Cloud Touch: Web Service Based Flight Booking App

Abstract

This web service application project was designed and developed to serve flight booking. The purpose is to design an interactive and seamless user experience while flight booking. The application is developed using JSP. MySQL workbench to provide data storage facility. In addition to these multiple services are integrated to empower the application like user log, and signup for user authentication. Google Maps services to check the distance between source and destination. Currency exchange API to concert the currency rate from one currency to another. The data used in the application are in JSON format, which facilitates easy manipulation of the data. This application addresses most of the features mentioned in the guidelines of the assignment.

Keywords: MySQL, API, Flight Booking, Web Services

1. Service Oriented Architecture

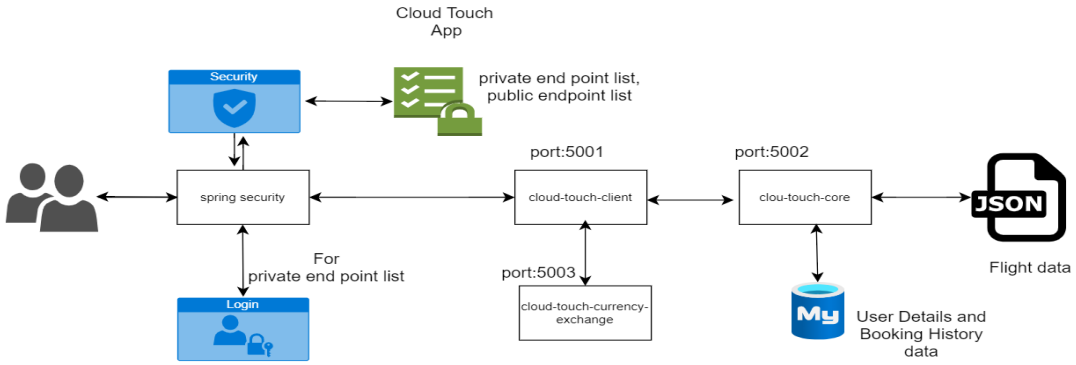

Figure 1. Service Oriented Architecture for the application.

The architecture encompasses core services, client services, and additional functionalities such as currency services. Data storage and retrieval are facilitated by MySQL Workbench. Core services communicate with subsidiary services through a specified port, as depicted in the provided diagram. The API handles responses generated by the core services.

In the context of scaling the application, JMeter assumes a crucial role. JMeter is employed to simulate performance testing of load balancing within the core services of the web application. This involves detailing the number of concurrent clients attempting to access the application simultaneously.
The application's user interface is constructed using JSP, a server-side technology that seamlessly integrates Java code into HTML pages.

2. Development Stack:

1. Programming Language: Java, with JSP for the front end, and utilizing the Spring Boot framework for backend development.
2. Development Platform: Utilizing Visual Studio and MySQL Workbench for software development and database management, respectively.
3. Testing Tool: Employing Apache JMeter for comprehensive testing of system performance and scalability.

4. Google Map API Integration: Integrating the Google Maps API seamlessly into the application to enhance location-based functionalities and services.

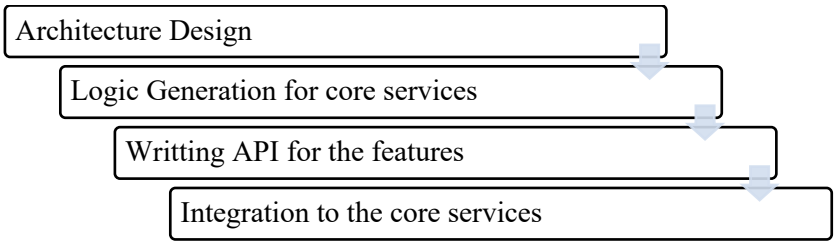

Figure 2. Workflow diagram for the application

3. Application Design & Features:

The application development steps involved below mentioned steps:
Developing a SOA (Service Oriented Architecture) based architecture, Logic generation for the core services, writing API for the feature, and finally integration as shown in the above workflow diagram.

Functional Description:

1. **RESTful API:** Implemented a RESTful web service for interaction with clients, allowing easy integration and interoperability. It consists of 3 web services. The main client web service, core service, and currency conversion service.

2. **Flight Reservation**: Allows users to search for available flights based on various criteria such as origin/destination, and date.

3. **Booking:** Enables users to book selected flights, updating the number of available seats accordingly.

4. **Integration with Currency Conversion Service (Auto-Conversion):** Automatically converts ticket prices to the preferred currency specified by the customer during flight booking.

5. **Performance Analysis:** Conducted analysis of performance and scalability challenges in SOA web services.

6. **Data Storage:** Utilized SQL Workbench database to store user details, such as username, e-mail address, flight booking details, etc.

7. **Modular Architecture:** Designed the service in a modular manner to promote scalability and maintainability. The flight details are stored in a JSON file. It is very scalable and easy to update flights.

8. **Validation:** Implemented validation mechanisms to ensure data integrity and handle invalid input effectively.

Design Decisions:

1. **External Service Integration:** Integrated an external currency conversion service to handle currency conversion seamlessly.

2. **Error Handling**: Implemented robust error handling mechanisms to address potential issues during currency conversion.

3. **Asynchronous Processing:** Utilized asynchronous processing to minimize service latency and enhance user experience.

Section C

4. Quality of Service Challenges:

Our online service for booking flights has problems with Quality of Service (QoS), particularly in terms of implementation and scalability. Scalability is essential as our company becomes more well-known so that our systems can effectively handle rising traffic volumes without compromising effectiveness. Increasing resource efficiency, reducing response times, and ensuring service availability during peak demand are some potential implementation process roadblocks. These challenges demonstrate how important it is to plan carefully and distribute resources wisely in order to maintain service quality as demand increases.

Ensuring scalability involves both strategic planning and technical considerations in order to support future expansion. Issues with scalability might arise from a lack of physical resources, poorly designed software, or network infrastructure bottlenecks. These difficulties can be mitigated by putting in place efficient load balancing techniques, horizontal scalability plans, and performance optimisation techniques. Proactive monitoring and capacity planning are also necessary to foresee and address scalability problems before they have an adverse effect on service quality.

5. QoS Testing and Performance Analysis:

We tested our implementation thoroughly for quality of service (QoS) in order to assess its scalability and performance. We simulated situations using technologies such as JMeter to evaluate the impact on performance and scalability. Performance profiling, load testing, and stress testing gave us important insights into how our system behaved in various scenarios. Through the identification of possible obstacles and the examination of system response under varying loads, we were able to fully comprehend the restrictions imposed by scalability and pinpoint areas that needed improvement.

To maximise system performance and guarantee constant service quality, performance analysis is essential. Through the examination of critical performance indicators like error rates, throughput, and response times, we can pinpoint areas of low performance and adjust system elements accordingly. Regular performance testing is necessary to assess system performance and continuously find opportunities for optimisation, both during development and after deployment.

6. Utilizing Cloud Computing for Scalability:

To address our scaling issues, we can make use of the scalability and flexibility of cloud computing. We can utilise auto-scaling capabilities from leading cloud providers such as AWS, Azure, and Google Cloud by moving to the cloud. By dynamically adjusting resources in response to measurements such as CPU utilisation and network traffic, these capabilities guarantee optimal performance while avoiding over-provisioning. But moving to the cloud comes with its own set of difficulties, such deciding which deployment type (public, private, or hybrid) to use and how to integrate new systems seamlessly with the old ones while managing security issues and possible vendor lock-in.

Pay-as-you-go pricing structures, worldwide availability, and on-demand resource provisioning are just a few of the many scalability benefits that come with cloud computing. Organisations may quickly scale their infrastructure to meet demand spikes by utilising cloud services, which eliminates the requirement for an initial hardware investment. Additionally, a variety of managed services and tools for monitoring, automation, and optimisation are provided by cloud providers, allowing businesses to simplify their processes and concentrate on providing value to their clients.

7. Challenges Associated with Cloud Migration:

Although moving to the cloud has advantages for scalability, there are drawbacks as well. Whether going public, private, hybrid, or multi-cloud, selecting the best deployment plan necessitates carefully weighing aspects including cost-effectiveness, security, compliance with regulations, and data protection. To reduce interruptions and preserve post-migration performance, careful planning is also necessary when transferring apps and data to the cloud. Organisations may face challenges during

migration, including vendor lock-in, security issues, data migration, and integration with current systems. For the cloud infrastructure shift to go smoothly, there needs to be a clear migration strategy in place, as well as comprehensive risk assessment and mitigation measures.

To summarise, managing the difficulties associated with cloud migration, conducting thorough testing, and strategically utilising cloud computing for scalability are all necessary to address Quality of application concerns in our travel booking web application. By overcoming these obstacles, we make sure that our service continues to provide a dependable and seamless user experience even as demand increases and technology advances.

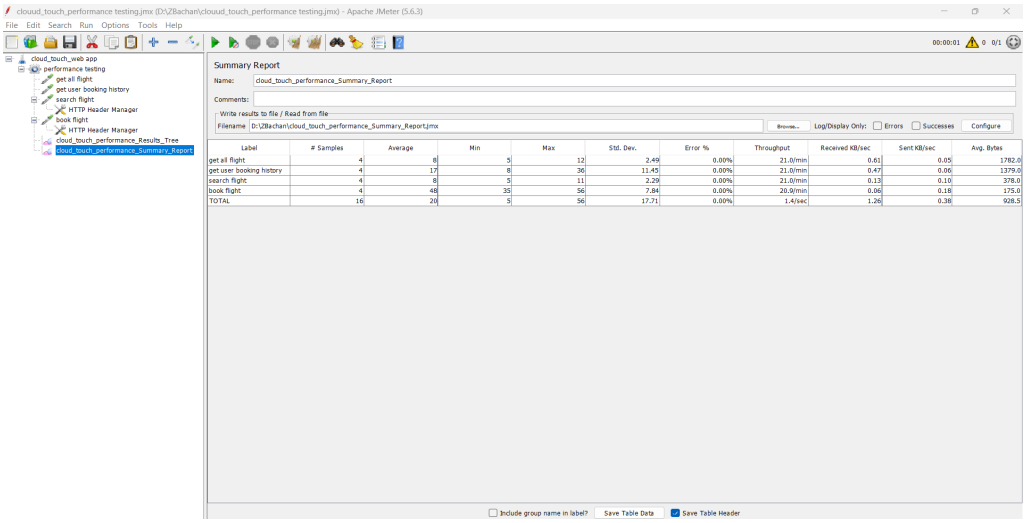

Figure 3. JMeter test report for each API

Section D

8. Application of Semantic Web and Linked Data Technologies

The Semantic Web and Linked Open Data (LOD) are critical components of intelligent information processing because they provide formal frameworks for organising and utilising massive amounts of interconnected data. Using semantically defined Linked Open Data can significantly improve existing services, especially in areas such as travel, where tailored advice and personalised experiences are highly looking for.

To demonstrate this possibility, let's look at how using semantically defined Linked Open Data may turn a basic travel service into a themed vacation package. Consider that we have created a travel software that assists users in organising trips according to their tastes and financial constraints. This programme can advance beyond conventional travel planning by incorporating LOD and suggesting themed vacations, such adventure travel, music festivals, food tours, and more.

A thorough examination of how public datasets from the Linked Open Data Cloud are used presents several chances to improve and expand the functionality of the trip application. A huge network of linked datasets involving a variety of topics, including geography, culture, entertainment, and demography, is included in the Linked Open Data Cloud. The programme can provide richer, more individualised holiday recommendations based on the unique characteristics of each user by utilising this abundance of data.

To provide themed holiday recommendations using Linked Open Data, one method is to mine relevant data sets to get insights into different topics. For example, it is possible to analyse datasets that include data about historical sites, outdoor activities, cultural events, and tourist attractions to find trends related to various holiday types. These dataset's content may be interpreted and categorised with the use of machine learning and natural language processing algorithms, which makes it easier to find themed vacation spots.

Including information from review sites, social media platforms, and travel blogs can also yield useful user-generated content that enhances vacation suggestions. The popularity and calibre of themed vacation spots and activities can be evaluated with the aid of sentiment analysis, user reviews, and rating analysis. This allows the application to suggest experiences that correspond with the interests and preferences of its users.

Additionally, the application can provide location-based recommendations based on user preferences and travel history by utilising geographic data from the Linked Open Data Cloud. Through the programme, users can propose themed holidays that align with their prior preferences and introduce them to novel and exciting experiences by fusing geographical data on popular areas with their past travel experiences.

Locating and gaining access to datasets related to travel, tourism, entertainment, and cultural heritage is necessary to find appropriate public data from the Linked Open Data Cloud. Data integration and discovery are made easier by platforms like the LOD Cloud Cache and LOD Laundromat, which offer easy access to a variety of Linked Open Data sources. The availability and calibre of specific datasets are further enhanced by cooperative activities within the Linked Open Data community, such as data curation programmes and ontology development projects.

To guarantee consistency and relevance, pre-treatment procedures, transformation, and integration of data are required when integrating public datasets into the created travel application. Interoperability and semantic enrichment are made possible by semantic technologies like RDF, OWL, and SPARQL, which also enable the representation, querying, and integration of Linked Open Data. Developers can easily integrate other datasets into the application ecosystem by creating mappings between the Linked Open Data vocabulary and the application's data model.

Finally, using semantically defined Linked Open Data presents an abundance of opportunities to improve travel services and generate tailored recommendations for themed vacations. Travel applications can deliver more personalised and engaging experiences by utilising the extensive network of interconnected datasets included in the Linked Open Data Cloud. Developers may unleash the transformational power of Linked Open Data and usher in a new era of intelligent information processing in the travel sector by effectively mining, analysing, and integrating data.

9. References:

Belleau, F., Nolin, M., Tourigny, N., Rigault, P., Morissette, J. (2008). Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. Journal of Biomedical Informatics, 41(5):706-16.

Berners-Lee, T. (1997). Cleaning up the User Interface, Section - The ”Oh, yeah?”-Button.

Berrueta, D., Phipps, J. (2008). Best Practice Recipes for Publishing RDF Vocabularies - W3C Working Group Note.

Haslhofer, B., Schandl, B. (2008). The OAI2LOD Server: Exposing OAI-PMH Metadata as Linked Data.

Heath, T. (2008a). Information-seeking on the Web with Trusted Social Networks – from Theory to Systems. PhD Thesis, The Open University.

Kobilarov, G., et al. (2009). Media Meets Semantic Web - How the BBC Uses DBpedia and Linked Data to Make Connections. Proceedings of the 6th European Semantic Web Conference (ESWC2009).