



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene



**Faculté d'Electronique et d'Informatique
Département Informatique**

Mémoire de Licence

Filière : Informatique

Spécialité : Ingénierie des Systèmes d'Information et des Logiciels

Thème

**Développement d'une application de commande à
distance pour un robot de surveillance**

Soutenu le :.../.../....

Sujet Proposé par :

Mme KAHLOUCHE Souhila (CDTA)

Mme BOUAGAR Saliha (USTHB)

Présenté par :

Mr BARACHE Syphax

Mr TIGHILT Massinissa

Devant le jury composé de:

M..... Président (e)

M..... Membre

Binôme N° : ISIL 026/21

Remerciements

Avant tout nous tenons à remercier Allah le plus puissant, qui nous a donné la force et la patience pour qu'on puisse accomplir ce modeste travail.

En second lieu, nous remercions spécialement Mme Kahlouche Souhila et Mme BOUAGAR Salha, nos encadreurs, pour l'orientation, la patience, l'aide, la confiance et l'assistance par leur disponibilité, qui ont constitué un apport considérable sans lequel notre travail n'aurait pas pu être mené à bon port.

Nous souhaitons aussi adresser notre profonde reconnaissance à tous les enseignants qui par leur compétence, la richesse et la qualité de leur enseignement nous ont soutenu dans la poursuite de nos études.

Nous remercions également les membres du jury d'avoir accepté d'évaluer ce modeste travail.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches, amis et famille, qui ont contribué de près ou de loin à l'achèvement de ce travail.

Merci à tous et à toutes.

Table des matières

Introduction générale	1
Chapitre 1 : Etat de l'art sur la robotique de surveillance	2
1.Introduction	2
1.1. La Robotique de surveillance.....	2
1.1.1. La Robotique.....	2
1.1.2. Les Systèmes de surveillance	2
1.2. Avantages.....	2
1.3. Inconvénients.....	2
1.4. Type robot de surveillance	2
1.4.1. Les robots de surveillance domestique.....	2
1.4.2. Le robot d'entrepôt.....	3
1.4.3. Spécialisé pour les datacenters	3
1.4.4. Le robot d'intervention	4
1.4.5. Les robots patrouilleurs	4
2. Travaux existant dans la robotique de surveillance	4
3. Conclusion	6
Chapitre 2 : Analyse et conception.....	5
1.Introduction	7
2.Identification des besoins	7
2.1. Spécification des besoins	7
2.1.1. Identification des acteurs :	7
2.1.2. Les besoins fonctionnels	7
2.1.3. Besoins non fonctionnels :	7
2.2. Messages échangés entre le système et les acteurs	7
2.2.1. Messages entrants vers le système :.....	7
2.2.2. Messages sortant du système	8
3.Diagramme cas d'utilisation	8
3.1. Description textuelle	8
3.1.1. Cas d'utilisation « authentification »	8
3.1.2. Cas d'utilisation « Gestion profil ».....	9
3.1.3. Cas d'utilisation « Gestion des comptes ».....	9
3.1.4. Cas d'utilisation « Manipulation robot ».....	10
4.Diagrammes d'activités	12
5.Diagramme de classe	14
5.1. Description du diagramme de classe.....	14
5.1.1. La classe Utilisateur.....	14

5.1.2. La classe robot.....	14
5.1.3. La classe Itinéraire	15
6.Développement du modèle dynamique	15
6.1. Diagramme d'état machine globale.....	15
6.2. Diagramme d'état machine authentification.....	16
6.3. Diagramme d'état machine de Gestion de comptes.....	16
6.4. Diagramme d'état machine de manipulation	17
7.Diagramme des composant du système	17
8.Diagramme de déploiement	18
9.Conclusion	19
Chapitre 3 : Réalisation et implémentation	1
1.Introduction	20
A propos du robot.....	20
2. Outils et langages utilisés	20
2.1. HTML	20
2.2. CSS	20
2.3. JavaScript.....	20
2.4. MySQL.....	20
3.Implémentation	21
3.1. Les serveurs.....	21
3.1.1. Serveur externe	21
3.1.2. Serveur embarqué	21
3.2. Transformation des données.....	22
4.Présentation de l'application	22
4.1. Interface authentification.....	22
4.2. Interface Accueil	23
4.3. Interface gestion des comptes	23
4.4. Interface gestion du profil.....	24
4.5. Interface manipulation robot.....	24
5. Test.....	26
6. Conclusion	27
Conclusion générale.....	28

Introduction générale

De nos jours, la robotique de surveillance est omniprésente et on la retrouve dans de nombreux domaines d'activité (militaire, industrielle, exploitation spéciale, etc.) et lieux de vie (maisons, espace public, institution publique et privée, etc.)

Les besoins de l'installation des systèmes de surveillance sont divers, toutefois la sécurité publique ainsi que la protection des biens mobiliers ou immobiliers font office d'éléments principaux dans la justification de la vidéosurveillance.

La surveillance, qui était autrefois accomplie uniquement par des agents de sécurité, a vu arriver la technologie comme une révolution. Les robots de surveillance permettent d'aider les agents de sécurité dans les tâches les plus difficiles, répétitives ou pénibles, tel que surveiller les zones non couvertes par des caméras de surveillance ou les endroits dangereux (station nucléaire, zone de guerre).

L'objectif de ce travail est de développer une application de commande à distance multi-utilisateurs pour un robot de surveillance via une application web avec l'établissement d'une communication en temps réel.

Notre travail est structuré en trois chapitres. Dans le premier chapitre qui s'intitule « Etat de l'art sur la robotique de surveillance », il a pour but de définir des concepts essentiels à la robotique de surveillance.

Le second chapitre s'intitule « Analyse et conception ». Dans ce chapitre, nous avons spécifié et analysé les besoins, présenté l'architecture du système, ainsi que les diagrammes (UML) associés aux différentes interactions de l'utilisateur avec l'application.

Dans le dernier chapitre, intitulé « Réalisation et implémentation », on a exposé les différents outils de développement et les environnements de travail utilisés, ainsi que le côté matériel. Des captures d'écrans des interfaces et un scénario de test montrant l'implémentation des différentes fonctionnalités de l'application font aussi partie de ce chapitre.

Ce présent mémoire se termine par une conclusion résumant les connaissances acquises à travers la réalisation du projet, ainsi que des perspectives de développement pouvant venir améliorer notre système dans le futur.

Chapitre 1 : Etat de l'art sur la robotique de surveillance

1.Introduction

La surveillance désigne le processus de suivi d'une situation, région ou personne. Cela se produit à titre d'exemple dans un scénario où la surveillance militaire des frontières et du territoire ennemi est essentielle à la sécurité d'un pays. La surveillance humaine est réalisée en déployant à proximité des zones sensibles, du personnel pour surveiller les changements. Cependant la capacité limitée des humains ne permet pas une surveillance efficace, ce qui a mené les développeurs à enrichir les systèmes de surveillance avec des robots. Outre l'avantage évident de protéger les vies du personnel, les robots terrestres et aériens peuvent également rechercher des détails qui ne sont pas visibles pour les humains. Et cela, grâce à des caméras de haute résolution et des divers capteurs, permettant d'obtenir des informations de grandes précisions.

De ce fait, l'utilisation des robots élargit considérablement le potentiel des systèmes de surveillance classique, qui peuvent évoluer du rôle passif traditionnel, où le système ne peut que détecter des événements et déclencher des alarmes, à la surveillance active, où un robot peut être utilisé pour interagir avec l'environnement, les humains ou d'autres robots pour des actions coopératives plus complexes (Dehual, Z.etal. ,2007) [1] ,(Vig, L.&Adams, J.A.,2007)[2].

1.1.La Robotique de surveillance

C'est un domaine basé sur deux concepts principaux qui sont : la robotique et les systèmes de surveillance.

1.1.1. La robotique

Est un ensemble des domaines scientifiques et industriels en rapport avec la conception et la réalisation de robots.

Dans le domaine industriel, la robotique produit des automates réalisant des fonctions précises sur des chaînes de montage. La robotique produit aussi des engins capables de se mouvoir dans différents milieux : dangereux (pollués, radioactifs...), aérien, sous-marins, spatiaux...

Outre l'industrie, la robotique concerne ainsi aujourd'hui la recherche scientifique, l'exploration spatiale et les activités de défense militaire ou de maintien de l'ordre. Elle intéresse également le secteur médical, pour les prothèses, les assistances aux chirurgiens ou aux infirmiers.

La robotique s'est également étendue récemment au marché grand-public avec des engins autonomes pour assurer seuls certaines fonctions (aspirateurs, tondeuses à gazon...) ou pour des activités ludiques (robots compagnons, robots jouets...).

1.1.2. Les systèmes de surveillance

Est un ensemble d'éléments mis en place permettant de détecter une situation de danger potentiel et de prévenir les conséquences. De manière générale, il se compose de pièces de détection d'un module de contrôle et d'un avertisseur.

1.2.Avantages

- Minimisation des situations à risque sur l'homme.
- Plus d'efficacité en surveillance.
- Réduction des couts de main-d'œuvre.

1.3.Inconvénients

- Difficulté de la mise œuvre des robots totalement autonomes.
- La fiabilité n'est pas toujours garantie.
- Source d'accident
- Non-respect de la vie privée

1.4.Type robot de surveillance

On peut dénombrer six types de robot de surveillance :

1.4.1. Les robots de surveillance domestique

Ce sont des robots qui assurent la sécurité dans les maisons. Ces robots qui grâce à leurs algorithmes peuvent détecter et informer des anomalies :

- prévenir dans le cas d'un incendie
- fuite d'eau
- la présence d'une personne étrangère dans la maison.

C'est ainsi qu'ils assurent la surveillance du domicile.

Ils peuvent aussi apporter une assistance aux personnes en pertes d'autonomie, ou les personnes âgées. C'est pour cela que ses robots sont définis comme des gardiens intelligents(Voir figure1).



Figure 1:robot de surveillance domestique

1.4.2. Le robot d'entrepôt

Ce robot qui opère dans les entrepôts des zones industrielles est plus spécifique que les robots qui sillonnent les bureaux, les aéroports et les rues. En effet il est conçu pour effectuer des rondes automatisées et détecter des incidents d'entrepôt comme les incendies. C'est ainsi que grâce à ses capteurs, micros et caméras intégrés il peut notifier des fuites d'eau, de présence de fumée et des intrusions. Donc c'est un robot de surveillance et de sécurité. (Voir figure2)



Figure 2:Le robot d'entrepôt.

1.4.3. Spécialisé pour les datacenters

Le robot spécialisé pour les datacenters est celui qui effectue la surveillance dans les salles d'hébergements. Ce type de robot est capable de se déplacer librement, en évitant les obstacles, dans les allées du datacenter sur la base de la cartographie des lieux. Il possède des capteurs et des outils de communication lui permettant d'assurer une surveillance 24h/24 et 7j/7. Pilotable à distance, il peut signaler une présence de chaleur ou d'humidité inhabituelle (Voir figure 3).



Figure 3:Robot pour les datacenters

1.4.4. Le robot d'intervention

Le robot d'intervention est un robot utile dans les interventions suite à des catastrophes naturelles, il permet d'explorer des lieux inaccessibles à l'homme, pour la recherche de victimes, il est également utilisés par les forces de l'ordre pour le désamorçage ou la destruction des colis piégés.

Il peut aussi intervenir lors d'un incendie pour inspecter et reconnaître la zone dangereuse dans le but d'éloigner l'Homme du risque (Voir figure4).



Figure 4:Robot d'intervention

1.4.5. Les robots patrouilleurs

Destiné à patrouiller des grands sites extérieurs, il s'adresse tout particulièrement aux sites industriels, aux ports, aéroports et aux sites sensibles. Équipé de caméras offrant une vue à 360° et d'un système de détection humaine, il signale les intrusions sur le site. IL est conçu pour sillonner inlassablement les espaces à sécuriser et intervenir rapidement en se rendant sur place en cas d'alarme. Fonctionnel sur tous types de terrains, de jour comme de nuit, il assure une sécurité sans failles(Voir figure 5).



Figure 5:Robot patrouilleur

2. Travaux existant dans la robotique de surveillance

La nécessité croissante d'une surveillance automatisée des environnements intérieurs, tels que les aéroports, les entrepôts, les usines de production, etc. a stimulé le développement de systèmes intelligents basés sur des capteurs mobiles. Ces dernières années, plusieurs projets mondiaux ont tenté de développer des plateformes de sécurité mobiles.

Un exemple notable est le Système mobile de détection, d'évaluation et d'intervention (MDARS). (Everett, H.&Gage, D.W.,1999) [3]. Ce projet visait à développer un système multirobots capable d'inspecter les entrepôts et les sites d'entreposage, d'identifier les situations anormales comme les inondations et les incendies, de détecter les intrus et de déterminer le statut des objets inventoriés à l'aide de transpondeurs RF spécialisés.

Dans le projet RoboGuard (Birk, A.&Kenn, H.,2001) [4], un dispositif de sécurité mobile semi-autonome utilise une architecture comportementale pour la navigation, tout en envoyant des flux vidéo aux gardes humains.

Le robot expert de surveillance nocturne d'aéroport (ANSER) (Capezio, F.etal. 2005) [5] est un véhicule terrestre sans pilote (UGV) utilisant un GPS non différentiel pour les patrouilles de nuit dans les aéroports civils et dans des zones étendues similaires, en interaction avec une station de surveillance fixe sous le contrôle d'un opérateur humain (Voir figure 6).



Figure 6: Robot UGV

Un robot de sécurité (Duckett, T.etal. ,2004) [6] pour la surveillance à distance des environnements intérieurs a également fait l'objet d'un projet de recherche au Laboratoire des systèmes d'apprentissage de l'AASS. L'objectif de ce projet était de développer une plateforme robotique mobile capable de patrouiller dans un environnement donné, d'acquérir et de mettre à jour des cartes, de surveiller des objets de valeur, de reconnaître des personnes, de distinguer les intrus des personnes connues et de fournir aux opérateurs humains à distance une analyse sensorielle détaillée (Voir figure7).



Figure 7: Robot patrouilleur d'intérieur

Un autre exemple de robot de sécurité est celui mis au point à l'Université de Waikato, à Hamilton, en Nouvelle-Zélande (Carnegie, D.A.etal. 2004) [7]. Il s'appelle MARVIN (Mobile Autonomous Robotic Vehicle for Indoor Navigation) et a été conçu pour agir comme un agent de sécurité dans les environnements intérieurs. Afin d'interagir avec les humains, le robot est équipé d'un logiciel de reconnaissance et de synthèse vocale ainsi que de la capacité de transmettre des états émotionnels, verbalement et non verbalement (Voir figure8).



Figure 8: Robot MARVIN

3. Conclusion

Dans ce chapitre nous avons présenté un état de l'art sur la robotique de surveillance, les domaines d'application, ainsi que les travaux existant dans ce domaine. Nous allons par la suite aborder l'étude conceptuelle afin de répondre au besoin de notre application.

Chapitre 2 : Analyse et conception

1.Introduction

Dans ce chapitre nous allons aborder l'étude conceptuelle de notre application, qui consiste à concevoir un système, satisfaisant les objectifs du projet, ces objectifs rappelons-les, consistent à commander à distance un robot de surveillance à travers une plateforme multiutilisateurs.

Dans notre étude nous nous sommes basés sur une modélisation orientée objet avec le langage de Modélisation Unifié (UML), pour ces avantages d'adaptation à un nombre important d'applications en termes de : types, tailles, et domaines.

2.Identification des besoins

2.1. Spécification des besoins

Cette section identifie les besoins fonctionnels puis non fonctionnels que notre système doit satisfaire.

2.1.1. Identification des acteurs :

Un acteur est une entité externe (utilisateur, système, etc.), qui interagit directement avec le système [8]. Dans notre application nous deux acteurs : Simple utilisateur et administrateur.

2.1.2. Les besoins fonctionnels

Les besoins fonctionnels représentent les fonctionnalités principales de notre solution sont :

Acteur	Fonction	Description
Utilisateur simple/Administrateur	Authentifier	Un utilisateur peut s'authentifier en saisissant un nom d'utilisateur et un mot de passe. Cela permettra d'accéder à l'application.
Administrateur	Gérer les comptes des utilisateurs	L'administrateur peut consulter les utilisateurs inscrits dans l'application ainsi que leurs informations, créer un compte pour un utilisateur, modifier leurs informations. Il peut aussi supprimer les utilisateurs
Utilisateur simple/Administrateur	Modifier profil	Un utilisateur a la possibilité de changer les informations de son profil.
Utilisateur simple/Administrateur	Visualiser le flux vidéo du robot	Un utilisateur peut visualiser le flux vidéo transmis par le robot s'il 'est connecté.
Utilisateur simple/Administrateur	Manipuler le robot	Un utilisateur peut manipuler le robot si le robot est connecté et si y a aucuns autres utilisateurs qui le manipule, ou des utilisateurs qui sont plus privilégiés pour le manipuler.
Utilisateur simple/Administrateur	Enregistrer un itinéraire	Un Utilisateur peut enregistrer un itinéraire s'il peut le manipuler le robot.
Utilisateur simple/Administrateur	Lancer un itinéraire	Un Utilisateur peut lancer un itinéraire s'il peut le manipuler le robot.

2.1.3. Besoins non fonctionnels :

Les besoins non-fonctionnels de notre solution sont suivants :

- Fonctionner dans n'importe quel navigateur (chrome, ...) ;
- Sécuriser les informations personnelles des utilisateurs ;
- Fluidité ;
- Ergonomie ; Simple à utiliser ;
- Utilisation dans un réseau interne (ne dépend d'une connexion internet).

2.2. Messages échangés entre le système et les acteurs

2.2.1. Messages entrants vers le système :

- Demande d'authentification (utilisateur / Administrateur).
- Création de comptes (Administrateur).

- Modification d'un compte (Administrateur).
- Suppression d'un compte (Administrateur).
- Modification des informations du profiles (utilisateur/administrateur).
- Les vitesses de déplacement du robot (utilisateur/administrateur).
- Le flux vidéo/audio (Robot/Utilisateur).

2.2.2. Messages sortant du système

- Pages d'authentification (utilisateur/administrateur)
- Menu principale (utilisateur/administrateur)
- Page gestion des comptes (Administrateur)
- Page gestions du profile (utilisateur/administrateur)
- Page manipulation du robot (utilisateur/administrateur)
- Flux vidéo/audio envoyé par le robot (utilisateur/administrateur)
- Les vitesses de déplacement du robot (Robot)

3. Diagramme cas d'utilisation

Un cas d'utilisation est la description d'un ensemble d'actions qu'un système effectue pour produire un résultat observable à un acteur. Les diagrammes de cas d'utilisation mettent l'accent sur ce que les acteurs font sur le système plutôt que sur la façon dont il le fait [9].

Dans ce qui suit, nous présentons un diagramme de cas d'utilisation global afférents aux acteurs cités précédemment (Voir figure 9).

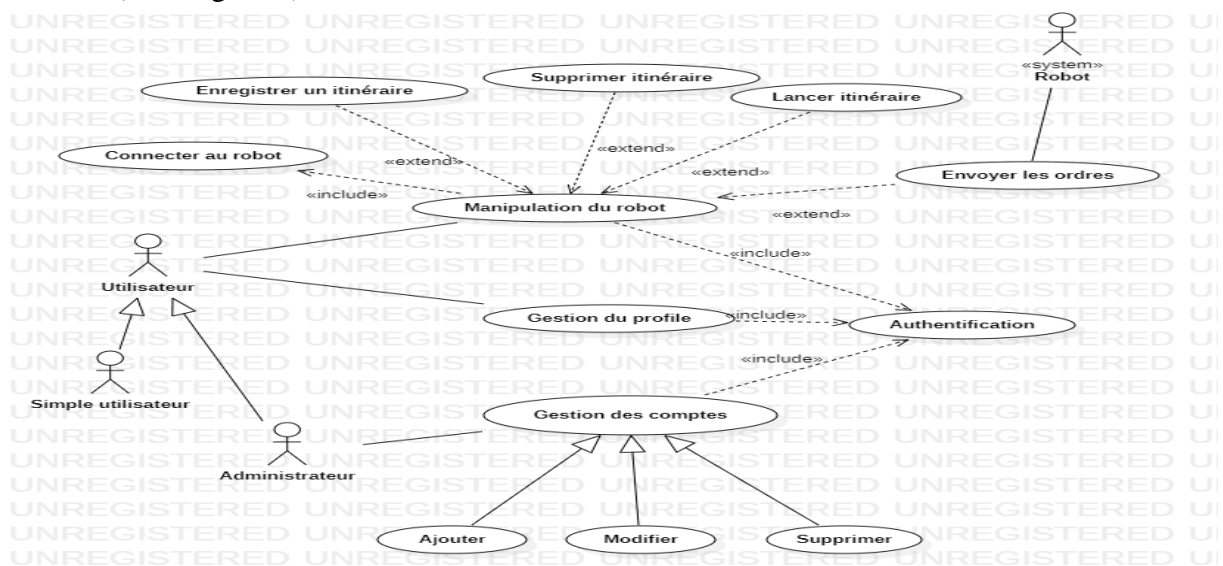


Figure 9: Diagramme cas utilisation globale

3.1. Description textuelle

3.1.1. Cas d'utilisation « authentification »

Titre	Authentification.
But	Sécuriser l'application.
Résumé	L'utilisateur accède à la page d'authentification pour s'identifier et accéder au menu principal de l'application.
Description des enchaînements	
Prés-condition	Affichage page authentification
Enchaînements nominaux	A- L'utilisateur accède à la page d'authentification. B- L'utilisateur saisis son pseudo et mot de passe. C- L'utilisateur click sur le bouton authentifier. D- Pseudonyme et mots de passe correct (accès a l'application).

Scenario alternatif	- Pseudo incorrect. Afficher le message d'erreur « Pseudonyme incorrecte ». -Pseudo correct mais mot de passe incorrect. Afficher message d'erreur « Mot de passe incorrect ».
Postcondition	Utilisateur redirigé vers la page menu principal de l'application.

3.1.2. Cas d'utilisation « Gestion profil »

Titre	Gestion de profil.
But	Mise à jour information profil.
Résumé	L'utilisateur accède à la page Gestion du profil, un formulaire s'affiche concernant ses informations personnelles et peut les modifier.
Description des enchaînements	
Précondition	Affichage page Gestion profil.
Enchaînements nominaux	A- L'utilisateur accède à la page Gestion profil. B- Formulaire information personnel s'affiche. C- L'utilisateur modifier ces informations. D- L'utilisateur clic sur le bouton modifier. E- Le système modifier les informations.
Scenario alternatif	-Si l'utilisateur décide de quitter la page de gestion de profil alors il sera redirigé vers la page menu principale. -Information saisis incorrecte. Afficher message erreur « Cette information est incorrecte ». -Champ obligatoire non remplis. Afficher message « Veuillez remplir tous les champs obligatoire ».
Postcondition	Le système mise à jour et enregistre les informations dans la base de données.

3.1.3. Cas d'utilisation « Gestion des comptes »

• Ajouter compte

Titre	Ajouter compte.
Bute	La gestion de tous les comptes de l'applications.
Résumé	L'administrateur accède à la page Gestion des comptes, peut voir tous les comptes et ajouter un compte.
Description des enchaînements	
Précondition	Affichage page Gestion des comptes.
Enchaînements nominaux	A- L'administrateur accède à la page Gestion des comptes. B- Table des comptes s'affiche. C- L'administrateur remplis le formulaire d'informations. D- Clic sur le bouton ajouter. E- Le système vérifier les informations modifier.
Scenario alternatif	-Si l'utilisateur décide de quitter la page de gestion des comptes alors il sera redirigé vers la page menu principale. -Information saisis incorrecte. Afficher message erreur « Cette information est incorrecte ». -Champ obligatoire non remplis. Afficher message « Veuillez remplir tous les champs obligatoire ».
Postcondition	Le système mis à jour et enregistre les informations dans la base de données

• Cas d'utilisation « Modifier compte »

Titre	Modifier compte
Bute	La gestion de tous les comptes de l'applications

Résumé	L'administrateur accède à la page Gestion des comptes, peut voir tous les comptes et modifier un compte.
Description des enchaînements	
Précondition	Affichage page Gestion des comptes
Enchaînements nominaux	A- L'administrateur accède à la page Gestion des comptes B- Table des comptes s'affiche C- L'administrateur sélectionne une ligne du tableau des comptes D- Le formulaire d'information se remplit avec les informations de la ligne sélectionné E- L'administrateur clic sur le bouton modifier. F- Le système modifier les informations.
Scenario alternatif	-Si l'utilisateur décide de quitter la page de gestion des comptes alors il sera redirigé vers la page menu principale. -Information saisis incorrecte. Afficher message erreur « Cette information est incorrecte » -Champ obligatoire non remplis. Afficher message « Veuillez remplir tous les champs obligatoire »
Postcondition	Le système mis à jour et enregistre les informations dans la base de données.

● **Cas d'utilisation « supprimer compte »**

Titre	Supprimer compte
But	La gestion de tous les comptes de l'application
Résumé	L'administrateur accède à la page Gestion des comptes, peut voir tous les comptes, supprimer un compte
Description des enchaînements	
Précondition	Affichage page Gestion profil
Enchaînements nominaux	A- L'administrateur accède à la page Gestion des comptes B- Table des comptes s'affiche C- L'administrateur sélectionne une ligne du tableau des comptes D- Le formulaire d'information se remplit avec les informations de la ligne sélectionné E- L'administrateur clic sur le bouton supprimer F- Le système vérifier les informations
Scenario alternatif	-Si l'utilisateur décide de quitter la page de gestion des comptes alors il sera redirigé vers la page d'accueil
Postcondition	Le système mis à jour et enregistre les informations dans la base de données

3.1.4. Cas d'utilisation « Manipulation robot »

● **Cas d'utilisation « Manipuler le robot »**

Titre	Manipuler le robot compte
But	Prendre les commandes du robot
Résumé	L'utilisateur accède à la page de manipulation robot, récupérer le flux vidéo des camera du robot et demande de prendre les commandes du robot s'il est autorisé il prend les commandes du robot
Description des enchaînements	
Précondition	Affichage page Manipulation robot
Enchaînements nominaux	A. L'utilisateur accède à la page manipulation du robot B. L'utilisateur clic sur le bouton connecter au robot

	C. Le flux vidéo du robot est afficher D. L'utilisateur click sur manipuler le robot. E. Le système dans la possibilité a l'utilisateur de manipuler le robot F. L'utilisateur manipule le robot
Scenario alternatif	B.1- Le robot n'est pas connecter alors afficher message « le robot n'est pas connecter » C.1- Le robot se déconnecte alors afficher message « le robot n'est pas connecter » F.1- Le robot est entrain d'être manipuler et le nouveau utilisateur a plus de privilège que l'utilisateur actuel. demande accepter, rendre la main au nouveau utilisateur et metre en attente l'ancien utilisateur et afficher message « Accès refuser »
Postcondition	L'utilisateur contrôle le robot

• Cas d'utilisation enregistrer un itinéraire

Titre	Enregistrer un itinéraire
But	Enregistrer un itinéraire
Résumé	L'utilisateur clic sur le bouton enregistrer itinéraire, ensuite il saisit l'itinéraire qu'il veut enregistrer, à la fin il clic sur le bouton fin enregistrement et le système enregistre son itinéraire.
Description des enchaînements	
Précondition	L'utilisateur contrôle le robot
Enchaînements nominaux	A- L'utilisateur clic sur le bouton enregistrer itinéraire B- L'utilisateur remplis les informations de l'itinéraire C- Utilisateur manipule le robot. D- L'utilisateur clic sur fin enregistrement de l'itinéraire
Scenario alternatif	- Erreur lors que le système enregistre l'itinéraire alors afficher un message d'erreur. - L'utilisateur se déconnecte alors mettre fin à l'enregistrement.
Postcondition	Le système enregistre l'itinéraire.

• Cas d'utilisation lancer un itinéraire

Titre	Lancer un itinéraire
But	Lancement d'un itinéraire
Résumé	L'utilisateur clic sur un itinéraire dans le tableau des itinéraires déjà enregistrer, clic sur le bouton lancer itinéraire.
Description des enchaînements	
Précondition	L'utilisateur contrôle le robot
Enchaînements nominaux	A- L'utilisateur clic sur un itinéraire dans le tableau des itinéraires. B- L'utilisateur clic sur lancer l'itinéraire C- Le système récupère le fichier de l'itinéraire et l'envoyer au robot D- Le robot exécute l'itinéraire
Scenario alternatif	- Erreur lors de la récupération de l'itinéraire afficher message d'erreur. - Erreur lors de l'exécution de l'itinéraire par le robot afficher message d'erreur.
Postcondition	L'itinéraire est bien exécuté.

• Cas d'utilisation supprimer un itinéraire

Titre	Supprimer un itinéraire
But	Suppression d'un itinéraire
Résumé	L'utilisateur clic sur un itinéraire dans le tableau des itinéraires déjà enregistrer, clic sur

	le bouton supprimer itinéraire.
Description des enchaînements	
Précondition	L'utilisateur contrôle le robot
Enchaînements nominaux	A. L'utilisateur clic sur un itinéraire dans le tableau des itinéraires. B. L'utilisateur clic sur Supprimer l'itinéraire.
Scenario alternatif	C. L'utilisateur clic sur le bouton supprimer itinéraire sans choisir un itinéraire alors afficher message d'erreur. D. Erreur lors que le système supprime un itinéraire alors afficher un message d'erreur.
Postcondition	Le système supprime l'itinéraire.

4.Diagrammes d'activités

Un diagramme d'activité permet de modéliser le comportement du système, dont la séquence des actions et leurs conditions d'exécution. Les actions sont les unités de base du comportement du système [10].

Dans cette partie, nous allons présenter les scénarios des cas d'utilisations les plus importants par des diagrammes d'activité :

- Diagramme d'activité du cas d'utilisation « gestion profil »

La figure 10 illustre le diagramme d'activité du cas d'utilisation gestion profil.

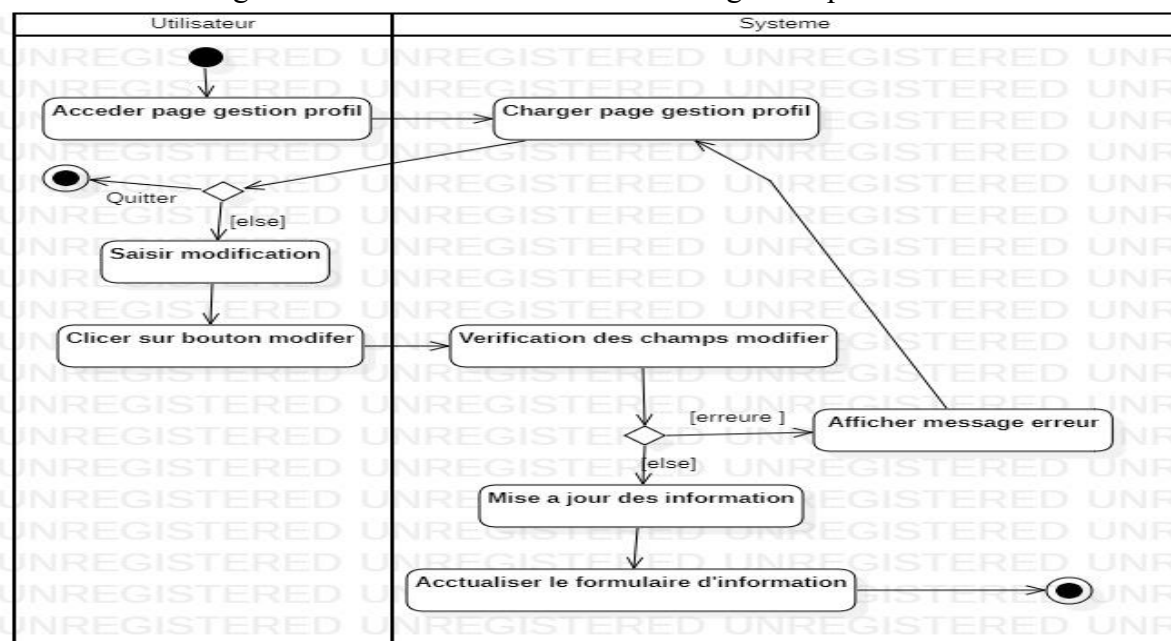


Figure 10:Diagramme d'activité du cas d'utilisation « gestion profil »

- Diagramme d'activité du cas d'utilisation « manipulation robot »

La figure 11 illustre le diagramme d'activité du cas de manipulation robot.

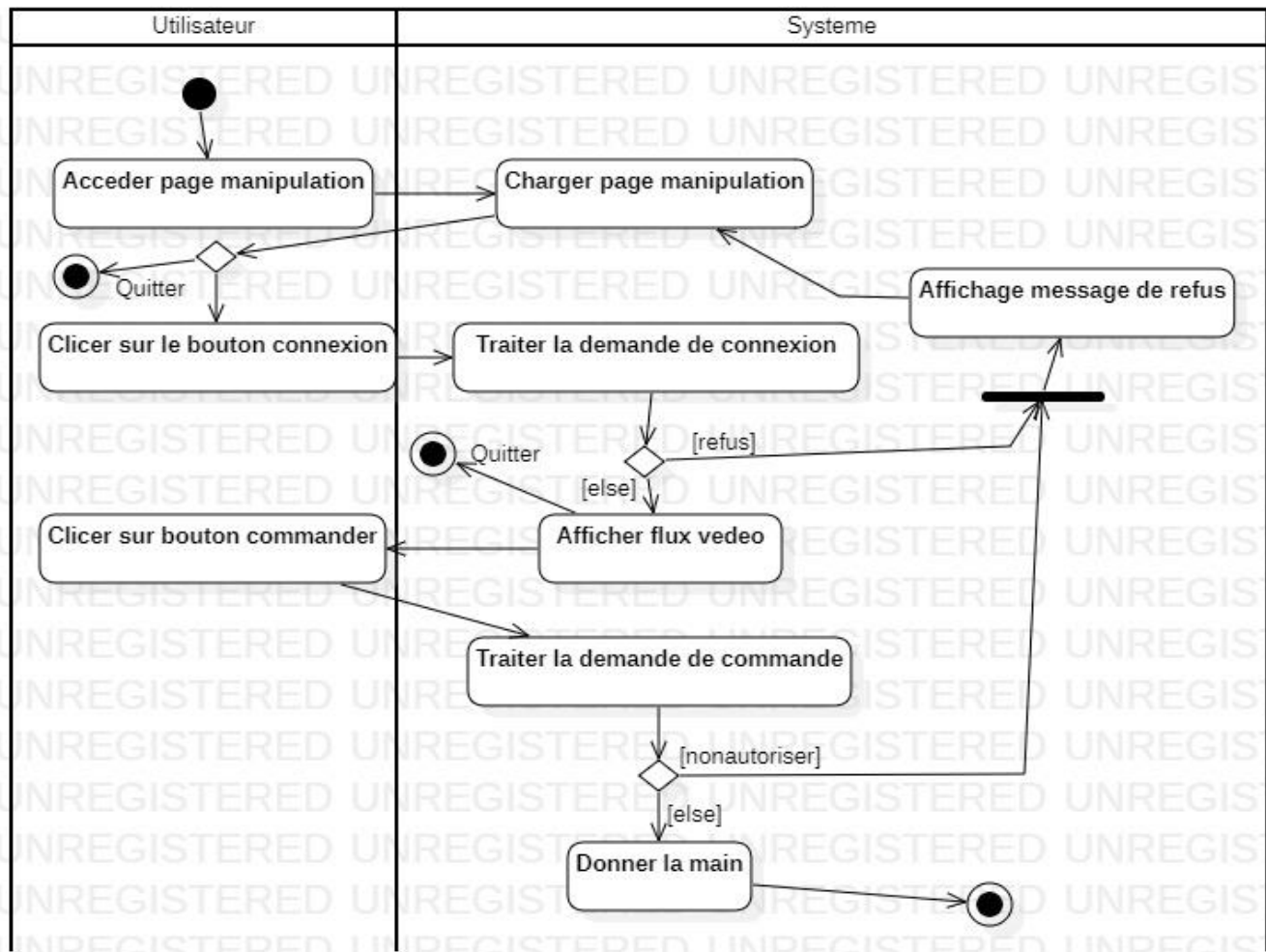


Figure 11:Diagramme d'activité du cas d'utilisation « manipulation robot »

La figure 12 illustre le diagramme d'activité du cas d'utilisation d'authentification.

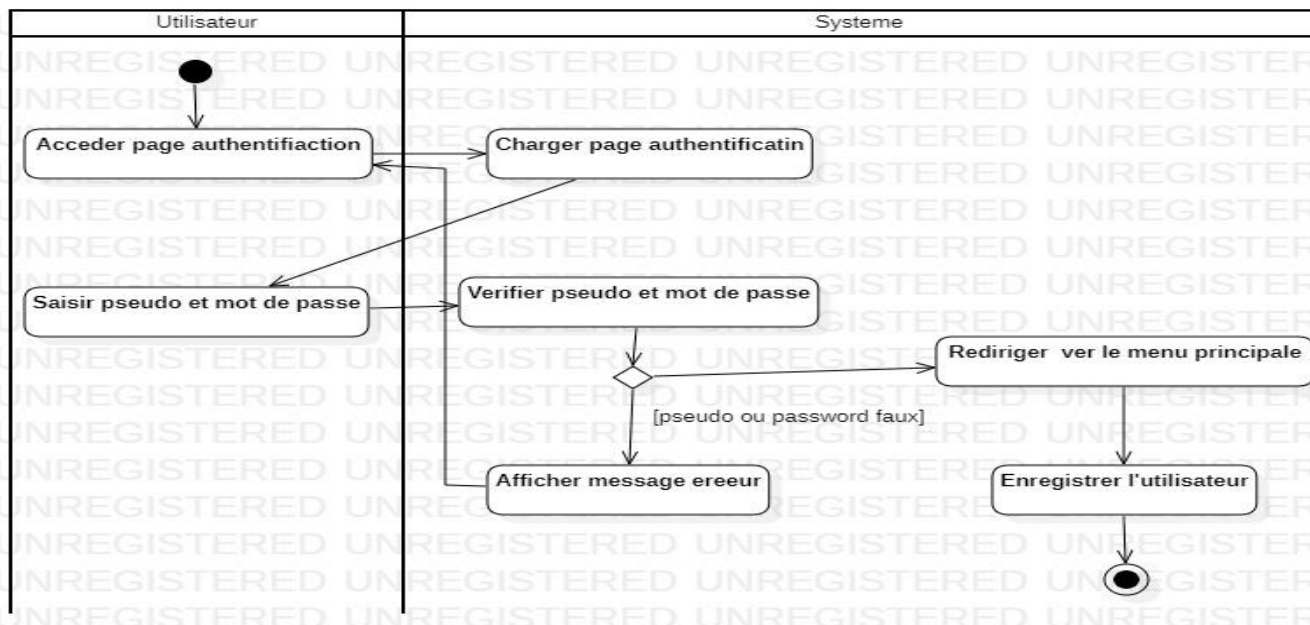


Figure 12:Diagramme d'activité du cas d'utilisation « authentification »

5. Diagramme de classe

- Un diagramme de classes fournit une vue globale d'un système en présentant ses classes, interfaces et collaborations, et les relations entre elles [11].

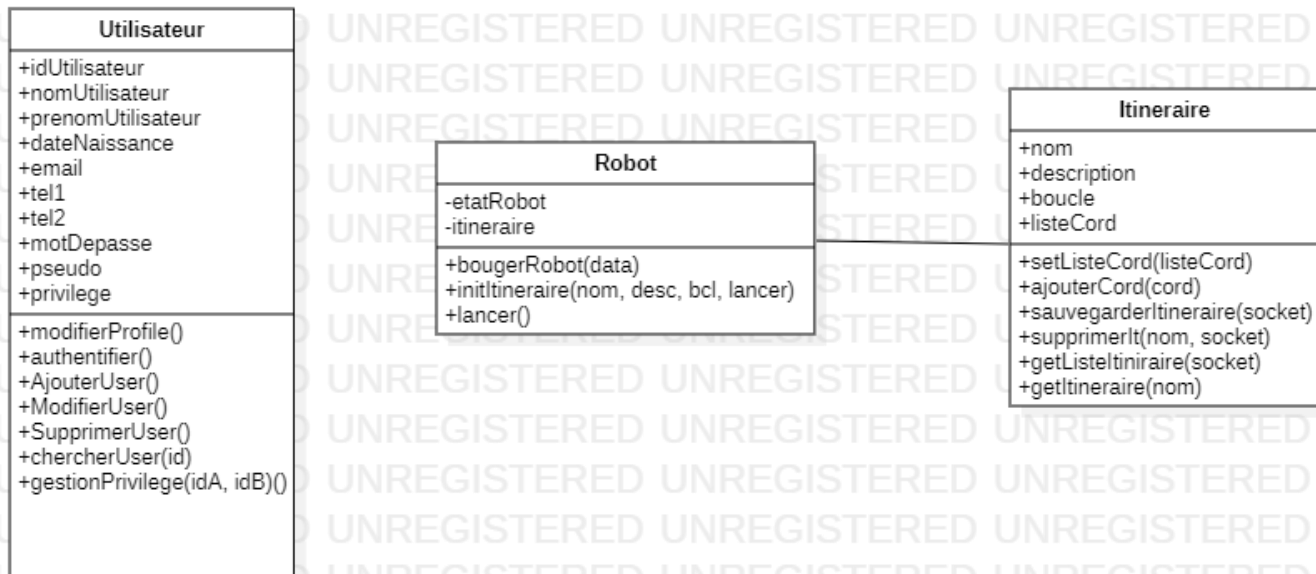


Figure 13: Diagramme de classe globale

5.1. Description du diagramme de classe

5.1.1. La classe Utilisateur

Attributs	Type	Description
IdUtilisateur	String	Identifiant de l'utilisateur, codifier sur 8 caractères numérique tel que (2 pour jours de naissance, 2 pour mois de naissance, 2 pour année de naissance et 2 code séquentiel si plusieurs utilisateurs sont née la même date)
NomUtilisateur	String	Nom de l'utilisateur
PrenomUtilisateur	String	Prénom de de l'utilisateur
DateNaissance	Date	Date de naissance de l'utilisateur
Email	String	Email de l'utilisateur
Tel1	String	Numéro de téléphone de l'utilisateur
Tel2	String	Numéro de téléphone de l'utilisateur
Pseudo	String	Pseudo de l'utilisateur avec qui il se connecte
MotDePasse	String	Mot de passe de l'utilisateur
Privilège	Int	Le privilège de l'utilisateur à manipuler le robot (0 correspond au plus grand privilège)
Méthodes	Type	Description
Authentifier ()	Boolean	Permet d'accéder a l'application en vérifiant si le pseudo est mot de passe sont juste
modifierProfile()	Boolean	Permet de modifier les informations du profil de l'utilisateur
ajouterUser()	Boolean	Permet d'ajouter un utilisateur
modifierUser()	Boolean	Permet de modifier les informations de l'utilisateur
supprimerUser()	Boolean	Permet de supprimer un utilisateur
getPrivilege(idA,idB)	Boolean	Permet de retourner l'utilisateur le plus privilégié
chercherUser()	Boolean	Permet de chercher un utilisateur

5.1.2. La classe robot

Attributs	Type	Description
EtatRobot	String	Etat du robot (connecter/déconnecter)

Itinéraire	Objet	Sauvegarde d'un itinéraire
Méthodes	Type	Description
bougerRobot(data)	Void	Envoyer les coordonnées au robot pour qu'il est exécute
initItineraire(nom, desc, bcl, lancer)	Boolean	Enregistrer un itinéraire
lancer()	Boolean	Permet de lancer un itinéraire

5.1.3. La classe Itinéraire

Attributs	Type	Description
Nom	String	Nom de l'itinéraire
Description	String	Description de l'itinéraire
Boucle	Boolean	Itinéraire est une boucle ou non
listeCord	ArrayList<>	La liste des coordonnées d'un itinéraire
Méthodes	Type	Description
sauvegarderItineraire(socket)	Boolean	Permet de sauvegarder l'itinéraire dans un fichier
getItineraire(nom)	Itinéraire	Permet de renvoyer un itinéraire
getListeItineraire(socket)	ArrayList<Itinéraire>	Permet de renvoyer la liste des itinéraire
ajouterCord(cord)	Boolean	Permet d'ajouter une cordonne (linéaire et angulaire) a l'itinéraire actuelle.
supprimerIt(nom, socket)	Boolean	Permet de supprimer un itinéraire.

6.Développement du modèle dynamique

6.1. Diagramme d'état machine globale

Les diagrammes d'état décrivent le comportement logique du système, une partie du système ou le protocole d'utilisation de celui-ci.

Ces diagrammes affichent les états possibles des objets et les transitions qui provoquent un changement d'état [12].

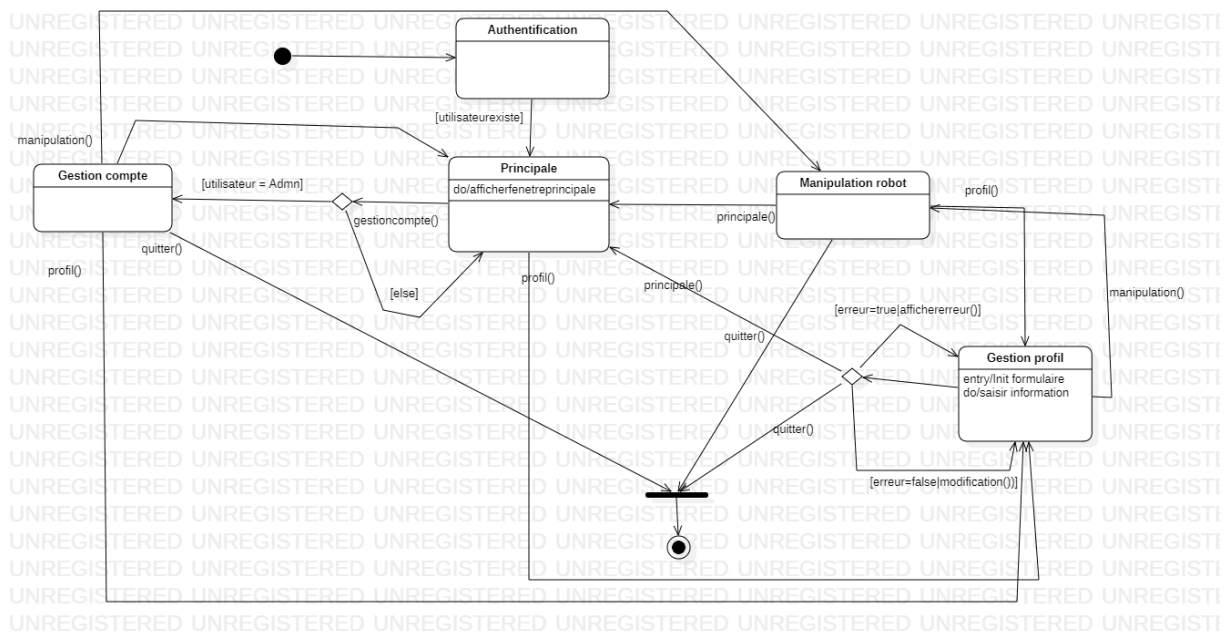


Figure 14: Diagramme d'état machine globale

6.2. Diagramme d'état machine authentication



Figure 15:Diagramme d'état machine authentication

6.3. Diagramme d'état machine de Gestion de comptes

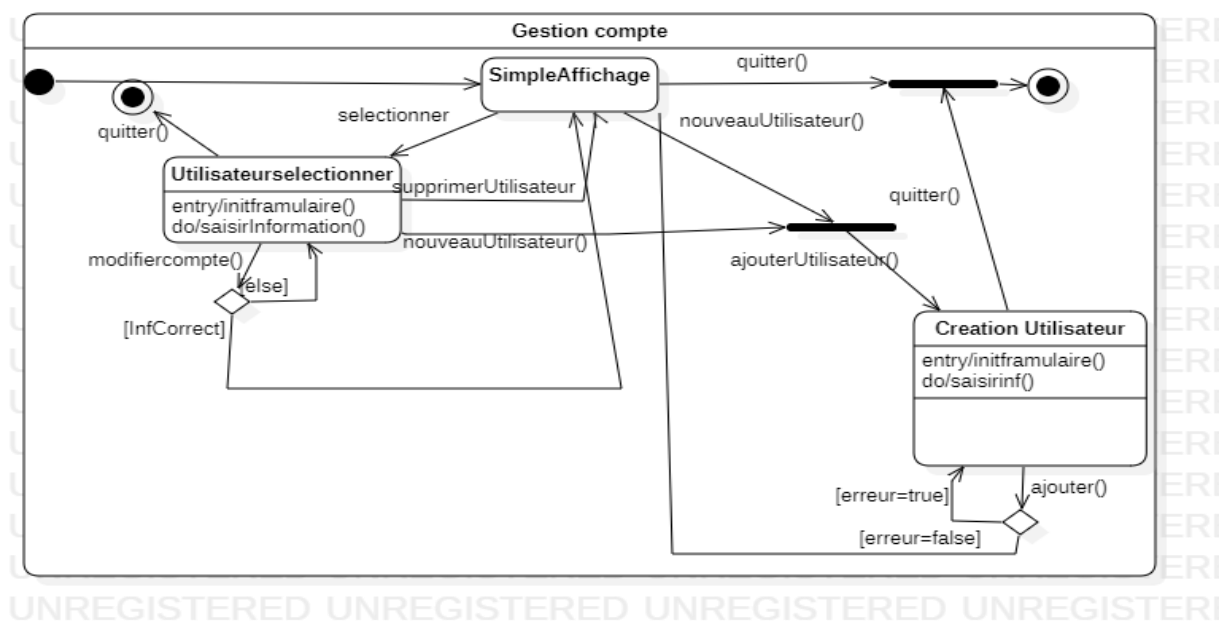


Figure 16:Diagramme d'état machine de Gestion de comptes

6.4. Diagramme d'état machine de manipulation

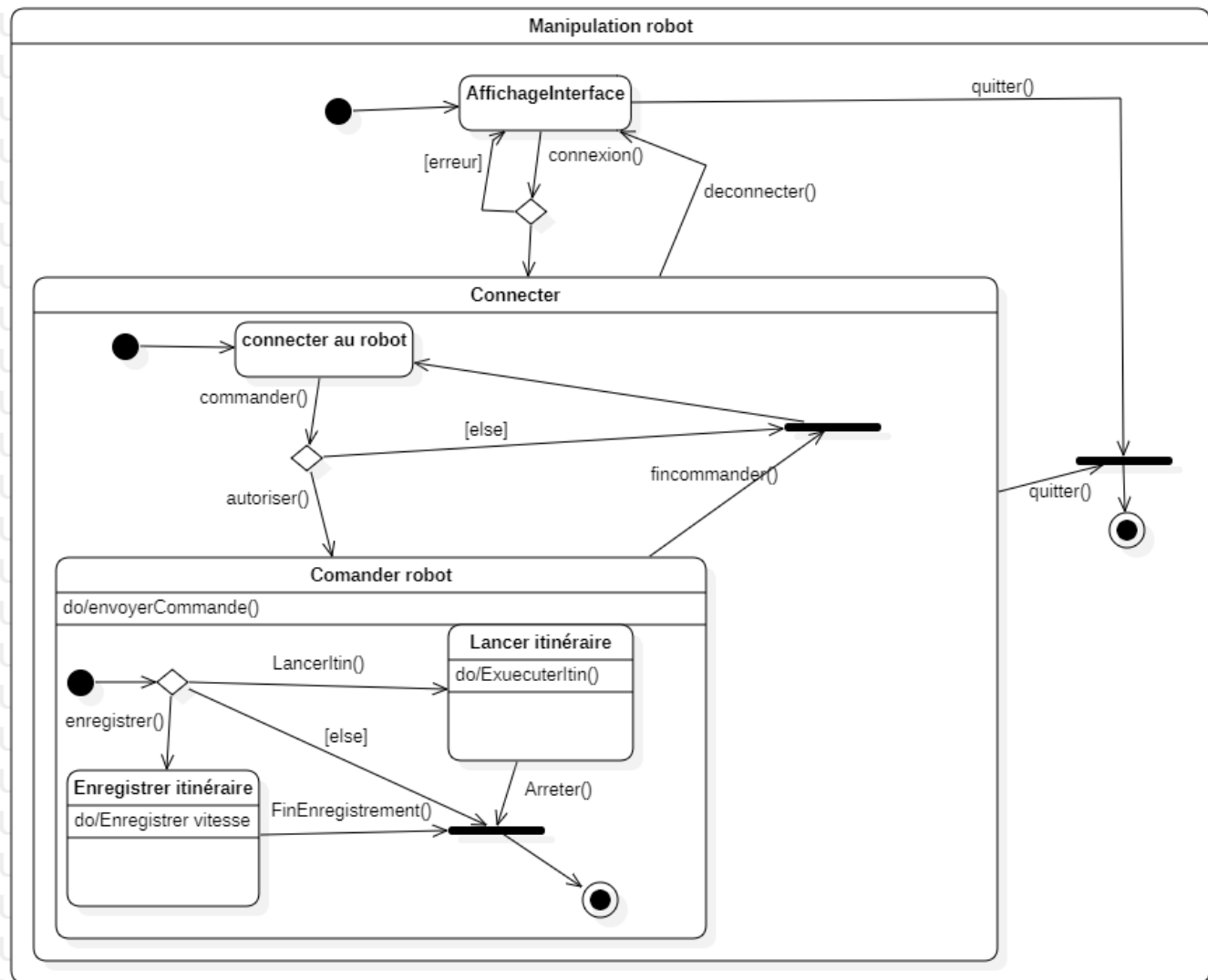


Figure 17:Diagramme d'état machine de manipulation

7.Diagramme des composant du système

Selon la spécification UML2.0, un diagramme de composants peut contenir des spécifications d'instance. Une spécification d'instance peut être définie par un ou plusieurs classificateurs. Vous pouvez utiliser les classes, les interfaces ou les composants comme un classificateur. Il est possible d'instancier un classificateur à l'aide de l'inspecteur d'objet ou de l'éditeur in-situ [13].

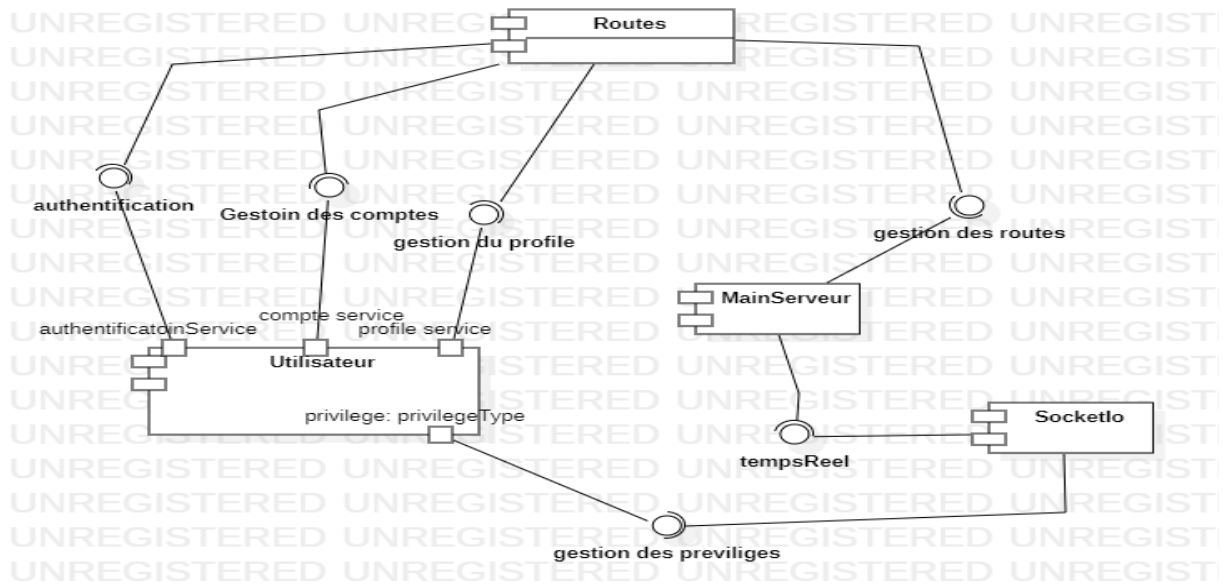


Figure 18:Diagramme des composant du système

8.Diagramme de déploiement

Le diagramme de déploiement spécifie un ensemble de constructions qui peut être utilisé pour définir l'architecture d'exécution de système qui représentent l'affectation d'artefacts logiciel a des nœuds. Les nœuds sont connectés via des chemins de communication pour créer des systèmes de réseau d'une complexité quelconque. Les nœuds sont généralement définis d'une manière imbriquée et représentent soit des périphériques matériels, soit des environnements d'exécution de logiciels. Les artefacts représentent des éléments concrets du monde physique qui sont le résultat d'un processus de développement. [14].

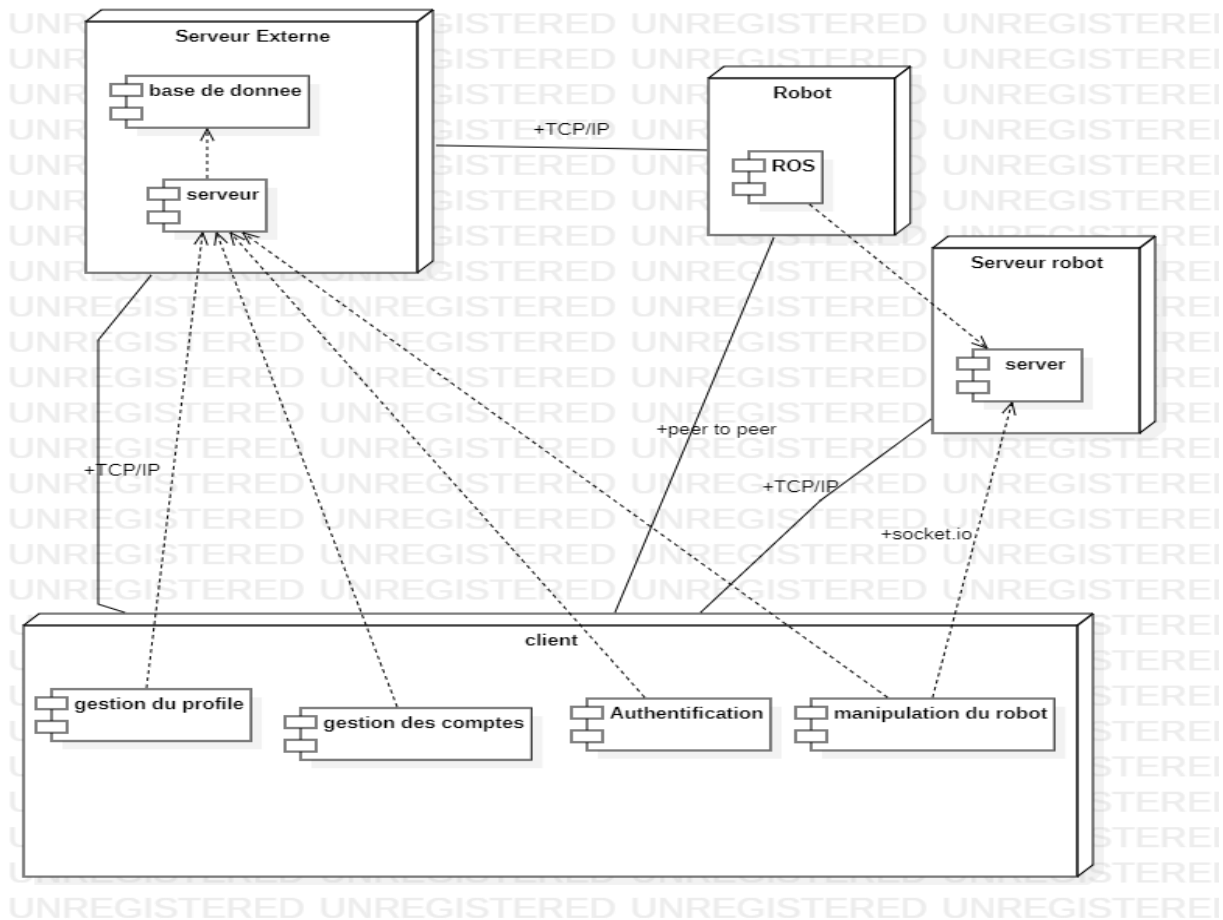


Figure 19:Diagramme de déploiement

9.Conclusion

Dans ce chapitre nous avons identifié les besoins fonctionnels et non fonctionnels. Ensuite, nous avons présenté la conception logicielle de l'application à travers le diagramme de cas d'utilisation, les diagrammes d'activités, le diagramme de classes, les diagrammes d'état machine, le diagramme de composant, et le diagramme de déploiement. Dans le chapitre suivant, nous allons passer à l'étape de la réalisation de notre application.

Chapitre 3 : Réalisation et implémentation

1.Introduction

Après avoir réalisé la phase de conception en mettant en évidence les aspects statiques et dynamiques de l'application, à travers les diagrammes UML, nous abordons la partie réalisation de l'application.

Dans ce chapitre, nous exposons l'architecture de notre application et son fonctionnement, l'environnement de développement à travers la définition des différents outils utilisés pour cette réalisation, ainsi que les principales fonctionnalités de cette application.

A propos du robot

Le robot mobile B21r est une plate-forme expérimentale construite par la société iRobot (Voir figure 20) pouvant se déplacer sur un terrain non accidenté ayant comme type de traction, la traction synchrone. Il dispose de quatre roues décentrées orientables tournant selon deux axes : une rotation selon l'axe y pour engendrer la translation, et une rotation selon l'axe verticale au sol pour engendrer une rotation sur lui-même. Ce robot mobile est muni de deux ceintures de capteurs à ultrasons, une ceinture de capteurs infrarouges, un laser, des capteurs tactiles placés le long de ses parois, et une caméra de profondeur Kinect.



Figure 20:Structure générale du robot

2. Outils et langages utilisés

2.1.HTML

HTML5 (Hypertext Markup Language5), est un langage de balisage pour la structure et la présentation des contenus Web. HTML5 prend en charge la syntaxe traditionnelle de style HTML et XHTML et d'autres nouvelles fonctionnalités dans son balisage, ses nouvelles API, son langage XHTML et sa gestion des erreurs.

2.2.CSS

Cascading Style Sheets (CSS) est un langage de feuille de style utilisé pour décrire la présentation d'un document écrit en HTML ou en XML.CSS décrit la façon dont les éléments doivent être affichés à l'écran, sur du papier, en vocalisation, ou sur d'autres supports [15].

2.3. JavaScript

JavaScript est un langage de programmation qui ajoute de l'interactivité à votre site Web. Cela se produit dans les jeux, dans le comportement des réponses lorsque vous appuyez sur des boutons ou lors de la saisie de données sur des formulaires ; avec un style dynamique ; avec l'animation, etc [16].

2.4. MySQL

Le plus populaire des serveurs de bases de données SQL Open Source, c'est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait

partie des logiciels de gestion de base de données les plus utilisés au monde, en concurrence avec Oracle, Informix et Microsoft SQL Server.

3. Implémentation

3.1. Les serveurs

Dans notre application nous avons utilisé deux serveurs : le serveur embarqué qui reçoit des données (vitesse angulaire et vitesse linéaire) et les envoie au robot pour le faire déplacer, et le serveur externe qui s'occupe de tout ce qui est lié à la base de données (de la gestion des utilisateurs et gestion des profils).

3.1.1. Serveur externe

- **Node.js**

Node.js est une plateforme logicielle libre et événementielle en JavaScript pour construire facilement des applications de réseau rapides et évolutives. Elle utilise la machine virtuelle V8 de Google Chrome. Node.js contient une bibliothèque de serveur HTTP intégrée, ce qui rend possible de construire ou de créer un serveur web sans avoir besoin d'un logiciel externe comme Apache ou autres. Il permet aussi de mieux contrôler la façon dont le serveur web fonctionne, tout en étant idéal pour les applications en temps réel intensives en données qui courent à travers des dispositifs distribués.

- **Les Modules npm**

Node.js contient plusieurs modules qui ont différents fonctionnements mais leur but est de faciliter l'utilisation du serveur.

- Express : La création de l'application et de ses comportements.
- Https : Créer le serveur HTTP sécurisé.
- Socket.io : Pouvoir interagir en temps réel avec les clients et le serveur.
- Simple-peer : Pour la communication media peer-to-peer.

Figure (Voir figure 21) ci-dessous est un extrait du fichier JavaScript du serveur qui montre les différents modules utilisés dans notre application.

```
const express = require('express');
var app = express();
var route = require('./modules/Root');
const https = require('https')
const fs = require('fs');
```

Figure 21: Liste des Modules utilisés.

3.1.2. Serveur embarqué

- **ROS (Robot Operating System)**

➤ **Définition**

Comme son nom l'indique, ROS (Robot Operating System) est un système d'exploitation pour robots, de même que les systèmes d'exploitation pour PC. ROS est middleware, quelque chose entre le software et le hardware. Il fournit des services proches d'un système d'exploitation (abstraction du matériel, gestion de la concurrence, des processus...) mais aussi des fonctionnalités de haut niveau (appels asynchrones, appels synchrones, base de données centralisée, système de paramétrage du robot...). [17]

➤ **L'organisation générale de ROS**

La philosophie de ROS se résume dans les 5 grands principes suivants :

- Peer to Peer : Les programmes individuels se concentrent sur une API définie (messages ROS, services, etc.).
- Distribué : Les programmes peuvent fonctionner sur plusieurs ordinateurs et communiquer sur le réseau.

- Multi langages : Les modules ROS peuvent être écrits dans n'importe quel langage pour lequel une bibliothèque cliente est disponible (C++, python, MATLAB, Java, etc.)
- Léger : Les bibliothèques autonomes sont entourées d'une fine couche ROS
- Gratuit et open source : la plupart des logiciels ROS sont ouverts et gratuits.

3.2. Transformation des données

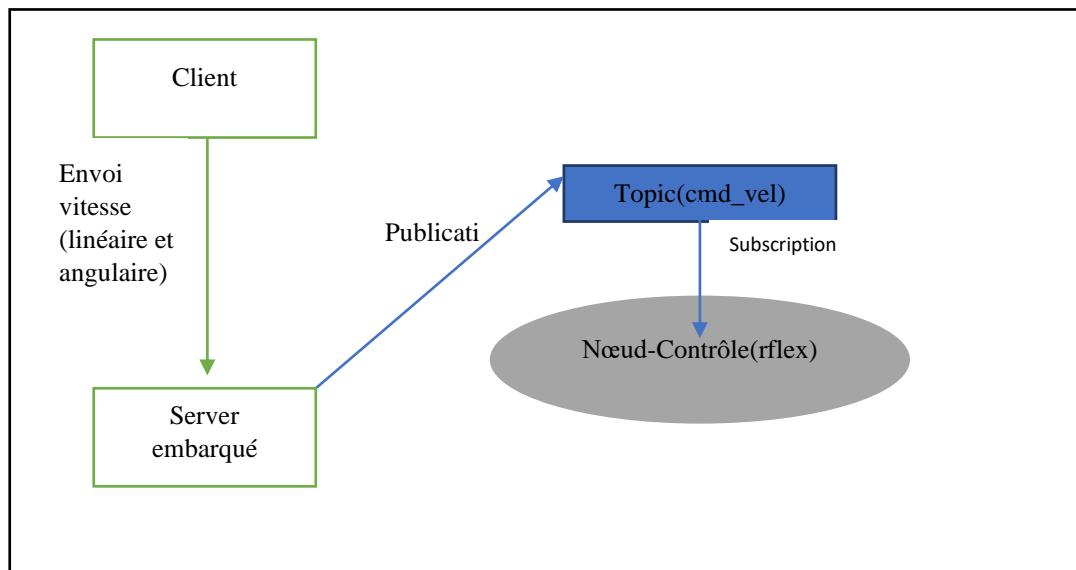


Figure 22: transformation des données entre les nœuds

- ✓ Le client envoie des données (vitesse angulaire et vitesse linéaire) au serveur embarqué.
- ✓ Le serveur récupère les données envoyées par le client puis les publie dans le topic (cmd_vel).
- ✓ Le nœud de contrôle (reflex) fait la suscription des données, et à partir de ces données il gère le robot.
- **Topic** : est un mode de communication asynchrone permettant une communication many-to-many.

4. Présentation de l'application

4.1. Interface authentification

Au lancement de notre application, un formulaire s'affiche à l'écran, il demande d'introduire le pseudonyme et le mot de passe d'authentification pour accéder au menu principal.

L'utilisateur saisit son login et mot de passe, ensuite il doit cliquer sur le bouton connexion (Voir figure23).

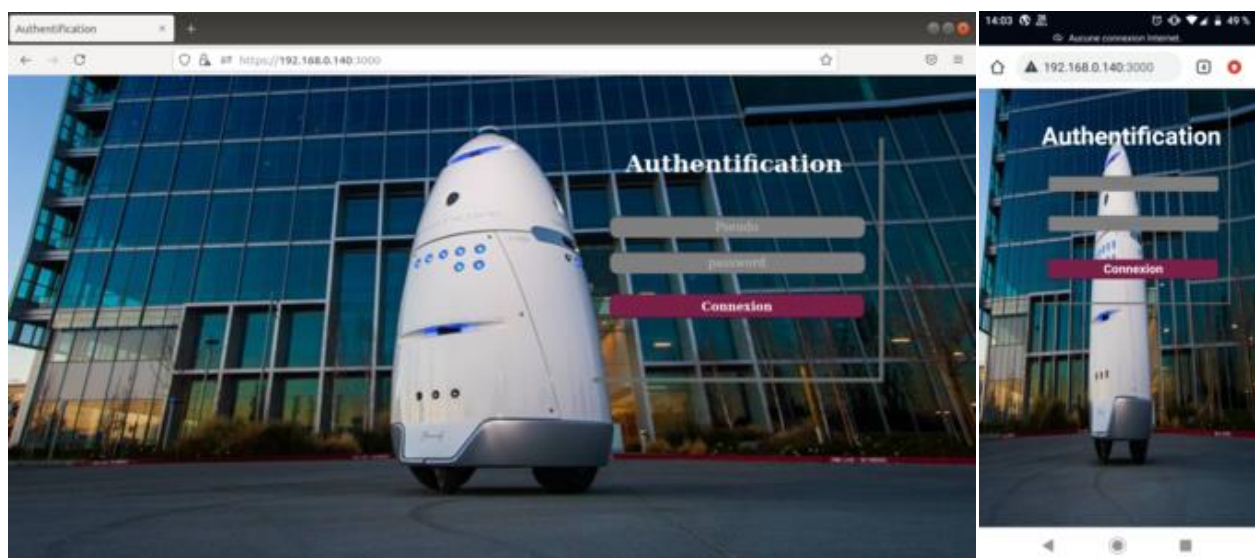


Figure 23: Interface « d'authentification ».

4.2.Interface Accueil

Cette interface comporte le menu principal où l'utilisateur pourra sélectionner la tâche à effectuer (Voir Figure 24). Ce menu contient cinq (05) boutons qui sont :

- Gestion profil : Pour les informations du profil de l'utilisateur.
- Manipulation du robot : Pour se connecter et contrôler le robot.
- Gestion des comptes : Pour afficher la liste des utilisateurs, modifier, ajouter, supprimer ; ce bouton ne s'affiche qu'à l'administrateur.
- Surveillance autonome : fonctionnalité qui sera implémenter dans la prochaine version de l'application.
- Déconnexion : pour déconnecter l'application et revenir à la page d'authentification.

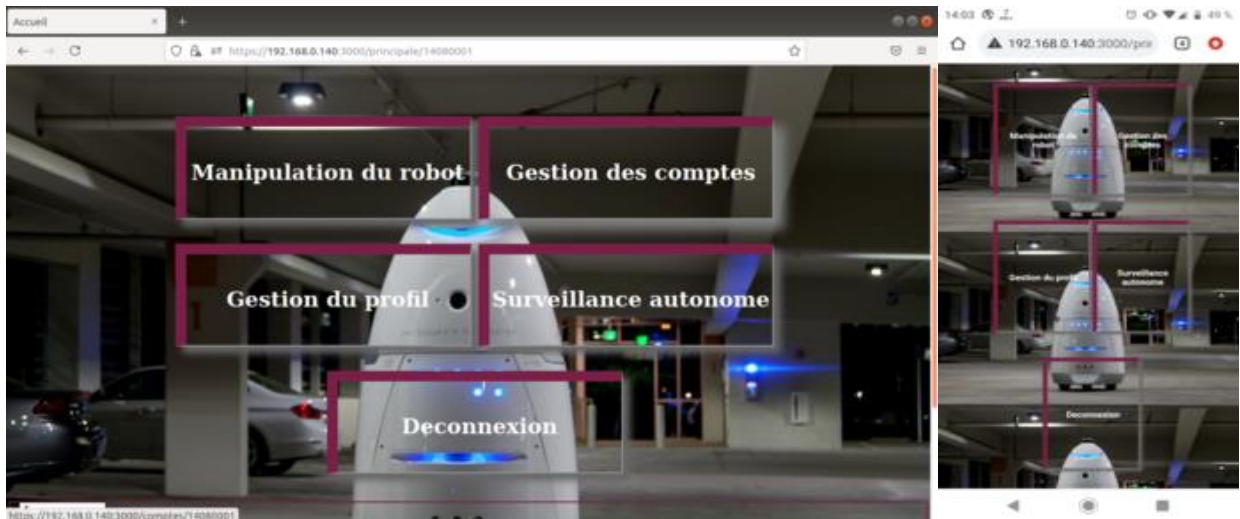


Figure 24:Interface « Accueil ».

4.3. Interface gestion des comptes

Les figures ci-dessous montrent l'interface de gestion des comptes de l'administrateur (Voir figure 25).

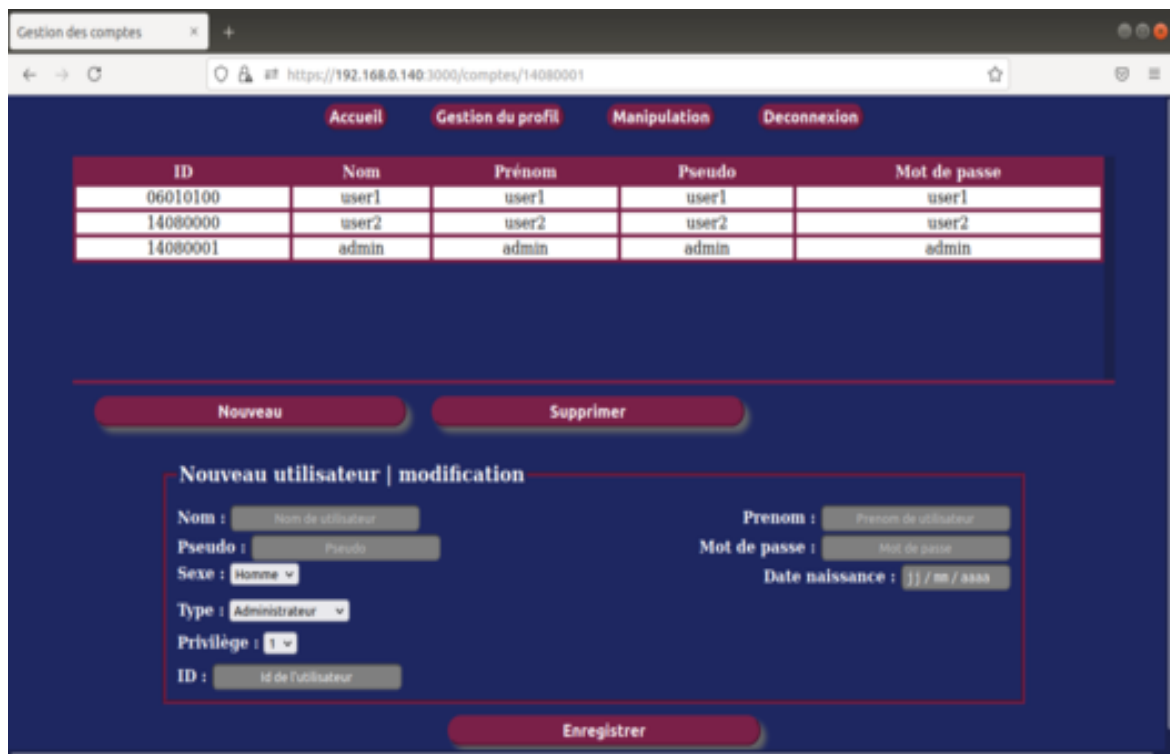


Figure 25:Interface « Gestion des comptes ».

4.4. Interface gestion du profil

Les figures ci-dessous montrent l'interface de gestion du profil de l'utilisateur (Voir figure 26).

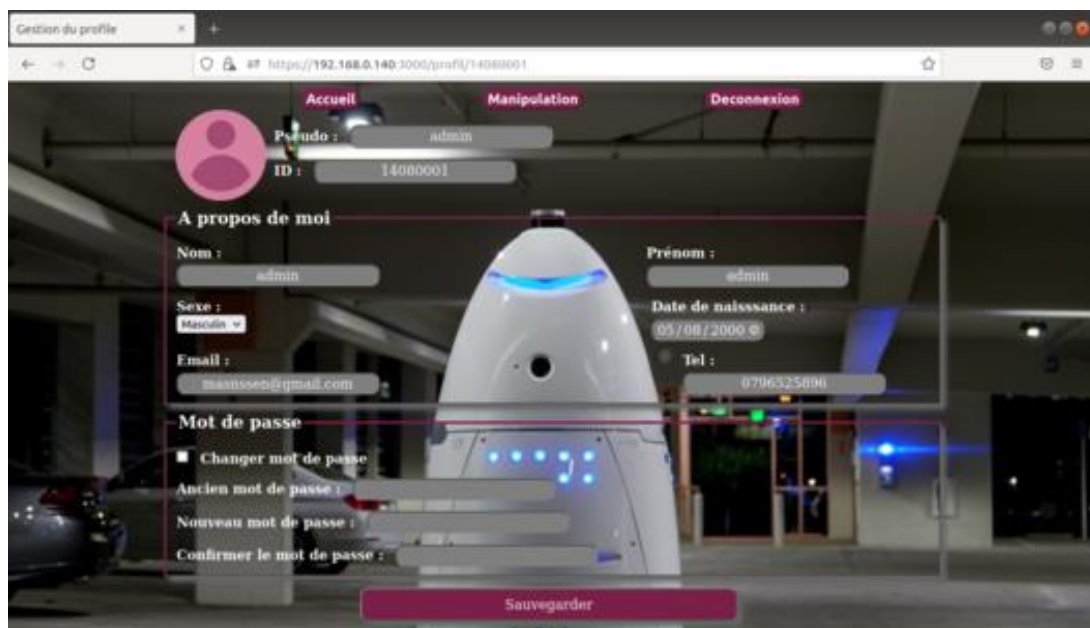


Figure 26: Interface « Gestion profil ».

4.5. Interface manipulation robot

Les figures ci-dessous montrent l'interface de manipulation du robot lorsque le robot n'est pas connecté (Voir figure 27).

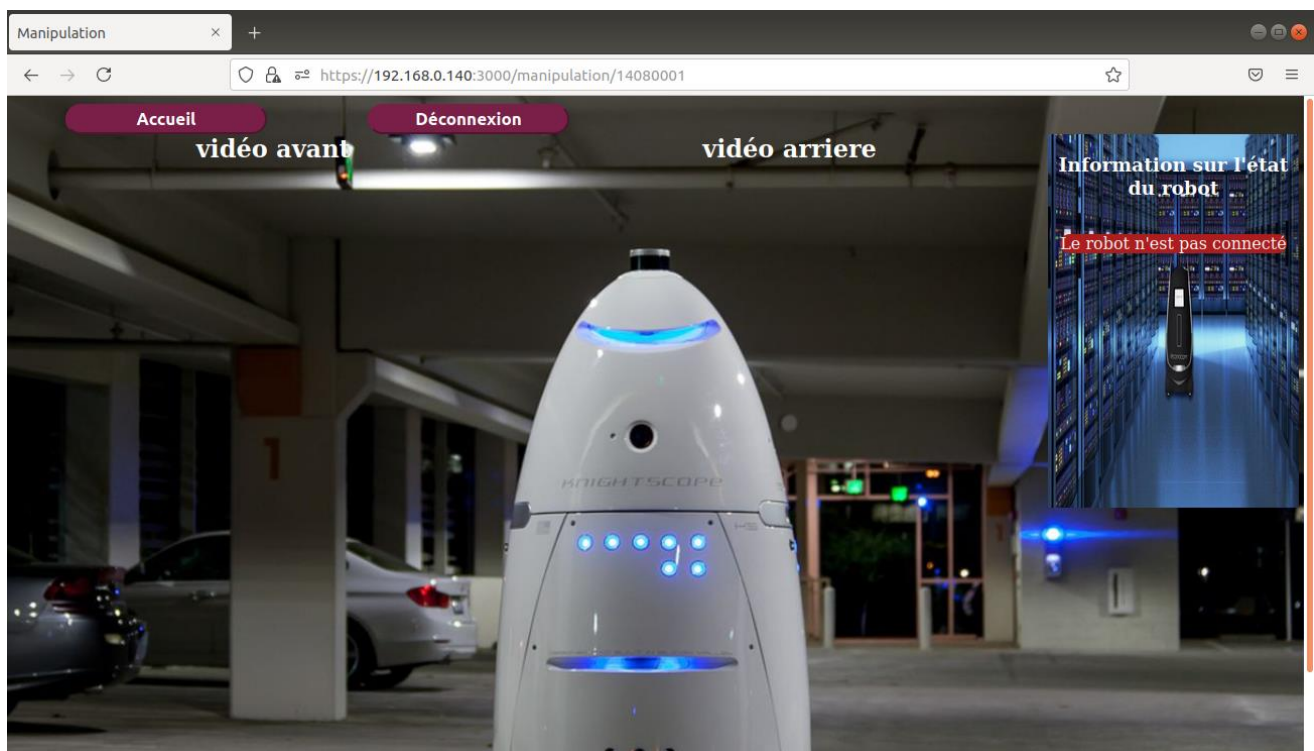


Figure 27: Interface « Manipulation robot (robot pas connecter) ».

Les figures ci-dessous montrent l'interface de manipulation du robot lorsque est connecté et l'utilisateur n'est pas connecter au robot. (Voir figure 28).

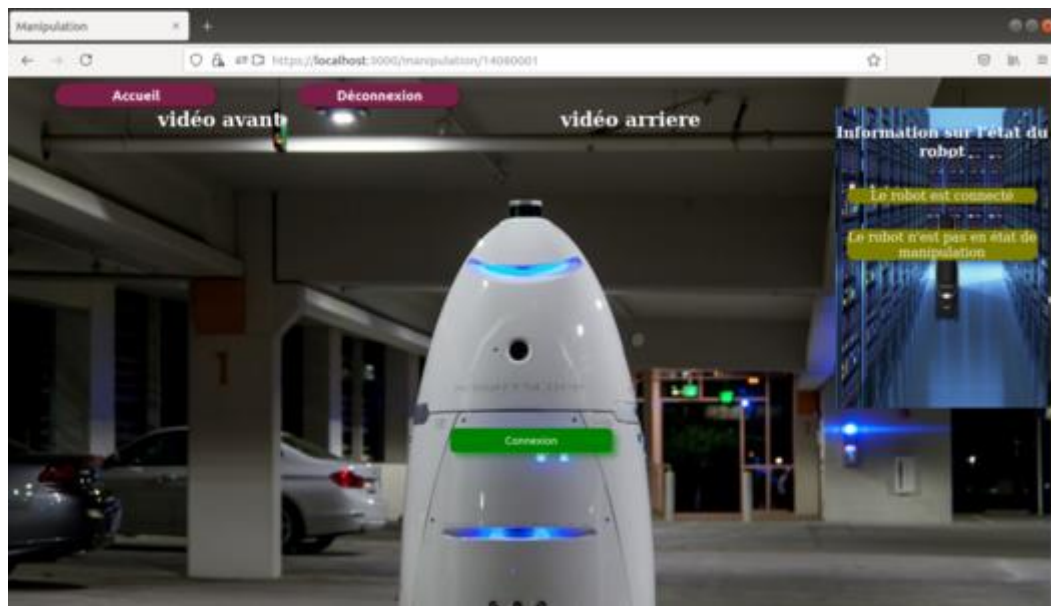


Figure 28:Interface « Manipulation robot (robot connecte et l'utilisateur n'est pas connecter) ».

La figure ci-dessous montrent l'interface de manipulation du robot lorsque est connecté, l'utilisateur est connecté au robot et il ne manipule pas le robot. Nous visualisons la vidéo avant et arrière du robot (Voir figure 29).

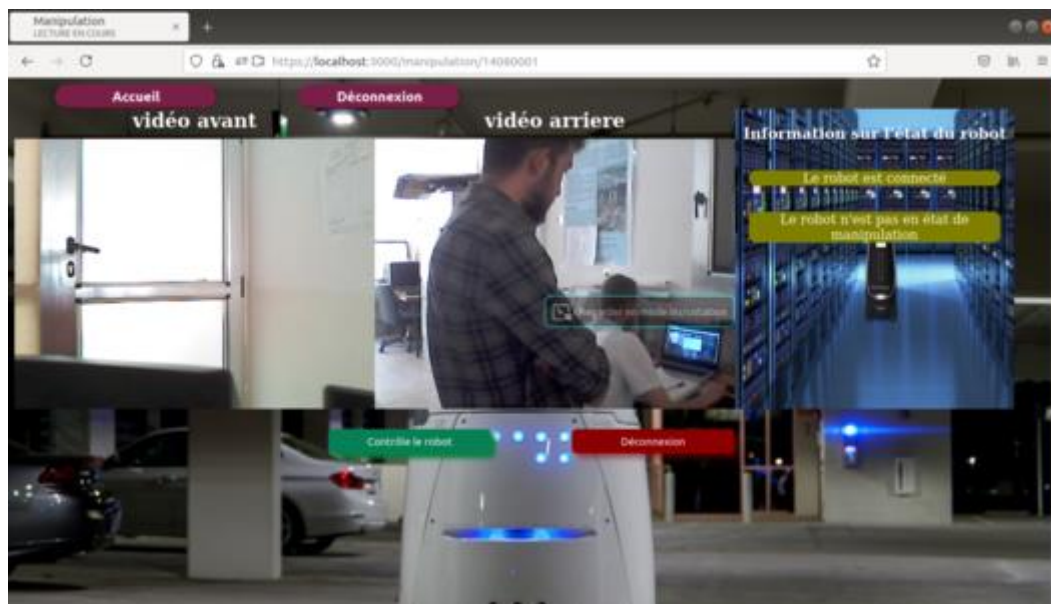


Figure 29:Interface « Manipulation robot (robot connecte et l'utilisateur est connecter) ».

Les figures ci-dessous montrent l'interface de manipulation du robot lorsque est connecté, l'utilisateur est connecté au robot et il manipule le robot. Nous visualisons dans la figure la vidéo avant et arrière du robot et les options que on peut faire quand nous sommes entrain de manipuler le robot qui sont :

- Déplacer le robot avec les flèches avant, arrière, droit, gauche.
- Enregistrement d'un itinéraire.
- Lancement d'un itinéraire.
- Suppression d'un itinéraire.
- Le contrôle de la vitesse du robot.
- Mode prudence qui est rendre le déplacement du robot pas à pas.

- Fin de contrôle.
- Déconnexion pour déconnecter l'application et revenir à la page d'authentification



Figure 30:Interface « Manipulation robot (robot connecté, utilisateur est connecter et il contrôle le robot) ».

5.Test

Dans ce test nous effectuerons la communication entre le PC embarqué (Robot B21r de CDTA) et PC à distance, en envoyant des vitesses (linéaire et angulaire) au moteurs et les flux de vidéo. Une fois que les données sont réceptionnées par le robot, il démarre d'une position initiale (voir figure 26) jusqu'à la position finale (voir figure 27).



Figure 31:Position finale de Robot B21r Figure 32: Position initial de Robot B21r

6. Conclusion

Nous avons présenté dans ce chapitre les détails techniques liés à la mise en œuvre de notre système. Nous avons commencé par présenter le robot mobile B21r, ainsi que les différents langages et technologies logicielles utilisées. Nous avons ainsi utilisé Node.js et ROS pour la partie serveur, les websockets pour la communication entre les navigateurs et les serveurs, et enfin PeerJs pour la communication peer-to-peer entre les navigateurs, nous avons présenté également les différentes interfaces et fonctionnalités de notre application. La fin du chapitre a été consacrée à la présentation des différents tests effectués en temps réel.

Conclusion générale

Au cours de ce mémoire, nous avons présenté les différentes étapes de la conception et la réalisation de notre application multiutilisateur de commande à distance pour un robot de surveillance.

L'élaboration de ce projet nous a permis, d'une part, d'approfondir nos connaissances pratiques et théoriques acquises durant nos trois années d'études et d'autre part, d'apprendre plein de nouveaux concepts tels que les fondamentaux de la robotique, ROS, la programmation JavaScript, le balisage Html, et les différents concepts de la programmation Web.

Ainsi, nous arrivons à la fin de ce rapport qui résume des mois de travail durant lesquelles nous avons réussi à mettre au point notre plateforme web et à réaliser la conception de celle-ci. Nous espérons qu'elle sera enrichissante pour tous ceux qui la consulteront.

Ce projet a été une expérience intéressante, qui nous a permis d'améliorer nos connaissances et nos compétences dans le domaine du recueil, analyse des besoins et de la programmation et dans le domaine de la robotique.

Bien que notre application ne soit qu'à sa première version, elle est néanmoins totalement fonctionnelle et offre un certain nombre de fonctionnalités intéressantes telles que :

- La gestion des utilisateurs de l'application ;
- La possibilité de visualiser le flux vidéo et audio des caméras du robots en temps réel par plusieurs utilisateurs ;
- La manipulation du robot en temp réel par un utilisateur ;
- Gestion des privilèges des utilisateurs lors de manipulation du robot.

Une éventuelle amélioration de cette version de notre application web, peut être envisagée par le rajout des fonctionnalités suivantes :

- Mettre en place un dispositif de commande vocal.
- Accès à l'application par empreinte digitale.
- Reconnaissance faciale et détection d'objets.
- Détection d'intrusion humaines.

Bibliographie

- [1] Development of a mobile platform for security robot. Proceedings of IEEE International Conference on Automation and Logistics, pp. 1262–1267, ISBN: 978-1-4244-1531-1, August 2007.
- [2] Coalition formation : From software agents to robots. Journal of Intelligent and Robotic Systems, Vol. 50, No. 1, pp. 85–118, ISSN: 0921-0296.
- [3] From Laboratory to Warehouse : Security Robots Meet the Real World, International Journal of Robotics Research, Special Issue on Field and Service Robotics, Vol. 18, No. 7, pp 760–768.
- [4] An industrial application of behavior-oriented robotics, Proceedings of IEEE International Conference on Robotics and Automation, Vol. 1, pp. 749–754, ISBN : 0-7803-6578-1, Seoul, Korea, May 2001.
- [5] GPS-Based Localization for a Surveillance UGV in Outdoor Areas, Proceedings of the Fifth International Workshop on Robot Motion and Control (RoMoCo'05), pp. 157–162, ISBN: 83-7143-266-6, Dymaczewo, Poland, June 2005.
- [6] Robotic Security Guard - Autonomous Surveillance and Remote Perception, Proceedings of IEEE International Workshop on Safety, Security, and Rescue Robotics, Bonn, Germany, May 2004.
- [7] A human-like semi-autonomous Mobile Security Robot, 2nd International Conference on Autonomous Robots and Agents, ISBN : 0-476-00994-4, Palmerston North, New Zealand, December 2004.
- [8]<https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-casutilisation> .
- [9]http://docwiki.embarcadero.com/RADStudio/Tokyo/fr/D%C3%A9finition_des_diagrammes_de_cas_d%27utilisation_UML_1.5.
- [10]http://docwiki.embarcadero.com/RADStudio/Tokyo/fr/D%C3%A9finition_des_diagrammes_d%27activit%C3%A9s_UML_2.0.
- [11]http://docwiki.embarcadero.com/RADStudio/Sydney/fr/D%C3%A9finition_des_diagrammes_de_classes_UML_1.5.
- [12]http://docwiki.embarcadero.com/RADStudio/Tokyo/fr/D%C3%A9finition_du_diagramme_de_machine_d%27%C3%A9tat_UML_2.0.
- [13]http://docwiki.embarcadero.com/RADStudio/Sydney/fr/D%C3%A9finition_des_diagrammes_de_composants_UML_2.0.
- [14]http://docwiki.embarcadero.com/RADStudio/Sydney/fr/D%C3%A9finition_des_diagrammes_de_d%C3%A9ploiement_UML_1.5.
- [15]<https://developer.mozilla.org/fr/docs/Web/CSS>.
- [16] https://developer.mozilla.org/enUS/docs/Learn/Getting_started_with_the_web/JavaScript_basics.
- [17] www.ros.org/.