Mounir AMGHAR 20215351

Une image contenant capture d’écran, Police, Graphique, conception

Description générée automatiquementTim HRDY 20210085

Filière L2 Info

Université d’Evry Val d’Essonne

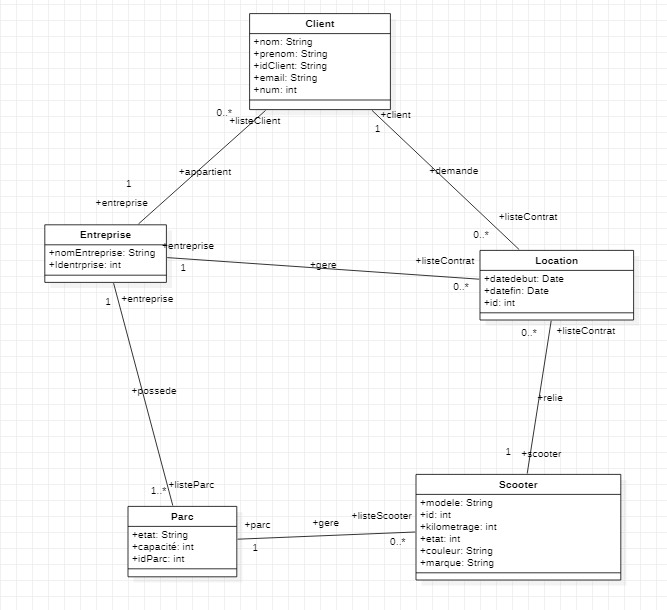
**Rapport : Gestion du parc de scooters pour le magasin "LOUSCOOT"**

**INTRODUCTION :**

Le projet « LOUSCOOT » est un logiciel de gestion de parc de scooters pour un commerce de location de scooters. Le logiciel permet de réaliser la location, la réception, la recherche et l'affichage de l'état des scooters dans le parc. Le programme offre un menu interactif permettant de choisir parmi différentes fonctionnalités citées ci-dessus ainsi que la saisie du parc de scooters et la possibilité de quitter le programme (tout en sauvegardant les données modifiées). Le présent rapport détaille la conception et la mise en œuvre de ce projet, ainsi que les problèmes rencontrés et les optimisations proposées.

**Mise en Œuvre du Projet :**

La mise en œuvre du projet a été réalisée sous 4 séances avec l’aide de notre chargé de TD, notamment pour la conception ainsi que la validation du diagramme UML que l’on a proposé. Les classes et méthodes demandées ont été implémentées une par une en respectant les spécifications fonctionnelles et les contraintes du projet données dans l’énoncé.



**Fonctionnalités implémentées :**

Les fonctionnalités suivantes ont été implémentées dans le projet :

* Location d'un scooter : L'utilisateur peut entrer le numéro d'identification d'un scooter à louer. Si le scooter existe et n'est pas déjà en location, il est marqué comme étant en location. Sinon, une erreur est signalée.

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

* Retour d'un scooter : L'utilisateur peut entrer le numéro d'identification d'un scooter à retourner. Si le scooter existe et est en location, l'utilisateur peut également entrer le nombre de kilomètres effectués avec le scooter, qui sera ajouté à son kilométrage. Le scooter est ensuite marqué comme disponible. Sinon, une erreur est signalée.

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

* Connaissance de l'état d'un scooter : L'utilisateur peut entrer le numéro d'identification d'un scooter pour connaître son modèle, son kilométrage et son état de location. Si le scooter n'existe pas dans le parc, une erreur est signalée.

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

* Affichage de l'état du parc de scooters : La méthode d'affichage dans PARC parcourt la liste des scooters et affiche le modèle, le numéro d'identification, le kilométrage et l'état de chaque scooter.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

* Saisie du parc de scooters : La méthode d’affichage dans PARC permet à l'utilisateur de visualiser le nombre total de Scooters, le nombre et ID des scooters loués et des scooters dispos ainsi que le kilométrage moyen.

Une image contenant texte, capture d’écran

Description générée automatiquement

* Quitter le programme : La condition de sortie dans PARC permet de quitter le programme en mettant fin à la boucle du menu interactif et en sauvegardant.

Une image contenant texte, Police, capture d’écran, nombre

Description générée automatiquement

**Problèmes rencontrés pendant la conception :**

Pendant la conception du projet, plusieurs problèmes ont été rencontrés et résolus pour garantir le bon fonctionnement du logiciel. Voici les principaux problèmes rencontrés et les solutions adoptées :

* Il a fallu décider de la structure de données appropriée pour stocker les informations sur les scooters, notamment leur modèle, leur numéro d'identification, leur kilométrage et leur état. Une solution a été d'utiliser une classe Scooter pour représenter un scooter avec les attributs appropriés, et de les stocker dans une structure telle qu’un Vecteur de scooters.
* Pour permettre la sauvegarde et la restauration des informations sur les scooters, un fichier csv a été utilisé pour stocker les données. Il a fallu mettre en place une logique pour lire les données à partir du fichier lors du démarrage du programme et pour écrire les données dans le fichier lors de la sauvegarde. Des manipulations de fichiers et de flux d'entrée/sortie ont été nécessaires pour implémenter cette fonctionnalité.
* Pour la fonction N°4 “état du parc” et N°5 « saisie du parc »dans la version interface graphique, comme un parc pouvait contenir plusieurs Scooter, utiliser des JLabels était impossible car il aurait fallu créer des centaines d’instances, pour cela nous avons eu recours à l’aide de nos chargés de TD qui nous ont mis sur la bonne voie, on a ainsi expérimenté les JList puis, nous nous sommes tournés vers les JTable pour une question de beauté ce qui nous a permis d’afficher nos données efficacement.

**CONCLUSION :**

Après des semaines de travail sur notre projet, nous pensons avoir réussi à implémenter toutes les fonctions demandées par l’énoncé et par notre prof. Cependant une interface un peu plus moderne pourrait clairement apporter un petit plus au projet, mais en vue de nos connaissances actuelles des interfaces graphiques, une telle tâche serait plutôt longue.