**Site Rater Documentation Version 2.0 (May 31,2014)**

Table of Contents

Table of Contents 1

Introduction: 3

PHP & HTML 3

index.php 3

dbconnect.php 3

admin.php 3

adminproc.php 4

admin\_pjt\_atft.php 4

admin\_pjt\_cate.php 4

admin\_pjt\_persona.php 4

admin\_pjt\_project.php 4

admin\_pjt\_scenario.php 4

admin\_pjt\_user.php 4

MySQL 5

The current database schema: 5

Tables and Data Dictionary 5

artifact 5

artifactTypes 6

category 6

languages 7

personaScenario 7

personae 7

project 7

projectArtifact 8

rating 8

ratingNarrative 8

scenario 8

scenarioCategory 9

userPersonae 9

userProfile 9

userRating 10

userRatingProgress 10

userScreenshot 11

Stored Procedures 11

addUser\_new 11

addArtifact 12

addUserRating 12

Work through Scenarios 12

Login 12

Add Project 13

Add Artifact 13

Add Scenarios 13

Add Persona 13

Add User 13

Add New Rating 13

Future Development Notification 13

# Introduction:

1. This document will illustrate the updated & extended part of TEDS Site Rater project, compared to original documentation.
2. Our development environment:
   1. MacOS Operation System.
   2. MAPP: Install the local environment on MacOS
      1. Apache 2.0, port 8888
      2. PHP 5.5.3
      3. MySQL 5.5.33, port 3306
   3. Database
      1. name: artifactRating2
      2. user: root
      3. password: dongh3d3long
3. At MySQL, we updated the values of tables to make them consist. We also updated procedures to make database updating working successfully.

# PHP & HTML

### index.php

This is the home page for admin interface. It's open to public, but only authorized user will be able to login and use it via pre-defined credentials. It embedded session\_inc.php for authentication. Updated PDO Connection to guarantee the connection with different versions of MySQL database.

### dbconnect.php

This file contains helper function db\_connect() which is a generic call to initialize the database connection. We changed the host, user and password parameter in order to develop extension on local machine.

$dbh = new PDO('mysql:host=localhost;port=3306;dbname=artifactRating2', 'root', 'root', array( // 'sunfl0w3r',

PDO::MYSQL\_ATTR\_USE\_BUFFERED\_QUERY => true,

PDO::ATTR\_PERSISTENT => true

));

### admin.php

This is the management site of administration. It contains several functions: overall management *Rating Process*, adding *Project*, adding *Artifacts*, adding *Persona*, adding *Scenario*, adding *Category*, adding *User*. Each function refer to php files to process the functions and store the added elements in database. Once clicking submit, all the filled information will be sent to *adminproc.php*.

### adminproc.php

This is the process file to process Rating Process functions: fetch information, call MySQL corresponding procedure, and update MySQL database.

1. For now, the languageID is assigned as 5, which represents *English* in database.
2. The artifactTypeID is assigned as 4, which represents WebPage.
3. Once adding persona, scenario, attifact, the process will update associative entities which are *ProjectArtifact*, *PersonaScenario*.
4. For Adding User, admin would need to fill all the information to generate new user.
5. Adding Categories by their id.

### admin\_pjt\_atft.php

This is the process file to process adding Artifacts. It is embedded in admin.html.

### admin\_pjt\_cate.php

This is the process file to process adding Categories description. It is embedded in admin.html

### admin\_pjt\_persona.php

This is the process file to process adding Personae. It is embedded in admin.html

### admin\_pjt\_project.php

This is the process file to process adding Projects. It is embedded in admin.html

### admin\_pjt\_scenario.php

This is the process file to process adding Scenario. It is embedded in admin.html. It will also process the association entity table.

### admin\_pjt\_user.php

This is the process file to process adding Scenario. It is embedded in admin.html. The user authority level is 1 by default.

# MySQL

The ER Diagram shows the schemas and relationships.

### The current database schema:

### erd.png

This section will document the updated tables and SPROCs (stored procedures) used in the MySQL database. And we also illustrate the data stored in tables.

### Tables and Data Dictionary

#### artifact

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| artifactID | int(11) | No |  | AUTO\_INCREMENT |
| artifactTitle | varchar(255) | No |  |  |
| artifactURL | varchar(255) | **No** | ***None*** |  |
| artifactTypeID | int(11) | No | **4** |  |
| artifactLanguage | int(11) | No | **5** |  |
| artifactDescription | varchar(150) | Yes | *NULL* |  |

Updated:

1. Change default Null of artifactURL to No, changed the default value to None
2. The default value of artifactTypeID is changed to 4, which represent Website
3. The default value of artifactLanguage is changed to 5, which represent English

#### artifactTypes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| artifactTypeID | int(11) | No |  | AUTO\_INCREMENT |
| artifactTypeTitle | varchar(100) | No |  |  |
| artifactTypeDescription | varchar(100) | Yes | *NULL* |  |

Explanation:

1. We already have 3 artifactTypes in the database, there are Website, Book, Audio File

#### category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| categoryID | int(11) | No |  | AUTO\_INCREMENT |
| categoryTitle | varchar(255) | Yes | *NULL* |  |
| parentCategoryID | int(11) | Yes | *NULL* |  |
| categoryLanguage | int(11) | No | **None** |  |
| categoryDescription | varchar(255) | Yes | *NULL* |  |

Explanation:

1. This is self associative table, the parentCategoryID refers to this table
2. The first 6 rows represent the categories: Ease of Use, Adaptability, Quality, Performance, Affection, and Noise Reduction. Their parenCategoryID is NULL
3. Rows from 7th is the sub-categories that under the one of first 6 rows, that means the parentCategoryID is one of 1 – 6.

Updated:

1. The default Language ID is not null. It must be the ID in the Language Table

#### languages

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| languageID | int(11) | No |  | AUTO\_INCREMENT |
| languageTitle | varchar(45) | No |  |  |

Explanation:

1. This is the table that almost other tables are linked to
2. There are 3 languages stored in database: English, Francais, Deutch
3. The ID for them are 5, 6, 7, the default is 5

#### personaScenario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| psID | int(11) | No |  | AUTO\_INCREMENT |
| personaID | int(11) | No |  |  |
| scenarioID | int(11) | No |  |  |

Explanation:

1. This is the associative entity that associate persona and scenario together by their id

#### personae

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| personaeID | int(11) | No |  | AUTO\_INCREMENT |
| personaTitle | varchar(45) | No |  |  |
| personaDescription | varchar(255) | Yes | *NULL* |  |
| personaLanguage | int(11) | No |  |  |

Explanation:

1. Currently, all language is 5, representing English
2. There are 4 personae for development testing

#### project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Type | Null | Default | Comments |
| projectID | int(11) | No |  | AUTO\_INCREMENT |
| projectTitle | varchar(45) | No |  |  |
| projectDescription | varchar(150) | Yes | **NULL** |  |
| projectLanguageID | int(11) | No |  |  |

Explanation:

1. There are 3 projects for development testing

#### projectArtifact

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| projectArtifactID | int(11) | No |  | AUTO\_INCREMENT |
| projectID | int(11) | No |  |  |
| artifactID | int(11) | No |  |  |
| isAnchor | bit(1) | Yes | *NULL* |  |

Explanation:

1. This is the associative entity that associates project and artifact.

#### rating

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| ratingID | int(11) | No |  | AUTO\_INCREMENT |
| ratingValue | int(11) | No |  |  |
| projectID | int(11) | No |  |  |

Explanation:

1. This table stores the rating value
2. It associate to project table.

#### ratingNarrative

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| narrativeID | int(11) | No |  | AUTO\_INCREMENT |
| userRatingID | int(11) | No |  |  |
| userNarrative | varchar(2000) | Yes | *NULL* |  |
| userScreenShot1 | varchar(255) | Yes | *NULL* |  |
| userScreenshot2 | varchar(255) | Yes | *NULL* |  |

Explanation:

1. This table store the narrative values of rating

#### scenario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| scenarioID | int(11) | No |  |  |
| scenarioTitle | varchar(45) | No |  |  |
| scenarioDescription | varchar(255) | Yes | *NULL* |  |
| scenarioLanguageID | int(11) | No |  |  |

Explanation:

1. This table stores the scenarios for TEDS evaluation.

#### scenarioCategory

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| SC ID | int(11) | No |  | AUTO\_INCREMENT |
| scenarioID | int(11) | No |  |  |
| categoryID | int(11) | No |  |  |

Explanation:

1. It is the associative entity that associate scenario and category

#### userPersonae

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| userPersonaeID | int(11) | No |  | AUTO\_INCREMENT |
| userID | int(11) | No |  |  |
| personaeID | int(11) | No |  |  |

Explanation:

1. This is the associative entity that associate user and personae

#### userProfile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| userID | int(11) | No |  | AUTO\_INCREMENT |
| email | varchar(100) | No |  |  |
| firstName | varchar(45) | No |  |  |
| lastName | varchar(45) | No |  |  |
| preferredLanguage | int(11) | Yes | ***5*** |  |
| passwordValue | varchar(50) | Yes | *Null* |  |
| **AuthorityLevel** | **int(11)** | **no** | **1** |  |

Explanation:

1. This is the User table to store user information

Updated:

1. The preferredLanguage is changed to 5, which represent English
2. Added new column AuthorityLevel, the default is 1, which represent normal user. 2 means admin.

#### userRating

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| **id** | int(11) | No |  | AUTO\_INCREMENT |
| **userRatingProcessID** | **int(11)** |  |  |  |
| userID | int(11) | No |  |  |
| ratingID | int(11) | No |  |  |
| userPersonaeID | int(11) | No |  |  |
| personaScenarioID | int(11) | No |  |  |
| scenarioCategoryID | int(11) | No |  |  |
| artifactID | int(11) | No |  |  |

Explanation:

1. This is an core associative entity

Updated:

1. The table id is changed to **id** from userRatingID
2. Add a new column which is userRatingProcessID, this is foreign key to userRatingProcess table.

#### userRatingProgress

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| userRatingProgressID | int(11) | No |  | AUTO\_INCREMENT |
| userID | int(11) | No |  |  |
| **personaID** | **int(11)** | No |  |  |
| **scenarioID** | **int(11)** | No |  |  |
| projectArtifactID | int(11) | No |  |  |
| isComplete | bit(1) | Yes | *NULL* |  |
| completionDate | datetime | Yes | *NULL* |  |

Explanation:

1. This is the table that represents the progress of rating.

Updated:

1. Add new column which is personaID, the foreign key to Persona Table
2. Add new column which is scenarioID, the foreign key to Scenario Table

#### userScreenshot

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| userScreenshotID | int(11) | No |  |  |
| userRatingID | int(11) | No |  |  |
| screenshot1 | varchar(255) | Yes | *NULL* |  |
| screenshot2 | varchar(255) | Yes | *NULL* |  |

Explanation:

1. This table store links of screenshot path

### Stored Procedures

The Site Rater system uses a number of stored procedures; these may be found in the proc table of the default mysql database during an install. We will illustrate updated PROCs.

#### addUser\_new

(in email varchar(100), in firstName varchar(45), in lastName varchar(45), in passwordValue varchar(50), in languageID INT, out userID INT)  
BEGIN  
START TRANSACTION;  
INSERT INTO userProfile  
(email, firstName, lastName, passwordValue, preferredLanguage)  
VALUES  
(email, firstName, lastName, passwordValue, languageID);  
COMMIT;  
SELECT last\_insert\_id() INTO userID;  
END$$

Explanation:

1. This stored procedure was replaced to addUser

#### addArtifact

(in artifactTitle varchar(45), in artifactURL varchar(255), in artifactTypeID INT, inartifactLanguageID int, out newArtifactID int)  
BEGIN  
START TRANSACTION;  
INSERT INTO artifact  
(artifactTitle, artifactURL, artifactTypeID, artifactLanguage)  
VALUES  
(artifactTitle, artifactURL, artifactTypeID, artifactLanguageID);  
COMMIT;  
SELECT last\_insert\_id() INTO newArtifactID;  
END$$

Explanation:

1.Delete DECLARE artifactURL varchar(255) default null;

#### addUserRating

(in userID INT, in ratingID INT, in userPersonaID INT, in personaScenarioID INT, in scenarioCategoryID INT, in artifactID INT, out newRatingID int, in userRatingProcessID int)  
BEGIN  
START TRANSACTION;  
INSERT INTO userRating  
(userID, ratingID, userPersonaID, personaScenarioID, scenarioCategoryID, artifactID, userRatingProcessID)  
VALUES  
(userID, ratingID, userPersonaID, personaScenarioID, scenarioCategoryID, artifactID, userRatingProcessID);  
ON DUPLICATE KEY UPDATE  
userRatingID = LAST\_INSERT\_ID(userRatingProcessID),  
ratingID = VALUES (ratingID);  
COMMIT;  
SELECT userRatingProcessID into newRatingID;  
END$$

# Work through Scenarios

### Login

Both raters and administrators can login with their user email and password. The administrator has higher authority level (level 2 in database), and he/she can add *project*, *artifact*, *scenarios,* personas, users, and add new ratings for users. Users can login for starting or continuing rating.

### Add Project

Administrator can add new project with information of project name, project description, and project language (default is English)

### Add Artifact

Administrator can add new rating artifacts for existing project, with artifact information of title, url, and project name.

### Add Scenarios

Administrator can add new scenarios with scenario title and description

### Add Persona

Administrator can add new personas with persona title and description

### Add User

Administrator can add new users for existing project, with user information of email, first name, last name, preferred language (default is English), and related personas.

### Add New Rating

Administrator can select project along with its artifact, persona along with its related scenario and user, to create new rating progress. The rating progress table can shows all rating progresses with their status. Admin can send rating initiation to users with artifact link.

# Future Development Notification

For the database development, we suggest adding new tables to contain more comments & screenshots of one project rating. And, the current stored procedures were designed to consisted with current MySQL and PHP version; we would recommend keep updating them for future MySQL and PHP version.

One of the most important problems that we need to fix in the future is to making the iframe being able to handle page re-direction within itself, so that raters can make ratings within the same context of the same site. After some research and double check, we found that there are only a few sites that will force the parent site to redirect to another url. This is no longer a universal problem. But in case of this kind of problem happening again, we think the follow potential solution may be helpful.

Here is an article regarding the detailed processes of how it works: <http://stackoverflow.com/questions/1794974/using-js-how-can-i-stop-child-iframes-from-redirecting-or-at-least-prompt-users?lq=1>

Basically, we need to write a javascript function to detect the page redirection event and check redirect destination, if it is different from the current parent page url, use javascript to force the page remain in the same domain and change the iframe src attribute.

Besides all the UI modifications, there are also some back-end and database changes that we have to make.