

Documentation of Submission Box, Version 3

(Developer Documentation)

4 May 2015

Bindu Madhavi Wuyyuru
Cameron Schermerhorn
Deepika patlolla
Riam Nassir
Reem Alshammari

Table of Contents



[Developer Documentation](#)

[Architecture Overview](#)

[Webpage Architecture](#)

[Webpage Flow](#)

[Successful Login Attempt and File Upload](#)

[Unsuccessful Login Attempt](#)

[Forgot Password](#)

[Webpage Files](#)

[Authentication.html](#)

[Authentication.php](#)

[assignments.php](#)

[connect.php](#)

[deleteDir.php](#)

[dropzone.js](#)

[forgot-password.php](#)

[loginError.html](#)

[logout.php](#)

[mkdir.php](#)

[submit.php](#)

[Database Architecture](#)

[Database Table Design](#)

[Database Scripts](#)

[CreateDatabase.sql](#)

[editEntry \(PERL\)](#)

[edit_database \(PERL\)](#)

[man_editEntry \(PERL\)](#)

[man_newEntry \(PERL\)](#)

[man_removeEntry \(PERL\)](#)

[newEntry \(PERL\)](#)

[removeEntry \(PERL\)](#)

[showAssignments \(PERL\)](#)

Developer Documentation

This section will describe the design and implementation used for revision three of Submission Box. This will serve as a starting point for future iterations so that developers can gain an insight into what the original ideas were.

Architecture Overview

Keeping the first two iterations of Submission Box in mind, the third iteration has not made any significant changes to existing architecture. Changes in this sense can be considered additions, meaning that all existing functionality has been preserved alongside new features.

The main portions of Submission Box revolve around the end user through a web interface. While the user could be using several different browsers, the architecture for producing the experiences are the same. Submission Box uses html, php, and javascript hosted on an apache server to deliver the viewable webpage. For populating data and allowing access, Submission Box uses a mysql database hosted on an external server. Combining both the database and webpage architecture pieces together delivers the experience to the end user. To control the environment, the client issues commands to the mysql database through an ssh client. Commands are parsed in the perl programming language, and passed to the mysql database.

Webpage Architecture

This section will expand on the webpage architecture discussed in the architecture overview section above. The website is hosted on an apache server, following the namespace of sb.teresco.org. For testing and development purposes, the production site used was sb3.teresco.org. The following examples will follow a typical path that will be taken when visiting Submission Box. Each successive path is a forward to a new page relating to Submission Box.

All pages are considered to follow the web address of SubmissionBox (i.e. www.sb.teresco.org/Authentication.html).

Webpage Flow

Successful Login Attempt and File Upload

Authentication.html >> Authentication.php >> Submit.php >> assignments.php >> mkdir.php >> logout.php

Unsuccessful Login Attempt

Authentication.html >> Authentication.php >> loginError.html

Forgot Password

Authentication.html >> forgot-password.php >> connect.php

The html files generally serve as the basic viewable landing pages. Cases where the user has to login, logout, or re authenticate due to bad password all cause a simple html redirect. During authentication, user input from these pages is passed to a php file. Files like Authentication.php, Submit.php, Assignments.php, and Mkdir.php all serve as a connector to the mysql database.

User input, say username and password, as passed to a php file to authenticate against the mysql database. If passwords do not match, or the user does not exist, then access to Submission Box is denied. When submitting files, the same principle of passing data to a php file is used. Information, such as the course name and assignment name, are passed from the Submit.php page to the Mkdir.php page. If the file size is acceptable, and the file matches the extension parameter, then this data is passed into the php file to upload to the server.

Mkdir.php serves as the link between file server and SubmissionBox. If successful, the php file will issue commands to the file server to create a new directory under its respective course, assignment, and user. It then passes the file into this new directory and uploads it for submission. A notification email is then sent to both user and client for confirmation.

Webpage Files

The following are descriptions of what each web page file does.

Authentication.html

First page seen by the end user. All users must authenticate their credentials through the input fields provided, and then hit submit. User can choose to reset password by using the forgot password link. Information is passed to Authentication.php to check against database. Theme and styling has been changed to use an external stylesheet, and whitespace has been modified to follow a newer flow.

Authentication.php

Information from Authentication.html is passed here. Username and user password will be authenticated against the Student table used for SubmissionBox. Passwords are obscured using MD5.

assignments.php

Used to build entities seen in Submit.php. This file will pull courses and assignments for a user that is currently enrolled in a course. No changes have been made to this file, aside from cleaning up spacing and indentation from old iterations.

connect.php

File is used during reset password functionality. This serves as a direct connection to the database hosted for SubmissionBox.

deleteDir.php

This file serves as a method to delete directories on the file server for Submission Box. Files are unlinked and the count for folder

submission is decremented by one. Not currently implemented or used.

dropzone.js

This javascript file serves as the direct base for the Dropzone implementation, which essentially is the drag and drop functionality that can be used for Submission Box. Removing this file will break functionality, causing errors to occur on web page viewing.

Options specifically for Submission Box can be found by searching for “my-awesome-dropzone” within this file. Main points here are that files will not be uploaded until the submit button is hit. Otherwise, files will immediately be processed and uploaded to the server, regardless of errors or legitimate entries. File is used within the Submit.php file.

forgot-password.php

File exists to implement the forgot password feature. User will be forwarded here from Authentication.html. Simple layout, the user will enter their username and hit submit. The information is collected and checked against the database used for Submission Box. If the username exists, an email is sent to the user. A pre-determined password is generated which is randomly built from a string of alphabetic characters, but uppercase and lowercase. A size of 8 characters is used during generation.

loginError.html

The user is forwarded here when an incorrect username or password is detected. Same functionality as Authentication.html, except an error message is displayed when the user reaches this page.

logout.php

User may hit log out at anytime to terminate the session. User is forwarded here, and then to a page where they can login again.

mkdir.php

Used during file submission. This file will create the directories needed on the file server used by Submission Box, and then place the uploaded file there. Error checking is done prior to file uploads. If an error exists, an error message is shown to the user. If file is legitimate to upload, a confirmation is shown to the user and an email is sent to both the client and user.

submit.php

The user is forwarded here once their credentials are authenticated. Courses and assignments are populated from the Assignments.php file. If a user is enrolled in only one course, the functionality was carried over so that the course is displayed immediately. If not, the user must choose the course manually.

Database Architecture

Submission Box uses a MySQL database for storing user information and class information. Edits and changes to this database can only be made by the client (unless a user is resetting their password) through an ssh

client. The scripts needed to modify the database are located in the /scripts folder within the Submission Box directory.

Database Table Design

```
drop table Student;
drop table Course;
drop table Enrollment;
drop table Assignment;
drop table Submission;

create table Student(
  StudentID varchar(20) not null,
  Password varchar(32) not null,
  primary key (StudentID));

create table Course(
  CourseID varchar(20) not null,
  primary key (CourseID));

create table Enrollment(
  StudentID varchar(20) not null,
  CourseID varchar(20) not null,
  foreign key (StudentID) references Student (StudentID),
  foreign key (CourseID) references Course (CourseID));

create table Assignment(
  AssignmentID int not null auto_increment,
  gFlag char(1) default 0,
  Name varchar(30) not null,
  CourseID varchar(20) not null,
  DueDate date not null,
  primary key (AssignmentID),
  foreign key (CourseID) references Course (CourseID));

create table Submission(
  StudentID int not null,
  AssignmentID int not null,
  SubmissionDate DATETIME not null,
  Grade DECIMAL(5,2),
  Comments varchar(4000),
  PRIMARY KEY (StudentID, AssignmentID),
  FOREIGN KEY (StudentID) REFERENCES Student(StudentID),
  FOREIGN KEY (AssignmentID) REFERENCES
Assignment(AssignmentID));

alter table Assignment add FileTypes VARCHAR(512);
```

Database Scripts

All database scripts reside in the /scripts directory within the Submission Box directory. These can only be issued by the client, or those who have strict permission to do so.

For perl files, if permissions are set up properly, the client can simply enter the perl filename to execute functionality. If not, the command that should be used is "perl perl_filename."

CreateDatabase.sql

Initial build of database; sets up tables necessary for basic SubmissionBox functionality. See Database Table Design section to see full table builds.

edit Entry (PERL)

This perl file is designated to edit existing data in the mysql database. If proper input is given as arguments, the file will effectively modify the data in database the respective database table. Input is parsed with perl using database integrations. Query is executed to parse and pass information to specified database. Improper arguments result in an error. See man_editEntry for proper usage.

edit_database (PERL)

Handles all functions from edit Entry, removeEntry, and new Entry. All mentioned perl files forward to edit_database to complete selected function.

man_editEntry (PERL)

The manual for how to execute the perl edit Entry file. A table of valid commands is shown to the user, along with proper input needed for each command.

man_newEntry (PERL)

The manual for how to execute the perl newEntry file. A table of valid commands is shown to the user, along with proper input needed for each command.

man_removeEntry (PERL)

The manual for how to execute the perl removeEntry file. A table of valid commands is shown to the user, along with proper input needed for each command.

newEntry (PERL)

This perl file is designated to enter new data into the mysql database. If proper input is given as arguments, the file will effectively insert the data into the database. Input is parsed with perl using database integrations. Query is executed to parse and pass information to specified database. Improper arguments result in an error. See man_newEntry for proper usage.

removeEntry (PERL)

This perl file is designated to remove existing data from the mysql database. If proper input is given as arguments, the file will effectively delete the data from the database. Input is parsed with perl using database integrations. Query is executed to parse and pass information to specified database. Improper arguments result in an error. See man_removeEntry for proper usage.

show Assignments (PERL)

Lists a current list of all assignments hosted by the mysql database table "Assignments." A list of the following is shown to the client: assignment id, assignment name, and course name. All data is respective to their own entity.