Project Part 3

Who: Rich Blanchard, Nika Shafranov, Elanor Hoak, Aubree Lytwyn
Title: Study Finder
Vision: We want everyone to be able to get help in a particular class if they need it.

Automated Tests:

The automated tests are created in a special test file used to run the application. Each test gives a part of the project an input and checks what the program produces to the desired output. If they match, the test passes, if not, it fails. Each test is structured the same way: output the name of the test and the result (pass/fail) so we know which tests passed or failed. If it gets to the point where there are more tests than we want printed out, then only failing tests will print their report (name and fail). The desired result is to have all tests pass.

User Acceptance Tests:

Project Name: Study Finder	
Test Case Template	
Test Case ID: TC-1	Test Designed by: Elanor Hoak
Test Priority: Medium	Test Designed date: Nov. 11, 2015
Module Name: Study Finder Add Class page	Test Executed by: Nov. 11, 2015
Test Title: Test Add New Class	Test Execution date: Nov. 11, 2015
Description: Tests adding a new class to the user's personal list. A valid class cannot already be in the user's list. This test tries to add the same new class twice, so it should only add it the first time.	
Pre-Conditions: User is logged in.	
Dependencies: Compares new class to classes already in the user's list.	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (P/F)	Notes
1	Select class from database					
2	Check class	Compare to	Not found	Not found	Pass	

	is new	user's list		(wasn't in list)		
3	Check class is new	Compare to user's list	Found	Found (was in list)	Pass	
Post-Conditions: User's personal class list contains a new class or is left the same.						

Project Name: Study Finder	
Test Case Template	
Test Case ID: TC-2	Test Designed by: Nika Shafranov
Test Priority: High	Test Designed date: 11/11/15
Module Name: Study Finder Share Location	Test Executed by: Nika Shafranov & Richard Blanchard
Test Title: Test Account Creation	Test Execution date: 11/12/15
Description: Test from one account whether a user can share their location with another user	
Pre-Conditions: Both users are logged in.	
Dependencies: Location sharing software implemented, user location services turned on	

Step	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (P/F)**	**Notes**
1	User 1: share location	Check own location on map	Location shared	Location shared	Pass	
2	User 2: check map for activity	Check User 1's location on map	Shared location is visible on User 2's profile and matches that of User 1	Location is visible	Pass	
Post-Conditions:						

Project Name: Study Finder
Test Case Template

Test Case ID: TC-3	Test Designed by: Aubree Lytwyn
Test Priority: Medium	Test Designed date: 11/11/2015
Module Name: Study Finder Find Class	Test Executed by: Aubree Lytwyn
Test Title: Test Find Class	Test Execution date: 11/11/2015
Description: This test makes sure that if a user searches for a class that does not exist then it returns as there is no class.	
Pre-Conditions: User needs to be logged in	
Dependencies:	

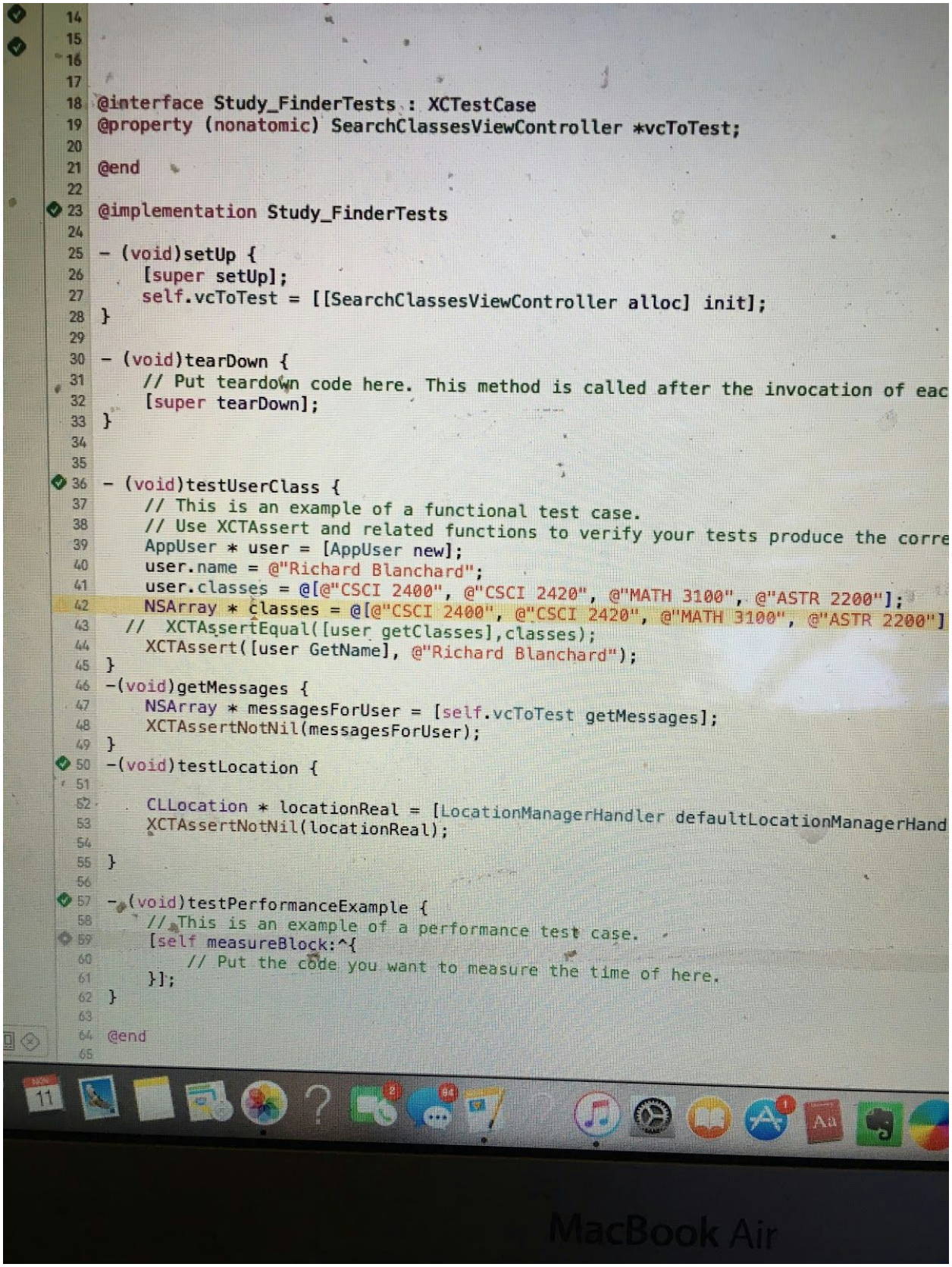
Step	Test Steps	Test Data	Expected Result	Actual Result	Status (P/F)	Notes
1	Search Valid Class that is in offered classes list (CSCI 3308)	check if on offered classes list	Class will appear and can be added	CSCI 3308	P	
2	Search class that is not in offered classes list (CSBI 2342)	check if not on offered classes list				
Post-Conditions:						

Project Name: Study Finder	
Test Case Template	
Test Case ID:ID-TC-4	Test Designed by:Rich Blanchard
Test Priority (Low/**Medium**/High):	Test Designed date: Nov. 10, 2015
Module Name:Study Finder Location	Test Executed by:Rich Blanchard
Test Title:Get Location	Test Execution date:Nov. 10, 2015

Description:This test makes sure the user's location is never nil or inaccurate
Pre-Conditions:User needs to have enabled Location Services and be logged in.
Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (P/F)	Notes
1	Get User's location		40 lat 105 long	40 lat 105 long	P	
2	Compare it to Boulder's latitude and longitude, make sure it is the same.					
Post-Conditions:We are now able to see where the user is and can show them users closeby.						

VCS: https://github.com/RichardBlanch/StudyFinder



```
@interface Study_FinderTests : XCTestCase
@property (nonatomic) SearchClassesViewController *vcToTest;

@end

@implementation Study_FinderTests

- (void)setUp {
    [super setUp];
    self.vcToTest = [[SearchClassesViewController alloc] init];
}

- (void)tearDown {
    // Put teardown code here. This method is called after the invocation of eac
    [super tearDown];
}


- (void)testUserClass {
    // This is an example of a functional test case.
    // Use XCTAssert and related functions to verify your tests produce the corre
    AppUser * user = [AppUser new];
    user.name = @"Richard Blanchard";
    user.classes = @[@"CSCI 2400", @"CSCI 2420", @"MATH 3100", @"ASTR 2200"];
    NSArray * classes = @[@"CSCI 2400", @"CSCI 2420", @"MATH 3100", @"ASTR 2200"]
  //  XCTAssertEqual([user getClasses],classes);
    XCTAssert([user GetName], @"Richard Blanchard");
}
-(void)getMessages {
    NSArray * messagesForUser = [self.vcToTest getMessages];
    XCTAssertNotNil(messagesForUser);
}
-(void)testLocation {

    CLLocation * locationReal = [LocationManagerHandler defaultLocationManagerHand
    XCTAssertNotNil(locationReal);

}

-(void)testPerformanceExample {
    // This is an example of a performance test case.
    [self measureBlock:^{
        // Put the code you want to measure the time of here.
    }];
}

@end
```

Here is a screen shot of the first tests we wrote. The first test tests whether the “getName” and “getClasses” functions return the expected outpit.

2. Test two tests are location handler. It makes sure the location is not nil.

3.The messages test makes sure that we return the correct array of messages, and that the messages returns an array, not nothing.