Progress Report – Week 1

Name: Saiprakash Bollam

Internship Role: Research Intern

Duration: 25th September 2025 - 8th October 2025

Organization: Computer Science Department, Binghamton University

Supervisor: Zerksis Umrigar

Email: umrigar@binghamton.edu

1. Introduction

This week focused on refactoring and improving the architectural structure of the Smart Contact Manager project. The primary task was to migrate business logic from controllers to service layers, ensuring better code separation, maintainability, and adherence to software design principles. AI tools were utilized extensively to review existing code patterns, identify redundant logic, and assist in restructuring methods for improved modularity and clarity.

2. Objectives

- Move business logic from controllers into service layer methods.

- Eliminate redundancy and improve code readability.

- Align the project structure with Spring Boot's MVC architecture standards.

- Refactor methods for scalability and easier unit testing.

- Use AI for automated code optimization and structural recommendations.

3. Key Tasks and Implementations

- Identified and extracted complex business logic from controller methods like ContactController, UserController, and PasswordController.

- Created corresponding service-layer methods in ContactServiceImpl and UserServiceImpl.

- Improved reusability by centralizing logic for form validation, user verification, and file handling.

- Simplified controllers to act purely as intermediaries between the UI and service layer.

- Implemented dependency injection for services, improving testing efficiency.

- Cleaned up commented or duplicate code, standardizing naming conventions.

- Enhanced error handling using custom exception classes and global advice handlers.

4. AI's Role in Development

AI was instrumental throughout the process:

- Code Review and Refactoring: Suggested optimal places to relocate business logic.

- Best Practices Enforcement: Recommended improved naming conventions and modular designs.

- Error Prevention: Helped detect unhandled null pointers and unnecessary dependencies.

- Debugging Assistance: Analyzed stack traces after refactor-induced errors and proposed fixes.

- Optimization: Advised consolidating repetitive data access logic into reusable service components.

AI effectively functioned as both a technical consultant and debugging assistant, accelerating refactor completion and enhancing design quality.

5. Challenges Faced

- Managing multiple dependent services while ensuring backward compatibility.

- Adjusting controller-level test cases after logic migration.

- Handling circular dependency issues during service injection.

- Ensuring proper exception propagation from service to controller.

- Adapting AI's generic recommendations to fit the project's existing architecture.

6. Insights on AI Usage

Pros:

- Greatly reduced the manual effort of code review and restructuring.

- Suggested improvements aligned with SOLID and MVC principles.

- Simplified debugging and improved project maintainability.

- Accelerated the refactoring process through context-aware prompts.

Cons:

- AI sometimes suggested over-generalized refactors not fully compatible with existing patterns.

- Required developer judgment to validate and adapt AI-generated code.

- Occasionally missed edge cases in dependency injection and bean lifecycle management.

7. Outcome and Learning

By the end of the week, controller classes were leaner, and all business logic was successfully moved to their appropriate service classes. The project is now more maintainable, scalable, and test-friendly. This phase reinforced understanding of Spring Boot architecture, layered design, and the significance of modularity. AI tools proved invaluable in automating refactoring suggestions while highlighting the importance of manual oversight.

8. Future Work

- Write unit tests for refactored service methods.

- Continue optimizing service interfaces for reuse.

- Implement API documentation (Swagger) for improved clarity.

- Begin preparing for final report compilation and presentation.