**Test strategy**

1. Requirements elicitation: The game consists of three reels that spin when you press the spin button. When the spin ends you will get one of the following three messages based on your winning status:

2) Features to be tested: Spin button, No win scenario, Small win scenario, Big win scenario

3) Types of testing to be performed:

Functional testing, UI testing, user acceptance, Compliance testing, localisation testing, Mobile testing, Automation testing, testing on different software and hardware configurations, Content Testing to validate the following aspects of the game:

o Is the content appropriate for the gamers?

o How easy it is to understand the content?

4) Bug Tracking tool: XYZ

5) Test entry criteria:

Defined and Approved Requirements

Test Plan, Test Cases and Test Data

Test Tools

Testable Game with stable Test Environment

6.) Test exit criteria:

Ensuring all critical Test Cases are passed

Identifying and fixing all the high-priority defects

Re-testing and closing all the high-priority defects to execute corresponding Regression scenarios successfully

**How to make game more testable**

1. QA should be able to alter the reel positions to see how different results show different messages.
2. Would be nice to have the strings for messages in a separate document (approved by someone)
3. Would be nice to have the animations in a separate file so QA can test if right animations are shown.
4. Some lines/reel animation to show the pattern of win would be nice for end user.
5. Also need to see how the game behaves when the system crashes or there are other unforeseen circumstances.
6. The game should be able to resume back to where the user left.
7. Also, need to test the game on varying hardware devices and fluctuating network.