Requirement Documentation: Simple Security System Application

Objective: Create a simple security system that captures images using a camera and logs events in a JSON file.

User Stories:

1. **User Story 1: Event Logging** (must have)
    a. **As a** security system administrator,
    b. **I want** to log events with details such as timestamp, event type, and image filename,
    c. **So that** I can keep a record of all activities detected by the system.

    Acceptance Criteria:
    1. Events are stored in a JSON file.
    2. Each event includes a timestamp, event type, and image filename.

2. **User Story 2: Camera Capture** (must have)
    a. **As a** security system administrator,
    b. **I want** the system to capture an image by accessing a camera feed via HTTP, when an event occurs,
    c. **So that** I can visually verify the event.

    Acceptance Criteria:
    1. The system captures an image using the computer's camera.
    2. The captured image is saved with a timestamped filename.

3. **User Story 3: Error Handling** (should have)
    a. **As a** developer,
    b. **I want** the system to handle errors gracefully,
    c. **So that** it can continue to operate or provide meaningful error messages.

    Acceptance Criteria:
    1. The system logs errors to a log file.
    2. The system provides clear error messages when the camera cannot be accessed, or an image cannot be captured.

4. **User Story 4: Unit Testing** (could have)
    a. **As a** developer,
    b. **I want** to create unit tests for the event logging and image capture functions,
    c. **So that** I can ensure the system works correctly.

    Acceptance Criteria:
    1. Unit tests are written key functions.
    2. Tests verify that events are logged correctly, and images are captured successfully.

5. **User Story 5: Logging** (could have)

a. **As a** security system administrator,
b. **I want** to log system activities and errors,
c. **So that** I can review the system's performance and troubleshoot issues.

Acceptance Criteria:

1. The system logs activities and errors to a log file.
2. Log entries include timestamps and relevant messages.

6. **User Story 6: Documentation of AI Assistant Use** (must have)
    a. **As a** developer,
    b. **I want** to document how an AI assistant was used in the development process,
    c. **So that** I can provide transparency and credit for the assistance received.

    Acceptance Criteria:

    1. A section in the documentation describes how the AI assistant was used.
    2. Examples of AI-generated code snippets or suggestions are included.

Functional Requirements:

- The system must use a dictionary to store event information.
- The system must save and load event data from a JSON file.
- The system must capture images using the computer's camera.
- The system must handle exceptions and log errors.
- The system must include unit tests for key functions.
- The system must log activities and errors to a log file.

Non-Functional Requirements:

- The system should be easy to set up and run.
- The system should provide clear and informative log messages.
- The system should handle errors gracefully without crashing.
- The system should have meaningful comments