**🔹 1. Project Overview**

**Project Name:** Hawkeye Security Software  
**Description:** A customizable school security platform that uses camera systems to detect weapons and alert school personnel.

**🔹 2. Scope of Testing**

**In Scope:**

* User and camera management
* Member access control
* Weapon detection alert system
* Snapshot viewing functionality

**Out of Scope:**

* Third-party integrations (future phase)
* Mobile platform (if not yet implemented)

**🔹 3. Objectives**

* Ensure relational integrity across all database tables
* Validate all foreign key and many-to-many relationships
* Confirm alert triggering and camera linking are functional

**🔹 4. Types of Testing**

* Functional Testing
* Database Integrity Testing
* UI/UX Testing (Admin Panel)
* Security Testing (Access Control)
* Regression Testing

**🔹 5. Environment Setup**

* Web App with Admin Panel
* Simulated or live camera feeds
* Email/SMS alert channel (test mode)

**🔹 6. Key Entities (from ERD)**

* User (owns Cameras)
* Camera (linked to User, Members, Alerts)
* Member (linked to multiple Cameras)
* Alert (linked to Cameras)
* Snapshot (URL per camera)

**🔹 7. Sample Test Cases**

**TC001 – Create User and Assign Camera**

* **Steps:**
  1. Create a user with name, phone, email, and password
  2. Add a camera and assign it to the user
* **Expected Result:** User and camera are stored with valid FK link

**TC002 – Link Camera to Multiple Members**

* **Steps:**
  1. Create two members
  2. Link both to a single camera
* **Expected Result:** camera\_member table contains correct mappings

**TC003 – Generate Alert for Camera**

* **Steps:**
  1. Insert a new alert with message/type/timestamp
  2. Link alert to a camera
* **Expected Result:** camera\_alert table reflects correct relationship

**TC004 – Validate Snapshot URL**

* **Steps:**
  1. Access camera snapshot\_url
  2. Ensure image is served/displayed
* **Expected Result:** Snapshot URL loads a valid image

**TC005 – Member Access Control**

* **Steps:**
  1. Log in as a member
  2. Query cameras linked to member
* **Expected Result:** Only cameras linked via camera\_member are visible

**🔹 8. Data Integrity Tests**

* **FK Validation:** Cannot delete user with active cameras
* **Cascade Deletes:** Deleting camera removes related entries (if enabled)
* **M:M Validity:** camera\_member and camera\_alert junctions reject duplicates

**🔹 9. Entry & Exit Criteria**

**Entry:**

* Dev environment is live
* Database schema is deployed with sample data

**Exit:**

* All test cases pass
* No critical bugs remain
* Stakeholder sign-off

**🔹 10. Deliverables**

* Test Plan Document (this file)
* Test Cases Sheet
* Screenshots or DB snapshots for test proof
* Bug report logs