Thesis Proposal

QUIZ APP- A new approach for better performance

-Harish Pindi

Committee Chair:

Dr. Robert D Mcilhenny

1. Objective:

The objective of this project is to develop a Quiz application, which provides a new approach of displaying ranking position based on their results after each exam through a Leaderboard. As a template of this project will be used to showcase my abilities of solving industry problems by applying knowledge acquired during my pursuit of the Master's degree.

2. Introduction

2.1 Industry Problem:

In general, the admin of an organization used to conduct several quizzes to their perspective candidates. The candidates might be able to attempt these exams and get to know only about their results. Also, for the future quizzes they might follow the same pattern.

The Main problem arises, the way candidate perceiving the result. Most of the methods just display the achieved result. The candidate might take the result as granted and would show no further improvements. The result of one exam should boost for the upcoming ones rather than degrading.

2.2 Proposed Solution:

The way the candidate knows about his ranking position, would accumulate chances of improvements in their perspective field. For instance, if an organization consists of thousands of people who are familiar with certain technologies. If there is a sudden occurrence of new technology which might be trending. The admin would like to know how often their candidates getting used to the new technology. The admin might be using this application to conduct quizzes. At the end of the exam, the user might be knowing result and their positions out of thousands. This position out of thousands would make them think of the situation and would seek for a better performance in the upcoming quiz exam.

2.3 Solution:

I came up with a web application, where the user can access the application through user login method and can attempt the exam. Since the exams are assigned only for a short period, the user can skip the questions and proceed further to avoid negative marking. After he attempts, he can get access to check his result through a leader board. Where he used to know about his position out of several people.

3. Technical Approach

3.1 Overview

QuizApp has been designed to have a tiered architecture. The real strength of QuizApp is in its robust and flexible data model. However, not all users want to have to know this data model inside and out. The API layer allows a developer to know Rest endpoints and read/save to them. This layer can be used in different channels like web, IOS, Android etc.,

3.2 Context

The backbone of QuizApp is the core API. This API has methods for all the basic functions like adding/Validating User, adding/updating an Exam, adding/updating Questions etc. These methods are provided in services.

The application is built on Eclipse IDE following the basic principles of MVC pattern. The basic idea of the application is reusability and hence the service and the UI layers are decoupled. All the DB interactions are exposed as rest endpoints and this rest endpoints can be used in all the channels irrespective of the domain and technology they incorporate.

3.3 Authorization and Authentication
QuizApp has a very granulated permit system. Every action is associated with a privilege. Admin can add the Exam, Questions and can see the Leaderboard. Whereas a user can only attempt an exam and view the Leaderboard.

3.4 Framework

Jersey Restful web services :

The Jersey usage gives a library to actualize Restful web services in a Java servlet holder. It is the reference usage of the JSR 311 detail. Jersey gives a servlet execution ,which examines predefined classes to recognize RESTful resources. The Jersey execution likewise gives a stage of client library to communicate with a RESTful web service.

There are many rest endpoints created in QuizApp using jersey web services framework, QuizApp uses POST and GET methods, Consumes and produces only Json data.

3.5 Angular JS :

AngularJS is a very powerful JavaScript Framework. It is generally utilized as a part of Single Page Application (SPA) projects. It expands HTML DOM with additional characteristics and makes it more receptive to client activities.

The QuizApp UI layer is completely written in Angular JS and this is a single page application and uses the services provided by QuizApp services module and renders the data.

3.6 Programming Languages and Tools:
3.6.1 J2EE7:

The server side programming language mainly used to implement business logic.

3.6.2 Maven**:**

Maven is an open source build tool customarily utilized as a part of Java and Java EE projects to arrange source documents, execute unit tests and gather appropriation artifacts. While Maven, works in Java projects and artifacts, for example. Ear and .War applications.

QuizApp is built with maven and all the artifacts required are assembled by maven and it also takes care of creating final a Web App (. war) file which can be deployed on the server.

3.6.3 Eclipse IDE (Neon):

Eclipse is an integrated development tool which is mainly used to develop Java applications.

3.6.4 MySQL (5.1):

QuizApp database resides in MySQL and data model contains a bunch of tables and views to store application data.

4. Schedule and Approach:

I followed Agile methodology, since it is an iterative approach to deliver the software in an incremental process. I used to deliver my project in bits rather than all once at a time. Each bits of the project took a duration of two weeks. Each bits of this project further includes with user functionalities so called user stories. It then required to prioritize and deliver this chunks during a period.

The entire Application is divided into two phases, one is to write the services and other is for UI. The entire application is divided into a bunch of stories and these stories are grouped to different sprints, ideally each sprint is for two weeks.

5. References:

[1] Web services:

L. Wang and J. Shen, "A Systematic Review of Bio-Inspired Service Concretization," in *IEEE Transactions on Services Computing*, vol. 10, no. 4, pp. 493-505, July-Aug. 1 2017.doi: 10.1109/TSC.2015.2501300.keywords: {Algorithm design and analysis;Heuristic algorithms; Optimization;Quality of service;Systematics;Taxonomy;Web services;Ant colony optimization;Web service composition;Web service concretization;genetic algorithm;particle swarm optimization;quality of service}, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7330016&isnumber=8002778

[2] API's:

W. Zhou, L. Li, M. Luo and W. Chou, "REST API Design Patterns for SDN Northbound API," *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, Victoria, BC, 2014, pp. 358-365.doi: 10.1109/WAINA.2014.153keywords: {application program interfaces;computer networks;hypermedia;service-oriented architecture;transport protocols;HTTP content negotiation mechanism;OpenStack;REST API design patterns;REST architectural style;RESTful networking protocol design;SDN northbound API;SOX;fixed resource URIs;generalized SDN controller;hypertext links;hypertext-driven approach;service oriented data networking;software-defined network;software-defined networking;Computer architecture;Firewalls (computing);Media;Ports (Computers);Protocols;Servers;XML;Controller;Hypertext Driven;Northbound API;OpenStack;Quantum;REST API;SDN}, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6844664&isnumber=6844560

[3] MVC architecture:

D. M. Selfa, M. Carrillo and M. Del Rocio Boone, "A Database and Web Application Based on MVC Architecture," 16th International Conference on Electronics, Communications and Computers (CONIELECOMP'06), 2006, pp. 48-48.doi:10.1109/CONIELECOMP.2006.6keywords: {MVC architecture;analysis and system design.;database systems;Application software;Computer architecture;Control systems;Databases;Independent component analysis;Logic;Programming;Service oriented architecture;Unified modeling language;User interfaces;MVC architecture;analysis and system design.;database systems},URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1604744&isnumber=33714

[4] **Jersey Restful web service:**

D. Kosmajac and V. Vujović, "Information systems security and security extension in Jersey RESTful framework," 2012 20th Telecommunications Forum (TELFOR), Belgrade, 2012, pp. 15561559.doi:10.1109/TELFOR.2012.6419518keywords: {information systems;security of data;service-oriented architecture;Jersey RESTful framework;RESTful service security;efficiency enhancement;information system security;security extension;service application security;service oriented technologies;software

solutions;subsystemdesign;subsystemsecurity;Electronicmail;Java;Security;Servers;Web services;XML;Informacionisistemi;Jersey;RESTful;servisi;sesije;sigurnost;tokeni},
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6419518&isnumber=6419130

[5] Maven Framework:

D. Garg, A. Singhal and A. Bansal, "A framework for testing web applications using action word based testing," 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, 2015, pp. 593-598. doi: 10.1109/NGCT.2015.7375190keywords: {Internet;automatic testing;program testing;software development management;software quality;Maven automation tool;Web application testing;action word based testing;automated testing;keyword based testing;keyword driven testing;software development life cycle;software quality;software testing;test case execution;test script;Automatic testing;Browsers;Manuals;Postal services;Reliability;Automated Testing;Keyword Driven Testing;MavenAutomationTool;SoftwareTesting},
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7375190&isnumber=7375067

[6] Angular JS:

URL: https://angularjs.org/

[7] JAVA:

C. L. Sabharwal, "Java, Java, Java," in *IEEE Potentials*, vol. 17, no. 3, pp. 33-37, Aug/Sep 1998. doi:10.1109/45.714612keywords: {Internet;object-oriented languages;parallel programming;Internet Web pages;Java platform;Java programming language;Java programs;Java source code;Solaris machine;Sun Microsystems;Web site;World Wide We;applets;application development tools;built-in capabilities;bytecode;client server development tools;client side;dynamic Web pages;homepages;intranet;networking applications;object oriented programming language;remote Web sites;virtual machine code;Computer languages;Costs;Internet;Java;Object oriented programming;Sun;Virtual machining;Web pages;Web sites;World Wide Web},
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=714612&isnumber=15498

[8] J2EE7:

URL: https://docs.oracle.com/javaee/7/tutorial/

[9] Eclipse IDE:

URL: https://eclipse.org/ide/