CCS Team

Heritage college  Development Projects II course

Systems test plan

Last Updated: March 24th, 2020

# Systems Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case # | Name | Actual Results | Pass/Fail |
| ChallengeControllerTests | | | |
|  | IndexGet\_ReturnsIndexView\_UserIsStudent\_UserIsAuthorized | Returns the index view for challenge | PASS |
|  | IndexGet\_ReturnsAccessDenied\_UserIsTeacher\_UserIsAuthorized | Returns access denied if teacher goes to challenge index page. | PASS |
|  | CreateGet\_ReturnsCreateView\_UserIsTeacher\_UserIsAuthorized | Returns the create challenge page for the teacher | PASS |
|  | CreateGet\_ReturnsAccessDenied\_UserIsStudent\_UserIsAuthorized | Returns access denied for create challenge page for the student | PASS |
|  | EditGet\_ReturnsEditView\_UserIsTeacher\_UserIsAuthorized | Returns edit challenge page for teacher | PASS |
|  | EditGet\_ReturnsAccessDenied\_UserIsStudent\_UserIsAuthorized | Returns access denied for the student | PASS |
|  | ChangeActive\_NonExistent\_ReturnsErrorView\_UserIsTeacher\_UserIsAuthorized | Returns an error view for a nonexistent challenge if the teacher tries to change the active state | PASS |
|  | Delete\_ReturnsTeacherIndex\_DeleteIsNotSuccessful\_UserIsTeacher\_UserIsAuthorized | Returns the edit challenge page when a delete is unsuccessful | PASS |
| LoginControllerTests | | | |
|  | IndexGet\_ReturnsErrorView\_UserIsNotAuthorized | Returns an error view if no one is logged in for the session | PASS |
|  | IndexGet\_ReturnsStudentIndex\_UserIsStudent\_UserAuthorized | Returns the student home page if a student is logged in | PASS |
|  | IndexGet\_ReturnsTeacherIndex\_UserIsTeacher\_UserAuthorized | Returns teacher home page if a teacher is logged in | PASS |
|  | IndexPost\_ReturnsLoginIndex\_InvalidCredentials | Returns the login page if the credentials are invalid. | PASS |
|  | IndexPost\_ReturnsTeacherIndex\_UserIsTeacher\_ValidCredentials | Returns the teacher home page if the credentials are of a real teacher credentials in AMS | PASS |
|  | IndexPost\_ReturnsStudentIndex\_UserIsStudent\_ValidCredentials | Returns the student home page if the credentials are of a real student credentials in AMS | PASS |
|  | Logout\_ReturnIndex | Returns the login page if someone logs out of their session | PASS |
| TeacherControllerTests | | | |
|  | IndexGet\_ReturnsLoginIndex\_UserIsNotAuthorized | Returns the login page if the user logged in is not authorized | PASS |
|  | IndexGet\_ReturnsStudentIndex\_UserIsStudent\_UserIsAuthorized | Returns the student home page if a student is logged in | PASS |
|  | IndexGet\_ReturnsTeacherIndex\_UserIsTeacher\_UserIsAuthorized | Returns teacher home page if a teacher is logged in | PASS |
|  | FilterChallengeTitlePost\_ReturnsIndexWithTeacher\_UserIsTeacher\_UserIsAuthorized | Checking when you search a challenge on the teacher home page returns the correct view and model | PASS |
| ChallengeModelTests | | | |
|  | SortingChallengesByDifficultyDecending\_ReturnsCorrectOrder | Returns all the teacher challenges when the order of difficulty is changed. | PASS |
|  | SetDifficultyLevelNumber\_ReturnsCorrectOrder | If the difficulty level string is equivalent to the number then this test case should pass | PASS |
|  | AddChallenge\_AddChallengeLanguage\_AddsSuccessfully | Checks if you can add a challenge language to a challenge | FAIL |
|  | GetStudentsAttemptingChallenge\_ReturnsNumberOfStudentsAttemptingChallenge | Returns the number of students attempting a challenge | PASS |
|  | GetStudentsAttemptingChallenge\_ChallengeDoesNotExist\_ThrowsExeption | Returns an exception if you a trying to get the number of students attempting a challenge if the challenge doesn’t exist | PASS |
|  | GetLastTimeChallengeWasAttempted\_ChallengeHasBeenAttempted\_ReturnsCorrectLastTime | Returns true if the last time a challenge was attempted is correct and the challenge in general was attempted by a student. | PASS |
|  | GetLastTimeChallengeWasAttempted\_ChallengeHasNotBeenAttempted\_ReturnsErrorMessage | Returns an error when trying to get the last time a challenge was attempted from a challenge that has not been attempted by a student. | PASS |
|  | GetLastTimeChallengeWasAttempted\_ChallengeDoesNotExist\_ThrowsException | Returns an error when trying to get the last time a challenge was attempted from a challenge that doesn’t exist. | PASS |
|  | GetChallengeTitle\_ReturnsChallengeTitle | Returns true if the challenge title that is being retrieved is correct | PASS |
|  | GetChallengeTitle\_ChallengeDoesNotExist\_ThrowsException | Returns an exception if you try to get a challenge title from a challenge that doesn’t exist. | PASS |
|  | GetChallengeLanguageNameList\_ReturnsChallengeList | Returns all the languages the challenge is created in | PASS |
|  | DeleteChallenge\_ReturnsTrue | Returns true if delete challenge was successful | PASS |
|  | DeleteChallenge\_ChallengeDoesNotExist\_ReturnsFalse | Returns false if you are trying to delete a challenge that doesn’t exist. | PASS |
|  | ToggleChallenge\_TurnActiveOff | Returns false if the challenge active state was turned off | PASS |
|  | ToggleChallenge\_TurnActiveOn | Returns true if the challenge active state was turned on | PASS |
|  | GetChallenge\_ChallengeExists | Checks if a certain challenge exists. If it does then is returns not null | PASS |
|  | GetChallenge\_ChallengeDoesntExist | Checks if a certain challenge does not exist. If it doesn’t then is returns null | PASS |
|  | EditChallenge\_ChallengeUpdates | Returns true if updating the challenge works | PASS |
|  | EditChallenge\_ChallengeDoesntUpdate | Returns an error when you try to update a challenge that doesn’t exist. | PASS |
|  | SanitizeChallenge\_Caught | if the challenge has been sanitized and it returns true | PASS |
|  | GetDifficultiesDropdown\_DifficultiesComesBack | Return if the difficulty levels | PASS |
|  | GetChallengeParameters\_TestCaseDoesntExist | Returns true if the number of challenge parameters is 0 when there are no test cases | PASS |
|  | GetChallengeParameters\_TestCaseExist\_ReturnsParameter | Returns true if the number of parameters for the challenge is correct when there are test cases | PASS |
|  | AllChallenges\_ShowsActive | Returns true if the number of challenges for the student are active | PASS |
|  | SortChallenges\_SortAscendingDifficultyLevel | Returns true if the challenges are in ascending order | PASS |
|  | SortChallenges\_SortDescendingDifficultyLevel | Returns true if the challenges are in descending order | PASS |
|  | SortTeacherChallenge\_TeacherDoesntExist | Returns an error if the challenges are being sorted by a nonexistent teacher | PASS |
|  | SortTeacherChallenge\_TeacherExists | Returns true if the challenges are being sorted by a teacher who exists | PASS |
|  | SortChallengesByDifficultyLevel\_SortIsCorrect | Returns true if the challenges sort is correct | PASS |
| TeacherModelTests | | | |
|  | GetTeacher\_TeacherExists | Returns true if you get a teacher from the database that exists. | PASS |
|  | GetTeacher\_TeacherDoesntExist | Returns an error if a teacher doesn’t exist in the database | PASS |
|  | AddTeacher\_Success | Returns true if the teacher was successfully added | PASS |
|  | AddTeacher\_Fail | Returns an error occurred when adding a new teacher | PASS |
|  | FilterChallengeTitle\_TeacherDoesNotExist | Returns false if teacher doesn’t exist and they try to filer the challenges | PASS |
|  | FilterChallengeTitle\_ChallengeTitleIsEmpty | Returns true if entering an empty string to filer the challenges is equal to all the challenges | PASS |
|  | FilterChallengeTitle\_ReturnsRightChallenge | Returns true if the filtered challenges are the correct count | PASS |
|  | FilterChallengeTitle\_ReturnsNoChallenge | Returns true if what is being filtered for the challenges does not exist | PASS |