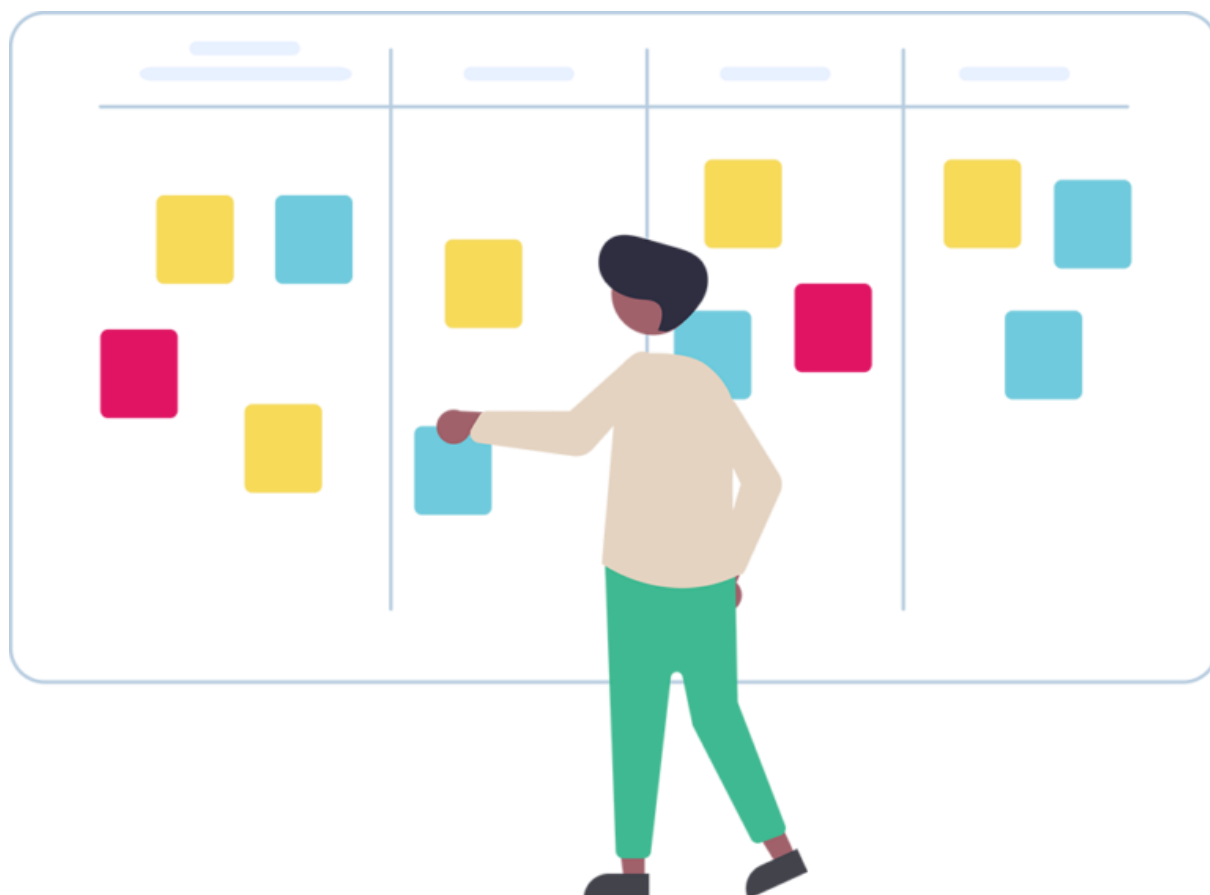


# Projet TPI – Gestionnaire de tâches

---



Tristan Gerber – MID4  
ETML N501  
Du 2 Mai 2022 au 1er juin 2022  
Chef de projet : Roberto Ferrari  
Experts : Joseph Brandenburg, Serge Wenger

## Table des matières

<b>1</b>	<b>SPÉCIFICATIONS.....</b>	<b>5</b>
1.1	TITRE .....	5
1.2	DESCRIPTION .....	5
1.3	MATÉRIEL ET LOGICIELS À DISPOSITION .....	5
1.4	PRÉREQUIS.....	5
1.5	CAHIER DES CHARGES.....	6
1.5.1	Objectifs et portée du projet .....	9
1.5.2	Caractéristiques des utilisateurs et impacts .....	9
1.5.3	Contraintes.....	9
1.6	LES POINTS SUIVANTS SERONT ÉVALUÉS .....	9
1.7	VALIDATION ET CONDITIONS DE RÉUSSITE .....	9
<b>2</b>	<b>ANALYSE.....</b>	<b>9</b>
2.1	OPPORTUNITÉS.....	9
2.2	MÉTHODOLOGIE DE PROJET .....	10
2.2.1	S'informer.....	10
2.2.2	Planifier .....	10
2.2.3	Décider.....	10
2.2.4	Réaliser.....	10
2.2.5	Contrôler.....	10
2.2.6	Évaluer .....	10
2.3	NORMES DE CODAGE .....	10
2.4	DÉPLOIEMENT DE L'APPLICATION .....	11
2.5	LANGAGES UTILISÉS .....	11
2.6	STOCKAGE DES DONNÉES.....	11
2.7	SAUVEGARDE DU PROJET .....	12
2.8	TESTS .....	12
<b>3</b>	<b>CONCEPTION .....</b>	<b>12</b>
3.1	MAQUETTES & STRUCTURE DE L'APPLICATION .....	12
3.2	INFOGRAPHIE DE LA DATE D'ÉCHÉANCE .....	15
3.3	NAVIGATION .....	16
3.4	CONCEPTION DES TESTS.....	16
3.5	PLANIFICATION DÉTAILLÉE .....	17
<b>4</b>	<b>RÉALISATION .....</b>	<b>18</b>
4.1	PROGRAMMES & VERSIONS UTILISÉS .....	18
4.2	BASE DE DONNÉES .....	19
4.3	SQLITE.....	19
4.4	MVVM & CODE .....	21
4.5	SPÉCIFICATIONS DU CODE .....	22
4.5.1	Binding .....	22
4.5.2	ObservableCollections .....	22
4.5.3	Pop-ups.....	23
4.6	COMPILATION .....	24
4.7	RÉALISATION DES TESTS .....	24
<b>5</b>	<b>CONCLUSION.....</b>	<b>25</b>
5.1	BILAN DES FONCTIONNALITÉS DEMANDÉES .....	25
5.2	POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET .....	25
5.3	RÉSULTAT VISUEL.....	26
5.4	BILAN DE LA PLANIFICATION.....	28
5.5	BILAN PERSONNEL .....	30

---

<b>6</b>	<b>WEBOGRAPHIE .....</b>	<b>30</b>
<b>7</b>	<b>TABLE DES ILLUSTRATIONS.....</b>	<b>32</b>
<b>8</b>	<b>ANNEXES .....</b>	<b>33</b>
8.1	PLANIFICATION .....	33
8.2	JOURNAL DE TRAVAIL .....	43

# Résumé du rapport du TPI

Dans le cadre de mon TPI, dont M. Ferrari est le chef de projet, je dois réaliser une application mobile sur Android de gestion de tâches. Le projet dure un mois et je travaille sur ce projet tous les jours sauf le mardi et le lundi après-midi, dans la salle N501 où un ordinateur et un téléphone m'est mis à disposition.

Dans cette application, l'utilisateur peut créer, supprimer et modifier ses tâches. Celles-ci sont triées par date d'échéance, et ont un code couleur particulier accordé. Un système de catégories est également mis en place : l'utilisateur peut créer, supprimer et modifier des catégories, les attribuer à des tâches et finalement trier les tâches par catégorie. Une ToDoList est mise en place, dans laquelle l'utilisateur peut ajouter ou enlever des tâches. Une planification est réalisée au début du projet afin d'en donner le fil conducteur.

Afin de réaliser cette application, j'utilise Xamarin, plateforme de développement mobile tournant avec XAML et C#. Bien que cette plateforme supporte Android, iOS et Windows Phone, je ne réalise mon application que pour Android. J'utilise une structure MVVM (Model View ViewModel), utilisée pour rendre les différents bouts de code indépendants des autres. Chaque View est composée d'un fichier XAML et d'un fichier C#, tandis que tout le reste est en C#. Grâce à un système ingénieux expliqué dans ce rapport, il est très facile d'interagir entre les Views, les ViewModels et les Models à l'aide de « Bindings ». La méthodologie de projet utilisée est la méthode des six pas, car elle est bien structurée et adaptée à un projet de cette envergure. Le système de sauvegarde de données utilisé est SQLite, car bien que je ne l'aie jamais utilisé avant ce projet, il est simple d'utilisation, stocke les données en local et ne nécessite donc pas de connexion à une base de données sur un serveur.

Dans ce projet, je réalise deux pages principales étant la liste des tâches et la ToDoList, le reste des pages étant sous forme de pop-ups - car le système de navigation par pages était trop lourd à mon goût - ce qui a représenté un grand défi ; l'installation d'un plugin a été nécessaire, et j'ai failli repasser à un système de navigation par pages. La planification est bien respectée, excepté quelques irrégularités, et on peut retrouver la méthodologie de travail des six pas dans les différents points du rapport. Ce projet a été un plaisir à faire, et je remercie mon chef de projet, M. Ferrari ainsi que mes deux experts, M. Brandenburg et M. Wenger pour le support qu'ils m'ont apporté.

Bonne lecture.

# 1 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 : S'informer.

## 1.1 Titre

Gestionnaire de tâches

## 1.2 Description

Ce projet consiste en la création d'une application de gestion des tâches. Cette application est fonctionnelle sur Android, réalisée en Xamarin. Elle comporte un système de création, suppression et modification de tâches et de catégories ainsi qu'une ToDoList à laquelle on peut ajouter des tâches.

## 1.3 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>. Afin d'installer Git Bash, il suffit de télécharger le Setup ou la version Portable en appuyant sur « next » à toutes les étapes.

**DB-Main** – Utilisé pour la modélisation de la base de données – Installé depuis le site de DB-Main : [GetIt – DB-Main](#)

## 1.4 Prérequis

Compétences en bases de données

Compétences en programmation

Compétences en outils bureautiques

Compétences en gestion de projets

➔ Modules 104 et 153

➔ Modules 120, 226A et B, 326, 403, 404, 411, 335  
P\_APPRO (Xamarin)

➔ Modules 104 et 153

➔ Modules 104 et 153

## 1.5 Cahier des charges

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









ETML Section informatique		Travail Spécialisé TPI	
<b>1 INFORMATIONS GENERALES</b>			
Candidat :	Nom : <b>GERBER</b>	Prénom : <b>TRISTAN</b>	
	 : tristan.gerber@eduvaud.ch	 :	
Lieu de travail :	ETML, Sébeillon 12 1004 Lausanne		
Orientation :	<input checked="" type="checkbox"/> 88601 Développement d'application <input type="checkbox"/> 88602 Informatique d'entreprise <input type="checkbox"/> 88603 Technique des systèmes		
Chef de projet :	Nom : FERRARI	Prénom : Roberto	
	 : roberto.ferrari@eduvaud.ch	 : 021 31 6 02 63	
Expert 1 :	Nom : BRANDENBERG	Prénom : Joseph	
	 : joseph.brandenberg@icloud.com	 : 078 604 39 03	
Expert 2 :	Nom : WENGER	Prénom : Serge	
	 : serge.wenger@matisa.ch	 : 021 631 22 13	
Période de réalisation :	Du <b>lundi 02 mai 2022 à 8h</b> au <b>mercredi 01 juin 2022 à 12h15</b>		
Horaire de travail : (Basé sur l'horaire officiel)	Lundi 08h00-11h25 - Mardi - - Mercredi 08h00-12h15 13h10-16h35 Jeudi 08h00-12h15 13h10-16h35 <i>Ascension 26 mai</i> Vendredi 08h00-11h25 13h10-16h35 <i>Pont de l'Ascension 27 mai</i> Attention : Toutes les demi-journées ont une pause obligatoire de 15 minutes.		
Nombre d'heures :	89 heures		
Planning	Analyse 20%, Implémentation 40%, Tests 20%, Documentation 20%		
Présentation :	Dates prévues : 9 ou 10 juin 2022		
<b>2 PROCÉDURE</b>			
<p>Le candidat réalise un travail personnel sur la base d'un cahier des charges reçu le 1<sup>er</sup> jour.</p> <p>Le cahier des charges est approuvé par les deux experts. Il est en outre présenté, commenté et discuté avec le candidat. Par sa signature, le candidat accepte le travail proposé.</p> <p>Le candidat a connaissance de la feuille d'appréciation avant de débiter le travail.</p> <p>Le candidat est entièrement responsable de la sécurité de ses données.</p> <p>En cas de problèmes graves, le candidat avertit au plus vite les deux experts et son CdP.</p> <p>Le candidat a la possibilité d'obtenir de l'aide, mais doit le mentionner dans son dossier.</p> <p>A la fin du délai imparti pour la réalisation du TPI, le candidat doit transmettre par courrier électronique le dossier de projet aux deux experts et au chef de projet. En parallèle, une copie papier du rapport doit être fournie sans délai en trois exemplaires (L'un des deux experts peut demander à ne recevoir que la version électronique du dossier). Cette dernière doit être en tout point identique à la version électronique.</p>			
<b>3 TITRE</b>			
Application de gestion de tâches.			
Fichier : TristanGerber-CDC-TPI-2022-V2.docx Auteur :		Version 9 du 06.04.2022 11:23:00 Dernière modification le 11.04.2022 Imprimé le 11.04.2022 08:55:00 à 08:55	

Figure 1 Cahier des charges - Page 1

## 4 MATÉRIEL ET LOGICIEL À DISPOSITION

PC standard de l'ETML (Windows 10)  
Suite Microsoft Office  
IDE C# (Visual Studio Community, Xamarin.Essentials)  
Tablette Android (pour les tests)

## 5 PRÉREQUIS

Compétences en base 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 → Module 302.  
Compétences en gestion de projets → Modules 306, 426 et 431.

## 6 DESCRIPTIF DU PROJET

Le candidat développe une application Android permettant de gérer des tâches.

- Il doit être possible de catégoriser les tâches.
- Il doit être possible d'afficher les tâches par catégorie.
- Il doit être possible d'afficher les tâches par ordre de date d'échéance.
- Il doit être possible d'insérer, de modifier ou de supprimer une tâche.
- Il doit être possible de marquer une tâche comme terminée. A ce moment elle ne doit plus être visible.
- Une infographie par date d'échéance. Par exemple différentes couleurs par proximité de l'échéance.
- 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.

## 7 LIVRABLES

Le candidat est responsable de livrer à son chef de projet et aux deux experts :

- Une planification initiale au format PDF au plus tard le 04 mai 2022.
- A la fin du TPI, son rapport de projet et son journal de travail imprimés,
- A la fin du TPI, son rapport de projet final et son journal de travail au format PDF,
- A la fin du TPI, une archive contenant :
  - un dossier contenant l'application complète (source, classes, exe),
  - une procédure d'installation et de mise en service de l'application.

## 8 POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET

La grille d'évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, ...).

En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants (Point A14 à A20) :

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

Figure 2 - Cahier des charges - Page 2

**9 VALIDATION**

	Lu et approuvé le :	Signature :
Candidat :		
Expert n°1 :		
Expert n° 2 :		
Chef de projet :		

*Figure 3 - Cahier des charges - Page 3*



### 1.5.1 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

### 1.5.2 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.

### 1.5.3 Contraintes

Temps pour le projet : 1 mois

Émulateur ou téléphone pour compiler

## 1.6 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

## 1.7 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

## 2 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 correspond aux trois premières étapes de la méthode des six pas : S'informer, Planifier et Décider

### 2.1 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.

## 2.2 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 maîtrisé et que le cahier des charges ne changera pas, il est plus adapté de tout planifier à l'avance.

Elle consiste en six étapes :

### 2.2.1 S'informer

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

### 2.2.2 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.

### 2.2.3 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.

### 2.2.4 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.

### 2.2.5 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.

### 2.2.6 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.

## 2.3 Normes de codage

Dans tout le programme en C#, les variables suivent le lowerCamelCase 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.

## 2.4 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 et l'exécution directe sur un téléphone évite tout problème lors du déploiement qui pourrait arriver si développé sur émulateur.

## 2.5 Langages 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 ».



Figure 4 - Logo C#

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.



Figure 5 - Logo XAML

Tous ces langages sont réunis par Xamarin, une plateforme de développement mobile développée par Microsoft.

Multiplateforme, elle supporte Android, iOS et Windows Phone.



Figure 6 - Logo Xamarin

## 2.6 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 données 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 données 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.

## 2.7 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.

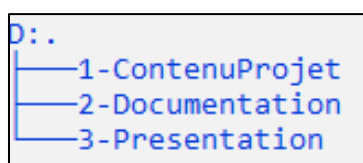


Figure 7 - Structure de fichiers

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

Figure 8 - Tableau des fichiers

## 2.8 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 crashes, 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.

## 3 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 : Planifier & Décider

### 3.1 Maquettes & Structure de l'application

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

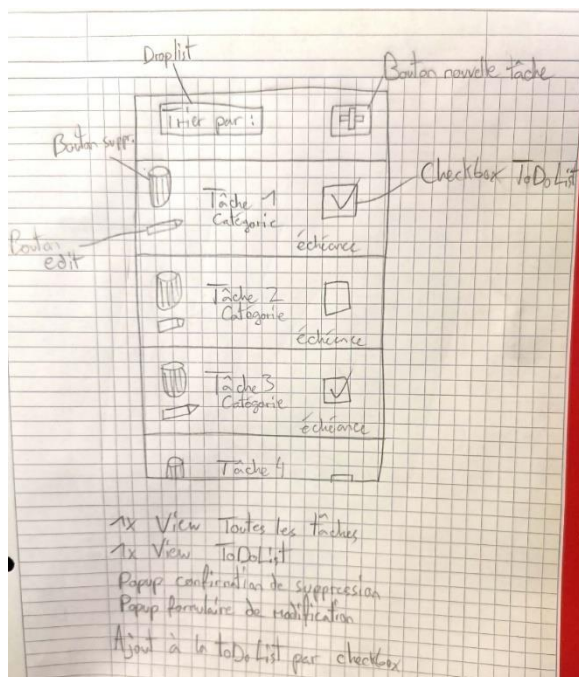


Figure 9 - Maquette initiale

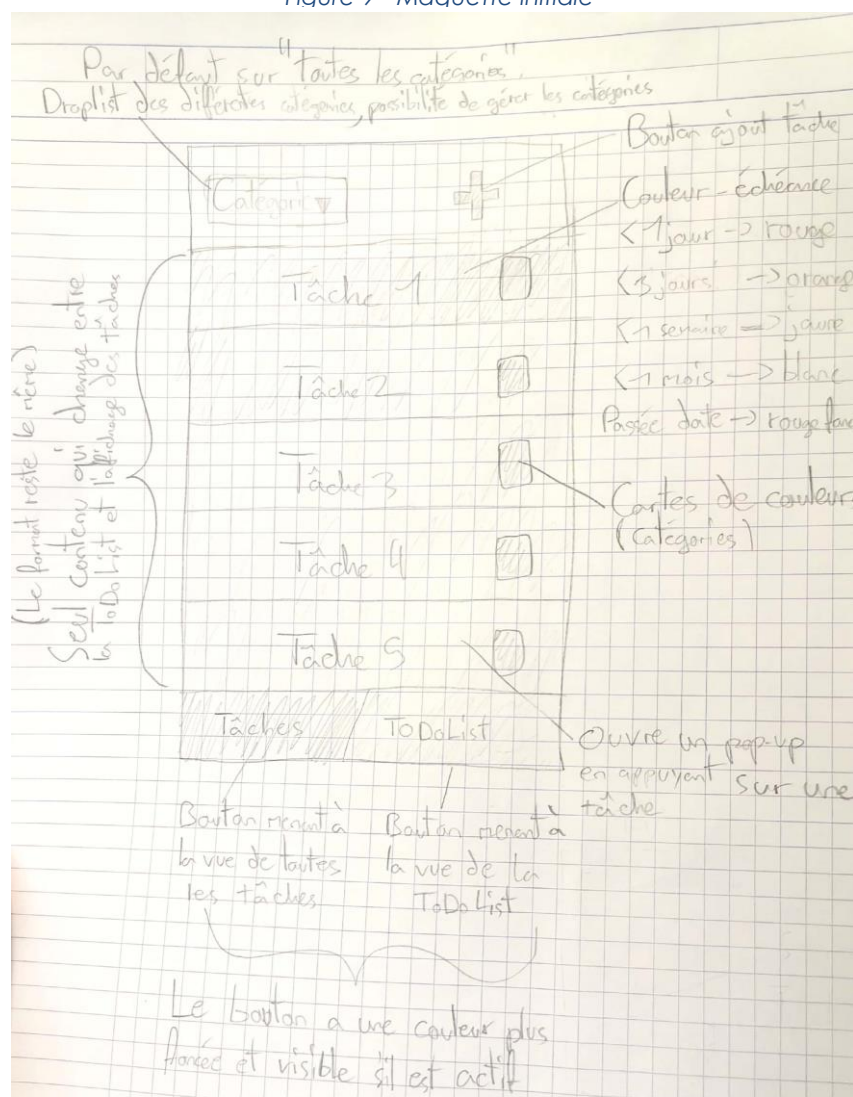


Figure 10 - Maquette finale 1

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.



L'application sera faite à l'aide de pop-ups ; au lieu de changer de page à chaque interaction, on utilise des pop-ups. Ainsi, l'utilisateur est toujours ancré à une page principale.

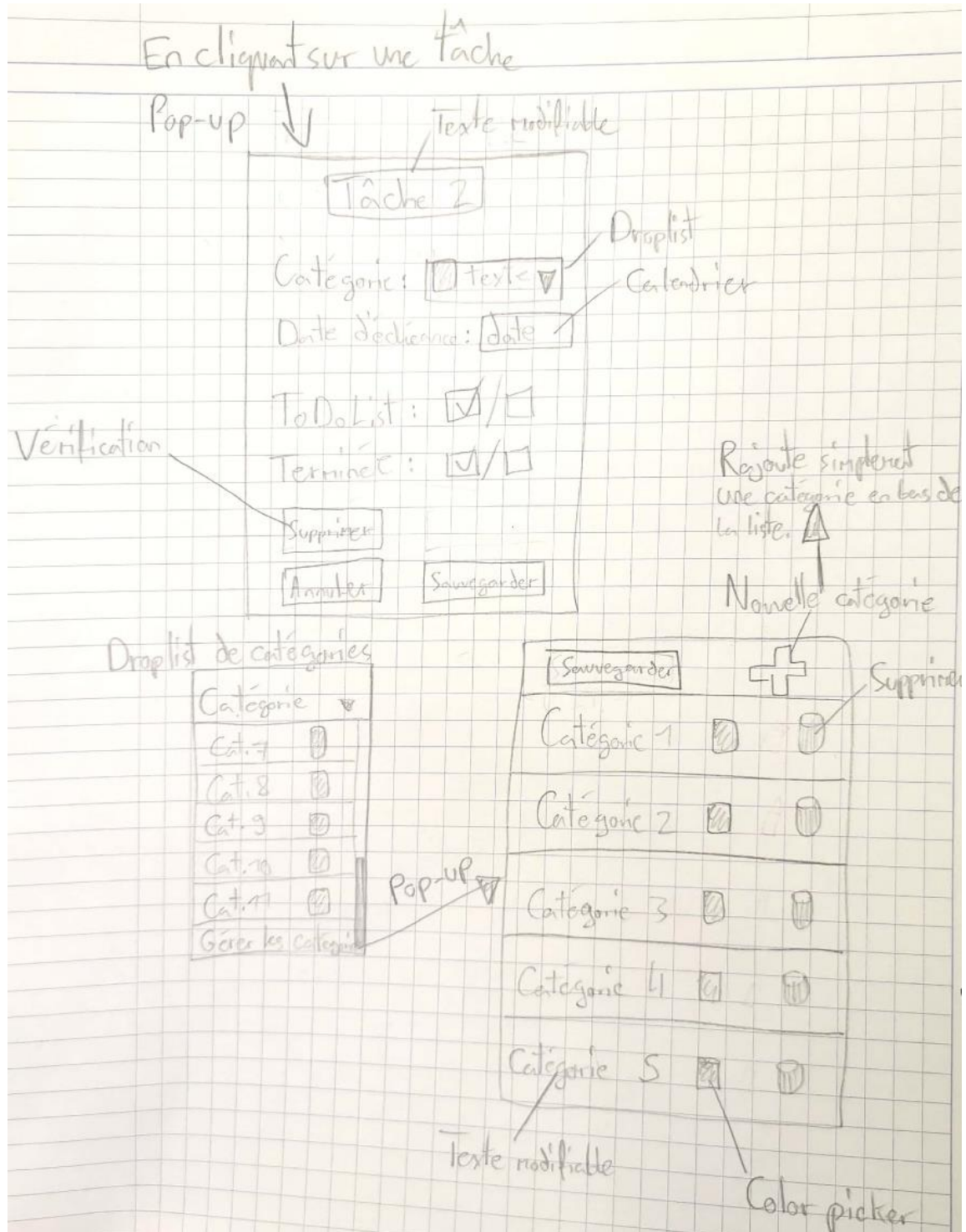


Figure 11 - Maquette finale 2

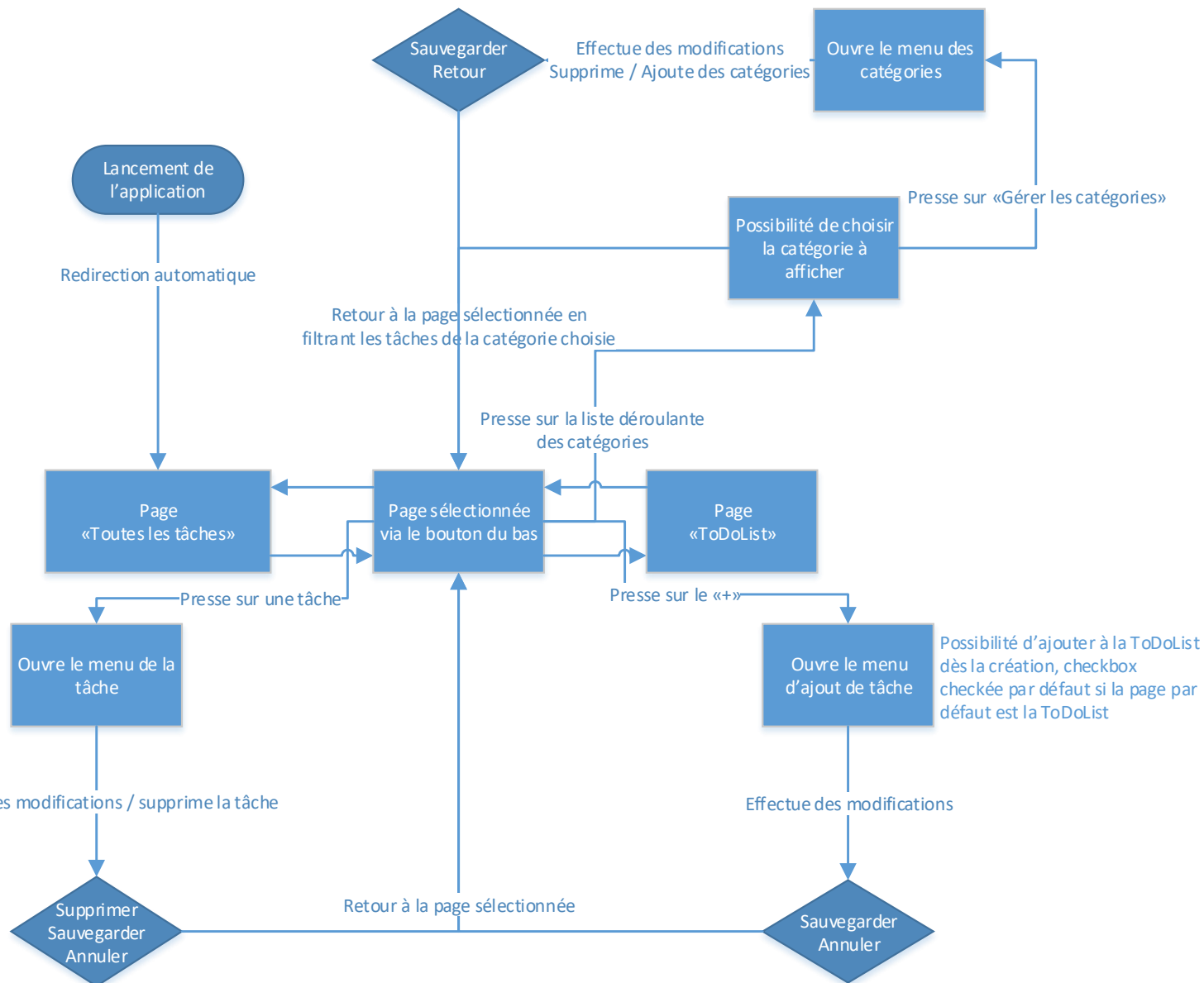


Figure 12 - Schéma Visio

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.

### 3.2 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.

### 3.3 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.

### 3.4 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 »



### 3.5 Planification détaillée

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

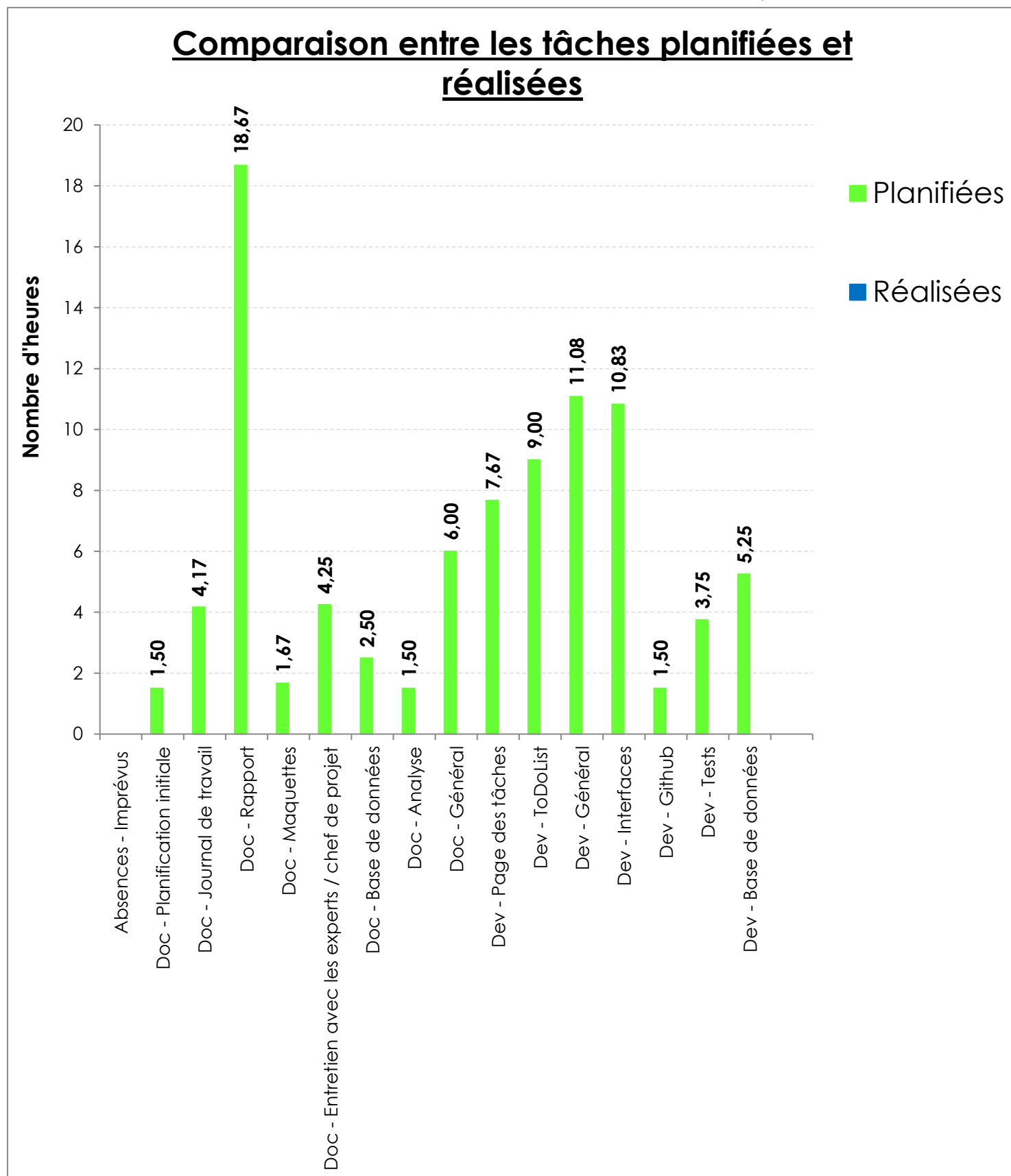


Figure 13 - Planification détaillée

## 4 RÉALISATION

C'est durant la réalisation que tout le code est produit. Dans cette partie, j'aide le lecteur à comprendre le fonctionnement de mon programme, les choix de développement réalisés et les outils utilisés. Cette partie correspond à la quatrième et cinquième étape de la méthode des six pas : Réalisation & Contrôle.

### 4.1 Programmes & versions utilisés

**Code** : Visual Studio 2019

**Planification** : Excel 2016

**Schémas** : Visio 2016

**NuGets** :






	<b>NETStandard.Library</b> par Microsoft A set of standard .NET APIs that are prescribed to be used and supported together. 18a36291e48808fa7ef2d00a764ceb1ec95645a5	2.0.3
	<b>Rg.Plugins.Popup</b> par Kirill Lyubimov Plugin for Xamarin forms. Allows you to open any page as a popup.	2.1.0
	<b>sqlite-net-pcl</b> par SQLite-net SQLite-net is an open source and light weight library providing easy SQLite database storage for .NET, Mono, and Xamarin applications. This version uses SQLitePCLRaw to provide platform independent versions of SQLite.	1.8.116
	<b>Xamarin.Essentials</b> par Microsoft Xamarin.Essentials: a kit of essential API's for your apps	1.6.1 1.7.3
	<b>Xamarin.Forms</b> par Microsoft Build native UIs for iOS, Android, UWP, macOS, Tizen and many more from a single, shared C# codebase	5.0.0.2012 5.0.0.2401

Figure 14 - NuGets

**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 :**



Figure 15 - 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

## 4.2 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.

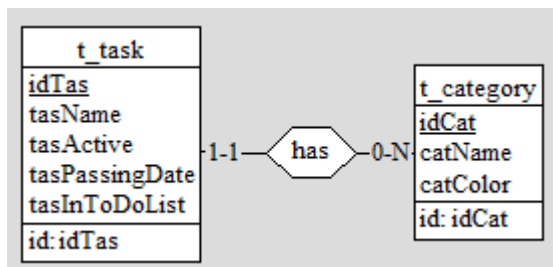


Figure 17 – MCD initial

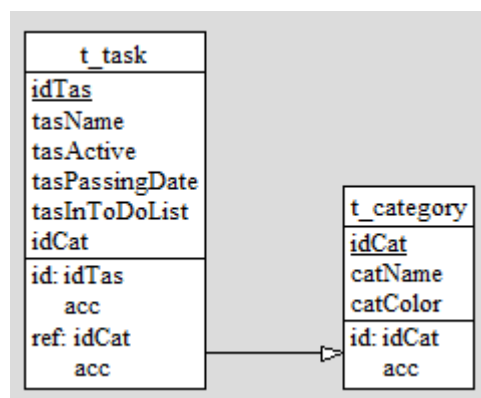


Figure 16 - MLD initial

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 (pas de différence pour l'utilisateur, mais l'application pourrait rencontrer des problèmes si elle devient trop lourde).

Un champ « Description » a également été ajouté à la table des tâches, car jugé nécessaire, et le champ « Active » s'est transformé en « Finished » car cela semblait plus cohérent.

Voici donc la base de données finale :

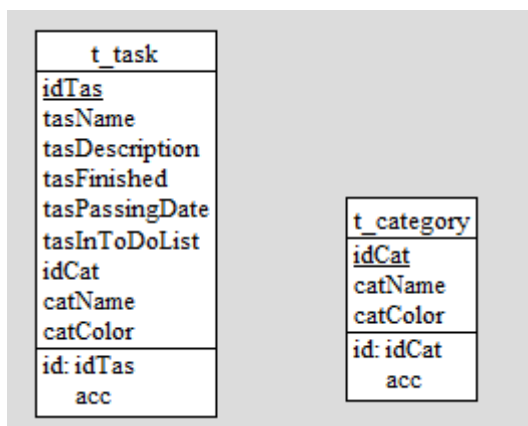


Figure 18 - MLD final

## 4.3 SQLite

SQLite est un système de stockage de données similaire à une base de données classique. Cependant, les données sont stockées localement, la plupart du temps sur un téléphone. Cela permet d'implémenter la base de données directement dans le code.

Par exemple, j'implémente les tables de ma base de données directement dans mon code en tant que classes :

```
public class CategoryModel
{
    [PrimaryKey, AutoIncrement]
    7 références
    public int Id { get; set; }
    7 références
    public string Name { get; set; }
    0 références
    public string Color { get; set; }
}
```

Figure 19 - Code - Model

Une fois les modèles des tables de la base de données créés, il faut créer la base de données contenant les informations en brut :

```
public class Database
{
    readonly SQLiteAsyncConnection database;

    1 référence
    public Database(string dbPath)
    {
        database = new SQLiteAsyncConnection(dbPath);
        database.CreateTableAsync<TaskModel>().Wait();
        database.CreateTableAsync<CategoryModel>().Wait();
    }
}
```

Figure 20 - Code - Database

Ici, on utilise une base de données « SQLiteAsyncConnection », puis on lui attribue un chemin local vers la base de données. Une fois la base de données créée, on lui ajoute des tables venant de nos modèles.

Dans le fichier App.xaml.cs (fichier .cs de App.xaml), que l'on peut voir comme le fichier principal de l'application, on instancie la base de données et on définit le chemin de la base de données :

```
public static Database Db
{
    get
    {
        if (db == null)
        {
            db = new Database(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData), "TaskDb.db3"));
        }
        return db;
    }
}
```

Figure 21 - Code - Instanciation de Database

La base de données étant déclarée comme public static, on peut y accéder n'importe où dans le programme. L'usage de la base de données est cependant restreint aux ViewModels par conformisme à la structure MVVM. Voici un exemple d'usage de la base de données :

```
if (!string.IsNullOrEmpty(_category.Name))
{
    await App.Db.SaveCategoryAsync(_category);
}
```

Figure 22 - Code - Utilisation de Database

Aucune requête SQL n'est directement lancée dans le code. Toutes les requêtes sont faites dans le code, en orienté objet. Par exemple, voici la méthode « SaveCategoryAsync » vue ci-dessus.

```
public Task<int> SaveCategoryAsync(CategoryModel cat)
{
    if (cat.Id != 0)
    {
        return database.UpdateAsync(cat);
    }
    else
    {
        return database.InsertAsync(cat);
    }
}
```

Figure 23 - Code - SaveCategoryAsync()

Ici, on utilise des méthodes directement implémentées dans « database », qui est de type « SQLiteAsyncConnection ».

Il est important de noter qu'afin d'utiliser SQLite en Xamarin, il faut installer l'extension **SQLite-net-pcl**.

## 4.4 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).

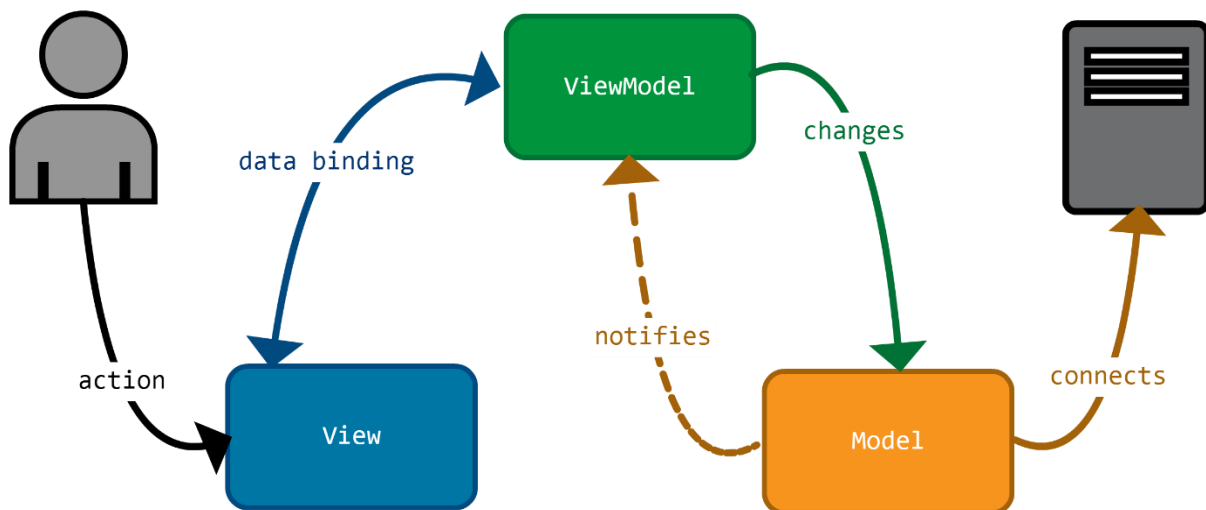


Figure 24 - Schéma MVVM

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. Ils sont utiles par exemple pour éviter d'écrire du code spécifique à une certaine plateforme dans les ViewModels. Dans mon programme, ils assurent la connexion avec la base de données et se chargent de toutes les fonctions directement en lien avec la base de données, pouvant être appelées depuis les ViewModels.

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`.

## 4.5 Spécifications du code

Un nouvel environnement est utilisé : Xamarin. Il possède son propre fonctionnement, qui est expliqué ci-dessous.

### 4.5.1 Binding

Le front-end (xaml) et le back-end (C#) sont reliés par un système de Bindings. C'est-à-dire que l'on peut accéder à des valeurs publiques des ViewModels et des Models directement depuis xaml. Pour ce faire, il faut tout d'abord connecter le ViewModel à la page .xaml.cs de la view. Ici, on a `TaskView` qui se connecte à `TaskViewModel`.

```
TaskViewModel _viewModel;  
0 références  
public TaskView()  
{  
    BindingContext = _viewModel = new TaskViewModel();  
    _viewModel.ToDoListOn = false;  
    InitializeComponent();  
}
```

Figure 25 - Code - Bindings

Une fois cela fait, il faut ajouter les ViewModels et les Models dans les lignes de tête du fichier xaml :

`xmlns:local="clr-namespace:GestTask.ViewModels"` pour les ViewModels ou

`xmlns:model="clr-namespace:GestTask.Models"` pour les Models

Ceci est simplement une inclusion des Models et ViewModels afin de ne pas devoir les appeler par leur source, mais uniquement par `local` ou `model`.

On peut ensuite définir à quel ViewModel ou Model on se connecte grâce à

`x:DataType="local:TaskViewModel"` pour les ViewModels ou `x:DataType="model:TaskModel"` pour les Models.

Lorsqu'on est dans un contexte de `x:DataType="local:TaskViewModel"`, on peut désormais accéder aux valeurs publiques de `TaskViewModel`. Voici des exemples de comment l'on pourrait s'y prendre :

Une commande : `Command="{Binding LoadTasksCommand}"`

Un texte : `Text="{Binding Name}"`

Un paramètre : `IsRefreshing="{Binding IsBusy, Mode=TwoWay}"`

### 4.5.2 ObservableCollection

Comme vu au dernier point, la View se connecte aux ViewModels et aux Models grâce aux Bindings. Si les données récupérées par les Bindings étaient des listes et qu'on modifiait celles-ci, il faudrait recharger la page pour que les Bindings rechargent les listes modifiées. C'est pour

cela que j'utilise des « ObservableCollection<T> ». Ces listes du type <T>, c'est-à-dire n'importe lequel, sont constamment observées par la View et modifient celle-ci dès leur modification.

Les ViewModels ont tout de même leur propre liste de tâches affichées, et celle-ci ne correspond pas forcément à celles trouvées dans la base de données, car il peut y avoir un filtre et bien d'autres modifications par rapport à la base de données. Il faut donc les mettre à jour lorsqu'on fait une modification dans la base de données.

Au lancement du ViewModel de la page des tâches, on exécute une méthode nommée « ExecuteLoadTasksCommand », qui est issue d'une commande de la View. Cette méthode est exécutée à chaque fois que la base de données est mise à jour. En la mettant à jour, la View voit que l'ObservableCollection a changé et change à son tour.

### 4.5.3 Pop-ups

Lors de la conception de l'application, j'ai décidé d'utiliser des Pop-ups pour naviguer dans l'application. Lors du P\_Appro, la navigation dans mon application se faisait sous forme de pages. Le problème avec ce type de navigation est que l'on ne sait très rapidement plus où l'on se trouve dans l'application, et j'ai pensé que sous forme de pop-ups, l'utilisateur aurait toujours une connexion avec la page principale sur laquelle il se trouve.

J'ai utilisé le NuGet Rg.Plugins.Popup, extension permettant d'afficher des pop-ups personnalisés en Xamarin. J'avais déjà utilisé cette extension durant le P\_Appro, mais je n'avais pas réussi à la faire marcher. L'utilisation des pop-ups était donc un défi pour ce projet, étant donnée l'incertitude sur le sujet.

J'ai finalement dans ce projet réussi à bien les implémenter. Afin d'en avoir dans son programme Android (je ne parlerai pas des processus à faire pour iOS), il faut tout d'abord installer le plugin « Rg.Plugins.Popup » dans son programme. Une fois cela fait, il faut initialiser le plugin dans la classe MainActivity avec la ligne « Rg.Plugins.Popup.Popup.Init(this) » :

```
public class MainActivity : global::Xamarin.Forms.Platform.Android.FormsAppCompatActivity
{
    protected override void OnCreate(Bundle savedInstanceState)
    {
        base.OnCreate(savedInstanceState);
        Rg.Plugins.Popup.Popup.Init(this);
        Xamarin.Essentials.Platform.Init(this, savedInstanceState);
        global::Xamarin.Forms.Forms.Init(this, savedInstanceState);
        LoadApplication(new App());
    }
}
```

Figure 26 - Code - Pop-ups

Les pop-ups peuvent ensuite être créés comme le serait une page xaml. En créant une page xaml, afin d'en faire un pop-up, il faut :

1. Modifier le type de la page (ContentView, ContentPage, ...) en « <pages:PopupPage »
2. Ajouter ces lignes en haut de page :

```
xmlns:animations="clr-namespace:Rg.Plugins.Popup.Animations;assembly=Rg.Plugins.Popup"
xmlns:pages="clr-namespace:Rg.Plugins.Popup.Pages;assembly=Rg.Plugins.Popup"
```

3. Définir le type de la page .xaml.cs :  

```
public partial class EditCategoryPopup : Rg.Plugins.Popup.Pages.PopupPage
```

4. Il ne reste désormais plus qu'à ouvrir la page de Pop-up :

```
_modalPage = new TaskMenuPopup(task, this);
await _popup.PushAsync(_modalPage, true);
```

5. Afin de fermer une page de Pop-up, il faut utiliser cette ligne :  
`await _popup.PopAsync();`

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

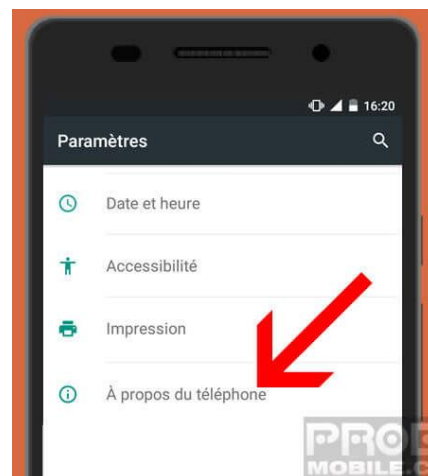


Figure 27 – Compilation  
[www.prodigemobile.com](http://www.prodigemobile.com)

## 4.7 Réalisation 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 »

Ces tests n'ont pas été réalisés d'une traite. Ils ont été réalisés au cours du projet, car chaque test fait d'une manière ou d'une autre partie des objectifs du projet. Tous les tests sont fonctionnels, sauf le 11<sup>ème</sup> : « 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. La liste



pourrait s'étendre à l'entièreté des fonctionnalités du programme, mais j'ai jugé plus simple de simplement vérifier la fonctionnalité du programme en le développant que de perdre du temps à écrire des tests à l'avance qui vont de toute façon être faits au cours du projet, et qui pourraient changer, comme l'a fait le 11<sup>ème</sup> de la liste.

## 5 CONCLUSION

Cette rubrique contient les différents résultats obtenus, ainsi que les conclusions des différents thèmes de ce projet et ce que l'on peut en tirer. Cette partie correspond aux deux dernières étapes de la méthode des six pas : Contrôle & Évaluation

### 5.1 Bilan des fonctionnalités demandées

Toutes les fonctionnalités demandées sont implémentées et fonctionnelles.

Il doit être possible de catégoriser les tâches	OK
Il 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 « ToDoList » à partir des tâches. Par exemple si l'utilisateur veut choisir les tâches qu'il veut réaliser dans la journée.	OK

Figure 28 - Bilan des fonctionnalités demandées

### 5.2 Points techniques évalués spécifiques au projet

Tous les points techniques évalués spécifiques au projet sont complétés.

L'ergonomie de l'application (user friendly et simple), la facilité d'utilisation.	L'application est très simple d'utilisation ; l'utilisateur peut aller où il le souhaite en deux touches d'écran.
La modélisation de la base de données (MCD, MLD et MPD).	La base de données est très simple, mais efficace. Il a été question de rajouter une table pour pouvoir attribuer de multiples catégories à une tâche, mais cette proposition a été repoussée aux modifications à effectuer si le temps le permet.
La pertinence de l'infographie proposée.	L'infographie des couleurs est simple, facile à comprendre et parfaitement fonctionnelle.
Le bon fonctionnement des créations et modification des tâches.	La création et modification des tâches marchent sans problèmes.
Le bon fonctionnement des affichages des tâches.	L'affichage des tâches est fonctionnel, et les tâches se mettent à jour à la moindre modification.
Le bon fonctionnement de création d'une « ToDoList ».	La ToDoList est implémentée et fonctionne sans accroc.
La qualité et lisibilité du code source (respect des normes ETML).	Le code est facile à lire, assez optimisé et commenté.

Figure 29 - Points techniques évalués spécifiques au projet

## 5.3 Résultat visuel

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

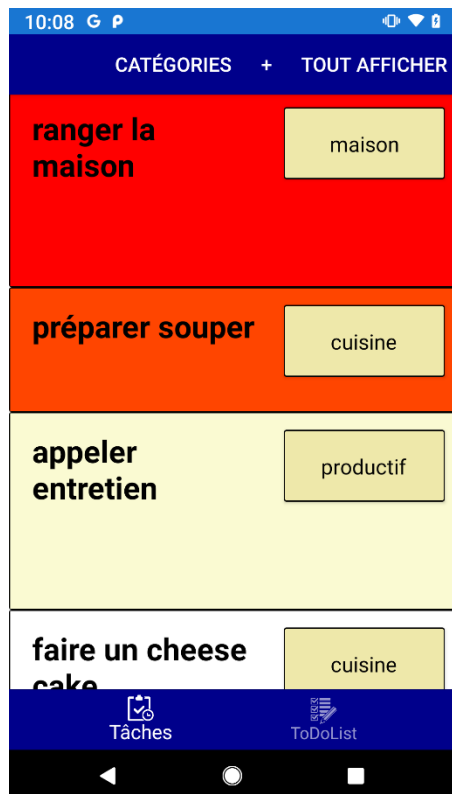


Figure 30 - Résultat - Liste des tâches

Sur la page des tâches, on peut voir les différentes tâches, triées par date d'échéance. Les couleurs suivent le code couleur défini 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.

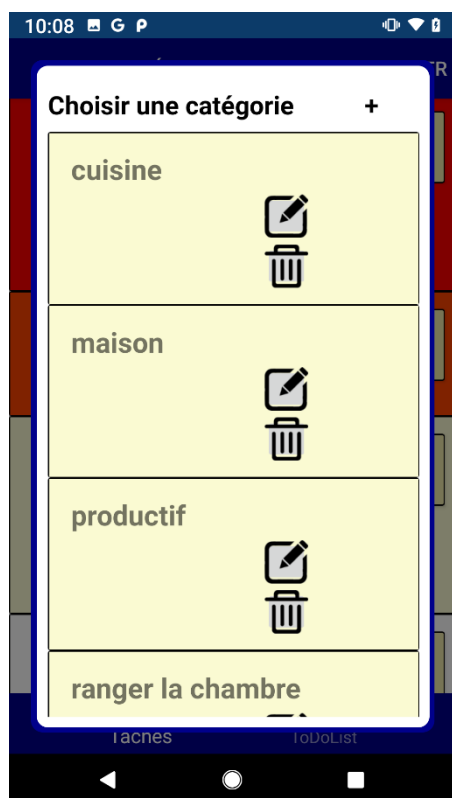


Figure 32 - Résultat - Menu des catégories

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.



Figure 31 - Résultat - Ajout de catégorie

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



Figure 34 - Résultat - Tâches filtrées

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 ».

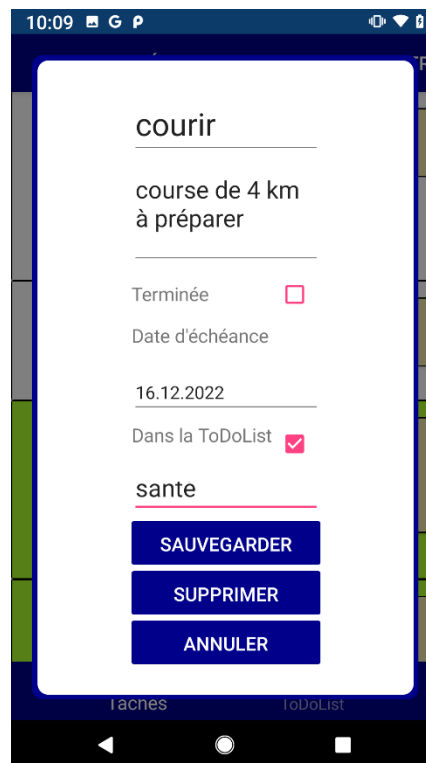


Figure 33 - Résultat - Menu de tâche

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.

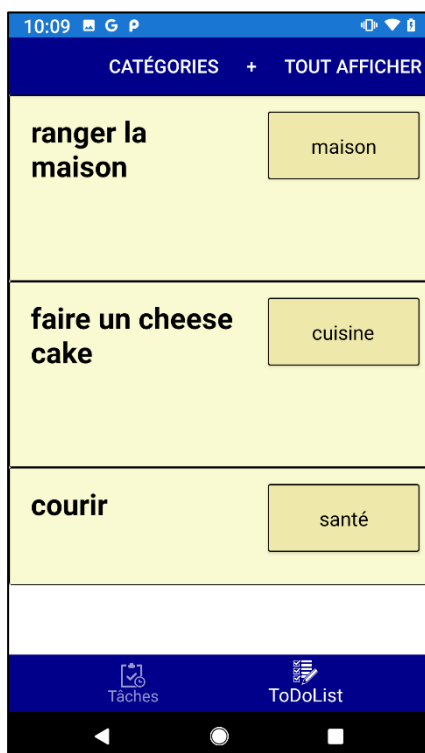


Figure 35 - Résultat - ToDoList

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.

## 5.4 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
<b>Total planifié</b>		<b>1008</b>
		84h
<b>Total réalisé</b>		<b>1008</b>
		84h

Figure 36 - Bilan de la planification

## Comparaison entre les tâches planifiées et réalisées

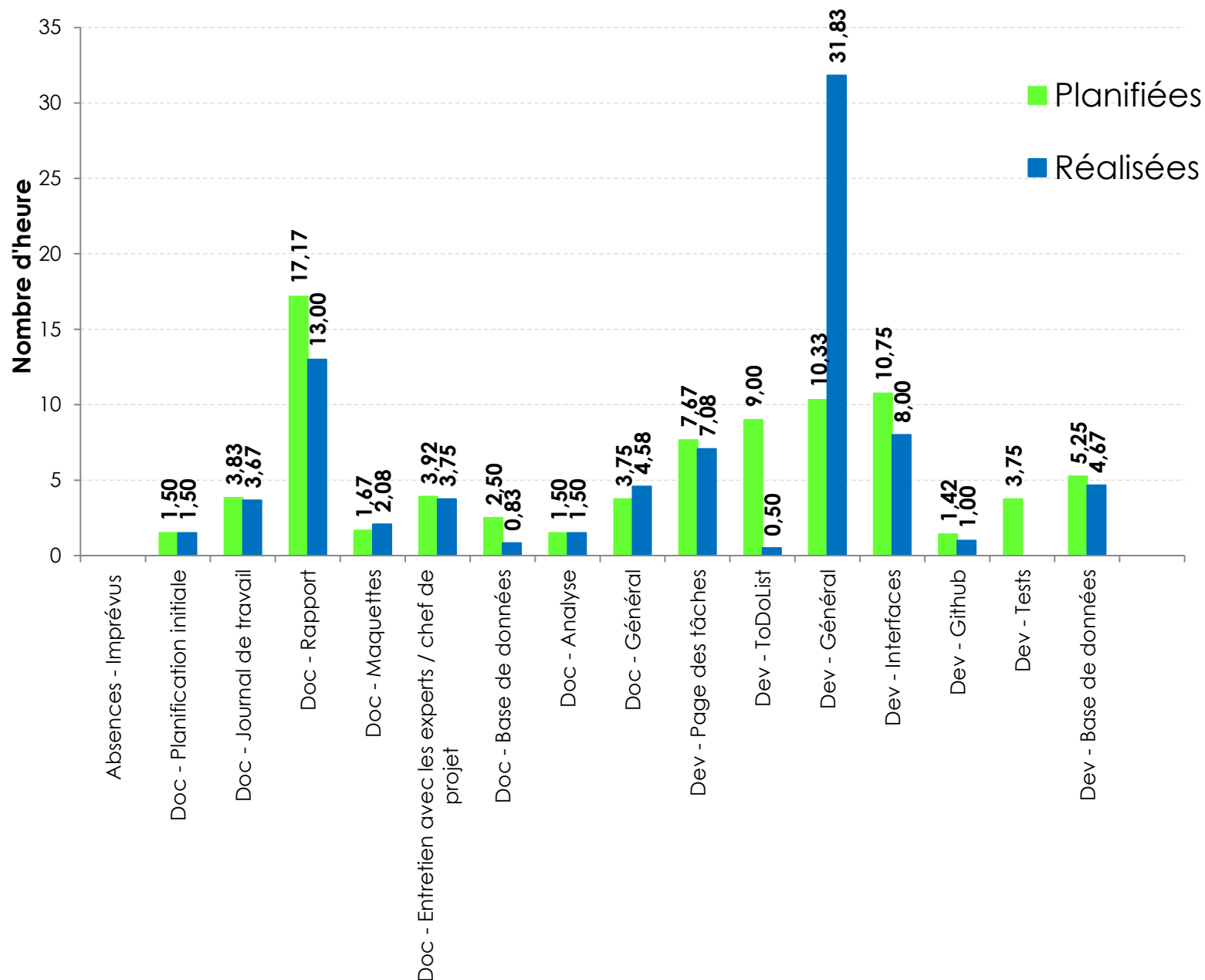


Figure 37 - Comparaison entre les tâches planifiées et réalisées

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.

## 5.5 Bilan personnel

Dans ce projet, j'ai pu utiliser mes compétences en documentation et développement, et j'ai pu constater l'amélioration par rapport aux derniers projets réalisés dans tous les domaines. Je suis particulièrement satisfait de mon analyse et de ma conception, qui laissaient à désirer lors du projet P\_Appro. Il est vrai que j'ai rencontré de nombreux problèmes lors du développement, mais cela n'a que renforcé ma confiance en moi à résoudre n'importe quel problème et à arriver à mon objectif, quel qu'il soit.

Je ne recommanderais pas en général Xamarin à quelqu'un voulant développer une application mobile, sauf si cette personne est un expert en C#, car il y a beaucoup de subtilités dans Xamarin que j'ai mis longtemps à comprendre. Le langage est également limité. Par exemple, il est impossible de mettre de droplist dans un ShellMenu, ce qui m'a retardé et m'a forcé à m'éloigner de la maquette initiale. Le système MVVM mis en place de base utilise des interfaces de type <T>, des héritages, des singletons et autre, ce qui peut être compliqué pour quelqu'un qui n'est pas totalement à l'aise avec C#. Une fois compris, le code est quand même beau, et ça vaut la peine de prendre le temps de comprendre ce qui se passe en arrière.

Je tiens finalement à remercier encore une fois mon expert, M. Ferrari, pour avoir si diligemment veillé sur moi, et à mes deux experts, M. Wenger et M. Brandenburg, pour le temps qu'ils m'ont accordé.

## 6 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.

Les liens affichés ci-dessous vont de la première recherche à la dernière recherche, correspondant aux liens dans le journal de travail.

[Comprenez les différents moyens de stocker des données sur Android - Gérez vos données localement pour avoir une application 100 % hors-ligne - OpenClassrooms](#)

[Testing Xamarin Applications with Visual Studio App Center - App Center Blog \(microsoft.com\)](#)

[Getting started · roorgames/Rg.Plugins.Popup Wiki \(github.com\)](#)

[PopupPage · roorgames/Rg.Plugins.Popup Wiki \(github.com\)](#)

[Using SQLite.NET with Android - Xamarin | Microsoft Docs](#)

[Store data in a local SQLite.NET database - Xamarin | Microsoft Docs](#)

[c# - Adding a button to the title bar Xamarin Forms - Stack Overflow](#)

[Rg Popup In Xamarin.Forms Using Fresh MVVM \(c-sharpcorner.com\)](#)

[Create a list of dropdown lists in Xamarin.Forms\(Xaml\) - Microsoft Q&A](#)

[xaml - How to bind entry from page with ViewModel - Stack Overflow](#)

[Sorting in Xamarin ListView control | Syncfusion](#)

[Xamarin Forms Toolbar – Xamarin Help](#)

[Toolbar Item In Xamarin Cross Platform With Toolbar Order Example \(c-sharpcorner.com\)](#)

[How to create a Custom Layout of Toolbar with Xamarin Forms? \(microsoft.com\)](#)

[how to add picker in toolbar in xamarin forms - Stack Overflow](#)

[Populate Picker Using MVVM \(c-sharpcorner.com\)](#)

[Xamarin Forms - Bindable Picker\(!!\) \(codemilltech.com\)](#)

[mvvm - Xamarin Forms Default value for Picker - Stack Overflow](#)

[Xamarin.Forms RefreshView - Xamarin | Microsoft Docs](#)

[c# - Convert a List<T> into an ObservableCollection<T> - Stack Overflow](#)

[Getting Started with Xamarin ListView control | Syncfusion](#)

[Load More in Xamarin ListView control | Syncfusion](#)

[How to show Xamarin.Forms ListView \(SfListView\) in popup using Rg.Plugin.Popup framework ? | Syncfusion KB](#)

[SyncfusionExamples/rgplugin-popup-listview-xamarin: How to load ListView using Rg.plugin.Popup in Xamarin forms ListView \(SfListView\)? \(github.com\)](#)

[c# - How to show picker choose before button command in MVVM xamarin? - Stack Overflow](#)

[Xamarin.Forms RefreshView - Xamarin | Microsoft Docs](#)

[Building An Edit Label Custom Control in Xamarin Forms. - Cool Coders \(doumer.me\)](#)

## 7 TABLE DES ILLUSTRATIONS

Figure 1 - Cahier des charges - Page 1 .....	6
Figure 2 - Cahier des charges - Page 2 .....	7
Figure 3 - Cahier des charges - Page 3 .....	8
Figure 4 - Logo C# .....	11
Figure 5 - Logo XAML .....	11
Figure 6 - Logo Xamarin .....	11
Figure 7 - Structure de fichiers .....	12
Figure 8 - Tableau des fichiers .....	12
Figure 9 - Maquette initiale .....	13
Figure 10 - Maquette finale 1 .....	13
Figure 11 - Maquette finale 2 .....	14
Figure 12 - Schéma Visio .....	15
Figure 13 - Planification détaillée .....	17
Figure 14 - NuGets .....	18
Figure 15 - Version d'Android .....	18
Figure 16 - MLD initial .....	19
Figure 17 - MCD initial .....	19
Figure 18 - MLD final .....	19
Figure 19 - Code - Model .....	20
Figure 20 - Code - Database .....	20
Figure 21 - Code - Instanciation de Database .....	20
Figure 22 - Code - Utilisation de Database .....	20
Figure 23 - Code - SaveCategoryAsync() .....	21
Figure 24 - Schéma MVVM .....	21
Figure 25 - Code - Bindings .....	22
Figure 26 - Code - Pop-ups .....	23
Figure 27 - Compilation www.prodigemobile.com .....	24
Figure 28 - Bilan des fonctionnalités demandées .....	25
Figure 29 - Points techniques évalués spécifiques au projet .....	25
Figure 30 - Résultat - Liste des tâches .....	26
Figure 31 - Résultat - Ajout de catégorie .....	26
Figure 32 - Résultat - Menu des catégories .....	26
Figure 33 - Résultat - Menu de tâche .....	27
Figure 34 - Résultat - Tâches filtrées .....	27
Figure 35 - Résultat - ToDoList .....	27
Figure 36 - Bilan de la planification .....	28
Figure 37 - Comparaison entre les tâches planifiées et réalisées .....	29



## 8 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.

### 8.1 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 /	4	0h20min	Entretien avec M. Ferrari pour répondre à des questions et vérifier l'avancement du projet	

chef de projet				
Total tranche	36	3h		
<b>Séquence 23</b>			Date: <b>mercredi, 25 mai 2022</b>	<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
Doc - Rapport	45	3h45min	Rédaction du rapport, notamment de la réalisation et la conclusion	
Total tranche	45	3h45min		
<b>Séquence 22</b>			Date: <b>lundi, 23 mai 2022</b>	<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 21</b>			Date: <b>vendredi, 20 mai 2022</b>	<b>Après-midi</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 20</b>			Date: <b>vendredi, 20 mai 2022</b>	<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
Dev - Général	36	3h	Développement général, amélioration du code	
Total tranche	36	3h		
<b>Séquence 19</b>			Date: <b>jeudi, 19 mai 2022</b>	<b>Après-midi</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 18</b>			Date: <b>jeudi, 19 mai 2022</b>	<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
Dev - Tests	45	3h45min	Test des différentes fonctionnalités suivant les tests écrits auparavant dans le rapport	
Total tranche	45	3h45min		
<b>Séquence 17</b>			Date: <b>mercredi, 18 mai 2022</b>	<b>Après-midi</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 16</b>			Date: <b>mercredi, 18 mai 2022</b>	
			<b>Matin</b>	
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 15</b>			Date: <b>lundi, 16 mai 2022</b>	
			<b>Matin</b>	
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 11</b>			Date: <b>jeudi, 12 mai 2022</b>	<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 10</b>			Date: <b>mercredi, 11 mai 2022</b>	<b>Après-midi</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 9</b>			Date: <b>mercredi, 11 mai 2022</b>	<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 8</b>			Date: <b>lundi, 9 mai 2022</b>	<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 7</b>			Date: <b>vendredi, 6 mai 2022</b>	<b>Après-midi</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 6</b>			<b>Date:</b> <b>vendredi, 6 mai 2022</b>	<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 5</b>			<b>Date:</b> <b>jeudi, 5 mai 2022</b>	<b>Après-midi</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 4</b>			<b>Date:</b> <b>jeudi, 5 mai 2022</b>	<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 3</b>			<b>Date:</b> <b>mercredi, 4 mai 2022</b>	<b>Après-midi</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 2</b>			<b>Date:</b> <b>mercredi, 4 mai 2022</b>	<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 1</b>			<b>Date:</b> <b>lundi, 2 mai 2022</b>	<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		

## 8.2 Journal de travail

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

<b>Séquence 26</b>			Date: <b>mercredi, 1 juin 2022</b>		<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>	
Doc - Général	45	3h45min	Finalisation du projet		
Total tranche	45	3h45min			
<b>Séquence 25</b>			Date: <b>lundi, 30 mai 2022</b>		<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>	
Doc - Rapport	26	2h10min	Rédaction du rapport		
Doc - Général	10	0h50min	Finalisation du projet		
Total tranche	36	3h			
<b>Séquence 24</b>			Date: <b>mercredi, 25 mai 2022</b>		<b>Après-midi</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>	
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			
<b>Séquence 23</b>			Date: <b>mercredi, 25 mai 2022</b>		<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>	

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		
<b>Séquence 22</b>			<b>Date:</b> <b>lundi, 23 mai 2022</b>	<b>Matin</b>
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 quelque chose. 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		
<b>Séquence 21</b>			<b>Date:</b> vendredi, 20 mai 2022	<b>Après-midi</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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 passer une seule variable définissant si l'on est dans la ToDoList que de tout recréer.	
Total tranche	36	3h		
<b>Séquence 20</b>			<b>Date:</b> vendredi, 20 mai 2022	<b>Matin</b>
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.	<a href="https://doumer.me/building-an-edit-label-custom-control-in-xamarin-forms/">https://doumer.me/building-an-edit-label-custom-control-in-xamarin-forms/</a>
Total tranche	36	3h		
<b>Séquence 19</b>			<b>Date:</b> jeudi, 19 mai 2022	<b>Après-midi</b>
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Dev - Page des tâches	20	1h40min	Implémentation de la suppression 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	<a href="https://docs.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/refreshview">https://docs.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/refreshview</a>

			connecté au mauvais ViewModel. Toujours un problème pour afficher le nom de la catégorie choisie dans le picker par défaut	
Doc - Journal de travail	5	0h25min	Rédaction du journal de travail journalier	<a href="https://stackoverflow.com/questions/60472467/how-to-show-picker-choose-before-button-command-in-mvvm-xamarin">https://stackoverflow.com/questions/60472467/how-to-show-picker-choose-before-button-command-in-mvvm-xamarin</a>
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		
<b>Séquence 18</b>			Date: <b>jeudi, 19 mai 2022</b>	
			<b>Matin</b>	
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 licence 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	<a href="https://github.com/SyncfusionExamples/rqplugin-popup-listview-xamarin">https://github.com/SyncfusionExamples/rqplugin-popup-listview-xamarin</a>



			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.	
Dev - Interfaces	20	1h40min	Modifications de l'interface allant avec le code. Réutilisation du design de popup trouvé sur le premier lien GitHub	<a href="https://www.syncfusion.com/kb/11353/how-to-show-xamarin-forms-listview-sfplugin-popup-using-rg-plugin-popup-framework">https://www.syncfusion.com/kb/11353/how-to-show-xamarin-forms-listview-sfplugin-popup-using-rg-plugin-popup-framework</a>
Total tranche	45	3h45min		
<b>Séquence 17</b>			<b>Date:</b>	<b>mercredi, 18 mai 2022</b>
				<b>Après-midi</b>
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	<a href="https://docs.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/refreshview">https://docs.microsoft.com/en-us/xamarin/xamarin-forms/user-interface/refreshview</a>
Doc - Journal de travail	4	0h20min	Rédaction du journal de travail journalier	<a href="https://stackoverflow.com/questions/56239550/xamarin-forms-default-value-for-picker">https://stackoverflow.com/questions/56239550/xamarin-forms-default-value-for-picker</a>
Dev - Github	1	0h5min	Utilisation de Github	<a href="https://codemilltech.com/xamarin-forms-bindable-picker/#:~:text=The%20ItemDisplayBinding%20property%20is%20of%20a%20BindingBase%20type%2C,determines%20what%20actually%20gets%20disp">https://codemilltech.com/xamarin-forms-bindable-picker/#:~:text=The%20ItemDisplayBinding%20property%20is%20of%20a%20BindingBase%20type%2C,determines%20what%20actually%20gets%20disp</a>

				<a href="#">layed%20within%20the%20Picker.</a>
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		
<b>Séquence 16</b>			<b>Date:</b> <b>mercredi, 18 mai 2022</b>	<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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}.	<a href="https://www.c-sharpcorner.com/article/populate-picker-using-mvvm/">https://www.c-sharpcorner.com/article/populate-picker-using-mvvm/</a>
Total tranche	45	3h45min		
<b>Séquence 15</b>			<b>Date:</b> <b>lundi, 16 mai 2022</b>	<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>

Doc - Entretien avec les experts / chef de projet	5	0h25min	Entretien avec M. Wenger, un de mes deux experts pour la seconde visite	<a href="https://www.c-sharpcorner.com/article/toolbar-item-in-xamarin-cross-platform-with-toolbar-order-example/">https://www.c-sharpcorner.com/article/toolbar-item-in-xamarin-cross-platform-with-toolbar-order-example/</a>	
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.	<a href="https://social.msdn.microsoft.com/Forums/en-US/47077ec2-6fea-479d-8737-7dfe809f7da5/how-to-create-a-custom-layout-of-toolbar-with-xamarin-forms">https://social.msdn.microsoft.com/Forums/en-US/47077ec2-6fea-479d-8737-7dfe809f7da5/how-to-create-a-custom-layout-of-toolbar-with-xamarin-forms</a>	
Doc - Journal de travail	1	0h5min	Rédaction du journal de travail de la journée	<a href="https://stackoverflow.com/questions/38988108/how-to-add-picker-in-toolbar-in-xamarin-forms">https://stackoverflow.com/questions/38988108/how-to-add-picker-in-toolbar-in-xamarin-forms</a>	
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	<a href="https://www.xamarinhelp.com/xamarin-forms-toolbar/">https://www.xamarinhelp.com/xamarin-forms-toolbar/</a>	
Doc - Rapport	10	0h50min	Rédaction du rapport	<a href="https://www.c-sharpcorner.com/article/toolbar-item-in-xamarin-cross-platform-with-toolbar-order-example/">https://www.c-sharpcorner.com/article/toolbar-item-in-xamarin-cross-platform-with-toolbar-order-example/</a>	
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		
<b>Séquence 12</b>			<b>Date:</b> jeudi, 12 mai 2022	<b>Après-midi</b>
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	<a href="https://docs.microsoft.com/en-us/answers/questions/243785/create-a-list-of-dropdown-lists-in-xamarinformsxaml.html">https://docs.microsoft.com/en-us/answers/questions/243785/create-a-list-of-dropdown-lists-in-xamarinformsxaml.html</a>
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, ...)	<a href="https://stackoverflow.com/questions/55414973/how-to-bind-entry-from-page-with-viewmodel">https://stackoverflow.com/questions/55414973/how-to-bind-entry-from-page-with-viewmodel</a>
Doc - Journal de travail	4	0h20min	Rédaction du journal de travail de la journée	<a href="https://help.syncfusion.com/xamarin/listview/sorting">https://help.syncfusion.com/xamarin/listview/sorting</a>
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)	<a href="https://docs.microsoft.com/en-us/answers/questions/243785/create-a-list-of-dropdown-lists-in-xamarinformsxam.html">https://docs.microsoft.com/en-us/answers/questions/243785/create-a-list-of-dropdown-lists-in-xamarinformsxam.html</a>
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, déblocage 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	<a href="https://www.c-sharpcorner.com/article/rg-popup-in-xamarin-forms-using-fresh-mvvm/">https://www.c-sharpcorner.com/article/rg-popup-in-xamarin-forms-using-fresh-mvvm/</a>
Doc - Entretien avec les experts / chef de projet	4	0h20min	Entretien avec M. Ferrari afin de parler de l'avancement du projet :	<a href="https://github.com/rotrorgames/Rg.Plugins.Popup/wiki/Getting-started">https://github.com/rotrorgames/Rg.Plugins.Popup/wiki/Getting-started</a>

Dev - Github	1	0h5min	Utilisation de github pour le push journalier	<a href="https://github.com/roorgames/Rg.Plugins.Popup/wiki/PopupPage">https://github.com/roorgames/Rg.Plugins.Popup/wiki/PopupPage</a>
Doc - Journal de travail	4	0h20min	Rédaction du journal de travail de la journée	
Total tranche	36	3h		
<b>Séquence 9</b>			<b>Date: mercredi, 11 mai 2022</b>	
			<b>Matin</b>	
Tâche	Tranche [5min]		Explications: qu'est-ce qui se fait et comment ?	Liens, références, ...
Dev - Général	45	3h45min	Code de la structure 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.	<a href="https://stackoverflow.com/questions/39384263/adding-a-button-to-the-title-bar-xamarin-forms">https://stackoverflow.com/questions/39384263/adding-a-button-to-the-title-bar-xamarin-forms</a>
Total tranche	45	3h45min		
<b>Séquence 8</b>			<b>Date: lundi, 9 mai 2022</b>	
			<b>Matin</b>	
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.	<a href="https://docs.microsoft.com/en-us/xamarin/get-started/quickstarts/database?pi_vots=windows">https://docs.microsoft.com/en-us/xamarin/get-started/quickstarts/database?pi_vots=windows</a>

Total tranche	36	3h		
<b>Séquence 7</b>			Date: <b>vendredi, 6 mai 2022</b>	<b>Après-midi</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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,	<a href="https://docs.microsoft.com/en-us/xamarin/android/data-cloud/data-access/using-sqlite-orm">https://docs.microsoft.com/en-us/xamarin/android/data-cloud/data-access/using-sqlite-orm</a>
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		
<b>Séquence 6</b>			Date: <b>vendredi, 6 mai 2022</b>	<b>Matin</b>
<b>Tâche</b>	<b>Tranche [5min]</b>		<b>Explications: qu'est-ce qui se fait et comment ?</b>	<b>Liens, références, ...</b>
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		
<b>Séquence 5</b>			Date: <b>jeudi, 5 mai 2022</b>	<b>Après-midi</b>

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 des bases (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		
<b>Séquence 4</b>			Date: <b>jeudi, 5 mai 2022</b>	<b>Matin</b>
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 données. Mauvaise installation : installation de DB_Main situé sur le K:/, serveur de l'ETML. La version nécessitait une licence, et après configuration et installation de la licence, 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. Ambiguïté 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		
<b>Séquence 3</b>			<b>Date:</b> <b>mercredi, 4 mai 2022</b>	<b>Après-midi</b>
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	<a href="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">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</a>
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 prédéfinis 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.	<a href="https://devblogs.microsoft.com/appcenter/testing-xamarin-applications-with-visual-studio-app-center/">https://devblogs.microsoft.com/appcenter/testing-xamarin-applications-with-visual-studio-app-center/</a>
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. Brandenburg : 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		