A thick dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

08/06/2020

Documentation Scrum'o'Wall

Travail de Diplôme Technicien ES en
informatique.

Session 2019 2020

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

Travail de Gaël Mariot
Suivi par Anne Terrier

CENTRE DE FORMATION PROFESSIONNEL TECHNIQUE

1	Résumé & Abstract	2
1.1	Résumé	2
1.2	Abstract.....	2
2	Lexique	3
3	Introduction	4
4	Cahier des charges	4
4.1	Objectifs du projet.....	4
4.2	Fonctionnalités	4
4.3	SWOT	4
4.4	Étude de l'existant.....	5
4.5	Planning	8
4.6	Outils.....	9
5	Analyse fonctionnelle	10
5.1	Description des fonctionnalités.....	10
5.2	Maquettes d'écrans.....	11
5.3	Fonctionnalités additionnelles	22
5.4	Manuel d'installation.....	23
6	Analyse Organique	24
6.1	Modèle de données.....	24
6.2	Modèle de classes	26
6.3	Apports personnels	30
6.4	Communications avec la base de données	30
6.5	Liaison Access.....	31
6.6	Glisser-déposer multipoint	31
7	Plan de tests.....	34
7.1	Test d'interface	34
7.2	Tests unitaires	37
8	Conclusion	38
9	Table des figures	39
10	Annexes	39

1 Résumé & Abstract

1.1 Résumé

Dans le cadre de ma formation en tant que Technicien en informatique au CFPT, il m'est demandé de réaliser un travail de diplôme pour démontrer mes capacités.

Le but de ce projet est de créer une application de gestion de projets agiles destinée à fonctionner sur un mur tactile. Mon idée de départ est de proposer un backlog sur lequel les membres d'une équipe Scrum peuvent interagir facilement. Afin d'enrichir l'application, j'y ai intégré d'autres fonctionnalités de gestion de projets et de gestion de sprints. On peut y trouver un diagramme de Burndown, des commentaires sur les User Stories, des checklists ainsi qu'une assignation d'utilisateurs aux projets/User Story/Objet de checklist.

Pour réaliser ce travail, je me suis orienté sur le développement d'une application centralisée intégrant les fonctionnalités tactiles. Toutes les données sont, par conséquent, stockées en local dans une base de données Access. L'application programmée en C# intègre la spécification graphique de Microsoft, Windows Presentation Foundation (WPF) pour l'interface utilisateur.

Afin de convenir au plus grand nombre, je me suis efforcé de rendre mon application modulable. Il est par exemple possible de personnaliser les colonnes disponibles dans les sprints via l'édition d'un projet. Il est également possible de lier des fichiers à une User Story pour les retrouver plus facilement. De plus, cette application peut gérer plusieurs utilisateurs simultanément grâce à une fonctionnalité multipoint ce qui me semble être un atout considérable. Je pense donc que c'est l'outil idéal pour remplacer les backlog papier.

1.2 Abstract

As part of my training as a technician in computer science at the CFPT, I am asked to complete a diploma work to demonstrate my capacities.

The purpose of this project is to create an application to manage agile projects to run on a touch wall. My initial idea is to provide a backlog on which Scrum team members can easily interact. To enrich the application, I integrated other project management and sprint management features. You can find a Burndown chart, a comment section on User Stories, checklists on User Stories as well as a user assignment on Projects, User Story and Checklist items.

To achieve this work, I focused on the development of a centralized application integrating touch functionalities. All data is therefore stored locally in an Access database. The application made in C# integrates Microsoft's graphical specification, Windows Presentation Foundation (WPF) for the user interface.

To suit the largest number of people, I tried to make my application modular. For example, it is possible to customize the columns available in the sprints through the edition of a project. It is also possible to link files to a User Story to find them more easily. In addition, this application can manage several users simultaneously with a multitouch functionality which seems to me to be a considerable attribute. So, I think it is the ideal tool to replace paper backlogs.

2 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.
- **CRUD** : Le CRUD est un acronyme signifiant « Create,Read,Update,Delete ». Il représente les opérations de création, lecture, mise à jour et suppression sur une base de données et sont les quatre opérations sont les piliers d'une base de données.
- **CRD** : Equivalent du CRUD mais sans la mise à jour.

3 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.

Ce projet porte sur la création d'un logiciel de gestion. En effet, l'informatique permet de faciliter la gestion de bien des façons et, ayant déjà accompli mon travail de CFC sur un outil de gestion, j'ai été intéressé par ce projet quand Mme. Terrier me l'a proposé.

4 Cahier des charges

4.1 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.

4.2 Fonctionnalités

4.2.1 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.

4.2.2 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.

4.2.3 Détection multipoint

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

4.3 SWOT

4.3.1 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.

4.3.2 Faiblesses

Ce projet est dangereux. Beaucoup d'applications reproduisent déjà le même rôle. Cependant, j'ai rajouté des options inédites telles que le diagramme de progression avec le burndown chart.

4.3.3 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#.

4.3.4 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.

4.4 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.

4.4.1 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-clés ou encore créer une checklist afin d'accomplir diverses tâches pour finir une carte.

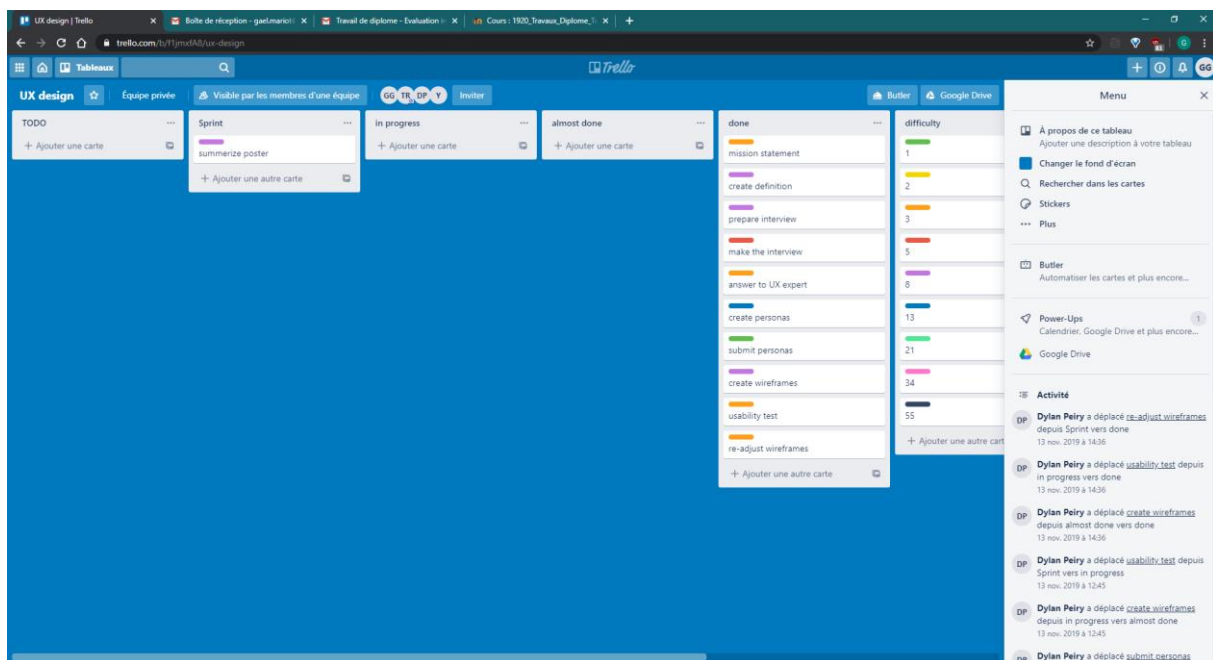


Figure 1 Capture d'écran de trello

<https://trello.com/>

4.4.2 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.

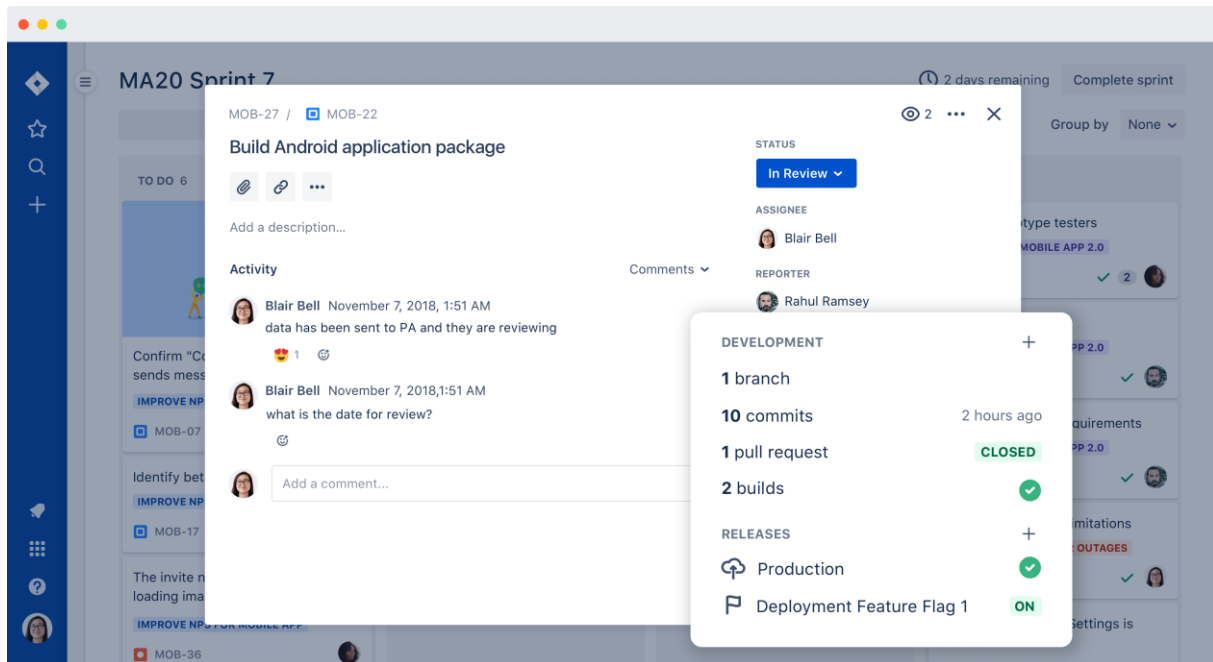


Figure 2 Capture d'écran de JIRA

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

4.4.3 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.

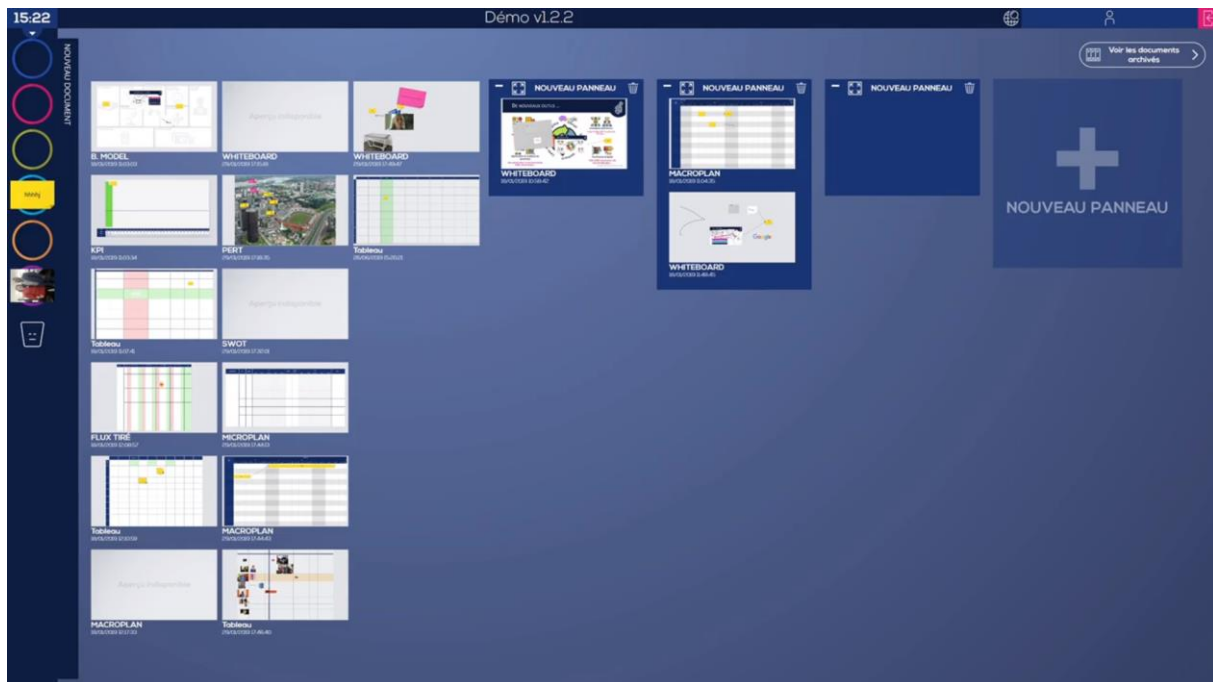


Figure 3 Capture d'écran de Ubikey

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

4.4.4 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.

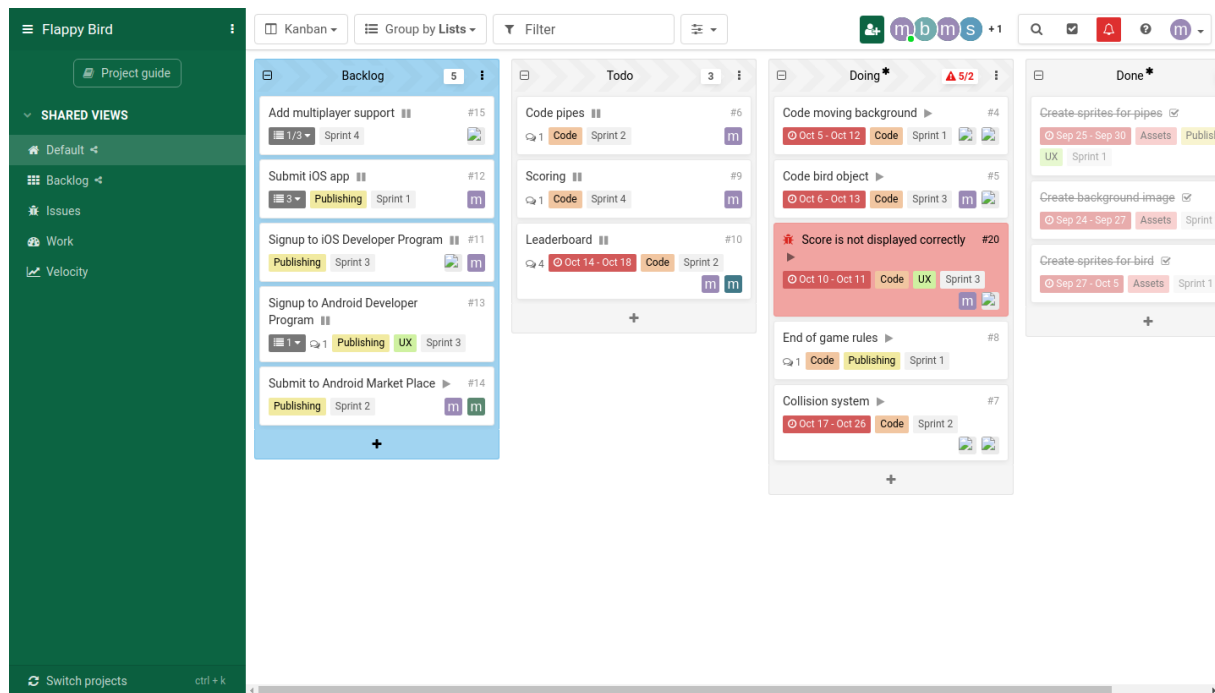


Figure 4 Capture d'écran de Kantree

<https://kantree.io/frhttps://kantree.io/fr/cas-usage/teletravail>

4.4.5 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 qui permet à l'équipe de développement de trouver un terrain d'entente plus facilement et de lier leurs pensées pour un même projet.

4.5 Planning

4.5.1 Prévu

Planning du travail de diplôme				
N° Tache	Description de la tâche	Temps estimé(j)	Temps réel (j)	Date limite
1	Documentation du projet, Journal de bord	18		
2	Modelisation de la base de données	1		
3	Création de la base de données	0.5		
4	Mise en place des systèmes de versioning	1		
5	Création du backend du logiciel	3		
6	Création de la fenêtre principale	1		
7	Création de la fenêtre de projet	1		
8	Création de la fenêtre de sprint	1		
9	Création d'un controlleur	3		
10	Liaison controlleur - fenêtre principale	1		
11	Liaison controlleur - fenêtre de projet	1		
12	Liaison controlleur - fenêtre de sprint	1		
13	Création du poster	2		
14	Implementation des mindmap	2		
15	Implémentation du gantt	2		
Jalon	Evaluation Intermédiaire 01			24.04.2020
Jalon	Rendu du rapport intermédiaire + poster			07.05.2020
Jalon	Evaluation Intermédiaire 02			08.05.2020
Jalon	Rendu version intermédiaire du résumé et de l'abstract			15.05.2020
Jalon	Evaluation Intermédiaire 03			22.05.2020
Jalon	Rendu travail			08.06.2020

Figure 5 Planning initial

4.5.2 Final

Planning du travail de diplôme				
N° Tache	Description de la tâche	Temps estimé(j)	Temps réel (j)	Date limite
1	Documentation du projet, Journal de bord	18	14	
2	Modelisation de la base de données	1	2	
3	Création de la base de données	0.5	1	
4	Mise en place des systèmes de versioning	1	0.25	
5	Création du backend du logiciel	3	2	
6	Création de la fenêtre principale	1	1	
7	Création de la fenêtre de projet	1	1	
8	Création de la fenêtre de sprint	1	1	
9	Création d'un contrôleur	3	2	
10	Liaison contrôleur - fenêtre principale	1	1	
11	Liaison contrôleur - fenêtre de projet	1	1	
12	Liaison contrôleur - fenêtre de sprint	1	1	
13	Création du poster	2	2	
14	Implementation des mindmap	2	3	
15	Implémentation du gantt	2	0	
16	Création de fenêtres pop-up		3	
17	Tests unitaires		3	
Jalon	Evaluation Intermédiaire 01			24.04.2020
Jalon	Rendu du rapport intermédiaire + poster			07.05.2020
Jalon	Evaluation Intermédiaire 02			08.05.2020
Jalon	Rendu version intermédiaire du résumé et de l'abstract			15.05.2020
Jalon	Evaluation Intermédiaire 03			22.05.2020
Jalon	Rendu travail			08.06.2020

Figure 6 Planning final

On peut voir que certaines tâches ont fait leur apparition et que je n'ai pas eu tout le temps voulu afin de travailler sur la documentation.

4.6 Outils

4.6.1 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.

4.6.2 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.

4.6.3 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 aux restrictions liées au 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

5 Analyse fonctionnelle

5.1 Description des fonctionnalités

Gestion des projets

- Créer, Modifier et Supprimer un projet
- Assigner des utilisateurs à un projet
- Assigner des états à un projet

Gestion des sprints

- Créer, Modifier et Supprimer un sprint
- Assigner des User Stories à un sprint

Gestion des User Stories

- Créer, Modifier et Supprimer une user story
- Assigner des utilisateurs à une user story
- Assigner des fichiers à une user story
- Assigner des listes à une user story
- Commenter une user story
- Voir l'historique des actions d'une user story

Gestion des listes

- Créer, Modifier et Supprimer une liste
- Ajouter des objets à la liste
- Assigner des utilisateurs

Gestion des utilisateurs

- Créer, Modifier et Supprimer un utilisateur

Gestion des états

- Créer, Modifier et Supprimer un état

Gestion des fichiers

- Créer, Modifier et Supprimer un fichier

Autres

- Créer et Supprimer des commentaires
- Création automatique des activités

5.2 Maquettes d'écrans

5.2.1 Fenêtre principale

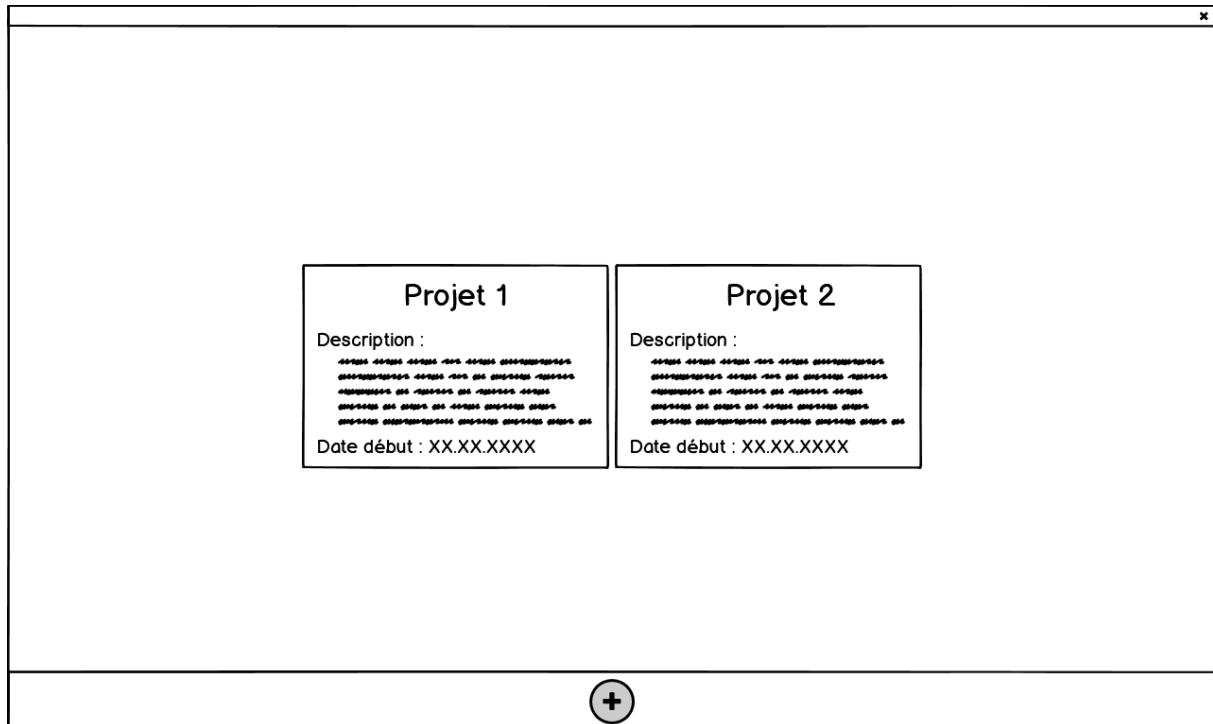


Figure 7 Maquette de la fenêtre de sélection de projet

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 via un menu contextuel.

5.2.2 Fenêtre de projet

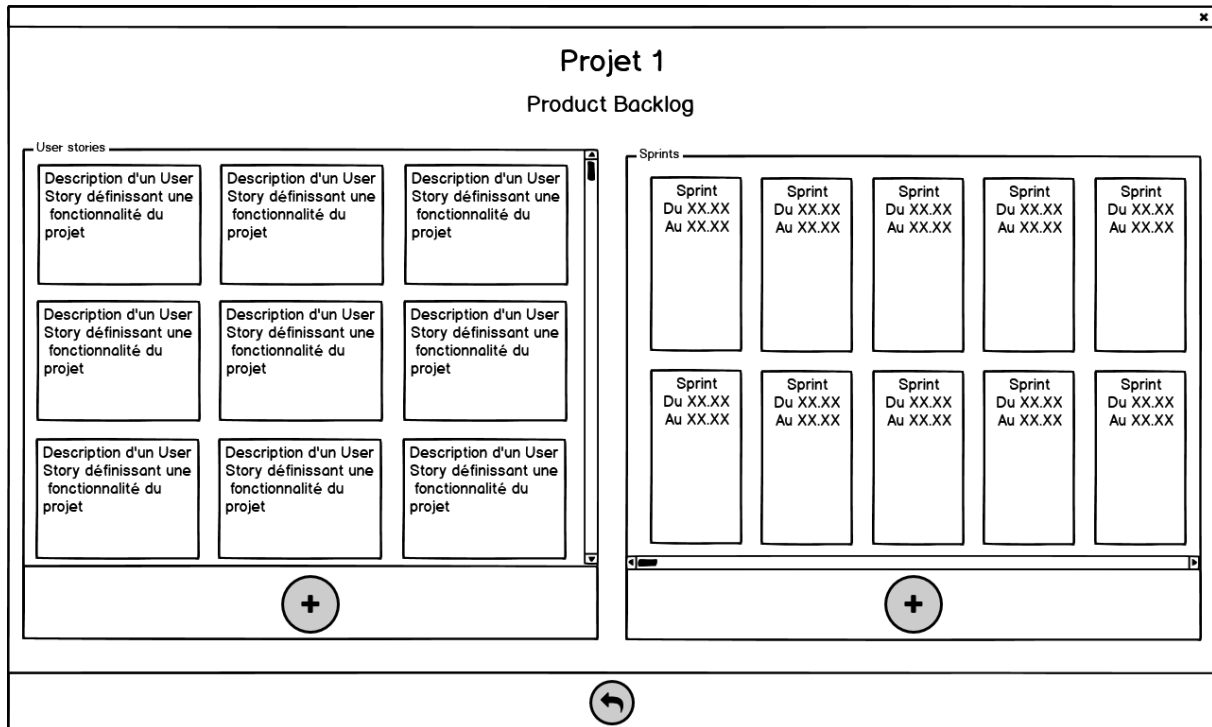


Figure 8 Maquette de la fenêtre de projet

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.

5.2.3 Fenêtre de sprint

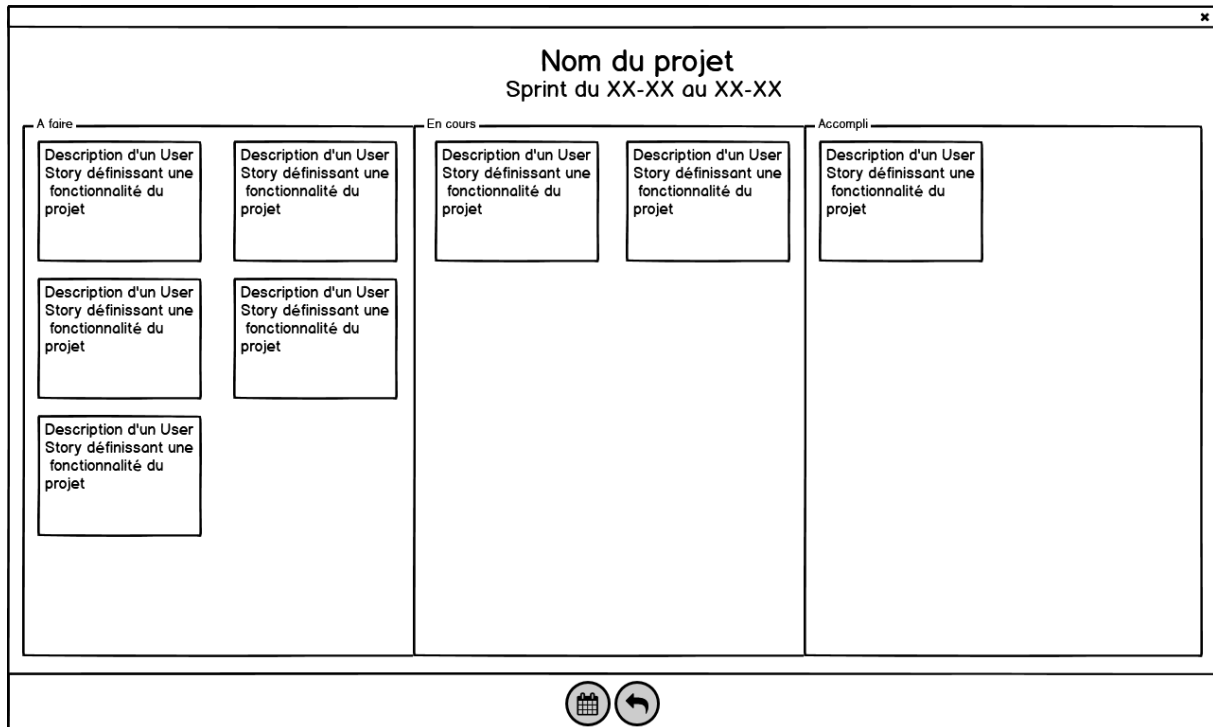


Figure 9 Maquette de la fenêtre de sprint

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.

Le bouton avec le calendrier amène à la Fenêtre de burndown chart

5.2.4 Fenêtre de burndown chart

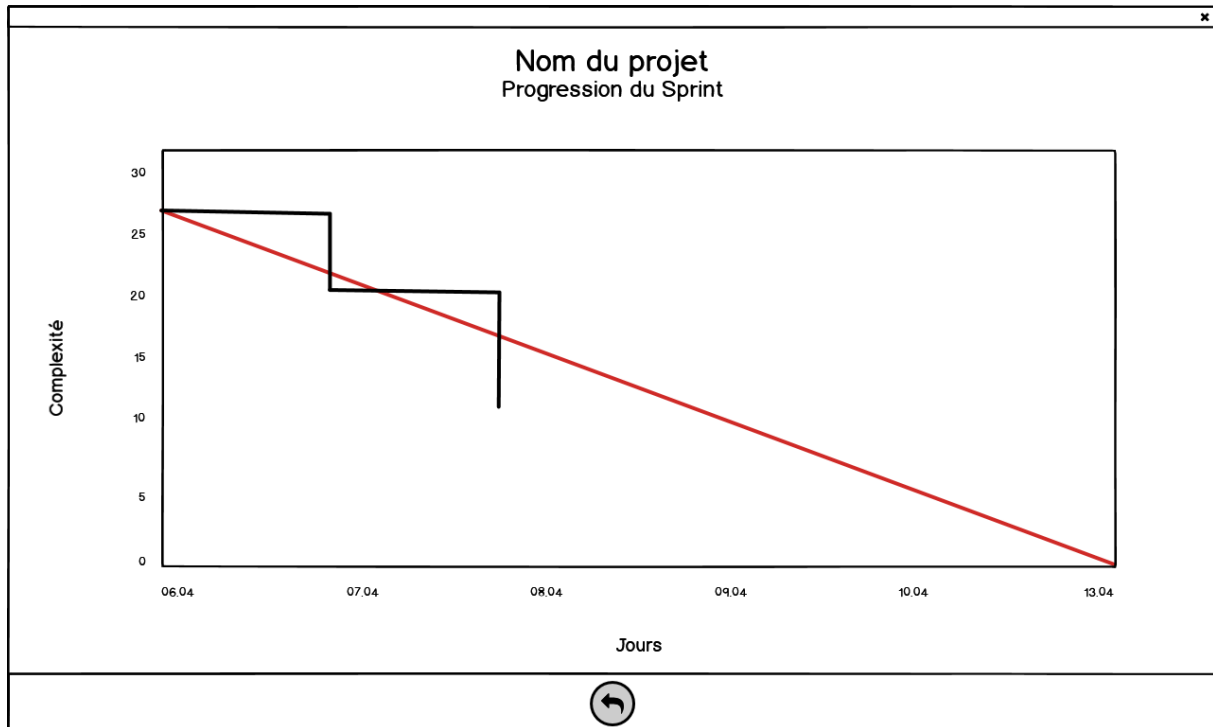


Figure 10 Maquette de la fenêtre du Burndown Chart

Cette fenêtre permet de visualiser la progression d'un sprint. La barre rouge représente l'avancée idéale du sprint et la barre noire représente l'avancée réelle du projet.

5.2.5 Fenêtre des utilisateurs

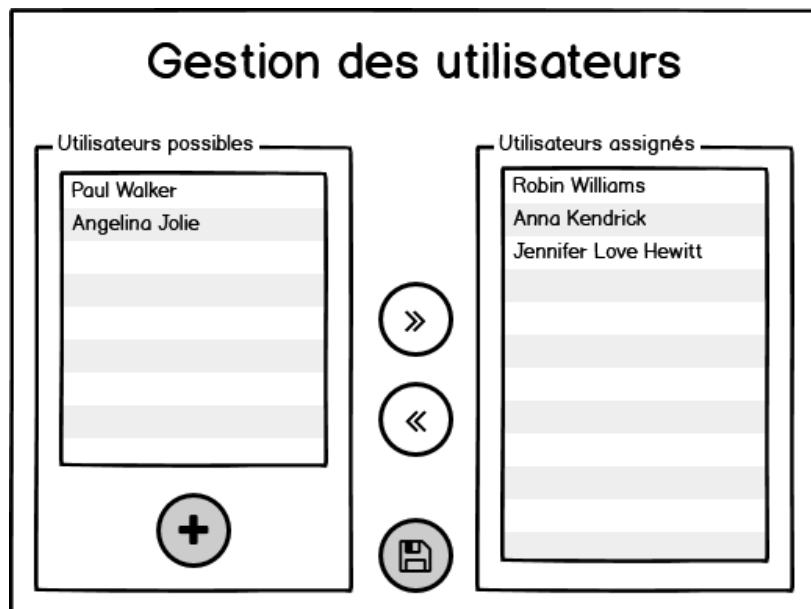


Figure 11 Maquette de la fenêtre des utilisateurs

Cette fenêtre permet de gérer les utilisateurs pour un UserStory, un projet ou un objet de checklist. Il affiche tous les utilisateurs et permet de rajouter et supprimer les utilisateurs assignés à l'objet. Cependant, aucun changement ne sera effectué si le bouton « Enregistrer » n'est pas pressé.

5.2.6 Fenêtre des états

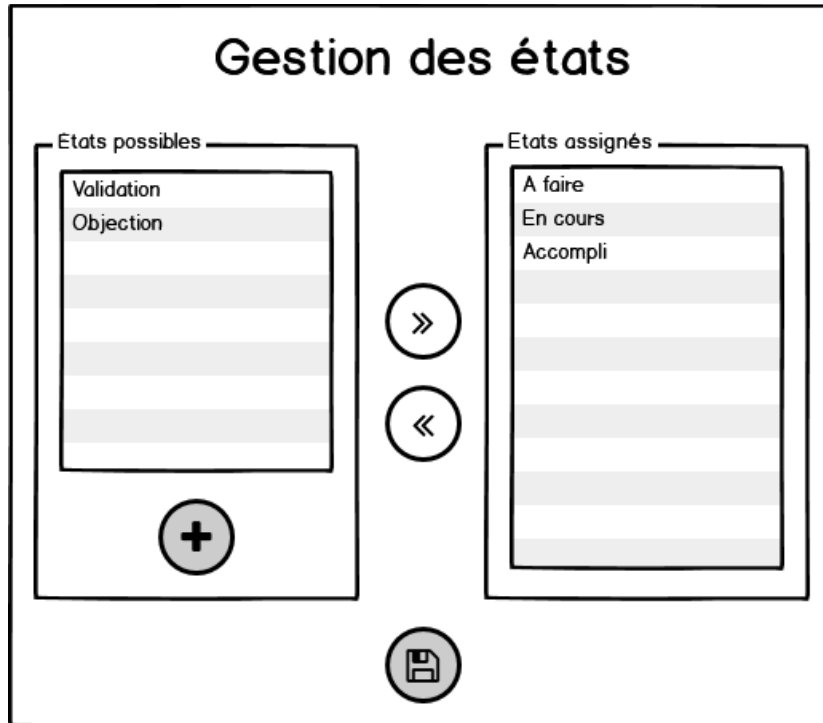


Figure 12 Maquette de la fenêtre des états

Cette fenêtre permet de gérer les utilisateurs pour un projet. Il affiche tous les états et permet de rajouter et supprimer les états au projet. Cependant, aucun changement ne sera effectué si le bouton « Enregistrer » n'est pas pressé.

5.2.7 Fenêtre des activités

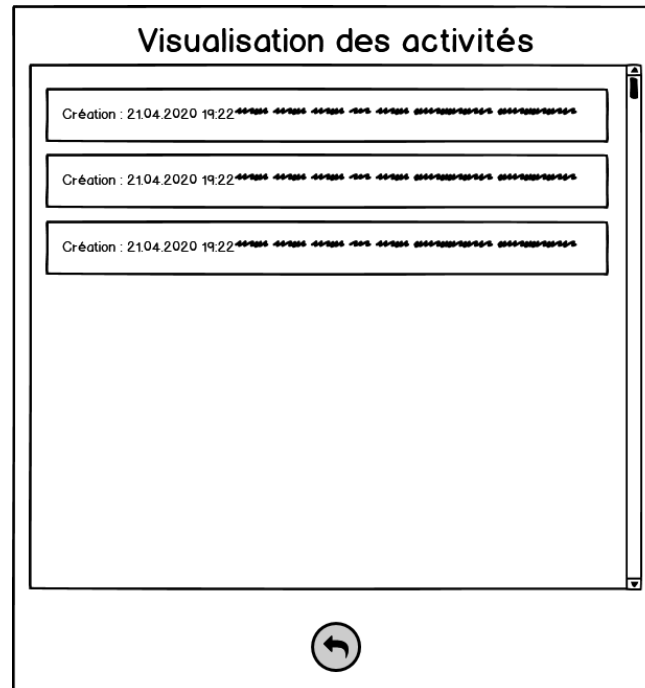


Figure 13 Maquette de la fenêtre des activités

Cette fenêtre permet d'afficher les activités concernant une User Story définie. Elle ne permet pas de créer d'activités car elles sont créées automatiquement.

5.2.8 Fenêtre des checklists

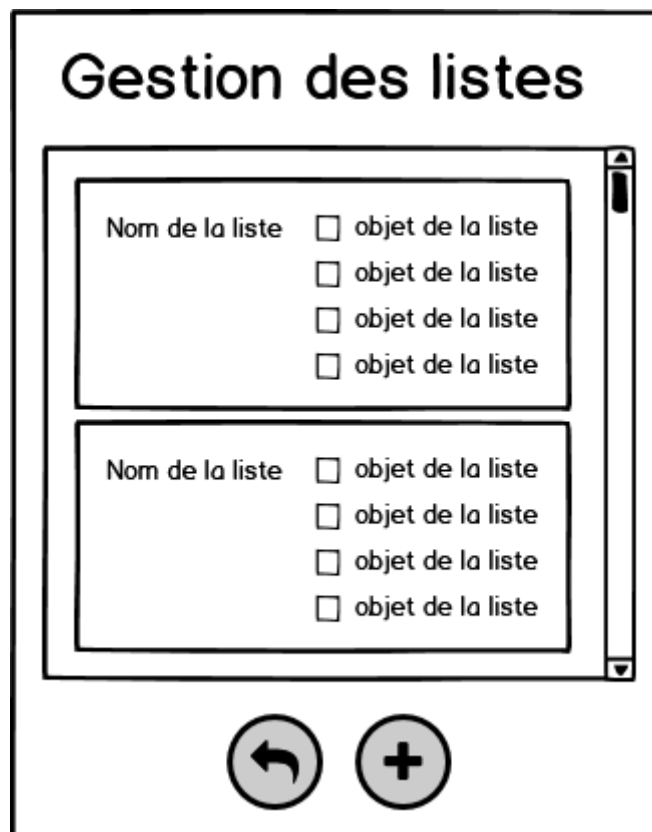


Figure 14 Maquette de la fenêtre des listes

Cette fenêtre permet de créer et de gérer les listes d'une User Story. En cliquant sur une liste, cela permet de modifier la liste. En appuyant sur le bouton « + », une redirection nous amène au pop-up de création de liste.

5.2.9 Fenêtre des commentaires

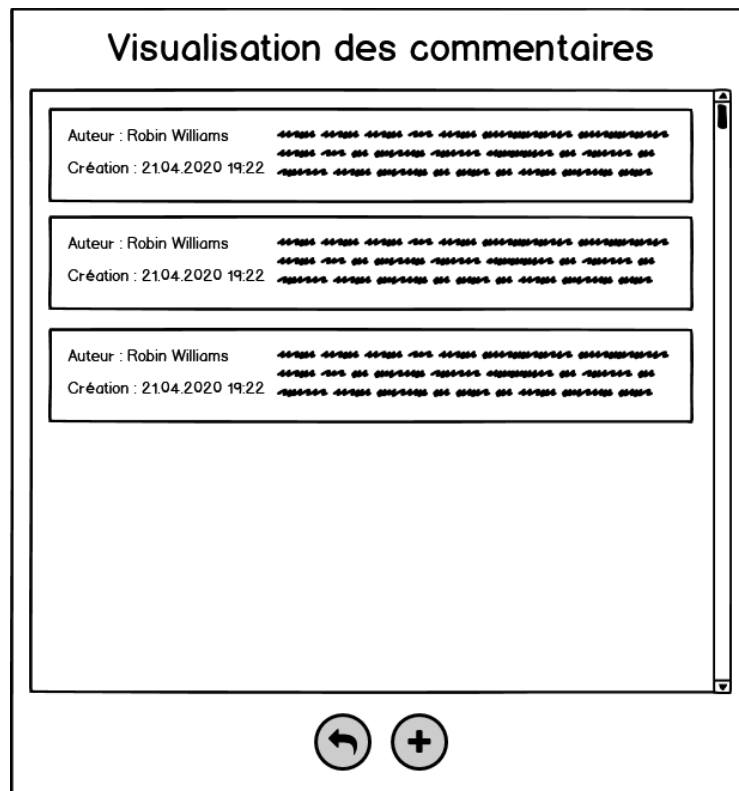


Figure 15 Maquette de la fenêtre des commentaires

Cette fenêtre permet de visualiser les commentaires. Les commentaires sont affichés par date et heure. Le bouton « + » permet de rajouter un commentaire en passant par le pop-up de création de commentaires.

5.2.10 Fenêtre des fichiers

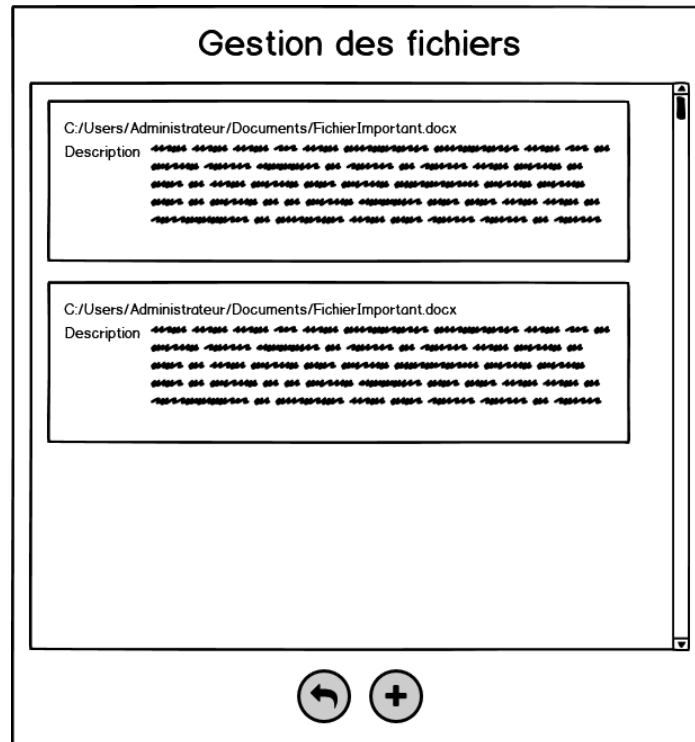


Figure 16 Maquette de la fenêtre de fichiers

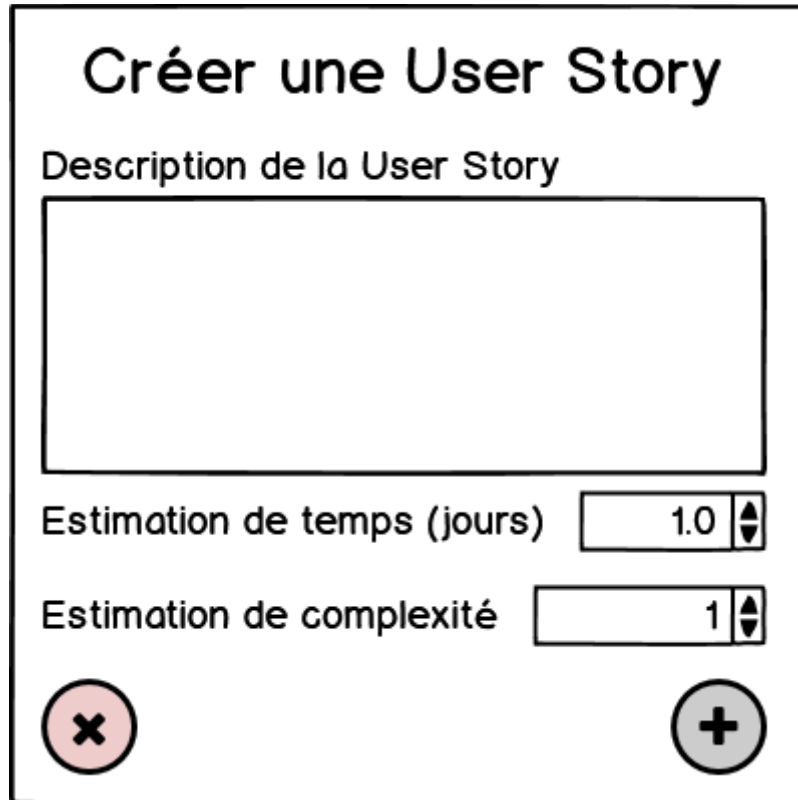
Cette fenêtre permet de gérer les fichiers liés à une User Story. Appuyer sur un fichier permet de le modifier. Appuyer sur le bouton « + » permet d'ajouter un fichier.

5.2.11 Pop-up Projets

Figure 17 Maquette du pop-up 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.

5.2.12 Pop-up UserStory



Créer une User Story

Description de la User Story

Estimation de temps (jours)

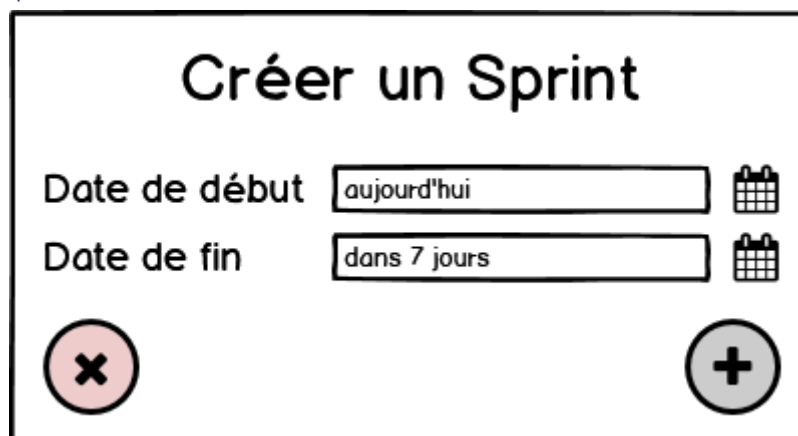
Estimation de complexité

Buttons: [X] [+]


Figure 18 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.

5.2.13 Pop-up Sprint



Créer un Sprint

Date de début 

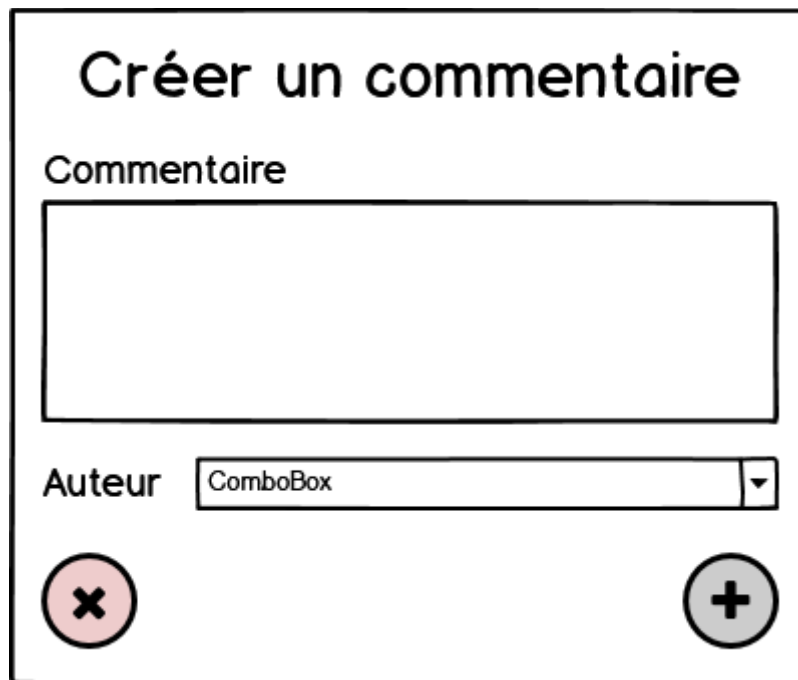
Date de fin 

Buttons: [X] [+]

Figure 19 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.

5.2.14 Pop-up des commentaires



Créer un commentaire

Commentaire

Auteur

ComboBox

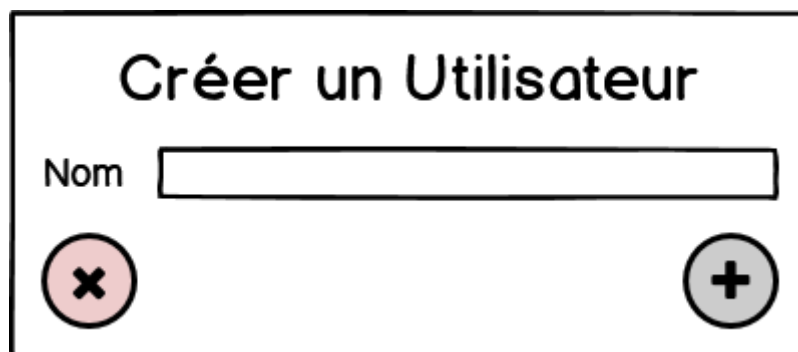
X

+

Figure 20 Maquette de pop-up de création de commentaire

Cette fenêtre permet de créer un commentaire. La date sera insérée automatiquement mais l'auteur devra être choisi parmi les utilisateurs assignés à la User Story. Le bouton « X » permet d'annuler la création. Le bouton « + » permet de confirmer la création si les champs requis ont été remplis.

5.2.15 Pop-up des utilisateurs



Créer un Utilisateur

Nom

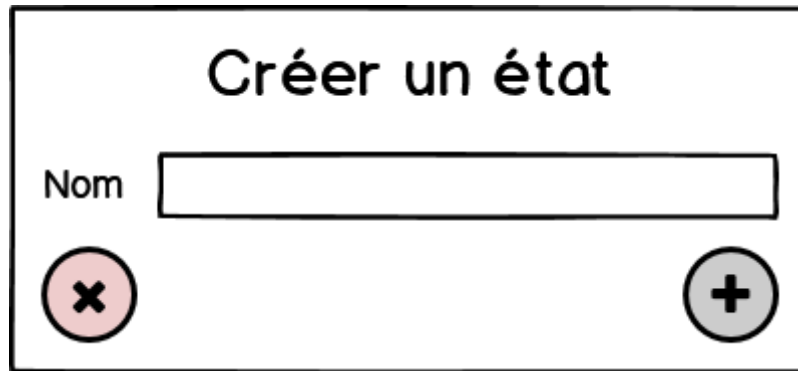
X

+

Figure 21 Maquette de pop-up de création d'utilisateur

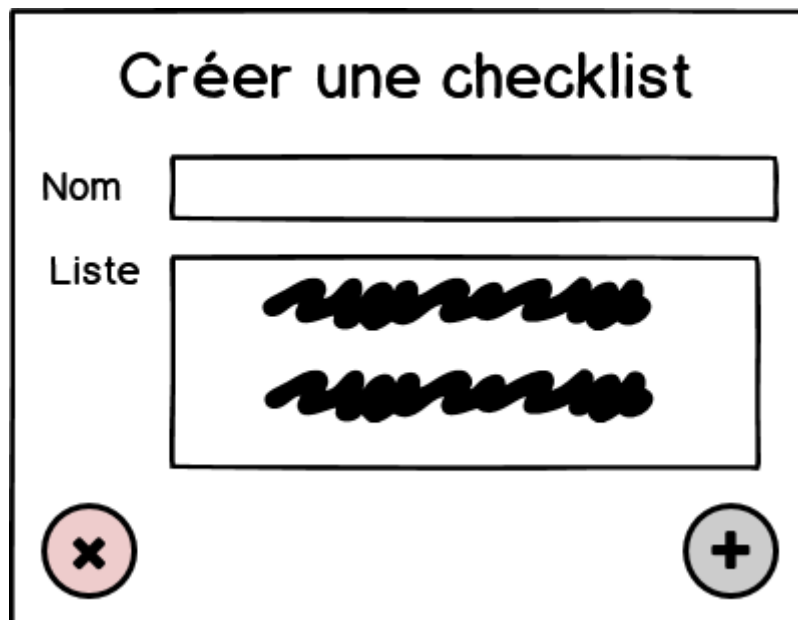
Cette fenêtre permet de créer un utilisateur en remplissant le nom de l'utilisateur. Le bouton « X » permet d'annuler la création. Le bouton « + » permet de confirmer la création si les champs requis ont été remplis.

5.2.16 Pop-up des états

A wireframe for a 'Créer un état' (Create a state) pop-up window. The window has a title bar at the top with the text 'Créer un état'. Below the title bar, there is a label 'Nom' followed by a text input field. At the bottom of the window, there are two circular buttons: a red button with a black 'X' on the left and a grey button with a black '+' on the right.*Figure 22 Maquette de pop-up de création d'état*

Cette fenêtre permet de remplir les informations nécessaires à la création d'un état. Il est nécessaire pour cela que le nom soit rempli. Le bouton « X » permet d'annuler la création. Le bouton « + » permet de confirmer la création si les champs requis ont été remplis.

5.2.17 Pop-up des checklists

A wireframe for a 'Créer une checklist' (Create a checklist) pop-up window. The window has a title bar at the top with the text 'Créer une checklist'. Below the title bar, there is a label 'Nom' followed by a text input field. Below that, there is a label 'Liste' followed by a text area containing two lines of scribbled text. At the bottom of the window, there are two circular buttons: a red button with a black 'X' on the left and a grey button with a black '+' on the right.*Figure 23 Maquette de pop-up de création checklist*

Cette fenêtre permet de remplir les informations nécessaires à la création d'une liste. Il est nécessaire pour cela que le nom soit rempli et que la liste ne soit pas vide. Le bouton « X » permet d'annuler la création. Le bouton « + » permet de confirmer la création si les champs requis ont été remplis.

5.2.18 Pop-up des fichiers

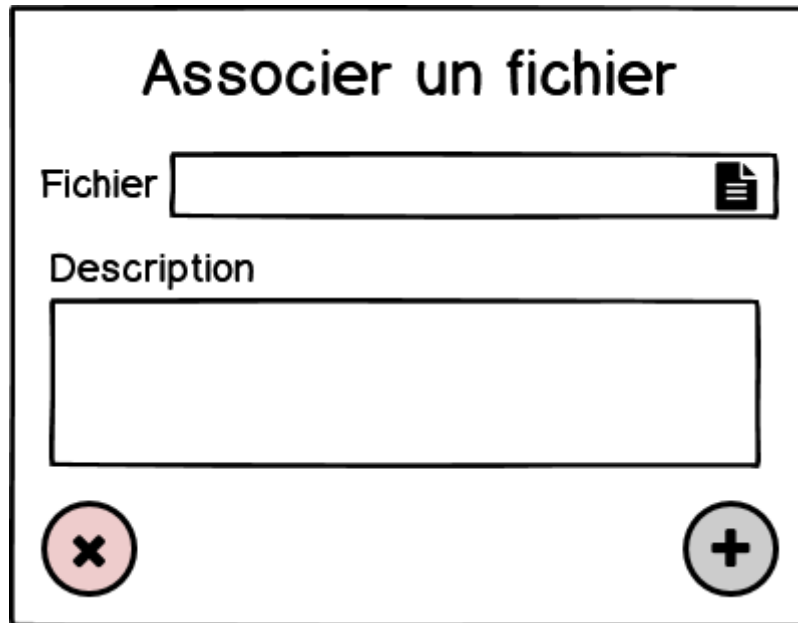
A wireframe of a 'Create File' pop-up window. The title 'Associer un fichier' is centered at the top. Below it is a label 'Fichier' followed by a text input field with a document icon on the right. Underneath is a label 'Description' followed by a larger text input field. At the bottom, there are two circular buttons: a red one with a black 'X' on the left and a grey one with a black '+' on the right.

Figure 24 Maquette de pop-up de création fichier

Cette fenêtre permet d'associer un fichier local avec une User Story. Il est nécessaire de choisir un fichier et de donner une description. Le bouton « X » permet d'annuler la création. Le bouton « + » permet de confirmer la création si les champs requis ont été remplis.

5.3 Fonctionnalités additionnelles

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

5.3.1 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.

Fonctionnalités

- Affichage dans le temps

5.3.2 Mindmap 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.

Fonctionnalités

- Créer, Modifier et Supprimer un mindmap
- Ajouter, Modifier et Supprimer des nœuds de mindmap

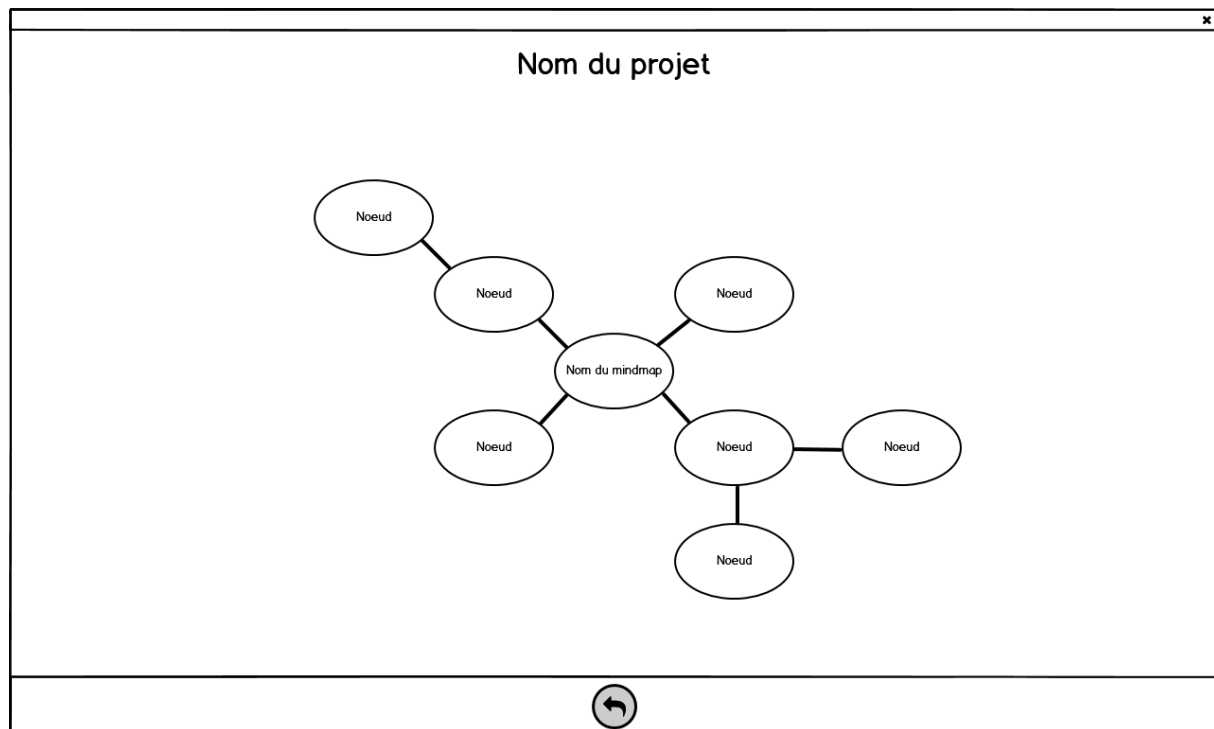


Figure 25 Maquette écran du mindmap

5.3.3 Protection de projets

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

Fonctionnalités

- Demande du mot de passe stocké dans la base de données

5.4 Manuel d'installation

5.4.1 Configuration de l'environnement

Afin de configurer l'environnement, il faut tout d'abord installer le moteur de base de données d'Access afin de pouvoir ouvrir les bases de données. Le moteur de base de données se situe sur cette page : <https://www.microsoft.com/fr-FR/download/details.aspx?id=39358>. De plus, le moteur de base de données étant créé uniquement en 64 bits, il n'est pas possible de lancer l'application en 32bits.

5.4.2 Activation du clavier virtuel

L'activation du mode tablette qui permet d'avoir le clavier virtuel quand un champ textuel est sélectionné n'est pas une possibilité. En effet, la présence de plus d'un écran empêche l'activation de cette fonctionnalité.

Cependant, il est possible d'activer cela sans activer le mode tablette. Cela requiert qu'aucun clavier ne soit connecté et qu'une option spécifique soit sélectionnée. Pour cela, il faut se rendre dans les **Paramètres > Périphériques > Saisie**. Dans ce menu, une option présente nécessite d'être activée. Elle est intitulée « **Afficher le clavier tactile lorsque vous n'êtes pas en mode tablette et qu'aucun clavier n'est connecté** »

5.4.3 Mise en place de la base de données

Afin d'avoir une base de données vide, trouvez le fichier « Empty.accdb » qui se situe dans le dossier « Installation ». C'est une base de données vierge contenant uniquement les données nécessaires.

Je vous conseille de copier ce fichier dans un endroit que vous retrouverez facilement et de le renommer avec un nom de votre choix.

6 Analyse Organique

6.1 Modèle de données

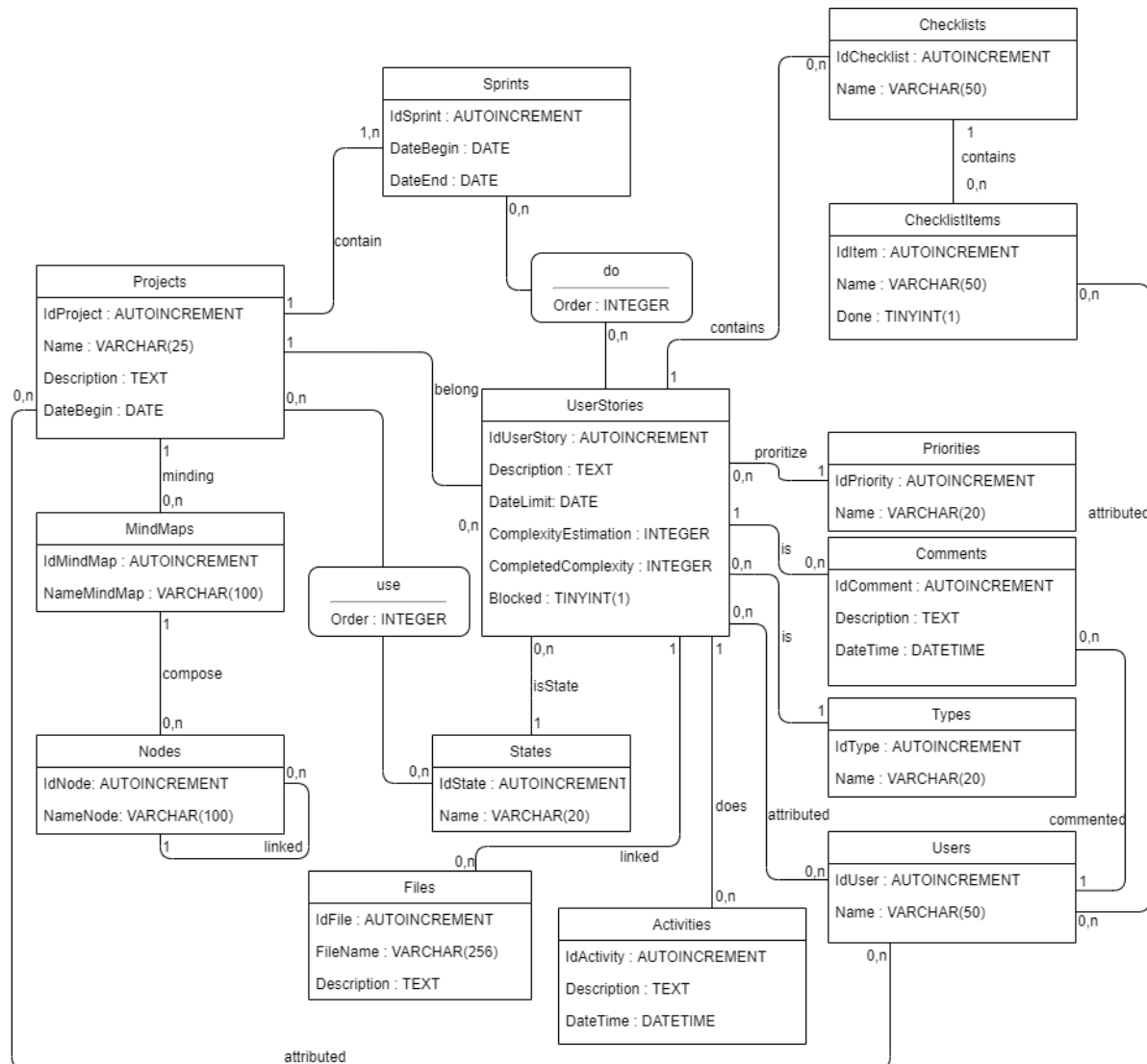


Figure 26 Modèle Conceptuel de Données

6.1.1 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.

6.1.2 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 à 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.

6.1.3 Table Sprint

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

Une table de liaison reliera la table *UserStory* à cette table afin de créer le contenu du sprint.

6.1.4 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.

6.1.5 Table MindMap

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

6.1.6 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.

6.1.7 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.

6.1.8 Table FileTypes

Cette table sert de constante.

Elle permet de garder en mémoire les types de fichiers.

6.1.9 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.

6.1.10 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.

6.1.11 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.

6.1.12 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.

6.1.13 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.

6.1.14 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.

6.2 Modèle de classes

Afin de séparer le projet au mieux, j'ai décidé d'utiliser une approche Modèle-Vue-Contrôleur. Nous avons d'abord les classes de données, représentées dans la Figure 27. Ensuite, les contrôleurs représentés dans la Figure 28. Finalement, les classes de la vue sont représentés dans la Figure 29.

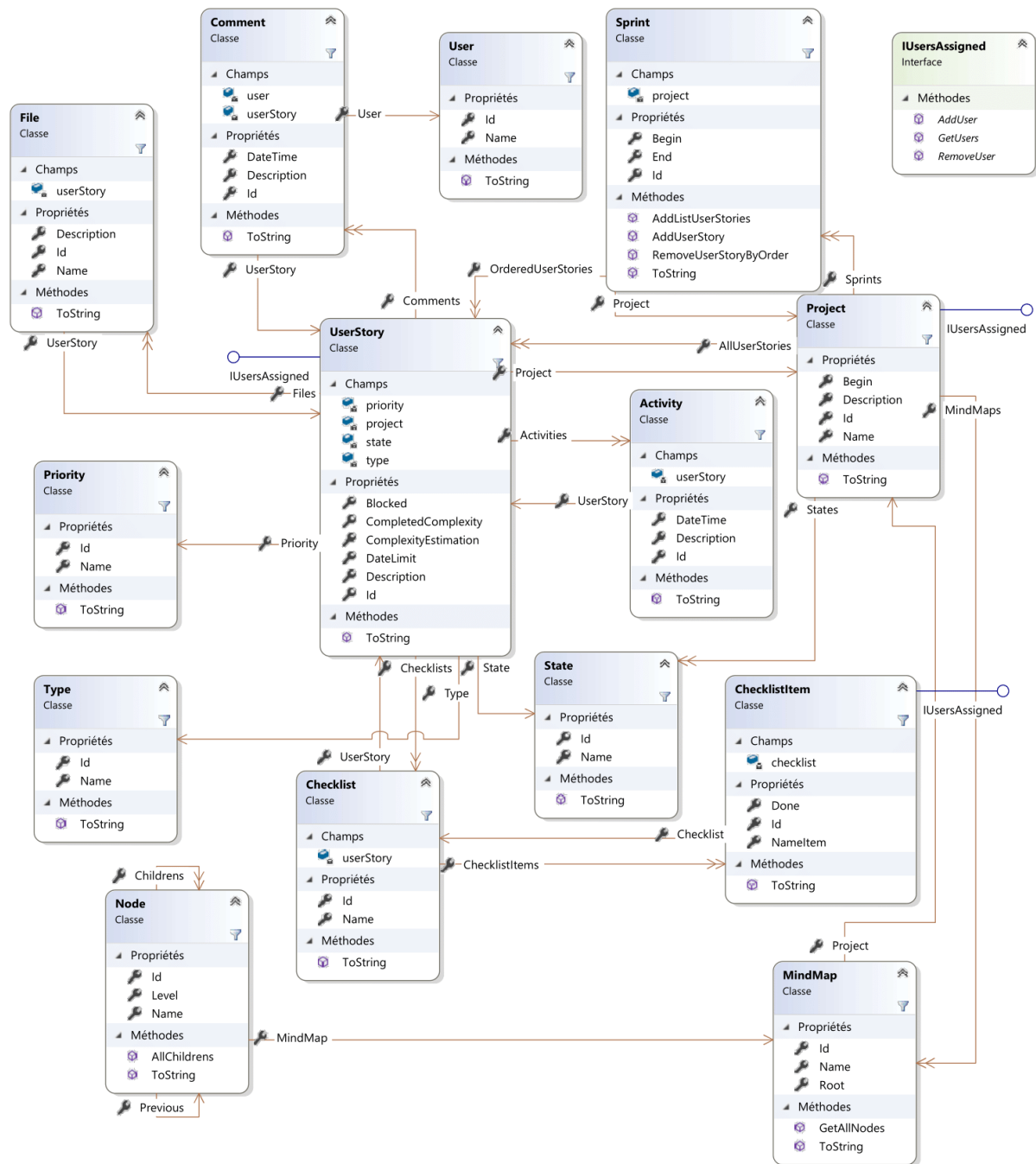


Figure 27 Diagramme des classes de données

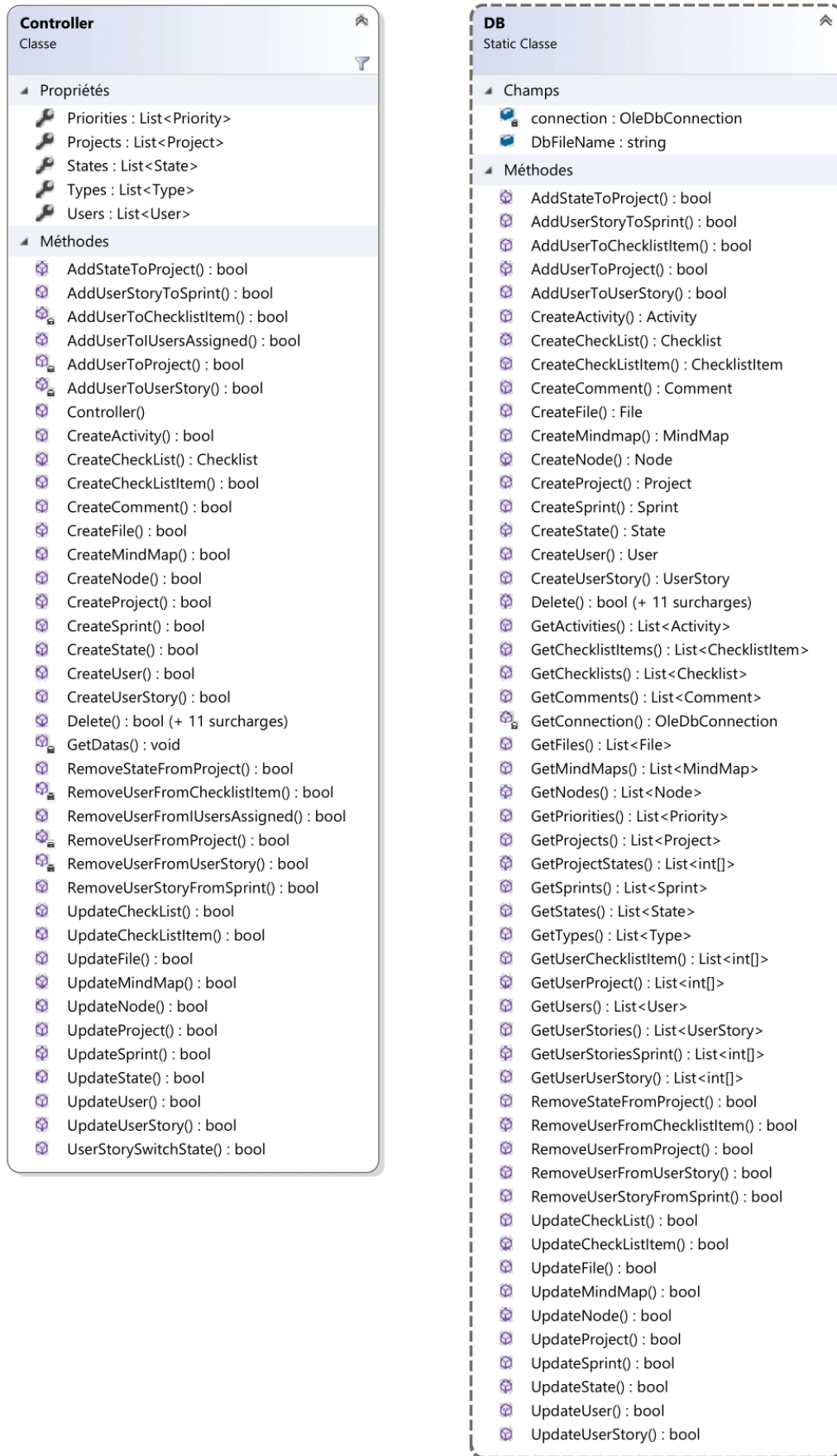


Figure 28 Diagramme des classes de contrôle

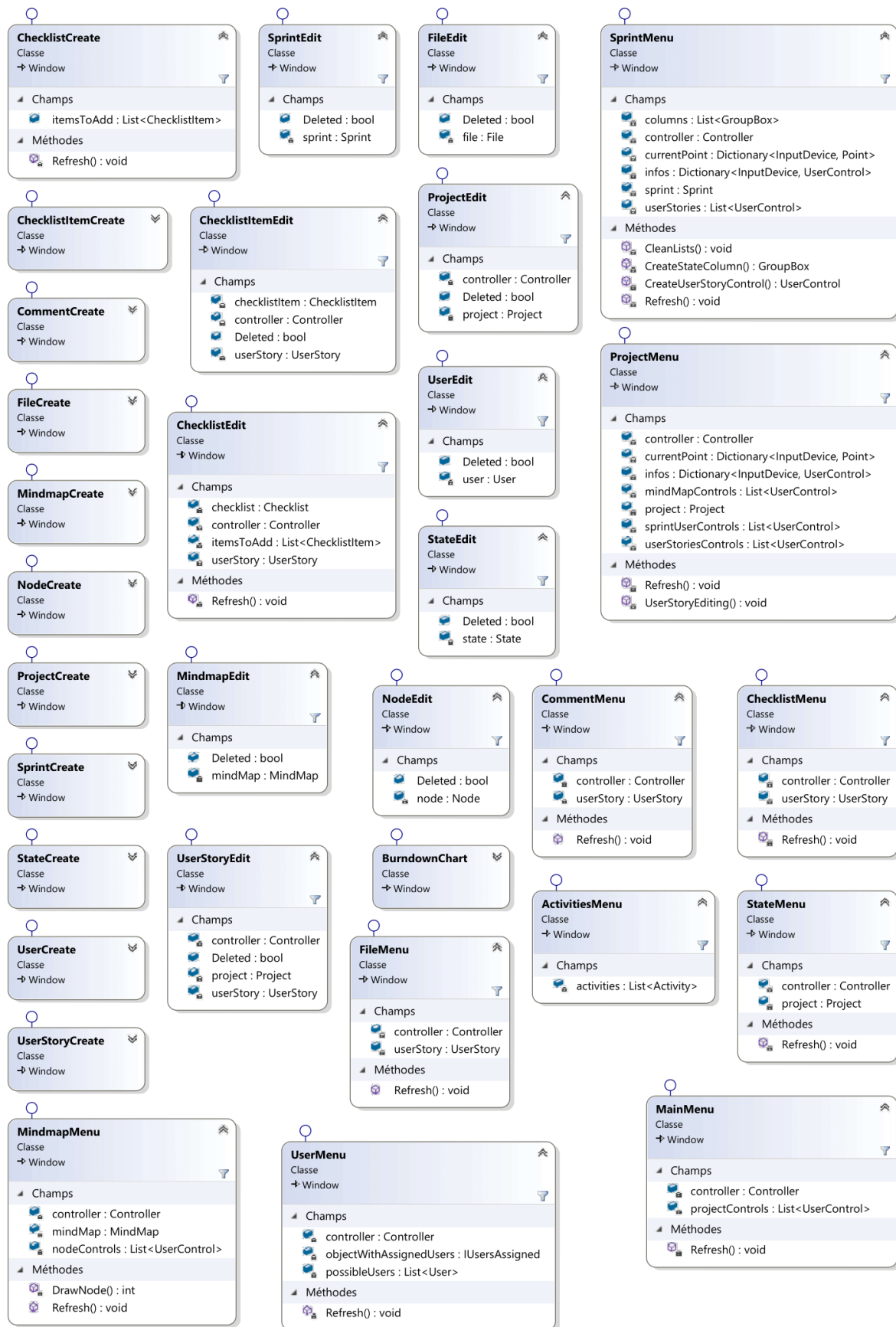


Figure 29 Diagramme des classes de vues

6.3 Apports personnels

Un projet complet de ce niveau ne peut pas être effectué dans son entièreté par une seule personne dans le temps qu'il m'a été donné. C'est pourquoi j'ai utilisé des outils préfabriqués afin de réaliser certaines tâches tel que draw.io pour réaliser certains diagrammes et Balsamiq pour réaliser les maquettes d'écrans. J'ai également utilisé les outils fournis avec Visual Studio pour les diagrammes de classes.

Pour ce qui est du code, je me suis basé sur WPF et sa gestion des évènements afin d'utiliser mes propres méthodes pour les évènements.

J'ai donc réalisé dans ce projet tout ce qui se situe dans les classes présentées dans le chapitre 6.2 Modèle de classes.

Pour ce qui est de la base de données, je l'ai réalisé de la conception à l'implémentation dans le programme en passant par la création dans Access.

6.4 Communications avec la base de données

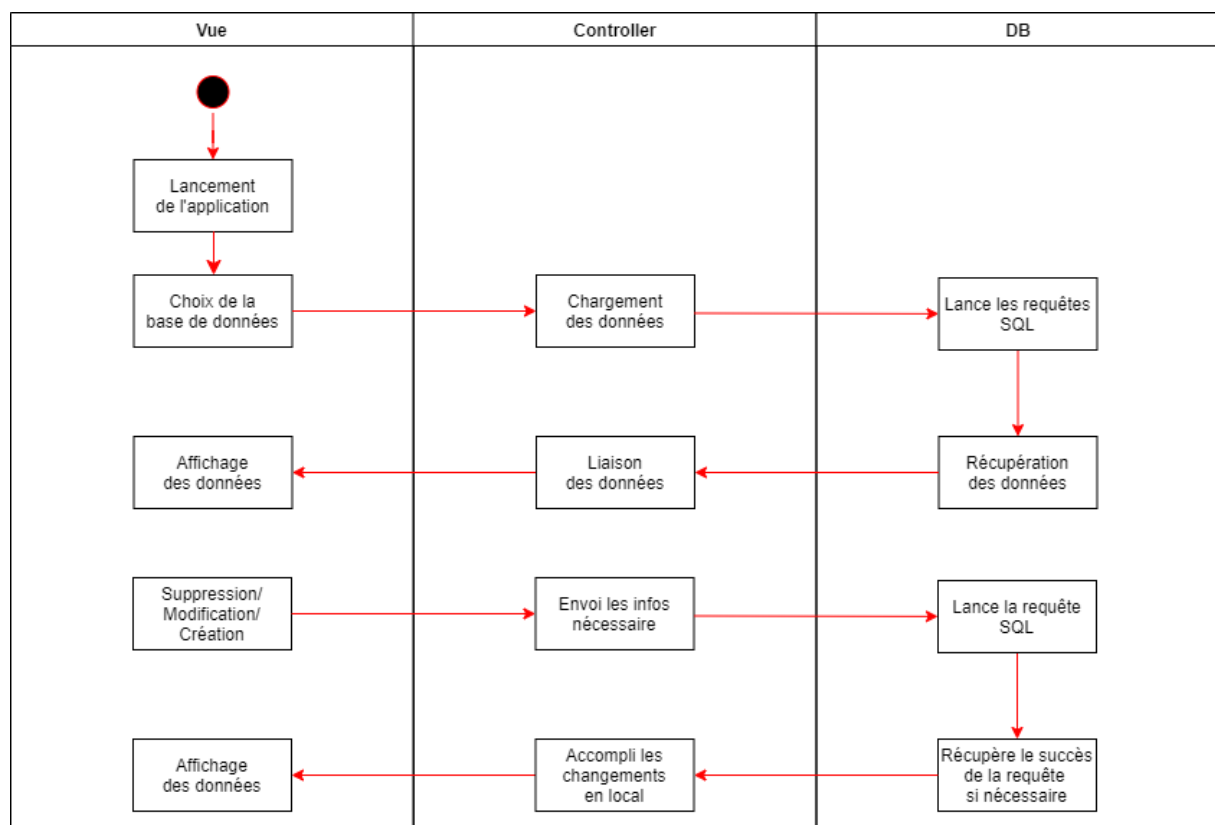


Figure 30 Diagramme de séquence de communication à la base de données

Afin de ne pas ralentir le projet avec les requêtes SQL d'acquisition de données, j'ai décidé de récupérer toutes les données au lancement de l'application. Cela permet d'avoir déjà toutes les données et de pouvoir traverser les vues d'une façon beaucoup plus fluide.

Cependant, afin de ne pas avoir de temps morts durant lesquels des milliers d'enregistrements se créent et se modifient, je n'ai pas trouvé pertinent de couper l'accès à la base de données totalement. J'accède à la base de données lors de la création, suppression et modification des enregistrements afin d'éviter des soucis en cas d'erreur fatale ou de coupure de courant.

6.5 Liaison Access

```
using System.Data.OleDb;

OleDbConnection connection = new
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + DbFileName
+ ";Persist Security Info=False;");

//Initialize variables
OleDbCommand cmd;
bool result;
//Open database, build sql statement and prepare
connection.Open();
cmd = connection.CreateCommand();
cmd.CommandText = "UPDATE TComments SET deletedFlag = ? WHERE IdComment = ?;";
cmd.Parameters.Add("deletedFlag", OleDbType.Boolean);
cmd.Parameters.Add("IdComment", OleDbType.Integer);
cmd.Parameters[0].Value = true;
cmd.Parameters[1].Value = comment.Id;
//Execute sql statement
cmd.Prepare();
result = cmd.ExecuteNonQuery() == 1;

//Close database
connection.Close();
return result;
```

Figure 31 Code d'utilisation de la liaison avec Access

La liaison avec une base de données Access utilise la classe de données de C#, OleDb. Il faut pour cela déclarer l'utilisation de cette classe dans les déclarations d'usage

Il faut ensuite créer un objet OleDbConnection qui permet d'établir la liaison entre l'application C# et la base de données Access.

J'ai inséré pour l'exemple la suppression d'un commentaire dans la base de données avec la suppression douce. Je commence par ouvrir la connexion. Je définis ensuite la requête SQL dans une chaîne de caractères.

J'ai ensuite créé les paramètres ainsi que leur type. Je leur attribue des valeurs et, finalement, j'exécute la requête et récupère le résultat (Dans ce cas, le nombre de lignes affecté égal à 1). A la fin je ferme la connexion et je retourne le résultat.

6.6 Glisser-déposer multipoint

Afin de réaliser le glisser-déposer, j'ai besoin de tableau associatifs qui se trouvent en C# sous la forme des dictionnaires. J'utilise une liste avec une position comme valeur et une avec les informations. (Voir Figure 32)

```
private readonly Dictionary<InputDevice, Point> currentPoint = new Dictionary<InputDevice, Point>();
private readonly Dictionary<InputDevice, UserControl> infos = new Dictionary<InputDevice, UserControl>();
```

Figure 32 Déclaration des variables nécessaire au glisser-déposer

Lors de l'appui sur un élément pouvant être glissé, je vérifie si l'identifiant est déjà existant car le programme peut réutiliser le même identifiant après avoir disparu. Après m'être assuré que l'identifiant n'existe pas déjà, je rajoute un enregistrement dans les listes. (Voir Figure 33)

```
private void UserStory_PreviewTouchDown(object sender, TouchEventArgs e)
{
    if (currentPoint.ContainsKey(e.Device))
    {
        currentPoint.Remove(e.Device);
        infos.Remove(e.Device);
    }

    currentPoint.Add(e.Device, e.GetTouchPoint(null).Position);
    infos.Add(e.Device, sender as UserControl);
}
```

Figure 33 Création d'un nouvel enregistrement dans le glisser-déposer

Lors du déplacement de l'élément, je mets à jour la position de l'élément. Cet exemple venant de la fenêtre de sprint, je mets à jour la taille de toutes les bordures à la taille par défaut. Ensuite, je parcours tous les éléments existants pour augmenter la taille des colonnes qu'un élément survole. Cela permet de voir visuellement la destination. (Voir Figure 34)

```
private void SprintMenu_PreviewTouchMove(object sender, TouchEventArgs e)
{
    //Update position of point
    if (currentPoint.ContainsKey(e.Device))
    {
        currentPoint[e.Device] = e.GetTouchPoint(null).Position;
    }
    //Set border thickness to default
    foreach (GroupBox col in columns)
    {
        col.BorderThickness = new Thickness(1);
    }
    //Verify each points to change border thickness if in bounds
    foreach (KeyValuePair<InputDevice, Point> keyValuePair in currentPoint)
    {
        Point p = keyValuePair.Value;
        foreach (GroupBox col in columns)
        {
            double leftBound = Canvas.GetLeft(col);
            double rightBound = leftBound + col.Width;
            double topBound = Canvas.GetTop(col);
            double bottomBound = topBound + col.Height;

            if (p.X > leftBound && p.X < rightBound &&
                p.Y > topBound && p.Y < bottomBound)
            {
                col.BorderThickness = new Thickness(3);
            }
        }
    }
}
```

Figure 34 Déplacement d'un élément

Finalement, lors du relâchement, je vérifie tout d'abord que l'identifiant de l'élément existe dans la liste. Ensuite, je récupère la colonne contenant la position de l'élément. Si une colonne est détectée, je change l'user story de colonne. Dans tous les cas, je finis en supprimant l'élément des listes et en rechargeant la fenêtre. (Voir Figure 35)

```

private void SprintMenu_PreviewTouchUp(object sender, TouchEventArgs e)
{
    if (currentPoint.ContainsKey(e.Device))
    {
        //Search the contained groupbox
        Point p = e.GetTouchPoint(null).Position;
        GroupBox gbxState = null;
        foreach (GroupBox col in columns)
        {
            double leftBound = Canvas.GetLeft(col);
            double rightBound = leftBound + col.Width;
            double topBound = Canvas.GetTop(col);
            double bottomBound = topBound + col.Height;
            if (p.X > leftBound && p.X < rightBound &&
                p.Y > topBound && p.Y < bottomBound)
            {
                gbxState = col;
                break;
            }
        }
        //if release in a groupbox, change state
        if (gbxState != null)
        {
            gbxState.BorderThickness = new Thickness(1);

            State state = gbxState.Tag as State;
            UserStory userStory = (infos[e.Device] as UserControl).Tag as UserStory;
            controller.UserStorySwitchState(userStory, state);
        }
        currentPoint.Remove(e.Device);
        infos.Remove(e.Device);
        Refresh();
    }
}

```

Figure 35 Relâchement de l'appui

7 Plan de tests

7.1 Test d'interface

N°	Description
Démarrage	
1	L'application compile et démarre.
2	L'application récupère les données de la base de données fournie.
3	L'application renvoie une erreur si la base de données n'est pas correcte.

N°	Description
Menu principal	
4	Les projets s'affichent sans erreur
5	Cliquer sur un projet amène sur la fenêtre de projet
6	Cliquer sur le bouton « + » amène sur le pop-up de création de projet
7	Cliquer sur le bouton « < » ferme l'application
7.5	Le pop-up crée en cas de confirmation et ne fait rien en cas d'annulation

N°	Description
Fenêtre de projet	
8	Cliquer sur une User Story ouvre la fenêtre de modification d'user stories
9	Cliquer sur un sprint ouvre la fenêtre de sprint
10	Cliquer sur le bouton « < » ramène au menu principal
11	Cliquer sur le bouton « Modifier » ouvre le pop-up de modification de projet
12	Cliquer sur le bouton « + » des sprint ouvre le pop-up de création de sprint
12.5	Le pop-up crée en cas de confirmation et ne fait rien en cas d'annulation
13	Cliquer sur le bouton « + » des user stories ouvre le pop-up de création de sprint
13.5	Le pop-up crée en cas de confirmation et ne fait rien en cas d'annulation

N°	Description
Fenêtre de sprint	
14	Le glisser déposer d'une User Story la change d'état dans la base et visuellement
15	Cliquer sur le bouton « Calendrier » ouvre la fenêtre du burndown chart
16	Cliquer sur le bouton « < » ramène à la fenêtre de projet
17	Le menu contextuel permet d'ajouter une user story
18	Cliquer sur une User story ouvre la fenêtre de modification d'user story

N°	Description
Fenêtre de burndown chart	
19	Affiche en rouge la droite idéale de progression de sprint
20	Affiche en noir la progression du sprint actuelle

N°	Description
Fenêtre de modification d'user story	
21	Cliquer sur le bouton « Fichiers » ouvre la fenêtre des fichiers
22	Cliquer sur le bouton « Commentaires » ouvre la fenêtre des commentaires
23	Cliquer sur le bouton « Checklists » ouvre la fenêtre des checklists
24	Cliquer sur le bouton « Activités » ouvre la fenêtre des activités
25	Cliquer sur le bouton « Utilisateurs assignés » ouvre la fenêtre des utilisateurs

N°	Description
Fenêtre de modification de projet	
26	Cliquer sur le bouton « Utilisateurs assignés » ouvre la fenêtre des utilisateurs
27	Cliquer sur le bouton « Colonnes » ouvre la fenêtre des états

N°	Description
Fenêtre des Fichiers	
28	Cliquer sur un Fichier ouvre le pop-up de modification de fichier
28.5	Le pop-up modifie en cas de confirmation et ne fait rien en cas d'annulation
29	Cliquer sur le bouton « + » ouvre le pop-up d'association de fichier
29.5	Le pop-up crée en cas de confirmation et ne fait rien en cas d'annulation
30	Cliquer sur le bouton « Retour » ferme la fenêtre

N°	Description
Fenêtre des commentaires	
31	Cliquer sur le bouton « + » ouvre le pop-up de création de commentaire
31.5	Le pop-up crée en cas de confirmation et ne fait rien en cas d'annulation
32	Cliquer sur le bouton « Retour » ferme la fenêtre

N°	Description
Fenêtre des checklists	
33	Cliquer sur une liste ouvre le pop-up de modification de liste
33.5	Le pop-up modifie en cas de confirmation et ne fait rien en cas d'annulation
34	Cliquer sur le bouton « + » ouvre le pop-up de création de liste.
34.5	Le pop-up crée en cas de confirmation et ne fait rien en cas d'annulation
35	Cliquer sur le bouton « Retour » ferme la fenêtre

N°	Description
Fenêtre des activités	
36	Cliquer sur le bouton « Retour » ferme la fenêtre

N°	Description
Fenêtre des états	
37	Cliquer sur le bouton « + » ouvre le pop-up de création d'état
37.5	Le pop-up crée en cas de confirmation et ne fait rien en cas d'annulation
38	Cliquer sur le bouton « > » fait passer l'élément sélectionné de gauche à droite
39	Cliquer sur le bouton « < » fait passer l'élément sélectionné de droite à gauche
40	Cliquer sur le bouton « Enregistrer » ferme la fenêtre
41	La fermeture de la fenêtre enregistre les modifications

N°	Description
Fenêtre des utilisateurs	
42	Cliquer sur le bouton « + » ouvre le pop-up de création d'état
42.5	Le pop-up crée en cas de confirmation et ne fait rien en cas d'annulation
43	Cliquer sur le bouton « > » fait passer l'élément sélectionné de gauche à droite
44	Cliquer sur le bouton « < » fait passer l'élément sélectionné de droite à gauche
45	Cliquer sur le bouton « Enregistrer » ferme la fenêtre
46	La fermeture de la fenêtre enregistre les modifications

7.2 Tests unitaires

J'ai créé des tests seulement pour les classes de contrôle. J'ai choisi de ne pas implémenter de fichier de test pour chaque classe car la classe de test « ControllerTests » vérifie les intégrités de chaque classe en même temps que leurs tests.

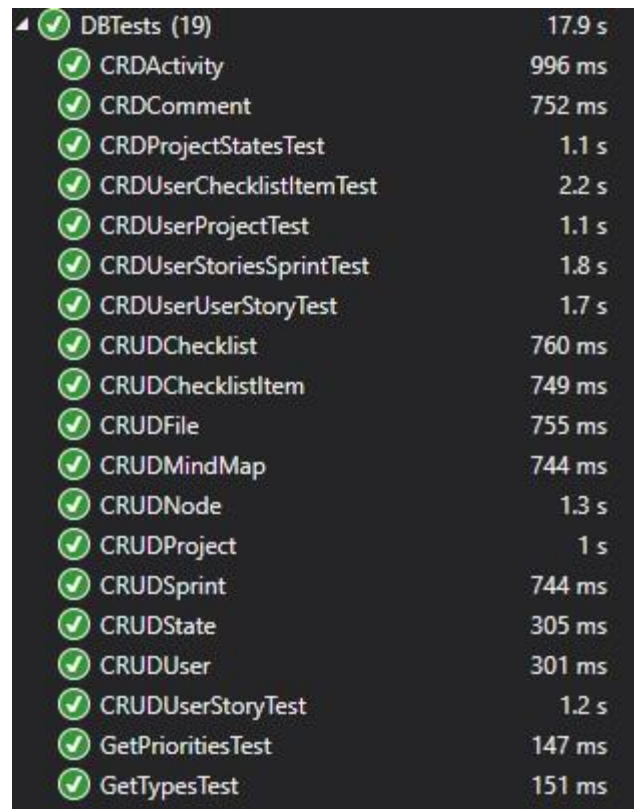


✓ ControllerTests (16)	1.2 min
✓ CRDActivity	5 s
✓ CRDComment	5 s
✓ CRDProjectStatesTest	4.3 s
✓ CRDUserUsersAssignedTest	6.6 s
✓ CRDUserStoriesSprintTest	5.2 s
✓ CRUDChecklist	4.7 s
✓ CRUDChecklistItem	5.1 s
✓ CRUDFile	4.8 s
✓ CRUDMindMap	4.1 s
✓ CRUDNode	4.5 s
✓ CRUDProject	3.7 s
✓ CRUDSprint	4.3 s
✓ CRUDState	3.8 s
✓ CRUDUser	4 s
✓ CRUDUserStoryTest	4.4 s
✓ ControllerTest	5.3 s

Figure 36 Résultat des tests sur la classe Controller

Comme vous pouvez le voir sur la Figure 36, Les tests ont été ordonnés par classe. J'ai préféré les ordonnées comme cela afin de ne pas avoir beaucoup de tests. Comme vous l'aurez peut-être remarqué, certaines classes ont besoin d'un CRUD complet alors que d'autres n'utilisent qu'un CRD.

Finalement, La méthode « ControllerTest » vérifie que le contrôleur récupère les données.



DBTests (19)	17.9 s
CRDActivity	996 ms
CRDComment	752 ms
CRDProjectStatesTest	1.1 s
CRDUserChecklistItemTest	2.2 s
CRDUserProjectTest	1.1 s
CRDUserStoriesSprintTest	1.8 s
CRDUserUserStoryTest	1.7 s
CRUDChecklist	760 ms
CRUDChecklistItem	749 ms
CRUDFile	755 ms
CRUDMindMap	744 ms
CRUDNode	1.3 s
CRUDProject	1 s
CRUDSprint	744 ms
CRUDState	305 ms
CRUDUser	301 ms
CRUDUserStoryTest	1.2 s
GetPrioritiesTest	147 ms
GetTypesTest	151 ms

Figure 37 Résultat des tests sur la classe DB

Comme vous pouvez le voir sur la Figure 37, j'ai utilisé des tests couvrant les opérations susceptibles de survenir réellement. Il y'a par exemple les activités qui sont CRD. Ceci montre que le test comprend la création, la lecture et la suppression sur cette table mais pas de modifications. En effet, la table activité ne sera jamais utilisée pour de la modification car cette option n'est pas pertinente.

Finalement, les méthodes dont le nom commence par « Get » sont les tables de constantes pour les types des User Stories ou pour les priorités qu'il est possible de donner.

8 Conclusion

J'ai rencontré des difficultés inédites durant ce projet. En effet, le confinement dû au Covid-19 m'a empêché de pouvoir travailler sur l'outil voulu. J'ai eu la chance que M. Garcia m'apporte un écran tactile afin de palier ce problème. De plus, le nombre de distractions existantes au sein de mon lieu de vie est élevé. J'ai donc dû me discipliner afin de travailler au mieux sur le projet. Je pense avoir réussi à fournir une application de qualité malgré ce point.

De plus, l'utilisation de WPF m'était totalement inconnue. J'ai bien sûr essayé de créer une application avant. Ce qui m'a aidé à surmonter cette difficulté fut les différents sites tel que microsoft ou stackoverflow. Beaucoup de gens ont déjà eu des problèmes similaires à ceux que j'ai eu et la communauté répondait avec des explications claires qui m'ont permis de mener à bien ce projet.

Ce projet n'est pas parfait et les rajouts possibles sont nombreux. On peut par exemple penser à l'ajout d'une gestion du projet dans le temps via un affichage sous la forme de gantt. Un changement afin d'ajouter le glisser-déposer à la souris pourrait également être réalisé. Il y a encore sûrement beaucoup de fonctionnalités auxquels je n'ai même pas pensé.

Finalement, je suis globalement content de mon projet. J'ai réussi à produire une application fonctionnelle malgré les difficultés rencontrées. J'ai même pu rajouter une fonctionnalité qui n'était pas prévue de base, la création de mindmap.

9 Table des figures

Figure 1 Capture d'écran de trello	5
Figure 2 Capture d'écran de JIRA.....	6
Figure 3 Capture d'écran de Ubikey.....	6
Figure 4 Capture d'écran de Kantree	7
Figure 5 Planning initial.....	8
Figure 6 Planning final.....	9
Figure 7 Maquette de la fenêtre de sélection de projet.....	11
Figure 8 Maquette de la fenêtre de projet	12
Figure 9 Maquette de la fenêtre de sprint	13
Figure 10 Maquette de la fenêtre du Burndown Chart	14
Figure 11 Maquette de la fenêtre des utilisateurs	14
Figure 12 Maquette de la fenêtre des états.....	15
Figure 13 Maquette de la fenêtre des activités.....	16
Figure 14 Maquette de la fenêtre des listes	16
Figure 15 Maquette de la fenêtre des commentaires	17
Figure 16 Maquette de la fenêtre de fichiers.....	18
Figure 17 Maquette du popup de création de projet.....	18
Figure 18 Maquette du popup de création de UserStory	19
Figure 19 Maquette du popup de création de sprint	19
Figure 20 Maquette de pop-up de création de commentaire	20
Figure 21 Maquette de pop-up de création d'utilisateur	20
Figure 22 Maquette de pop-up de création d'état	21
Figure 23 Maquette de pop-up de création checklist.....	21
Figure 24 Maquette de pop-up de création fichier	22
Figure 25 Maquette écran du mindmap	23
Figure 26 Modèle Conceptuel de Données	24
Figure 27 Diagramme des classes de données.....	27
Figure 28 Diagramme des classes de contrôle	28
Figure 29 Diagramme des classes de vues	29
Figure 30 Diagramme de séquence de communication à la base de données	30
Figure 31 Code d'utilisation de la liaison avec Access	31
Figure 32 Déclaration des variables nécessaire au glisser-déposer	31
Figure 33 Création d'un nouvel enregistrement dans le glisser-déposer	32
Figure 34 Déplacement d'un élément	33
Figure 35 Relâchement de l'appui	34
Figure 36 Résultat des tests sur la classe Controller.....	37
Figure 37 Résultat des tests sur la classe DB.....	38

10 Annexes