



Rapport d'analyse

Editeur graphique de fichiers XML de données MicMacV2

Réalisé par :
Loïs Bilheran, Maëlle Klein, Zijian Liu, Laurie Behloul

Référents :
Jean Francois Hangouet et Florent Geniet

Commandité par :
Christophe Meynard



11 Mars 2024

Table des matières

1	Introduction	2
2	Contexte du Projet	2
2.1	Les différents parties	2
2.2	Enjeux	2
2.3	Impact financier et social	3
3	Reformulation du Besoin	3
3.1	Objectifs de l'étude	3
3.2	Contraintes	3
3.3	Acteurs	5
4	Analyse Fonctionnelle	6
4.1	Diagramme de classe	6
4.2	Diagramme d'activité	7
4.3	Diagramme de séquence	8
5	Étude Technique	9
5.1	Choix du Langage de Programmation	9
5.2	Choix des Outils et des Bibliothèques	9
5.3	Données Utilisées	9
5.4	Gestion des Contraintes	10
6	Réalisation et suivis du projet	11
6.1	Risques	11
6.2	Planning prévisionnel	11
7	Conclusion	12

1 Introduction

Dans le cadre de notre Projet de Développement Informatique, nous avons entrepris la réalisation d'un éditeur graphique de fichiers XML de données MicMacV2. L'objectif principal de ce projet est de développer un outil permettant de manipuler et d'éditer des données au format XML, spécifiquement conçues pour être utilisées avec le logiciel de photogrammétrie MicMacV2. Cette initiative découle d'un besoin réel exprimé par notre commanditaire, soucieux de disposer d'un moyen efficace pour visualiser et modifier les données MicMacV2 de manière intuitive et rapide.

2 Contexte du Projet

2.1 Les différents parties

Plusieurs parties sont impliqués dans notre projet, chacun contribuant de manière significative à la réussite de celui-ci. Dans un premier temps, nous avons notre commanditaire Christophe Meynard. Son rôle consiste à définir les besoins et les attentes du projet, à fournir les ressources nécessaires et à superviser son avancement. L'ENSG fait aussi partie des commanditaires du projet. Des livrables différents sont attendus pour l'évaluation du projet. Pour nous épauler tout au long de ce projet, nous avons nos encadrants : Jean Francois Hangouet et Florent Geniet. Il nous guide pour chaque étape du processus de développement, mais également Thierry Saffroy et Véronique Pereira pour nous aider sur la partie gestion de projet. Enfin, notre équipe de développement est composée de Loïs Bilheran qui est à la fois chef de projet et développeur logiciel, Zijian Liu qui est également développeur logiciel, Maëlle Klein qui est en charge du développement de l'interface graphique et Laurie Behloul qui documente le projet à chaque séance et qui développe également l'interface graphique. Cette collaboration entre les différents acteurs, nous permet de mener à bien notre projet. Cela vise à répondre aux besoins du commanditaire, à respecter les contraintes fixées et à fournir une solution fonctionnelle pour la communauté géomatique.

2.2 Enjeux

Les enjeux de ce projet sont multiples. Tout d'abord, l'ajout d'une interface graphique à MicMacV2 permettra de simplifier considérablement la manipulation des fichiers XML de données, ce qui réduira les risques d'erreurs de syntaxe et facilitera la création et la modification de ces fichiers. De plus, en facilitant l'utilisation de MicMacV2, notre éditeur graphique pourra contribuer à l'amélioration de cet outil open source, en augmentant sa popularité et en élargissant sa base d'utilisateurs. Il deviendra une alternative solide aux logiciels payants de photogrammétrie, tel que MétaShape. Cette démarche renforcera ainsi le positionnement de l'IGN en tant que diffuseur open source de données géographiques et de créateur de géo-commun, et pourra offrir aux spécialistes la possibilité de réaliser leurs travaux de manière gratuite et efficace. En outre, ce projet s'inscrit dans une démarche d'innovation et de modernisation des outils de traitement d'images et de données géospatiales.

2.3 Impact financier et social

L’impact financier révèle que le coût direct du projet s’élève à zéro euro, puisque aucune dépense spécifique n’est engagée pour sa mise en œuvre. Cette situation découle en partie du caractère scolaire qui anime ce projet, ainsi que de l’utilisation exclusive de matériels et de logiciels open source, ce qui réduit considérablement les coûts liés à l’acquisition de licences et d’équipements. Le coût indirect du projet est estimé à 210 euros, principalement attribuable aux frais de transport et de repas engagés par les membres de notre équipe. Ces coûts sont à notre charge et ne représentent donc pas une dépense directe pour le projet. Sur le plan social, ce projet revêt également une importance particulière. En offrant une solution open source et intuitive pour la manipulation des données géospatiales, il contribue à démocratiser l’accès aux outils de photogrammétrie. De plus, ce projet nous permet de renforcer notre réseau professionnel et de consolider nos liens au sein de notre équipe. Cela nous permet d’explorer en interne les notions de collaboration et de soutien nous faisant grandir ensemble sur un plan autant individuel que collectif.

3 Reformulation du Besoin

3.1 Objectifs de l’étude

Notre projet vise à concevoir un éditeur graphique de fichiers XML de données MicMacV2, s’intégrant à l’environnement du logiciel de photogrammétrie. Les objectifs de ce projet sont les suivants :

- Création d’un formulaire de création de fichier XML de données : l’outil développé permettra aux utilisateurs de générer facilement de nouveaux fichiers XML de données en fournissant un fichier XML de spécifications.
- Fonction de modification des fichiers XML existants : en cas de fourniture d’un fichier XML de données existant, le formulaire sera pré-rempli pour permettre une modification simple.
- Livrables pour le commanditaire : Le projet livrera le code de l’outil, la documentation de l’outil, la documentation du code, le rapport d’analyse final, un README présentant le projet et faisant office de tutoriel d’installation et un tutoriel d’utilisation.
- Présentation lors de l’événement GéoDev² : Le projet sera présenté lors de cet événement afin de partager les résultats obtenus avec la communauté et de recueillir des retours d’expérience.

3.2 Contraintes

Pour atteindre ces objectifs, le projet est soumis à diverses contraintes :

- Échéance de livraison : Le projet doit être prêt avec tous les livrables requis pour le commanditaire pour le 15 mai 2024 au soir.
- Utilisation de Python ou C++
- Modularité et compatibilité : L’outil doit être conçu de manière modulaire et flexible pour assurer sa compatibilité avec différents formats de données, notamment XML et JSON.

- Interface utilisateur simple d'utilisation : L'interface graphique de l'outil doit être intuitive et efficace.
- Livrables pour l'école : Le projet reste un projet scolaire, des livrables spécifiques sont donc attendus. Le projet livrera le code de l'outil, la documentation de l'outil, la documentation du code, le rapport d'analyse final, un README présentant le projet et faisant office de tutoriel d'installation, un tutoriel d'utilisation et une vidéo de démonstration de l'outil pour le 15 mai au soir.
- CoTech : Le rapport d'analyse est à envoyer le 11 Mars au soir, et le CoTech aura lieu le 18 Mars. Le rapport d'analyse finale à rendre pour le 12 Mai au soir.
- CoPil : Le CoPil de lancement aura lieu le 28 Février et le CoPil de clôture le 13 Mai. Les livrables attendus pour ces séances seront les présentations. Des séances de coaching sont prévues le 20 Mars, 03 Avril et 24 Avril. Les rendus de gestion de projet sont à fournir pour le 13 Mai et comprennent les différents outils de gestion de projets actualisés, les présentations des CoPil et le compte rendus de chaque point/réunion avec le commanditaire.

3.3 Acteurs

La réalisation de cet éditeur graphique de fichiers XML de données MicMacV2 vise principalement les professionnels et les spécialistes du domaine de la géomatique et de la photogrammétrie. Le public cible comprend donc essentiellement les photogrammètres ainsi que toute personne travaillant dans les domaines de la modélisation 3D. Ces utilisateurs ont besoin d'une solution efficace pour visualiser, manipuler et éditer les fichiers XML de données MicMacV2 dans le cadre de leurs activités professionnelles. Par exemple, les paramètres internes d'un appareil photo ne sont pas toujours renseignés dans les fichiers XML de données et il faut les ajouter.

Ces différentes personnes ont une utilisation commune du logiciel, qui a été décrite dans un diagramme d'utilisation. Pour l'utilisateur, son besoin est de créer ou modifier un XML de données en fournissant le XML de spécification associé, et le XML de données à modifier si tel est le cas.

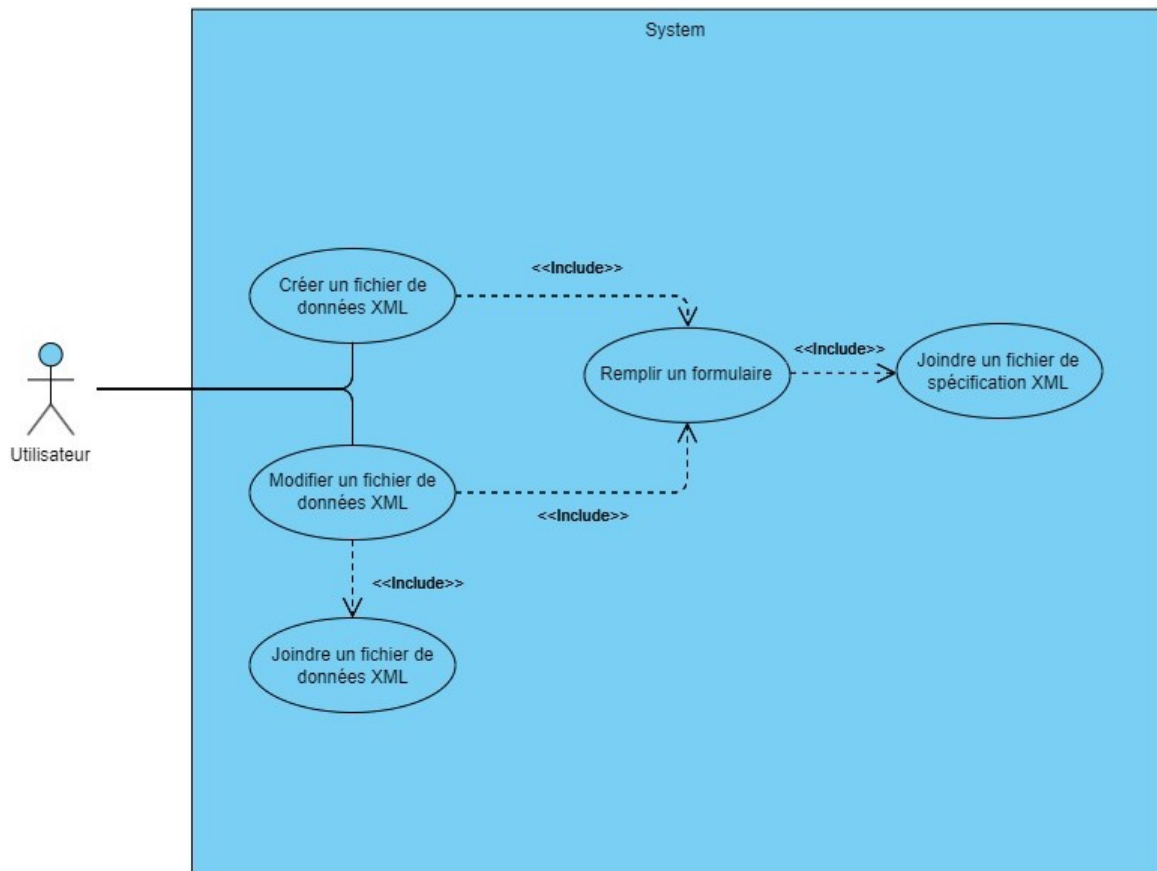


Figure 1: Diagramme d'utilisation

4 Analyse Fonctionnelle

4.1 Diagramme de classe

Le diagramme de classe offre une vue statique de la structure interne du système, mettant en évidence les différentes entités, leurs attributs et leurs relations. Le diagramme suivant représente l'assemblage de nos premières idées de la structure interne de notre future proposition. On a commencé par créer une classe Data abstraite duquel seront héritées les classes XML et Json (le nom de ces deux classes représentent juste le format du fichier d'origine). Cela permettra aux futurs développeurs d'utiliser des formats de données différents. Ensuite, la classe Strategy permettra de différencier les stratégies pour les données xml et json. Nous avons également construit une classe GUI qui représente l'interface utilisateur contenant des champs issus de la classe Champ. Enfin, ce diagramme vise à respecter les principes SOLID au mieux et n'est qu'une première version de test pour le moment. Nous aimerions préciser que les constructeurs ne sont pas dans ce diagramme car nous voulons mettre que les éléments importants.

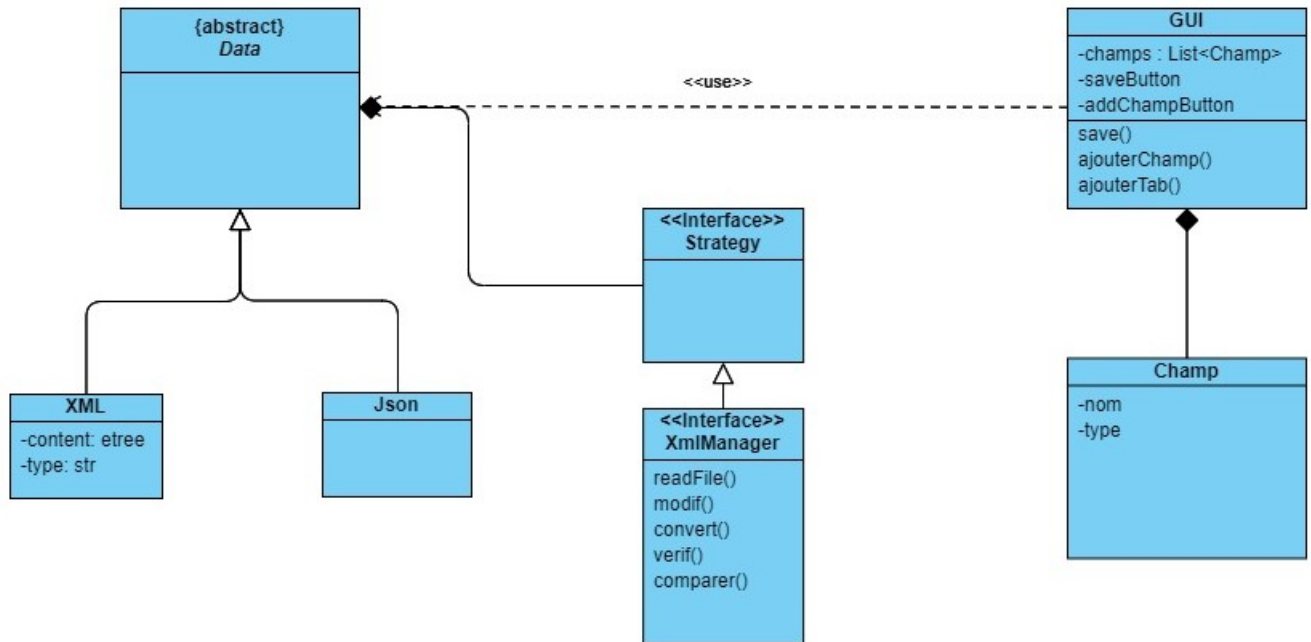


Figure 2: Diagramme de classe

4.2 Diagramme d'activité

Le diagramme d'activité suivant illustre le processus de gestion d'un fichier XML. Il commence par vérifier la conformité du fichier XML de spécification. Il faudra également vérifier le fichier XML de données s'il est inséré. Une fois les fichiers de spécifications et de données (si présent) validés, le système examine les potentielles modifications des champs; s'ils sont modifiés et ont été changés, le système les modifie. Par la suite, les champs modifiés sont enregistrés dans un nouveau fichier XML. Ce cycle itératif assure que seuls les fichiers XML corrects et actualisés progressent vers la fin du processus.

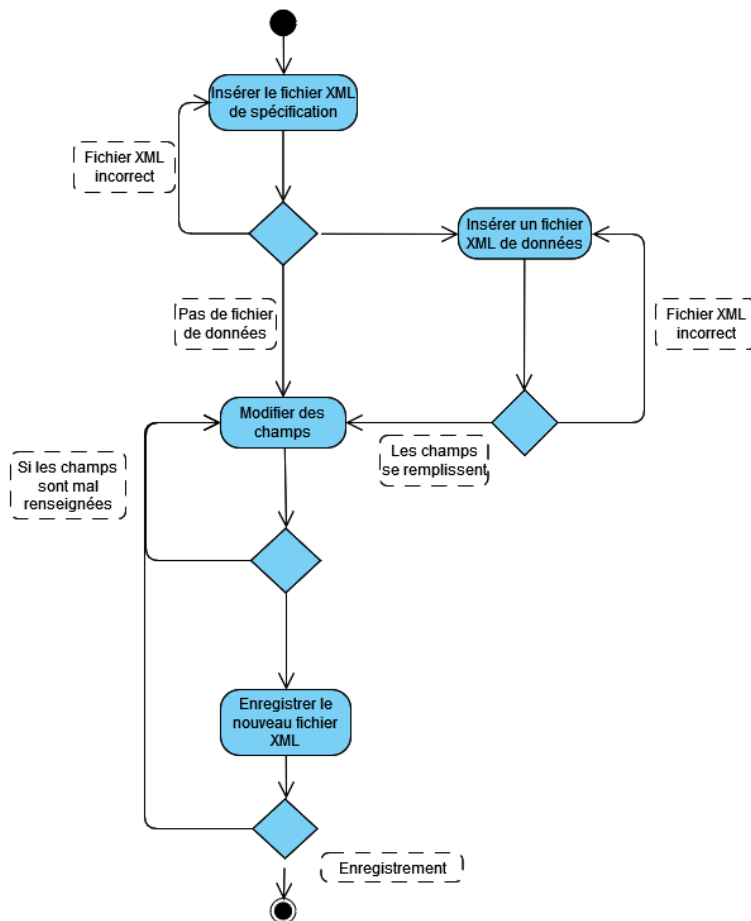


Figure 3: Diagramme d'activité

4.3 Diagramme de séquence

Le diagramme de séquence suivant représente l'interaction entre les différentes classes et objets dans une séquence temporelle. Tout d'abord, l'utilisateur va joindre un fichier XML de spécification via le terminal. Ensuite, le programme principal (Main) va créer un objet de spécification en interne et vérifier sa conformité par rapport aux spécifications. Si c'est le cas, les champs seront créés en fonction des données spécifiées. En parallèle, l'utilisateur peut également joindre un fichier XML de données via le terminal au même moment où il joint le fichier XML de spécification. Dans ce cas, le programme effectue une validation des données et actualise les champs. Ensuite, le Main génère l'interface utilisateur graphique (GUI) contenant et affichant les champs. Lorsque les champs sont modifiés, Le GUI se rafraîchit automatiquement pour mettre à jour les modifications. Enfin, lorsque l'utilisateur clique sur le bouton de sauvegarde, le programme génère un nouveau fichier XML de données (qui peut écraser l'ancien) en fonction des données modifiées dans l'interface. Si la fenêtre est fermée, le programme se termine.

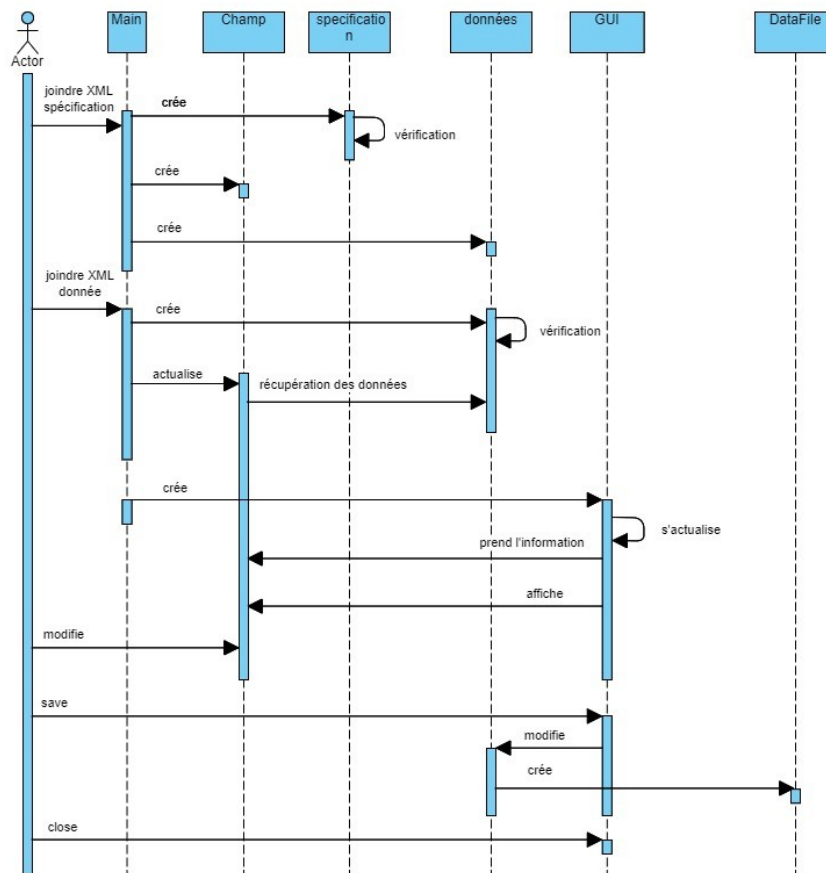


Figure 4: Diagramme de séquence

5 Étude Technique

5.1 Choix du Langage de Programmation

Nous avons choisi Python comme langage de programmation principal pour plusieurs raisons. Nous avons le choix entre Python et C++, puisqu'il s'agissait d'une contrainte imposée par le commanditaire d'utiliser l'un de ses deux langages. Comme notre équipe a eu l'occasion de se familiariser avec le langage Python et n'a en revanche aucune base en C++, choisir une approche en C++ représenterait une perte de temps considérable. Notre choix s'est donc porté sur Python.

5.2 Choix des Outils et des Bibliothèques

Dans le cadre de notre projet, nous avons opté pour des outils et des bibliothèques très exploitées en python:

- **Qt** : Cette API permet de développer des interfaces graphiques.
- **lxml** : Cette bibliothèque nous permet de parcourir, de modifier et de générer des documents XML de manière efficace. Elle permet également de transformer des XML en arbre de donnée.

5.3 Données Utilisées

Dans le cadre de notre projet, nous travaillons avec des fichiers XML de données et de spécifications offrant une approche arborescente et structurée pour représenter une multitude de données. Cette approche permet une organisation hiérarchique des données facilitant leur manipulation et leur interprétation. Les fichiers XML de données contiennent des informations importantes pour la photogrammétrie, telles que les paramètres de calibrage et les coordonnées de point de contrôle. En parallèle, les fichiers XML de spécifications détaillent la structure et le format de ces données. Les fichiers XML de micmac v1 et v2 ne sont pas compatibles. En plus des fichiers XML, nous envisageons également la possibilité de travailler avec des données au format JSON. Bien que notre projet se concentre principalement sur les fichiers XML, la prise en charge de JSON pourrait élargir la portée de notre éditeur, permettant aux utilisateurs de travailler avec différents formats de données. Il nous faudra être flexible dans la conception de notre interface utilisateur et adapter nos algorithmes.

5.4 Gestion des Contraintes

Lors de la conception de notre solution, nous devons faire face aux impératifs de temps, aux ressources disponibles, et aux exigences fonctionnelles. Pour y faire face, nous avons mis en place plusieurs stratégies pratiques. Par exemple, nous avons élaboré un tableau des actions à entreprendre, sur le site Trello, avec des priorités définies et des membres désignés pour chaque tâche. De plus, nous avons établi une matrice de risques pour anticiper et atténuer les éventuelles difficultés. Enfin, nous avons élaboré un diagramme de Gantt pour planifier et suivre la progression de notre projet.

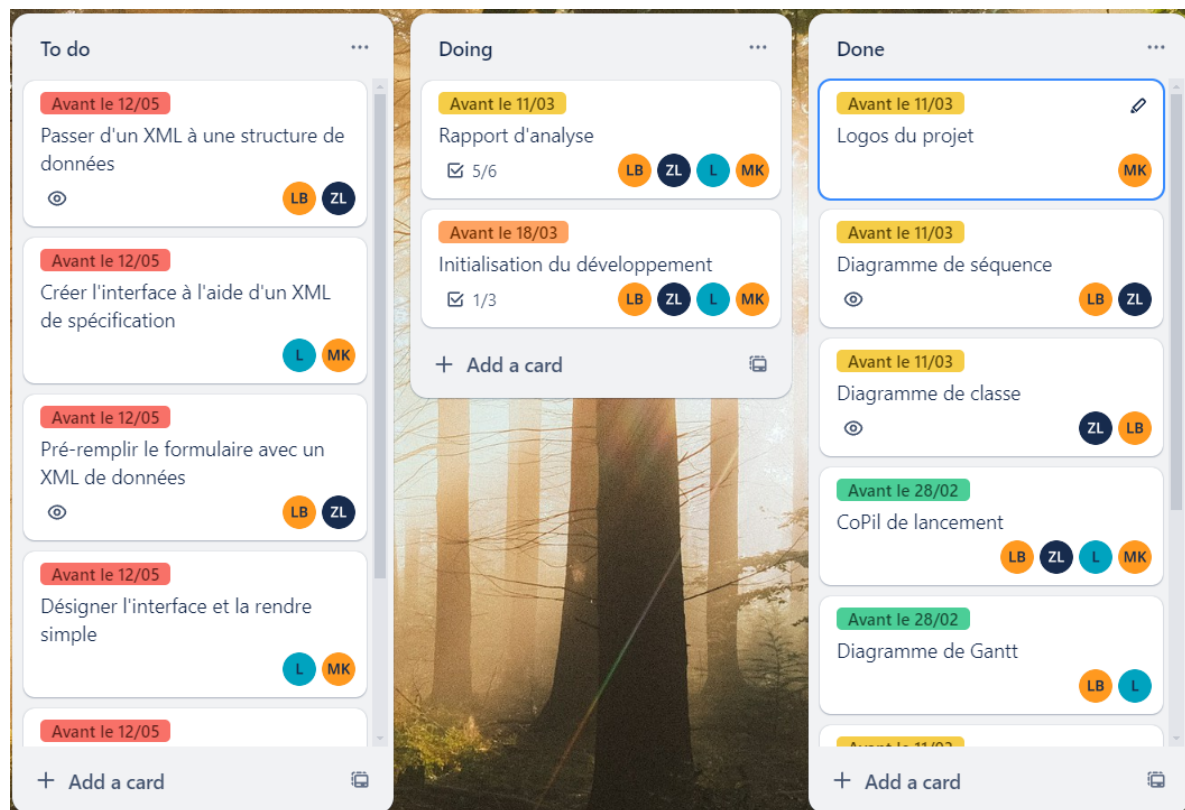


Figure 5: Tableau des actions sur Trello

6 Réalisation et suivis du projet

6.1 Risques

La matrice de risque est un outil permettant d'identifier, évaluer et gérer les risques associés au projet. Elle permet de visualiser les différents risques qui pourraient survenir tout au long du projet, leur probabilité de réapparition et leur impact sur les objectifs du projet. Dans notre analyse des risques, nous avons relevé plusieurs scénarios susceptibles d'affecter le déroulement et la réussite de notre projet:

Nature du risque	Probabilité	Conséquence	Solution corrective	Evolution
Pertes des données/travaux réalisés	Moyenne	Perte de temps et efforts importants pour récupérer les données.	Sauvegarde multiple sur plusieurs support (ordinateur, clé USB...), dépôt Git.	
Bugs logiciels	Haute	Dysfonctionnement de l'outil et impact sur la productivité.	Mise à jour régulière du code et réalisation de tests.	
Interruption électrique sur notre lieu de travail	Faible	Perte de productivité et indisponibilité de l'outil.	Ordinateur portable, télétravail.	
Manque de communication interne	Faible	Malentendus, erreurs de coordination, doublons de travaux	Groupe messenger de communication, dépôt GitHub, tableaux des taches sur Trello	+
Manque de communication externe	Faible	Direction prise ne correspondant pas au besoin du commanditaire	Visio à chaque fin de séance ou toutes les 2 semaines, discussions fréquentes par mail	+

Figure 6: Matrice de risque

6.2 Planning prévisionnel

Nous avons réalisé un diagramme de Gantt afin de planifier, de suivre et de visualiser les différentes tâches et activités nécessaires à la réalisation de notre projet. Ce diagramme présente les tâches du projet sous forme de barres horizontales sur un axe de temps, ce qui permet de visualiser la chronologie des activités et les dépendances entre elles.



Figure 7: Diagramme de Gantt

En combinant la matrice de risque et le diagramme de Gantt, nous essayons de faire en sorte d'anticiper au mieux les obstacles que nous pourrions rencontrer.

Pour faciliter la communication dans l'équipe, un groupe Messenger a été créé. Nous échangeons également régulièrement par mail avec le commanditaire et avons créé un Git pour notre projet.

Le logo a été conçu en s'inspirant de sa représentation visuelle initiale présente sur MicMac. Pour maintenir une cohérence visuelle, nous avons repris les couleurs du logo précédent. L'objectif de notre projet étant de simplifier la création et la modification de fichier est retranscrit dans notre logo nommé "XML Editor". Parmi les trois propositions de design, seule celle présentée en page de garde a été retenue.



Figure 8: Logos non retenus

7 Conclusion

En définitive, la phase d'analyse nous a permis de mettre en lumière les enjeux cruciaux et les objectifs clés de notre démarche. En nous appuyant sur une analyse approfondie du contexte, des besoins des utilisateurs et des contraintes techniques, nous avons établi une vision claire de notre projet.