|  |
| --- |
| Shadow Scan |

Pour projets avec XCL

V2.0

Table des matières

[1 Analyse préliminaire 5](#_Toc188610005)

[1.1 Introduction 5](#_Toc188610006)

[1.2 Objectifs 5](#_Toc188610007)

[1.3 Gestion de projet 5](#_Toc188610008)

[1.4 Planification initiale 6](#_Toc188610009)

[2 Analyse / Conception 7](#_Toc188610010)

[2.1 Contexte produit 7](#_Toc188610011)

[2.2 Contextes techniques 8](#_Toc188610012)

[2.2.1 Opérationnel 8](#_Toc188610013)

[*Placer le produit dans son contexte d’opérations.* 8](#_Toc188610014)

[2.2.2 Validation 8](#_Toc188610015)

[2.2.3 Développement 8](#_Toc188610016)

[2.2.4 Justification des choix 8](#_Toc188610017)

[2.3 Concept 9](#_Toc188610018)

[2.4 Analyse fonctionnelle 10](#_Toc188610019)

[2.5 Stratégie de test 10](#_Toc188610020)

[2.6 Risques techniques 10](#_Toc188610021)

[2.7 Planification 10](#_Toc188610022)

[3 Réalisation 11](#_Toc188610023)

[3.1 Points de design spécifiques 11](#_Toc188610024)

[*3.1.1* *…* 11](#_Toc188610025)

[*3.1.2* *…* 11](#_Toc188610026)

[*3.1.3* *…* 11](#_Toc188610027)

[3.2 Déroulement 11](#_Toc188610028)

[3.2.1 Sprints 11](#_Toc188610029)

[3.2.2 Stories 12](#_Toc188610030)

[~~3.3~~ ~~Dossier de réalisation~~ 12](#_Toc188610031)

[3.4 Mise en place de l’environnement de travail 12](#_Toc188610032)

[3.5 Mise en place de l’environnement de test 12](#_Toc188610033)

[3.6 Déploiement du produit 13](#_Toc188610034)

[3.7 Description des tests effectués 13](#_Toc188610035)

[3.8 Bilan 13](#_Toc188610036)

[3.8.1 Erreurs restantes 13](#_Toc188610037)

[3.8.2 Stories 13](#_Toc188610038)

[3.8.3 Dette technique 13](#_Toc188610039)

[3.9 Recours à l’intelligence artificielle 13](#_Toc188610040)

[3.10 Liste des documents fournis 14](#_Toc188610041)

[4 Conclusions 14](#_Toc188610042)

[5 Annexes 15](#_Toc188610043)

[5.1 Résumé du rapport du TPI / version succincte de la documentation 15](#_Toc188610044)

[5.2 Sources – Bibliographie 15](#_Toc188610045)

[5.3 Journal de travail 15](#_Toc188610046)

[~~5.4~~ ~~Manuel d'Installation~~ 15](#_Toc188610047)

[5.5 Manuel d'Utilisation 15](#_Toc188610048)

[5.6 Archives du projet 15](#_Toc188610049)

*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 but de ce projet est de créer un moyen de monitorer les accès à certaines ressources sur les post des élèves.

Lors d’un test ou un travail le prof va pouvoir définir une liste de ressource interdites (site web, app, fichiers).

Si durant la durée du test ou du travail un élève outrepasse les limitations, il en sera notifié et le prof aussi sans que rien se passe. Le but est seulement de vérifier ce qui se passe et non de l’empêcher.

## Objectifs

Objectifs de formation :

Le but de ce projet est de se préparer au TPI, se remettre dans le bain du fonctionnement des projet et de prendre connaissance de la grille d’évaluation du TPI.

Objectifs du produit :

Le but du produit est de pouvoir créer une liste de ressource interdites, de lancer une vérification sur ces ressource interdite.

D’être informé si une de ces ressources est utilisée (notifications des deux côtés : prof et élève)

## Gestion de projet

IceScrum

Agile avec des sprints de 2 semaine

## Planification initiale

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Sprint 1 | | | | | | | | | |
|  |  |
| 27.01.2024 -> 07.02.2024 | |  |  |  |  |  |  |  |  |  |  |
|  | Etablir la connexion gRPC entre les clients (prof et élève) | | | | | | | |  |
|  |  |  |  |
| Semaine 5->6 | |  |  |  |  |  |  |  |  |  |  |
|  | lire / modifier / exporter / importer une bibliothèque sur le pc prof | | | | | | | |  |
|  |  |  |  |
| ~43.5 H | |  |  |  |  |  |  |  |  |  |  |
|  | lire / modifier / exporter / importer sous liste sur le pc prof | | | | | | | |  |
|  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Sprint 2 | | | | | | | | | |
| 10.02.2024 -> 21.02.2024 | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  | Détection des post allumés et si l'agent est allumé | | | | | | | |  |
| Semaine 7->8 | |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  | Scan de l'utilisation de ressource interdites | | | | | | | |  |
| ~43.5 H | |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Sprint 3 | | | | | | | | | |
|  |  |
| 24.02.2024 -> 07.03.2024 | |  |  |  |  |  |  |  |  |  |  |
|  | Envoyer des infos depuis le client (élève) au prof | | | | | | | |  |
|  |  |  |  |
| Semaine 9->10 | |  |  |  |  |  |  |  |  |  |  |
|  | Début / fin du mode test | | | | | | | |  |
|  |  |  |  |
| ~43.5 H | |  |  |  |  |  |  |  |  |  |  |
|  | Lire les dépassements après le test (prof) | | | | | | | |  |
|  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |

# Analyse / Conception

## Contexte produit



***Placer le produit dans son contexte d’utilisation. Par exemple :***

* ***Site Web Internet***
* ***Application, Web, intranet***
* ***Application mobile***
* ***Infrastructure dupliquer des sites multiples***

***Présenter les utilisateurs du système ainsi que les rôles qu’ils peuvent avoir***

***Présenter les données/information que le système prend en charge***

*Au minimum :*

* *Un ou plusieurs schémas de contexte montrant le produit dans son environnement d’utilisation, ainsi que ses utilisateurs. Ce type schéma doit être accompagné d’explications textuelles*

## Contextes techniques

### Opérationnel

Le post maitre est un laptop d’un prof (chacun utilise le sien) le port 55052 ouvert

Le post client (élève) est un poste d’une classe, avec la partie client du code qui tourne comme service non-stop (démarrage automatique)

### *Placer le produit dans son contexte d’opérations.*

***Ce chapitre doit impérativement contenir au moins une représentation graphique***

*Au minimum :*

* *Un ou plusieurs schémas de contexte montrant le produit dans son environnement technique. Ce type schéma doit être accompagné d’explications textuelles*

### Validation

(EN COURS DE DEFINITION)

Le post maitre est un laptop avec le port 55052 ouvert

Le post client (élève) est un poste d’une classe, avec la partie client du code qui tourne comme service non-stop (démarrage automatique) / avec la partie client du code qui tourne comme une simple application non-stop (démarrage automatique)

***Décrire les moyens qui seront mises en œuvre pour faire valider les résultats produits par le product owner***

### Développement

Le post maitre est un post fix avec le port 55052 ouvert et 0 droits admins

Le post client (élève) est un laptop avec un compte admin local (Shadow) qui permet d’exécuter le coté client comme un simple programme.

***Décrire l’environnement dans lequel le projet va être réalisé.***

***Ce chapitre doit permettre à tout nouvel arrivant dans l’équipe de projet d’installer son environnement de travail***

### Justification des choix

Laptop pour le dev car on a besoin des droits admins

Finalement le but est que le prof (la console, la partie de monitoring) puisse monitorer les post des élèves avec le moins d’efforts possible donc il faut que la partie client soit automatique d’où le service qui démarrera automatiquement et le prof s’y connectera et le scan pourra commencer.

***Pour tous les contextes ci-dessus.***

## Concept



*Le concept complet avec toutes ses annexes :*

*Un ou plusieurs schéma conceptuel, accompagné d’explications textuelles, de type :*

* *Modèle conceptuel des données. Expliquer chaque entité/attribut possédant une particularité.*
* *Schémas d’architecture montrant la structure interne du produit*
* *Diagramme de flux*
* *Diagramme d’état.*

*Générez les images à partir des applications utilisées. N’ayez recours à la capture d’écran que s’il n’y a pas d’autre moyen de faire.*

*Par exemple :*

* *Multimédia : carte de site, ~~maquettes papier~~, story board préliminaire, …*
* *Bases de données : interfaces graphiques, modèle conceptuel.*
* *~~Programmation : interfaces graphiques, maquettes, analyse fonctionnelle…~~* ***Ceci est dans l’analyse fonctionnelle ci-dessous***
* *…*

## Analyse fonctionnelle

***Décrivez de manière précise en vous appuyant sur des éléments graphiques (maquettes, schémas) la manière dont vous envisagez que votre utilisateur va travailler avec votre produit.***

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

## Stratégie de test

***Décrire quels sont les MOYENS utilisés pour faire les tests, ne pas décrire les tests à effectuer !!!***

***Décrire l’environnement dans lequel se fait la sprint review***

*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.*

## Risques techniques

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

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

## 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.~~*

***Cette section n’est présente que si la planification initiale a dû être revue suite à l’analyse***

# 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

### Sprints

***Pour chaque sprint : Résumer le déroulement du sprint, le résultat de sa revue, sa retrospective***

### Stories

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

## ~~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…~~*

***Ce chapitre est éclaté en trois ci-dessous***

## 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***

## Mise en place de l’environnement de test

* *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*
* ***La marche à suivre pour préparer l’environnement (ne pas oublier les données de test, ainsi que les mots de passe éventuels)***

***Ce chapitre décrit précisément comment un employé qualifié peut recréer le contexte décrit par la stratégie de test***

## Déploiement du produit

* *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*
* ***La marche à suivre pour préparer l’environnement (ne pas oublier les données de test, ainsi que les mots de passe éventuels)***

***Ce chapitre décrit précisément comment un employé qualifié peut mettre votre produit en production***

## 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~~*

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

## Bilan

### Erreurs restantes

*S'il reste encore des erreurs :*

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

### Stories

***Ce qu’on pensait faire vs ce qu’on a fait***

### Dette technique

***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***

## Recours à l’intelligence artificielle

***Comment avez-vous utilisé l’IA dans votre projet.***

***Si vous ne l’avez pas utilisée, pourquoi ?***

***Ce chapitre doit contenir au minimum 200 mots***

## 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…*

# Conclusions

*Développez en tous cas les points suivants :*

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

# Annexes

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

## 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)*

## Journal de travail

|  |  |  |  |
| --- | --- | --- | --- |
| **~~Date~~** | **~~Durée~~** | **~~Activité~~** | **~~Remarques~~** |
|  |  |  |  |
|  |  |  |  |

***Référence à votre journal de travail (en PDF)***

## ~~Manuel d'Installation~~

## Manuel d'Utilisation

## Archives du projet

*Media, … dans une fourre en plastique*