Object-Oriented Programming  
Assignment 1  
Testing Document

Jack Jones (22055565)

Contents

[Unit Tests 3](#_Toc157285639)

[CardTests 3](#_Toc157285640)

[DatabaseControllerTests 4](#_Toc157285641)

[DeckTests 5](#_Toc157285642)

[EnumHelperTests 5](#_Toc157285643)

[PasswordHelperTests 6](#_Toc157285644)

[Manual Testing 7](#_Toc157285645)

[Create User 7](#_Toc157285646)

[Too Short Username 7](#_Toc157285647)

[Duplicate Username 10](#_Toc157285648)

[None Matching Password 12](#_Toc157285649)

[Password Too Short 14](#_Toc157285650)

[Password One Case 16](#_Toc157285651)

[Password No Digits 18](#_Toc157285652)

[Password Has Space 20](#_Toc157285653)

[Password No Special Characters 22](#_Toc157285654)

[Valid Username and Password 24](#_Toc157285655)

[Log In 26](#_Toc157285656)

[No Username 26](#_Toc157285657)

[No Password 27](#_Toc157285658)

[User Does Not Exist 28](#_Toc157285659)

[Invalid Credentials 29](#_Toc157285660)

[Correct Admin Details 30](#_Toc157285661)

[Correct Player Details 31](#_Toc157285662)

[Leaderboard 32](#_Toc157285663)

[View Guidebook 33](#_Toc157285664)

[Reset All Credit 34](#_Toc157285665)

[Change Password 37](#_Toc157285666)

[Set User Balance 41](#_Toc157285667)

[Balance Above the Maximum 41](#_Toc157285668)

[Balance Below the Minimum 43](#_Toc157285669)

[Valid Balance 46](#_Toc157285670)

[Detailed Create User 49](#_Toc157285671)

[Combo Box Should Contain Same Role Types as Enum 50](#_Toc157285672)

[Message Handling 51](#_Toc157285673)

[Unable to Send Message When Balance Positive 51](#_Toc157285674)

[Able to Send Message When Balance 0 or Below 52](#_Toc157285675)

[Admin Should be able to View Messages 54](#_Toc157285676)

[On Decline of Message, Balance Unchanged 55](#_Toc157285677)

[On Message Approval, Player Balance Set to 100 57](#_Toc157285678)

[Game Play 58](#_Toc157285679)

[On Fold 58](#_Toc157285680)

[On Win 60](#_Toc157285681)

[On Lose 62](#_Toc157285682)

[References 64](#_Toc157285683)

# Unit Tests

Unit testing is a process of automated testing, where individual methods and functions are tested, as well as reviewing and testing all termination points (Amazon, 2023). For my project, each of my none form classes, where possible, would have a dedicated unit test file. I used the framework NUnit.

## CardTests

This set of unit test is to test the class which models a card.

A black background with white text

Description automatically generated

Image Unit Test results for Card Model

Most of these tests are based around getting the numeric value of the card. This includes, getting the value of “royal” cards (Jack, Queen, King), numeric cards (2-10), and both low value and high value aces. The latter test (which is named **ShouldSwitchAceValueOnToggleAce**) also tests the **ToggleAce** method of the card class, as well as the **GetCardValue** method.

The final of these tests, tests the **ToString** method of the card class. It asserts that a card set up as a “two of clubs” returns the correct Unicode card value (\U0001F0D2):

🃒

Figure Unicode Character for 2 of Clubs

## DatabaseControllerTests

This set of unit tests are for the class DatabaseController.

A screenshot of a computer

Description automatically generated

Image Unit Test results for DatabaseController

One of the tests is around checking that an administrator user role type, has the role type id of 1. This tests the **GetRoleTypeIdForRoleType** method.

Another test checks that the DefaultAdmin user has the correct role type (Admin). It tests the **GetRoleTypeForUsername** method.

A third test checks that there is the expected number of tables (6) created. It tests the **GetNumberOfTables** method.

There is also a test that checks that the **GetRoleTypeIdForRoleType** returns the correct value for an admin user.

One of the other tests, checks that the **GetUsernames** method returns a list of users, that is both not empty, and contains the DefaultAdmin.

As well as this, there is a test which checks that the MessageStates table, contains the same number of records, as enumerators in the Enum MessageStates. It tests the **GetMessageStates** method of the database controller.

The penultimate test of this test class is a similar test, but for role types. It checks that there is the same number of records in the RoleTypes table, as enumerators in the Enum RoleTypes. It tests the **GetRoleTypes** method.

The final test is to check that the DefaultAdmin exists. It does this using the **CheckUserExists** method of the database controller.

## DeckTests

This suite of tests is to test the class which models a deck.

A black screen with white text

Description automatically generated

Image Tests for the Deck Class

One of these tests, ensures that the **ShuffleDeck** method, appropriately shuffles the list of cards. It does this by setting up a deck, creating a list of its cards, calling the **ShuffleDeck** method, creating a new list of the cards, and comparing that the sequence has changed. There is an infinitely small chance, that this method will fail due to the shuffle order being the same as the original order. This is around 1 in 52 factorial or around (1.23979993 \* 10-66) % chance of happening, and as such I believe it to be a minimal risk (Matthews, 2010).

The other test, simpler, just checks to ensure that there are 52 unique cards in the deck. It does this by relying on the override method in Card, **GetHashCode**, which generates a unique identifier, and adds each card into a HashSet. The length of this HashSet is then checked to make sure there is 52 cards. It checks the **Cards** property of the deck class.

## EnumHelperTests

This class tests the helper class EnumHelpers.



Image Test Results for Enum Helper

The test in this class checks that the **GetStringListFromEnum** method works correctly, as a generic method. I provide the test with the expected list of values of role types (admin and player), and it checks for equivalency (so that the order does not matter) when the method is called against the Enum **RoleTypes**.

## PasswordHelperTests

This class tests the method of the helper class **PasswordHelper**.

A screenshot of a computer

Description automatically generated

Image Test results for PasswordHelper

One of the tests is to test that given the same string and initial salt, the same hashed result is produced each time. The test does this 100 times, to ensure it is always consistent. It is a test of the method **HashPassword**.

There are also several tests of the **PasswordIsValid** method. Six of these are testing that a password is marked invalid if it contains a space, when it doesn’t contain a digit, when it contains no special characters, when the password is null, when it is in a single case, or when it is too short. There is also a method to check if a password is valid.

There is another test of the **HashPassword** method to check that a hashed password is returned given some plaintext and a salt.

Another test of the **HashPassword** method, checks that when no salt is provided, a salt is automatically generated. There is also a test that when a salt is provided, the same salt is returned.

# Manual Testing

Some notes:

* If a process uses the same function call as another, it will only be tested in one place. For example, the password validation will only be checked once (on the low level create user) as it is from one class.

## Create User

### Too Short Username

When I try and create a new user using the low level create user form, when I try to use a username which is too short, I should get a message box informing me of the mistake.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer login screen

Description automatically generated

A screenshot of a computer

Description automatically generated

### Duplicate Username

When a user tries to create a user using the low level create user form, using a username that already exists, an error message should occur.

**Note: In this example, a user with the username testUser exists.**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer login

Description automatically generated

A screenshot of a computer login

Description automatically generated

### None Matching Password

When a user tries to create a user using the low level create user form, using passwords that do not match, an error message should occur.

**Note: In this example, pbPassword contains the content “password” whereas pbPasswordConfirm contains the content “hello”**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer login screen

Description automatically generated

A screenshot of a computer login

Description automatically generated

### Password Too Short

When a user tries to create a user using the low level create user form, using a password that is too short, an error message should show.

**Note: In this example, pbPassword and pbPasswordConfirm, both have the value “hello”**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### Password One Case

When a user tries to create a user using the low level create user form, using a password that is too one case (for example all lowercase), an error message should show.

**Note: In this example, pbPassword and pbPasswordConfirm, both have the value “abcdef”**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a login screen

Description automatically generated

A screenshot of a computer login

Description automatically generated

### Password No Digits

When a user tries to create a user using the low level create user form, using a password that has no digits, an error message should show.

**Note: In this example, pbPassword and pbPasswordConfirm, both have the value “Abcdef”**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a login screen

Description automatically generated

A screenshot of a computer

Description automatically generated

### Password Has Space

When a user tries to create a user using the low level create user form, using a password that has a space, an error message should show.

**Note: In this example, pbPassword and pbPasswordConfirm, both have the value “Abc de1”**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer login screen

Description automatically generated

A screenshot of a computer

Description automatically generated

### Password No Special Characters

When a user tries to create a user using the low level create user form, using a password that has no special characters, an error message should show.

**Note: In this example, pbPassword and pbPasswordConfirm, both have the value “Passw0rd”**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a login screen

Description automatically generated

A screenshot of a computer

Description automatically generated

### Valid Username and Password

When a user tries to create a new user using the low level create user form, where the username and password are valid, a success message should show, and the user should be added to the database.

**Note: In this example, pbPassword and pbPasswordConfirm will both have the value “testUser1!”**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer login screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Log In

### No Username

When no username is provided on the main window, when trying to log in, a message box should show.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### No Password

When no password is provided on the main window, when trying to log in, a message box should show.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### User Does Not Exist

When a username does not exist when trying to log in, a message box should show.

**Note: In the system, the way checking a user does not exist works, is by checking if a hash exists for that username.**

A screenshot of a computer

Description automatically generated

A screenshot of a computer login

Description automatically generated

### Invalid Credentials

When trying to log in on the main window, where the username exists, but the password is wrong, a message should be shown.

A screenshot of a computer login

Description automatically generated

A screenshot of a computer

Description automatically generated

### Correct Admin Details

When logging in on the main window, with correct admin credentials, the admin home page should be shown.

**Note: In this example, the value of pbPassword is “DefaultAdminPassword123!”**

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

### Correct Player Details

When logging in on the main window, with correct player credentials, the player home page should be shown.

**Note: In this example, the value of pbPassword is “testUser1!”**

A screenshot of a computer login screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

## Leaderboard

The leaderboard should contain a list of 5 users, ranked by their balance. When more than one user has the same balance, they should have the same rank.

A screenshot of a computer login

Description automatically generated

A screenshot of a computer

Description automatically generated

## View Guidebook

When the view guidebook button is pressed, the guidebook PDF should open in the default PDF reader.

A screenshot of a computer

Description automatically generated

A screenshot of a video game

Description automatically generated

## Reset All Credit

On the admin home page, when the reset all credit button is pressed, the balance of all players should be set to £100.

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Change Password

On the admin home page, when a user’s password is reset, it should be used to log in.

* Initial Credentials:
  + Username: testUser2
  + Password: testUser2Password!
* New Credentials:
  + Username: testUser2
  + Password: testUser2Password@

A screenshot of a login screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a login screen

Description automatically generated

A screenshot of a login screen

Description automatically generated

A screenshot of a computer

Description automatically generated

## Set User Balance

### Balance Above the Maximum

When an admin user tries to set a user’s balance above the maximum (£10,000), it should be forced to be £10,000.

A screenshot of a computer login screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### Balance Below the Minimum

When an admin user tries to set a user’s balance below the minimum (£0), it should be forced to be £0.

A screenshot of a computer login screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### Valid Balance

When an admin user tries to set a user’s balance below the minimum (£0) and the maximum (£10,000), it should be accepted.

A screenshot of a computer login screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

## Detailed Create User

A screenshot of a computer login screen

Description automatically generated

A screenshot of a computer

Description automatically generated

### Combo Box Should Contain Same Role Types as Enum

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Message Handling

My project has the concept of balance reset request messages. This section will look at how messages are used in the application.

### Unable to Send Message When Balance Positive

When a player’s balance is above 0, they should not be able to send a message request.

A screenshot of a login screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

### Able to Send Message When Balance 0 or Below

A screenshot of a login screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

### Admin Should be able to View Messages

A screenshot of a computer login screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

### On Decline of Message, Balance Unchanged

A screenshot of a computer

Description automatically generated

A screenshot of a login screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

### On Message Approval, Player Balance Set to 100

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Game Play

### On Fold

During the game of blackjack, if the user chooses to fold, they should lose half their bet amount i.e.

A screenshot of a computer screen

Description automatically generated

A screenshot of a game

Description automatically generated

A screenshot of a computer

Description automatically generated

### On Win

During the game of blackjack, if the user wins, they should win back their bet amount i.e.

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a game

Description automatically generated

A screenshot of a computer screen

Description automatically generated

### On Lose

During the game of blackjack, if the loses, they should lose their bet amount i.e.

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a game

Description automatically generated

A screenshot of a computer screen

Description automatically generated

# References

* Amazon (2023). *What is Unit Testing? - Unit Testing Explained - AWS*. [online] Amazon Web Services, Inc. Available at: https://aws.amazon.com/what-is/unit-testing/.
* Matthews, R. (2010). *What are the odds of shuffling a deck of cards into the right order?* [online] www.sciencefocus.com. Available at: https://www.sciencefocus.com/science/what-are-the-odds-of-shuffling-a-deck-of-cards-into-the-right-order.