

KATA-MANGA

Table des matières

1	Analyse préliminaire	3
1.1	Introduction	3
1.2	Objectifs	3
1.3	Planification initiale	4
2	Analyse / Conception	4
2.1	Concept	4
2.2	Stratégie de test	4
2.3	Risques techniques	4
2.4	Planification	4
2.5	Dossier de conception	5
3	Réalisation	5
3.1	Dossier de réalisation	5
3.2	Description des tests effectués	6
3.3	Erreurs restantes	6
3.4	Liste des documents fournis	6
4	Conclusions	6
5	Annexes	7
5.1	Résumé du rapport du TPI / version succincte de la documentation	7
5.2	Sources – Bibliographie	7
5.3	Journal de travail	7
5.4	Manuel d'Installation	7
5.5	Manuel d'Utilisation	7
5.6	Archives du projet	7

NOTE L'INTENTION DES UTILISATEURS DE CE CANEVAS:

Toutes les parties en italiques sont là pour aider à comprendre ce qu'il faut mettre dans cette partie du document. Elles n'ont donc aucune raison d'être dans le document final.

De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n'aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l'alourdir inutilement.

1 Analyse préliminaire

1.1 Introduction

kata-manga est un concept de projet spécialement conçu pour l'entraînement des apprentis aux TPI. Mon projet consistera d'un Backend en .NET C# avec un Swagger et une base de donnée en MySQL, d'un Frontend en React Typescript avec Tailwind CSS pour le style des pages, le tout sera conteneurisé à l'aide de Docker.

Le site répertorie une liste des 100 animes les plus populaires sur My anime list. Il y sera possible d'ajouter, modifier, supprimer, et voir des mangas. Le site servira donc aux utilisateurs de pouvoir chercher et voir les détails d'un manga en spécifique et de créer une nouvelle entrée quand un nouvel anime sera sorti.

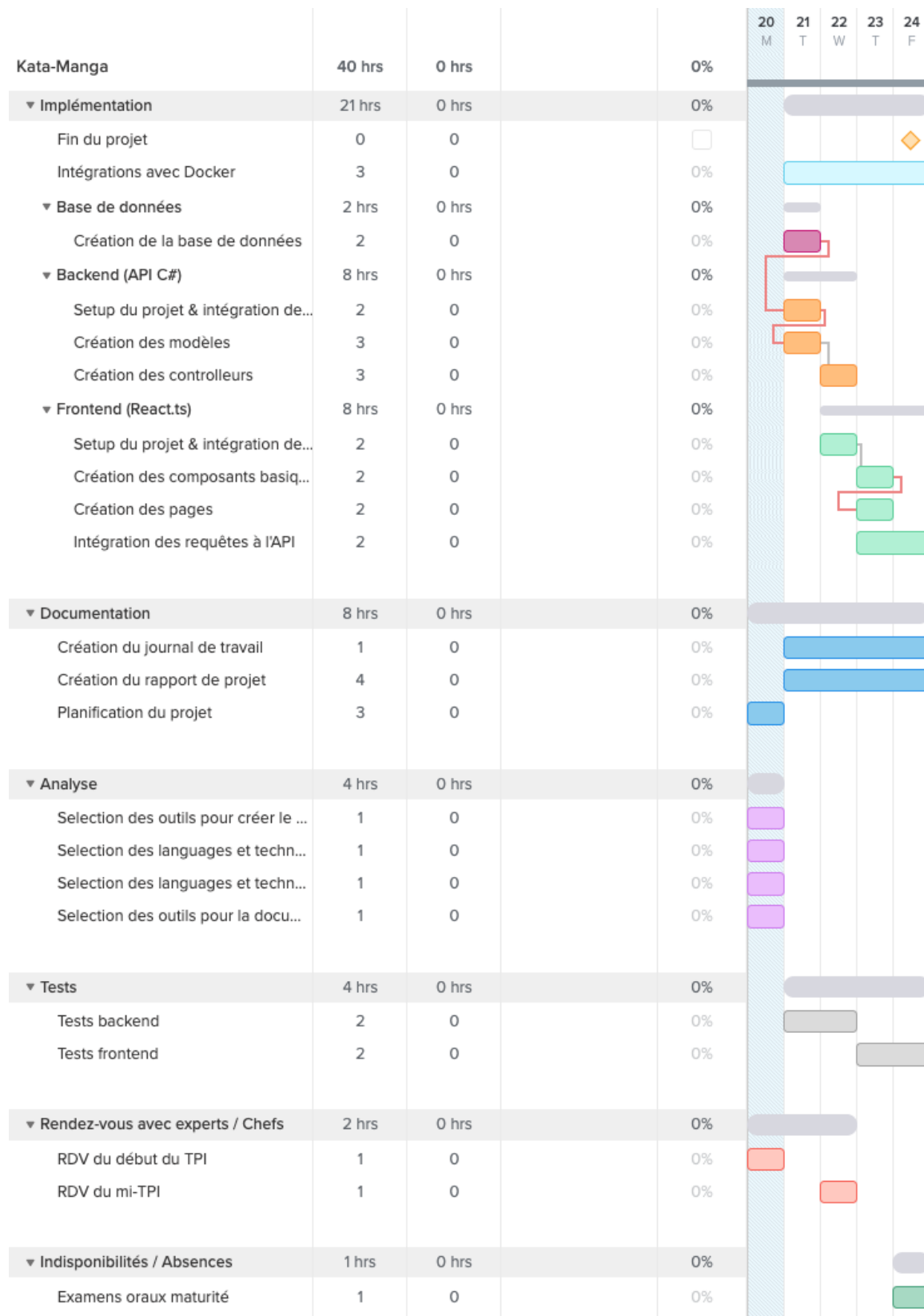
Pour ce projet de TPI, la méthode de projet Agile **Scrum for One** va être utilisée. Il s'agit d'une version de la méthode de gestion de projet Scrum adaptée pour les développeurs travaillant seul sur des projets courts.

1.2 Objectifs

Créer un site web avec un Frontend et un Backend fonctionnel. Le Frontend sera capable de pouvoir faire des requêtes au Backend pour effectuer des actions CRUD sur les mangas dans la base de données.

Le projet doit pouvoir être lancé à l'aide de docker, et donc avec une seule commande.

1.3 Planification initiale



2 Analyse / Conception

2.1 Concept

Base de données: MySQL avec pour administrer phpmyadmin, pour la génération du diagramme je vais utiliser Datagrip JetBrains.

Frontend: React Typescript avec Tailwind CSS

Backend: .NET C# avec Swagger

Déploiement/Ops: Docker

Le concept complet avec toutes ses annexes:

Par exemple :

- *Multimédia: carte de site, maquettes papier, story board préliminaire, ...*
- *Bases de données: interfaces graphiques, modèle conceptuel.*
- *Programmation: interfaces graphiques, maquettes, analyse fonctionnelle...*
- *...*

2.2 Stratégie de test

Décrire la stratégie globale de test:

- *types de des tests et ordre dans lequel ils seront effectués.*
- *les moyens à mettre en œuvre.*
- *couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).*
- *données de test à prévoir (données réelles ?).*
- *les testeurs extérieurs éventuels.*

2.3 Risques techniques

Il y a un risque de retardement du planning dû à la dockerisation du backend .NET. Ceci est dû à un potentiel manque de compétence de ma part, étant donné que je n'ai encore jamais conteneurisé un backend .NET pour des projets tout au long de mon apprentissage.

- *risques techniques (complexité, manque de compétences, ...).*

Décrire aussi quelles solutions ont été appliquées pour réduire les risques (priorités, formation, actions, ...).

2.4 Planification

Révision de la planification initiale du projet :

- *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
- *partage des tâches en cas de travail à plusieurs.*

*Il s'agit en principe de la planification **définitive du projet**. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l'historique.*

2.5 Dossier de conception

Fournir tous les document de conception:

- *le choix du matériel HW*
- *le choix des systèmes d'exploitation pour la réalisation et l'utilisation*
- *le choix des outils logiciels pour la réalisation et l'utilisation*
- *site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, ...*
- *bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.*
- *programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme...*

Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !

3 Réalisation

3.1 Dossier de réalisation

Décrire la réalisation "physique" de votre projet

- *les répertoires où le logiciel est installé*
- *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
- *les versions des systèmes d'exploitation et des outils logiciels*
- *la description exacte du matériel*
- *le numéro de version de votre produit !*
- *programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

NOTE : Évitez d'inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n'incluez que cette partie...

3.2 Description des tests effectués

Pour chaque partie testée de votre projet, il faut décrire:

- *les conditions exactes de chaque test*
- *les preuves de test (papier ou fichier)*
- *tests sans preuve: fournir au moins une description*

3.3 Erreurs restantes

S'il reste encore des erreurs:

- *Description détaillée*
- *Conséquences sur l'utilisation du produit*
- *Actions envisagées ou possibles*

3.4 Liste des documents fournis

Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions

- *le rapport de projet*
- *le manuel d'Installation (en annexe)*
- *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
- *autres...*

4 Conclusions

Développez en tous cas les points suivants:

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

5 Annexes

5.1 Résumé du rapport du TPI / version succincte de la documentation

5.2 Sources – Bibliographie

<https://www.lucidchart.com/blog/scrum-for-one#:~:text=Scrum%20for%20one%3A%20A%20tutorial%20on%20adapting%20Agile%20Scrum%20methodology%20for%20individuals&text=Many%20teams%2C%20particularly%20in%20software,products%20faster%2C%20and%20continuously%20iterate.>

Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)... Et de toutes les aides externes (noms)

5.3 Journal de travail

Date	Durée	Activité	Remarques

5.4 Manuel d'Installation

5.5 Manuel d'Utilisation

5.6 Archives du projet

Media, ... dans une fourre en plastique