

GROUP 7 PROJECT ANALYSIS

Sudoku

Group 7 Project Analysis, Rev 10

CMSC 495 (6380) Current Trends and Projects in Computer Science (2205)

University of Maryland Global Campus

Summer 2020 – Professor Hung Dao

July 1, 2020

Group 7

Ben Brandhorst

Christopher Breen

Christopher Smith

Revision	Date	Description	Contributor
1	6/1/2020	Cover page and revision table created	Chris Breen
2	6/2/2020	Possible enhancements list created	Chris Smith
3	6/2/2020	Possible risks and risk mitigations list created	Chris Smith
4	6/2/2020	Input Data, Output Data, Context Diagram, Processing Steps	Chris Breen
5	6/4/2020	Subsystems Diagram	Ben Brandhorst
6	6/5/2020	Subsystem-requirements mapping	Chris Smith
7	6/5/2020	Subsystem descriptions	Chris Smith
8	6/16/2020	Upgraded context diagram based on Stephanie Lightner peer review; refined newgame.html verbiage based on Zachary Finnegan peer review	Chris Breen
9	6/24/2020	Removed about.html and associated references, removed about.html from context diagram.	Chris Breen
10	7/1/2020	Added concern for Chrome regarding lack of SameSite attribute in header	Chris Smith

Input Data

The Sudoku web app contains multiple web pages, with different input from the user for each webpage. The “Menu” includes Home, How to Play, Start a New Game, and Leaderboards.

Individual webpage inputs are as follows:

- [index.html](#). Menu
- [howtoplay.html](#). Menu
- [new_game.html](#). Form for users to input their name, desired difficulty level, Menu
- [leaderboard.html](#). Button or link to play any game depicted on the leaderboard; Menu
- [play.html](#). Checkbox to toggle pencil mark display ON or OFF; “Verify Solutions” button to validate the current status of the board. Clicking on an individual cell will open a scratchpad canvas for the user to “erase” or restore pencil marks.

Output Data

Each webpage will display different outputs to the user. They are as follows:

- [index.html](#). Text and graphics welcoming the user to the Sudoku application.
- [howtoplay.html](#). A brief history of the game, rules, and beginner techniques for solving puzzles.
- [newgame.html](#). A form is displayed with a textbox for player name input and radio button options for available difficulty levels.
- [play.html](#). A welcome message, the puzzle board with a partial solution and “pencil marks” in place of missing numbers, along with a checkbox label indicating pencil marks are on. The “Verify Solutions” button will output a message indicating if the player has

made any mistakes thus far, or if the puzzle is complete – if the user solution is correct.

Updates to the puzzle will be shown in response to user changes to the pencil marks.

- [leaderboard.html](#). Statistics of games previously played by all players.

Context Diagram

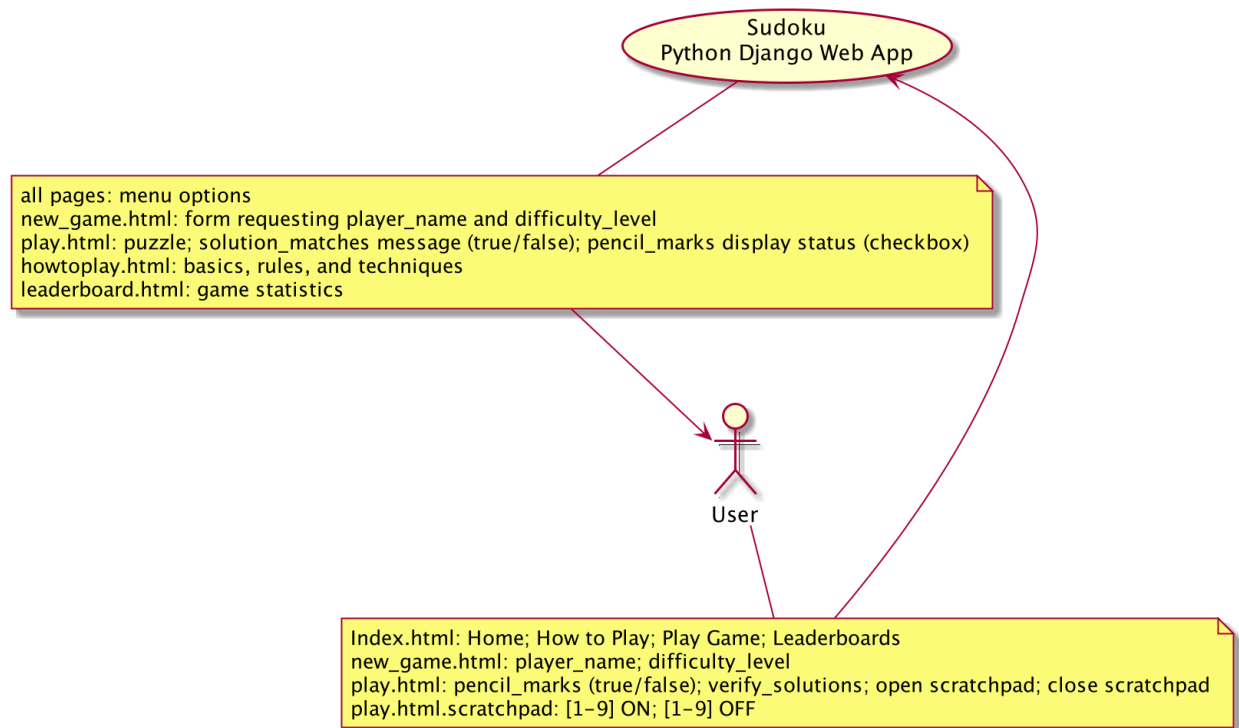


Figure 1 - Context Diagram designed with PlantUML plug-in for IntelliJ

Data Processing Steps (requirement IDs identified in brackets, e.g. [SUD-##])

The user arrives at the home page (index.html) [SUD-5] where they are presented with several menu options: Home (current page) [SUD-5], How to Play (howtoplay.html) [SUD-6], Start a New Game (newgame.html) [SUD-7], Leaderboards (leaderboard.html) [SUD-8], and About Us (about.html) [SUD-19].

If the user clicks How to Play, howtoplay.html is rendered, displaying various text and graphics that include a brief history of Sudoku [SUD-28] and explains strategies [SUD-30, 31]

and basic rules [SUD-29] for playing the game of Sudoku. There are no new inputs on this webpage, only the menu options mentioned above.

If the user clicks Start a New Game, `newgame.html` is rendered, providing additional user input for the player's name and desired difficulty level for the game, as well as a "Create Game" button to begin game generation [SUD-10, 26]. The application shall display a progress indicator [SUD-27]. At the same time, the creation subsystem takes the difficulty level as input and randomly creates a Sudoku solution, then randomly hides cells (quantity also determined by difficulty level), then applies the appropriate solution techniques [SUD-11, 13, 14, 15, 16, 17, 18, 23, 24, 25, 49, 50, 52] based on the user chosen difficulty level in an attempt to solve the game. If the puzzle is solved using only techniques of the associated difficulty level, the puzzle is sent to `play.html` for output [SUD-9, 34]. The solution to the puzzle board is stored in the Django Session [SUD-61] to validate the current state of the game against the solution anytime the user clicks "Verify Solutions" [SUD-61]. The current state of the game shall be maintained in the Django Session, the JavaScript global variable, and in the SQLite3 database [SUD-22] to enable reliable gameplay as well as act as input to the leaderboard. Users will have a checkbox they can toggle, which will show or hide pencil marks [SUD-35]. When a user clicks on a specific cell, a "Scratchpad" will appear [SUD-36] that displays the numbers 1-9, indicating which numbers have already been removed, and which are still included in the pencil marks. The current cell associated with the scratchpad will be uniquely identified [SUD-60]. Clicking on a specific number will toggle its status as included or excluded [SUD-37]. If the user clicks back on the current working cell, the scratchpad will disappear [SUD-38]. If the user clicks a new cell, with or without closing the previous scratchpad, the scratchpad will repopulate the available numbers for the newly clicked cell [SUD-39]. Closing the scratchpad will update the data stores (JavaScript front-end, Django Session backend, and SQLite3 DB) with the current status of the puzzle [SUD-40]. If the puzzle is completed successfully and warrants display on the leaderboard, the user will be prompted to enter, if not previously provided, or confirm their name [SUD-41].

If the user clicks on leaderboard, `leaderboard.html` is rendered, retrieving from the SQLite3 database the date and time, player name if provided by the user, difficulty level, and

the number of hints used, for each of the fastest five puzzles solved for each difficulty level, and outputting to the webpage [SUD-32]. Buttons or links are available for the current user to (re)play any game they see on the leaderboard [SUD-33]. Regardless of leaderboard status, all games played will be stored in the persistent SQLite3 database to track game usage statistics by the owners of the application [SUD-42].

Subsystem Diagram

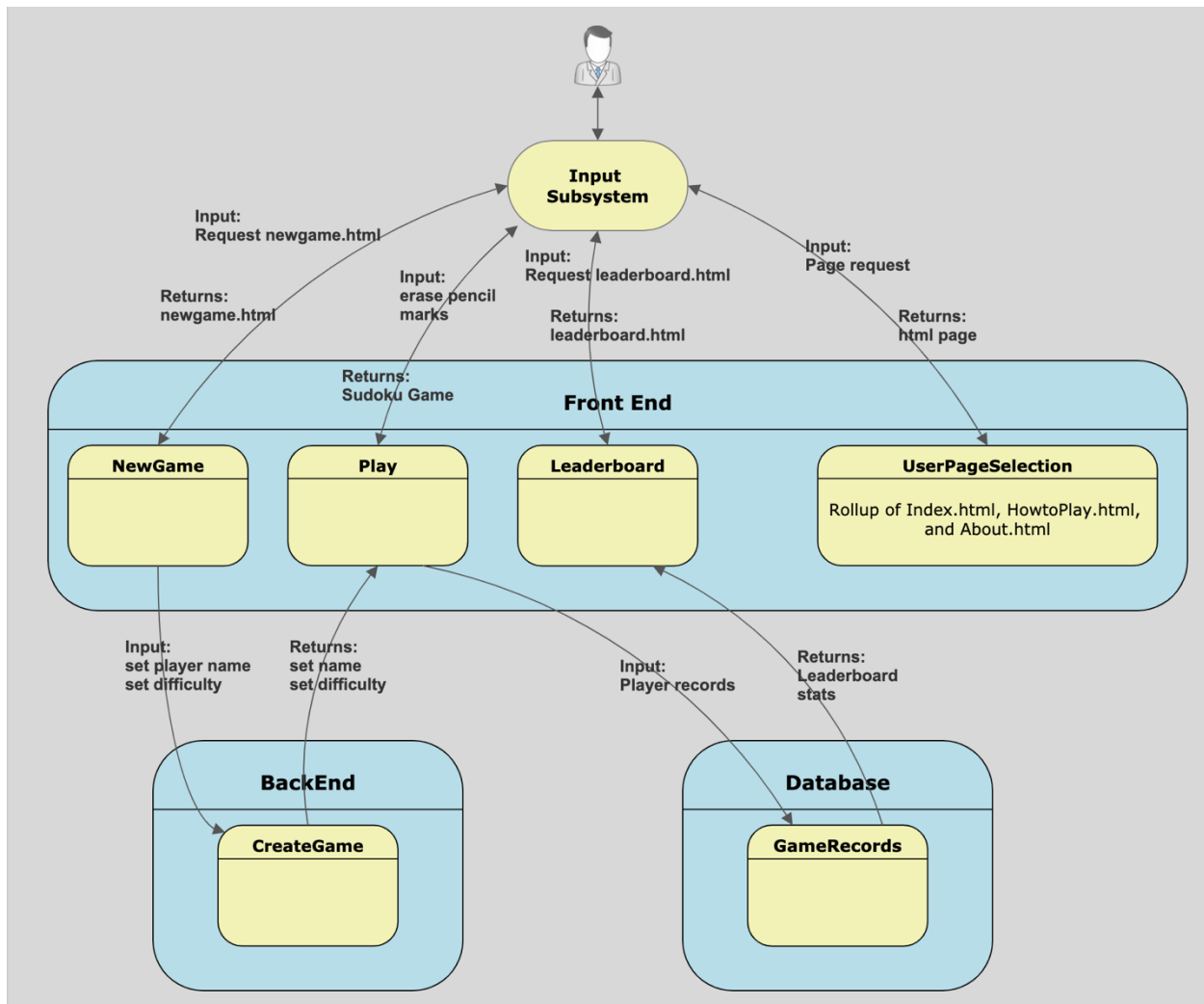


Figure 2 - Subsystem Diagram designed with SmartDraw

Subsystem Descriptions

- a. The input subsystem takes input data from the user involving new game requests, erasing of pencil marks, leaderboard page requests, and index page requests.
- b. NewGame subsystem receives input from the Input subsystem for new game requests and accepts input from the user for player name and difficulty level, which is then sent to the CreateGame subsystem, a backend component.
- c. CreateGame subsystem receives player name and difficulty level from the NewGame subsystem and sends this data to the Play subsystem to display the specified game.
- d. Play subsystem receives set username and difficulty level from the CreateGame subsystem and returns the Sudoku game itself, allowing input for pencil marks from the user to play the game. Additionally, the Play subsystem sends player records to GameRecords subsystem to store pertinent information for the board being played.
- e. GameRecords subsystem receives player records from the Play subsystem and sends leaderboard statistics to the Leaderboard subsystem.
- f. The leaderboard subsystem receives leaderboard statistics from GameRecords and allows the user to view the current leaderboard through the Input subsystem.
- g. UserPageSelection subsystem operates as a directory, taking page requests from the Input subsystem and returning the chosen HTML page.

Subsystem/Requirement Mapping

Requirement #	Subsystems
SUD-5	UserPageSelection
SUD-6	Input, UserPageSelection
SUD-7	Input, UserPageSelection
SUD-8	Input, UserPageSelection
SUD-9	NewGame, CreateGame, Play
SUD-10	NewGame, CreateGame
SUD-11	NewGame, CreateGame
SUD-13	NewGame, CreateGame
SUD-14	NewGame, CreateGame
SUD-15	NewGame, CreateGame
SUD-16	NewGame, CreateGame
SUD-17	NewGame, CreateGame

Requirement #	Subsystems
SUD-18	NewGame, CreateGame
SUD-19	Input, UserPageSelection
SUD-22	GameRecords, Leaderboard
SUD-24	NewGame, CreateGame
SUD-25	NewGame, CreateGame
SUD-26	Input, NewGame
SUD-27	NewGame, CreateGame
SUD-28	UserPageSelection
SUD-29	UserPageSelection
SUD-30	UserPageSelection
SUD-31	UserPageSelection
SUD-32	GameRecords, Leaderboard
SUD-33	GameRecords, Leaderboard
SUD-34	Play
SUD-35	Play
SUD-36	Play
SUD-37	Play
SUD-38	Play
SUD-39	Play
SUD-40	Play, GameRecords
SUD-41	Play, GameRecords
SUD-42	Play, GameRecords
SUD-49	NewGame, CreateGame
SUD-50	NewGame, CreateGame
SUD-52	NewGame, CreateGame
SUD-60	Play
SUD-61	NewGame, CreateGame

Possible Enhancements

1. Add a button to allow the user to reset either the entire board.
2. Add a listener (right-clicking, for example) that reset pencil marks in individual cells.
3. Add links to tutorial videos that show the user techniques to solving more difficult boards.
4. Add a button that fills in remaining cells, operating as a surrender option that forfeits leaderboard placement for the user.
5. Add a difficulty option for (a) timed mode(s) that fails once an arbitrary amount of time has passed.
6. Add a button to display the number of hints remaining.

7. Add a toggle that displays a timer beneath the board that reflects the amount of time that has elapsed since the board was created.

Possible Risks and Risk Mitigation

1. The web application may be vulnerable to CSRF attacks between various pages. To remedy this, an anti-CSRF token should be implemented per session to prevent malicious transmission of code.
2. Vulnerability to SQL injection. Mitigating this risk involved using parameterized queries of whitelisted characters to ensure that the database is not compromised.
3. Vulnerability to general code injection in username textbox. Akin to risk #2, whitelisting characters are vital in preventing a would-be malicious user from injecting code through the textbox. While credential safeguarding is not an issue in this application, performance issues may arise without preventing unexpected character entry. Risk mitigated.
4. Chrome users may encounter an issue regarding cross-site cookies that lack a SameSite attribute related to the application's LinkedIn badges. This warning stems from recent configuration changes on Chrome's end pertaining to how cookies are handled between different websites.