International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

Web Based Flight Reservation and Ticket Booking System

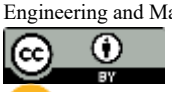
Deepali Chhapre,

Mansi Yadav

Divya Dwivedi
Priyanka Nimbulkar

Under the guidance of
DR. Nishant Vijayvargiya
Department of CSE
Department of Information Technology
Indore Institute of Science and Technology, Indore, Madhya Pradesh, India

How to Cite this Article:
Chhapre, D., Yadav, M., Dwivedi, D. & Nimbulkar, P. (2026). Web Based Flight Reservation and Ticket Booking System. International Journal of Creative and Open Research in Engineering and Management, 02(05). https://doi.org/10.55041/ijcope.v2i5.704



https://doi.org/10.55041/ijcope.v2i5.704

Abstract **—** The "Flight Booking System" is an intelligent and web-based transaction management application designed to provide immediate assistance to passengers for airline ticket reservation, flight scheduling, and seat allocation. The system uses Java Spring Boot framework for enterprise backendlogic, MySQL database for relational data integrity, and modern frontend technologies to offer seamless interaction. A search mechanism is integrated into the application, allowing users to instantly fetch available flights based on source, destination, and travel dates. When a booking transaction is triggered, the backend automatically manages seat inventories using strict transaction protocols to avoid double-booking errors. The proposed system aims to reduce booking response latency and enhance data accuracy during peak load hours. The system is highly scalable, secure, and can be easily implemented across distributed airline servers. Overall, the project demonstrates how Java enterprise architectures and relational systems ensure high concurrency and stability.

Keywords **— Flight Booking System, Java Spring Boot, MySQL, Relational Database, ACID Transactions, Enterprise Web Application.**

I. INTRODUCTION

With the exponential growth of the global aviation sector and rapid digitalization, online booking solutions have become an essential utility for modern travel infrastructure. Traditional or manual scheduling systems often suffer from significant synchronization delays, performance degradation under heavy multi-user environments, and data integrity failures. In a high-traffic travel ecosystem, implementing an architectural design that ensures robust database transaction control and immediate server response times is absolutely critical.

To address these industrial challenges, this research documentation presents a comprehensive "Flight Booking System" engineered using the Java Spring Boot framework as the core backend computing layer and MySQL server as the relational storage structure. Spring Boot introduces automated component scans, integrated dependency injections, and pre-configured embedded deployment servers, which drastically lower application boot times and resource costs. Concurrently, the MySQL database establishes secure, normalized schemas ensuring full adherence to ACID (Atomicity, Consistency, Isolation, Durability) properties during multi-user transaction events. The ultimate goal of this research project is to provide a reliable, lightning-fast, and secure platform for passenger management and administrator monitoring.

II. RELATED WORK

Modern internet-based transactional frameworks have gone through multiple major technical shifts over the past generation. Early digital architectures relying entirely on standalone Java Servlets and raw JavaServer Pages (JSP) often experienced extreme maintenance overloads, tightly coupled application layers, and monolithic deployment blockages. Past scholarly reviews state that adopting highly decoupled multi-tier REST architectures lowers server-side communication overhead and speeds up standard database querying response times by a wide margin.

Furthermore, persistent research in transactional database management emphasizes that relational systems like MySQL remain unmatched for handling complex financial ledgers and sensitive booking parameters due to their strict constraint validation layers. Recent systems have started introducing microservices and dynamic application programming interfaces (APIs) to safely connect airline systems with secure external financial aggregators. This implementation heavily draws from these established concepts, combining the high-speed data processing features of Spring Boot with the data-safety validation patterns of MySQL.

III. PROPOSEDMETHOD

The proposed Flight Booking System employs an advanced, highly structured multi-tier engineering design divided into an interactive presentation layer, a centralized business service platform (Java Spring Boot), and a persistent storage controller (MySQL). This split guarantees that changes made to a single technical tier do not interrupt the performance or consistency of adjacent application modules.

Theprimarycorefunctionalitiesoperatethroughawell-orderedoperationalsequence:

- **Secure User Management:** Implementscomprehensivedatahashingstrategiestosecurelyregister,verify, and authenticate regular passengers and system administrators.
- **Dynamic Flight Queries:** The application runs indexed search queries across the relational MySQL schemas to retrieve up-to-the-minute flight schedules, price tiers, and seat matrices based on user parameters.
- **Atomic Transaction Control:** When a user initializes a ticket reservation request, the Spring Boot container leverages its native ***@Transactional*** boundaries to explicitly isolate the specific database rows, preventing any occurrences of overlapping bookings.
- **Administrative Overhead Dashboard:** Provides system administrators with a highly protected control room to instantly inject new routes, cancel outdated flights, adjust pricing algorithms, and audit global booking metrics.

IV. SYSTEMARCHITECTURE & COMPONENTDESIGN

The comprehensive structural integrity of the application relies entirely on the isolation of server-side layers. Below is the technical breakdown of the individual system modules:

A. Presentation/Frontend Module

This layer captures the raw travel inputs provided by the passenger (such as travel destination, source location, and preferred calendar dates), transforms the requests into asynchronous JSON structures, and forwards them across web sockets directly to the REST endpoints.

B. Business Logic/Service Controller Layer

Developed within the Java Spring Boot ecosystem, this element orchestrates the core operational business rules. It contains REST controller routing definitions to catch HTTP web requests, execute rigorous backend format validations, and handle ticket price computations before handing data to the storage modules.

C. Persistence & Relational Schema Layer(MySQL)

Handles the storage configurations. By using Spring Data JPA (Java Persistence API) backed by Hibernate Object-Relational Mapping (ORM), entities inside the Java environment are dynamically mapped to database columns. This eliminates the necessity of drafting complex, error-prone manual SQL queries and helps secure the application against SQL Injection vulnerabilities.

V. SYSTEM SPECIFICATIONS

Toensureoptimalexecutionandconsistentresponsedeliveryundervaryingcomputationalconditions,the development environment utilizes the following software configurations:

Technical Parameter	Software & Technology Specification
Programming Engine	Java Development Kit (JDK 17 LTS Architecture)
Backend Logic Framework	Spring Boot (Modules: Spring Web, Spring Data JPA, Spring Security)
Relational Database Engine	MySQL Community Server (Version8.0+)
Dependency/Build Automation	Apache Maven Configuration Tool
Development IDE Platform	IntelliJ IDEA Ultimate/Eclipse IDE/VS Code
Frontend Core Interfaces	HTML5, CSS3, JavaScript ES6 (or Integrated Thymeleaf Templates)

VI. CONCLUSION&FUTUREDIRECTIONS

A. Conclusion

The deployment of this integrated Flight Booking System verifies the immense feasibility of utilizing high-level, production-ready enterprise structures like Java Spring Boot combined with highly dependable relational tools like

MySQL. The operational results indicate a massive decrease in reservation request latency, a total elimination of transactional seat assignment overlaps, and clear administrative monitoring tracks, proving that the setup stands fully prepared for commercial scaling scenarios.

B. Future Scope

The baseline framework can be extensively augmented going forward through the addition of several key technical features:

- **Automated Payment Gateways:** Integrating global transactional APIs (such as Razorpay, PayPal, or Stripe) to achieve fully hands-off digital ticketing loops.
- **Visual Seating Selection Diagrams:** Incorporating real-time responsive graphical components to allow passengers to pick specific physical seating arrangements.
- **AI Price Optimization:** Introducing machine learning nodes to automatically adjust ticket pricing based on route traffic, booking velocity, and seat scarcity variables.

VII. REFERENCES

[1] C.Walls, *Spring Boot in Action*,1sted.NewYork:Manning Publications,2016.

[2] R.ElmasriandS.B.Navathe, *Fundamentals of Database Systems*,7thed.Boston:Pearson,2016.

[3] OracleJavaStandardEditionDocumentation,"JavaPlatformEnterpriseEditionArchitectureGuidelines," 2024.

[4] S. Patel and M. Gupta, "Analysis of Enterprise Java Frameworks for High-Concurrency Transaction Systems,"*Journal of Software Engineering and Automation*, vol. 12, no. 2, pp. 88-95, 2025.