

Hotel Reservation System

Akash Sadanand Hande
School of Computing
National College of Ireland
X17156220@student.ncirl.ie

Abstract—This report describes a cloud application service solution for hotel reservations i.e. Hotel Reservation System (HRS), developed in the J2EE framework and delivered as Software as a Service (SaaS). The application follows MVC architecture with Java classes as Model, JSP as view and Servlet as Controller as well as MySQL at the database. The application is deployed on Microsoft Azure which is a very famous cloud computing service for deploying the application. The application has responsive web pages, built in bootstrap and HTML which can adjust dynamically, considering the characteristics of the device used for accessing the application. HRS helps it's end users to search hotels in the specific city, features for basic viewing and reserve it accordingly. This application is composed of the client application which consumes multiple APIs namely Google Sign-In API, Geocoding API, Maps JavaScript API, Travel Reminder and services application which provides RESTful services for booking and hotel searches.

Keywords—Hotel Reservation System, Client Application, Service Application, REST, Web Service

I. INTRODUCTION

The Hotel Reservation System (HRS) is a cloud-based software system which helps customers across the globe to reserve hotels in some of the cities in Ireland. Key features include central search and reservations, online booking engine, basic viewing and retail point of sale. Apart from that HRS provides services to peer application i.e. Travel Reminder and Travel Planner for booking and registered hotels in the HRS. Deployed application links on Azure is mentioned below:

Client Application	http://hotelnci.azurewebsites.net/index.jsp
Service 1	http://hotelnci.azurewebsites.net/rest/UserService/users/cork
Service 2	http://hotelnci.azurewebsites.net/rest/UserService/booking

Table 1 URL of application

Rest of the paper is organized in the following way. Application Goals with different features of application have been summarized in Section II. Section III depicts the literature survey on “object storage and messaging infrastructures in the cloud”. The architecture of the application has been described in section IV. Section V gives a detailed description of the implementation of the application. Testing methodologies have been described in section VI. Finally, section VII concludes this paper.

II. APPLICATION GOALS

The developed application allows end users to search hotels from different cities of Ireland, have a basic viewing of hotels, check the location of the hotel on google map and reserve it accordingly. HRS being a cloud-based solution allows the end user to access it with any device capable of running any one of the web browsers. HRS is developed to be compatible for all browsers and responsive in nature so that it can be accessed from any size of the device and automatically scale it to fit the screen. When user wants to reserve room in hotel from some of the cities of Ireland (Dublin, Galway, Cork, and Limerick), he can visit HRS, search for hotels in the city, can have basic viewing of the hotel such as facilities provided by hotel, rate of the room, address, rating etc., see location of hotel on google map and if he agreed to book the room he can proceed to book the room. HRS is formed from client applications and service application as described below.

A. Client Application

The HRS client application is majorly composed of three API's provided by Google i.e. Google Sign-In, Geocoding, and Maps. Functional workflow of the HRS application is illustrated in the below flowchart:

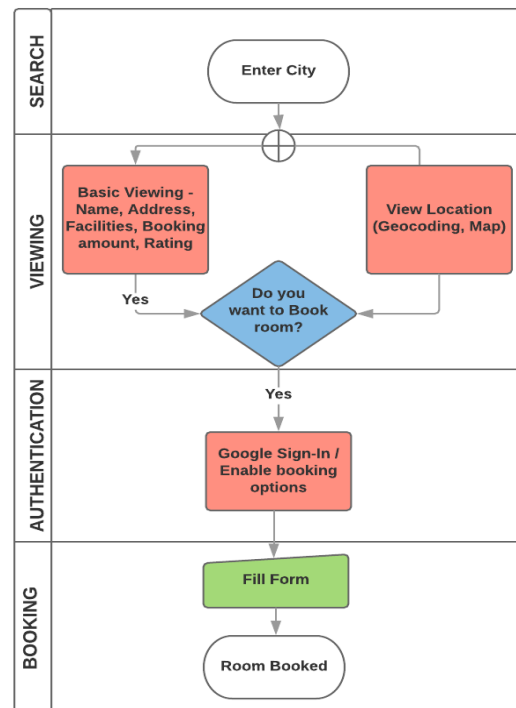


Figure 1 Flow chart of daily operations [1]

The client application has four modules namely search, viewing, authentication and booking as shown in the flowchart mentioned above. In the search, module application provides a list of the hotels in response for city entered by the end user which are registered with the HRS system. Viewing module provides basic viewing such as name, address, facilities, booking amount and rating etc. The second part of the viewing module is composed of Google Geocoding and Maps which are used to show the location of the hotel on the map so that user gets an idea about the geographical location of the hotel. If the user is satisfied with basic viewing, hotel location and he want to book a room he can log in to HRS by using his google login credentials and can enable booking options, as they are not enabled before login and user is not able to proceed for booking. Booking module allows the user to fill form asking for basic details such as name, address, contact details etc. after submitting form room will be booked in that hotel and end user gets confirmation on the screen.

The client application also consumes one-second party API known as Travel Reminder, provided by colleague. Travel Reminder provides the functionality to set a reminder on Google calendar so that end user can get reminder while leaving the place for boarding the mode of travel to reach the hotel.

B. Service Application

The service application part of the HRS provides two services namely booking and search hotels. Booking service provides the list of bookings made in the HRS and searches service takes the city as a parameter and provides the list of hotels present in that specific city. Both the services data are retrieved from the database and transmitted back to the client in the form of XML. For utilization of services, the request must pass the authentication process of HRS which follows “Basic Auth” for REST APIs. Here is the basic specification of HRS service application:

<i>Specification</i>	<i>Follows</i>
Language	Java
Architectural Style	REST
Authentication	Basic Auth
Authentication Model	Username and password
Terms of Service	URL
Supported response format	XML

Table 2 Specification of Service application

III. BACKGROUND RESEARCH

In today’s digital era, where web technology is becoming an essential part of the business in various sectors such as banking, finance, education, and communication from small scale to large scale enterprises, applications are migrating to cloud for scalability, flexibility, and simplicity. Recent advances in web application development have allowed software developers to improve the quality of software to collaborate other software both within the organization and among organizations for achieving mutual benefits. These

software services are developed in different forms such as SOAP services, message-based services, and most recent REST services. Several software services providing publicly available web application programming interface has been grown rapidly [2] leads to several issues and challenges for the researchers. Research area includes secured data exchange, retrieve accurate, real-time, detailed data automatically and importantly infrastructures in the cloud for object storage.

A. Importance of messaging infrastructure

Virtual Enterprise (VE) is a logical, temporary and dynamic network of autonomous organizations that cooperatively form unique business opportunities [3]. To form VE information and message exchange (IME) process is introduced for the exchange of messages between two different software applications which may be part of the different organization. In the VE environment most of the department such as marketing, sales, human resource etc. are distributed and connected via networks but due to autonomous and diverse nature of software applications, there are various barriers of message exchange in two systems.

To design independent infrastructure supporting various business environment, the priorities include data exchange, information delivery, object storage, sharing between different services and location of services. While developing a product with the collaborative design process, supports in the gathering and organizing of design information with overall design concept and purpose is necessary. Previous research on messaging infrastructure demonstrated the data exchange by using Object Oriented Methodology (OOM), Simple Object Access Protocol (SOAP), and Representational State Transfer (REST) which uses Web Service Description Language (WSDL). Among all of them, REST is the most famous and widely used architectural style for defining web service and message passing infrastructures.

B. Web Service and Architectural Styles

A web service allows the use of software application via a network where message passing or data exchange is done by URL queries and Input/output is processed via HTTP protocol. In the web services data gets exchanged in the form of XML so this protocol is known as Simple Object Access Protocol (SOAP) where Web Service Description Language (WSDL) is used to define the structure of the object which is used for communication between two parties.

RESTful web service is the type of web service which provides four basic features namely scalability, Declarative nature, lightweight objects and importantly provides HTTP methods (GET, PUT, POST, DELETE etc.) for management of resources on the server. RESTful services involve several important aspects such as orchestration and choreography as part of its composition view [4]. The lifecycle of web service composition includes service deployment (deployment, service registry) and service composition (request, definition service selection, constructed composite service, execution and sending response data) [5]. In [4] the authors proposed a process for composing SLAs associated with a workflow and lifecycle of web services which importantly includes composition, evaluation, and optimization. Research in [6] presents benefits along with examples of Amazon SQS

programming interface developed in ActionScript. They propose an interface to enable Adobe Flex mobile developers to facilitates integration efforts within tablets and smartphones.

C. Object Storage

In the massive communication era of software development, web-based solution providers must work with huge amount of data produced from various resources such as gaming, social media, high-quality video etc. and this data is mostly unstructured. To work with this unstructured data raising trend in the computing community is object storage which provides APIs allowing direct access to objects via URL, unlike HTTP protocols. Most widely used and promising framework of object storage systems is swift; released by OpenStack and aiming at designing of IaaS architectures. Instead of considering as a traditional file system, Swift must be considered as a highly scalable system to store, recover and delete objects along with related metadata with the help of HTTP API [7].

Research in [7] aims towards characterizing Swift-based object storage system from an availability point of view and finding out the configuration of infrastructure which guarantees “five nine” SLA. Authors in [8] have analyzed the reliability of geo-replicated cloud storage system which used backbone of OpenStack swift characterized by different bandwidth resources. They are analyzing reliability which is carried out by considering recovery scheme designed from scratch.

D. Message Queues in cloud applications

Amazon Simple Queue Service (Amazon SQS) is very famous message-oriented middleware in the cloud provided by Amazon which ensures at least once delivery, best effort ordering with standard queues and high throughput, exactly once processing in FIFO queue. ActiveMQ is another powerful open source messaging and Integration Pattern servers provided by Apache foundation with support to JMS and J2EE. RabbitMQ is lightweight, easy to deploy, cloud-based, open source message broker which runs on many operating systems and cloud environments. Azure also provides two types of queue mechanisms i.e. storage queues which are part of Azure storage infrastructure featuring REST-based interface and service bus queues which are part of Azure messaging infrastructure that supports queuing as well as publish and advanced integration patterns.

Software development kits are provided from leading cloud providers which provides basic functionalities in creating service applications such as authentication, request retries and error handling etc. Amazon, Microsoft, CloudMQ, and Apache provide development kits and libraries for multiple programming languages such as Java, .NET, PHP and Ruby. For proper use of these libraries, the developer needs to have basic technical knowledge about the programming environment in which he is developing to reduce the time for development.

IV. SOLUTION ARCHITECTURE

Software solution architecture is the process of defining the high level of the structured skeleton of application which is used to fill the gap between business requirement and technical specification while improving manageability,

security, and performance of the application [9]. Specifically, the objective of solution architecture is to deeply understand the high-level business requirement, the behavior of the application before starting of development of the application, interdependency between various modules of the application and development of independent modules. The architectural design of application highly affects development, deployment, and maintenance of the software solution. Below is the detailed description of HRS solution architecture with framework adoption:

A. Client Application

A software framework is a baseline for developing software applications which provide a foundation platform to software developers to construct their application with specific programming language [9]. In today’s technical era, developers can choose any framework from a large variety of choices to build their software application with unique features of everyone. HRS client application is adopting J2EE (JSP/Servlet) framework for the development because of its decent features. Hotel Reservation System (HRS) is a cloud-based application which is followed MVC architecture by which HRS gets divided in a model built with Java classes, view composed of JSP and Servlet as Controller as well as MySQL at the database layer. As client application is composed of three different APIs from google namely sign in, geocoding and maps below sequence diagram illustrate the sequence diagram of the HRS application:

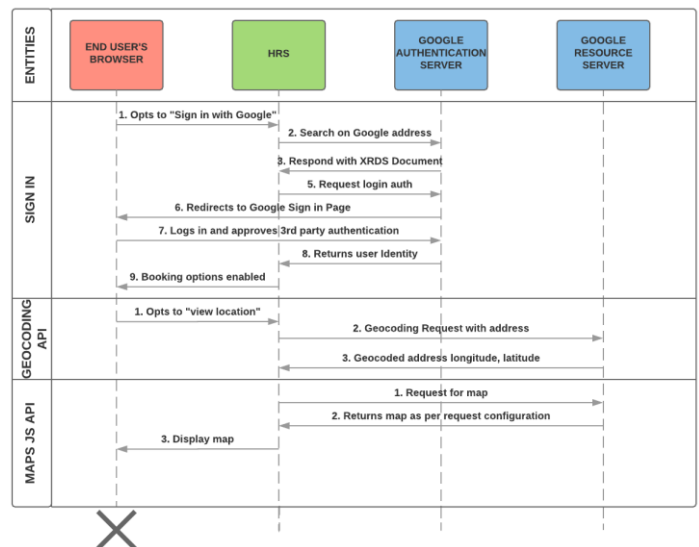


Figure 2 Sequence diagram of client application [1]

B. Service Application

To develop service application of Hotel Reservation System which is a RESTful web service java provides standard and portable JAX-RS API. HRS adopts Jersey RESTful web service framework which is an open source framework for developing RESTful web service in Java and provides support for JAX-RS API [10]. HRS provides two major services namely booking and search which follows GET HTTP methods and provides output in XML format. Service application uses stateless and singleton

enterprise java beans as Jersey root resource classes, and annotation for the methods in EJB's local interface to form structured XML as an output. Service application also uses JDBC connection to connect to the MySQL database which lies at the database tier of the application and give the latest result as part of booking and search services.

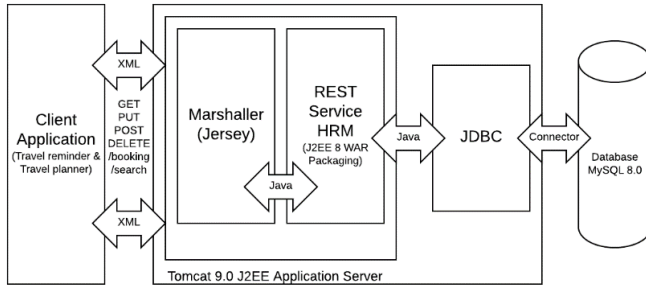


Figure 3 Architecture of service application [1]

V. IMPLEMENTATION

Software implementation involves the transformation of detailed technical design into the valid program with the help of selected framework and programming language to produce a working application. Along with programming implementation phase involves unit testing, debugging, compiling and constructing the complete executable product. Hotel Reservation System is implemented in a J2EE framework with the help of various programming languages at different layers as explained below:

A. Presentation Layer

Graphical user interface of HRS is developed in JSP with the help of programming languages such as HTML 5 which is standard markup language used to create web pages, bootstrap for browser compatibility and responsive behavior of web pages, jQuery for client-side scripting and CSS for styling [11][12][13][14]. HTML is much more elastic and introduced a lot of new tags and attributes for basic validation in its fifth version but most importantly it makes user browser independent of third-party plugins. Apart from these technologies HRM also uses Java Server Pages Standard Tag Library (JSTL) especially for common and structural tags such as iteration and conditional tasks [15]. All commercial API's are consumed from presentation layer especially from JavaScript and authenticated with API keys to form client application of HRS.

For the production environment of HRS "App Service" has been created which is using Java version 8 and Tomcat 9.0 as the web container.

B. Database Layer

Database layer of the HRS is designed in MySQL which is an open source relational database management system and is connected to the java application by using JDBC and MySQL connector. HRS uses the latest version of MySQL (8.0) for the development as well as the production environment. For the development purpose, the MySQL community has MySQL Workbench which is a graphical tool for working with MySQL servers and

databases. MySQL Workbench is mainly used for SQL development, Data modeling, Server administration and Data migration.

For the production purpose of HRS "Azure Database for MySQL server" with the version 5.7 has been created in West Europe region and connected with java application by using JDBC connection.

C. Development Tools and Deployment Environment

Development tools are the software applications which provides a platform for software developers to create, debug and maintain their product. Tools may be simple command line execution or may be part of a single large program called Integrated Development Environment. Development and deployment of HRS are done with the help of the below tools:

1) Java Development Environment

Eclipse is an Integrated Development Environment (IDE) most widely used in Java development also contains base workspace and an extensible plug-in system for customizing the environment. Development of HRS is done in Eclipse Oxygen, JDK 1.8 and Tomcat 9.0 as the web container. Apart from that, multiple external libraries such as Jersey, JSTL etc. are part of the HRS system.

2) Database Development

Database development of HRS is done in MySQL workbench 8.0 which uses MySQL-connector-java-8.0.11.jar to connect java application. For the development purpose, we used the MySQL server at the local machine.

3) Deployment on Azure

Microsoft Azure is a Platform as a Service provided by Microsoft which provides decent features such as Auto manage and scale. Hotel Reservation System is developed and deployed in three different parts which are a client application, service application, and database. Client application and service application are part of Dynamic web project and database is developed in MySQL workbench. For the deployment on cloud Azure Toolkit for Eclipse 3.9.0 plugin has been used which is set up in Eclipse and can directly connect to the Azure account where the user can deploy a web application. For the deployment of the database, MySQL Workbench has been configured to connect MySQL database on the cloud so that queries can be executed directly.

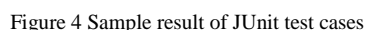
VI. TESTING

Test Driven Development (TDD) is a software development approach which is used for the development of Hotel Reservation System in which development is totally dependent on small iteration. Here are some of the steps which carried out in each iteration:

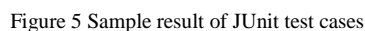
- 1) The software developer must fully understand the technical as well as business requirement.
- 2) Test cases get developed before the development of actual programs.

- During the development of HRS JUnit test cases has been written as well as performance testing of HRS has been carried out by using JMeter. Here is the brief of testing strategies carried out to ensure the quality of the product.

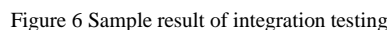
This is the bottom level of the software testing in which individual component or unit of the software application is tested. Unit testing in HRS is done by writing JUnit test cases, a major focus of the test cases is for data access objects which are interacting with the database directly and failure of the code can harm whole application.



During development of HRS, service application provided by HRS has been tested by using external tool postman which is API development and testing environment and provides built-in features for API testing.



Integration testing is a part of software testing in which different modules are combined and tested as a group so that tester can test the behavior of the application as a group. HRS is tested for Integration Testing of second party API integration i.e. Travel Reminder. For this integration of client application JUnit test case has been written to check whether second party API is working with HRS code or not.



Software performance testing is a type of testing practice performed to determine how the system behaves for scalability and responsiveness under the workload. For performance testing of HRS JMeter is used, which is an open source software tool provided by the Apache foundation for analyzing and measuring the performance of services and web applications. Test cases have been written with the help of plugin provided by BlazeMeter which a load testing platform provider is and generates test cases for JMeter. Below summary report has been generated after a test run for ten users and five ramps up period. Report on performance testing and test scripts have been submitted in the project.



The hotel industry is a vast business venture for hotel owner and community of tourists and/or travelers who are traveling for various reasons. Hotel Reservation System is a cloud-based search engine to search hotels in main cities in Ireland and provides a solution for book them. HRS consumes three APIs provided by Google (Google sign in, Geocode, Maps) to form its client application and provides two APIs to serve two different applications (Travel reminder and Travel planner) with its service application. Apart from that HRS consumes one-second party APIs namely Travel Reminder to set travel reminder in Google Calendar.

This study took sample hotels for demonstration purpose. The major objective is to consume different commercial as well as second party APIs to form client application as well as a service application.

- [1] Lucidchart. (2018). Online Diagram Software & Visual Solution | Lucidchart. [online] Available at: <https://www.lucidchart.com/?noHomepageRedirect=true> [Accessed 25 Jul. 2018].
- [2] Maleshkova, M., Pedrinaci, C., Domingue, J., 2010. Investigating Web APIs on the World Wide Web, in: 2010 Eighth IEEE European Conference on Web Services. Presented at the 2010 Eighth IEEE European

Conference on Web Services, pp. 107–114.
<https://doi.org/10.1109/ECOWS.2010.9>

- [3] Helaakoski, H., Iskanius, P., Peltomaa, I., 2007. Agent-Based Architecture for Virtual Enterprises to Support Agility, in: Establishing the Foundation of Collaborative Networks, IFIP — The International Federation for Information Processing. Presented at the Working Conference on Virtual Enterprises, Springer, Boston, MA, pp. 299–306. https://doi.org/10.1007/978-0-387-73798-0_31
- [4] Blake, M.B., Cummings, D.J., 2007. Workflow Composition of Service Level Agreements, in: IEEE International Conference on Services Computing (SCC 2007). Presented at the IEEE International Conference on Services Computing (SCC 2007), pp. 138–145. <https://doi.org/10.1109/SCC.2007.136>
- [5] Quan Z, S., n.d. Web services composition: A decade's overview [WWW Document]. ResearchGate. URL https://www.researchgate.net/publication/262806779_Web_services_composition_A_decade's_overview (accessed 7.25.18).
- [6] Popovic, K., Hocenski, Z., 2012. REST-style Actionsript programming interface for message distribution using Amazon Simple Queue Service, in: 2012 Proceedings of the 35th International Convention MIPRO. Presented at the 2012 Proceedings of the 35th International Convention MIPRO, pp. 512–516.
- [7] Carullo, G., Mauro, M.D., Galderisi, M., Longo, M., Postiglione, F., Tambasco, M., 2017. Object Storage in Cloud Computing Environments: An Availability Analysis, in: Green, Pervasive, and Cloud Computing, Lecture Notes in Computer Science. Presented at the International Conference on Green, Pervasive, and Cloud Computing, Springer, Cham, pp. 178–190. https://doi.org/10.1007/978-3-319-57186-7_15
- [8] Iliadis, I., Sotnikov, D., Ta-Shma, P., Venkatesan, V., 2014. Reliability of Geo-replicated Cloud Storage Systems, in: 2014 IEEE 20th Pacific Rim International Symposium on Dependable Computing. Presented at the 2014 IEEE 20th Pacific Rim International Symposium on Dependable Computing, pp. 169–179. <https://doi.org/10.1109/PRDC.2014.30>
- [9] Techterms.com. (2018). Framework Definition. [online] Available at: <https://techterms.com/definition/framework> [Accessed 25 Jul. 2018].
- [10] Jersey.github.io. (2018). Jersey. [online] Available at: <https://jersey.github.io/index.html> [Accessed 25 Jul. 2018].
- [11] W3.org. (2018). HTML 5.2. [online] Available at: <https://www.w3.org/TR/html5/> [Accessed 25 Jul. 2018].
- [12] W3.org. (2018). Cascading Style Sheets. [online] Available at: <https://www.w3.org/Style/CSS/Overview.en.html> [Accessed 25 Jul. 2018].
- [13] js.foundation, J. (2018). jQuery. [online] JQuery.com. Available at: <https://jquery.com/> [Accessed 25 Jul. 2018].
- [14] Mark Otto, a. (2018). Bootstrap. [online] Getbootstrap.com. Available at: <https://getbootstrap.com/> [Accessed 25 Jul. 2018].
- [15] Oracle.com. (2018). JSTL Documentation. [online] Available at: <http://www.oracle.com/technetwork/java/index-141251.html> [Accessed 25 Jul. 2018].