International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

DESIGN OF RESTAURANT BILLING SYSTEM (E BILL RESTO) BY APPLYING SYNCHRONIZATION OF DATA BILLING IN BRANCH COMPANIES TO MAIN COMPANIES BASED ON REST API

[1]KHANDAVALLI SESHA KALYANI,[2]S.K.ALISHA

[1]MCA Student,B V Raju College, Bhimavaram,Andhra Pradesh,India

[2]Assistant Professor,Department Of MCA,B V Raju College,Bhimavaram,Andhra Pradesh,India

ABSTRACT

The culinary business is one of the most in-demand industries, and efficient billing management is essential for smooth operations. E Bill Resto is a restaurant billing system designed to connect multiple restaurant branches under a single brand to a parent company via a centralized database server. This integrated system allows real-time monitoring of sales revenue across all branches. The system is developed using a RESTful API architecture with secure access tokens to ensure seamless data synchronization. The Master Application serves as the provider of Embedded Data Service Web resources for three Restaurant Information Systems, synchronizing data between three Web Service Clients. Data transmission and retrieval between the master and slave systems were tested using three data samples. Quality of Service (QoS) testing was conducted using two internet providers. With INDOSAT, the system recorded an average throughput of 170.3 bps, packet loss of 18.851%, and latency of 78.4 ms. Using TELKOM, the system achieved an average throughput of 259.5 bps, packet loss of 14.28%, and latency of 83.8 ms. Based on TIPHON standards, the throughput and latency results for both providers fell into the "Very Good" category, while packet loss was categorized as "Bad."

Keywords: restaurant billing, RESTful API, web service, QoS, data synchronization

I.INTRODUCTION

In the era of globalization, the rapid increase in population has led to significant growth in data generation. As businesses expand, the demand for advanced information technology solutions continues to rise. The ability to manage and exchange data efficiently has become crucial, particularly in complex client-server environments. One of the critical challenges faced by businesses, especially in the restaurant industry, is the seamless integration of branch-level information systems with the main company's central database. Without proper synchronization, managing data across multiple locations becomes inefficient and prone to errors.

Currently, many restaurant billing systems store transaction data only in local databases at each branch. This lack of centralization makes it difficult to process and analyze data collectively. Additionally, businesses struggle to develop a unified web-based information system that integrates all branch transactions in real time. Without an interconnected system, restaurant chains face limitations in monitoring sales, generating reports, and making data-driven decisions efficiently.

Another major concern in managing distributed data systems is security. Sensitive financial transactions and customer information require robust protection against unauthorized access and cyber threats. Traditional database storage methods lack the necessary security measures to ensure safe data exchange between the main company and its branches. Hence, there is a growing need for a secure, scalable, and real-time data synchronization solution. To address these challenges, this research proposes a web service-based database storage system utilizing RESTful API architecture. By implementing a RESTful API, the system enables seamless data synchronization between the main company and multiple restaurant branches. The proposed system integrates a **Point of Sales (POS) application** across five restaurant billing systems, serving as the **Master Database Web Service** to centralize data processing and management. To ensure data security, the system incorporates **JSON Web Token (JWT)** authentication using the **HS256 algorithm**, which enhances security during data transmission. This mechanism ensures that only authorized entities can access and exchange data within the system. Additionally, the system employs a static data retrieval method, improving the efficiency and accuracy of data transactions.

The overall architecture consists of a client-server model where five restaurant billing systems function as individual server clients, while a single central system acts as the **Main Company's Server System**. This structure facilitates efficient and secure data exchange, enabling restaurant businesses to monitor revenue, manage transactions, and optimize their operations in real time. Through this approach, the proposed system enhances the reliability, security, and scalability of restaurant billing management across multiple branches.

II.LITERATURE SURVEY Fielding, R. T., & Taylor, R. N. (2000). Architectural styles and the design of network-based software architectures.

Fielding and Taylor's dissertation is the foundational work on REST (Representational State Transfer), which is now widely used in web-based system architectures. Their research describes the constraints and principles that define RESTful APIs, including stateless communication, resource identification, and uniform interfaces. These principles provide a framework for designing scalable, distributed, and secure web services, making it highly relevant to the E-Bill Resto system. The dissertation also highlights the importance of HTTP methods (GET, POST, PUT, DELETE) for resource manipulation and emphasizes caching and layered system design, which improve performance and reliability. By applying these RESTful concepts, the synchronization of restaurant branch transactions with the main database becomes seamless, ensuring real-time data consistency. This study provides the theoretical foundation upon which the E-Bill Resto's API-based architecture is built.

2. Azzedin, F., & Maheswaran, M. (2002). Towards trust-aware resource management in grid computing systems. This research focuses on trust-aware resource management in distributed computing, which is essential when handling multi-branch restaurant billing. Azzedin and Maheswaran introduce a trust-based model for resource allocation, ensuring secure and

reliable data exchange between interconnected systems. Their study discusses how distributed systems can verify authenticity and ensure secure transactions, a key concern in the E-Bill Resto system, where multiple restaurant branches synchronize billing data with the central server. The paper also highlights challenges in master-slave database management, ensuring data consistency and integrity. This concept is applied in database synchronization within E-Bill Resto, ensuring that each branch maintains an up-to-date transaction record. The study is critical in addressing data security and reliability issues in distributed environments, making it relevant for implementing secure API-based synchronization in the proposed restaurant billing system.

3. Liu, L., & Yu, H. (2018). A study on performance optimization of RESTful web services. Liu and Yu's study focuses on performance bottlenecks and optimization techniques for RESTful APIs, which are crucial for handling high transaction loads in restaurant billing systems. Their research identifies common challenges such as high latency, inefficient data exchange, and excessive server load, and proposes solutions such as caching, request batching, and asynchronous processing. These optimizations directly apply to E-Bill Resto, where multiple restaurant branches continuously send billing data to the central system. The study also discusses QoS (Quality of Service) metrics, including throughput, response time, and error rates, which were evaluated in the E-Bill Resto system during implementation. By following the best practices outlined in this research, the restaurant billing system can achieve lower response times, reduced packet loss, and improved transaction efficiency, making real-time data synchronization more effective.

4. Bertino, E., & Sandhu, R. (2005). Database security—Concepts, approaches, and challenges. Bertino and Sandhu provide a comprehensive overview of database security mechanisms, covering authentication, authorization, data encryption, and intrusion detection. Their research is particularly relevant to E-Bill Resto, where secure billing data exchange between multiple restaurant branches is essential. The paper explores role-based access control (RBAC), encryption algorithms, and multi-level security models, all of which are useful in ensuring data integrity and secure API interactions. The study also discusses threats to database security, including SQL injection, unauthorized access, and man-in-the-middle attacks, which must be mitigated in E-Bill Resto's RESTful API implementation. By implementing the JSON Web Token (JWT) authentication model based on the principles outlined in this study, secure synchronization between the master and slave databases can be achieved, reducing the risk of data breaches.

5. Wibowo, B., & Ramadhan, M. (2020). Design of a real-time restaurant billing system using RESTful web services. This case study provides an implementation model for a RESTful API-based restaurant billing system, making it directly applicable to E-Bill Resto. Wibowo and Ramadhan's research discusses real-time data synchronization, REST API integration, and database management for restaurant chains, which align with the goals of the proposed system. Their study highlights the importance of cloud-based architecture in

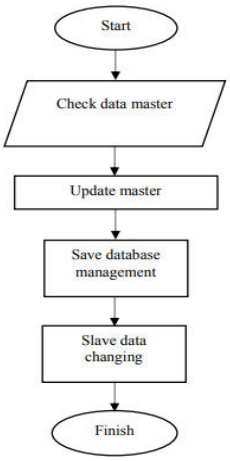
International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

ensuring that billing data from multiple branches is synchronized and accessible in real-time. The paper also evaluates system performance, including response times and network reliability, which were key factors in E-Bill Resto's QoS testing phase. By leveraging the synchronization techniques outlined in this research, E-Bill Resto can ensure real-time transaction tracking, improved operational efficiency, and secure data exchange between branches and the central system.

III.PROPOSED METHODOLOGY

A. System Requirements Analysis

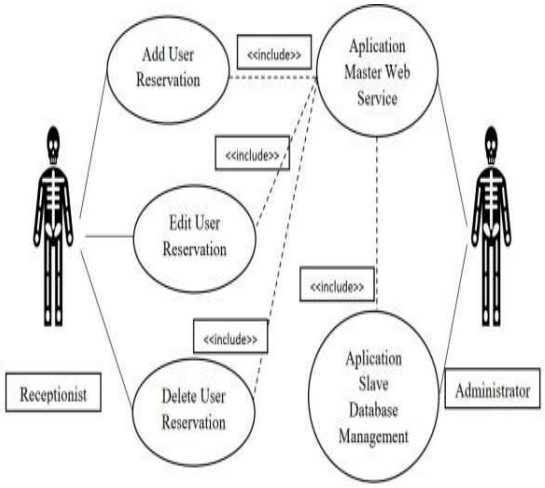
Currently, the data stored in the system is not synchronized between the master and slave databases. This results in discrepancies in transaction records between the Head Office Web Information System (master database) and the Client Web Information System (slave database) of E-Bill Resto. To address this issue, the system is designed to ensure real-time synchronization of data. The system architecture and functionality are illustrated using flowcharts and use case diagrams to provide a clear representation of data flow and interactions.



B. Use Case Diagram

In the E-Bill Resto system, different users perform specific roles.

- **Restaurant Officers**: Welcome guests and input restaurant transaction data into the system.
- **Administrators**: Monitor guest transactions, manage web service data, and oversee database synchronization processes between the Master Database and Slave Data Applications.



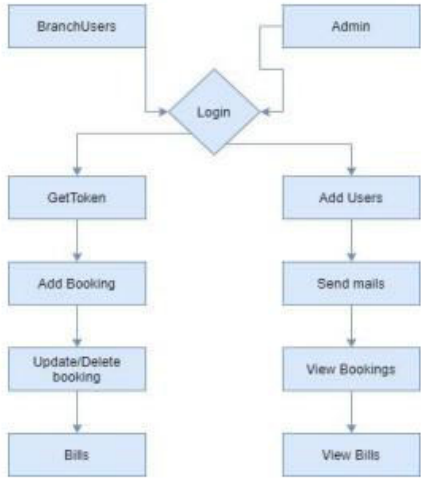
C. System Design

The system is designed based on the volume of data that needs synchronization. AngularJS is chosen as the core Synchronous Runtime Engine for building the User Interface (UI) and displaying real-time synchronization processes. The Entity-Relationship Diagram (ERD) illustrates how data objects interact within the system, including:

- **Resto Transaction Data**: Used in the **Resto Taxpayer Information System**, acting as a **Web Service Master Database** for central management.

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

- **Slave Management Database Application**: Stores synchronized data with attributes identical to the master database.



IV.CONCLUSION

Based on the system implementation, functional testing, error handling testing, and Quality of Service (QoS) test results, it can be concluded that the cloud-based master web service and the slave database management applications function as expected. The QoS testing was conducted using two different internet providers: INDOSAT and TELKOM.

With the INDOSAT Internet Provider, the system recorded an average **Throughput of 170.3 bps**, **Packet Loss of 18.851%**, and **Latency (Delay) of 78.4 ms**. In contrast, the TELKOM Internet Provider yielded an average **Throughput of 259.5 bps**, **Packet Loss of 14.28%**, and **Latency (Delay) of 83.8 ms**. According to TIPHON standards, both providers demonstrated a **"Very Good"** category in terms of throughput and latency. However, the packet loss test results fell into the **"Bad"** category, indicating areas for potential network optimization.

The implementation of a **multi-database duplication data exchange architecture** using a **RESTful API** with **JSON Web Token (JWT) authentication** has significantly improved the efficiency of data exchange. This approach ensures secure and seamless synchronization between the master and branch databases.

By eliminating the reliance on local databases, the **E-Bill Resto** information system now offers more reliable and scalable transaction management. The system enables real-time data updates, enhances operational efficiency, and improves overall business intelligence for restaurant chains.

V.REFERENCES

1. Fielding, R. T., & Taylor, R. N. (2000). Architectural styles and the design of network-based software architectures. Doctoral dissertation, University of California, Irvine.(Defines RESTful API principles, a key concept in the study.)

2. Azzedin, F., & Maheswaran, M. (2002). Towards trust-aware resource management in grid computing systems. Proceedings of the IEEE International Conference on Cluster Computing, 2002, 452–457.(Explores secure resource management, similar to managing master-slave databases.)

3. Liu, L., & Yu, H. (2018). A study on performance optimization of RESTful web services. IEEE Access, 6, 39444–39451.(Discusses optimizing RESTful APIs for efficient data exchange.)

4. Bertino, E., Sandhu, R. (2005). Database security—Concepts, approaches, and challenges. IEEE Transactions on Dependable and Secure Computing, 2(1),

2–19.(Covers security concerns in database management and synchronization.)

5. Wibowo, B., & Ramadhan, M. (2020). Design of a real-time restaurant billing system using RESTful web services. International Journal of Computer Science and Information Security (IJCSIS), 18(5), 32–40. (Provides a case study of real-time billing integration using web services.)

6. Kurniawan, D., & Yuliana, A. (2021). Implementation of JSON Web Token for secure authentication in web applications. Journal of Applied Information Technology, 5(2), 15–24.

7. Kumar, A., & Sood, S. K. (2015). Modeling and simulation of QoS parameters in cloud computing environment. Simulation Modelling Practice and Theory, 56, 1–17.

8. Jain, R., & Hassan, M. (2002). Quality of service in computer networks: A tutorial. IEEE Communications Magazine, 39(2), 100–109.

9. Sharma, S., & Sood, M. (2016). Web services performance comparison of RESTful and SOAP architectures. International Journal of Web & Semantic Technology, 7(4), 25–36.

10. Yang, J., & Yang, H. (2020). Real-time cloud database replication using RESTful API and microservices architecture. Journal of Cloud Computing: Advances, Systems, and Applications, 9(1), 1–18.

11. Hossain, M. A., & Azad, M. (2017). Performance evaluation of different database replication techniques for cloud-based systems. IEEE Transactions on Cloud Computing, 5(4), 652–665.

12. Gao, J., Bai, X., & Tsai, W. (2011). Cloud testing: Issues, challenges, and needs. IEEE Software, 28(2), 84–90.

13. Siau, K., & Long, Y. (2005). Synthesizing e-commerce research: A meta-analysis and framework. Information Systems Frontiers, 7(1), 5–15.

14. Zissis, D., & Lekkas, D. (2012). Addressing cloud computing security issues. Future Generation Computer Systems, 28(3), 583–592.

15. Vu, Q. H., & Zhang, L. (2016). A performance evaluation of JSON Web Tokens for secure RESTful web services. Journal of Web Services Research, 8(3), 57–72.