28.04.2020

Travail de Gaël Mariot

Suivi par Anne Terrier

Centre de formation Professionnel Technique

Documentation Scrum’o’Wall

Travail de Diplôme Technicien ES en informatique.

Session 2019 2020

Table des matières

[1 Résumé & Abstract 3](#_Toc38986452)

[1.1 Résumé 3](#_Toc38986453)

[1.2 Abstract 3](#_Toc38986454)

[2 Lexique 3](#_Toc38986455)

[3 Introduction 3](#_Toc38986456)

[4 Cahier des charges 3](#_Toc38986457)

[4.1 Objectifs du projet 3](#_Toc38986458)

[4.2 Fonctionnalités 3](#_Toc38986459)

[4.2.1 Stockage multi-projet 3](#_Toc38986460)

[4.2.2 Historique des sprints 3](#_Toc38986461)

[4.2.3 Détection multipoint 3](#_Toc38986462)

[4.3 SWOT 4](#_Toc38986463)

[4.4 Étude de l’existant 4](#_Toc38986468)

[4.5 Outils 7](#_Toc38986474)

[5 Analyse fonctionnelle 7](#_Toc38986478)

[5.1 Maquettes 8](#_Toc38986479)

[5.2 Description des outils 12](#_Toc38986486)

[5.2.1 Visual Studio 12](#_Toc38986487)

[5.2.2 Access 12](#_Toc38986488)

[5.3 Description des fonctionnalités 12](#_Toc38986489)

[5.3.1 12](#_Toc38986490)

[5.4 Fonctionnalités additionnelles 12](#_Toc38986491)

[5.4.1 Gantt intégré 12](#_Toc38986492)

[5.4.2 Cartes mentales intégré 12](#_Toc38986493)

[5.4.3 Protection de projets 12](#_Toc38986494)

[6 Analyse Organique 13](#_Toc38986495)

[6.1 Modèle de données 13](#_Toc38986496)

[6.2 Modèle de classe 15](#_Toc38986511)

[6.3 Planning 16](#_Toc38986512)

[6.3.1 Prévu 16](#_Toc38986513)

[6.3.2 Final 16](#_Toc38986514)

[7 Manuel utilisateur 16](#_Toc38986515)

[8 Manuel d’installation 16](#_Toc38986516)

# Résumé & Abstract

## Résumé

## Abstract

# Lexique

* **SWOT** : Une méthode d’analyse permettant de comparer un projet ainsi que sa faisabilité en analysant ses forces (Strength), ses faiblesses (Weaknesses), ses opportunités (Opportunities) et ses menaces (Threat).
* **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.
* **Sprint :** Période de travail courte d’un projet.
* **User Story :** Courte description d’une fonctionnalité. Habituellement formulée en cas d’utilisation.
* **Backlog :** Ensemble des User Stories d’un projet.
* **MindMap :** Une méthode pour pouvoir récolter des idées sur un sujet ou une problématique.

# 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

## 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.

## Fonctionnalités

### 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 utilisé 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.

## Étude de l’existant

Afin de pouvoir comparer, j’ai sélectionné des applications qui gèrent les fonctions tactiles. Je me suis arrêté aux plus connues.

### Trello

Trello est un produit racheté par la compagnie « Atlassian ». Il permet d’afficher un tableau contenant diverses listes créées par l’utilisateur. Dans ces listes, des cartes sont créées et on peut leur attribuer des utilisateurs, des mots-clefs ou encore créer une checklist afin d’accomplir diverses tâches pour finir une carte.



<https://trello.com/>

### JIRA

JIRA est un produit de la compagnie « Atlassian ». Il permet de créer et distribuer des tâches aux divers employés. Une fenêtre permet de définir des dates cible pour sortir certaines versions. Il existe un moyen de pouvoir visualiser les performances de son équipe en temps réel.



<https://www.atlassian.com/software/jira>

### Ubikey

Ubikey est une application payante. Elle permet de travailler à plusieurs sur le même écran et de connecter plusieurs écrans à distance. La collaboration via un téléphone portable ou un autre ordinateur est également possible.

Cependant, cette application ne propose pas d’outils de visualisation de la progression.



<https://www.ubikey.fr/>

### Kantree

Kantree est un logiciel de gestion de projet payant. Il permet de collaborer en temps réel sur plusieurs dispositifs et d’afficher les Sprints sous diverses formes telles que le format classique en colonnes, une matrice contenant chaque colonne pour chaque employé ou encore un calendrier montrant le temps estimé. Il permet également de sauvegarder le projet en local pour continuer à gérer son projet en étant hors ligne.



<https://kantree.io/fr>

### Résumé comparatif

Mon projet se distinguera de ces projets car toutes les données sont stockées localement et n’ont pas besoin d’internet. Cela signifie qu’aucun service ne doit être entretenu pour garder les données sur un serveur distant.

De plus, aucun de ces projets n’a de mindmap intégré dans leurs fonctionnalités. Le mindmap est une étape permet de pouvoir

## 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* versio*n 2.10.18* pour le poster, *Balsamiq Wireframes* pour les maquettes de l’interface utilisateur et *Draw.io* pour les diagrammes et le MCD

# Analyse fonctionnelle

## Maquettes

### Fenêtre principale



Figure 2 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 3 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. Elle permettra également de les modifier. La partie de droite, quant à elle, permettra de rajouter et d’éditer des Sprints et de leur rajouter des Use Case via un glisser-déposer sur le sprint voulu. Les sprints dont la date sera déjà passée seront légèrement grisés et ne pourront pas être modifiés. Le sprint en cours sera mis en évidence.

### Fenêtre de sprint

Une image contenant capture d’écran

Description générée automatiquement

Figure 4 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 existants.

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

### Pop-up Projets

Une image contenant capture d’écran

Description générée automatiquement

Figure 5 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.

### Pop-up UserStory

Une image contenant capture d’écran

Description générée automatiquement

Figure 6 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.

### Pop-up Sprint

Une image contenant capture d’écran, dessin

Description générée automatiquement

Figure 7 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.

## Description des outils

### Visual Studio

### Access

## Description des fonctionnalités

### 

## 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 permettra d’avoir une idée de la date de fin de projet et du nombre de jours nécessaires à la complétion d’un projet.

### Cartes mentales intégré

Le logiciel pourra créer des mindmap. Cela permettrait aux équipes de pouvoir faire des brainstormings de façon plus efficace et sans avoir de problèmes pour retrouver le mindmap sur un aspect d’un projet.



Figure 1 Maquette écran du mindmap

### Protection de projets

Il pourrait être possible d’intégrer une gestion de mots de passe pour limiter l’accès aux projets aux autres utilisateur du poste. Cela pourrait être utile si des informations sensibles se situent dans un projet.

# Analyse Organique

## Modèle de données

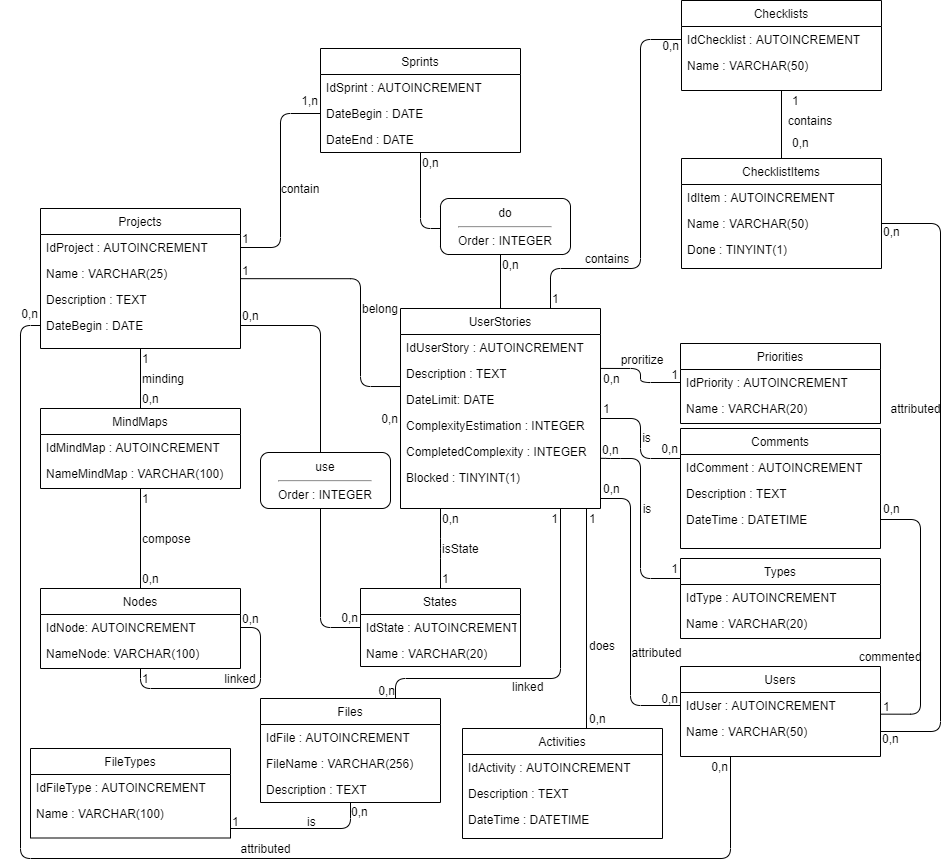


Figure 8 Modèle Conceptuel de Données

### Table project

Cette table contiendra la description, le nom ainsi que la date de début du projet. C’est cela qui permettra d’identifier le projet sur la première page de l’application.

Cette table sera liée à la table *state* pour définir quelles colonnes seront affichés dans le sprint. Elle sera également liée à la table *user* pour déterminer l’attributions d’utilisateurs dans le projet.

### Table UserStory

Cette table contient les différentes fonctionnalités. Chaque User Story contient une description de type TEXT, une estimation de complexité afin de pouvoir attribuer les tâches aux bonnes personnes, la valeur de la complexité accomplie pour pouvoir réaliser un suivi.

Une table de liaison reliera la table Sprint et cette table afin de créer le contenu du sprint. Une table de liaison avec tous les différents utilisateurs sera également créée pour montrer les utilisateurs attribués à cette tâche.

### Table Sprint

Cette table permet de déterminer la date de début et de fin des sprints. On utilisera ses dates afin d’identifier les différents sprints et de les afficher dans la page du projet.

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 tâche peut avoir.

Une table de liaison la reliera avec la table *project*. Cette table de liaison déterminera tous les états disponibles dans les sprints du projet.

### 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.

### Table Files

Cette table permet aux utilisateurs de lier un fichier à une tâche. Les fichiers sont stockés sur l’ordinateur et non dans la base. L’attribut *filename* permet de retrouver le fichier grâce à son chemin d’accès. Un attribut *description* permet de définir le rôle de ce fichier dans la tâche.

### Table FileTypes

Cette table sert de constante.

Elle permet de garder en mémoire les types de fichiers.

### Table Activities

Cette table permet d’avoir des activités liées aux tâches dans le projet. Elle permet de garder une trace sur les actions lors du travail sur un projet.

### Table Type

Cette table sert de constante.

Elle permet de garder en mémoire si la tâche créée est une fonctionnalité ou si cette tâche permet la résolution d’un problème. Elle ne contient que très peu d’enregistrements.

### Table Comments

Cette table permet aux utilisateurs de pouvoir commenter une tâche. Les commentaires ont une date et une heure afin de pouvoir les ordonner et un utilisateur y sera assigné pour savoir à qui s’adresser.

### Table Priorities

Cette table sert de constante.

Elle permet de garder en mémoire les ordres de priorité et ne contient que très peu d’enregistrements.

### Table Checklists

Cette table permet de créer une checklist qui servira à fractionner les tâches d’une UserStory. Un nom est attribué par les utilisateurs pour donner un nom à la liste.

### Table ChecklistItems

Cette table permet de créer un objet dans la checklist liée. L’attribut *done* permet de savoir si la tâche créée par cet enregistrement est effectuée. L’attribut *name* permet d’afficher dans les vues un identifiant déterminé par l’utilisateur.

Une table de liaison la reliera avec la table *users* afin de pouvoir attribuer des utilisateurs aux différentes tâches que cette table va créer.

## Modèle de classe

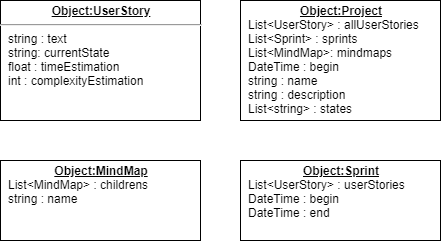


Figure 9 Diagramme de classe

## Planning

### Prévu



Figure 10 Planning initial

### Final

TODO Remplir à la fin

# Manuel utilisateur

# Manuel d’installation