

Bug Report

Data Writer - Jason Querubin:

Bug Name:	Write One Course
Description:	DataWriter has failed when retrieving an the difficulty of a course due to a null pointer exception from getCourses() method
Test Case:	testWritingOneCourse
Location of error:	DataWriter

Bug Name:	Write Multiple Courses
Description:	DataWriter has failed when retrieving an the difficulty of a course due to a null pointer exception from getCourses() method
Test Case:	testWritingMultipleCourses
Location of error:	DataWriter

Bug Name:	Null User Case
Description:	Test fails because of conversion error when attempting to convert a null value to a UUID type
Test Case:	testWritingNullUser
Location of error:	DataWriter; getUserJSON

Bug Name:	Null Course Case
Description:	Test fails due to undefined error occurring with constructor full of null parameters.
Test Case:	testWritingNullCourse
Location of error:	DataWriter; Course

UserList - Rhaynie Bongiorno:

Bug Name:	Email/Username Swap
-----------	---------------------

Description:	In the “addUser” method, the parameters for the method do not line up with the Student, Teacher, or Admin constructors, causing the username and email to not line up properly in the UserList.
Test Case:	addUserStandard test
Location of error:	UserList class, addUser method

Bug Name:	Blank Account
Description:	There seems to be no protections against having a “blank” user, or user with only spaces from being created.
Test Case:	addUserSpaces test
Location of error:	UserList class, addUser method

Bug Name:	Account with invalid password
Description:	The addUser Method does not check for whether or not a password meets the password requirements before creating a user.
Test Case:	addUserInvalidPass test
Location of error:	UserList class, addUser method

Bug Name:	Account with invalid email
Description:	The addUser Method does not check for whether or not an email is valid (if it contains an @address or not).
Test Case:	addUserInvalidEmail test
Location of error:	UserList class, addUser method

Bug Name:	deleteUser does not remove users
Description:	The deleteUser method only executes if the hasUser method returns false. This means that if the user exists, the code does not execute and the user is not removed.
Test Case:	removeUserFirstItem, removeUserLastItem
Location of error:	UserList class, deleteUser method

Comment- China Riley:

Bug Name:	Comment hasReplies
Description:	hasReplies method return a value of true when the value should return false.
Test Case:	hasReplies
Location of error:	Comment class hasReplies method, line 62

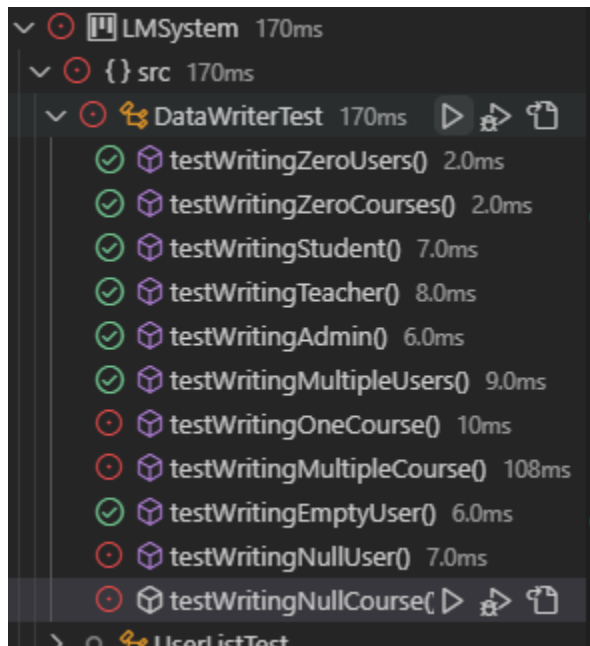
Quiz - Will Brown:

Bug Name:	Quiz getPotentialAnswers
Description:	getPotentialAnswers method returns location of potential answers, not individual strings of potential answers
Test Case:	hasPotentialAnswers
Location of error:	Quiz class, getPotentialAnswers method

Report:

Take a screenshot showing your test cases, and if they passed or failed.

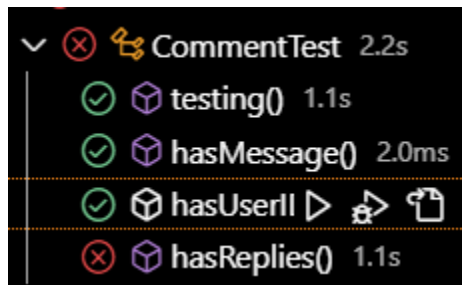
Data Writer - Jason Querubin:



UserList - Rhaynie Bongiorno:

✓	✗	LMSystem	136ms	
✓	✗	{ } src	136ms	
✓	✗	UserListTest	136ms	
✓	✓	testing()	0.0ms	
✓	✓	hasUserUUID()	2.0ms	
✓	✓	hasUserUsername()	0.0ms	
✓	✓	hasUserFirstItem()	0.0ms	
✓	✓	hasUserLastItem()	1.0ms	
✓	✓	hasUserInvalid()	0.0ms	
✓	✓	hasUserNull()	2.0ms	
✓	✓	hasUserEmpty()	0.0ms	
✓	✓	getUserUUID()	0.0ms	
✓	✓	getUserUsername()	0.0ms	
✓	✓	getUserFirstItem()	0.0ms	
✓	✓	getUserLastItem()	0.0ms	
✓	✓	getUserInvalid()	0.0ms	
✓	✓	getUserNull()	0.0ms	
✓	✓	getUserEmpty()	0.0ms	
✗	✓	addUserStandard()	44ms	
✗	✓	addUserSpaces()	3.0ms	
✗	✓	addUserNull()	2.0ms	
✓	✓	addUserInvalidType()	0.0ms	
✗	✓	addUserInvalidPass()	3.0ms	
✗	✓	addUserInvalidEmail()	2.0ms	
✓	✓	addUserExistingUser()	0.0ms	
✗	✓	removeUserFirstItem()	5.0ms	
✗	✓	removeUserLastItem()	2.0ms	
✓	✓	removeUserNull()	46ms	▶ ⚙ 📄
✓	✓	removeUserEmpty()	24ms	

Comment- China Riley:



Quiz - Will Brown: