**1. Fonctionnalités principales**

**a. Gestion des tâches**

* Créer, modifier, supprimer des tâches.
* Ajouter des descriptions, des échéances, des priorités et des catégories.
* Marquer une tâche comme terminée ou en cours.
* Afficher une liste des tâches en cours et des tâches terminées.
* Tri des tâches par échéance, priorité, ou catégorie.

**b. Système de gamification**

* **Système de récompenses :**
  + Débloquer des badges en fonction du nombre de tâches complétées, du respect des échéances, etc.
  + Système de niveaux basé sur les points gagnés après avoir accompli des tâches (exemple : +10 points pour une tâche terminée).
  + Les tâches plus complexes ou urgentes rapportent plus de points.
* **Défis quotidiens/hebdomadaires :**
  + Proposer des défis pour inciter les utilisateurs à accomplir un certain nombre de tâches dans une période donnée.
* **Personnalisation de l'application :**
  + Débloquer des thèmes ou des couleurs personnalisées en fonction de la progression.

**c. Notifications**

* Notifications pour rappeler à l'utilisateur ses tâches à accomplir, en fonction des échéances.
* Alertes pour les défis ou missions spéciales à accomplir.

**d. Statistiques**

* Tableau de bord avec les statistiques de progression (tâches accomplies, points obtenus, niveau actuel, etc.).
* Graphiques montrant l'évolution des tâches terminées par jour/semaine.

**2. Spécifications techniques**

**a. Interface utilisateur (UI)**

* Utilisation de **RecyclerView** pour afficher les tâches sous forme de liste.
* Écran principal avec différentes sections : tâches à venir, tâches terminées, badges obtenus, progression.
* Utilisation de **Fragments** pour naviguer entre différentes vues (liste des tâches, tableau de bord, personnalisation).
* Intégration de thèmes personnalisables, avec des styles et couleurs déblocables.

**b. Base de données**

* Utilisation de **Room** (Android Architecture Components) pour stocker localement les tâches, points, niveaux, et badges.
* Structure de la base de données :
  + **Table des tâches** : ID, titre, description, échéance, priorité, état (terminée/en cours).
  + **Table des badges** : ID, nom, description, débloqué ou non.
  + **Table des niveaux** : ID, points nécessaires, niveau actuel.

**c. Gestion des récompenses**

* Créer une classe ou un service qui gère les points et l'attribution des badges.
* Conditions pour débloquer des badges (par exemple, terminer 10 tâches sans retard, compléter toutes les tâches d'une semaine, etc.).
* Mise à jour des points et du niveau après chaque tâche terminée.

**d. Notifications**

* Utilisation de **AlarmManager** ou **WorkManager** pour gérer les notifications.
* Notifications locales avec des rappels de tâches ou des félicitations lorsque l'utilisateur atteint un objectif ou débloque un badge.

**e. Architecture**

* Utilisation du **pattern MVVM** (Model-View-ViewModel) pour organiser le code :
  + **Model** : gestion des données et interactions avec la base de données.
  + **ViewModel** : logique métier et gestion des états de l'application.
  + **View** : affichage des données à l'utilisateur via les Fragments ou Activities.

**3. Améliorations futures**

* Synchronisation des tâches avec un service cloud (Firebase, par exemple) pour sauvegarder les données.
* Ajout d'une fonctionnalité de partage ou de compétition entre amis pour comparer les niveaux et badges.
* Intégration d'une barre de progression visuelle pour chaque tâche.

**4. Outils et bibliothèques**

* **Retrofit/Volley** pour les requêtes réseau (si nécessaire).
* **Glide** pour le chargement des images (badges, thèmes, etc.).
* **Material Design** pour créer une interface moderne et agréable.
* **Firebase** pour ajouter des services tels que les notifications push ou l'authentification (si nécessaire).