Partie 3 :

Le cas d’utilisation choisi pour faire le diagramme de classes et le diagramme d’interaction est la configuration du réseau d’arrosage.

Réponse partie 3 :

Le diagramme de concept est une décomposition de divers concepts associés à un problème. Ce diagramme représente le plus fidèlement possible le monde réel. On exclut tout concept/élément logiciel de ce diagramme. Dans ce diagramme, on tente de diviser au maximum les concepts.

Le diagramme de classe, quant à lui, représente les différents composants logiciels d’un problème. Ce diagramme peut aussi être vu comme une décomposition logique d’un problème. Une classe peut donc représenter plusieurs concepts différents pour suivre un modèle logique du point de vue logiciel.

Réponse partie 4 : GRASP

Polymorphisme :

La responsabilité de la classe AbsDispositifArrosage est de définir une interface pour les différentes sortes de dispositifs d’arrosages. En effet, cette interface définit la fonction arroser, qui sera implémentée par ses classes enfants et utilisées de façon polymorphique.

Diagram

Description automatically generated

Expert :

La responsabilité de ModuleControle est de définir tous les éléments de module de contrôle. Comme c’est la classe ModuleControle qui a toutes les informations nécessaires pour configurer le module, comme fuseauHoraire, heureLocale, latitude et longitude, alors on lui donne la responsabilité de le faire.

Text, letter

Description automatically generated

Faible couplage :

Une autre responsabilité de ModuleControle est de contrôler les parcelles qui lui sont associées. Donc, comme ModuleControle et Parcelle ont déjà un lien entre eux, nous avons connecté Parcelle et ModuleControle sans lier Parcelle à ReseauArrosage pour diminuer le couplage entre les classes. Ceci suit les principes du patron Faible Couplage.

Diagram

Description automatically generated

Contrôleur :

La responsabilité de ReseauArrosage est de détecter l’accès de l’utilisateur et de lui permettre de configurer le réseau d’arrosage. Il agit donc d’intermédiaire entre l’utilisateur et le reste du système. Le seul événement du système, soit l’accès de l’utilisateur, est reçue par cette classe et traitée par les autres classes, comme ModuleControle. Ceci suit donc les principes du patron Contrôleur.

Diagram

Description automatically generated