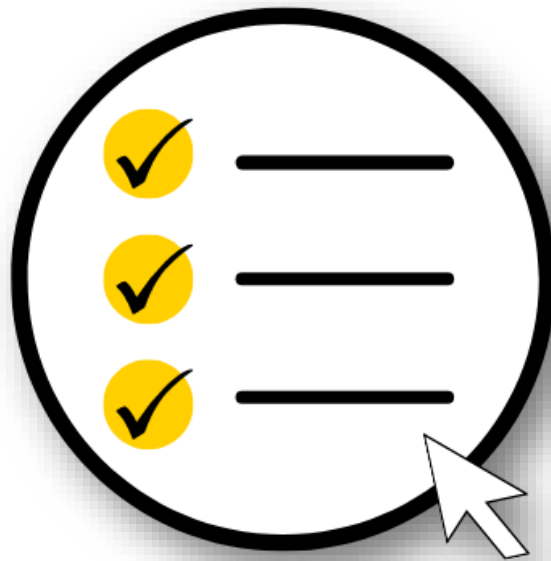


TPI : Advanced To Do List



Théo Ghaemmaghami
ETML

Table des matières

Remerciements	3
Introduction	4
1 Planification initiale	5
2 Analyse	15
2.1 Planification détaillée	15
2.2 Méthodologie	15
2.2.1 Méthode de gestion de projet	15
2.2.2 Sauvegarde	17
2.3 Stratégie de test	17
2.4 Conception	17
2.4.1 Architecture MVC	20
2.4.2 Modélisation de la base de données	17
2.4.3 UwAmp	19
2.4.4 Maquettes	19
3 Réalisation	22
3.1 Compte utilisateur	22
3.1.1 Hachage	22
3.1.2 Création de compte et login	22
4 Tests	24
5 Conclusion	25
6 Annexes	26

Remerciements

Avant d'entamer ce rapport, j'aimerais remercier les gens qui ont été impliqué de loin ou de près dans ce projet.

Tout d'abord, je tiens à remercier mon chef de projet, Monsieur Chenaux, d'avoir accepté d'être mon chef de projet pour ce TPI.

Je remercie également et particulièrement les deux experts, Monsieur Berney et Monsieur Wenger, d'avoir accordé de leur temps, d'avoir contribué à l'amélioration de mon projet grâce à leurs retours

Introduction

Dans le cadre de ma formation d'informaticien et dans le but d'obtenir mon Certificat Fédéral de Capacité d'informaticien, un Travail Personnel Individuel est à réaliser pour conclure mon apprentissage. L'objectif étant de mettre en pratique ce que j'ai appris durant ces deux années passées à l'Ecole des Métiers – Ecole Technique de Lausanne.

Ayant la possibilité de choisir dans quel domaine nous souhaitons faire ce projet, j'ai décidé de me lancer dans un projet de « Développement Desktop » car c'est dans cela que je souhaite me spécialiser par la suite. En effet, aspirant à poursuivre mes études en Informatique Logicielle à l'HEIG-VD, j'ai vu dans ce projet une opportunité de me replonger dans la programmation.

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 (idées de solutions). Ce chapitre contient également l'inventaire et la description des travaux qui auraient déjà été effectués pour ce projet.

1 Planification initiale

Cette planification initiale a été la première étape de ce projet. Elle a été réalisée lors du premier jour de TPI.

Avant tout, je souhaiterais préciser que ma planification initiale inclut des plages de 1h30 les vendredis, intitulées « Révision globale du code ».

Ces sessions sont initialement prévues pour être dédiées à l'optimisation du code, revoir des parties éventuellement obsolètes, ... cependant, selon les besoins, ces 1h30 peuvent être également utilisées pour rattraper tout retard dans d'autres aspects du projet, comme la rédaction du rapport.

Séquence 1			Date: lundi, 13 mai 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Réunion	3	0h15min	Découverte et validation du Cahier des Charges		
Réunion	5	0h25min	Discussion avec Monsieur Wenger (expert)		
Autres	8	0h40min	"Mise en place" du projet : création du GitHub, mise en page des fichiers (JdT, rapport, planif, ...)		
Analyse - Planification initiale	30	2h30min	Rédaction de la planification initiale		
Doc - Journal de travail	2	0h10min	Rédaction du journal de travail		
Total tranche	48	4h			

Séquence 2			Date: lundi, 13 mai 2024		Après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Analyse - Planification initiale	27	2h15min	Rédaction de la planification initiale		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	28	2h20min			

Séquence 3			Date: mardi, 14 mai 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Analyse - Documentation	24	2h	Choix de la méthode de gestion de projet, mise en place d'une stratégie de test, analyse du CDC (architecture MVC ?)		
Doc - Rapport	23	1h55min	Création et rédaction du rapport		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	48	4h			

Séquence 4			Date: mardi, 14 mai 2024		Après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Absences - imprévus	12	1h	Rendez-vous médical (IRM)		
Implémentation - Base de données	15	1h15min	Modélisation (MCD, MLD) de la base de données (sur Looping)		
Doc - Journal de travail	1	0h5min	Création de la base de données (MySQL)		
			Rédaction du journal de travail		
Total tranche	28	2h20min			

Séquence 5			Date: mercredi, 15 mai 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	15	1h15min	Rédaction du rapport		
Implémentation - Interfaces	12	1h	Création de l'interface du formulaire de création d'un compte		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	28	2h20min			

Séquence 6			Date: jeudi, 16 mai 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Implémentation - Fonctionnalités	12	1h	Implémentation d'une méthode de hashage		
Implémentation - Fonctionnalités	12	1h	Implémentation de la fonctionnalité de création d'un compte		
Tests	8	0h40min	Tests de la fonctionnalité (hashage, sécurité, insertion, ...)		
Implémentation - Interfaces	12	1h	Création de l'interface du formulaire de login (connexion à un compte)		
Implémentation - Fonctionnalités	3	0h15min	Implémentation du login (hashage, sécurité, ...)		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	48	4h			

Séquence 7			Date: jeudi, 16 mai 2024		Après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Absences - imprévus	12	1h	Rendez-vous médical (CHUV)		
Implémentation - Interfaces	9	0h45min	Implémentation du login (hashage, sécurité, ...)		
Tests	8	0h40min	Test de la fonctionnalité de login (sécurité, insertion, ...)		
Implémentation - Interfaces	8	0h40min	Création de l'interface principale de l'application + responsive (redimensionnement)		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	38	3h10min			

Séquence 8			Date: vendredi, 17 mai 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Implémentation - Interfaces	24	2h	Suite de la création de l'interface principale de l'application + responsive (redimensionnement)		
Doc - Rapport	23	1h55min	Rédaction du rapport		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	48	4h			

Séquence 9			Date: vendredi, 17 mai 2024		Après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Implémentation - Révisions	18	1h30min	Révision globale du code (ajustements nécessaires, code incohérent, imprévues ...)		
Implémentation - Fonctionnalités	12	1h	Implémentation de la fonctionnalité permettant de renommer les colonnes (formulaire ? Possibilités ?)		
Tests	7	0h35min	Tests de la fonctionnalité		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	38	3h10min			

Séquence 10			Date: mardi, 21 mai 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Implémentation - Fonctionnalités	39	3h15min	Implémentation de la fonctionnalité permettant d'ajouter de nouvelles colonnes (placements !, responsive)		
Tests	8	0h40min	Tests de la fonctionnalité		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	48	4h			

Séquence 11			Date: mardi, 21 mai 2024		Après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Implémentation - Fonctionnalités	27	2h15min	Implémentation de la fonctionnalité permettant de supprimer une colonne (placements des colonnes ?)		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	28	2h20min			

Séquence 12			Date: mercredi, 22 mai 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Tests	8	0h40min	Test de la fonctionnalité permettant de supprimer une colonne		
Doc - Rapport	19	1h35min	Rédaction du rapport		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	28	2h20min			

Séquence 13			Date: jeudi, 23 mai 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Implémentation - Fonctionnalités	18	1h30min	Création de l'objet Tâche		
Implémentation - Interfaces	18	1h30min	Création du formulaire d'ajout d'une nouvelle tâche (? Image, hyperlien)		
Implémentation - Fonctionnalités	11	0h55min	Implémentation de l'ajout d'une nouvelle tâche		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	48	4h			

Séquence 14			Date: jeudi, 23 mai 2024		Après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Implémentation - Fonctionnalités	3	0h15min	Suite de l'implémentation permettant l'ajout d'une nouvelle tâche		
Tests	8	0h40min	Test de la fonctionnalité		
Implémentation - Fonctionnalités	26	2h10min	Implémentation du Drag & Drop d'une image dans une tâche (format ?, resize de l'image)		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	38	3h10min			

Séquence 15			Date: vendredi, 24 mai 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Implémentation - Fonctionnalités	4	0h20min	Suite de l'implémentation du Drag & Drop d'une image		
Tests	8	0h40min	Test de la fonctionnalité		
Doc - Rapport	35	2h55min	Rédaction du rapport		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	48	4h			

Séquence 16			Date: vendredi, 24 mai 2024		Après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Implémentation - Révisions	18	1h30min	Révision globale du code (ajustements nécessaires, code incohérent, ...)		
Doc - Rapport	19	1h35min	Rédaction du rapport		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	38	3h10min			

Séquence 17			Date: lundi, 27 mai 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Implémentation - Fonctionnalités	47	3h55min	Implémentation du Drag & Drop entre colonnes		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	48	4h			

Séquence 18			Date: lundi, 27 mai 2024		Après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Tests	12	1h	Test de la fonctionnalité		
Implémentation - Révisions	15	1h15min	Révision du code dans son intégralité - optimisation		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	28	2h20min			

Séquence 19			Date: mardi, 28 mai 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Implémentation - Révisions	21	1h45min	Suite de la révision du code - optimisation		
Tests	26	2h10min	Tests de toutes les fonctionnalités		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	48	4h			

Séquence 20			Date: mardi, 28 mai 2024		Après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Tests	27	2h15min	Suite des tests		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	28	2h20min			

Séquence 21			Date: mercredi, 29 mai 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	27	2h15min	Rédaction du rapport		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	28	2h20min			

Séquence 22			Date: jeudi, 30 mai 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	47	3h55min	Rédaction du rapport		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	48	4h			

Séquence 23			Date: jeudi, 30 mai 2024		Après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	37	3h5min	Rédaction du rapport		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	38	3h10min			

Séquence 24			Date: vendredi, 31 mai 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	47	3h55min	Rédaction du rapport		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	48	4h			

Séquence 25			Date: vendredi, 31 mai 2024		Après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	37	3h5min	Rédaction du rapport		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	38	3h10min			

Séquence 26			Date: lundi, 3 juin 2024		Matin
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	47	3h55min	Rédaction du rapport		
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail		
Total tranche	48	4h			

Séquence 27			Date: lundi, 3 juin 2024		Après-midi
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...	
Doc - Rapport	16	1h20min	Rédaction du rapport		
Doc - Journal de travail	6	0h30min	Rédaction du journal de travail		
Autres	6	0h30min	FIN DU TPI - Envoi des livrables (! Nommage, formats, adresse mail !)		
Total tranche	28	2h20min			

2 Analyse

2.1 Planification détaillée

2.2 Méthodologie

2.2.1 Méthode de gestion de projet

La méthode de gestion de projet « Waterfall », plus particulièrement la méthode des « 6 pas » a été choisie et va être appliquée à ce projet.

Les 6 étapes de cette méthode sont les suivantes :

S'informer – Planifier – Décider – Réaliser – Contrôler – Evaluer

Le cahier des charges (par conséquent, les objectifs du projet également) ayant clairement été défini et validé au lancement du projet ainsi que la date de rendue, la méthode des 6 pas s'est rapidement présentée comme une solution fiable, envisageable et adaptée à la situation.

De plus, la structure des canevas fournis pour le journal de travail, la planification ainsi que le rapport correspond à la méthodologie des 6 pas. Par exemple, la tâche « Analyse – Planification initiale » s'inscrit totalement dans les étapes « S'informer, Planifier et Décider ».

En outre, étant une personne que l'on peut qualifier de perfectionniste, cette méthode structurée me permettra de m'éviter de tomber dans un peaufinage excessif des détails, qui conduirait à une mauvaise gestion du temps et entraînerait un potentiel retard. De ce fait, la progression linéaire et segmentée des 6 pas me servira de fil conducteur et m'encadrera dans mon travail.

Concrètement, comment cette méthode va s'appliquer à mon projet ?

J'ai d'abord élaboré une planification initiale qui, implicitement, suit les 6 pas. En effet, comme je l'ai déjà expliqué précédemment, la tâche « Analyse – Planification initiale » est la parfaite représentation de cette méthodologie, car pour faire cette planification, j'ai commencé par lire et analyser le cahier des charges (s'informer), j'ai ensuite dû faire ma planification initiale (planifier) et en même temps choisir (décider) l'ordre dans lequel j'allais m'y prendre, par quelles fonctionnalités j'allais commencer...

J'ai ensuite séparé les fonctionnalités comme des « mini-projets » et à chaque mini-projet la méthode des 6 pas est appliquée.

Par exemple, j'ai fait de la « Possibilité de se connecter de manière sécurisée avec des comptes utilisateurs. », qui est une fonctionnalité demandée dans le cahier des charges, un mini-projet.

S'informer : Cette étape consiste à comprendre ce qu'il y a à réaliser et prendre conscience des défis à relever. La documentation sur internet peut déjà commencer à ce stade si j'estime que c'est utile.

Qu'est-ce que cette fonctionnalité implique ?

La création d'un formulaire permettant à l'utilisateur de se logger et un formulaire permettant de créer un compte si l'utilisateur n'a pas de compte.

Y a-t-il potentiellement des points qui vont me poser un ou des problèmes ?

L'aspect « sécurité » de la fonctionnalité (exécution des requêtes sécurisée, utiliser quelle méthode de hachage du mot de passe, ...) et l'aspect « Notification » du modèle MVC (valider qu'une donnée a été insérée correctement dans la base de données et avertir l'utilisateur via la vue).

Ces points problématiques sont-ils importants ?

L'aspect sécurité l'est. Tandis que l'aspect « Notification » l'est moins, car il n'est pas mentionné dans le cahier des charges, c'est uniquement une contrainte de l'architecture MVC.

Comment puis-je résoudre ces problèmes ?

En me documentant sur internet, en utilisant ChatGPT, en demandant de l'aide à un camarade ou à mon chef de projet.

Planifier et Décider : C'est le moment de mettre en place ma planification détaillée pour ce mini-projet sur laquelle je vais me baser pour la partie « Réaliser ».

Si j'ai détecté de potentiels problèmes à la question précédente, combien de temps leurs résolutions risquent de me prendre ?

Cela risque de prendre une heure supplémentaire.

Quelles sont les priorités ?

D'abord, l'utilisateur doit pouvoir se connecter à un compte existant et doit également pouvoir créer un nouveau compte. Ensuite, la solution doit fonctionner de manière sécurisée. Pour finir, le code doit être optimisé.

Combien de temps maximum j'accorde à la réalisation de cette fonctionnalité (mini-projet) ?

Jusqu'au vendredi 17 mai.

Décider : Ce sont les décisions importantes à prendre avant de se lancer dans la réalisation afin de structurer sa façon de faire.

S'il y a plusieurs solutions pour la résolution des problèmes pressentis, quelle(s) méthode(s) vais-je utiliser ?

A propos de la recherche de solutions, je vais surtout me documenter sur internet. Si cela ne suffit pas, j'utiliserai ChatGPT et si cela n'est pas encore suffisant, je demanderai de l'aide à mon chef de projet. Concernant le hachage du mot de passe, je pense utiliser BCrypt et mes recherches internet me confirmeront mon choix.

Quels sont les critères de réussite ?

L'utilisateur doit pouvoir créer un compte et se connecter et le processus doit être sécurisé et doit au minimum protéger contre les injections SQL et haché le mot de passe.

Réaliser : La réalisation est le cœur de la tâche. C'est la pratique.

Contrôler : Le contrôle est l'étape de test. C'est le moment de s'assurer que ce qui a été fait dans la partie « Réaliser » répond aux attentes et aux objectifs.

La solution a-t-elle été validée par les tests de validation de la stratégie de tests ?

Evaluer : Cette partie permet de faire une courte rétrospective ce qui vient d'être fait qui permettra de déterminer plusieurs choses.

Y a-t-il des objectifs supplémentaires que je peux considérer si j'ai davantage de temps à ma disposition ?

La façon dont je traite les « notifications (MVC) », que j'ai laissé de côté, pourrait être revue et optimisée (code répétitif). Le code pourrait globalement être optimisé.

A quel point ces objectifs supplémentaires sont-ils importants ? Vont-ils réellement amener une plus-value au projet ?

L'optimisation du code pourrait rendre le tout plus simple à gérer, mais ce n'est pas primordial car la solution reste fonctionnelle et sans cela.

Si j'ai perdu du temps, où et pourquoi ai-je perdu du temps ?

La conception des maquettes des interfaces m'a fait « perdre » du temps car je n'avais pas prévu du temps pour cela dans ma planification. J'ai également sous-estimé le temps que la recherche de documentation sur internet m'a prise.

2.2.2 Sauvegarde

La perte de données est un risque omniprésent, encore plus dans un projet comme celui-ci et plus généralement dans le métier d'informaticien. Il est crucial de hautement considérer ce danger.

C'est pourquoi j'ai décidé d'établir une « routine » de backup facile à mettre en place afin de minimiser ce risque sans que cela ne soit trop coûteux en termes de temps.

La routine est la suivante :

- **Utilisation de GitHub :** GitHub, en plus de permettre le versioning et d'optimiser la portabilité du projet, est une véritable solution de sauvegarde viable. C'est pour ces raisons que j'ai décidé de créer mon projet « actif » sur GitHub.
- **Push régulier sur GitHub :** Au minimum, avant chaque pause ou fin de journée (toutes les 1h30), un commit et un push sont effectués, afin de garantir que les dernières modifications ont été sauvegardées.
- **Sauvegarde sur SSD et en local :** En complément de GitHub, une copie du TPI est également sauvegardée sur mon disque dur externe ainsi que sur l'ordinateur de travail, en local, après chaque mise à jour sur GitHub.

2.3 Stratégie de test

2.4 Conception

2.4.1 Modélisation de la base de données

Avant de me lancer dans la création et l'écriture du script SQL de ma base de données, j'ai appliqué et suivi la méthode MERISE pour modéliser ma base de données afin de m'assurer de sa cohérence et qu'elle pourra supporter mon application.

MERISE, qui signifie Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise, est une méthode d'analyse, de conception et de gestion de projet, surtout utilisée dans le monde de l'informatique.

Dans le cadre de l'élaboration d'une base de données, MERISE propose des « outils » permettant de réaliser une base de données de manière méthodique. Suivre cette méthodologie consiste à d'abord élaborer un Modèle Conceptuel des Données (MCD), qui est un schéma qui met particulièrement en évidence comment les éléments sont liés entre eux.

Cette étape analytique permet de faire une première « ébauche » de la structure de la base de données à partir de ce qui est demandé (dans mon cas, par le cahier des charges).

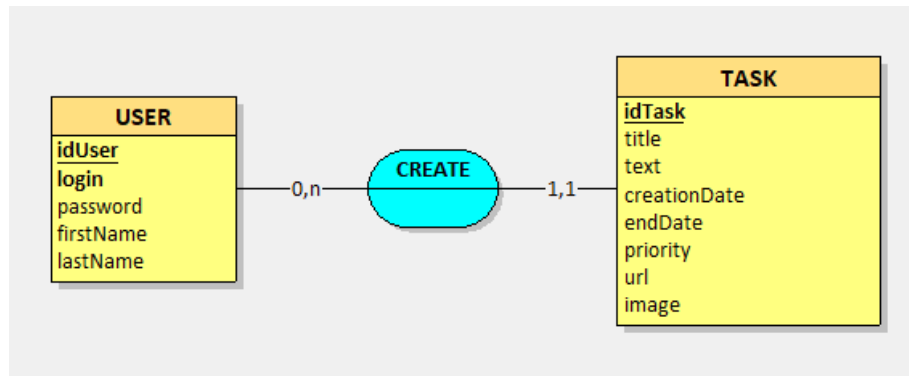


Figure 1 : Modèle Conceptuel des Données

Une fois que le MCD est terminé, le Modèle Logique des Données (MLD) peut être construit. Le MLD reprend le contenu du MCD et le transpose dans un format plus « technique », prêt à être implémenté dans un Système de Gestion de Base de Données. Par exemple ici, les entités TASK et USER deviennent des tables et les associations (CREATE), qui représentent les relations entre les entités, sont traduits en clés étrangères.

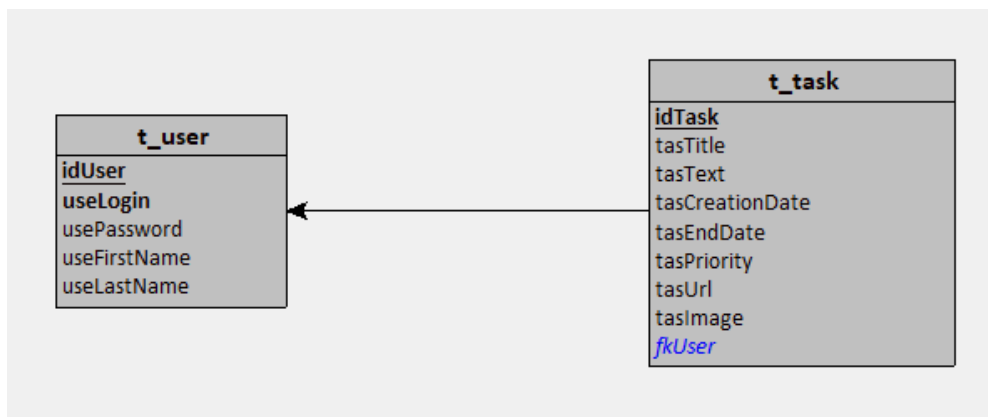


Figure 2 : Modèle Logique des Données (fait sur Looping)

Le Modèle Physique des données (MPD) est la dernière étape de la méthode MERISE dans la conception d'une base de données. Il représente concrètement la structure de la base de données en détaillant notamment les types de données (int, varchar, ...). Contrairement au MCD et au MLD, le MPD est spécifique au système de gestion de base de données, ici SQL.

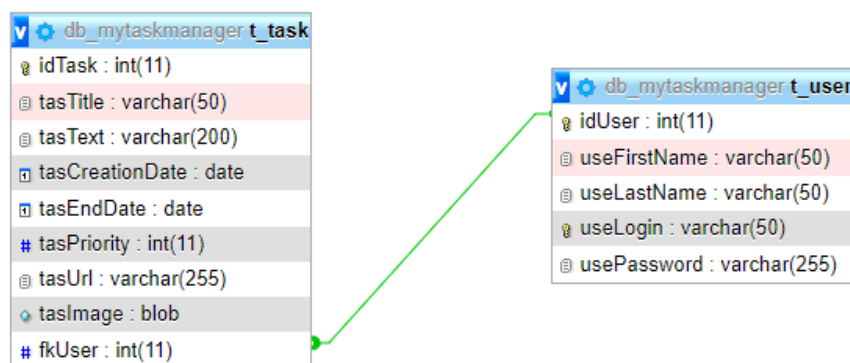


Figure 3 : Modèle Physique des Données (représentation depuis PHPMYAdmin)

2.4.2 UwAmp

UwAmp est un serveur Wamp léger et portable permettant d'exécuter Apache, PHP, PHPMyAdmin, etc. qui donne la possibilité de tester son site web ou encore sa base de données.

J'ai choisi d'utiliser UwAmp en lieu et place de Docker car UwAmp est une solution portable, ne nécessitant aucune installation. A travers les divers projets (PHP, MySQL) que j'ai menés, je me suis familiarisé avec UwAmp et toujours dans cette quête de gestion du temps, j'ai préféré privilégier un logiciel que je connais très bien et qui répond à ce dont j'ai besoin pour réussir ce projet à la place de Docker, qui, malgré ses nombreuses qualités, n'est pas un outil avec lequel j'ai eu l'opportunité de vraiment travailler.

2.4.3 Maquettes

Pour établir une cohérence graphique, j'ai réalisé des maquettes de chaque interface de mon application sur Figma.

J'ai commencé par chercher un logo pour mon application. J'ai essayé d'utiliser ChatGPT 4 pour générer un logo correspondant à une application « To Do List », en vain. Les résultats proposés ne m'ont pas satisfait malgré les multiples essais.

J'ai donc fini par réaliser le logo moi-même sur Canva. Je voulais une image simpliste et parlante. J'ai choisi le jaune comme couleur principale car, au-delà d'être une couleur qui attire l'œil, elle me rappelle les Post-It que beaucoup utilisaient pour noter leurs tâches.



Figure 4 : Post-It

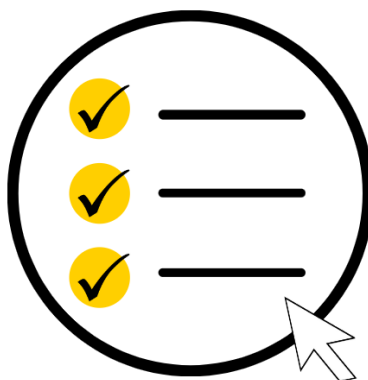


Figure 5 : Logo de MyTaskManager

2.4.4 Architecture MVC

MVC est une architecture logicielle qui permet de séparer le développement d'une application en 3 « composants » distincts : **Modèle**, **Vue** et **Contrôleur**.

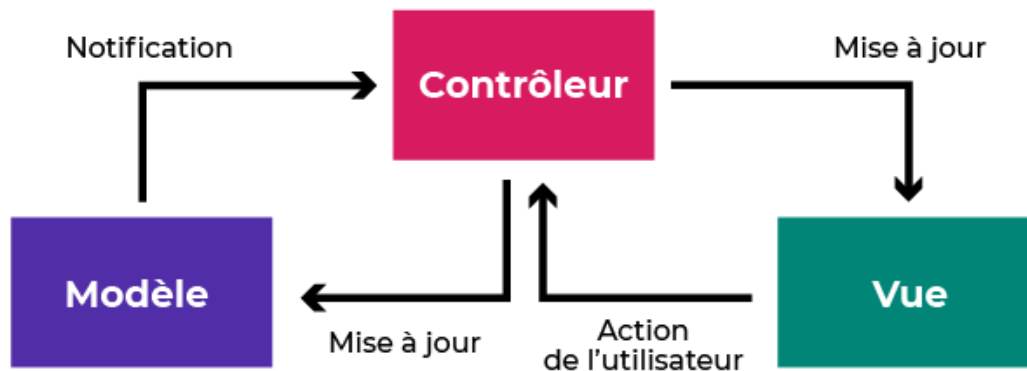


Figure 6 : Fonctionnement du MVC schématisé

Chaque composant se charge d'un aspect spécifique de l'application.

Le Modèle : Il contient les données manipulées par le programme. Il assure la gestion de ces données et garantit leur intégrité. Dans le cas typique d'une base de données, c'est le modèle qui a accès à la base de données¹. Concrètement, il contient les opérations servant à manipuler les données, en incluant, par exemple, une méthode permettant d'ajouter (créer) un nouvel utilisateur.

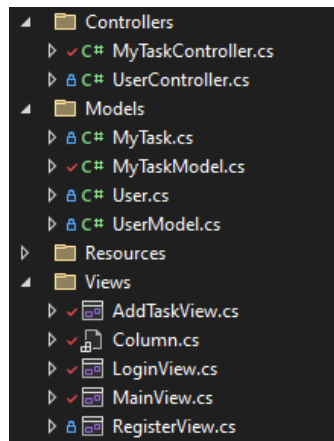
La Vue : Elle gère toute la logique interface utilisateur. Elle génère l'interface que l'utilisateur va voir et utiliser. Elle est chargée d'afficher les données (fournies par le Modèle) et de recevoir les actions de l'utilisateur (clic de souris, remplissage d'un champ, etc.).

Le Contrôleur : C'est l'intermédiaire entre la Vue et le Modèle. Le contrôleur reçoit les événements enclenchés par l'utilisateur depuis la Vue et avertit, par conséquent, le modèle en lui indiquant quoi faire. Inversement, le modèle notifie le contrôleur des changements pour que ce dernier mette à jour la Vue.

Le pattern MVC étant la seule architecture logicielle que l'on a réellement étudiée et dans l'intérêt de « structurer » le code, ce modèle MVC a donc logiquement été utilisé et suivi pour le développement de l'application.

Voici comment j'ai organisé mon MVC :

¹ Explication de l'architecture MVC :
<https://www.irif.fr/~carton/Enseignement/InterfacesGraphiques/Cours/Swing/mvc.html>



Toujours dans l'optique de structurer le code et répartir spécifiquement les aspects de l'application, j'ai opté pour une séparation entre la partie Utilisateur et la partie Tâche du programme. C'est pourquoi j'ai deux Contrôleurs (MyTaskController et UserController) ainsi que deux Modèles (MyTaskModel et UserModel).

3 Réalisation

3.1 Compte utilisateur

3.1.1 Hachage

Les comptes utilisateurs contenant des informations particulièrement sensibles, notamment comme un mot de passe, il est indispensable de protéger et sécuriser ces données.

C'est là que le hachage intervient. En effet, le hachage est un processus unidirectionnel qui permet de convertir une donnée comme un mot de passe en texte chiffré à l'aide « d'algorithmes de hachage »².

Dans le cadre de mon application, j'ai opté pour l'utilisation de la bibliothèque BCrypt, principalement car elle offre une méthode de hachage spécialement conçue et adaptée pour les mots de passe. Au-delà du côté « pratique » en termes de développement, BCrypt utilise également le salage dans sa fonction de hachage. Le salage ajoute des caractères aléatoires au hachage, assurant ainsi que chaque hash généré est unique. Par exemple, dans le cas où deux utilisateurs ont le même mot de passe, BCrypt produit deux hashes distincts, ce qui renforce la sécurité et permet d'éviter les « attaques par tables arc-en-ciel »³.

```
string hashedPassword = BCrypt.Net.BCrypt.HashPassword(newUser.Password);
```

3.1.2 Création de compte

Comme je l'ai déjà mentionné précédemment, mon application propose la possibilité de créer un compte utilisateur. Lorsque l'utilisateur a rempli le formulaire ([design du formulaire](#)) de création d'un compte, au clic du bouton « Valider », l'action est récupérée par la vue et envoyé au Contrôleur.

² Définition du hachage de Mail in Black :
<https://www.mailinblack.com/ressources/blog/le-salage-de-mots-de-passe-une-couche-de-securite-indispensable/>

³ Pourquoi BCrypt est une solution pour le hachage des mots de passe :
<https://nordvpn.com/fr/blog/what-is-bcrypt/>

```
public bool RegisterUserInDatabase(User newUser)
{
    try
    {
        // Ouvre la connexion à la base de données
        _connection.Open();

        // Stocke et hache le mot de passe
        string hashedPassword = BCrypt.Net.BCrypt.HashPassword(newUser.Password);

        // Prépare la requête SQL pour insérer un nouvel utilisateur dans la base de données
        MySqlCommand query = new MySqlCommand(
            "INSERT INTO t_user VALUES(NULL, @useFirstName, @useLastName, @useLogin, @usePassword)", _connection);

        // Data Binding (permet d'éviter les injection SQL)
        query.Parameters.AddWithValue("@useFirstName", newUser.FirstName);
        query.Parameters.AddWithValue("@useLastName", newUser.LastName);
        query.Parameters.AddWithValue("@useLogin", newUser.Login);
        query.Parameters.AddWithValue("@usePassword", hashedPassword);

        // Exécute la requête SQL
        int result = query.ExecuteNonQuery();

        // Vérifie si l'opération a fonctionné
        if (result > 0)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    catch (Exception)
    {
        return false;
    }
    finally
    {
        _connection.Close();
    }
}
```

```
public partial class RegisterView : Form
{
    2 références
    public UserController UserController { get; set; }

    1 référence
    public RegisterView(UserController userController)
    {
        InitializeComponent();
        UserController = userController;
    }

    1 référence
    private void btnCreateUser_Click(object sender, EventArgs e)
    {
        if(UserController.RegisterUser(tbxFirstName.Text, tbxLastName.Text, tbxLogin.Text, tbxPassword.Text))
        {
            MessageBox.Show("Nouveau compte utilisateur crée avec succès !", "Création d'un compte");
            this.Close();
        }
        else
        {
            MessageBox.Show("Echec de la tentative de création d'un compte !", "Création de compte");
        }
    }
}
```

4 Tests

5 Conclusion

6 Annexes

6.1 Table des illustrations

Figure 1 : Modèle Conceptuel des Données.....	Erreur ! Signet non défini.
Figure 2 : Modèle Logique des Données (fait avec le logiciel Looping)	Erreur ! Signet non défini.
Figure 3 : Modèle Physique des Données (représentation depuis PHPMysqlAdmin)	Erreur ! Signet non défini.
Figure 4 : Fonctionnement du MVC schématisé	20