**Software Implementation and Testing Document**

**For**

**Group**

**Casino and More**

Version 2.0

**Authors**:

Brandon Mohammed

Alex Tedeschi

David Risi

# Programming Languages (5 points)

*List the programming languages use in your project, where you use them (what components of your project) and your reason for choosing them (whatever that may be).*

We use Java solely for all components of the project. Specifically, the Java Swing library. We choose to use Java due to its many easy to use graphical components and easy implementable options such as ActionListener’s and Paintcomponents. As our game has multiple different screens, we implemented new games with JFrames so they open in a separate window instead of the same JPanel.

# Platforms, APIs, Databases, and other technologies used (5 points)

*List all the platforms, APIs, Databases, and any other technologies you use in your project and where you use them (in what components of your project).*

We are not currently using any API’s or databases. As we continue to create the different games we may need to make use of some API’s or databases for the AI. We are using VsCode and IntelliJ to code the software and test on downloaded jdks.

# Execution-based Functional Testing (10 points)

*Describe how/if you performed functional testing for your project (i.e., tested for the* ***functional requirements*** *listed in your RD).*

So far we have tested Roulette, Main Menu, Blackjack, and the how to play menu. These features have not been completed fully, therefore only the completed parts can be tested. We tested main menu and how to play by compiling on jdks and seeing the output. We tested Roulette and Blackjack by compiling the software and tested the completed features to make sure they work properly.

# Execution-based Non-Functional Testing (10 points)

*Describe how/if you performed non-functional testing for your project (i.e., tested for the* ***non-functional requirements*** *listed in your RD).*

We have tested the animation to spin the roulette wheel when the “spin” button is pressed. This is a non-functional requirement. We tested this by compiling the roulette program and clicking on the spin button. Money system in roulette has also been tested this allows a user to keep track of their money throughout the program.

# Non-Execution-based Testing (10 points)

*Describe how/if you performed non-execution-based testing (such as code reviews/inspections/walkthroughs).*

We have all reviewed and walked through all the code implemented so far. We plan on continuing to do this to catch any possible error cases we may have missed.