|  |  |
| --- | --- |
|  | Chapitre 1**Présentation du cadre du projet** |

***Introduction***

Dans ce chapitre, nous présenterons le cadre générale du projet et JBA expertise, la société hôte de notre projet de fin d’études. Et nous ferons une étude et une critique de l’existant qui ont menée a la naissance de la solution à proposer dans le cadre de projet.

***I. Cadre générale du projet***

Ce travail s’inscrit dans le cadre du projet de fin d’études à l’Institut Supérieur des Etudes Technologiques de Bizerte (ISET). Il vient compléter notre formation universitaire acquise au sein de cet établissement, en vue de nous introduire dans la vie professionnelle grâce à la réalisation d’un projet réel dans lequel nous mettrons en œuvre nos connaissances acquises en développement informatique.

***II. Présentation de l’entreprise d’accueil***

***JBA expertise*** est une entreprise unipersonnelle Fondée en avril 2012 au capital de 1 EURO par Jawhar BEN ABDALLAH, Chef de projet et expert dans l’étude et la réalisation des systèmes d’information. Son siège est immatriculé auprès des greffes et tribunaux de la ville de Nanterre***,*** spécialisée dans les développent et conseil pour la mise en place des plateformes basées sur les technologies Java/J2EE.

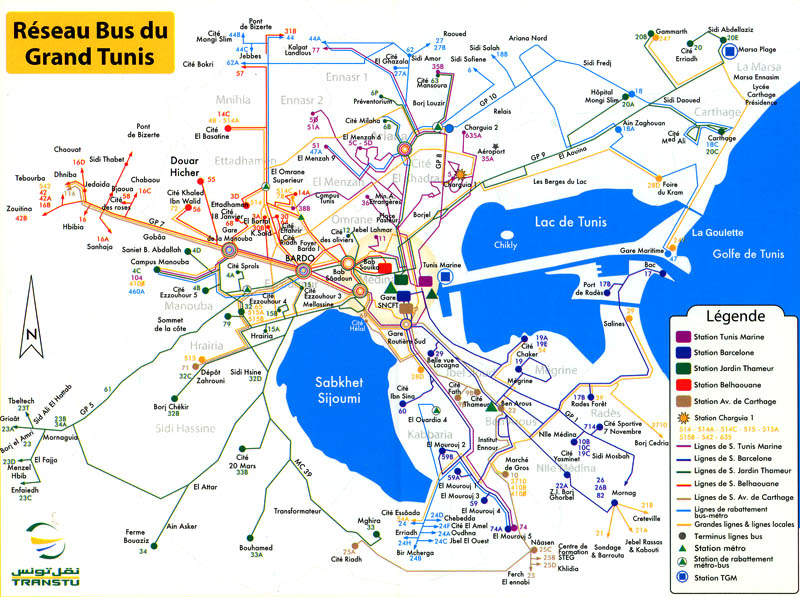
***III. Etude de l’existant***

***III.1 Description de l’existant***

De nos jours les sociétés de transports informent leurs clientèles des horaires et des itinéraires des voyages via les voies classiques tels que les guides, les affichages papier dans les stations, information au guichet et leurs sites web.

**III.1.1 Les guides**

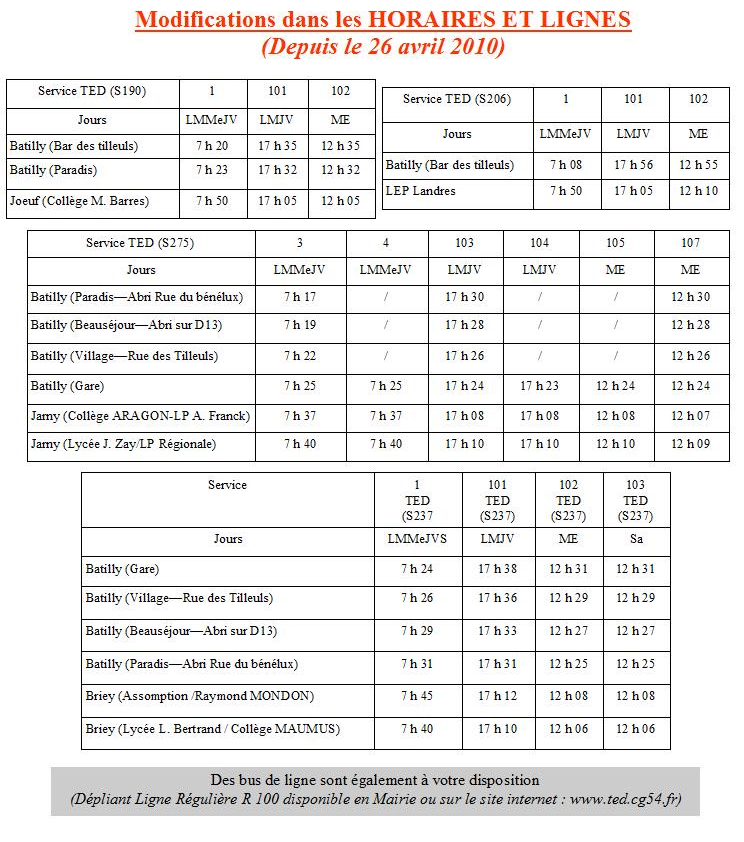
Les sociétés de transport impriment des guides qui sont une sorte d’un document écrit qui regroupe l’ensemble d'informations concernant leurs réseaux routiers et les trajets des voyages et qui seront distribuées sur leurs clientèles.



**Figure : Guide voyageur de la société TRANSTU**

**III.1.2 L’affichage papier**

C’est une autre façon pour que les sociétés de transport communiquent leurs clients donc il s’agit d’accrocher des papiers dans chaque station qui contient les horaires des voyages programmés.



**Figure : Exemple d’un affichage papier des horaires des voyages**

**III.1.3 information au guichet**

Le voyageur s’adresse au guichet ou bien à l’agent dans les stations pour obtenir plus des renseignements sur les horaire des voyages, pour lui aidé dans l’élaboration de ses déplacement (itinéraires, localisation dans le quartier, horaires, conditions de trafic) ou bien lui propose des itinéraires de substitution en cas de perturbation.



**Figure :**

**III.1.4 Les sites web**

C’est une solution un peu avancés pour fournir plus des renseignements aux clients, elle consiste à mettre en ligne des informations qui sont plus ou moins dynamiques par rapport aux autres solutions citées précédemment.



**Capture d'écran des sites web des sociétés tunisiennes SRTB et TRANSTU**

***III.2 Critique de l’existant***

Les affichages en papiers et les guides sont les moyens les plus répandus que les sociétés de transports utilisent pour informer leurs clients des horaires des voyages. Ces moyens se caractérisent par leur cout élevé (cout d’impression, cout de distribution) et la non fiabilité des résultats car les informations seront figées à l’ instant de l’impression et ne reflétant pas la réalité.

Suite à l’évolution des technologies de l’information les sociétés de transports essaient de plus en plus d’être plus proche de leurs clients en leurs fournissant l’information via leur sites d’internet. Mais nous remarquons que dans la plus part des cas que les informations sont obsolètes et dépassés par les événements faute de mise à jour ce qui ne servira pas grand-chose aux clients et ne donnera pas satisfaction.

***III.3 Solution proposée***

Tenant compte des critiques présentés précédemment, nous avons pensé à développer un système d’information en ligne pour les sociétés de transports en commun qui répond aux objectifs et qui pallie aux lacunes constatées au niveau du processus existant.

Ce système va faciliter la tâche au passager et lui permet d’optimiser ses déplacements et ses voyages à travers plusieurs nouveaux services, tels que les horaires des prochains passages à un point d’arrêt en temps réel, le suivi des retards, les calculs des itinéraires et l’abonnement au service des alertes pour recevoir des informations concernant les perturbations sur le réseau de transport.

Ce système vapermettre à l’administrateur de la société de gérer les données du réseau de transport : tels que la gestion des stations, la gestion des lignes, la gestion des parcours, la gestion des alertes, et la supervision du trafic du réseau.

***IV. Méthodologie adoptée***

Pour bien mener notre projet il s'avère nécessaire, de suivre une méthodologie de développement. Notre choix s’est porté vers la processus unifié(PU ou UP en anglais pour Unified Process) qui est une démarche de développement logiciel orienté objet.il s’adresse au membre de l’équipe de développement chargée des activités du cycle de vie qui sont la formulation des besoins, l’analyse, la conception et les tests, d’autre termes des activités produisent des modèle UML. Le processus unifié est pilotée par les cas d’utilisation, l’itération, l’incrémentation, centré sur l’architecture et utilise le langage UML.

Le processus unifié comporte quatre phases qui sont :

* Préetude (Inception) : c’est ici que nous faisons une étude de faisabilité en évaluant la valeur ajoutée du développement et la capacité technique à le réaliser.
* Elaboration : au niveau de cette phase nous confirmons l’adéquation du système aux besoins des utilisateurs et nous livrons l’architecture de base.
* Construction : dans cette phase nous produisons progressivement les différentes fonctions étudiés du système.
* Transition : à la phase finale le produit est livré en version bêta à la disposition des utilisateurs.

***IV.2 Langage de modélisation UML***

Pour concevoir notre application nous avons choisi le langage de modélisation UML (Unified Modelling Language) vu qu’il est standard et peut être appliqué à toutes les méthodes de développement orientées objet. Il se base sur un ensemble de diagrammes allant des cas d’utilisation qui sert à exprimer les besoins de l’utilisateur jusqu’aux diagrammes de déploiement qui illustrent le déploiement physique des composant logiciels et matériels du système.

***V. Planification du projet :***

La planification est parmi les phases d'avant-projet. Elle consiste non seulement à délimiter le périmètre temporel du projet, mais aussi à prévoir le déroulement des activités tout au long de la période allouée au stage.

Les étapes de travail que nous avons suivi :

* Etude du besoin et rédaction d’un document de spécifications des travaux à réaliser
* Définition des cas d’utilisation
* Conception des écrans de l’application
* Conception des données
* Développement et tests unitaires
* Rédaction du rapport du stage

Ci-dessous le chronogramme de la répartition des semaines entre les différentes étapes d’élaboration du projet.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mois | Février | | | | Mars | | | | Avril | | | | Mai | | | | Juin | |
| Semaines | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| Etude de l’existant |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Spécification des besoins |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Conceptions des écrans |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Conceptions des données |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Développement et tests unitaires |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Rédaction du rapport du stage |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

***Conclusion***

Ce chapitre a été consacré à la spécification du projet, à travers lequel nous avons présenté son cadre général ainsi une description et critique des solutions existantes utilises par les sociétés des transports en communs.

Le chapitres suivant sera consacres à la spécification des besoins du projet.

|  |  |
| --- | --- |
|  | Chapitre 2**Spécification des besoins** |

***Introduction***

La spécification est la première étape dans un projet. Cette étape est déterminante pour le bon déroulement du projet.

Elle consiste à connaitre le travail demandé et les différents problèmes, posés par le sujet du point de vue organisationnel et technique.

Au niveau de ce chapitre nous allons spécifier en détail les différents besoin fonctionnels et non fonctionnels de notre système d’information, ainsi que les différents acteurs et les cas d’utilisation.

***I. Les besoins fonctionnels***

Les besoins fonctionnels listent les opérations réalisables de notre application. Ce sont des besoins spécifiant un comportement d'entrée / sortie du système.

Dans cette partie, nous exposons l’ensemble des besoins fonctionnels auxquels devraient répondre notre application.

* **Gestion des Station :** Cette fonctionnalité permet à l’administrateur de la société de transport d’ajouter, de modifier, et de supprimer des stations
* **Gestion des Lignes :** Cette fonctionnalité permet à l’administrateur de la société de transport d’ajouter, de modifier, et de supprimer des lignes
* **Gestion des parcours :** Cette fonctionnalité permet à l’administrateur de la société de transport d’ajouter, de modifier, et de supprimer des parcours
* **Gestion des programmes des voyages :** Cette fonctionnalité permet à l’administrateur de la société de transport d’ajouter, de modifier, et de supprimer des programmes des voyages
* **Gestion des conducteurs :** Cette fonctionnalité permet à l’administrateur de la société de transport d’ajouter, de modifier, et de supprimer des conducteurs
* **Gestion des véhicules :** Cette fonctionnalité permet à l’administrateur de la société de transport d’ajouter, de modifier, et de supprimer des véhicules
* **Alerter les perturbations sur le réseau :** Cette fonctionnalité permet à l’administrateur de la société de transport d’informer ses clients voyageurs de l’état du trafic réseau en cas des perturbations.
* **Suivre le trafic réseau :** Cette fonctionnalité permet à l’administrateur de suivre le comportement des bus sur le réseau de transport.
* **Rechercher itinéraire :** Cette fonctionnalité offre au voyageur la possibilité de rechercher le plus court chemin entre deux emplacement.
* **S’informer sur le prochain passage à une station :** Cette fonctionnalité offre au voyageur la possibilité de savoir l’heure des prochains passages du véhicule de transport à une station choisie.
* **Visualiser les points d’arrêt sur GMap :** Cette fonctionnalité offre au voyageur la possibilité la de géolocaliser les points d’arrêts sur Google Map
* **Recevoir des alertes concernant les perturbations sur les lignes :** Cette fonctionnalité offre au voyageur la possibilité de choisir ses parcours préférés et recevoir des alerte en fonction de l‘état du trafic sur le réseau de la société

***II. Les besoins non fonctionnels***

Les besoin non fonctionnels représentent les exigences implicites auxquelles le système doit répondre. Parmi ces besoins on cite :

* La rapidité de traitement : Vu que le besoin de trouver une réponse dans les brefs délais, il est impérativement nécessaire que la durée d’exécution des traitements s’approche le plus possible du temps réel.
* La performance et l’utilité : c'est-à-dire donnant des résultats utiles, précis et exacts
* L’ergonomie de l’application : l’application doit présenter des interfaces simples afin que l’utilisateur puisse naviguer d’un écran vers l’autre sans aucune difficulté.
* La maintenabilité : Le projet doit être bien organisé et le code doit être suffisamment clair pour permettre de futures évolutions et améliorations.

***III. Les diagrammes de cas d’utilisation***

Les cas d'utilisation permettent de représenter le fonctionnement du système vis-à-vis de son environnement extérieur. Le diagramme de cas d’utilisation se compose de :

* Acteurs : ce sont les entités externes (personne humaine ou robot) qui utilisent le système.
* Cas d’utilisation : ce sont les fonctionnalités proposées par le système.

***III.1 présentation des acteurs***

Les acteurs principaux avec lesquels interagit le système sont :

* **Administrateur** : chargé principalement de la gestion des stations, des lignes des parcours, des programmes, et les déclarations des alertes.
* **Voyageur** : consulte l'application sans inscription pour utiliser toutes les fonctionnalités permettant de faciliter ses déplacements, à savoir la localisation des stations, la recherche des plus rapides itinéraires, s’informer sur le prochain passage à une station, et l’inscription au service des alertes.

***III.2 Les cas d'utilisation***

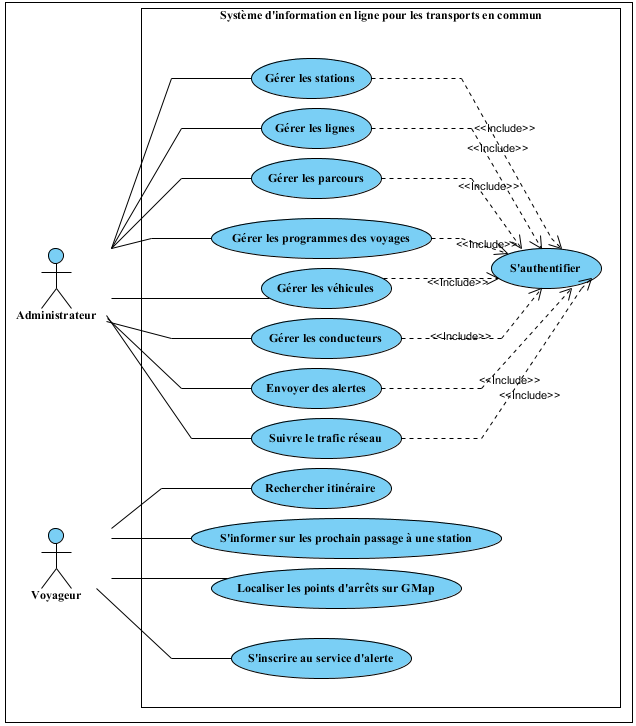
Durant notre étude nous avons identifié les différents cas d’utilisation pour chaque acteur du système.

Le tableau ci-dessous contient une classification des cas d’utilisation par acteur :

|  |  |
| --- | --- |
| **Acteur** | **Cas d’utilisation** |
| Administrateur | * Gérer les stations * Gérer les lignes * Gérer les parcours * Gérer les programmes des voyages * Gérer les véhicules * Gérer les conducteurs * Envoyer des alertes * Suivre le trafic réseau |
| Voyageur | * Rechercher itinéraire * S’informer sur les prochains passages à une station * Localiser les points d’arrêts sur Gmap * Recevoir des alertes concernant les perturbations sur les lignes |

Par la suite, nous illustrons graphiquement ce tableau par un diagramme de cas d’utilisation global.

III.2.1 Diagramme de cas d'utilisation globale



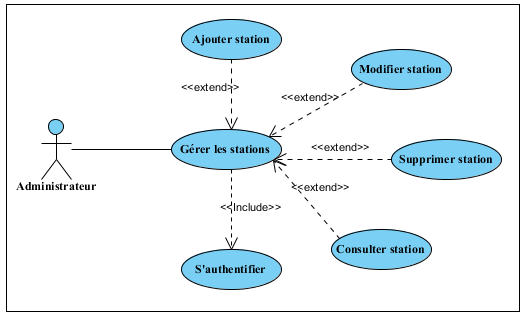
**Figure : Diagramme des cas d’utilisation général**

Ce diagramme représente les cas d’utilisation sans en montrer les détails, chaque cas d’utilisation sera détaillé plus bas.

**2. Description des cas d’utilisation**

Afin de décrire les interactions entre les cas d’utilisation, nous présentons ces derniers de façon textuelle. Il s’agit donc d’associer à chaque cas d’utilisation un nom, un objectif, les acteurs qui y participent, les pré-conditions et des scénarii.

***III.2.2 Raffinement de cas d’utilisation « Gérer les stations »***



**Figure : Diagramme des cas d’utilisation « *Gérer les stations »***

**Cas d'utilisation " Ajouter une station"**

|  |
| --- |
| **Titre : *Ajouter une station***  **But :** *ajouter une nouvelle station au réseau routier.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système l’ajout d’une nouvelle station.  *Enchaînement (a) :* Créer une station  L’utilisateur saisie un nom pour la station.  L’utilisateur saisie les coordonnés GPS de la station  *Enchaînement (b) :* Valider la création.  Valider les données.  **Post conditions** : Le système crée la nouvelle station. |

**Cas d'utilisation " Modifier une station"**

|  |
| --- |
| **Titre : *Modifier une station***  **But :** *mettre à jour des informations concernant une station.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  Il existe forcément une station enregistré à modifier  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système la modification d’une station.  *Enchaînement (a)* **:** Modifier une station.  L’utilisateur modifie le nom de la station et ses coordonnés GPS.  *Enchaînement (b)* : Valider la modification  Valider les données.  **Post conditions** : Le système enregistre la modification de la station. |

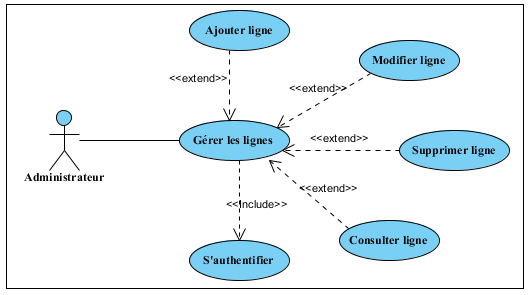
**Cas d'utilisation " Consulter une station"**

|  |
| --- |
| **Titre : *consulter une station***  **But :** *consulter une station.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  Il existe forcément une station enregistré à consulter  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système de lui retourner les informations concernant une station choisie.  *Enchaînement (a)* **:** Consulter une station  L’utilisateur sélectionne la station à consulter.  Le système affiche la station. |

**Cas d'utilisation " Supprimer une station "**

|  |
| --- |
| **Titre : *supprimer une station***  **But :** *supprimer une station.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  Il existe forcément une station enregistré à supprimer  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système de supprimer une station.  *Enchaînement (a)* **:** supprimer une station  L’utilisateur valide la suppression. |

***III.2.3 Raffinement de cas d’utilisation « Gérer les lignes »***

******

**Figure : Diagramme des cas d’utilisation « *Gérer les lignes »***

**Cas d'utilisation " Ajouter une ligne"**

|  |
| --- |
| **Titre : *Ajouter une ligne***  **But :** *ajouter une nouvelle ligne au réseau routier.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système l’ajout d’une nouvelle ligne.  *Enchaînement (a) :* Créer une ligne  L’utilisateur saisie un nom pour la ligne.  *Enchaînement (b) :* Valider la création.  Valider les données.  **Post conditions** : Le système crée la nouvelle ligne. |

**Cas d'utilisation " Modifier une ligne"**

|  |
| --- |
| **Titre : *Modifier une ligne***  **But :** *mettre à jour des informations concernant une ligne.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  Il existe forcément une ligne enregistré à modifier  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système la modification d’une ligne.  *Enchaînement (a)* **:** Modifier une ligne.  L’utilisateur modifie le nom de la ligne.  *Enchaînement (b)* : Valider la modification  Valider les données.  **Post conditions** : Le système enregistre la modification de la ligne. |

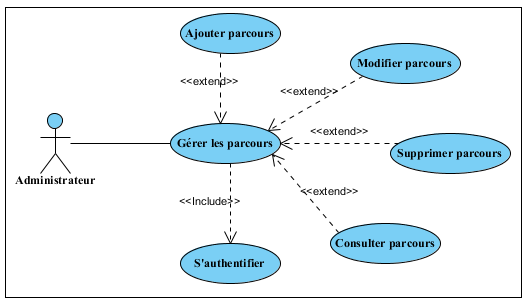
**Cas d'utilisation " consulter une ligne"**

|  |
| --- |
| **Titre : *consulter une ligne***  **But :** *consulter une ligne.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  Il existe forcément une ligne enregistré à consulter  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système de lui retourner les informations concernant une ligne choisie.  *Enchaînement (a)* **:** Consulter une ligne  L’utilisateur sélectionne la ligne à consulter.  Le système affiche la ligne. |

**Cas d'utilisation " Supprimer une ligne "**

|  |
| --- |
| **Titre : *supprimer une ligne***  **But :** *supprimer une ligne.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  Il existe forcément une ligne enregistré à supprimer  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système de supprimer une ligne.  *Enchaînement (a)* **:** supprimer une ligne  L’utilisateur valide la suppression. |

***III.2.4 Raffinement de cas d’utilisation « Gérer les parcours »***

****

**Figure : Diagramme des cas d’utilisation « *Gérer les parcours »***

**Cas d'utilisation " Ajouter un parcours"**

|  |
| --- |
| **Titre : *Ajouter un parcours***  **But :** *ajouter un nouveau parcours au réseau routier.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système l’ajout d’un nouveau parcours.  *Enchaînement (a) :* Créer un parcours  L’utilisateur choisie la ligne auquel appartient le parcours  L’utilisateur saisie un nom pour le parcours.  L’utilisateur sélectionne les stations qui construisent le parcours  *Enchaînement (b) :* Définir les durées.  L’utilisateur saisie les durée entre les stations et les durées d’arrêts dans les stations  *Enchaînement (c) :* Valider la création.  Valider les données.  **Post conditions** : Le système crée le nouveau parcours. |

**Cas d'utilisation " Modifier un parcours"**

|  |
| --- |
| **Titre : *Modifier un parcours***  **But :** *mettre à jour des informations concernant un parcours.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  Il existe forcément un parcours enregistré à modifier  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système la modification d’un parcours.  *Enchaînement (a)* **:** Modifier un parcours.  L’utilisateur modifie le nom du parcours.  Modifier les stations qui construisent le parcours  *Enchaînement (b) :* Redéfinir les durées  L’utilisateur modifie les durées entre les stations et les durées d’arrêts dans les stations  *Enchaînement (c)* : Valider la modification  Valider les données.  **Post conditions** : Le système enregistre la modification du parcours. |

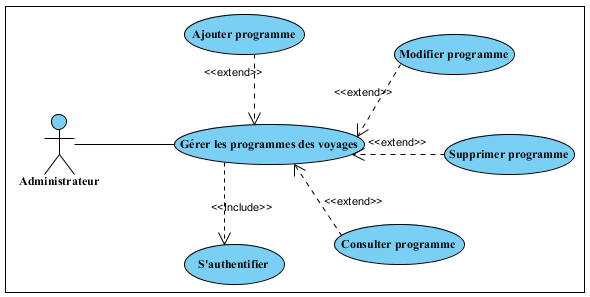
**Cas d'utilisation " Consulter un parcours"**

|  |
| --- |
| **Titre : *consulter un parcours***  **But :** *consulter un parcours.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  Il existe forcément un parcours enregistré à consulter  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système de lui retourner les informations concernant un parcours choisi.  *Enchaînement (a)* **:** Consulter un parcours  L’utilisateur sélectionne le parcours à consulter.  Le système affiche le parcours. |

**Cas d'utilisation " Supprimer un parcours "**

|  |
| --- |
| **Titre : *supprimer un parcours***  **But :** *supprimer un parcours.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  Il existe forcément un parcours enregistré à supprimer  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système de supprimer un parcours.  *Enchaînement (a)* **:** supprimer un parcours  L’utilisateur valide la suppression. |

***III.2.5 Raffinement de cas d’utilisation « Gérer les programmes»***

****

**Figure : Diagramme des cas d’utilisation « *Gérer les programmes »***

**Cas d'utilisation " Ajouter un programmes"**

|  |
| --- |
| **Titre : *Ajouter un programme***  **But :** *ajouter un nouveau programme de voyage.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système l’ajout d’un nouveau programme de voyage.  *Enchaînement (a) :* Créer un programme  L’utilisateur saisie la date et l’heure du début du programme  L’utilisateur choisie le parcours à effectuer.  L’utilisateur affecte le véhicule et le conducteur à réaliser le programme  *Enchaînement (b) :* Valider la création.  Valider les données.  **Post conditions** : Le système ajoute le nouveau programme. |

**Cas d'utilisation " Modifier un programme"**

|  |
| --- |
| **Titre : *Modifier un programme***  **But :** *mettre à jour des informations concernant un programme de voyage.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  Il existe forcément un programme enregistré à modifier  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système la modification d’un programme.  *Enchaînement (a)* **:** Modifier un programme.  L’utilisateur modifie les détails du programme.  *Enchaînement (b)* : Valider la modification  Valider les données.  **Post conditions** : Le système enregistre la modification du programme. |

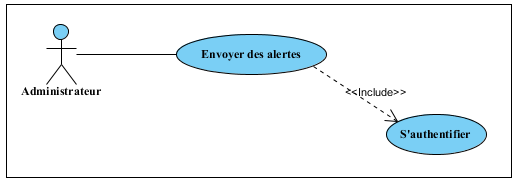
**Cas d'utilisation " Consulter un programme"**

|  |
| --- |
| **Titre : *consulter un parcours***  **But :** *consulter les detail concernant un programme de voyage.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  Il existe forcément un programme enregistré à consulter  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système de lui retourner les informations concernant un programme choisi.  *Enchaînement (a)* **:** Consulter un programme  L’utilisateur sélectionne le programme à consulter.  Le système affiche les informations concernant le programme choisi. |

**Cas d'utilisation " Supprimer un programme "**

|  |
| --- |
| **Titre : *supprimer un programme***  **But :** *supprimer un programme.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  Il existe forcément un programme enregistré à supprimer  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système de supprimer un parcours.  *Enchaînement (a)* **:** supprimer un parcours  L’utilisateur valide la suppression. |

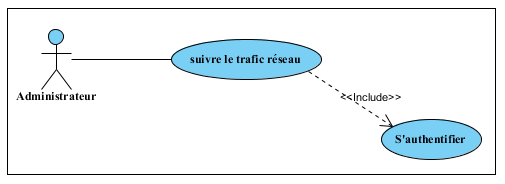
***III.2.6 Raffinement de cas d’utilisation « Envoyer des alertes»***

******

**Cas d'utilisation " Envoyer des alertes"**

|  |
| --- |
| **Titre : Envoyer des alertes**  **But :** Envoyer des alertes au voyageur*.*  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  Il existe une perturbation sur une ligne du réseau routier  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système l’ajout d’une alerte.  *Enchaînement (a)* **:** ajouter une alerte  L’utilisateur choisie la ligne qui est perturbée  L’utilisateur saisie une description de la cause de l’alerte.  *(b)* : Envoyer l’alerte  **Post conditions** : Le système envoie un mail au voyageur inscrit pour le suivie des perturbations sur la ligne concerné par l’alerte. |

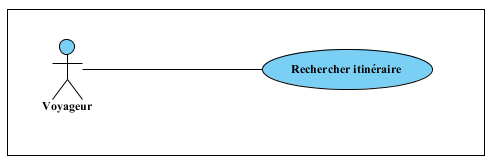
***III.2.6 Raffinement de cas d’utilisation « Suivre le trafic réseau»***

******

**Cas d'utilisation "Suivre le trafic réseau"**

|  |
| --- |
| **Titre : Suivre le trafic réseau**  **But :** suivre les déplacements des bus sur le réseau routier en temps réel  **Acteurs :** *Administrateur*  **Préconditions :**  L’administrateur est authentifié.  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système de lui fournir les informations concernant les entrées et les sorties des bus dans les points d’arrets.  **Post conditions** : le système affiche les événements d’entrés sorties réalisés |

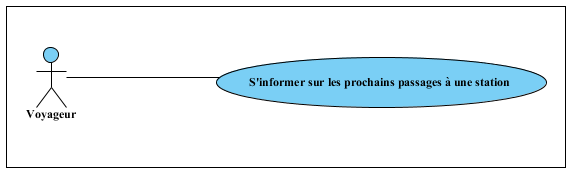
***III.2.6 Raffinement de cas d’utilisation « Rechercher itinéraire»***

******

**Cas d'utilisation " Recherche itinéraire"**

|  |
| --- |
| **Titre : Recherche itinéraire**  **But :** *rechercher l’itinéraire le plus rapide entre deux points d’arrêts en fonction des voyages programmés*  **Acteurs :** *Voyageur*  **Préconditions :**  Le voyageur accède à l’application.  Il existe des voyages programmés  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système de rechercher le chemin le plus rapide entre deux points d’arrêts.  *Enchaînement (a)* **:** Saisir les informations du voyage.  L’utilisateur choisi le point de départ et le point d’arrivé.  L’utilisateur saisie la date et l’heure de départ  *Enchaînement (b)* : Rechercher itinéraire  Le système cherche le chemin le plus rapide.  **Post conditions** : Le système affiche l’itinéraire |

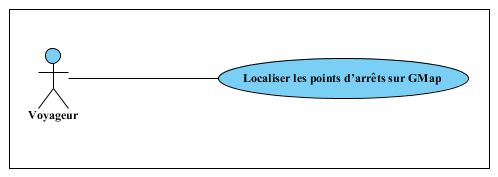
***III.2.6 Raffinement de cas d’utilisation «*s’informer sur les prochains passages à une station*»***

******

**Cas d'utilisation "s’informer sur le prochain passage à une station"**

|  |
| --- |
| **Titre : s’informer sur le prochain passage à une station**  **But :** *s’informer sur l’heure des prochains passages du bus à un point d’arrêts en temps réel*  **Acteurs :** *Voyageur*  **Préconditions :**  Le voyageur accède à l’application.  Il existe des voyages programmés  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système de rechercher l’heure des prochains passages des bus à une station  *Enchaînement (a)* **:** Choisir la station de départ.  L’utilisateur choisi sa station de départ.  Le système cherche et affiche les lignes qui passent par cette station.  *Enchaînement (b)* : Choisir la destination (parcours d’aller ou de retour)  L’utilisateur choisi la ligne correspondante a son trajet et sa destination.  **Post conditions** : Le système affiche l’heure des prochains passages vers la destination souhaité |

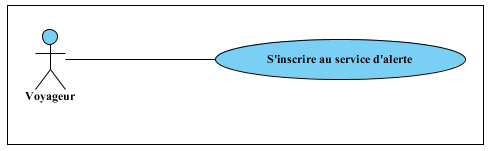
***III.2.6 Raffinement de cas d’utilisation «*Localiser les points d’arrêts sur GMap*»***

******

**Cas d'utilisation "Localiser les points d’arrêts sur GMap"**

|  |
| --- |
| **Titre : Localiser les points d’arrêts sur Gmap**  **But :** *localiser les points d’arrêts du réseau routier sur la carte GoogleMap*  **Acteurs :** *Voyageur*  **Préconditions :**  Le voyageur accède à l’application.  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système d’afficher les stations sur une carte.  **Post conditions** : Le système affiche marque les stations sur la carte GoogleMap |

***III.2.6 Raffinement de cas d’utilisation «*S’inscrire au service d’alerte*»***



**Cas d'utilisation "S’inscrire au service d’alerte "**

|  |
| --- |
| **Titre : *Recevoir des alertes concernant les perturbations sur les lignes***  **But :** *s’inscrire au service d’alerte pour recevoir des informations concernant toutes perturbations sur une ligne.*  **Acteurs :** *Voyageur*  **Préconditions :**  Le voyageur accède à l’application.  **Scénario nominal :**  Ce cas d’utilisation commence lorsque l’utilisateur demande au système de s’inscrire au service d’alerte.  *Enchaînement (a)* **:** s’inscrire  L’utilisateur choisie une ligne laquelle veut recevoir des alertes sur le changement de son etat  L’utilisateur saisie son adresse mail  *Enchaînement (b)* **:** recevoir une alerte en mail.  Le système envoie une alerte en mail pour tous les voyageurs inscrits |

Conclusion

Après avoir achevé la phase de spécification des acteurs ainsi que la description des cas d’utilisation, nous entamons la phase de conception de notre projet en s’appuyant sur le formalisme du langage UML.

|  |  |
| --- | --- |
|  | Chapitre 3**Conception** |

***Introduction***

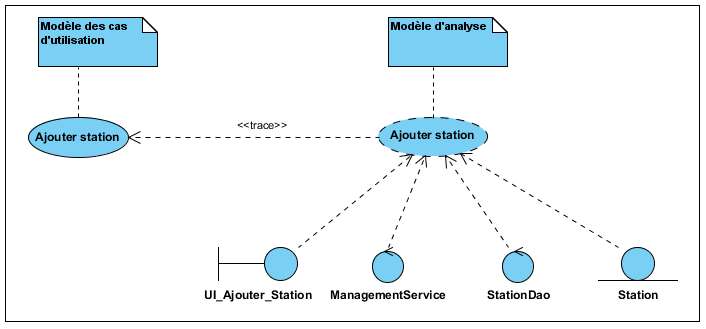
Ce chapitre se consacre en premier lieu, à présenter l’architecture globale à adopter pour la mise en place de notre système

La deuxième partie constitue une suite du chapitre précédent, afin d'enrichir l'analyse fonctionnelle effectuée auparavant à l'aide des diagrammes de cas d'utilisation UML. Les parties qui suivent présentent quelques diagrammes de séquences et le diagramme de classes.

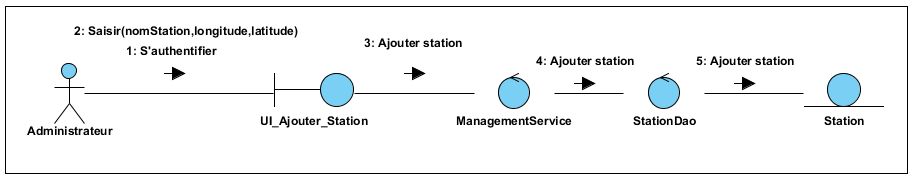
**I. Analyse**

Au cours de cette partie nous allons approfondir notre compréhension du système en effectuant le modèle d’analyse des différents cas d’utilisation qui tourne autour de la traçabilité entre le modèle des cas d’utilisation et le modèle d’analyse, et le diagramme de collaboration.

**Cas d’utilisation « Ajouter Station »**

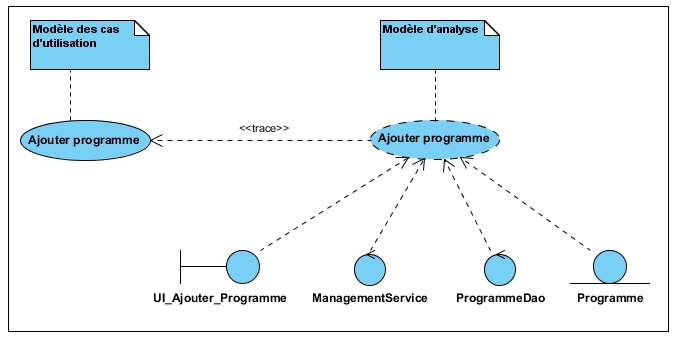
****

**Traçabilité entre le modèle de cas d’utilisation et le modèle d’analyse du cas d’utilisation « Ajouter station»**

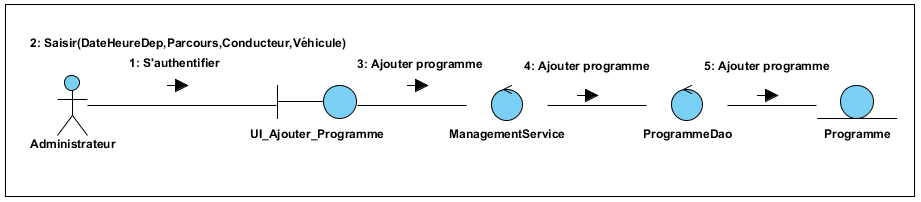
****

**Diagramme de collaboration du cas d’utilisation « Ajouter station»**

**Cas d’utilisation « Ajouter programme »**

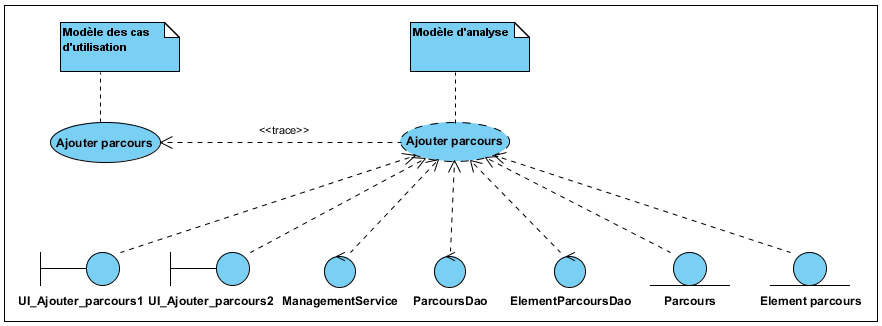
****

**Traçabilité entre le modèle de cas d’utilisation et le modèle d’analyse du cas d’utilisation « Ajouter programme »**

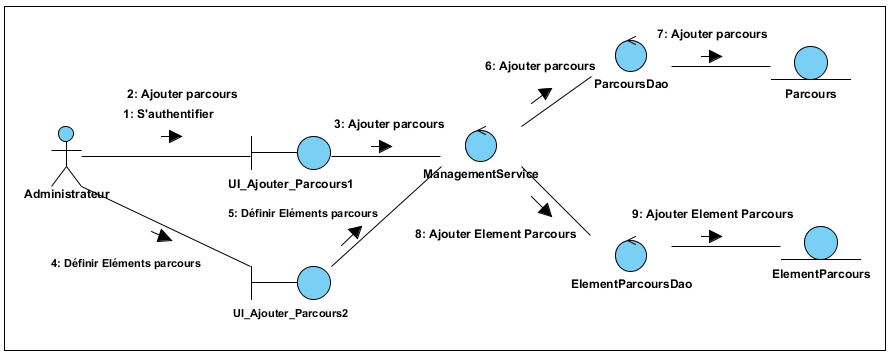
****

**Diagramme de collaboration du cas d’utilisation « Ajouter programme »**

**Cas d’utilisation « Ajouter parcours »**

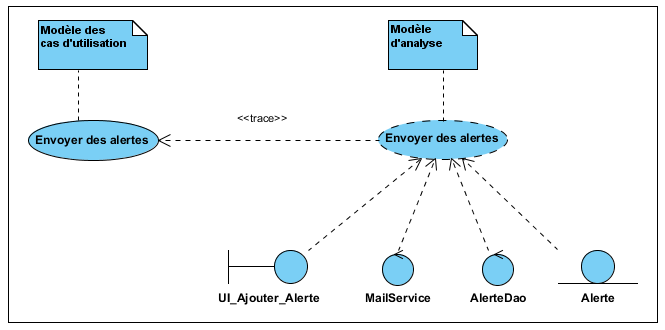
****

**Traçabilité entre le modèle de cas d’utilisation et le modèle d’analyse du cas d’utilisation « Ajouter parcours»**

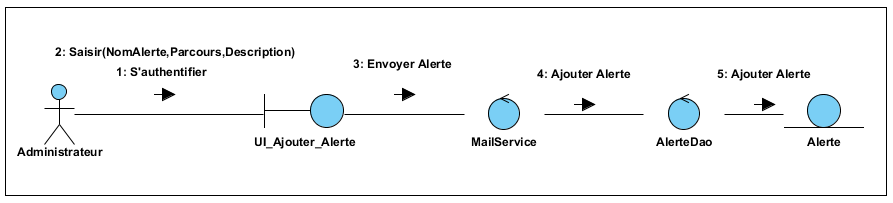
****

**Diagramme de collaboration du cas d’utilisation « Ajouter parcours»**

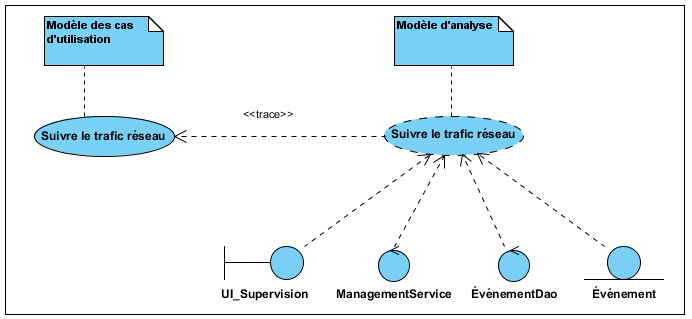
**Cas d’utilisation « Envoyer des alertes»**

****

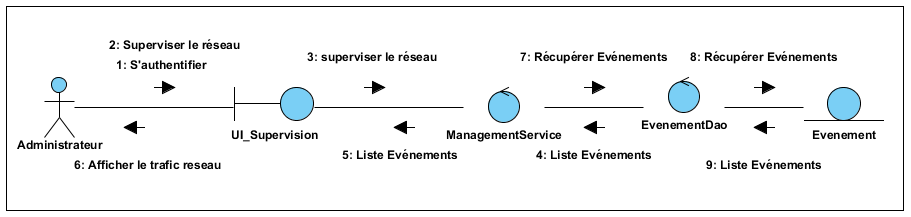
**Traçabilité entre le modèle de cas d’utilisation et le modèle d’analyse du cas d’utilisation « Envoyer des alertes»**

**Diagramme de collaboration du cas d’utilisation « Envoyer des alertes»**

**Cas d’utilisation « Suivre le trafic réseau»**

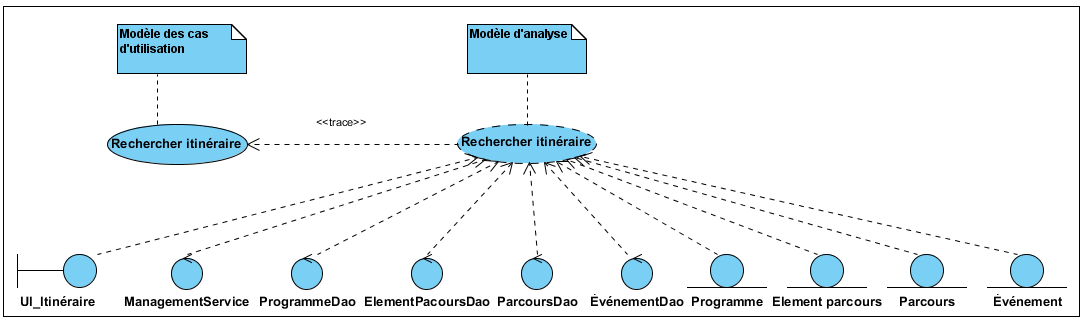
****

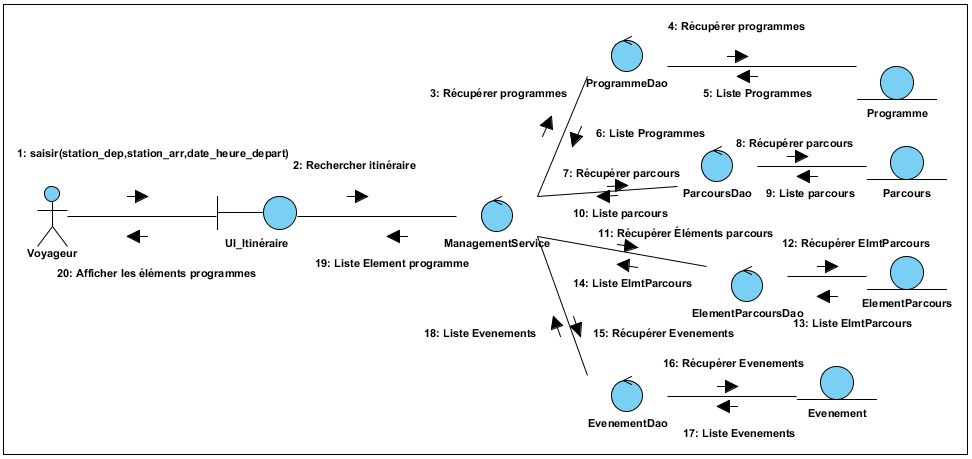
**Traçabilité entre le modèle de cas d’utilisation et le modèle d’analyse du cas d’utilisation « Suivre le trafic réseau»**

****

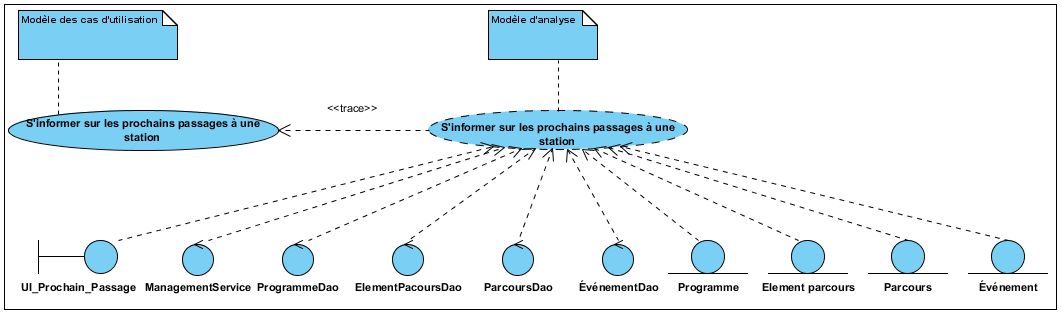
**Diagramme de collaboration du cas d’utilisation Suivre le trafic réseau»**

**Cas d’utilisation « Rechercher itinéraire »**

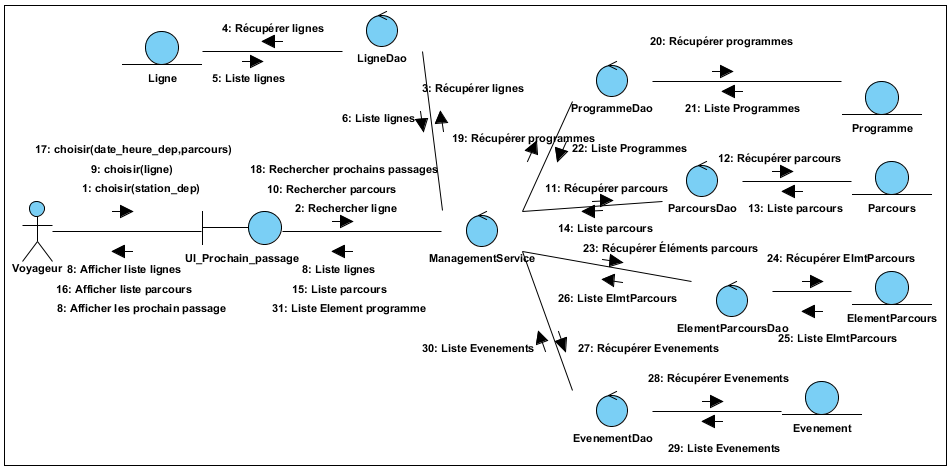
**Traçabilité entre le modèle de cas d’utilisation et le modèle d’analyse du cas d’utilisation « Rechercher itinéraire »**

**Diagramme de collaboration du cas d’utilisation « Rechercher itinéraire»**

**Cas d’utilisation « S’informer sur les prochains passages à une station »**

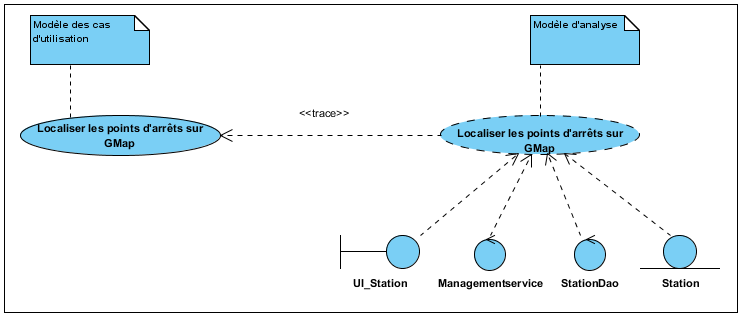
****

**Traçabilité entre le modèle de cas d’utilisation et le modèle d’analyse du cas d’utilisation « rechercher itinéraire »**

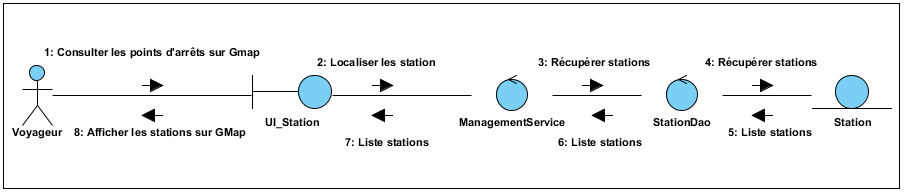
****

**Diagramme de collaboration du cas d’utilisation « S’informer sur les prochains passages à une station»**

**Cas d’utilisation « Localiser les points d’arrêts sur GMap»**

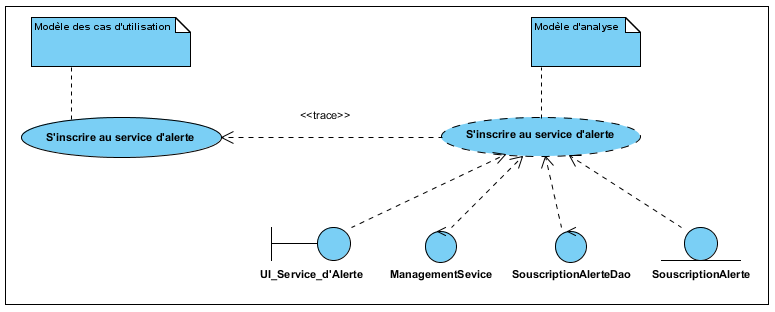


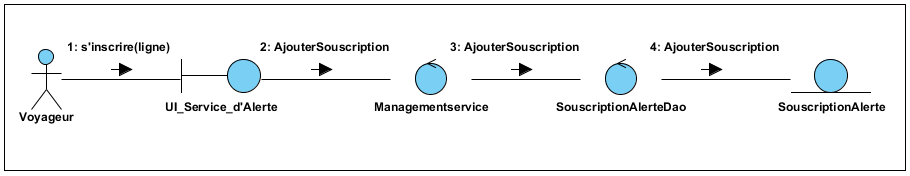
**Traçabilité entre le modèle de cas d’utilisation et le modèle d’analyse du cas d’utilisation «Localiser les points d’arrêts sur GMap»**

****

**Diagramme de collaboration du cas d’utilisation « Localiser les points d’arrêts sur GMap»**

**Cas d’utilisation « S’inscrire au service d’alerte »**

** Traçabilité entre le modèle de cas d’utilisation et le modèle d’analyse du cas d’utilisation « « S’inscrire au service d’alerte »**

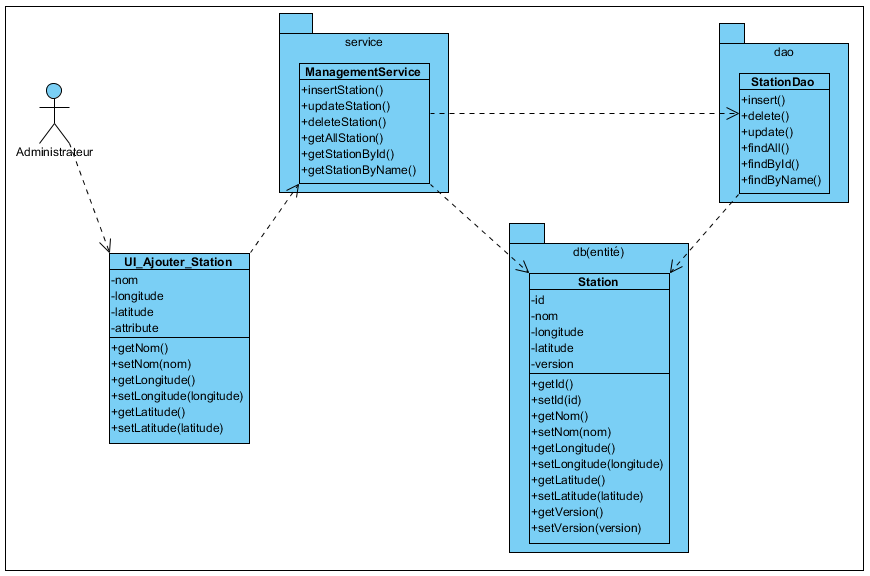
** Diagramme de collaboration du cas d’utilisation « « S’inscrire au service d’alerte »**

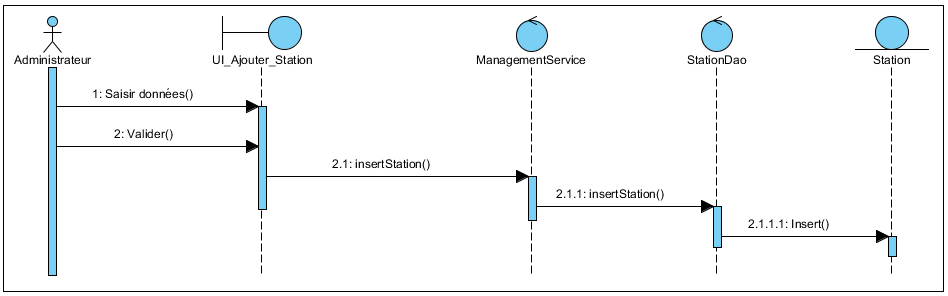
**Etude conceptuelle :**

Dans cette partie nous allons effectuer une étude conceptuelle des cas d’utilisation en détaillant l’analyse effectué dans la partie précédente à travers : les diagrammes de classes et les diagrammes de séquences.

**Conception du cas d’utilisation «Ajouter Station »**

Diagramme de classes « Ajouter station »

**Diagramme de classes de conception réalisant un cas d’utilisation «Ajouter station »**

**Diagramme de séquence du cas d’utilisation «Ajouter station »**

|  |  |
| --- | --- |
|  | Chapitre 4**Réalisation** |

**Introduction**

Pour aboutir à la finalité de notre projet, nous entamons la partie réalisation à la quelle nous commençons dans un premier lieu d’argumenter notre choix matériel et logiciel pour le développement du projet ensuite nous présentons le processus de déploiement par des captures d’écran pour bien décrire le travail réalisé.

**I. Environnement de développement**

Le choix du bon environnement de développement est très important pour la réalisation du projet. Cela se fait suivant plusieurs facteurs : la puissance d’execution, la facilité d’utilisation, la disponibilité de plusieurs fonctionnalités, la communication avec d’autre environnement, etc.

Au cours de cette phase, nous présentons les outils et les langages de programmation que nous l’avons utilisé afin de réussir la phase de développement de notre projet.

**I.1. Environnement matériel**

Pour la réalisation de notre projet, nous avons utilisé un ordinateur DELL tournant sur le système d’exploitation Microsoft Windows 7 dont ses caractéristiques sont les suivantes :

* Un processus Intel core i5
* Une mémoire vive de 4Go
* Un disque dur de 500 Go
* Un système d’exploitation 32 bit

**I.2. Environnement logiciel**

**I.2.1 Subversion (SVN)**

Subversion est un logiciel de gestion de versions, il s'agit d'un logiciel qui surveille et enregistre tous les états des différentes versions d'un répertoire ou d’un fichier au fil du temps.

Subversion est principalement utilisé dans le cadre du développement de logiciels.

Le développement d'un logiciel est composé de multiples modifications de fichiers au fil du temps alors que SVN permet d'enregistrer tous ces changements pour, par exemple, pouvoir avoir une trace explicite et exhaustive de tous les changements faits au code source, ou pouvoir revenir au code tel qu'il était a un instant passé .

**I.2.2 Eclipse Juno**

Eclipse est un IDE (Integrated Development Environment) ou environnement de développent intégré dont le but est de fournir une plateforme modulaire afin de permettre la réalisation des développements informatique

**I.2.3 Apache Tomcat**

Un serveur d’application JEE libre entièrement écrit en java et peut être utilisé sur tout système d’exploitation fournissant une machine virtuelle Java (JVM) .

**Apache Derby**

Afin de stocker et manipuler les données, nous avons besoin d’un Système de gestion de base de données. Pour cela, nous avons choisi le SGBD Apache Derby qui est entièrement écrit en java et qui a l'avantage d'être gratuit et facilement embarquable.

**Pacestar UML Diagram**

C’est un logiciel de création des diagrammes UML  permet :

* La modélisation d’une application informatique.
* La création des neuf types de diagrammes UML de manière simple et graphique :
  + le diagramme de cas d’utilisation,
  + le diagramme de classes,
  + le diagramme de séquence,
  + le diagramme de collaboration
  + le diagramme d’état
  + le diagramme d’activité
  + le diagramme de déploiement
* L’exportation de ces diagrammes dans de nombreux formats (GIF,PNG , PS ...).
* La génération des classes Java.

est un logiciel très stable et mature, de plus sa finition est exemplaire et il est un bon outil de modélisation UML.

**JSF(Java Server Faces)**

JSF est une technologie dont le but est de proposer un Framework qui facilite et standardise le développement d'applications web avec Java.

L’objectif de JSF, est de procurer un environnement de développement permettant de construire une interface de type web, sans devoir toucher au code HTML et JavaScript.

Ceci est réalisé par la mise en place d'un mapping entre l'HTML et les objets concernés.

**OpenFaces**

Pour concevoir l’interface graphique de l’application, nous avons choisie la bibliotheque OpenFaces qui est une bibliothèque open-source de composants JSF riches couplée à une solution de requêtage Ajax.

**JavaMail**

La **JavaMail API** est utilisée pour émettre et recevoir des mails

**C3P0**

C3P0 est un pool de connexion JDBC open-source . C3P0 possède un algorithme évolué pour gérer les connexion a la source de données, qui peut être utilisé directement dans un programme en production.

Le chargement de pilote jdbc a chaque demande de connexion est couteuse. il revient donc d'utiliser un pool de connexion qui est un mécanisme permettant de créeer un ensemble de connexion (instancier les connexion une fois pour toute) et de les réutiliser a la demande par un utilisateur.

**GMaps4JSF**

GMaps4JSF est une bibliotheque qu ivise à intégrer Google Maps avec JSF.

L’avantage est que nous pouvons manipuler des cartes seulement avec quelques lignes de code (tags JSF) sans toucher du javascript.

**JUnit**

**JUnit** est un Framework de test unitaire pour le langage de programmation Java.

JUnit est un framework open source pour le développement et l'exécution de tests unitaires automatisables. Le principal intérêt est de s'assurer que le code répond toujours aux besoins même après d'éventuelles modifications. Plus généralement, ce type de tests est appelé tests unitaires de non régression.

**Java**

Pour développer l’application, j'ai choisi le langage de programmation java. Il a l'avantage d'être gratuit, connu, rapide à l'exécution, et il est orienté objet, ce qui nous facilitera beaucoup les choses pour structurer le projet grâce à la richesse de ses API (l’api swing l’api pour l’impression).

**JavaMail**

JavaMail est une API qui permet d'utiliser le courrier électronique (e-mail) dans une application écrite en java (application cliente, applet, servlet, EJB, ... ). Son but est d'être facile à utiliser, de fournir une souplesse qui permette de la faire évoluer et de rester le plus indépendant possible des protocoles utilisés.