Use Motor-driven Grippers to remove obstacles.

2. Non-Functional Requirements:

1. **Performance:**
    - Detect obstacles or heat sources within 2 seconds.
    - Respond to earthquakes within 3 seconds.
2. **Efficiency:**
    - Operate continuously for at least 2 hours on a fully charged battery.
3. **Safety:**
    - Automatically stop when encountering impassable obstacles.
    - Ensure audible alerts are clear.
4. **Ease of Use:**
    - Allow easy programming via USB Port.
5. **Mobility:**
    - Ensure smooth movement on flat surfaces.

3. Constraints:

- Use only built-in components.
- Ensure compatibility with current robot software.
- Optimize power consumption.

4. Timeline:

- Complete implementation within 9-10 weeks.

---

5. Priority of Feature:

1. **Must Have:**
    - Avoid obstacles.
    - Follow paths.
    - Detect fires.
    - Sense earthquakes.
    - Indicate emergency status using LED.
2. **Should Have:**
    - Automatically change paths.
    - Display guidance using LCD.

3. **Could Have:**
    - Use a 3-axis Compass for navigation.
    - Perform rescue tasks using Motor-driven Grippers.
4. **Won't Have:**
    - Draw maps using a Pen Holder (future implementation).

---

6. User Stories:

1. **Obstacle Avoidance:**
    - As a user, I want the robot to detect and avoid obstacles to ensure safe navigation.
2. **Path Following:**
    - As a user, I want the robot to follow predefined paths for efficient rescue operations.
3. **Fire Sensing:**
    - As a user, I want the robot to detect fires and issue alerts.
4. **Earthquake Sensing:**
    - As a user, I want the robot to sense vibrations and alert me in case of an earthquake.
5. **Audible Alerts:**
    - As a user, I want the robot to emit audible alerts during emergencies.
6. **Automatic Path Change:**
    - As a user, I want the robot to change its path automatically when facing obstacles.
7. **Emergency Light Signals:**
    - As a user, I want the robot to signal emergencies using colored lights.
8. **Stopping During Earthquakes:**
    - As a user, I want the robot to stop moving when detecting earthquakes.

---

7. Acceptance Criteria:

1. **Avoiding Obstacles:**
    - **Given** the robot is guiding people,
    - **When** it detects an obstacle using the Distance Sensor,
    - **Then** it must change its path and continue moving.
2. **Fire Sensing:**
    - **Given** the robot is in monitoring mode,
    - **When** it detects a heat source using the Infrared Sensor,
    - **Then** it must trigger an alert and display a warning.

3. **Earthquake Sensing:**
    - **Given** the robot is in monitoring mode,
    - **When** vibrations exceed a certain threshold,
    - **Then** it must trigger an alert and display a warning.
4. **Path Following:**
    - **Given** the robot is moving along a predefined path,
    - **When** it uses Edge and Line Sensors,
    - **Then** it must follow the path accurately.
5. **Guiding People:**
    - **Given** the robot is guiding people,
    - **When** alerts or signals are needed,
    - **Then** it must use LED lights and emit audible alerts.
6. **Using Gripper Arm:**
    - **Given** the robot is tasked with moving an object,
    - **When** its Motor-driven Grippers are activated,
    - **Then** it must securely hold and move the object.

---

8. Use Cases

Use Case: Obstacle Avoidance

Actor: Robot

Goal: Avoid obstacles while guiding people.

Precondition: Robot is moving along a predefined path.

Steps:

1. Robot detects obstacles using Ultrasonic Sensor.
2. Robot changes path to avoid collision.
3. Robot continues guiding people.

Outcome: Obstacle is avoided, and the robot keeps moving safely.

Use Case: Fire Sensing

Actor: Robot

Goal: Detect fire and alert users.

Precondition: Robot is in monitoring mode.

Steps:

1. Robot detects heat source with Infrared Sensor.
2. Robot triggers an audible alert and LED warning.
3. Robot continues its operation.

Outcome: Fire detected, alert issued.

Use Case: Earthquake Sensing

Actor: Robot

Goal: Detect earthquakes and alert users.

Precondition: Robot is in monitoring mode.

Steps:

1. Robot detects vibrations using 3-axis Accelerometer.
2. If vibration exceeds threshold, robot triggers an alert.
3. Robot stops moving if necessary.

Outcome: Earthquake detected, alert issued, robot stops.

Use Case: Path Following

Actor: Robot

Goal: Follow predefined paths.

Precondition: Robot is ready to follow a path.

Steps:

1. Robot follows path using Edge and Line Sensors.
2. Robot adjusts to stay on the path.

Outcome: Robot successfully follows the path.

9. Glossary

• Feature: A high-level capability of the Smart Rescue Robot.

Example: “Detect and avoid obstacles.”

• User Story: A description of a user’s need or requirement for the robot.

Example: “As a user, I want the robot to detect and avoid obstacles to ensure safe navigation.”

• Scenario: Describes different ways the robot interacts with its environment or users.

Example: “The robot detects a fire and triggers an alert.”

• Acceptance Criteria: A set of conditions that determine when a user story is complete.

Example: “Given the robot detects an obstacle using the Ultrasonic Sensor, it must change its path and continue moving.”

• Functional Requirement: Specific actions or features the robot must support.

Example: “The robot must detect fires using an Infrared Sensor and trigger an alert.”

• Non-Functional Requirement: Describes the system’s quality attributes, like performance, safety, and efficiency.

Example: “The robot must operate for at least 2 hours on a fully charged battery.”

---

10. Implementation plan :

The Third Week	Project requirements
The Fourth Week	Priority of feathers , User Stories and Acceptance criteria
The Fifth Week	Walk the specified path and Avoid obstacles
The Sixth Week	Program the robot to deal with natural disasters
The Seventh Week	Activate the warning lights
The eighth Week	Activate the warning sound for danger
The ninth Week	Test the robot