Proposal for MSE Capstone Project

Project Title:

Twiddler

Student Name: Yufan Xue

Rui Li

Faculty Advisor: Kenny Hunt

Date of submission: 07/02/2014

TWIDDLER

OBJECTIVE

This project will develop a web application named “Twiddler” to manage assessment activities for academic programs. The assessment activities are around hierarchically categorized ‘student learning outcomes’

The application will support multiple Universities (or other types of academically clients) and will run within Amazons web services using EC2 for the server, RDS for the database and S3 for document storage

BACKGRAOUND

STUDENT LEARNING OUTCOME

A student learning outcome (SLO) is just a short statement that describes a specific objective for some academic unit. Students who successfully complete the academic activities for that unit are expected to achieve the outcome described by the SLO. Consider the following examples:

* Students will be able to write a quicksort algorithm in Java.
* Students will be able to explain the political theories: Communism, Democracy, and Populism
* Students will be able to use an Autel diagnostics tool to identify engine errors for any supported vehicle type.

Faculty often define academic courses and programs by generating a list of SLO’s that characterize graduates of the course or program

SLO HEIRARCHY

A University might define a program as a collection of courses and each course might be defined as collection of SLOs. Graduates of the program are then characterized by the union of all course SLOs in the program. In addition, Universities often summarize a program by listing several high-level SLOs for the program. These high-level SLOs are more general and abstract than the course-level SLOs and are meant to summarize (not specify) the course-level SLOs. Consider the following examples of program-level SLOs:

* Students will be able to develop efficient and well organized software for a variety of computational problems
* Students will be able to explain historical political structures and how they have been shaped by cultural perspectives
* Students will be able to diagnose and repair a wide variety of mechanical engines.

SLOs are organized hierarchically and there should be no pre-defined ‘level’ or ‘types’ in hierarchy. While ‘program’ and ‘course’ are typical types, there might be others and Universities should be able to define their own Universities might, for example, define the following levels in their SLO hierarchy: ‘emphases’, ’competency’, ’skill’, or ‘concentration’

ASSESSMENT

The main reason that Universities define and use SLOs is to access how well the program trains students. The University might want to know if students are really able to ‘develop efficient and well organized software’ or ‘explain historical political structures’ when they graduates. As a result, students will be asked to demonstrate their ability with respect to an SLO. This is usually a quiz, exam, homework assignment or other thing. The results are then collected and recorded by the Faculty.

Twiddler must support the collection and recording of assessment data for each SLO in the system. These documents will (initially) have no required structured; however Universities will oven want to define a structure for consistent reporting.

CURRENT PROJECT

ROLES

There are three roles. These roles are listed below and their capabilities are briefly outlined. There should also be a ‘public’ or ‘non-authenticated’ user that is able to view all “public” documentation

1. An application administrator.

a. The Twiddler administrator operates on behalf of the company that sells Twiddler

b. The Twiddler administrator should be able to create, update, view and delete licenses.

c. A license is associated with a client (usually a University) and will allow for X users until an expiration date.

d. The Twiddler administrator should be able to create, update, view and delete clients

e. Each client will have an account administrator and some number of licensed users.

f. Each license will have a document that defines the licensing agreement (i.e. the contract)

2. An account administrator

a. The account administrator operates on behalf of a Twiddler customer. Typically a University.

b. The account administrator can only manage information related to the customer that they represent.

c. The account administrator should be able to create, update, view and delete their list of licensed users.

d. The account administrator should be able to create, update, view and delete groups of licensed users.

e. The account administrator should be able to enable, disable, and view any data that is associated with the account.

f. The account administrator should be able to create, update, view and delete the permissions settings for each group and for every licensed user.

3. A licensed user

a. The licensed user is anyone an account administrator awards a license to.

b. A licensed user should be able to reset their password. We will likely use Amazon SES for this purpose.

c. A licensed user should be able to create, update, view and delete outcomes (ones they own) and assessments.

d. A licensed user should be able to control who has access to each document they create (access includes read, write and delete privileges).

e. A licensed user should be able to create, update, view and delete assessment documents.

f. Some licensed users are ‘publishers’.

i. A publisher is able to ‘publish’ any document

ii. Only documents that are ‘published’ are part of the formal documentation record.

Published documents have a version, timestamp and publisher data.

iii. Published documents may change over time, but all versions of a published document are available within the system.

DOCUMENTS

Primary documents are SLOs, SLO groups, and assessment documents. The application will support any document type the user chooses to upload.

All information is searchable.

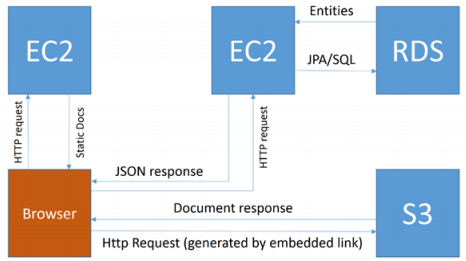
For users, search is done server-side.

For documents, search does not fire until three characters are typed into a search box. A ranked list of N items per category is returned to the user and further entries into the search field are processed on the client.

Access to documents is strictly controlled

HIGH-LEVEL ARCHITECTURE

Twiddler will be deployed on Amazon web services. The system will use an EC2 machine for deploying static HTML, JavaScript, CSS and any other supporting documents. The system will use a separate EC2 machine to supply all JSON data to the browser. RDS will be used to store user data, access control lists, assessment document meta-data, SLOs and SLO groups. S3 will be used to store assessment documentation.



CHANLEGES

The following are some of the challenges in this project

* Complex requirements
* Complex API provided by Amazon AWS as EC2, S3, RDS
* Search engine(search based on content of each web page content)

PROJECT SCHEDULE

The following schedule is proposed by the student, and is agreed by sponsors.

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **From** | **To** | **Credits** |
| Requirements analysis (user-stories) | Sep 02,2014 | Oct 01,2014 | 1 |
| Sprint 1 | Oct 02,2014 | Oct 21, 15,2014 | 1 |
| Sprint 2 | Oct 22,2014 | Nov 14, 2014 | 2 |
| Sprint 3 | Nov 15,2014 | Dec 07 ,2014 | 2 |
| Sprint 4 | Jan 01,2014 | Jan 14,2014 | 1 |
| Sprint 5 | Jan 15, 2015 | Feb 07, 2015 | 2 |
| Sprint 6 | Feb 08, 2015 | March 01, 2015 | 1 |
| Test | March 02, 2015 | March 31, 2015 | 1 |
| Project writeup | Apr 01, 2015 | May 10, 2015 | 1 |

Total 12