|  |
| --- |
| Plot That Line |

Belkhiria Sofiène

Table des matières

[1 Analyse préliminaire 3](#_Toc175917886)

[1.1 Introduction 3](#_Toc175917887)

[1.2 Objectifs 3](#_Toc175917888)

[1.3 Gestion de projet 4](#_Toc175917889)

[2 Analyse / Conception 4](#_Toc175917890)

[2.1 Gameplay 4](#_Toc175917891)

[2.2 Concept 4](#_Toc175917892)

[2.3 Analyse fonctionnelle 4](#_Toc175917893)

[2.4 Stratégie de test 4](#_Toc175917894)

[3 Réalisation 4](#_Toc175917895)

[3.1 Points de design spécifiques 4](#_Toc175917896)

[*3.1.1* *…* 5](#_Toc175917897)

[*3.1.2* *…* 5](#_Toc175917898)

[*3.1.3* *…* 5](#_Toc175917899)

[3.2 Déroulement 5](#_Toc175917900)

[3.3 Mise en place de l’environnement de travail 5](#_Toc175917901)

[3.4 Description des tests effectués 6](#_Toc175917902)

[3.5 Erreurs restantes 6](#_Toc175917903)

[4 Conclusions 6](#_Toc175917904)

[5 Annexes 7](#_Toc175917905)

[5.1 Manuel de référence 7](#_Toc175917906)

[5.2 Journal de travail 7](#_Toc175917907)

*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.*

# Analyse préliminaire

## Introduction

Le projet Plot That Line se déroule en parallèle du module I323 et a pour but pédagogique l’utilisation du paradigme fonctionnel avec la librairie LinQ en C#.

Le projet en lui-même consiste à développer une application graphique d’analyse de donnée en C#, le jeu de donné et la technologie d’affichage sont quant à eux au choix.

## Objectifs

* Développement d’une application d’analyse des statistiques NBA.
* Elaboration d’un fil rouge de développement à base de user stories de qualité.
* Mise en place d’un historique de développement du projet fiable via des commits bien nommé et atomique.
* Mise en place d’un historique du temps via un journal de retraçant le temps passé sur les différents pants du projet
* Développement de maquette basse fidélité de l’application
* Elaboration d’une documentation expliquant le déroulement les objectifs ainsi que le fonctionnement du projet.

## Gestion de projet

La gestion de ce projet se fera avec les outils suivants :

* IceScrum pour l’élaboration des user stories.
* GitHub/desktop afin de suivre l’évolution du code ainsi que son accessibilité.
* Excel pour le journal de travail.
* Word pour la création du rapport de projet.

A noté que la planification se fera aussi à l’aide des différentes user stories.

# Analyse / Conception

## Domaine

Les données choisies sont les statistiques de base de chaque match NBA de la saison 2023-2024, ces statistiques sont :

* La date du match.
* L’heure de début du match local au lieu du match.
* L’équipe Visiteuse.
* Les points de l’équipe visiteuse.
* L’équipe à domicile.
* Les points de l’équipe à domicile.

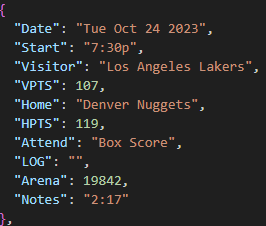
Ces données seront donc affichées sur une échelle d’une semaine chaque jour représentant la moyenne de point par match que l’équipe marque sur l’ensemble de la saison pour le jour spécifique.

Ces données pourront alors être utiliser pour :

* Des tentatives de prédiction de résultats par les différents organismes de paris.
* Les équipes techniques des différentes équipes NBA afin de mieux se préparer pour les matchs.
* Les analystes de statistique NBA avancé afin d’émettre des théories quant à la raison des résultats des équipes sur l’ensemble de la saison.

Le programme devrait lire les données de différentes manières :

* Via des requêtes API renvoyant une réponse au format JSON
* Via la lecture d’un fichier JSON ayant des clefs identiques que le canevas ci-dessous.

**

A noté que les champs : « Attend, LOG, Arena, Notes » ne sont pas obligatoire.

## Concept

* *Diagramme de classe*
* *Diagramme(s) d’état*

## Analyse fonctionnelle

***Reprendre le contenu des User Stories d’IceScrum : Story + tests d’acceptance (avec IceTools) + maquettes***

## Stratégie de test

Afin de tester et de vérifier au mieux le fonctionnement de l’application plusieurs tests ont été mis en place tels que :

* Test unitaire afin de vérifier la logique du code.
* Test par les paires afin de vérifier la solidité du code.
* Test par des proche non informaticien afin de vérifier l’affordance et la solidité de l’application en cas d’utilisation inattendue.

La stratégie de test à été accès sur les 3 fonctionnalité les plus important du programme qui sont :

* La génération du graphique.
* L’importation des données via un fichier. json adéquat.
* Les différents filtres du graphique

Chaque type de test permets de vérifier ces points de différentes manières.

1. Génération du graphique.
   1. Le test unitaire si dessous vérifie que le graphique générer n’est pas vide (mettre screen du test).
   2. Le test par les paires à permis de vérifier que peu importe combien de fois on charge le même fichier le graphique ne change pas.
   3. Le test par des utilisateurs a permis de tester et d’assurer le bon fonctionnement du graphique malgré des zoom et dézoome continue.
2. Importation des données via un fichier.
   1. Le test unitaire ci-dessous vérifie que la classe DataSelection.cs ne soit pas vide une fois que la méthode LoadData ait été effectuée. (Mettre screen).
   2. Le test par les paires a permis d’assurer la gestion d’erreur du code dans le cas où les clefs du fichier. json ne correspondent pas à celle attendue ou que les valeurs de ses clef ne soient pas du bon type.
3. Filtres du graphique.
   1. Le test unitaire ci-dessous permet de vérifier que la listes des jours afficher dans le graphique soit équivalente à la liste des jours devant être afficher dans les filtres (Mettre screen).
   2. Les différents tests par les paires ont testé la cohérence d’affichage entre le filtre des équipes et du filtre sur jours (Capacité des deux types de filtre à être activer en même temps).
   3. Les différents tests réaliser par les utilisateurs ont permis d’identifier certains bugs liés à l’utilisation de plusieurs filtres.

*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 ?)* ***et comment elles seront mises en place****.*
* *les testeurs extérieurs éventuels.*

# Réalisation

## Points de design spécifiques

***Ce chapitre est constitué de plusieurs sous-chapitre.***

***Chaque sous-chapitre explique un point de design technique particulier, quelque chose que vous avez dû inventer pour répondre au besoin et qui ne peut pas s’expliquer par de simples commentaires dans le code.***

***Il s’agit d’explications techniques sur le fonctionnement du système. Les explications sont appuyées par des diagrammes, ou de très brefs éléments de code.***

***NE PAS mettre ici des pratiques usuelles que tout professionnel de la branche connaît déjà. Par exemple, n’EXPLIQUEZ PAS ICI CE QU’EST LE PATTERN MVC.***

***Exemple (simplifié à l’extrême) : Protection contre des formulaires mal intentionnés ou modifiés***

* ***Au moment de générer le formulaire, le script php :***
  + ***Concatène les noms de tous les champs contenus dans le formulaire***
  + ***Calcule un hash SHA256 de la chaîne obtenue***
  + ***Ajoute un input nommé « CSRF » de type hidden dans le form***
* ***A la réception du POST du fromulaire***
  + ***Concatène les noms des indices de $\_POST***
  + ***Calcule un hash SHA256 de la chaîne obtenue***
  + ***Vérifie que la valeur du champ CSRF correspond***

### *…*

### *…*

### *…*

## Déroulement

***Résumer comment s’est passé la réalisation de chaque story, ses difficultés, les alternatives envisagées mais rejetées, ses surprises, …***

## Mise en place de l’environnement de travail

* ***Comment accéder au code source***
* *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*

***Ce chapitre décrit précisément comment un employé qualifié peut recréer l’environnement dans lequel vous avez effectué ce travail***

## Description des tests effectués

***Reprendre les tests d’acceptance d’IceScrum au moyen de la feuille ad hoc d’IceTools***

## Erreurs restantes

*S'il reste encore des erreurs :*

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

***Reporter la*** [***dette technique***](https://www.premaccess.com/qu-est-ce-que-dette-technique-comment-la-maitriser/#:~:text=La%20dette%20technique%20survient%20quand,de%20plus%20en%20plus%20fr%C3%A9quents.) ***connue. S’appuyer sur la pratique des // TODO***

# Conclusions

*Développez en tous cas les points suivants :*

* *Objectifs atteints / non-atteints*
* *Points positifs / négatifs*
* *Difficultés particulières*
* *Suites possibles pour le projet (évolutions & améliorations) (Graphique en fonction de : l’heure de début de match, le nombre de fuseau horaire traversé par l’équipe visiteuse, si l’équipe joue en prime time)*

# Annexes

## Journal de travail