|  |
| --- |
| **Dictionary Class** |
| Rules / Constraints | Valid Equivalence Classes | Invalid Equivalence Classes |
| Number of characters of the guessed word must be greater than 0. | words.toString().lenght() > 0 | words.toString().lenght() = 0  words.toString().lenght() = null |
| The words may only contain letters and spaces as characters. | [A-Za-z/s] | [./\/-/=/#/@/%/(/0-9)/] … ! [A-Za-z/s] |
| The Maximal Length of a word from the Dictionary cannot exceed 10 for design purposes. | words.toString().lenght() = 10 | words.toString().lenght() = 11 |
| The word must not be null or an invisible space only | “apple” | “ ” |

|  |
| --- |
| **Scoreboard Class** |
| Rules / Constraints | Valid Equivalence Classes | Invalid Equivalence Classes |
| The numberPlayers must be a positive integer | numberPlayers = 0 - 999… | numberPlayers = -1 - -999… |
| The numberPlayers begins at 0 and doesn’t have a determined limit. | numberPlayers = 0 - 999… | numberPlayers = -38 - 999… |
| Only player objects should be store in the players DLNode | players.getObject().equals(Player) | players.getObject().equals(Scorboard) |

|  |
| --- |
| **HangmanLogic Class** |
| Rules / Constraints | Valid Equivalence Classes | Invalid Equivalence Classes |
| letterAnswers SLNode only contain characters of the current word played | letterAnswers<String> | letterAnswer<Long> |
| classValidationId is a variable that stores a number of type long when serialized and must be randomly generated and checked. Of length 30 | 135609945446954606059978429194 | 135609945446954606059978429194559519  ||  1356099454 |
| numlives begins at zero every for every single player new game | numLives = 6 | Numlives = 8 |
| instantiateValidationID should assure the id from a deserialized game state is the same as the serialized game state id previously saved. | classValidationId = playerClassID | classValidationId != playerClassID |
| giveHint() should only provide a single character hint | Number of hints is equal to 1 | Number of hints exceed 1 |
| saveGameState() serializes the game state | Save id is recorded and the object is serialized | The save id isn’t recorded and the object isn’t serialized |
| loadGameState() deserialized the games state | Deserialized the object and assures the saved id is valid | Doesn’t deserialize and the saved id is invalid. |

|  |
| --- |
| **Player Class** |
| Rules / Constraints | Valid Equivalence Classes | Invalid Equivalence Classes |
| name must be a string allowed hyphens, apostrophes. | [A-Za-z/s/-/’] | [./\/-/=/#/@/%/(/0-9)/] … ! [A-Za-z/s/-/’] |
| name must not exceed 30 characters for design purposes. | name.lenght() = 29 | name.length() = 30 |
| numberGamesWon must be a positive Integer | 0 – 999… | -1 - -999… |
| numberGamesPlayed must be a positive Integer | 0 – 999… | -1 - -999… |
| numberGamesWon cannot exceed the number of gamesPlayed. | numberGamesWon <= numberGamesPlayed | numberGamesWon > numberGamesPlayed |

|  |
| --- |
| **HangmanFrame Class** |
| Rules / Constraints | Valid Equivalence Classes | Invalid Equivalence Classes |
| Must be visible | setVisible = true | setVisible = null || false |
| Must be closable | System.exit(0) | Doesn’t close on exit |
| Saves the game state | Calls logic to serialize the game state and indicates it to the user | The frame does nothing |
| Tells the user when the program executes an action such as saving, closing, displaying game state, etc. | The frames display information messages in JOptionPanes | The games doesn’t display a message when actions are performed. |
| The game displays to the user the rules of the game | The games display all the rules of the game to the user on the click of the “infoMenuItem”. | The Game doesn’t display the rules. |