

CSC470: Software Engineering Final Report
TESS: The Extraordinary Sudoku Solver

Team: #1
David Koval and Joseph Mammo
Instructor: Dr. Ahyoung Lee

4/24/17
v1.1

Table of Contents

1	Summary	3
2	Introduction	3
2.1	Purpose	3
2.2	Scope	3
2.3	Definitions, acronyms, and abbreviations	3
3	Goals	3
4	Specific Requirements	3
4.1	Functional Requirements	3
4.2	Non-Functional Requirements	6
4.3	System Model	6
4.3.1	Actors	6
4.3.2	Use Cases	7
4.3.3	Use Case Diagram	8
4.3.4	Prototyping	8
4.3.5	Traceability Matrix	8
5	System Design	8
5.1	Design Overview	9
5.2	Realistic Constraints and Professional Standards	9
5.2.1	Constraints	9
5.3	Alternative Designs and Design Choices	10
6	System Implementation	10
7	System Testing	11
7.1	Test Plan	11
7.2	Test Results	11
8	Conclusions	11

1 Summary

High level summary with 1 page. the project goals, motivation or problem issues (requirements). Design considerations and choices to solve the problems to achieve the goals. Implementation, validation and testing plans.

!!!

2 Introduction

Overall introduction

2.1 Purpose

2.2 Scope

2.3 Definitions, acronyms, and abbreviations

Term	Definition
DESC	Description
Grid	This is where all the values are stored for the player to see.
ID	Identification
PW	Priority Weights
User	Whoever will be using the app

3 Goals

This is the goals section.

4 Specific Requirements

This is the specific requirements section.

Be sure to include numbering scheme including Identifier (RQ1, RQ2, and RQn) and PW (the priority weights, may be the highest priority = 5 and the lowest priority = 1) to allow traceability.

Provide a high-level use case diagram for the high-level system models and a traceability matrix for the requirement validation.

4.1 Functional Requirements

ID:R1

TITLE: Play Game Option

DESC: The user first opens up the app, they should be able to choose the option to play a Sudoku puzzle. They user should be able to stay as long as they want to on this screen.

PW: 3

How to add
todo

ID:R2

TITLE: Select Difficulty

DESC: The user should be able to select a difficulty setting that better suites needs at any time. There should be 5 difficulty options for the user to choose from. 1 being the easiest all the way down to 5 being the hardest.

PW: 2

ID:R3

TITLE: Back Option

DESC: The user should be able to return to the main screen from the select difficulty screen if they don't select a difficulty. The user can remain on the select difficulty as long as they want to.

PW: 1

ID:R4

TITLE: Continue Game

DESC: The user should be able to continue a game that has been previously started whether or not the app has been closed. All of the user's input should be saved so they could be brought up again should the user want to continue a game.

PW: 4

ID:R5

TITLE: Get Puzzle

DESC: When the user selects a difficulty, a puzzle with the selected difficulty should be retrieved from the database for the user to be able to play it and enjoy the game.

PW: 2

ID:R6

TITLE: Solver Option

DESC: On started, the user should be able to select the Solver option in the app to go straight to the solver part of the app.

PW: 5

ID:R7

TITLE: Input Sudoku Puzzle To Solve

DESC: The user needs to be able to input any sort of Sudoku puzzle that the user has, with or without any extra numbers the user wishes to input.

PW: 5

ID:R8

TITLE: Check Current Board

DESC: The user should be able to check the status of the current puzzle they are working on. The user should be able to view the inputs that are conflicting

with each other. They should be marked in some way for the user to be able to see them clearly.

PW: 4

ID:R9

TITLE: Delete an Input Value

DESC: The user should be able to delete an input value that they put in previously or a value that the system put in automatically.

PW: 5

ID:R10

TITLE: Clear Board

DESC: The user should be able to clear the entire board easily and effortlessly.

PW: 3

ID:R11

TITLE: Solve Current Board

DESC: The user should be able to have the option to solve the current board that they are working on. It should use the input the system put in as well as the inputs the user decided to add.

PW: 5

ID:R12

TITLE: Play Sudoku Game

DESC: The user should be able to play a Sudoku game by selecting certain boxes and input a value into them.

PW: 4

ID:R13

TITLE: Hint Option

DESC: The user should have the option to get a hint on the current Sudoku game they are playing. The hint should display any correct value on the given board.

PW: 1

ID:R14

TITLE: Download Mobile Application

DESC: The user should be able to download the mobile application either from the Play Store or via Email to their Android phone. The download should be free.

PW: 5

4.2 Non-Functional Requirements

ID:RQ1

TITLE: System Availability

DESC: The system needs to be available to the user 99.9% of the time, whether or not the system is being used. The system can only be down for 0.1% of the time for updates or maintenance.

ID:RQ2

TITLE: Solve Time

DESC: The app should be able to solve a given Sudoku puzzle, whether there is or isn't a solution in under 5 seconds.

ID:RQ3

TITLE: Search Algorithm

DESC: The Suduko solver must utilize some search algorithm such as depth first search or breadth first search or any other searching algorithm.

ID:RQ4

TITLE: Pruning Technique

DESC: There must be some sort of pruning technique with solving the problem because 9^{81} is not feasible.

ID:RQ5

TITLE: Database Storage

DESC: The Database needs be stored locally on the device and new puzzles should be added to the device through updates or pulled from an external database.

ID:RQ6

TITLE: Easy to Use

DESC: The app should be very simple to use to any player with some general idea of how Sudoku puzzles work.

ID:RQ7

TITLE: Private Information

DESC: The app should not keep any personal information about the user. There app should not ask the user for any passwords.

4.3 System Model

4.3.1 Actors

- Player: This is the primary actor who will be using the app.

- Database: This is the datastore that on the device that contains all the Sudoku puzzles for the user to play.

4.3.2 Use Cases

ID:UC1

TITLE: Select Play Game

ACTORS: Player

DESC: Player opens up the app and selects to play a Sudoku puzzle.

ID:UC2

TITLE: Select Solver

ACTORS: Player

DESC: Player opens up the app and selects the Solver option to solve a Sudoku puzzle they might have.

ID:UC3

TITLE: Selecting a difficulty

ACTORS: Player and Database

DESC: Player selects a difficulty from the options available and a Sudoku board is filled with a puzzle similar to the difficulty the Player selected.

ID:UC4

TITLE: Quitting Time

ACTORS: Player and Database

DESC: When Player decides to quit the puzzle they are currently working on, the puzzle is saved to the database so Player can continue playing it.

ID:UC5

TITLE: Continuing

ACTORS: Player and Database

DESC: When Player decides to play a game again and selects the continue option to bring up the previously saved game they were playing.

ID:UC6

TITLE: Give me a hint

ACTORS: Player

DESC: The Player is unsure of any other moves, so Player decides to select the hint option in order to receive a hint about the current Sudoku puzzle they are working on.

ID:UC7

TITLE: Solve the Puzzle

ACTORS: Player

DESC: The Player is tired of playing and wants to see the solution to the Sudoku puzzle, so they decided to tap the solve button to get all the answers.

ID:UC8

TITLE: Clear the Board

ACTORS: Player

DESC: The Player decides to start a new puzzle, so the Player hits the clear button to erase everything on the board.

ID:UC9

TITLE: Playing the game

ACTORS: Player

DESC: The Player enjoys playing a Sudoku game by clicking the buttons and entering the numbers that they might think will solve the puzzle.

4.3.3 Use Case Diagram**4.3.4 Prototyping****4.3.5 Traceability Matrix**

REQs	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9
R1	3	X		X						
R2	2			X						X
R3	1			X	X	X				
R4	4			X	X	X				
R5	2			X						
R6	5		X							X
R7	5							X		X
R8	4						X	X		X
R9	5									X
R10	3								X	X
R11	5							X		X
R12	4						X	X	X	X
R13	1						X			X
R14	5	X	X							
Max PW		5	5	4	4	4	4	5	4	5
Total PW		8	10	12	5	5	9	18	7	32

Figure 1: Traceability Matrix of the TESS System.

5 System Design

This is the system design section.

5.1 Design Overview

Provide an overview of the design, including diagrams, key design subsections, and how they relate or connect to one another (e.g., Interaction, structural models).

5.2 Realistic Constraints and Professional Standards

Identify and discuss realistic constraints on the problem, such that constraints may include economic, environmental, social, ethical, health and safety, manufacturability, policy issues, etc.

5.2.1 Constraints

ID:C1

TITLE: Memory Space

DESC: The app should have a small footprint. The entire app should take up less than 5MBs of memory, this is including with future updates as well.

ID:C2

TITLE: Internet Connection for Updates

DESC: The app must have an internet connection established to be able to retrieve updated information from a master database.

ID:C3

TITLE: Programming Language

DESC: The app must be written in mainly Java, there may be other languages within the program, but the main part of it should be in Java.

ID:C4

TITLE: Android Device Only

DESC: The system must work on Android Devices, specifically the Galaxy Note 5 first, the following versions will include other phone models.

ID:C5

TITLE: Time to Solve

DESC: The app should take less than 5 seconds to output the solved Sudoku puzzle or report that it can't be solved.

ID:C6

TITLE: Database Usage

DESC: The app must utilize some sort of database to hold the stored puzzles.

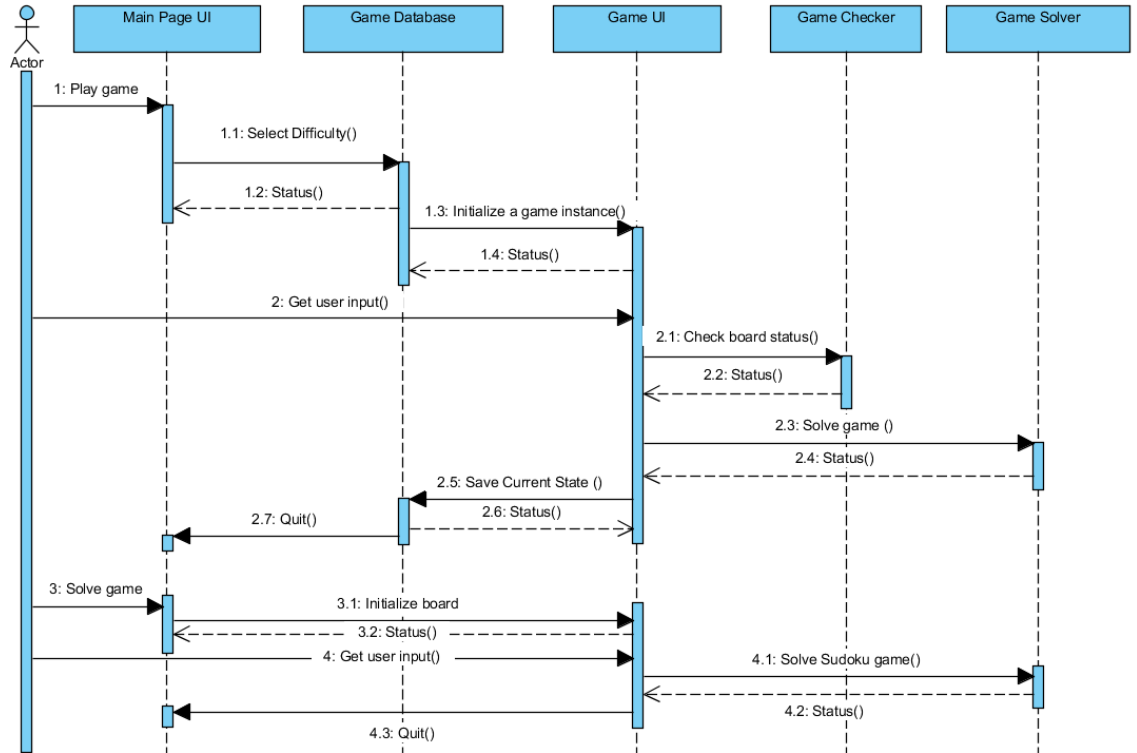


Figure 3: Sequence diagram of a system.

7 System Testing

This is the system testing section.

7.1 Test Plan

Provide your test plan with unit testing (Black-box testing and White-box testing), integration testing (Top-down or bottom-up approach) and system testing.

7.2 Test Results

Show your test results and evaluate them.

8 Conclusions

This is the conclusions section.

Overall summary of design methodologies, key creative approaches and potential contribution/impact. [1]

References

- [1] J. Mammo, “Hahahahaha,” *Computer Networks*, vol. 71, pp. 1–30, 2017.