

Software Requirements Specification

[The underlined statements in boldface are the changes/additions. The few parts removed can be cross-checked with the initial (Stage 1) project outlay or in the status of completion.]

In short, the program shall solve and generate a standard 9x9 Sudoku Grid using some basic logical and guessing (whenever required) techniques in a fluid and intuitive GUI. Thus the problem is split into two linked but different parts –

A. In case of solving a user-given incomplete Grid:

1. If an invalid Sudoku grid, i.e. one which is unsolvable due to inconsistencies in the given digits, an appropriate error message explaining the user the problem will be displayed.
2. Saving and loading a puzzle from a file will also be implemented.
3. This functionality can be used for checking a Grid sourced from elsewhere or to get some hints into its solving.

B. In case of generating a random, new and incomplete Grid for the user to solve:

1. To make it more interesting and give it a slight twist, instead of a numerical Sudoku, various pictures – such as logos of popular brands, images of fruits, etc. – will replace numbers if the user wants so (can be done manually easily by replacing the stock images with desired ones).
2. A timer will also be present for the user to track his speed and proficiency in solving.
3. Again, hints will be provided for the user to get some further insight into the puzzle solving.

It shall generate only valid Sudokus or in other words, grids with a single solution only. Further, the Sudoku generated is a new, fresh, original Sudoku grid worthy of publishing (even rotationally symmetrical) from a sample space of 12.8 crore (precisely 12,80,24,064) possible grids (constraint by the generating algorithm).