E4D : ÉTUDE DE CAS

Durée : 5 heures **Coefficient : 5**

CAS CRAB

Ce sujet comporte 15 pages dont 7 pages d'annexes.
Il est composé de quatre dossiers indépendants.
Le candidat est invité à vérifier qu'il est en possession d'un sujet complet.

Matériels et documents autorisés

Lexique SQL sans commentaire ni exemple d'utilisation des instructions.
Règle à dessiner les symboles informatiques.

Tous les types de calculatrice sont INTERDITS pour cette épreuve.

Liste des annexes

Barème

Dossier 1 : Gestion des contrats et des rechargements	**25 points**
Dossier 2 : Interconnexion des sites des stations et du siège central	**15 points**
Dossier 3 : Maintenance corrective et préventive des bornes	**42 points**
Dossier 4 : Suivi de projet	**18 points**

Total : 100 points

Présentation du contexte

Promulguée le 12 juillet 2010, la loi portant engagement national pour l'environnement, dite « Grenelle 2 », est un texte d'application et de territorialisation du Grenelle Environnement et de la loi Grenelle 1. Un des chantiers de cette loi porte sur le développement des véhicules électriques et hybrides rechargeables, en favorisant l'émergence de l'offre industrielle nationale, en stimulant la demande et en encourageant la possibilité de créer et d'entretenir des infrastructures de recharge électrique nécessaires à l'usage de ces véhicules.

Bien que la plupart des infrastructures de recharge va relever de la sphère privée (90%), les bornes de recharge accessibles au public, placées dans des parkings ou sur voirie, offriront l'assurance aux utilisateurs de pouvoir y accéder en dehors de cette sphère privée (domicile, travail) et des stations services. Elles constituent un gage de fiabilité de l'ensemble du système, complément indispensable pour encourager l'utilisation du véhicule électrique.

Les communes sont naturellement impliquées dans le déploiement de ces bornes, en raison du fort impact sur la voirie et les places de stationnement.

La ville de R. fait partie des douze agglomérations pilotes appelées à déployer une première vague d'infrastructures de recharge pour véhicules hybrides et électriques.

Elle a confié à la société prestataire de services Chargéon la mise en place et l'exploitation d'un réseau de points de recharge sous forme de bornes intelligentes standardisées. Soucieuse de se forger une expérience solide sur ce marché émergent, la société Chargéon fait évoluer son système d'information, colonne vertébrale permettant de réaliser les principales opérations nécessaires au bon fonctionnement des différents sous-systèmes de l'infrastructure de recharge.

En tant que développeur d'applications, vous participez aux différentes missions liées à ce projet baptisé CRAB – Chargement Rapide Automatisé de Batteries.

Dossier 1	**Gestion des contrats et des rechargements**

L'accès aux bornes de recharge par un usager est sujet à la souscription d'un contrat de recharge. Un portail *web* accessible aux usagers doit être mis en place pour faciliter le suivi de leur consommation.

Les bornes de recharge électrique

Les bornes de recharge sont implantées dans des stations situées dans les parkings et sur la voirie de la ville. Chaque station accueille une ou plusieurs bornes et est localisée par ses coordonnées GPS (latitude et longitude) et l'adresse de la rue dans laquelle elle est située. Sont également mémorisées la date de mise en service de chaque borne et sa dernière date de révision.

Ces bornes de recharge de batteries sont préférentiellement du type « recharge normale » (puissance de 3 kW ou kilowatt) ou « semi-rapide » (puissance de 24 kW), mais certaines sont de type « recharge rapide » (puissance de 50 kW).

Les contrats de recharge

Afin de pouvoir utiliser les bornes de recharge mises à disposition par la mairie de R., tout possesseur d'un véhicule électrique doit souscrire un contrat de recharge référençant ce véhicule auprès des services municipaux. Les informations à renseigner sont les suivantes : nom, prénom, téléphones fixe et mobile, adresse postale, adresse de courriel, numéro d'immatriculation de la voiture, date du contrat. Il est également important de connaître le modèle de la batterie du véhicule (référence, capacité et fabricant). En effet, si toutes les batteries supportent le type de charge « normal », certaines ne supportent pas la charge « rapide », voire « semi-rapide ». Un usager peut bien entendu posséder plusieurs contrats. Pour chaque contrat, une carte magnétique est délivrée, cette carte permettra de se connecter aux bornes de recharge.

Les bornes de recharge comptabilisent l'énergie délivrée en kilowatt-heure (KWh).

Deux formules de contrat ont été retenues :

- Le forfait prépayé : l'usager règle à l'avance un certain nombre de KWh. Les opérations de rechargement de batterie sont autorisées tant que le solde restant de KWh n'est pas épuisé. L'usager peut à tout moment racheter des KWh.
- L'abonnement : l'usager choisit une durée d'abonnement qui lui permettra de réaliser au cours de cette période autant d'opérations de rechargement que nécessaire. On mémorisera les dates de début et de fin d'abonnement. L'usager peut renouveler son abonnement, ce qui a pour effet de repousser la date de fin.

Les opérations de rechargement

Après avoir immobilisé son véhicule, l'usager présente sa carte magnétique auprès du lecteur de carte de la borne. Cette lecture permet alors d'identifier le contrat de recharge et son propriétaire.

L'usager déroule le câble de son chargeur de batterie et le connecte à la prise de la borne prévue à cet effet. Le système détermine alors le nombre de KWh nécessaires et informe l'usager du temps de rechargement.

Plusieurs contrôles sont alors effectués :

- la borne détecte automatiquement les caractéristiques de la batterie du véhicule, qui doit correspondre au modèle déclaré dans le contrat ;
- la borne vérifie que la batterie connectée supporte bien le type de charge qu'elle délivre (« normal », « semi-rapide » ou « rapide ») ;
- la borne s'assure de la validité des dates du contrat dans le cas d'un abonnement ou du solde de KWh restant dans le cas de la formule prépayée.

L'échec d'un de ces contrôles bloque le rechargement ; cet échec est alors enregistré en regard de l'opération de rechargement en cours. À des fins statistiques, la cause de l'échec doit être mémorisée.

Si tous les contrôles sont positifs, le rechargement peut commencer.

À l'issue de l'opération, l'heure de début et de fin du rechargement ainsi que le nombre de KWh délivrés sont enregistrés.

Travail à faire	
1.1	Élaborer un schéma conceptuel des données concernant la gestion des contrats de recharge et le suivi des rechargements.

Dossier 2	**Interconnexion des sites des stations et du siège central**

Documents à utiliser : annexes 1A et 1B

Les bornes de recharge présentes dans les stations doivent accéder en consultation et mise à jour aux informations stockées sur les serveurs d'applications hébergés dans les locaux de la société Chargéon.

Les bornes d'une station sont reliées par liaison filaire à un routeur-modem GPRS – *General Packet Radio Service* -, ce dernier permettant de joindre les locaux de la société Chargéon (cf. annexe 1A).

Le routeur-modem GPRS offre la possibilité de constituer un réseau privé virtuel (RPV appelé aussi VPN : *Virtual Private Network*). On parle de RPV lorsqu'un organisme interconnecte ses sites via une infrastructure partagée telle qu'Internet.

Le RPV correspond à une interconnexion de réseaux locaux via une technique de "tunnel" : le protocole de «tunnelisation» encapsule les données à transmettre, données qui sont elles-mêmes chiffrées.

Travail à faire	
2.1	Indiquer l'objectif principal de sécurité visé par le RPV lors des échanges de données sur Internet.

La table de routage du « routeur station RPV » est la suivante :

Destination réseau	Masque réseau	Adresse de passerelle	Adresse IP de l'interface du routeur
192.168.10.0	255.255.255.240	192.168.10.1	192.168.10.1
192.168.1.0	255.255.255.0	192.168.5.2	192.168.5.1

Travail à faire	
2.2	Expliquer par un calcul la valeur du masque réseau 255.255.255.240 de la première ligne de la table de routage du « routeur station RPV » présentée ci-dessus.
2.3	Indiquer l'adresse IP de la passerelle des bornes de la station.
2.4	Par l'observation de la table de routage du « routeur station RPV », montrer que l’accès aux bornes de la station est limité aux hôtes de la société Chargéon.

Un extrait de la table de filtrage actuelle du pare-feu situé dans les locaux de *Chargéon* est présenté ci-dessous :

N°	Interface Entrée	Interface Sortie	Adresse Source	Port Source	Adresse destination	Port Destination	Action
10	192.168.0.2	192.168.1.254	192.168.10.0/28	Tous	192.168.1.10	53	Accepter
20	192.168.1.254	192.168.0.2	192.168.1.10	53	192.168.10.0/28	Tous	Accepter

Remarque : les règles sont numérotées de 10 en 10 de manière à ce que l'insertion d'une nouvelle règle soit aisée.

L'algorithme utilisé par le service de filtrage est décrit ci-dessous.

Pour chaque paquet à traiter :

- en suivant l'ordre des règles de 1 à n, rechercher la première règle applicable ;
- si une des règles est applicable, alors appliquer l'action au paquet et arrêter le parcours de la table ;
- si aucune règle n'est applicable, refuser le paquet.

L'annexe 1B recense l'association de quelques services réseau avec leurs numéros de port standards.

Travail à faire	
2.5	Indiquer le rôle des règles numéro 10 et 20.

Le logiciel embarqué sur les bornes de la station fait appel à des services *web* situés sur le serveur d'applications d'adresse IP 192.168.1.30. Ces services *web* sont invoqués à l'aide de requêtes HTTP GET.

Travail à faire	
2.6	Écrire les deux règles à ajouter pour permettre aux bornes de la station d'invoquer les services *web* hébergés sur le serveur d'applications.

Dossier 3	Maintenance corrective et préventive des bornes

Première partie : Gestion des interventions de maintenance corrective

Afin d'assurer la qualité de service attendues par les usagers, il s'agit d'optimiser la gestion des pannes pouvant survenir dans les infrastructures de recharge nécessaires à l'usage des véhicules électriques.

Toutes les interventions de maintenance corrective se font à la suite d'un incident déclaré.

Pour faciliter la planification des interventions, les types d'incidents les plus courants sont répertoriés dans la base de données avec une indication du temps de réparation prévu pour ce genre de panne. Une fois l'intervention réalisée par le technicien, l'incident est clôturé.

Il est nécessaire de conserver un historique des opérations dans une base accessible par tous les intervenants de maintenance. Cette base permet d'effectuer différentes statistiques aussi bien sur la fiabilité des bornes que sur la réactivité de la maintenance, dont l'analyse permettra d'augmenter la performance du système.

Voici un extrait du schéma relationnel de la base de données pour la gestion de la maintenance préventive.

TECHNICIEN (id, nom, prénom)
- Clé primaire : id

STATION (id, nom, adresseRue, coordLat, coordLong)
- Clé primaire : id

BORNE (id, état, dateMiseEnService, dateDernièreRévision, idStation)
- Clé primaire : id
- Clé étrangère : idStation en référence à id de STATION
- *Le champ état prend la valeur « HS » ou « ES » selon que la borne est hors service ou en service. Elle est hors service dès qu'un incident est en cours de traitement sur cette borne.*

TYPEINCIDENT (id, description, tempsRéparationPrévu)
- Clé primaire : id
- *tempsRéparationPrévu est exprimé en nombre de minutes.*

INCIDENT (id, remarques, dateHeure, dateHeureClôture, idBorne, idType)
- Clé primaire : id
- Clé étrangère : idBorne en référence à id de BORNE
- Clé étrangère : idType en référence à id de TYPEINCIDENT
- *Le champ dateHeureClôture est renseigné lorsque l'incident est clôturé.*
- *Le champ idType n'est pas renseigné si le type de l'incident n'est pas un incident déjà connu.*

INTERVENTION (id, dateHeureDébut, dateHeureFin, idIncident, idTechnicien)
- Clé primaire : id
- Clé étrangère : idIncident en référence à id de INCIDENT
- Clé étrangère : idTechnicien en référence à id de TECHNICIEN

On dispose de la fonction *DATEDIFF (partie_date, dateDebut, dateFin*) qui renvoie l'intervalle entre deux dates, où le paramètre *partie_date* spécifie la partie de date dans laquelle l'intervalle doit être mesuré (*year* | *quarter* | *month* | *week* | *day* | *hour* | *minute* | *second* | *millisecond*). Le résultat est un entier signé égal à (dateFin - dateDebut) exprimé en parties de date.

Travail à faire	
3.1	Rédiger les requêtes SQL permettant d'obtenir les résultats suivants : A) Liste par ordre alphabétique des noms et prénoms des techniciens ayant réalisé une intervention sur la borne d'identifiant B1. B) Liste des interventions démarrées plus de 24 heures après l'incident (identifiant de l'intervention, remarques, date et heure de l'incident, date et heure de début de l'intervention). C) Nombre d'incidents non clôturés à ce jour. D) Liste des stations (nom) ayant eu plus de dix incidents.

Seconde partie : Gestion de la maintenance préventive

Documents à utiliser : annexes 2A, 2B et 2C

Les techniciens doivent également mener des actions de maintenance préventive sur les bornes de recharge. Ces révisions sont fonction du type de borne. Elles sont programmées à intervalles de temps réguliers, mais peuvent aussi être déclenchées lors de l'atteinte d'un seuil d'utilisation.

La solution informatique doit permettre chaque mois de répartir équitablement les tâches de maintenance préventive sur l'ensemble des techniciens. Cette partie de l'application doit être réalisée à l'aide d'un langage orienté objet. Un extrait du diagramme de classes utilisé est présenté en ***annexe 2A***, leur description littérale en ***annexe 2B***. L'ensemble des objets est instancié à partir de la base de données dès le lancement de l'application.

La classe « Maintenance » est chargée de programmer les visites de stations à réaliser dans le mois, puis de les affecter aux techniciens. Une visite concerne une station, et précise la durée totale nécessaire pour réaliser les révisions sur les bornes.

La répartition des visites aux différents techniciens doit être équitable en affectant chaque nouvelle visite au technicien actuellement le moins occupé en temps total de maintenance préventive.

Les classes techniques « Collection » et « Date » sont présentées en ***annexe 2C***.

Travail à faire	
3.2	Écrire la méthode « getDuréeRévision » de la classe « Borne ».
3.3	Écrire la méthode « estARéviser » de la classe « Borne ».
3.4	Écrire le constructeur de la classe « Visite ».
3.5	Écrire la méthode « getVisiteAFaire » de la classe « Station ».
3.6	Écrire la méthode « affecterVisites » de la classe « Maintenance ». *Le candidat est libre d'ajouter et d'écrire toutes les méthodes (non citées dans les annexes) qu'il juge nécessaires.*

Dossier 4	Suivi de projet

Documents à utiliser : Annexe 3a

Vous êtes chargé(e), en l'absence temporaire du chef de projet, d'assurer le suivi du projet CRAB et de mettre à jour les éléments de planification en fonction des événements qui vous sont soumis.

Vous disposez des éléments suivants sur le projet :

- la planification du projet (cf. annexe 3) comprenant les tâches assignées aux différents métiers : maquettiste (M), chef de projet (CP), développeur (D), architecte (A) ;
- les règles de gestion de projet à respecter :
  - la phase « Recette fonctionnelle » ne peut intervenir qu'à l'issue de la réalisation des développements ;
  - les phases « Formation » et « Déploiement » ne peuvent intervenir qu'après la recette fonctionnelle ;
  - règles de criticité d'utilisation des ressources développeur :
    - seuil normal : 50 % ;
    - seuil critique : 75 % ;
- les données concernant les ressources humaines :
  - effectif des développeurs : 4 ;
  - coût / jour d'un développeur : 400 € ;
  - taux d'occupation des ressources (cf. annexe 3).

Travail à faire	
4.1	Indiquer en jours la durée du projet CRAB.
4.2	Calculer le taux d'occupation mensuel des ressources développeur pour les mois de juin, juillet et août. Indiquer ce qu'on peut en conclure.

Dans la phase « Réalisation des développements », la tâche « Sécurité » prend du retard alors qu'elle doit être réalisée avant la tâche « Simulations (tests techniques) » de cette même phase.

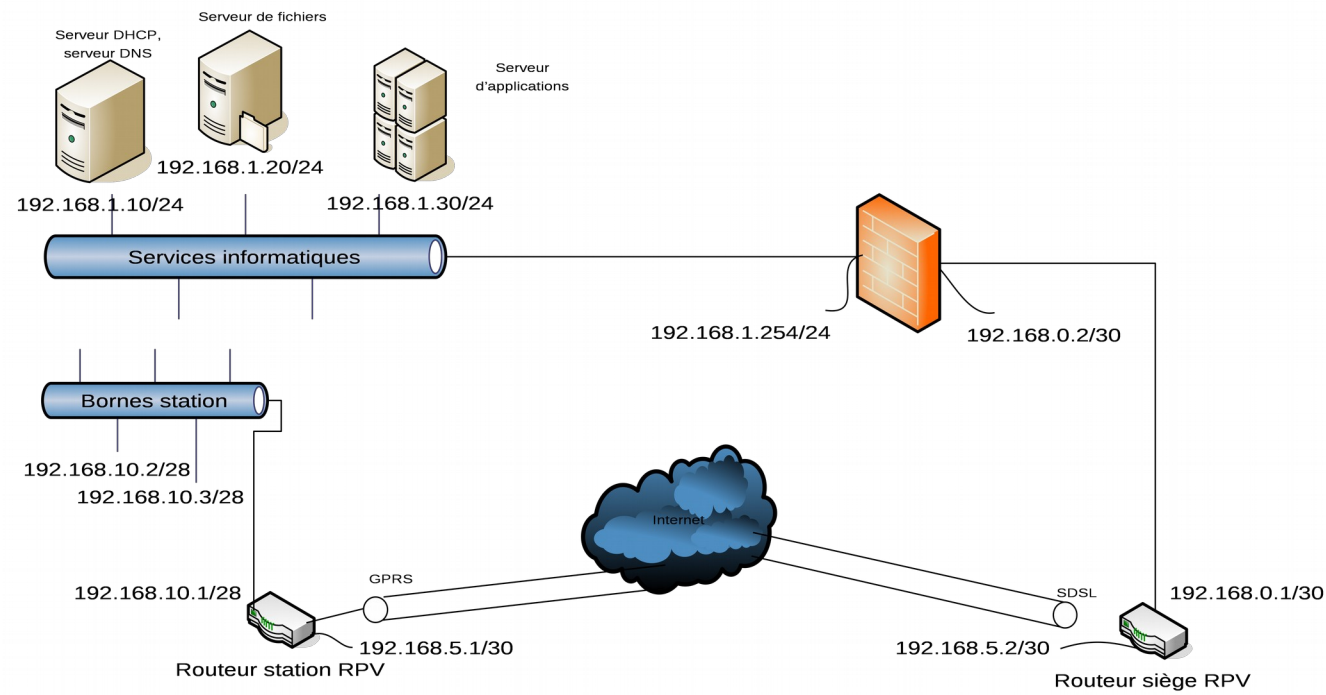
L'estimation de la charge initiale semble en effet irréaliste ; il est fort probable que cette tâche « Sécurité » s'alourdisse de 20 jours-hommes (JH). La prise en compte de cette surcharge sera faite à compter du 15 juillet.

Travail à faire	
4.3	Indiquer l'impact de ce changement sur la planification du projet si un seul développeur est affecté à la tâche « Sécurité » de la phase « Réalisation des développements » en justifiant la réponse.

Pour des raisons contractuelles, la durée totale du projet ne doit pas être modifiée.

Travail à faire	
4.4	Indiquer à qui le chef de projet peut confier la tâche « Sécurité » de la phase « Réalisation des développements » compte tenu du plan de charge des développeurs en respectant cette contrainte contractuelle de délai. Indiquer l'impact de ce changement sur le taux d'occupation mensuel des ressources développeur.
4.5	Calculer l'impact de ce changement sur le coût du projet.

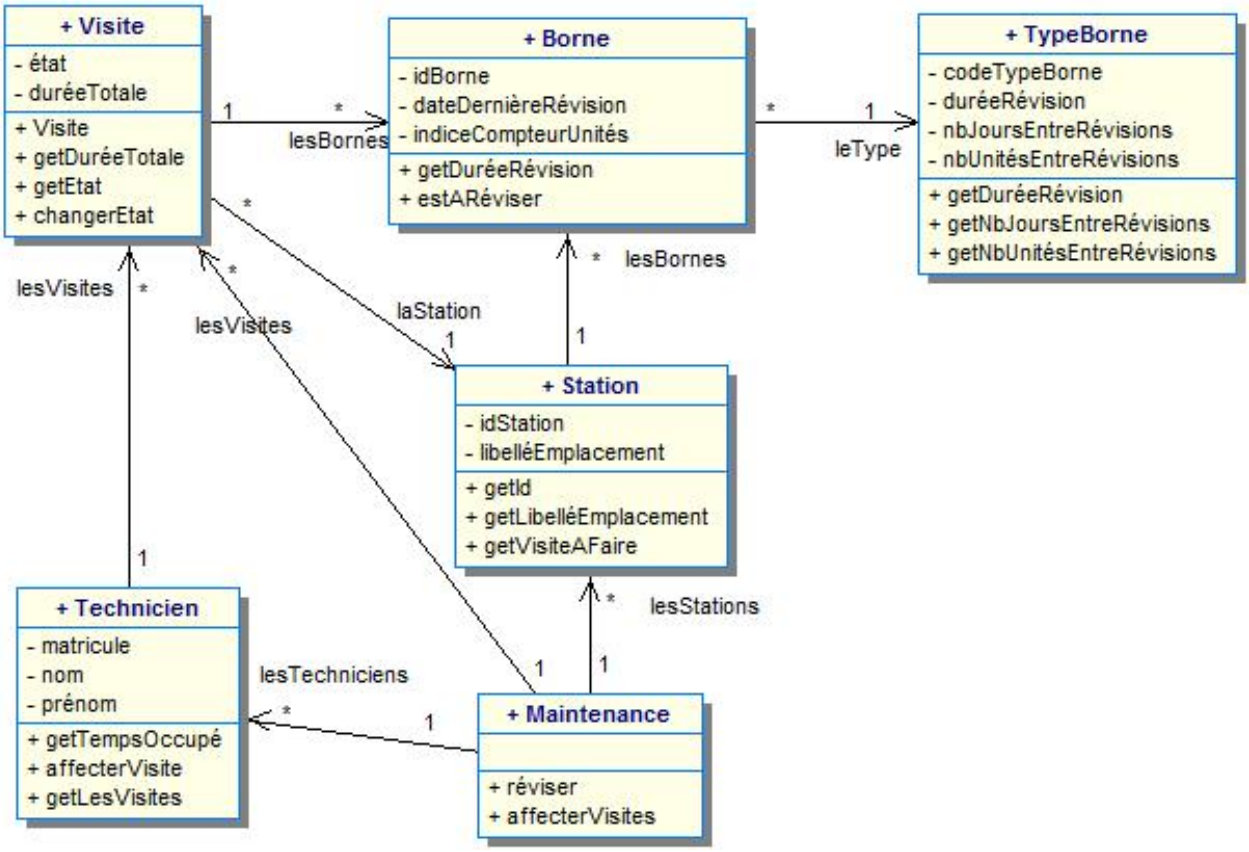
Annexe 1A – Extrait du schéma du réseau



Annexe 1B – Association des noms et numéros de ports de quelques services réseau

File Transfer Protocol	21
Telnet	23
Simple Mail Transfer Protocol	25
Domain Name Server	53
Oracle Sql*Net	66
Web HTTP	80

Annexe 2A – Diagramme partiel des classes métiers



<u>Remarque</u> : Les paramètres des méthodes ne sont pas présentés sur ce diagramme.

Annexe 2B – Description littérale des classes métiers

Classe Station

Attributs privés :

idStation : Entier *// identifiant de la station*

libelléEmplacement : Chaîne *// libellé de l'emplacement de la station*

lesBornes : Collection de Borne *// les bornes de la station*

Méthodes publiques :

Fonction getId() : Entier *// retourne l'identifiant de la station*

Fonction getLibelléEmplacement() : Chaîne *// retourne le libellé de l'emplacement*

Fonction getVisiteAFaire() : Visite

// retourne une instance de classe Visite recensant toutes les bornes de la station

// qui nécessitent d'être révisées, ou null s'il n'y a aucune borne à réviser

FinClasse

Classe TypeBorne

Attributs privés :

codeTypeBorne : Chaîne *// code du type de borne*

duréeRévision : Entier *// durée en minutes requise pour réaliser*
// la révision sur les bornes de ce type

nbJoursEntreRévisions : Entier
// nombre de jours qui séparent deux révisions successives d'une borne de ce type

nbUnitésEntreRévisions : Entier
// nombre d'unités de recharge au-delà duquel il faut envisager une nouvelle révision

Méthodes publiques :

Fonction getDuréeRévision() : Entier
// retourne la durée en minutes requise pour réaliser la révision sur les bornes de ce type

Fonction getNbJoursEntreRévisions() : Entier
// retourne le nombre de jours au-delà duquel il faut envisager une révision
// sur les bornes de ce type

Fonction getNbUnitésEntreRévisions() : Entier
// retourne le nombre d'unités de recharge au-delà duquel il faut envisager une révision
// sur les bornes de ce type

FinClasse

Classe Borne

Attributs privés :

idBorne : Entier *// identifiant de la borne*

dateDernièreRévision : Date *// date de la dernière révision effectuée sur la borne*

indiceCompteurUnités : Entier
// nombre d'unités de recharge délivrées depuis la dernière révision,
// ce compteur étant remis à zéro suite à chaque révision

leType : TypeBorne *// type de la borne*

Méthodes publiques :

Fonction getDuréeRévision() : Entier
// retourne la durée en minutes requise pour réaliser la révision sur la borne,
// cette durée étant fonction du type de la borne

Fonction estARéviser() : Booléen
// retourne vrai lorsque la borne doit être révisée, soit parce que le temps qui sépare
// deux révisions pour ce type de borne s'est écoulé depuis la date de la dernière révision,
// soit parce que le nombre d'unités de recharge délivrées par la borne
// depuis la dernière révision a atteint le seuil établi pour ce type de borne ;
// retourne faux sinon

FinClasse

Classe Visite

Attributs privés :

état : Caractère
// état de la visite : 'p' pour programmée, 'a' pour affectée, 'r' pour réalisée

duréeTotale : Entier
// durée totale en minutes requise pour réaliser l'ensemble des révisions
// prévues sur les bornes de la station

laStation : Station
// la station concernée par la visite

lesBornes : Collection de Borne
// la collection des bornes de laStation concernées par la visite

Méthodes publiques :

Visite(lesBornesAVisiter : Collection de Borne, uneStation : Station)
// ce constructeur valorise tous les attributs privés de la classe Visite, y compris l'état et la
// durée totale de la visite

Fonction getDuréeTotale() : Entier
// retourne la durée totale en minutes requise pour réaliser l'ensemble
// des révisions prévues sur les bornes de la station

Fonction getEtat() : Caractère
// retourne l'état de la visite

Procédure changerEtat()
// modifie l'état de la visite, de 'p' programmée à 'a' affectée, ou de 'a' affectée à 'r' réalisée

FinClasse

Classe Technicien

Attributs privés :

matricule : Entier — *// matricule du technicien*
nom : Chaîne — *// nom du technicien*
prénom : Chaîne — *// prénom du technicien*
lesVisites : Collection de Visite — *// ensemble des visites affectées au technicien*

Méthodes publiques :

Fonction getTempsOccupé() : Entier
// retourne la durée totale en minutes des visites affectées au technicien

Procédure affecterVisite(uneVisite : Visite)
// ajoute la visite uneVisite dans les visites affectées au technicien

Fonction getLesVisites() : Collection de Visite
// retourne l'ensemble des visites affectées au technicien

FinClasse

Classe Maintenance

Attributs privés :

lesStations : Collection de Station *// l'ensemble des stations*

lesTechniciens : Collection de Technicien *// l'ensemble des techniciens*

lesVisites : Collection de Visite *// l'ensemble des visites à réaliser*

Méthodes publiques :

Procédure réviser()

// Etablit l'ensemble des visites à réaliser sur les stations

Procédure affecterVisites()

// Affecte les visites à réaliser aux techniciens, en répartissant équitablement le travail
// entre les techniciens. Chaque visite est affectée au technicien le moins occupé en temps
// total de maintenance préventive. L'état de chaque visite doit alors être mis à jour.

FinClasse

Annexe 2C – Description des classes techniques

Classe Collection de <nom de la classe>

Méthodes publiques

Fonction cardinal() : Entier

// Renvoie le nombre d'objets de la collection

Fonction obtenirObjet(unIndex : Entier) : Objet de la classe

// Retourne l'objet d'index unIndex, le premier objet de la collection a pour index 1

Procédure ajouter(unObjet : Objet de la classe)

// Ajoute un objet à la collection

FinClasse

Pour instancier une collection :

uneCollection : Collection de <classe>

uneCollection ← new Collection() de <classe>

Pour parcourir par itération les éléments d'un objet Collection, il est possible d'utiliser :

Pour chaque <objet> dans <collection> faire

// instructions avec <objet>

FinPour

Classe Date

Attributs privés :

année : Entier

mois : Entier

jour : Entier

Méthode publique à portée classe :

fonction aujourdhui() : Date *// renvoie la date du jour*

// Exemple d'appel : an ← Date.aujourdhui().année()

// la variable entière an reçoit l'année de la date du jour

Méthodes publiques :

fonction année() : Entier *// renvoie l'année*

fonction mois() : Entier *// renvoie le mois*

fonction jour() : Entier *// renvoie le jour*

fonction différence(uneDate : Date) : Entier

// renvoie le nombre de jours de différence entre l'objet Date courant et le paramètre uneDate
// si l'objet Date courant correspond à une date postérieure au paramètre uneDate,
// le nombre de jours retourné est positif.
// Dans le cas contraire, le nombre de jours retourné est négatif.

Fin Classe Date

Annexe 3 – Planification du projet CRAB et taux d'occupation des ressources développeur

Légende :

1 mois = 30 jours délai, soit 20 Jours/Homme (J/H)

Echelle :

1 mois
0,5 mois

PROJET CRAB

TACHES	Assignée A	Coût en €	Charge en JH	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre
Conception / Maquettage		**38 000**	**80**									
Conception / Maquettage	M1,M2	30 000	60									
Documentation	D1	8 000	20									
Validation des charges	CP	**8 000**	**10**									
Conception de l'architecture technique	A1,A2	**84 000**	**140**									
Conception des développements		**56 000**	**140**									
Ecrans	D1,D4	16 000	40									
Contrôles	D1,D4	8 000	20									
Sécurité	D1,D4	16 000	40									
Simulations (Tests techniques)	D3	16 000	40									
Réalisation des développements		**44 000**	**110**									
Ecrans	D2	4 000	10									
Contrôles	D2,D3	8 000	20									
Sécurité	D2,D3	8 000	20									
Simulations (Tests techniques)	D1,D2	24 000	60									
Recette fonctionnelle	CP,D4	**24 000**	**40**									
Formation utilisateurs	D1,D2	**8 000**	**20**									
Déploiement	D1,D2	**8 000**	**20**									
Pilotage	CP	**20 000**										
		290 000										

Plan de charge des développeurs D1, D2, D3, D4

Développeur affecté à une tâche

	Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre
D4									
D3									
D2									
D1									

Taux d'occupation mensuel des ressources (développeurs)

Février	Mars	Avril	Mai	Juin	Juillet	Août	Septembre	Octobre
	12,50%	12,50%	62,50%				50,00%	37,50%