

Turbo Finder



« Projet d'application de recherche de fichier »

Rossier, Quentin
Les Autins 14
1584, Villars-Le-Grand
quentin.rossier@cpnv.ch

Philippe, Baumann
Chemin du château 20
1543, Grandcour
philippe.baumann@cpnv.ch

Table des matières

Analyse préliminaire.....	5
Introduction	5
Organisation	5
Objectifs.....	5
Planification initiale	6
Analyse	7
Cahier des charges détaillé.....	7
Fonctionnalités ajoutées pour le confort de l'utilisateur.....	7
Définition de l'audience	8
Analyse concurrentielle	8
Cas d'utilisation.....	9
Scenarios	10
Maquettes fonctionnelles.....	14
Etat « search »	14
Etat « Récent »	15
Etude de faisabilité.....	16
Nom de l'application	16
Stratégie de test.....	16
Cas-tests	17
Planification détaillées	24
Conception	26
Diagramme d'activité	26
Rendu visuel finale	27
Description du matérielle	27
Fonctionnement du système.....	28
UI :.....	28
LogHandler :	28
Searcher :	29
Nugets	29
Réalisation.....	29

Description de tests effectués	29
Erreurs restantes.....	30
Dossier d'archivage	30
Problèmes rencontrés	31
Livraison	32
Liste des documents fournis.....	32
Conclusion	32
Objectifs atteints et non atteints	32
Points positifs et négatifs sur le projet.....	33
Difficultés particulières	33
Suite et améliorations possible	33
Comparaison des plannings.....	34
Liste des problèmes restants.....	34
Commentaires personnel	34
Annexes.....	34
Sources – bibliographie.....	34
Archives du projet.....	35

Analyse préliminaire

Introduction

Le client nous a contacté, une équipe de deux techniciens en développement d'application, afin de développer une application de recherche de fichiers. Elle devra pouvoir rechercher dans le nom du fichier, sur une plage de dates, et à l'intérieur du fichier. Nous disposons d'environ 2 mois et demi afin de réaliser ce projet.

Organisation

Chef de projet 1 :	Nom :	BAUMANN	Prénom :	Philippe
		philippe.baumann@cpnv.ch		079 922 08 36
Chef de projet 2 :	Nom :	ROSSIER	Prénom :	Quentin
		Quentin.ROSSIER@cpnv.ch		079 860 95 90
Mandataire :	Nom :	CHEVILLAT	Prénom :	Jerome
		Jerome.CHEVILLAT@cpnv.ch		024/55 + 76079

Objectifs

Voici la liste des objectifs que le projet doit remplir afin de satisfaire notre client :

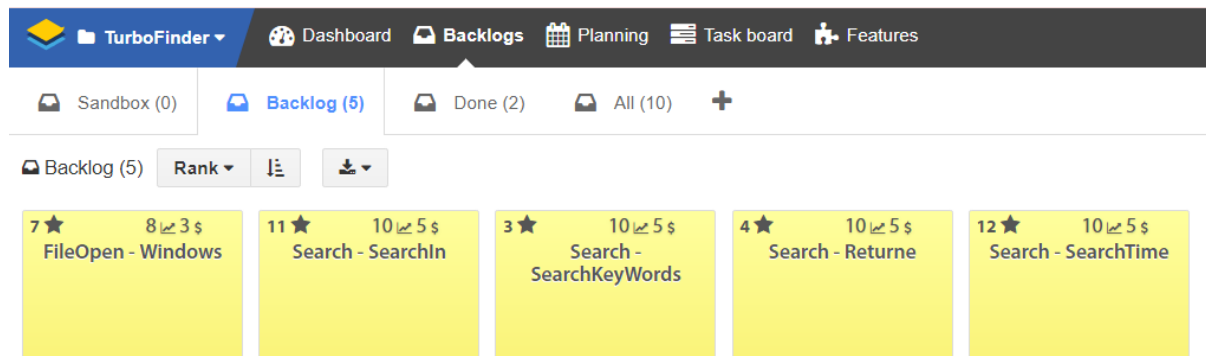
L'application doit pouvoir :

- Rechercher les documents par les attributs suivants :
 - La date
 - Le nom
 - L'auteur
 - La taille
 - L'extensions
- Rechercher à l'intérieur du fichier
- Ouvrir les fichiers depuis l'application (uniquement ceux qui disposent de logiciel par défaut)
- Ouvrir l'emplacement du fichier dans l'explorateur Windows
- Changer l'emplacement de la recherche

Tous les objectifs cité ci-dessus devront être terminé avant la date finale du projet.

Planification initiale

Nous planifierons notre projet sur Icescrum, un site permettant de travailler en méthodologie scrum de manière assez poussée.

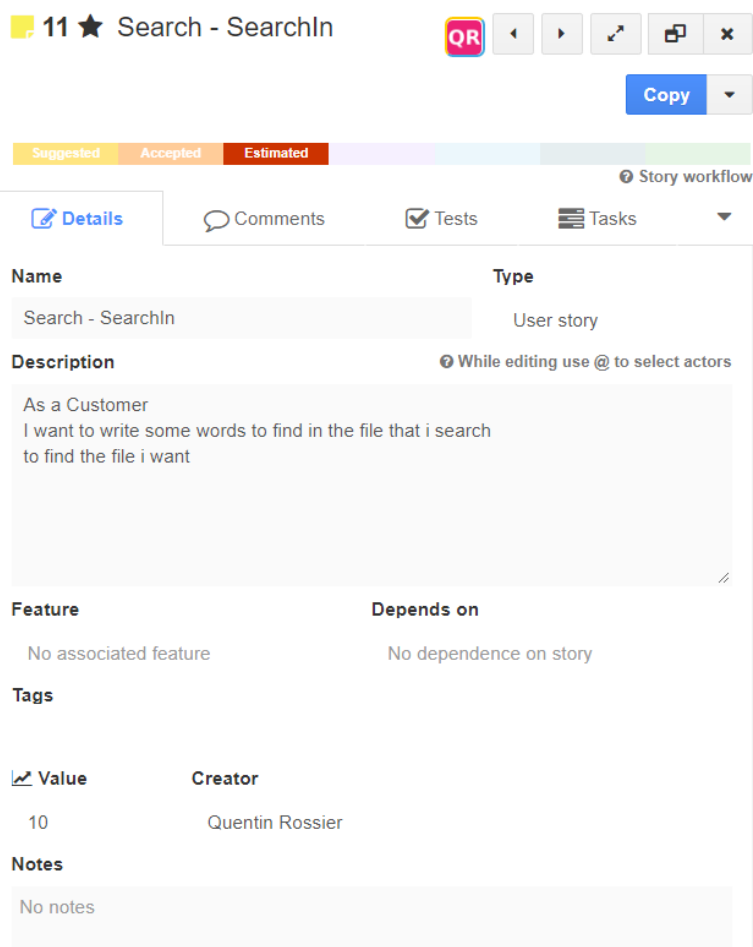


Nous avons créé des UsersStories dans un backlog que nous planifions par sprint. Un sprint dure une semaine ou l'on a 4h30 de travail à notre disposition. Nous effectuerons un meeting par semaine afin de discuter de l'avancée du projet et du projet en lui-même.

Chaque Userstory a un nom, une description de ce qu'elle doit faire et une valeur d'effort estimée par les intervenants.

Nous ajusterons les valeurs d'effort au fur et à mesure de l'avancée du projet.

L'onglet « Tasks » permet de définir quelles tâches réaliser exactement afin de terminer cette user story ce qui nous met au clair quant au travail à réaliser.



Grandes dates du projet

- | | |
|---|------------|
| • Début du projet | 31.08.2018 |
| • Présentation d'une partie fonctionnelle du projet | 24.11.2018 |
| • Remise de la réalisation et de la documentation | 14.12.2018 |
| • Présentation du projet final | 20.12.2018 |
| • Fin du projet | 20.12.2018 |

Analyse

Cahier des charges détaillé

Recherches :

L'utilisateur doit pouvoir trouver un ou plusieurs fichiers à l'aide d'un champ de recherche et de plusieurs filtres utilisables :

- Date
- Nom du fichier
- Auteur
- Taille
- Extensions

En plus de ses filtres, l'utilisateur doit aussi pouvoir taper du texte qui sera rechercher à l'intérieur des fichiers.

Endroit de recherche :

L'utilisateur pourra aussi changer le pointeur de la recherche dans un disque et un dossier souhaité.

Ouverture de fichiers :

Une fois les fichiers trouvés, l'utilisateur doit pouvoir ouvrir un fichier trouvé avec l'application par défaut de l'ordinateur, s'il y en a une. Où ouvrir le chemin du fichier dans l'explorateur Windows.

Fonctionnalités ajoutées pour le confort de l'utilisateur

Historique des recherches

L'utilisateur pourra afficher ses anciens résultats de recherche dans une page.

Prévisualisation du fichier

Une fois un fichier trouvé, l'utilisateur pourra le sélectionner afin d'en afficher ses informations.

Définition de l'audience

L'application servira au mandataire de la solution et à ses collègues

L'application servira au commanditaire de la solution, à sa famille et à ses proches qui pourront visionner les photos.

Il est possible que par la suite, le projet puisse être vendu comme application avec des modifications apportées pour le grand public.

Analyse concurrentielle

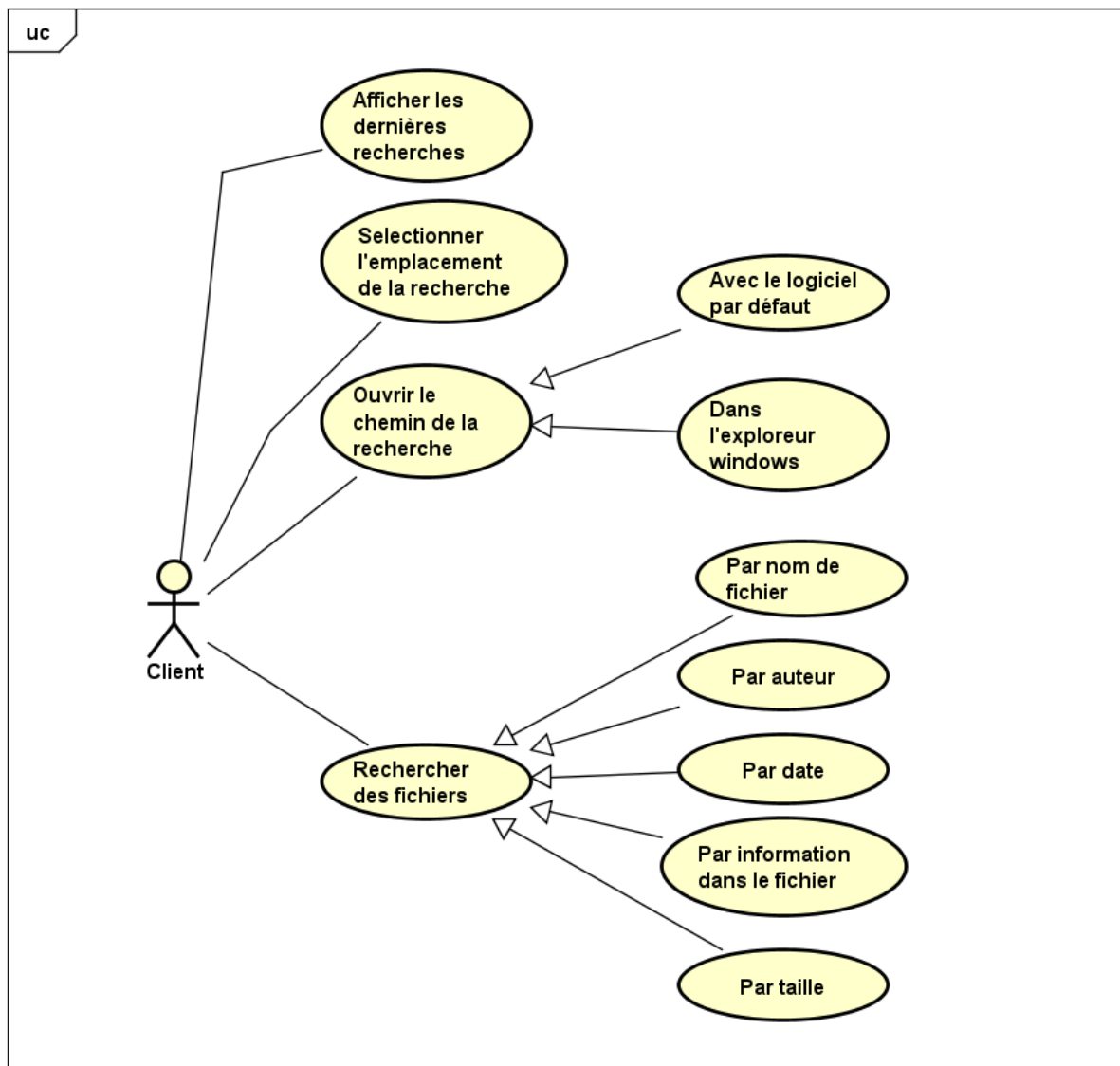
Explorateur Windows : permet de simple fonction de tri et de recherche, pose des problèmes lors de traitement d'un nombre important de fichier.

« SMF – Search my Files » est un logiciel qui permet de faire des recherches rapides et affichées en temps réelle. L'interface et la lisibilité est amoindrie et il ne peut pas accéder aux lecteurs réseau.

« Copernic Desktop Search » à une interface bien plus simple et épurée, garantissant une bonne lisibilité. Il est rapide et permet d'affiner grandement sa recherche en ajoutant des critères à choix. Il permet aussi de trier ses mails et d'enregistrer des recherches en favoris. Mais il ne permet pas non plus de chercher des informations sur des lecteurs réseau.

Cas d'utilisation

Voici le diagramme des cas d'utilisations de notre logiciel :



powered by Astah

Scenarios

Identifiant	Scenario_SelectFolder	
En tant que	Utilisateur	
Je veux	Sélectionner le dossier dans lequel effectuer ma recherche	
Pour	Trouver le fichier que je cherche	
Charge estimée	6h	
Priorité	Must	
Scénario	Étapes	Description
L'utilisateur parcourt le chemin du disque pour pointer sur le fichier ou il souhaite faire la recherche U : Utilisateur S : Système	1	U : Clique sur la liste déroulante
	2	S : Affiche plusieurs choix de disques et l'option « parcourir »
	3	U : Clique sur le bouton « parcourir »
	4	S : Ouvre l'explorateur Windows au dernier emplacement choisis sur l'application
	5	U : Parcourt l'explorateur Windows
	6	U : Sélectionne un dossier dans la liste
	7	S : Retourne sur l'application et affiche la liste des fichiers dans le dossier
Extensions	3a	<i>Choix dans la liste déroulante</i> U : Clique sur un choix de la liste déroulante
	4a	S : Affiche la liste des fichiers dans le dossier choisis

Identifiant	Scenario_SearchKeyWords	
En tant que	Utilisateur	
Je veux	Recherche un fichier avec un ou des mots clés	
Pour	Trouver le fichier que je cherche	
Charge estimée	5h	
Priorité	Must	
Scénario	Étapes	Description
L'utilisateur lance une recherche avec un ou des mots-clés U : Utilisateur S : Système	1	U : Entre un critère de recherche
	2	U : Sélection le filtre à appliquer
	3	U : Clique sur le bouton de recherche
	4	S : Lance une recherche basé sur les critères établis par l'utilisateurs
	5	S : Affiche le/les résultat(s) dans la liste
Extensions	3a	<i>Le champ de recherche est vide</i> S : Affiche un message d'erreur indiquant que le champs de recherche est vide

Scénario	Étapes	Description
Extensions	3b	<i>Certains champs comportent des caractères non autorisés</i> S : Affiche un message d'erreur indiquant que le champ de recherche comportent des caractères invalides

Identifiant	Scenario_SearchTime	
En tant que	Utilisateur	
Je veux	Recherche un fichier avec une plage de temps	
Pour	Trouver le fichier que je cherche	
Charge estimée	6h	
Priorité	Must	
Scénario	Étapes	Description
L'utilisateur lance une recherche avec une notion de temps U : Utilisateur S : Système	1	U : Entre un critère de recherche de type « date »
	2	U : Sélectionne le filtre à appliquer « date »
	3	U : Clique sur le bouton de recherche
	4	S : Lance une recherche basé sur les critères établis par l'utilisateurs
	5	S : Affiche le/les résultat(s) dans la liste
Extensions	3a	<i>Le champ de recherche est vide</i> S : Affiche un message d'erreur indiquant que le champs de recherche est vide
	3b	<i>Certains champs comportent des caractères non autorisés</i> S : Affiche un message d'erreur indiquant que le champ de recherche comportent des caractères invalides

Identifiant	Scenario_SearchIn	
En tant que	Utilisateur	
Je veux	Recherche un fichier en fonction de son contenu	
Pour	Trouver le fichier que je cherche	
Charge estimée	8h	
Priorité	Must	
Scénario	Étapes	Description
L'utilisateur lance une recherche avec une notion de temps	2	U : Sélection le filtre à appliquer « In »
	3	U : Clique sur le bouton de recherche

Scénario	Étapes	Description
U : Utilisateur S : Système	4	S : Lance une recherche basé sur les critères établis par l'utilisateurs
	5	S : Affiche le/les résultat(s) dans la liste
Extensions	3a	<i>Le champ de recherche est vide</i> S : Affiche un message d'erreur indiquant que le champs de recherche est vide
	3b	<i>Certains champs comportent des caractères non autorisés</i> S : Affiche un message d'erreur indiquant que le champ de recherche comportent des caractères invalides

Identifiant	Scenario_SearchAutor	
En tant que	Utilisateur	
Je veux	Trouver un fichier par son auteur	
Pour	Trouver un fichier en ne connaissant que son auteur	
Charge estimée	3h	
Priorité	Must	
Scénario	Étapes	Description
L'utilisateur sélectionne un fichier et ouvre l'explorateur Windows sur le fichier choisi U : Utilisateur S : Système	1	U : Entrer une date dans le champ de recherche
	2	U : Sélection le filtre à appliquer « Auteur »
	3	U : Clique sur le bouton de recherche
	4	S : Lance une recherche basé sur les critères établis par l'utilisateurs
	5	S : Affiche le/les résultat(s) dans la liste
Extensions	3a	<i>Le champ de recherche est vide</i> S : Affiche un message d'erreur indiquant que le champs de recherche est vide
	3b	<i>Certains champs comportent des caractères non autorisés</i> S : Affiche un message d'erreur indiquant que le champ de recherche comportent des caractères invalides

Identifiant	Scenario_SearchDate	
En tant que	Utilisateur	
Je veux	Trouver un fichier par sa date de création	
Pour	Trouver un fichier dont je ne me rappelle que de la date de création	
Charge estimée	3h	
Priorité	Must	
Scénario	Étapes	Description
L'utilisateur sélectionne un fichier et ouvre l'explorateur Windows sur le fichier choisi U : Utilisateur S : Système	1	U : Entrer une date dans le champ de recherche
	2	U : Sélection le filtre à appliquer « Date »
	3	U : Clique sur le bouton de recherche
	4	S : Lance une recherche basé sur les critères établis par l'utilisateurs
	5	S : Affiche le/les résultat(s) dans la liste
Extensions	3a	<i>Le champ de recherche est vide</i> S : Affiche un message d'erreur indiquant que le champs de recherche est vide
	3b	<i>Certains champs comportent des caractères non autorisés</i> S : Affiche un message d'erreur indiquant que le champ de recherche comportent des caractères invalides

Identifiant	Scenario_OpenWindows	
En tant que	Utilisateur	
Je veux	Ouvrir un fichier dans l'explorateur Windows	
Pour	Avoir accès au fichier dans l'explorateur Windows	
Charge estimée	3h	
Priorité	Must	
Scénario	Étapes	Description
L'utilisateur sélectionne un fichier et ouvre l'explorateur Windows sur le fichier choisi U : Utilisateur S : Système	1	U : Sélectionne un fichier recherché ou récent
	2	S : Surligne le fichier et affiche une prévisualisation de celui-ci
	3	U : Clique sur le bouton « Open in Explorer »
	4	S : Ouvre l'explorateur Windows et le place à l'endroit où le fichier est classé en le sélectionnant

Identifiant	Scenario_OpenAppli	
En tant que	Utilisateur	
Je veux	Ouvrir un fichier avec l'application par défaut	
Pour	Accéder au fichier rapidement	
Charge estimée	4h	
Priorité	Must	
Scénario	Étapes	Description
L'utilisateur ouvre le fichier sélectionné avec l'application par défaut suivant le format du fichier U : Utilisateur S : Système	1	U : Sélectionne un fichier recherché ou récent
	2	S : Surligne le fichier et affiche une prévisualisation de celui-ci
	3	U : Clique sur le bouton « Open»
	4	S : Ouvre l'application par défaut de l'utilisateur, correspondant au type de fichier choisi.
Extensions	4a	<i>Aucune application de base sélectionnée</i> S : Impossible d'ouvrir le fichier, affichage d'un message d'erreur.

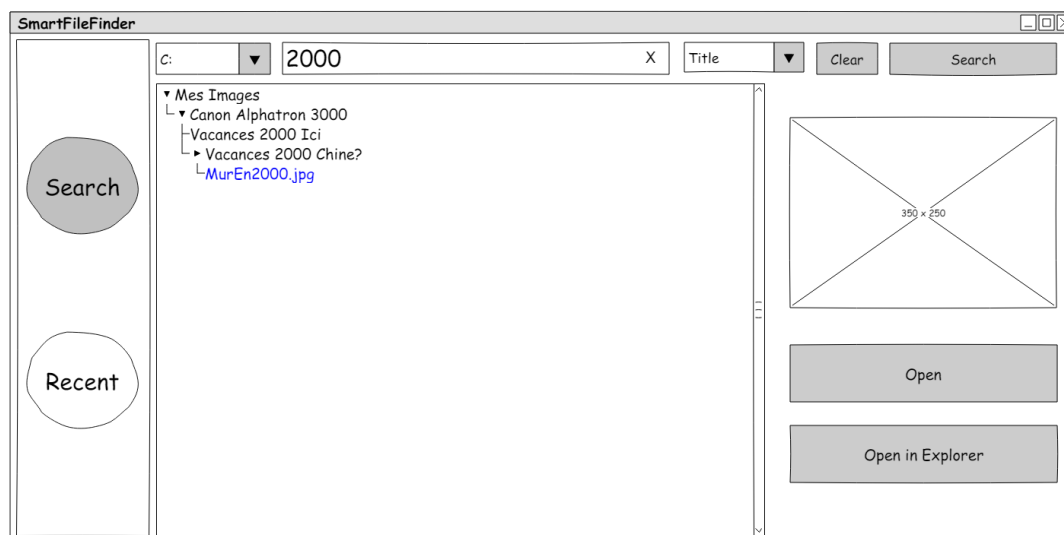
Identifiant	Scenario_Recent	
En tant que	Utilisateur	
Je veux	Afficher le résultat de mes anciennes recherches	
Pour	Ne pas à avoir faire encore la recherche et gagner du temps	
Charge estimée	6h	
Priorité	Should	
Scénario	Étapes	Description
L'utilisateur reprend un fichier déjà chercher. U : Utilisateur S : Système	1	U : Clique sur l'onglet « Recent »
	2	S : Affiche dans l'ordre chronologique les anciens résultat de recherche
Extensions	2a	<i>Aucune recherche n'a été effectuée</i> S : N'affiche rien

Maquettes fonctionnelles

Etat « search »

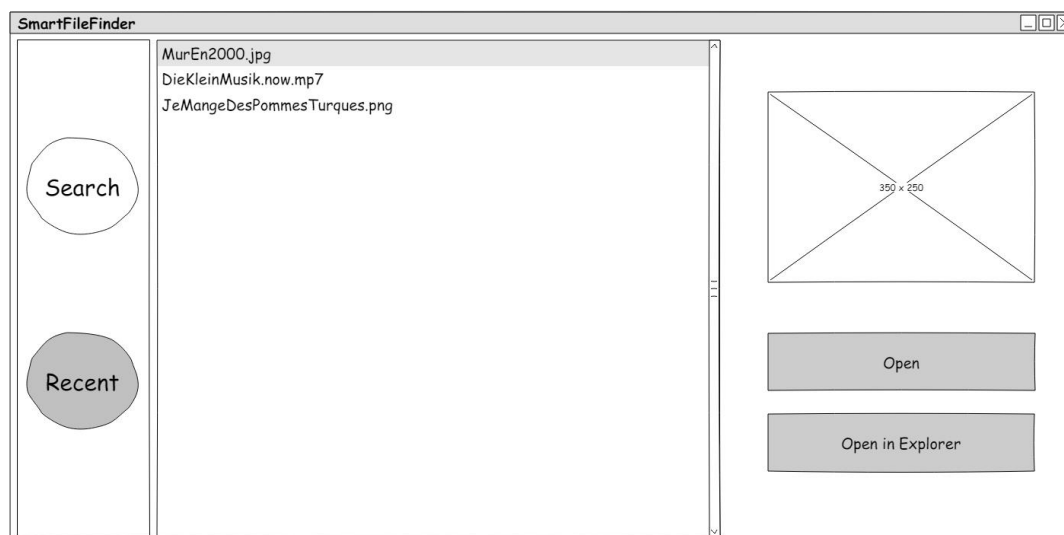
L'application est en mode « search ». Ici nous pouvons sélectionner le lecteur et le dossier ou nous voulons effectuer la recherche. Un affichage de ce qui se trouve dans

le dossier et du résultat de la recherche. Sur la droite, une liste déroulante avec les options de recherches, un affichage du fichier sélectionné, et la possibilité de l'ouvrir dans l'explorateur, ou avec l'application de base. Un bouton « clear » servira à effacer les anciens critères de recherche.



Etat « Récent »

Ici, l'application affiche les anciens résultats de recherche. Toujours avec la prévisualisation du fichier, et les deux possibilités d'ouverture.



Etude de faisabilité

Nous avons choisi de faire ce projet en C# car les deux programmeurs sont à l'aise avec le C#. Il n'y a donc pas de phase d'apprentissage pour les développeurs. Un gain de temps est ainsi effectué.

Après une discussion entre les deux développeurs, le projet est faisable au niveau du temps fournis et des demandes du client.

Seul la fonction de recherche risque de nous imposer un travail de recherche et de test plus important. La méthode LINQ nous permettrait de faire des recherches plus rapide malgré le grand nombre de fichier à chercher.

Nous explorerons aussi des méthodes de codages plus professionnelle qui laisserons la possibilité de modifier le logiciel déployé chez le client.

Nom de l'application

TurboFinder

Fait référence au turbo d'une voiture pour la vitesse et à la fonction recherche au cœur de l'application.

Le nom est déjà utilisé comme nom de site web :

<http://turbofinder.cermotor.com.pl/turbo-sklep/login.php>

Qui est un site polonais de vente de moteur et autre matériel mécanique.

Ça ne devrait pas poser de problème vu l'ancienneté et le sujet du site, bien différent de notre application. Le barrage de la langue est aussi évident. Les noms de domaine « TurboFinder. *** » Sont tous libre. Nous avons donc un large choix si nous souhaitons promouvoir notre application avec un site web.

Aucune autre application ne semble porter ce nom. Nous n'allons donc pas être confondu où prendre un nom indisponible.

Stratégie de test

Pour tester l'application, nous allons tester tour à tour les différentes fonctionnalités. Les tests vont suivre les cas d'utilisations en premier lieux pour vérifier le bon fonctionnement des fonctions.

Les tests seront effectués au fil du développement de l'application avec des tests unitaires. Lorsque le projet sera terminé, nous testerons l'ensemble des fonctionnalités tour à tour dans l'application finale.

Environnement de test :

- Un ordinateur privé

- Un ordinateur de notre environnement de travail

Données de test utilisés :

Notre propre disque physique et réseau avec son arborescence de fichier et dossier

Participant aux tests :

Les deux développeurs seront les deux testeurs de l'application.

	Unitaire	Intégration	Système
Fonctionnel	Effectué en cours de réalisation pour chaque développeur Aucun rapport n'est fournis	Le développeur fusionne les deux parties de code. Aucun rapport n'est fournis	L'application est compilée et lancée sur un post lambda Les cas-tests seront exécutés
Performance	Aucun	Aucun	Mise en place d'une arborescence de fichier volumineuse Cas-tests : 10'000 fichiers
Robustesse	Effectué en cours de réalisation pour chaque développeur Aucun rapport n'est fournis	Aucun	Effectué sur l'ensemble des user stories

Cas-tests

Nom	Search Name
Synopsis	Recherche du fichier « DocPHP »
Environnement	Visual Studio
Objectif	Valider la recherche de fichier par nom
Données	Un fichier « DocPHP.docx » Placé dans des dossiers
Prérequis	
Auteur	

	Actions	Résultat attendu	Résultat
1	Lancement de l'application et écriture de « DocPHP » sur la barre de recherche	L'application s'ouvre et le champ de recherche affiche « DocPHP »	OK
2	Clic sur le bouton « search » et choisit un dossier où se trouve, dessous, le fichier « DocPHP »	Affiche le fichier « DocPHP » et les autres fichiers qui correspondent au critère de recherche	OK

Nous sous-entendons que l'application se lance correctement à partir de maintenant.

Nom	Search In Docx		
Synopsis	Recherche du texte dans un fichier docx		
Environnement	Visual Studio		
Objectif	Valider la recherche dans le contenu d'un fichier word		
Données	Un fichier « DocPHP.docx » Placé dans des dossiers avec l'intitulé recherché		
Prérequis			
Auteur			
	Actions	Résultat attendu	Résultat
1	Ecrire « Cahier des charges détaillé » Dans le champs de recherche et sélectionné dans la liste box « interne »	Aucune intervention du logiciel	OK
2	Clic sur le bouton « search » et choisit un dossier où se trouve, dessous, le fichier contenant le champs de recherche	Affiche les fichiers qui contiennent le critère de recherche	OK

Nom	Search In XLS		
Synopsis	Recherche du texte dans un fichier .xls		
Environnement	Visual Studio		
Objectif	Valider la recherche dans le contenu d'un fichier Excel		
Données	Un fichier « DocPHP.docx » Placé dans des dossiers avec l'intitulé recherché		
Prérequis			
Auteur			
	Actions	Résultat attendu	Résultat
1	Ecrire « Cahier des charges détaillé » Dans le champs de recherche et sélectionné dans la liste box « interne »	Aucune intervention du logiciel	OK
2	Clic sur le bouton « search » et choisit un dossier ou se trouve, dessous, le fichier contenant le champs de recherche	Affiche les fichiers qui contient le critère de recherche	OK

Nom	Search In PDF		
Synopsis	Recherche du texte dans un fichier PDF		
Environnement	Visual Studio		
Objectif	Valider la recherche dans le contenu d'un fichier PDF		
Données	Un fichier « DocPHP.docx » Placé dans des dossiers avec l'intitulé recherché		
Prérequis			
Auteur			
	Actions	Résultat attendu	Résultat
1	Ecrire « Cahier des charges détaillé » Dans le champs de recherche et sélectionné dans la liste box « interne »	Aucune intervention du logiciel	OK
2	Clic sur le bouton « search » et choisit un dossier ou se trouve, dessous, le fichier contenant le champs de recherche	Affiche les fichiers qui contient le critère de recherche	OK

Nom	Search Autor		
Synopsis	Recherche d'un fichier selon son auteur		
Environnement	Visual Studio		
Objectif	Valider la recherche par auteur		
Données	Un fichier « DocPHP.docx » Placé dans des dossiers avec un auteur définit		
Prérequis			
Auteur			
	Actions	Résultat attendu	Résultat
1	Ecrire «Quentin» Dans le champs de recherche et sélectionné dans la liste box « interne »	Aucune intervention du logiciel	KO
2	Clic sur le bouton « search » et choisit un dossier ou se trouve, dessous, le fichier avec l'auteur « Quentin »	Affiche le fichier « DocPHP » et les autres fichier qui a été créé par l'auteur « Quentin »	KO

Nom	Search Date		
Synopsis	Recherche d'un fichier selon une plage de date		
Environnement	Visual Studio		
Objectif	Valider la recherche dans le contenu d'un fichier		
Données	Un fichier « DocPHP.docx » Placé dans des dossiers crée le jour même		
Prérequis			
Auteur			
	Action	Résultat attendu	Résultat
1	Sélectionner « Date » sur l'interface	Affiche un choix de plage de dates	OK
2	Ecrit un jour, un mois, ou une années	Aucune intervention du logiciel	OK
3	Clic sur le bouton « search » et sélectionne le dossier ou le fichier a été crée	Affiche-le, ou les fichiers créés durant cette plage de date. Le fichier « DocPHP » Doit s'y retrouver	OK

Nom	Search Extension		
Synopsis	Recherche d'un fichier selon une extension		
Environnement	Visual Studio		
Objectif	Valider la recherche d'un fichier via son extension		
Données	Un fichier « DocPHP.docx » Placé dans des dossiers		
Prérequis			
Auteur			
	Actions	Résultat attendu	Résultat
1	Sélectionner « Extensions » sur l'interface et écrit « docx » sur la plage de recherche	Aucune intervention du logiciel	OK
2	Clic sur le bouton « search » et choisit un dossier où se trouve, dessous, le fichier contenant le champs de recherche	Affiche les fichiers « .docx », et donc notre fichier « DocPHP.docx »	OK

Nom	Search Size		
Synopsis	Recherche d'un fichier selon la taille du fichier		
Environnement	Visual Studio		
Objectif	Valider la recherche d'un fichier via son extension		
Données	Un fichier « DocPHP.docx » Placé dans des dossiers, vérifié ça taille (430 Ko)		
Prérequis			
Auteur			
	Actions	Résultat attendu	Résultat
1	Sélectionner « Taille » sur l'interface et écrit « 430 » sur la plage de recherche	Aucune intervention du logiciel	KO
2	Clic sur le bouton « search » et choisit un dossier où se trouve, dessous, le fichier contenant le champs de recherche	Affiche tous les fichiers ayant une taille d'environ 430 Ko	KO

Nom	Search Without		
Synopsis	Recherche sans aucun critère		
	Visual Studio		
Objectif	Valider la recherche sans critère écrit		
Données	Une arborescence de fichier quelconque		
Prérequis			
Auteur			

	Action	Résultat attendu	Résultat
1	Clic sur le bouton search et sélectionne une arborescence de fichier	Affiche tous les fichiers et sous fichier de cette arborescence	OK

Nom	Search Big		
Synopsis	Recherche dans une grande arborescence de fichiers		
	Visual Studio		
Objectif	Valider la robustesse du logiciel		
Données	Une arborescence de fichier contenant plusieurs sous dossier pour environ 10'000 fichiers au total		
Prérequis			
Auteur			
	Action	Résultat attendu	Résultat
1	Clic sur le bouton search et sélectionner l'arborescence de fichier crée pour le test	Affiche tous les fichiers et sous fichier de cette arborescence.	OK

Nom	Log Reading		
Synopsis	Lecture du log dans la page « LogView »		
	Visual Studio		
Objectif	Valider la fonction du log		
Données	Un fichier log avec déjà quelques recherche effectué et notée dedans		
Prérequis			
Auteur			
	Action	Résultat attendu	Résultat
1	Clic sur le bouton « LogView »	Affiche tous les fichiers et sous fichier de cette arborescence.	KO
2	Clic sur le bouton « LogView »	Affiche un fichier de log contenant une entête datant de la dernière recherche, et ensuite contenant le résultat de la recherche	KO

Nom	Log Writing		
Synopsis	Écriture de la recherche dans le fichier de log		
	Visual Studio		
Objectif	Valider la fonction d'écriture dans le fichier de log		
Données	Une arborescence de fichier quelconque		
Prérequis			
Auteur			
	Action	Résultat attendu	Résultat
1	Clic sur le bouton search et sélectionne une arborescence de fichier	Affiche tous les fichiers et sous fichier de cette arborescence.	KO
2	Clic sur le bouton « LogView »	Affiche un fichier de log contenant des entêtes de date avec le résultat de la recherche effectuée à se jour	KO

Nom	Open Explorer		
Synopsis	Ouvrir le fichier sélectionné dans l'explorateur Windows		
	Visual Studio		
Objectif	Valider la fonction ouverture de fichier dans l'explorateur		
Données	Un fichier recherché, dans le log ou après une recherche		
Prérequis			
Auteur			
	Action	Résultat attendu	Résultat
1	Clic sur un chemin de fichier	Surligne la ligne du fichier sélectionné	OK
2	Clic sur « Open in Explorer »	Ouvre le fichier dans l'explorateur Windows	OK

Nom	Open Logiciel		
Synopsis	Ouvrir le fichier sélectionné avec le logiciel par défaut		
	Visual Studio		
Objectif	Valider la fonction ouverture de fichier via le logiciel		
Données	Un fichier recherché, dans le log ou après une recherche		
Prérequis			
Auteur			

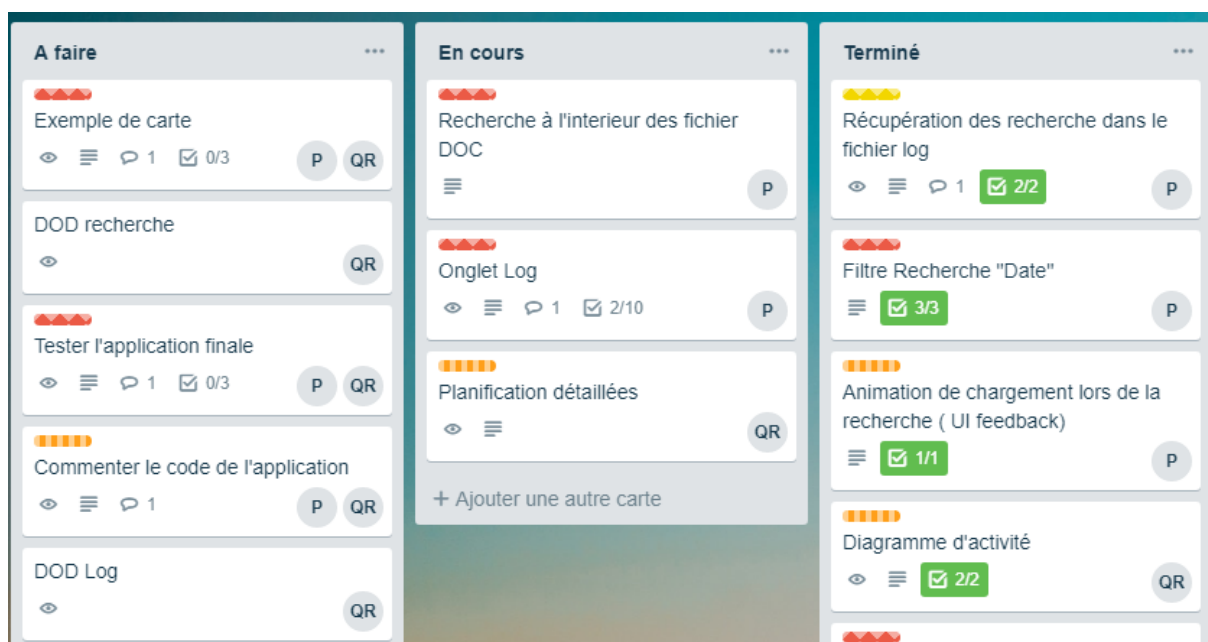
	Action	Résultat attendu	Résultat
1	Clic sur un chemin de fichier	Surligne la ligne du fichier sélectionné	OK
2	Clic sur « Open »	Ouvre le fichier sélectionné avec le logiciel par défaut de l'utilisateur	OK

Planification détaillées

Malheureusement, notre première méthode de management de projet sur Icescrum s'est avérée infructueuse. Cette manière de faire ne convenait pas à notre rythme de travail trop lente et espacée dans le temps. Nous prenions trop de temps pour nous organiser, remplir les informations sur Icescrum et mettre à jour notre avancée du projet.

Nous avons donc changé de méthode d'organisation durant le projet pour opter pour une organisation sur Trello. Nous avons écrit un document décrivant notre organisation sur Trello. Il se trouve dans les annexes de cette documentation

Nous avons donc fait trois colonnes. A faire, en cours et terminé. Cela ne permet de remplir les cartes à réaliser, de sélectionner les plus importantes et les réalisées.



Voici un exemple de carte :

Onglet Log

Dans la liste [En cours](#)

MEMBRES

ÉTIQUETTES

P

+

Must

+

Description

Éditer

En tant qu'utilisateur

Je veux pouvoir voir mes anciennes recherches et ne pas être bloqué par une mauvaise manipulation

Pour gagner du temps sur les fichiers dont j'ai souvent besoin

Test

Cacher les éléments complétés

Supprimer...

33%

✓

Création du fichier log s'il n'existe pas

Gestion de l'affichage si aucun fichier log à été crée et que l'on veut voir les logs

Impossible d'ouvrir la ligne de date

Sélection du dossier ouvert par défaut

Gestion des erreurs

✓

Lecture du log dans les fichiers interne à l'application

Ajouter un élément...

AJOUTER À LA CARTE

Membres

Étiquettes

Checklist

Date limite

Pièce jointe

POWER-UPS

Obtenir des Power-ups

ACTIONS

Déplacer

Copier

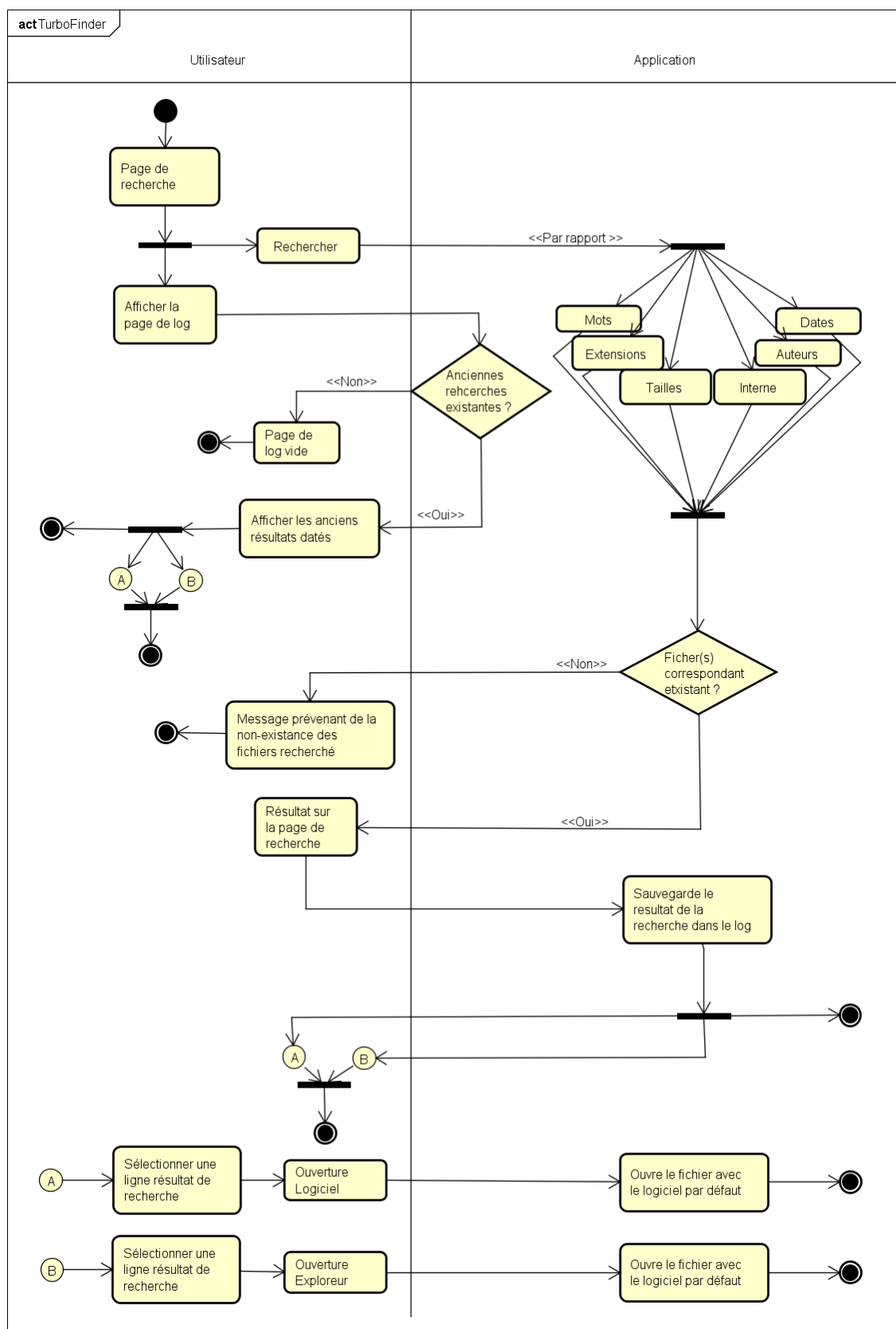
Suivre

Archiver

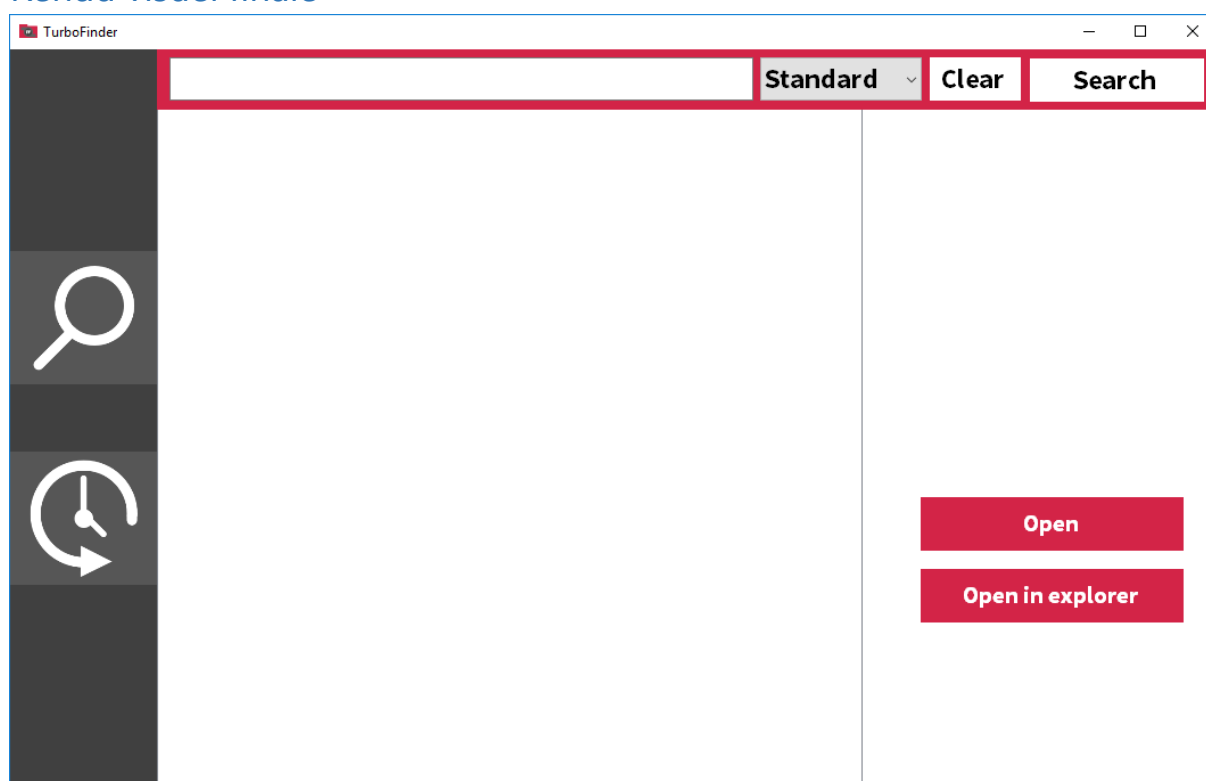
Nous décrivons le but de la carte dans la description de celle-ci. Une checklist comprenant la liste des étapes à réaliser pour valider la carte et d'avoir un suivis visuelle pour tous les membres du groupes.

Conception

Diagramme d'activité



Rendu visuel finale



Voici la page de base de notre application. Nous avons choisi un style assez épuré, permettant l'affichage de nombreuses recherches au centre de l'application.

La liste déroulante permet de sélectionner tous les type de recherche que l'on souhaite.

Le bouton « Clear » efface les anciennes recherches ainsi que le champ de recherche

Le bouton « Search » permet de sélectionner un dossier ou effectuer sa recherche et lance ensuite la recherche avec les filtres choisis par l'utilisateur.

Description du matérielle

Liste du matériel HW :

Le CPNV met à notre disposition un poste informatique complet ainsi qu'une connexion internet. Il n'y a pas d'équipement hardware spécifique requis pour la réalisation du projet.

Liste des logiciels utilisés :

- Visual studio 2017

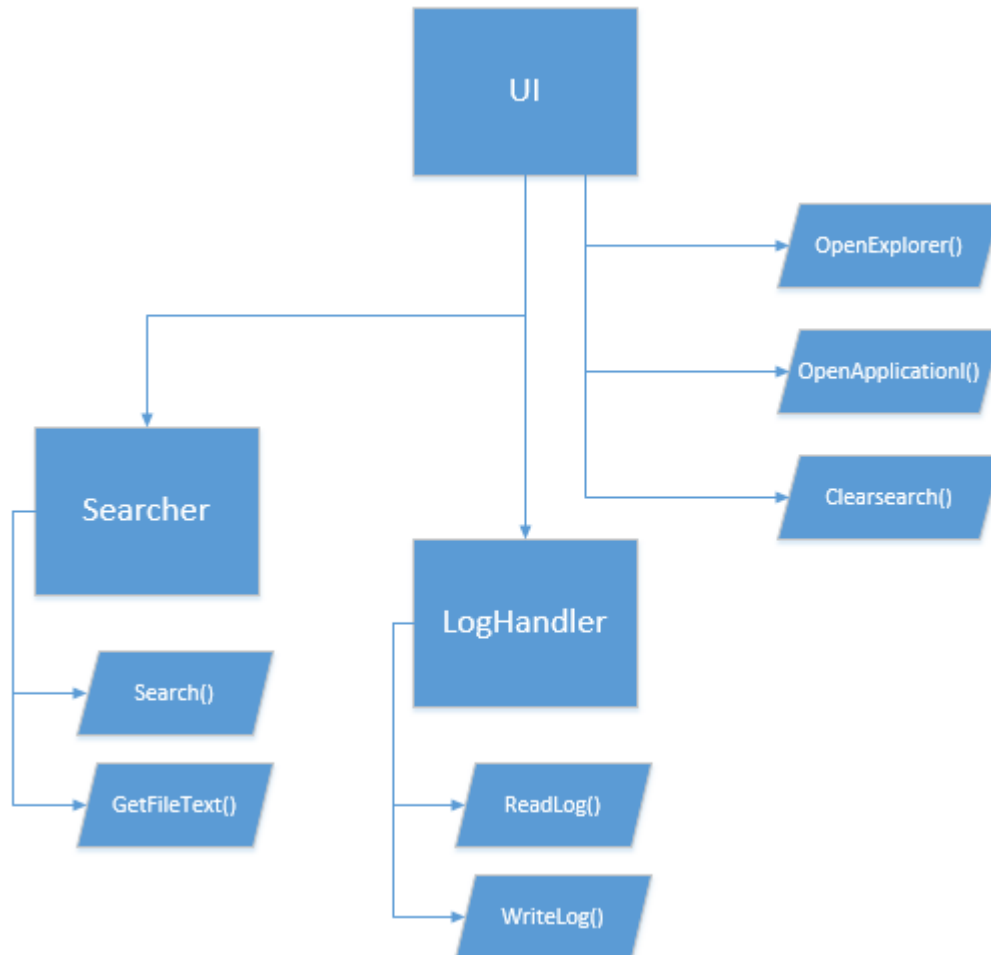
Sera notre logiciel de développement

- Suite office 2016 (Word, Excel, Outlook)
- Trello
- Astah

- GitHub Desktop

Nous servirons à documenter le projet et à nous organiser dans la réalisation de celui-ci

Fonctionnement du système



UI :

Est l'interface utilisateur. Elle contient ses trois fonctionnalités :

- **OpenExplorer()**
 - Permet d'ouvrir le chemin de la recherche dans l'explorateur Windows
- **OpenApplication) :**
 - Permet d'ouvrir le chemin de la recherche avec l'application par défaut définie sur l'ordinateur
- **ClearSearch() :**
 - Permet de vider la recherche

LogHandler :

Est la classe qui s'occupe du log des recherches

- ReadLog() :
 - Affiche le contenu du fichier log
- WriteLog() :
 - Ecrit dans le fichier de log, le résultat de la recherche, avec un entête daté

Searcher :

Est la classe qui s'occupe des fonctions de recherches

- Search() :
 - Recherche dans l'arborescence de fichier en fonction des critères de recherche
- GetFileText() :
 - Lit le contenu du fichier

Nugets

Voici les NuGets que nous avons utilisé durant le projet :

- WinFormAnimation
 - Permet d'animer les contrôles et les valeurs.
- Circular Progress Bar
 - Utilise le WinFormAnimation, permet de réaliser une barre de progression circulaire. Utiliser durant le chargement de la recherche
- DocX
 - Permet de manipuler les documents Word. Ce qui nous permet d'aller les lire et rechercher dedans

Réalisation

Description de tests effectués

Synthèse des cas-tests

Cas-test	Date	Personne	Résultat	Commentaire
Search Name	13.12.2018	Philippe	OK	
Search Date	13.12.2018	Philippe	OK	Façon de faire bien spécifique
Search Extension	13.12.2018	Philippe	OK	
Search In	13.12.2018	Philippe	OK	Fonctionne seulement avec PDF, docx, xls.
Search size	13.12.2018	Quentin	KO	Fonctionnalité non incluse
Search Big	13.12.2018	Quentin	OK	Environ 15 secondes
Search without	13.12.2018	Quentin	OK	
Open Logiciel	13.12.2018	Quentin	OK	

Cas-test	Date	Personne	Résultat	Commentaire
Open Explorer	13.12.2018	Quentin	OK	
Log Writing	13.12.2018	Quentin	KO	Fonctionnalité non incluse
Log Reading	13.12.2018	Quentin	KO	Fonctionnalité non incluse

Erreurs restantes

Si nous cherchons dans une arborescence de fichier ou l'application n'as pas les droits d'aller chercher, la recherche se termine et affiche un message d'erreur.

Mis à part cette erreur, dans les fonctions implémentées, il n'y a pas d'erreur restante. Le logiciel gère les potentiels mauvaises utilisations du client et ainsi, n'arrête pas inopinément son fonctionnement.

La lecture et l'écriture d'un log de recherche as été mis en attente afin de terminer le projet avec les objectifs principaux en état de marche. Ce n'était pas dans notre cahier des charges mais nous pensions que c'était une amélioration utile à l'expérience de l'utilisateur.

Les recherches par auteur et taille du fichier ont été mise au dépourvu car nous jugions plus important les autres fonctionnalités.

Dossier d'archivage

Le projet a été créé sur GitHub afin que les deux participants puissent retrouver le projet et effectuer des modifications depuis n'importe quel lieu de travail. Voici l'arborescence de dossiers que nous avons créé pour ce projet :

```

├── COD
│   ├── TurboFinder
│   │   ├── packages                Liste des packages de l'application
│   │   │   ├── CircularProgressBar.2.3.0  NuGet barre de progression ronde
│   │   │   ├── DocX.1.2.1                NuGet Gestion des fichiers words
│   │   │   └── WinFormAnimation.1.3.0.0    NuGet pour les animations de console
│   │   └── TurboFinder
│   │       ├── bin
│   │       │   ├── Debug
│   │       │   └── app.publish

```

—Properties	
—Resources	
—DOC	Dossier de toute la documentation
—img	Dossier des images utilisés lors du projet
—icone	
—logo	
—Pic	Captures d'écran et images utilisées pour le projet
—Shema	Captures des schémas crée pour le projet
—wireframe	Captures des wireframe de l'application
—Schéma	Liste des Schémas crée pour la documentation
Readme	Fichier de présentation du GitHub

Problèmes rencontrés

Problème	Analyse	Solution
GithubDesktop nous à poser problèmes lors de merge de fichier. Impossible de pull ou de remettre en place la branche	Les conflits doit être résolus pour permettre le pull de la branche	Par manque de temps, nous avons recrée la branche
Les recherches peuvent être interrompues par le manque d'accès au fichier de l'arborescence recherchée	Il nous manque les droits sur certains fichiers	Aucune solution n'a été trouvée.
Anticiper l'arborescence Excel pour extraire le texte dans les bonnes cellules	Tentative de récupération du texte dans un fichier Excel	Après plusieurs heures de recherche, jointure des tutoriels et documentations afin de réaliser une solution
Crée un SplashScreen sans que la form main n'apparaisse plus		Il manquait une réactivation de la form main (le SplashScreen était optionnel)
Apprendre le Linq	Du temps nécessaire afin d'apprendre le bon	Suivie de tuto et de documentations diverses

	fonctionnement de cette librairie	
--	--------------------------------------	--

Livraison

Liste des documents fournis

En annexe, sur GitHub vous trouverez les documents suivant :

- Journaux de bord des deux participants
- Le manuel d'installation
- Le manuel d'utilisation
- La documentation
- Le code source du projet
- Le dossier d'installation du projet

Voici le lien du GitHub de l'équipe

https://github.com/PhilCPNV/MAW_filefinder_QRR_PBN

Conclusion

Objectifs atteints et non atteints

Objectifs atteints :

La majorité des objectifs cités dans le cahier des charges ont été réalisés. Malheureusement, d'autres n'ont pas été accomplis.

Les recherches par :

- Nom
- Date
- Extensions
- Le contenu des fichiers PDF, Excel, Word

L'ouverture des fichiers trouvés :

- Dans l'explorateur Windows
- Avec l'application de base du client

Et l'on peut changer l'emplacement de la recherche pour des disques locaux ou sur le réseau.

Objectifs non atteints :

Les recherches par :

- Auteur

- Taille de fichier

N'ont pas été réalisées.

Points positifs et négatifs sur le projet

Point positif :

Ce projet nous a permis d'approfondir nos connaissances avec le fonctionnement des classes en C#.

La gestion un peu maladroite de ce projet nous a montré les difficultés et les pièges de cette étape de travail dans la réalisation d'un projet.

La découverte de la puissance du module Linq et ces diverses utilités.

Comprendre le fonctionnement et réussi à utiliser la méthode « async ».

Point négatif :

Le peu de temps par semaine allouée à ce projet, et la petite taille de l'équipe qui nous permet uniquement de faire une « simulation » de gestion de projet agile.

Ne pas avoir réussi à intégrer la fonctionnalité du log de recherche.

Ne pas pouvoir rendre les objets Winform de manière à charger et s'intégrer de manière asynchrone ce qui me/nous poussera à utiliser de préférence WPF à partir de maintenant.

Difficultés particulières

L'organisation du projet a été une difficulté en soit. Trouver le bon mélange entre trop ou trop peu d'organisation pour ne pas que le projet parte en vrille, ou au contraire soit bloqué par d'interminable obligations d'organisations.

Trouver un module non payant pour lire les fichiers Word et Excel. Ainsi que de réussir à l'intégrer au projet.

Rendre la tâche de chargement asynchrone.

Apprendre à utiliser la librairie Linq

Suite et améliorations possible

- Nous pourrions terminer le projet comme cité dans le cahier des charges si le client souhaite avoir ses options.
- La finition de l'option du log de recherche serait envisageable.
- Un bouton qui annule la recherche en cours.

- Trouver une solution pour le manque de droit qui empêche la recherche dans certaines arborescences.
- Crée un retour d'erreur lors de la lecture du contenu d'un fichier
- Crée un gestionnaire de plug-in pour les extensions de type spécifique (image, vidéo...)
- Amélioré la fonction de lecture pour les PDF
- Ajouter explicitement la lecture des tags métadonnées

Comparaison des plannings

Due au changement de méthode de planification du projet, nous n'avons pas de comparaison d'avant et après la réalisation du projet. Notre Trello comporte la suite des cartes que nous devions réalisés pour terminer le projet depuis le changement depuis l'escrum.

Liste des problèmes restants

Les seules fonctionnalités qu'il manque depuis le cahier des charges sont les fonctionnalités de recherche par auteur ou par taille de fichier. Autrement, la partie du log que nous souhaitions ajouter en tant qu'amélioration pour le client n'a pas été réalisée.

Commentaires personnel

Ce projet nous a montré que réaliser une organisation correcte sur le long terme, demande du travail et de la rigueur et que ce n'est pas si facile de trouver les bonnes méthodes d'organisation dès le départ. Cela dépend du projet, de la taille de l'équipe et cette expérience enrichissante nous permettra d'éviter de refaire les mêmes erreurs.

Annexes

Sources – bibliographie

Nous avons consulté différents tutoriel et résolution de problème sur ses sites :

Site visualisé pour des informations sur la gestion de projet :

GitHub

<https://cpnv-es.github.io/> Documentation sur le bon usage de GitHub, Trello et autre.

Site visualisé pour des informations sur la programmation :

Open classroom

Stackoverflow

Documentation officiel de Microsoft

Reading Word and Excel Files tanks to

<https://stackoverflow.com/questions/23102010/open-xml-reading-from-excel-file>

<https://stackoverflow.com/questions/3633615/how-can-i-read-docx-file>

Async Tasking :

<https://stackoverflow.com/questions/18013523/when-correctly-use-task-run-and-when-just-async-await>

Linq :

<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/how-to-query-the-contents-of-files-in-a-folder-lin>

Access Control:

<https://docs.microsoft.com/en-us/dotnet/api/system.io.directory.getaccesscontrol?view=netframework-4.7.2>

Image Metadata:

<https://docs.microsoft.com/en-us/dotnet/framework/winforms/advanced/how-to-read-image-metadata>

Checking modified date of files:

<https://stackoverflow.com/questions/3360324/checking-when-a-file-was-last-modified-in-c-sharp-on-windows-os>

Get one single ListView SelectedItem

<https://stackoverflow.com/questions/15091400/get-single-listview-selecteditem>

Archives du projet

En annexe, sur GitHub vous trouverez donc la liste des documents fournis décrit plus haut.

Lien vers le GitHub de l'équipe :

https://github.com/PhilCPNV/MAW_filefinder_QRR_PBN

Lien vers notre organisation Trello :

<https://trello.com/b/Y9JXS3ex/mawturbofinder>