09.04.2020

Gaël Mariot

Centre de formation Professionnel Technique

Documentation Scrum’o’Wall

Travail de Diplôme ES 2020

Table des matières

[1 Résumé & Abstract 3](#_Toc37249001)

[1.1 Résumé 3](#_Toc37249002)

[1.2 Abstract 3](#_Toc37249003)

[2 Introduction 3](#_Toc37249004)

[3 Cahier des charges 3](#_Toc37249005)

[3.1 Titre du projet 3](#_Toc37249006)

[3.2 Objectifs du projet 3](#_Toc37249007)

[3.3 Description détaillée 3](#_Toc37249008)

[3.3.1 Développement agile 3](#_Toc37249009)

[3.3.2 Stockage multi-projet 3](#_Toc37249010)

[3.3.3 Historique des sprints 3](#_Toc37249011)

[3.3.4 Détection multipoint 3](#_Toc37249012)

[3.4 SWOT 3](#_Toc37249013)

[3.4.1 Forces 3](#_Toc37249014)

[3.4.2 Faiblesses 4](#_Toc37249015)

[3.4.3 Opportunités 4](#_Toc37249016)

[3.4.4 Menaces 4](#_Toc37249017)

[3.5 Outils 4](#_Toc37249018)

[3.5.1 Interface graphique 4](#_Toc37249019)

[3.5.2 Base de données 4](#_Toc37249020)

[3.5.3 Environnement 4](#_Toc37249021)

[4 Fonctionnalités additionnelles 4](#_Toc37249022)

[4.1 Gantt intégré 4](#_Toc37249023)

[4.2 Cartes mentales intégré 4](#_Toc37249024)

[5 Maquettes 5](#_Toc37249025)

[5.1 Fenêtre principale 5](#_Toc37249026)

[5.2 Fenêtre de projet 6](#_Toc37249027)

[5.3 Fenêtre de sprint 7](#_Toc37249028)

[6 Modèle de données 8](#_Toc37249029)

[6.1 Table *project* 8](#_Toc37249030)

[*6.2* Table *UserStory* 8](#_Toc37249031)

[*6.3* Table *Sprint* 9](#_Toc37249032)

[*6.4* Table *State* 9](#_Toc37249033)

[*6.5* Table *MindMap* 9](#_Toc37249034)

[*6.6* Table *Node* 9](#_Toc37249035)

[7 Modèle de classe 9](#_Toc37249036)

[8 Planning 10](#_Toc37249037)

[8.1 Prévu 10](#_Toc37249038)

[8.2 Final 10](#_Toc37249039)

[9 Étude de l’existant 10](#_Toc37249040)

[10 Manuel utilisateur 10](#_Toc37249041)

[11 Manuel d’installation 10](#_Toc37249042)

# Résumé & Abstract

## Résumé

## Abstract

# Introduction

La validation de la formation de technicien ES en informatique du Centre de Formation Professionnel Technique nécessite la réalisation d’un projet sur une durée de neuf semaines afin de démontrer les compétences acquises durant ce cursus.

# Cahier des charges

## Titre du projet

Scrum’o’Wall

## Objectifs du projet

Le but de ce projet est de créer un logiciel interagissant avec le mur tactile NCI Lab situé dans la salle de Technicien ES du CFPT Informatique. Ce logiciel permettra la gestion de projets en méthode agile et utilisera la fonctionnalité multipoint de l’écran permettant ainsi l’utilisation du logiciel par plusieurs utilisateurs simultanément.

## Description détaillée

### Développement agile

Le développement agile est une méthode de travail très répandue dans le pôle informatique car il permet de pouvoir maintenir un produit plus facilement et se concentre sur des petites parties d’un projet effectuées lors de sprint plutôt que sur le projet entier d’un coup.

### Stockage multi-projet

Le logiciel devra pouvoir contenir plusieurs projets et les garder en mémoire afin de permettre à plusieurs équipes de travailler sur le même mur. Cela permet d’apporter un gain de place aussi bien sur le stockage de la machine que physiquement.

### Historique des sprints

Le logiciel permettra de voir les sprints selon leur ordre chronologique pour permettre aux utilisateurs de voir le chemin qu’ils auront parcouru et ils pourront planifier en avances les sprints suivants.

### Détection multipoint

Le logiciel pourra gérer une interaction multipoint permettant à plusieurs utilisateurs de travailler sur tout l’écran simultanément.

## SWOT

### Forces

Ce projet permettra d’apporter une gestion plus facile et centralisée des projets avec les méthodes agiles. De plus, le fait de pouvoir interagir avec les tableaux rien qu’en les touchant rendra l’application encore plus intuitive.

### Faiblesses

Ce projet est cependant un peu faible techniquement. C’est pourquoi la détection multipoint est indispensable et représentera le défi de l’application afin d’être fonctionnel et intéressant pour l’utilisateur.

### Opportunités

C’est la première fois que je crée une application prévue pour un appareil tactile avec C# et j’ai donc eu l’opportunité de pouvoir m’atteler à un projet qui pourra être utiliser tout en découvrant la détection multipoint avec C#.

### Menaces

Malheureusement, le mur tactile n’est plus mis à jour et la dernière version des pilotes du mur tactile est prévue pour Windows 8. De plus, le mur tactile est difficile à calibrer afin de ne pas avoir d’erreur de réception.

Enfin, le Covid-19 étant survenu juste avant le début du projet, je ne peux pas commencer mon projet sur le matériel ciblé. Même si l’école d’informatique m’a fourni un matériel permettant d’accomplir mon projet tout en restant confiné, la différence entre le mur et l’écran pourraient me nuire.

## Outils

### Interface graphique

Afin de réaliser l’interface graphique, j’ai décidé d’utiliser WPF. En effet, la détection multipoint est déjà gérée sur WPF et non sur Windows Form. De plus, WPF permet de créer des applications multi-plateformes.

### Base de données

Afin d’intégrer une base de données, j’ai choisi d’utiliser Access car c’est une technologie avec laquelle je suis déjà familier. En effet, n’étant déjà pas familier avec WPF et la détection multipoint, j’ai pensé qu’il serait mieux de ne pas miser sur trop d’inconnus et utiliser un système de base données que je connais bien et qui est très bien intégré avec C#. De plus, Access est bien plus facile à configurer pour l’utilisateur et ne requiert pas de serveur afin de le faire fonctionner.

### Environnement

Afin de réaliser ce projet, j’ai à ma disposition un ordinateur avec Windows 10 ainsi que l’édition Community de Visual Studio 2019. On m’a également fourni un écran tactile *Iiyama ProLite T2735MSC* afin de pallier les problèmes du Covid-19

Pour ce qui est des créations graphiques, je vais utiliser l’application graphique GIMP version 2.10.18 pour le poster, Balsamiq Wireframes pour les maquettes de l’interface utilisateur et Draw.io pour les diagrammes et le MCD

# Fonctionnalités additionnelles

Voici des fonctionnalités qui pourront être rajoutées si le temps le permet.

## Gantt intégré

Le logiciel aura un diagramme de Gantt intégré pour chaque projet. Il se mettra à jour à chaque création de Use Case. Il ordonnera

## Cartes mentales intégré

Le logiciel pourra créer des mindmap

# Maquettes

## Fenêtre principale



Figure Maquette de la fenêtre de sélection de projet

Sur cette fenêtre, une mosaïque avec les différents projets déjà créé sera afficher. Un bouton permettra de rajouter un projet. Un maintien prolongé permettra également de rajouter un projet par un menu contextuel.

## Fenêtre de projet

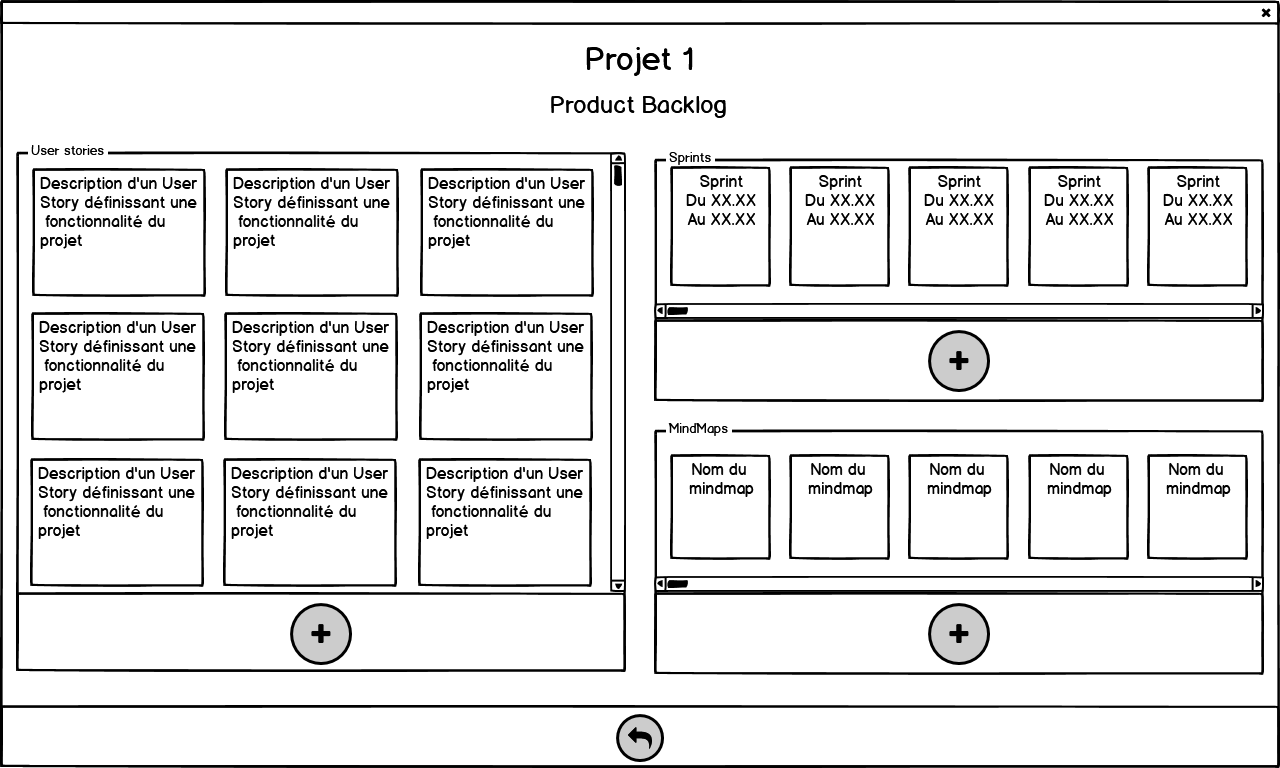


Figure Maquette de la fenêtre de projet

Cette fenêtre s’affichera dès l’ouverture d’un projet. On peut changer le nom du projet ainsi que sa description en cliquant sur le bouton à côté du nom ou en déclenchant le menu contextuel via un appui prolongé.

La partie gauche permettra de créer des fonctionnalités à rajouter dans le projet à travers des Use Case ou les modifier alors que la partie de droite permettra de rajouter et d’éditer des Sprints et leur rajouter des Use Case via un glisser-déposer sur le sprint voulu. Les sprints déjà passé dans la date seront légèrement grisé et ne pourront pas être modifié alors que l’actuel sera mis en évidence.

## Fenêtre de sprint



Figure Maquette de la fenêtre de sprint

Cette fenêtre affiche le sprint sur lequel on aura appuyé. Un bouton permet de revenir au backlog. Un appui prolongé ouvrira le menu contextuel qui permettra de rajouter des Use Case ou de modifier les Use Case existant.

Le menu contextuel permettra de rajouter des colonnes pour permettre aux gens de gérer au mieux leur projet de la façon qui leur convient.

## PopUp Projets

Une image contenant capture d’écran

Description générée automatiquement

Figure Maquette du popup de création de projet

Ce popup apparait quand la modification ou l’ajout d’un projet est demandé. En appuyant sur la croix en bas à gauche, la fenêtre se ferme sans changement alors qu’en appuyant sur la droite, les changements s’effectuent.

## PopUp UserStory

Une image contenant capture d’écran

Description générée automatiquement

Figure Maquette du popup de création de UserStory

Ce popup apparait quand la modification ou l’ajout d’une UserStory est demandée. En appuyant sur la croix en bas à gauche, la fenêtre se ferme sans changement alors qu’en appuyant sur la droite, les changements s’effectuent.

## PopUp Sprint

Une image contenant capture d’écran, dessin

Description générée automatiquement

Figure Maquette du popup de création de sprint

Ce popup apparait quand la modification ou l’ajout d’un sprint est demandé. En appuyant sur la croix en bas à gauche, la fenêtre se ferme sans changement alors qu’en appuyant sur la droite, les changements s’effectuent.

# Modèle de données

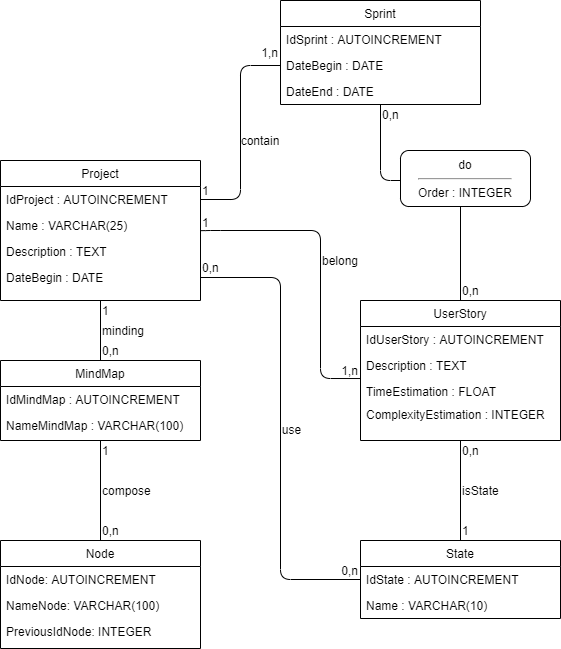


Figure Modèle Conceptuel de Données

## Table *project*

La table *project* contiendra les grandes lignes des différents projets enregistrés dans la base de données. Cette table sera liée à la table state pour définir quelles colonnes seront affichés dans le sprint.

## Table *UserStory*

Cette table contient les différentes fonctionnalités. Chaque User Story contient une description de type TEXT, une estimation de temps en FLOAT car cette estimation sera sur une journée et cela permettra de mettre de fraction de journée et une estimation de complexité afin de pouvoir attribuer les tâches aux bonnes personnes. Une table de liaison reliera la table Sprint et cette table afin de créer le contenu du sprint.

## Table *Sprint*

Cette table permet de déterminer la date de début et de fin des sprints. Une table de liaison reliera la table UserStory et cette table afin de créer le contenu du sprint.

## Table *State*

Cette table stocke tous les états possibles qu’une UserStory peut avoir. Il permet également, grâce à une liaison avec la table project de pouvoir rajouter des colonnes au sprint. La liaison se trouve entre cette table et project afin de n’avoir à changer les états qu’à un endroit.

## Table *MindMap*

Cette table permet la création de mindmap en gardant leurs noms en mémoire et en les liant aux projets correspondants.

## Table *Node*

Cette table contient tous les différents nœuds du mindmap. Cette table contient une référence sur elle-même afin de définir le nœud parent. Cela me permettra par la suite d’utiliser une fonction récursive pour l’affichage.

# Modèle de classe

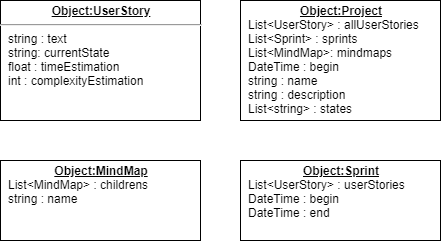


Figure Diagramme de classe

# Planning

## Prévu



Figure Planning initial

## Final

TODO Remplir à la fin

# Étude de l’existant

# Manuel utilisateur

# Manuel d’installation