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

***Introduction***

Dans ce chapitre introductif, nous présenterons le cadre générale du projet suivi d’une présentation générale de JBA expertise, la société dans laquelle nous avons effectué notre projet de fin d’études. Ensuite nous allons découvrir et décrire la solution existante qui assure la communication entre les entreprises de transport et ces clients voyageurs, et nous entamons les différentes critiques de ce qui existe ainsi que la solution envisageables.

***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 mettons 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 :***

Lorsque nous avons reçu notre sujet à JBA expertise, sujet qui consiste à développer un prototype de système d’information pour les petites entreprises de transport en commun, la question qui s’est posée immédiatement c’est pourquoi ? N’y aurait-il pas de solutions prêtes à être utilisées ?

C’est ça la question qui nous a menés à effectuer une étude de l’existant, d’où on a trouvé que les clients voyageurs des sociétés de transport rencontrent des problèmes ennuyeux lors de ses déplacements à cause de l’absence totale d’un moyen efficace qui lui fournit l’information qu’ils souhaitent sur l’heure de prochain passage du bus dans une station ,l’itinéraire le plus rapide entre deux endroits ,ou bien sur l’état du trafic d’une ligne préféré. Ces difficultés peuvent se développer à des incidents.

En outre, pour retrouver une information qui peut lui aider, le voyageur doit avoir recours à des moyens classiques tels que le guide et le simple affichage papier dans les stations ou bien s’adresser au site internet de la société s’il y en a qui n’est pas à jour et qui ne possède pas l’information en temps réel.

***III.2 Critique de l’existant***

Nous avons remarqué que les solutions existantes n’ont jamais assuré l’information au voyageur et ne répondent pas à ses besoins souhaités.

D’une part le voyageur rencontre une perte de temps en attendant l’arrivée du moyen de transport sans avoir aucune idée sur l’heure de son arrivée à la station.

D’autre part le voyageur n’as pas une solution lui permettra de rejoindre une destination souhaité dans les meilleurs délais possibles. En plus, absence d’un service d’alerte pour les perturbations sur les lignes.

***III.3 Solution proposée***

Afin d’éviter les problèmes exposés ci-dessus nous avons pensé à développer un prototype d’un système d’information en ligne pour les transports en commun qui peut être amélioré dans les prochains jours et que nous l’avons nommé SIELTEC.

La solution que nous proposons dans ce contexte est formée par trois applications différentes qui sont :

* SIELTEC
* SIELTEC Admin
* Le simulateur

***III.3.1 SIELTEC***

**SIELTEC** est une application web publique qui va faciliter la tâche au passager et lui permet d’optimiser ses déplacements des 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.

***III.3.2 SIELTEC Admin***

**SIELTEC Admin** est une deuxième application web destiné à l’administrateur de l’entreprise de transport (et non aux voyageurs) lui permet la gestion des données du système d’information : tels que ajout de lignes, ajout de nouvelles stations, modifications de parcours…

***III.3.3 Simulateur***

**Le simulateur** est notre troisième application qui est un dispositif permettant de reproduire d’une façon virtuelle le comportement réel des moyens de transports sur les différentes lignes.

***IV. Méthodologie adoptée***

Pour organiser notre travail, nous avons choisi de suivre le Processus Unifié et l’UML comme un langage de modélisation.

***IV.1 Processus Unifié(UP)***

Le Processus Unifié (PU ou UP en anglais pour **Unified Process**) est une méthode de développement logiciel construite sur UML, elle est itérative et incrémentale, centrée sur l’architecture, conduite par les cas d’utilisation et pilotée par les risques.

***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 fait

* Etude du besoin
* Définition des cas d’utilisation
* Conception des écrans de l’application
* Conception des données
* Développement
* Rédaction du rapport du stage

***Conclusion***

Dans ce chapitre, nous avons analysé et critiqué l’existant et proposé des solutions. Dans le chapitre suivant, nous allons présenter les besoins fonctionnels et non fonctionnels.

|  |  |
| --- | --- |
|  | Chapitre 2**Etat de l’art** |

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

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.
* **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 possibilité au voyageur 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 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.

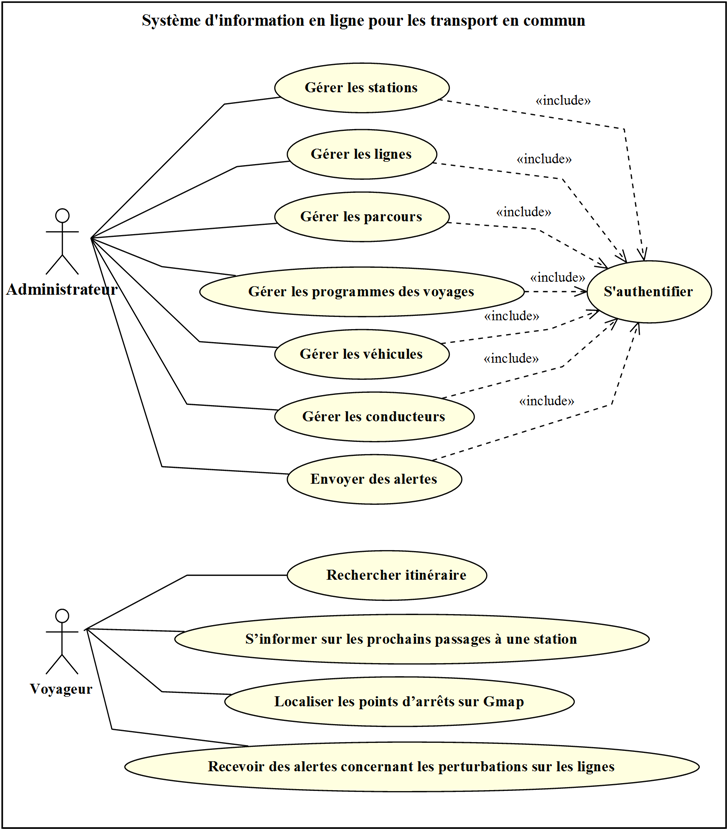
***III.2 Les cas d'utilisation***

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 :

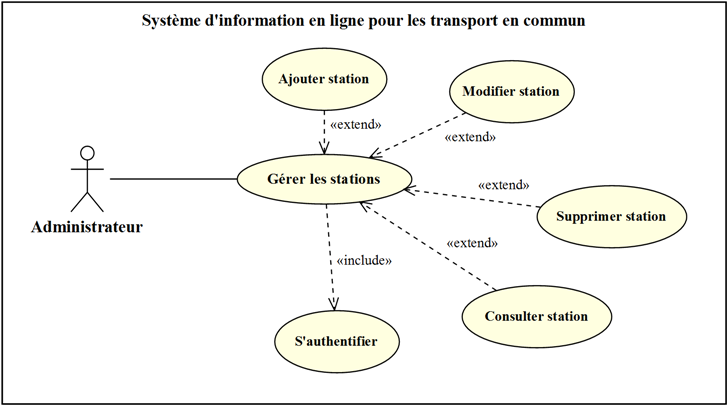
|  |  |
| --- | --- |
| **Acteur** | **Cas d’utilisation** |
| Administrateur | * Gérer les stations * Gérer les lignes * Gérer les parcours * Gérer les programmes des voyages * Gérer les véhicules * Gérer les conducteurs * Envoyer des alertes |
| Voyageur | * Rechercher itinéraire * S’informer sur les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

III.2.1 Diagramme de cas d'utilisation globale



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

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

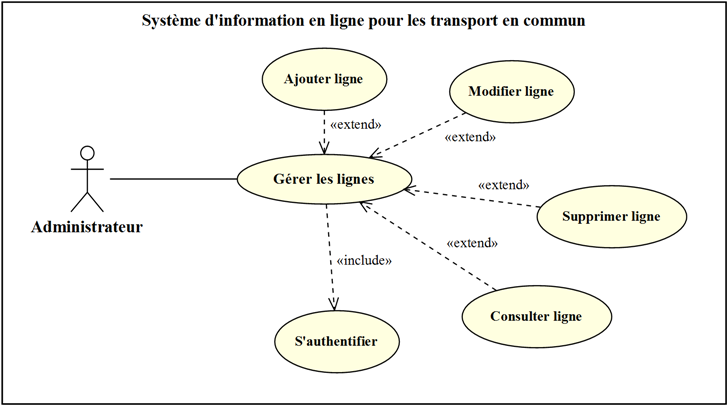


**Figure : Diagramme des cas d’utilisation « *Gérer les stations »***

Description textuelle :

|  |  |
| --- | --- |
| Titre | Gérer les stations |
| Acteur | Administrateur |
| Objectif | Ajouter, Modifier, Supprimer des stations |
| Scénario nominal | 1. L’administrateur consulte l’application.  2. l’administrateur accède au menu station.  3. L’administrateur choisir l’opération à effectuer : Ajouter, modifier, supprimer, ou simple consultation.  4. En cas d’ajout ou de modification d’une station le système affiche un formulaire que l’administrateur doit le remplir.  5. En cas de suppression l’utilisateur doit cliquer sur le bouton de suppression.  6. En cas de consultation le système affiche une liste qui contient tout les stations. |

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

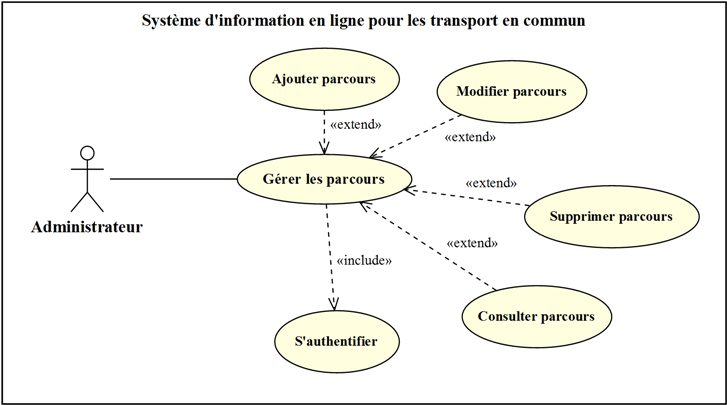
******

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

Description textuelle :

|  |  |
| --- | --- |
| Titre | Gérer les lignes |
| Acteur | Administrateur |
| Objectif | Ajouter, Modifier, Supprimer des lignes |
| Scénario nominal | 1. L’administrateur consulte l’application.  2. l’administrateur accède au menu ligne.  3. L’administrateur choisir l’opération à effectuer : Ajouter, modifier, supprimer, ou simple consultation.  4. En cas d’ajout ou de modification d’une ligne le système affiche un formulaire que l’administrateur doit le remplir.  5. En cas de suppression l’utilisateur doit cliquer sur le bouton de suppression.  6. En cas de consultation le système affiche une liste qui contient tout les lignes. |

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

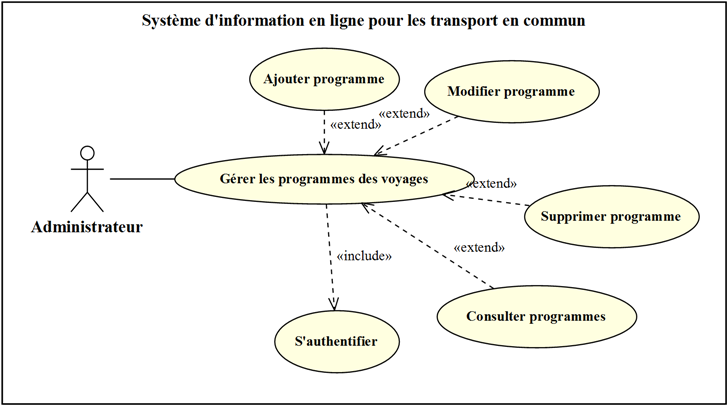
******

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

Description textuelle :

|  |  |
| --- | --- |
| Titre | Gérer les parcours |
| Acteur | Administrateur |
| Objectif | Ajouter, Modifier, Supprimer des parcours |
| Scénario nominal | 1. L’administrateur consulte l’application.  2. l’administrateur accède au menu parcours.  3. L’administrateur choisir l’opération à effectuer : Ajouter, modifier, supprimer, ou simple consultation.  4. En cas d’ajout ou de modification l’administrateur doit sélectionner tout d’abord les stations et définir les différents délais du parcours.  5. En cas de suppression l’utilisateur doit cliquer sur le bouton de suppression.  6. En cas de consultation le système affiche une liste qui contient tout les parcours. |

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

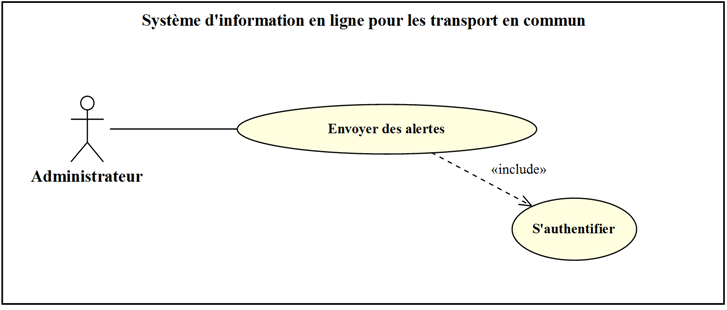
******

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

Description textuelle :

|  |  |
| --- | --- |
| Titre | Gérer les programmes |
| Acteur | Administrateur |
| Objectif | Ajouter, Modifier, Supprimer des programmes |
| Scénario nominal | 1. L’administrateur consulte l’application.  2. l’administrateur accède au menu programmes.  3. L’administrateur choisir l’opération à effectuer : Ajouter, modifier, supprimer, ou simple consultation.  4. En cas d’ajout ou de modification d’un programme le système affiche un formulaire que l’administrateur doit le remplir.  5. En cas de suppression l’utilisateur doit cliquer sur le bouton de suppression.  6. En cas de consultation le système affiche une liste qui contient tout les programmes. |

***III.2.2 Raffinement de cas d’utilisation « Envoyer des alertes»***

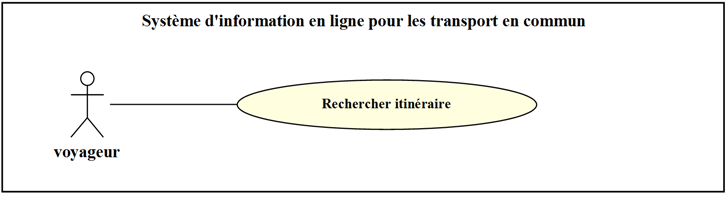
******

**Figure : Diagramme des cas d’utilisation « *Envoyer des alertes »***

Description textuelle :

|  |  |
| --- | --- |
| Titre | Envoyer des alertes |
| Acteur | Administrateur |
| Objectif | Envoyer des alertes au voyageur |
| Scénario nominal | 1. L’administrateur consulte l’application.  2. l’administrateur accède au menu alerte.  3. L’administrateur doit choisir le parcours concernés par l’alerte et écrire une description pour plus détailler la cause de l’alerte au voyageur.  4.Aprés la validation un e-mail sera envoyé au voyageur. |

***III.2.2 Raffinement de cas d’utilisation «Rechercher itinéraire»***

******

**Figure : Diagramme des cas d’utilisation « *Rechercher itinéraire»***

Description textuelle :

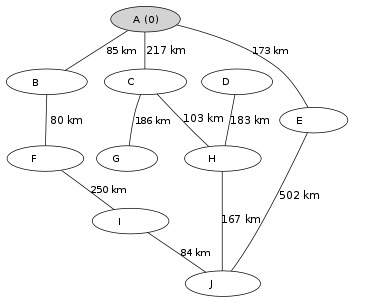
|  |  |
| --- | --- |
| Titre | Rechercher itinéraire |
| Acteur | Voyageur |
| Objectif | Rechercher le chemin le plus rapide entre deux stations en appliquant **l**'**algorithme de « Dijkstra »** |
| Scénario nominal | 1. Le voyageur consulte l’application.  2. Le voyageur choisie la fonctionnalité de recherche d’itinéraire.  3. Le voyageur choisie la station de départ, la station d’arrivée et la date de départ  4. Le système applique l’algorithme de Dijkstra pour déterminer le plus rapide chemin et affiche le résultat au voyageur. |

**Algorithme de Dijkstra**

En théorie des graphes, l'**algorithme de Dijkstra** sert à résoudre le problème du plus court chemin. Il permet, par exemple, de déterminer le plus court chemin pour se rendre d'une ville à une autre.

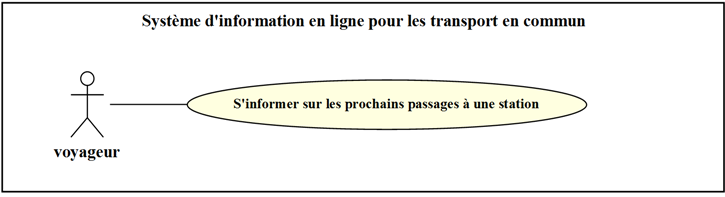
Principe :

Le principe de l'algorithme de Dijkstra est de trouver le chemin ayant le poids le plus faible entre 2 noeuds, sachant que le poids d'un chemin est la somme des poids des arêtes qui le composent.



**Figure : Exemple d’un graphe**

***III.2.2 Raffinement de cas d’utilisation «S’informer sur les prochains passage a une station»***

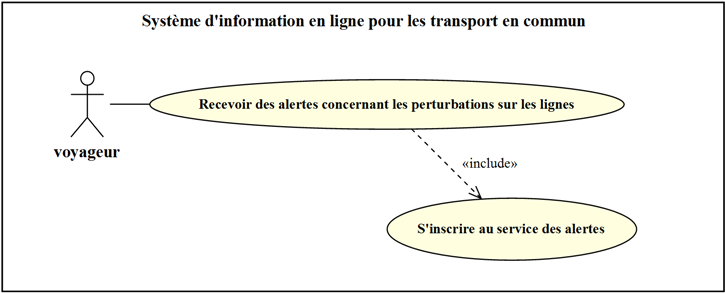
******

**Figure : Diagramme des cas d’utilisation «S’informer sur les prochains passage à une station»**

Description textuelle :

|  |  |
| --- | --- |
| Titre | S’informer sur les prochains passages a une station |
| Acteur | Voyageur |
| Objectif | S’informer sur l’heure de passage du véhicule de transport a une station choisie par le voyageur. |
| Scénario nominal | 1. Le voyageur consulte l’application.  2. Le voyageur choisie la fonctionnalité prochain passage.  3. Le voyageur choisi une station.  4. Le système recherche les lignes qui passent par la station choisie et affiche le résultat au voyageur.  5. Le voyageur choisi la ligne et sa destination  6. Le système calcule l’heure des prochains passages et affiche le résultat au voyageur |

***III.2.2 Raffinement de cas d’utilisation «Recevoir des alertes concernant les perturbations sur les lignes»***

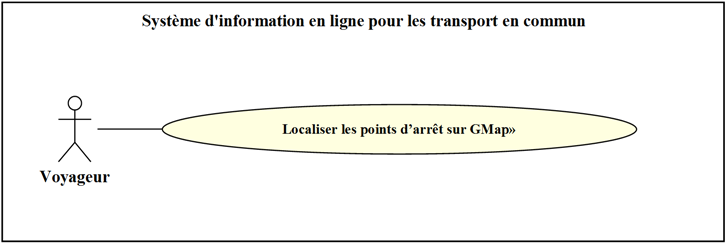
******

**Figure : Diagramme des cas d’utilisation «Recevoir des alertes concernant les perturbations sur les lignes»**

Description textuelle :

|  |  |
| --- | --- |
| Titre | Recevoir des alertes concernant les perturbations sur les lignes |
| Acteur | Voyageur |
| Objectif | Le voyageur reçoit des alertes à tout moment de perturbation ou des incidents sur ses lignes préférer |
| Scénario nominal | 1. Le voyageur consulte l’application.  2. Le voyageur choisie  3. Le système envoie un e-mail au voyageur au moment d’une perturbation |

***III.2.2 Raffinement de cas d’utilisation «Localiser les points d’arrêt sur GMap»***

******

**Figure : Diagramme des cas d’utilisation «Localiser les points d’arrêt sur GMap»**

Description textuelle :

|  |  |
| --- | --- |
| Titre | **Localiser les points d’arrêt sur GMap** |
| Acteur | Voyageur |
| Objectif | Le voyageur peut géolocaliser les stations autour d’une point choisie |
| Scénario nominal | 1. Le voyageur consulte l’application.  2. Le voyageur choisi la fonctionnalité de localiser station.  3. Le système marque les points d’arrêt sur Google map |