**Projet TPI – Gestionnaire de tâches**



Tristan Gerber – MID4

ETML N501

Du 2 Mai 2022 au 1er Juin 2022

Chef de projet : Roberto Ferrari

Experts : Joseph Brandenberg, Serge Wenger

**Table des matières**

[1 Spécifications 3](#_Toc104800126)

[1.1 Titre 3](#_Toc104800127)

[1.2 Description 3](#_Toc104800128)

[1.3 Matériel et logiciels à disposition 3](#_Toc104800129)

[1.4 Prérequis 3](#_Toc104800130)

[1.5 Cahier des charges 3](#_Toc104800131)

[1.5.1 Objectifs et portée du projet 3](#_Toc104800132)

[1.5.2 Caractéristiques des utilisateurs et impacts 4](#_Toc104800133)

[1.5.3 Contraintes 4](#_Toc104800134)

[1.5.4 Si le temps le permet … 4](#_Toc104800135)

[1.6 Méthodes de validation des solutions 4](#_Toc104800136)

[1.7 Les points suivants seront évalués 4](#_Toc104800137)

[1.8 Validation et conditions de réussite 4](#_Toc104800138)

[2 Analyse 4](#_Toc104800139)

[2.1 Opportunités 5](#_Toc104800140)

[2.2 Méthodologie de projet 5](#_Toc104800141)

[2.2.1 S’informer 5](#_Toc104800142)

[2.2.2 Planifier 5](#_Toc104800143)

[2.2.3 Décider 5](#_Toc104800144)

[2.2.4 Réaliser 5](#_Toc104800145)

[2.2.5 Contrôler 5](#_Toc104800146)

[2.2.6 Évaluer 5](#_Toc104800147)

[2.3 Normes de codage 6](#_Toc104800148)

[2.4 Déploiement de l’application 6](#_Toc104800149)

[2.5 Langages utilisés 6](#_Toc104800150)

[2.6 Stockage des données 6](#_Toc104800151)

[2.7 Sauvegarde du projet 7](#_Toc104800152)

[2.8 Tests 7](#_Toc104800153)

[3 Conception 8](#_Toc104800154)

[3.1 Maquettes & Structure de l’application 8](#_Toc104800155)

[3.2 Infographie de la date d’échéance 11](#_Toc104800156)

[3.3 Navigation 12](#_Toc104800157)

[3.4 Conception des tests 12](#_Toc104800158)

[3.5 Planification détaillée 13](#_Toc104800159)

[4 Réalisation 14](#_Toc104800160)

[4.1 Dossier de Réalisation 14](#_Toc104800161)

[4.1.1 Programmes & versions utilisés 14](#_Toc104800162)

[4.1.2 Base de données 15](#_Toc104800163)

[4.1.3 MVVM & Code 15](#_Toc104800164)

[4.2 Compilation 16](#_Toc104800165)

[4.2.1 Réalisation des tests 16](#_Toc104800166)

[5 Conclusion 17](#_Toc104800167)

[5.1 Bilan des fonctionnalités demandées 17](#_Toc104800168)

[5.2 Points techniques évalués spécifiques au projet 17](#_Toc104800169)

[5.3 Résultat visuel 18](#_Toc104800170)

[5.4 Bilan de la planification 20](#_Toc104800171)

[5.5 Bilan personnel 22](#_Toc104800172)

[6 Webographie 23](#_Toc104800173)

[7 Table des illustrations 23](#_Toc104800174)

[8 Annexes 24](#_Toc104800175)

[8.1 Planification 24](#_Toc104800176)

[8.2 Journal de travail 34](#_Toc104800177)

# Spécifications

En dernière année, un projet qui sert d’examen final, un TPI (Travail Pratique Individuel) est réalisé par les élèves. Ce rapport concerne mon TPI dont le chef de projet est M. Ferrari. Cette partie du rapport correspond à la première étape de la méthode des six pas.

## Titre

Gestionnaire de tâches

## Description

Ce projet consiste en la création d’une application de gestion des tâches.

## Matériel et logiciels à disposition

**PC standard de l’ETML**

**Visual Studio 2019** – Utilisé pour coder – Préinstallé

**Windows 10** – Système d’exploitation de l’ordinateur

**Suite Office** – Utilisé pour réaliser la documentation – Préinstallé

**Pencil Project** – Utilisé pour réaliser les maquettes – Installé avec l’exécutable : *K:\INF\Maitres-Eleves\Outils\\_APP-Portables\03-developpement\PencilProjectPortable\_3.0.3\_English.paf.exe*

**Téléphone de modèle Google Pixel 2** – Utilisé pour héberger, lancer et tester l’application – Fourni par M. Lymberis.

**GitHub & Git Bash** – Utilisé dans le cadre de la sauvegarde et du versionning du projet – Installé depuis [*https://git-scm.com/downloads*](https://git-scm.com/downloads)*. Afin d’installer Git Bash, il suffit de télécharger le Setup ou la version Portable en appuyant sur « next » à toutes les étapes.*

DBMain

## Prérequis

Compétences en bases de données  Modules 104 et 153

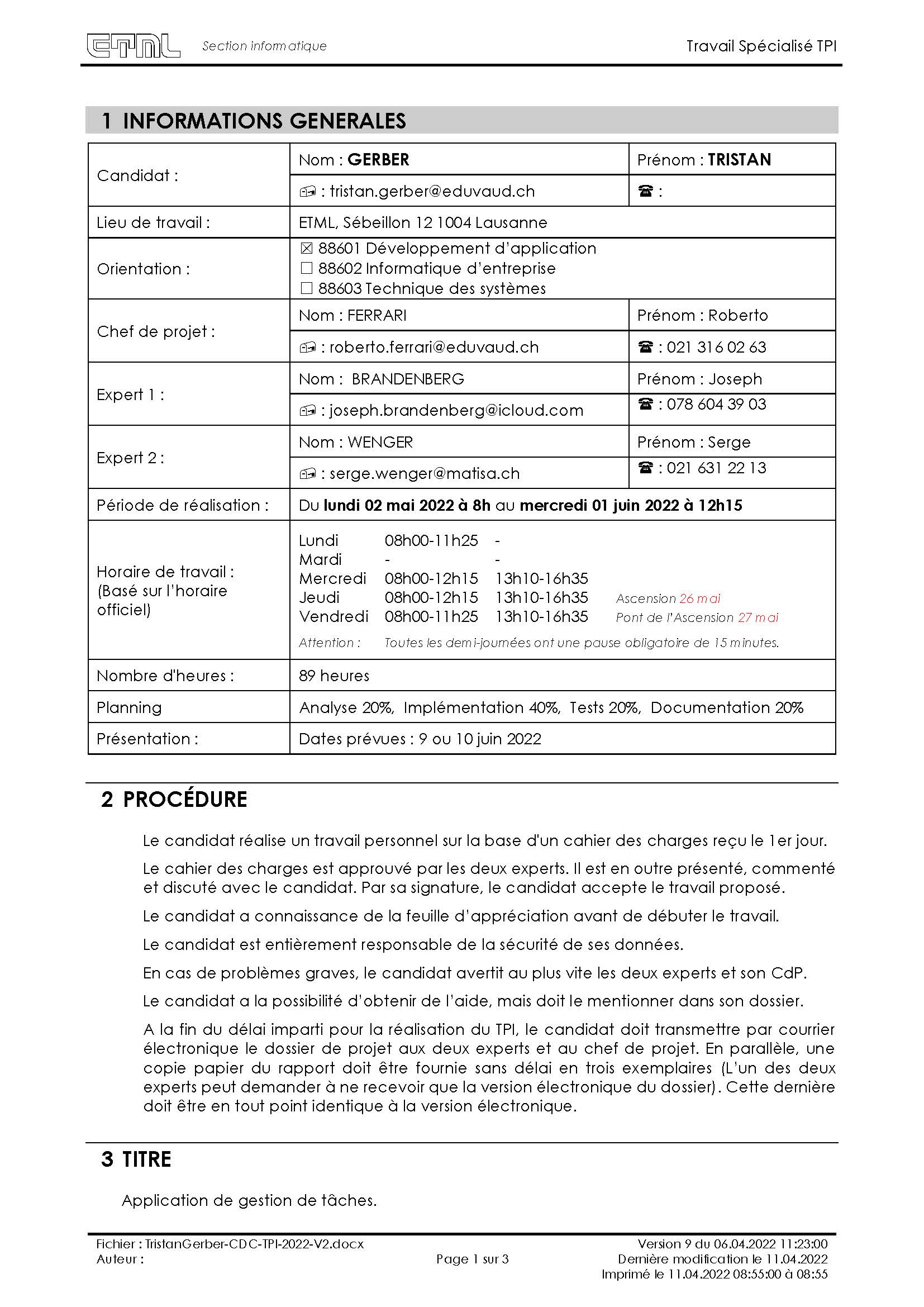
Compétences en programmation  Modules 120, 226A et B, 326, 403, 404, 411, 335

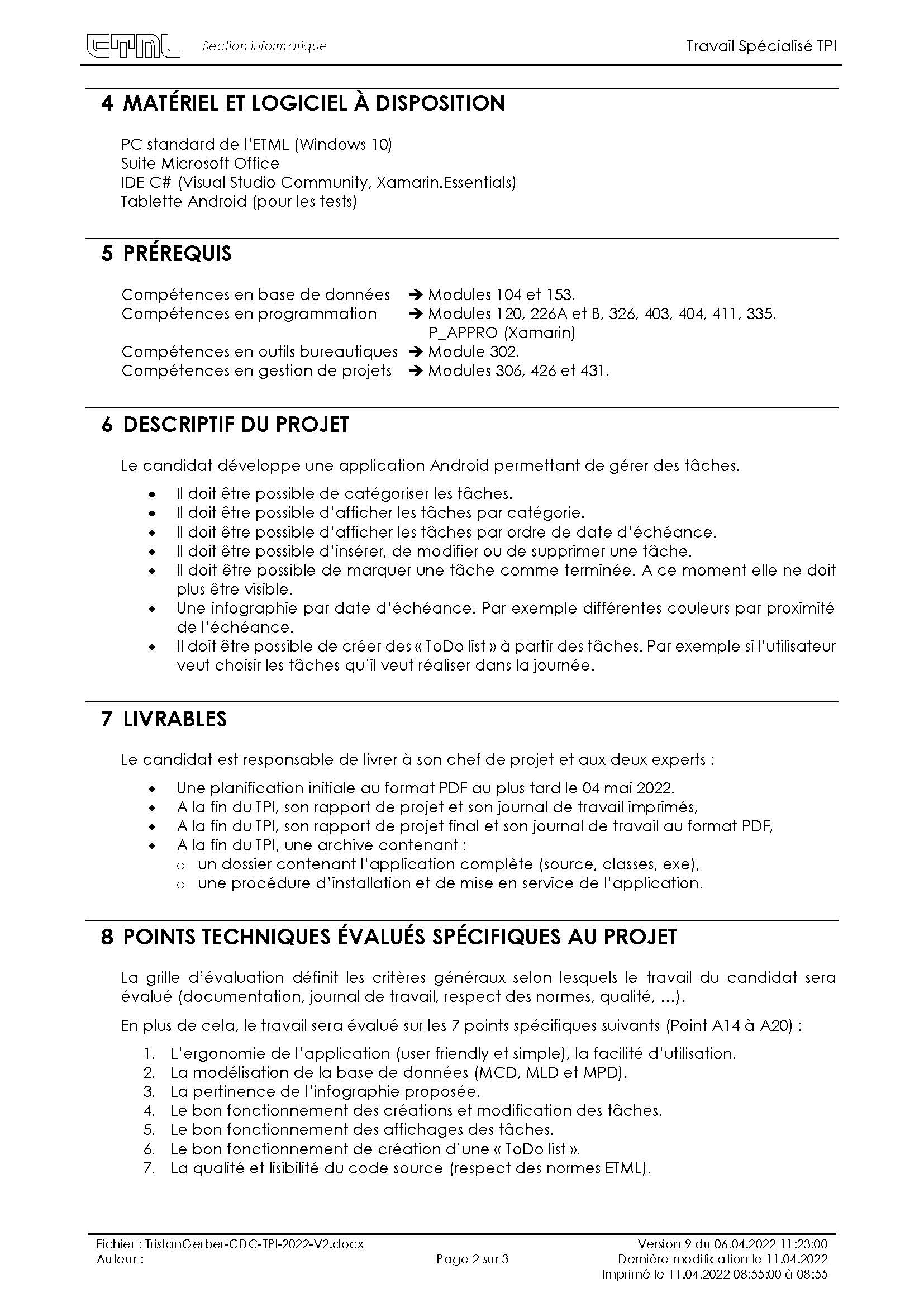
P\_APPRO (Xamarin)

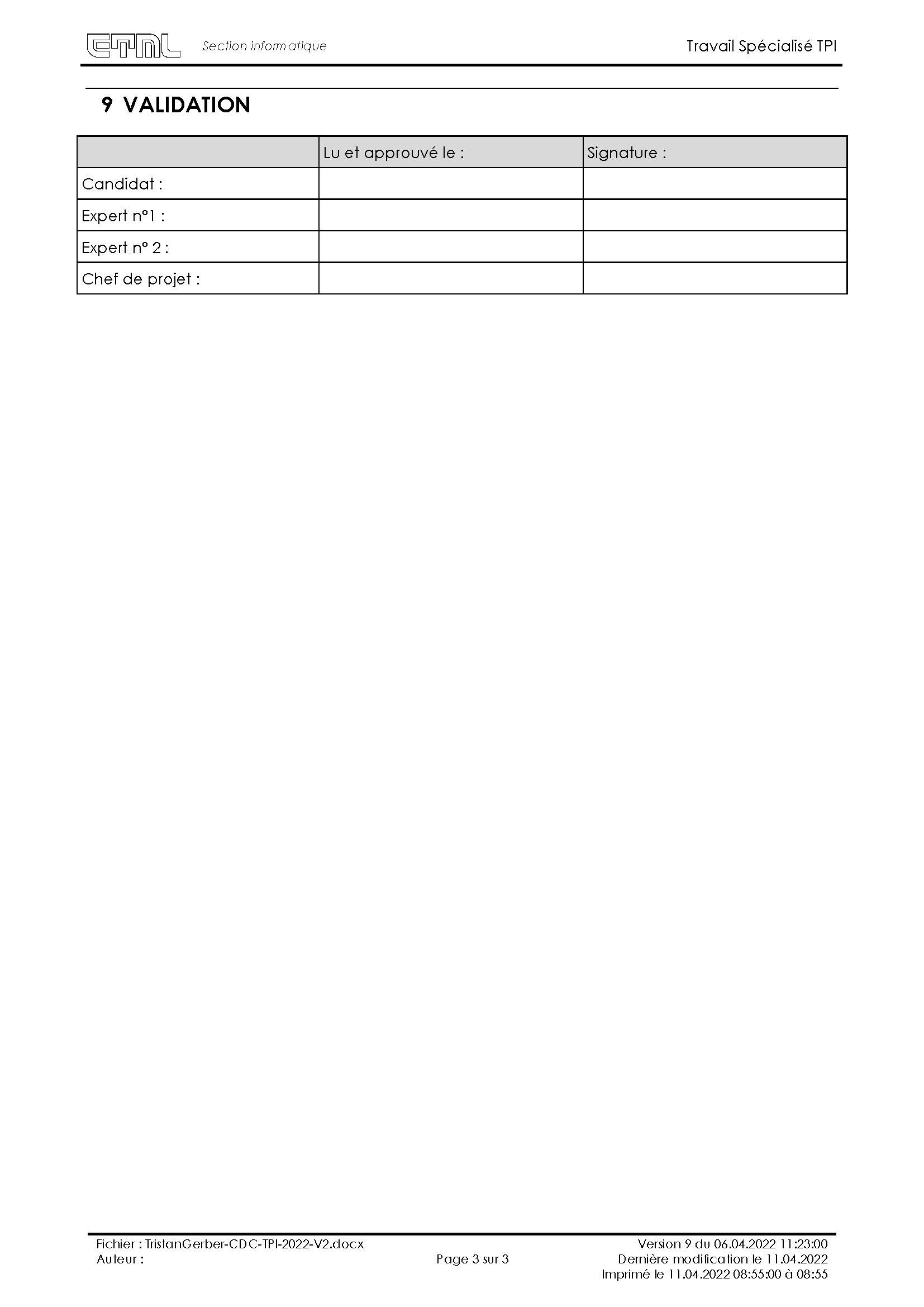
Compétences en outils bureautiques  Modules 104 et 153

Compétences en gestion de projets  Modules 104 et 153

## Cahier des charges

Ci-dessous on retrouve le cahier des charges qui m’a été donné au début du projet.





Le cahier des charges est disponible :

* Sous le fichier [DLS-Cdc-GestionStockEtml.docx](file:///C:\Users\trigerber\Downloads\DLS-Cdc-GestionStockEtml.docx)
* En annexe à ce rapport.

### Objectifs et portée du projet

1. Catégorisation des tâches
2. Affichage des tâches par catégorie
3. Affichage des tâches par ordre de date d’échéance
4. Insertion, modification et suppression de tâches
5. Possibilité de marquer une tâche comme terminée, la rendant invisible
6. Infographie par date d’échéance
7. Création de « ToDoList » à partir des tâches

### Caractéristiques des utilisateurs et impacts

Le public cible est constitué de personnes de toutes les tranches d’âge, sexe ou nationalité parlant français et souhaitant gérer leurs tâches quotidiennes et prendre en main leur temps. L’interface doit être simple d’utilisation, l’application utilisable en tout temps avec ou sans connexion à internet.

### Contraintes

Temps pour le projet : 1 mois

Émulateur ou téléphone pour compiler

### Si le temps le permet …

## Méthodes de validation des solutions

Tests manuels sur la sécurité de l’application et du bon fonctionnement des différentes fonctionnalités.

Si le temps le permet, Test de l’application sur différents appareils de différentes tailles.

L’application devra être testée sur iOS. Il donc faudra utiliser un émulateur iOS ou directement connecter un téléphone de ce même OS au PC. L’application sera également à tester sur des écrans de tailles différentes et des nouvelles versions d’Android et d’iOS.

## Les points suivants seront évalués

* Le rapport
* Les planifications (initiale et détaillée)
* Le journal de travail
* Le code et les commentaires
* Les documentations de mise en œuvre et d’utilisation

## Validation et conditions de réussite

* Compréhension du travail
* Possibilité de transmettre le travail à une personne extérieure pour le terminer, le corriger ou le compléter
* Etat de fonctionnement du produit livré
* Produit livré dans les temps

# Analyse

Lors de l’analyse, on regarde le projet à faire sous tous les angles afin de se faire une bonne idée du travail à fournir et de la direction à prendre. Cette partie du rapport correspond à la deuxième et troisième étape de la méthode des six pas.

## Opportunités

Ce projet a pour but de décider quant à ma qualification à recevoir mon CFC. Il servira de consolidation pour les bases en Xamarin déjà acquises précédemment durant le P\_APPRO et viendra vérifier mes compétences en gestion de projet, en programmation, en bureautique et en gestion de projets.

## Méthodologie de projet

Afin d’avoir un projet structuré, la méthode des 6 pas est utilisée lors de tout le projet. Ce choix est dû à sa bonne structuration. Celle-ci permet une meilleure planification à l’avance au détriment de la flexibilité lors de la création du projet. Étant donné que les objectifs de ce projet sont clairs, que le domaine est maitrisé et que le cahier des charges ne changera pas, il est plus adapté de tout planifier à l’avance.

Elle consiste en six étapes :

### S’informer

Des recherches sont réalisées autour du mandat afin d’analyser les besoins, les possibilités, les limitations et les mesures à prendre.

### Planifier

Définir le temps mis à disposition, les priorités, les solutions et les moyens à disposition afin d’avoir des tâches claires et séparées durant l’entièreté du projet.

### Décider

À l’aide des informations trouvées auparavant, décider comment et à l’aide de quel outil chaque tâche va être réalisée et les avantages et inconvénients de la solution par rapport aux autres.

### Réaliser

Réaliser les différentes tâches, s’ajuster à la planification et la décision sur les tâches, constater ce qui peut être véritablement fait dans les conditions de travail actuelles.

### Contrôler

Comprendre ce qui a pu mal se passer, ce qui s’est bien passé, le respect de la planification, les modifications à envisager, les conséquences des erreurs.

### Évaluer

Mesurer l’état d’avancement des travaux, faire une rétrospective sur les tâches qui ont posé des problèmes, trouver des améliorations à faire sur l’approche de projet, tirer une leçon du déroulement du projet.

## Normes de codage

Dans tout le programme en C#, les variables suivent le lowerCamerCase tandis que les constantes, les fonctions et les classes suivent le UpperCamelCase. Les commentaires en C# sont tous sous cette forme :

// Creates a row in the database with the borrow's informations

Chaque méthode et classe de l’application comporte un sommaire et chaque page de code comporte un entête sous cette forme :

/\* Developper : Tristan Gerber

\* Place : ETML, N501

\* Project creation date : 02.05.2022

\* Last updated : 02.05.2022 \*/

La langue utilisée dans les commentaires est la même que celle du code, l’anglais. C’est une langue dans laquelle je suis à l’aise et d’après moi, il est logique d’écrire le nom des objets et des commentaires dans la même langue que s’écrit le code.

La langue utilisée pour la partie visible de l’application est évidemment le Français, étant donné que les utilisateurs de l’application parleront sans doute cette langue.

## Déploiement de l’application

Le programme tournant sur un téléphone, le choix s’impose d’avoir :

* Un émulateur (Pour Android, iOS et Windows Phone)
* Un téléphone (Android)

M. Lymberis, s’occupant de la salle N501, me mettant à disposition un téléphone, je préfère le second choix. La création d’émulateurs aurait pris trop de temps sur le projet, l’émulateur iOS nécessite une installation spéciale (via iTunes) et l’exécution directe sur un téléphone évite tout problème lors du déploiement qui pourrait arriver si développé sur émulateur.

## Afficher l’image sourceLangages utilisés

Le C# sera utilisé pour le back-end, soit toutes les fonctions en liaison avec la base de données, les modèles et le côté algorithmique et logistique du programme. Il est très important car il est également utilisé par XAML en tant que « code-behind file ».

Le XAML est utilisé en tant que front-end. Il comprend tout un système de styles permettant de modifier la partie graphique de l’application. Il est un « dialecte » de XML, mettant en valeur la liaison au C#, ou la séparation du code visible par l’utilisateur et le code lui-même.

Tous ces langages sont réunis par Xamarin, une plateforme de développement mobile développée par Microsoft. Multiplateforme, elle supporte Android, iOS et Windows Phone.

## Stockage des données

Après une analyse des différentes possibilités de stockage de données, quatre options s’imposent :

1. **Shared Preferences** : Système de stockage de données basé sur un système de Clé/Valeur. Pratique pour stocker des valeurs désorganisées, p.ex un email ou un nom d’utilisateur.
2. **SQLite** : Système de base de données pouvant organiser et stocker de grosses quantités de données structurées sur le téléphone de l’utilisateur. Il ne peut stocker des données que localement.
3. **Stockage interne / externe**: Stocke des fichiers dans la mémoire interne et externe (Carte SD)
4. **Base de donnée MySql**: Système de base de données distant, hébergé sur un serveur de la salle de classe. Utilisable uniquement avec une connexion à internet, lorsque la base de donnée tourne sur le serveur.

L’option retenue est SQLite, car elle semble être la plus adaptée au projet. Les Shared Preferences ne sont pas assez structurées pour l’ampleur du projet et la base de données MySql perd son sens car il faut être sur le même réseau que le serveur, qu’il peut y avoir de problèmes du côté serveur et qu’il faut une connexion à internet. Or, l’application n’a pas besoin d’une connexion pour fonctionner. Quant au stockage interne / externe de fichiers, il manque de structure et c’est également le domaine dans lequel je possède le moins de compétences.

L’option retenue est SQLite car elle possède une bonne structure de base de données.

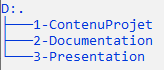
## Sauvegarde du projet

Les fichiers du projet seront retrouvés :

* Sur Teams dans le dossier :
* Sur GitHub :
* Sur l’ordinateur « INF-N501-M804 » dans le dossier « D:\DATA\trigerber-TPI »

La structure des fichiers utilisée suit celle-ci depuis le dossier trigerber-TPI.

|  |  |
| --- | --- |
| 1-ContenuProjet | Contient tout le code, les ressources et l’application finie produits lors de mon projet. |
| 2-Documentation | Contient la totalité de la documentation produite lors de mon projet. |
| 3-Presentation | La présentation PowerPoint finale ainsi que les outils nécessaires à celle-ci. |



## Tests

Les tests font partie intégrante du projet et ont une grosse importance dans celui-ci.

Après analyse, deux choix principaux sont à ma disposition afin de réaliser les tests :

1. Tests Unitaires : Pratique pour les applications réalisant des calculs, possédant des algorithmes compliqués et beaucoup de processus tournant en arrière-plan.
2. Tests d’Interface Utilisateur (UI Tests) : Permet de tester une interface utilisateur sur différents appareils. Peut reconnaître les crashs, les inconsistances dans les transitions par exemple ainsi que des problèmes de layout sur différents appareils.
3. Tests manuels suivant des procédures prédéfinies.

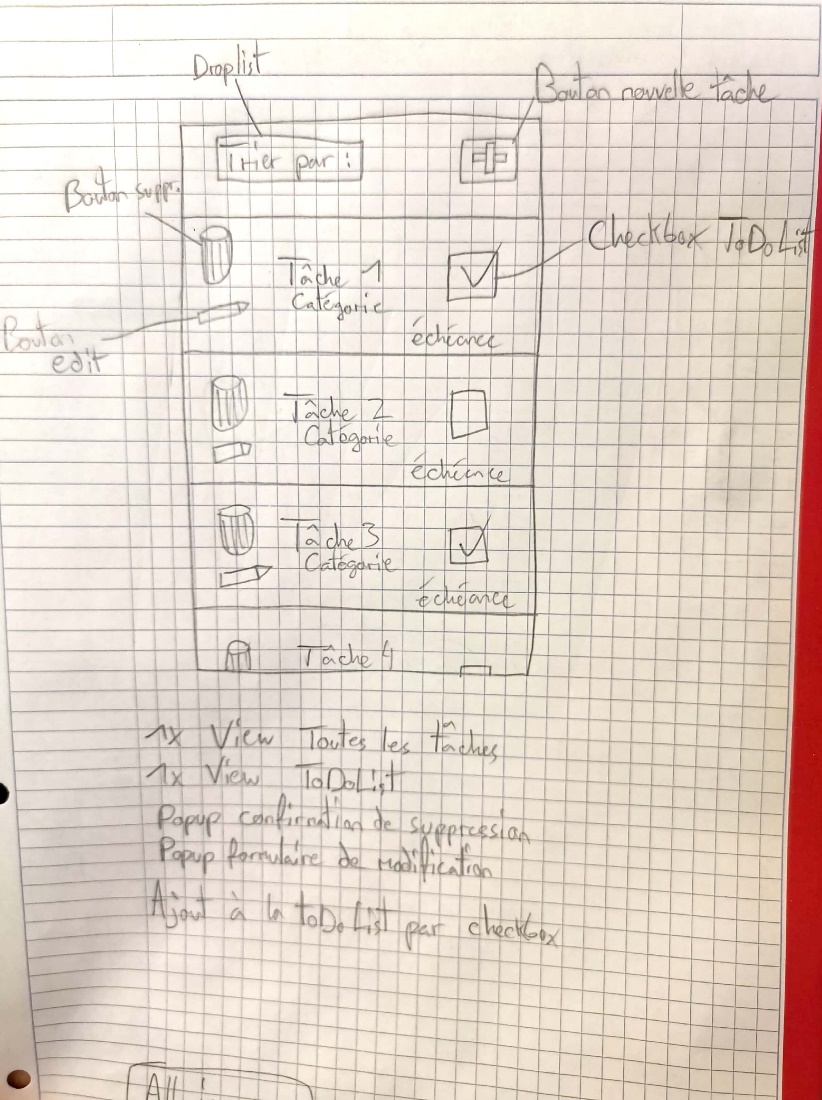
Les UI Tests étaient la méthode choisie par défaut, cependant, après discussion avec le chef de projet, il a été décidé de réaliser des tests manuels car l’application et le temps mis à disposition pour sa création étaient trop petits pour apprendre une méthode entière de test pouvant vite retarder le projet.

# Conception

Dans ce document est montré le résultat de l’analyse réalisée précédemment, les maquettes du fonctionnement attendu de l’application, la structure des pages ainsi que différentes mesures prises dans la conception de l’application. Cette partie correspond à la deuxième et troisième étape de la méthode des six pas.

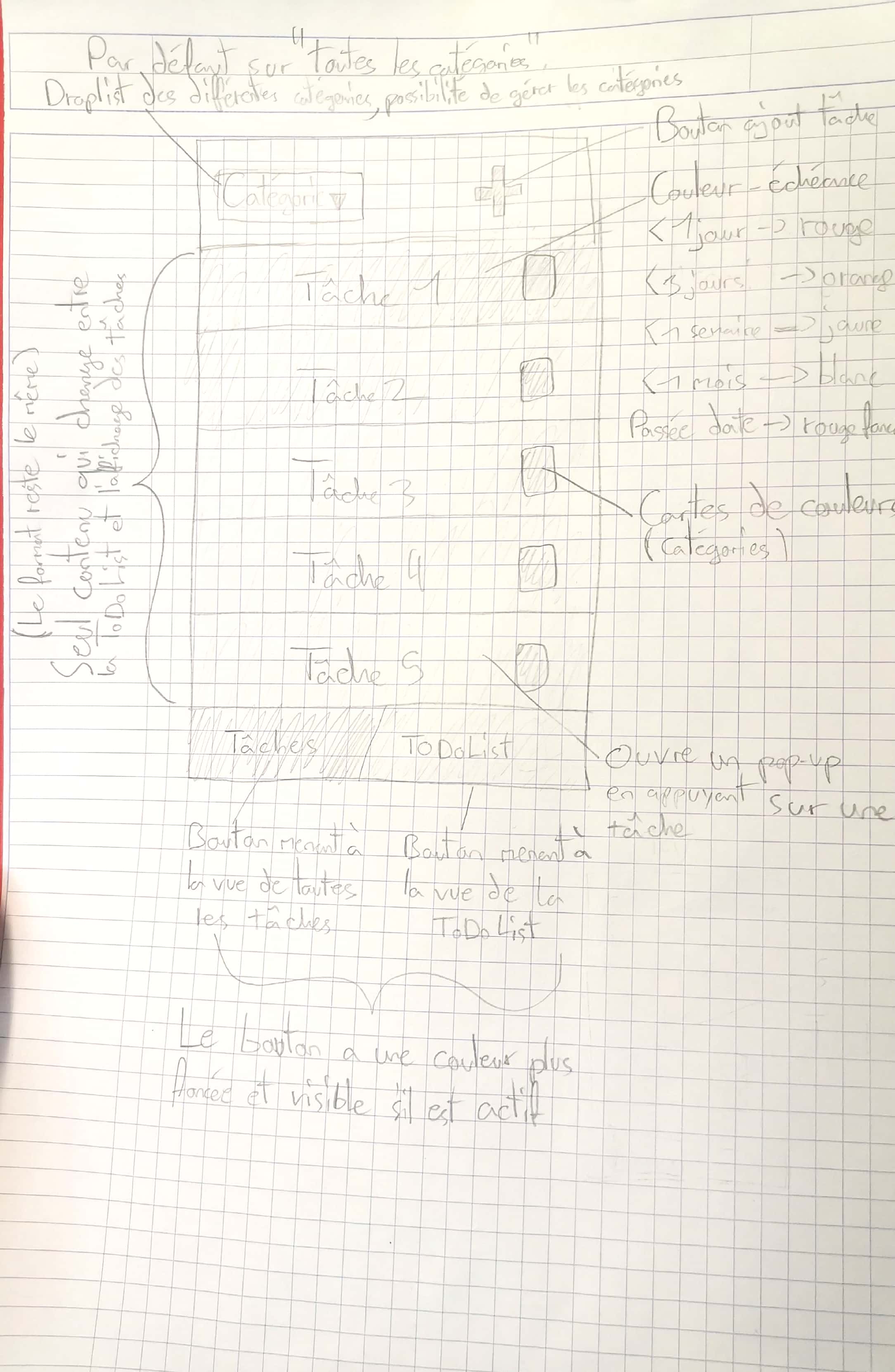
## Maquettes & Structure de l’application

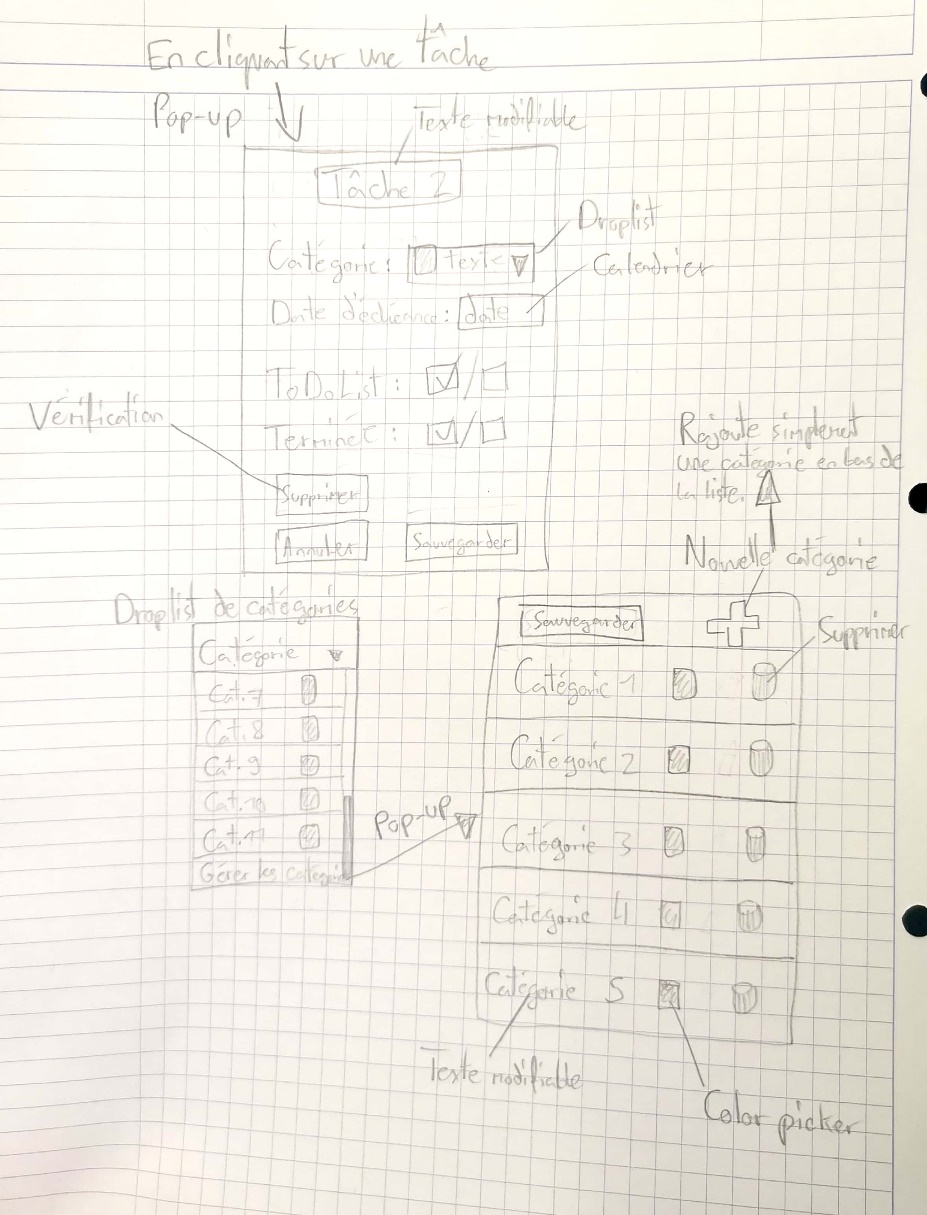
Au début du projet, une maquette « Tout en un » était le modèle de base sur lequel je partais. L’utilisateur pouvait accéder à la modification, suppression, ajout, tri et ajout à la ToDoList des tâches sur une seule même page.



Cependant, cette solution me paraissait trop compacte et pourrait engendrer des problèmes dans le cas où l’on rajoute des informations ou des outils, qui n’auraient plus assez de place ou qui transformeraient un conteneur de tâche en menu de tâche, ajoutant trop de redondance dans le programme.

J’ai finalement opté pour une solution plus aérée dans laquelle on a juste le nom de la tâche et sa catégorie. En appuyant dessus, on peut accéder à un menu différent permettant d’en observer les détails et l’éditer. Depuis la page principale des tâches, on peut appuyer sur le « + » pour ajouter une tâche à la liste, choisir la catégorie à afficher en appuyant sur la droplist « Catégorie », cliquer sur une tâche pour accéder à son menu. La navigation entre la page des tâches et la page de ToDoList se fait par le biais d’un menu de navigation fixé au bas de l’application comprenant deux boutons liés à ces deux pages.







La navigation se fait via un menu statique tout au bas de l’application (voir figure x). Il n’y a que deux pages principales, toutes les autres étant sous forme de pop-ups. La navigation des pop-ups se fait comme s’ils étaient des pages, c’est-à-dire avec les flèches de navigation de base du téléphone.

## Infographie de la date d’échéance

Les tâches affichées dans la liste des suivent un code de couleurs correspondant au temps restant avant la date d’échéance :

* > 1 mois 🡪 Blanc
* < 1 mois 🡪 Doré – Verger Clair Jaune (LightGoldenrodYellow)
* < 1 semaine 🡪 Doré - Verger (Goldenrod)
* < 3 jours 🡪 Orange
* < 1 jour 🡪 Orange-Rouge
* Passée date 🡪 Rouge

Les couleurs ne sont pas appliquées dans la ToDoList, étant donné que les tâches sont toutes à accomplir le jour même.

Il faut noter que si la tâche est cochée comme terminée, sa date de péremption est égale à DateTime.MaxValue, c’est-à-dire l’année 2100. Cela la propulse tout en bas de la liste et sa couleur devient Vert-Jaune.

## Navigation

La navigation principale se fait avec les deux boutons du menu. L’utilisateur peut ensuite naviguer à travers l’application en appuyant sur une tâche dans la liste de tâche pour accéder à son menu, en appuyant sur une catégorie dans le menu de catégories pour y accéder, en appuyant sur le « + » pour accéder au menu d’ajout de tâches, etc…

Le retour en arrière se fait avec les flèches de navigation de base du téléphone.

Toutes les pages sauf les deux principales – La page des tâches et la ToDoList – sont sous forme de pop-ups.

## Conception des tests

Les tests manuels consistent en cette liste :

1. La navigation entre l’onglet des tâches et la ToDoList est fonctionnelle
2. Les tâches sont triées par ordre croissant du temps restant avant leur date d’échéance
3. Les tâches suivent le code de couleur définit plus tôt correspondant au temps restant avant leur date d’échéance.
4. L’ajout de tâche fonctionne et la tâche est directement triée avec les autres par date d’échéance
5. La suppression de tâche fonctionne, la tâche disparaît de la liste
6. Lorsqu’une tâche est cochée comme terminée, elle n’apparaît plus dans la liste
7. Le menu de tâche s’ouvre comme pop-up en appuyant sur une tâche
8. Toutes les modifications sauvegardées dans le menu de tâche sont entrées dans la base de données
9. Après avoir ajouté un objet à la ToDoList, il est visible dans celle-ci
10. Après avoir supprimé un objet de la ToDoList, il n’est plus visible dans celle-ci
11. Le menu des catégories s’ouvre comme pop-up en cliquant sur « Gérer les catégories » dans la droplist de catégorie.
12. La suppression, modification et ajout de catégories sont enregistrés dans la base de données et affichées dans la droplist de catégorie lors d’un appui sur le bouton « sauvegarder »

## Planification détaillée

La planification initiale a été réalisée le Mercredi 4 Mai, soit après environ 1 journée de travail

# Réalisation

## Dossier de Réalisation

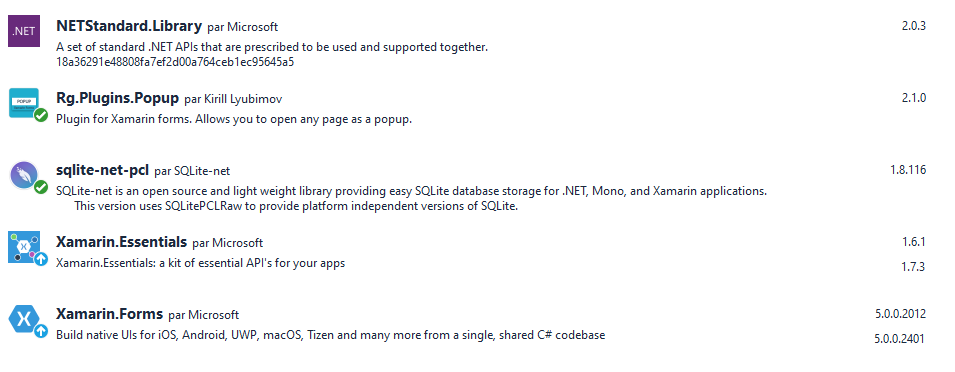
### Programmes & versions utilisés

**Code** : Visual Studio 2019

**Planification** : Excel 2016

**Schémas** : Visio 2016

**NuGet** :



**Rg.Plugins.Popup** permet de d’afficher des Pop-Ups et d’assurer leur bon fonctionnement.

**Sqlite-net-pcl** est nécessaire sur Xamarin pour utiliser SQLite.

**Téléphone & Version d’Android :**

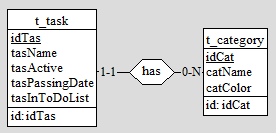


**Spécifications de l’ordinateur :**

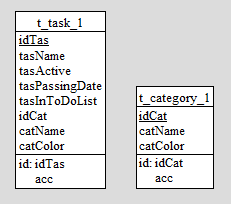
* Nom : INF-N501-M804
* Processeur : Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz 3.41 GHz
* Mémoire RAM installée : 32,0 Go
* Type de système : Système d’exploitation 64 bits, processeur x64
* OS : Windows 10 Éducation, Version 20H2

### Base de données

Les modèles de base de données sont très simples. Ils consistent en deux tables, la table des tâches et la table des catégories.



À cause d’un problème dans le code pour la connexion entre les deux tables et du manque de temps pour le régler, les deux tables ont été séparées, les attributs de la table des catégories étant chargés automatiquement dans celle des tâches au lancement et à chaque modification de la table des catégories.



Cette modification ne change rien au fonctionnement du programme, si ce n’est le rendre un peu plus lent.

### MVVM & Code

La structure du code est en MVVM (Model / View / ViewModel), ce qui signifie que ce que l’utilisateur voit (View) est séparé mais connecté à un ViewModel. Le Model est un modèle de données (expliqué plus bas).

La View, vue en français est tout ce que l’utilisateur peut voir. Elle est codée en XAML (C# en arrière), et est liée au ViewModel avec plusieurs paramètres, notamment le terme « Binding » qui lie une valeur affichée avec une valeur du ViewModel ou une commande qui exécute du code.

Les ViewModels contiennent des méthodes qui traitent tout ce qui est dans le programme. Ils sont en lien avec le Model, la View et les Services.

Les Services sont en dehors de MVVM, ils peuvent faire office de classe statique, de variables globales. Dans mon programme, ils assurent la connexion avec la base de donnée

Les Modèles représentent la base de données. Pour la base de données contenant t\_task et t\_category, il y a un modèle TaskModel et un modèle CategoryModel.

## Compilation

Afin de compiler l’application sur le téléphone, il faut avant tout mettre le téléphone en mode développeur. Cela se fait en suivant ces étapes :

1. Ouvrir les paramètres du téléphone
2. Aller dans « À propos du téléphone »
3. Trouver le numéro de build
4. Appuyer sur le numéro de build jusqu’à ce qu’une notification avertisse que le mode développeur est activé.

Il faut ensuite le connecter à l’ordinateur contenant le code et il se connectera automatiquement à Visual Studio si l’environnement de code Xamarin est ouvert.

Une fois cela fait, l’application peut être lancée depuis l’ordinateur et se lancera sur le téléphone.

Afin de débugger l’application, tout ce qu’il y a à faire est de mettre un break point et de lancer le débugger. L’application sur téléphone se figera et l’on pourra débugger le code.

### Réalisation des tests

Les tests n’ont pas été réalisés d’une traite. Ils ont été réalisés au cours du projet, car chaque test prédéfini fait d’une manière ou d’une autre partie des objectifs du projet. Tous les tests sont fonctionnels, sauf le 11ème : *« Le menu des catégories s’ouvre comme pop-up en cliquant sur « Gérer les catégories » dans la droplist de catégorie*». Le moyen d’ouvrir le menu des catégories a légèrement changé ; il s’agit d’un bouton, au lieu d’un choix dans une droplist comme prévu. Cela n’a aucun impact sur le bon fonctionnement du programme.

# Conclusion

Cette rubrique contient les différentes conclusions des différents thèmes de ce projet et ce que l’on peut en tirer.

## Bilan des fonctionnalités demandées

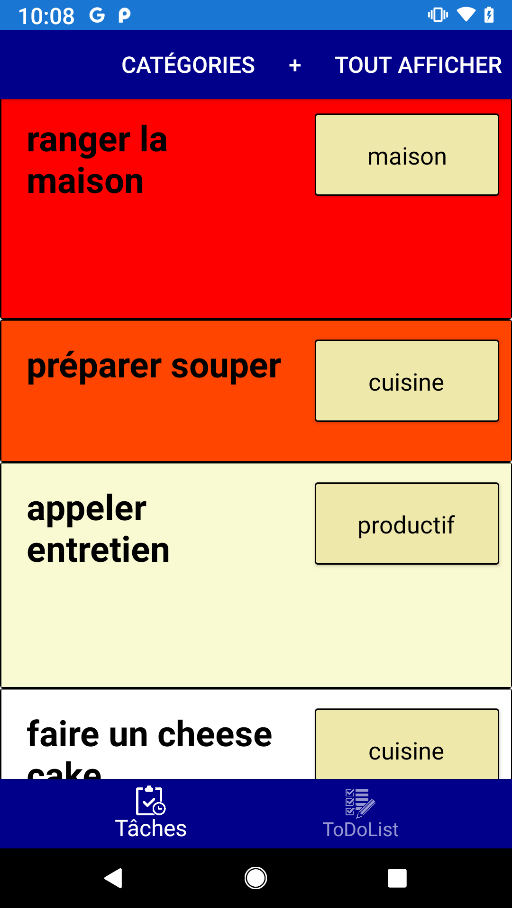
|  |  |
| --- | --- |
| Il doit être possible de catégoriser les tâches | OK |
| l doit être possible d’afficher les tâches par catégorie. | OK |
| Il doit être possible d’afficher les tâches par ordre de date d’échéance. | OK |
| Il doit être possible d’insérer, de modifier ou de supprimer une tâche. | OK |
| Il doit être possible de marquer une tâche comme terminée. A ce moment elle ne doit plus être visible. | OK |
| Une infographie par date d’échéance. Par exemple différentes couleurs par proximité de l’échéance. | OK |
| Il doit être possible de créer des « ToDo list » à partir des tâches. Par exemple si l’utilisateur veut choisir les tâches qu’il veut réaliser dans la journée. | OK |

## Points techniques évalués spécifiques au projet

|  |  |
| --- | --- |
| L’ergonomie de l’application (user friendly et simple), la facilité d’utilisation. | OK |
| La modélisation de la base de données (MCD, MLD et MPD). | OK |
| La pertinence de l’infographie proposée. | OK |
| Le bon fonctionnement des créations et modification des tâches. | OK |
| Le bon fonctionnement des affichages des tâches. | OK |
| Le bon fonctionnement de création d’une « ToDo list ». | OK |
| La qualité et lisibilité du code source (respect des normes ETML). | OK |

## Résultat visuel

Le design n’est pas très élaboré, mais il est satisfaisant et facile à comprendre.

Sur la page des tâches, on peut voit les différentes tâches, triées par date d’échéance. Les couleurs suivent le code couleur définit plus haut :

> 1 mois 🡪 Blanc

< 1 mois 🡪 Doré – Verger Clair Jaune (LightGoldenrodYellow)

< 1 semaine 🡪 Doré - Verger (Goldenrod)

< 3 jours 🡪 Orange

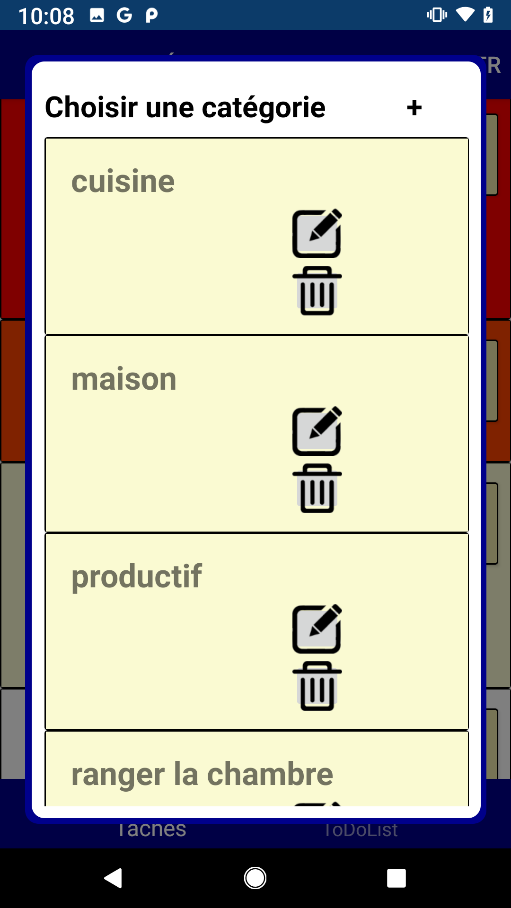
< 1 jour 🡪 Orange-Rouge

Passée date 🡪 Rouge

Le nom de leur catégorie est affiché dans une box à droite de la tâche.

Depuis cette page, l’utilisateur peut ajouter une tâche en cliquant sur le « + » en tête de page, ouvrir le menu des catégories en cliquant sur « Catégories », afficher toutes les tâches en cliquant sur « Tout afficher », ouvrir le menu d’une tâche en cliquant sur la tâche en question.

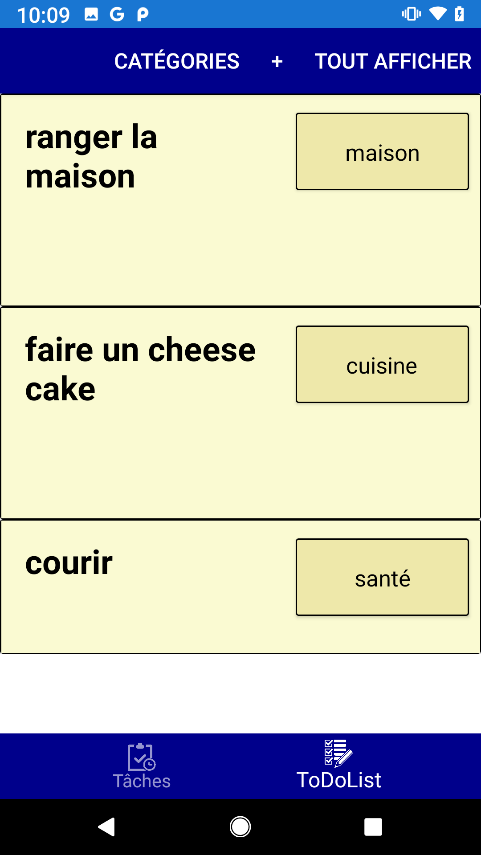
L’utilisateur a également la possibilité de passer sur la ToDoList en cliquant sur le bouton « ToDoList » dans le menu du bas.

Dans le menu des catégories, apparaissant comme pop-up, l’utilisateur peut sélectionner une catégorie à afficher en cliquant dessus, il peut modifier la catégorie en cliquant sur le bouton de modification et la supprimer en cliquant sur le bouton de suppression.

Il peut ajouter une nouvelle catégorie, ce qui ouvre le même pop-up que lors de la modification.



Lorsque l’utilisateur clique sur une catégorie dans le menu des catégories, seules les tâches de cette certaine catégorie s’affichent. Ce filtre peut être enlevé en cliquant sur « Tout afficher ».



En cliquant sur une tâche, ou en cliquant sur le bouton de création de tâche, l’utilisateur peut modifier une tâche comme bon lui semble. Il peut ensuite sauvegarder la tâche et celle-ci sera modifiée dans la base de données.

Le menu d’ajout de tâche est le même, mais sans le bouton de suppression.

Une tâche ne peut pas être terminée et dans la ToDoList en même temps.

Enfin, la ToDoList est similaire à la liste des tâches, mais sans l’infographie de couleurs pour des raisons citées plus tôt.

Le tri est également possible dans la ToDoList, mais indépendant de la liste des tâches.

## Bilan de la planification

Voici la planification et journalisation détaillée du projet.

|  |  |  |
| --- | --- | --- |
| **Tâches - objectifs** |  | **Tranche 5min** |
|  |
| Absences - Imprévus | 0h | 0 |
| 0h | 0 |
| Doc - Planification initiale | 1h30min | 18 |
| 1h30min | 18 |
| Doc - Journal de travail | 3h50min | 46 |
| 3h40min | 44 |
| Doc - Rapport | 17h10min | 206 |
| 13h | 156 |
| Doc - Maquettes | 1h40min | 20 |
| 2h5min | 25 |
| Doc - Entretien avec les experts / chef de projet | 3h55min | 47 |
| 3h45min | 45 |
| Doc - Base de données | 2h30min | 30 |
| 0h50min | 10 |
| Doc - Analyse | 1h30min | 18 |
| 1h30min | 18 |
| Doc - Général | 3h45min | 45 |
| 4h35min | 55 |
| Dev - Page des tâches | 7h40min | 92 |
| 7h5min | 85 |
| Dev - ToDoList | 9h | 108 |
| 0h30min | 6 |
| Dev - Général | 10h20min | 124 |
| 31h50min | 382 |
| Dev - Interfaces | 10h45min | 129 |
| 8h | 96 |
| Dev - Github | 1h25min | 17 |
| 1h | 12 |
| Dev - Tests | 3h45min | 45 |
| 0h | 0 |
| Dev - Base de données | 5h15min | 63 |
| 4h40min | 56 |
| **Total planifié** | | **1008** |
| *84h* |
| **Total réalisé** | | **1008** |
| *84h* |

Bien que la plupart du temps réparti pour les tâches soit respecté, on peut observer une forte différence entre la planification et la réalisation dans la tâche Dev – Général. Ceci est dû principalement à la réalisation de la ToDoList, qui était en tous points similaire à la page des tâches. Une fois cette page réalisée, la ToDoList pouvait être faite en quelques minutes.

Beaucoup de développement a été nécessaire pour l’implémentation des pop-ups, qui – même si utilisés dans la page des tâches – étaient notés comme Dev – Général, car considérés comme à part.

Le temps attribué aux tests est nul, contrairement à celui attribué dans la planification. Ceci est expliqué plus haut : les tests approfondis ont été jugés non-nécessaires. Des tests manuels ont été réalisés tout au cours du projet à la place, mais ceux-ci vont plus dans le Dev – Général que dans Dev – Tests.

## Bilan personnel

# Webographie

La webographie contient tous les liens des sites qui m’ont apporté des réponses à mes questions ou qui m’ont aidé. Les liens consultés durant une recherche qui n’ont pas été utiles sont laissés de côté, faute de pertinence.

[Comprenez les différents moyens de stocker des données sur Android - Gérez vos données localement pour avoir une application 100 % hors-ligne - OpenClassrooms](https://openclassrooms.com/fr/courses/4568746-gerez-vos-donnees-localement-pour-avoir-une-application-100-hors-ligne/4568753-comprenez-les-differents-moyens-de-stocker-des-donnees-sur-android)

[Testing Xamarin Applications with Visual Studio App Center - App Center Blog (microsoft.com)](https://devblogs.microsoft.com/appcenter/testing-xamarin-applications-with-visual-studio-app-center/)

[Getting started · rotorgames/Rg.Plugins.Popup Wiki (github.com)](https://github.com/rotorgames/Rg.Plugins.Popup/wiki/Getting-started)

[PopupPage · rotorgames/Rg.Plugins.Popup Wiki (github.com)](https://github.com/rotorgames/Rg.Plugins.Popup/wiki/PopupPage)

[Using SQLite.NET with Android - Xamarin | Microsoft Docs](https://docs.microsoft.com/en-us/xamarin/android/data-cloud/data-access/using-sqlite-orm)

[c# - Adding a button to the title bar Xamarin Forms - Stack Overflow](https://stackoverflow.com/questions/39384263/adding-a-button-to-the-title-bar-xamarin-forms)

[Rg Popup In Xamarin.Forms Using Fresh MVVM (c-sharpcorner.com)](https://www.c-sharpcorner.com/article/rg-popup-in-xamarin-forms-using-fresh-mvvm/)

# Table des illustrations

**Aucune entrée de table d'illustration n'a été trouvée.**

# Annexes

Ici sont les annexes de ce projet : la planification et le journal de travail qui m’ont accompagné tout au long de ce projet.

## Planification

Voici la planification détaillée du projet, réalisée le Mercredi 4 Mai, soit après environ 1 journée de travail.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Séquence** | **26** |  | **Date:** | **mercredi, 1 juin 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Général | 45 | 3h45min | Finalisation du projet |  | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **25** |  | **Date:** | **lundi, 30 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Journal de travail | 2 | 0h10min | Rédaction du journal de travail |  | |
| Dev - Github | 1 | 0h5min | Utilisation de Github |  | |
| Doc - Rapport | 33 | 2h45min | Rédaction du rapport, notamment de la réalisation et la conclusion |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **24** |  | **Date:** | **mercredi, 25 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail |  | |
| Dev - Github | 1 | 0h5min | Utilisation de Github |  | |
| Doc - Rapport | 27 | 2h15min | Rédaction du rapport, notamment de la réalisation et la conclusion |  | |
| Doc - Entretien avec les experts / chef de projet | 4 | 0h20min | Entretien avec M. Ferrari pour répondre à des questions et vérifier l'avancement du projet |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **23** |  | **Date:** | **mercredi, 25 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Rapport | 45 | 3h45min | Rédaction du rapport, notamment de la réalisation et la conclusion |  | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **22** |  | **Date:** | **lundi, 23 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Journal de travail | 2 | 0h10min | Rédaction du journal de travail |  | |
| Dev - Github | 1 | 0h5min | Utilisation de Github |  | |
| Dev - Interfaces | 33 | 2h45min | Amélioration de l'interface générale de l'application |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **21** |  | **Date:** | **vendredi, 20 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail |  | |
| Dev - Github | 1 | 0h5min | Utilisation de Github |  | |
| Dev - Général | 27 | 2h15min | Développement général, amélioration du code |  | |
| Doc - Entretien avec les experts / chef de projet | 4 | 0h20min | Entretien avec M. Ferrari pour répondre à des questions et vérifier l'avancement du projet |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **20** |  | **Date:** | **vendredi, 20 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Général | 36 | 3h | Développement général, amélioration du code |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **19** |  | **Date:** | **jeudi, 19 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail |  | |
| Dev - Github | 1 | 0h5min | Utilisation de Github |  | |
| Dev - Général | 27 | 2h15min | Correction des potentielles erreurs liées aux tests |  | |
| Doc - Entretien avec les experts / chef de projet | 4 | 0h20min | Entretien avec M. Ferrari pour répondre à des questions et vérifier l'avancement du projet |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **18** |  | **Date:** | **jeudi, 19 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Tests | 45 | 3h45min | Test des différentes fonctionnalités suivant les tests écrits auparavant dans le rapport |  | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **17** |  | **Date:** | **mercredi, 18 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail |  | |
| Dev - Github | 1 | 0h5min | Utilisation de Github |  | |
| Dev - ToDoList | 12 | 1h | Finalisation de la réalisation de la page de ToDoList |  | |
| Dev - Interfaces | 10 | 0h50min | Finalisation de la réalisation de l'interface de la page de ToDoList |  | |
| Doc - Rapport | 5 | 0h25min | Rédaction de la partie du rapport concernant la réalisation de la page de ToDoList |  | |
| Doc - Entretien avec les experts / chef de projet | 4 | 0h20min | Entretien avec M. Ferrari pour répondre à des questions et vérifier l'avancement du projet |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **16** |  | **Date:** | **mercredi, 18 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - ToDoList | 35 | 2h55min | Réalisation de la page de ToDoList |  | |
| Dev - Interfaces | 10 | 0h50min | Réalisation de l'interface de la page de ToDoList |  | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **15** |  | **Date:** | **lundi, 16 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Journal de travail | 2 | 0h10min | Rédaction du journal de travail |  | |
| Dev - Github | 1 | 0h5min | Utilisation de Github |  | |
| Dev - ToDoList | 23 | 1h55min | Réalisation de la page de ToDoList |  | |
| Dev - Interfaces | 10 | 0h50min | Réalisation de l'interface de la page de ToDoList |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **14** |  | **Date:** | **vendredi, 13 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail |  | |
| Dev - Github | 1 | 0h5min | Utilisation de Github |  | |
| Dev - ToDoList | 12 | 1h | Réalisation de la page de ToDoList |  | |
| Dev - Interfaces | 10 | 0h50min | Réalisation de l'interface de la page de ToDoList |  | |
| Doc - Rapport | 5 | 0h25min | Rédaction de la partie du rapport concernant la réalisation de la page de ToDoList |  | |
| Doc - Entretien avec les experts / chef de projet | 4 | 0h20min | Entretien avec M. Ferrari pour répondre à des questions et vérifier l'avancement du projet |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **13** |  | **Date:** | **vendredi, 13 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - ToDoList | 26 | 2h10min | Réalisation de la page de ToDoList |  | |
| Dev - Interfaces | 10 | 0h50min | Réalisation de l'interface de la page de ToDoList |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **12** |  | **Date:** | **jeudi, 12 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail |  | |
| Dev - Github | 1 | 0h5min | Utilisation de Github |  | |
| Dev - Page des tâches | 10 | 0h50min | Finalisation de la réalisation de la page affichant toutes les tâches |  | |
| Dev - Interfaces | 16 | 1h20min | Finalisation de la réalisation de l'interface de la page affichant toutes les tâches |  | |
| Doc - Rapport | 5 | 0h25min | Rédaction de la partie du rapport concernant la réalisation de la page des tâches |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **11** |  | **Date:** | **jeudi, 12 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Page des tâches | 35 | 2h55min | Réalisation de la page affichant toutes les tâches |  | |
| Dev - Interfaces | 10 | 0h50min | Réalisation de l'interface de la page affichant toutes les tâches |  | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **10** |  | **Date:** | **mercredi, 11 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail |  | |
| Dev - Github | 1 | 0h5min | Utilisation de Github |  | |
| Dev - Page des tâches | 12 | 1h | Réalisation de la page affichant toutes les tâches |  | |
| Dev - Interfaces | 10 | 0h50min | Réalisation de l'interface de la page affichant toutes les tâches |  | |
| Doc - Rapport | 5 | 0h25min | Rédaction de la partie du rapport concernant la réalisation de la page des tâches |  | |
| Doc - Entretien avec les experts / chef de projet | 4 | 0h20min | Entretien avec M. Ferrari pour répondre à des questions et vérifier l'avancement du projet |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **9** |  | **Date:** | **mercredi, 11 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Page des tâches | 35 | 2h55min | Réalisation de la page affichant toutes les tâches |  | |
| Dev - Interfaces | 10 | 0h50min | Réalisation de l'interface de la page affichant toutes les tâches |  | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **8** |  | **Date:** | **lundi, 9 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Base de données | 32 | 2h40min | Implémentation de la base de données dans l'application à l'aide de SQLite |  | |
| Doc - Base de données | 4 | 0h20min | Rédaction du rapport sur le fonctionnement et la structure des bases de données de l'application |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **7** |  | **Date:** | **vendredi, 6 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Base de données | 22 | 1h50min | Apprentissage de SQLite |  | |
| Doc - Entretien avec les experts / chef de projet | 4 | 0h20min | Entretien avec M. Ferrari pour répondre à des questions et vérifier l'avancement du projet |  | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail |  | |
| Dev - Github | 1 | 0h5min | Utilisation de Github |  | |
| Doc - Base de données | 5 | 0h25min | Rédaction du rapport sur le fonctionnement et la structure des bases de données de l'application |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **6** |  | **Date:** | **vendredi, 6 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Base de données | 9 | 0h45min | Apprentissage de SQLite |  | |
| Doc - Rapport | 27 | 2h15min | Rédaction de la partie du rapport sur le fonctionnement et l'implémentation des tests, définition des différents tests à réaliser |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **5** |  | **Date:** | **jeudi, 5 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Général | 20 | 1h40min | Création de la structure de l'application correspondant aux normes ETML et à celles citées dans le rapport, configuration de l'environnement de programmation |  | |
| Doc - Rapport | 11 | 0h55min | Rédaction du rapport, notamment de la réalisation, des choix effectués et des options prises dans le code |  | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail pour la journée |  | |
| Dev - Github | 1 | 0h5min | Utilisation de GitHub |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **4** |  | **Date:** | **jeudi, 5 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Base de données | 21 | 1h45min | Conception de la base de données |  | |
| Dev - Général | 14 | 1h10min | Création de la structure de l'application correspondant aux normes ETML et à celles citées dans le rapport, configuration de l'environnement de programmation |  | |
| Doc - Maquettes | 10 | 0h50min | Création des maquettes finales |  | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **3** |  | **Date:** | **mercredi, 4 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Github | 5 | 0h25min | Mise en place d'un répertoire GitHub pour la sauvegarde du projet |  | |
| Doc - Rapport | 10 | 0h50min | Rédaction de la partie de conception du rapport |  | |
| Doc - Entretien avec les experts / chef de projet | 4 | 0h20min | Entretien avec M. Ferrari pour répondre à des questions et vérifier l'avancement du projet |  | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail |  | |
| Doc - Planification initiale | 13 | 1h5min | Fin de la rédaction de la planification initiale |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **2** |  | **Date:** | **mercredi, 4 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Analyse | 18 | 1h30min | Analyse et recherches nécessaires |  | |
| Doc - Maquettes | 10 | 0h50min | Conception des maquettes, de différents choix possibles |  | |
| Doc - Rapport | 12 | 1h | Rédaction de la partie d'analyse et début de la partie de conception |  | |
| Doc - Planification initiale | 5 | 0h25min | Début de la rédaction de la planification initiale |  | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **1** |  | **Date:** | **lundi, 2 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Entretien avec les experts / chef de projet | 15 | 1h15min | Lecture du cahier des charges avec le chef de projet et un des experts, questionnement et compréhension de celui-ci et que du déroulement du projet en général. |  | |
| Doc - Rapport | 21 | 1h45min | Rédaction de la première partie du rapport |  | |
| Total tranche | 36 | 3h |  |  | |

## Journal de travail

Voici le journal de travail, rempli à chaque fin de journée.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Séquence** | **26** |  | **Date:** | **mercredi, 1 juin 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Général | 45 | 3h45min | Finalisation du projet |  | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **25** |  | **Date:** | **lundi, 30 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Rapport | 26 | 2h10min | Rédaction du rapport |  | |
| Doc - Général | 10 | 0h50min | Finalisation du projet |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **24** |  | **Date:** | **mercredi, 25 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail |  | |
| Doc - Entretien avec les experts / chef de projet | 4 | 0h20min | Test de l'application par M. Ferrari, questions pour la suite, … |  | |
| Doc - Rapport | 26 | 2h10min | Réalisation du rapport, notamment de la réalisation |  | |
| Dev - Github | 2 | 0h10min | Utilisation de Github |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **23** |  | **Date:** | **mercredi, 25 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Général | 15 | 1h15min | Correction des derniers problèmes, notamment la suite de cette question de refresh, mais pour charger les catégories dans le Picker. P.ex si on voulait créer une catégorie, on ne pouvait ensuite pas l'utiliser dans le Picker de catégories d'une nouvelle tâche sans redémarrer l'application. Ajout du système de couleurs dans la liste des tâches selon l'échéance et finalisation de l'affichage des catégories par nom. Correction des problèmes trouvés lors des tests. |  | |
| Dev - Interfaces | 20 | 1h40min | Changement des interfaces de l'application, ajout de Grids, de Frames, |  | |
| Doc - Rapport | 10 | 0h50min | Rédaction du rapport, notamment sur la réalisation |  | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **22** |  | **Date:** | **lundi, 23 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Général | 36 | 3h | J'ai essayé de récupérer le nom de la catégorie dans un label. Cela paraît simple, mais le label est dans une liste auto-générée, basée sur la table des tâches. Il est donc possible de récupérer l'id de la catégorie, mais pour accéder au nom ou de la couleur de la catégorie, il faut exécuter du code et retourner quelquechose. Je ne savais pas comment faire, après de nombreuses recherches, et au vu du temps restant, apprendre à faire un lien entre Binding, la table des tâches et la table des catégories était trop à risque. J'ai donc à regret fait quelque chose de terrible : mettre le nom de la catégorie et sa couleur dans la table des tâches, tout en gardant la table des catégories à l'extérieur. À chaque fois que la page se refresh, on charge le nom et la couleur d'une catégorie dans la tâche liée. |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **21** |  | **Date:** | **vendredi, 20 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail journalier |  | |
| Dev - Github | 1 | 0h5min | Utilisation de Github |  | |
| Doc - Entretien avec les experts / chef de projet | 4 | 0h20min | Entretien avec M. Ferrari à propos de l'avancement du projet : La priorité est de régler les bugs afin d'avoir un maximum de tâches fonctionnelles par rapport au cahier des charges. Le design et les ajouts sont à faire si le temps le permet. |  | |
| Dev - Général | 21 | 1h45min | Implémentation de la modification des catégories, du filtre par catégories, d'un système de réinitialisation des filtres, correction de problèmes liés à la catégorie, implémentation d'un rafraichissement des pages en passant une valeur dans le constructeur. |  | |
| Dev - ToDoList | 6 | 0h30min | Copie de la page des tâches pour la ToDoList. Suppression du ViewModel de la ToDoList et réutilisation du ViewModel de la page des tâches, car il était plus approprié de passe une seule variable définissant si l'on est dans la ToDoList que de tout recréer. |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **20** |  | **Date:** | **vendredi, 20 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Général | 36 | 3h | Je voulais rendre une catégorie modifiable, mais pour cela il faut transformer un label en entry dans laquelle l'utilisateur peut écrire, et ensuite modifier la catégorie, etc… Après avoir perdu trop de temps en essayant d'implémenter un custom control, un label qui se modifie en cliquant dessus, j'ai opté pour une solution plus simple : l'utilisateur peut uniquement créer ou supprimer des catégories. À la création d'une catégorie, un nouveau pop-up s'ouvre et demande le nom et la couleur de la catégorie, qui va s'ajouter dans la liste des catégories. Il y aura même la possibilité de modifier une catégorie via ce menu. | <https://doumer.me/building-an-edit-label-custom-control-in-xamarin-forms/> | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **19** |  | **Date:** | **jeudi, 19 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Page des tâches | 20 | 1h40min | Implémentation de la supression et de l'ajout de catégories et des connexions entre les views et les viewmodels (p.ex passer un paramètre via une commande), rien ne s'affichait par défaut en ouvrant le menu d'une tâche, il était connecté au mauvais ViewModel. Toujours un problème pour afficher le nom de la catégorie choisie dans le picker par défaut | <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/refreshview> | |
| Doc - Journal de travail | 5 | 0h25min | Rédaction du journal de travail journalier | <https://stackoverflow.com/questions/60472467/how-to-show-picker-choose-before-button-command-in-mvvm-xamarin> | |
| Dev - Github | 1 | 0h5min | Utilisation de Github |  | |
| Dev - Interfaces | 10 | 0h50min | Mise en page des pop-ups et des catégories, des boutons sur la liste des catégories |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **18** |  | **Date:** | **jeudi, 19 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Général | 25 | 2h5min | Utilisation et implémentation d'une alternative de liste fonctionnant apparemment avec les pop-ups : SfListView de Syncfusion. Après avoir suivi les étapes, le pop-up n'affiche toujours rien. Au démarrage de l'application, un message notifie que j'utilise une version d'essai de Syncfusion et qu'il faut une lisence pour ne pas avoir ce message. J'abandonne donc l'idée de syncfusion. Débuggage du programme : Lors de l'appui sur le bouton des catégories, il y a bien une connexion avec le ViewModel ; le constructeur s'exécute, mais jamais la commande de chargement des catégories n'est exécutée. Le problème est finalement résolu : Un bout de code que je croyais inutile (base.OnAppearing et \_viewModel.OnAppearing --> IsBusy = true) était en fait indispensable à cette liste. Il ne l'était pas aux autres pop-ups. Transformation de toutes les listes en ObservableCollection, afin d'avoir ces listes constamment à jour quand modifiées. Essai de le faire directement dans le service "Database", mais un problème survient, vite réglé : Les ObservableCollection ne supportent pas l'async. | <https://github.com/SyncfusionExamples/rgplugin-popup-listview-xamarin> | |
| Dev - Interfaces | 20 | 1h40min | Modifications de l'interface allant avec le code. Réutilisation du design de popup trouvé sur le premier lien GitHub | <https://www.syncfusion.com/kb/11353/how-to-show-xamarin-forms-listview-sflistview-in-popup-using-rg-plugin-popup-framework> | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **17** |  | **Date:** | **mercredi, 18 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Entretien avec les experts / chef de projet | 4 | 0h20min | Entretien avec M. Ferrari : Considération d'une tâche pouvant avoir plusieurs catégories, du changement de "tâche active" en "tâche terminée" et d'une confirmation de suppression | <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/refreshview> | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail journalier | <https://stackoverflow.com/questions/56239550/xamarin-forms-default-value-for-picker> | |
| Dev - Github | 1 | 0h5min | Utilisation de Github | <https://codemilltech.com/xamarin-forms-bindable-picker/#:~:text=The%20ItemDisplayBinding%20property%20is%20of%20a%20BindingBase%20type%2C,determines%20what%20actually%20gets%20displayed%20within%20the%20Picker.> | |
| Dev - Général | 27 | 2h15min | Menu des tâches - Étant affichée en tant que Pop-up, la liste des catégories ne fonctionne pas de la même manière que la liste des tâches. À l'ouverture, rien ne s'affiche. Expérimentations avec xaml (alternatives à la RefreshView, ajout de stacklayout et autres composants afin de trouver ce qui ne va pas) |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **16** |  | **Date:** | **mercredi, 18 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Général | 45 | 3h45min | Insertion de catégories dans la base de données, implémentation du menu de catégories et des pickers servant à sélectionner les catégories. C'était assez compliqué car je travaille en MVVM, donc pour remplir le Picker, qu'on remplit normalement avec des Picker.Items.Add, je dois utiliser des Bindings avec le ViewModel. J'ai mis longtemps à comprendre que le système de Binding était orienté objet, ou quelque chose de similaire. On peut donc faire une commande de la sorte : {Binding SelectedCategory.Name}. | <https://www.c-sharpcorner.com/article/populate-picker-using-mvvm/> | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **15** |  | **Date:** | **lundi, 16 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Entretien avec les experts / chef de projet | 5 | 0h25min | Entretien avec M. Wenger, un de mes deux experts pour la seconde visite | <https://www.c-sharpcorner.com/article/toolbar-item-in-xamarin-cross-platform-with-toolbar-order-example/> | |
| Dev - Général | 30 | 2h30min | Après confirmation, il est impossible d'ajouter un picker dans le menu du haut en Xamarin. Il faut donc revoir comment l'application fonctionnera. Au lieu d'une liste déroulante de catégories, il y aura un bouton de filtre dans le menu ouvrant un pop-up, menu des catégories, dans lequel on peut choisir la catégorie à afficher et ajouter, supprimer ou modifier des catégories. Début de création de la page. | <https://social.msdn.microsoft.com/Forums/en-US/47077ec2-6fea-479d-8737-7dfe809f7da5/how-to-create-a-custom-layout-of-toolbar-with-xamarin-forms> | |
| Doc - Journal de travail | 1 | 0h5min | Rédaction du journal de travail de la journée | <https://stackoverflow.com/questions/38988108/how-to-add-picker-in-toolbar-in-xamarin-forms> | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **14** |  | **Date:** | **vendredi, 13 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Général | 20 | 1h40min | Recherches sur la toolBar de Xamarin Forms, ajouts au design et modifications sur les pop-ups | <https://www.xamarinhelp.com/xamarin-forms-toolbar/> | |
| Doc - Rapport | 10 | 0h50min | Rédaction du rapport | <https://www.c-sharpcorner.com/article/toolbar-item-in-xamarin-cross-platform-with-toolbar-order-example/> | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail de la journée |  | |
| Dev - Github | 2 | 0h10min | Utilisation de Github |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **13** |  | **Date:** | **vendredi, 13 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Page des tâches | 22 | 1h50min | L'utilisateur peut désormais afficher les tâches, les modifier, les supprimer et en ajouter. Il peut ajouter ou retirer une tâche à la ToDoList, qui affichera uniquement les tâches qui sont dedans. Début de réalisation du système de catégories. Problème : on ne peut pas implémenter de Picker (ou dropdown list) dans le menu de base. Il faut soit le créer customisé, nécessitant l'installation de NuGets et de code extérieur et long à comprendre, soit utiliser une alternative au Picker. |  | |
| Dev - Interfaces | 14 | 1h10min | Modification du menu implémenté de base dans le programme, de la couleur, la taille et l'affichage des tâches et des pop-ups |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **12** |  | **Date:** | **jeudi, 12 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Page des tâches | 18 | 1h30min | Implémentation de la modification de tâche. Pour l'instant, tout se fait dans le code, lorsque l'utilisateur appuie sur sauvegarder, sans prendre en compte les informations rentrées par l'utilisateur. Problème avec la liaison entre le ViewModel et la View pour passer en paramètre des données | <https://docs.microsoft.com/en-us/answers/questions/243785/create-a-list-of-dropdown-lists-in-xamarinformsxam.html> | |
| Dev - Interfaces | 12 | 1h | Réalisation des interfaces des pop-ups de la page des tâches, ajout des différents outils nécessaire à la modification d'une tâche (datepicker, droplist, entry, ...) | <https://stackoverflow.com/questions/55414973/how-to-bind-entry-from-page-with-viewmodel> | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail de la journée | <https://help.syncfusion.com/xamarin/listview/sorting> | |
| Dev - Github | 2 | 0h10min | Utilisation de github pour le push journalier |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **11** |  | **Date:** | **jeudi, 12 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Page des tâches | 25 | 2h5min | Réalisation du pop-up d'ajout de tâche et du menu des tâches. Ajout de tâches réussi, en connexion avec la base de données. Le menu des tâches est exactement le même pop-up que le menu d'ajout, mais avec les informations actuelles de la tâche sélectionnée, tandis que celle des nouvelles tâches est vide. L'utilisateur peut modifier ce qu'il veut de la tâche. Recherches en lien du tri, et de comment le réaliser, mais aucune solution acceptable pour l'instant (installer un add-on ou changer les IDs, mauvaise idée) | <https://docs.microsoft.com/en-us/answers/questions/243785/create-a-list-of-dropdown-lists-in-xamarinformsxam.html> | |
| Dev - Interfaces | 20 | 1h40min | Réalisation des interfaces des pop-ups de la page des tâches |  | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **10** |  | **Date:** | **mercredi, 11 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Général | 27 | 2h15min | Restructuration du code qui s'est éparpillé dans des Views dans les ViewModels, débloquage du projet : ajout d'un bouton dans le menu de l'application --> ouverture du pop-up, réussite de l'ouverture d'un pop-up. Expérimentations sur les pop-ups, qui marchent sur les deux différentes pages | <https://www.c-sharpcorner.com/article/rg-popup-in-xamarin-forms-using-fresh-mvvm/> | |
| Doc - Entretien avec les experts / chef de projet | 4 | 0h20min | Entretien avec M. Ferrari afin de parler de l'avancement du projet : | <https://github.com/rotorgames/Rg.Plugins.Popup/wiki/Getting-started> | |
| Dev - Github | 1 | 0h5min | Utilisation de github pour le push journalier | <https://github.com/rotorgames/Rg.Plugins.Popup/wiki/PopupPage> | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail de la journée |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **9** |  | **Date:** | **mercredi, 11 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Général | 45 | 3h45min | Code de la struture générale afin de pouvoir ajouter, supprimer ou modifier une tâche. Problème avec les types de pages, l'utilisation de méthodes dans la sous-page de xaml et dans le ViewModel. Décision d'implémenter les pop-ups plus tard, car jugés trop compliqués à mettre en place, et pourraient trop retarder le projet. Pour l'instant, le fonctionnement de l'application sera une priorité. Beaucoup de recherches pour SQLite sont difficiles à implémenter dans le modèle MVVC, je dois comprendre tout le code et le modifier avant de l'implémenter. | <https://stackoverflow.com/questions/39384263/adding-a-button-to-the-title-bar-xamarin-forms> | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **8** |  | **Date:** | **lundi, 9 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Base de données | 36 | 3h | Implémentation de la base de données SQLite dans l'application. Changement de la structure du code : Une interface était mise en place, mais elle n'acceptait qu'un type par utilisation de l'interface. Une base de données devait être faite, consistant de multiples types. L'interface a donc été supprimée. | <https://docs.microsoft.com/en-us/xamarin/get-started/quickstarts/database?pivots=windows> | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **7** |  | **Date:** | **vendredi, 6 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Base de données | 20 | 1h40min | Modélisation du MCD, MLD, MPD de la base de données, Utilisation de SQLiteStudio pour créer le MPD et remplir la base de données de données. Découverte de SQLite dans Xamarin. Utilisation du NuGet sqlite-net-pcl, | <https://docs.microsoft.com/en-us/xamarin/android/data-cloud/data-access/using-sqlite-orm> | |
| Doc - Journal de travail | 4 | 0h20min | Rédaction du journal de travail de la journée |  | |
| Doc - Entretien avec les experts / chef de projet | 4 | 0h20min | Entretien avec M. Ferrari : Clarifications sur les maquettes, Discussion à propos de SQLite et de son fonctionnement |  | |
| Doc - Rapport | 7 | 0h35min | Rédaction du rapport, notamment sur les maquettes, le fonctionnement de l'application, la navigation et l'analyse & conception des tests |  | |
| Dev - Github | 1 | 0h5min | Utilisation de github pour le push journalier |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **6** |  | **Date:** | **vendredi, 6 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Maquettes | 10 | 0h50min | Réflexions et ajouts sur les maquettes, notamment sur les menus des catégories et des tâches |  | |
| Doc - Rapport | 26 | 2h10min | Rédaction du rapport, notamment sur les maquettes, le fonctionnement de l'application, la navigation et l'analyse & conception des tests |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **5** |  | **Date:** | **jeudi, 5 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Dev - Général | 25 | 2h5min | Recherches sur des possibilités afin de créer une tâche ou une catégorie. Choix : Rg.Plugins.Popup. Je préfère avoir des popups pour gérer une tâche que d'avoir une nouvelle page. Recherches sur Rg Popups et apprentissage desbases (comment setup le plugin, comment ouvrir un popup, la navigation entre popups, ...) |  | |
| Doc - Journal de travail | 5 | 0h25min | Rédaction du journal de travail de la journée |  | |
| Doc - Rapport | 5 | 0h25min | Rédaction du rapport, notamment sur les maquettes |  | |
| Dev - Github | 1 | 0h5min | Utilisation de github pour le push journalier |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **4** |  | **Date:** | **jeudi, 5 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Base de données | 10 | 0h50min | Installation de DB\_Main afin de modéliser la base de donnée. Mauvaise installation : installation de DB\_Main situé sur le K:/, serveur de l'ETML. La version nécessitait une license, et après configuration et installation de la license, celle-ci était échue. DB\_Main est en fait maintenant gratuit, il faut donc le télécharger sur internet. Un guide d'installation a également été réalisé, ce qui a pu rallonger le temps d'installation. Finalement, la base de données est modélisée sur papier. Elle est très simple et consiste en uniquement deux tables, catégories et tâches. |  | |
| Doc - Maquettes | 5 | 0h25min | Quelques modifications et réflexions sont faites sur la maquette déjà créée, notamment sur les menus des tâches. |  | |
| Dev - Général | 30 | 2h30min | Création d'un projet en Xamarin, basé sur le template "À onglets" proposé par Visual Studio. Ambiguité de renommage avec une classe que j'ai nommée "Task", qui est déjà un Type à part --> Re-création du projet et renommage de Task en TaskModel. Création de la structure de base du programme, des pages ToDoList et Tâches, ainsi que la navigation entre celles-ci. Mise en place des entêtes de code et du nommage correspondant aux normes citées dans mon rapport. Création et mise en lien des Modèles, des Services, des ViewModels et des VIews. |  | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **3** |  | **Date:** | **mercredi, 4 mai 2022** | **Après-midi** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Planification initiale | 13 | 1h5min | Fin de la rédaction de la planification initiale | <https://openclassrooms.com/fr/courses/4568746-gerez-vos-donnees-localement-pour-avoir-une-application-100-hors-ligne/4568753-comprenez-les-differents-moyens-de-stocker-des-donnees-sur-android> | |
| Doc - Rapport | 13 | 1h5min | Rédaction de la partie d'analyse et structure générale du rapport |  | |
| Doc - Journal de travail | 5 | 0h25min | Rédaction du journal de travail de mercredi et lundi |  | |
| Doc - Entretien avec les experts / chef de projet | 5 | 0h25min | Entretien avec M. Ferrari : Présentation de la maquette choisie, du fonctionnement de l'application, du système de stockage de données choisi (SQLite), Du système de testing (Tests manuels prédéfinits au lieu de UI Testing car jugé inadapté à l'ampleur du projet) |  | |
| Total tranche | 36 | 3h |  |  | |
| **Séquence** | **2** |  | **Date:** | **mercredi, 4 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Analyse | 18 | 1h30min | Recherches sur mon choix pour le système de stockage, le système de tests, sur le fonctionnement de l'application. | <https://devblogs.microsoft.com/appcenter/testing-xamarin-applications-with-visual-studio-app-center/> | |
| Doc - Maquettes | 10 | 0h50min | Conception des maquettes, de différents choix possibles |  | |
| Doc - Rapport | 12 | 1h | Rédaction de la partie d'analyse et début |  | |
| Doc - Planification initiale | 5 | 0h25min | Début de la rédaction de la planification initiale |  | |
| Total tranche | 45 | 3h45min |  |  | |
| **Séquence** | **1** |  | **Date:** | **lundi, 2 mai 2022** | **Matin** |
| **Tâche** | **Tranche [5min]** |  | **Explications: qu'est-ce qui se fait et comment ?** | **Liens, références, …** | |
| Doc - Entretien avec les experts / chef de projet | 15 | 1h15min | Entretien avec M. Ferrari et M. Brandenberg : lecture du cahier des charges et questionnement sur le déroulement du projet |  | |
| Doc - Rapport | 21 | 1h45min | Rédaction initiale du rapport : Structure, spécifications, analyse Le rapport est repris du dernier rapport fait durant le projet P\_APPRO, étant donné la ressemblance du projet avec celui réalisé en ce moment. |  | |
| Total tranche | 36 | 3h |  |  | |