

# **Journal de médiation**



Table des matières

1	Analyse préliminaire .....	3
1.1	Introduction .....	3
1.2	WiseJ.net.....	3
1.2.1	Caractéristiques principales.....	3
1.2.2	Avantages.....	3
1.2.3	Utilisation typique.....	3
1.3	Outils Utilisé.....	4
1.4	Objectifs.....	4
1.5	Planification initiale .....	5
2	Analyse préliminaire .....	6
2.1	Introduction .....	6
2.2	Critères d'ergonomie (Bastien et Scapin) .....	6
2.3	Cryptographie .....	7
2.4	Stratégie de test.....	8
2.5	Risques techniques .....	8
2.6	Planification .....	8
2.7	Dossier de conception .....	8
3	Réalisation.....	9
3.1	Dossier de réalisation .....	9
3.2	Description des tests effectués .....	9
3.3	Erreurs restantes .....	9
3.4	Liste des documents fournis .....	9
4	Conclusions .....	10
5	Annexes.....	11
5.1	Résumé du rapport du TPI / version succincte de la documentation .....	11
5.2	Sources – Bibliographie.....	11
5.3	Journal de travail .....	11
5.4	Manuel d'Installation .....	11
5.5	Manuel d'Utilisation.....	11
5.6	Archives du projet.....	11

# **1 Analyse préliminaire**

## **1.1 Introduction**

Le projet consiste à développer une application Web basé sur WiseJ.net qui permet aux médiateurs de tenir leur journal de médiation de manière simplifiée et centralisée. Cette application met l'accent sur la facilité d'accès notamment par l'utilisation sur différents appareils (smartphone, tablette, ordinateur portable, ordinateur fixe).

Ce projet est destiné aux médiateurs du canton qui souhaitent abandonner leur vieux tableur Excel, avec la possibilité de l'utiliser n'importe où et sur n'importe quelle plateforme.

## **1.2 WiseJ.net**

WiseJ.net est un Framework de développement web basé sur .net, conçu pour permettre aux développeurs de créer rapidement des application web interactives, responsive et complexe en utilisant du C# sans avoir besoin de maîtriser HTML, CSS ou JavaScript.

### **1.2.1 Caractéristiques principales**

- Développement en C# (ou VB.net)
- Interface graphique similaire à WindowsForm
- Cross-platform (fonctionne sur tous les navigateurs modernes)
- Intégration facile avec MySQL
- Responsive natif pour PC, tablette et mobiles

### **1.2.2 Avantages**

- Aucune connaissance approfondie du web nécessaire
- Très rapide à prendre en main pour les développeurs WinForm
- Parfait pour moderniser des applications desktop vers le web
- Beaucoup d'extensions disponible en package NuGet

### **1.2.3 Utilisation typique**

- Application métier (ERP, CRM, outil de gestion interne)
- Dashboard interactif
- Logiciel de saisie ou de consultation de données
- Portail utilisateur/admins

Wisej.NET est un framework .NET moderne qui permet de créer des applications web dynamiques avec la même simplicité que des applications desktop, en C#, tout en offrant un rendu fluide, responsive et professionnel dans un navigateur. Il combine le meilleur du monde WinForms avec la puissance du web.

---

### 1.3 Outils Utilisé

Pour mener à bien ce projet, j'aurai besoin de plusieurs outils :

Application	Utilisation
Visual studio 2022	IDE de développement .net
Github desktop	Logiciel desktop pour pousser les différentes versions du projet
Github	Site Web qui héberge les différentes versions du projet
IceScrum	Gestion de projet
SwissCenter	Site Web qui héberge toutes les composantes de l'application (web, BD, mail)
Azure	Site web qui héberge l'application pour la phase de test

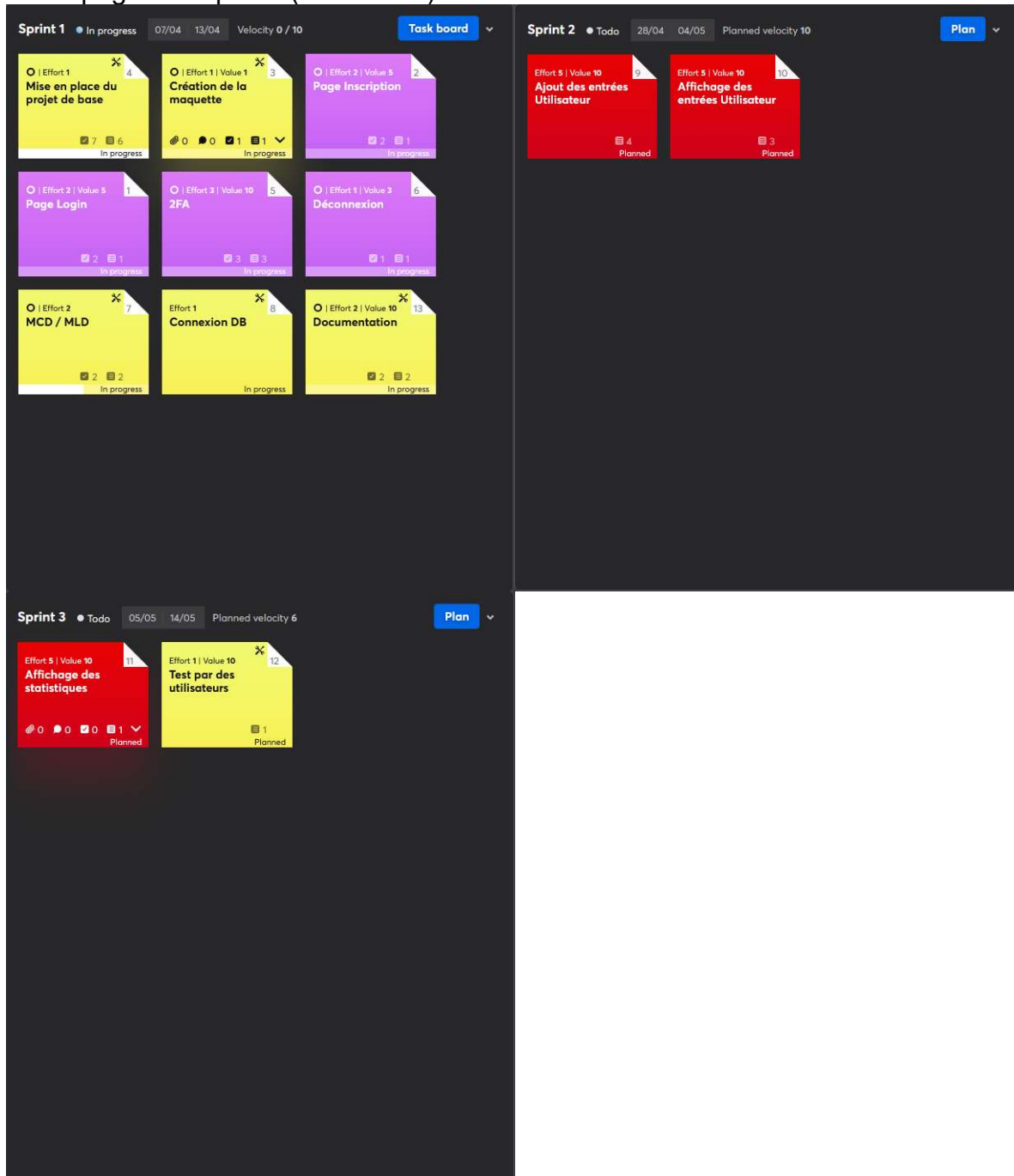
### 1.4 Objectifs

Les objectifs du projet consistent au développement d'une application web qui permet d'entrer des données de rendez-vous des médiateurs. Voici la liste des objectifs défini par le cahier des charges.

- Les données doivent être cryptée
- L'ergonomie de l'application doivent suivre les critères de Bastien et Scapin
- L'application permet de s'enregistrer et de se connecter
- L'application permet d'entrer et de sauver des données
- L'application permet d'afficher des statistiques sur la répartition du temps de travail
- Les cas critiques sont traité comme des entrée erronées (ex :Type de champ)
- L'application est utilisable sur smartphone
- Les données stockées ne sont disponible qu'à l'utilisateur qui les a insérées

## 1.5 Planification initiale

J'ai décidé d'utiliser la méthode agile pour réaliser la planification de ce projet. Voici le découpage des sprints (1 semaine).



Voici les différents sprint goal pour chaque sprint :

Sprint 1 : L'application doit pouvoir gérer l'inscription, la connexion ainsi que toute la partie authentification à 2 facteurs. La base du projet doit être mise en place (IceScrum, repo Git, SwissCenter, base du projet C# avec les implémentations du 1er sprint).

Sprint 2 : L'application doit gérer l'ajout d'entrées utilisateur ainsi que l'affichage de ces entrées dans une liste. Les entrées utilisateur ne doivent être vue uniquement par l'utilisateur concerné.

Sprint 3 : L'application doit pouvoir afficher des statistiques globales des entrées de l'utilisateur. Uniquement les données de cet utilisateur doivent être affichée dans les statistiques.

Durant toute la phase de développement du projet, je vais documenter mon avancement à travers ce document, ainsi qu'à travers mon journal de travail.

## **2 Analyse préliminaire**

### **2.1 Introduction**

Dans cette partie, je vais aborder les différentes analyses et recherches qui m'ont amené à la réalisation du projet. J'ai notamment dû me référer à des critères d'ergonomie (Bastien et Scapin)

### **2.2 Critères d'ergonomie (Bastien et Scapin)**

Les critères d'ergonomes de Bastien et Scapin sont une référence en ergonomie des interface homme-machine. Ils ont été définis en 1993 et sont encore aujourd'hui largement utilisé pour évaluer ou concevoir des interfaces utilisateur efficace et confortables.

Ils se composent en 8 critères principaux avec leurs exemples respectifs :

- Guidage : L'interface doit guider l'utilisateur dans ce qu'il peut ou doit faire
  - o Menus clairs, Boutons explicites, Messages d'aide
- Charge de travail : l'interface doit minimiser les efforts cognitifs et physique
  - o Remplissage automatique, Groupement logique des champs
- Contrôle explicite : L'utilisateur doit garder le contrôle sur ses actions
  - o Confirmation avant suppression, Choix clairs et sans actions cachées
- Adaptabilité : L'interface doit s'adapter aux besoins et préférences des utilisateurs
  - o Mode sombre / clair, Interface responsive (mobile / PC)
- Gestion des erreurs : L'interface doit éviter les erreurs et aider à les corriger facilement
  - o Messages d'erreur clairs, validation de format (email, code)
- Homogénéité / Cohérence : L'interface doit être cohérente dans tout le système
  - o Même design pour tous les boutons, navigation similaire sur tout le site
- Signifiante des Codes et Dénomination : Les icônes, couleurs, labels doivent être compréhensible immédiatement
  - o Poubelle = supprimer, Disquette = enregistrer, Aucun jargon technique pour l'utilisateur

- Compatibilité : L'interface doit respecter les habitudes et attentes de l'utilisateur
  - o Raccourcis clavier connus (Ctrl + S), Position du bouton « OK » à droite

## 2.3 Cryptographie

Le cryptage des données est fondamental de nos jours pour sécuriser des accès à des données sensibles. La cryptographie est la science qui permet de protéger les informations en les rendant illisible pour tout personne non autorisée. Elle est essentielle en informatique pour garantir la confidentialité, l'intégrité, l'authenticité et parfois la non-répudiation des données.

Les 4 grands objectifs :

- Confidentialité : Seuls les destinataires autorisés peuvent lire l'information
- Intégrité : L'information n'a pas été modifiée durant le transport ou le stockage
- Authentification : On peut confirmer l'identité d'un utilisateur ou d'une source
- Non-répudiation : L'auteur d'un message ne peut pas nier l'avoir envoyé

Types de cryptographie :

- Cryptographie symétrique :
  - o Même clé pour chiffrer et déchiffrer
  - o Rapide, mais nécessite de partager la clé secrètement
  - o Exemples : AES, DES
- Cryptographie asymétrique
  - o Une clé publique pour chiffrer
  - o Une clé privée pour déchiffrer
  - o Très utilisé pour l'échange sécurisé de données et des signatures numériques
  - o Exemples : RSA, ECC
- Fonction de hachage
  - o Transforme une donnée en empreinte fixe
  - o Fonction irréversible
  - o Utile pour stocker des mots de passe, vérifier l'intégrité
  - o Exemple : SHA-256, SHA-3, bcrypt

Quelques concepts clé :

- Chiffrement : Rendre un message illisible sans la clé
- Déchiffrement : Rendre un message lisible avec la clé
- Hachage : Résumer une donnée en empreinte unique
- Signature numérique : Garantir l'identité et l'intégrité d'un message
- Sel : Valeur aléatoire ajouté à un mot de passe avant le hachage
- Clé publique / privée : System de clés utilisées pour chiffrer ou signer

Dans mon application, le hachage des mots de passe avec l'ajout de sel serra utiliser pour chiffrer les mots de passes

## 2.4 Stratégie de test

Concernant ma stratégie de test, j'ai décidé de la faire en 2 parties. La première, en testant via mes tests d'acceptation sur IceScrum (sur PC et sur tablette). Puis la seconde en envoyant l'accès à l'application à différents utilisateurs choisis pour me faire un retour concret sur le UI, les bugs ainsi que l'utilisation globale de l'application.

## 2.5 Risques techniques

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

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

## 2.6 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.*

## 2.7 Dossier de conception

*Fournir tous les documents 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 champ) 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 : Evitez 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

*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*