Software as a Service (SaaS) for Management information system using multiple tenants

Rachana Desale, Purva Kolhatkar, Anju More, Piyush Katira, Vishal Kokane Prof.S.M.Jaybhaye

(Department of Information Technology, Sinhgad College of Engineering, Pune University,Pune)

ABSTRACT

Today, reliable broadband internet access, service-oriented architectures (SOAs), and the cost inefficiencies of managing dedicated on-premises applications are driving a transition toward the delivery of decomposable, managed, shared, Web-based services called software as a service (SaaS).

Tenants using a multitenant service operate in virtual isolation from one another; Organizations can use and customize an application as though they each have a separate instance, yet their data and customizations remain secure and insulated from the activity of all other tenants. The single application instance effectively morphs at runtime for any particular tenant at any given time. Multitenancy is an architectural approach that pays dividends to both application providers and users. Operating just one application instance for multiple organizations yields tremendous economy of scale for the provider. Only one set of hardware resources is necessary to meet the needs of all users, a relatively small, experienced administrative staff can efficiently manage only one stack of software and hardware, and developers can build and support a single code base on just one platform (operating system, database, etc.) rather than many. The economics afforded by multitenancy allow the application provider in turn offer the service at a lower cost to customers.

Keywords-Cloud Computing, Multitenancy, SaaS, Virtualized database.

I. Introduction

Leave management system is an MIS that is required in every company, however big or small the company might be. Hence using the concepts of SaaS for multiple tenants we are going to build a real time flexible leave management system which will morph according to any company requirements and keep individual customizations and data of each company intact.

II. Cloud Computing

Cloud Computing is Internet based (Cloud) development with the use of computer technology (Computing). Cloud Computing is an emerging technology in the area of parallel and distributed computing. Clouds consist of a collection of virtualized resources, which include both computational and storage facilities that can be provisioned on demand, depending on the user's needs.

Cloud computing is defined as a computing capability that provides an abstraction between the computing resource and its underlying technical architecture (e.g. servers, storage, networks), enabling convenient, on demand network access.

There are 3 types of cloud computing:

1) SaaS-Software as a Service
Applications run on hosted servers as a service.
2) PaaS-Platform as a Service
3) IaaS-Infrastructure as a Service

III. SaaS

SaaS (Software as a Service) is an application hosted on a remote server and accessed through internet.

Software as a Service (SaaS) is a model in which application is hosted as a service to customers who access it via the Internet Software as a service, sometimes referred to as "on-demand software", is a software delivery model in which software and associated data are centrally hosted on the cloud. Software as a service (or SaaS) is a way of delivering applications over the Internet as a service. Instead of installing and maintaining software, you simply access it via the Internet, freeing yourself from complex software and hardware management. The most important difference between SaaS applications and ordinary apps is multi-tenancy.

- **A multitenant web application framework for SaaS (IEEE Paper)[8]:**

In the framework, multiple tenants share a single application server, and this lowers operational costs. Industry-specific, tenant specific and department-specific customizations are supported by multilevel inheritance. It promotes reuse of user interface and business logic components. The system can scale by adding SaaS application servers. This also improves the availability of the system.

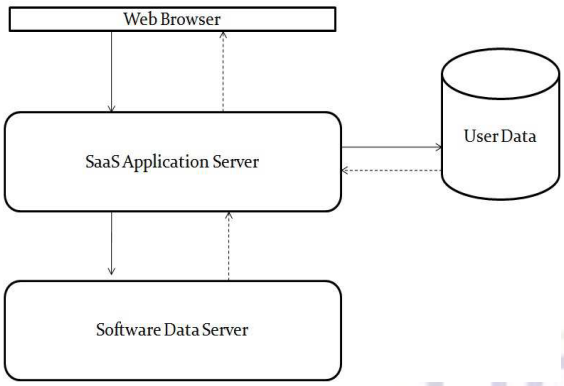


Fig.1. Web application framework for SaaS

IV. Virtualized Database

Virtualization enables compute and storage resources to be pooled and allocated on demand. This enables both the sharing of single server resources for multi-tenancy, as well as the pooling of server resources into a single logical database or cluster. Database virtualization provides increased flexibility, more granular and efficient allocation of pooled resources, and more scalable computing.

Database virtualization means different things to different people. In our application, different organizations have different employees and have different leave records, status and different leave policies of different organizations. Everything is managed with the help of virtualized database. Our focus is that databases are accessible and manageable as if they were a single database. Virtualization provides a common framework for better availability, scalability, mobility, manageability and security. So it is better to use virtualized database in multitenant web based application.

Advantages of virtual database:

- Enhanced database performance.
- Pooling and sharing computing resources either splitting them (multi-tenancy) or combining them (clustering).
- Simplification of administration and management to approve or reject leaves.

V. Multitenancy

Multitenancy is the requirement for SaaS Vendor to be successful. Multitenancy refers to a principle in software architecture where a single instance of the software runs on a server, serving multiple client organizations (tenants). Multitenancy is contrasted with a multi-instance architecture where separate software instances (or hardware systems) are set up for different client organizations. With a multitenant architecture, a software application is designed to virtually partition its data and configuration, and each client organization works with a customized virtual application instance. This capability to have multiple organizations (called tenants in the SaaS nomenclature), co-exist on the same application without compromising the security of data for those organizations defines the application as a multi-tenant one.

In Multitenant Architecture all users and applications share a single, common infrastructure and code base which is centrally maintained.

There are several levels of multi-tenancy:

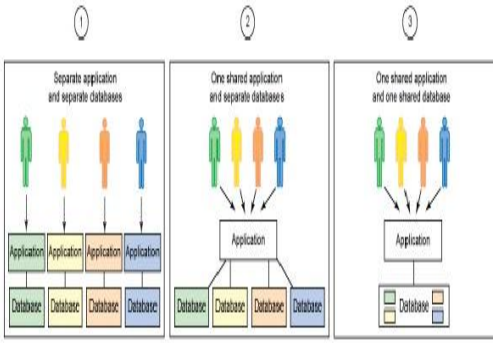


Fig.2. Multitenancy models

1. Simple virtualization in the cloud where only the hardware is shared.
2. Single application with separate databases per tenant.
3. Single application and shared database (highest efficiency, true multi-tenancy).

VI. Existing Approaches For Multitenancy Software

Architectural Approach of Salesforce's [1]:

The significant development in SaaS for multiple tenants has been done by the company Force.com. Using standards-based Web service APIs and native platform development tools,

Force.com is a generalized Internet application development and delivery platform on which individual enterprises and service providers have built all types of custom business applications, including supply chain management, billing, accounting, compliance tracking, human resource management, and claims processing applications. Force.com developers can easily build all components of a Web-based application, including the applications data model (tables, relationships, etc.), user interface (data entry forms, reports, etc.), business logic (workflows, validations, etc.), integrations with other applications, and more.

Force.com's foundation is a metadata driven software architecture that enables multitenant applications by using pivot tables, field tables etc.

VII. Existing Tools

Existing System:

There are a few companies which have provided Leave management system using Software as a Service as follows:-

- **Justlogin.com's eLeave:[10]**
  - Online application for leave without the need for paper work.
  - Email notifications to approving officer and Applicants.
  - Leave approval through normal email without the need to login.

- **HR2000 eLeave:[11]**

1. Another leave management system, however it is based on multi-instance architecture.
2. While the database is updated online this software requires users to install its instance on their own machines

- **EGOTEC's SaaS Leave Management and Time Recording:-**

1. Department heads have a view of the lists of the leave requests to be approved. Integration of Leaves in the time-recording process.
2. Absence calendar integration in Google Calendar.
3. Views and reports: List view of several employees, their leaves and absences.

VIII. Experimental setup

Hardware Interfaces

- Windows operating system
- Linux operating system

Software Interfaces:

Software Used	Description
Operating System	Windows XP, Windows 7,Linux
Database	MySql
Ruby on Rails	To implement the project we have used Ruby on Rails as it is made open source for web development

Developing Software as a service (SaaS) for multiple tenants and developing an application demonstrating it. (Real time flexible leave management system).

The aim of this project is to develop "Designing architecture for Software as a Service for multiple tenants developing an application demonstrating it (Real time flexible leave management system)".

The leave management system plays an important core of many companies. Developed delivery solution will provide suitably designed as well as the customized service packages to its employees. The leave management system keeps detailed records of all types of leaves taken by employees, leave application form, leave status, leave balance, employee database etc

1. Creating Rails Application using MVC Architecture:

The MVC (Model-View-Controller) architecture is a way of decomposing an application into three parts: the model, the view and the controller.

- Model consists of data part, database and any programmatic operation (calculations).
- View describes about web pages what end user sees.
- Controller is the programming glue that joins model to the view.

2. Deploying Rails Application on cloud:

Heroku is a platform for instant deployment of Ruby on Rails Web applications. In the Heroku system, servers are invisibly managed by the platform and are never exposed to users. Applications are automatically dispersed across different CPU cores and servers, maximizing performance and minimizing contention. Heroku has an advanced logic layer than can automatically route around failures, ensuring seamless and uninterrupted service at all times.

IX. Database

MySql Database.

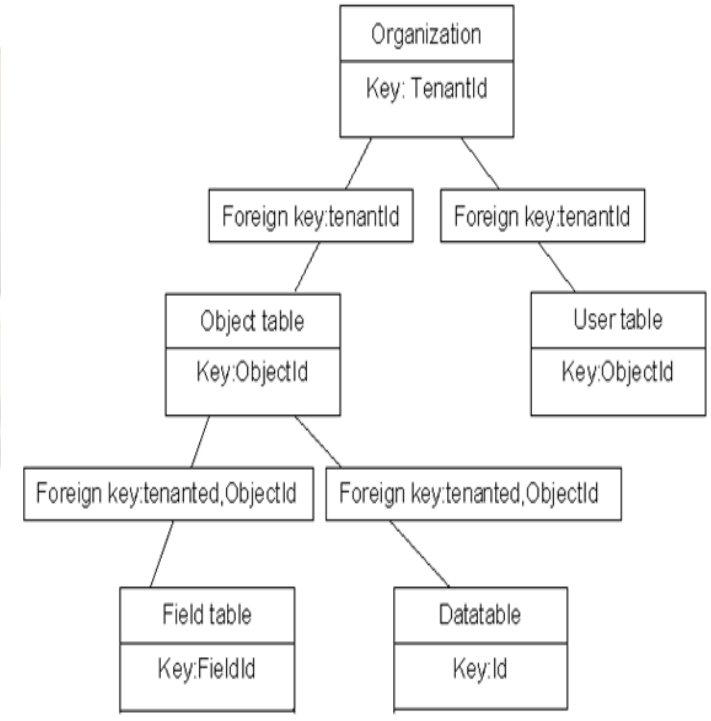


Fig.3. Diagrammatic representation of metadata-driven multi-tenant database

We have used metadata in the organizations, objects, fields tables to define application organizations, objects, fields and to map corresponding data stored in the large datatables database table.

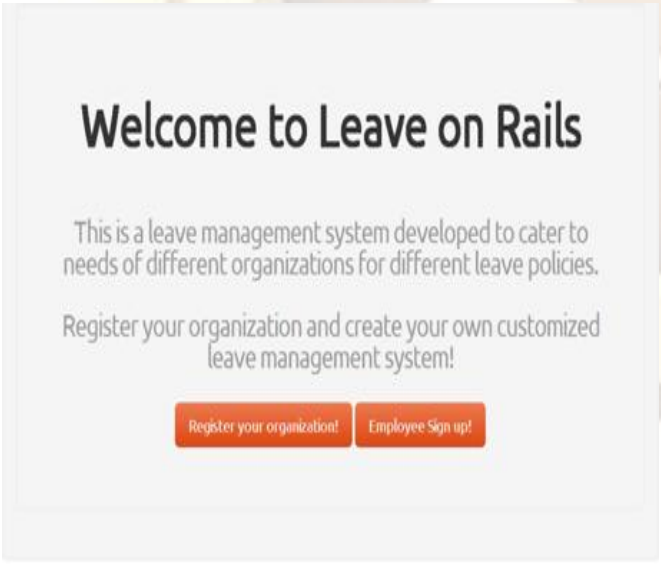
X. Comparisons and evaluation

Difference in our System

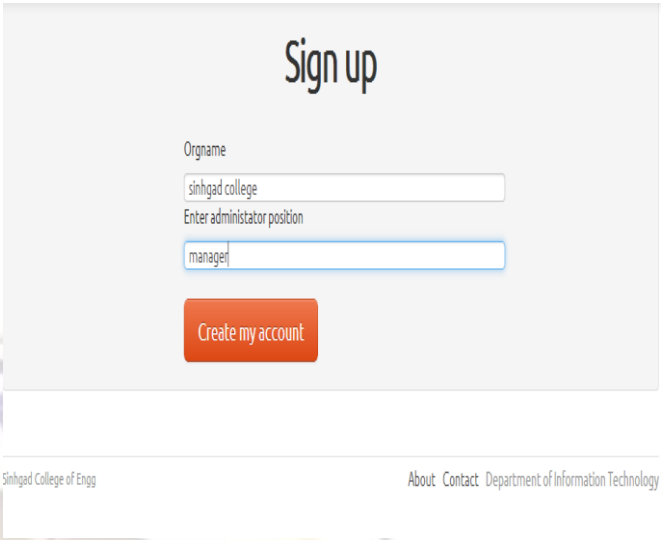
Many software are based on multi-instance architecture where separate software instances are set up for each client organization. The one's that support multi-tenancy in various ways they tend to be too complex and to lack generality. The multitenant architecture that is going to be designed is based on the architecture developed by salesforce.com. With a multitenant architecture, a software application is designed to virtually partition its data and configuration, and each client organization works with a customized virtual application instance. Instead of collecting data from multiple data sources, with potentially different database schemas, all data for all customers is stored in a single database schema. Thus, running queries across customers, mining data, and looking for trends is much simpler. This project will provide SaaS for multiple tenants which will be less complex and generic than the ones which have been provided before.

Screenshots of GUI implementation:-

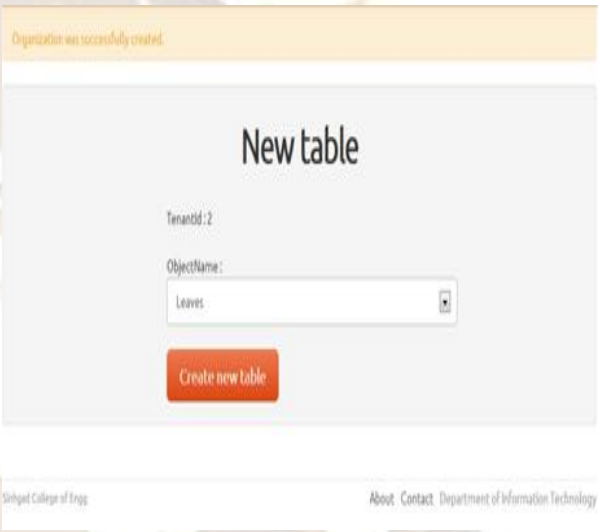
Home Page:-



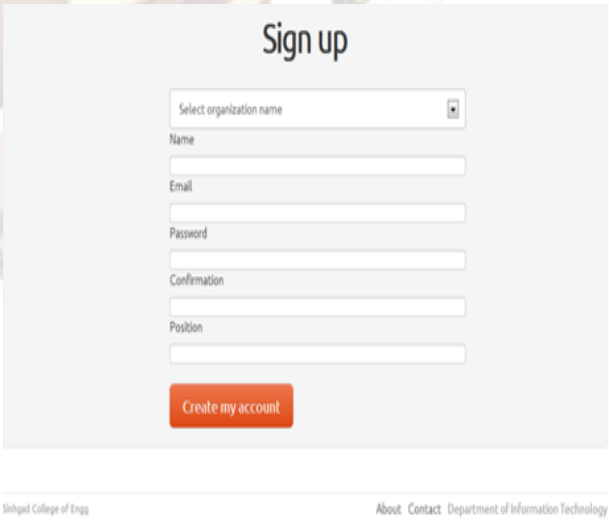
Organization Sign up:-



Creating New Table:-



Employee Sign up:-



XI. Conclusion

In this paper, we have developed a multi-tenant application framework to support and manage the variability of SaaS applications and tenants-specific requirements.

References

[1]. Force.com, "The Force.com Multitenant Architecture Understanding the Design of Salesforce.com's Internet Application Development Platform", White paper

[2]. Qihong Shao "Towards Effective and Intelligent Multi-tenancy SaaS" ,UMI Dissertation publishing, May 2011

[3] International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-1, March 2012 421

[4] UNDERSTANDING The Cloud Computing Stack SaaS, Paas, IaaS, © Diversity Limited, 2011 Non-commercial reuse with attribution permitted.

[5] Volume 2, Issue 1, January 2012 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering.

[6] K.H. Bennett et al., "An Architectural Model for Service-Based Software with Ultra Rapid Evolution," Proc.IEEE Int'l Conf. Software Maintenance (ICSM 01),IEEE CS Press, 2001, pp. 292–300.

[7] Haitham Yaish, Madhu Goyal, George Feuerlicht, "An Elastic Multi-tenant Database Schema for Software as a Service", 2011 Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing.

[8] Wonjae Lee, Min Choi," A Multi-tenant Web Application Framework for SaaS", 2012 IEEE Fifth International Conference on Cloud Computing

[9] Lei Ju, Bikram Sengupta,Abhik Roychoudhury," Tenant Onboarding in Evolving Multi-tenant Software-as-a-Service Systems", 2012 IEEE 19th International Conference on Web Services.

[10] www.justlogin.com/leave-management-software,ELeave.

[11] www.hr2000.com,e-Leave Management System.