

République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة برج بوعريريج
Université de Bordj Bou Arreridj
Faculté de Mathématiques et d'informatique

Département d'informatique

*Rapport de projet de fin d'études
En vue de l'obtention du Diplôme de
Master en Informatique
Option : informatique Décisionnelle*



Thème :

***Proposition d'un système intelligent générique pour le transport
urbain par bus. Application à la ville de BBA***

Réalisé par :

- ***Ouidir Oussama***
- ***Leghlam Abde Rraouf***

Soutenu publiquement le : 15 septembre 2020 devant le jury composé de :

Président : Mrs : FARES NOUR EL HOUDA *M.A à L'U. El Bachir El Ibrahimi-BBA*
Examineur : Mr : SAHA ADEL *M.A à L'U. El Bachir El Ibrahimi-BBA*
Encadreur : Mr : MAACHE SALEH *MAA à L'U. El Bachir El Ibrahimi-BBA*

Promotion :2019/2020



Remerciement

Nous avons à remercier gracieusement notre encadreur
monsieur **MAACHE SALAH** Qui a bien assuré la
direction de ce mémoire avec ses précieuses aides
Et orientations ainsi que Des enseignants qui ont
contribués à notre formation.

Transport Intelligent (**Transport Tracker**) qui
Notre encadreur proposé ce thème de projet, et
Tous ceux qui Ont En L'amabilité de
Nous aider de près on de loin et plus sans
Oublier notre Promotion et notre famille
<<**Informatique Décisionnelle**>>.

Sommaire

Remercîment.....	I
Table des Matières	II-IV
Liste des Figures.....	V
Liste des Tableaux.....	VI

Chapitre 01

INTRODUCTION GENERAL ET PROBLEMATIQUE

1 . Le contexte	9
Les domaines concernés.....	9
1.2.Les Motivations	10
1.3.Les Problématique.....	11
1.3.Les solutions	12
1.4. Le Plan de mémoire.....	13
CONCLUSION	13

Chapitre 02

Etat de l'Art

Introduction	15
2.1. Les origines des systèmes de transport intelligent	15
2.1.1. Premières études (années 60 – années 70)	15
2.1.2. Premières applications (années 80 –milieu des années 90)	15
2.1.3. Les grands projets (milieu des années 90 – aujourd'hui)	15
2 .2. Application pour les transports intelligents	16
2.2.1. Information à l'intention des voyageurs	16
2.2.2. Gestion du trafic	16
2.2.3. Services de transport en commun	17

2.2.4. Services de paiement électronique	17
2.2.5. Exploitation de véhicules commerciaux	17
2.2.6. Gestion des urgences	17
2.2.7. Systèmes avancés de sécurité des véhicules	17
2.2.8. Gestion de l'information	17
2.2.9. Gestion de travaux d'entretien et de construction	17
2.3. les projets au domaine de transport intelligent	17
2.3.1. FLOW Labs	18
2.3.2. Le Centre de gestion de la mobilité urbaine (CGMU)	19
2.4. Application mobile dans le domaine de transport	20
2.4.1. Application Citymapper	20
2.4.2. Application Moovit	21
Conclusion.....	Error! Bookmark not defined.

Chapitre 03

Architecture et Modélisation

Introduction	25
3.1. Planning des tâches	25
3.2. Architecture proposée pour un système de transport intelligent	25
3.2.1 description du système	27
3.2.2. Composants du système	27
3.2.3. les services de transport intelligent de notre modèle proposé	28
3.3. Identification des acteurs du système	30
3.4. Identification des cas d'utilisation du système	30
3.5. Conception	31
3.5.1. Diagramme de classe	31
3.5.2. Diagramme de cas d'utilisation	37
3.5.2.1 Diagramme de Cas d'utilisation d'utilisateur, administrateur et capteurs	37
3.5.2.2. Diagramme de Cas d'utilisation d'authentification	38
3.5.2.3. Diagramme de Cas d'utilisation capteur bus	39
3.5.2.4. Diagramme de Cas d'utilisation capteur arrêt	40

3.5.2.5. Diagramme de Cas d'utilisation Capteurs –Serveur:	41
3.5.2.6. Diagramme de cas d'utilisation Utilisateur-Serveur	42
3.5.3. Diagramme de séquence	43
3.5.3.1. Diagramme de séquence d'authentification	43
3.5.3.2. Diagramme de séquence de création et activation des capteurs	44
3.5.3.3. Diagramme de séquence d'utilisateur – application mobile	44
3.5.3.4. Diagramme de séquence capteur- serveur	46
Conclusion	46

Chapitre 04

IMPLEMENTATION ET BILAN

Introduction	48
4.1.Choix des outils et langages utilisés	48
4.2.Flux de données	50
4.3.Presentation de l'application	52
4.3.1. Coté web	52
4.3.2. Coté Mobile	58
Conclusion	59
Conclusion Générale	61
Bibliographie	63

Liste des figures

FIGURE 1: BUS ETUS-----	9
FIGURE 2: LES DOMAINES CONCERNES -----	10
FIGURE 3: PERT DE TEMPS -----	11
FIGURE 4: ARCHITECTURE DES STI GENERALE -----	16
FIGURE 5: COLLECTE DE DONNEES -----	18
FIGURE 6: OPTIMISATION DE L'IA -----	18
FIGURE 7: STI DE CGMU-----	20
FIGURE 8: IHM DE CITYMAPPER -----	21
FIGURE 9: IHM DE MOOVIT -----	22
FIGURE 10: L'ORGANISATION DES TACHES QU'ON A PREVUE PENDANT LA PERIODE DE NOTRE TRAVAIL -----	25
FIGURE 11: ARCHITECTURE DU SYSTEME-----	26
FIGURE 12: DIAGRAMME DE CLASS.-----	32
FIGURE 13: CAS D'UTILISATION GENERALE. -----	37
FIGURE 14: CAS D'UTILISATION D'AUTHENTIFICATION -----	38
FIGURE 15: CAS D'UTILISATION DE CAPTEUR DE BUS-----	39
FIGURE 16: CAS D'UTILISATION DE CAPTEUR D'ARRET -----	40
FIGURE 17: CAS D'UTILISATION CAPTEURS -SERVEUR-----	41
FIGURE 18: CAS D'UTILISATION UTILISATEUR-SERVEUR -----	42
FIGURE 19: DIAGRAMME DE SEQUENCE D'AUTHENTIFICATION -----	43
FIGURE 20: DIAGRAMME DE SEQUENCE DE CREATION ET ACTIVATION DES CAPTEURS -----	44
FIGURE 21: DIAGRAMME DE SEQUENCE D'UTILISATEUR – APPLICATION MOBILE-----	45
FIGURE 22: DIAGRAMME DE SEQUENCE CAPTEUR- SERVEUR-----	46
FIGURE 23: DIAGRAMME DE FLUX DE DONNEES CONCEPTUEL -----	51
FIGURE 24: AUTHENTIFICATION -----	53
FIGURE 25: TABLEAU DE BORD D'ADMINISTRATEUR-----	53
FIGURE 26: METHODE D'AJOUTER UTILISATEUR-----	54
FIGURE 27: AJOUTER UTILISATEUR AVEC EMAIL -----	54
FIGURE 28: TEST APPLICATION SUR CLOUD -----	55
FIGURE 29: SUIVI DES ACTIFS EN TEMPS REEL -----	56
FIGURE 30: MANIPULER LA BASE DE DONNEE -----	57
FIGURE 31: HEBERGEMENT D'APPLICATION -----	57
FIGURE 32: APPLICATION MOBILE -----	58
FIGURE 33: TRACKER D'UTILISATEUR SUR BASE DE DONNEE.-----	59

Liste des Tableaux

TABLEAU 1: CAPTEURS INTELLIGENT LOGIQUE ET PHYSIQUE -----	28
TABLEAU 2: IDENTIFICATION DES CAS D'UTILISATION DU SYSTEME -----	30
TABLEAU 3: L'IDENTIFICATION DES CLASSES -----	31
TABLEAU 4: CLASS AGENCE -----	33
TABLEAU 5: CLASS ROUTE -----	33
TABLEAU 6: CLASS VOYAGE -----	34
TABLEAU 7: CLASS ARRET -----	34
TABLEAU 8: CLASS TEMPS_D'ARRET -----	35
TABLEAU 9: CLASS BUS -----	35
TABLEAU 10: CLASS UTILISATEUR -----	36
TABLEAU 11: CLASS CAPTEUR -----	36
TABLEAU 12: CLASS ADMINISTRATEUR -----	36
TABLEAU 13: IMPLEMENTATION DES COMPOSANTS -----	50
TABLEAU 14: LES MODULES -----	51
TABLEAU 15: LES AUTRES MODULES -----	52



CHAPITRE 01

**INTRODUCTION GENERAL Et
PROBLEMATIQUE**

1 . Le contexte :

La wilaya de Bordj Bou Arreridj s'étend sur une superficie de 10 982 km². La population résidente est estimée à 1 000 475 habitants en 2020, le besoin des habitants de cette ville a été augmenter, mais les moyens de communication et les nouvelles technologies de communication et d'information tout seul ne répondent pas aux besoins des citoyens, et pour cela le transport intelligent va résoudre leurs problèmes.

A BBA, d'importants problèmes de circulation et de congestion sont apparus, le bus va se retarder, les clients de transport public commun ne connaissent pas l'heure exacte de passage de leur bus. Donc le transport intelligent, résoudre le problème de retard, en informant le client de l'heure exacte de l'arrivée des prochains bus et des cas spéciaux inattendus et les arrêts en temps réels et si le bus est vide, ou un bus qui a des options avancées comme la climatisation la disponibilité de wifi gratuit, place pour les handicapés ...etc.

Les informations seront diffusées aux clients à partir des différents outils de communication, qui peut être une application mobile, site web de la société, par des SMS, panneaux d'affichage sur les routes...etc.

Alors et pour cela on a proposé des solutions pour les sociétés (voir la figure 1 [1]) et les clients pour suivre et gérer les actifs en mouvement des bus, améliorer le transport et aider le client et offrir une visibilité en temps réel de nos bus, itinéraires et horaires.



Figure 1: bus ETUS

Les domaines concernés (voir la figure 2) :

- ***Le transport*** est le déplacement d'objets, de marchandises, ou d'individus d'un endroit à un autre. [2]
- ***Les transports urbains*** sont les différents moyens de transport propres à une ville ou à une agglomération, adaptés à cet environnement. [3]

- *L'Internet des Objets* dans les transports Créez une base solide et tirez parti de l'Iot pour améliorer l'expérience, la sécurité et l'efficacité des passagers. [4]
- *Les systèmes de transport intelligents* un ensemble hétérogène qui combine des technologies des télécommunications et de l'information avec l'ingénierie des transports.[5]
- *Les systèmes d'aide à la décision* améliorent l'efficacité opérationnelle et la performance en renforçant la capacité des parties prenantes à prendre des décisions plus rapides et plus intelligentes basées sur les informations, plutôt que sur la seule intuition. [6]

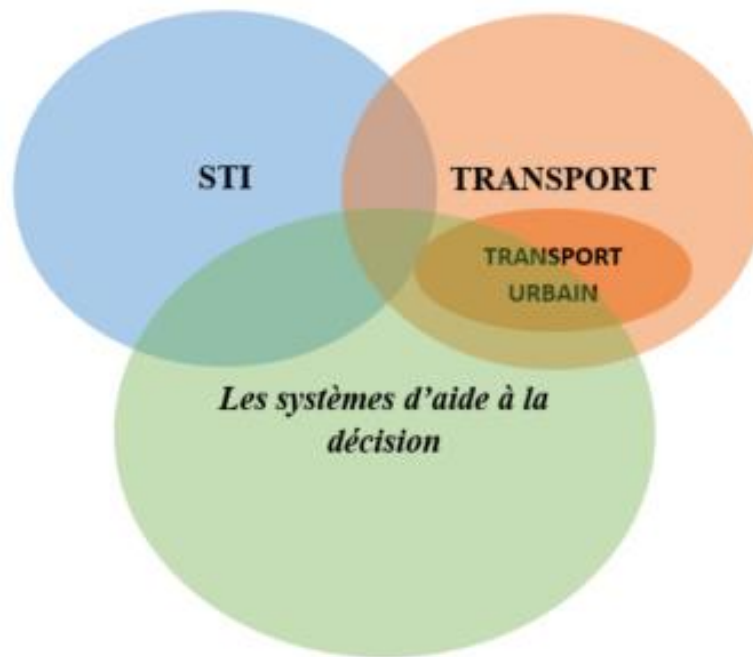


Figure 2: Les domaines concernés

1.2.Les Motivations :

Le choix du projet de fin d'études, est une étape essentielle, Plusieurs motivations ont orienté notre choix :

- Les systèmes de transport intelligent sont des nouveaux domaines qui nécessitent une exploration en profondeur.
- L'occasion de travailler avec des nouvelles technologies telle que la technologie Google Cloud Platform.
- Notre sujet fait le lien entre l'informatique décisionnelle et de gestion.

1.3. Les Problématique:

Nous allons approcher les principaux problèmes rencontrés en matière de transport urbain dans le monde et en Algérie(BBA) et qui traduisent les difficultés à organiser les transports pour répondre aux besoins des citoyens.

Nous allons citer quelques problèmes :

- ***Manque de moyens de transport en commun :***

Les habitants de cette ville se demandent pourquoi la ville BBA ne s'est pas dotée de plusieurs bus de l'Etablissement public de transport urbain (ETUS) comme les autres grandes villes du pays.

- ***Perte du temps :***

Dans certains quartiers les passagers perdent plus de 20 minutes à attendre. Beaucoup de passager rate chaque jour leurs bus soit parce que ces bus arrivent tôt aux stations ou bien parce que le bus a dépassé la station. Les passagers ne savent jamais les horaires de sortie des bus. Pour cela plusieurs personnes préfèrent prendre leurs voitures climatisées avec tous les comforts et, plutôt de prendre le transport en commun. (Voir la figure 3 [7])



Figure 3: Pert de temps

- ***Circulation routière (Embouteillage) :***

C'est un problème qui dure depuis des années, l'Algérie souffre énormément des embouteillages, tout ça car il y a un manque d'organisation.

La circulation routière se complique davantage au fil des jours dans la ville de BBA, notamment au niveau des grandes artères aux heures de pointe, cette densité de la circulation générant des embouteillages au niveau des grands axes routiers de la ville a des conséquences très grave.

- ***Absence des arrêts :***

Certains quartiers ne sont pas desservis, telle que la ligne El Annassar, la population se rabat sur les taxis clandestins ou la marche à pied.

Les habitants de El Annassar continuent à trouver d'énormes difficultés pour se déplacer en dehors de leur quartier.

Les informations concernant les arrêts, par exemple la température de l'arrêt, les bus passant par chaque arrêt, ainsi que le nombre des passagers en attente de bus, les services de l'arrêt comme le wifi gratuit.

- ***Manque d'infrastructure intelligente :***

Une manque d'infrastructure intelligente comme des panneaux d'affichage, les plaques pour s'informer, et en générale il y a manque d'information en temps réel qui influence éventuellement à la décision du passager.

1.3.Les solutions :

La ville est environnement dans lequel vit une très large part de la population qui utilise le moyen de transport urbain, elle nécessite une bonne gestion des flux des voyageurs dans l'espace urbain dans le but de satisfaire les passagers.

L'objectif de la modélisation des réseaux de transport urbain est d'obtenir une représentation utilisable et une description parfaite pour pouvoir l'exploiter.

La solution proposé est que les systèmes de transport doivent être intelligent ou tous les composant du système et l'infrastructure doivent être intelligent, qui utilise les nouvelles technologies comme les smart phone, les réseaux de Magnétomètres sans fils, les capteurs sans fil et les caméras de surveillance dont toute ses activités et ses relations tel que tous les objets de système vont être connectés entre eux et avec l'environnement ainsi que l'utilisation des aspects algorithmiques de régulation et de prédiction du trafic, et l'utilisation optimale des données relatives à la route, à la circulation et aux déplacements.

Pour cela on a proposé des solutions :

La solution propose pour les gérants de la société un système de gestion d'information et donnant un aperçu des actifs suivis. Il affiche des informations à l'aide de nouvel technologie, avec les données de bus et de localisation.

Le passager aura à sa disponibilité deux moyens d'information : une application mobile et un écran d'affichage sur les différentes stations qui utilise le fournisseur de localisation fusionné, pour signaler son emplacement à la base de données en temps réel et prédit les temps de trajet et stocke les emplacements des bus, accrochée à la route et fournit une synchronisation des données en temps réel et signaler son emplacement à la base de données en temps réel et afficher une carte stylisée montrant les emplacements et les itinéraires des bus.

1.4. Le Plan de mémoire :

Nous avons élaboré un plan de travail qui est scinde en quatre chapitres :

❖ **Le 1^{er} chapitre :** introduction général et problématique

Est le point de départ, il consiste une vue globale sur le projet, on a présenté le projet ainsi que la discussion de la problématique avec des exemples réels, ainsi que les solutions proposées, les objectifs attendu par utilisateurs.

❖ **Le 2^{eme} chapitre :** état de l'art

Pour une étude sur les modèles existant et donne une description sommaire du model propose et définir objectifs de se model.

❖ **Le 3^{eme} chapitre :** Architecture et modélisation et conception

Port sur l'analyse des besoins décrits dans le chapitre précédent, il met la description du model ou du système, ainsi que la représentation de notre système utilisant quelque diagramme UML, et il représente l'architecture propose pour le système.

❖ **Le 4^{eme} chapitre :** Implémentation et bilan

Il présente notre choix de l'environnement et du langage de réalisation de l'application et notre système côté technique, comment il fonctionne, , analysés et conçus dans les précédents chapitres.

CONCLUSION :

Dans ce chapitre on a présenté en premier lieu notre travail de manier générale, commençons par le contexte de notre projet ainsi que les motivations de ce dernier, en deuxième lieu on a introduit les problématiques de transport intelligent, et en fin les solutions qu'on a proposées pour résoudre ces problèmes pour améliorer le transport public.



CHAPITRE 02



Etat de l'Art

Introduction :

Après avoir fini de présenter tous les notions de base de notre projet, on va présenter dans ce chapitre l'état de l'art de système de transport intelligent, objectifs, les applications qui existent (les modèles existants) et les travaux de recherche en ce domaine.

2.1. Les origines des systèmes de transport intelligent :

2.1.1. Premières études (années 60 – années 70) :

Les attrayants projets de cette époque c'étaient : le programme de recherche CACS « Compréhensive Automobile trafic Control System » qui s'est étendu de 1973 à 1979. Il s'agissait du premier partenariat public-privé dans le monde ayant testé en zone urbaine un système de navigation interactif embarqué possédant un écran ; le projet ERGS « Electronic Route Guidance System » aux États-Unis et le projet similaire ALI « Auto fahrer Leitund Information System » en Allemagne. Malheureusement, vu les faibles capacités de calcul des systèmes embarqués de cette période et de l'importante puissance requise pour le serveur principal, ces projets n'ont jamais vu le jour. [8]

2.1.2. Premières applications (années 80 –milieu des années 90) :

La période des années 80 a été marquée par plusieurs avancées technologiques. Le domaine de transport a largement bénéficié de ces évolutions qui ont abouti à l'essor de nombreux projets. Ces derniers s'intéressaient au développement des solutions réelles, concrètes et fonctionnelles afin de les commercialiser. Parmi ces projets, on cite les issues européennes représentées par le programme EUREKA créé en 1986 et son projet PROMETHEUS « Program for Européen Traffic with Highest Efficiency and Unprecedented Safety », qui est une initiative soutenue par des constructeurs automobiles visant à développer les transports intelligents. Globalement, les projets cités ci-dessus issus de nombreux efforts (privés et publics), ont aidé à accélérer les recherches dans le domaine des STI. [9]

2.1.3. Les grands projets (milieu des années 90 – aujourd'hui) :

Paris, en 1994 avec l'organisation du premier congrès mondial sur les STI, la phase actuelle de l'histoire a commencé. Cette phase est fortement marquée par sa dimension mondiale. En 1994 aux États-Unis, le projet IVHS a été renommé ITS American « Intelligent Transportation System America » pour étaler le champ de recherche. [10]

2.2. Application pour les transports intelligents :

En va présenter tous les services de transport intelligent qui ont apparu jusqu'à maintenant (voir la figure 4 [11]) :

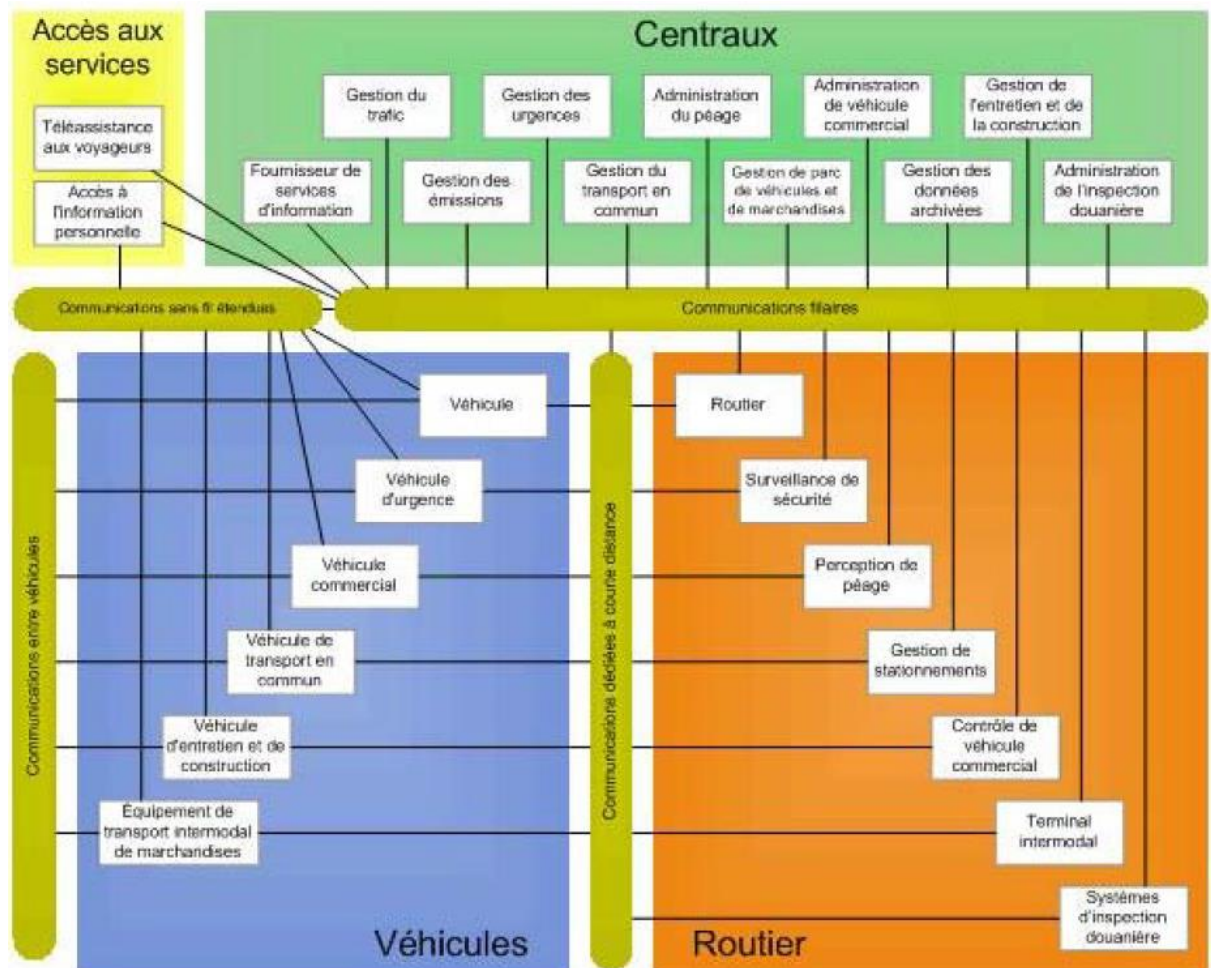


Figure 4: Architecture des STI générale

2.2.1. Information à l'intention des voyageurs :

- Information sur les déplacements avant le départ.
- Guidage routier et navigation.

2.2.2. Gestion du trafic :

- Régulation du trafic.
- Gestion des incidents.

2.2.3. Services de transport en commun :

- Gestion du transport et commun.
- Information en cours de route.

2.2.4. Services de paiement électronique :

- Services de paiement électronique.

2.2.5. Exploitation de véhicules commerciaux :

- Vérification électronique de véhicules commerciaux.
- Inspection automatisée de sécurité routière.

2.2.6. Gestion des urgences :

- Notification des urgences et sécurité personnelle.
- Gestion des véhicules d'urgence.

2.2.7. Systèmes avancés de sécurité des véhicules :

- Prévention des collisions longitudinales.
- Prévention des collisions latérales.

2.2.8. Gestion de l'information :

- Données archivées.

2.2.9. Gestion de travaux d'entretien et de construction :

- Activités d'entretien et de construction.

2.3. Les projets au domaine de transport intelligent :

Parmi les projets plus importante au domaine de transport intelligent :

2.3.1. FLOW Labs :

Le programme de recherche **FLOW Labs** qui s'est étendu de 2018 à 2020 :

- Les véhicules connectés, l'intelligence artificielle, les méga données et l'IoT créent de nouvelles opportunités intéressantes pour la gestion du trafic. Flow Labs construisent **la première plateforme de gestion du trafic autonome au monde** pour permettre des routes plus claires, plus propres et plus sûres pour citoyens.
- Il collecte et organise des millions de points de données d'exploitation en utilisant l'IA pour optimiser les normes et procédures d'exploitation. Les données que la plateforme logicielle basée sur le cloud ingère sont transformées et améliorées en données significatives qui permettent aux points de données d'exploitation d'alimenter automatiquement les innovations, d'augmenter l'efficacité et d'innover la mobilité en quelques clics.
- Il utilise des données provenant de plusieurs sources (voir la figure 5 [12]) pour vous aider à **comprendre** avec précision **ce qui se passe sur routes**, y compris les temps de trajet, la vitesse des véhicules, les files d'attente ou les mouvements de virage.

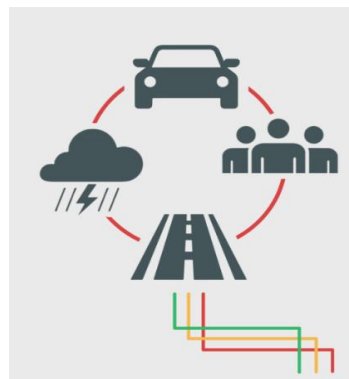


Figure 5: Collecte de données

- Plateforme propriétaire d'apprentissage automatique et d'intelligence artificielle **élabore les meilleurs plans de synchronisation** (voir la figure 6 [13]) pour besoins et prédit avec précision performances sur le terrain.

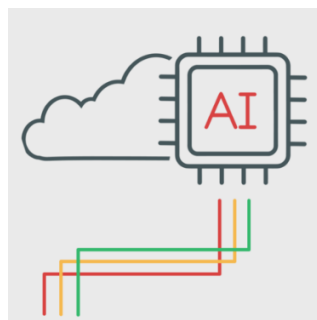


Figure 6: Optimisation de l'IA

- En quelques clics, nouveaux plans de synchronisation sont prêts à l'emploi et peuvent être **mis en œuvre automatiquement et instantanément** sur le terrain.

2.3.2. Le Centre de gestion de la mobilité urbaine (CGMU) :

La nouvelle version **CGMU** qui s'est étendu de 2013 à 2018 :

Le CGMU est une salle de surveillance du réseau artériel où se trouvent divers équipements : mur vidéo, système de gestion des caméras de surveillance, etc. Ces équipements permettent à la Ville de rassembler l'information sur l'état du réseau routier montréalais. C'est par cette salle que transite toute l'information provenant du réseau artériel et celle que l'on veut communiquer aux différents outils sur le terrain, dont les feux de circulation ou les panneaux de jalonnement dynamique, par exemple. Ce centre comporte de nombreux avantages (voir la figure 7 [14]) :

- Gestion des feux de circulation en temps réel.
- Amélioration de la circulation des autobus.
- Surveillance accrue du réseau routier et d'infrastructures.
- Optimisation des interventions et de la circulation en cas d'urgence.
- Échange rapide d'information avec les partenaires soit le Service des incendies de Montréal (SIM), la Société de transport de Montréal (STM), le ministère des Transports, de la Mobilité durable et de l'Électrification des transports (MTMDET) et la sécurité civile.
- Communication plus efficace avec les centres de contrôle des partenaires.

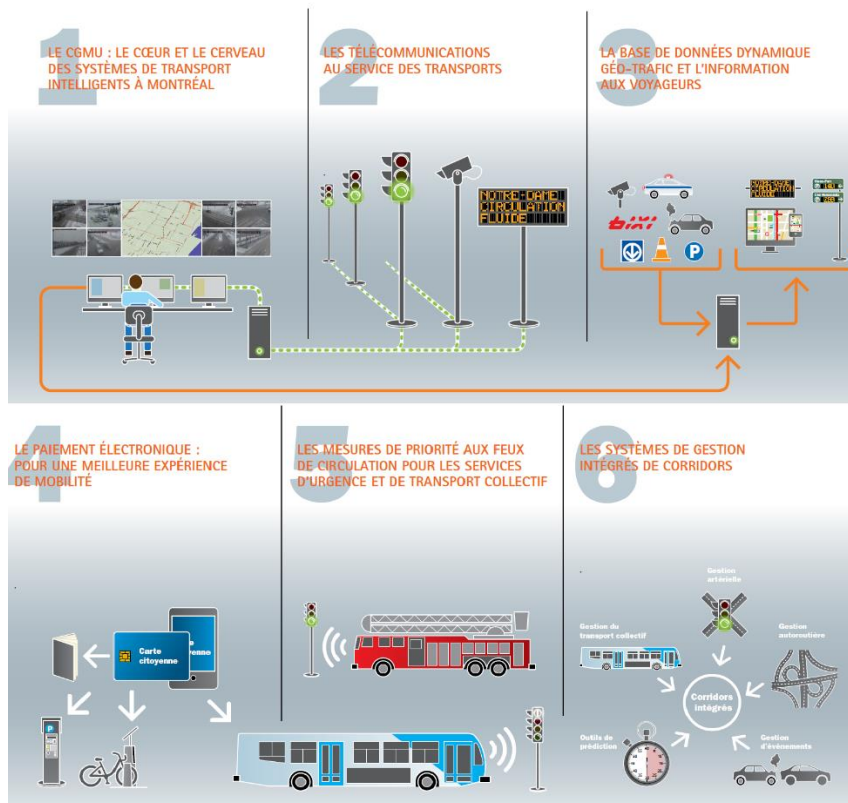


Figure 7: STI de CGMU

2.4. Application mobile dans le domaine de transport :

2.4.1. Application Citymapper :

Tout est très simple avec **Citymapper** (voir la figure 8 [15]). À partir de votre position et du lieu d'arrivée, l'appli génère plusieurs itinéraires équivalents, indiquant les temps de trajet, tarifs estimés et prochains passages aux stations de départ. Les horaires sont mis à jour, et l'on vous communique même le wagon dans lequel monter pour être au plus proche de la sortie une fois à quai. L'état du trafic est pris en compte avec possibilité de configurer des alertes pour les lignes régulièrement empruntées.

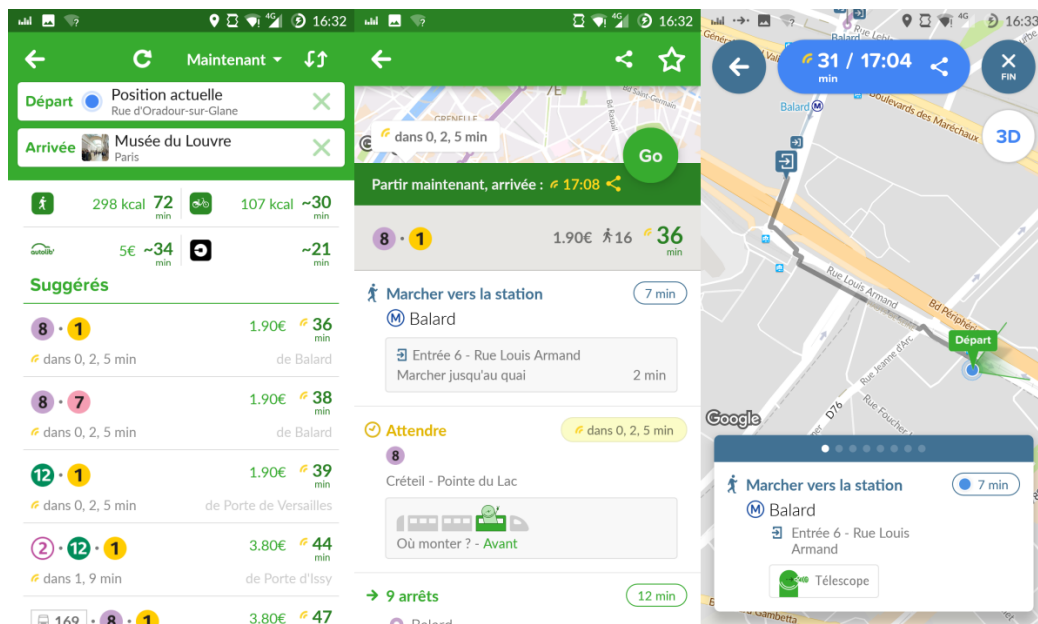


Figure 8: IHM de citymapper

CRITIQUE :

C'est une application simple à utiliser et elle a pas mal d'utilité mais Le problème qu'elle ne détecte pas par exemple les retarde des bus parce qu'elle ne donne pas des informations en temps réel, par contre elle donne des informations statiques.

2.4.2. Application Moovit :

Moovit mise tout sur l'exhaustivité de son application (voir la figure 9 [16]). Saisissez votre point de rendez-vous et l'appli produit une série d'itinéraires suggérés. Les filtres affinent les résultats selon les types de trajets et de moyens de transport souhaités. Les propositions estiment la durée du voyage, marche à pied comprise. L'info trafic dresse instantanément un état des lieux général du réseau. Intégré au système de navigation embarquée de **Moovit**, un module d'alerte prévient les voyageurs lorsqu'il est temps de descendre. Pratique pour qui a préféré ranger son smartphone durant le trajet. Petit plus, l'application se comporte comme un Waze des réseaux publics et offre à ses utilisateurs la possibilité d'informer la communauté sur les conditions de transport en temps réel : arrêts manqués, taux de remplissage de la rame, incidents en cours, températures.

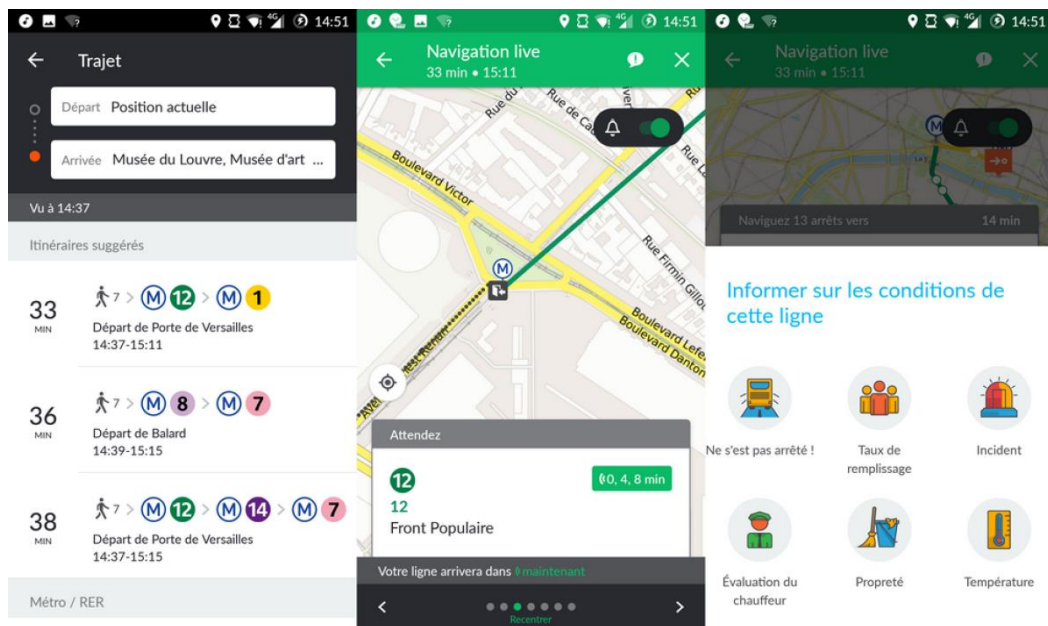


Figure 9: IHM de moovit

CRITIQUE :

C'est une application simple à nombreux filtres permettant d'adapter l'itinéraire aux conditions de transport souhaitées, module d'alerte pour signaler la prochaine descente lorsque l'écran est éteint, mais Interface qui manque parfois de clarté.

En bénéfice de ses travaux et en va trouver solutions pour les problèmes :

- Une meilleur décision base sur l'information, l'intégration des différent systèmes et l'interopérabilité.
- La circulation s'effectue en continu sur toutes les voies sans délai d'attente.
- Information en temps réel pour les passagers avant de prendre de bus et pendant les déplacements.
- Uniformisation des décisions liées à la mobilité.
- L'accès direct offre une solution de rechange aux nombreux axes congestionnés de la région.
- Le système de péage intelligent est basé sur une technologie qui a fait ses preuves dans le monde.
- L'utilisation optimale des données relatives aux routes, à la circulation et aux déplacements.

Conclusion :

Dans ce chapitre en as présenté en détail les domaines des systèmes de transport intelligent et travaux de recherche et quelque application en ce domaine pour résoudre les problèmes liés au trafic et améliorer l'exploitation des systèmes de transport.



CHAPITRE 03



Architecture et Modélisation

Introduction :

Dans ce chapitre, en va présenter en premier lieu description du model ou du system qu'on a proposé en suite en va donner notre architecture informatique du model qu'on a proposé, en suite en va suivrons le processus UP, donner notre conception commençons par une définition des principaux acteurs de système, les besoins fonctionnel et non fonctionnel du projet, tout cela avec les diagrammes des cas d'utilisation et les diagrammes de séquences.

3.1. Planning des taches :

La planification des taches c'est une étape important avant le début des projets (voir la figure 10).

	Février	Mars	Avril	Mai	Juin	Juillet	Aout	Septembr e
Le choix de thème								
Collection des informations.								
Spécification des Besoins								
Conception et Modélisation								
Réalisation, Implémentation								
Rédaction du Rapport								

Figure 10: l'organisation des taches qu'on a prévue pendant la période de notre travail

Tous simplement en doit appliquer le processus UP :

3.2. Architecture proposée pour un système de transport intelligent :

Comme tous les systèmes, le système de transport intelligent a une architecture, l'architecture qu'on a proposée pour ce système est monter dans la (figure 11)

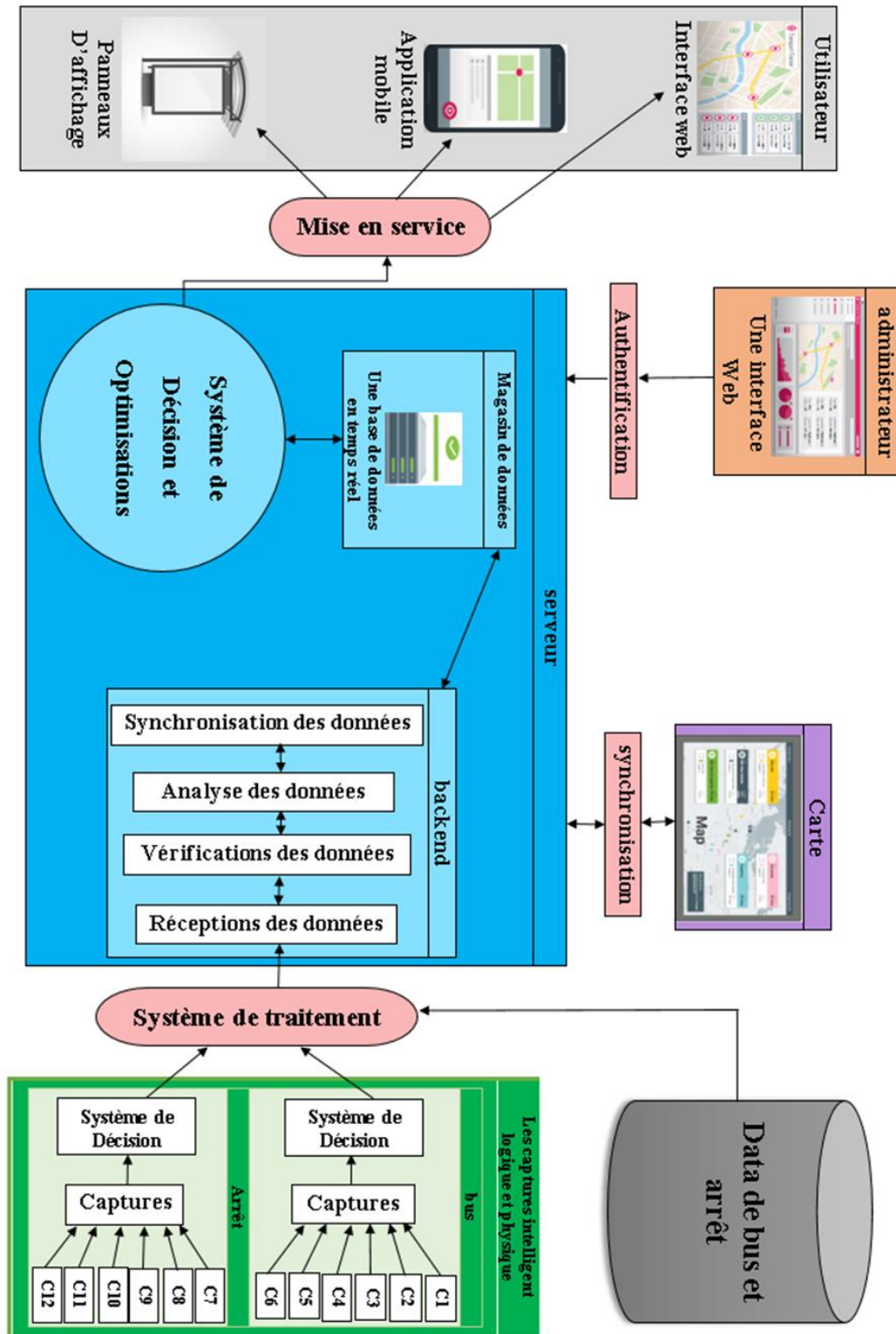


Figure 11: Architecture du système

3.2.1 description du système :

Notre système est un ensemble d'applications conçues pour suivre une gamme d'actifs en mouvement (tels que des bus) et les visualiser sur une carte en direct. Le référentiel se compose d'un certain nombre de composants, représentant les différentes parties mobiles du système de suivi de bus. Il comprend une application Android sur les appareils Android et déployée sur les bus suivis, ainsi qu'une interface utilisateur d'administration. Le dépôt contient également le côté écran du système, conçu pour alimenter des téléviseurs LCD de 60 pouces. Il existe `backend`, qui reçoit les emplacements signalés par l'application Android, ainsi qu'un calendrier fourni au format intelligent et effectue des mises à jour régulières d'une base de données en temps réel. Il existe aussi un système de décisions par exemple en cas d'un accident dans un chemin, il informe le client de ce cas, et il lui propose un autre chemin automatique et il s'informe aussi leur client. Le client reçoit les mises à jour de la base de données et les dessine sur le téléviseur LCD.

3.2.2. Composants du système :

- **LOCALISATEUR DE BUS :**

Une application Android que vous installez sur des appareils mobiles et que vous mettez dans chaque bus que vous souhaitez suivre.

- **DATA DE BUS ET ARRET :**

Ensembles des données définit un format de fichier sur les bus et les arrêts permettent aux agences de publier leurs informations de transports et aux développeurs de créer des applications qui utilisent ces données de manière interopérable.

- **APERÇU DE L'ADMINISTRATEUR :**

Une interface Web qui donne aux administrateurs un aperçu des bus suivis.

- **BACKEND :**

Le composant côté serveur qui gère l'état de la base de données en temps réel (réception, analyse, vérification des données avec la synchronisation) et augmenter la charge de communication avec la base de donnée.

• **CARTE :**

Une interface Web qui affiche les informations de planification et les emplacements des bus.

• **MAGASIN DE DONNEES :**

Une base de données en temps réel qui stocke les données de localisation du bus envoyées par le localisateur de bus et fournit une synchronisation des données en temps réel avec la carte et la vue d'ensemble de l'administrateur.

• **SYSTEME DE DECISIONS :**

C'est cerveau du système de société, c'est lui qui donne les décisions finales d'après les informations traitées et il indique les cas spéciaux, dans le système.

• **CAPTEURS INTELLIGENT LOGIQUE ET PHYSIQUE :**

Tableau 1:Capteurs intelligent logique et physique

Objet	capteur
<p style="text-align: center;">Bus</p>	<ul style="list-style-type: none"> • C1 : Capteur physique de position de bus. • C2 : Capteur physique de nombre de place vide et occuper. • C3 : Capteur physique de l'état de bus. • C4 : Capteur logique de position de bus. • C5 : Capteur logique de nombre de place vide et occuper. • C6 : Capteur logique de l'état de bus.
<p style="text-align: center;">Arrêt</p>	<ul style="list-style-type: none"> • C7 : Capteur physique de nombre de personne attendre dans l'arrêt. • C8 : Capteur physique de caméras vidéo. • C9 : Capteur physique des bus les plus proches pour chaque arrêt. • C10 : Capteur logique de nombre de personne attendre dans l'arrêt. • C11 : Capteur logique de caméras vidéo. • C12 : Capteur logique des bus les plus proches pour chaque arrêt.

3.2.3. Les services de transport intelligent de notre modèle proposé :

En va présenter les services de transport intelligent de notre modèle proposé :

✓ INFORMATION A L'INTENTION DES VOYAGEURS :

- Information sur les déplacements avant le départ.
- Information à l'intention des conducteurs en cours de route.
- Guidage routier et navigation.
- Information sur les services aux voyageurs.

✓ GESTION DU TRAFIC :

- Régulation du trafic.
- Gestion des incidents.
- Avertissement dynamique et application des règlements automatisés.

✓ SERVICES DE TRANSPORT EN COMMUN :

- Gestion du transport en commun.
- Information en cours de route.

✓ GESTION DES URGENCES :

- Notification des urgences et sécurité personnelle.
- Gestion des véhicules d'urgence.
- Intervention et évacuation en cas de catastrophe.

✓ SYSTEMES AVANCES DE SECURITE DES VEHICULES :

- Prévention des collisions longitudinales.
- Prévention des collisions latérales.
- Prévention des collisions aux intersections.
- Visibilité accrue pour éviter les collisions.
- Contrôle de l'état de sécurité.
- Déploiement de dispositifs de retenue avant collision.

✓ GESTION DE L'INFORMATION :

- Données archivées.

3.3. Identification des acteurs du système :

Un acteur est une entité qui définit le rôle joué par un utilisateur ou par un système qui interagit avec le système modélisé. Les types d'acteur qui participent à notre système ont les suivant :

- Utilisateur de site web et l'application mobile,
- Administrateur.
- Les captures intelligent.

3.4. Identification des cas d'utilisation du système :

Tableau 2: Identification des cas d'utilisation du système

ACTEUR	CAS D'UTILISATION
Utilisateur	<ul style="list-style-type: none"> ▪ Consulter les tarifs des déplacements. ▪ Consulte l'état de route. ▪ Consultation en temps réel la position et des emplacements et les itinéraires des bus. ▪ Choisir le bus. ▪ Choisir l'arrêt. ▪ Choisir le plus court chemin.

ACTEUR	CAS D'UTILISATION
Administrateur	<ul style="list-style-type: none"> ▪ Voir un aperçu rapide de tous les actifs suivis (la position et des emplacements et les itinéraires des bus, prix des déplacements...etc.) ▪ Géré et sécurisé le système (administration, réseau de transport, les bus, les arrêts, prix des déplacements... etc.) ▪ Consulte le chiffre d'affaire la société.

ACTEUR	CAS D'UTILISATION
<p>Captures Intelligent</p>	<ul style="list-style-type: none"> ▪ Collecter toutes les informations concernons les emplacements des bus, arrêts, les itinéraires des bus, les utilisateurs. ▪ Envoyer une information au système en cas d'urgence. ▪ Envoyer ses informations au système (administrateur).

3.5. Conception :

3.5.1. Diagramme de classe :

Dans cette section nous aborderons l'identification des objets métiers à partir de l'expression initiale des besoins de notre projet.

nous utilisons le diagramme de classes (voir la figure 12) pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci :

Tableau 3:l'identification des classes

Class	Définit
Agence	Agences de transports en commun ayant un service représenté dans cet ensemble de données.
Route	Itinéraires en transports en commun. Un itinéraire est un ensemble de trajets présentés aux usagers comme relevant du même service.
Voyage	Trajets pour chaque itinéraire. Un trajet est une série d'au moins deux arrêts desservis à des horaires précis.
Bus	représente moyenne de transport.
Arrêt	Arrêts où les usagers peuvent monter et descendre. Définit également les stations et leurs entrées.
Temps d'arrêt	Heures d'arrivée et de départ d'un bus depuis des arrêts spécifiques, pour chaque trajet.
Utilisateur	représente les utilisateurs des bus, tel que un passager peut prend un bus, et qui peut attendre des bus dans les différents arrêts.
Capteur	Représente les moyennes de communication entre arrêt, bus et le centre opérationnel.
Administrateur	Gérer agence.

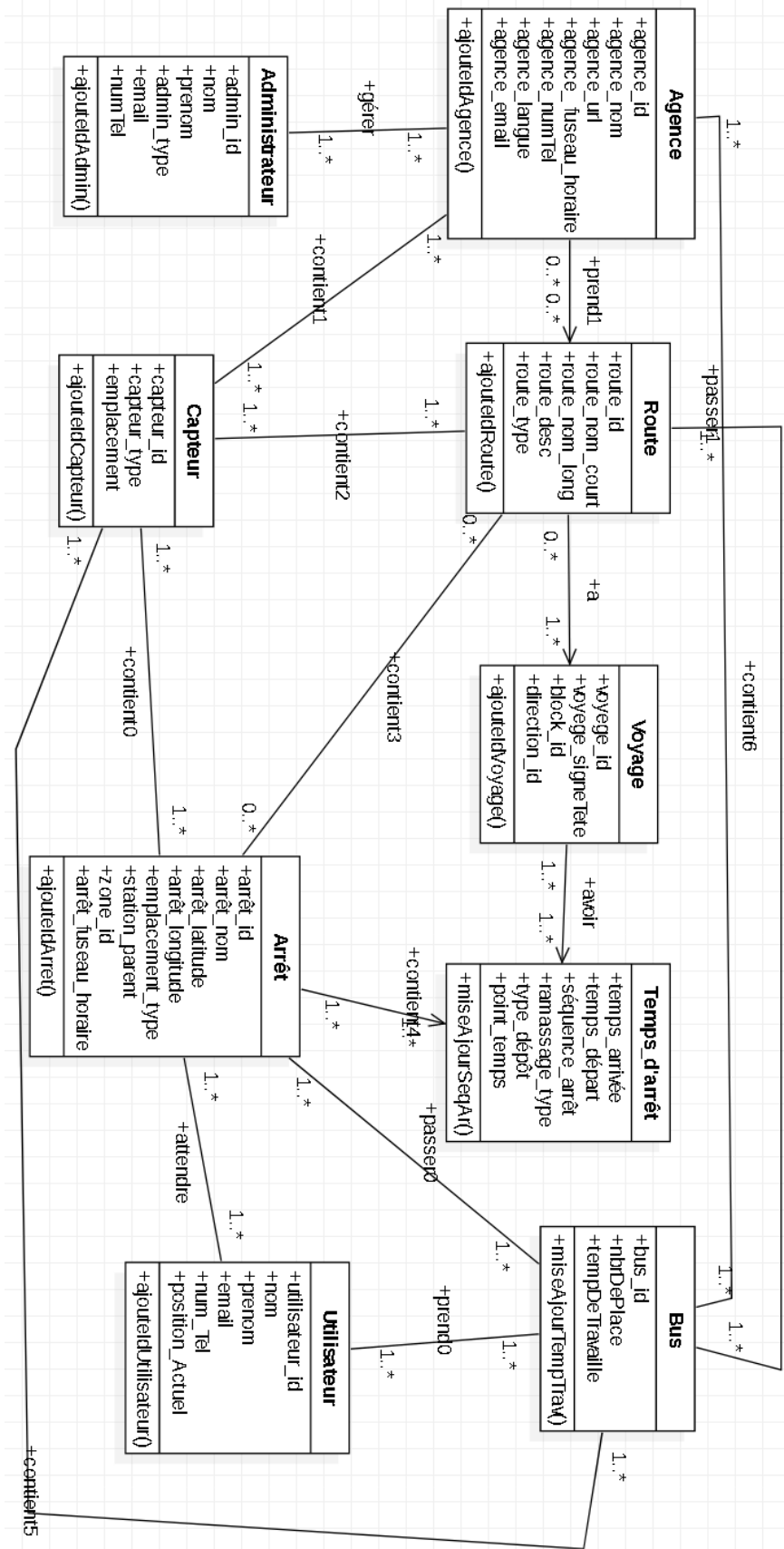


Figure 12: diagramme de class.

Définition des champs :

Class Agence :

Tableau 4: class agence

Nom de champ	type	Description
agence_id	ID	Souvent identique au nom de l'agence.
agence_nom	texte	Nom complet de l'agence de transports.
agence_url	url	URL de l'agence de transports.
agence_fuseau_horaire	Fuseau horaire	Fuseau horaire de la zone où se trouve l'agence de transports en commun. Si plusieurs agences sont spécifiées dans l'ensemble de données.
agence_numTel	Numéro de téléphone	Numéro de téléphone de l'agence spécifiée. Ce champ est une valeur de chaîne qui indique le numéro de téléphone habituel pour la zone desservie par l'agence.
agence_langue	Code de langue	Langue principale utilisée par cette agence de transports en commun.
agence_email	Adresse e-mail	Adresse e-mail consultée régulièrement par le service client de l'agence.

Class Route :

Tableau 5: class route

Nom de champ	Type	Description
route_id	ID	Définit un itinéraire.
route_nom_court	Texte	Version courte du nom d'un itinéraire. Il s'agit généralement d'un identifiant court, abstrait comme "32".
route_nom_long	Texte	Le nom complet d'un itinéraire. Ce nom est généralement plus descriptif que la version courte.
route_desc	Texte	Description d'un itinéraire. Les informations fournies doivent être utiles et de qualité.
route_type	Énumération	Décrit le moyen de transport utilisé pour un itinéraire. Par exemple (1 : train. Utilisé pour les trajets interurbains ou longue distance. 2 : bus. Utilisé pour les lignes de bus courte et longue distance.)

Class Voyage :

Tableau 6: class voyage

Nom de champ	Type	Description
voyage_id	ID	Définit un trajet.
voyege_signeTete	Texte	Texte qui apparaît sur la signalétique indiquant aux usagers la destination du trajet. Ce champ permet de différencier les modèles de service sur un même itinéraire.
block_id	ID	Identifie le bloc auquel appartient le trajet. Un bloc comprend un seul trajet ou de nombreux trajets séquentiels effectués par le même bus.
direction_id	Énumération	Indique la direction du trajet. Ce champ n'est pas utilisé pour les itinéraires, mais il permet de distinguer les trajets en fonction de leur direction lors de la publication des horaires.

Class Arrêt :

Tableau 7: class arrêt

Nom de champ	Type	Description
arrêt_id	ID	Identifie un arrêt, une station ou une entrée de station.
arrêt_nom	Texte	Nom de l'emplacement. Utilisez un nom compréhensible à la fois pour les habitants de la région et pour les touristes.
arrêt_latitude	Latitude	Latitude de l'emplacement.
arrêt_longitude	Longitude	Longitude de l'emplacement.
emplacement_type	Énumération	Type d'emplacement : 0 ou vide : arrêt ou quai (lieu où les usagers montent dans une bus de transport en commun ou en descendant).
station_parent	ID qui fait référence	Spécifie la hiérarchie entre les différents emplacements.
zone_id	ID	Définit la zone tarifaire d'un arrêt.
arrêt_fuseau_horaire	Fuseau horaire	Fuseau horaire de l'emplacement.

Class Temps_d'arrêt :

Tableau 8: class temps_d'arrêt

Nom de champ	Type	Description
temps_arrivée	Heure	Heure d'arrivée à un arrêt donné pour un trajet spécifique inclus dans un itinéraire.
temps_départ	Heure	Heure de départ depuis un arrêt donné pour un trajet spécifique inclus dans un itinéraire.
séquence_arrêt	Entier non négatif	Ordre des arrêts desservis lors d'un trajet particulier. Les valeurs doivent augmenter à mesure du trajet.
ramassage_type	Énumération	Indique les possibilités de montée à bord.
type_dépôt	Énumération	Indique les possibilités de descente du bus.
point_temps	Énumération	Indique si les heures d'arrivée et de départ spécifiées pour un arrêt sont strictement respectées par le bus, ou s'il s'agit plutôt d'heures approximatives et/ou interpolées.

Class Bus :

Tableau 9: class bus

Nom de champ	Type	Description
bus_id	ID	Identifiant de bus.
nbrDePlace	Entier non négatif	Nombre de place dans le bus.
tempDeTravail	Heur	Temps de travail par journées.

Class Utilisateur :

Tableau 10: class utilisateur

Nom de champ	Type	Description
utilisateur_id	ID	Identifiant de utilisateur.
Nom	Texte	Nom de utilisateur.
prénom	Texte	Prénom de utilisateur.
Email	Email	Emaile de utilisateur.
Num_Tel	Numéro de téléphone	Numéro de téléphone de utilisateur.
Position_Actuel	Positions	Emplacement de bus actuel.

Class Capteur :

Tableau 11: class capteur

Nom de champ	Type	Description
Capteur_id	ID	Identifiant de capture.
Capteur_type	Texte	Type de capture.
emplacement	Position	La positionnement de capture.

Class Administrateur :

Tableau 12: class administrateur

Nom de champ	Type	Description
admir_id	ID	Identifiant de Administrateur.
Nom	Texte	Nom de Administrateur.
prénom	Texte	Prénom de Administrateur.
Admin_type	Texte	type de Administrateur.
Email	Email	Email de Administrateur.
numTel	Numéro de téléphone	Numéro de Téléphone de Administrateur.

3.5.2. Diagramme de cas d'utilisation :

3.5.2.1 Diagramme de Cas d'utilisation d'utilisateur, administrateur et capteurs :

Nous structurons dans cette section, le rôle de chaque acteur de système dans un diagramme de cas d'utilisation générale (voir la Figure 13) :

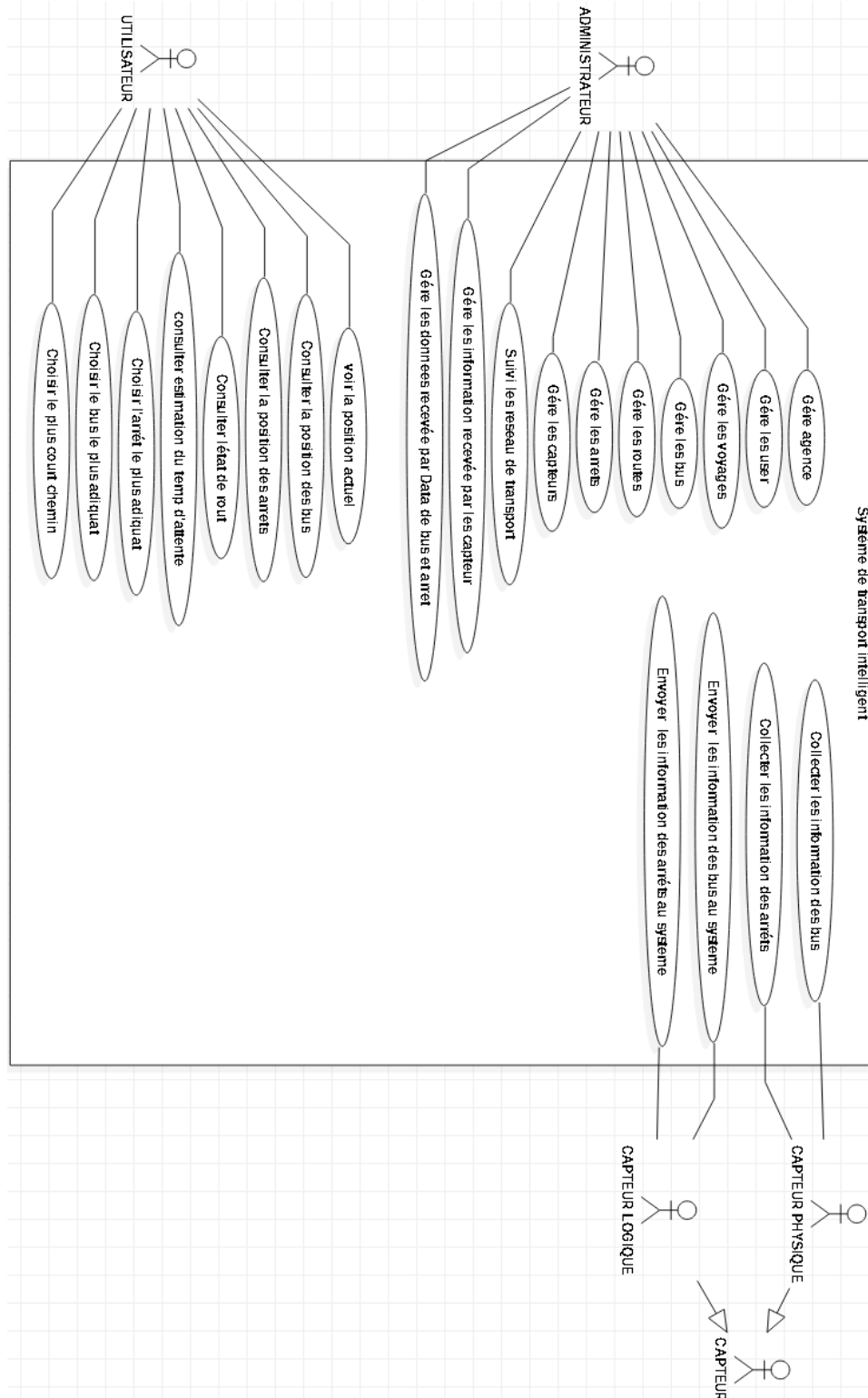


Figure 13: Cas d'utilisation générale.

3.5.2.2. Diagramme de Cas d'utilisation d'authentification :

Nous structurons dans cette section, les fonctionnalités qui permet au gérant de la société de gérer les profils des administrateurs et d'affecter les droits d'accès au système dans un diagramme de cas d'utilisation (voir la Figure 14) :

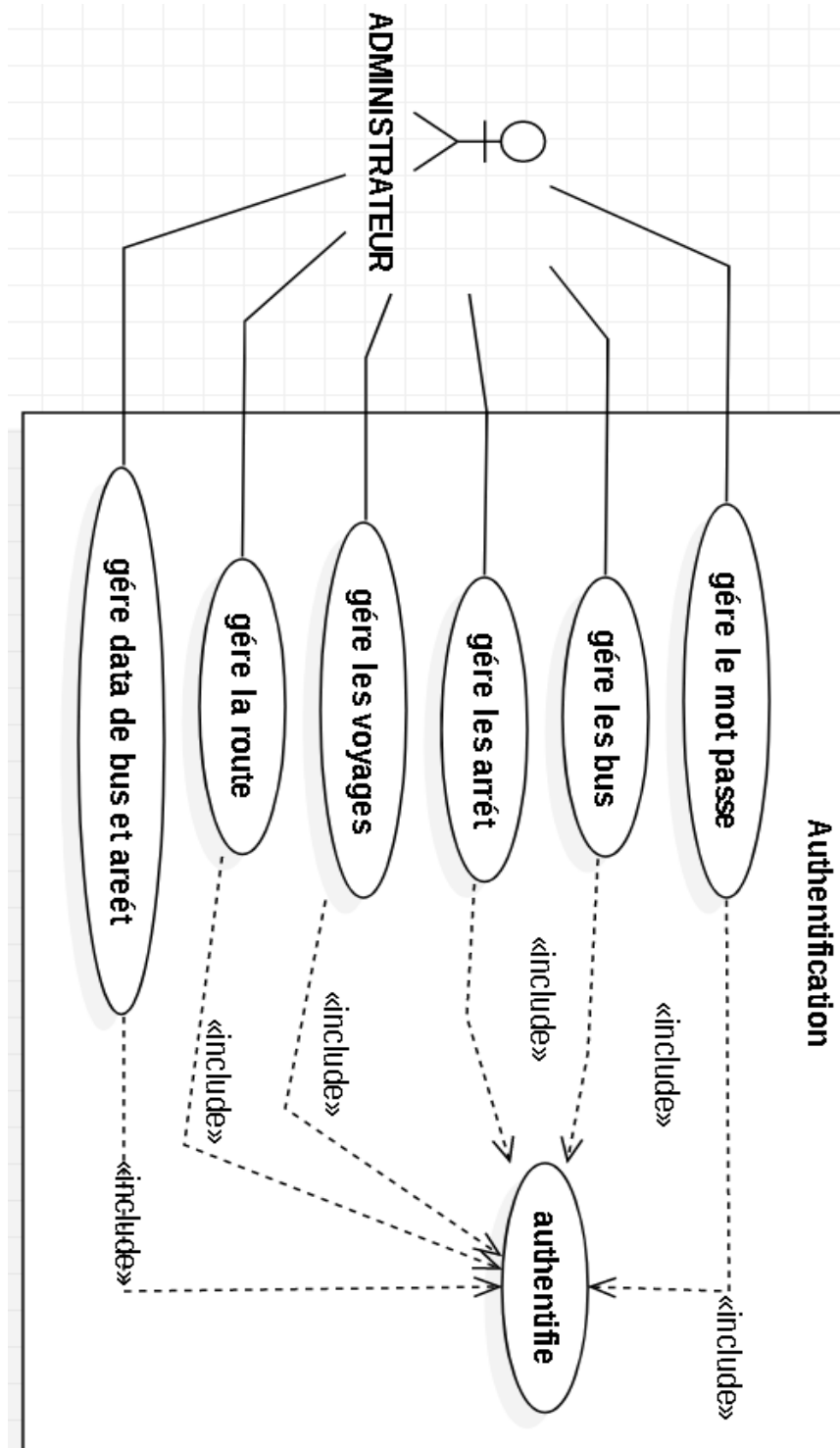


Figure 14: Cas d'utilisation d'authentification

3.5.2.3. Diagramme de Cas d'utilisation capteur bus :

Nous structurons dans cette section les fonctionnalités principales des capteurs existant dans les bus, et plus précisément les informations retenues par les capteurs (voir la Figure15) :

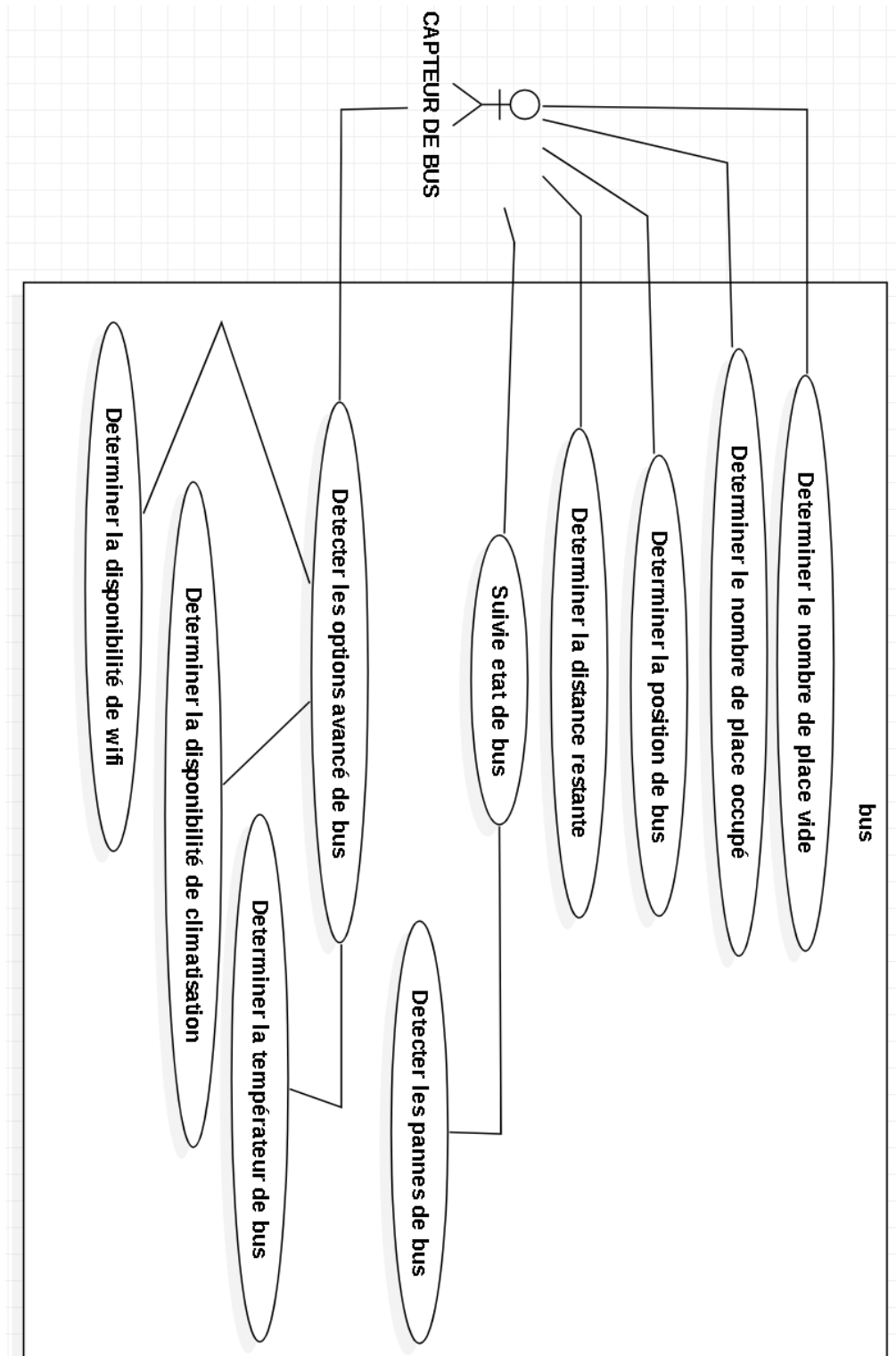


Figure 15: Cas d'utilisation de capteur de bus

3.5.2.4. Diagramme de Cas d'utilisation capteur arrêt :

Nous structurons dans cette section les fonctionnalités principales des capteurs existant dans les arrêts, et plus précisément les informations retenues par les capteurs (voir la Figure16) :

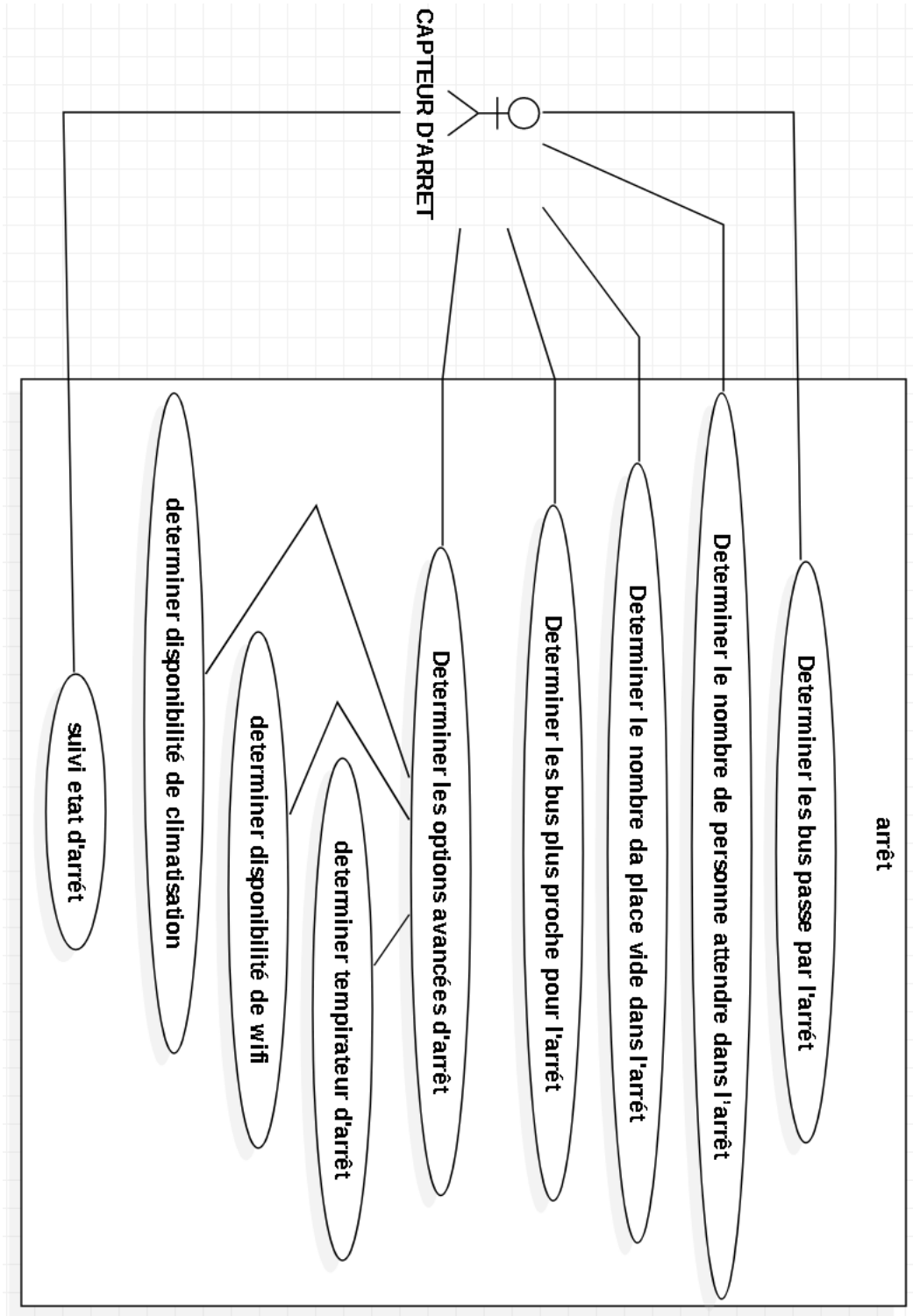


Figure 16: Cas d'utilisation de capteur d'arrêt

3.5.2.5. Diagramme de Cas d'utilisation Capteurs -Serveur :

Nous structurons dans cette section diagramme de cas d'utilisation Capteurs -Serveur (voir la Figure 17) :

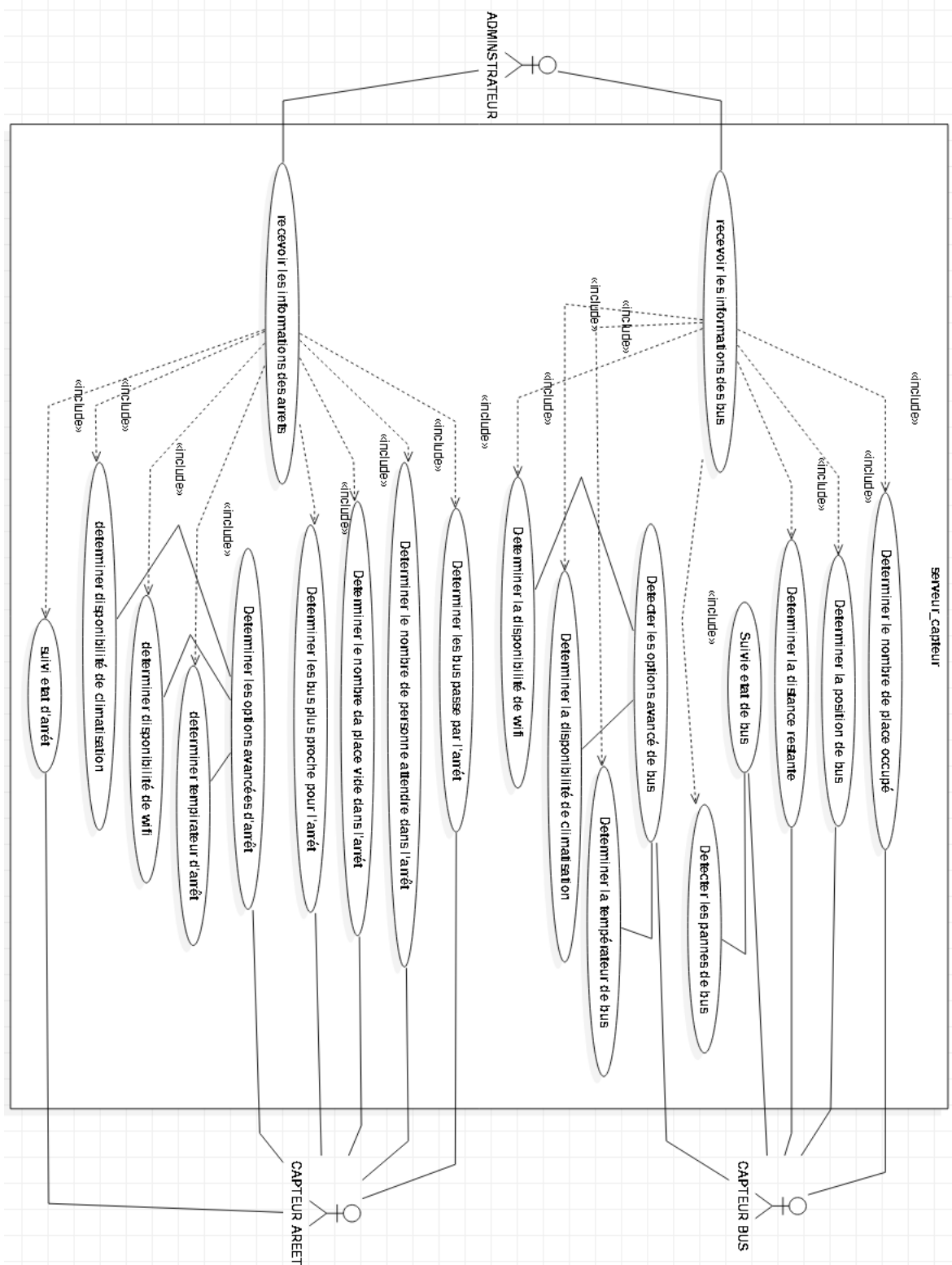


Figure 17: Cas d'utilisation Capteurs -Serveur

3.5.2.6. Diagramme de cas d'utilisation Utilisateur-Serveur :

Nous structurons dans cette section diagramme de cas d'utilisation Utilisateur-Serveur (voir la Figure 18) :

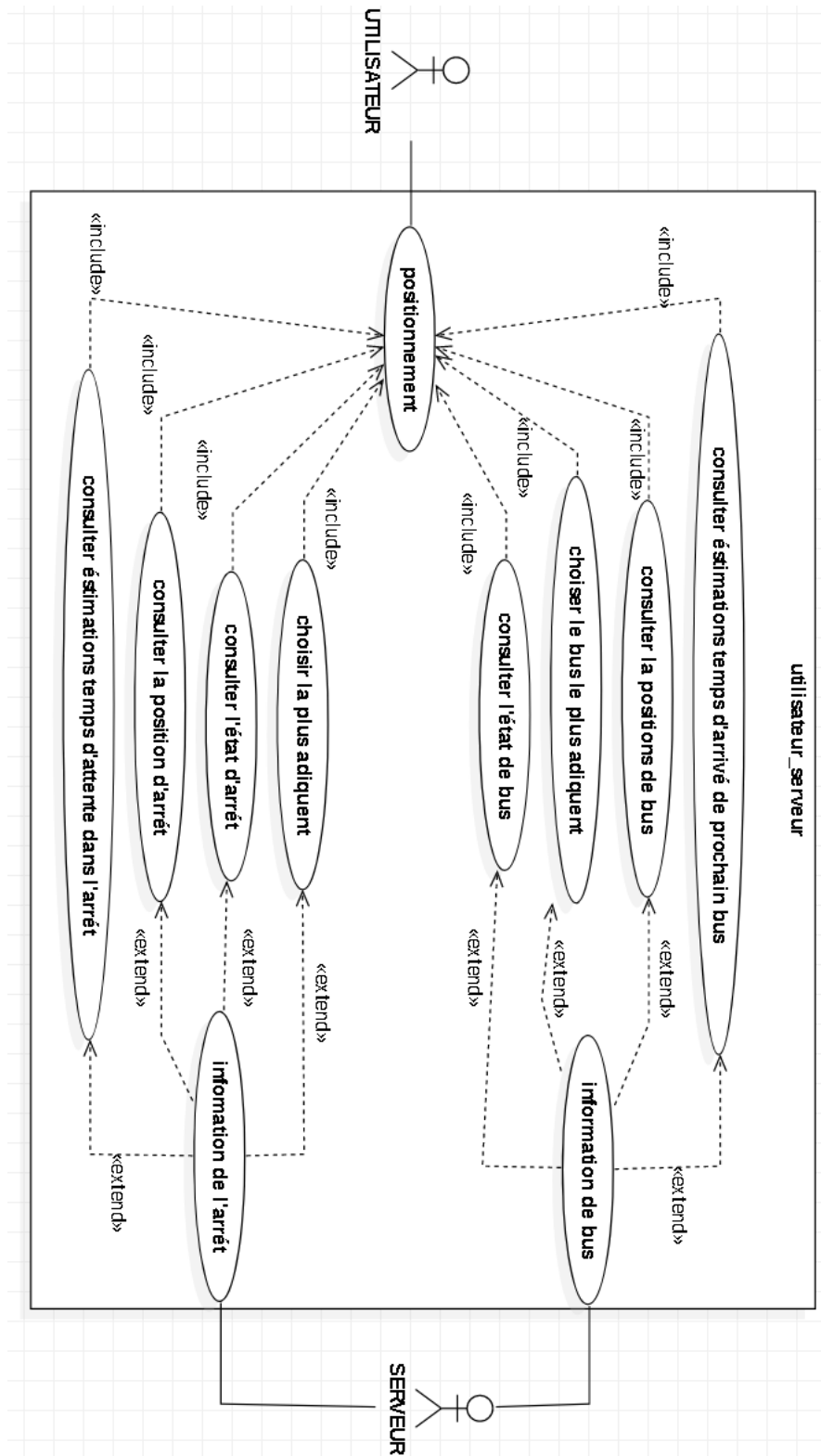


Figure 18: Cas d'utilisation Utilisateur-Serveur

3.5.3. Diagramme de séquence :

3.5.3.1. Diagramme de séquence d'authentification :

Nous structurons dans cette section, Diagramme de séquence d'authentification (voir la Figure 19) :

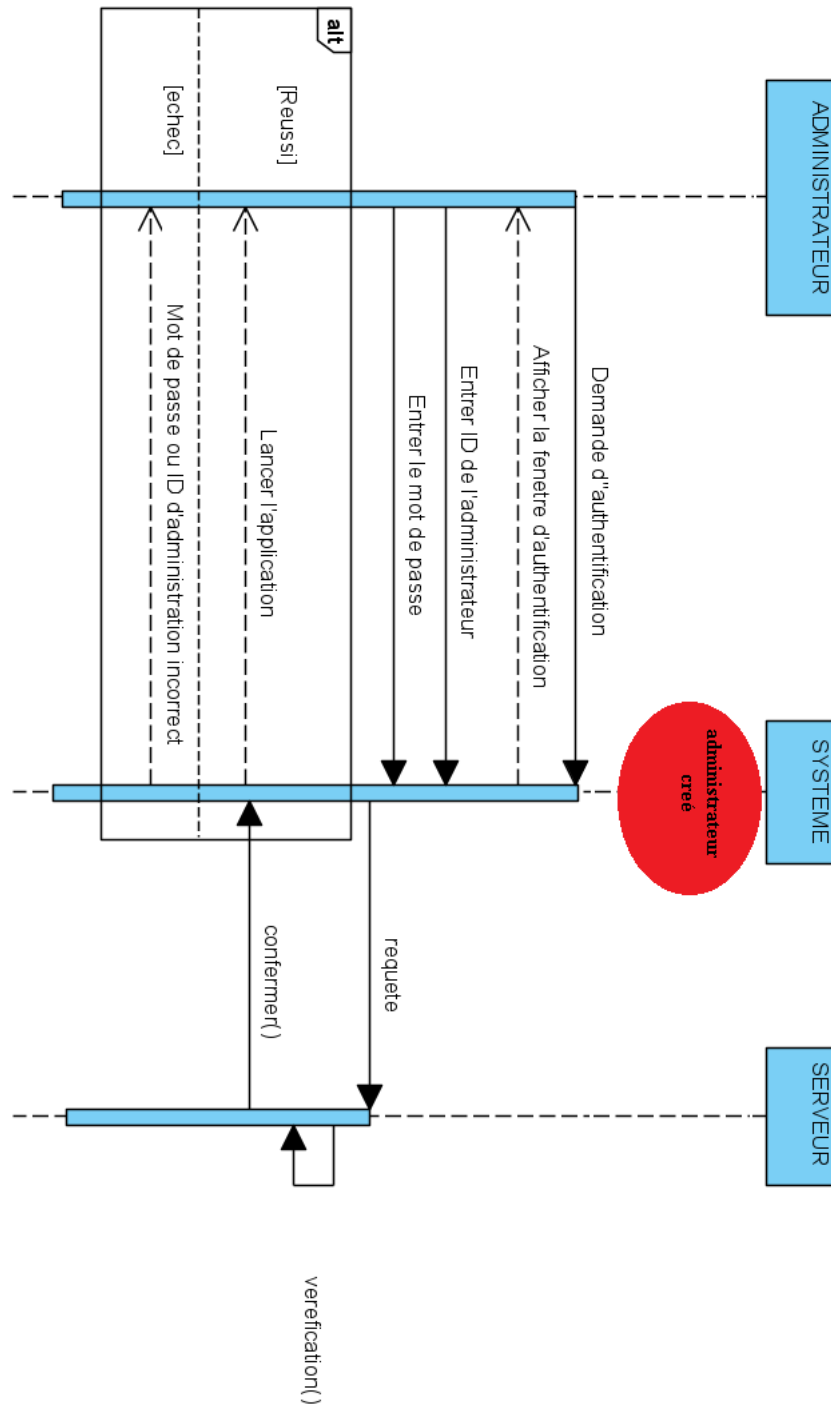


Figure 19: Diagramme de séquence d'authentification

3.5.3.2. Diagramme de séquence de création et activation des capteurs :

Nous structurons dans cette section, Diagramme de séquence de création et activation des capteurs (voir la Figure 20) :

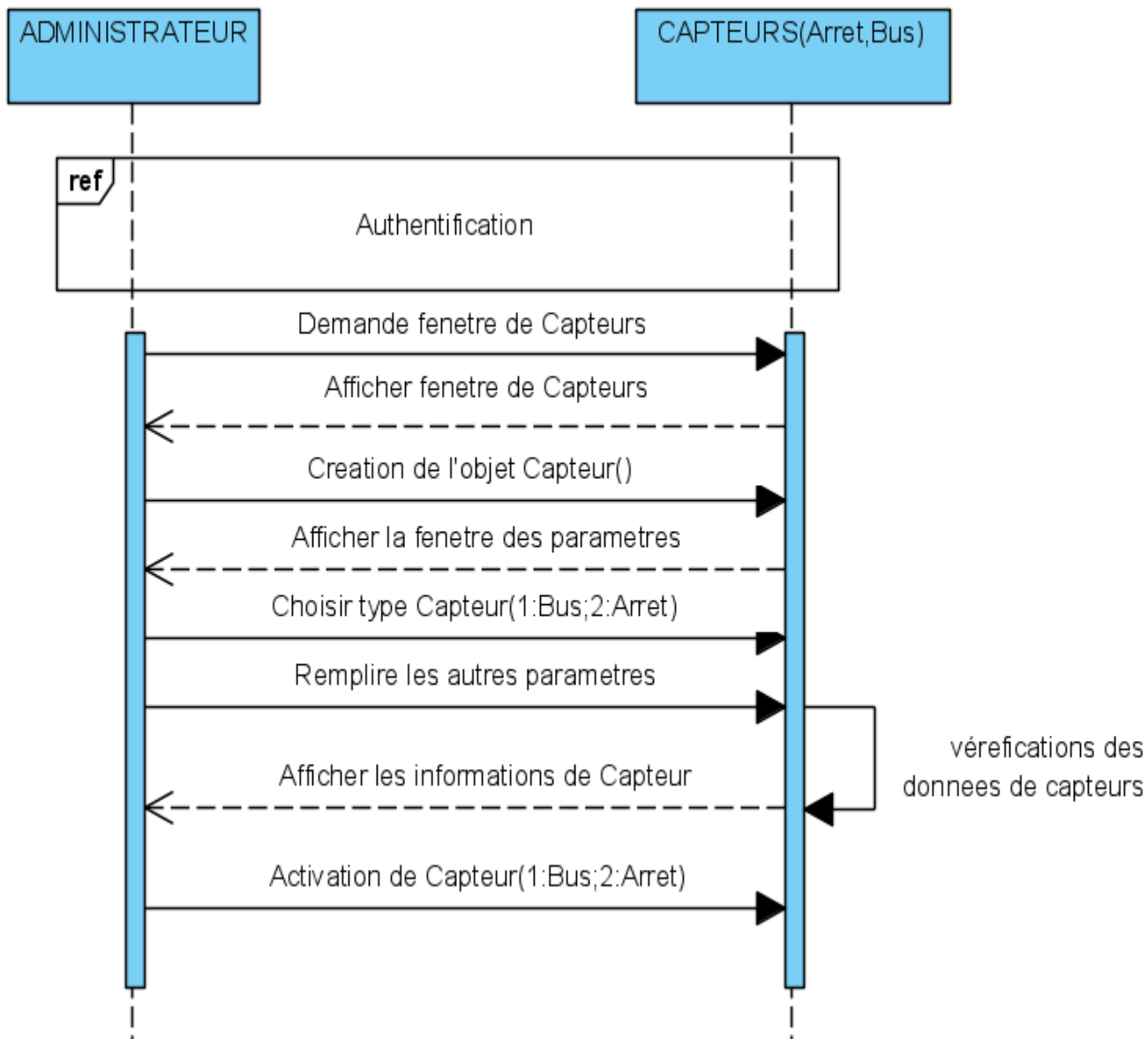


Figure 20: Diagramme de séquence de création et activation des capteurs

3.5.3.3. Diagramme de séquence d'utilisateur – application mobile :

Nous structurons dans cette section, Diagramme de séquence d'utilisateur – application mobile (voir la Figure 21) :

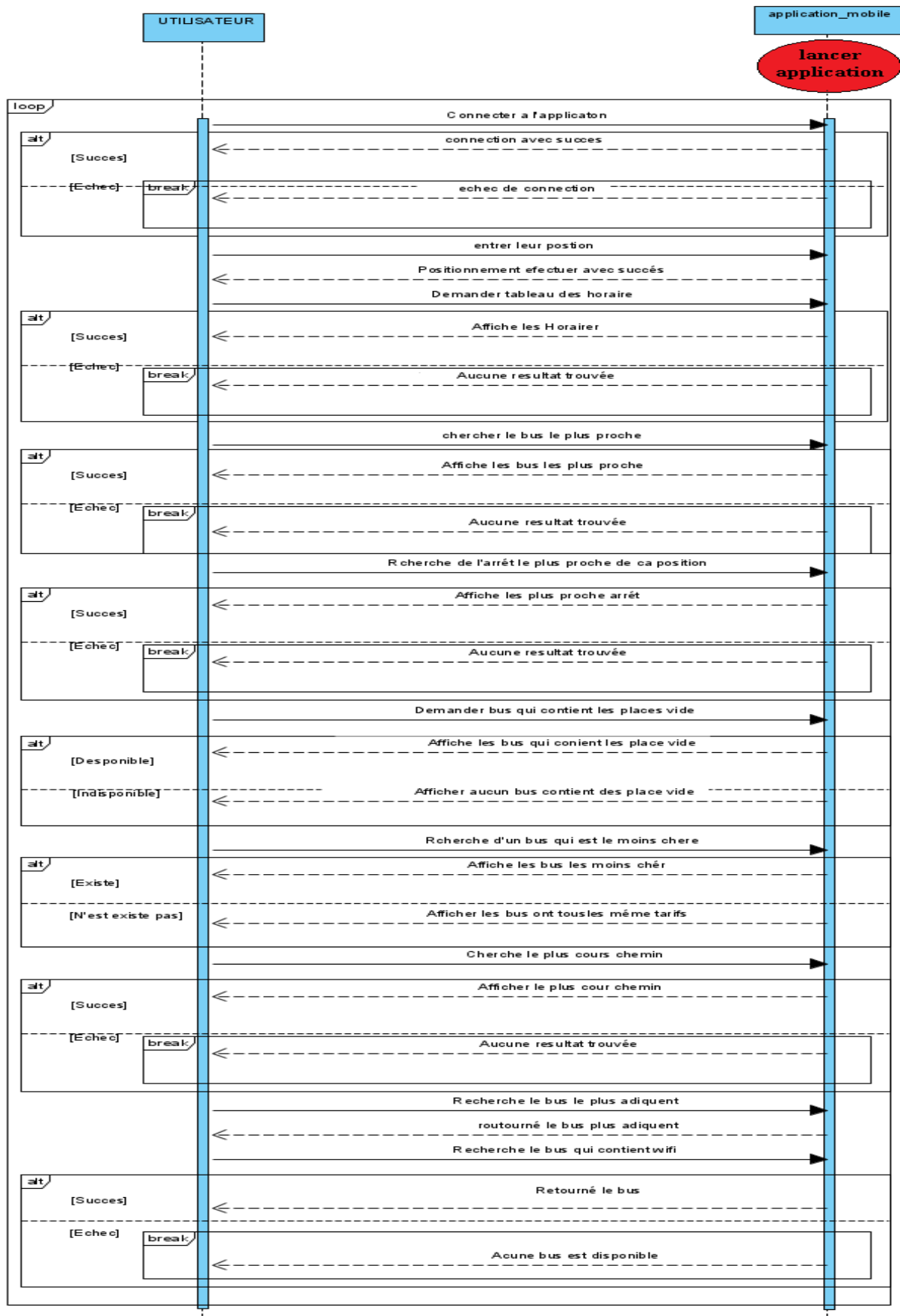


Figure 21: Diagramme de séquence d'utilisateur – application mobile

3.5.3.4. Diagramme de séquence capteur- serveur :

Nous structurons dans cette section, Diagramme de séquence capteur- serveur (voir la Figure 22) :

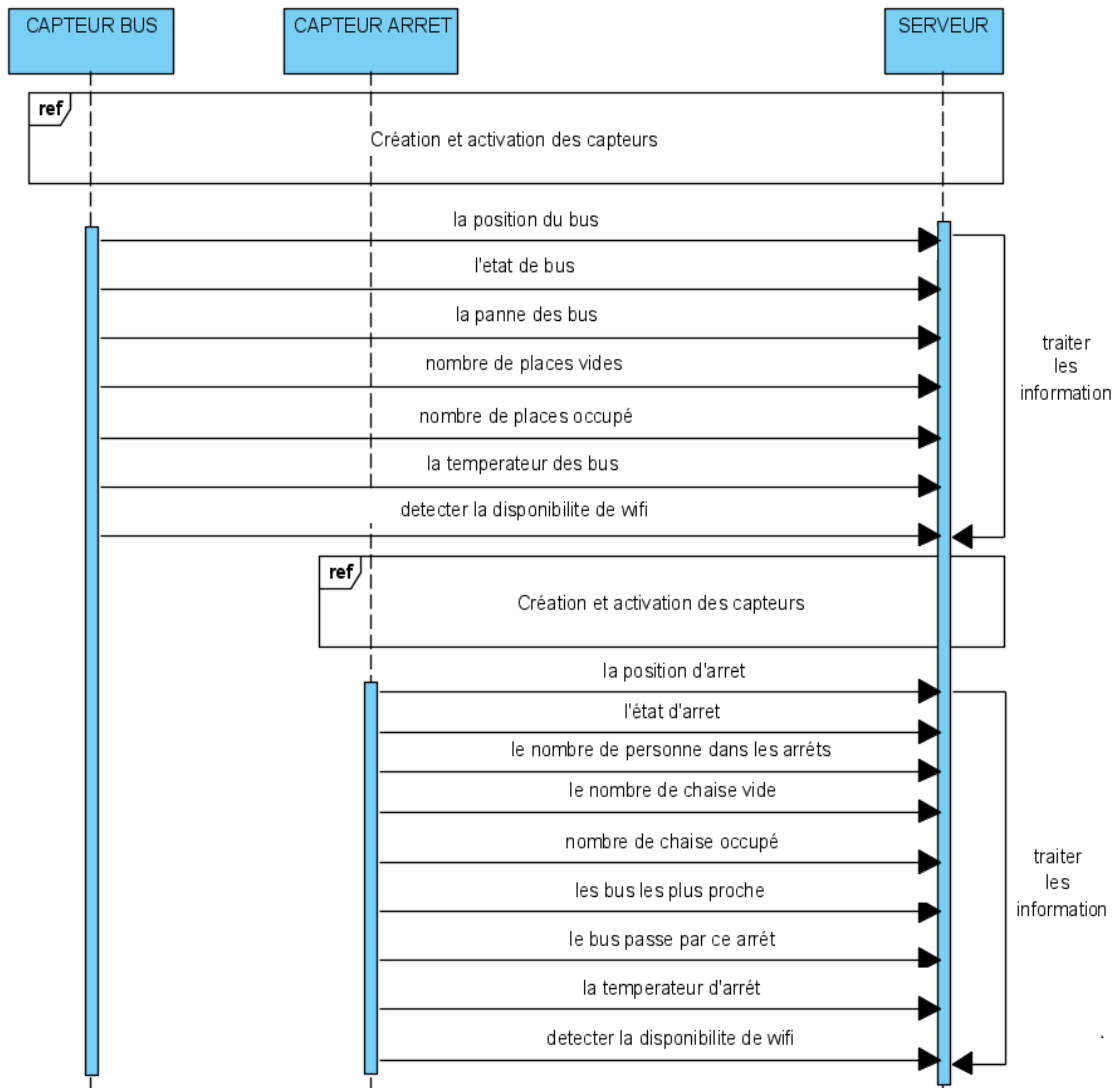


Figure 22: Diagramme de séquence capteur- serveur

Conclusion :

Ce chapitre a donné une vision sur notre travail, et a donné l’aspect conceptuel de notre projet à travers les différents diagrammes décrits en UML, suivi le processus UP, diagramme qui représente les différentes classes métiers de notre système, ensuite des diagrammes qui illustrent les différentes cas d’utilisations, et enfin des diagrammes de séquence qui illustrent le processus de visualisation des cas d’utilisation.



CHAPITRE 04



IMPLEMENTATION ET BILAN

Introduction :

Dans ce chapitre en va présenter d'abord les outils pour la réalisation de notre projet, et en va tracer quelques étapes de réalisation, après nous passons à la présentation de l'application mobile, le backend de système et map spécialisé avec des capture écran ainsi les composants et les fonctionnalités de chacune d'elle.

4.1.Choix des outils et langages utilisés :

Google Cloud Platform :

Google Cloud Platform (GCP) est la plateforme Cloud de Google pour les développeurs et les gestionnaires d'infrastructure qui regroupe son PaaS et son IaaS ainsi qu'un ensemble d'API et de services connexes comme les bases de données hébergés, des outils Big Data, de Machine Learning ou d'optimisation du réseau. [17]

Google Maps Platform :

La plate-forme Google Maps est un ensemble d'API et de SDK qui permet aux développeurs d'intégrer Google Maps dans des applications mobiles et des pages Web, ou de récupérer des données de Google Maps. Il existe plusieurs offres. Selon vos besoins, vous pouvez vous retrouver à utiliser une ou une combinaison des API et SDK. [18]

Firebase Platform :

Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel.[19]

Android Studio :

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle.[20]

Visual Paradigme :

Visual Paradigme est un outil UML prenant en charge UML 2.En plus de la prise en charge de la modélisation, il fournit des capacités de génération de rapports et d'ingénierie de code, y compris la génération de code . Il peut faire de l'ingénierie inverse des diagrammes à partir du code et fournir une ingénierie aller-retour pour divers langages de programmation .[21]

JavaScript :

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les technologies HTML et CSS. [22]

Node.js :

Node.js est une plateforme logicielle libre en JavaScript orientée vers les applications réseau événementielles hautement concurrentes qui doivent pouvoir monter en charge. [23]

General Transit Feed Specification :

GTFS (General Transit Feed Specification), également connu sous le nom de *GTFS statique* ou de *flux statique* pour le différencier de l'extension GTFS-realtime, définit un format de fichier commun pour les horaires de transports en commun et les informations géographiques associées. Les "flux" GTFS permettent aux agences de publier leurs informations de transports en commun et aux développeurs de créer des applications qui utilisent ces données de manière interopérable. [24]

JavaScript Object Notation :

JSON (JavaScript Object Notation) est un format d'échange de données léger. Il est facile pour les humains de lire et d'écrire. Il est facile pour les machines d'analyser et de générer. Il est basé sur un sous-ensemble de la norme du langage de programmation JavaScript. [25]

Extensible Markup Language :

Le **XML**, pour Extensible Markup Language, désigne un langage informatique (ou métalangage pour être plus précis) utilisé, entre autres, dans la conception des sites Web et pour faciliter les échanges d'informations sur Internet. [26]

HyperText Markup Language :

Le HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. [27]

Cascading Style Sheets :

Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML. [28]

Cette solution implémente le suivi des actifs. Pour une utilisation dans un environnement de production, vous avez donc besoin d'une licence Google Maps Platform Premium, licence Google cloud Platform, licence Firebase Consol et une licence GTFS.

Tableau 13:implémentation des composants

Magasin de données	Une base de données Firebase Realtime qui stocke les emplacements des véhicules, accrochés à la route avec l' API Roads . Firebase assure la synchronisation des données en temps réel avec le backend et la carte.
Localisateur de véhicule	Une application Android qui utilise le fournisseur de localisation fusionné, dans les API de localisation des services Google Play , pour signaler son emplacement à la base de données Firebase Realtime.
Backend	Le backend intégré à Node.js, qui traite les emplacements de la base de données Firebase Realtime et prédit les temps de trajet à l'aide de l' API Directions .
Carte	Une application Web qui utilise l' API JavaScript Maps pour afficher une carte stylisée indiquant les emplacements et les itinéraires des bus
Aperçu de l'administrateur	Une interface Web pour les administrateurs, donnant un aperçu des actifs suivis. Il affiche une carte à l'aide de l' API Maps Static , avec les données de véhicule et de localisation de la base de données Firebase Realtime.

4.2.Flux de données :

Vous trouverez ci-dessous un diagramme de flux de données conceptuel (voir la figure 23) pour le suivi du transport, en se concentrant sur les modules principaux :

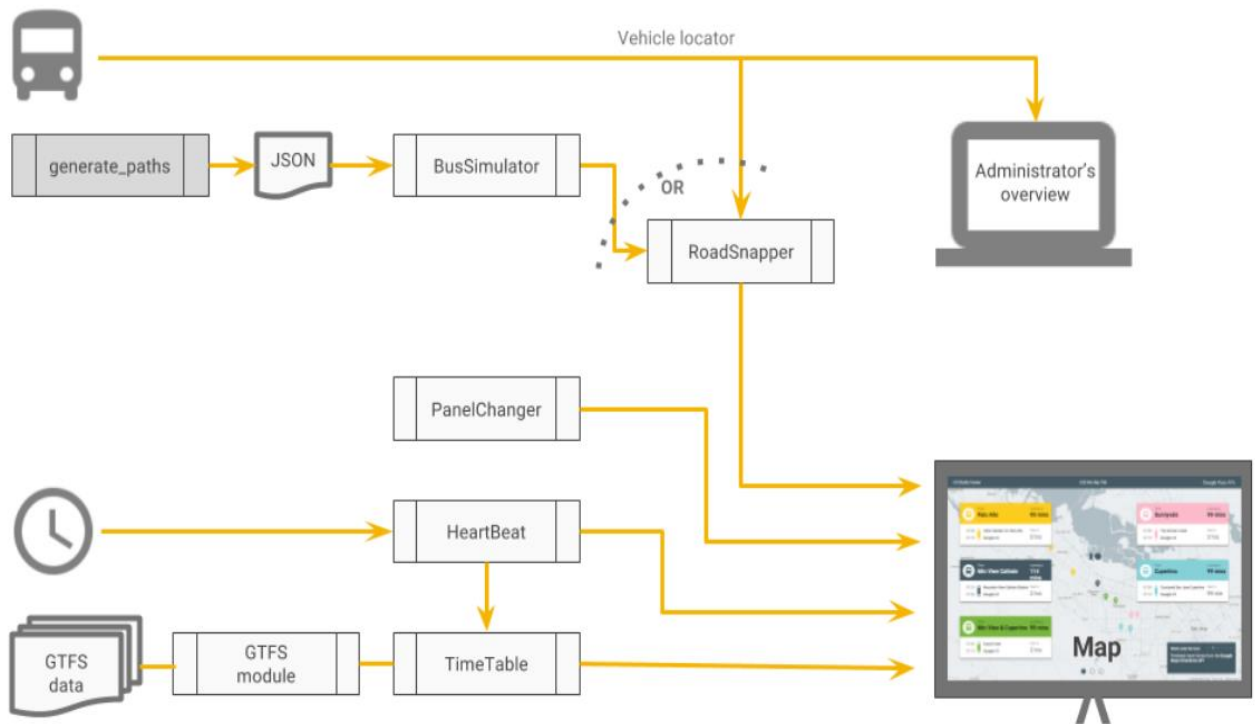


Figure 23: Diagramme de flux de données conceptuel

Modules :

Tableau 14:les modules

bus_simulator.js	Définit la BusSimulator classe, qui simule les positions des bus. Il utilise le signal horaire de HeartBeat pour rechercher les emplacements de bus dans le paths.json magasin généré .
gtfs.js	Définit la GTFS classe, qui charge les données GTFS (General Transit Feed Specification) dans une base de données SQLite en mémoire, et fournit des fonctionnalités pour interroger le calendrier GTFS.
heart_beat.js	Définit la HeartBeat classe qui publie les mises à jour de l'heure, réelles ou simulées, dans la base de données Firebase Realtime.
main.js	la logique principale de l'application Node.js
panel_changer.js	Définit la PanelChanger classe qui publie la partie pertinente du panels_config.json fichier. Le frontal utilise ce fichier pour déterminer la zone géographique à afficher, ainsi que les marqueurs d'hôtel appropriés pour chaque panneau.
road_snapper.js	Définit la RoadSnapperclasse, qui utilise l' API Roads pour capturer les signaux GPS sur les routes réelles.
time_table.js	Définit la TimeTable classe, contenant la logique métier principale du Transport Tracker. En résumé : <ul style="list-style-type: none"> • Interrogez la base de données GTFS en réponse aux notifications de changement d'heure de l'HeartBeat instance, via l'heure publiée dans la base de données Firebase Realtime. • Collectez les temps de trajet prévus à partir de l' API Directions et mettez en cache les résultats. • Regroupez les données transformées en fonction de la configuration du panneau de la carte.

Autres modules :

Tableau 15:les autres modules

generate_paths.js	construit une prédiction de l'emplacement des bus à des moments donnés.
GTFS répertoire	Les informations horaires des véhicules à chenilles. Les exemples de données sont basés sur le calendrier du Google I / O Bus Tracker. Cette table de timetable est stockée dans un ensemble de fichiers CSV qui suivent GTFS, la spécification générale du flux de transit .
package.json	Les informations requises par npm pour télécharger correctement les dépendances de développement et d'exécution, et pour exécuter l'application résultante.
panels_config.json	La configuration de la disposition des panneaux dans la carte Transport Tracker. Définit les trois panneaux de l'affichage, chacun avec un groupe d'itinéraires.
paths.json	Emplacements et horaires prévus des véhicules. Le generate_paths.js script crée ce fichier. Remarque: n'ouvrez pas ce fichier dans l'éditeur, car il est très volumineux.
serviceAccountKey.json	Les informations d'identification de votre compte de service Firebase.
tracker_configuration.json	Diverses valeurs de configuration, y compris la clé d'API Google Maps et les valeurs de configuration pour Firebase Realtime Database.

4.3.Presentation de l'application :

Dans cette partie en va présenter notre application :

4.3.1. Coté web :

L'authentification d'administrateur :

Cette fenêtre (voir la figure 24) permet aux administrateurs d'accéder au système après d'une entre correcte d'un nom d'administrateur et un mot de passe.

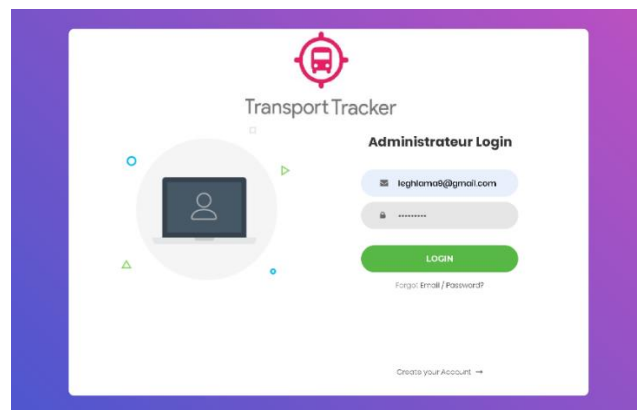


Figure 24: authentication

Tableau de bord d'administrateur :

Cette fenêtre (voir la figure 25) permet aux administrateurs de gère le système et voir une vision sur le système et aussi les statistiques et l'état de système.

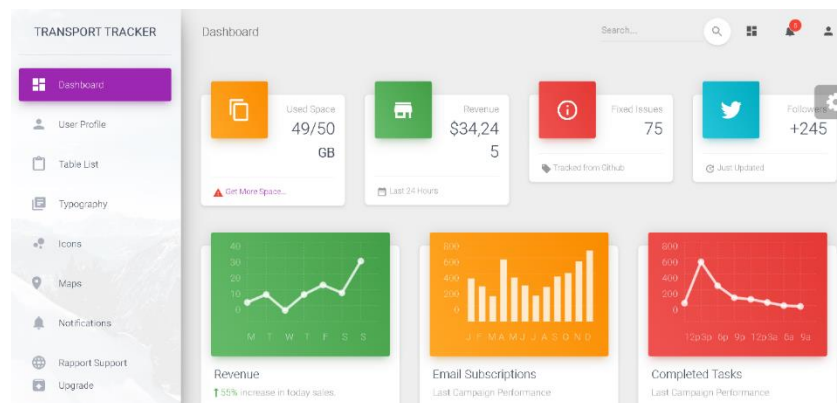


Figure 25: Tableau de bord d'administrateur

Méthode d'ajouter utilisateur :

Cette fenêtre (voir la figure 26) permet aux administrateurs de gère les méthodes de fournir accès aux utilisateurs sur firebase.

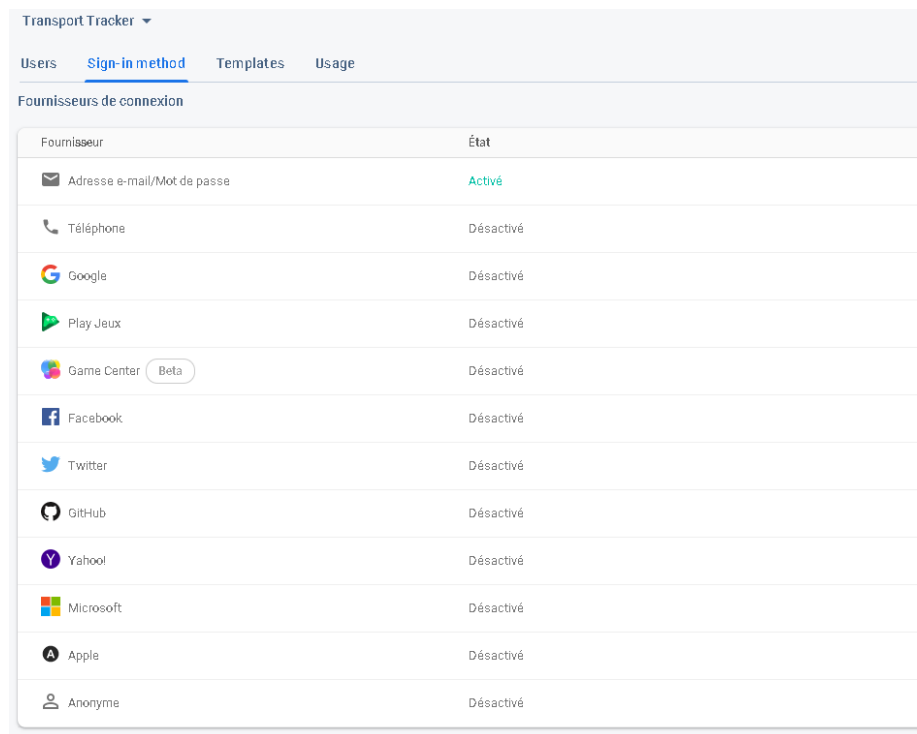


Figure 26: Méthode d'ajouter utilisateur

Ajouter utilisateur avec email :

Cette fenêtre (voir la figure 27) permet aux administrateurs de gère (ajouter) les utilisateurs sur firebase.

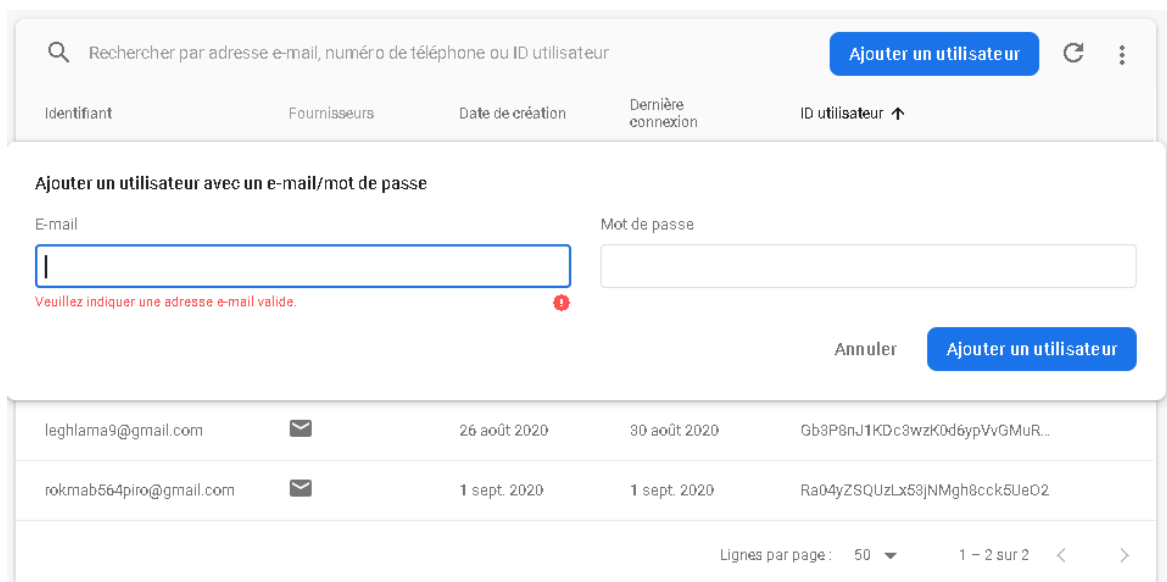


Figure 27: Ajouter utilisateur avec email

Test Application sur Cloud :

Cette fenêtre (voir la figure 28) permet aux administrateurs d'accéder à la Platform Google Cloud, activer Google Cloud Shell, en cliquant sur le bouton en haut à droite de la barre d'outils Cloud Shell, la fenêtre Cloud Shell s'ouvre en bas de la console Cloud. C'est ici que vous pouvez entrer des commandes pour interagir avec le Shell, en fin exécutez l'application avec la commande (**npm run main**)

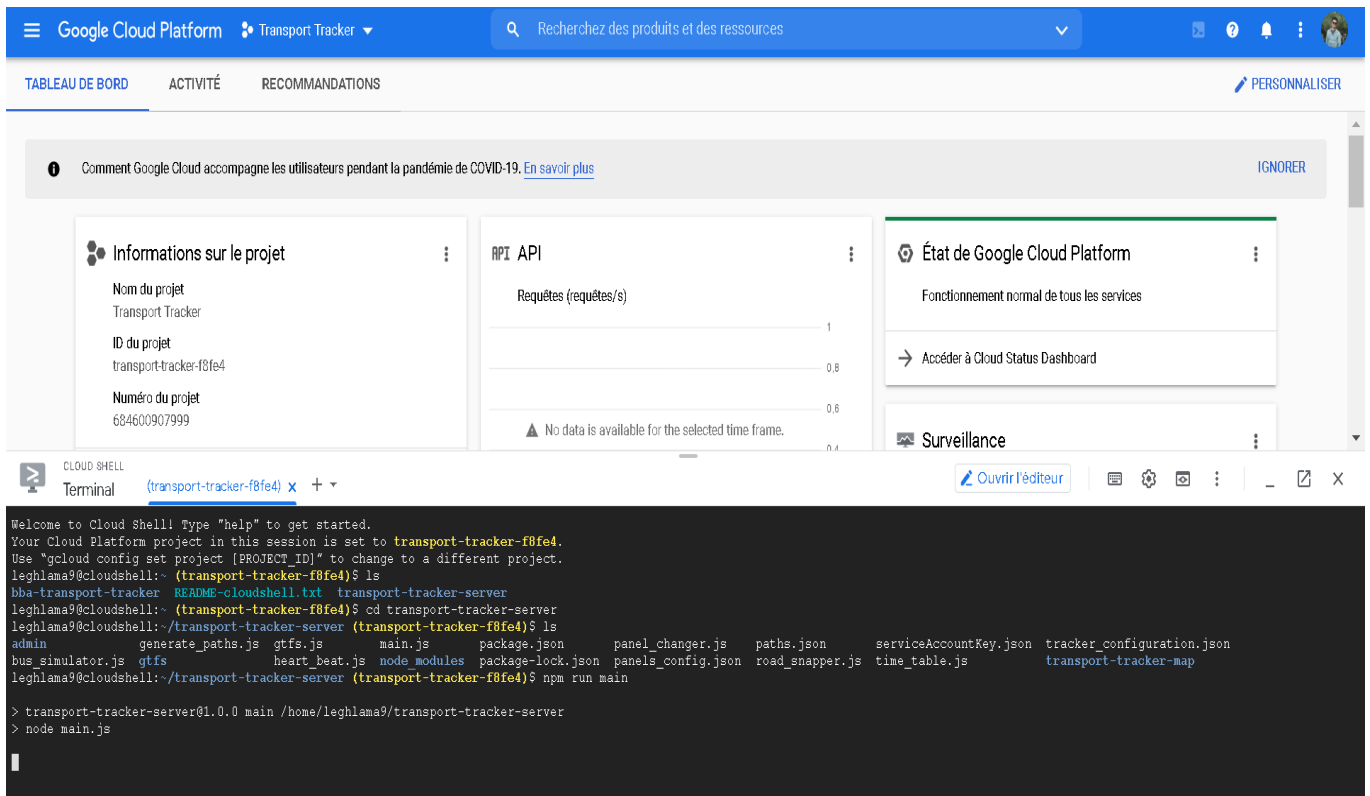


Figure 28: Test Application sur Cloud

Après utiliser le client Node.js pour les services Google Maps pour demander facilement des données aux services Web des API Google Maps, hébergez le contenu dynamique de notre application Web avec Firebase Hosting, appliquez un style personnalisé à une carte Google. En fin nous disposez de notre propre solution complète de suivi des actifs en temps réel (une carte de transport tracker (voir la figure 29)).

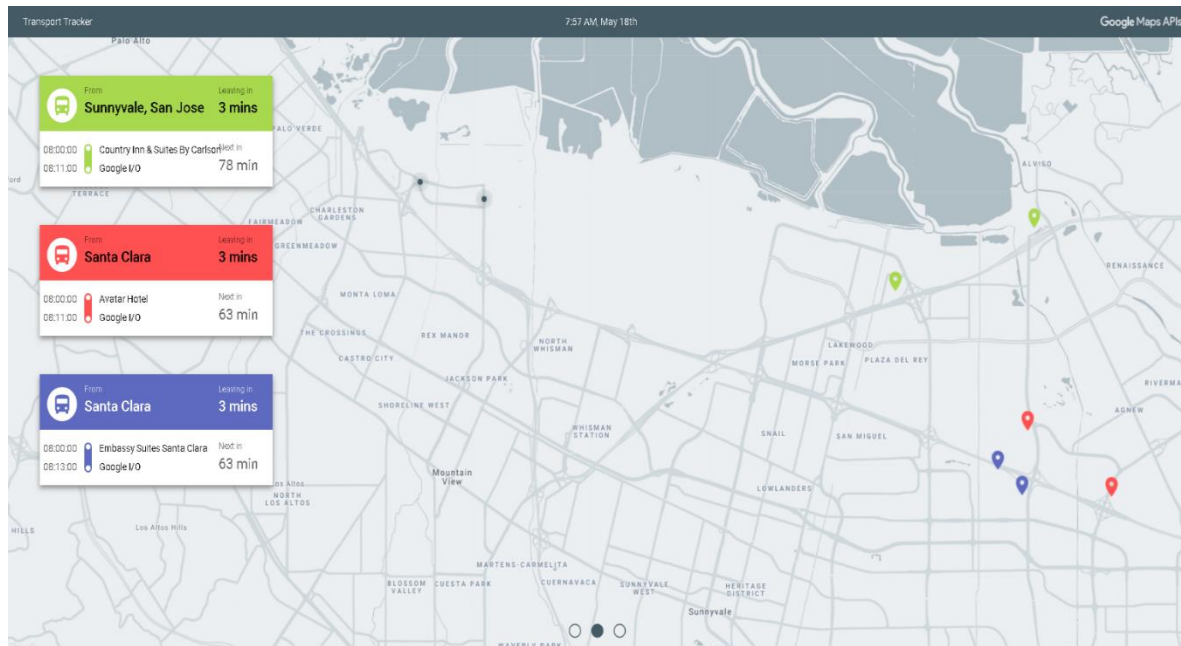


Figure 29: suivi des actifs en temps réel

Manipuler la base de donnée :

Cette fenêtre (voir la figure 30) permet aux administrateurs de gère la base de donnée sur Consol firebase.

Les références de base de données définissent les chemins dans votre base de données Firebase Realtime :

- **Bus-locations** : Les emplacements des bus sur la carte.
- **Map** : Les limites de la carte et l'ensemble particulier de cartes d'itinéraire qui apparaissent sur la carte à tout moment.
- **Panels** : Les données horaires (itinéraires et heures d'arrivée estimées) affichées sur la carte et les arrêts.
- **Current-time** : L'horodatage qui détermine le calendrier et les données associées affichés sur la carte.
- **Raw-bus-locations** : nouvelles itinéraires des bus à l'aide d'API Google map direction.
- **Raw -locations** : Les emplacements des utilisateurs.

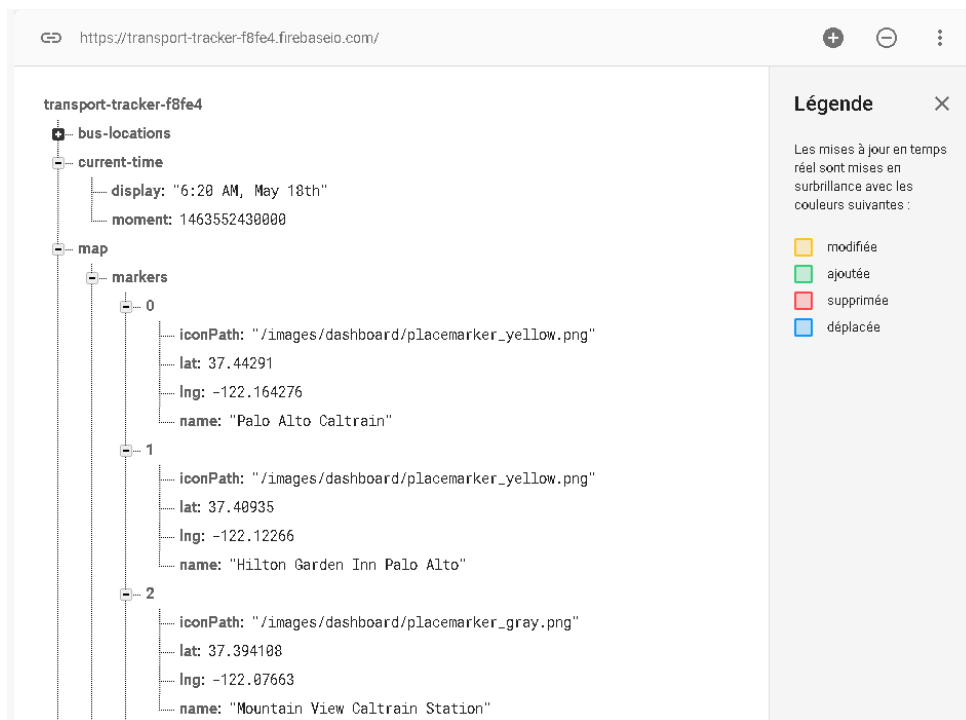


Figure 30: Manipuler la base de donnée

Hébergement d'application :

La première chose que nous allons faire est de mettre une carte sur une page Web, indiquant l'emplacement. Pour cela, nous utiliserons Firebase Hosting (voir la figure 31) pour héberger le contenu dynamique de cette application Web

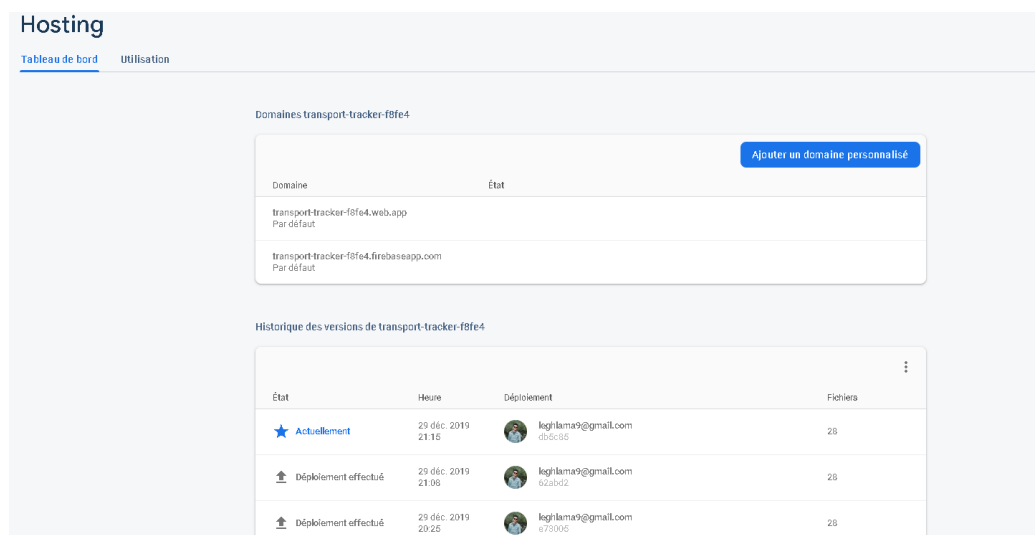


Figure 31: Hébergement d'application

4.3.2. Coté Mobile :

Transport tracker est une application Android (voir la figure 32) qui envoie des mises à jour de localisation à la base de données Firebase Realtime, voir les données horaires (itinéraires et heures d'arrivée estimées) affichées sur la carte et les arrêts et Les emplacements des bus sur la carte.

Après saisie id, email, mot de passe click sur butons Start tracking, application mobile vous donne votre emplacement en temps réel et donne la liste des bus qui vous demande aussi fournir un lien a l'application web pour voir tout.

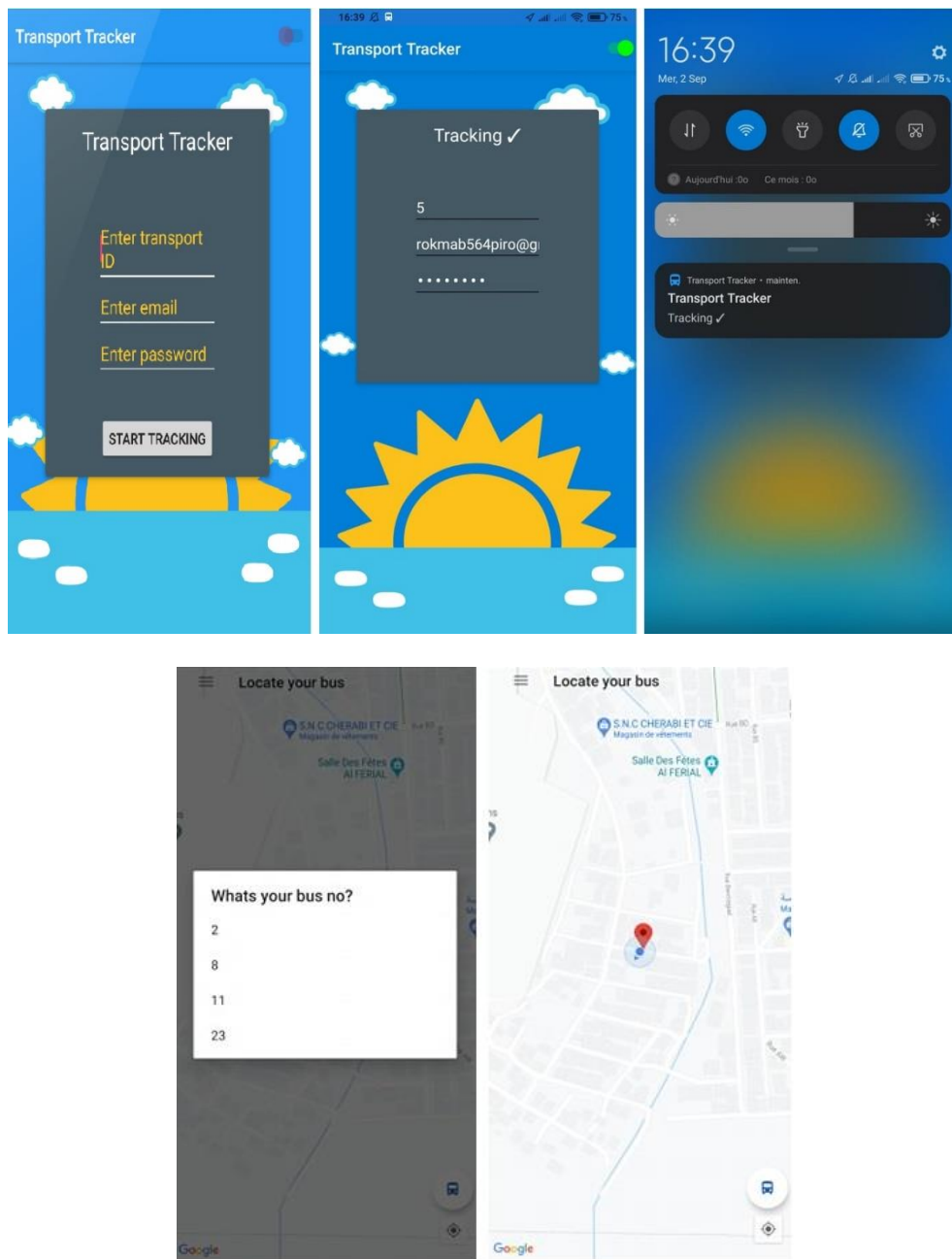


Figure 32: application mobile

Suivi d'utilisateur sur base de donnée :

A l'aide du champ raw-location (voir la figure 33) sur la base de donnée en peut suivi localisation d'utilisateur avec synchronisation et aussi énergie de batterie de mobile

(Power < 20 désactiver tous les services activer a par GPS pour économie la batterie).

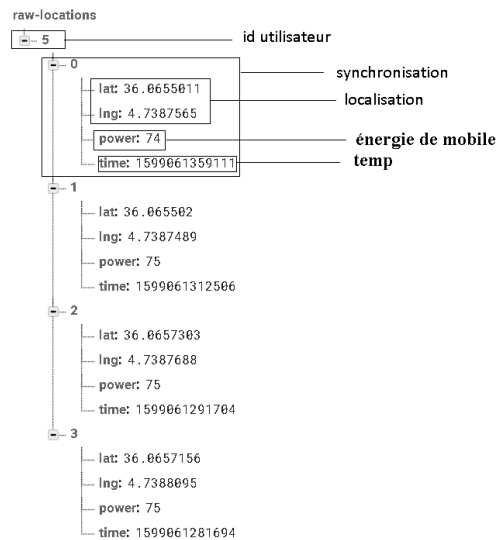
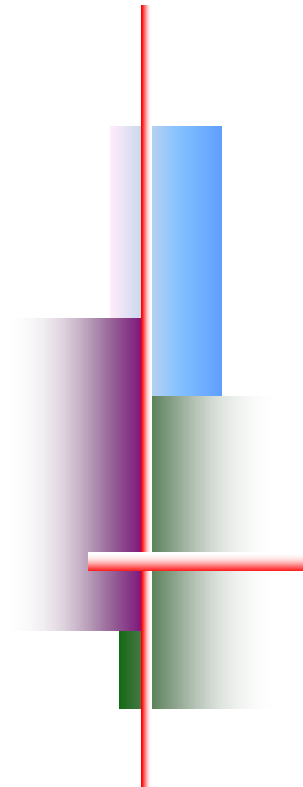


Figure 33: Suivi d'utilisateur sur base de donnée.

Conclusion :

Dans ce chapitre, on a présenté les outils et logiciel du projet. Par la suite, nous avons élaboré quelques aperçus du fruit de notre travail à travers des interfaces résultant.



Conclusion Générale

Conclusion Générale

Le travail développé dans ce mémoire porte sur la problématique de transport intelligent au niveau de la ville de Bordj Bou Arredj mais notre travail a été appliqué au niveau de la ville San Francisco à cause de Covid-19 et Les APIs de GTFS et autre APIs comme API Roads non disponible sur notre ville.

Ce mémoire propose une résolution au domaine de transport intelligent. Nous avons donc étudié dans un premier temps la notion de transport intelligent et différents domaines informatiques concernés et les différentes technologies de Système de Transport Intelligent, puis en a donné quelques travaux (Application) dans ce domaine.

Après nous avons suivi la méthode UP, donné une proposition d'un architecte générale sur le fonctionnement d'un système de transport intelligent, Avec une conception, et une modélisation à ce type de système utilisant l'application orienté objet (Unified Modeling Language).

Par ailleurs, nous avons inscrit sur la Plateforme de Google cloud et autres plateformes pour permettre de tester et appliquer nos connaissances acquises au sein de la faculté des Mathématiques et informatique. Ainsi qu'en a enrichi nos connaissances aux nouveaux concepts.

Après l'application des nouvelles technologies de programmation complexe nous pouvons dire que les résultats trouvés nous permettent de valider notre travail de fin d'étude qui peut être appliqué pour la restructuration de transport intelligent de n'importe quelle ville.



Bibliographie

Bibliographie

1. <http://www.etusa.dz/>
2. Livre de Systèmes de transport intelligents : Modélisation, information et contrôle (Français) Broché – 23 mai 2007 de Saïd Mamer
3. Systèmes de transport intelligents : Modélisation, information et contrôle (Français) Broché – 23 mai 2007 de Saïd Mamer
4. Article de Sayed Ali Ahmed El Mustafa sur Applications, défis et technologies futures de l'Internet des objets
5. Systèmes de transport intelligents : Modélisation, information et contrôle (Français) Broché – 23 mai 2007 de Saïd Mamer
6. Cour de doctorant M. NAILI.MAKHLOUF Système d'aide à la décision.
7. <https://defimedia.info/express-rs-45-m-de-pertes-pour-les-compagnies-de-transport>
8. Livre de Systèmes de Transport Intelligents (ITS) de Gérard Ségarra (Auteur) le 23 novembre 2018
9. Livre de Systèmes de Transport Intelligents (ITS) de Gérard Ségarra (Auteur) le 23 novembre 2018
10. Livre de Systèmes de Transport Intelligents (ITS) de Gérard Ségarra (Auteur) le 23 novembre 2018
11. Livre de Systèmes de Transport Intelligents de Dmbli Lazaro (Auteur)
12. <https://www.flowlabs.ai/>
13. <https://www.flowlabs.ai/>
14. <http://www1.ville.montreal.qc.ca/>
15. <https://citymapper.com/>
16. <https://moovitapp.com/>
17. <https://cloud.google.com/docs>
18. <https://cloud.google.com/maps-platform?hl=fr>
19. <https://firebase.google.com/docs/>
20. <https://developer.android.com/studio>
21. <https://www.visual-paradigm.com/>
22. <https://devdocs.io/javascript/>
23. <https://nodejs.org/en/docs/>
24. <https://developers.google.com/transit/gtfs?hl=fr>
25. <https://www.lemagit.fr/definition/JSON-JavaScript-Object-Notation>
26. https://fr.wikipedia.org/wiki/Extensible_Markup_Language
27. https://fr.wikipedia.org/wiki/Hypertext_Markup_Language
28. <https://devdocs.io/css/>

ملخص :

في هذه المذكرة، اقترحنا بنية جديدة لشبكة النقل الحضري لمدينة برج بوعرييج، والتي تلبي احتياجات المسافرين ، لتحقيق هذا الهدف أعدنا منهجية قابلة للتطبيق على جميع شبكات النقل الحضرية تعتمد على أساليب عديدة لحل المشاكل باستخدام برامج ولقد قمنا بتوثيق عملنا من خلال محاكاة حركة الحافلات التي تعمل على كل خط من الشبكة.

Abstract:

In this dissertation, we put forward this new structure of transport intelligent of the city of BBA, which meets the needs of passengers, to meet this goal we have a practical methodology to all urban transport networks based on different resolution methods using different software and we have confirmed our work by simulating the movement of buses running on each line of the network, this visualization was made under Google Cloud Platform.

Résumé :

Dans ce mémoire nous avons proposé une nouvelle structure de transport Intelligent de la ville de BBA, qui répond aux besoins des passagers, pour répondre à cet objectif nous avons disposé d'une méthodologie applicable à tous les réseaux de transport urbain basés sur différentes méthodes de résolution en utilisant des différents logiciels et nous avons validé notre travail par la simulation du déplacement des bus circulant sur chaque ligne du réseau, cette visualisation a été faite sous Google cloud Platform.