|  |
| --- |
| [Nom de la société] |
| DOCUMENTATION TECHNIQUE |
| ConnectionToLife |

|  |
| --- |
| FLORENT Victor  [Date] |

Table des matières

[Mise en contexte et expression des besoins 2](#_Toc160707740)

[Liste des besoins 2](#_Toc160707741)

[Solution proposée 3](#_Toc160707742)

[Tâches et planning 4](#_Toc160707743)

[Monétisation 5](#_Toc160707744)

[Installer et utiliser l’interface 6](#_Toc160707745)

[Installation 6](#_Toc160707746)

[Utilisation 6](#_Toc160707747)

[Nota Bene 7](#_Toc160707748)

# Mise en contexte et expression des besoins

Dans le contexte du cours de Développement Logiciel Avancé, un client nous demande de concevoir un projet composé du Jeu de la Vie de Conway, amélioré par un chat multijoueur, avec la possibilité de composer une simulation et de changer les règles avec les autres joueurs.

## Liste des besoins

|  |  |
| --- | --- |
| Besoin | Priorité |
| Simuler une partie du Jeu de la Vie de Conway selon des règles personnalisées | Essentielle, mais n’est pas la priorité du projet |
| S’authentifier | Haute |
| Discuter avec d’autres personnes connectées | Moyenne |
| Proposer et voter une nouvelle règle | Haute |

NB : La création de compte ne sera pas présente dans le livrable, cela se fera directement en base de données.

# Solution proposée

La solution sera divisée en plusieurs parties :

* Une partie serveur qui se charge de transmettre la communication entre les utilisateurs par le biais de sockets.

TODO : Justifier

* Une API Rest se chargeant de l’authentification et de la simulation du Jeu de la Vie

TODO : Justifier -> Sécurité et simplicité

* Une interface WPF utilisable par le client

Une base de données MongoDB est utilisée pour fournir la liste des utilisateurs. Les mots de passe sont hachés et salés. Nous utilisons une base de données NoSQL comme nous n’avons pas besoin de relations entre données. Nous utilisons MongoDB Atlas afin d’avoir un accès simple à la base de données, sans avoir à faire d’installation particulière.

# Tâches et planning

Lot 1 : Jeu de la Vie individuel

1. Création et génération d’une grille pour simuler le Jeu de la Vie (1/4 jour)
2. Générer une itération du jeu (à partir de règles personnalisées) (1/4 jour)
3. Afficher le résultat d’une itération (1/8 jour)

Lot 2 : Début de connexion multi-utilisateurs

1. Pouvoir s’authentifier (1/2 jour)
2. Pouvoir chatter avec d’autres utilisateurs (1 jour)

Lot 3 : Jeu de la Vie multijoueur

1. Partager l’état de la grille à la fin d’une simulation (1 jour)
2. Proposer une nouvelle règles aux autres utilisateurs (1/2 jour)
3. Voter pour confirmer la nouvelle règle (1/2 jour)

Comme une seule personne travaille sur ce projet, ces tâches seront faites dans cet ordre.

Total en jours pour un développpeur : 4+1/8 jours

# Monétisation

Freemium ou par abonnement

Vente du projet et mises à jour : 500€

Connexion : 1€

1000 Itérations : 1€

# Installer et utiliser l’interface

## Installation

Le client obtiendra une application WPF. Ce logiciel sera connecté à une application serveur ainsi qu’une API.

## Utilisation

Connexion (login-mdp) : admin-admin ou test-sosis

Le bouton start/stop permet de démarrer/arrêter la simulation, en fonction de la règle choisie.

Les règles possèdent le schéma suivant : ##A##D, # étant des nombres.  
Les nombres avant A représentent le nombres de voisins qu’une cellule vivante doit avoir pour rester vivant. Les nombres avant D représentent la même chose, mais pour une cellule morte, afin de naître.

L’écran à droite permet de dialoguer avec les autres utilisateurs.

Pour proposer une nouvelle règle, envoyez « RULE=# », # étant la nouvelle règle (voir le format ci-dessus)

TODO : Scénarios pour tester fonctionnalités, mettre le reste en README

# Nota Bene

Dans la release, l’ensemble des projets sont trouvables puisqu’il n’y a pas de déploiement de fait. Dans un vrai scénario, le client n’aurait que le nécessaire pour démarrer l’exécutable : il ne devrait pas avoir accès au serveur ni à l’API Rest.