

**Oublie-pas**

---

Table des matières

1	Analyse préliminaire .....	3
1.1	Introduction .....	3
1.2	Objectifs.....	3
1.3	Planification initiale .....	4
2	Analyse / Conception.....	5
2.1	Concept .....	5
2.2	Stratégie de test.....	7
2.3	Risques techniques .....	7
2.4	Planification .....	8
2.5	Dossier de conception .....	8
2.5.1	Choix de l'IDE .....	8
2.5.2	Base de données.....	8
2.5.3	Maquette.....	9
3	Réalisation.....	10
3.1	Dossier de réalisation .....	10
3.1.1	Version d'Android .....	10
3.1.2	Création des layouts .....	10
3.2	Description des tests effectués .....	13
3.3	Erreurs restantes .....	13
3.4	Liste des documents fournis .....	13
4	Conclusions .....	13
5	Annexes.....	14
5.1	Résumé du rapport du TPI / version succincte de la documentation .....	14
5.2	Sources – Bibliographie.....	14
5.3	Journal de travail .....	14
5.4	Manuel d'Installation .....	14
5.5	Manuel d'Utilisation.....	14
5.6	Archives du projet.....	14

# **1 Analyse préliminaire**

## **1.1 Introduction**

"Oublie-pas" est une application mobile, Réalisée dans le cadre de mon TPI, conçue pour aider les utilisateurs, en particulier ceux qui ont des difficultés d'organisation liées au TDAH, à définir et à suivre leurs objectifs quotidiens. L'application permettra de créer des listes d'objectifs personnalisables pour la journée et de définir des rappels à des heures spécifiques pour encourager la régularité et la discipline dans l'accomplissement des tâches.

## **1.2 Objectifs**

- L'application doit être codée en Kotlin.
- L'utilisateur devra pouvoir créer objectif et rappel, composé d'un titre et une description.
- L'utilisateur doit pouvoir choisir s'il veut un rappel ou non.
- Les objectifs doivent être catégorisés.
- Les catégories doivent être personnalisable.
- Les objectifs auront différents statuts en fonction de si le rappel est dépassé ou non.
- Les objectifs doivent être affiché sous forme de liste avec uniquement le titre et la date.
- Les objectifs terminés doivent être séparé par un menu.
- Le tri des objectifs doit pouvoir se faire selon :
  - La date du rappel
  - La date de création
  - Leur statut
- L'utilisateur doit pouvoir afficher le détail d'un objectif en cliquant dessus.
- Toutes les informations d'un objectif doivent pouvoir être modifiable par l'utilisateur
- L'utilisateur doit pouvoir supprimer un objectif
- La couleur et la police des objectifs doivent être affiché différemment selon la date de rappel.
- Les objectifs doivent être sauvegardé dans une base de données.

### 1.3 Planification initiale

Du a certains problèmes avec l'application Microsoft Project le nombre d'heures total affiché pour le projet TPI et la documentation est de 92 heure.

Voici les valeurs correctes :

Projet TPI : 90 heures

Documentation : 18 heures

1	✈	<b>Projet TPI</b>	<b>92 heures?</b>	<b>Mar 30.04.24</b>	<b>Jeu 30.05.24</b>
2	✈	<b>Analyse</b>	<b>18 heures</b>	<b>Mar 30.04.24</b>	<b>Ven 03.05.24</b>
3	✈	Introduction au projet	2 heures	Mar 30.04.24	Mar 30.04.24
4	✈	Rédaction objectifs	1 heure	Mar 30.04.24	Mar 30.04.24
5	✈	Diagramme de Gantt	4 heures	Mar 30.04.24	Mar 30.04.24
6	✈	Analyse Concept	2 heures	Jeu 02.05.24	Jeu 02.05.24
7	✈	Stratégie de test	2 heures	Jeu 02.05.24	Jeu 02.05.24
8	✈	Risques Techniques	1 heure	Jeu 02.05.24	Jeu 02.05.24
9	✈	Planification	1 heure	Jeu 02.05.24	Jeu 02.05.24
10	✈	Dossier de conception	1 heure	Jeu 02.05.24	Jeu 02.05.24
11	✈	<b>Implémentation</b>	<b>35 heures</b>	<b>Ven 03.05.24</b>	<b>Jeu 16.05.24</b>
12	✈	<b>Création des layouts</b>	<b>4 heures</b>	Ven 03.05.24	Ven 03.05.24
16	✈	Classe navigation	2 heures	Ven 03.05.24	Ven 03.05.24
17	✈	<b>Android Manifest</b>	<b>1 heure</b>	Lun 06.05.24	Lun 06.05.24
18	✈	ajout des scènes	0.5 heure	Lun 06.05.24	Lun 06.05.24
19	✈	paramètre et permission de l'application	0.5 heure	Lun 06.05.24	Lun 06.05.24
20	✈	<b>Classe RecycleurAdapter</b>	<b>2 heures</b>	<b>Lun 06.05.24</b>	<b>Lun 06.05.24</b>
21	✈	Rendre les boutons cliquable	1 heure	Lun 06.05.24	Lun 06.05.24
22	✈	Créer le layout du recycleur	1 heure	Lun 06.05.24	Lun 06.05.24
23	✈	ajouter le recycleur dans la classe home	2 heures	Mar 07.05.24	Mar 07.05.24
24	✈	Classe Alarm	4 heures	Mar 07.05.24	Mar 07.05.24
25	✈	Classe AlarmReceiver	4 heures	Lun 13.05.24	Lun 13.05.24
26	✈	Database Objectif	3 heures	Lun 13.05.24	Lun 13.05.24
27	✈	<b>Classe DetailsObjectif</b>	<b>6 heures</b>	<b>Mar 14.05.24</b>	<b>Mar 14.05.24</b>
28	✈	Implementation suppression d'objectif	3 heures	Mar 14.05.24	Mar 14.05.24
29	✈	Ajout modification des valeurs par l'utilisat	3 heures	Mar 14.05.24	Mar 14.05.24
30	✈	Implementation changement couleur de police	2 heures	Jeu 16.05.24	Jeu 16.05.24
31	✈	Try des objectifs interactif	3 heures	Jeu 16.05.24	Jeu 16.05.24
32	✈	<b>Tests</b>	<b>15 heures</b>	<b>Ven 17.05.24</b>	<b>Mar 21.05.24</b>
33	✈	Test sur differents appareils	7.75 heures	<b>Ven 17.05.24</b>	<b>Ven 17.05.24</b>
34	✈	Test selon les scénario de test préparé	4 heures	Ven 17.05.24	Mar 21.05.24
35	✈	Correction de bug	1 heure	Jeu 16.05.24	Ven 17.05.24
36	✈	<b>Documentation</b>	<b>92 heures?</b>	<b>Mar 30.04.24</b>	<b>Jeu 30.05.24</b>
37	✈?	Dossier de réalisation	5 heures		
38	✈?	Description des tests effectués	1 heure		
39	✈?	Résumé du rapport du TPI	2 heures		
40	✈?	Sources – Bibliographie	1 heure		
41	✈?	Journal de travail	4 heures		
42	✈?	Manuel d'Installation	0.5 heure		
43	✈?	Manuel d'Utilisation	0.5 heure		
44	✈?	Archives du projet	1 heure		
45	✈?	Conclusion	1 heure		
46	✈?	Correction orthographe	1 heure		
47	✈?	Mise en page	1 heure		

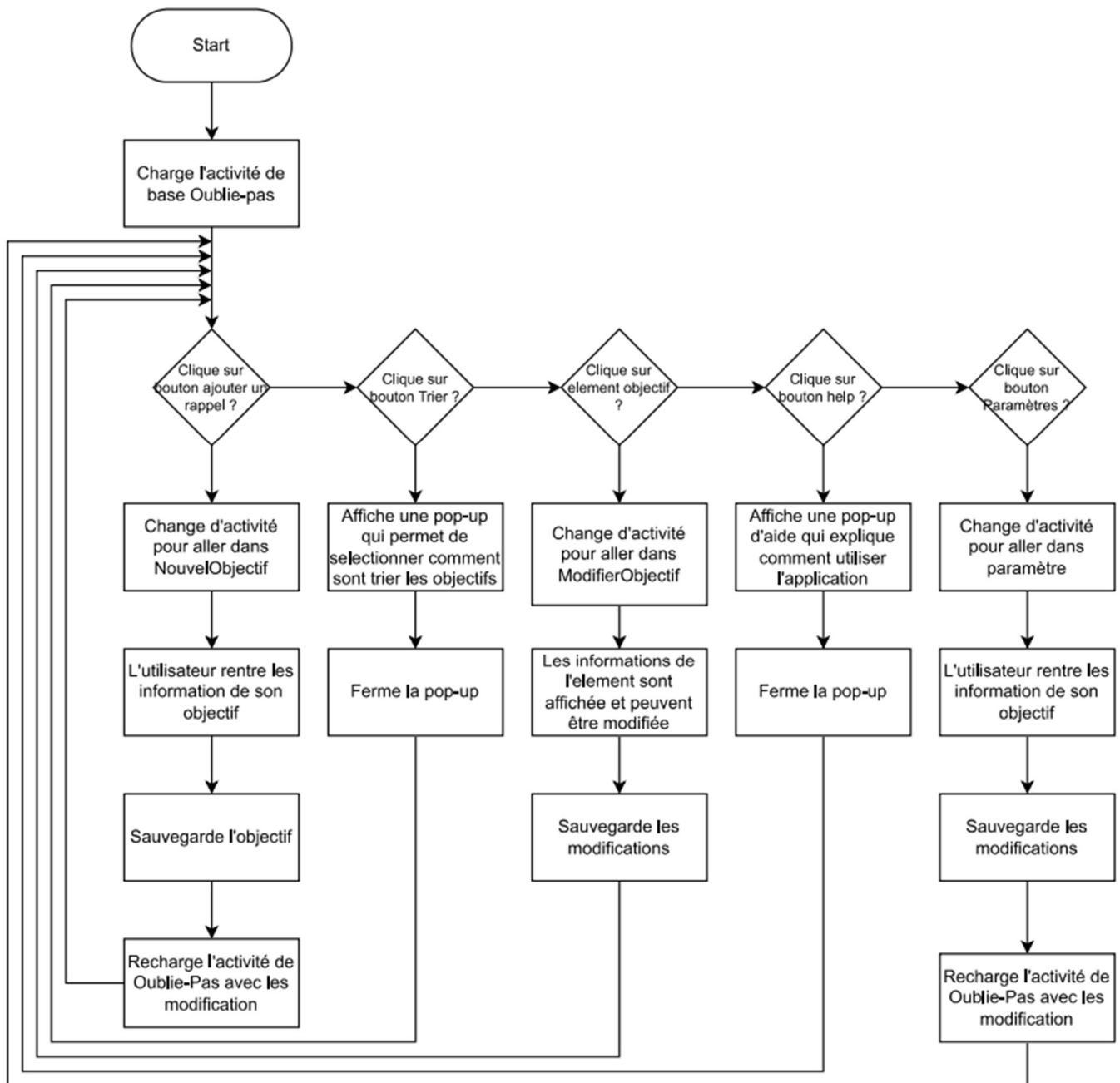
## 2 Analyse / Conception

### 2.1 Concept

The image displays four mobile application wireframes arranged in a 2x2 grid. Each wireframe is enclosed in a rounded rectangle with a black border and a small circle containing a question mark in the top right corner.

- Oublie-Pas:** Features a title bar with the text "Oublie-Pas". Below it is a "Trier" button. The main content area contains a list of 12 items, each with a "Titre" label, a date "1/1/2000", and a checkbox. At the bottom, there are two buttons: "Ajouter un rappel" and "Parametres".
- Nouvel Objectif:** Features a title bar with the text "Nouvel Objectif". Below it is a "Titre" label and a text input field. This is followed by a "Description" label and another text input field. Below these are two labels, "Rappel" and "Date/heure", each with a corresponding input field (a radio button for "Rappel" and a text field for "Date/heure"). Below these are six "Catégorie custom" labels, each with a checkbox. At the bottom is a "Sauvegarder" button.
- Paramètre:** Features a title bar with the text "Paramètre". Below it is a "Nouvelle catégorie" label and a text input field. This is followed by an "Ajouter la catégorie" button. Below this are ten "Catégorie custom" labels, each with a checkbox. At the bottom is a "Sauvegarder" button.
- Modifier objectif:** Features a title bar with the text "Modifier objectif". Below it is a "Titre" label and a text input field. This is followed by a "Description" label and another text input field. Below these are two labels, "Rappel" and "Date/heure", each with a corresponding input field (a radio button for "Rappel" and a text field for "Date/heure"). Below these are six "Catégorie custom" labels, each with a checkbox. At the bottom, there is a label "Date de création : 1/1/2000" and a "Sauvegarder" button.

Voici un diagramme qui montre dans les grandes lignes le déroulement d'une partie du point de vue d'un utilisateur.



## 2.2 Stratégie de test

La stratégie qui sera utilisée pour tester l'application.

L'application sera transmise au minimum a 5 personnes et demandé d'exécuter les actions ci-dessous puis me donner un retour

Ajouter un rappel	Va à l'écran Nouvel Objectif, l'utilisateur rentre les informations et met un rappel et le sauvegarde.	✓ X
Rappel sonne	Le rappel envoie une notification a l'heure demandée	✓ X
Cliquer sur le bouton	Passe la liste d'objectif sur un autre type de tri, date de rappel, date de création puis statut	✓ X
Cliquer un objectif	Passe sur Modifier objectif de l'objectif cliqué	✓ X
Modifier l'objectif	Modifie les informations et cliquer sur sauvegarder	✓ X
Supprimer un objectif	Cliquer sur un objectif, puis cliquer sur supprimer	✓ X
Cliquer sur bouton paramètre	Passe sur paramètre.	✓ X
Ajouter une catégorie	Rentre le nom de la catégorie que l'utilisateur veut créer, et clique sur le bouton ajouter la catégorie.	✓ X
Voir la nouvelle catégorie	La catégorie s'affiche dans la liste en dessous, s'affiche dans paramètre et dans modifier objectif	✓ X
Changer les paramètre	Clique sur les objectifs que l'utilisateur veut voir sur le menu principal puis cliquer sur sauvegarder	✓ X

## 2.3 Risques techniques

Durant la conception de la maquette du projet, j'ai été beaucoup plus ambitieux en ce qui concerne le design par rapport à mes projets précédents. J'ai cherché à créer une interface utilisateur attrayante et conviviale pour offrir une expérience agréable aux utilisateurs.

Une partie du développement de l'application me préoccupe également : l'ajout de catégories personnalisées. Cette fonctionnalité nécessite de repasser par toutes les étapes de la création de l'application, notamment la conception de trois mises en page (layouts), la création d'une base de données, ainsi que l'apprentissage de la programmation des boutons à cocher, une fonctionnalité avec laquelle je n'ai pas d'expérience préalable. L'intégration de toutes ces composantes représente un défi chronophage pour une fonctionnalité qui semble relativement simple.

## **2.4 Planification**

J'ai créé un Projet GitHub afin de faire la planification de mon projet.  
C'est pour moi la façon la plus pratique et efficient étant donné que c'est également là où je stocke toute mes données.

Vous pouvez trouver la version PDF entière en annexe ou directement sur GitHub via ce lien : <https://github.com/users/Bifmann/projects/2/views/1>

## **2.5 Dossier de conception**

### **2.5.1 Choix de l'IDE**

L'application sera intégralement codée depuis l'IDE Android studio Code.  
C'est l'IDE officiel de google pour le développement Android. Et prend en charge nativement Kotlin ce qui me donne accès a beaucoup de fonctionnalité intéressante pour notamment le débogage.

Android studio possède également un émulateur android inclus ce qui me permettra de tester Oublie-pas sur une multitude d'OS et format d'écran.

### **2.5.2 Base de données**

Oublie-pas utilisera non-pas plusieurs fichiers JSON comme spécifié dans le cahier des charges pour stocker les données persistantes de l'applications mais Room, une bibliothèque de persistance de données pour Android qui agit comme une couche d'abstraction sur SQLite.

Cela permet une gestion plus robuste, type-safe et moins sujette aux erreurs des données grâce à la vérification au moment de la compilation des requêtes SQL et l'intégration avec le code Kotlin.

Contrairement au stockage des données au format JSON, Room offre une meilleure performance pour les opérations de données complexes et garantit l'intégrité des données à travers des transactions et des requêtes optimisées, cela est également beaucoup plus simple à mettre en place et sera bien plus stable et fiable.

Seront stocké comme données persistantes :

- Les objectifs
- Les paramètres de l'application

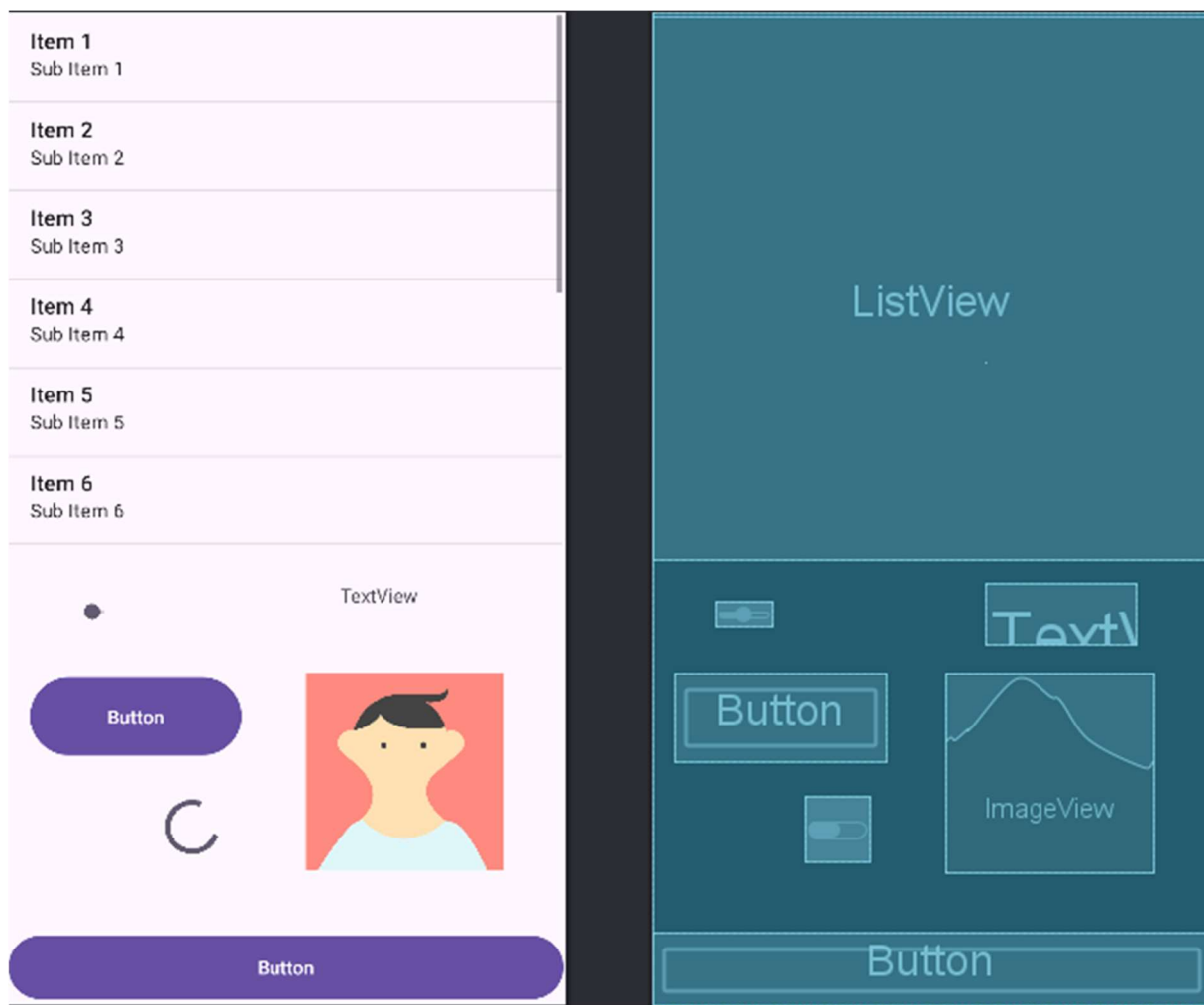


### 2.5.3 Maquette

Voir fichier annexe.

Les couleurs choisies et le format sera celui d'Android par défaut. En plus de me faire gagner du temps au développement, ce sont exactement les couleurs et simplicité que je recherche. Est de plus automatiquement adapté au dark mode.

Voici un Screenshot de plusieurs éléments d'Android studio.



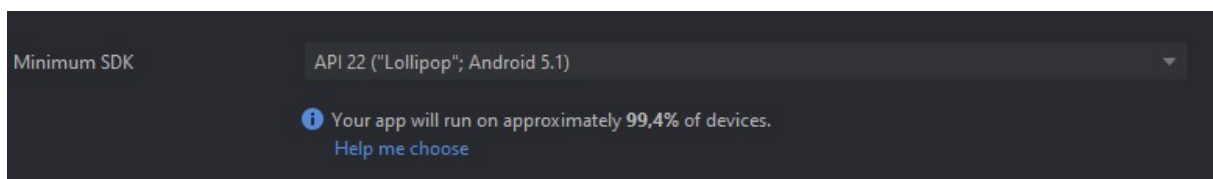
### 3 Réalisation

#### 3.1 Dossier de réalisation

##### 3.1.1 Version d'Android

Pour créer mon projet j'ai choisi comme SDK minimum l'api 21 qui correspond à Android 5.0 (Lollipop version).

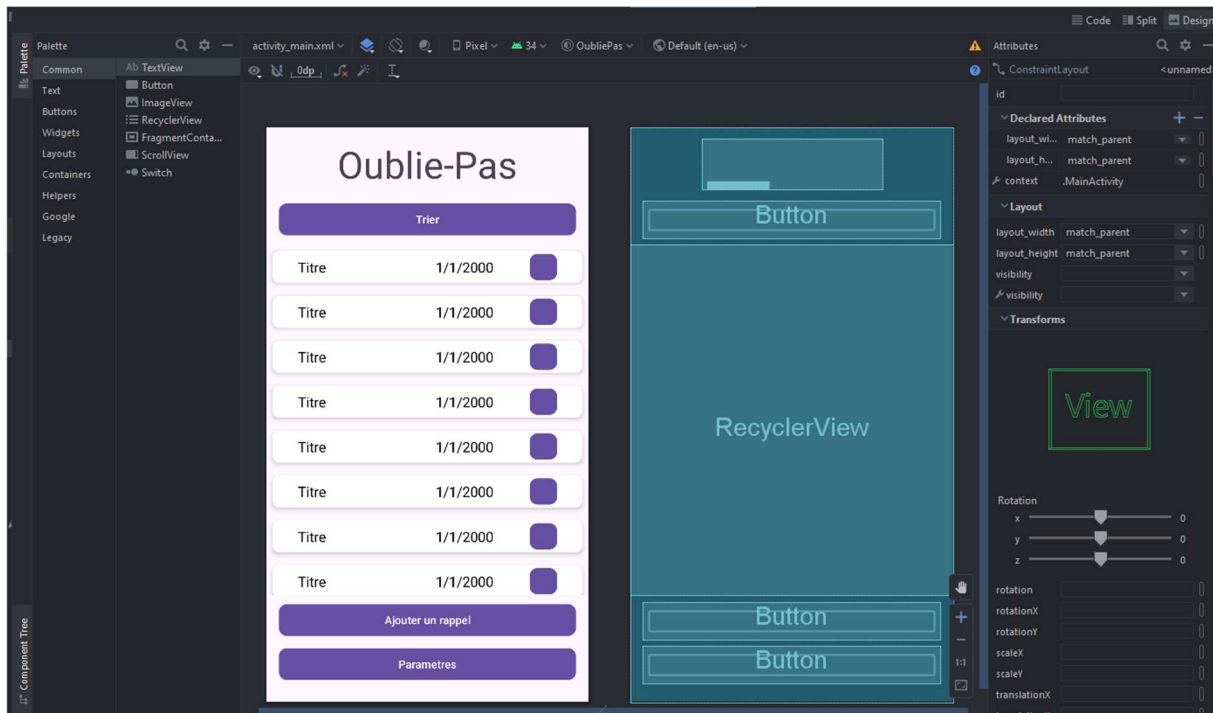
Cela a pour but que l'application puisse fonctionner sur le maximum de téléphones (99.4%).



##### 3.1.2 Création des layouts

J'ai choisi de commencer par directement m'occuper de toute la partie graphique de l'application en premier afin de ne plus avoir à y toucher au futur et fonctionner avec les mêmes bases partout dans mon projet.

J'ai donc reproduit toutes mes maquettes dans android studio, séparé en 4 fichiers ces « Activité » sont codé en XML grâce au mode « Design » d'Android studio et stockée dans le dossier « layout ».



### 3.1.3 Préparations des classes principales

Afin de pouvoir travailler fluidement par la suite du projet j'ai préparé dès le départ les classes de chaque activité, ainsi que la plupart des évènements de click.

Premièrement, les quatre classes « d'activité », c'est là-dedans que vas se trouver la logique propre de ce qui se passe dans chacune des activités. Par exemple `NouvelObjectif` appellera les fonctions de la base de données et rentrera les données des nouveaux objectifs.

- `MainActivity.kt`
- `ModifierObjectif.kt`
- `NouvelObjectif.kt`
- `Parametres.kt`

Ensuite j'ai créé les deux adaptateurs de mes recycleurs, Les adaptateurs servent à lier les données aux vues, en transformant des éléments de données en éléments d'interface utilisateur. Les recycleurs optimisent l'affichage des listes en recyclant les vues, ce qui permet de gérer efficacement de grandes collections de données. Les classes d'adaptateurs modifient les recycleurs par défaut d'Android pour afficher des listes de `CardView`.

- `RecycleurAdapter.kt`
- `RecycleurAdapterCategorie.kt`

Et finalement pour que l'utilisateur puisse naviguer entre les différentes scènes j'ai adapté et implémenté ma classe `NavigationHandler`, provenant d'un de mes projets précédent.

### 3.1.4 Ajout d'un système de notification.

Dans la réalisation du projet, un système d'alarme et de notification est nécessaire. Étant encore débutant dans ces fonctions particulières, j'ai créé un projet de test à part afin de suivre un tutoriel et de créer mes classes, réutilisables dans de multiples projets.

Le système de notification est composé de deux classes :

- `Notification.kt`
- `NotificationHelper.kt`

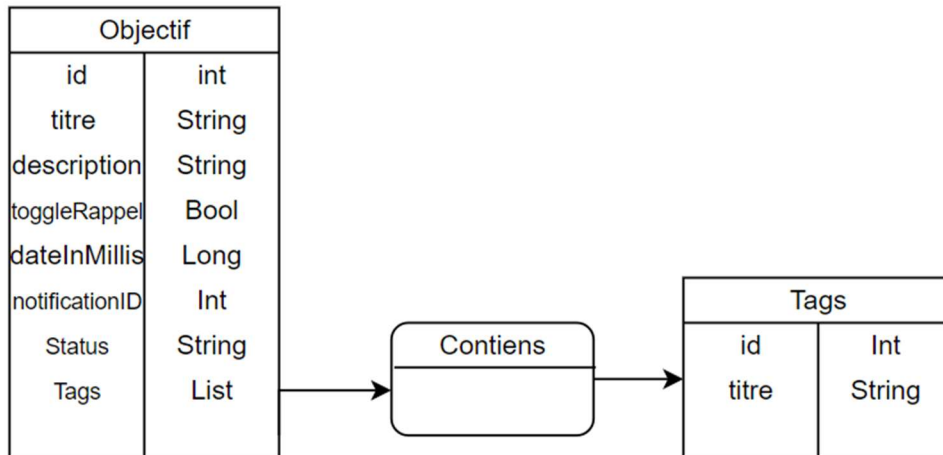
La classe `Notification` sert de cadre et va définir quel type de notification est créé ainsi que servir de `BroadcastReceiver`.

La classe `NotificationHelper` est là pour créer un canal de notification dans l'application et planifier, déplanifier les notifications à une heure donnée.

Avec ces deux classes, utilisable dans n'importe quel projet, je peux définir des notifications avec un titre, description à l'heure que je désire. Le design est également modulable au besoin en modifiant la classe notification.

### 3.1.5 Création d'une base de données.

N'ayant pas créer de MCD lors de l'analyse de mon projet j'ai commencé par en créer un rapidement a la demande de monsieur Ferrari.



**RoomData.kt** : Cette classe définit l'entité de la base de données avec les champs spécifiés.

**RoomDataDao.kt** : Cette interface définit les méthodes d'accès aux données.

**AppDatabase.kt** : Cette classe crée et gère l'instance de la base de données.

#### *Décrire la réalisation "physique" de votre projet*

- les répertoires où le logiciel est installé
- la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)
- les versions des systèmes d'exploitation et des outils logiciels
- la description exacte du matériel
- le numéro de version de votre produit !
- programmation et scripts: bibliothèques externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.

*NOTE : Evitez d'inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n'incluez que cette partie...*

### 3.2 **Description des tests effectués**

*Pour chaque partie testée de votre projet, il faut décrire:*

- *les conditions exactes de chaque test*
- *les preuves de test (papier ou fichier)*
- *tests sans preuve: fournir au moins une description*

### 3.3 **Erreurs restantes**

*S'il reste encore des erreurs:*

- *Description détaillée*
- *Conséquences sur l'utilisation du produit*
- *Actions envisagées ou possibles*

### 3.4 **Liste des documents fournis**

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

- *le rapport de projet*
- *le manuel d'Installation (en annexe)*
- *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
- *autres...*

## 4 **Conclusions**

*Développez en tous cas les points suivants:*

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

## **5 Annexes**

### **5.1 Résumé du rapport du TPI / version succincte de la documentation**

### **5.2 Sources – Bibliographie**

*Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)... Et de toutes les aides externes (noms)*

### **5.3 Journal de travail**

### **5.4 Manuel d'Installation**

### **5.5 Manuel d'Utilisation**

### **5.6 Archives du projet**