# Testing Strategy for Elevens Java application

## Type of Operating system and software package used

* Windows 10 pro and Home
* Testing carried out on IntelliJ Idea Community Edition
* With Java JDK 11 and JUnit 4 for unit test.

## Types of testing

The testing that we carried out was mostly Junit testing, carried out on the methods of the associated classes throughout the app, however there were some methods in the classes that we were not able to test using Junit because they would have required external classes or collections to be used. This was not in our scope, so we tested these in a main method with an instantiated version of the class file to be tested.

## Purpose and scope of testing carried out

To test the working elevens card game java application, we created test classes for each class file used in the app. Each of these test classes were initially created with the methods stubbed out that should be created in their associated class file. Testing each of these stubs in the test classes thoroughly will ensure that the methods will work as expected, to bring about the functionality required by the design specification, previously laid out in our design and development.

We went through each element as thorough as possible to ensure working order of each method tested, this helped us to see that our app had no underlying flaws that were not being picked up by other conventional test. Such as taking a demo through and running through the app itself, as a black box test methodology, where you are trying out the functionality of the app as a user’s perspective where you are not looking at the code, just at a surface level seeing the output of the functionality of the elevens game.

As another way to quality check the code and unit tests we have written, we had numerous zoom call to code quality check and pair programme some of the functionality and tests. This helped to maintain a more consistent and higher quality code through the app and as a result of this the app has had few bugs and tests passed when we went through Junit testing the classes.

## Test classes and other methods

Each class create in the elevens game java application had a test class create for it where we would Junit test the methods contained within, where was possible and where it wasn’t possible, we tested the methods in a main method using an instance of the class.

The test classes tested in the application are as follows:

* CardTest
* CardSlotsBagTest
* CardNodeTest
* DeckTest
* HouseTest
* RankTest
* RoundQueueTest
* RoundTest

## Testing against requirements

We will have a full run through the elevens game application to fully test the playable functionality. Throughout these tests we referenced the requirements of the elevens game set to us in the design document, checking the functionality works in accordance with what is described in the requirements. Checking them off as we tested through playing the game and any aspects which were not covered sufficiently, we implemented the requirement missed out on. We then tested after each new piece of functionality was added and checked against our requirements until satisfied.

//TODO can you do a requirements against test matrix and wheather is it passed or failed

|  |  |  |
| --- | --- | --- |
| Requirement | Test Description | Meets requirements |
| Level 1 | How you met requirement, example, Was able to selection a new game in the console menu which created a new game. | Yes |
| Level 2 |  | Yes/No |

//TODO UML

//TODO FLOW chart

etc

Launch app => menu => Game Menu

* Quit application