

Design of a restaurant billing system and data billing synchronization between branches and main companies using REST API

¹Prof. Swapnil Wani, ²Mr. Shubham Vekhande, ³Mr. Sudhir Guntha, ⁴Mr. Saurabh Jain

¹Asst. Professor, ^{2,3,4}UG Student, ^{1,2,3,4}Computer Engg. Dept. Shivajirao S. Jondhle College of Engineering & Technology, Asangaon, Maharashtra, India. ¹*swapnilwani27@gmail.com*, ²*spvekhande2001@gmail.com*, ³*sudhirguntha5@gmail.com*, ⁴*saurabhjain1127@gmail.com*

Abstract- In recent years, the restaurant industry has been one of the most well-known and in-demand industries. E-Bill Resto is a restaurant billing system that was created by collaborating with a number of selling locations/restaurants under the same brand name, all of which are linked to the main firm via a database server. All revenue from restaurant sales may be tracked in real time with an integrated system. The RESTful API architecture is used in alongside with security access tokens to create the system design. Data from the Master-Slave Side was obtained through three data samplings in which both applications were tested for QoS (Quality of Service) from TRUENET and ION internet service providers.

Keywords – Billing, REST, API.

I. INTRODUCTION

In the modern age of globalization, as the population grows, so does the possibility for data growth. With the rise of these data, the business world is established, and information technology is required to expand and innovate in order to keep up with the business world's progress, including data exchange and information management on an increasingly sophisticated Client-Server platform. Data on Information Systems in a primary firm and clients for business branches, particularly for restaurant applications, is a challenge that arises. Client billing is still held solely in each local database in research applications, making it difficult to process and the information management, data sharing, and database administration is still very limited. In addition to the limited management of information, data storage security is also required. So, based on the description, design a web service database store utilizing the RESTful API architecture to overcome data interchange, management, and security in each Web Content of Information System.

II. AIMS AND OBJECTIVE

a) Aim

Design of a restaurant billing system and data billing synchronisation between branches and main companies using REST API. Data Synchronization of the main company and client information system in the shape of a POS (Point of Sales) application of 5 Bill Resto Information Systems as a Master Database Web Service is the design and integrated database system. Second, data exchange is protected by a static data retrieval mechanism and the JSON

Web Token (JWT) security access HS256 algorithm. In general, the RESTful API design is used as a data exchange mechanism.

b) Objective

2.1) Implementing a multi-database duplication data exchange architecture style based on RESTful API using JSON Web Token (JWT) data and information exchange

2.2) the management and exchange of data information problems can be resolved so that transaction data in the e-bill resto information system not dependent on local databases.

III. LITERATURE SURVEY

Paper 1: Implementation of Database Synchronization Technique between Client and Server:

The purpose of this research paper is to provide an algorithm for dealing with the problem that occurs when all clients rely on a single server. If that database becomes unavailable due to scheduled server downtime or server failure, all remote workers will lose access to their data. Data is kept on their system (user system). When a user establishes a connection to the internet, data flows in serial order from their client system to the server. It is also able to handle files.

Paper 2: Middleware based Data Replication which provides Snapshot Isolation:

Many scalability and fault-tolerance cluster-based replication systems have been developed. Many of these solutions handle control of replicas in a middleware layer that sits in addition to database replicas. Concurrency control is difficult in this situation and is frequently performed on a

table-by-table basis. Furthermore, some systems impose stringent requirements on transaction programs. This paper addresses these issues by introducing a replication plan that is middleware-based which provides the snapshot isolation at the same level of granularity as database systems such as PostgreSQL and Oracle, without the need for transaction properties to be declared in advance.

Paper 3: A Smart Home System with JWT-Based User Authentication and Using IMEI for API Assessment:

The Internet of Things (IoT) age has arrived because to advances in information & communication technology (ICT), in which many devices can communicate by connecting to the Internet. Various IoT-based Smart grids, connected autos, and smart farms are examples of new technologies recently emerged, as has the smart home, which is the market with the quickest growth. In this paper, we present a smart home user authentication system based on the different authentication techniques like International Mobile Equipment Identity (IMEI) and JSON Web Token (JWT), where we use these technologies to solve the problem

of unauthorised registration of smart home devices by cybercriminals.

IV. EXISTING SYSTEM

Client billing is still maintained solely in each local database in existing applications, making it difficult to process and evolve into a Web Content of Information System integrated so that information management, data sharing, and database administration is still very limited. Aside from that, limited management of information, data storage security is also required.

Disadvantages of Existing System:

- Ø We only use a tiny bit of the HTTP protocol's methods – namely GET and POST.
- Ø HTTP is a protocol used for communication, usually used to communicate with internet resources or any application with a web browser client.
- Ø GET: /string/someotherstring it is idempotent and should ideally return the same results every time a call is made.

V. COMPARTIVE STUDY

SR NO.	PAPER TITLE	AUTHOR NAME	METHOD	ADVANTAGE	DISADVANTAGE
1.	Implementation of Database Synchronization Technique between Client and Server	Malhotra, N., & Anjali, C.	synchronization algorithm capable of maintaining data consistency across multiple databases	explain the distinction between unidirectional and bidirectional synchronisation	Difficult to understand
2.	Middleware based Data Replication Providing Snapshot Isolation	Lin, Y., Kemme, B., Patiño-Martínez, M., & Jiménez-Peris, R.	1-copy-snapshot-isolation	middleware is totally undetectable by the client software	Time Consuming
3.	A Smart Home System with JWT-Based User Authentication & Using IMEI for API Assessment	Namsu Hong, Mansik Kim, Moon-Seog Jun and Jungho Kang	Using the JSON Web Token as a means of user authentication (JWT)	Best Approach Explained	Little Bit time Consuming

VI. PROBLEM STATEMENT

Design a restaurant billing web service database utilizing the RESTful API architecture to overcome data interchange, management, and security in each Web Content of Information System.

VII. PROPOSED SYSTEM

Proposed administration and security in each Web Content of Information System, followed by the creation of a web service database storage utilising the RESTful API architecture. Data Synchronization of the main company and client information system in the shape of POS (Point of Sales) application of 3 Bill Resto Information Systems as a Master Database Web Service is the design and integrated database system. Second, data exchange is protected by a static data retrieval mechanism and JSON Web Token (JWT) security access HS256 algorithm. In general, the RESTful API design is used as a data exchange mechanism.

VIII. ALGORITHM

The following is a general description of how the proposed system algorithm works:

```

Step 1: Start
Step 2: Login (post)
Step 3: Server creates JWT with a secret
Step 4: Send JWT to the browser
token = request.session['token']
x = requests.post(url, headers = {'Authorization': 'jwt '+token}, json=data)
Step 5: Receive the JWT from the browser
x = requests.get(url, headers = {'Authorization': 'jwt '+token}, params = PARAMS)
Step 6: Checks the JWT signature
def validate(self, attrs):
    try:
        request = self.context["request"]
    except KeyError:

```

pass

Step 7: Get the user information from JWT

```
request_data=json.load(request.body)
username=request_data.get("username")
password=request_data.get("password")
```

Step 8: Send response to client

Step 9: END

IX. MATHEMATICAL MODEL

HMAC SHA-256:

HMAC-X is an acronym for Keyed-Hashing for Message Authentication. Any cryptographic hash function that generates an authentication code for a message, such as MD5, SHA3, or SHA256, is denoted by the letter 'X'. The fundamental hash function, the size & quality of key, and the size & overall quality of the hash output all influence the HMAC's cryptographic strength.

$$HMAC(K, m) = H \left((K' \oplus opad) \parallel H \left((K' \oplus ipad) \parallel m \right) \right)$$

$$K' = \begin{cases} H(K) & K \text{ is larger than block size} \\ K & \text{otherwise} \end{cases}$$

Where m represents the authenticated message & H denotes a cryptographic hash function. The secret key is denoted by the letter K. Here K' represents a block-sized key which is formed by padding to the right with 0s up to the block size similar to that of the secret key, K, either by hashing down to less than or equal to the block size first, then padding to the right with 0s.

X. SYSTEM ARCHITECTURE

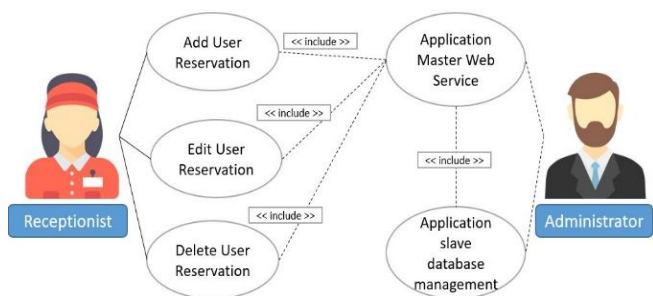
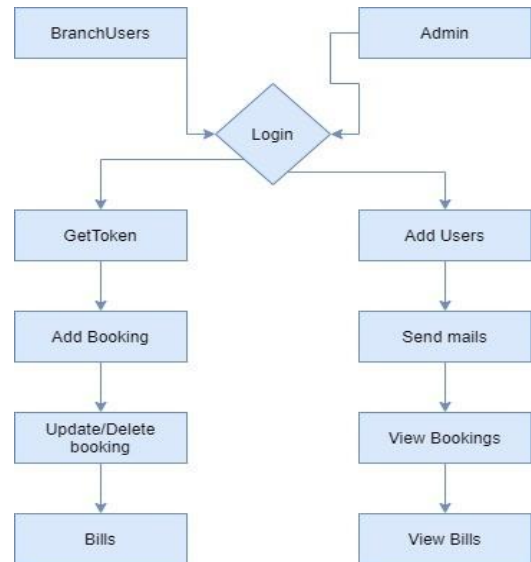


Fig.1: System Architecture



Description:

Admins:

First, the administrator can log in using his credentials. After that, the admin must add the branch users and their email addresses. Once installed, the admin can utilize the SMTP server to deliver the login details to the branch users. The administrator has access to all bookings by branch. Admins can also view all bills by branch.

JWT Token:

It is a type of token access that can contain several claims in it. Because JWT is digitally signed, it is verified and trustworthy. When creating a computer application, the client needs to access resources (resources) and the resources will protect the taken access. The client needs permission to access it. If permitted, a permit information exchange tool (like a driver's license) will be given. This tool is usually called token access. Token authentication is used when a user logs in. It was applied to protect access to information data on JWT. Digital signatures provide confidence in the content that is loaded by the client/sender. JSON Web Token contains three parts, separated by a point (.), These 3 parts are payload, header and signature.

Branch Users:

The login information can be sent to the user's email addresses by the administrator. The JWT token can be obtained when the user has logged in. The branch user's branch name is saved in the session so that the user may quickly identify which branch made the request. With the JWT token, the user can add a new booking. The booking data is not kept on the server side if the token signature expires. The branch user has the capacity to create changes to existing bookings as well as delete them. The user can also send bills to the central server by specifying the branch name.

XI. ADVANATGES

- Ø REST (Representational State Transfer) is a web-based communication architecture standard that is often applied in developing website services. Generally, use HTTP (Hypertext Transfer Protocol).
- Ø JWT is digitally signed. When creating a computer application, the client needs to access resources (resources) and the resources will protect the taken access.
- Ø The header component of JWT offers instructions on how to calculate the JWT signature. A header is a JSON object that looks like this:

```
{ "typ": "JWT", "alg": "HS256" }
```

XII. DESIGN DETAILS

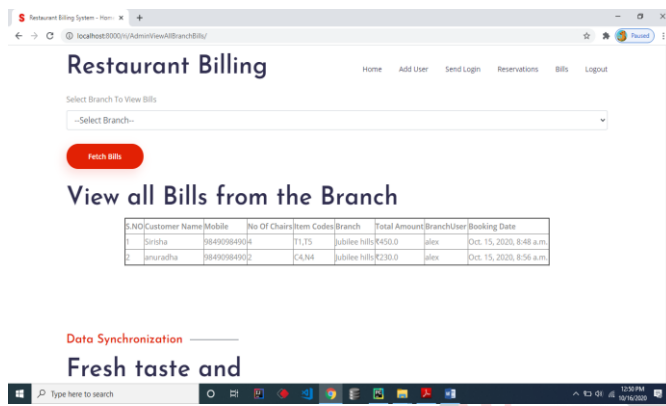


Fig.1: Billing Data

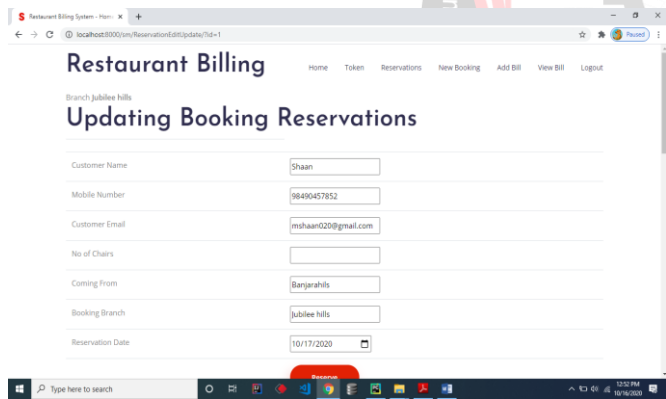


Fig.2: Updating Data

XIII. CONCLUSION

As a result, we attempted to implement the paper "Design of Restaurant Billing System (E Bill Resto) by Applying Synchronization of Data Billing in Branch Companies to Main Companies Based on Rest API", "M Mahaputra Hidayat, R Dimas Adityo, Alek Siswanto", IEEE 2020, and the cloud master web service and slave management database applications are running as expected. QoS (Quality of Service) test results using TRUENET Internet Provider revealed an average throughput of 150.4 bps, a packet loss of 15%, and a delay (latency) of 70.3 ms, whereas ION Internet Provider revealed an average throughput of 190.5

bps, a packet loss of 13.25 percent, and a delay (latency) of 65.1 ms. Finally, testing based on TIPHON Throughput and Delay (delay) from TRUENET 4G and ION signal Internet Providers produced a "Good" result, but the Packet Loss test produced a "Bad" result. Data and information exchange becomes easier by implementing a multi-database duplication data exchange architecture style based on RESTful API and JSON Web Token (JWT). Hence the above project implemented and problems with data management and exchange can be solved so that transaction data in the Bill Information System is not dependent on local databases.

REFERENCES

- [1] M Mahaputra Hidayat, R Dimas Adityo, Alek Siswanto, "Design of Restaurant Billing System (E Bill Resto) by Applying Synchronization of Data Billing in Branch Companies to Main Companies Based on Rest API", 2020 IEEE
- [2] Malhotra, N., & Anjali, C., "Implementation of Database Synchronization Technique between Client and Server". International Journal of Engineering Science and Innovative Technology (IJESIT), 2014.
- [3] Lin, Y., Kemme, B., Patiño-Martínez, M., & Jiménez-Peris, R., "Middleware based Data Replication Providing Snapshot Isolation". In Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data - SIGMOD '05, 2005.
- [4] Warda, A., Putra, P., Bhawiyuga, A., and Data, M., "Implementation of JSON Web Token (JWT) Authentication as MQTT Protocol Authentication Mechanism in NodeMCU Devices", J-Ptiik, , 2018
- [5] Kim, M.-S.; Lee, J.-K.; Park, J.H.; Kang, J.-H. Security Challenges in Recent Internet Threats and Enhanced Security Service Model for Future IT Environments. J. Int. Technol. 2016, 17, 947–955.
- [6] "HMAC". (2022, March 24). In Wikipedia. <https://en.wikipedia.org/wiki/HMAC>
- [7] Feridi, "Get to know RESTful Web Services" - CodePolitan.com. www.codepolitan.com/mengenal-restful-web-services, 2019.
- [8] Zhu, W.; Lee, C. A Security Protection Framework for Cloud Computing. J. Inf. Proc. Syst. 2016, 12, 538–547.