**ACTIVITY 3 - Step-by-Step Solution**

**Number Guessing Game**

1. Create a blank process and, on the Design tab, in the File group, select New > State Machine. The New State Machine window is displayed.
2. In the Name field type a name for the automation, such as "First State Machine", and leave the default project location or add a subfolder. Click Create. The Designer panel is updated accordingly.
3. Create two integer variables, InitialGuess and RandomNumber. The first variable stores your guess, while the second stores the random number.
4. Add a State activity to the Designer panel and connect it to the Start node. This is the initial state, and it is used to generate a random number.
5. Double-click the activity. This State activity is displayed expanded in the Designer panel.
6. In the Properties panel, in the DisplayName field, type Initializing Random Number. This enables you to easily tell states apart.
7. In the Entry section, add an Assign activity.
8. In the To field, add the RandomNumber variable.
9. In the Value field, type new Random().Next(1,100). This expression generates a random number.
10. Return to the main project view and add a new State activity.
11. Connect it to the previously added activity.
12. Double-click the last added State activity. This activity is displayed expanded in the Designer panel.
13. In the Properties panel, in the DisplayName field, type Guess Number. This state is used to prompt the user to guess a number.
14. In the Entry section, add an Input Dialog activity.
15. Select the Input Dialog, and in the Properties panel, add an appropriate Label and Title to prompt the user to guess a number between 1 and 100.
16. In the Result field, add the InitialGuess variable. This variable stores the user’s guess.
17. Return to the main project view and create a transition that points from the Guess Number state to itself.
18. Double-click the transition. The transition is displayed expanded in the Designer panel.
19. In the Properties panel, in the DisplayName field, type Try Smaller. This message is displayed on the arrow, enabling you to run through your automation easier.
20. In the Condition section, type InitialGuess > RandomNumber. This verifies if the user’s guess is bigger than the random number.
21. In the Action section, add a Message Box activity.
22. In the Text field, type something similar to "Your guess is too big. Try a smaller number." This message is displayed when the user’s guess is bigger than the random number.
23. Return to the main project view and create a new transition that points from the Guess Number state to itself.
24. Double-click the transition. The transition is displayed expanded in the Designer panel.
25. In the Properties panel, in the DisplayName field, type "Try Bigger". This message is displayed on the arrow, enabling you to run through your automation easier.
26. In the Condition section, type InitialGuess < RandomNumber. This verifies if the guess is smaller than the random number.
27. In the Action section, add a Message Box activity.
28. In the Text field, type something similar to "Your guess is too small. Try a bigger number." This message is displayed when the users guess is smaller than the random number.
29. Return to main project view and add a Final State activity to the Designer panel.
30. Connect a transition from the Guess Number activity to the Final State.
31. In the Properties panel, in the DisplayName field, type "Correct Guess".
32. In the Condition field, type InitialGuess = RandomNumber. This is the condition on which this automation steps to the final state and end.
33. Double-click the Final State activity. It is displayed expanded in the Designer panel.
34. In the Entry section, add a Message Box activity.
35. In the Text field, type something similar to "Congratulations. You guessed correctly! The number was " + RandomNumber.ToString + "." This is the final message that is to be displayed, when the user correctly guesses the number.
36. Press F5. The automation is executed correctly.