

Internship Progress Report – 1

Name: Saiprakash Bollam

Internship Role: Research Intern

Duration: 24th December 2025 - 6th January 2026

Organization: Computer Science Department, Binghamton University

Supervisor: Zerksis Umrigar

Email: umrigar@binghamton.edu

1. Introduction

During this week, the primary focus was on **debugging and fixing failing automated test cases** within the Smart Contact Manager application. Following multiple framework upgrades and security improvements in earlier weeks, several integration and controller-level tests began failing due to changes in request handling, validation logic, and response behavior.

AI-powered tools were extensively used to **analyze test failures, trace root causes, and iteratively refine both application code and test cases**, ultimately restoring full test-suite stability.

2. Objectives

- Identify failing test cases and analyze root causes.
 - Restore full test coverage without compromising application security or behavior.
 - Align controller logic with Spring Boot 3.x testing best practices.
 - Improve test reliability and realism.
 - Leverage AI assistance to accelerate debugging and reduce manual effort.
-

3. Issues Identified

The main failing test was related to the **contact image upload functionality** in the `ContactControllerTest`.

Key Problems Identified

- `ImageService.uploadImage()` was not being invoked during test execution.
- Multipart file handling behaved differently in **MockMvc test environments**.
- Test image data failed custom file validation due to invalid file signatures.
- HTTP status expectations in tests did not align with actual controller behavior.

- Redirect handling caused mismatch between expected and actual responses.
-

4. Fixes Implemented

4.1 Controller-Level Fixes

- Replaced the use of `MultipartHttpServletRequest` with **Spring-standard form binding** via `ContactForm.getContactImage()`.
- Ensured multipart file handling aligns with MockMvc behavior.
- Maintained security and validation logic without introducing test-only shortcuts.

4.2 Test-Level Fixes

- Updated failing tests to use **valid image byte signatures** (JPEG magic bytes) instead of plain text.
 - Adjusted HTTP status expectations to match actual controller responses.
 - Ensured realistic test inputs that pass custom `FileValidator` checks.
 - Preserved correctness of assertions while improving test robustness.
-

5. AI's Role in Fixing Tests

AI tools such as **GitHub Copilot** and **ChatGPT** played a critical role throughout this process:

- Interpreted Maven test failure logs and stack traces efficiently.
- Guided root-cause analysis instead of surface-level fixes.
- Identified mismatches between controller logic and test expectations.
- Highlighted the impact of custom validators on test data.

- Suggested best practices for multipart file testing in Spring Boot.
- Helped iteratively validate fixes through repeated test execution.

AI assistance significantly reduced trial-and-error debugging and enabled a **methodical, structured resolution process**.

6. Insights on AI Usage

Advantages

- Faster identification of root causes in complex failures.
- Reduced cognitive load when navigating large codebases.
- High-quality explanations of Spring testing internals.
- Improved confidence in final fixes due to systematic reasoning.
- Enhanced learning of real-world testing edge cases.

Limitations

- AI suggestions required **manual verification** before implementation.
 - Some recommendations needed contextual refinement.
 - Runtime behavior still required local test execution for confirmation.
-

7. Outcomes & Learnings

- All **85 automated tests passed successfully** with zero failures.
- Improved understanding of multipart handling in Spring MVC tests.

- Strengthened alignment between application behavior and test expectations.
- Reinforced importance of **realistic test data** when validators are involved.
- Demonstrated how AI can act as a **debugging partner rather than a replacement**.

This week highlighted the importance of maintaining **test reliability after major architectural and security changes**.