**YOUR LOGO**

**Connect Four Game**

**Siddhesh Malvankar**

**26-Nov-19**

**Version 1.0.0**

| REVISION HISTORY | | | |
| --- | --- | --- | --- |
| DATE | VERSION | DESCRIPTION | AUTHOR |
| 26-Nov-2019 | 1.0.0 | Technical Specification for Connect Four Game | Siddhesh Malvankar |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[Connect Four Game 1](#_Toc25682318)

[Siddhesh Malvankar 1](#_Toc25682319)

[26-Nov-19 1](#_Toc25682320)

[Version 1.0.0 1](#_Toc25682321)

[INTRODUCTION 3](#_Toc25682322)

[PURPOSE 3](#_Toc25682323)

[INTENDED AUDIENCE AND PERTINENT SECTIONS 3](#_Toc25682324)

[PROJECT SCOPE 3](#_Toc25682325)

[DOCUMENT CONVENTIONS 3](#_Toc25682326)

[REFERENCES 3](#_Toc25682327)

[DESCRIPTION 4](#_Toc25682328)

[PRODUCT PERSPECTIVE 4](#_Toc25682329)

[FEATURES 4](#_Toc25682330)

[USER OVERVIEW 4](#_Toc25682331)

[OPERATING ENVIRONMENT 4](#_Toc25682332)

[CONTRAINTS: IMPLEMENTATION / DESIGN 4](#_Toc25682333)

[DOCUMENTATION 4](#_Toc25682334)

[Api End-to-End Testing with Swagger 6](#_Toc25682335)

[ASSUMPTIONS / DEPENDENCIES 7](#_Toc25682336)

[REQUIREMENTS OF EXTERNAL INTERFACE 8](#_Toc25682337)

[USER INTERFACES 8](#_Toc25682338)

[HARDWARE INTERFACES 9](#_Toc25682339)

[SOFTWARE INTERFACES 10](#_Toc25682340)

[COMMUNICATION INTERFACES 10](#_Toc25682341)

[ADDITIONAL NONFUNCTIONAL REQUIREMENTS 11](#_Toc25682342)

[PERFORMANCE 11](#_Toc25682343)

[SAFETY 11](#_Toc25682344)

[SECURITY 11](#_Toc25682345)

[SOFTWARE QUALITY 11](#_Toc25682346)

[APPENDICES 12](#_Toc25682347)

[APPENDIX A: GLOSSARY OF TERMS 12](#_Toc25682348)

[APPENDIX B: ANALYSIS DOCUMENTATION 12](#_Toc25682349)

[APPENDIX C: ISSUES 12](#_Toc25682350)

# INTRODUCTION

## PURPOSE

This is technical specification document for Connect Four Game. Connect Four is designed and developed using Microsoft technology stack.

## INTENDED AUDIENCE AND PERTINENT SECTIONS

Any user who is capable of understanding basics of C# programing and JavaScript programming .can leverage this document to under the solution structure of this project.

## PROJECT SCOPE

Simple game to be designed using the best capabilities of .net technologies and demonstrate how modern world applications can be designed.

## DOCUMENT CONVENTIONS

N/A

## REFERENCES

Game uses Min-max algorithm to device the next move against the player.

<https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/>

# DESCRIPTION

## PRODUCT PERSPECTIVE

Game is intended to be played by one player with AI being its opponent. Objective of the Game is to achieve consecutive moves to connect 4 coins.

## FEATURES

This application is build using .NET Web API and Angular app for interactive user experience.

* Solution Architecture follows Clean Architecture Pattern
* Segregation of concerns and dependency injection using Autofac
* Routing techniques of .NET core routes
* Ease of API testing using Swagger

## USER OVERVIEW

Any naïve user can play the game which involves easy moves to enjoy the game.

## OPERATING ENVIRONMENT

Windows Operating System

Software Dependencies

* Microsoft Visual Studio 2017 or Higher
* Microsoft Visual Studio Code
* Node JS
* Angular CLI 8

## CONTRAINTS: IMPLEMENTATION / DESIGN

Limitations involve a single player game playing against AI

## DOCUMENTATION

Project followed architecture pattern of **Clean Architecture.**

The key rule behind Clean Architecture is: **The Dependency Rule**. The gist of this is simply that dependencies are encapsulated in each "ring" of the architecture model and these dependencies can only point inward.

.NET Solution includes 2 projects

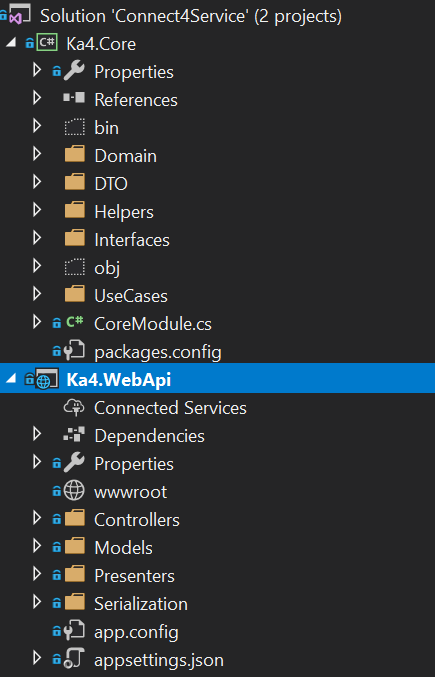
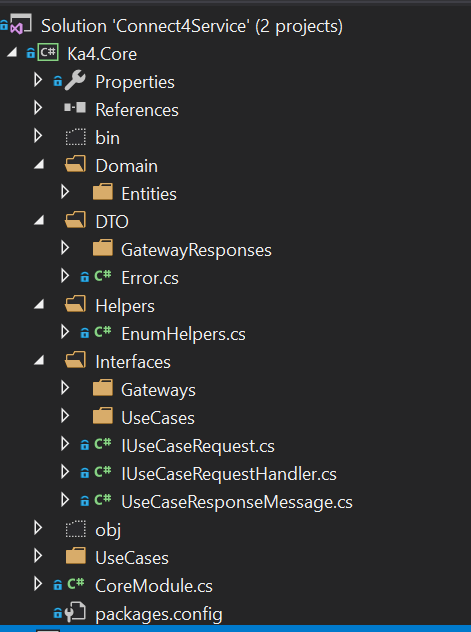
**Web API**

Layer responsible for interaction in the form or Api or Presenters. In the context of our API, this means it accepts input in the form of http requests over the network (e.g., GET/POST/etc.) and returns its output as content formatted as JSON.

**Core**

Maps to the layers that hold the Use Case and Entity concerns and is also where our External Interfaces get defined. These innermost layers contain our domain objects and business rules. The code in this layer is mostly pure C# - no network connections, databases, etc. allowed. Interfaces represent those dependencies, and their implementations get injected into our use cases.

**Folder Structure**

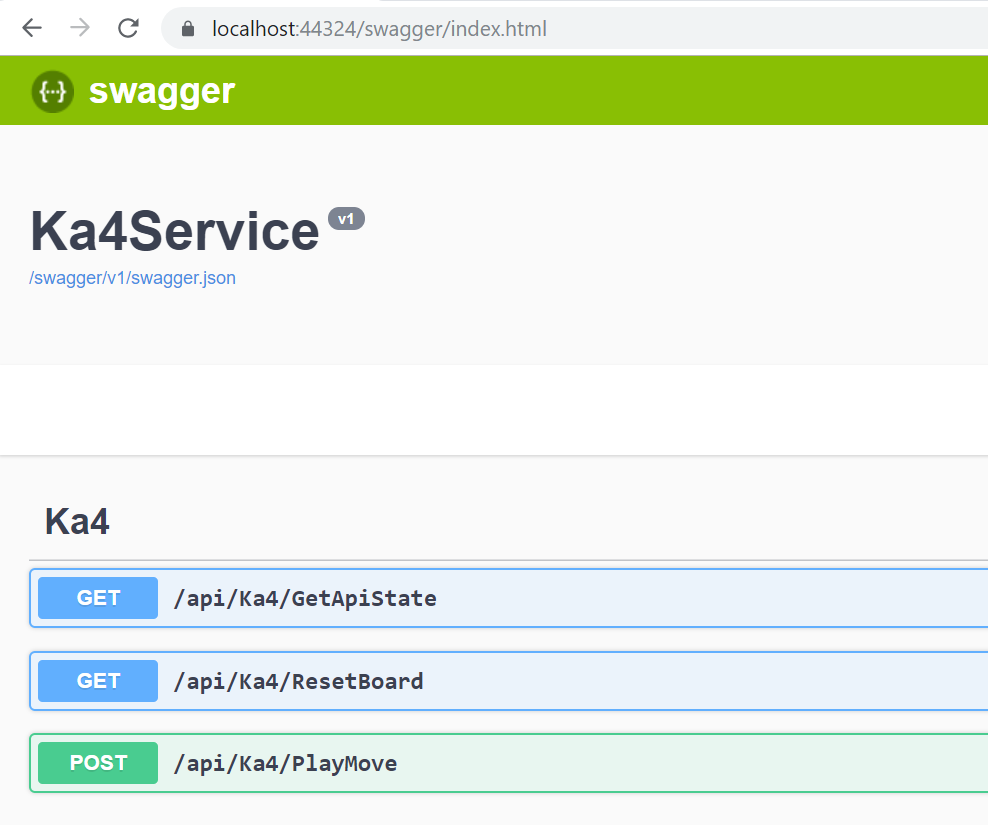


**Dependency Injection Using Autofac**

Plays a pivotal role in our Clean Architecture by enforcing the Dependency Inversion Principle across the application.Autofac helps to wire up dependency amount individual project and to register them altogether in the startup services. Modules provide a very nice way to orgranize your dependencies on a per-project basis when working with Autofac.

### **Api End-to-End Testing with Swagger**

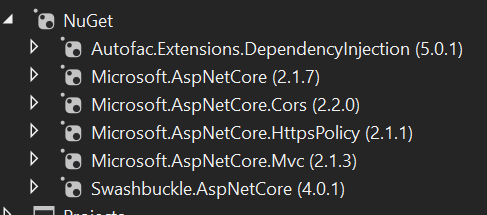
Swagger tool helps test API has we develop. Web API project is integrated with swashbuckle package to integrate swagger.



Please follow below steps to run the program

1. Complete project is available on GitHub repository at <https://github.com/siddheshmalvankar/Connect4Game.git>
2. Download “Connect4Service” and “connect4UI” directories for .net Web API and angular project respectively
3. Open Connect4Service.sln in VS 2017 and let the project restore packages.
4. Run the project which will host the Web API on localhost port 44324
5. Now open the connect4UI project in Visual studio Code.
6. Run command npm install in the terminal
7. Then run command ng serve which will run the application on the port localhost:4200
8. Web API endpoints are also available on <https://localhost:44324/swagger/index.html>

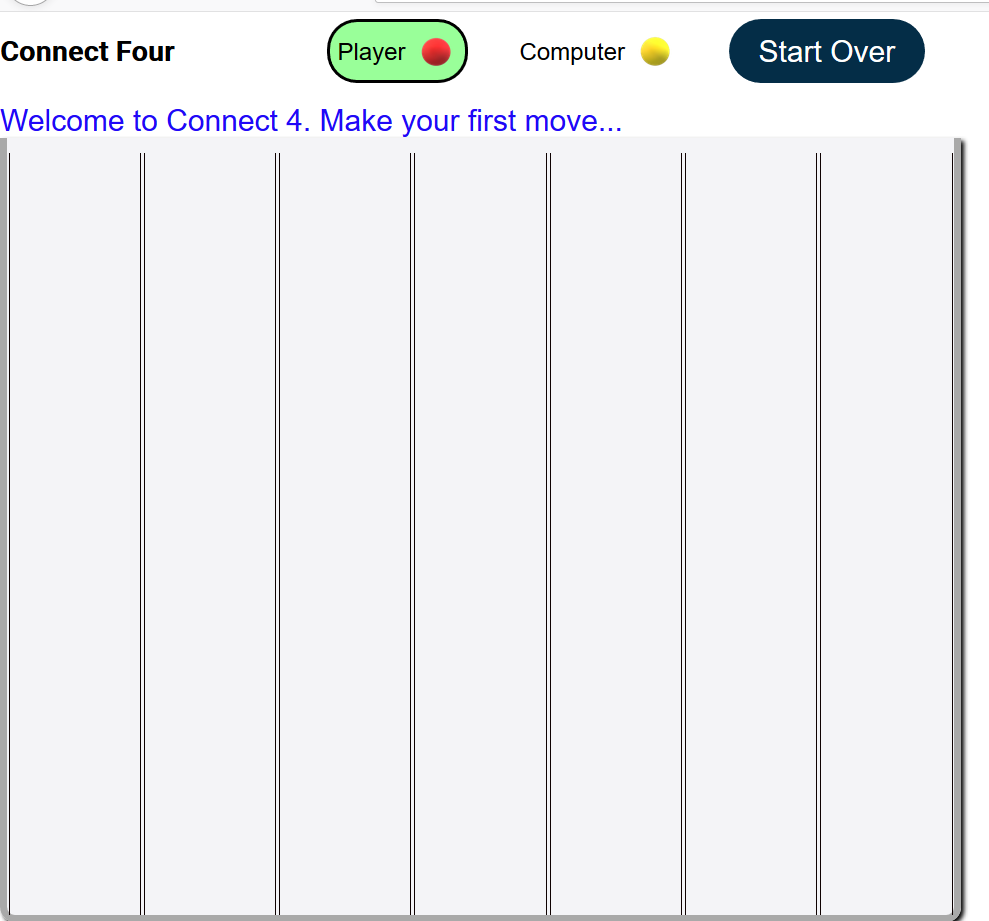
## ASSUMPTIONS / DEPENDENCIES



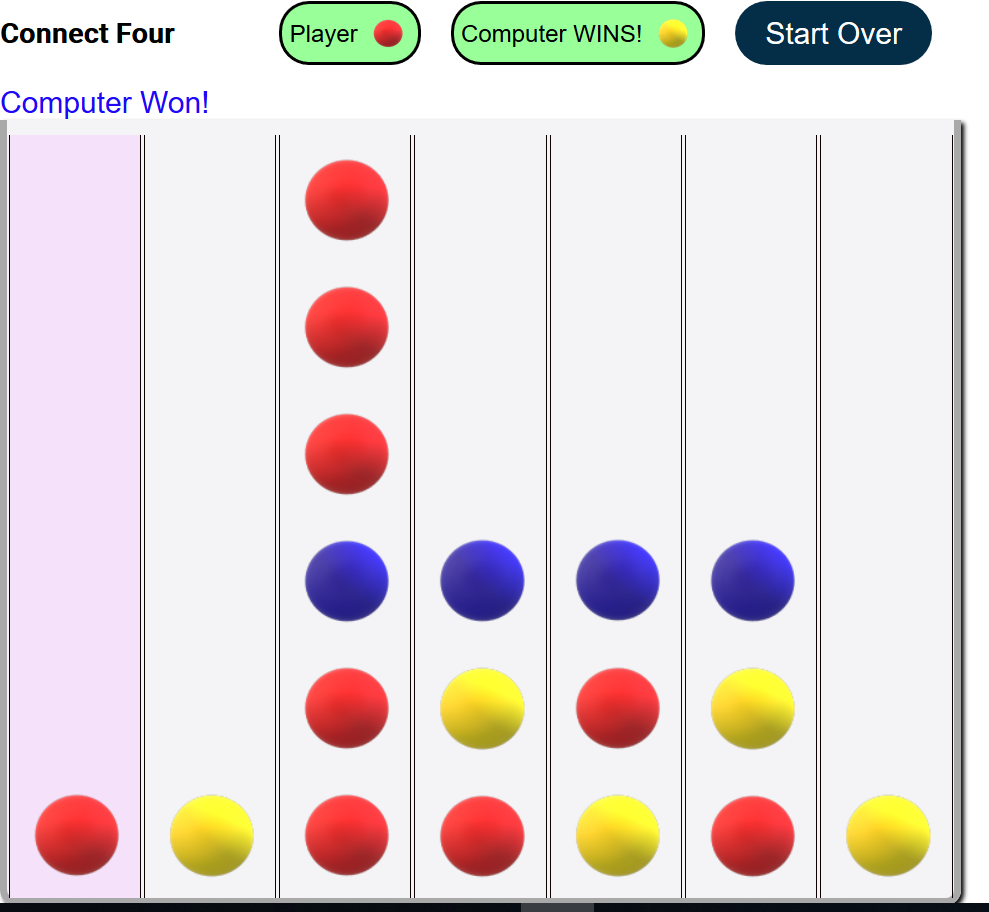
# REQUIREMENTS OF EXTERNAL INTERFACE

## USER INTERFACES

**Game Board When Launched**



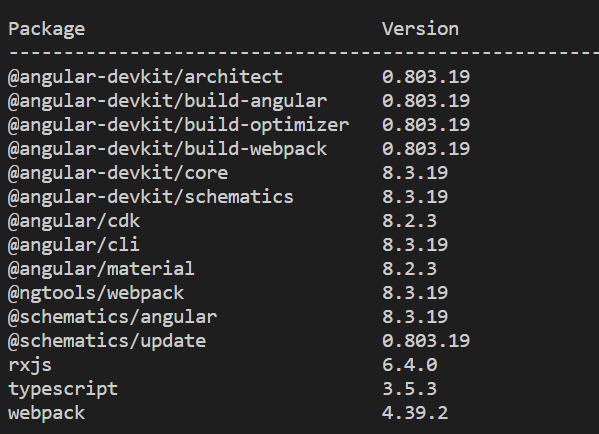
**Computer Won**



## HARDWARE INTERFACES

N/A

## SOFTWARE INTERFACES



## COMMUNICATION INTERFACES

N/A

# ADDITIONAL NONFUNCTIONAL REQUIREMENTS

## PERFORMANCE

N/A

## SAFETY

Valid client certificate is not available so all connection are over http.

## SECURITY

System Exceptions and Security is not fully implemented.

## SOFTWARE QUALITY

N/A

# APPENDICES

## APPENDIX A: GLOSSARY OF TERMS

N/A

## APPENDIX B: ANALYSIS DOCUMENTATION

N/A

## APPENDIX C: ISSUES

List all unresolved issues, TBDs, pending decisions, findings required, conflicts, etc.

| ISSUES | | |
| --- | --- | --- |
| ID | DESCRIPTION | PARTY RESPONSIBLE |
| 1 | Currently board doesn’t represent winning moves for diagonal positions. |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |