Review on Decision Making Tool Based on Cloud Multi-tenancy

Prof. Vidya Shitole[1], Rutuja Kulkarni[2], Kshitija Kulkarni[3],Kajal Kadam[4]
Department of Computer Engineering
[1, 2, 3, 4]SRTTC College of engineeringSavitribai Phule Pune UniversityPune India

Abstract*-Multinational companies have several branches over the world. For taking a decision regarding an organization employees need to communicate with each other and for taking each decision it is not convenient to meet every time and also it is not possible. So, we are making a decision making tool which takes the opinion from employees of the company and analyze it and take optimal decision which has support of historical data and previous result. Our system generates the result and can represent it in the form of pie-charts and graphical format.*

Keywords-cloud, cloud computing, multi-tenancy.

I. INTRODUCTION

Many companies and even govt. sectors need some platform or some tool to manage their communication and it also help them to take some crucial decisions on different level of management. For betterment and progress of any private or government organization it is important to see the issue or any idea from different views to get good results in future. These include discussion of colleagues or members of organization to manage their abstraction at different levels. It improves the quality of decisions. It gives the decisions on the basis of analysis done by our decision tool. It gives the formal review of the decision by representing the information in visual form to get easily understand by all members in the community. It's remarkable and unusual platform that will be available for any organization that wants to avail it. Its Completely non identical from the social website as its motive and use is limited to only organization domain and only their members have complete access to it. It is more than communicating between walls of the office, this means you can chat with colleagues across the world in real time.

We are managing the different ideas of members in organization using our decision making tool. This tool also analyzes the previous record and helps us to take best decision for future improvement. For better understanding we are representing the result in the form of graphs. The cloud is a recent technology which helps these multiple organization to share and analyze the data ubiquitously. Cloud multi- tenancy allows accessing the same instance of application for different organization.

II. ARCHITECTURE

1. System Architecture

Routing System -Routing is the act of moving informationacross an inter-network from a source to a destination. It's alsoReferred to as the process of choosing a path over which to send the packets. Routing protocols use metrics to evaluate what path will be the best for a packet to travel. A metric is a standard of measurement; such as path bandwidth, reliability, delay, current load on that path etc; that is used by routing algorithms to determine the optimal path to a destination.

Routing is the process of selecting best paths in a network.In the past, the term routing also meant forwarding network traffic among networks. However, that latter function is better described as forwarding. The routing algorithm is a formula that is stored in the router's memory. The routing algorithm your protocol uses is a major factor in the performance of your routing environment. The purpose of the routing algorithm is to make decisions for the router concerning the best paths for data.

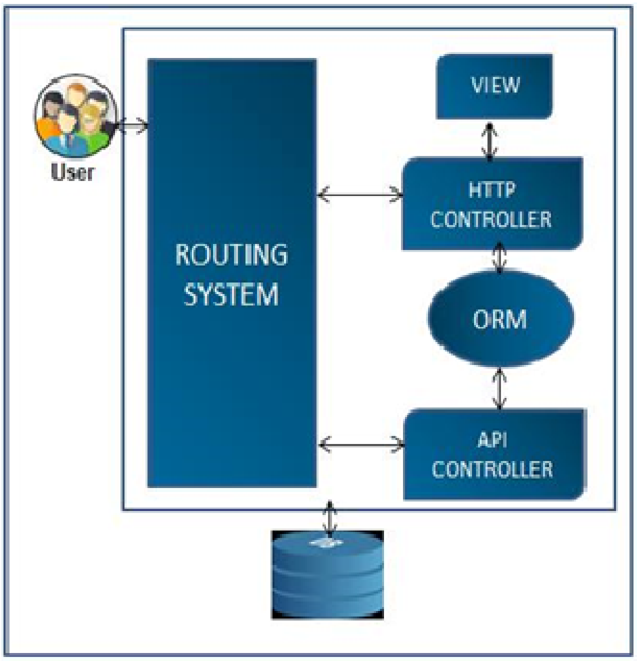


Figure 1

II. RUBY ON RAILS

It is free and open source framework and best for web applications. it’s easy to use and it provides the structure for all code they write. The Rails framework helps developers to build websites and applications, because it abstracts and simplifies common repetitive tasks. That’s why for the sake of simplicity in web designing we choose this framework to develop this web tool. Rails is good for rapid application development (RAD), as the framework makes it easy to accommodate changes

Adding users and authentication

As its multi-tentant application multiple number of users are accesing this remotely.so for the security purpose of the users information we use a library for keeping the confidentiality of user information and authentication.

It’s a gem in ruby that is most commonly used for authentication.

DEVISE gem

Devise is a popular authentication solution for Rails applications. It provides a full gamut of features, and can be configured to meet all, but the most unusual, requirements. In this tutorial, we are going to look at how Devise is implemented on a high level, go through configuration and common use patterns of the gem, how to set up authentication with Devise, and finally demonstrate how to write tests with Devise.

Installing and using devise gem

To have Devise completely installed in an existing Rails app, (where a user’s model class is named, User), the following six shell commands are required:

1.echo "gem 'devise'" >> Gem file
Gem file and add line: gem 'devise'
2.Bundle install
Fetch and install the gems
3.rails generate devise:install

This command is going to install a config file in the initializers folder and then it will also install some messages that you can customize for the notices that you see in the application.

Now, once this is done, it makes some messages here such as the config action mailer options and this is for when it sends out an email and you want to make sure that they put in the right host name in the URL. So when you're in development they recommmend... If you're using localhost:3000, to go and operate with that URL.

And then they recommend having a route in your Rails application and putting in notices and alerts so that you can see what's going on.

4. rails generate devise user

This is going to generate a User model for us, and it will populate it with an email address and a password, and a bunch of other automatically tracked options, such as the last time they signed in, the last IP address they signed in, and a bunch of things like that, and we're just going to use the defaults and generate this. Now that we have that we can run

5. rake db:migrate

to create our users, and if we refresh our application after generating on the devise stuff, you will get an undefined method 'devise' for User When you create your users with devise it adds a line in there that says devise and the options that it's going to run:

Create user table

6. rails generate devise:views users

and this will install the views, the html.erb templates for all the different devise screens that you can have. I recommend doing that because I almost always customize it and add bootstrap-html around the form fields so that it looks pretty. And then lastly, we need to run

Creates (minimal) views

CKEditor

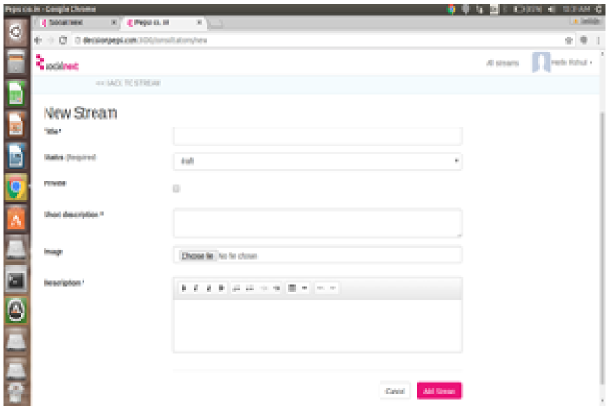


Figure 2

Web content is created in HTML, a markup language that includes both text and special tags that describe this text as well as add semantic meaning to it. These tags are used to mark particular content elements as "headings", "paragraphs", "bold text", etc. CKEditor merely hides the HTML tags from the user and shows their equivalents in the graphical user interface (like the Bold toolbar button) as well as outputs the text with formatting already applied.

This has some consequences, though. CKEditor works on HTML code and to do it well, it needs to get proper HTML code as input. If the source document is incorrect, CKEditor will attempt to fix it in order to be able to apply its content transformations.

CKEditor is not a human-being, though, so it is unable to guess your intentions when you created incorrect HTML code which may lead to some unexpected consequences. In order to avoid problems the input provider (you!) needs to make sure that what CKEditor gets is clean, standards-compliant HTML code. If this is the case, CKEditor will work as intended and output clean, standards-compliant HTML code in return.

We successfully used this tool for our post module as shown infigure.

III. ORM

ORM is the acronym for Object Relational Mapping.Object-relational mapping (ORM, O/RM, and O/R mapping) in computer software is a programming technique for converting data between incompatible type systems in object-oriented programming languages. In Ruby On rails framework ORM is implemented through active record

1. What is Active Record?

Active Record is the M in MVC - the model - which is the layer of the system responsible for representing business data and logic. Active Record facilitates the creation and use of business objects whose data requires persistent storage to a database. It is an implementation of the Active Record pattern which itself is a description of an Object Relational Mapping system.

2. Active Record as an ORM Framework

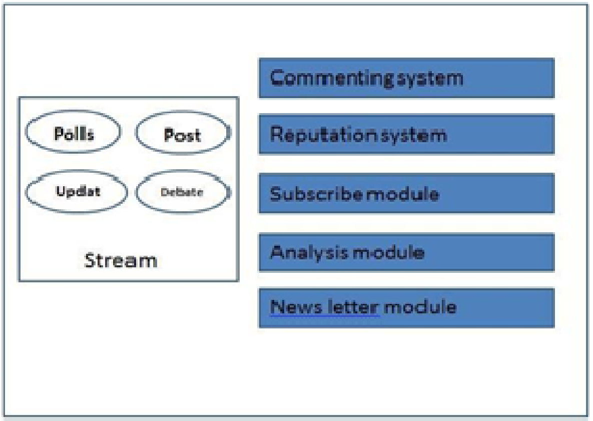
Active Record gives us several mechanisms, the most important being the ability to:

Represent models and their data.

Represent associations between these models.

Represent inheritance hierarchies through related models.Validate models before they get persisted to the database.Perform database operations in an object-oriented fashion

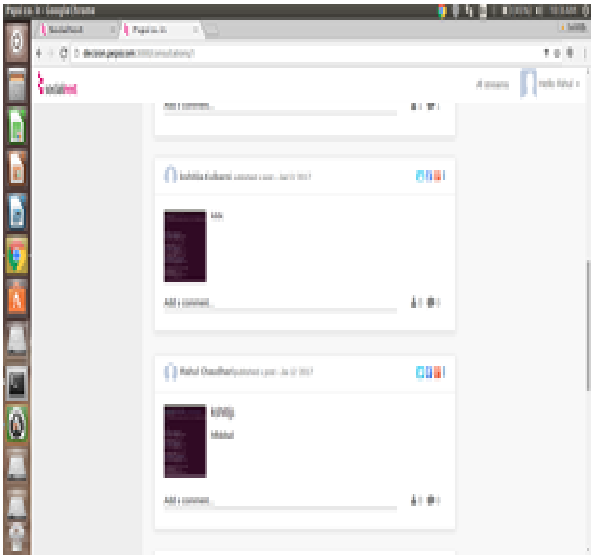
IV. COMPONENT DIAGRAM



V. MODULES IN SYSTEM

1. Post:

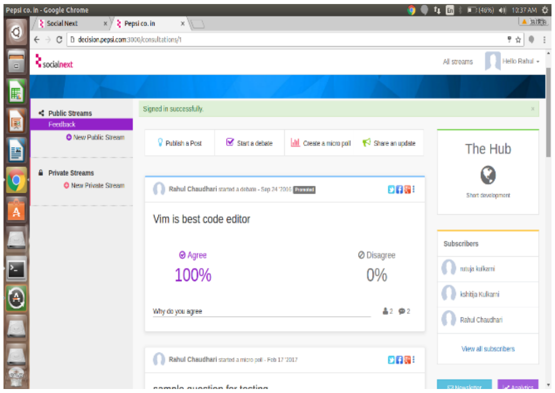
We can post anything related to organization events or Updates about projects



Figure

2. Debate

A formal discussion on a particular matter ina public meeting or legislative assembly, in which opposing arguments are put forward and which usually ends with a vote.



Vim is best code editor

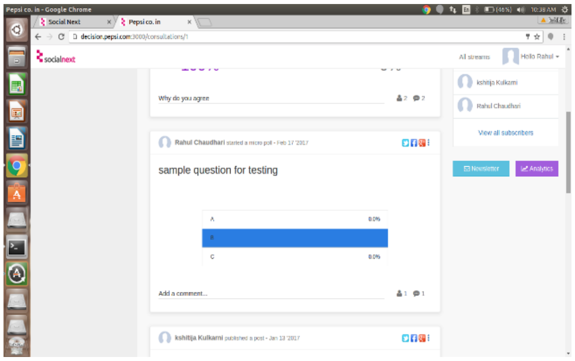
Agree 100%

Disagree 0%

Figure 3

3. Poll

People give their judgement or point of view about particular subject or issue related to organization



sample question for testing

Subscription model: - it is a simple model where user get added to private stream. We have 2 types of stream.private stream and public stream and both are uses subscription module

Public stream:-if stream is public then all user of a system become a subscriber of the system.

Private stream:- if the stream is private then only admin will choose whom to add as a subscriber. Then subscriber will get notification for any activity happens.

Newsletter module: - for notification purpose we use newsletter module which just send mail to all subscriber of the stream. Newsletter will be send by admin manually to all stream subscribers.

Apart from these we have basic and standard modules called authentication and authorization modules.

Authentication: Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be. Authentication is any process by which a system verifies the identity of a User who wishes to access it. Since Access Control is normally based on the identity of the User who requests access to a resource, Authentication is essential to effective Security.

Authorization:- Authorization, by contrast, is the mechanism by which a system determines what level of access a particular authenticated user should have to secured resources controlled by the system.

Authentication = login + password (who you are)

Authorization = permissions (what you are allowed to do)

VI. MULTI-TENANCY

Our application is multi-tenant. Multi tenancy is critical technology to allow one instance of application to serve multiple organizations by sharing resources.

Multi- Multiple, independent are served.

Tenant- It is any legal entity responsible for data and is provided on contractual basis.

In cloud computing model, a "SaaS provider" is the organization that provides a domain specific SaaS App to its users. "Multi-tenancy" is for the benefit of the service provider so they can manage the resource utilization more efficiently, but multi-tenancy is not to the tenant's advantage at all.

The multitenant application design was created to enable multiple users (tenants) to access the same application logic simultaneously. Each tenant has its own view of the application that it uses, administers, and customizes as adedicated instance of the software while remaining unaware of other tenants that are using the same application.

Multitenant applications ensure that tenants do not have access to data and configuration information that is not their own.

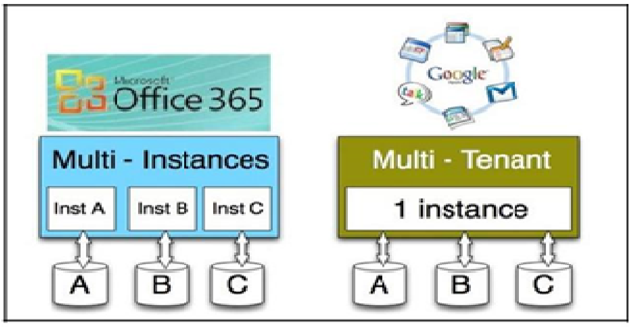


Figure 3

Tenants can individually customize features of the application, such as:

- User Interface - Tenants can define a specialized "look and feel" for their application interface.
- Business Process - Tenants can customize the rules, logic, and workflows of the business processes that are implemented in the application.
- Data Model - Tenants can extend the data schema of the application to include, exclude, or rename fields in the application data structures.
- Access Control - Tenants can independently control the access rights for users and groups.

Multitenant application architecture is often significantly more complex than that of single-tenant applications. Multitenant applications need to support the sharing of various artifacts by multiple users (including portals, data schemas, middleware, and databases), while maintaining security levels that segregate individual tenant operational environments.

Common characteristics of multitenant applications include:

- Usage Isolation - The usage behavior of one tenant does not affect the application availability and performance of other tenants.
- Data Security - Tenants cannot access data that belongs to other tenants.
- Recovery - Backup and restore procedures are separately executed for the data of each tenant.
- Application Upgrade - Tenants are not negatively affected by the synchronous upgrading of shared software artifacts.
- Scalability - The application can scale to accommodate increases in usage by existing tenants and/or increases in the number of tenants.
- Metered Usage - Tenants are charged only for the application processing and features that are actually consumed.
- Data Tier Isolation - Tenants can have individual databases, tables, and/or schemas isolated from other tenants. Alternatively, databases, tables, and/or schemas can be designed to be intentionally shared by tenants.

VII. CONCLUSION

We implemented system which can successfully take decision by analyzing the historical data and the views or judgments by authorized members of organization.this tool has ability to take quick and optimal decisionand it also represents the analysis in graphical format so it makes it easier to perceive the analysis for taking future steps. It will lead to overall progress of organization.

FUTURE SCOPE

We can add multiple features like video conferencing or online live video streaming. We can also implement some intelligent algorithms for decision making by the tool itself so there will be less efforts by team of organization

REFERENCES

[1] Dan Gonzales, Member, IEEE, Jeremy Kaplan, Evan Saltzman, Zev Winkelman, Dulani Woods ” Cloud-Trust - a Security Assessment Model for Infrastructure as a Service (IaaS) Clouds”

[2] Pooyan Jamshidi, Aakash Ahmad, and Claus Pahl, Member, IEEE, “Cloud Migration Research: A Systematic Review”

[3] Bachle M, Kirchberg P. Ruby on Rails. IEEE Software, Vol. 24, Iss. 6, pp. 105 - 108

[4] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[5] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.