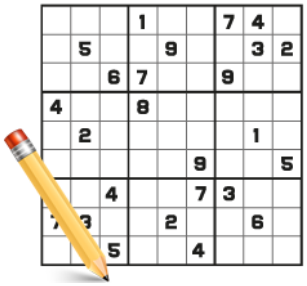

# CSIS 604 Distributed Systems

## Homework Assignment 1 **(INDIVIDUAL WORK!!)**

## Sudoku – Due February 21 in Class

**THE IDEA:** You must build **(from scratch)** an online Sudoku game. A distributed programming approach must be taken such that:

- Frontend JavaScript code is used for the user interaction with the game.
- Backend PHP code is used to generate the board states, determine difficulty levels, etc. JSON or XML is used to communicate between JavaScript and PHP.
- A MySQL database server asynchronously stores game state information as the user plays.

**GAME PLAY:** Your MySQL database only stores one Sudoku game at a time – you can manually initiate it with a board state when you build the db. You only have one web page which displays:

- A NEW GAME button to create a new board state along with a level of difficulty dropdown (1-5 where 1 is the easiest and 5 is the hardest.) When the button is pressed, a new board is randomly generate by PHP, stored in the database (wiping out the existing one), and displayed. If a difficulty level of "1" is chosen, then in the each 3x3 block on the board, 1 tile should be left blank. If difficulty is "2" then 2 tiles should be left blank, etc.
- The current state of whichever Sudoku game is being stored in the database.
    - When a user clicks on a blank board space, they should be able to select/enter a value. They should not be able to change existing/correct values. The game should accept the value if it is correct, but reject it otherwise and not allow them to put an incorrect value in a space. AJAX must be used for backend communication.
    - As the user enters correct values, the state of the board should be automatically stored (AJAX) in the MySQL database. (A user should be able to close the browser and pull up the game on another computer and see exactly where they left off).
- A HINT button should be present which when clicked simply fills in a random space with the correct value. AJAX is used for all backend calls.
- A LAST MOVE MADE AT <div> or <span> should display (at all times) the time stamp for the last time the board state was updated/stored on the database.

All code needs to be uploaded in **one .zip** file to Dropbox on OAKS, and your program must be demonstrated in class to the instructor. The grading rubric is as follows:

| | |
|---|---|
| 10% | Front end HTML webpage that shows all parts of the above requirements. |
| 15% | JavaScript is working and correctly listens to events and triggers AJAX requests to the backend |
| 10% | PHP correctly generates random boards states |
| 10% | Difficult levels working when generating board states |
| 5% | MySQL database has been designed and correctly stores a board state |
| 10% | PHP writes a board state to the MySQL database |
| 5% | Hint button 100% correctly working |
| 5% | New Game button 100% correctly working |
| 5% | Time stamp <div> or <span> is working |
| 20% | Automatic update of game state to database working |
| 5% | Front end design is professionally done with slick color scheme, pictures, etc. |

**-100%** **Any suspected dishonesty: copying code from online resources, fellow students, etc. You must figure this out yourself and code it yourself. Submissions will be crossed checked and also run through an online plagiarism detection tool.**