iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70774

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Review Paper on NextGen Banking: A Secure and Scalable Banking System

Sachin Tiwari[1], Ankita Dubey[2], Udit Narayan Tiwari[3], Anujsen Yadav[4]
[1, 2, 3, 4]*Scholars,* [2]*Assistant Professor, Department of Information Technology, Goel Institute of Technology and Management, Lucknow, India*

Abstract: ***This research introduces NextGen Banking, an innovative web-based banking system that prioritizes security, scalability, and superior user experience. Leveraging the Spring Boot framework with MVC architecture and PostgreSQL integration, the platform automates essential banking operations—including user account management, transactions, loans, and card services. The study delves into the system's technical architecture, layered design, and robust security mechanisms, while also addressing key implementation challenges and solutions. Designed for modern financial institutions, NextGen Banking offers a seamless, efficient, and future-proof alternative to traditional banking systems.***
Keywords: Spring Boot, Banking System, Spring Security, MVC Architecture, PostgreSQL, Digital Transactions

I. INTRODUCTION

A. Introduction and Background

With the advent of fintech and digital transformation, traditional banking systems are being replaced with more intelligent, scalable, and user-friendly platforms. However, many legacy systems still suffer from poor integration, weak security, and low scalability. NextGen Banking proposes a complete solution using Spring Boot and Java technologies that bring modularity, data consistency, and modern security practices into the core banking experience.

B. Need for a Smart Desktop Assistant

Customers today demand real-time access, faster transactions, and complete control over their financial data. Most small-scale banking solutions fail to meet these expectations. NextGen Banking provides a unified platform with a clean user interface, seamless backend operations, and comprehensive features like card management, loan tracking, and transaction history — all while prioritizing security.

C. Research Objectives

- To build a secure, full-stack banking application using Java Spring Boot.
- To implement database management with entity relationships using JPA and PostgreSQL.
- To develop core modules including user registration, account handling, card services, and transaction monitoring.
- To explore secure authentication using Spring Security.

II. LITERATURE REVIEW

Several banking solutions exist, such as online banking portals of ICICI, HDFC, and other national and international institutions. These solutions often utilize enterprise frameworks like Java EE or .NET. While effective, they are often closed-source and hard to extend or study for academic purposes. Open-source systems like Apache Fineract provide microfinance software but have steep learning curves. Spring Boot has emerged as a powerful framework for building scalable enterprise applications. Combined with tools like Spring Security and JPA, it offers clean abstraction, making it ideal for academic prototypes of real-world banking systems. Previous student projects often lack transactional handling or fail to implement proper security standards, which NextGen Banking resolves.

III. SYSTEM ARCHITECTURE / PROPOSED METHODOLOGY

The overall design of JARVIS follows a modular structure that integrates multiple technologies to enable smooth and efficient interaction between the user and the system. The architecture is divided into interconnected layers, each responsible for specific tasks such as voice input, processing, command execution, and system response.

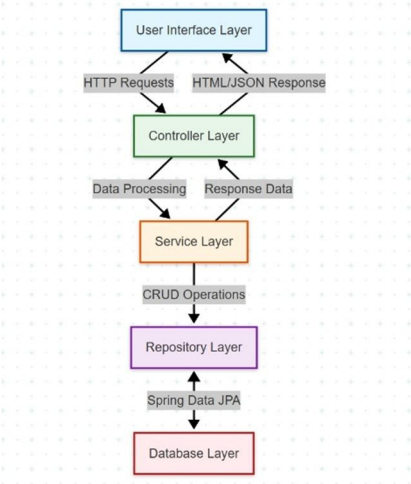
©IJRASET: All Rights are Reserved

A. *Design Overview*

The application follows a three-layered architecture:

- Presentation Layer: JSP pages rendered via controllers
- Business Logic Layer: Services handling use-cases
- Data Access Layer: Spring Data JPA repositories interacting with PostgreSQL

B. *System Flow Diagram*



C. *Technologies Used*

- Spring Boot
- Spring MVC
- Spring Security
- Spring Data JPA
- PostgreSQL
- JSP, HTML, CSS
- IntelliJ IDEA, GitHub

IV. IMPLEMENTATION DETAILS

This section describes the technical foundation of the JARVIS desktop assistant, including tools, libraries, and how the system is structured.

A. *Software and Hardware Requirements*

- OS: Windows 10 / Linux
- RAM: 8 GB or higher
- Java Version: 17 or above
- Spring Boot: 3.x
- PostgreSQL: 14+
- IDE: IntelliJ IDEA

B. *Libraries / APIs Used*

- spring-boot-starter-web
- spring-boot-starter-data-jpa
- spring-boot-starter-security
- postgresql JDBC Driver
- thymeleaf or JSP engine

V. APPLICATIONS AND USE CASES

A. *Real-World Scenarios*

- Banking Institutions: Core banking modules for digital account access
- Startups: Financial services firms with modern web platforms
- Educational Tools: College projects on full-stack enterprise applications

B. *Productivity Benefits*

- Quick transaction handling
- Secure and scalable database
- Easy admin management of accounts and loans

C. *Security Enhancements*

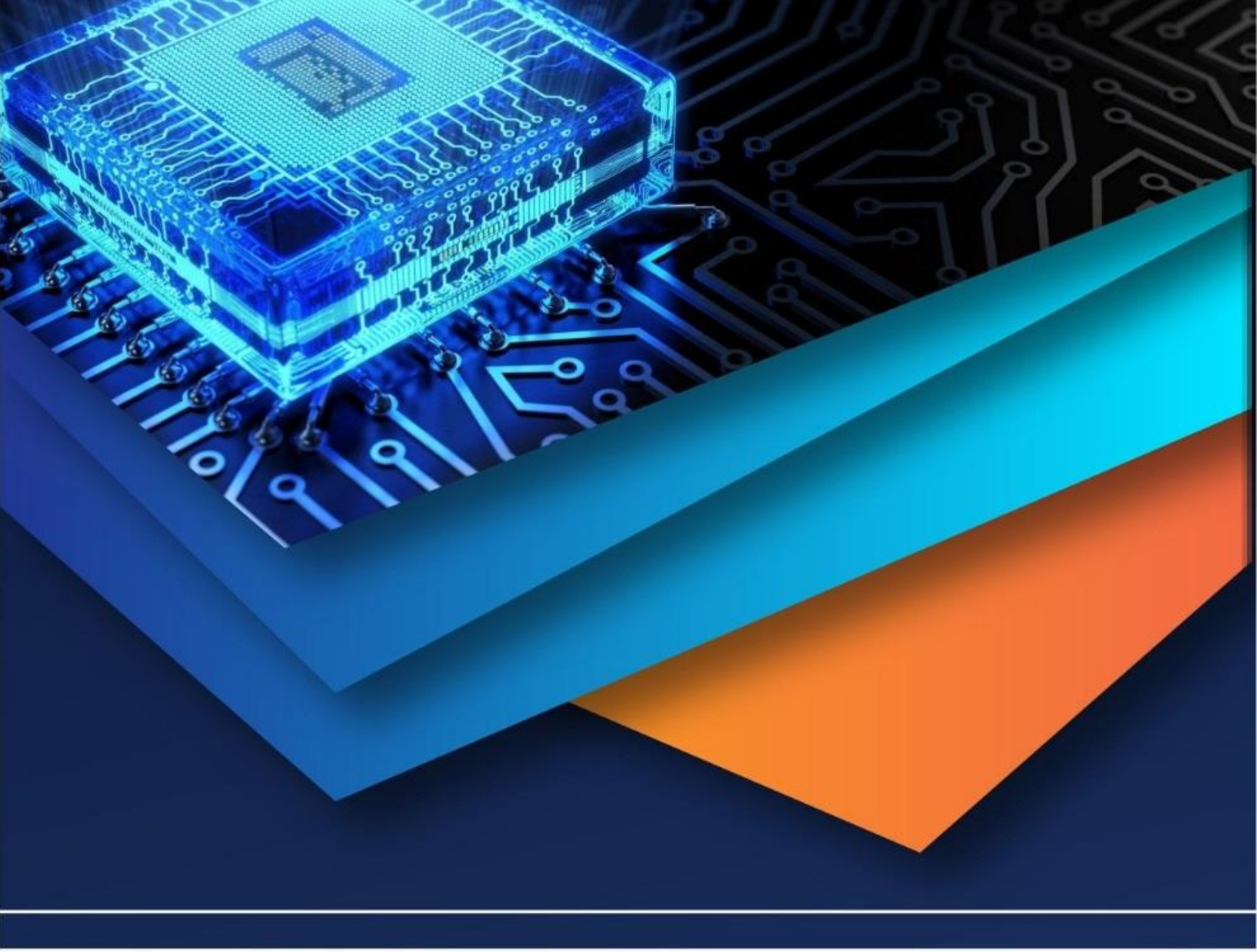
- Password hashing with BCrypt
- Role-based access (User vs Admin)
- Prevention of SQL injection via ORM

VI. CONCLUSION

NextGen Banking bridges the gap between theoretical knowledge and enterprise-level implementation in the fintech domain. It showcases how Spring Boot, combined with robust database design and security practices, can create a complete digital banking system. The modularity and extensibility make it suitable for future upgrades like payment gateway integration and AI-based fraud detection. As a final-year project, it holds academic, professional, and practical value.

REFERENCES

[1] Özer, C. B. (2022). *Spring Boot Online Banking REST API*. Graduation Thesis, Department of Computer Engineering, Adana. Retrieved from ResearchGate
[2] Deshmukh, K. R., & Pursharthi, B. (2023). *Building a Modern Banking System: Implementation of a Java Spring Boot and Angular Framework-Based Solution*. International Journal of Research Publication and Reviews, 4(6), 6876–6880. Retrieved from ResearchGate+1IJRPR+1
[3] Goel, L. (2024). *Microservices Architecture with Spring Boot for Financial Services*. International Journal of Creative Research Thoughts, 12(6). Retrieved from IJCRT
[4] Thalla, R. R., & Elluri, M. (2022). *Banking Application (Backend Development Using Spring Boot)*. International Research Journal of Modernization in Engineering Technology and Science, 4(6). Retrieved from IRJMETS Journal+1Semantic Scholar+1
[5] Spring Boot Documentation. (n.d.). *Spring Boot Reference Guide*. Retrieved from Spring
[6] PostgreSQL Global Development Group. (n.d.). *PostgreSQL 14 Documentation*. Retrieved from PostgreSQL
[7] Ali Ihsan Hashas. (2024). *Core Banking Application with Spring Boot: Microservices Architecture and Immutability Principles*. Retrieved from Medium
[8] Page, J. L. (2024). *Building a Core Banking Microservice with Spring Boot and MongoDB*. Retrieved from Medium
[9] Bansikah, J. (2024). *Building a Simple CRUD Banking Application with Spring Boot and Swagger UI*. Retrieved from DEV Community
[10] Abhi9720. (2023). *Open-Source Project: Banking Portal REST API Using Spring Boot*. Retrieved from DEV Community
[11] Mohammed-eid35. (n.d.). *Bank Management System - Spring Boot*. Retrieved from GitHub
[12] Paul Dragoslav. (n.d.). *Spring Boot Banking*. Retrieved from GitHub
[13] Kartik1502. (n.d.). *Spring Boot Microservices Banking Application*. Retrieved from GitHub
[14] Phelumie. (n.d.). *Spring Boot Banking API*. Retrieved from GitHub
[15] Java Guides. (2023). *Banking Application Using Spring Boot*. Retrieved from Java Guides
[16] Neeraj Kumar. (2024). *A Spring Boot Banking Application*. Retrieved from LinkedIn
[17] Spring Security Documentation. (n.d.). *Spring Security Reference*. Retrieved from Spring
[18] World Bank Group. (2020). *Digital Financial Services: Challenges and Opportunities*. Retrieved from World Bank
[19] IBM Developer. (2023). *Building Secure REST APIs with Spring Boot and JWT*. Retrieved from IBM Developer
[20] JHipster. (n.d.). *JHipster - Wikipedia*. Retrieved from en.wikipedia.org



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)