|  |
| --- |
| *QXPhotoviewer* |

*Illustration*



*Nom, prénom*

*Adresse*

*NPA, Ville*

*E-mail*

*Classe*

*Date de création*

Table des matières

[1 Analyse préliminaire 4](#_Toc499799832)

[*1.1* Introduction 4](#_Toc499799833)

[*1.2* Organisation 4](#_Toc499799834)

[*1.3* Objectifs 4](#_Toc499799835)

[*1.4* Planification initiale 5](#_Toc499799836)

[2 Analyse 6](#_Toc499799837)

[*2.1* Cahier des charges détaillé 6](#_Toc499799838)

[*2.2* Définition de l’audience 7](#_Toc499799839)

[2.3 Analyse concurrentielle 7](#_Toc499799840)

[2.4 Cas d’utilisation 7](#_Toc499799841)

[1.1 Use cases 7](#_Toc499799842)

[*1.2* *Scénarios* 7](#_Toc499799843)

[1.3 Maquettes fonctionnelles 11](#_Toc499799844)

[2.5 Etude de faisabilité 13](#_Toc499799845)

[2.6 Nom du site et du domaine 14](#_Toc499799846)

[*2.7* Stratégie de test 14](#_Toc499799847)

[*2.8* Budget initial 14](#_Toc499799848)

[*2.9* Planification détaillée 14](#_Toc499799849)

[*2.10* Historique 16](#_Toc499799850)

[3 Conception 17](#_Toc499799851)

[3.1 Analyse de l’environnement 17](#_Toc499799852)

[3.2 Architecture du programme 17](#_Toc499799853)

[3.3 Définition de la charte graphique 17](#_Toc499799854)

[3.4 Maquette graphique 17](#_Toc499799855)

[3.5 Conception de la BDD 17](#_Toc499799856)

[3.6 Conception du code 17](#_Toc499799857)

[3.7 Plugins et librairies 17](#_Toc499799858)

[4 Réalisation 17](#_Toc499799859)

[4.1 Dossier de réalisation 17](#_Toc499799860)

[4.2 Description de tests effectués 17](#_Toc499799861)

[4.3 Erreurs restantes 17](#_Toc499799862)

[4.4 Dossier d’archivage 17](#_Toc499799863)

[5 Mise en service 17](#_Toc499799864)

[5.1 Rapport de mise en service 17](#_Toc499799865)

[5.2 Liste des documents fournis 17](#_Toc499799866)

[6 Conclusions 18](#_Toc499799867)

[7 Annexes 18](#_Toc499799868)

[7.1 Sources – bibliograpgie 18](#_Toc499799869)

[7.2 Journal de bord de chaque participant 18](#_Toc499799870)

[7.3 Manuel d’installation 18](#_Toc499799871)

[7.4 Manuel d’utilisation 18](#_Toc499799872)

[7.5 Archives du projets 18](#_Toc499799873)

*NOTE L’INTENTION DES UTILISATEURS DE CE CANEVAS:  
Toutes les parties en italiques sont là pour aider à comprendre ce qu’il faut mettre dans cette partie du document. Elles n’ont donc aucune raison d’être dans le document final.*

*De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n’aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l’alourdir inutilement.*

***N’oubliez pas d‘adapter les entête et pieds de page.***

# Analyse préliminaire

## Introduction

*Ce chapitre décrit brièvement le projet, le cadre dans lequel il est réalisé, les raisons de ce choix et ce qu'il peut apporter à l'élève ou à l'école. Il n'est pas nécessaire de rentrer dans les détails (ceux-ci seront abordés plus loin) mais cela doit être aussi clair et complet que possible. Ce chapitre contient également l'inventaire et la description des travaux qui auraient déjà été effectués pour ce projet.*

*Ces éléments peuvent être repris de la fiche signalétique*

## Organisation

*Elève : Nom, prénom, e-mail et téléphone*

*Responsable de projet (enseignant : Nom, prénom, e-mail et téléphone)*

*Expert 1 : Nom, prénom, e-mail et téléphone*

*Expert 2 : Nom, prénom, e-mail et téléphone*

*Intervenants externes : Client, Testeur(s)*

Mandataire : Chevillat Jerome, [Jerome.CHEVILLAT@cpnv.ch](mailto: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 dois pouvoir :

* Normaliser les noms de photos entre les différents appareilles
* Automatisée le nommage des photos
* Classer les photos par ordre chronologique
* Ajouter des informations personnelles aux photos et aux dossiers
* Visionner les photos sur différents appareils
* Faire des recherches pour trouver des photos
* Garder les informations ajoutées lors de changement d’application.
* L’accès au visionnage des photos dois pouvoir se faire depuis n’importe quel appareilles et système d’exploitation.

## *N:\COMMUN\ELEVE\INFO\SI-T1a\Trim1\MAW1.1\Groupes\Groupe QX\Planification\Image\Planification Initiale\V2.PNG*Planification initiale

### Grande dates du projet

* Début du projet 22.08.2017
* Présentation d’une partie fonctionnelle du projet 18.09.2017
* Remise de la réalisation et de la documentation Début janvier
* Fin du projet Début janvier

# Analyse

## Cahier des charges détaillé

Liste détaillées des objectifs :

### Normaliser le nom des photos et les classer

A chaque ajout de photo, l’application devras normaliser les noms des photos afin de s’y retrouver. L’application prendra la date de la prise de photo et un champ que l’utilisateur devras remplir, pour ensuite crée un nom global pour toutes les photos ajoutées pour qu’elle puissent être classées dans un ordre chronologique avec un nom clair.

### Information personnelles :

L’utilisateur doit pouvoir ajouter des informations personnelles sur les photos :

* Des lieux
* Des personnes
* Des tags
* Des favoris

Ainsi que d’autre informations sur les dossiers crée :

* Des commentaires

### Recherches :

L’utilisateur dois pouvoir trouver une ou plusieurs photos et/ou un dossier précis à l’aide d’un champ de recherche et de plusieurs filtres :

* Date
* Favoris
* Personnes
* Tags
* Format de photo

### Stockage :

Les données ajoutées par l’utilisateur doivent être stockées de tel manières à ce qu’on puisse les réutiliser dans un autre projet

### Visibilité sur d’autres appareils :

Les dossiers et photos doivent être visible sur d’autre appareil connecté au même réseau local. Les informations ajoutées par l’utilisateur ne doivent pas forcément être visible, mais le classement des fichiers doit être respectés lors du visionnage des photos.

## Définition de l’audience

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 publique.

## Analyse concurrentielle

Picasa, dont le client à parler, n’est à présent plus mis à jour. Et a été remplacé par « Google photos ». Cette solution ne convient pas car, les photos sont stockées sur le cloud. Chose que le client ne souhaite pas.

SunlitGreen Photo Manager est un logiciel qui ne répond que partiellement au désire du client. Il est impossible de mettre les photos en favoris, de les rechercher autrement que par leurs titres ainsi que de les renommer automatiquement

*Analyse de sites concurrents et/ou de sites aux fonctionnalités proches ou pouvant être transposées à ce site. Analyse également du contenu et de l’ergonomie de ces sites. En extraire les choses à faire et/ou ne pas faire*

## Cas d’utilisation

## Use cases

On utilise ce programme de gestion de photos pour :

* Classer les photos et les dossiers par ordre chronologique
* Ajouter automatiquement la date au nom des photos
* Associer des informations personnelles à des dossiers et des photos
* Rechercher des photos à l’aide de recherche par mots-clefs et des filtres activable
* Visionner les photos sur un appareil externe
* Conserver les tags des photos dans le temps

## *Scénarios*

|  |  |
| --- | --- |
| Identifiant | Scenario\_AffichePhoto |
| En tant que | Utilisateur |
| Je veux | Afficher les photos d’un dossier |
| Pour | Les voir dans l’ordre chronologique |
| Charge estimée | 4h |
| Priorité | Elevée |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| J'accède à l'application | La page d'accueil s'affiche |
| Je clic sur un dossier choisis | Les photos contenu dans le dossier sont affichées par ordre chronologique |

|  |  |
| --- | --- |
| Identifiant | Scenario\_AjoutPhoto |
| En tant que | Utilisateur |
| Je veux | Ajouter des photos |
| Pour | Les ajouter dans l’ordre chronologique |
| Charge estimée | 8h |
| Priorité | Elevée |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| J'accède à l'application | La page d'accueil s'affiche |
| Je double clic sur le dossier voulu | Les photos du dossier d’affichent |
| Je clic sur le bouton "ajouter des photos" | L'explorateur de fichier s'ouvre |
| Je sélectionne les photos voulues | Les photos sont ajoutés et s’affichent par ordre chronologique dans l'application |
| Je remplis le nom de base des photos | Les photos se renomment avec le nom de base ajouté et la date. |

|  |  |
| --- | --- |
| Identifiant | Scenario\_AficheTag |
| En tant que | Utilisateur |
| Je veux | Afficher les informations personnelles d’une photo ou d’un dossier |
| Pour | Consulter les tags |
| Charge estimée | 8h |
| Priorité | Elevée |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| J'accède à l'application | La page d'accueil s'affiche |
| Je clic sur une photo ou un dossier | Un menu contenant les informations personnelles s'affiche |

|  |  |
| --- | --- |
| Identifiant | Scenario\_ModifTag |
| En tant que | Utilisateur |
| Je veux | Modifier les informations personnelles d’une photo ou d’un dossier |
| Pour | Mettre à jour les tags |
| Charge estimée | 8h |
| Priorité | Elevée |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| J'accède à l'application | La page d'accueil s'affiche |
| Je clic sur une photo ou un dossier | Un menu contenant les informations personnelles s'affiche |
| Je remplie les tags et je clic sur le bouton "sauvegarder" situé en bas de la liste de tags | Les tags de la photo ou du dossier ont été enregistré |

|  |  |
| --- | --- |
| Identifiant | Scenario\_Triage |
| En tant que | Utilisateur |
| Je veux | Trier des photos |
| Pour | Afficher les photos voulue |
| Charge estimée | 8h |
| Priorité | Elevée |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| J'accède à l'application | La page d'accueil s'affiche |
| Je clic le bouton « filtre » | Une liste de filtre s’affiche |
| Je sélectionne les filtres souhaité | Un premier filtrage des photos est effectué |
| Je commence à taper la recherche voulue dans la barre de recherche | La barre de recherche propose des choix selon ce qui a été tapé |
| Une fois que j’ai fini de taper la recherche voulue, j’appuie sur la présélection du choix écrit | Les photos correspondant au filtres sélectionnées et à la recherche tapé s’affichent |

|  |  |
| --- | --- |
| Identifiant | Scenario\_AfficheExterne |
| En tant que | Utilisateur |
| Je veux | Voir des photos depuis un téléphone |
| Pour | Visionner des photos sur un appareil externe |
| Charge estimée | 10h |
| Priorité | élevée |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| J'accède à l'application avec une connexion au réseau de la maison | Les dossiers de photos s’affichent. |
| Je sélectionne un dossier pour voir ses photos | Les photos du dossier s’affichent par ordre chronologique. |

|  |  |
| --- | --- |
| Identifiant | Scenario\_Sauvegarde |
| En tant que | Utilisateur |
| Je veux | Changer d’application sans perdre tout le travail effectué |
| Pour | Eviter de devoir refaire toute les notations personnelles sur les photos. |
| Charge estimée | 10h |
| Priorité | Elevée |

Les informations personnelles des photos seront conservées dans les Meta données. De ce fait, n’importe quelle application capable de lire ses tags pourront récupérer et utiliser les informations de la photo.

|  |  |
| --- | --- |
| Identifiant | Scenario\_Recherche |
| En tant que | Utilisateur |
| Je veux | Faire une recherche via le champs de recherche avec un résultats correspondant |
| Pour | Trouver une photos, un dossier précis |
| Charge estimée | 6h |
| Priorité | Moyenne plus |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| J'accède à l'application | La page d'accueil s'affiche |
| Je place ma souris sur la barre de recherche et je clic | Le curseur clignote dans la barre de recherche |
| J’écris le nom de la photo recherchée | Propose des solutions au fur et à mesure de la frappe. |
| Clic ou presse « Entrée » sur le bon champ voulu ou dans la barre de recherche | Affiche le résultat de la recherche (photo ou dossier) |

|  |  |
| --- | --- |
| Identifiant | Scenario\_ErrorRecherche |
| En tant que | Utilisateur |
| Je veux | Faire une recherche via le champs de recherche sans résultats correspondant |
| Pour | Trouver une photos, un dossier précis sans trouver. |
| Charge estimée | Identique au Scénario\_001 |
| Priorité | Identique au Scénario\_001 |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| J'accède à l'application | La page d'accueil s'affiche |
| Je place ma souris sur la barre de recherche et je clic | Le curseur clignote dans la barre de recherche |
| J’écris le nom de la photo recherchée | Propose des solutions au fur et à mesure de la frappe. Puis plus aucune présélection |
| Presse « entrée » dans la barre de recherche | Affiche un message prévenant qu’il n’y a pas de résultats pour cette recherche. |

|  |  |
| --- | --- |
| Identifiant | Scenario\_Filtrage |
| En tant que | Utilisateur |
| Je veux | Faire une recherche via un ou des filtres |
| Pour | Trouver des photos, des dossiers, une personne précise, des favoris. |
| Charge estimée | 4h |
| Priorité | Moyenne |

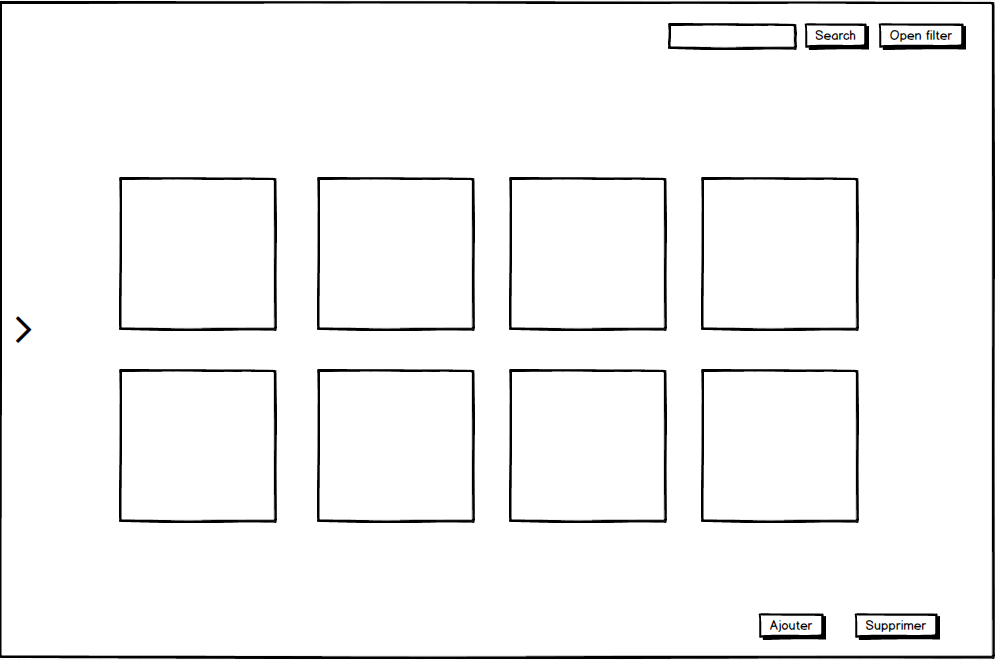
|  |  |
| --- | --- |
| **Action** | **Réaction** |
| J'accède à l'application | La page d'accueil s'affiche |
| Je clic le bouton « filtre » | Une liste de filtre s’affiche |
| Je sélectionne un premier filtre en cliquant sur le filtre « favoris » | Affiche uniquement les photos qui ont été mise en favoris. |
| Je choisis un deuxième filtre en ajoutant une plage de date. | Trie les photos favoris avec la plage de date demandée |
| Je choisis un troisième filtre en ajoutant la personne « Claude » à mes recherches | Affiche un message prévenant qu’il n’y a pas de résultats pour cette recherche. |

|  |  |
| --- | --- |
| Identifiant | Scenario\_Errorspace |
| En tant que | Utilisateur |
| Je veux | Sauvegarder les images et leurs informations |
| Pour | Garder la traces de mes modifications |
| Charge estimée | 10h |
| Priorité | Elevée |

|  |  |
| --- | --- |
| **Action** | **Réaction** |
| J'accède à l'application | La page d'accueil s'affiche |
| J’effectue une modification sur une photo en ajoutant des tags | La sauvegarde se fait automatiquement. |
| Je modifie le titre du dossier | Le dossier est sauvegardé automatiquement avec la modification du titre. |
| J’ajoute une centaine de photos d’un coup | Un message s’affiche pour prévenir l’utilisateur que le disque dur est saturé. La sauvegarde n’a donc pas eu lieu. |

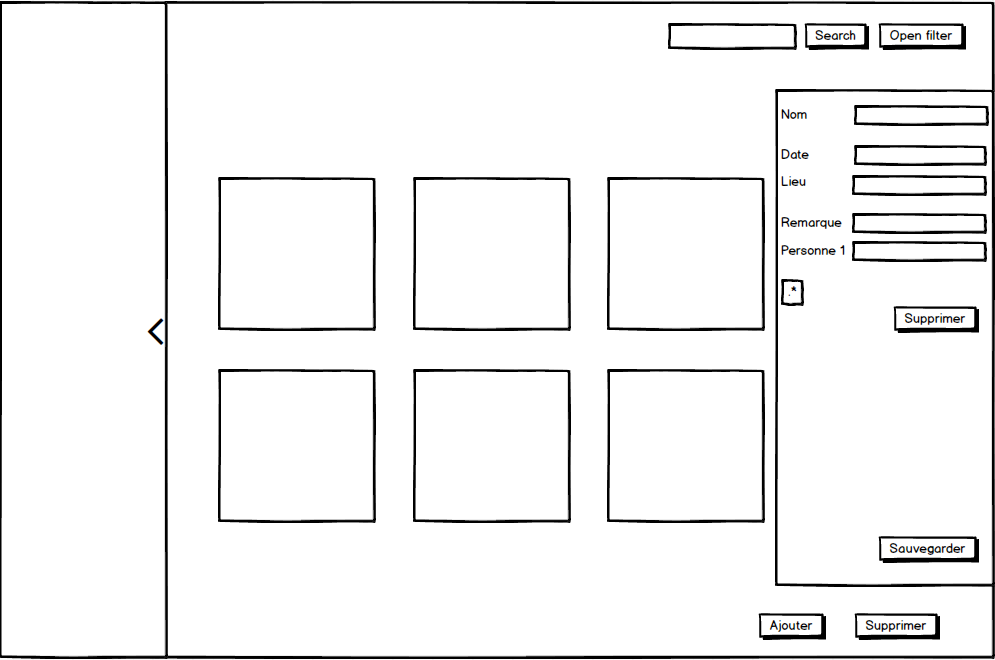
## Maquettes fonctionnelles

### Etat de base



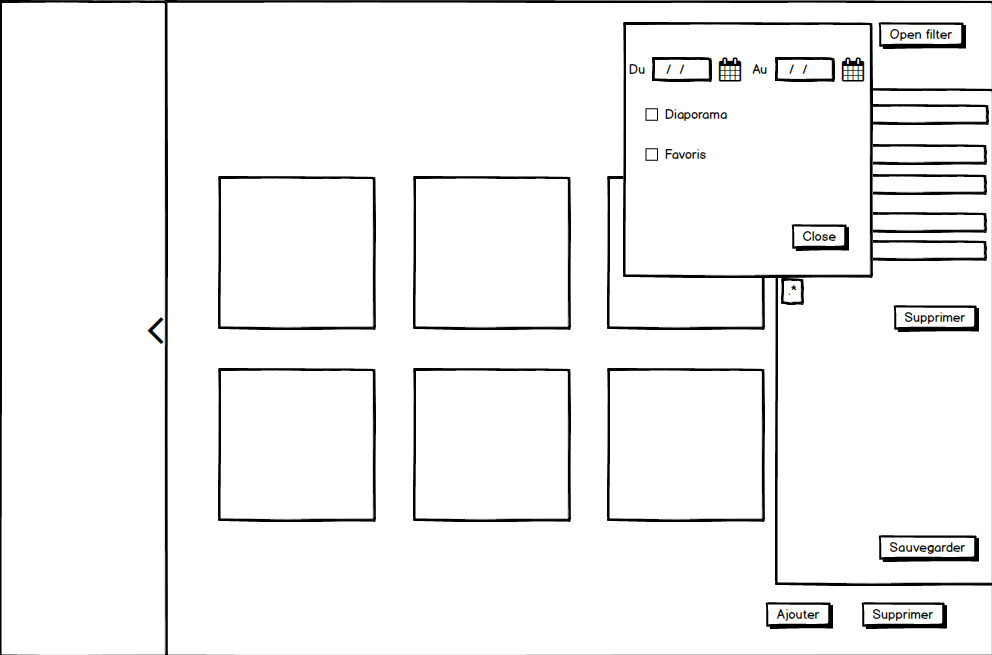
Tous les menus sont fermés et les photos du dernier dossier ouvert lors de la dernière fermeture du programme sont affichés.

### Menu principaux ouverts



A gauche, l’arborescence de fichier est ouverte, tout comme le menu d’édition des informations personnelles à droite, qui s’affiche lors d’un simple clic sur une photo.

### Menu filtre



Le menu des filtres sert à trier selon les contraintes choisies par l’utilisateur.

## Etude de faisabilité

Nous hésiter entre choisir le Framework ASP.NET ou WPF

Car ASP.NET est une solution web, et l’utilisateur souhaitait avoir une solution en local.

Nous avons choisi de faire ce projet en C# avec le Framework .Net car les deux programmeurs sont à l’aise avec le C# et l’un déjà utiliser se Framework. La phase d’apprentissage est donc réduite pour les deux développeurs.

Le Framework WPF est une Interface graphique utilisateur, utilisé pour faire des applications de bureau. Utilisant du XAML pour réaliser le design de l’application.

Au fur et à mesure de l’avancée dans le projet, nous avons rencontré aucun souci majeur qui nous a fait changer de solution. Nous n’avons pas encore trouvé de solutions permettant de crée de l’espace pour ajouter des métadonnées inexistantes sur une photo.

## Nom de l’application

Nous avons pensé à plusieurs noms pour cette application :

**QXPhotos**

Fait référence aux initiales des deux développeurs. Ce nom est libre mais on ne retrouve pas le coté gestion dans le titre.

**PhotoViewer**

Ce nom est simple, trop simple. Il décrit mieux l’application mais pas le coté de gestion.

**QXPhotosViewer**

Est donc le nom que nous avons choisi pour notre application. Un mélange des deux autres idées qui présente bien le projet en ajoutant une touche personnelle au nom.

## 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 inutilisé dans la salle de classe

Données de test utilisés :

Le client nous a fournis un échantillon de photos. Nous les utiliserons dans les tests de notre application.

Participant aux tests :

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

*Décrire la stratégie de test:*

* *Liste des tests pour la validation par le client*
* *Les moyens à mettre en œuvre.*
* *Données de test à prévoir (données réelles fournies par le client ?).*
* *Testeurs.*

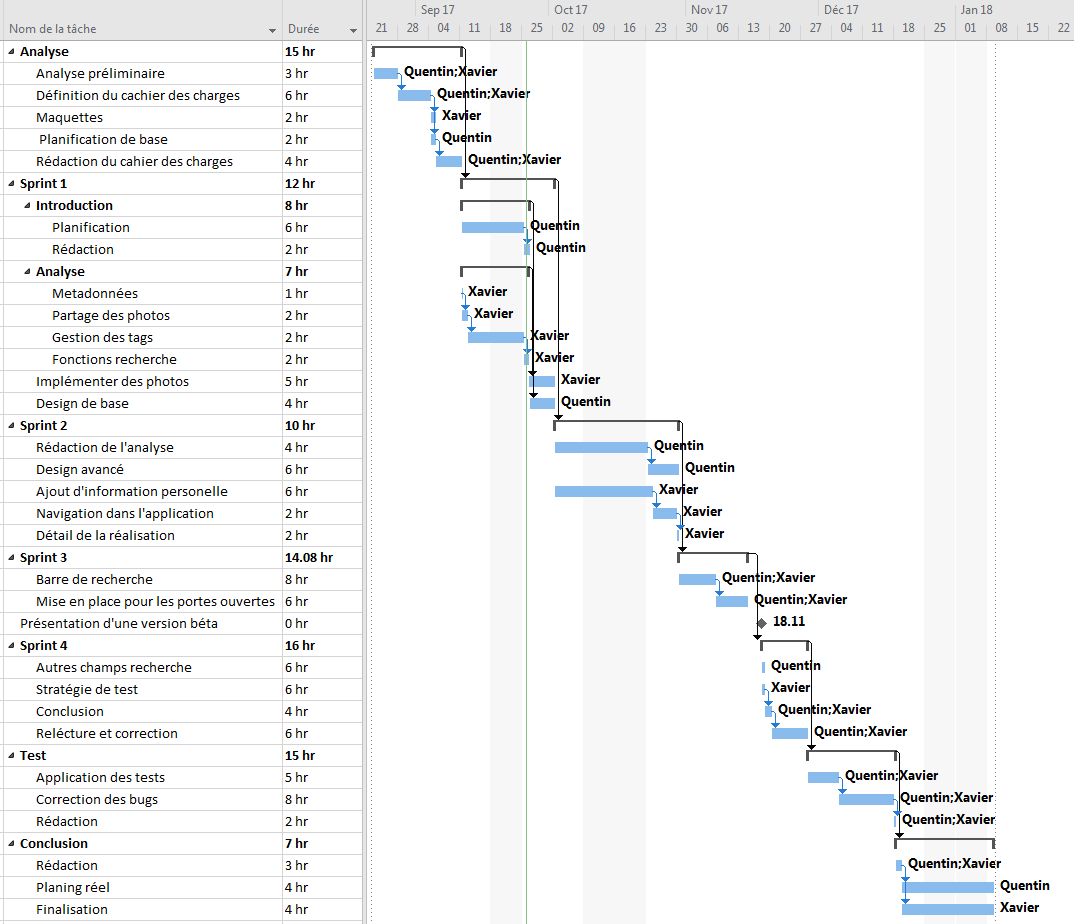
|  |  |  |  |
| --- | --- | --- | --- |
|  | Unitaire | Intégration | Système |
| Fonctionnel | Effectué en cours de réalisation pour chaque développeur | Aucun | L’application est lancée sur un post lambda  Les cas d’utilisateurs seront effectué un à un |
| Performance | Aucun | Aucun | L’affichage des images (100) |
| Robustesse | Effectué en cours de réalisation pour chaque développeur | Aucun |  |

## Budget initial

Les applications nécessaires pour ce projet sont fournies par le CPNV. Il n’y a donc aucun budget prévu à cet effet.

Pour le salaire, nous prenons une estimation de 30CHF / h pour un étudiant en technicien ES. Avec une estimation du temps de travail de 50h, la réalisation du projet couterait 3000 CHF.

## Planification détaillée



Voici les grandes dates du projet :

|  |  |  |
| --- | --- | --- |
| Analyse | 22.08.2017 | 11.09.2017 |
| Rendu du CDC | 15.09.2017 | |
| Sprint 1 | 03.05.2017 | 09.05.2017 |
| Sprint 2 | 12.05.2017 | |
| Sprint 3 | 09.05.2017 | 30.05.2017 |
| Porte ouverte | 02.06.2017 | |
| Sprint 4 | 12.06.2017 | 13.06.2017 |
| Tests | 28.11.2017 | 18.12.2017 |
| Conclusion | 19.12.2017 | 19.12.2017 |

Le projet devra être rendu le 02.06.2017 à 8h50 ou avant. Avec une version papier et électronique pour chaque expert et responsable du projet.

La documentation sera écrite en même temps que les autres tâches. Ainsi que la planification et l’historique qui seront tenu à jour au fil du projet.

Code couleur :

De base : la tâche a été résolue dans les temps impartie.

Vert : la tâche a été terminée plus rapidement que prévue.

Rouge : la tâche a été plus longue que prévue.

Historique

28.04.2017

Modification du canevas, ajout des cas d’utilisation. Changement de tâches et des heures attribué dans la partie Analyse

30.11.2017

Date de fin annoncée au 19 décembre 2017. Plutôt que prévue, il va falloir trouver des solutions.

*Révision de la planification initiale (Gantt) du projet :*

*Planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases (jalons).*

*Listes des tâches, durées et interdépendances.*

*Il s’agit ici de détailler la planification initiale du projet.   
Elle sera ensuite affinée régulièrement (ajout/suppressions de tâches, durées…). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliqués dans l’historique.*

## Historique

*Pour le cahier des charges : lister toutes les modifications demandées par le client, ainsi que celles décidées pour d'autres raisons*

# Conception

## Analyse de l’environnement

### Liste du matériel HW :

Le CPNV met à notre disposition un poste informatique complet ainsi qu’une connexion internet.

*Déterminer le matériel HW.*

### Liste des logiciels utilisés :

* Visual studio 2017

Sera notre logiciel de développement

* Suite office 2016 (Word, Excel, Project, Outlook)

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

*compris IDE, frameworks, librairies, etc.*

*Détailler tout ce qui va permettre de développer et de maintenir le projet.*

## Architecture du programme

Principe de l’architecture

Qui parle avec qui ?

(Il faut faire comme si on laissait le code à un autre programmeur. Il faut lui fournir des informations.)

Image detail = model

MainWindow.xaml = vue

MainWindow.xaml.cs = Codebehind utilise

UriToBitmmapConverter.cs = class pour convertir les images

Shéma

Codebehind MainWindow.xaml.cs

* Selectionne un dossier et crée une liste d’image
  + Récupère les metadata d’une image
  + Ajouter des metadata aux images

GetTags() = récupère les metadata de l’images

AddTag() = ajouter des metadata aux images

Lors d’un clic sur le bouton « AddTags () » La methode « TagButton\_Click() » est appellée. Celle-ci appelle la méthode « AddTags() » en lui fournissant les metadata voulue par l’utilisateur. Une fois ceci fait, elle appelle la methode « GetTags() » Pour afficher les nouveaux tags.

Lors d’un clic sur une image, on appelle la methode « GetTags () » et affiche les résultats

Lors d’une recherche, reprend la liste d’images affichée puis tri avec les critères de l’utilisateur, avant de retourner la nouvelle liste.

Sélectionne le dossier voulu, crée une liste d’image et renomme les photos avec la date de prise de vue récupéré avec la méthode « GetTags() » (ou de modification si l’autre date n’existe pas) et le nom du dossier

Ajoute de l’espace dans les images pour les metadata

## Définition de la charte graphique

Après avoir visualiser plusieurs applications du genre, Nous avons opté pour un visuel simple aux teinte épurée, vu que les images prendront la majeure partie du visuel de l’application.

Les couleurs du site seront du bleu clair, du blanc, du noir, et du gris. Le bleu ne sera que peu présent pour équilibrer. Le gris servira à mettre en évidence certaine partie de l’application.

La police d’écriture choisis est « Lato ». Qui est simple et lisible.

## Maquette graphique

(Reprise des captures d’écran de l’application)

## Conception du code

shéma

## Plugins et librairies

Kepuik

# Réalisation

## Dossier de réalisation

Fichier de l’application

Image detail = model

MainWindow.xaml = vue

MainWindow.xaml.cs = controleur

UriToBitmmapConverter.cs = class pour convertir les images

## Description de tests effectués

Réponse à la stratégie de test

## Erreurs restantes

Bha.. tout ?

Le seul truc qui marche pas : l’ajout d’espace pour les tags.

Recherche quand les tags sont vides, ça bug

La gestion des erreurs

## Dossier d’archivage

Le projet a été créé sur GitHub afin que les deux participants puissent retrouver le projet et effectuer des modifications dessus sans se perdre. Tout en gardant un historique des modifications

Voici l’arborescence de dossier que nous avons créé pour ce projet :

Liste des dossiers du projet (word, et tout ça)

───Canevas Liste de canevas utiles pour le projets

├───CDC

│ ├───CDC Cahier des charges et ses versions

│ │ └───Ancien Ancienne version

│ ├───Maquette Fichier de maquettes

│ │ └───img Capture d’écran des maquettes

│ ├───Retour Retour prof du cahier des charges

│ └───Uses-cases Liste des Uses-cases du projet

├───Documentation Liste des versions de la documentation

├───Planification Liste des versions de la planification

│ └───Image Capture d’écran des versions de la planification

├───Présentation

│ └───Portes-ouvertes Informations sur la présentation des portes ouvertes

└───PV Liste des PV du projet

## Problèmes rencontrés

L’accès aux metadata des images était bloqué, en conséquence, on ne pouvait pas les modifiers.

Analyse :

On à trouver après des recherches que c’est la liste des images affiché qui bloquait l’accès aux metadata.

Solution /choix

La liste d’image est détruite avant la modification des metadata. Une fois ceci effectuée, elle est reconstruite.

Poids des images : lors du chargement de la liste des images, la RAM était saturé par cette liste.

Analyse :

Les miniatures gardait le même poids que l’image originale.

Choix /soluce

Création d’une classe pour réduire la taille des miniatures.

Lors d’ajout d’une metadata qui n’a jamais été assignée, il est impossible de l’assigner.

Analyse :

L’image ne contient pas la place pour les metadata. Il faut donc crée cet espace.

Soluce / choix

Création d’une classe pour ajouter de la place pour les metadata dans l’image.

Lors d’une recherche, si un des metadata est vide, une erreur survient.

Analyse :

La recherche s’attend à un résultat non nul.

Soluce :

Pas de solution trouvée actuellement.

# Mise en service

## Rapport de mise en service

Comment on instale xd

.exe

tada

## Liste des documents fournis

Manuel d’installation

Page de présentation format A4.

# Conclusions

## Objectifs

Voici la liste des objectifs atteints :

* Normaliser les noms de photos avec une date.
* Retirer le nommage « à la main » des photos importées.
* Classer les photos par ordre chronologique
* Ajouter des informations personnelles sur les photos tel que :
  + Des lieux (dans les commentaires
  + Des personnes ( l’auteur de la photo peut être utilisé com tel)
  + Des tags (fait)
  + Des favoris (oui mais on en fait rien)
* Être visible depuis plusieurs appareils comme :
  + Un téléphone
  + Une tablette
  + Un ordinateur
* Avoir des sauvegardes de données réutilisable en cas de changement de logiciel.
* Faires des recherches grâce à :
  + Une barre de recherche en auto complétion

Voici la liste des objectifs non atteints :

* Si la metadata n’existe pas, on ne peut pas la rajouter.
* Ajouter des informations personnelles sur les dossiers tel que :
  + Des commentaires
  + Des filtres (favoris, date, format de photo, personne, tags)

## Points positifs et négatifs

Xavier positif

Ça ma permit de voir de nouvelles technologie de traitement d’images

Xavier négatif

Trop peu de temps pour terminer entièrement le projet.

## Difficultés particulières

Xavier dur

La gestion des metadata a pris bien plus de temps que prévu initialement

## Suite et améliorations possible

*Développez en tous cas les points suivants:*

Amélioration :  
Terminer le projet en bonne et due forme

Amélioration des performances globale

Suite possible

Faire une application pour mobile

Ajout de fonctionnalité de traitement d’image supplémentaire par exemple des filtres graphique (sépia, noir et blanc, etc…)

* *Suites possibles pour le projet (évolutions & améliorations)*

# Annexes

## Sources – bibliographie

GitHub, openclassroom, stackowerflow, code project,

Géstion de tags:HTTP://blog.andreweichacker.com/2009/02/reading-and-writing-tags-for-photos-in-wpf

## Journal de bord de chaque participant

En annexe

## Manuel d’installation

annexe

## Archives du projet

Annexe

Clef usb, dropbox, etc.

GIT