Review Article

IJCRR
Section: Healthcare
Sci. Journal
Impact Factor
4.016

AN EFFICIENT REAL TIME OHR SYSTEM BY INTEGRATING GOOGLE CLOUD SERVICES

Karthikayini T., Padmapriya M.K., Arya S.

Department of Computer Science Engineering, New Horizon College of Engineering, Bangalore, KA, India.

ABSTRACT

The proposed approach is fully developed based on Google cloud applications and Google cloud services. In this paper, we have developed a web application for hall reservations by integrating Google UI service and Google apps services. It avoids usage of back end servers which in turn reducing the complexity of usage and maintenance. The cloud service also provides flexibility for reserving and cancelling the booked halls. Integrating Google cloud services enables quick and easy generation of reservation reports at anytime from anywhere through any device. Google authorization has in addition provided a centralized access to the resources with unique authentication identifier for every user. The system is made centralized at both the user-level and developer-level where the developer can revise the system from anywhere.

Key Words: Google apps, Google calendar, Google spreadsheet, Google UI service, Google sites, OHR(online hall reservation)

INTRODUCTION

In the recent times, there are various platforms for online hall reservation systems are available. All the traditional hall reservation systems are processed by using back end servers for maintaining the event data. There is [1] a need of installing and maintaining the rack server for those traditional systems. The online booking system for the meeting and seminar halls will provide flexible and sustainable services for saving time and emulate mistakes. From the other hands, users are looking for an interactive and easy way to communicate and do their jobs via internet. The usefulness of providing such a reservation services is to help administration staffs in their daily work by making their reservation up to schedule, connected, and generating reports easily [1]. Rozinah [1] began to study what factors affect the speed of the online reservation system in the centralized environment by classifying the respondents to the booking system by their age, gender, time, date of access, and time to request and to get response. Classification was done on the time constraint basis and it is compared with manual applications on booking system survey. Karthikayini [6] proposed an efficient online lesson plan management system which automate the tasks, requires less level of monitoring and optimization, Server is capable enough for the centralized system which provides efficient query performance by using Google cloud services.

Ahsan Habib [2] began to study what factors affect this propensity in a negative manner i.e. what discourages individuals from adopting the technology. In this approach, middleware is used instead of database, by further reducing the complexity of data maintenance. Bobbitt and Dabholkar [3] has attempted to integrate the various attitude-based theories with external factors (such as the product/service category and perceived risks) to explain why individuals may choose technology-based self-service options [3].

Abu Zakir Rizvi [2] showed that in a single server system, all pressure goes through the server. As a result, it increases the possibility of causing server failure. It also reduces the ability to increase the scale the size of server as the clients increase by the time. Using multi-server system, server load is reduced by distributing the load among other server and that's the way to improve the server related major issues like failure possibility, scalability etc. A middleware based three-tier architectural system provides a more easy way to manage clients' request to different RMs/servers and to giving output back to clients from server. Ultimately, this approach provides benefits such as reusability, flexibility, manageability, maintainability, and scalability.

This paper mainly focuses on the existing booking system for the meeting and seminar halls of Universities, which

Corresponding Author:
Karthikayini T., Department of Computer Science Engineering, New Horizon College of Engineering, Bangalore, KA, India.
E-mail: karthikayini@outlook.com

Received: 02.05.2015 **Revised:** 30.05.2015 **Accepted:** 23.06.2015

face several issues regarding the booking procedures that mainly performed manually. Overcoming the drawbacks in [2], this proposed approach is used in developing web applications integrated with distributed Google servers. It avoids the back-end connectivity of the web application with database, single server or even with a multiple servers. Further, this system provides no complexity of maintenance, additional flexibility, any manageability and scalability.

Instead of having local server for reservation system, the proposed system uses Google Calendar as a data storage (service provided by Google) to develop an online web application for hall reservations. Google app script is an online scripting tool which is quite similar to java script, is used to create the proposed system. The whole architecture is designed for the user-friendliness and automatism. Because exiting models always needs a system to be updated all the time. But in proposed system, once the system is developed, each amendment can be made automatically and can achieve consistency easily.

The proposed system is developed in a centralized way and can be used in a decentralized manner such as through any device, at anytime and from anywhere. The developers can do troubleshooting, code optimization and usage can be done globally from anywhere.

AN OVERVIEW OF GOOGLE CLOUD SERVICES

Google's providing both SaaS and PaaS solutions in cloud computing. Some of the example for SaaS solutions including Google Apps which including Gmail, Doc, etc., and PaaS includes Google App Engine [8]. In the Platform as a Service (PaaS) space Google is a key player. App Engine is a platform to create, store and run applications on Google's servers using development languages as java and python. App Engine contains various tools to deal with the data store, site management and its resource utilization, checking errors and classification. A user can serve the app from his own domain name (such as http://www.example.com/) using Google Apps. Or, he can serve his app using a free name on the appspot.com domain.

A user can share his application with the world, or limit access to members of organization. App Engine costs nothing to get started. All applications can use up to 1 GB of storage and enough CPU and bandwidth to support an efficient app serving around 5 million page views a month, absolutely free[5]. Applications requiring more storage or bandwidth can purchase which is divided into five buckets: CPU time, bandwidth in, bandwidth out, storage, and outbound email.

Google App Engine enables users to build a basic web application very quickly. Configuring and setting up an application is quick and easy. The Google App Engine Architecture provides a new approach without dealing with web servers and load balancers but instead deploying the applications on the Google App Engine cloud by providing instant access and scalability. The Google App Engine Software Development Kit (SDK) provides Java and Python programming languages. The languages have their own web server application that contains all Google App Engine services on a local computer. The web server also simulates a secure sandbox environment. The Google App Engine SDK has APIs and libraries including the tools to upload applications. The Architecture defines the structure of applications that run on the Google App Engine [1].

JavaScript: The Google App Engine allows implementation of applications using Java Script and Google scripting language and running them on its interpreter. The Google App Engine provides rich APIs and tools for designing web applications, data modeling, managing, accessing apps data, support for mature libraries and frameworks like Django [1].

Google offers services like cloud endpoints, translate APIs, prediction APIs, also a variety of storage capabilities like Google cloud SQL, Cloud data store, Cloud storage for different volume of data. Even the Petabytes of data can be processed in seconds using Google Compute Engine. In addition Google provides Big Data service like Big Query which doesn't require an infrastructure to manage, a database administrator by using familiar SQL and can take advantage of a pay-as-you-go model [7].

ARCHITECTURE OF ONLINE HALL BOOKING SYSTEM

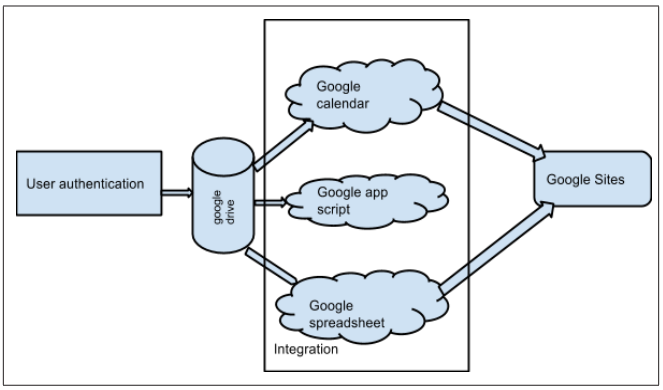

Figure 1: Architecture of online hall booking system

The entire architecture (Fig.1) speaks about the integration of various Google products which are used to design the proposed system. It shows that there is a way to integrate any Google products and can develop the application. Various Google products such as Google calendar, Google app script, Google spreadsheet & Google site is

integrated here. According to the requirement of application, the developer can choose the Google products to make the application more useful & user-friendly for the end-level users.

In the proposed system, the user authentication is needed in order to develop the application. Google drive which is an online storage cloud resource with the collection of various Google products such as Google Document, Google Spreadsheet, Google Calendar, Google App script and more. Google app script, an online script editor tool, is used to develop the scripts. Google calendar is acting as a database for storing the hall reservation information in the respective date, month and year. Plenty of classes and methods available from the Google app script tutorial, out of which, Calendar class and its methods were chosen to manipulate the data. In Google calendar, either default calendar or any calendars can be created and used in the script. Day agenda, week, month agenda is available in it. Once the end-level user registers the halls, the inputs are prompted and exported to the specified Google calendar events. The Google spreadsheet is used for report generation and be either implemented in one of two ways.

a. The user-inputs can be directly imported to Google Spreadsheet and can be kept visible for users in order to ensure whether the same hall is booked for the same date and same time. User can view the sheet and decide for booking without any overlaps with the previous registration. The system will deny the user if he tries to book on the same day and same timings for a same hall.
b. Data can also be imported from Google calendar to Google spreadsheet. Different sheets can be used for displaying each and every month hall reservations.

A separate website is created for the hall reservation online application using Google site. It is possible to make the site visible for the entire world. Sharing and managing access permissions are also available for the site. Hall reservation app script URL link and Hall cancellation URL link is inserted in a site and it is made visible lively.

System Design and implementation:

The Fig 2 describes that; the user will be first authenticated through the email-id in order to access the application. Google site is created especially for this application and provide the access permission and sharing settings. In order to access the site, it redirects the user directly to Gmail sign-in page. Appscript link is inserted for both hall reservation and hall cancellation. To book the hall, Google calendar is inserted as a view inside the site itself for the users, to look at the already booked events. After viewing the calendar, the user can decide accordingly to reserve the halls or cancel the halls. So that, it's an initial level of revising that no overlaps have occurred among the events by the user-level itself. User can click on listed links such as Hall-reservation and Hall-cancellation.

Designing user interface:

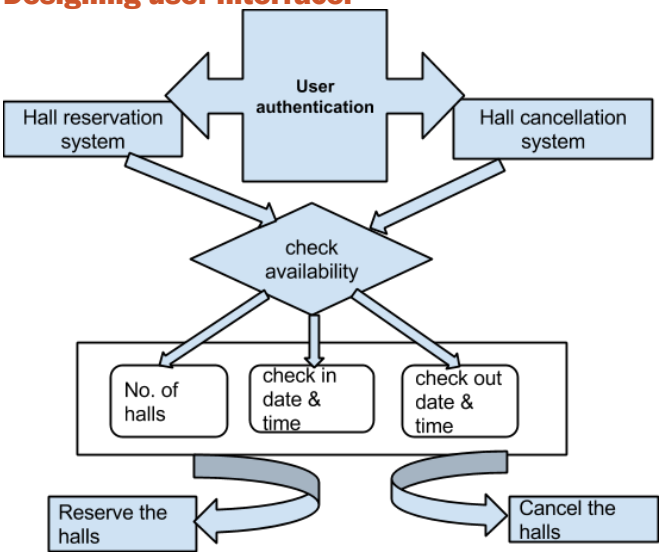

Figure 2: System design

Hall -reservation:

Hall reservation link will be available in the site. Once the user clicks the link, this system will authorize the user for accessing the application. If not, it denies the user to access the application and just displays a message that "You are not authorized to access this app". After user authentication, it lets that particular app script to execute and shows the panel to the user. Now user can start registering the form and can click submit. The server handler will directly calls the Class Calendarapp, as well as Spreadsheet app and imports the user-inputs directly according to the date picked by the user. Incase if the user gives the same date, same time and trying to book the same hall, then the system denies the user to do it.

Hall-cancellation:

If the user desires to cancel the reserved halls, system will ensure for the authenticated user and then it abandons the reserved halls. Both hall reservation and hall booking system are synchronized to Google calendar through app script. All the events which is registered and cancelled by a particular user will be reflected in Google calendar along with date and time stamps. Here the user can be able to cancel the booked halls if and only if he is the one who created that particular event and nobody else can cancel that event. The system will deny the user for cancelling the events created by other users. The user fills out the form and the system is automatically removing those events from the Google calendar and Google Spreadsheet respectively.

Advantages:

The proposed system provides an automated reservation system and once the system is developed, it requires 0% man power. The query performance will be better when compared to other conventional systems. There is no probability of server crash due to the mirroring storage technology; the documents are distributed to the servers located in various places in the world. Even though the mirroring technology enables increases memory usage, various algorithms and compression techniques are implemented to overcome the disadvantages of the mirroring technology. Ultimately the system remains fault tolerant and provides 24x7 supports for incoming requests. The traditional system uses a separate standalone server and database for running the application which requires a man power. Instead using database to store the events, the proposed system is using Google calendar to store the events. So it's an effective and intelligent way to use this product. Google Calendar is attached to the users Gmail-id itself. Hence forth, there is no need to install any kind of software and the system is made using ready-made Google products. The privileged users can access the product from anywhere, at any time through any device.

CONCLUSION AND FUTURE WORK

With the vast and rapidly growing technology "Cloud Computing" which is a new evolving research field since the entire world is moving towards cloud, that attracts the attention of an astonishingly diverse set of software developers or researchers from various related fields. We have used Google APIs, global cloud service provider that gives high energy efficiency and supports Go-Green environment by reducing paper work. It gives a substantial impact to the entire world economy if it is implemented in all major business areas. The proposed approach will emerge and has the potential to become a very popular in near times. We believe that our work has created an efficient automatic framework. In future, one more module can be included for modifying the reserved slots and Google analytics can be integrated to the site to check the number of users visited, including the browser and the country as well.

REFERENCES

1. Omar Abdullah M. Al-Maktari, Rozinah Jamaludin, Al-Samarraie Hosam, "The Acceptance of Online Booking System (OBS) Based on the Theory of Reasoned Action (TRA): A Case of Sana'a University", International Journal of Scientific & Engineering Research, Volume 3, Issue 2, February-2012.
2. Mohammad Badrul Alam Miah, Md. Abu Zakir Rizvi, Md. Ahsan Habib, Khandaker Hamidul Haque,"A Multipurpose Online Reservation System In Distributed Environment Using Middleware Architecture" International Journal Of Scientific & Technology Research Volume 2, Issue 2, February 2013.
3. L. M. Bobbitt and P. A. Dabholkar, "Integrating attitudinal theories to understand and predict use of technology-based self-service: the internet as an illustration," International Journal of Service Industry Management, vol. 12, pp. 423-450, 2001.
4. Rabi Prasad Padhy, Manas Ranjan Patra and Suresh Chandra Satapathy, "X-as-a-Service: Cloud Computing with Google App Engine,Amazon Web Services, Microsoft Azure and Force.com", International Journal of Computer Science and Telecommunications Volume 2, Issue 9, December 2011.
5. Site:baselinklabs.blogspot.com/2014_09_01_archive.html
6. Site:pnrsolution.org/DataCenter/Vol3/Issue1/32.pdf
7. Site: https://cloud.google.com/