

Barathon: rapport intermédiaire

Bourguignon Maxime
Jacobi Jordan
Simon Christophe
Vanneste Jean

23 octobre 2018

Introduction

Dans ce document, nous allons aborder plusieurs aspects de notre projet appelé « Barathon ». Ce rapport intermédiaire est composé de plusieurs parties qui sont :

- Description du projet ;
- Diagrammes réalisés ;
- Conventions de codage ;
- Critères de qualité.

Ce document n'est pas exhaustif mais est assez représentatif de l'idée que nous nous faisons de notre application.

1 Description du projet

Le projet « Barathon » doit permettre à un utilisateur de trouver facilement des bars à proximité et de se constituer un itinéraire. L'application est pensée pour tenir compte des préférences de l'utilisateur telles que les événements, le budget, sa position géographique, ... Elle est pensée pour être déployée en tant qu'application mobile.

Après la création d'un profil de préférences, l'utilisateur pourra se servir de notre algorithme pour trouver le parcours qui correspond le mieux à ses critères.

Pour l'aspect base de données, nous souhaitons utiliser une DB noSQL de type graphe (neo4j) qui s'intègre particulièrement bien avec notre projet et le langage de programmation imposé qu'est le java.

L'idée est de parcourir le graphe en comparant les critères de l'utilisateur et les caractéristiques des lieux pour retourner les plus adéquats.

Pour l'implémentation, nous allons utiliser le design pattern Strategy qui se prête bien aux deux types de recherches qui sont :

- Trouver un lieu à proximité ;
- Trouver un itinéraire parcourant différents établissements.

2 Diagrammes réalisés

Dans cette section, vous trouvez ci-dessous les deux diagrammes réalisés sur le projet.

2.1 Diagramme des cas d'utilisation

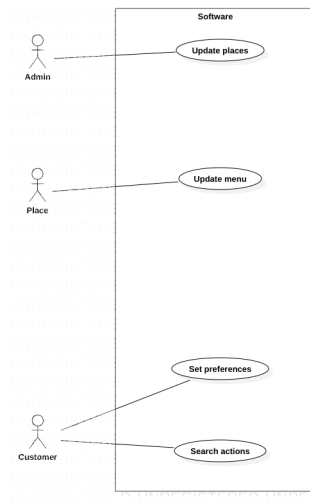


FIGURE 1 – Diagramme des cas d'utilisation du projet Barathon

2.2 Diagramme de classes

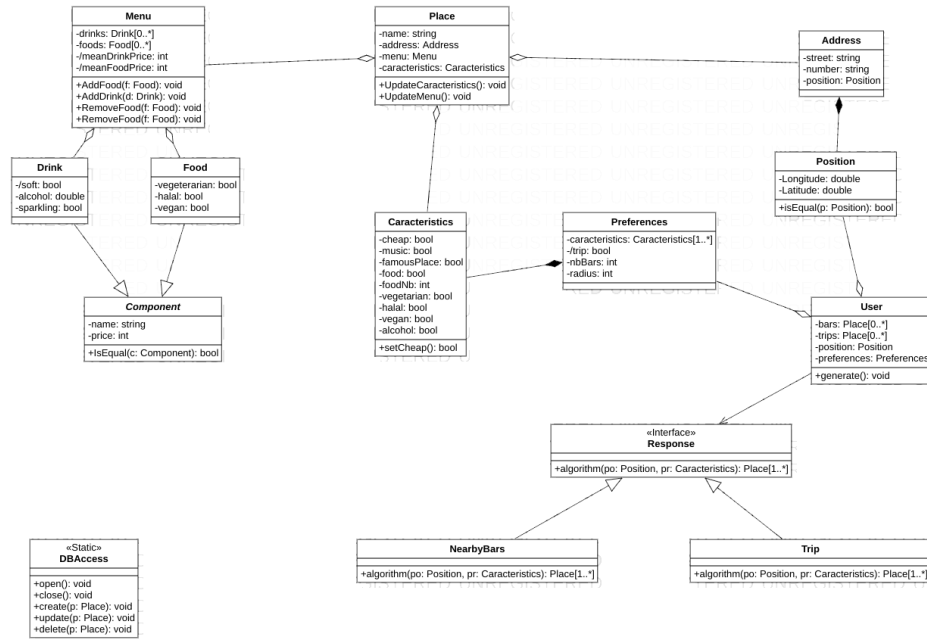


FIGURE 2 – Diagramme de classes du projet Barathon

3 Conventions de codage

Pour les conventions de codage, nous allons utiliser les conventions de codage standard de java. Cela nous permettra de vérifier facilement que notre code respecte celles-ci avec des applications tels que PMD ou Checkstyle.

Les conventions de codage standard de java que nous suivons sont celles-ci :

- Le nom des classes est en camelcase avec la première lettre en majuscule ;
- Le nom des variables est en camelcase avec la première lettre en minuscule ;
- Les variables d'une seule lettre sont utilisées localement ;
- Le nom des constantes est en majuscule avec un underscore pour séparer les mots ;
- Le nom des fichiers en minuscule avec des tirets pour séparer les noms ;
- L'indentation est de 4 espaces ;
- Les commentaires se trouvent au dessus des déclarations de classes.

Pour de plus amples informations, vous pouvez retrouver nos conventions de codage sur le site d'oracle : <https://www.oracle.com/technetwork/java/codeconvtoc-136057.html>.

4 Critères de qualité

Comme critères de qualité, nous avons choisi :

- Une densité de commentaires située entre vingt et trente pour cent ;
- Une couverture de code de quarante pour cent au minimum pour la partie « logique » ;
- Une rapidité d'exécution : exécution de l'algorithme de recherche d'itinéraire en moins de 30 secondes sur un téléphone mobile Android de dernière génération.

Il s'agit d'un travail que nous devons échanger. Pour faciliter la relève, nous voulons un programme qui est bien documenté, et qui est composé de fonctions fiables.

Pour notre projet, une application de recherche dans une database, nous voulons que celle-ci s'exécute vite. Cette vitesse va dépendre de la requête et du hardware, mais nous voulons optimiser celle-ci le plus possible.

Conclusion

Nous nous rendons déjà compte que choisir des critères de qualité n'est pas chose aisée.

Nous espérons proposer un projet intéressant pour nos futurs successeurs.