CITY MALL SMART DEIVERY

Nihareeka Meher 4th Year, Department of CSE(AI), Gandhi Institute for Technology, BPUT, India
nihareekameher2021@gift.edu.in
Professor, Department of CSE, Gandhi Institute for Technology, BPUT, India

Abstract—
This project focuses on the testing and debugging of a newly introduced **Scheduled Delivery Mode** in the City Mall e-commerce platform, a rapidly growing digital retail service in Myanmar. The feature aims to enhance user experience by allowing customers to select preferred delivery slots, addressing common complaints about unpredictable delivery times. The project follows a structured Software Testing Life Cycle (STLC), covering requirement analysis, test planning, execution, defect tracking, and reporting. Various tools and technologies including Java, Spring Boot, MySQL, and Selenium were used to ensure functionality, performance, and integration across modules such as cart, order, and payment. Manual and automated test cases were executed to validate slot availability, user interface responsiveness, and backend data consistency. Results show that the system performs reliably under real-world conditions, providing a smoother and more predictable delivery process. The feature is now production-ready, with future improvements planned for dynamic slot management and user personalization.
Keywords:
JAVA,SPRING BOOT,SQL

I. INTRODUCTION

With the rapid expansion of e-commerce in Myanmar, platforms like City Mall are striving to improve user experience by offering more flexible and reliable delivery options. One common issue faced by customers is the lack of control over delivery timings. To address this, a **Scheduled Delivery Mode** was introduced, allowing users to select preferred time slots during checkout. This project focuses on testing and debugging this feature to ensure its accuracy, usability, and integration with existing systems. A structured testing approach was adopted to validate slot booking, cancellation, and system performance under various real-world conditions.

II. LITERATURE REVIEW

The literature review explores previous research and practices related to delivery scheduling in ecommerce platforms, with a focus on slot-based delivery models. Studies emphasize the importance of customer satisfaction in last-mile logistics and the need for systems that provide time-slot flexibility and real-time feedback. Platforms like Amazon and Big Basket have implemented similar delivery features, validating their effectiveness in high-demand environments. Academic resources also support the use of the Software Testing Life Cycle (STLC) and Agile development for incremental feature deployment and defect tracking. Additionally, research highlights best practices in database design, concurrency management, and user interface responsiveness—critical for slot reservation systems. This review informed the design and implementation of City Mall's Scheduled Delivery Mode by providing guidance on system architecture, testing strategies, and user-centered design principles. The insights gathered ensured that the new feature was not only technically sound but also aligned with evolving user expectations and real-world operational challenges.

III.   **SYSTEM DESIGN**

The system design of the Scheduled Delivery Mode feature was structured using a modular, layered architecture to ensure scalability, maintainability, and seamless integration with the existing City Mall platform. The backend is built using Java and Spring Boot, adhering to the Model-View-Controller (MVC) pattern to separate concerns between the user interface, business logic, and data access layers. The delivery slot management system interacts with the cart, order, and payment modules through RESTful APIs, enabling smooth data flow. MySQL is used as the database, with normalized tables for customers, orders, delivery slots, and locations. Slot status is updated dynamically based on booking actions, with safeguards for concurrency and data consistency. The frontend, developed using HTML, CSS, JavaScript, and Bootstrap, supports responsive slot selection interfaces that adapt across devices. Real-time feedback, intuitive design, and error handling mechanisms ensure a user-friendly experience. This robust system architecture provides the foundation for reliable delivery scheduling and future feature enhancements.

IV. **IMPLEMENTATION**

The implementation of the Scheduled Delivery Mode feature in the City Mall e-commerce platform involved both frontend and backend development, closely aligned with the system architecture and testing methodologies. The backend was developed using Java and Spring Boot, enabling robust service layering and modularity. Delivery slots were managed through a structured database in MySQL, where each slot was associated with a date, time window, status, and location. Using Spring Data JPA, the application performed CRUD operations on slot records efficiently, ensuring real-time availability tracking. Slot selection was integrated into the checkout process, where the user interface—built with HTML, CSS, JavaScript, and Bootstrap—offered a calendar and time picker for users to select available delivery windows. RESTful APIs facilitated communication between the frontend and backend, handling slot retrieval, booking, updates, and cancellations. Emphasis was placed on data validation, concurrency control, and transactional safety to prevent double bookings and maintain data integrity. Frontend responsiveness was tested across devices, ensuring consistent user experience. Various testing tools such as Postman and Selenium were used to verify API responses and UI behavior. The implementation phase concluded with a successful integration of the new feature into the existing platform, ready for production deployment and aligned with customer expectations and business goals.

V. **RESULTS**

The Scheduled Delivery Mode feature was successfully tested and validated across multiple parameters, including functionality, performance, user interface responsiveness, and system integration. Functional testing confirmed that the system accurately displayed available delivery slots, prevented overbooking, and correctly linked selected slots with customer orders. The feature operated consistently across web and mobile platforms, providing a seamless user experience. Integration testing showed smooth communication between modules such as cart, payment, order tracking, and the new delivery scheduling system. Performance testing under simulated high-traffic conditions revealed that the system could handle multiple concurrent slot bookings without data conflicts or delays. All major and critical defects identified during the testing phase were resolved, and no blocking issues remained. Usability testing indicated that users found the interface intuitive and helpful. Overall, the results demonstrated that the feature met all defined objectives and was ready for deployment, significantly enhancing delivery reliability and customer satisfaction for the City Mall platform.

VI. **CONCLUSION**

The development and testing of the Scheduled Delivery Mode feature in the City Mall platform marks a significant step toward improving delivery reliability and enhancing user satisfaction in the growing e-commerce space of Myanmar. Through a structured Software Testing Life Cycle (STLC), the project systematically validated functionality, performance, data consistency, and user experience across multiple scenarios and devices. The implementation of slot selection, booking validation,

realtime feedback, and database synchronization was successfully achieved with minimal issues postdebugging. Integration with existing systems such as cart, order, and payment modules was seamless, demonstrating the effectiveness of modular design and API-driven architecture. Testing tools like Selenium and Postman helped ensure system robustness and responsiveness under varying conditions. Overall, the project achieved its key objectives and provided a scalable and user-friendly feature ready for production deployment. This work lays the foundation for further enhancements such as predictive slot allocation, delivery tracking, and more personalized user experiences.

ACKNOWLEDGEMENT

REFERENCES

- Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education.
- Oracle. (2023). *Java SE Documentation*. Retrieved from https://docs.oracle.com/javase/8/docs/
- Spring Framework. (2024). *Spring Boot Documentation*. Retrieved from https://spring.io/projects/spring-boot
- SeleniumHQ. (2023). *Selenium WebDriver Documentation*. Retrieved from https://www.selenium.dev/documentation/