

LINK SHARING APPLICATION

Pallavi Acharya

4th year student, Dept. of CSE, RV College of Engineering Bangalore, India

Abstract - A web application is developed which is used to share the documents and/or link among a group of people who are subscribed to that topic. One of the oldest applications of the internet include the file sharing application. A user can upload a file at a space that is common and other users can download the file from there. The objective of this project is to create a web application that is user friendly and where users can not only create a topic but can also share their views on the topic along with the relevant links and/or documents among the subscribers of the topic.

Key Words: User interface, Grails framework, Apache groovy, Convention over configuration, Spring and Hibernate

1.INTRODUCTION

The purpose of this project is to develop an user-friendly interface to enable users to create a topic and share their views on the topic along with the relevant links and/or documents among the subscribers of the topic .One of the oldest applications of the internet include the file sharing application. A user can upload a file at a space that is common and other users can download the file from there. The main motivation of this project is to build well organized application that help us to share resources like document or link among the subscribers of the topic and another motivation is to learn about the tools and technologies required to build this project. The web Application is built on the grails framework and is written in Apache groovy language. To develop an user-friendly interface to enable users to create a topic and share their views on the topic along with the relevant links and/or documents among the subscribers of the topic. The objective of this project is to create a web application that is user friendly and where users can not only create a topic but can also share their views on the topic along with the relevant links and/or documents among the subscribers of the topic. The creator of the topic also has the ability to decide who can see the post and access the documents/links of their topic by setting the topic to private and not open for subscription..

The users can send invitations via mail to invite users to subscribe to a private topic. On the other hand public topics can be viewed, subscribed and shared by anyone using the application Interface provide the features like

rating the posts of other users , subscribing to topics, user authentication, users can also specify how serious they are towards a particular topic by setting the seriousness.

Section II covers the existing Literature on Link Sharing, Section III methodology, while Section IV discusses the results we achieved. Section V concludes.

2.LITERATURE REVIEW

Grails is based on Groovy, which provides an immediate productivity boost. Groovy's syntax is substantially more succinct than Java's, making it much easier to accomplish tasks in one line of Groovy code that would take multiple lines of Java code. In comparison to other web frameworks, Grails offers a lot of advantages. By the way, Grails is built on top of Spring MVC, and you can combine other components using Spring.

Database migrations and versioning ensure that your applications are always in sync with your database schema. This is a problem that affects the majority of Java web apps. Artefacts - which make it easy to create new controllers and components. No longer must you construct a controller, configure it, and place it in the appropriate location in your web project. Scaffolding also gives you with all of the necessary components to begin constructing the structure. Simpler validation (in comparison to Struts 1.x), for example, username provides two validation constraints for a username field: it must be of a certain length and not be blank. In the other Java web app frameworks, it's a little more difficult. Webflow (through Spring webflow) is built-in, making the creation of complex workflows much easier. Webflow is supported by Struts 2, but only through a plugin. SpringMVC also features webflow interceptors. Struts 2 is substantially better than Struts 1 in terms of data binding. SpringMVC is a fantastic option as well.

Grails is particularly good at making it simple to start a new project. It's as easy as running a command that builds the project structure and all of the folders required for the classes you'll be adding later. Model classes, controllers, services, and web pages can all be added with a comparable level of effort. The only thing you have to worry about is appropriately naming and positioning your items. Unlike Java, there is almost no boilerplate code that must be present simply because it is required. This is made feasible in part by the use of Spring and Hibernate,

two of Grails' pillars, as well as the concept of coding by convention.

Table-1: Grails benefits over other frameworks

	Grails	Rails	Spring MVC	Spring Boot
Targets JVM	X		X	X
Designed for Groovy	X			
Extensible via plugins	X	X		
Built on Spring	X		X	X
Integrated ORM	X	X		
Full Stack	X	X		
Convention over Configuration	X	X		

Grails comes with Apache Tomcat as a development server, which is required to operate the project. All we have to do now is execute the project in our IDE, and the server will be launched with our code installed. In addition, the GORM with Hibernate will create the database for us. We may either configure the JDBC connection properties to use an existing database or leave it as is to use an in memory instance. We can edit the code once the Grails server is up and running (it takes a little longer than a Spring MVC application), and the hot deploy mechanism will keep our debug session up to date. The entity classes are the only ones that can't be reloaded this way.

The simplicity of working with data was one of the first things that drew our attention to Grails. Reading data from a database is a procedure that must be repeated several times. And, in many cases, it is straightforward. As an example, obtaining one or more entities that fulfil a set of criteria and then aggregate them. Instead of using a static finder, why not use a dynamic finder? It's a method of querying data in which methods are constructed dynamically during runtime. It's as simple as following a naming convention.

3.METHODOLOGY

The implementation details of the developed web application are discussed in this section including the technologies, tools that are used to develop the application, system design, database architecture, features and functionalities developed. Grails is a web application framework that uses technologies such as spring (java technology) and hibernate (java technology) under a single interface. It also provides a complete development mode. It is based on the apache groovy. Apache groovy is an object-oriented programming language which is java compatible to the java syntax. It has features similar to languages like Ruby, Smalltalk and Python since it is a static as well as a dynamic language. It is a language that can be used for both scripting and programming for the Java platform. Groovy interoperates with java libraries and code seamlessly. All relational databases supported by hibernate are supported by grails too, it also supports other databases like MongoDB and Redis. Plugins: Many grails plugins are also used in the web Application these include data Tables plugin for displaying the data to the user which allows the user to choose how many rows appear on the screen and allows searching within the section based on field data. Rating widget plugin which allows users to rate posts and display it aesthetically on the screen. Spring Security plugin that is used to manage and control authentication and provide security to the application. Mail plugin is also used to allow sending mail from the application to send email invites to other users.

The Architecture of the system is shown in the figure. The architecture shown in the figure is slightly different from the typical architecture of n-tiers. The web application pages when loaded for the first time act similar to a normal web application but after interaction with the user when the user creates a request it is handled by the Ajax in the background (Asynchronously). Hence the side on the right of the architecture is used when a user interacts with a page and an Ajax call is made in the background for the user's request. The other side is of the architecture shows the requests or operations that cannot use Ajax and when a page load is required. In the architecture client UI is the web pages that will appear on the clients' browser. The presentation layer converts the outputs or result from the business logic and displays it to the end user in a readable format. The business logic layer handles all the logic related to a request for example the logic for searching a post based on the topic name, description or owner's name. The layer isolates the business logic layer from the other layers hence allowing user to be able to share and control the business logic. The business logic layer is between the presentation layer and the data access layer. It takes a client's request analyzes and processes the request, retrieves data from the data access layer and gives a suitable data response to the presentation layer. The data access layer allows application to access the data

from the database by executing Sql queries or stored procedures. All the data base connection details, methods to create a database connection and opening a database connection exist in this layer. It does not contain any business logic and does not manipulate any data from the database or vice versa. It is just an interface for the database. All the applications data is stored in the database tier. Data access layer retrieves data from the database that is oracle. In the Architecture Fig 11, whenever a user creates a new request from the server, Ajax engine receives a java Script call and forwards that request in the background to the web services without changing the display of the web page. The web services analyzes and processes the request and then retrieves the data from the database (Oracle in this case) manipulates the retrieved data if required converts it into JSON and sends it to the Ajax so that it could be displayed to the user after transforming into HTML. The benefit of this architecture is that every layer is isolated. So if a change is required it can easily be performed without disturbing the entire flow of the application.

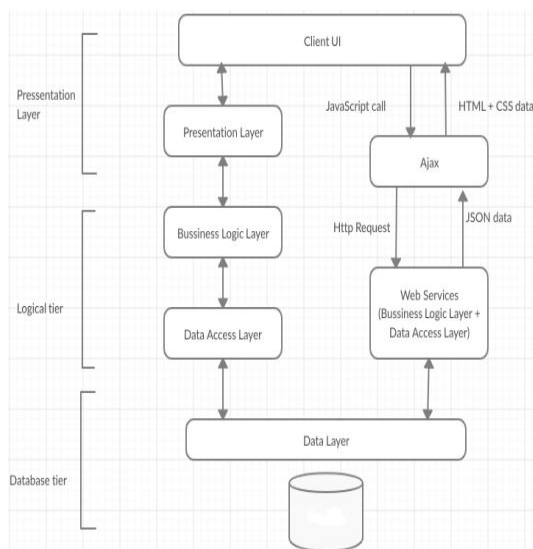


Fig-1: System Architecture

Class diagrams are the main building block in object-oriented modeling. A class diagram depicts the structure of a system by displaying the system's classes, their characteristics, actions (or methods), and interactions between objects. The class diagram of link sharing application is shown in the Fig-1.

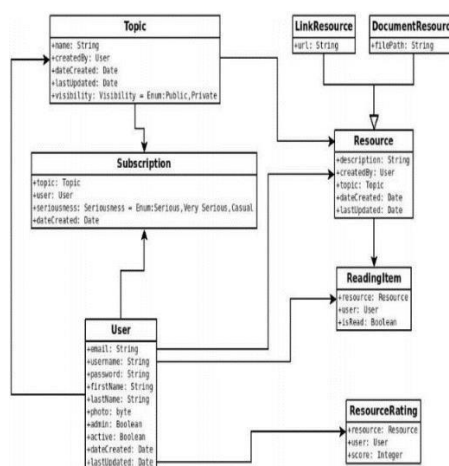


Fig-2: Class diagram of Link sharing application

4. RESULTS

The project's outcomes include a modernised Link sharing program and user interface. A user-friendly web application is constructed, in which users can not only create a subject but also share their thoughts on the issue, as well as pertinent links and/or documents, with the topic's subscribers. The entire application is written in the Apache Groovy programming language. H2 database is used to store user information. Non-functional requirements have been addressed in addition to functional requirements. The responsive user interface has been thoroughly tested to ensure that users have the greatest possible experience.

A web application is created that may be used to share documents and/or links between subject subscribers. Users can create topics and control the visibility of those subjects. The user can also specify how serious they are about the subject. A user can subscribe to an already existing public subject or a private subject by receiving an invitation from the topic's originator, subscriber, or admin.

Incorporating convention over configuration, the Grails framework eliminates much of the configuration effort typically required in building robust web applications. The efficiency and precision of Grails code means faster development with fewer bugs and less code. Interactive development is a hallmark of the Grails framework because teams can carve out functionality in short sprints, with checkpoints and testing along the way. As an open source framework, Grails can be broadly adopted and used with no vendor lock-in, contracts, license management, or run-time licensing fees. The Grails framework includes a development-time reloading agent that supports the dynamic reloading of code changes, thereby reducing the number of container restarts in the development environment.

The application can be used by nearly 1000 user of a organization to access the documents, upload documents and links. The web application works very smoothly because of the use of Ajax. The amount of time it takes to upload the documents depends on the internet speed. Grails application takes less time to upload the documents. 500KB files with internet speed of 1Mbps takes less than 3 seconds. 10MB files with speed of 1Mbps takes around 1.33 minutes to upload. The application developed using grails framework is very responsive and easy to use.

5. CONCLUSION

An interactive web application is developed on grails framework using the groovy language as per the requirements provided by the company. Oracle database is used for storing the application data. A web application is created that is user friendly and where users can not only create a topic but can also share their views on the topic along with the relevant links and/or documents among the subscribers of the topic. The creator of the topic also has the ability to decide who can see the post and access the documents/links of their topic by setting the topic to private and not open for subscription. The users can send invitations via mail to invite users to subscribe to a private topic. On the other hand public topics can be viewed, subscribed and shared by anyone using the application. The application also has many other features some of these include rating the posts of other users , subscribing to topics, user authentication, registration for new users, users can also specify how serious they are towards a particular topic by setting the seriousness, users can browse topics that are publicly available and can also search for posts related to topics that are either public or subscribed by the user or both, unread posts related to the subscribe topic shows up in the inbox and the user can also mark a post as read or unread . Application also uses technologies like Ajax for asynchronous data transfer from and to the server and JavaScript for client side validation and making the web pages more interactive.

ACKNOWLEDGMENT

The authors acknowledge the guidance and mentorship of Prof. Jyoti Shetty, Department of Computer Science and Engineering, RV College of Engineering.

6. REFERENCES

[1] P. Thung, C. Ng, S. Thung and S. Sulaiman, "Improving a Web Application Using Design Patterns: A Case Study" in International Symposium in Information Technology 2010, Malaysia:IEEE, Kuala Lumpur, Sept. 2020.

[2] S. M. Blackburn, "The DaCapo benchmarks: Java benchmarking development and analysis", In Conference on Object-Oriented Programming Systems Languages and Applications (OOPSLA), pages 169–190, 2016.

[3] J. de Halleux and N. Tillmann, "Moles: Tool-assisted environment isolation with closures", In International Conference on Objects, Models, Components, Patterns (TOOLS), pages 253–270, 2016.

[4] A. Gal et al, "Trace-based just-in-time type specialization for dynamic languages" In Conference on Programming Language Design and Implementation (PLDI), pages 465–478, 2009.

[5] E. Kiciman and B. Livshits, "Ajaxscope: a platform for remotely monitoring the client-side behavior of Web 2.0 applications", In Symposium on Operating Systems Principles (SOSP), pages 17–30, 2007.

[6] B. Livshits and E. Kiciman, "Doloto: code splitting for network-bound Web 2.0 applications", In Conference on Foundations of Software Engineering (FSE), pages 350–360, 2008.

[7] J. Mickens, J. Elson, and J. Howell. "Mugshot: Deterministic capture and replay for JavaScript applications" In Symposium on Networked Systems Design and Implementation (NSDI), 2010.

[8] K. Pattabiraman and B. Zorn. "DoDOM: Leveraging DOM invariants for Web 2.0 applications robustness testing", In International Symposium on Software Reliability Engineering (ISRE), 2010.

[9] P. Ratanaworabhan, B. Livshits, and B. Zorn. "JSMeter: Comparing the behavior of JavaScript benchmarks with real Web applications", In Conference on Web Application Development, 2010.

[10] G. Richards, C. Hammer, B. Burg, and J. Vitek, "The eval that men do: A largescale study of the use of eval in JavaScript applications", In European Conference on Object-Oriented Programming (ECOOP), 2011.

[11] G. Richards, S. Lebesne, B. Burg, and J. Vitek, "An analysis of the dynamic behavior of JavaScript programs", In Conference on Programming Language Design and Implementation (PLDI), pages 1–12, 2010. LINK SHARING APPLICATION Department of Computer Science and Engineering 2020-2021 Page 59

[12] K. Vikram, A. Prateek, and B. Livshits. "Ripley: automatically securing Web 2.0 applications through replicated execution", In Conference on Computer and Communications Security (CCS), pages 173–186, 2009.

[13] S. Floyd, V. Jacobson, "Link-sharing and resource management models", in IEEE/ACM Transactions on Networking' 04, 2015

[14] Bosi, C, "Developing Web Application Based on Grails Framework" in Computer Applications and Software, 2009, Issue 8.

[15] Hao, Z. & Guang-Xin, "Discussion of Domain Oriented Web Application Development Based on Grails" in Microcomputer Application, 2009.

[16] Mora-Murguía, L.P., Alor-Hernández, G., Olivares-Zepahua, B. A., Reyes-Hernández, L. A. & Chavez-Trejo, A. M,

“Best Practices from Web Development using Grails and Django”, Mexico City, Technological Trends in Computing, IPN,2014.

[17] Heiden, S, “The TXM Platform: Building Open-Source Textual Analysis Software Compatible with the TEI Encoding Scheme”, in 24th Pacific Asia Conference on Language, Information and Computation,2006.

[18] Hermawan, H. &Sarno, R. (2012), “Developing Distributed System With Service Resource Oriented Architecture”, In Conference on Programming Language Design and Implementation, Vol.10, No.2.

[19] Alessandro Rossini³, Adrian Rutle , “Groovy and Grails Meet Eclipse Modelling Framework” , 2017, Bergen University.

[20] PGovindRaj, “Enhancing Application Performance through GORM Optimizations” DOI:10.5120/9402-3880.

[21] GraemeRocher, Jeff Brown, “The definitive guide to grails” , In Conference on Computer and Communications Security, Second edition, 2008, pp.23-56, issue 9

[22] Pereira, M. & Martins, J. A, “aRDF: A plugin to expose RDFa semantic information using Grails” EATIS, 6th Euro American Conference, 2012, Issue 11.

[23] Waller, M.P, Dresselhaus, T. & Yang, J, “JACOB: An Grails framework for computational chemistry” Journal of Computational Chemistry, 2011, Vol. 34, Issue 16. LINK SHARING APPLICATION Department of Computer Science and Engineering 2020-2021 Page 60