Mardi 17 décembre 2019

A picture containing object

Description automatically generated

Développement informatique avancé : Orienté Applications

-

Projet JAVA : Réalisation d’une application utilitaire

*« Application de gestion des parkings étudiants »*

Groupe 6

DE LEENER Nathan – GORTZ Gaëtan – LUK Brian

I) *Cahier des charges*

1. Contexte du projet :

Dans le cadre du cours « Développement informatique avancé :application (théorique) », les élèves de TI doivent créer une application web qui permet aux étudiants de l’Ephec de pouvoir réserver une place de parking durant un certain temps.

1. Description du projet :

L’application doit être capable de gérer le parking à disposition des étudiants. Chaque utilisateur doit créer un compte afin de pouvoir utiliser l’application et réserver une place de parking (nom utilisateur, mot de passe, plaque du véhicule, email, …etc.). Les étudiants auront aussi la possibilité de réserver une place pour une période déterminée et de signaler les utilisateurs qui enfreignent les règles (système de pénalité).

1. Objectifs du projet :

L’objectif est, pour les étudiants de l’Ephec, de pouvoir stationner leur véhicule en toute sécurité et de pouvoir trouver une place sans que les autres usagers entravent la circulation.

1. Contraintes techniques :
   * Développement en Java
   * MVC
   * L’interface de programmation JDBC
2. Besoins fonctionnels :

En tant qu’utilisateur, je souhaite pouvoir créer un compte et me logger afin que l’application sache qui je suis.

En tant qu’utilisateur, je souhaite avoir une interface graphique simple et intuitive permettant de réserver facilement les places disponibles.

En tant qu’utilisateur, à la fin de ma réservation, je souhaite pouvoir libérer une place afin que d’autre utilisateurs puissent en profiter.

En tant qu’utilisateur, je souhaite pouvoir signaler un utilisateur mal garé afin qu’il ne puisse plus avoir accès et ennuyer les autres utilisateurs.

A close up of text on a white background

Description automatically generated*II) Diagramme UML*

*III) Nos choix d’implémentation*

La réalisation de notre application utilitaire permettant de gérer les parkings étudiants, nous avons décidé d’utiliser une base de données. Cela nous semble être le choix le plus judicieux pour réaliser notre application, car pour mettre à jour nos données nous pouvons utilisé l’API « Java.SQL ».

Pour la mise en place de notre base de données, à défaut d’avoir accès à un serveur distant, nous avons utilisé Xampp. Ce qui nous a permis de créer un serveur local sur nos machines et ainsi de développer sans encombres notre application sans pour autant devoir avoir un serveur physique.

Ensuite, nous avons mis en place et développé notre base de données à l’aide de phpMyAdmin, ce qui nous a permis de coder cette dernière en y intégrant des données de bases, indispensable pour effectuer nos tests.

Concernant l’interface graphique, nous avons utilisé pour le layout « GridLayout » ce qui nous semble idéal pour réaliser un programme avec une fenêtre de connexion.

Pour passer d’une interface à une autre, au lieu de créer les frames quand on clique sur les boutons, nous avions opté pour que les frames soient toutes créées dès qu’on lance l’application et de révéler les frames grâce à la méthode setVisible() car il s’agissait de la méthode la plus simple à mettre en place.

*IV) Les difficultés que nous avons rencontrées*

Lors de la réalisation de ce projet, nous avons rencontré plusieurs difficultés…

Tout d’abord, nous avons éprouvé des difficultés à mettre en place le projet en se servant correctement de l’outil GitHub. Il a fallu quelques *commits* de test et une concertation d’équipe pour se lancer correctement dans le projet.

Ensuite, nous avons eu besoin de pas mal de concertation afin de se répartir au mieux les tâches à réalisées. Notamment, lorsqu’il s’agissait de savoir qui allait réaliser quelle partie du code. En effet, certaines parties étant liées à d’autres, il fallait que chacun puisse avancer sans dépendre des autres. Pour cela nous devions définir les lignes directrices de notre code, comme la manière de nommer nos variables, les conventions de codages que nous utilisions et bien entendu la structure de nos classes.

Pour cette étape, le diagramme UML nous a bien servi et nous a permis d’avoir une bonne vision globale.

Par la suite, nous avons rencontré certaines difficultés par rapport à l’élaboration de l’interface utilisateur, tant pour l’interface graphique que pour l’interface console. Notre problème étant que les deux interfaces étaient dépendantes l’une de l’autre. C’est-à-dire que nous tentions de faire fonctionner les deux en parallèle, ce qui nous retournait des erreurs lors de l’exécution. Nous avons compris finalement notre erreur et avons séparé les deux.

*V) Les pistes d’amélioration*

Nous avons de nombreuses pistes concernant les améliorations possibles à apporter à notre application.

Premièrement, nous pourrions améliorer l’interface graphique en la rendant plus attractive visuellement. Notamment en y ajoutant des image/icones pour les boutons, des listes déroulantes pour les parkings à réserver, des boutons supplémentaires pour effectuer des retours ou des déconnections. De plus, il faudrait ajouter un affichage indiquant l’état des réservations et le nombre de signalements de chaque utilisateur.

Deuxièmement, nous pourrions créer une accès administrateur afin de rajouter, supprimer ou modifier les parkings, places et utilisateurs sans devoir nécessairement modifier la base de donnée manuellement. Nous avons déjà créé certaines méthodes dans le code permettant d’agir en tant qu’administrateur.

Troisièmement, nous pourrions mettre la base de donnée dans un serveur distant afin que celle-ci soit accessible par quiconque souhaite utiliser l’application.

Quatrièmement, il faudrait réécrire certaines classes du code de manière plus propre, claire et lisible en essayant de garder une réelle uniformisation dans la manière de codée entre les différentes classes. Il faudrait aussi revoir sa structure afin que les différents composants du code soient plus structurés.

Cinquièmement, les messages d’erreur -par exemple lorsque l’on a trop d’infractions- n’apparait pour le moment que dans l’interface console et il serait judicieux de faire apparaître ceux-ci également dans l’interface graphique. De plus lorsqu’une entrée n’est pas valide, nous sommes pour le moment obligé de relancer le programme. Il faudrait plutôt afficher un message d’erreur annonçant l’erreur et permettant à l’utilisateur de ré-entrer les données erronées.

*VI) Conclusion personnelle des membres du groupe*

* Nathan DE LEENER  
  Pour conclure ce projet, je dirais qu’il m’a permis de prendre connaissance d’un ensemble de problématiques liées à la réalisation d’un projet de développement de plus grande ampleur que ce à quoi nous avions déjà été confronté. En devant travailler sur du code en parallèle avec d’autres personnes cela m’a montré la complexité et l’importance de définir une structure claire et des conventions précises de codage. Concernant mes apports durant le projet, je me suis occupé dans la partie codage de la réalisation des classes *Parking*, *Place* et une partie de la classe *User*, j’ai également retravaillé l’ensemble du code à la fin du projet pour essayer d’avoir une certaine uniformisation dans celui-ci. Et concernant la partiedocumentation, je me suis occupé de faire le cahier des charges, l’UML et le rapport.
* Gaëtan GORTZ  
  Pour conclure ce projet, après un petit rush pour finir dans les temps, je suis assez content de ce que nous avons pu produire bien que cela ne remplit pas tout ce que nous avions prévu lors du premier cahier des charges. Le fait de travailler en équipe oblige à avoir confiance en les membres, ce qui a été difficile pour moi qui voulait tout contrôler. Après quelques commits de chacun, je me rendu compte que je pouvais leur faire confiance et qu’ils respectaient ce qu’ils devaient faire. J’ai un peu travaillé à l’uniformisation du code, car Nathan et moi avions les mêmes conventions dû à notre passé au sein de l’UCL. Mais ma plus grosse contribution a été la base de données et ses liens avec notre programme.
* Brian LUK  
  Pour conclure, malgré le fait que j’ai commencé le projet sans aucune notion de java contrairement aux autres membres du groupe, j’ai essayé de rattraper mon retard et d’apporter le plus d’aide possible. J’ai pris connaissance de mes difficultés rencontrées et des concepts que je ne comprenais pas à mes débuts. Je me suis occupé de toute la partie MVC du projet, de la réalisation de la classe User et Offense et j’ai aidé pour faire la mise en commun entre l’interface graphique/console et la base de données. Pour le rapport, j’ai écrit les pistes d’amélioration que j’ai constaté et j’ai également aidé à écrire le cahier des charges.

Pour conclure, même si je n’ai pas travaillé autant que mes 2 collègues, je suis content d’avoir pu faire l’interface graphique et console et que le projet fonctionne.