**The Software Institute**

**Qualitest Cohort 1**

Test Plan X

for

Blackjack App

Version 1.1

**Tester: A. Quanunga, Sam King**

**Test Manager: G. Davis, S. Syuleyman**

13/09/2021

# Test Plan Identifier

Test Plan X

Version date: 13/09/2021

Version number: 1.1

Revision History:

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version Number | Author | Comment |
| 13/09/2021 | 1.0 | Arkaan Quanunga, Sam King | Initial Draft |
| 14/09/2021 | 1.1 | Arkaan Quanunga, Sam King | Added Test Process & Deliverables |
|  |  |  |  |

# 1.0 Introduction:

Test Plan X prescribes the scope, approach, resources, and schedule of all testing activities of the project Blackjack App.

The plan identifies the items tested, the features to be tested, the types of testing to be performed, the personnel responsible for testing, the resources and schedule required to complete testing, and the risks associated with the plan.

## Scope

|  |  |
| --- | --- |
| Abbreviation | Word |
| L | Low |
| M | Medium |
| H | High |

The features to be tested in Test Plan X includes:

|  |  |  |
| --- | --- | --- |
| Feature | Risk Level | Description |
| Number of Decks | L | Sets the number of decks to be eight by default |
| Shuffle | H | Shuffles the Deck of Cards Created |
| Deal Cards | H | Deal a set number of cards to the players |
| Dealer Stands on 17 or above | H | The dealer stops drawing Cards when points reach 17 |
| Dealer Hits below 17 | H | The dealer keeps drawing Cards until points reach 17 |
| Split Pairs | L | When the player Draws the Same Value Card, they can split their hand into two |
| Double on 9/10/11 or 12 | L | The player can double the bet if the points are between 9 to 12 |
| Set Player Money | M | Creates a money Account for the player to use when betting. |
| Set Player Bet | M | Sets the Player Bet from the money they have |
| Blackjack Pays | H | Player wins 3:2 |
| Normal Pays | H | Player wins 2:1 |

## 1.1.2 Out of Scope

The feature not to be tested includes:

* User Interfaces
  + This is Acceptance Testing and will be performed when Web app is developed.
* Hardware Interfaces
  + There is no hardware involved in this software; therefore, hardware interfaces testing is unnecessary.
* Security and Performance
  + Very low level of software so doesn't need to test Security or Performance until Web app development.
* Split Pairs won't be tested as its an additional feature and needs to be added later into the Blackjack App

# 2.0 Test Objectives/ Quality Objectives

The test objectives for Blackjack App are the following:

* To verify the functionality of the Blackjack App game
* To prevent defects from occurring at component testing
* To find failures and defects throughout the development process
* To reduce the level of risks of inadequate software quality
* To build confidence in the quality of Blackjack App
* To validate whether Blackjack App is complete and works for the users as intended

The test objectives to exclude in Blackjack App are the following:

* Comply with the legal requirements of the minimum age for gambling set to be 18.
  + As people are not betting, it's not gambling.
* Comply with the legal requirements of holding a license for running an Online Casino
  + As people are not betting actual money, this is unnecessary.

# 3.0 Test Approach

Test Plan X uses the following Test Strategies for testing:

1. Analytical – Test strategy based on analysis of the requirement of the feature.
2. Reactive- Test strategy where testing is reactive to the component being tested and the events that occur during the test execution.

## 3.1 Test Environment

Mentions the minimum requirements that will be used to test Blackjack App:

* Windows 8 and above
* Office 2013 and above
* Java 16

## 3.1.1 Test Tools

The following tools are used for static analysis testing:

* IntelliJ IDEA 2021.1 Community Edition by JetBrains for static analysis of code
* Sonar Cloud for checking code smells, security vulnerabilities, and bugs.

The following tools are used for version control:

* Git 2.33 by Linus Torvalds for version control
* Git Hub for cloud-based hosting service, managing Git Repositories

The following tools are used for the Continuous Integration Process of Static Analysis

* GitHub/workflow/build.yml file from Sonar Cloud

The following tool is used to manage and build Blackjack App:

* Apache Maven 3.8.1 based on a project object model (POM)

The following tool is used for test design and implementation:

Behaviour-driven development is being used.

* IO. Cucumber JUnit 6.11.0 - Dependency
* Info. Cukes Cucumber Junit 1.2.6 - Dependency
* Cucumber for Java 212.4746.52 - Plug-In. IntelliJ.

The following tool is used for test execution and logging:

* Junit 5.8 testing framework for writing unit tests – Dependency
* Jacoco 0.8.7 for measuring code coverage in a codebase through visual reports
* Maven sure-fire plugin 3.0.0-M5 for executing the unit tests

Additional Comments:

Few tools require a little special training to get used to.

# 3.2 Test Levels

In Test Plan X, 3 types of testing were conducted:

1. Component Testing- All the classes/Objects created must be thoroughly tested through unit tests and by performing test cases based on testable requirements
2. Component Integration Testing- Components should be able to interact with each other through integration testing.
3. System Testing- The System should be able to function correctly according to the required functionality and non-functionality.
4. Regression Testing to ensure that a change to the system does not introduce new defects

# 3.3 Configuration Management

As the software currently is small and therefore doesn't consist of many components. Configuration Management will not be performed to save time.

# 3.4 Regression Test Rules

* The tests will be created and carried out every time a new feature is added to the system.
* The tests will be created and carried on when a new requirement is added to an existing feature of the system.
* The tests will be created and carried out when the code base is fixed to solve defects.
* All the tests cases in the test suites will run simultaneously to ensure changes in the source code don't create new bugs or defects in the source code.

# 3.5 Test Techniques

The two common categories of test techniques being used in Test Plan X:

1. Black Box testing techniques
2. White Box testing techniques

## 3.5.1 Black Box techniques

In Test Plan X 5 black box techniques are being used:

### 1.&2. Equivalence Partitioning Testing & Boundary Value Analysis Testing:

Player Money:

|  |  |  |
| --- | --- | --- |
| Invalid | Valid | Invalid |
| All Negative Values - 0 | 1-10000 | >10000 |

Player Bet:

|  |  |  |
| --- | --- | --- |
| Invalid | Valid | Invalid |
| -1< | 1-10000 | >10000 |

Player Points:

|  |  |  |
| --- | --- | --- |
| Invalid | Valid | Invalid |
| -1< | 0-21 | >21 |

Dealer Points:

|  |  |  |
| --- | --- | --- |
| Invalid | Valid | Invalid |
| -1< | 0-21 | >21 |

Number of Decks:

|  |  |  |  |
| --- | --- | --- | --- |
| Invalid | Invalid | Valid | Invalid |
| -1< | 0 | 1-8 | >8 |

### Decision Table testing:

|  |  |  |  |
| --- | --- | --- | --- |
| Decision Table for Player Choices | Rules | | |
| 1 | 2 | 3 |
| Conditions |  |  |  |
| Player Points<21 | Y | Y | N |
| Player Points 9,10,11 or 12 | Y | N | N |
| Player Points>=21 | N | N | Y |
| Actions |  |  |  |
| Player choice Hit or Stay | Y | Y |  |
| Player Choice to Double down | Y |  |  |
| Game Finished- Choice for continuing to play |  |  | Y |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Decision Table for Player Wins | Rules | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Conditions |  |  |  |  |  |  |  |
| Player Points= 21 | Y | N | N | N | Y | N | N |
| Dealer Points< Player Points | Y | Y | N | N | N | N | Y |
| Dealer Points> Player Points | N | N | Y | N | N | Y | N |
| Dealer Points = Player Points | N | N | N | Y | Y | N | N |
| Dealer Points>21 | N | N | N | N | N | Y | N |
| Player Points>21 | N | N | N | N | N | N | Y |
| Actions |  |  |  |  |  |  |  |
| Blackjack Win- 3:2 | Y |  |  |  |  |  |  |
| Push Win - Player keeps the bet |  |  |  | Y | Y |  |  |
| Player Win- 2:1 |  | Y |  |  |  |  | Y |
| Player Lose- 0 |  |  | Y |  |  | Y |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decision table for Dealer Choices | Rules | | | |
| 1 | 2 | 3 | 4 |
| Conditions |  |  |  |  |
| Dealer points<17 | Y | N | Y | N |
| Dealer points>=17 | N | Y | N | Y |
| Player Points>21 | N | N | Y | Y |
| Actions |  |  |  |  |
| Dealer Hit | Y |  |  |  |
| Dealer Stay |  | Y |  |  |
| Game Finished- Choice for continuing to play |  |  | Y | Y |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Decision Table for Player Hand When 2 Cards are drawn | Rules | | | | |
| 1 | 2 | 3 | 4 | 5 |
| Conditions |  |  |  |  |  |
| Player Hand draws Ace | Y | N | Y | N | Y |
| Player Hand draws a pair of identically ranked cards | N | Y | Y | N | N |
| Player Hand Draws any other card | N | N | N | Y | N |
| Player Hand Draws a Blackjack Value of 21 | N | N | N | N | Y |
| Actions |  |  |  |  |  |
| Player Hand points can increase dynamically by +1 or +11 | Y |  | Y |  | Y |
| Player Hand CAN Splits in two |  | Y | Y |  |  |
| Add points of the Card Drawn |  |  |  | Y |  |
| Game Finished- Choice for continuing to play |  |  |  |  | Y |

|  |  |  |  |
| --- | --- | --- | --- |
| Decision Table for Continue Playing | Rules | | |
| 1 | 2 | 3 |
| Conditions |  |  |  |
| Player has Money | Y | Y | N |
| Player clicks Yes | Y | N | Y |
| Player clicks No | N | Y | N |
| Action |  |  |  |
| Continue Playing | Y |  |  |
| Exit Game |  | Y | Y |

### State Flow Diagram for State Transition testing

Diagram

Description automatically generated

### Use Cases

This part will be done using Cucumber and Gherkin plugins in the Maven framework.

The following Use Cases will be tested. Diagram

Description automatically generated 3.5.2 White Box Testing Techniques:

1. Statement Coverage
2. Decision Coverage- Test Plan X focuses on covering all the decisions in Blackjack App. This ensures 100% statement coverage.

* This is done by using Junit for designing the Unit tests
* Sonar Cloud with Jacoco plugin to analyse the Decision coverage in Blackjack App

# Item Pass/Fail Criteria:

* For component testing:
  + 100% of test cases must be completed with a 5% threshold which may contain minor defects
  + 80% Code coverage must be achieved at Component testing for the software to be progressed further.
* All Player betting functions must work correctly
* All Card Distribution functions must work correctly
* All Card Value calculations work correctly
* The system is easy to use by the end-users
* All Player betting calculations are correct.
* All points of integration within components work as defined in the requirement

# 5.0 Suspension Criteria and Resumption Requirements:

## 5.1 Normal Criteria

At the end of each day (6:00 pm), testing will be suspended. At that time, all test cases executed during the day should be marked as such. The testing will resume by performing automated regression testing at the start of the day.

When all test cases have been executed, the test will be suspended, and the results documented for the Test Summary Report

## 5.2 Abnormal Criteria

If at least 40% of the test cases failed, the testing would be stopped immediately for the development team to fix any errors.

If the defect backlog continually increases over a one-week period, testing should be suspended. This time frame will allow developers time to fix existing defects without pressure and confusion about adding new defects to the backlog.

If a critical feature is found to have severe defects, testing should be suspended until the defects have been fixed. When the fixed unit is moved back into the test environment, any previously performed tests that affect the test object should be achieved again to ensure new defects were not created because of the fix. (Regression Testing).

# 6.0 Deliverables

The deliverables of the test activities are outlined in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Deliverable | Owner | Review | Due Date |
| Test Plan | A. Quanunga, S. King | G. Davis, S. Syuleyman | 13 September 2021 |
| Test Environment | A. Quanunga, S. King | - | 16 September 2021 |
| Test Suite | A. Quanunga, S. King | Peer Review | 20 September 2021 |
| Test Scripts | A. Quanunga, S. King |  | 20 September 2021 |
| Test Stubs, Drivers | A. Quanunga. S. King |  | 20 September 2021 |
| Test Defect Reports | A. Quanunga, S. King | G. Davis, S. Syuleyman | 23 September 2021 |
| Test Results | A. Quanunga, S. King |  | 23 September 2021 |
| tEST TRaceability matrix | S. King |  | 23 September 2021 |
| Test Evaluation Report | A. Quanunga, S. King | G. Davis, S. Syuleyman | 24 September 2021 |

# 7.0 Test Tasks

## Test Process

### 1. Test planning:

* Create a Test plan consisting of the following:
* Define the test objectives and constraints
* Use suitable test techniques and tasks

### 2. Test Monitoring:

* Checking test results and logs against specified coverage criteria in the Test plan
* Assess the level of quality of the software based on the test results and logs
* Determine if additional tests are required.

### 3. Test analysis & Design

* Analyze the Requirement specification for unclear elements and discuss with the project owner- includes ambiguities, inconsistencies, inaccuracies, and superfluous statements.
* Prioritize test conditions for each feature considering functional characteristics.
* Design and prioritize test cases for each feature
* Design Test environment and identify any required infrastructure and tools

### 4. Test implementation

* Create test suites from test procedures and create automated test scripts
* Arrange test suites with the test execution schedule
* Build Test Environment and verify everything needed has been set up

### 5. Test Execution

* Record IDs and version so test item, object, and test tools
* Executing tests
* Comparing actual results with expected results
* Reporting defects based on failures observed
* Logging the outcome of test execution
* Repeating test activities – regression testing and confirmation testing

### 6. Test Completion

* Checking whether all defect reports are closed.
* Evaluate Code Coverage
* Determine if Test completion Criteria and Success Criteria are completed.
* Create a Test summary report
* Use information gathered to improve test process maturity

# 8.0 Resource Availability

## 8.1 Roles

The following table shows the staffing assumptions for the test of Blackjack App

|  |  |  |
| --- | --- | --- |
| **Human Resources** | | |
| **Role** | **Name** | **Responsibilities** |
| **Test Manager** | G. Davis, S. Syuleyman | * Provide Technical Direction * Acquire appropriate resources * Management reporting |
| **Test Designer** | A. Quanunga, S. King | * Generate Test plan * Generate Test Suite * Evaluate the effectiveness of test effort |
| **System Tester** | A. Quanunga, S. King | * Execute tests * Log results * Recover from errors * Document defects |
| **Test System Administrator** | A. Quanunga, S. King | * Ensure test environment and assets are managed and maintained |
| **Designer** | A. Quanunga, S. King | * Identifies and defines the test classes * Identifies and describes the test packages |
| **Implementer** | A. Quanunga, S. King | * Create the test cases and packages implemented in the Test Suite. |

## 8.2 System

The following System resource is available:

|  |  |
| --- | --- |
| Resource | Configurations |
| 1 x Dell Inspiron 15 7000 Gaming Laptop | Operation System: Microsoft Windows 10.0.19042 Build 19042  System Type: x64-based PC  Processor: Intel® Core™ i7-7700HQ CPU @ 2.80 GHz, 2801 MHz, 4 Core(s)  Installed memory (RAM) : 16.0 GB |
| 2 x Wi-Fi- Internet Service Provider | 150 MB/S Download Speed, 30 MB/S Upload  13 MB/S Download Speed, 4 MB/S Upload |
| 1 x ACER Nitro N50-110 | Operation System: Microsoft Windows 10.0.19042 Build 19042  System Type: x64- based PC  Processor: Intel® Core™ I5-10400F CPU @2.90 GHz, 2901 MHz, 4 Core(s)  Installed memory (RAM) : 8.0 GB |

## 8.3 Staffing and Training Needs

Initially, the tester will be required to obtain one week of training to use the tools mentioned in the tools section. This is to ensure that the tester does not face any future difficulties during the testing process and hence not disrupt the Schedule of testing.

# 9.0 Schedule

The schedule consists of test milestones, all item transmittal events, the time required to do each testing task, and periods of use for each testing resource.

## 9.1 Project Milestones

The project milestones in Test Plan X are set by considering the development activity so that slippage is reduced. These milestones use expert-based techniques to estimate each task's test effort based on the tested project features.

Adaptations to the schedule are required as the SDLC continues through Test monitoring and Control.

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone Task | Effort(per day) | Start Date | End Date |
| Test Planning | 2 | 01 13 September 2021 | 03 15 September 2021 |
| Test Monitoring & Control | 10 | 03 15 September 2021 | 25 September 2021 |
| Test Analysis & Design | 3 | 15 September 2021 | 18 September 2021 |
| Test Implementation | 4 | 18 September 2021 | 22 September 2021 |
| Test Execution | 3 | 22 September 2021 | 25 September 2021 |
| Test Completion & Evaluation | 1 | 25 September 2021 | 26 September 2021 |

## 9.2 Periods of Usage for testing resource

System Resource will be used for standard x 40 hours per week.

The tester is recommended to work for x 40 hours per week.

# 10. Risk and Contingency

There are several risks associated with this project.

1. There is a lack of Human resource as all the testing is done by one tester.
2. Reviews are not being performed during this project as it is a small project
3. There can be changes to original requirements such as double down feature changed from 9-12 to every player turn.
4. The training time for the tester may be insufficient.

The deadline for the overall testing is set to be 25th September 2021.

* If additional features are added to the software design specification, the number of testing will increase.
* If scope of the plan is changed and there is increase in overall testing
  + Resources may be added to the team if too many additional features are added to stick to the current deadline of 25th September 2021.
* If development team takes too long to develop the project:
  + The number of tests performed may be reduced to high risk features and stick to the deadline
  + The threshold of minor defects may increase from 5% to 10% in testing.

# 11.Test Plan Approvals

The undersigned acknowledge they have reviewed the *<Blackjack App>* **Test Plan X** document and agree with the approach it presents. Any changes to this Requirements Definition will be coordinated with and approved by the undersigned or their designated representatives.

|  |  |
| --- | --- |
| Signature: |  |
| Print Name: |  |
| Title: |  |
| Role: | Test Manager |
| Date: |  |
| Signature: |  |
| Print Name: |  |
| Title: |  |
| Role: | Test Manager |
| Date: |  |