Mon retour d'expérience sur ce projet Java, en montrant mon implication, les difficultés rencontrées et ce que j'ai appris :

J'ai commencé par m'occuper de toute la partie interface utilisateur et de l'exécution globale du programme, car il fallait rendre la simulation utilisable simplement, sans refaire des modifications dans le code à chaque test.

Point d'entrée du projet – classe App

J'ai d'abord travaillé sur la classe App, qui est la porte d'entrée du programme.
C'est elle qui permet de :

- Choisir le mode interactif (--interactive)
- Sélectionner la stratégie (fcfs ou nearest)
- Choisir un fichier de configuration
- Lancer la simulation en mode automatique ou via le menu console.

Interface console – ConsoleUI

Ensuite, j'ai développé l'interface console complète, ce qui a été une grosse partie de mon travail.

J'ai ajouté :

- Un menu principal propre et structuré
- Un menu ascenseur permettant :
    - D'afficher les ascenseurs
    - D'ajouter un ascenseur (avec contrôles)
    - De modifier un ascenseur (avec saisies optionnelles)
    - De supprimer un ascenseur
    - D'annuler la dernière action (ajout/modification/suppression)
- Un menu stratégie (pour changer entre fcfs et nearest)
- La possibilité de lancer une simulation en direct
- L'historique des 5 dernières actions

Pour rendre tout ça robuste, j'ai dû gérer énormément de cas.

Gestion des saisies – ConsoleIO

J'ai créé une classe ConsoleO qui s'occupe des :

- lectures d'entiers dans un intervalle
- confirmations (o/n)
- effacement de l'écran
- "appuyer sur Entrée pour continuer"
- …

C'était pour réduire les erreurs liées à l'utilisateur.

Génération des rapports JSON

J'ai également réalisé la partie export JSON, qui était indispensable pour analyser les résultats du simulateur :

- JsonReportWriter : export des stats globales + toutes les requêtes
- ElevatorStopsJsonWriter : détails de chaque arrêt d'ascenseur
- ResidentsReportJsonWriter : historique complet des trajets de chaque résident

Cette partie m'a obligé à comprendre comment la simulation enregistrait les pickups, les dropoffs, les arrêts, etc.

Intégration avec GitHub

J'ai aussi participé à la mise en place du dépôt GitHub.
Pour être honnête, on a beaucoup travaillé à l'école, directement sur les PC, donc on n'a pas fait énormément de commits individuels au début.
On a utilisé GitHub surtout vers la fin, quand il a fallu rassembler nos trois parties. J'ai quand même appris à créer le dépôt, mettre en place le .gitignore etc… C'est clairement un point où je dois encore progresser, mais ce projet m'a fait avancer là-dessus aussi.

Difficultés rencontrées :

Au niveau de l'interface utilisateur :

Le plus compliqué a été de gérer proprement les erreurs de saisie : donc les utilisateurs qui tapent n'importe quoi etc …, éviter les NullPointerException dans les modifications et de vérifier les IDs existants / non existants.

C'est ce genre de détail "bête" qui fait perdre du temps mais qui est essentiel pour que le programme fonctionne avec toutes les conditions possibles

La partie annulation :

Mettre en place le système d'annulation a été dure, il fallait sauvegarder l'état d'avant, l'état après, et restaurer l'un ou l'autre selon le choix de l'utilisateur.

L'Intégration avec la simulation :

Il fallait faire attention à ne pas casser la simulation si on modifiait un ascenseur, bien recréer l'horloge (SimulationClock) à chaque lancement et utiliser la bonne stratégie selon les choix du menu

J'ai dû relire plusieurs fois la classe Simulation pour comprendre quand elle ouvre les portes, quand elle marque les pickups, etc.

Ce projet m'a beaucoup appris, d'un point de vue Java, j'ai énormément progressé sur l'organisation d'un projet multi-packages, les interfaces utilisateurs en console, les fichiers JSON et sur Maven et la génération d'un jar exécutable

D'un point de vue travail d'équipe, il a fallu répartir les tâches, fusionner les parties entre nous, discuter des choix, des noms de variables, des architectures, s'adapter aux contraintes du temps car on avait également d'autres (DS, TP).

Même si on aurait aimé ajouter plus de fonctionnalités ou une interface graphique, le résultat final est bien et fonctionnel.
C'était un projet enrichissant et qui m'a réellement aidé à progresser.