# CS350 Team 1 – Full Evaluation Report

**CS350 Team 1 – Full Evaluation Report 1**

Executive Summary 2

**Part 1: Documentation (21 points) 2**

1.1 User Manual (9 points) 2

A. Installation & Environment Setup (3 points) 2

B. Feature Walkthrough (4 points) 3

C. Troubleshooting (2 points) 3

1.2 Consistent Documentation Style (12 points) 3

Documentation Total: 16 / 21 points (76.2%) 4

**Part 2: Implementation (44 points) 4**

2.1 Style & Readability (10 points) 4

2.2 Functional Accuracy (26 points) 4

A. Functional Requirements (20 points) 4

B. Non-Functional Requirements (3 points) 5

C. Implicit Requirements (3 points) 5

2.3 Implementation Details (8 points) 6

A. Defensive Logic (4 points) 6

B. Error Message Quality (4 points) 6

Implementation Total: 26.5 / 44 points (60.2%) 6

**Part 3: Testing (35 points) 6**

3.1 Test Plan (6 points) 6

3.2 Test Case Specification (17 points) 7

Format & Techniques 7

Coverage (6 points) 7

3.3 Requirements Traceability (12 points) 7

Testing Total: 15 / 35 points (42.9%) 7

**Final Summary 8**

**Key Findings 8**

Strengths 8

Critical Weaknesses 9

**Recommendations 9**

Immediate (Critical) 9

Short-Term 9

Medium-Term 9

**Project:** SafeHome Security System (Team1)

**Evaluation Date:** 2025-12-04

**Total Points:** 100

## Executive Summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Points** | **Score** | **Percentage** |
| **Documentation** | 21 | **16.0** | 76.2% |
| **Implementation** | 44 | **26.5** | 60.2% |
| **Testing** | 35 | **15.0** | 42.9% |
| **Total** | **100** | **57.5** | **57.5%** |

**Overall:** The SafeHome project demonstrates a well-structured codebase with good coding practices and generally follows the SRS and SDS specifications. However, **several critical bugs** were identified during evaluation, including input validation issues (DoS vulnerability), logic errors (silent failures), and a lack of testing techniques (EP/BVA) which allowed over 25 bugs to slip through.

# Part 1: Documentation (21 points)

## 1.1 User Manual (9 points)

### A. Installation & Environment Setup (3 points)

**Score: 2 / 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criterion** | **Points** | **Status** | **Deduction** |
| Step-by-step instructions | 1 | ✅ Present | - |
| Complete dependency list | 1 | ❌ **Incomplete** | **-1** |
| Correct file paths | 1 | ✅ Present | - |

**Critical Issue:** requirements.txt is missing **Pillow (PIL)**. The application **crashes immediately** on startup for new users following the guide.

### B. Feature Walkthrough (4 points)

**Score: 2 / 4**

|  |  |  |  |
| --- | --- | --- | --- |
| **Criterion** | **Points** | **Status** | **Deduction** |
| Scenario-based workflow | 1.5 | ⚠️ Partial | **-0.5** |
| Non-technical language | 1 | ⚠️ Partial | **-0.5** |
| Feature description accuracy | 1.5 | ❌ Failed | **-1.0** |

**Major Discrepancy:**

* **HOME Mode:** The README says Button 8 "ARMS" the system, but the code actually **DISARMS** it.
* **Missing Docs:** No guide for "Sensor Trigger Simulation" (feature not implemented) or simultaneous PIN changes.

### C. Troubleshooting (2 points)

**Score: 1 / 2**

* ✅ Lists common generic issues.
* ❌ **Fails to cover actual system bugs** (e.g., Pillow crash, silent PIN failure, camera password flows). (-1 pt)

## 1.2 Consistent Documentation Style (12 points)

**Score: 10 / 12**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Points** | **Score** | **Status** | **Note** |
| **A. Cover Page** | 1 | 1 | ✅ | - |
| **B. Table of Contents** | 2 | 2 | ✅ | - |
| **C. Page Numbering** | 1 | 1 | ✅ | - |
| **D. Versioning** | 2 | 0 | ❌ | **No version info or revision history** |
| **E. Headings** | 3 | 3 | ✅ | - |
| **F. Formatting** | 2 | 2 | ✅ | - |
| **G. Appendix** | 1 | 1 | ✅ | - |

### 📉 Documentation Total: 16 / 21 points (76.2%)

|  |  |
| --- | --- |
| **Major Deductions** | **Points** |
| Missing Pillow dependency (Crash) | -1.0 |
| Feature description mismatch (Home Mode) | -1.0 |
| Incomplete scenarios & troubleshooting | -1.0 |
| Versioning history absent | -2.0 |
| **Total Deduction** | **-5.0** |

# Part 2: Implementation (44 points)

## 2.1 Style & Readability (10 points)

* **Style Guide (5/6):** PEP 8 followed well, but **comment styles are inconsistent** across modules (-1 pt).
* **Internal Doc (3/4):** Good overall, but some public methods lack docstrings and parameter descriptions (-1 pt).

## 2.2 Functional Accuracy (26 points)

### A. Functional Requirements (20 points)

**Score: 12 / 20**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Issue Description** | **Deduction** |
| **Camera Password** | **Logic Error:** Asks for NEW password before verifying CURRENT one. | -2.0 |
| **Guest PIN** | **Silent Failure:** UI shows "Success" but data is **NOT saved** to DB. | -2.0 |
| **Panic Button** | **Logic Error:** Works even when system is powered OFF. | -1.0 |
| **PIN Security** | **Security Violation:** Allows Master and Guest PINs to be identical. | -1.0 |
| **Sensor Trigger** | **Missing Feature:** Sensor trigger handling code does not exist. | -2.0 |
| **Monitoring** | **Missing Feature:** No call to monitoring service on Panic. | -1.5 |
| **Shutdown** | **UI Gap:** Missing "Please wait" messages. | -0.5 |

### B. Non-Functional Requirements (3 points)

**Score: 1 / 3**

* ❌ **Performance (-1):** No input length limits. Vulnerable to **DoS attacks** (tested with 10k+ chars).
* ❌ **Security (-0.5):** PIN uniqueness not enforced.

### C. Implicit Requirements (3 points)

**Score: 1 / 3**

* ❌ **Silent Failures (-1):** Simultaneous Master+Guest PIN change **bypasses validation** and **fails to save** without any error message.
* ❌ **Error Recovery (-0.5):** Some states require restart.

## 2.3 Implementation Details (8 points)

### A. Defensive Logic (4 points)

**Score: 2 / 4**

* **Validation Mismatch:** UI allows delay > 60, Backend rejects it (-0.5).
* **Missing Limits:** Zone Name, User ID, and Passwords accept **unlimited length strings** (-1.5).

### B. Error Message Quality (4 points)

**Score: 2.5 / 4**

* **Silent Failures:** Users are misled by success messages when operations fail (-1.0).
* **Guidance:** Some errors lack actionable steps (-0.5).

### Implementation Total: 26.5 / 44 points (60.2%)

|  |  |
| --- | --- |
| **Major Critical Flaws** | **Impact** |
| **Simultaneous PIN Failure** | **Critical:** Data loss + Security bypass + No user feedback. |
| **Sensor Trigger Missing** | **Critical:** Core security logic is unimplemented. |
| **DoS Vulnerability** | **High:** Unlimited input length allows crashing the system. |
| **Camera Logic Reversal** | **Medium:** Wrong security flow for password changes. |

# Part 3: Testing (35 points)

## 3.1 Test Plan (6 points)

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Points** | **Score** | **Notes** |
| Scope & Strategy | 4 | **3** | In/Out scope boundaries not defined. |
| Environment & Tools | 2 | **1** | OS, Browser, Hardware specs missing. |

## 3.2 Test Case Specification (17 points)

### Format & Techniques

**Score: 3 / 11**

* ❌ **Input Specification:** "Test with fixture" is vague. Concrete values (e.g., "1234") missing. (-1 pt)
* ❌ **Expected Result:** "Assertion passes" is not a valid oracle. (-1 pt)
* ❌ **Testing Techniques (0/6):**
  + **No Equivalence Partitioning:** Missed length limit bugs.
  + **No Boundary Value Analysis:** Missed Delay=60 mismatch.
  + **No Negative Testing:** Missed silent PIN failures.

### Coverage (6 points)

**Score: 5 / 6**

* Detailed reports provided, but multiple critical files (e.g., button\_actions.py) have **0% coverage**. (-1 pt)

## 3.3 Requirements Traceability (12 points)

|  |  |  |  |
| --- | --- | --- | --- |
| **Matrix** | **Points** | **Score** | **Notes** |
| **SRS Traceability** | 6 | **1** | No matrix; Empty references; No reverse trace. |
| **SDS Traceability** | 6 | **2** | No mapping table; Incomplete component coverage. |

### Testing Total: 15 / 35 points (42.9%)

|  |  |
| --- | --- |
| **Deduction** | **Reason** |
| **Techniques Ignored** | No EP/BVA/Negative testing led to **25+ uncaught bugs**. |
| **Traceability Absent** | Impossible to verify if all requirements are tested. |
| **Specification Vague** | Test cases lack concrete data and oracles. |

# Final Summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **Subcategory** | **Points** | **Score** |
| **Documentation** | User Manual | 9 | 5.0 |
|  | Style | 12 | 10.0 |
| **Implementation** | Style & Doc | 10 | 8.0 |
|  | Functional Accuracy | 26 | 14.0 |
|  | Implementation Details | 8 | 4.5 |
| **Testing** | Plan & Env | 6 | 4.0 |
|  | Case Spec & Coverage | 17 | 8.0 |
|  | Traceability | 12 | 3.0 |
| **Total** |  | **100** | **57.5** |

# Key Findings

## Strengths

1. **Code Quality:** Clean architecture (Services/Controllers) and PEP 8 compliance.
2. **Core Features:** Safety Zone management and basic Control Panel flows work well.
3. **Visuals:** Professional documentation styling and detailed coverage reports.

## Critical Weaknesses

|  |  |
| --- | --- |
| **Issue** | **Impact** |
| ❌ **Installation Crash** | Pillow missing from requirements causes immediate crash. |
| ❌ **Silent Data Loss** | Simultaneous PIN changes fail silently and bypass validation. |
| ❌ **Security Gaps** | No input length limits (DoS) & Master/Guest PINs can be identical. |
| ❌ **Missing Core Logic** | Sensor triggers are not implemented; Panic works when OFF. |
| ❌ **Testing Failure** | Lack of formal testing techniques allowed 25+ bugs to remain. |

# Recommendations

### Immediate (Critical)

1. **Fix Dependencies:** Add Pillow to requirements.txt.
2. **Implement Sensor Logic:** Add missing code for sensor triggers.
3. **Fix PIN Logic:** Ensure simultaneous PIN changes are validated and saved atomically.
4. **Correct Docs:** Align README "HOME mode" description with actual code behavior.

### Short-Term

1. **Security Hardening:** Enforce "Master ≠ Guest" rule and add input length limits (User ID, Zone Name).
2. **Fix Camera Flow:** Verify current password *before* accepting new one.
3. **Traceability:** Create SRS/SDS to Test mapping matrices.

### Medium-Term

1. **Apply Testing Techniques:** Refactor tests using Equivalence Partitioning and Boundary Value Analysis.
2. **Versioning:** Add revision history to all documents.
3. **Coverage:** target 0% coverage files (e.g., button\_actions.py).

**End of Team1 Full Evaluation Report**