**Sudoku Application - Project Test & ICD**

**Rev. v1.1**

**Abel Tabor, Thomas Edwards, Brian Jefferson**

**CMSC 495**

**Section 7384**

**Group 5**

**4/19/2022**

**Professor Dao**

**Revision History**

| **Name** | **Date** | **Description** |
| --- | --- | --- |
| Thomas Edwards | 4/14/2022 | Started document, added Revision History, added Test Plan Scenario tables, \*added document copies to GitHub Repository\*. |
| Abel Tabor | 4/17/2022 | Started and completed ICD and started on Test Cases adding Test Case 1 and 2 |
| Thomas Edwards | 4/19/2022 | Added Test Cases 3-6 for additional testing of functionality; Restructured test cases in progressive format; minor editing. Additional comments added to the Test Plan description. |
| Abel Tabor | 4/19/2022 | Added test case for user name input. |
| Thomas Edwards | 4/19/2022 | Added overview of normal game operation and denoted key error handling scenarios to be tested. |
|  |  |  |
|  |  |  |

**Project Test Plan**

**Overview**

Our group utilized this document to organize the testing of each scenario outlined in the sudoku application. The test plan outlines all application input scenarios and their expected outputs. Additionally, All classes/subsystems that communicate have their own ICD tables.

**Normal application operation:** on application launch, the user shall have the option to **“Start New Game”** or **“View Scoreboard”**. Upon selecting a new game, the game board will be generated and the game timer will start. The user will utilize the individual tile/square drop-downs to select non-duplicate numbers based on their generated sudoku board. So long as the user has not ended the game, the game timer will continue until the user completes the game. Upon completion, the game timer will stop and the user will be prompted to enter their name for the scoreboard. Next, the scoreboard will be updated with the user's name, score, and game time.. The scoreboard will only be updated after completion of a full game. The user will only be prompted to enter their name after a full game.

**Key Error Handling Scenarios to be tested:**

* **Inavalid user input on board:** \*Mitigated by restricting entries to drop-down buttons on board\*
* **Partial game entries:** \*No username prompt required, no update to scoreboard; game ends\*
* **Username entries:** \*No more than 10 characters: Upper/lower case letters and numbers\*

**Interface Control Documentation**

**Data between MenuGUI and GameGUI:**

| *From* | *To* | *Name* | *Type* | *Value Range* | *Default Value* |
| --- | --- | --- | --- | --- | --- |
| MenuGUI | GameGUI | showGame | boolean | true - false | false |

**Data between MenuGUI and LeaderGUI**

| *From* | *To* | *Name* | *Type* | *Value Range* | *Default Value* |
| --- | --- | --- | --- | --- | --- |
| MenuGUI | LeaderGUI | showLeader | boolean | true - false | false |

**Data between Database and LeaderGUI**

| *From* | *To* | *Name* | *Type* | *Value Range* | *Default Value* |
| --- | --- | --- | --- | --- | --- |
| Database | LeaderGUI | userData | String[] | null - Strings | null |

**Data between SudokuGenerator and GameGUI**

| *From* | *To* | *Name* | *Type* | *Value Range* | *Default Value* |
| --- | --- | --- | --- | --- | --- |
| SudokuGenerator | GameGUI | fullBoard | int[][] | 0-9 | 0 |
| SudokuGenerator | GameGUI | unsolvedBoard | int[][] | 0-9 | 0 |
| GameGUI | SudokuGenerator | userBoard | int[][] | 0-9 | 0 |
| SudokuGenerator | GameGUI | puzzledSolved | boolean | true - false | false |

**Data between GameGUI and GameLogic**

| *From* | *To* | *Name* | *Type* | *Value Range* | *Default Value* |
| --- | --- | --- | --- | --- | --- |
| GameLogic | GameGUI | score | int | 0-10,000 | 10,000 |
| GameLogic | GameGUI | time | int[] | 0-60 | 0, 0, 0 |
| GameLogic | GameGUI | timeString | String | a-z | null |
| GameGUI | GameLogic | name | String | a-z | null |

**Data between GameLogic and Database**

| *From* | *To* | *Name* | *Type* | *Value Range* | *Default Value* |
| --- | --- | --- | --- | --- | --- |
| GameLogic | Database | score | int | 0-10,000 | 10,000 |
| GameLogic | Database | time | int[] | 0-60 | 0, 0, 0 |
| GameLogic | Database | name | String | a-z | null |
| Database | GameLogic | validEntry | boolean | true - false | null |

**Functional Requirements**

| **Unique ID** | **Requirements** | **Specification** |
| --- | --- | --- |
| 01 | Menu GUI | The menu GUI shall allow users to select an option of starting a new game or viewing the leaderboards |
| 02 | Sudoku Generator | The sudoku generator shall take an empty board and randomly generate a fully solved sudoku board and shall take an unsolved board and determine if it is capable of being solved. The generator shall also be capable of returning a boolean on the user submitted board. |
| 03 | Game GUI | The GUI shall display a number of given numbers in a non-interactive manner and the unknown numbers in an interactable manner. The GUI shall allow the user to interact with the puzzle in order to solve it. |
| 04 | Game Time & Game Score | The program shall track the amount of time the user spends on any given puzzle and keeps track of their total score. |
| 05 | Database | A database shall be incorporated into the application in order to store and display usernames, scores, and times. |
| 06 | Leaderboard GUI | A leaderboard GUI shall display a ranked list of previous users with their time and score. |

**Test Cases**

**Note: Key error handling scenarios denoted in orange**

| **Test Case** | **Requirement** | **Test Description** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| 1 | 01, 05, 06 | **\*App launch/Check Scoreboard\***  User launches application and selects to see leaderboard, exits game. | Leaderboard GUI opens and entries from the database are shown. **\*empty scoreboard\*** Application closes |  |  |
| 2 | 01, 02, 03, 04, 05, 06 | **\*Game launch/end with no input(s)\***  User launches the application, starts a game, and immediately ends game. | Normal startup, game starts, game ends.  **\*No prompt to enter name/ Scoreboard not updated\*** |  |  |
| 3 | 01, 02, 03, 04, 05, 06 | **\*Partial game play\***  User launches application, inputs a couple of numbers, and exits game; Check scoreboard.  **\*Attempt duplicate input\*** | Normal startup, game start/end.  **\*Duplicate input not allowed/ No prompt to enter name/ Scoreboard not updated\*** |  |  |
| 4 | 01, 02, 03, 05 | **\*Game launch/play with invalid name input\***  User launches application, completes a puzzle and attempts to enter invalid names that include: invalid characters, attempts at SQL injection, more than the allocated number of characters in a name | Normal startup, game starts and continually asks user for valid name when given invalid user inputs.  **\*Invalid names are not allowed, Scoreboard not updated until valid entry made\*** |  |  |
| 5 | 01, 02, 03, 04, 05, 06 | **\*Single, full game\***  User launches the application and selects “start a new game”. User plays game and solves puzzle and submits name onto leaderboard. | GameGUI and SudokuGenerator generate a board for the user to solve, once user has entered in solved board, user information is logged into the database |  |  |
| 6 | 01, 02, 03, 04, 05, 06 | **\*Multiple games played\***  Launch and execute multiple games as different users.  **\*Attempt multiple invalid username entries\*** | Scoreboard should reflect an accurate number of historical players and respective game data.  **\*Invalid names are not allowed, Scoreboard not updated until valid entry made\*** |  |  |