

Rapport TP Programmation Orientée Objet : Projet MeetingPro

GRIMM-KEMPF Matthieu
DAGON Lucas

26 mai 2025

Résumé

Ce rapport présente le développement du projet MeetingPro, réalisé dans le cadre du cours de Programmation Orientée Objet. Nous y exposons la démarche de conception, les choix techniques effectués, ainsi que les difficultés rencontrées et les résultats obtenus.

Table des matières

1	Méthodologie	2
2	Fonctionnement du système	3
2.1	Gestion des utilisateurs	3
2.2	Gestion des salles	4
3	Difficultés rencontrées	5
4	Résultats	5

1 Méthodologie

Le développement de notre application s’est appuyé sur une approche structurée et itérative, suivant les grands principes de la programmation orientée objet ainsi que du modèle MVC (Modèle-Vue-Contrôleur). Dès le début du projet, nous avons discuté afin de définir clairement la vision globale de l’application, son périmètre fonctionnel ainsi que la manière de répartir les responsabilités entre les membres de l’équipe.

Répartition des rôles

Nous avons opté pour une séparation logique entre le **frontend** (interface graphique et interactions utilisateur) et le **backend** (gestion des données et logique métier). Cette division du travail permettait de travailler en parallèle tout en limitant les conflits, et favorisait également une architecture propre et maintenable.

Choix techniques

Concernant l’interface graphique, le cahier des charges imposait l’utilisation de la bibliothèque **Tkinter** [3], qui, bien que rudimentaire, offre suffisamment de fonctionnalités pour réaliser une interface fonctionnelle. Pour la persistance des données, nous avons opté pour un système de fichiers au format **JSON** afin de sauvegarder les états des utilisateurs et des salles. Bien que le sujet préconisait un fichier unique, nous avons préféré adopter une approche à fichiers multiples, chaque entité ayant son propre fichier. Cette solution nous semblait plus adaptée à la sérialisation d’objets, et permettait une meilleure évolutivité du système.

Organisation du code

Avant d’entamer le développement, nous avons conçu une structure de projet inspirée de l’architecture MVC :

- **Modèle (Model)** : les classes métier représentant les entités (Person, Room, Conference, ComputerRoom).
- **Vue (View)** : l’interface graphique conçue avec Tkinter.
- **Contrôleur (Controller)** : les fonctions de gestion des interactions entre la Vue et le Modèle.

Cette structuration nous a permis d’avoir une séparation claire des responsabilités et de rendre notre application plus facile à tester, maintenir et faire évoluer.

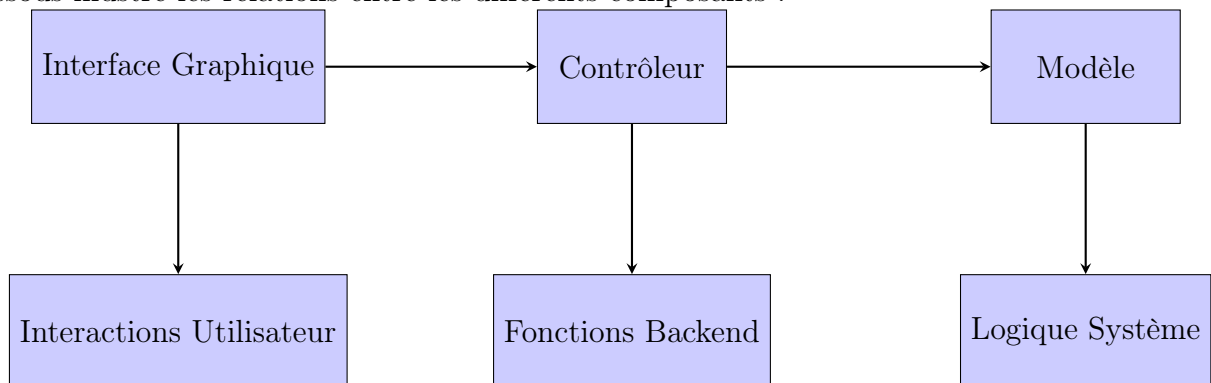
Cycle de développement

Nous avons suivi une démarche agile couplée avec le modèle V en découpant le projet en plusieurs phases :

1. **Spécification fonctionnelle** : définition des fonctionnalités clés.
2. **Conception UML et modélisation des classes**.
3. **Implémentation incrémentale** : développement progressif avec tests réguliers.
4. **Intégration et débogage**.
5. **Finalisation et documentation**.

2 Fonctionnement du système

Le système repose sur une architecture respectant le modèle MVC. Le schéma ci-dessous illustre les relations entre les différents composants :

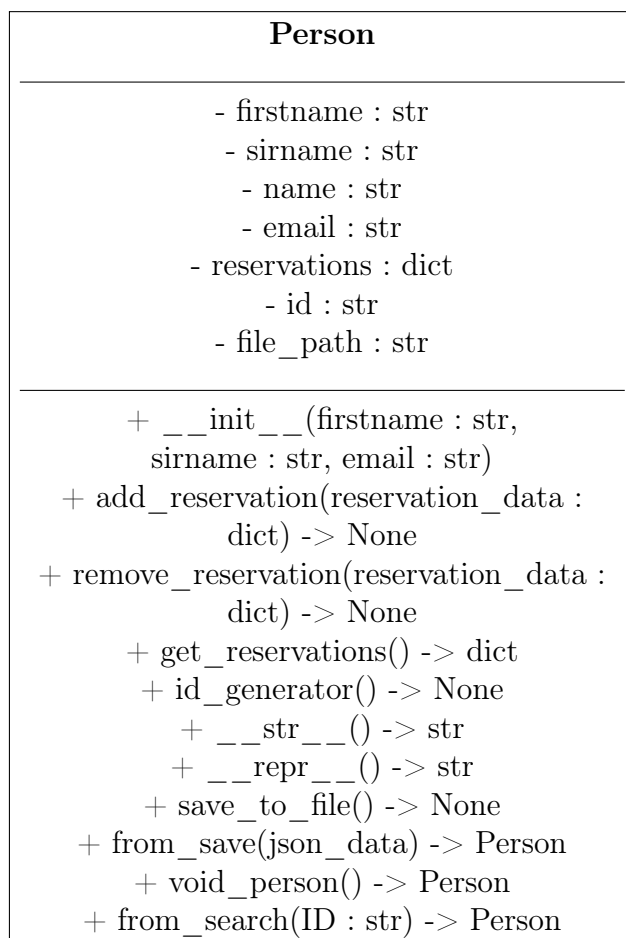


Deux groupes de classes principaux ont été définis :

2.1 Gestion des utilisateurs

La classe **Person** gère les informations des clients ainsi que leurs réservations. Elle permet l'ajout, la suppression et la consultation de réservations, ainsi que la sauvegarde dans un fichier JSON.

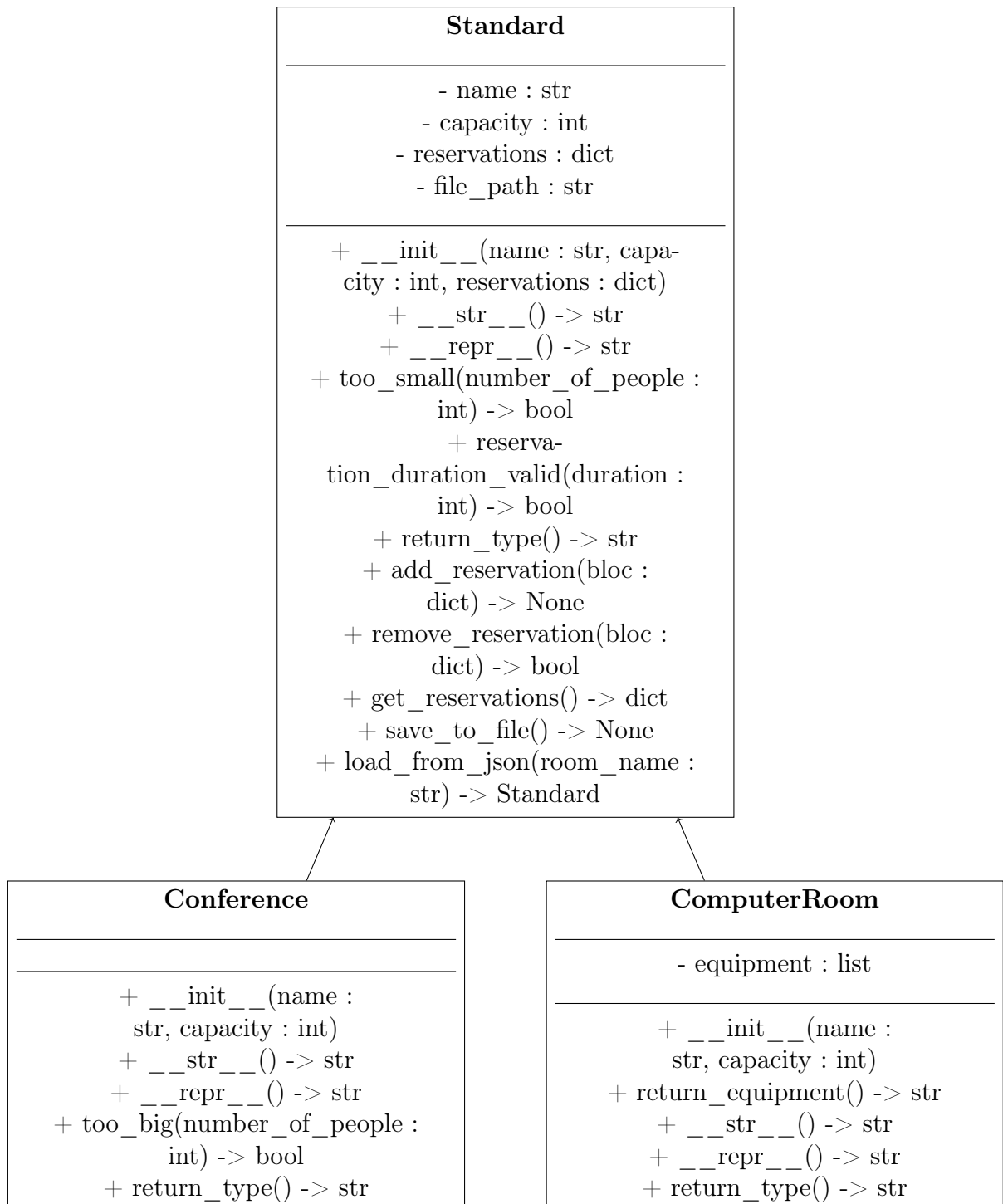
Une représentation de cette classe est la suivante :



2.2 Gestion des salles

La classe **Standard** définit les fonctionnalités communes à toutes les salles. Deux classes héritant de **Standard** spécialisent son comportement pour l'adapter aux types de salles également disponibles à la location : **Conference** vérifie que la salle n'est pas surdimensionnée pour l'événement, et **ComputerRoom** liste les équipements informatiques disponibles dans ce type de salle.

La représentation de ces classes et de leurs relations est la suivante :



3 Difficultés rencontrées

Plusieurs difficultés techniques et organisationnelles ont été rencontrées au cours du projet. Parmi celles-ci :

- Comprendre et maîtriser la bibliothèque `Tkinter` pour construire une interface graphique fonctionnelle et ergonomique.
- Gérer les erreurs lors de l'accès aux fichiers JSON (lecture, écriture, corruption).
- Mettre en place un système de sauvegarde robuste avec création dynamique de fichiers.
- Assurer une bonne coordination entre les différentes parties du code (classes, interface, contrôleur).

4 Résultats

Le projet MeetingPro permet aujourd'hui de :

- Créer, modifier et supprimer des utilisateurs.
- Ajouter et gérer des réservations de salles.
- Afficher dynamiquement les données relatives à chaque salle.
- Sauvegarder automatiquement les données sous forme de fichiers JSON séparés.

Le code est modulaire, extensible, et conforme aux principes de la programmation orientée objet.

Références

Références

- [1] Philipp ACSANY. « Working With JSON Data in Python ». In : *Real Python* (déc. 2024). URL : <https://realpython.com/python-json>.
- [2] L'Équipe ALPHORM et L'Équipe ALPHORM. « Maîtriser la manipulation JSON en Python ». In : *Blog Alphorm* (jan. 2025). URL : <https://blog.alphorm.com/maitriser-la-manipulation-json-en-python>.
- [3] *Graphical User Interfaces with Tk*. [Online ; accessed 19. May 2025]. Mai 2025. URL : <https://docs.python.org/3/library/tk.html>.
- [4] *json — JSON encoder and decoder*. [Online ; accessed 19. May 2025]. Mai 2025. URL : <https://docs.python.org/3/library/json.html>.
- [5] *SHA-256 - Hash SHA256 Mot de Passe - Décoder, Encoder en Ligne*. [Online ; accessed 19. May 2025]. Mai 2025. URL : <https://www.dcode.fr/hash-sha256>.
- [6] *Stack Overflow*. [Online ; accessed 19. May 2025]. Mai 2025. URL : <https://stackoverflow.com/questions/43562401/using-hashlib-sha256-to-create-a-unique-id-is-this-guaranteed-to-be-unique>.
- [7] *Stack Overflow*. [Online ; accessed 19. May 2025]. Mai 2025. URL : <https://stackoverflow.com/questions/16129652/accessing-json-elements>.
- [8] *Stack Overflow*. [Online ; accessed 19. May 2025]. Mai 2025. URL : <https://stackoverflow.com/questions/62527331/what-does-hexdigest-do-in-python>.
- [9] *W3Schools.com*. [Online ; accessed 19. May 2025]. Mai 2025. URL : https://www.w3schools.com/python/python_json.asp.