# Document de Spécifications Techniques

## Projet Fil Rouge Dev Avancé : Jeu de la Vie

### Table des Matières

Table des matières

[Document de Spécifications Techniques 1](#_Toc157157032)

[Projet Fil Rouge Dev Avancé : Jeu de la Vie 1](#_Toc157157033)

[Table des Matières 1](#_Toc157157034)

[1. Introduction 1](#_Toc157157035)

[2. Objectifs du Projet 1](#_Toc157157036)

[3. Fonctionnalités et Spécifications 2](#_Toc157157037)

[4. Architecture Technique 2](#_Toc157157038)

[5. Plan de Réalisation 3](#_Toc157157039)

[6. Bonnes Pratiques 3](#_Toc157157040)

### 1. Introduction

Le projet "Jeu de la Vie" est une initiative de développement avancé visant à implémenter une version automatisée du célèbre automate cellulaire conçu par John Conway. Cette application web permettra aux utilisateurs de simuler le Jeu de la Vie en se basant sur des configurations initiales stockées dans une base de données.

### 2. Objectifs du Projet

Fournir une interface utilisateur pour la connexion et la gestion de profils.

Implémenter les règles du Jeu de la Vie avec la possibilité de récupérer et modifier les configurations initiales.

Assurer la maintenabilité et l'évolutivité de l'application.

Les cellules vivantes initiales seront placés de manière aléatoire sur la grille à chaque nouveau lancement.

### 3. Fonctionnalités et Spécifications

#### 3.1 Connexion Utilisateur

Authentification sécurisée et gestion de sessions.

Profils utilisateurs avec configurations personnalisées du Jeu de la Vie.

#### 3.2 Règles du Jeu

Simulation basée sur des règles prédéfinies avec des cellules vivantes et mortes.

Interactions selon les règles classiques du Jeu de la Vie (naissance, survie, mort).

#### 3.3 Base de Données

Stockage des configurations initiales du jeu et des profils utilisateurs.

Modélisation des données permettant une récupération et une mise à jour efficaces.

### 4. Architecture Technique

#### 4.1 Choix Technologiques

Backend : .NET Core pour la logique métier et la gestion de la base de données.

Frontend : Dans une premier temps affichage console en .NET puis éventuellement évoluer par la suite sur une interface Angular () pour une interface utilisateur dynamique et réactive.

Base de Données : SQL Server pour le stockage et la gestion des données.

#### 4.2 Structure de la Base de Données

Tables pour utilisateurs/mdps et configurations de jeu.

#### 4.3 Sécurité

Hashage des mots de passe avec salage pour une meilleure sécurité.

Utilisation de HTTPS pour sécuriser les communications.

### 5. Plan de Réalisation

#### 5.1 Planning de Développement

Phase 1 : Conception et modélisation (0.75j) -> Préparation du contenu de la V1 + Rédaction de la V1 du document des specs techniques

Phase 2 : Développement Backend (2.5j) -> 0.5j logique du jeu + 1.5j pour la gestion des sockets et des threads + 0.5 implémentation de l’ORM

Phase 3 : Développement Frontend (1j) -> Affichage + Connexion au back

Phase 4 : Intégration et tests (2j) (pas pour V1)

#### 5.2 Liste des Tâches

Conception de la base de données

Implémentation de l'authentification

Développement de la logique du jeu

Création de l'interface utilisateur

Mise en place des tests unitaires (pas pour V1)

#### 5.3 Tests Unitaires

Tests pour chaque composant et fonctionnalité clé.

Utilisation du framework nUnit pour le back .NET. (pas pour V1)

### 6. Bonnes Pratiques

#### 6.1 Gestion de Version avec GitHub

Utilisation de GitHub pour le versionnage du code et la collaboration.

Intégration continue pour automatiser les tests et le déploiement.

#### 6.2 Documentation

Documentation technique et fonctionnelle mise à jour régulièrement.

#### 6.3 Sécurité et Confidentialité

Salage des mdps

Interdiction à un tiers d’avoir accès au resultat d’une partie (pas pour V1)