SOFTWARE REQUIREMENTS SPECIFICATION

JOHN CONWAY'S GAME OF LIFE

UMMAR IKRAM (19L-1049)

NAWAL ALI (19L-1048)

UMAR FAROOQ (19L-1045)

JUNAID AFZAL (19L-0962)

HUMA KARIM (19L-0927)

ABDUL MANNAN (19L-1080)

FAST-NUCES LAHORE
SOFTWARE DESIGN & ANALYSIS

1. INTRODUCTION

CATEGORY: DESKTOP APPLICATION

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, audience, scope, references, and overview of the SRS. This document aims to gather/analyze and give an in-depth insight of the complete **John Conway's Game of Life software system** by defining the problem statement in detail. The detailed requirements of the software system are provided in this document.

1.1 PURPOSE

The purpose of this project is to build an interactive desktop application based on the mathematical rules defined by Cambridge mathematician John Conway in his article. The project is part of the semester coursework at FAST-NU for the course: **Software Design & Analysis**.

1.2 INTENDED AUDIENCE AND READING SUGGESTIONS

This project is a desktop prototype for John Conway's Game of Life web application and it is restricted within the academic premises. This has been implemented under the guidance of college professor(s).

1.3 PROJECT SCOPE

The project will focus primarily on the concepts of cellular automation. The goal will be to follow the best programming and design practices to ensure maximum learning. The project will greatly benefit its developers in terms of getting a glimpse of how software is developed in the real world.

1.4 Definitions, Acronyms, and Abbreviations

Cellular Automation	A collection of "colored" cells on a grid of specified shape that evolves through a number of discrete time steps according to a set of rules based on the states of neighboring cells.
FAQ	Frequently Asked Questions
CRUD	Create, Read, Update, and Delete are the four basic operations of persistent storage.

1.5 REFERENCES

- Play John Conway's Game of Life
- John Horton Conway

2. FUNCTIONALITY

This subsection contains the requirements for the project. These requirements are organized by the features discussed in the classroom. Features from the document are then refined into a use case diagram and sequence diagram to best capture the functional requirements of the system.

1. **Manually Select/Deselect Cells (Ummar Ikram)**
   User can select or deselects cell(s) on grid by clicking on them to create different patterns.
2. **Start (Umar Farooq)**
   User can start the game cycle. It stores the reset/initial state in the database/file and then starts to apply all the rules on the current state and increments counter after every cycle.
3. **Next (Abdul Manan)**
   User can get to the next cycle by applying all the rules on the current state. It also increments counter by 1.
4. **Stop (Junaid Afzal)**
   User can stop the game loop cycle and counter from incrementing.
5. **Reset (Junaid Afzal)**
   User can reset the game cycle by loading reset/initial state from database/file to the grid and resetting the counter back to 0.
6. **Save state (Abdul Manan)**
   User can save the current state on the grid in a database/file using this function.
7. **Load state (Umar Farooq)**
   User can load any previously saved states from the database/file to the grid.
8. **Speed control (Huma Karim)**
   User can control the speed of the game cycle by adding appropriate delay in rendering.
9. **Grid Zoom (Ummar Ikram)**
   User can zoom in/out of the grid by changing individual cell size rendered on the grid.
10. **Counter (Huma Karim)**
    User can use this to keep track of the cycle/generation passed until the initial state.
11. **View saved states (Mohammad Nawal Ali)**
    User can see the previously saved states by loading them from the database/file.
12. **Delete saved states (Mohammad Nawal Ali)**
    User can delete any previously saved states from the database/file.
13. **Clear (Mohammad Nawal Ali)**
    User can clear the grid by removing all selected cells.