

# Internship Progress Report – Week 2

**Name:** Saiprakash Bollam

**Internship Role:** Research Intern

**Duration:** 9th October 2025 - 22th October 2025

**Organization:** Computer Science Department, Binghamton University

**Supervisor:** Zerksis Umrigar

**Email:** umrigar@binghamton.edu

# 1. Introduction

This week's focus was on upgrading the project's Spring Framework version and refactoring dependency configurations to align with the latest stable release. The objective was to enhance security, performance, and compatibility while ensuring that existing features such as authentication, QR code integration, and testing frameworks continued to function seamlessly after the upgrade. The process demanded deep analysis of dependency conflicts, migration of deprecated configurations, and testing to maintain consistent application behavior across modules.

---

## 2. Objectives

- Upgrade Spring Boot and Spring Framework versions to the latest LTS release.
  - Ensure backward compatibility for core components and service layers.
  - Update deprecated annotations, configurations, and dependency management files.
  - Resolve conflicts between updated dependencies and existing modules.
  - Use AI assistance to automate refactor suggestions, fix errors, and optimize dependency resolution.
- 

## 3. Key Tasks and Implementations

- Upgraded from Spring Boot 2.x to Spring Boot 3.x, aligning dependencies with Jakarta namespace changes.
- Migrated JPA and Hibernate configurations from `javax.*` to `jakarta.*` packages.
- Updated security configurations to align with the new Spring Security 6 structure.
- Fixed issues related to deprecated annotations in Controllers and Services.
- Updated Maven dependencies including Spring DevTools, Lombok, Thymeleaf, and testing libraries.

- Reconfigured application properties for new environment profiles and encryption support.
- 

## 4. AI's Role in Development

AI tools played a critical role in streamlining the upgrade process:

- **Dependency Management:** Suggested compatible library versions and assisted in resolving Maven conflicts.
  - **Code Migration:** Identified deprecated annotations and provided replacement patterns.
  - **Error Resolution:** Helped interpret migration-related stack traces and fixed namespace mismatches.
  - **Performance Optimization:** Recommended minor refactors to improve build times and reduce memory overhead.
  - **Testing Guidance:** Ensured test cases were compatible with updated framework classes.
- 

## 5. Challenges Faced

- Encountered multiple version conflicts among transitive dependencies, particularly between Spring Security and Hibernate.
  - Some deprecated configurations were not directly replaceable and required code restructuring.
  - Maven build failures occurred due to missing Jakarta dependencies after the upgrade.
  - Thymeleaf templates initially failed due to namespace inconsistencies in expression handling.
  - Ensuring consistent behavior across local, test, and production profiles after upgrading core components.
-

## 6. Insights on AI Usage

### Pros:

- Accelerated dependency migration by identifying compatible versions.
- Helped automate repetitive refactor tasks (e.g., javax to jakarta conversions).
- Provided detailed debugging assistance for complex Maven errors.
- Enhanced understanding of modern Spring Boot and Jakarta ecosystem practices.

### Cons:

- AI occasionally recommended experimental or unstable dependency versions.
  - Some suggestions required manual verification to ensure alignment with project stability.
  - Limited accuracy in handling version-specific issues in complex dependency graphs.
- 

## 7. Outcome and Learning

By the end of this upgrade cycle, the Smart Contact Manager successfully transitioned to the latest Spring Boot 3.x and Spring Framework 6 ecosystem. The migration resulted in improved security, better performance, and long-term maintainability. AI proved invaluable in resolving compatibility issues and accelerating the upgrade process. This week strengthened my understanding of dependency management, version control, and migration best practices in enterprise-grade applications.