

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

Faculté des Mathématiques et d'Informatique



MEMOIRE

De fin d'étude pour l'obtention du diplôme de
Master
En Informatique

Spécialité : Réseaux & Multimédias

THEME

**Conception et réalisation d'un système de Vidéosurveillance
automatique**

Réalisé par : **BELKAALOUL Abdallah**

Soutenu publiquement le : **06/07/2019**

Président :

Dr BENSEFIA Hassina Maitre de conférences à l'université de B.B.A

Examineurs

Mr BELAZZOUGE Mouhoub Maitre-assistant à l'université de B.B.A

Mlle BELALTA Ramla Maitre-assistante à l'université de B.B.A

Encadreur

Mr MOUSSAOUI Boubakeur Maitre-assistant à l'université de B.B.A

Promotion : 2018/2019

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

وَقُلْ رَبِّ زِدْنِيْ عِلْمًا

صَدَقَ اللّٰهُ الْعَظِیْمُ

Remerciement

Je remercie en premier lieu ALLAH de m'avoir donné non seulement le courage mais aussi la force et la volonté nécessaire pour la réalisation de ce travail.

Mes vifs remerciements s'adressent à mes parents, mes frères et mes sœurs.

Je tiens à exprimer mes profondes gratitude et mes sincères remerciements à mon encadreur **Mr. Moussaoui Boubakeur** pour la haute qualité de son encadrement, son suivi, sa disponibilité et ses conseils. Encore une fois, merci beaucoup.

Mes vifs remerciements vont également aux membres de jury pour l'intérêt qu'ils ont porté à mon sujet en acceptant d'examiner mon travail et de l'enrichir par leurs propositions.

Je profite aussi cette occasion pour remercier vivement tous mes enseignants, de l'école primaire à l'université, pour leurs efforts épargnés afin d'arriver à cette réussite.

J'adresse mes sincères remerciements à toute personne qui a contribué de près ou de loin à la réalisation de ce projet

Merci

Dédicaces

A l'aide de DIEU tout puissant, qui trace le chemin de ma vie, j'ai pu arriver à réaliser ce modeste travail que je dédie :

A la mémoire de ma sœur Nour el Houda, qui était et restera toujours dans mon esprit et dans mon cœur, que le miséricordieux l'accueille dans son éternel paradis.

A mes chers parents, qui n'ont cessé de m'encourager et de sacrifier pour je puisse franchir tout obstacle durant toute mes années d'étude que le dieu me les garde en très bonne santé.

A mes frères et sœurs pour leur soutien moral et leur encouragement

A ma cousine Amal qui m'a aidé à réaliser ce travail.

A mes chers amis qui sont toujours présent et fidèles.

A tous mes enseignants.

Que dieu les protèges tous...

Table des matières

INTRODUCTION GENERALE	6
Chapitre 01 : Concepts de base	8
1.1 Introduction.....	8
1.2 Les applications mobiles.....	8
1.2.1 Définition	8
1.2.2 L'apport des applications mobiles	8
1.2.3 Les limites	10
1.2.4 Classification des applications mobiles	10
1.2.5 Les étapes de création d'une application mobile	14
1.2.6 Les systèmes d'exploitation mobiles	16
1.2.7 Le marché des applications mobiles.....	18
1.3 La Vidéosurveillance	19
1.3.1 Définition	19
1.3.2 L'apparition de la vidéosurveillance.....	19
1.3.3 Les buts de la vidéosurveillance	20
1.3.4 Domaines d'application	20
1.3.5 Types des systèmes de la vidéosurveillance	20
1.4 Conclusion	22
Chapitre 02 : La mise en place du microcontrôleur	23
2.1 Introduction.....	23
2.2 Définition du microcontrôleur	23
2.3 Présentation des différents microcontrôleurs.....	23
2.4 Choix du Raspberry Pi.....	26
2.5 Spécification matériels du Raspberry pi	26
2.6 Spécification Logiciels du Raspberry pi.....	29
2.6.1 Le système d'exploitation	29
2.6.2 Logiciels compatibles	30
2.7 Démarrage du Raspberry Pi.....	30
2.8 Connexion à Distance	31
2.9 Les différents utilisations du Raspberry Pi.....	33
2.9.1 Les serveurs	34
2.9.2 La domotique	34
2.9.3 Les systèmes embarqués	34
2.9.4 Les ordinateurs.....	35
2.10 Conclusion	35

Chapitre 03 : Analyse et conception	36
3.1 Introduction.....	36
3.2 Analyse des objectifs	36
3.2.1 Objectif du Travail.....	36
3.2.2 Public visé.....	36
3.2.3 Le contenu.....	36
3.3 Spécification des besoins	36
3.3.1 Les acteurs.....	36
3.3.2 Besoins fonctionnel	37
3.3.3 Besoins non-fonctionnels	37
3.4 Planification.....	38
3.4.1 Architecture	38
3.4.2 Cas d'utilisations.....	38
3.4.3 Diagramme de cas d'utilisation	39
3.4.4 Diagramme de classes.....	43
3.4.5 Diagramme d'activités.....	44
3.5 Conclusion	46
Chapitre 04 : Réalisation	47
4.1 Introduction.....	47
4.2 Système proposé	47
4.3 Les composants principaux de la solution	48
4.4 Partie Matériels.....	48
4.4.1 Le choix du Raspberry Pi.....	48
4.4.2 Raspberry pi comme serveur web.....	49
4.4.3 Collection des données des capteurs et des caméras	49
4.4.4 Présentation du python.....	49
4.4.5 Configuration des capteurs avec le Raspberry Pi	50
4.5 Protocoles de communication.....	53
4.5.1 Le protocole http.....	53
4.5.2 Le protocole MQTT.....	54
4.6 Partie Mobile	58
4.6.1 Application Mobile	58
4.6.2 Le choix d'Android.....	58
4.6.3 Android Studio.....	59
4.6.4 Présentation de l'application.....	61
4.7 Mesures de performances.....	58
4.7 Conclusion	66
Conclusion générale	67

Liste des figures

Figure 1.1	: Nombre d'utilisateurs des smartphones.....	10
Figure 1.2	: La part des applications disponibles par catégorie.....	12
Figure 1.3	: Part des ventes des systèmes d'exploitation mobile aux utilisateurs finaux.....	18
Figure 1.4	: les revenus projetés des applications mobiles de 2015 à 2020.....	19
Figure 1.5	: Système analogique.....	21
Figure 1.6	: système sur réseaux IP.....	22
Figure 1.7	: Système hybride.....	22
Figure 2.1	: Carte Raspberry pi.....	23
Figure 2.2	: Carte Arduino	24
Figure 2.3	: Carte STM32.....	24
Figure 2.4	: Carte Beaglebone black.....	24
Figure 2.5	: Bloc d'alimentation du Raspberry Pi.....	27
Figure 2.6	: les Ports GPIO.....	28
Figure 2.7	: La Pi Camera.....	28
Figure 2.8	: Boitier pour Raspberry Pi.....	29
Figure 2.9	: les différents composants du Raspberry Pi.....	29
Figure 2.10	: le slogan officiel du Raspbian.....	30
Figure 2.11	: La table de configuration du Raspberry pi.....	31
Figure 2.12	: outil de connexion à distance.....	33
Figure 2.13	: écran du démarrage du Raspberry Pi	33
Figure 3.1	: Architecture du système.....	38
Figure 3.2	: Diagramme de cas d'utilisations.....	40
Figure 3.3	: Diagramme de classes.....	43
Figure 3.4	: Diagramme d'activité authentification.....	44
Figure 3.5	: Diagramme d'activité consulter les cameras.....	45
Figure 3.6	: Diagramme d'activité publier les données.....	46
Figure 4.1	: système à être réalisé.....	47
Figure 4.2	: schéma de d'une requête http.....	54
Figure 4.3	: Niveaux de Qualité de service MQTT.....	56
Figure 4.4	: Architecture d'un projet en Android studio.....	60
Figure 4.5	: Authentification.....	62
Figure 4.6	: Créer nouvel utilisateur.....	63
Figure 4.7	: page principale.....	63
Figure 4.8	: Système de suivi.....	64
Figure 4.9	: Camera.....	64
Figure 4.10	: Pourcentage des mouvements détectés par rapport aux endroits.....	65

Liste des tableaux

Tableau 01 : Tableau de comparaison technique entre les cartes électroniques.....	25
Tableau 02 : Cas d'utilisation consulter les données.....	41
Tableau 03 : Cas d'utilisation acquisition des données.....	41
Tableau 04 : Cas d'utilisation gérer les comptes.....	42
Tableau 05 : tableau démonstratif de diagramme de classes.....	44
Tableau 06 : Les composants principaux de la solution.....	48

Liste des Abréviations :

UML	Unified Modeling Language
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
API	Application programming interface
iOS	iPhone operating system
SDK	Software Development Kit
OS	Operating system
PDA	Personal digital assistant
IP	Internet protocol
GPIO	General Purpose Input/Output
CMOS	Complementary Metal Oxide Semiconductor
SSH	Secure SHel
VNC	Virtual Network Computing
SGBD	Système de gestion de base de données
MQTT	Message Queuing Telemetry Transport
HTTP	Hypertext Transfer Protoco
XMPP	Extensible Messaging and Presence Protocol
XML	Extensible Markup Language

Introduction Générale

Aujourd'hui, le téléphone mobile n'est plus seulement utilisé pour la communication et l'échange des messages courts, de nouveaux usages sont apparus tel que la messagerie électronique, la navigation sur Internet, la musique, la vidéo, etc. Grâce à la généralisation des smartphones ainsi qu'au développement des logiciels et des réseaux, les applications mobiles sont capables de satisfaire un large éventail des besoins notamment dans les domaines de la sécurité et du confort des usagers.

Le besoin de renforcer le niveau de sécurité se fait de plus en plus ressentir dans les différents domaines d'activités, la plus part des propriétaires souhaitent accroître la sécurité en protégeant les vies et les biens, un des moyens très utilisés est la **Vidéosurveillance**. La **Vidéosurveillance** consiste à surveiller des lieux publics ou privés, à l'aide des caméras qui transmettent des images et des vidéos à un équipement d'acquisition pour les traiter et les afficher sur un écran ou un PDA. Comment donc profiter de toutes ces technologies (Smartphone, caméra IP, Internet,...) afin de surveiller à distance un lieu sensible ?

L'objectif de ce travail est de développer une application mobile sous Android afin de sécuriser un endroit bien précis, ce dernier est équipé d'un système de vidéosurveillance à la base d'un capteur de mouvement et d'une caméra reliés à un équipement intelligent pour le stockage des informations, le traitement des données acquises et l'envoi des notifications en cas de danger ressenti. Notre application permet de faciliter la vie et de faire éliminer plusieurs difficultés, nous adoptons la technologie de **Vidéosurveillance** et des **applications mobiles** pour faire communiquer avec du matériel (capteurs de mouvement, des Pi cameras, et des microcontrôleurs...).

Notre projet tourne sur deux volets, le premier consiste à construire et à réaliser un système qui permet de détecter tous mouvement dans la zone d'intérêt, dont le second consiste à concevoir et réaliser une application mobile qui permet de consulter les cameras en ligne après la réception des notifications en cas de détection de mouvement. Le propriétaire a la possibilité de poursuivre la vidéo diffusée en ligne, même sans notification (le cas normal).

Ce mémoire est organisé en quatre chapitres :

Le premier chapitre s'intitule «Concepts de base », il a pour l'objectif d'introduire les principaux concepts de notre système, tel que les applications mobiles et la vidéosurveillance.

Le deuxième chapitre est dédié à la description du matériel utilisé, on a donné une description plus approfondie sur le microcontrôleur « Raspberry pi » vue son importance dans le système.

Le troisième chapitre concerne « l'analyse et la conception » du projet, Dans ce chapitre nous avons défini les besoin des utilisateurs. Pour la conception du système, le langage de modélisation UML est utilisé pour cette fin.

Dans le dernier chapitre, intitulé « Réalisation », on a exposé les outils et l'environnement de travail en premier lieu, puis on a parlé sur le fonctionnement du système ainsi que les activités de notre application mobile.

Chapitre I

Concepts de base

1.1 Introduction

Les applications mobiles ont pris une place importante dans notre vie quotidienne, et prennent de plus en plus d'espace dans l'utilisation de nos terminaux mobiles, ces applications ont révolutionnées nos usages à l'internet, le commerce, grâce à cette révolution tous les domaines sont concernés par les contraintes de ce développement notamment le domaine de **la vidéosurveillance**.

Dans ce chapitre nous allons présenter les notions de bases des applications mobiles et la vidéosurveillance afin de pouvoir connaître ses secrets et son importance dans notre vie.

1.2 Les applications mobiles :

1.2.1 Définition :

Une application mobile est un logiciel applicatif développé pour s'exécuter sur un appareil électronique mobile¹, tel qu'un assistant personnel, un téléphone portable, un smartphone, une tablette tactile, ces applications mobiles servent à fournir aux utilisateurs des services similaires à ceux du PC(1).

Une application, comme son nom l'indique, remplit une (des) fonction(s), une utilité précise qui peut être très variée. La spécificité de ces applications réside dans la possibilité de s'exécuter dans des endroits différents au fur et à mesure qu'on se déplace.

1.2.2 L'apport des applications mobiles :

Étant donné la progression du nombre d'utilisateurs de smartphones, il est devenu presque naturel pour une entreprise de chercher des solutions mobiles compatibles avec les centres d'intérêt des mobinautes, mais aussi avec leurs nouvelles habitudes.

Ainsi, utiliser les nouvelles technologies mobiles, spécialement les applications, pour apporter une proposition de valeur différente du marché est devenue la priorité d'un bon nombre d'entreprises afin de se démarquer et attirer plus de clients.

¹ Un smartphone est un téléphone mobile disposant en général d'un écran tactile, d'un appareil photographique numérique, des fonctions d'un assistant numérique personnel et de certaines fonctions d'un ordinateur portable

Sur le plan ergonomique et graphique, une application mobile permet une navigation fluide et un usage simple tout en proposant des fonctionnalités diverses comme les alertes ou encore l'utilisation de la pleine capacité du téléphone (GPS, boussole, appareil photo, contacts, agenda, etc..)

Les avantages tirés du choix d'une application mobile peuvent être résumés comme suit :

- (a) L'accessibilité : Les applications mobiles sont facilement accessibles pour la clientèle car elle est stockée directement sur son téléphone. C'est ainsi une forme d'engagement envers votre marque.
- (b) La variété des langages de programmation : Les applications mobiles peuvent être créées avec une grande variété de langages de programmation, une application peut être développée par plusieurs langages de programmation.
- (c) Profiter des technologies existantes : Les applications mobiles peuvent tirer profit de la présence du GPS, de la caméra, et de toute autre fonctionnalité embarquée sur les smartphones pour une grande interactivité.
- (d) L'instantanéité des notifications : Les applications mobiles peuvent envoyer ou recevoir les notifications, ces notifications seront consultées par le mobile user à l'instant même quel que soit son emplacement géographique, en plus l'envoi de ces notifications est fait sans l'intervention de l'être humain dans quelques applications comme c'est le cas de notre application.

Le diagramme de la figure 03 nous souligne l'importance des smartphones à travers le nombre d'utilisateurs dans le monde de 2015 et 2020.

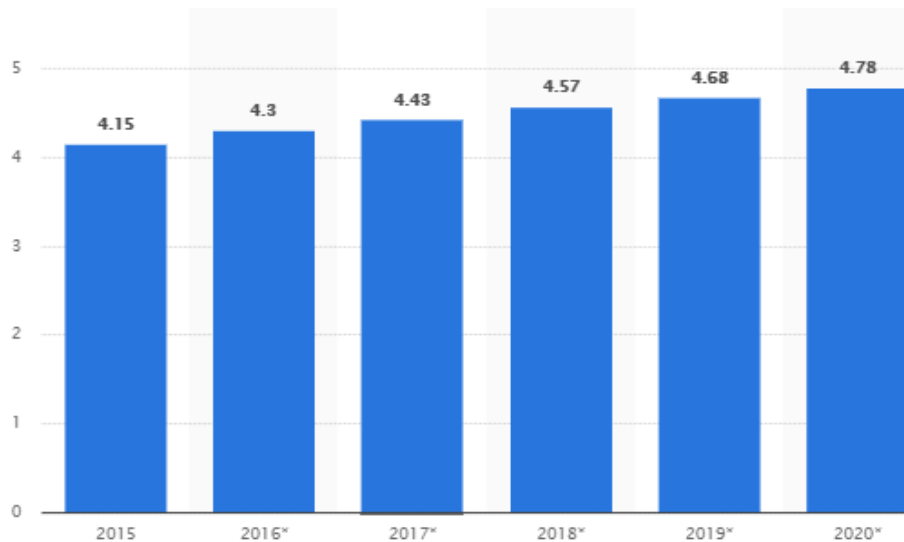


Figure 1.1 : Nombre d'utilisateurs des smartphones dans le monde de 2015 à 2020(en milliard)

1.2.3 Les limites :

- (a) Les applications mobiles sont spécifiques à chaque plate-forme, c'est-à-dire qu'une application mobile dédiée pour iOS (iPhone operating System) ne fonctionnera pas sur Android et vice-versa. Il faudra donc spécifier le système d'exploitation mobile avant le développement de ces applications
- (b) Les applications mobiles nécessitent une certaine forme de promotion car elles ne seront disponibles que sur des magasins d'applications spécifiques à chaque système (Apple Store pour iOS ²ou Google Play³ pour Android).
- (c) Les smartphones évoluent assez rapidement ce qui implique que très peu de standard existent pour les développeurs, notamment au niveau de la taille de l'écran, la résolution, la présence ou non de clavier, etc.

1.2.4 Classification :

Pour bien comprendre l'usage et l'utilité des applications mobiles il faut les classer selon des catégories, selon le type et la complexité.

² L'App Store est un magasin d'applications distribué par Apple sur les appareils mobiles fonctionnant sous iOS.

³ Google Play est un magasin d'applications distribué par Google sur les appareils mobiles fonctionnant sous Android.

1.2.4.1 Selon le type :

Les applications mobiles se regroupent en trois types :

1. Application native : Il s'agit d'une application conçue pour une grande partie de systèmes d'exploitation fiables par les smartphones en se référant à un langage particulier à chacun d'eux.

Ce mode d'application est accessible seulement sur les plateformes d'applications suivent ses particularités et ses formules. Ces plateformes retirent 25% du prix de vente pour une application native payante.

Le souci des applications natives est que les utilisateurs doivent avoir un système d'exploitation mobile donné pour qu'ils puissent les utiliser. Pour assurer un usage plus exponentiel de ces applications mobiles, il faut penser à lancer la même application mobile compatible à tout système d'exploitation mobile.

2. Application web : Toute application conçue avec HTML et CSS de plus opérationnelle sur navigateur internet pour un smartphone appelées application web.

Peu importe la marque du smartphone, on peut accéder à l'application web par le biais de son navigateur et donc on n'a pas besoin de la télécharger. Vu qu'elle ne tient pas en compte les divergences persistantes entre les systèmes d'exploitation et les marques de smartphone, l'application web manque d'ergonomie et de plus elle ne se sert pas de la mémoire du smartphone ce qui la place en infériorité par rapport à l'application native.

3. Application hybride : Il s'agit d'une application mobile qui fusionne entre les caractéristiques de web application et celles de l'application native. De cette manière, l'application mobile sera accessible sur toutes les plateformes d'application(5).

Ce type d'application mobile minimise les charges et la durée de son développement même si cela sera au détriment de perfectionnement et de la qualité qui caractérise l'application native.

Notons que les applications hybrides sont accessibles exclusivement sur iPhone et Android.

1.2.4.2 Selon la catégorie :

Les applications mobiles sont regroupées en plusieurs catégories et parmi ses catégories on trouve (1) les jeux, (2) l'éducation, (3) styles de vie....

Le diagramme de la figure 1.2 nous montre la part des applications mobiles active en 2018 selon les catégories.

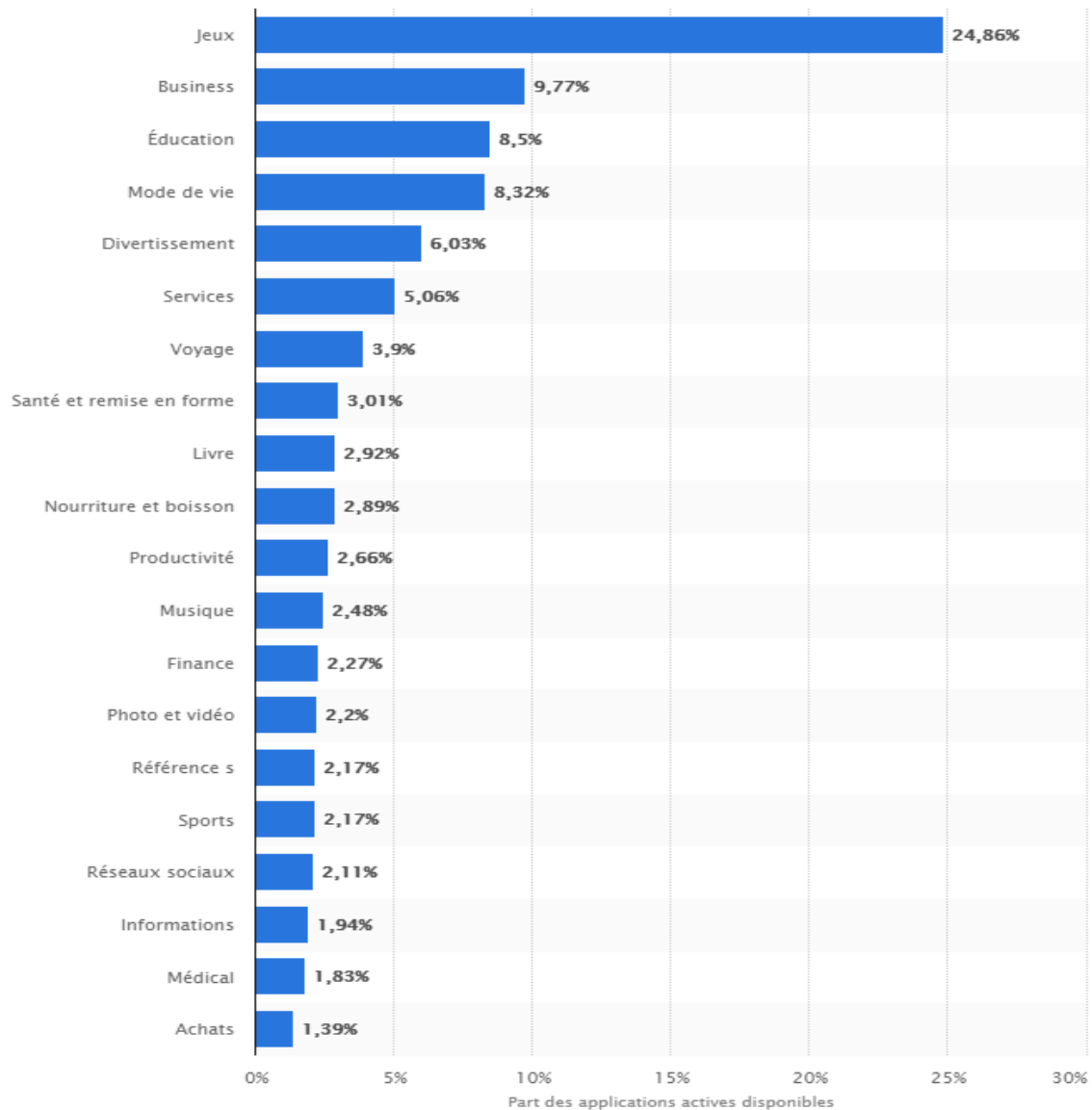


Figure 1.2 : La part des applications disponibles par catégorie

Il est clair dans la figure 1.2 que les jeux dominent le monde des applications mobiles avec un taux de téléchargement de 24.86%, notre objectif est développer une application mobile pour le confort des Johns (surveiller leur maisons à distance) et pour les industries (surveiller une

usine à distance), on remarque que cette catégorie représente un taux de téléchargement aussi important.

1.2.4.3 Classification selon la complexité :

Selon la complexité du produit, une application Mobile peut posséder des caractéristiques et des éléments différents.

1. Des applications simples : Les applications simples exécutent des fonctions de base et il n'est pas nécessaire d'intégrer des API⁴ tierces. En outre, une application simple peut gérer sans développement backend.

Les applications avec des fonctionnalités de base peuvent généralement inclure des composants d'interface utilisateur standard, des connexions sociales, des notifications push, un profil personnel, une liste de flux et des marqueurs de carte.

2. Applications de complexité moyenne : Les applications de cette catégorie peuvent nécessiter l'intégration d'une solution tierce. Par exemple, pour fournir une application avec des fonctionnalités de paiement ou des discussions en temps réel. Les applications de complexité moyenne peuvent inclure des fonctionnalités d'interface utilisateur personnalisées, l'intégration avec des API et un serveur backend simple.

3. Applications à complexité élevée : Ces applications possèdent des fonctionnalités avancées. Il peut s'agir du streaming audio / vidéo, des chats, de la synchronisation des données en temps réel, de la conception et de l'animation personnalisées, de l'intégration de services tiers, des achats intégrés, du backend complexe, etc.

Pour couper court à la poursuite, estimer la dépense pour votre application, vous devez comprendre ce qui suit. La complexité du projet influence la taille de l'équipe de développement. La taille de l'équipe a un impact direct sur le temps nécessaire pour mettre en œuvre le projet, le nombre des heures passées constitue le coût final. Il est évident cependant de fournir aux clients le processus transparent et clair de l'estimation des coûts.

⁴ API : une interface de programmation d'application, est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciel

1.2.5 Les étapes de création d'une application mobile :

Pour mettre en œuvre une bonne application mobile il faut passer par les étapes suivantes :

- (a) La conception
- (b) Le prototypage
- (c) Le développement
- (d) Le Test
- (e) Les feedbacks

La conception : Une idée, aussi bonne soit-elle, n'est pas suffisante pour passer à la réalisation de votre œuvre et d'en faire un succès, vous devez avant tout consacrer une séance de brainstorming au tout début de votre projet, afin de rassembler un maximum d'informations concernant l'utilisation de votre application. Cela passe notamment par une phase d'étude des besoins de ses futurs utilisateurs.

Lors de cette étape, il faut pouvoir :

- (a) Rédiger un cahier de charges (type d'applications à développer, design, définition de la cible, des fonctionnalités de l'application).
- (b) Préparer le déploiement de la stratégie utilisée tout au long du processus de développement mobile.
- (c) Dresser le business plan et les possibilités de financement.
- (d) Penser à un plan marketing en cas de monétisation de l'application.
- (e) Fixer des objectifs à atteindre.

Cette étape est détaillée dans le chapitre 03, sauf que quelques étapes n'apparaissent pas vu la nature du produit (n'est pas commercial)

Étape de prototypage : Une fois que les premières bases de l'application ont été déterminées sur le cahier de charges lors de la phase de conception, on le temps de dessiner les différents écrans qui vont donner vie au projet, à main levée ou en utilisant un logiciel adapté. Il s'agit de MockUps ou de Wireframes.

Cette opération permettra d'avoir une vision haute niveau de l'ensemble de la navigation dans l'application.

Le développement : Si l'application est native s'adressant aux devices Apple, iPhone, iPad ou iPod touch, il faut :

- (a) Préparer le matériel adéquat pour pouvoir réussir votre développement mobile, à savoir un Mac et le SDK⁵ (Software Development Kit) iOS.
- (b) Écrire ou faire écrire les lignes de code en utilisant les langages Objective-C ou Swift sous Xcode.

Par contre, si l'application native est destinée à la plate-forme de Google, peu importe le système d'exploitation de votre machine de développement (macOS, Windows ou Linux), le développement mobile d'une application Android pour tablette ou smartphone nécessite :

- (a) L'installation d'Android Studio (Environnement de développement des applications Android) et le SDK associé.
- (b) L'apprentissage des bases de Java ou Kotlin.

Si vous êtes habitué à d'autres langages de programmation comme *JavaScript* ou que vous cherchez à gagner en temps et en coûts, les applications hybrides s'avèrent être une bonne solution pour créer des applications compatibles avec plusieurs système d'exploitation en se basant sur le même code.

Arrivé à cette phase, vous avez dû tester et finaliser votre application mobile avant de la proposer en vente sur la plate-forme de téléchargement Play Store de *Google* ou App Store d'Apple.

Le test : Cette étape n'est pas obligatoire mais je la conseille fortement. L'idée est simple, on commence par les tests de l'application par un groupe d'utilisateurs cibles avant la mise sur le marché.

Cette étape permet de prendre en compte des retours utilisateurs pour mieux adapter l'application avant son lancement officiel.

Les feedbacks : Cette étape est également très importante et pourtant souvent oubliée par les entrepreneurs « débutants ».

L'idée est simple : mettre en place un système pour avoir des retours des utilisateurs.

⁵ SDK signifie software Development kit : c'est un ensemble d'outils d'aide à la programmation pour concevoir des logiciels, jeux, applications mobiles, etc, pour un terminal et/ou un système d'exploitation spécifique

Ainsi cette étape consiste à mettre en place une veille sur les mots clés de l'application. Cela elle permettra de savoir qui en parle et donc de savoir quels sont les avis de ces utilisateurs (pour ensuite adapter l'application si nécessaire)

Note : les étapes de développement et de test sont détaillées dans le chapitre 04, sauf l'étape des feedback n'apparue pas vue que l'application n'est utilisée par un nombre d'utilisateur qui nous permet de prendre les feedback.

1.2.6 Systèmes d'exploitation mobiles :

Les systèmes d'exploitation mobiles (OS) peuvent être définis comme les logiciels permettant à un smartphone ou un téléphone mobile basique de fonctionner. Ils permettent de ce fait aux utilisateurs de pouvoir passer un appel téléphonique, naviguer sur leurs téléphones parmi toutes les rubriques, télécharger des applications ou encore paramétrer et personnaliser leurs smartphones.

On distingue une vingtaine de système d'exploitation sur le marché, certains étant développés directement par et pour les fabricants de téléphones eux-mêmes, d'autres formant des conglomérats pour mutualiser les coûts et effectuer des synergies, et d'autres souhaitant uniquement se focaliser sur le développement de systèmes d'exploitation.

Plus généralement, on peut évoquer la présence de quatre OS sur le marché des systèmes d'exploitation, et que nous vous proposons de découvrir ci-dessous, par ordre d'importance : Android (Google), iOS (Apple Inc.), Windows Phone (Microsoft) et Blackberry OS (RIM).

1.2.6.1 Caractéristique des systèmes d'exploitation mobiles :

Un système d'exploitation mobile regroupe un ensemble des fonctionnalités qui gèrent :

- (a) Les microprocesseurs et l'ordonnancement.
- (b) Les Entrées/Sorties.
- (c) La sécurité.
- (d) Les fonctionnalités multimédia.
- (e) La mémoire

1.2.6.2 Les systèmes d'exploitation mobiles disponibles :

Les principaux systèmes d'exploitations mobile sont : Android, IOS, Bada, BlackBerry OS, Symbian OS, Windows Mobile, Windows Phone 7.

Android : est un système d'exploitation open source utilisant le noyau Linux, pour smartphones, PDA et terminaux mobiles conçu par Android, une startup rachetée par Google, et annoncé officiellement le 5 novembre 2007. D'autres types d'appareils possédant ce système d'exploitation existent, par exemple des téléviseurs et des tablettes.

L'avantage de ce système d'exploitation se résume dans le fait qu'il est mis à disposition de tous les fabricants de téléphones, à condition qu'ils respectent certaines conditions liés à la mise en avant de services Google. Pour le reste, l'OS est personnalisable à souhait, par les consommateurs comme par les fabricants. De nos jours, la majorité des téléphones vendus dans le monde tournent sous Android. On peut ainsi citer de très nombreuses marques de smartphones, parmi lesquelles Samsung, Acer, HTC, LG, Sony ou encore Huawei (6).

IOS D'Apple : Anciennement connu sous le nom iPhone OS, iOS est le système d'exploitation mobile développé par Apple pour iPhone, iPod et l'iPad, il est dérivé de OS X dont il partage des fondations (le noyau hybride XNU basé sur le micronoyau Mach, les services Unix et Cocoa, etc.) iOS est le deuxième système d'exploitation le plus répandu sur le marché

Bada : est le système d'exploitation pour smartphone haut de gamme et milieu de gamme de Samsung basé sur le système d'exploitation propriétaire SHP OS. Tous les téléphones sous Bada ont un nom commençant par Samsung Wave.

BlackBerry OS : est un système d'exploitation qui fonctionne sur le smartphone BlackBerry. Il permet aux développeurs de mettre en place des applications en utilisant les APIs BlackBerry, mais toute application doit être signée numériquement par le compte RIM du développeur.

Symbian OS : est un système d'exploitation le plus utilisé pour des smartphones et PDA, qui a été conçu par Symbian Ltd. Symbian OS fournit les fonctionnalités essentielles du système d'exploitation, notamment le cœur du système, ainsi que les API communes et une interface utilisateur de référence. Il a été adopté par différents fabricants de téléphones portables de 2G et 3G (Nokia, Sony Ericsson, Motorola, Samsung, etc.). Les principales interfaces utilisateur sont S60 série (pour clavier

Windows Phone : Il s'agit du système d'exploitation développé par Microsoft, principalement utilisé sur la gamme Lumia de Nokia. Ce Système d'exploitation permet une navigation personnalisable grâce à l'utilisation de tuiles dynamiques et dispose de Skype pour les appels par internet et la vidéoconférence. Le Windows Store, boutique d'applications de l'OS mobile de Microsoft, est également de plus en plus fourni.

Le diagramme de la figure 03 nous montre la part de marché mondiale détenue par les principaux systèmes d'exploitation pour smartphones en termes de ventes aux utilisateurs finaux de 2009 à 2018.

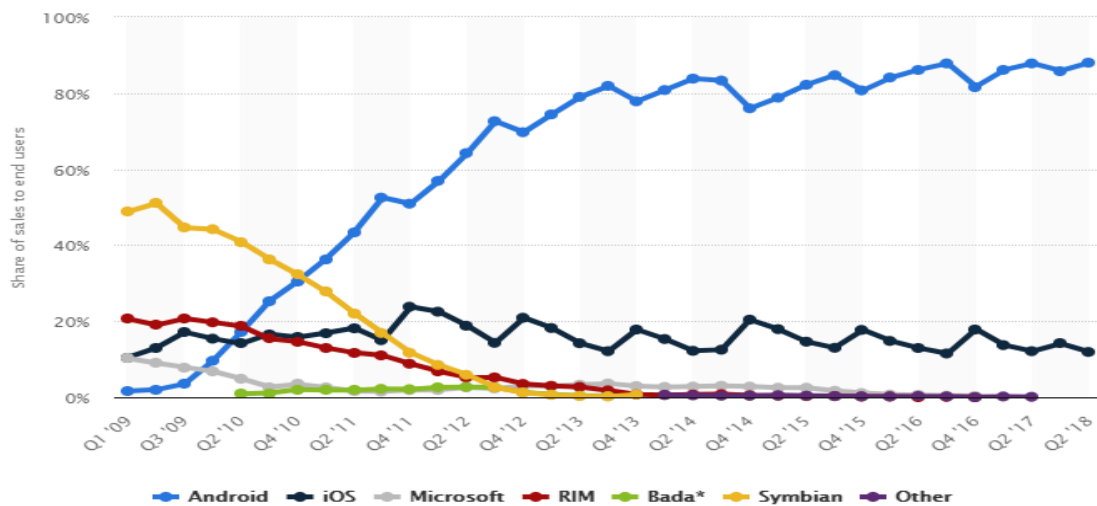


Figure 1.3 : Part des ventes des systèmes d'exploitation mobile aux utilisateurs finaux

Il est clair, dans la figure 03 que le Symbian entre 2009 et 2011 a dominé le marché des systèmes d'exploitation mobile. Après l'an 2010, l'Android et iOS d'Apple en font écarter le Symbian et les autres systèmes d'exploitation mobile du marché, iOS a connu une stabilisation depuis son apparition jusqu'à nos jours, mais l'Android a connu une croissance qui dépasse le 40 millions de téléchargements et donc domine le marché des systèmes d'exploitation mobile. Pour cette raison, nous avons opté pour le choix d'Android comme système pour notre application.

1.2.7 Le marché des applications mobiles :

Les applications mobiles font recette année après année, selon le site (statista.com) les revenus des applications connaissent une croissance de 100 milliards de dollars de 2016 à 2020 et le diagramme de la figure 04 nous montre cette croissance.

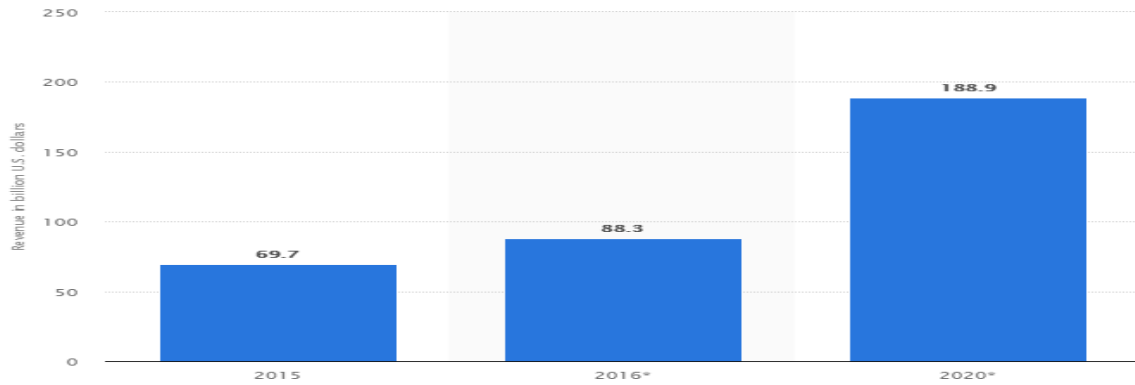


Figure 1.4 : les revenus projetés des applications mobiles de 2015 à 2020

1.3 La Vidéosurveillance

1.3.1 Définition :

La vidéosurveillance consiste à placer des caméras de surveillance dans un lieu public ou privé pour visualiser et/ou enregistrer en un endroit centralisé tous les flux de personnes au sein d'un lieu ouvert au public pour surveiller les allées et venues, prévenir les vols, agressions, fraudes et gérer les incidents et mouvements de foule. Au début des années 2000, les caméras font leur apparition en nombre important dans de nombreuses villes européennes. Londres est réputée comme étant la ville où la vidéosurveillance est la plus importante(3).

1.3.2 L'apparition de la Vidéosurveillance :

La vidéosurveillance s'est développée d'abord au Royaume-Uni, en réponse aux attaques de l'IRA(Armée républicaine irlandaise en anglais Irish Republican Army). Les premières expériences au Royaume-Uni dans les années 1970 et 1980 ont conduit à des programmes de grande ampleur au début des années 1990. Ces succès conduisirent le gouvernement à faire une campagne auprès de la population, et lança une série d'installations de caméras. Aujourd'hui, les caméras au Royaume-Uni couvrent la plupart des centres villes, et de nombreuses gares et parkings. Une étude donna le chiffre approximatif de 400 000 caméras à Londres et 4 millions au Royaume-Uni au total. D'autres pays comme la France ont installé des systèmes de vidéosurveillance. En 1998 le nombre de caméras en France était estimé à un million dont 150 000 dans le domaine public. Ces caméras sont présentes dans divers lieux tels que les aéroports, les gares, les routes, les transports publics. Ces installations vidéo commencent aussi à fleurir dans les villes. À Avignon par exemple, une enquête à propos de la

vidéosurveillance a révélé que 71 % des Avignonnais sondés étaient favorables à l'installation d'un tel système dans les parkings(9).

1.3.3 Les buts :

Les raisons de l'installation de systèmes de vidéosurveillance sont diverses, toutefois la sécurité publique ainsi que la protection des biens mobiliers ou immobiliers font office d'éléments phares dans la justification de la vidéosurveillance. En Angleterre, les attentats de juillet 2005 sont également un moteur pour l'augmentation du nombre de caméras. Cette menace qui a toujours été présente n'a jamais vraiment créé un sentiment d'insécurité, mais les attentats du 11 septembre 2001 ont changé la donnée. Les gens ont pris conscience que personne n'était intouchable. Toutefois la mise en place de la vidéosurveillance ne peut s'expliquer uniquement par l'insécurité grandissante ou la protection des biens. Certaines autres raisons moins connues du grand public existent également. La mise en place de la vidéosurveillance permet une amélioration de la gestion des incidents ainsi qu'une augmentation de l'efficacité et de la rapidité d'intervention. Par exemple, dans la prévention du suicide ou encore lors d'accidents qui pourraient survenir sur la voie publique. Elle permet ainsi indirectement, de maintenir les primes d'assurances à un niveau raisonnable.

La surveillance des axes routiers sert à informer en temps réel les automobilistes sur les conditions du trafic. Quelques affaires de crimes ont été résolues grâce aux enregistrements fournis par les caméras de surveillance(2).

1.3.4 Domaines d'application

La tendance de l'économie mondiale actuelle exige aux entreprises d'être réactive devant les demandes de plus en plus gourmandes de moyen de connectivité et infrastructure de communication et de marketing, la technologie de la vidéo sur réseau IP redynamise les applications de vidéosurveillance par de nombreuses fonctionnalités comme le contrôle à distance, la vidéo en temps réel. Ce qui rend plusieurs secteurs d'activités interactives. On dénombre trois grandes catégories publiques dans lesquelles l'on retrouve ces systèmes de surveillance : Les aéroports, les transports publics et les gares.

1.3.5 Les types de systèmes de Vidéosurveillance

Les systèmes de vidéosurveillance obéissent presque tous à un même schéma de base englobent tout un ensemble de d'équipement orientés sécurité et peuvent être réparties en trois grands groupes :

- (a) Les systèmes de vidéosurveillance analogique.
- (b) Les systèmes de vidéosurveillance IP.
- (c) Les systèmes « hybrides ».

1.3.5.1 Les systèmes analogiques :

Le système analogique est un système de vidéosurveillance utilisant un magnétoscope traditionnel, ce système est entièrement analogique dans lequel les caméras analogiques avec sorties coaxiales sont reliés au magnétoscope pour l'enregistrement

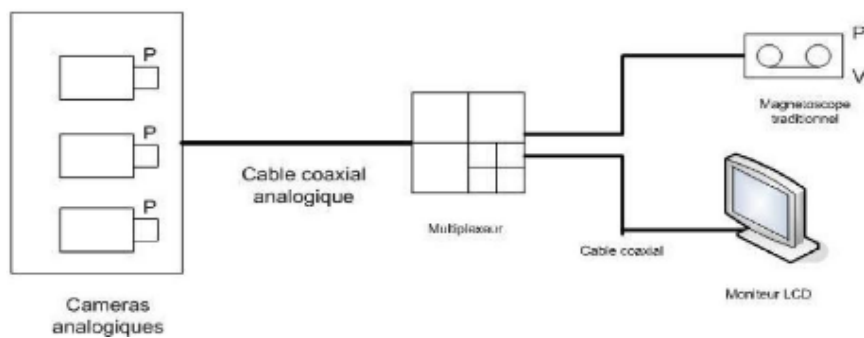


Figure 1.5 : Système analogique

1.3.5.2 Système sur réseaux IP :

Ce système relie un réseau de caméras IP, qui peut compter de nombreuses unités, à un système d'enregistrement numérique. D'une part, cela permet de pouvoir stocker une quantité importante d'images, sans perte de qualité, tout en pouvant les consulter rapidement grâce à des logiciels de traitement. D'autre part, le fait d'informatiser un système de surveillance permet de profiter des technologies de communication comme Internet. Ainsi, les caméras sont « visibles » et gérables depuis n'importe où dans le monde. L'évolution des téléphones mobiles a créé la "**vidéosurveillance mobile**" avec l'accès aux vidéos via Internet mobile sur PDA ou via GSM GPRS sur téléphone GSM doté de Java. Cette technologie permet également d'économiser et de mutualiser les câbles réseaux qui sont généralement disponibles dans les bâtiments récents(a).

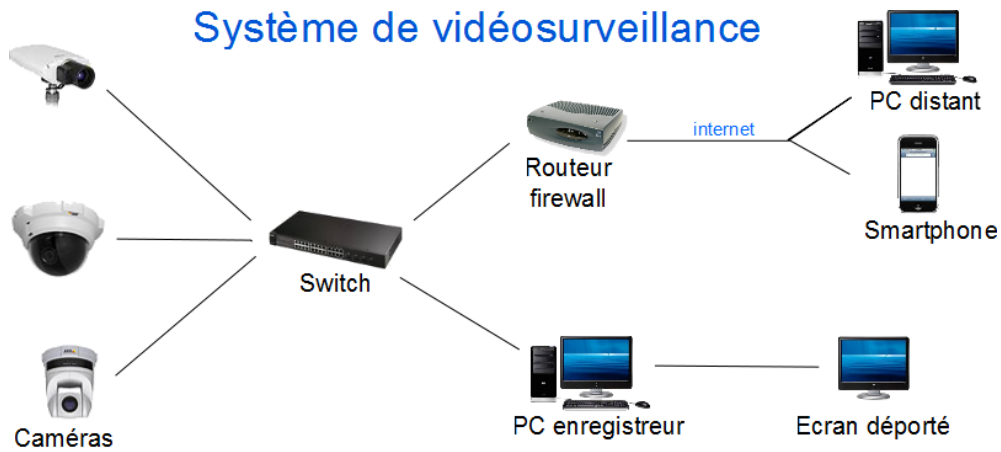


Figure 1.6 : système sur réseaux IP

1.3.5.3 Système hybride :

Un système hybride est un système de vidéo surveillance analogique utilisant un enregistreur numérique. L'enregistrement ne se fait plus sur bandes mais sur des disques durs.

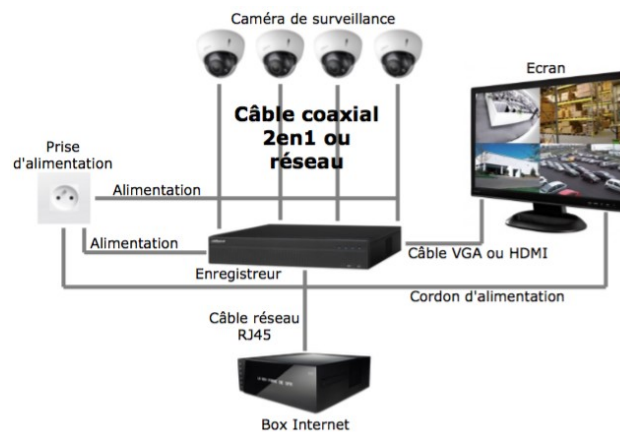


Figure 1.7 : Système hybride

1.4 Conclusion :

Dans ce chapitre, nous avons donné un aperçu sur les applications mobiles, ainsi que ses différents objectifs, puis nous avons présenté les différents systèmes d'exploitation mobiles, A la fin nous avons entamé les notions de base de la vidéosurveillance qu'est une partie primordiale dans notre projet.

Dans le chapitre suivant on voit en détail c'est quoi la carte électronique Raspberry pi, ses composant, son domaine d'utilisation et son emplacement dans notre projet.

Chapitre II

La mise en place du microcontrôleur

2.1 Introduction

Dans ce chapitre, on va présenter la carte électronique Raspberry Pi comme un module de notre projet de vidéosurveillance. On va voir c'est quoi la carte Raspberry Pi, ses composants et ses fonctionnalités.

2.2 Définition d'un microcontrôleur :

Un microcontrôleur (en notation abrégée μc , ou uc ou encore MCU en anglais) est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires (mémoire morte et mémoire vive), unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique, une vitesse de fonctionnement plus faible (de quelques mégahertz jusqu'à plus d'un gigahertz) et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels.

2.3 Présentation des différents microcontrôleurs :

Le Raspberry Pi : est un ordinateur de la taille d'une carte de crédit et pouvant être connecté à un moniteur ou à un téléviseur. On peut l'utiliser à l'aide d'un clavier et d'une souris standard. Il est capable de faire tout ce qu'un ordinateur peut faire, depuis la navigation sur Internet jusqu'à la lecture de vidéos haute définition en passant par la création de feuilles de calcul, le traitement de texte et les jeux(h).



Figure 2.1 : Carte Raspberry pi

L'Arduino : est une carte qui se connecte sur l'ordinateur pour être programmée, et qui peut ensuite fonctionner seule si elle est alimentée en énergie. Elle permet de recevoir des informations et d'en transmettre depuis ou vers des matériels électroniques : diodes, potentiomètres, récepteurs, servomoteurs, moteurs, détecteurs.

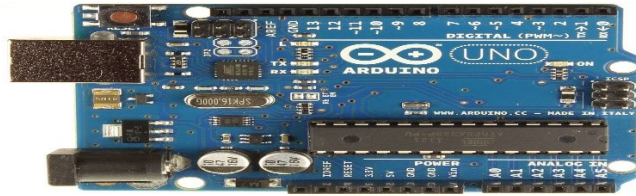


Figure 2.2 : Carte Arduino

STM32 : c'est un circuit intégré à base d'un microcontrôleur composé d'un processeur ARM, une mémoire RAM et une interface de débogage

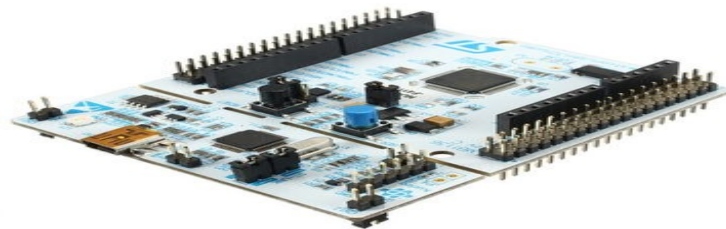


Figure 2.3 : Carte STM32

Beaglebone : C'est une mini-carte électronique qui possède les fonctionnalités d'un ordinateur basique.

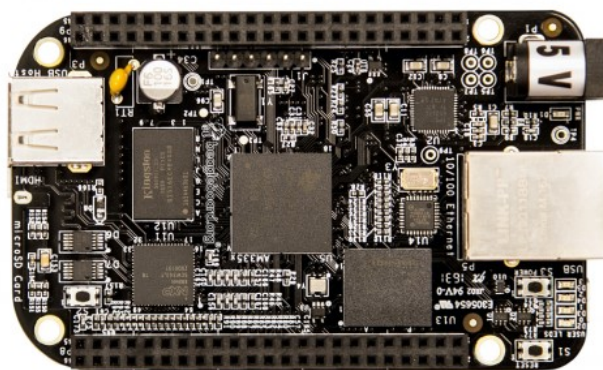


Figure 2.4 : Carte Beaglebone black

Le tableau de la figure 12 représente une comparaison technique entre les microcontrôleurs :

Nom de la carte	Arduino	Beaglebone	Raspberry pi	STM 32
Naissance	2005	2008	2012	2007
Prix en Dollars	30	90	25	10
Processeur	ATmeg328 8bits	Sitara335x Sur un cotex M8	BCM2835 sur un ARM11- cpu intégrée	Cortex
RAM	2 ko	256 Mo	512Mo	64 Ko
Mémoire Flash	32 Ko	MicroSD-4G	MicroSD-32G	Max 2048 Ko
Consommation (w)	05	2.5	3.5	0.3
Ethernet	Non	Oui	Oui	Non
USB	Non	1	4	Micro-USB
Sortie vidéo	Non	Non	HDMI, RCA	Non
Système d'exploitation	RTOS	Linux	Linux	RTOS
Disponibilité dans le Marché Algérien	Oui	Non	Oui	Non
Convertisseur Analogique/numérique	oui	Non	Non	Oui

Tableau 01 : Tableau de comparaison technique entre les cartes électroniques

Analyse du tableau comparatif : Nous remarquons qu'en termes de mémoire vive ou mémoire externe, la capacité de Raspberry est plus importante que les autres cartes.

- (a) Le Raspberry possède des connectiques (HDMI, port Ethernet, port USB) de plus par rapport au Beaglebone, STM32 et Arduino UNO
- (b) Le Raspberry et le Beaglebone se différencient par leur exécution du système Linux qui facilite les communications avec le monde extérieur.
- (c) Le Raspberry pi et l'arduino sont disponibles dans le marché.
- (d) Le Raspberry pi ne lit pas les signaux analogiques mais l'arduino convertit les signaux analogiques en valeurs numériques

2.4 Choix du Raspberry pi :

Après l'analyse du tableau de la figure 12 on a déduit que les cartes Raspberry pi et arduino représentent des outils plus performants et mieux adaptés à nos besoins.

Le Raspberry pi possède des caractéristiques convenables au fonctionnement de notre système. De plus, la disponibilité dans le marché algérien et son faible cout.

L'utilisation des capteurs gaz exige un outil qui convertir les signaux analogiques en valeurs numériques et on a trouvé cette caractéristique dans l'arduino.

2.5 Spécifications Matériels du Raspberry Pi :

Le Raspberry pi possède de :

Processeur ARM : Les architectures ARM sont des architectures de processeurs, à faible consommation, introduites à partir de 1983 par « Acorn Computers » et développées depuis 1990 par « ARM Ltd ». Raspberry Pi B possède un processeur intégré Quad-core ARM Cortex-A53 1.2 GHz (Broadcom BCM2837).

Mémoire Vive RAM : C'est la mémoire dans laquelle le Raspberry place les données lors de son traitement, elle a une capacité de 1Go.

Une connectique variée :

1. HDMI : « High Definition Multimedia Interface » permet de relier le Raspberry Pi à un dispositif compatible : écran LCD ou un vidéoprojecteur, téléviseur
2. Port USB 2.0 : Le port « Universal Serial Bus » est un port série qui sert à connecter le Raspberry aux autres périphériques, et il y a 4 ports USB. Le Raspberry Pi fonctionne avec pratiquement n'importe quels claviers et souris USB. On peut aussi utiliser la plupart des claviers et des souris sans fil (ceux qui marchent avec un dongle que l'on branche sur un port USB).
3. Port Ethernet : C'est un port qui correspond au protocole international Ethernet de réseau local à commutation de paquets.
4. Prise RCA : « Radio Corporation of America » est un connecteur électrique utilisé dans le domaine audio/vidéo via jack.

5. Un slot les cartes SD : Le Raspberry a besoin d'une mémoire externe supplémentaire pour fonctionner. Ce slot permet de connecter la mémoire externe. On peut utiliser notre propre carte SD dans le Raspberry Pi, mais elle devra être préparée avec une image disque d'un système d'exploitation.

6. Une prise jack : C'est une connectique audio-vidéo de dimension 3.5 mm

7. Bloc d'alimentation : l'image de la figure 13 représente le bloc d'alimentation USB classique avec un câble USB-A vers micro-USB. L'alimentation doit être capable de fournir au moins 700 mA, mais 1 A donné au Raspberry Pi un peu de marge qui sera utilisée par les périphériques connectés à ses ports USB. Si vous regardez attentivement les spécifications de l'alimentation, vous devriez être capable de déterminer sa capacité. Il arrive que la capacité soit exprimée en watts (W) , si tel est le cas, elle doit être au moins de 3 W. Si elle indique 5 W, cela correspond à 1 A.



Figure 2.5 : Bloc d'alimentation du Raspberry Pi

8. GPIO : Les broches GPIO « General Purpose Input/Output » du Raspberry Pi Permis de contrôler d'autres composants électroniques ainsi que des interfaces telles que des LED, des moteurs et des relais. Ces diverses interfaces sont généralement regroupées sous le terme de « sorties ». Pour ce qui est des « entrées » , le Raspberry Pi peut lire et interpréter l'état de boutons, d'interrupteurs, de capteurs de température, de lumière, de mouvement ou de proximité, etc., la liste est très longue. En utilisant les interfaces numériques asynchrones (port série vu au

chapitre précédent) et synchrones (SPI et I2C), les applications potentielles sont encore plus nombreuses. Le Raspberry pi B comporte 40 broches.

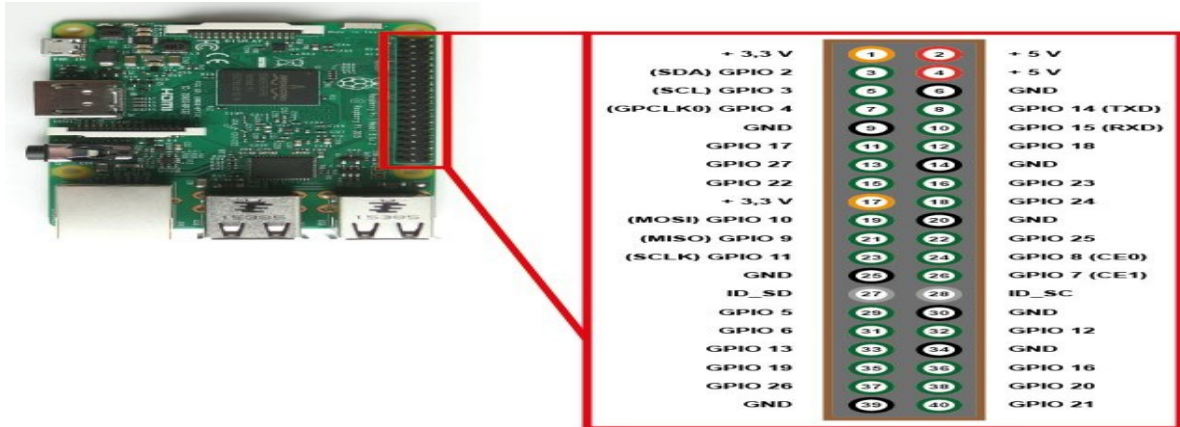


Figure 2.6 : les Ports GPIO

9. Le module de caméra Raspberry Pi : est une caméra CMOS⁶ de 5 MP avec un objectif à focale fixe capable de capturer des images fixes ainsi que des vidéos hautes définitions. Les images fixes sont capturées à une résolution de 2592 x 1944, tandis que la vidéo est prise en charge à 1080p à 30 ips, 720p à 60 ips et 640x480 à 60 ou 90 ips. La caméra est prise en charge dans la dernière version de Raspbian, le système d'exploitation préféré de Raspberry Pi.



Figure 2.7 : La Pi Camera

⁶ CMOS (Complementary Metal Oxide Semiconductor) : est un capteur d'image intégré dans la plus part des caméras, ce capteur transforme les signaux lumineux en signaux numérique

10. Boîtier : Le Raspberry Pi est livré nu, sans protection. Cela permet d'abaisser son prix, mais cela le rend vulnérable à la casse. Il est par conséquent préférable d'acheter un boîtier dès que vous le pouvez ou bien d'en réaliser un vous-même. Le choix du boîtier dépend de ce vous envisagez de faire avec votre Raspberry Pi



Figure 2.8 : Boitier pour Raspberry Pi

L'image de La figure 17 nous montre les différents composants de la Raspberry pi :

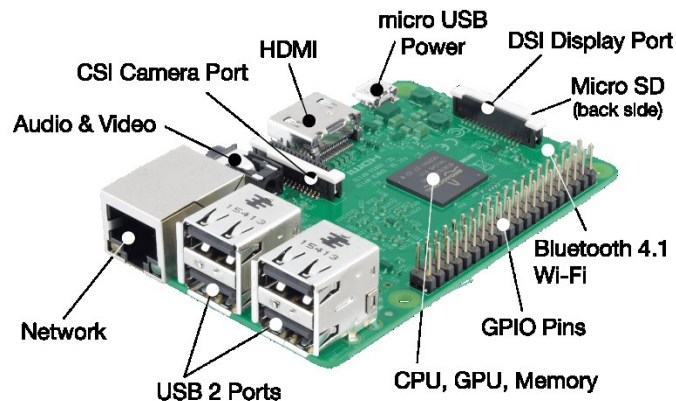


Figure 2.9 : les différents composants du Raspberry Pi

2.6 Spécifications logiciel du Raspberry Pi :

2.6.1 Le système D'exploitation :

La plupart des systèmes qui fonctionnent sur Raspberry Pi sont des versions du système d'exploitation Linux. Parce que Linux est open source, les développeurs peuvent l'adopter pour des buts spécifiques. La distribution recommandée est Raspbian. C'est un système d'exploitation libre basé sur Debian optimisé pour le matériel Raspberry Pi. Cependant

Raspbian fournit plus qu'un simple système d'exploitation : il est livré avec plus de 35.000 paquets, des logiciels précompilés qui nous facilitent le développement soft. Il y a autre système d'exploitation créé par Microsoft spécialement pour le domaine des IoT Windows 10 IoT Core qui peut être installé sur notre carte Raspberry Pi B.

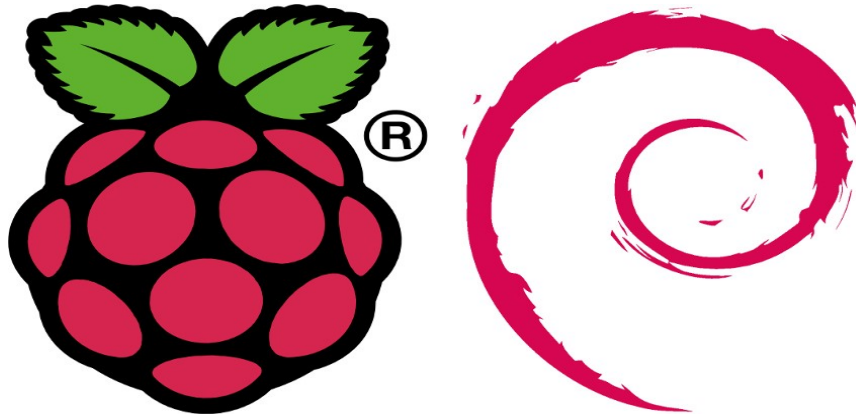


Figure 2.10 : le slogan officiel du Raspbian

2.6.2 Logiciels compatible avec le Raspberry Pi :

Toutes les applications compatibles avec le système d'exploitation et le processeur ARM, ou utilisant un environnement d'exécution virtuel (Java, émulateurs...) sont susceptibles de fonctionner : Iceweasel, KOffice, Python. Les principales contraintes portent sur les performances du processeur et la mémoire vive disponible (256 Mo). Ce dernier point a toutefois été corrigé avec l'arrivée de versions embarquant 512 Mo de mémoire vive, puis d'une version embarquant 1 Go de mémoire vive(g).

Le logiciel open source Aseba, pour programmer des robots facilement et efficacement, est disponible sur Raspberry Pi. En utilisant Raspberry Pi avec Aseba et le Thymio , un véritable laboratoire d'enseignement de la robotique peut être créé avec un coût très bas.

2.7 Démarrage du Raspberry Pi :

Pour démarrer Raspberry Pi pour la première fois on est obligé de suivre les étapes suivantes :

- (a) Installer le système d'exploitation sur une carte SD.
- (b) Insérer la carte SD dans son support.
- (c) Insérer la carte SD dans son emplacement.
- (d) Brancher un clavier USB et une souris au Raspberry PI.
- (e) Connecter la sortie HDMI au téléviseur ou moniteur.

(f) Brancher l'alimentation du pi.

Après le démarrage de la Raspberry pi on trouve une interface qui nous permet de :

- (a) Choisir la langue du système
- (b) Configurer les paramètres réseau
- (c) Choisir la langue du clavier
- (d) Configurer les paramètres de localisation
- (e) Configurer les paramètres de connexion.
- (f) Modifier les paramètres de boot.
- (g) mettre à jour le système

Une fois que ses paramètres sont définis on clique sur Finish et puis on peut utiliser à Raspberry pi

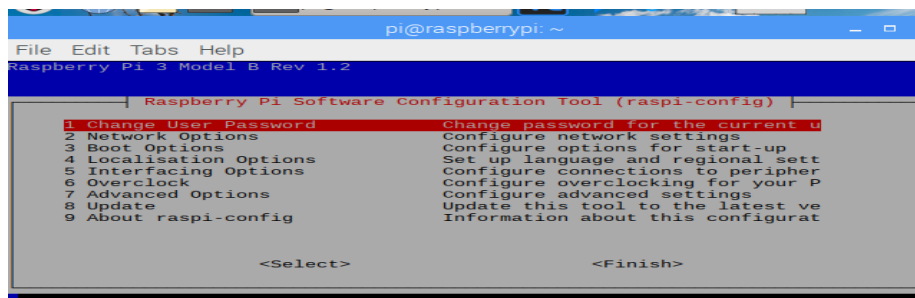


Figure 2.11 : La table de configuration du Raspberry pi

Si on veut reconfigurer ses paramètres en lance la commande suivante :

```
sudo raspi-config
```

2.8 Connexion à distance :

Pour pouvoir se connecter à distance à notre Raspberry, il faut le connecter à un réseau local et faire appel au protocole SSH.

Connexion SSH : Le serveur SSH est activé par défaut sur le Raspberry Pi. Sous Linux, le client étant intégré dans la plupart des distributions. Par contre, sous Windows, nous

devons installer « Putty » qui est un client SSH et Telnet. Ensuite, il suffit d'entrer l'adresse IP du Raspberry dans Putty pour pouvoir se connecter. Afin d'accéder à l'interface graphique, nous avons utilisé le système de partage de bureau graphique VNC

VNC (virtuel Network Computing): VNC est un système de partage de bureau graphique qui nous permet de contrôler à distance l'interface de bureau d'un ordinateur (exécutant VNC Server) à partir d'un autre ordinateur ou périphérique mobile (exécutant VNC Viewer). VNC Viewer transmet le clavier et les événements relatifs à la souris ou au toucher au serveur VNC, puis reçoit en retour des mises à jour de l'écran.

Nous verrons le bureau du Raspberry Pi dans une fenêtre de l'autre ordinateur ou d'appareil mobile. Nous pourrions le contrôler comme si nous travaillions sur le Raspberry Pi lui-même.

Configuration de VNC server :

Sur le Raspberry Pi, on exécute les commandes suivantes pour assurer que ce dernier possède la dernière version de VNC.

```
sudo apt-get update
sudo apt-get install realvnc-vnc-server realvnc-vnc-viewer
```

Puis on active le serveur vnc par la commande suivante :

```
vncserver
```

Etablissement de connexion :

Sur le périphérique que nous utiliserons pour prendre le contrôle, on télécharge VNC Viewer. Pour de meilleurs résultats, on utilise l'application compatible de RealVNC on suivre les étapes suivantes nous pouvons le contrôler.

- (a) En entre l'adresse IP du Raspberry pi.
- (b) En entre les informations d'authentification (nom d'utilisateur et le mot passe).
- (c) Une fois connecté, nous pouvons servir du bureau de notre Raspberry sur l'ordinateur.

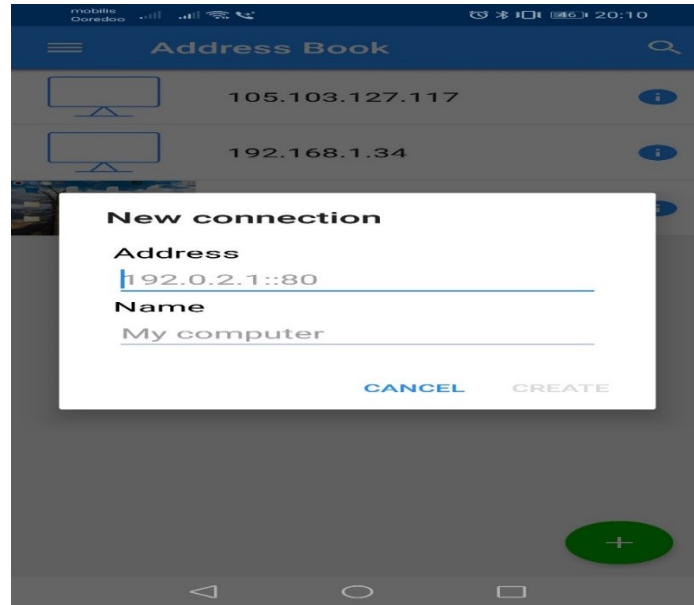


Figure 2.12 : outil de connexion à distance



Figure 2.13 : écran du démarrage du Raspberry Pi

2.9 Les différentes utilisations du Raspberry Pi :

Il est bien évidemment impossible de dresser une liste exhaustive des projets réalisables avec le Raspberry pi. Cependant, il est possible de distinguer un certain nombre d'usages globaux.

2.9.1 Les serveurs :

Dans le cadre d'un serveur, le Raspberry sera branché et fonctionne en permanence sans la moindre interruption. Il restera probablement proche du box internet, et sera contrôlée à distance via SSH, le moins souvent possible.

1. Le serveur web : Il s'agit d'un serveur visant à héberger un ou plusieurs sites web, qui devront être accessibles depuis l'extérieur.

2. Le serveur de stockage : Il s'agit d'un serveur qui permet de stocker les fichiers pour y avoir accès depuis n'importe où. Il s'agira en général d'un serveur dit « FTP », c'est-à-dire une machine exécutant un logiciel permettant de traiter le protocole « File Transfert Protocole », qui est un protocole dédié au transfert de fichiers.

3. Le serveur de service : Il s'agit d'un serveur un peu à part, qui vise à héberger un service, en général pour pouvoir y accéder depuis n'importe où. On entend par « service » un logiciel, mais qui ne sera pas un client, c'est-à-dire un logiciel tournant au niveau de la machine de l'utilisateur, et permettant de faire des appels vers une entité centrale. Il va justement s'agir de cette entité centrale, responsable des opérations, des calculs, de la coordination des différents clients, etc.

2.9.2 La domotique

Le problème avec la domotique c'est qu'il n'existe pas beaucoup d'interfaces complètes, et elles sont assez chères, mais le Raspberry Pi est présent pour résoudre ce problème. En effet, comme il est de taille réduite, il peut facilement s'intégrer dans une pièce. De plus, avec ses nombreuses sorties, notamment GPIO, il peut accueillir de nombreux modules pour communiquer avec des équipements domotiques.

2.9.3 Les systèmes embarqués

Le Raspberry pi étant particulièrement petite, et possédant des entrées GPIO, il est souvent utilisé dans le cadre de projets liés à l'électronique, en tant que contrôleur central d'un système informatique, etc. De nombreux projets émergent, comme le contrôle d'une voiture télécommandée avec caméra, des drones, ou même un amusant projet sur le piratage de drone.

2.9.4 Un ordinateur

Tout simplement On l'oublie souvent, mais avant tout, le Raspberry Pi est un ordinateur. Petit, bon marché, et souvent suffisant pour bien des utilisations. C'est sa raison d'exister. IL fournit une alternative aux personnes n'ayant pas les moyens d'acheter un ordinateur standard. Il permet aussi d'encourager l'apprentissage de la programmation.

2.10 Conclusion :

Dans ce chapitre dédié à la présentation du Raspberry Pi, nous avons présenté les divers composants de cette carte, son utilité spécialement pour notre projet sur la Vidéosurveillance.

Durant le chapitre suivant nous allons présenter les objectifs du projet puis on va représenter graphiquement les taches de chaque entité du système

Chapitre III

Analyse et Conception

3.1 Introduction :

Il est nécessaire de passer par une phase d'analyse et de conception l'analyse consiste à définir les besoins d'utilisateur et les contraintes du système. Tandis que la conception permet de donner une représentation graphique pour les entités du système.

Le langage de modélisation UML sera choisi dans notre cas.

3.2 Analyse des objectifs :

3.2.1 Objectif du travail

L'objectif de ce travail est la réalisation d'un système de Vidéosurveillance automatique qui détecte les vols et les incendies d'une façon automatique et ça va permettre aux utilisateurs d'augmenter le niveau de la sécurité physique.

3.2.2 Public visé

Notre système de télésurveillance est destiné à l'usage des entreprises et des établissements qui désirent sécuriser et protéger leurs locaux et leurs biens.

3.2.3 Le contenu

Notre système contient trois parties principales :

La première partie représente le système principal qui est chargé d'acquisition des images et la détection des mouvements et des incendies,

La deuxième partie représente les serveurs, sur lesquels on va publier les données avant la transmission.

La troisième partie est l'application mobile qui permet l'accès à distance par l'utilisateur de visualiser leurs différents camera et la réception des notifications en cas de prise de vol ou une incendie.

Ces parties seront détaillées dans le chapitre 04

3.3 Spécification des besoins

3.3.1 Les acteurs

La conception du système repose sur les besoins des utilisateurs, Cesdernier peuvent être des administrateur ou des utilisateurs finaux, la réussite de

l'application nécessite la réponse à tous besoins en prenant en considération les contraintes et les dépendances entre les besoins.

L'administrateur :

- (a) L'ajout des utilisateurs
- (b) La suppression des comptes utilisateur
- (c) La modification des comptes
- (d) La gestion de base de données

L'utilisateur :

- (a) Visualisation des caméras.
- (b) Consultation des messages
- (c) Réception des notifications

3.3.2 Besoins fonctionnels

- (a) Visualisation du flux des vidéos des caméras par l'application mobile.
- (b) Détection du mouvement.
- (c) Enregistrement du flux vidéo.
- (d) Capture des images.
- (e) Déclenchement d'alarme.
- (f) Envois des messages de diffusion
- (g) Gestion des vidéos et des images (suppression, relecture, archivage)

3.3.3 Besoins non-fonctionnels

A part les besoins fondamentaux, notre futur système doit répondre aux critères suivants :

D'une ergonomie simple et interface facile et intuitive : la solution doit présenter une interface ergonomique englobant toutes les fonctionnalités offertes. La manipulation de l'interface ne doit pas nécessiter des connaissances poussées en informatique, elle doit être simple et claire afin de s'adapter aux connaissances informatiques de nos utilisateurs.

Robuste et maintenable : Le système doit permettre le stockage des informations concernant tous les internautes inscrits et le différent traitement utiles pour le fonctionnement correct.

Sécurisé : notre système doit garantir à l'utilisateur l'intégrité des données c'est –à-dire qu'ils gardent leur forme et leur contenu original. En outre, il doit protéger la Confidentialité en assurant la validité de l'identité de l'utilisateur .ceci peut se faire entre autres par le moyen d'un mot de passe assurant le contrôle d'accès.

Notre système doit également certifier la disponibilité qui s'avère primordiale pour le bon fonctionnement.

Fiable et rapide : Notre système doit garantir la rapidité et la fiabilité de la recherche des informations, ainsi qu'une gestion optimale des ressources.

3.4 Planification

3.4.1 Architecture

Notre système se base sur l'application mobile qui permet de suivre les caméras en ligne et reçois des messages d'alerte en cas d'une prise de vol ou un incendie, les capteurs publies des messages dans le serveur (publish) et l'application mobile récupère ces messages d'une façon automatique par son abonnement dans le topic (subscribe).

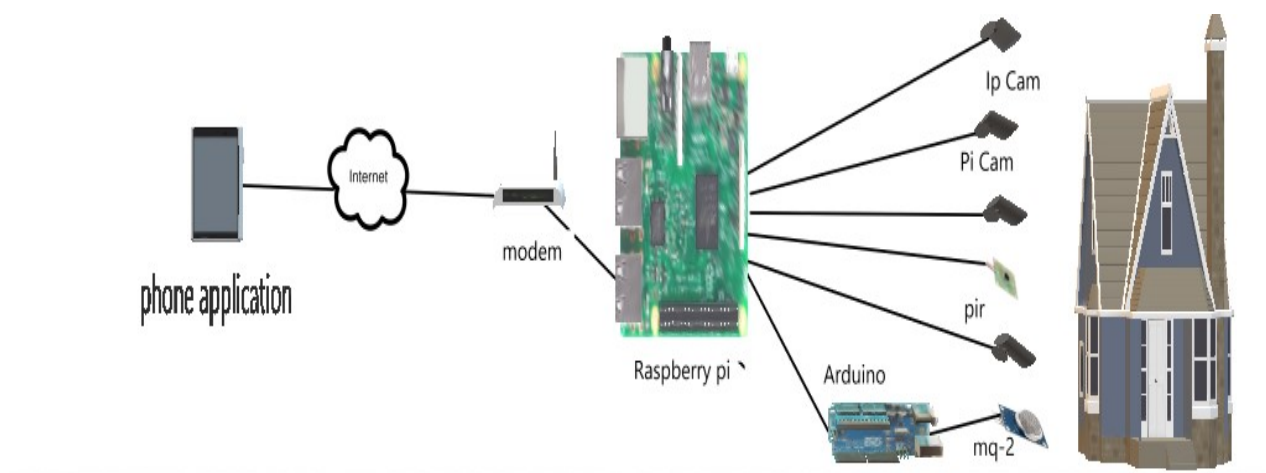


Figure 3.1 : Architecture du système

3.4.2 Cas d'utilisations

Un cas d'utilisation définit une manière d'utiliser le système et permet d'en décrire les exigences fonctionnelles. D'après Bittner et Spence, « Un cas d'utilisation, défini simplement, permet de décrire une séquence d'événements qui, pris tous ensemble, définissent un système

faisant quelque chose d'utile ». Chaque cas d'utilisation contient un ou plusieurs scénarios qui définissent comment le système devrait interagir avec les utilisateurs (appelés acteurs) pour atteindre un but ou une fonction spécifique d'un travail. Un acteur d'un cas d'utilisation peut être un humain ou un autre système externe à celui que l'on tente de définir

3.4.3 Diagrammes des cas d'utilisations :

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés.

Notre système, m'amène à distinguer entre deux acteurs : utilisateur (employée) et Administrateur.

Le diagramme de cas d'utilisation de la figure 23, met en évidence les grandes relations fonctionnelles entre les acteurs (principaux) et le système.

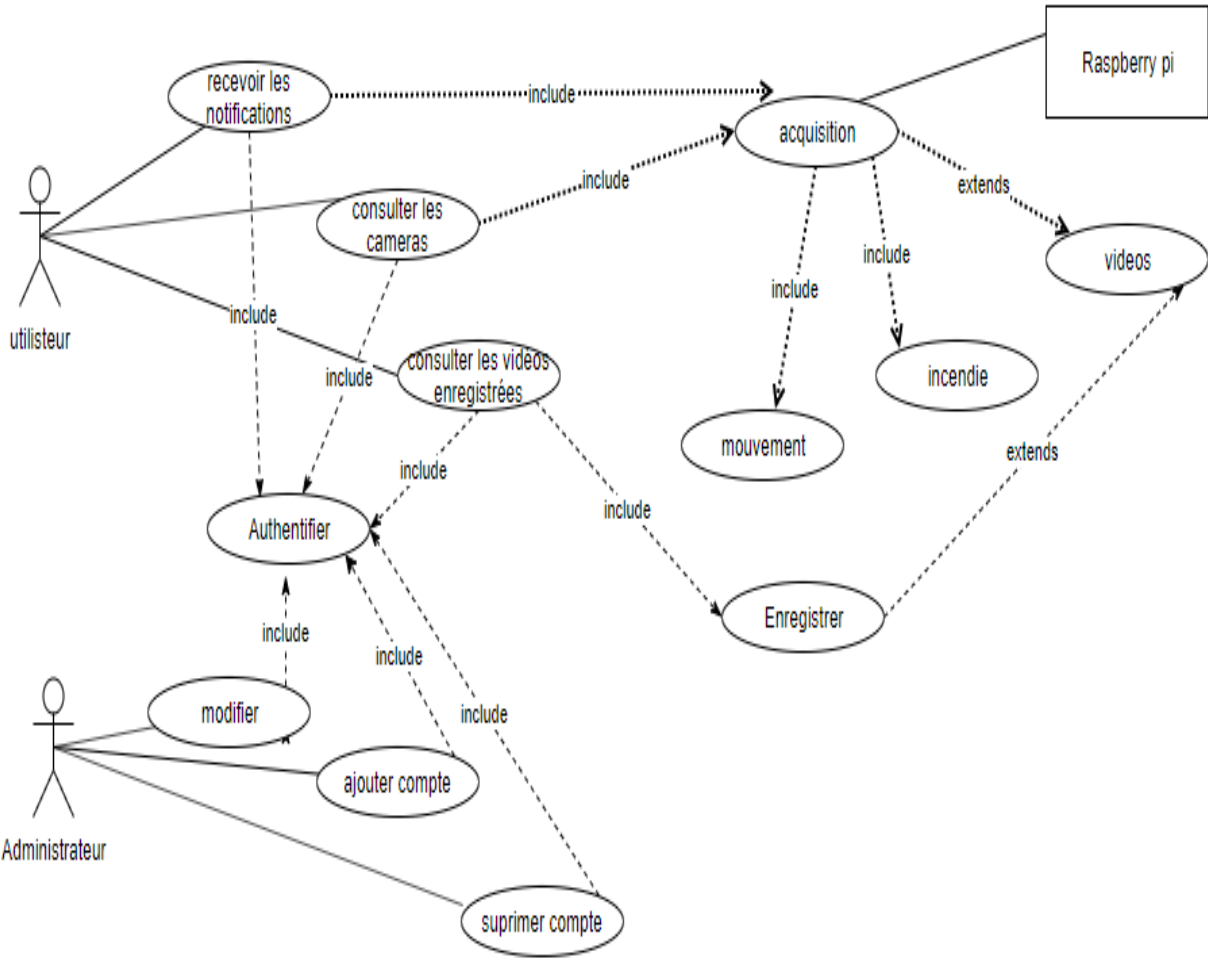


Figure 3.2 : Diagramme de cas d'utilisation

Cas d'utilisation	Consulter les données
Acteurs principaux	Utilisateur
Objectif	l'utilisateur utilise l'application pour consulter les données suivantes : (b) vidéo (c) l'état des capteurs (d) les cameras
préconditions	Les capteurs et les caméras sont placés et installé dans l'environnement. Connexion internet. L'activation du réseau. Lancer l'application. S'authentifier
Scénario nominale	l'utilisateur lance son application. L'application vérifier les données de s'authentification.
Scénario alternative	Si le wifi n'est pas activer alors l'application demande une connexion a l'internet Si l'identifiant incorrect : - l'application indique que les informations saisis sont incorrects.

Tableau 02 : Cas d'utilisation consulter les données

Cas d'utilisation	Acquisition des données
Acteurs secondaires	Microcontrôleur (Raspberry Pi)
Acteurs principal	L'administrateur
Objectif	Le microcontrôleur contient plusieurs capteurs qui sont en liés, pour capter les évènements et les publier dans le broker
Prés conditions	Le microcontrôleur est bien placé et bien installé L'existence d'une connexion à un réseau (LAN ou Internet). Le serveur fonctionne est en écoute.
Scénario nominale	1. L'administrateur allume le microcontrôleur. 2. le microcontrôleur capte Les données et les publier dans le broker

Tableau 03 : Cas d'utilisation acquisition des données

Cas d'utilisation	Gérer les comptes utilisateurs
Acteur principal	Administrateur
Objectif	L'administrateur a la possibilité de gérer les comptes des utilisateurs : Ajouter (nom_util, tel_util, mail_util, pass_util). Pouvoir de mettre à jour les informations des utilisateurs de système (pseudo Name - nom d'utilisateur -, numéro de téléphone, email, mot de passe). Peut supprimer compte d'utilisateur.
Prés conditions	L'application serveur fonctionne très bien et toujours en écoute.
Scénario nominal	L'administrateur lance le sgbd. 2. L'application demande de s'authentifier. 3. L'administrateur s'authentifier au système. 4. l'administrateur choisit la base de données 5. l'administrateur choisit la table user 6. l'administrateur gère les comptes.

Tableau 04 : Cas d'utilisation gérer les comptes

3.4.4 Diagramme de classes :

Le diagramme de classe représente l'architecture conceptuelle du système, il exprime de façon générale de la structure interne du système et montre les relations entre les classes composant le système. Dans cette partie, nous allons proposer un diagramme de classe conforme à notre projet.

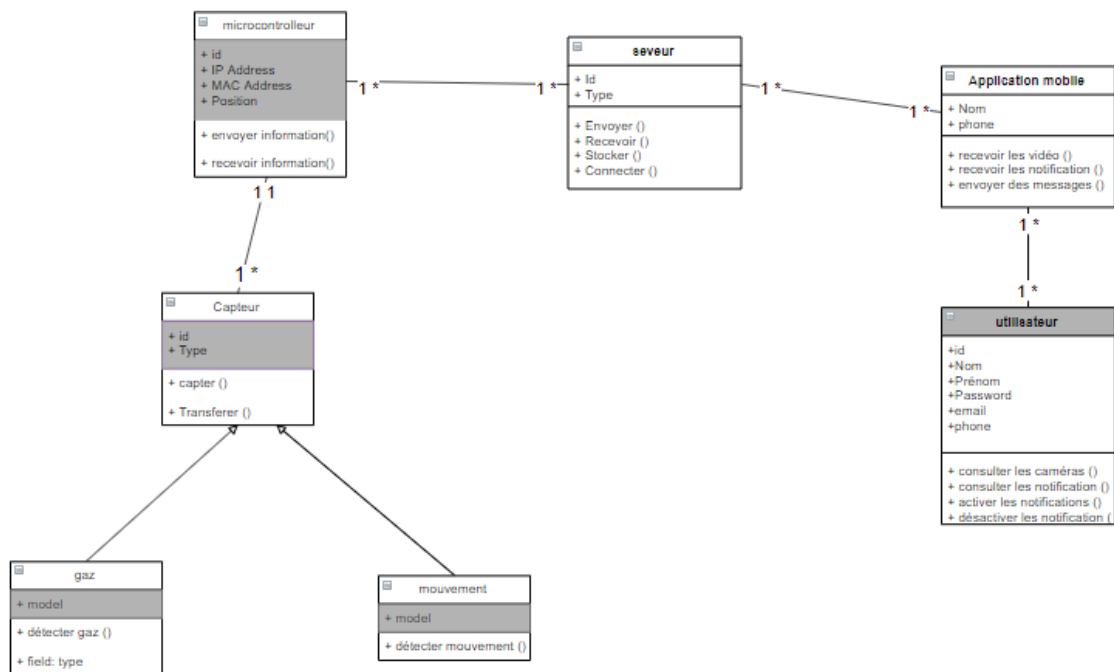


Figure 3.3 : Diagramme de classes

Classe	Description
Administrateur	Représenter l'administrateur qui gère les compte utilisateur et régler la sensibilité des capteurs
capteur	Représente la classe mère des sous classes capteur du mouvement et du gaz
Utilisateur	Représenter l'utilisateur qui consulter les camera et recevoir les notifications

Tableau 05 : tableau démonstratif de diagramme de classes

3.4.5 Diagrammes d'activités :

Le diagramme d'activité est une représentation proche de l'organigramme, la description d'un cas d'utilisation par un diagramme d'activité correspond à sa traduction algorithmique. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis.

3.4.5.1 Diagramme D'activité, Authentification

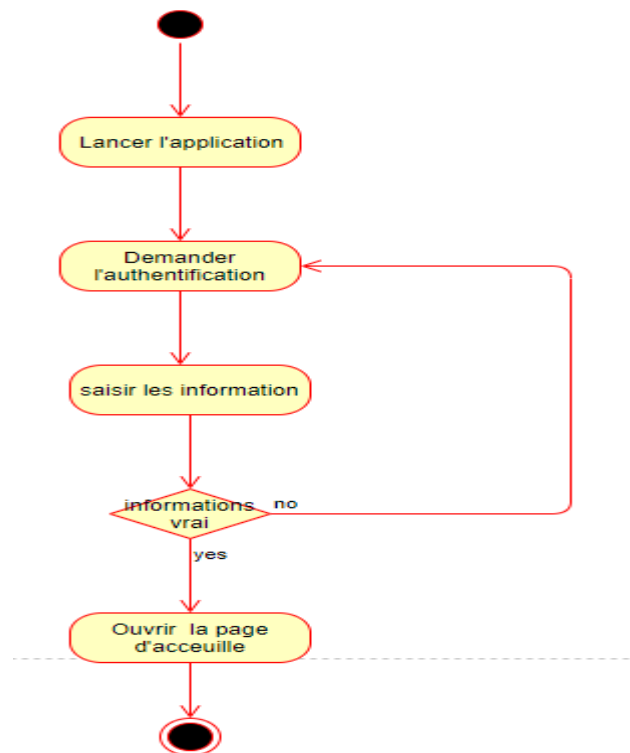


Figure 3.4 : Diagramme d'activité s'authentifier

3.4.5.2 Diagramme D'activité consulter les caméras :

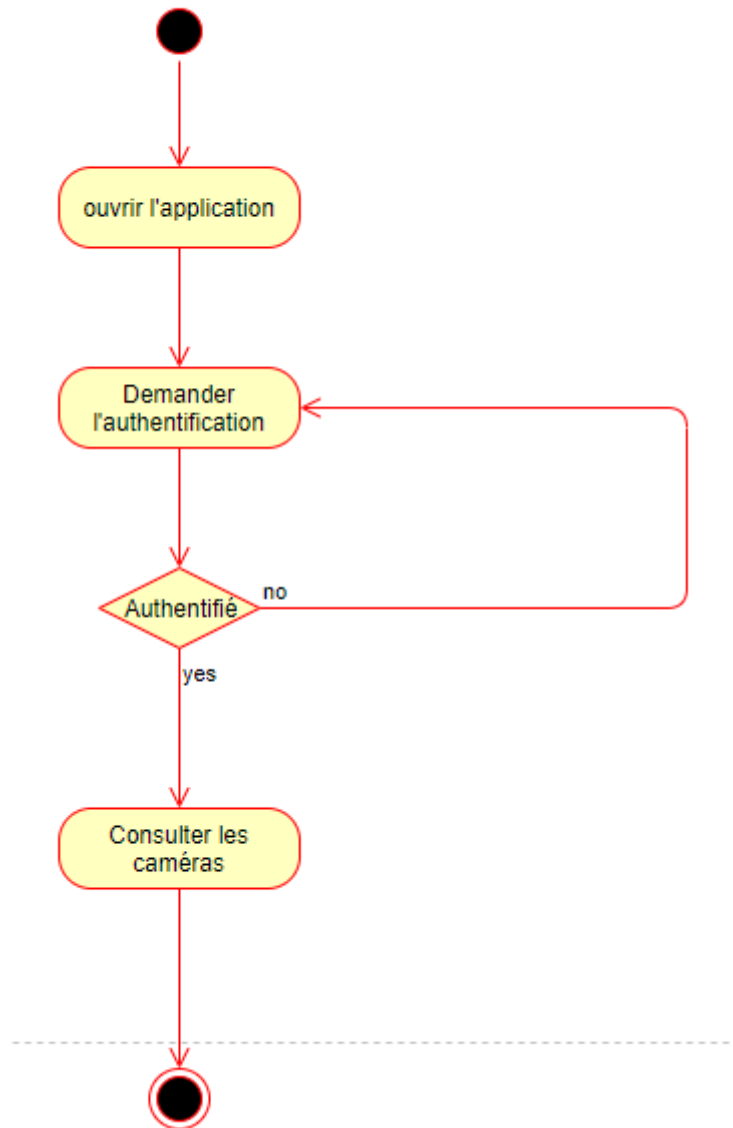


Figure 3.5 : Diagramme d'activité consulter les cameras

3.4.5.3 Diagramme d'activité publier les données :

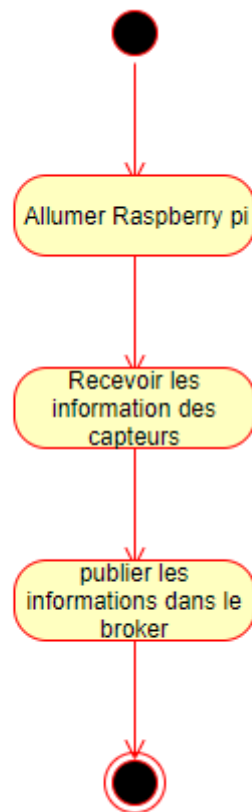


Figure 3.6 : Diagramme d'activité publier les données

3.5 Conclusion

La Conception est une étape très importante qui précède la réalisation de tout système, Dans ce chapitre nous proposons une architecture matérielle et logicielle pour notre système, Dans le prochain chapitre nous allons décrire les outils permettant la réalisation de notre système et un aperçu sur les fonctionnalités réalisées

Chapitre IV

Réalisation

4.1 Introduction

A ce stade de processus, les cas d'utilisations sont déterminés, le problème été analysé en profondeur, nous avons défini une conception mieux appropriée aux besoins du système.

Dans cette phase Nous visons de trouver des solutions informatiques et techniques pour mettre en œuvre et construire le prêt à fabriquer, je dois élaborer les différentes couches de système analysé et leurs interactions, d'abord à un niveau général puis à un niveau détaillé, en tenant compte des contraintes informatiques et techniques : langage, base de données, matériel, etc.

Selon le cycle de développement suivi, cette phase met en valeur une solution finale, qui se base sur les différents modèles de l'analyse.

Notre système est composé de deux partie principaux, une partie Matérielle qui se base sur des petits composants électroniques (capteurs, cameras, microcontrôleurs, Mobile) est chargé de la collecte d'informations (photos, détection de mouvement, incendie...) et une partie logiciel qui nous permettre de consulter les informations collectés et nous permettre de suivre les cameras en direct (Application mobile, base de données) et une partie qui représente l'intermédiaire entre l'application mobile et le système d'acquisition

4.2 Système proposé :

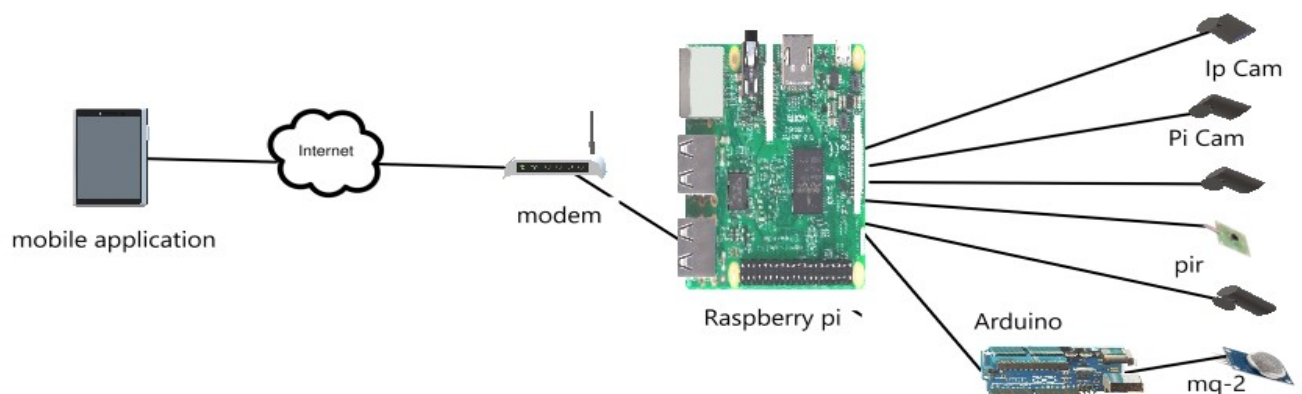


Figure 4.1 : le système à être réalisé

4.3 Les composants principaux de la solution

Langages	Python, Java, HTML
Environnements de Développement Et de déploiement	Android Studio : pour l'application mobile Thonny : Pour les scripts python
SGBD	MySQL
Matériels	Raspberry pi, Arduino, Kit Capteurs, Pi Camera, Pc, Smartphone
Protocoles de communication	MQTT, http

Tableau 06 : Les composants principaux de la solution

Le tableau de la figure 33 représente les différents outils utilisés dans la réalisation du système, pendant la réalisation du système on a utilisé Thonny comme éditeur de code pour python à cause de sa simplicité et son existence dans plusieurs systèmes d'exploitation tel que Raspbian.

Le MySQL est utilisé comme système de gestion de base de données pour plusieurs raisons tel que :

- (a) La rapidité : le MySQL est très rapide, la récupération des données se fait d'une façon très rapide.
- (b) La facilité : MySQL est très simple à utiliser que la plupart des SGBD
- (c) La sécurité : MySQL dispose d'un système de sécurité permettant de gérer les personnes et les machines pouvant accéder aux différentes bases.

Le choix des autres outils est justifié dans le chapitre 02 et le chapitre 04

4.4 Partie Matériel :

4.4.1 Le choix du Raspberry pi :

Comme il est indiqué dans le chapitre 02 notre choix du Raspberry pi n'est fortuit, mais obéit à des caractéristiques matériels et logiciel, de plus la disponibilité dans le marché algérien et son faible coût.

4.4.2 Raspberry pi comme serveur web

Pour pouvoir accéder aux cameras à distance via Wifi ou via internet on a besoin d'utiliser la Carte Raspberry Pi comme serveur Web capable d'héberger un ou plusieurs sites Internet et assurer la communication avec les clients et répondre à ses demandes grâce au protocole réseau http.

4.4.3 Collection des données des capteurs et des caméras :

Il est arrivé le moment de créer des scripts qui nous permis de récupérer les données des capteurs, le langage utiliser s'appelle **Python**. Il a un grand avantage d'être facile à apprendre tout en étant assez puissant pour créer des scripts intéressants.

4.4.4 Présentation du python :

Python est un langage de programmation créé en 1989 par Guido van Rossum, aux Pays-Bas. Le nom Python vient d'un hommage à la série télévisée Monty Python's Flying Circus dont G. van Rossum est fan. La première version publique de ce langage a été publiée en 1991. La dernière version de Python est la version 3. Plus précisément, la version 3.7 a été publiée en juin 2018. La version 2 de Python est désormais obsolète et cessera d'être maintenue après le 1er janvier 2020. Dans la mesure du possible évitez de l'utiliser(i).

La Python Software Fondation est l'association qui organise le développement de Python et anime la communauté de développeurs et d'utilisateurs.

Le langage python se caractérise par :

- (a) La multiplateforme : C'est-à-dire qu'il fonctionne sur de nombreux systèmes d'exploitation : Windows, Mac OS X, Linux, Android, iOS, depuis les mini-ordinateurs Raspberry Pi jusqu'aux supercalculateurs.
- (b) La gratuité : c'est-à-dire Vous pouvez l'installer sur autant d'ordinateurs que vous voulez (même sur votre téléphone !).
- (c) Le haut niveau : Il demande relativement peu de connaissance sur le fonctionnement d'un ordinateur pour être utilisé.

- (c) C'est un langage interprété. Un script Python n'a pas besoin d'être compilé pour être exécuté, contrairement à des langages comme le C ou le C++.
- (d) Il est orienté objet. C'est-à-dire qu'il est possible de concevoir en Python des entités qui miment celles du monde réel (une cellule, une protéine, un atome, etc.) avec un certain nombre de règles de fonctionnement et d'interactions.
- (e) Il est relativement simple à prendre en main.
- (f) Enfin, il est très utilisé en imagerie et plus généralement en Vidéosurveillance

4.4.5 Configuration des composants avec le Raspberry pi :

4.4.5.1 Pi Camera :

Pi Camera est un accessoire pour le Raspberry pi qui permet aux utilisateurs de prendre de photos et d'enregistrer des vidéos en full hd

4.4.5.2 diffusion de la vidéo en direct :

Le script-ci-dessus est composé de deux partie une partie qui permet de créer une page web « HTML » et une partie pour diffuser vidéo en direct sur la page.

```
import io
import picamera
import logging
import socketserver
from threading import Condition
from http import server

PAGE="videosurveillance"\
<html>
<head>
<title>Raspberry Pi - Surveillance Camera</title>
</head>
<body>
<center><h1>Raspberry Pi - Surveillance Camera</h1></center>
<center></center>
</body>
</html>
"""

class StreamingOutput(object):
    def __init__(self):
        self.frame = None
```

```

self.buffer = io.BytesIO()
self.condition = Condition()

def write(self, buf):
    if buf.startswith(b'\xff\xd8'):
        # New frame, copy the existing buffer's content and notify all
        # clients it's available
        self.buffer.truncate()
        with self.condition:
            self.frame = self.buffer.getvalue()
            self.condition.notify_all()
        self.buffer.seek(0)
    return self.buffer.write(buf)

class StreamingHandler(server.BaseHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/':
            self.send_response(301)
            self.send_header('Location', '/index.html')
            self.end_headers()
        elif self.path == '/index.html':
            content = PAGE.encode('utf-8')
            self.send_response(200)
            self.send_header('Content-Type', 'text/html')
            self.send_header('Content-Length', len(content))
            self.end_headers()
            self.wfile.write(content)
        elif self.path == '/stream.mjpg':
            self.send_response(200)
            self.send_header('Age', 0)
            self.send_header('Cache-Control', 'no-cache, private')
            self.send_header('Pragma', 'no-cache')
            self.send_header('Content-Type', 'multipart/x-mixed-replace;
boundary=FRAME')
            self.end_headers()
            try:
                while True:
                    with output.condition:
                        output.condition.wait()
                        frame = output.frame
                    self.wfile.write(b'--FRAME\r\n')
                    self.send_header('Content-Type', 'image/jpeg')
                    self.send_header('Content-Length', len(frame))
                    self.end_headers()
                    self.wfile.write(frame)
                    self.wfile.write(b'\r\n')
            except Exception as e:
                logging.warning(
                    'Removed streaming client %s: %s',
                    self.client_address, str(e))
        else:
            self.send_error(404)
            self.end_headers()

class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):
    allow_reuse_address = True
    daemon_threads = True

with picamera.PiCamera(resolution='1200x600', framerate=24) as camera:
    output = StreamingOutput()

```

```

#Uncomment the next line to change your Pi's Camera rotation (in
degrees)
#camera.rotation = 90
camera.start_recording(output, format='mjpeg')
try:
    address = ('192.168.1.34',8000)
    server = StreamingServer(address, StreamingHandler)
    server.serve_forever()
finally:
    camera.stop_recording()

```

Les Bibliothèques :

Threading : c'est bibliothèque qui contient des classes de gestion des processus

Http : c'est une bibliothèque de gestion des url utilisant les protocoles http et https

Les modules :

Io : est un module de gestion des flux d'entrée sortie

Pi Camera : module de pilotage de camera du Raspberry pi depuis des scripts python.

Logging : c'est un module qui fait l'enregistrement des événements liés au fonctionnement de l'application.

4.4.5.3 Le capteur de mouvement PIR :

Est un capteur passif d'infrarouge permet de détecter le mouvement d'un corps humain (en effet, la chaleur du corps produit suffisamment de lumière infrarouge pour être mesurée). Le capteur peut détecter un mouvement jusqu'à une distance maximale de 6, et ce type de capteur sortir les résultats sous forme numérique.

4.4.5.4 Détection du mouvement :

Le script ci-dessus permet de détecter le mouvement par le capteur de mouvement et publier une notification sur le broker pour être récupéré par le client, ce script est écrit en langage python

```

from gpiozero import MotionSensor
from picamera import PiCamera
import time
import paho.mqtt.client as paho

```

```

PIR= MotionSensor(4)
mqttserver = "192.168.1.34"
mqtttopic = "usine/hangar1/mouvement"
counter=0
i = 0
while True:
    PIR.wait_for_motion()
    camera.start_
    mqttc = paho.Client()
    mqttc.connect("192.168.1.34", 1883)
    mqttc.publish(mqttopic, "motion detected" + str(counter));

    time.sleep(10)
    counter= counter+1

```

Les bibliothèques

gpiozero : cette bibliothèque comprend des interfaces avec de nombreux composants simples de la vie quotidienne, ainsi que des éléments plus complexes tels que des capteurs, des convertisseurs analogique-numérique, des LED couleur, des kits de robotique, etc.

Les modules :

MotionSensor_ : ce module est pour gérer le capteur de mouvement.

Pho.mqtt.client : ce module nous permet de publier des messages dans le broker

4.5 Protocoles de communication :

Dans notre projet nous avons utilisé le protocole de **HTTP** afin de pouvoir consulter les caméras en ligne.

Le protocole **MQTT** est utilisé dans notre projet afin de faciliter la publication et la récupération des données générées par les capteurs

4.5.1 Le protocole http (Hypertext Transfer Protocol) :

Http est un protocole de communication client-serveur développé pour le World Wide Web. HTTPS (avec S pour secured, soit « sécurisé ») est la variante du HTTP sécurisée par l'usage des protocoles SSL ou TLS.

HTTP est un protocole de la couche application. Il peut fonctionner sur n'importe quelle connexion fiable, dans les faits on utilise le protocole TCP comme couche de transport.

Les clients HTTP les plus connus sont les navigateurs Web permettant à un utilisateur d'accéder à un serveur contenant les données. Il existe aussi des systèmes pour récupérer automatiquement le contenu d'un site tel que les aspirateurs de site Web ou les robots d'indexation.

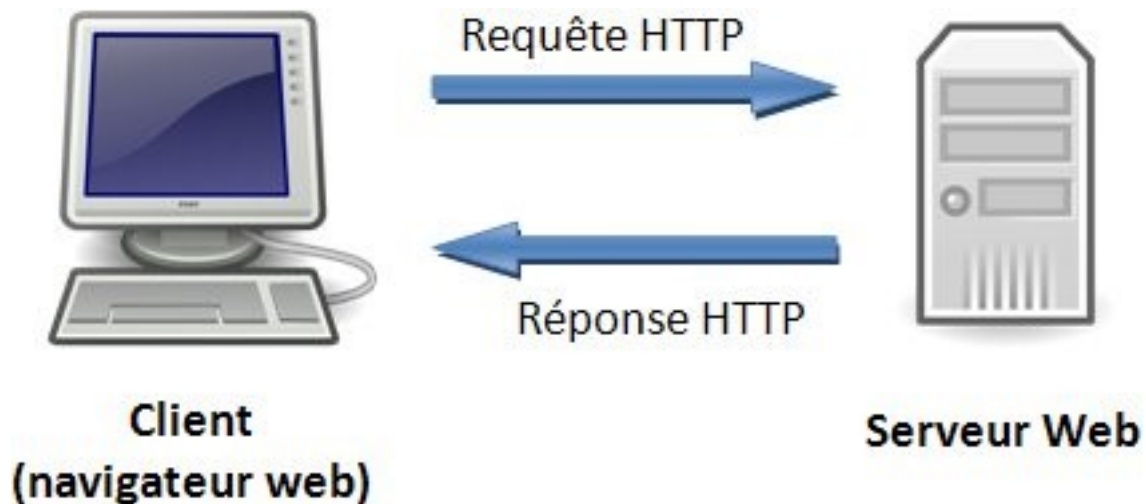


Figure 4.2 : schéma de d'une requête http

4.5.2 Le protocole MQTT (Message Queuing Telemetry Transport) :

MQTT est un protocole qui permet concrètement aux objets d'envoyer des informations sur un sujet donné à un serveur qui fonctionne comme un broker (courtier) de messages(9). Le broker pousse ces informations vers les clients qui se sont précédemment abonnés, Nous avons choisi le protocole MQTT pour de nombreuses raisons :

- (a) Le MQTT échange les données en temps réel ce qui nous aide à créer et à envoyer des notifications en même temps qu'une chose anormale détectée.
- (b) Il fonctionne pour les appareils à faible puissance, dans notre cas c'est le Raspberry pi
- (c) Un protocole facile à utiliser en raison du principe de publication / abonnement.

4.5.2.1 Principe de fonctionnement :

Contrairement au principe du client/serveur utilisé sur le Web, MQTT utilise celui de la publication/souscription : plusieurs clients se connectent à un serveur unique (appelé broker) pour soit publier des informations, soit souscrire à leur réception.

4.5.2.2 Le principe Publication/ abonnement du protocole MQTT :

Le protocole MQTT est basé sur le principe de la publication de messages et de l'abonnement à des sujets, ou "pub / sub". Plusieurs clients se connectent au broker et s'abonnent aux sujets qui les intéressent. Les clients se connectent également à courtier et publier des messages sur des sujets. De nombreux clients peuvent s'abonner aux mêmes sujets et faire avec les informations comme ils veulent. Le courtier et MQTT agissent comme une interface simple et commune pour tout ce à quoi se connecter

4.5.2.3 La qualité de service :

Pour publier les messages, Le protocole MQTT définit trois niveaux de qualité de service (QoS) :

QoS0 Livraison une fois maximum : Le message envoyé n'est pas stocké par le Broker. Il n'y a pas d'accusé de réception. Le message sera perdu en cas d'arrêt du serveur ou du client. C'est le mode par défaut.

QoS1 Livraison au moins une fois : Le message sera livré au moins une fois. Le client renvoie le message jusqu'à ce que le broker envoi en retour un accusé de réception.

QoS2 Livraison exactement une fois : Le broker sauvegarde le message et le transmettra jusqu'à ce qu'il soit reçu par tous les souscripteurs connectés.

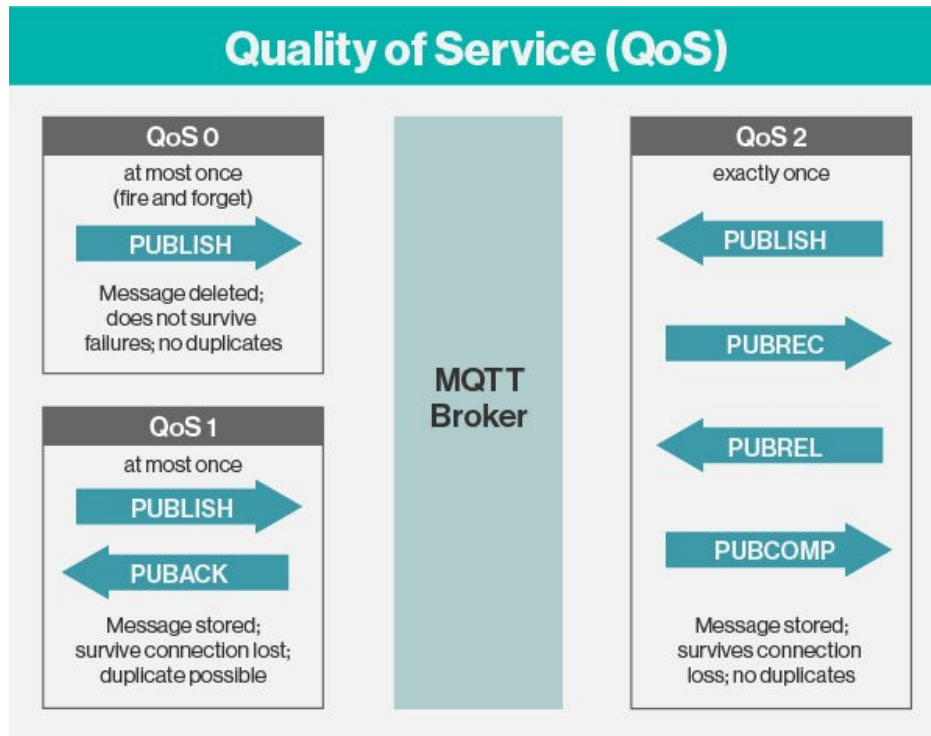


Figure 4.3 : Niveaux de Qualité de service MQTT

4.5.2.4 Avantages de MQTT :

- (a) **Légereté** : Beaucoup moins verbeux que HTTP, avec un côté asynchrone natif.
- (b) **Flexibilité** : MQTT est basé sur la couche réseau TCP/IP, utilisée par les protocoles internet dont http .On peut donc le trouver sur n'importe quelle plateforme matérielle, que ce soit un Arduino, un Raspberry Pi, un PC ou même un Cloud Microsoft Azure ou Amazon AWS.De plus, on peut faire passer n'importe quel message sur les topics, par exemple du binaire ou du JSON, selon les besoins.
- (c) **Sécurité** : De nombreuses possibilités de sécurisation sont disponibles. Mot de passe, authentification par certificats client et serveur, chiffrement SSL/TLS, listes de contrôle d'accès...C'est une qualité à ne pas négliger dans un monde où les objets connectés constituent un vecteur d'attaque important.
- (d) **Intégrité des données** : MQTT introduit la notion de qualité de service (QOS) qui permet à un client de s'assurer qu'un message a bien été transmis, avec différents niveaux de fiabilité. Il y a aussi une fonctionnalité

assez formidable : le **Will message**, qu'on pourrait traduire par "testament". Il s'agit d'un topic qui est envoyé automatiquement lorsqu'un client se déconnecte. Cela permet à n'importe quel client d'être notifié post mortem de la déconnexion d'un autre client et d'agir en conséquence.

4.5.2.5 Implémentations MQTT :

Depuis sa version 3.1.1 est maintenant un standard OASIS, (Consortium mondial qui travaille pour la standardisation de formats de fichiers ouvert). Il existe de nombreuses implémentations dans la plupart des langages de programmation (C, C++, Java, Python,). La plus connue est nommée mosquitto (avec deux t), mais des implémentations propriétaires existent aussi comme HiveMQ, qui propose de meilleures performances et une plus grande facilité de passage à l'échelle sur des serveurs positionnés dans le Cloud par exemple . MQTT est donc un protocole léger, facile à appréhender, très souple et sécurisé. Il est de plus en plus au cœur des projets IoT et supporté par la plupart des services de Cloud.

MQTT est un protocole dynamique qui continue à évoluer. Par exemple, il intègre dorénavant les Web Sockets qui lui ouvrent une porte sur le monde du Web .

4.5.2.6 Mosquitto :

Mosquitto est un courtier de messages open source (sous licence BSD) qui implémente le protocole de transport MQ Telemetry version 3.1. MQTT fournit une méthode légère d'exécution de la messagerie à l'aide d'un modèle de publication / abonnement. C'est le rend approprié pour la messagerie «de machine à machine» comme avec des capteurs de faible puissance ou des appareils mobiles tels que des **téléphones**, des ordinateurs embarqués ou des microcontrôleurs comme le **Raspberry pi**

4.5.2.7 Installer Mosquitto sur Raspberry Pi

Avant d'installer Mosquitto, il est toujours bon de mettre à jour le système

```
sudo apt-get update
```

L'installation du Broker Mosquitto se fait par la commande suivante.

```
sudo apt-get install mosquitto
```

Pour faire des tests sur le Raspberry, on peut installer les clients `mosquitto_sub` (pour souscrire), `mosquitto_pub` (pour publier des messages) et `mosquitto_passwd` (pour configurer des utilisateurs et les mots de passe).

```
sudo apt-get install mosquitto-clients
```

Pour développer avec python, on installe python-par la commande suivante :

```
sudo apt-get install mosquitto mosquitto-clients  
python-mosquitto.
```

4.6 Partie Mobile :

Il est arrivé le moment de créer une application mobile qui nous permis de consulter les caméras et de recevoir les données générées par les capteurs. Dans cette phase nous somme devant le défi de choix du système d'exploitation mobile et l'environnement de développement.

4.6.1 Application mobile :

Il existe plusieurs types de système d'exploitation pour le téléphone mobile, chaque système à ses avantages et ses limites. Il est donc très difficile de choisir la plateforme répondant bien à l'objectif d'une société ou d'une personne.

Dans cette partie nous allons présenter la raison du choix d'Android, nous aurons aussi à présenter ce système et les outils pour créer une application.

4.6.2 Le choix Android :

Android est un système d'exploitation open Source pour smartphones, PDA et autres terminaux mobiles, conçu par Android, une start-up rachetée par Google en juillet 2005. Il est donc gratuit et librement modifiable, ce qui explique d'ailleurs le nombre de mobiles qui l'utilise et ce indépendamment de leur fabricant. Cet aspect lui permet également d'être présent sur

d'autres types d'appareils possédant ce système d'exploitation tels que les téléviseurs et les tablettes. Ceci peut d'ailleurs constituer une opportunité de se détacher du seul cadre du téléphone mobile.

Le noyau Linux lui fournit une grande mémoire, la gestion de processus, le modèle de sécurité, le soutien des bibliothèques partagées...etc. Le SDK de l'Android offre complètement les APIs, avec un accès facile pour développer l'application.

Android est un système d'exploitation puissant et moderne, qui se caractérise par la simplicité et la flexibilité, cela signifie que le système est développé avec un simple langage java, et il s'adapte à beaucoup de structures différentes.

4.6.3 L'environnement de développement des applications Android

Android Studio est l'environnement de développement des applications mobiles Android. Il est basé sur IntelliJ IDEA⁷ et utilise le moteur de production Gradle⁸. Il peut être téléchargé sous les systèmes d'exploitation Windows, MacOS et Linux.

4.6.3.1 Caractéristiques d'Android studio :

Android studio se caractérise par :

- (a) Un environnement unifié où on peut développer pour tous les appareils Android.
- (b) Instant Run pour appliquer des modifications à l'application en cours d'exécution sans créer de nouveau APK.
- (c) Outils et frameworks de test étendus.
- (d) Outils de nettoyage pour détecter les problèmes de performances, de convivialité, de compatibilité des versions et autres
- (e) Prise en charge de java, kotlin, C++ et NDK.

⁷ IntelliJ IDEA également appelé « IntelliJ », « IDEA » ou « IDJ » est un environnement de développement intégré de technologie Java destiné au développement de logiciels informatiques

⁸ Gradle est un moteur de production fonctionnant sur la plateforme Java. Il permet de construire des projets en Java, Scala, Groovy voire C++.

4.6.3.2 L'architecture d'un projet en Android Studio :

- (a) App : le code de l'application
- (b) build : le code compilé
- (c) lib : les bibliothèques natives
- (d) src : les sources de l'application
- (e) main/java : les classes
- (f) main/res : vos ressources (XML, images, ...)
- (g) test : les tests unitaires
- (h) Des fichiers de configuration :
- (i) build.gradle (2 instances): règles de dépendance et de compilation
- (j) settings.gradle: liste de toutes les applications à compiler (si plusieurs)

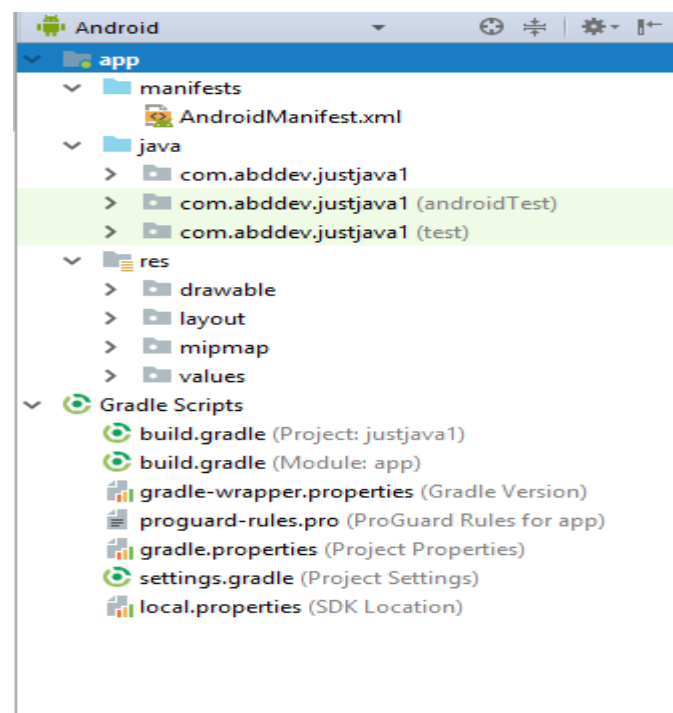


Figure 4.4 : Architecture d'un projet en Android studio

4.6.3.3 Les éléments d'une application Android :

- a. des activités (android.app.Activity): il s'agit d'une partie de l'application présentant une vue à l'utilisateur

- b. des services (android.app.Service): il s'agit d'une activité tâche de fond sans vue associée
- c. des fournisseurs de contenus (android.content.ContentProvider): permet le partage d'informations au sein ou entre applications
- d. des widgets (android.appwidget.): une vue accrochée au Bureau d'Android
- e. des Intents (android.content.Intent): permet d'envoyer un message pour un composant externe sans le nommer explicitement
- f. des récepteurs d'Intents (android.content.BroadcastReceiver): permet de déclarer être capable de répondre à des Intents
- g. des notifications (android.app.Notifications): permet de notifier l'utilisateur de la survenue d'événements

4.6.4 Présentation de l'application :

L'application mobile représente une partie très importante dans notre système, elle permet aux utilisateurs de consulter les caméras en direct, recevoir les notifications en cas de mouvement ou d'incendie, elle permet aussi de publier des messages de diffusion pour les autres utilisateurs de l'application.

Cette application connecte au microcontrôleur localement via wifi ou à un autre réseau via internet.

Notre application est composée de 4 activités principales : Login(Authentication), inscrire, menu principale, caméras.

Login (s'authentifier) :

L'utilisateur de cette application doit d'abord s'authentifier. Un nom d'utilisateur et un mot de passe sont nécessaires pour des raisons de sécurité.

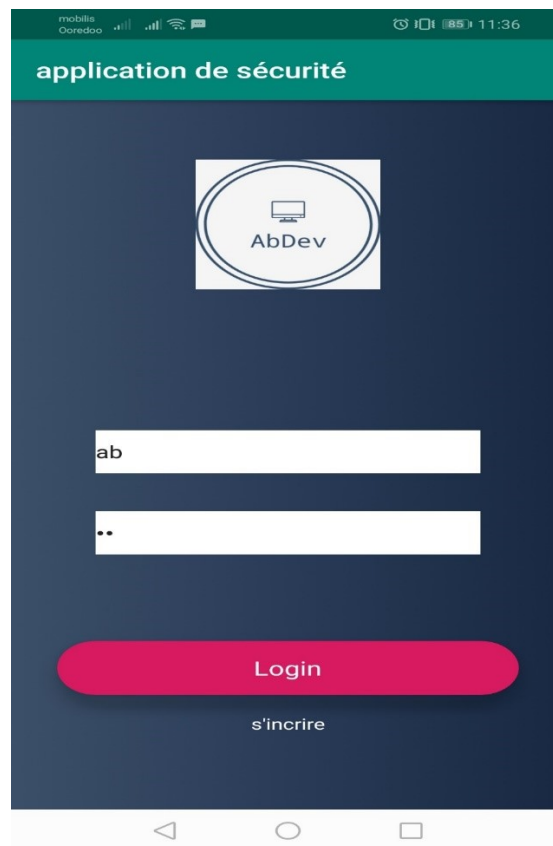


Figure 4.5 : Authentification

Créer nouvel utilisateur :

L'administrateur peut créer de nouveaux utilisateurs de son application

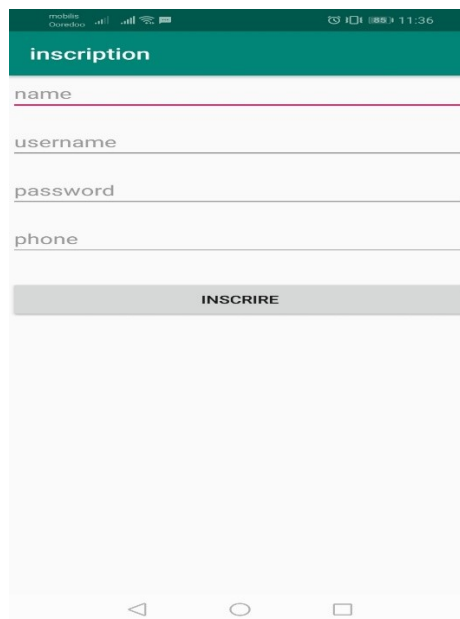


Figure 4.6 : Créer nouvel utilisateur

Page principale : dans cette activité permet aux utilisateurs de publier des messages de diffusion et consulter les notifications.



Figure 4.7 : page principal

Système de suivi : Cette activité permet à l'utilisateur de sélectionner la camera.



Figure 4.8 : Système de suivi

Camera : cette activité permet à l'utilisateur de consulter la camera sélectionnée da la l'activité principal.

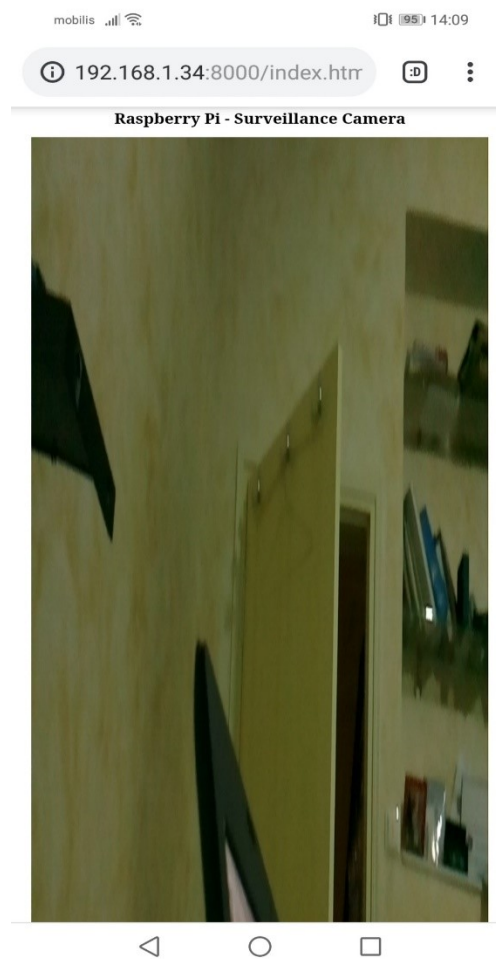


Figure 4.9 : Camera

4.7 Mesures de performances :

Après la conception et la réalisation d'un système il est important de mesurer les performances du système et voir es ce que le système atteint les objectifs tracé au début, on a appliqué des tests sur notre système dans plusieurs endroits , le diagramme de la figure 4.10 nous montre les résultats des tests.

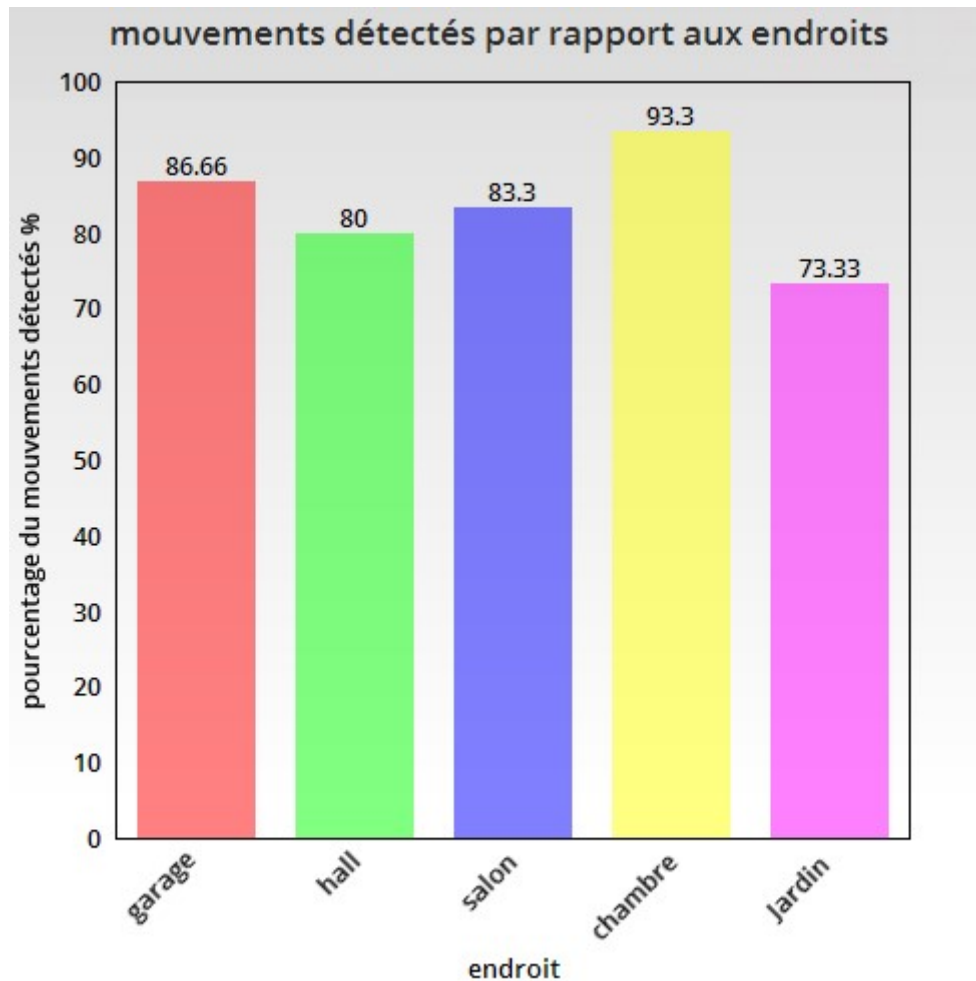


Figure 4.10 : pourcentage des mouvements détectés par rapport aux endroits

Le diagramme de la figure 4.10, représente les résultats des tests de détection de mouvement, dans ce test on a installé notre système dans plusieurs endroits, dans chaque endroit on a appliqué 30 mouvements, dans une chambre de 9 m² le système a détecté 28 mouvements sur 30 mouvements appliqués, dans le hall, le garage le salon a détecté de 24 à 26 mouvements sur 30 mouvements appliqués, ces dernier ont une superficie qui varié entre 25 et 30 m², pour le jardin le système a détecté 22 mouvements sur 30 mouvements appliqués et ça revient à la superficie qui dépasse 40 m² plus les obstacles tel que les arbres les piliers, etc.

Les résultats trouvés montre que notre système est très performant sur les endroits fermés plus que les endroits ouverts, chose qui est logique à cause des bruits de l'extérieur, des obstacles qu'on trouve, un deuxième facteur qui réduit la performance de notre système de détection est la superficie de la zone couverte, plus que la superficie augmente le nombre de détections réduit, le cas de la chambre et le salon presque 10% des mouvements ne sont pas détecter.

4.7 Conclusion :

Dans ce chapitre nous avons présenté les différents composants du notre système, puis nous avons définis les outils utilisé dans réalisation du notre système (Matériels électroniques, langages de programmation, environnements de développement, etc.). à la fin de chapitre nous avons fait une petite présentation à notre application mobile, quelque morceaux de codes important pour le développement d'une application pareil et les résultats de quelques tests afin de mesurer les performances du système sont aussi présentés.

Conclusion Générale

Notre projet consiste à concevoir et réaliser un système vidéosurveillance automatique afin d'augmenter le niveau de sécurité d'un endroit bien précis (bureau, hangar de stockage...).

Nous avons convergé la technologie des applications mobile avec l'outil de la vidéosurveillance pour pouvoir satisfaire un large éventail des besoins d'utilisateurs, notamment le besoin de la sécurité et le confort des personnes.

Au début nous avons cité les moyens matériels et logiciels nécessaire pour la réalisation du projet. on a défini pour cela, la carte électronique Raspberry pi du côté matériel et la programmation sous Android du côté logiciel. Dans la phase conception, nous avons commencé par la définition des besoins et la spécification des exigences, puis nous avons modélisé les fonctionnalités identifiées à l'issue de cette spécification utilisant UML afin de rendre notre système cohérent et répond aux objectifs visés. Enfin nous somme passé à la partie réalisation du projet, commençant par le développement du système d'acquisition utilisant quelques outils électroniques (microcontrôleur, capteurs) et le langage programmation Python, puis nous avons développés une application mobile qui permet de consulter les caméras en ligne et recevoir des notifications en cas de prises de vols.

Ce travail nous a permis d'apprendre énormément de choses concernant le développement des systèmes informatique (développement de deux système hétérogènes et assurer la communication autonome entre eux), ainsi cette chance nous a permis de découvrir de nouveaux protocoles de communication tel que MQTT, XMPP, également nous avons appris à manipuler toute une panoplie d'outils : Thonny, PhpMyAdmin, Android studio et quelques langages de programmation : PHP, html, CSS, java, python, XML.

Comme perspective, on aimerait bien ajouter des drones qui prennent des charges utiles pour remédier aux problèmes découverts par ce qui convient.

Bibliographie :

1. Lakhdar Imlouli, Guide de la vidéosurveillance, Les éditions du NE, aout 2014
2. Lacene Beddiaf, Vidéosurveillance, Principes et technologies, DUNOD, février 2013.
3. Alain Bauer et François Freinet, Vidéosurveillance et vidéo protection, Puf, octobre 2015.
4. Sylvain Hébuterne, Développez une application Android, programmation en Java sous Android, Studio, ENI France, octobre 2017.
5. Dogan Ibrahim, Android App For electronic designer, Elektor, janvier 2019.
6. Mark Murphy, L'art du développement Android, PEARSON, juin 2009.
7. Eben Upton, Raspberry Pi Le guide de l'utilisateur, DUNOD, avril 2017.
8. Gérard swinnen, Apprendre à programmer avec python 3, EYROLLES, février 2012.
9. Benhmida Amel, Vers une nouvelle architecture de vidéosurveillance basée sur la scalabilité, orientée vers l'application, Université de Bordeaux

Webographie

1. https://iot.goffinet.org/iot_protocoles.html, 12/09/2018
2. <https://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operating-system/>, 09/05/2019
3. <https://www.statista.com/statistics/218089/global-market-share-of-windows-7/>, 09/05/2019
4. <https://www.draw.io/>, 27/04/2019
5. <https://fr.statista.com/statistiques/671769/nombre-d-applications-sur-itunes-app-store/>, 09/05/2019
6. <https://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql/1975880-gestion-des-utilisateurs>, 03/07/2019
7. https://projetsdiy.fr/mosquitto-broker-mqtt-raspberry-pi/#Quel_Broker_MQTT_open-source_choisir, 15/12/2018
8. <http://www.univ-orleans.fr/lifo/Members/Jean-Francois.Lalande/enseignement/android/cours-android.pdf>, 20/01/2019
9. <http://www.theses.fr/2017LORIS436>, 25/05/2019
10. <https://www.agilistechlabs.com/impact-mobile-applications-customers/>, 25/01/2019
11. <http://guides.mistergooddeal.com/guide-achat/smartphone-systeme-exploitation.html>, 30/04/2019
12. <https://www.appstud.com/fr/guides/agence-mobile/app065/>, 02/05/2019

Résumé

Le besoin de renforcer le niveau de sécurité se fait de plus en plus ressentir dans les différents domaines d'activités, la plus part des propriétaires souhaitent accroître la sécurité en protégeant les vies et les biens, un des moyens très utilisés est la **Vidéosurveillance**.

Dans le but de faciliter la vie et de faire éliminer les difficultés autours de notre vie, nous avons adopté la technologie de **Vidéosurveillance** et des **applications mobiles** pour créer un système de vidéosurveillance automatique

Ce système est composé d'une partie matérielle et une partie logicielle, la partie matérielle a pour but d'acquérir des vidéos et de détecter tous mouvement. Le coté logiciel, on a développé une application mobile qui fonctionne sous Android, cette dernière a pour le but de consulter les vidéos en ligne et de recevoir les notifications.

Mots clés : Application mobile, Vidéosurveillance, détection du mouvement, Android

ملخص

بما اننا في زمن التطور التكنولوجي المتزايد، هنالك دائما شعور بالحاجة إلى زيادة مستوى الأمن في مختلف المجالات، حيث يرغب الناس في تطوير مستوى الأمن لحماية الأرواح والممتلكات، ومن أكثر الوسائل شيوعًا المراقبة عن طريق الفيديو.

ومن أجل تسهيل الحياة وإزالة الصعوبات المحيطة بنا، جمعنا بين تقنيتي المراقبة بالفيديو وتطبيقات الهاتف المحمول لإنشاء نظام مراقبة فيديو تلقائي.

يتكون هذا النظام من جانب مادي يشمل أجهزة الاستقبال والمراقبة، وجانب البرمجيات يتضمن تطبيق الهاتف المحمول، والذي يمكننا من الاطلاع على الفيديوهات ومتابعة واستقبال الاشعارات في حصول طارئ ما.

الكلمات المفتاحية: تطبيقات الجوال ، كشف الحركة، المراقبة بالفيديو، أندرويد.

Abstract

The need to increase the security's level is being dealt with in various areas of activity, most clients wants more security to protect their lives and properties, and one of the most popular means is video surveillance.

In order to make our life easier and eliminate difficulties around our lives, we have adopted video surveillance technology and mobile applications to create an automatic video surveillance system.

This system consists of a hardware part and a software part. The hardware part has the purpose of video acquisition and detect all motions, for the software part we had developed a mobile application that runs on Android, the latter has the purpose of viewing online videos and receiving notifications.

Key words : Mobile App, CCTV, Motion Detection, Android

