

Diagrammes UML et structures du projets

Ce diagramme UML illustre l'architecture globale de l'application **PC Manager**, en mettant en évidence les relations entre les principales classes, leurs attributs, et méthodes.

Résumé des éléments clés :

1. Entités principales :

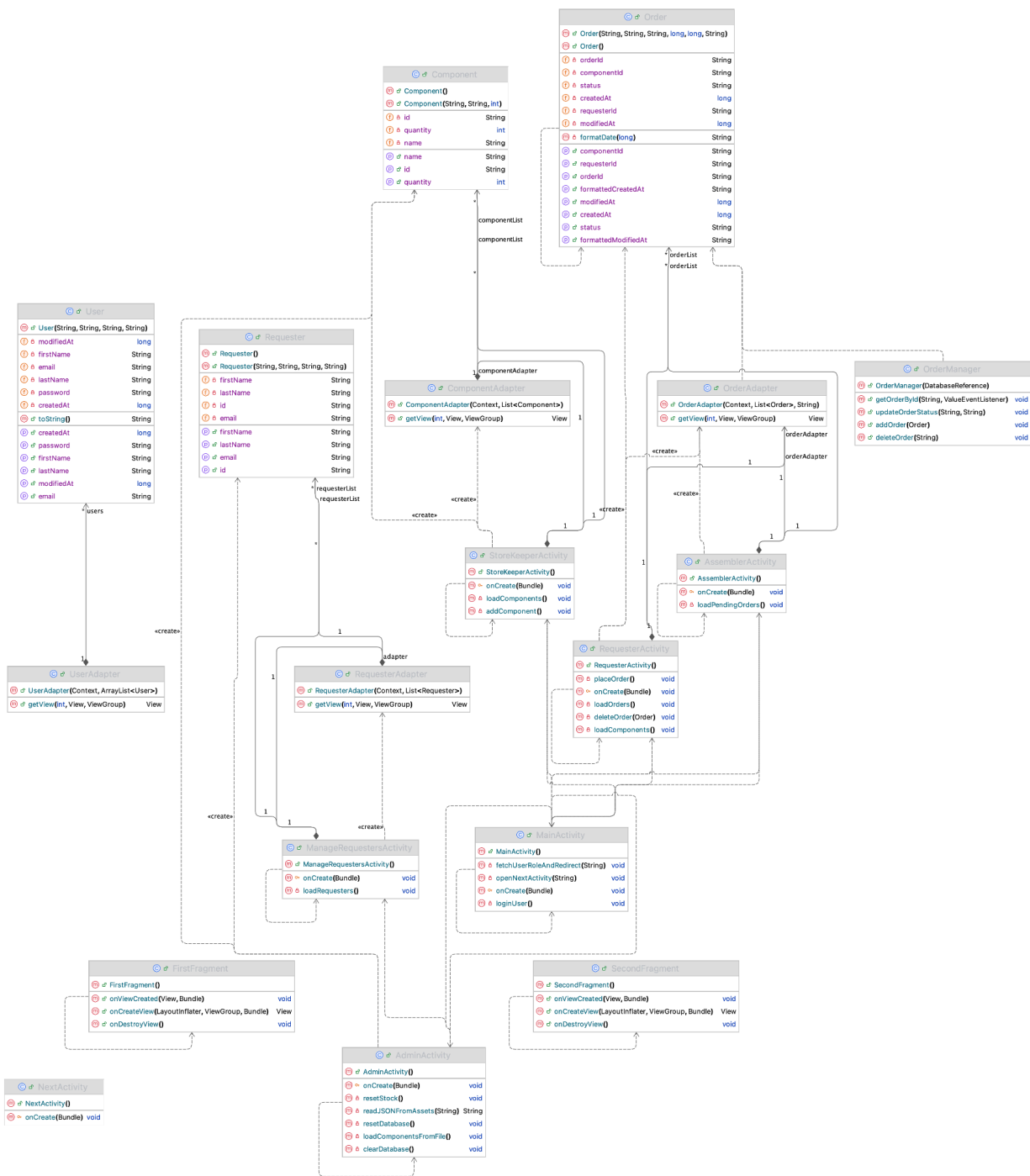
- **Order (Commande)** : Gère les informations sur les commandes (statut, composant, utilisateur).
- **Component (Composant)** : Représente les éléments en stock (nom, quantité).
- **User (Utilisateur)** : Inclut les rôles (admin, demandeur, assembleur).

2. Activités et rôles :

- **AdminActivity** : Gère les utilisateurs et les composants.
- **RequesterActivity** : Permet de passer des commandes.
- **AssemblerActivity** : Valide ou rejette les commandes selon le stock.

3. Interactions clés :

- Relié à Firebase pour la gestion en temps réel des données (commandes, composants, utilisateurs).
- Utilisation d'adaptateurs (**OrderAdapter**, **RequesterAdapter**) pour afficher dynamiquement les données.



Les diagrammes suivants représentent différents processus liés à la gestion des fonctionnalités dans l'application, en particulier les opérations réalisées par le rôle "AdminActivity". Voici une révision succincte de chaque diagramme :

- Ces diagrammes illustrent les interactions clés entre l'utilisateur, les différentes activités (notamment **AdminActivity**), et les composants de l'application pour la gestion des données. Ils mettent en évidence les processus principaux, tels que la réinitialisation de la base de données (**resetDatabase**), la réinitialisation des stocks (**resetStock**), et la suppression des données existantes (**clearDatabase**). Chaque processus suit une séquence claire : lecture des données depuis des fichiers JSON, traitement par **AdminActivity**, et mise à jour des nœuds Firebase correspondants (**Requesters**, **Components**, etc.). Ces diagrammes soulignent également l'organisation modulaire des fonctionnalités, facilitant leur maintenance et leur évolutivité.
-

Analyse générale

- Ces diagrammes mettent en évidence une séparation claire des responsabilités entre les différentes méthodes et entités.
- **AdminActivity** agit comme une passerelle principale pour gérer les opérations critiques sur Firebase.
- Les diagrammes détaillent les flux de données, permettant une compréhension approfondie du processus de gestion dans l'application.

