

**UNIVERSITE KASDI MERBAH OUARGLA**

**Faculté des Science et Technologie et des Science de la matière**  
**Département de mathématique et d'Informatique**



**Mémoire**

Domaine : Mathématique et Informatique

Filière : Informatique

Spécialité : Informatique Fondamentale

Présenté par : Telli Fatiha et Benjeddou Khadija

**Une application mobile :**  
**Une télécommande Bluetooth d'une souris**  
**optique**

Soutenu publiquement

Le : 29/9/2013

Devant le jury :

M. BELOUAAR Houcine

Président

UKM Ouargla

M. DJEDIAI Hmida

Encadreur

UKM Ouargla

M. ZGA Adel

Examineur

UKM Ouargla

**Année Universitaire : 2012 /2013**

# Remerciements

Avant tout, Nous remercions **الله** *ALLAH* tout puissant de nous avoir donné la force et le courage pour terminer ce travail.

Mes remerciements s'adressent particulièrement au Mr. DJDIAI Hmida, Pour tous les conseils.

Mes remerciements s'adressent particulièrement au Fekiar Mohammed el amine, Pour toute l'aide.

Nous n'oublions pas non plus nos enseignant, qui tout au long du cycle d'études à l'Université d'Ouargla.

Nous adressons une particulièrement affective à nos Amis de l'Université d'Ouargla qui rendu agréable nos longue années d'études.

Un grand merci également à tous ceux qui ont contribué, de près ou de loin, à l'aboutissement de ce travail.

Fatiha &Khadidja

# Dédicaces

*Je dédie ce travail*

*A ma mère*

*A mon père*

*En témoignage de mon affection et de ma profonde*

*Gratitude pour leur soutien moral et financier et leurs*

*Encouragements;*

*A mes sœurs et frères ...*

*A toute ma famille proche soit elle ou lointaine*

*A tous mes ami(e)s*

*A tous ce qui m'aiment et que je l'aime...*

Fati ha

# Dédicaces

*Je dédie ce travail*

*A ma mère*

*A mon père*

*En témoignage de mon affection et de ma profonde*

*Gratitude pour leur soutien moral et financier et leurs*

*Encouragements;*

*A mes sœurs et frères ...*

*A toute ma famille proche soit elle ou lointaine*

*A tous mes ami(e)s*

*A tous ce qui m'aiment et que je l'aime...*

Khadi dja

## Sommaire

### Introduction Generale

#### Chapitre I

<b>Les Applications Mobile</b> .....	<b>14</b>
<b>1. Introduction</b> .....	<b>6</b>
<b>2. Les Applications Mobiles</b> .....	<b>7</b>
<b>2.1 Definition</b> .....	<b>7</b>
<b>2.2 Caracteristiques D'application Mobile</b> .....	<b>8</b>
<b>2.4 Les Differents Types D'applications Mobiles</b> .....	<b>12</b>
<b>A) Application Native</b> .....	<b>12</b>
<b>B) Application Web Ou Web App</b> .....	<b>12</b>
<b>C) Applications Hybrides</b> .....	<b>12</b>
<b>2.5 Comparaison Entre Application Native Et Web App[4]</b> .....	<b>14</b>
<b>2.6 Elements De Comparaison Entre Les Applications Mobile [6]</b> .....	<b>15</b>
➤ <b>Couts De Mise En Œuvre</b> .....	<b>15</b>
➤ <b>Qualite, Rapidite Des Applications</b> .....	<b>15</b>
➤ <b>Publication Et Mises A Jour</b> .....	<b>15</b>
➤ <b>Monetisation</b> .....	<b>15</b>
<b>2.7 Statistiques Sur Le Marche</b> .....	<b>16</b>
<b>3. Les Technologies D'accès Sans Fil</b> .....	<b>18</b>
<b>3.2 Les Categories De Reseaux Sans Fil</b> .....	<b>18</b>
<b>3.2.1 Wwan</b> .....	<b>19</b>
<b>3.2.1.1 LES DIFFERENTES GENERATIONS DE TELEPHONIE MOBILE [10]</b> .....	<b>19</b>
<b>3.2.2 Wman</b> .....	<b>20</b>
<b>3.2.2.1 Wi Max</b> .....	<b>20</b>
<b>3.2.3 Wlan</b> .....	<b>20</b>
<b>3.2.3.1 Wifi</b> .....	<b>21</b>
<b>3.2.4 Wpan</b> .....	<b>21</b>
<b>3.2.4.1 Bluetooth</b> .....	<b>22</b>
<b>4. CONCLUSION</b> .....	<b>25</b>

#### Chapitre II

<b>Technologies Et Outils Pour Le Developpement Des Applications Mobiles .....</b>	<b>26</b>
<b>1. Introduction.....</b>	<b>27</b>
<b>2. Les Systemes Exploitation Mobile .....</b>	<b>28</b>
<b>2.1 Definition .....</b>	<b>28</b>
<b>2.2 ROLES DU SYSTEME D'EXPLOITATION .....</b>	<b>29</b>
<b>LES ROLES DU SYSTEME D'EXPLOITATION SONT DIVERS :.....</b>	<b>29</b>
• <b>Gestion Des Entrees/Sorties :.....</b>	<b>29</b>
• <b>Gestion Du Processeur :.....</b>	<b>29</b>
• <b>Gestion De La Memoire Vive : .....</b>	<b>29</b>
• <b>Gestion De L'execution Des Applications : .....</b>	<b>29</b>
• <b>Gestion Des Droits :.....</b>	<b>29</b>
• <b>Gestion Des Fichiers :.....</b>	<b>30</b>
• <b>Gestion Des Informations : .....</b>	<b>30</b>
<b>2.3 COMPOSANTES DU SYSTEME D'EXPLOITATION .....</b>	<b>30</b>
<b>2.4 Ios D'apple.....</b>	<b>31</b>
<b>L'app Store.....</b>	<b>31</b>
• <b>Caracteristique.....</b>	<b>31</b>
• <b>Versions .....</b>	<b>31</b>
<b>2.5 Android De Google.....</b>	<b>32</b>
• <b>Googaleplay .....</b>	<b>32</b>
• <b>Caracteristiques .....</b>	<b>32</b>
• <b>Versions.....</b>	<b>33</b>
<b>2.6 Windows Ce De Microsoft .....</b>	<b>34</b>
• <b>Microsoft Corporation.....</b>	<b>34</b>
• <b>Caracteristiques .....</b>	<b>34</b>
• <b>Versions .....</b>	<b>34</b>
<b>2.7 Symbian De Nokia.....</b>	<b>36</b>
• <b>Nokia Corporation .....</b>	<b>36</b>
• <b>Caracteristiques .....</b>	<b>36</b>
• <b>Architecture.....</b>	<b>36</b>

• Versions .....	37
3. Les Plateformes .....	39
3.1 Le.Net Compact Framework De Microsoft .....	39
3.1.1 L'architecture De La Plateforme .Net Compact Framework. ....	40
3.2 Java 2 Micro Edition J2me .....	41
3.2.1Architecture De J2me .....	42
4 .Conclusion .....	45

### Chapitre III

La Conception .....	46
1. Introduction.....	46
Etape 1 : Analyse Des Besoins.....	46
A) Services : .....	46
B) Contraintes .....	46
Etape 2 : La Conception.....	46
1. Conception Globale .....	47
2. Conception Detaillee : .....	50
3. Introduction Uml.....	56
3.1 Les Diagrammes Structurels .....	57
3.1.1 Les Diagrammes De Classe .....	57
3.1.2 Le Diagramme De Packages.....	60
3.1.4 Le Diagramme De Composant .....	60
3.2les Diagrammes Comportementaux.....	61
3.2.1le Diagramme De Cas D'utilisation.....	61
3.2.2 Le Diagramme Global D'interaction .....	62
3.2.3le Diagramme De Temps .....	62
3.2.4le Diagramme D'activite.....	62
3.2.5 Diagramme D'etats-Transitions.....	65
3.2.6 Diagramme De Sequence.....	65
3.2.7les Diagrammes De Communication.....	65
4. Points Forts Et Points Faibles.....	66

### Chapitre VI

<b>L'implementation .....</b>	<b>68</b>
<b>1. Introduction.....</b>	<b>70</b>
<b>2. Environnement De Developpement Partie Client: Netbeans.....</b>	<b>70</b>
<b>3. Langage De Programmation : Java.....</b>	<b>70</b>
<b>4. Technologie Adoptee : Rme .....</b>	<b>71</b>
<b>8. Communication Entre Telephone Portable (Client) Et L'ordinateur (Server): .....</b>	<b>74</b>
<b>8.1 Initialisation De Stack.....</b>	<b>74</b>
<b>8.2 La Gestion De L'equipement .....</b>	<b>75</b>
<b>8.3 La Decouverte De L'equipement .....</b>	<b>77</b>
<b>8.4 La Decouverte Du Service.....</b>	<b>78</b>
<b>9. La Communication : Transmission Entre Le Telephone Portable Et L'ordinateur</b>	<b>79</b>
<b>10. Programmation Pour Le Controle Automatique De La Souris.....</b>	<b>80</b>
<b>11. Conclusion .....</b>	<b>80</b>
<b>Conclusion General.....</b>	<b>82</b>
<b>Bibliographie .....</b>	<b>84</b>

## Liste de Figures

Figure 1: i phone et ces application en détail.....	7
Figure 2 : domaines d'application mobile .....	9
Figure 3 : tableau expliquer les domaines .....	11
Figure 4 : Comparaison entre application native et Web App.....	14
Figure 5 : Les Application mobiles utilisées le plus souvent .....	16
Figure 6 : Facebook et Google au top des applications mobiles.....	17
Figure 7 : Wireless networks categories.....	18
Figure 8 : Les différentes générations de téléphonie mobile .....	19
Figure 9 :Le système d'exploitation.....	28
Figure 10 : version d'Android.....	33
Figure 11 : les caractéristiques de Windows CE.....	36
Figure 12 : Version de Symbian .....	39
<b>Figure 13 : l'architecture de la plateforme .NET Compact Framework.....</b>	<b>40</b>
Figure 14 : Architecture de J2ME.....	43
Figure 15 : CDC& CLDC sur L'Architecture de J2ME.....	44
Figure 16 : La bibliothèque de J2ME.....	45
Figure 17 : Architecture générale du système .....	47
Figure 18 : Architecture générale deuxième du système.....	48
Figure 19 : Schéma de L'utilisateur .....	50
Figure 20 : Schéma pour le terminal mobile.....	53
Figure 21 : Schéma pour le terminal PC.....	54
Figure 22 : Les diagrammes de classe.....	59
Figure 23 :Le diagramme de cas d'utilisation .....	62
Figure 24 : Cycle de vie d'une MIDLET .....	73
<b>Figure 25 : APIs de Bluetooth en architecture de J2ME .....</b>	<b>74</b>

## **Introduction générale**

Aujourd'hui, le téléphone mobile n'est pas seulement utilisé pour la communication, et l'échange des messages courts comme autre fois, De nouveaux usages sont apparus tels que les jeux, la lecture audio, etc..... Sur un plan plus pratique, Le téléphone mobile dispose des variétés d'utilisations et l'appareil est devenu plus utile que jamais auparavant. L'essor du mobile a connu une évolution considérable en particulier dans les pays en développement. Les réseaux de télécommunication mobiles sont également en expansion : le nombre d'utilisateurs de mobile est en constante progression et la couverture territoriale est largement répandue.

Grâce à la généralisation des téléphones portables tactiles à écrans larges ainsi qu'au développement des logiciels et des réseaux, les applications mobiles sont capables de satisfaire un large éventail de besoins, sans oublier un développement technologique continu qui en fait un outil encore plus essentiel.

Malgré l'accroissement de la puissance de ces appareils, leur capacité reste limitée face à celle des ordinateurs portables et PC. Il faut donc prendre en considération ces limitations qui dépendent de leur taille (écran), de la puissance des processeurs et de la capacité de mémorisation . Sur le plan technologique on distingue plusieurs aspects qui varient en fonction des gammes et des versions des équipements :

- les interfaces de connectivité sont multiples: Bluetooth, WIFI, GSM,....
- les plateformes de développement sont hétérogènes: Apple, J2ME, Android et symbian, Windows CE.

L'objectif de notre projet présenté dans ce mémoire est l'étude des applications mobiles et leurs techniques et outils de développement et Construire une application réelle en fonction d'un processus à l'aide de technologie J2ME et les outils correspondants. C'est une application qui permet de télécommander une souris de micro portable à partir un équipement mobile.

Notre mémoire est organisé en quatre chapitres, On a commencé par une introduction générale

### **Chapitre I : Les Applications mobile**

Dans ce chapitre on a fait une représentation de l'application mobile, on a fait la notion de l'application mobile puis ses Caractéristiques, statistique sur marché et on a fait aussi Les technologies d'accès sans fil.

### **Chapitre II: Technologies et Outils pour le développement des applications mobiles**

Dans ce chapitre nous avons présenté les systèmes exploitation de mobile et les plateformes de développement.

### **Chapitre III : Conception de base**

Dans ce chapitre nous avons présenté la conception de notre application et nous utilisons la méthode UML.

### **Chapitre VI: L'implémentation**

Dans ce chapitre on a présenté l'implémentation.





Chapitre I

# Les Applications Mobile

## 1. Introduction

Aujourd'hui, les applications mobiles ont pris une place importante dans notre vie quotidienne. Il en existe plusieurs types, Les applications mobiles couvrent une grande variété de domaines, ces applications est considère l'étendue de la communication avec le monde extérieur. Il y a beaucoup d'applications mobiles qui sont encore les applications de téléphone unique, comme un éditeur de note ou un jeu ...., sans aucune communication avec le monde extérieur.

Dans ce chapitre on a fait la notion de l'application mobile et ses Caractéristiques, types, Statistiques sur le marché. Et on a fait Les technologies d'accès sans fil : Les catégories de réseaux sans fil, WWAN (GSM, GPRS, UMTS...), WMAN (Wi MAX),WLAN (Wifi), WPAN (Bluetooth).

## 2. Les applications mobiles

### 2.1 Définition

➤ Une application mobile est un logiciel applicatif développé pour être installé sur un appareil électronique mobile, tel qu'un téléphone portable, un « smartphone ». [1]

➤ Une application mobile est un programme téléchargeable de façon gratuite ou payante et exécutable à partir du système d'exploitation du téléphone. [2]

Ce petit logiciel s'appuie d'une manière générale sur le principe de widgets que nous connaissons sur nos ordinateurs.

Pour télécharger une application sur un téléphone mobile, il existe différentes possibilités:

- transfert depuis un ordinateur via un câble de connexion,
- à partir d'un service mobile,
- via une boutique logicielle accessible depuis un téléphone mobile (App Store d'Apple, Windows Market Place, Nokia OVI, AndroidMarket, etc.)
- le cas échéant l'application est dite native ; elle est déjà dans le téléphone lors de l'achat du téléphone (l'opérateur ou le fabricant l'a ajouté comme fonction de base) [3]

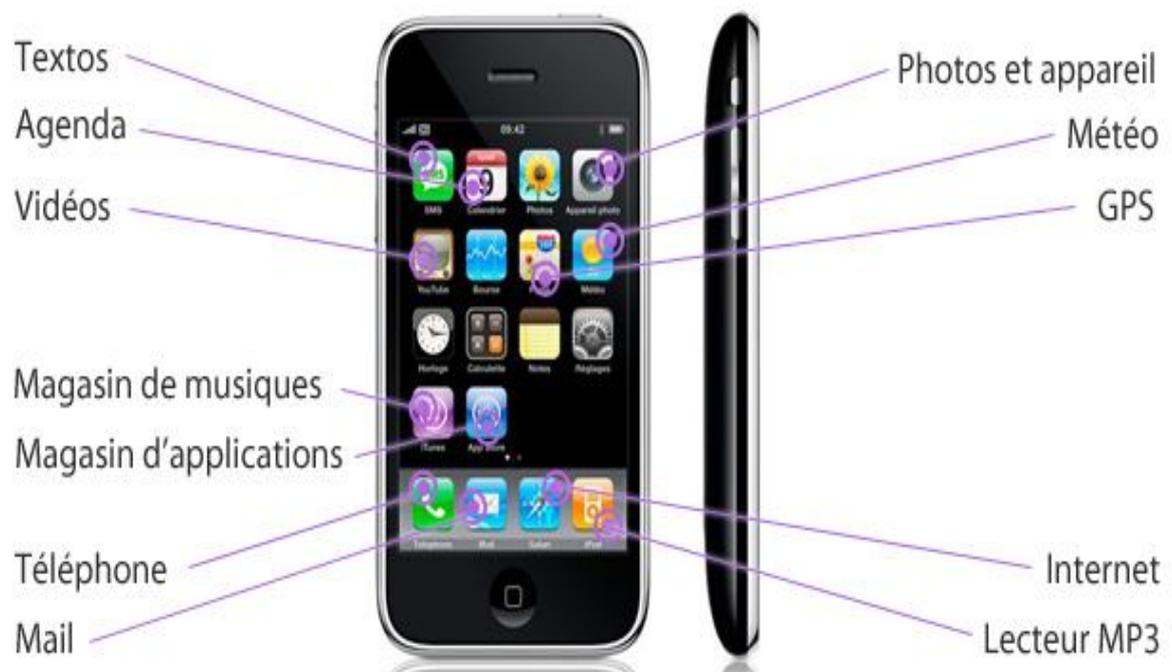


Figure 1: i phone et ces application en détail

## 2.2 Caractéristiques d'application mobile

Des contraintes techniques qu'il est nécessaire de prendre en compte lors de la conception d'une application mobile :

- Tailles d'écrans variables, pouvant dans certains cas être assez réduite
- Possibilité limitée de saisie de données
- Puissance du processeur, pouvant être limité sur les premiers smartphones
- Tailles de la mémoire pouvant varier
- Autonomie du smartphone
- Débits variables de la bande passante Internet.

Outre ces aspects techniques, il est également important de prendre en compte l'ergonomie de l'application mobile à réaliser, et c'est un point crucial à ne pas négliger

Une application mobile doit respecter certaines règles :

- Utiliser des images petites et légères
- Utiliser des éléments facilement accessibles
- Maitriser l'utilisation du javascript pour économiser la batterie
- Adapter le mode de saisie des informations. [4]

Les Solutions mobiles adaptées aux exigences et contraintes des clients.

Mobile repose sur cinq domaines de compétences :

- **Techniques** iOS, Android, Windows, Objective-C, Java, C#, XAML, ...
- **Architecture** : Performance, fiabilité, Intégration, sécurité, Evolutivité ,...
- **Design & Ergonomie**
- **Fonctionnelles** spécifiques à la mobilité
- **Démarche Projet** et aux conseils relatifs aux projets de mobilité.

### 2.3 Domaines d'application mobile

Avec les possibilités matérielles incorporées aux terminaux (caméra, GPS, gyroscope, ...), les applications Smartphones et Tablettes peuvent intégrer des fonctionnalités spécifiques et dédiées pour les utilisateurs, permettant ainsi d'enrichir le spectre fonctionnel et imaginer des usages non couverts jusqu'à présent par les systèmes d'information

- Géo-localisation, Itinéraires
- Scan de Code barre, Flash, QR Code
- Réalité augmentée
- M-commerce, Paiement mobile
- Push et notification
- Gestion de documents, dématérialisation, Workflow
- Analyse d'Audience
- Gestion et Sécurisation de parc et de déploiement de terminaux mobiles.
- ... [5]



Figure 2 : domaines d'application mobile

Le tableau suivante expliquer les domaines [5] :

<p><b>Application Mobile Multi plateforme</b></p>  <p><b>Site Mobile adaptée</b></p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Web mobile</li> <li><input type="checkbox"/> Responsedesign</li> </ul> 	<p><b>M-BUSINESS</b></p> <p>Apporter une puissante relation commerciale à l'aide d'appareils mobiles</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Personnalisation et fidélisation (carte de fidélité numérique, ...)</li> <li><input type="checkbox"/> Marketing géo-localisé (push marketing, diffusion de produits, ...)</li> <li><input type="checkbox"/> Aide à la vente (tags, QR code, réalité augmentée)</li> <li><input type="checkbox"/> Comparaison et suggestion de produits</li> <li><input type="checkbox"/> Achat à distance, solutions de paiement</li> <li><input type="checkbox"/> Recommandation sociale (commentaires, avis, évaluations de consommateurs, ...)</li> </ul>	<p><b>SCAN DE CODES</b></p> <p>Les Tags de type QR codes, Code Barre, Flash Code et autres peuvent contenir de l'information délivrée par les Smartphones pour de nombreux usages :</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Identification (inventaire, positionnement, jeux,...)</li> <li><input type="checkbox"/> Apport d'information complémentaire (texte, image, vidéo, ...)</li> <li><input type="checkbox"/> ...</li> </ul> 	<p><b>STORE PRIVE, MDM</b></p> <p>Les plateformes de références permettent la diffusion gratuite ou payante d'applications publiques. Afin de répondre au contexte d'applications internes, conçoit des stores privés, sécurisés et personnalisés pour restreindre la diffusion des applications d'une entreprise à ses employés, partenaires, ...</p> <p>De plus, en y intégrant les outils MDM (Mobile Device Management), ces entreprises peuvent sécuriser, monitorer, gérer et contrôler les terminaux déployés.</p>
--	---	--	---

<p><b>REALITE AUGMENTEE</b>          Proposer un nouveau moyen de visualisation informationnelle via des interfaces utilisateur à la fois simples et enrichies et par la superposition, en temps réel, d'un modèle virtuel 3D ou 2D à la perception naturelle de la réalité. L'intérêt consiste notamment à :</p> <ul style="list-style-type: none"> <li>□□ Actionner un levier à la prise de décision (par exemple l'acte d'achat)</li> <li>□□ Aide à la décision pour l'utilisateur (orientation, indicateur...)</li> <li>□□ Apporter des éléments de jeux, d'attraction dans son espace réel.</li> </ul>	<p><b>NOTIFICATIONS, PUSH</b>          Le Push ou Notification est un envoi de messages, de sons à propos d'un événement interne à une application même si celle-ci n'est pas ouverte. C'est donc le serveur qui pousse des informations vers les Smartphones et Tablettes, éventuellement en fonction de critères liés au terminal, à l'utilisateur, la localisation, ...          Ce mécanisme de notification est une alternative et/ou complément aux possibilités du principe des SMS.</p> 	<p><b>GEO-LOCALISATION, ITINERAIRES</b>          En utilisant différents procédés géo-localisation, ces fonctionnalités permettent:</p> <ul style="list-style-type: none"> <li>□□ d'améliorer l'expérience utilisateur en apportant une information localisée</li> <li>□□ d'afficher la localisation courante d'un utilisateur et de procéder à des recherches autour de soi.</li> <li>□□ d'accompagner l'utilisateur à travers une fonctionnalité de navigation ou d'itinéraire</li> <li>□□ de proposer un service, un produit, suivant une information de géo-localisation.</li> </ul>	<p><b>STATISTIQUES</b>          Des informations et des analyses, en temps réel, apportent des données très utiles pour l'approche marketing, ou l'étude des usages des utilisateurs.</p> <p>Identifier les fonctions les plus utilisées, savoir exactement comment, où, quand et par qui l'application est utilisée, Identifier rapidement les éventuels problèmes.</p>
---	---	--	--

Figure 3 : tableau expliquer les domaines

## 2.4 Les différents types d'applications mobiles

On peut distinguer plusieurs solutions pour créer ces applications mobiles : les applications natives, les applications web et les applications hybrides. [6]

### a) Application native

Une application mobile native est une application mobile spécifique à un système d'exploitation mobile, développée avec le langage et les outils associés fournis par l'éditeur du système d'exploitation mobile, et installée directement sur le mobile. Cette installation de l'application native se faisant soit au travers d'un téléchargement via Internet soit par déploiement depuis un ordinateur connecté au mobile.

Aspects dominants : Accès à toutes les fonctionnalités du mobile, rapidité d'exécution, possibilité de notifier (push) l'utilisateur, eco-système associés aux magasins (store) d'application permettant de générer des revenus.

### b) Application web ou Web App

L'application mobile web est un site Internet qui a été adapté pour une restitution sur un navigateur de mobile. De la même manière que les sites web, une application mobile web est généralement placée sur un serveur HTTP et se manipule en actionnant des éléments d'interface XHTML à l'aide d'un navigateur web.

Aspect dominant : une application mobile web adresse l'ensemble des utilisateurs de mobile, et non une population spécifique utilisant une marque précise de mobile.

### c) Applications hybrides

Une application hybride est un **mélange de code natif et d'affichage de vues HTML/javascript**. Concrètement toutes les plateformes mobiles proposent un composant de type WebView, permettant d'afficher du contenu web soit sur une partie de l'écran, soit en plein écran, et en utilisant le moteur HTML du navigateur intégré au système. Ces applications hybrides peuvent être distribuées sur les stores des systèmes mobiles.

Plusieurs stratégies sont alors possibles, selon que l'on place le curseur plus du côté natif ou plus du côté HTML :

- ne réaliser que certains écrans voir même que certains composants d'IHM en HTML.
- réaliser tous les écrans en HTML mais garder la logique applicative en code natif, notamment les effets de transitions entre écrans et la gestion du scrolling.
- réaliser les écrans en HTML, et les transitions / scrolling en javascript. Le code natif peut alors se cantonner à quelques composants techniques très ciblés. De la même manière selon les applications la logique métier peut être codée en javascript ou bien en code natif.

## 2.5 Comparaison entre application native et Web App[4]

	Application mobile native	Application mobile web
<b>Portabilité</b>	Développement spécifique à chaque plateforme	Navigateur web... mais une intégration distincte selon la plateforme
<b>Référencement</b>	Arriver à se positionner dans une boutique d'applications Accessible par la recherche dans une boutique d'applications	Accessible par les moteurs de recherche classiques et liens externes éventuels
<b>Accessibilité technique</b>	Dépendante de la plateforme et de l'éventuelle validation par une boutique d'applications Mode offline possible	Eventuelle dépendance aux navigateurs Mode offline (quasi) impossible, support HTML5 nécessaire
<b>Exploitation du mobile</b>	Utilise toutes les possibilités du mobile (GPS, contacts, caméra, voix, calendrier, etc.)	Se limite aux possibilités du navigateur
<b>Développement / Coût</b>	Plus long, plus fastidieux Nécessite un SDK + la connaissance d'un langage spécifique	Généralement moins onéreux HTML / JavaScript / CSS
<b>Effet immersif</b>	Plus de possibilités, richesse fonctionnelle et multimédia Logique marketing forte	Limité. Des possibilités plus importantes avec l'arrivée de HTML5
<b>Expérience utilisateur</b>	Maximale. Possibilité de notifier l'utilisateur (Push) même quand l'application n'est pas utilisée	Limitée mais conforme à l'utilisation classique du Web (hors animation de type Flash)
<b>Mises à jour</b>	Processus de soumission à un magasin d'application. Contraignant dans le cas de l'AppStore d'Apple Donc mises à jour en mode par action de l'utilisateur	Mise à jour instantanée sur le serveur Web, en une seule opération
<b>Développement / Courbe d'apprentissage</b>	Dépend des compétences existantes sur le SDK du mobile visé	Compétences HTML / CSS / JavaScript plus classiques
<b>Potentiel performance</b>	Maximum (traitement en local plus nombreux)	Dépend du développement du site, de la connexion, ...

**Figure 4 : Comparaison entre application native et Web App**

## 2.6 Éléments de comparaison entre les applications mobile [6]

### ➤ Coûts de mise en œuvre

Développer une application native pour plusieurs plateformes mobiles peut coûter très cher, de par la multitude de langages et technologies mises en œuvre. Selon le nombre de plateformes cibles, une technologie web ou même hybride sera souvent moins coûteuse. De plus il sera souvent plus simple de disposer de développeurs maîtrisant les technologies web, que les diverses plateformes mobiles.

### ➤ Qualité, rapidité des applications

Difficile de rivaliser avec les applications natives, Celles-ci seront presque toujours plus rapides.

Mais cela dépend fortement du type d'application en jeu et de ses fonctionnalités.

### ➤ Publication et mises à jour

Une importante contrainte des applications natives est que celles-ci doivent être approuvées avant diffusion sur leur store respectif (sauf pour Android), ce qui peut s'avérer long et contraignant. Une application hybride permet de limiter ce désagrément, et une web app de s'en affranchir complètement. Le même problème se pose pour les mises à jour, il n'est souvent pas possible de diffuser un patch correctif en urgence ou même rapidement sur une application native.

### ➤ Monétisation

Les magasins d'applications permettent très facilement de vendre les applications, mêmes si Apple, Google et consorts prélèvent leur part sur les prix de vente, généralement autour de 30%.

Même si les stores d'applications web commencent timidement à apparaître, leur usage est encore très restreint.

Les stores servent aussi de moteur de recherche et de vitrines pour les applications, et permettent ainsi de les mettre en avant et de les faire découvrir.

## 2.7 Statistiques sur le marché

Le nombre de téléchargements d'applications mobiles est actuellement en forte hausse. Cette tendance va de pair avec la vente des Smartphones.

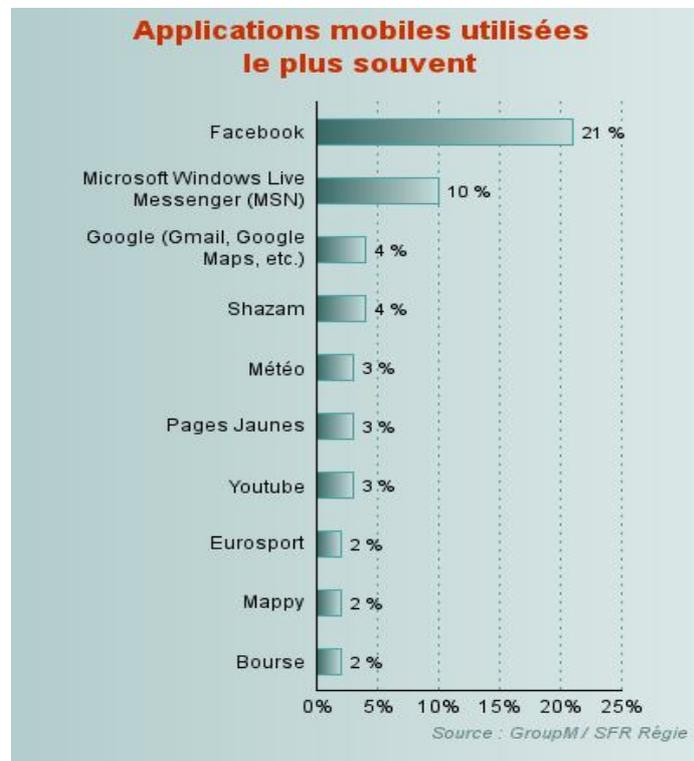
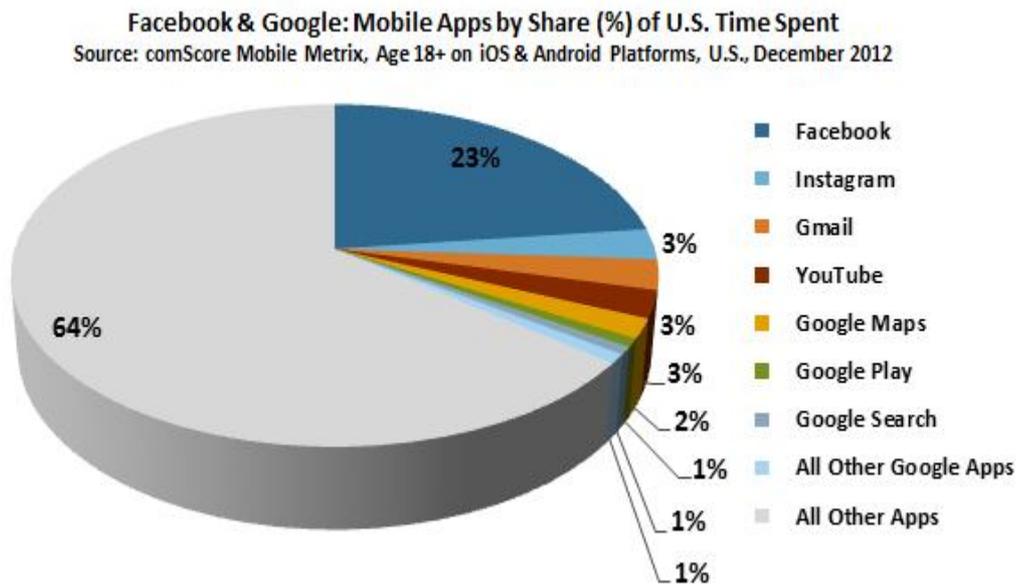


Figure 5 : Les Application mobiles utilisées le plus souvent

**Facebook est de loin l'application mobile la plus utilisée par les possesseurs de smartphones.** Plus d'un sondé sur cinq surfant sur Internet avec un "téléphone intelligent" dit utiliser souvent l'application développée par le réseau social. Le service de messagerie instantanée de Microsoft, Windows Live Messenger arrive en seconde position, cité par 10 % des internautes. **L'ensemble des services de Google (Maps, Gmail, Youtube, etc.) n'est cité que par 4 % des possesseurs de smartphones.** [7]

- Facebook et Google au top des applications mobiles : ces chiffres ne concernent que les Etats- Unis



**Figure 6 : Facebook et Google au top des applications mobiles**

### 3. Les technologies d'accès sans fil

Grâce aux réseaux sans fil, un utilisateur a la possibilité de rester connecté tout en se déplaçant dans un périmètre géographique plus ou moins étendu, c'est la raison pour laquelle on entend parfois parler de "mobilité".

Un **réseau sans fil** (en anglais *wireless network*) est, comme son nom l'indique, un réseau dans lequel au moins deux terminaux (ordinateur portable, PDA, etc.) peuvent communiquer sans liaison filaire. Ils sont basés sur une liaison utilisant des ondes radio-électriques (radio et infrarouges) en lieu et place des câbles habituels. Il existe plusieurs technologies se distinguant d'une part par la fréquence d'émission utilisée ainsi que le débit et la portée des transmissions. et les **réseaux sans fil** permettent de relier très facilement des équipements distants d'une dizaine de mètres à quelques kilomètres. De plus l'installation de tels réseaux ne demande pas de lourds aménagements des infrastructures existantes comme c'est le cas avec les réseaux filaires (creusement de tranchées pour acheminer les câbles, équipements des bâtiments en câblage, goulottes et connecteurs), ce qui a valu un développement rapide de ce type de technologies.

#### 3.2 Les catégories de réseaux sans fil

On distingue habituellement plusieurs catégories de réseaux sans fil, selon le périmètre géographique offrant une connectivité (appelé *zone de couverture*) : [8]

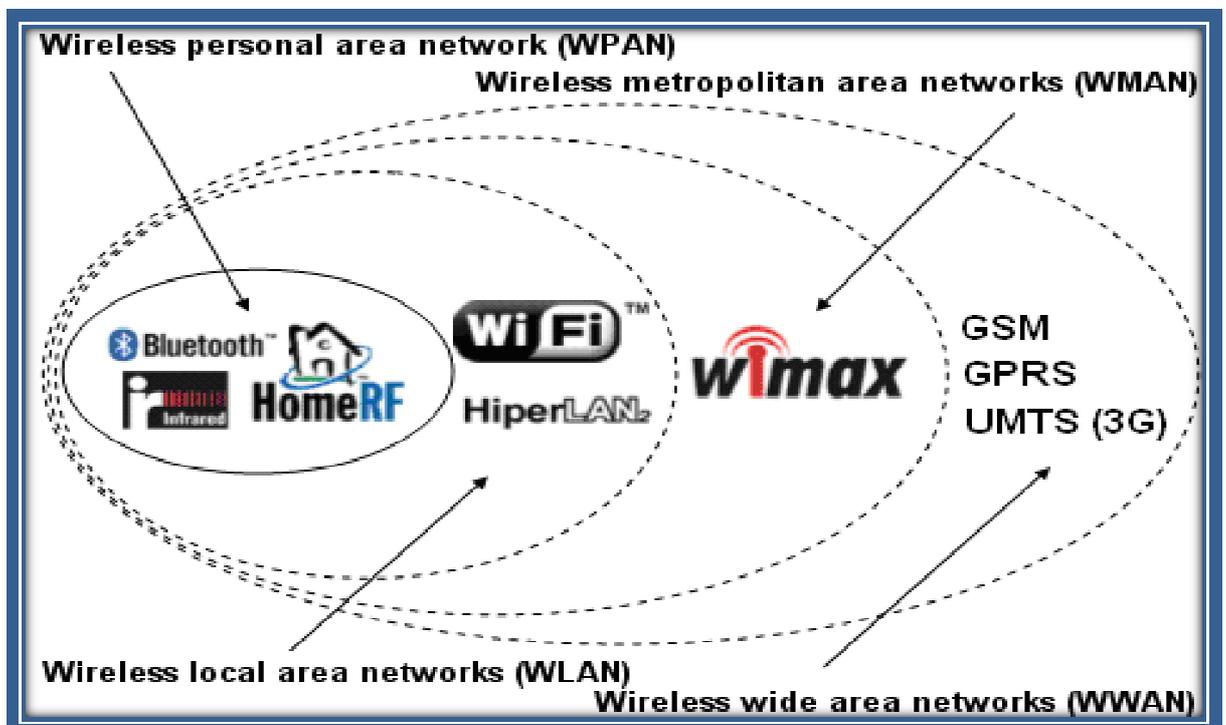


Figure 7 : Wireless networks categories

### 3.2.1 WWAN

Le réseau étendu sans fil (**WWAN** pour Wireless Wide Area Network) est également connu sous le nom de réseau cellulaire mobile. Il s'agit des réseaux sans fil les plus répandus puisque tous les téléphones mobiles sont connectés à un réseau étendu sans fil. Les principales technologies sont les suivantes :

- GSM (Global System for Mobile Communication ou en français Groupe Spécial Mobile)
- GPRS (General Packet Radio Service)
- UMTS (Universal Mobile Telecommunication System) [9]

#### 3.2.1.1 Les différentes générations de téléphonie mobile [10]

Génération	Acronyme	Intitulé	Débits théoriques Descendant / Montant
<b>1G</b>	Radiocom 2000	Radiocom 2000 (France Télécom), SFR 2000 (SFR)	Analogique
<b>2G</b>	GSM (échanges de type voix uniquement)	Global System for Mobile Communication	9,6 Kbits/s / -
<b>2.5G</b>	GPRS (échange de données sauf voix)	Global Packet Radio Service	171,2 Kbits/s / 17,9 Kbits/s
<b>2.75G</b>	EDGE (basé sur réseau GPRS existant)	Enhanced Data Rate for GSM Evolution	384 Kbits/s / 64 Kbits/s
<b>3G</b>	UMTS	Universal Mobile Telecommunications System	1,9 Mbits/s / -
<b>3.5G ou 3G+</b>	HSDPA (dérivé de l'UMTS)	High Speed Downlink Packet Access	14 Mbit/s / -
<b>3.75G</b>	HSPA+	High Speed Packet Access +	14 Mbits/s / 5,8 Mbits/s
<b>3.75G ou H+ ou Dual Carrier</b>	DC-HSPA+	Dual-Cell High Speed Packet Access +	42 Mbits/s / -
<b>3.9G</b>	LTE	Long Term Evolution	326 Mbits/s / -
<b>4G</b>	LTE-Advanced	Long Term Evolution Advanced	1 GBits/s / 500 Mbits/s

**Figure 8 : Les différentes générations de téléphonie mobile**

### 3.2.2 WMAN

Le réseau métropolitain sans fil (**WMAN** pour *Wireless Metropolitan Area Network*) est connu sous le nom de **Boucle Locale Radio (BLR)**. Les WMAN sont basés sur la norme *IEEE 802.16*. La boucle locale radio offre un débit utile de 1 à 10 Mbit/s pour une portée de 4 à 10 kilomètres, ce qui destine principalement cette technologie aux opérateurs de télécommunication.

La norme de réseau métropolitain sans fil la plus connue est le Wi MAX, permettant d'obtenir des débits de l'ordre de 70 Mbit/s sur un rayon de plusieurs kilomètres. [11]

#### 3.2.2.1 Wi MAX

l'abréviation pour *Worldwide Interoperability for Microwave Access*. Il s'agit d'un standard de réseau sans fil métropolitain créé par les sociétés Intel et Alvarion en 2002 et ratifié par l'IEEE (*Institute of Electrical and Electronics Engineer*) sous le nom IEEE-802.16. Plus exactement, **Wi MAX** est le label commercial délivré par le *Wi MAX Forum* aux équipements conformes à la norme IEEE 802.16, afin de garantir un haut niveau d'interopérabilité entre ces différents équipements. Les équipements certifiés par le Wi MAX Forum peuvent ainsi arborer le logo suivant :



L'objectif du Wi MAX est de fournir une connexion internet à haut débit sur une zone de couverture de plusieurs kilomètres de rayon. Ainsi, dans la théorie, le Wi MAX permet d'obtenir des débits montants et descendants de 70 Mbit/s avec une portée de 50 kilomètres. Le standard Wi MAX possède l'avantage de permettre une connexion sans fil entre une station de base (en anglais *Base Transceiver Station*, notée *BTS*) et des milliers d'abonnés sans nécessiter de ligne visuelle directe (en anglais *Line Of Sight*, parfois abrégés *LOS*) ou *NLOS* pour *Non Line Of Sight*). Dans la réalité le Wi MAX ne permet de franchir que de petits obstacles tels que des arbres ou une maison mais ne peut en aucun cas traverser les collines ou les immeubles. Le débit réel lors de la présence d'obstacles ne pourra ainsi excéder 20 Mbit/s. [12]

### 3.2.3 WLAN

Le *réseau local sans fil* (noté **WLAN** pour *Wireless Local Area Network*) est un réseau permettant de couvrir l'équivalent d'un réseau local d'entreprise, soit une portée d'environ une centaine de mètres. Il permet de relier entre-eux les terminaux présents dans la zone de couverture. Il existe plusieurs technologies concurrentes :

Le Wifi (ou IEEE 802.11), soutenu par l'alliance WECA (Wireless Ethernet Compatibility Alliance), l'organisme chargé de maintenir l'interopérabilité entre les matériels répondant à la norme 802.11 offre des débits allant jusqu'à 54Mbps sur une distance de plusieurs centaines de mètres. [13]

### 3.2.3.1 Wifi

La norme *IEEE 802.11 (ISO/IEC 8802-11)* est un standard international décrivant les caractéristiques d'un réseau local sans fil (*WLAN*). Le nom **Wi-Fi** (contraction de *Wireless Fidelity*) correspond initialement au nom donné à la certification délivrée par la Wi-Fi Alliance, anciennement WECA (*Wireless Ethernet Compatibility Alliance*). Par abus de langage (et pour des raisons de marketing) le nom de la norme se confond aujourd'hui avec le nom de la certification. Ainsi un réseau Wifi est en réalité un réseau répondant à la norme 802.11. Les matériels certifiés par la Wi-Fi Alliance bénéficient de la possibilité d'utiliser le logo suivant :



Grâce au Wi-Fi, il est possible de créer des réseaux locaux sans fils à haut débit pour peu que l'ordinateur à connecter ne soit pas trop distante par rapport au point d'accès. Dans la pratique, le Wifi permet de relier des ordinateurs portables, des ordinateurs de bureau, des assistants personnels (PDA) ou tout type de périphérique à une liaison haut débit (11 Mbps ou supérieur) sur un rayon de plusieurs dizaines de mètres en intérieur (généralement entre une vingtaine et une cinquantaine de mètres) à plusieurs centaines de mètres en environnement ouvert. [14]

### 3.2.4 WPAN

Le réseau personnel sans fil (appelé également *réseau individuel sans fil* ou *réseau domestique sans fil* et noté **WPAN** pour *Wireless Personal Area Network*) concerne les réseaux sans fil d'une faible portée : de l'ordre de quelques dizaines mètres. Ce type de réseau sert généralement à relier des périphériques (imprimante, téléphone portable, appareils domestiques, ...) ou un assistant personnel (PDA) à un ordinateur sans liaison filaire ou bien à permettre la liaison sans fil entre deux machines très peu distantes.

La principale technologie *WPAN* est la technologie **Bluetooth**, proposant un débit théorique de 1 Mbps pour une portée maximale d'une trentaine de mètres. Bluetooth, connue aussi sous le nom *IEEE 802.15.1*, possède l'avantage d'être très peu gourmande en énergie, ce qui la rend particulièrement adaptée à une utilisation au sein de petits périphériques.

La technologie **Bluetooth** a été originairement mise au point par *Ericsson* en 1994. En février 1998 un groupe d'intérêt baptisé *Bluetooth SpecialInterest Group (Bluetooth SIG)*, réunissant plus de 2000 entreprises dont Agere, Ericsson, IBM, Intel, Microsoft, Motorola, Nokia et Toshiba, a été formé afin de produire les spécifications Bluetooth 1.0, qui furent publiées en juillet 1999. [15]



### 3.2.4.1 Bluetooth

Bluetooth est une technologie de réseau personnel sans fils, les appareils Bluetooth ne nécessitent pas d'une ligne de vue directe pour communiquer, ce qui rend plus souple son utilisation et permet notamment une communication d'une pièce à une autre, sur de petits espaces.

L'objectif de Bluetooth est de permettre de transmettre des données ou de la voix entre des équipements possédant un circuit radio de faible coût, sur un rayon de l'ordre d'une dizaine de mètres à un peu moins d'une centaine de mètres et avec une faible consommation électrique.

Ainsi, la technologie Bluetooth est principalement prévue pour relier entre-eux des périphériques (imprimantes, téléphones portables, appareils domestiques, oreillettes sans

fil, souris, clavier, etc.), des ordinateurs ou des assistants personnels (PDA), sans utiliser de liaison filaire. La technologie Bluetooth est également de plus en plus utilisée dans les téléphones portables, afin de leur permettre de communiquer avec des ordinateurs ou des assistants personnels et surtout avec des dispositifs mains-libres tels que des oreillettes bluetooth. [16]

- **Caractéristiques**

Le Bluetooth permet d'obtenir des débits de l'ordre de 1 Mbps, correspondant à 1600 échanges par seconde en full-duplex, avec une portée d'une dizaine de mètres avec un émetteur de classe II et d'un peu moins d'une centaine de mètres avec un émetteur de classe I.

Le standard Bluetooth définit en effet 3 classes d'émetteurs proposant des portées différentes en fonction de leur puissance d'émission :

<b>Classe</b>	<b>Puissance (affaiblissement)</b>	<b>Portée</b>
<b>I</b>	100 mW (20 dBm)	100 mètres
<b>II</b>	2,5 mW (4 dBm)	15-20 mètres
<b>III</b>	1 mW (0 dBm)	10 mètres

- **Normes Bluetooth**

Le standard Bluetooth se décompose en différentes normes :

- IEEE 802.15.1 définit le standard Bluetooth 1.x permettant d'obtenir un débit de 1 Mbit/sec ;
- IEEE 802.15.2 propose des recommandations pour l'utilisation de la bande de fréquence 2.4 GHz (fréquence utilisée également par le Wifi). Ce standard n'est toutefois pas encore validé ;
- IEEE 802.15.3 est un standard en cours de développement visant à proposer du haut débit (20 Mbit/s) avec la technologie Bluetooth ;

- IEEE 802.15.4 est un standard en cours de développement pour des applications Bluetooth à bas débit. [16]

## 4. Conclusion

L'application mobile réalisée dans une technologie n'est pas nécessairement compatible avec toutes les plateformes (c'est d'ailleurs impossible). Alors de portabilité et de compatibilité de l'application avec le parc de téléphones que l'on trouve sur le marché mondial.

Aussi une application est réalisée à partir de différents langages de programmation car il existe des environnements très hétérogènes au niveau des plateformes (les OS mobiles) installées sur les téléphones portables.

Dans le prochain chapitre, Nous allons définir les systèmes d'application mobiles et les outils pour développer ces applications.

## Chapitre II

# Technologies et Outils pour le développement des applications mobiles

## 1. Introduction

Comme l'ordinateur, téléphone mobile fonctionne grâce à son système d'exploitation. Actuellement il existe plusieurs systèmes d'exploitation grâce à la concurrence, et les entreprises qui produisent ces systèmes battent entre eux pour offrir le meilleur système, ils améliorent leurs produits à chaque fois, si pour ça, nous trouvons plusieurs versions par un système.

Le système d'exploitation mobile est différent de l'un à l'autre : la langue qui a été développée, interface, existence de la portabilité ou non, .....Etc.

Dans ce chapitre, nous allons aborder les notions qui appartiennent au système d'exploitation mobile puis voir les systèmes plus intéressants (ios, android, symbian, windows) et les plateformes (Java 2 Micro Edition J2ME, Le.NET Compact Framework de Microsoft).

## 2. Les systèmes exploitation mobile

### 2.1 Définition

Le système d'exploitation de téléphone mobile, est un ensemble de programmes responsables de la liaison entre les ressources matérielles de l'appareil et ses applications logicielles. Il assure le démarrage et le fonctionnement de téléphone mobile. [17]

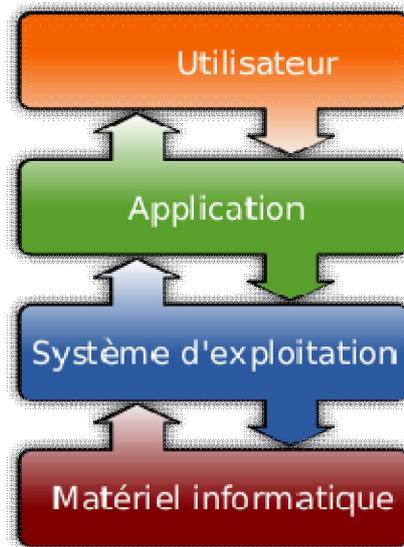


Figure 9 :Le système d'exploitation

**Autre définition :** Un **système d'exploitation mobile** est un système d'exploitation conçu pour fonctionner sur un appareil mobile. Ce type de système d'exploitation se concentre entre autres sur la gestion de la connectivité sans fil et celle des différents types d'interface. [18].

Le **système d'exploitation** (noté *SE* ou *OS*, abréviation du terme anglais *Operating System*), est chargé d'assurer la liaison entre les ressources matérielles, l'utilisateur et les applications (traitement de texte, jeu vidéo, ...). Ainsi lorsqu'un programme désire accéder à une ressource matérielle, il ne lui est pas nécessaire d'envoyer des informations spécifiques au périphérique, il lui suffit d'envoyer les informations au système d'exploitation, qui se charge de les transmettre au périphérique concerné via son pilote. En l'absence de pilotes il faudrait que chaque programme reconnaisse et prenne en compte la communication avec chaque type de périphérique

Le système d'exploitation permet ainsi de "dissocier" les programmes et le matériel, afin notamment de simplifier la gestion des ressources et offrir à l'utilisateur une interface homme-machine (notée «IHM») simplifiée afin de lui permettre de s'affranchir de la complexité de la machine physique. Mais ce n'est pas là son seul rôle. [19]

## 2.2 Rôles du système d'exploitation

Les rôles du système d'exploitation sont divers :

- **Gestion des entrées/sorties :**

(Comme on vient de le voir) le système d'exploitation permet d'unifier et de contrôler l'accès des programmes aux ressources matérielles par l'intermédiaire des pilotes (appelés également gestionnaires de périphériques ou gestionnaires d'entrée/sortie).

- **Gestion du processeur :**

Le système d'exploitation est chargé de gérer l'allocation du processeur entre les différents programmes grâce à un **algorithme d'ordonnancement**. Le type d'ordonnanceur est totalement dépendant du système d'exploitation, en fonction de l'objectif visé.

- **Gestion de la mémoire vive :**

Le système d'exploitation est chargé de gérer l'espace mémoire alloué à chaque application et, le cas échéant, à chaque usager. En cas d'insuffisance de mémoire physique, le système d'exploitation peut créer une zone mémoire sur le disque dur, appelée «**mémoire virtuelle**». La mémoire virtuelle permet de faire fonctionner des applications nécessitant plus de mémoire qu'il n'y a de mémoire vive disponible sur le système. En contrepartie cette mémoire est beaucoup plus lente.

- **Gestion de l'exécution des applications :**

Le système d'exploitation est chargé de la bonne exécution des applications en leur affectant les ressources nécessaires à leur bon fonctionnement. Il permet à ce titre de «tuer» une application ne répondant plus correctement.

- **Gestion des droits :**

Le système d'exploitation est chargé de la sécurité liée à l'exécution des programmes en garantissant que les ressources ne sont utilisées que par les programmes et utilisateurs possédant les droits adéquats.

- **Gestion des fichiers :**

Le système d'exploitation gère la lecture et l'écriture dans le système de fichiers et les droits d'accès aux fichiers par les utilisateurs et les applications.

- **Gestion des informations :**

Le système d'exploitation fournit un certain nombre d'indicateurs permettant de diagnostiquer le bon fonctionnement de la machine.

## 2.3 Composantes du système d'exploitation

Le système d'exploitation est composé d'un ensemble de logiciels permettant de gérer les interactions avec le matériel. Parmi cet ensemble de logiciels on distingue généralement les éléments suivants : [19]

- Le **noyau** (en anglais *kernel*) représentant les fonctions fondamentales du système d'exploitation telles que la gestion de la mémoire, des processus, des fichiers, des entrées-sorties principales, et des fonctionnalités de communication.
- L'**interpréteur de commande** (en anglais *shell*, traduisez «coquille» par opposition au noyau) permettant la communication avec le système d'exploitation par l'intermédiaire d'un langage de commandes, afin de permettre à l'utilisateur de piloter les périphériques en ignorant tous des caractéristiques du matériel qu'il utilise, de la gestion des adresses physiques, etc.
- Le **système de fichiers** (en anglais «*file system*», noté *FS*), permettant d'enregistrer les fichiers dans une arborescence.
- L'**interface usager**.
- Les **pilotes**.

On trouve plusieurs systèmes d'exploitation sur des les mobiles, alors nous choisissons plus intéressant: iOS, Android, Symbian OS, Windows CE .A chaque OS mobile correspond une technologie et un « store » où les applications peuvent être téléchargées, de manière gratuite ou payante : iOS (Apple), Android (Google Play), Symbian (Nokia), Windows CE (Microsoft)

## 2.4iOS d'Apple

- **iOS**(anciennement **iPhone OS**) est le système d'exploitation mobile développé par Apple pour l'iPhone. Il est dérivé de Mac OS X dont il partage les fondations (le kernel hybride XNU basé sur le micro-noyau Mach, les services Unix et Cocoa, etc.). iOS comporte quatre couches d'abstraction, une couche « Core OS », une couche « Core Services », une couche « Media » et une couche « Cocoa »Le système d'exploitation occupe moins d'un demi-gigaoctet (Go) de la capacité mémoire totale de l'appareil.[20]

### L'App Store

est la plateforme de téléchargement des applications mobiles iPhone( similaire au Google Play )Selon les cas, les applications obtenues sur l'App Store sont gratuites ou payantes.

- **Caractéristique**

Logicielle de l'iPhone est caractérisée par

**Le BaseBand** : C'est donc un micro logiciel autonome qui s'occupe en temps réel de toutes les interactions avec les périphériques de communication de l'appareil : Bluetooth, Wi-Fi et GSM.

**Le BootLoader** : C'est une partie du BaseBand, dont le rôle principal est d'assurer le démarrage de l'iPhone, de contrôler son activation, et sa compatibilité avec la carte SIM insérée

**Le firmware** : il s'agit d'un logiciel interne de l'appareil, cette fois responsable de la gestion de sa partie systémique (l'écran, le clavier tactile, etc.).

**Le SeckPack** : C'est une partie de la mémoire flash de l'appareil contenant entre autres des informations sur le verrouillage de celui-ci.

- **Versions**

- **La version 2.0**d'iOS a été dévoilée à l'occasion d'une conférence le 6 mars 2008,

- **La version 2.2.1** a été la dernière avant la sortie, le 17 juin 2009, soit un an après la version 2.0, de la mise à jour 3.0 qui a apporté son lot de fonctionnalités tel que le copier / coller,
- **La version 4.0** du firmware, présentée le 8 avril 2010,
- **La version 5.0** du firmware a été annoncée le 6 juin 2011 qui apporte la présentation du système **Icloud et la 6.0** le 11 juin 2012.

## 2.5 Android de Google

• **Android** est un système d'exploitation open source utilisant le noyau Linux, pour 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, des radio-réveils, des autoradios et même des voitures. [21]

### • Googaleplay

Anciennement dénommé AndroidMarket, est le magasin en ligne de Google. Celui-ci permet de télécharger des logiciels, des livres, des films ou de la musique, payants ou non. Il est aussi possible de les noter et de les commenter. En septembre 2011, il y avait plus de 520 000 applications sur AndroidMarket, dont 65 % sont gratuites.

- AndroidMarket a été remplacé par Google Play Store le 6 mars 2012.



### • Caractéristiques

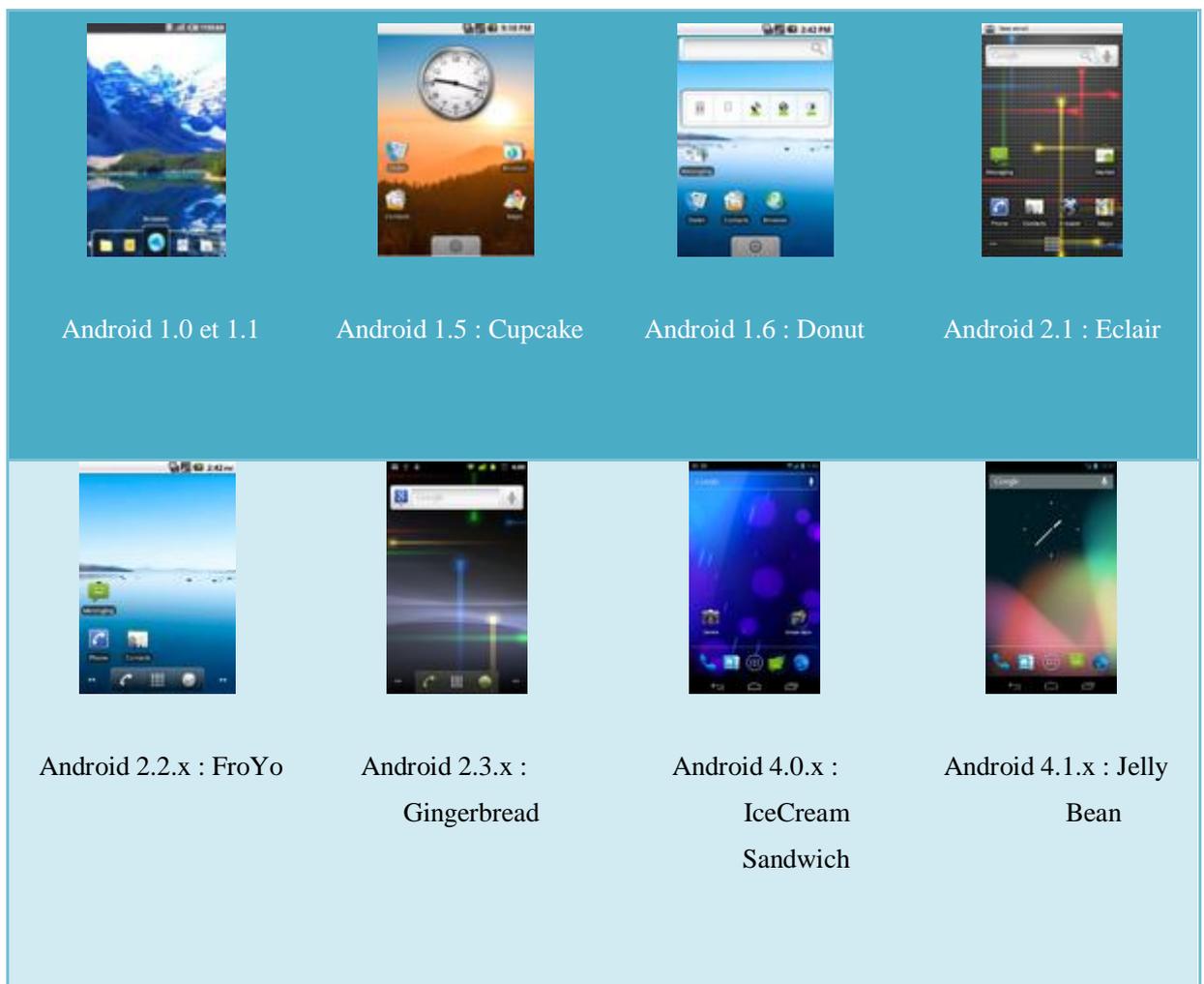
- Android est un système d'exploitation fondé sur un noyau Linux.
- il comporte une interface spécifique, développée en Java, toutefois il est possible de passer outre cette interface, en programmant ses applications en C, mais le travail de portabilité en sera plus important.
- la majorité des périphériques Android sont basés sur l'architecture ARM.
- Android a été conçu pour intégrer au mieux des applications existantes de Google comme le service de courrier Gmail, celui de cartographie, Google Maps, ou encore Google Agenda, Google Talk, YouTube. Un accent particulier est mis sur la géolocalisation avec

Google Latitude et la météo correspondant à la ville la plus proche disponible sur le menu principal.

- **Versions**

:Android est régulièrement mis à jour

La première version d'Android a été distribuée fin 2007. Depuis, de nombreuses versions ont vu le jour. La version la plus récente est la 4.2.1 Jelly Bean.



**Figure 10 : version d'Android**

## 2.6 Windows CE de Microsoft

- **Windows CE**(parfois abrégé WinCE) est une variation de Windows pour les systèmes embarqués et autres systèmes minimalistes, utilisée notamment dans les PC de poche ou *Handheld*. [22]

- **Microsoft Corporation**

Son activité principale consiste à développer et vendre des systèmes d'exploitation et des logiciels.

- **Caractéristiques**

- Windows CE est prévu pour fonctionner sur des périphériques à faible empreinte mémoire. La taille d'une image système sans support graphique peut être de l'ordre du Mo.
- Windows CE peut fonctionner sans support de stockage, exclusivement en RAM/ROM
- Windows CE est un système 32 bits pouvant fonctionner sur architecture ARM, x86, MIPS, SHx
- Windows CE répond aux contraintes d'un système temps-réel avec un ordonnanceur préemptif à 256 niveaux de priorités fixes et un temps de latence des interruptions déterministe. Sous certaines conditions, un mécanisme d'inversion de priorité peut être pris en charge par l'OS. L'unité d'exécution est le thread.
- Windows CE prend en charge un environnement graphique basé sur une mécanique événementielle (GWES - GraphicWindowing and Events Subsystem) et intégrant une couche GDI (GraphicDevice Interface) allégée par rapport à celle disponible sur XP.
- Communication: Bluetooth, IrDA, Ethernet, Wifi, NDIS 5.1, TCP/IP, ...

- **Versions**

- **1.0** : Windows CE est sorti pour la première fois en 1996 (nom de code "Pegasus").
- **2.0, 2.11, 2.12** : Versions fondées sur l'architecture 1.0. Jusqu'à la version 2.12 l'ordonnancement était fondé sur un système à 8 niveaux de priorités variables.

- **3.0** : La première rupture technologique a lieu avec CE 3.0 en 1999 avec l'intégration d'un ordonnanceur préemptif à 256 niveaux de priorités fixes rendant Windows CE éligible en tant que système temps réel. Windows CE 3.0 marque aussi l'apparition des outils graphiques pour le développement système et applicatif.
- **4.0, 4.1, 4.2** : Sorties entre 2001 et 2003.
- **5.0** : Sortie fin 2004.
- **6.0** : Windows Embedded CE 6.0 marque la seconde rupture technologique dans cette famille d'OS.

Le tableau suivant présente certaines des caractéristiques de Windows CE :

Windows CE	
	
	
<b>Société</b>	Microsoft
<b>programmé en</b>	C
<b>OS famille</b>	Windows CE
<b>modèle de code source</b>	Code source fermé (shared source kernel)
<b>première version</b>	16 Novembre 1996
<b>Dernière version stable</b>	7.0/ 1Mars2011
<b>Langue disponible (s)</b>	Multi lingual
<b>plates-formes supportées</b>	x86, MIPS, ARM, (SuperHup to 6.0 R2)

Type de noyau	MonolithicHybrid
Official site web	WindowsCE

Figure 11 : les caractéristiques de Windows CE

## 2.7 Symbian de Nokia

- **Symbian OS** est un système d'exploitation pour téléphones portables conçu par Symbian ltd. Il est l'héritier du système d'exploitation Epc32 qui équipa les Psion, et est né d'un consortium entre les différents constructeurs (souvent des Nokia). Il dispose de nombreuses API spécifiques pour la communication mobile voix et données, et implémente la majorité des standards de communication : IPv4/IPv6, MMS, Bluetooth, GPRS/UMTS, Java, WAP, SyncML...

Il a été adopté par différents fabricants de téléphones portables de deuxième génération (GSM et GPRS) et troisième génération (UMTS). [23]

- **Nokia Corporation**

est une entreprise de télécommunications, ses principaux produits sont des téléphones mobiles et appareils informatiques portables. Symbian OS est acheté en 2008 à 100 % par Nokia.

- **Caractéristiques**

Symbian OS fournit les fonctionnalités essentielles du système d'exploitation.[24]

- **Architecture**

- ✓ **Domaines de la technologie et des packages**

Le design de Symbian est subdivisé en domaines technologiques, dont chacun comprend un certain nombre de progiciels. Chaque domaine de la technologie a sa propre feuille de route.

- ✓ **Symbian noyau**

Le noyau de Symbian (EKA2) soutient suffisamment rapide réponse en temps réel pour construire un téléphone single-core autour d'elle, c'est-un téléphone dans lequel un noyau mono-processeur exécute à la fois les applications de l'utilisateur et la pile de signalisation.

✓ **DesignSymbian**

Fonctionnalités multitâche préemptif et la protection de la mémoire, comme les autres systèmes d'exploitation (en particulier ceux créés pour une utilisation sur les ordinateurs de bureau).

Symbian OS a été créé avec trois principes de conception des systèmes à l'esprit:

1. l'intégrité et la sécurité des données des utilisateurs est primordiale
2. Temps d'utilisation ne doit pas être gaspillé
3. Toutes les ressources sont limitées

✓ **Système d'exploitation**

La Partout modèle contient les couches suivantes, de haut en bas:

Couche UI Framework

Couche Application de services

Java ME

Couche OS services

Services OS générique

Services de communications

Services multimédias et graphiques

Services de connectivité

la couche Base de services

Services Noyau et couche d'interface matérielle.

• **Versions**



Menu principal de l'original S60v3



Menu principal de Symbian S60v5



Symbian v8.1 avec une interface S60v2, sur un Nokia N70



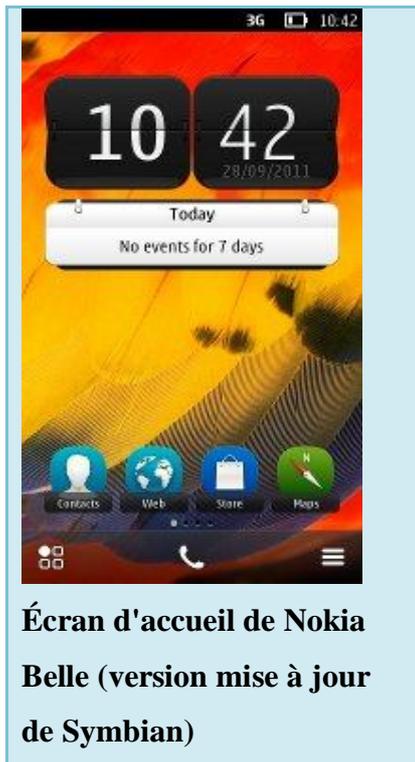
Symbian v9.3 avec une interface S60v3, sur un Nokia 6710



Symbian v9.1 avec une interface S60v3, sur un Nokia E61



Symbian Belle, sur un Nokia C7



**Figure 12 : Version de Symbian**

### **3. Les Plateformes**

Une plate-forme est une base de travail à partir de laquelle on peut écrire, lire, utiliser, développer un ensemble de logiciels.

Pour le développement des applications sur le système mobile dont deux technologies qui sont utilisés, ce sont la technologie J2ME de Sun utilisant le langage Java, et la technologie .NET Compact Framework de Microsoft

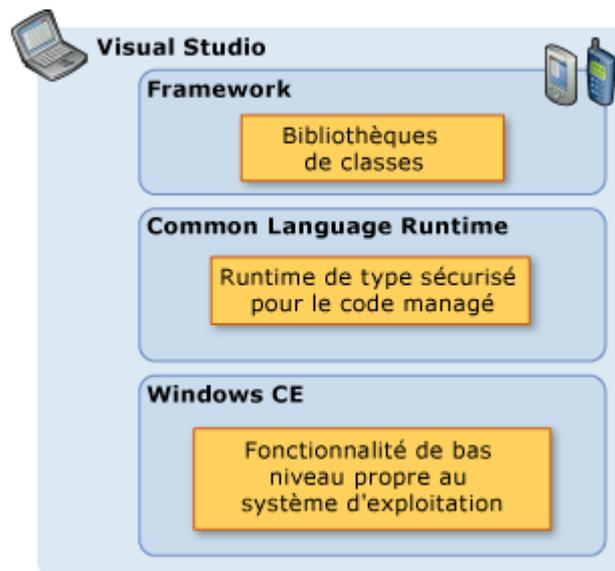
#### **3.1 Le.NET Compact Framework de Microsoft**

Le **.Net Compact Framework** de Microsoft (.Net CF) est une version du .NET Framework spécialement conçue pour être exécutée sur des terminaux mobiles de type Windows CE ou Windows Mobile tels que des PDA, des téléphones portables, des panels PC industriels...

Le **.Net Compact Framework** est un sous-ensemble du .Net Framework, certaines classes sont communes avec le .Net Framework et d'autres sont spécifiques au Compact Framework comme l'Input Panel Windows CE. [25]

### 3.1.1 L'architecture de la plateforme .NET Compact Framework.

L'illustration suivante résume l'architecture de la plateforme .NET Compact Framework. [26]



**Figure 13 : l'architecture de la plateforme .NET Compact Framework.**

#### a) Windows CE

Le .NET Compact Framework utilise le système d'exploitation Windows CE pour les fonctionnalités principales et les quelques fonctionnalités spécifiques au périphérique. Plusieurs types et assemblies, comme pour les Windows Forms, les graphiques, la fonctionnalité de dessin et les services Web, ont été régénérés afin d'être exécutés de façon efficace sur les périphériques, plutôt que d'être copiés à partir du .NET Framework complet.

Le .NET Compact Framework fournit l'interopérabilité suivante avec Windows CE :

- Compatibilité avec la sécurité native.
- Intégration complète avec les programmes d'installation natifs.
- Interopérabilité avec le code natif utilisant COM Interop et l'appel de code non managé.

## **b) Common LanguageRuntime**

Le Common Language Runtime (CLR) du .NET Compact Framework a également été régénéré pour permettre aux ressources soumises à des contraintes d'être exécutées sur une mémoire restreinte et d'utiliser efficacement la puissance de la batterie.

Une couche d'adaptation de plateforme, non représentée dans l'illustration, existe entre Windows CE et le Common Language Runtime pour mapper les services et les interfaces de périphérique requis par le CLR et le Framework sur les services et interfaces Windows CE.

## **C) Framework**

Le .NET Compact Framework est un sous-ensemble du .NET Framework et contient également des fonctionnalités exclusivement conçues pour le .NET Compact Framework. Il fournit les fonctionnalités et la simplicité d'utilisation qui permet de faire passer les développeurs d'applications de périphérique natives au .NET Framework et de faire passer les développeurs d'applications bureautiques aux périphériques.

## **d) Visual Studio**

L'utilisation de Microsoft Visual Studio 2005 pour développer des applications Smart Device est aussi simple qu'avec les applications bureautiques. Le développement Smart Device dans Visual Studio comprend un ensemble de types de projet et des émulateurs qui ciblent le développement pour Pocket PC, Smartphone et Windows CE incorporé.

## **3.2 Java 2 Micro Edition J2ME**

Java 2 Micro Edition est une architecture technique dont le but est de fournir un socle de développement aux applications embarquées. L'intérêt est de proposer toute la puissance d'un langage tel que Java associé aux services proposés par une version bridée du Framework J2SE : J2ME. Les terminaux n'ayant pas les mêmes capacités en terme de ressources que les ordinateurs de bureau classiques (mémoire, disque et puissance de calcul), la solution passe par la fourniture d'un environnement allégé afin de s'adapter aux différentes contraintes d'exécution. Cependant, comment faire en sorte d'intégrer la diversité de l'offre à un socle technique dont la cible n'est pas définie à priori ? La solution proposée par J2ME consiste à regrouper par catégories certaines familles de produits tout en

proposant la possibilité d'implémenter des routines spécifiques à un terminal donné. L'architecture J2ME se découpe donc en plusieurs couches : [27]

**a) Les profile:**

Ils permettent à une certaine catégorie de terminaux d'utiliser des caractéristiques communes telles que la gestion de l'affichage, des évènements d'entrées/sorties (pointage, clavier, ...) ou des mécanismes de persistance (Base de données légère intégrée). Ces profils sont soumis à spécifications suivant le principe du JCP (Java Community Process).

**b) Les configurations:**

Elles définissent une plate-forme minimale en terme de services concernant un ou plusieurs profils donnés.

**c) Les machines virtuelles:**

En fonction de la cible, la machine virtuelle pourra être allégée afin de consommer plus ou moins de ressources (KVM, CVM, ...)

**d) Le système d'exploitation:**

L'environnement doit s'adapter au système d'exploitation existant (Windows CE, Palm Os, SavaJe, ...)

### **3.2.1 Architecture de J2ME**

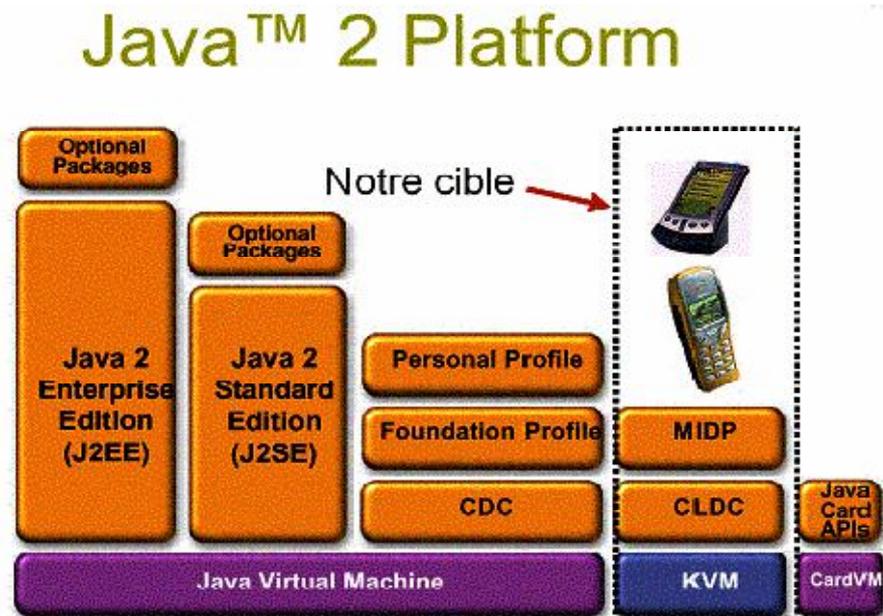


Figure 14 : Architecture de J2ME

Cette architecture en couche a pour but de factoriser pour des familles de produits données un ensemble d'API permettant à une application de s'exécuter sur plusieurs terminaux sans modification de code. Dans cette optique, la plate-forme propose deux configurations :

### 1. CDC (Connected Device Configuration):

Spécifie un environnement pour des terminaux connectés de forte capacité tels que les « set top boxes », les téléphones à écran, la télévision numérique, ...

### 2. CLDC (Connected Limited Device Configuration):

Cible les périphériques à ressources limitées ou faibles tels que les téléphones mobiles, les assistants personnels, ou les périphériques légers sans fil (wireless)

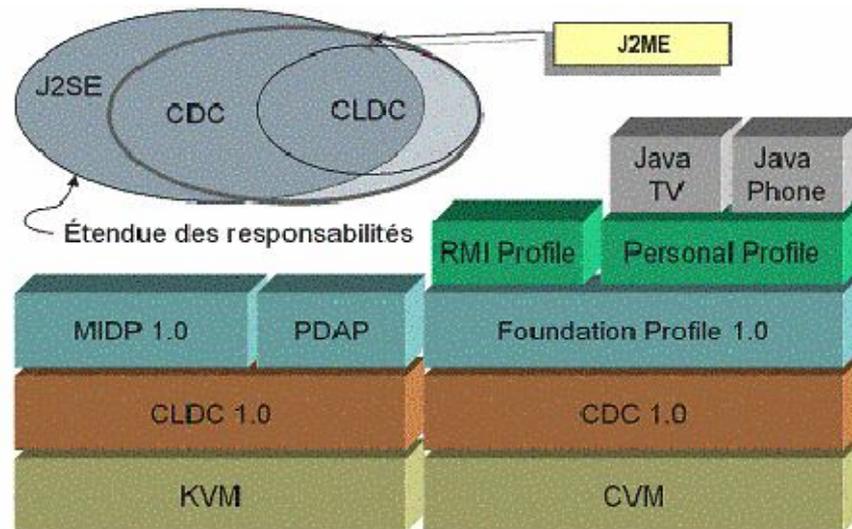


Figure 15 : CDC & CLDC sur L'Architecture de J2ME

### 3.2.2 MIDP (Mobile Information Device Profile)

MIDP est à la base de l'implémentation des classes liées à un profil donné. Vous y trouverez les méthodes permettant de gérer l'affichage, la saisie utilisateur et la gestion de la persistance (base de données). Il existe aujourd'hui deux implémentations majeures de profils MIDP. L'une, plus spécifique, destinée aux Assistants de type Palm Pilot (PalmOs) et l'autre, totalement générique, proposée par Sun comme implémentation de référence (RI). Ces profils sont en libre téléchargement sur le site de Sun et intègrent plusieurs émulateurs permettant de tester les applications de manière logicielle. Les API liées à MIDP font aussi partie de ce package :

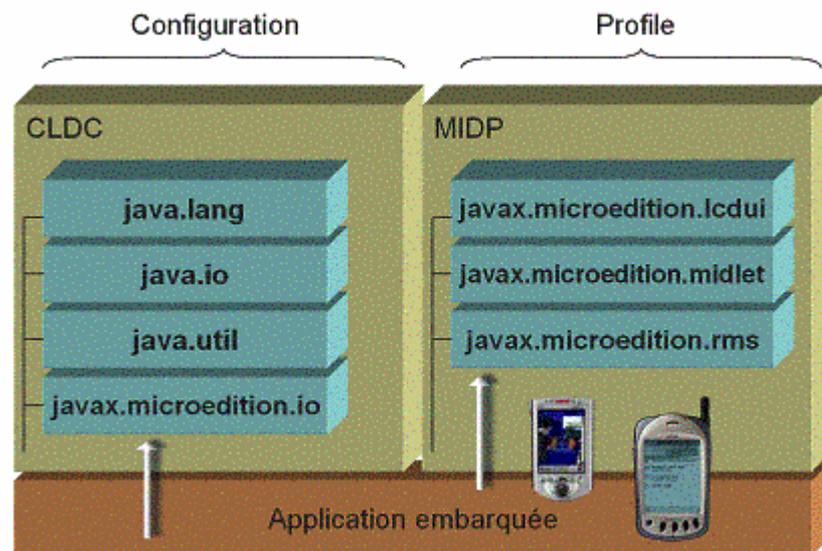


Figure 16 : La bibliothèque de J2ME

#### ❖ Les Midlets

Les Midlets sont l'élément principal d'une application Java.

## 4 .Conclusion

Dans ce chapitre On a vu les systèmes exploitation mobile sont différent entre eux et sur des appareils différent et le système exploitation mobile possède beaucoup caractéristiques la capacité de stockage est limitée, la capacité de traitement est limitée, ....

## Chapitre III

# La conception

## 1. Introduction

La souris, (*en anglais mouse ou mice*), est un périphérique de pointage (*en anglais pointing device*), servant à déplacer un curseur sur l'écran et permettant de sélectionner, déplacer, manipuler des objets grâce à des boutons. On appelle ainsi "clic" le fait d'appuyer (de cliquer) sur un bouton afin d'effectuer une action.

Notre application proposée simuler quelques les commandes de la souris d'ordinateur a distance a partir d'un téléphone mobile et nous utilisons la technologie Bluetooth pour faire la connexion entre les appareils. Le téléphone doit être compatible Bluetooth et supporter le langage J2ME et le PC doit être équipé d'un adaptateur Bluetooth.

Pour pouvoir exploiter cette approche précédente, on doit d'abord construire notre application qui doit être basé sur ses fondements. Afin de construire un tel système, on doit passer par plusieurs étapes, et comme il s'agit d'un outil logiciel, il est préférable de passer par un cycle de développement d'un logiciel.

### Etape 1 : Analyse des besoins

L'analyse des besoins est la première phase dans le cycle de vie d'un logiciel. Elle consiste à définir les services qui seront rendus à l'utilisateur et les contraintes sous lesquelles ce logiciel devra fonctionner :

#### a) Services :

Notre projet est contrôle notre ordinateur à distance à partir d'un téléphone mobile le but atteindre est de permettre à l'utilisateur utilisé la téléphone mobile au lieu la souris d'ordinateur de changer le l'emplacement de pointeur, ouvrir, fermer.

#### b) Contraintes

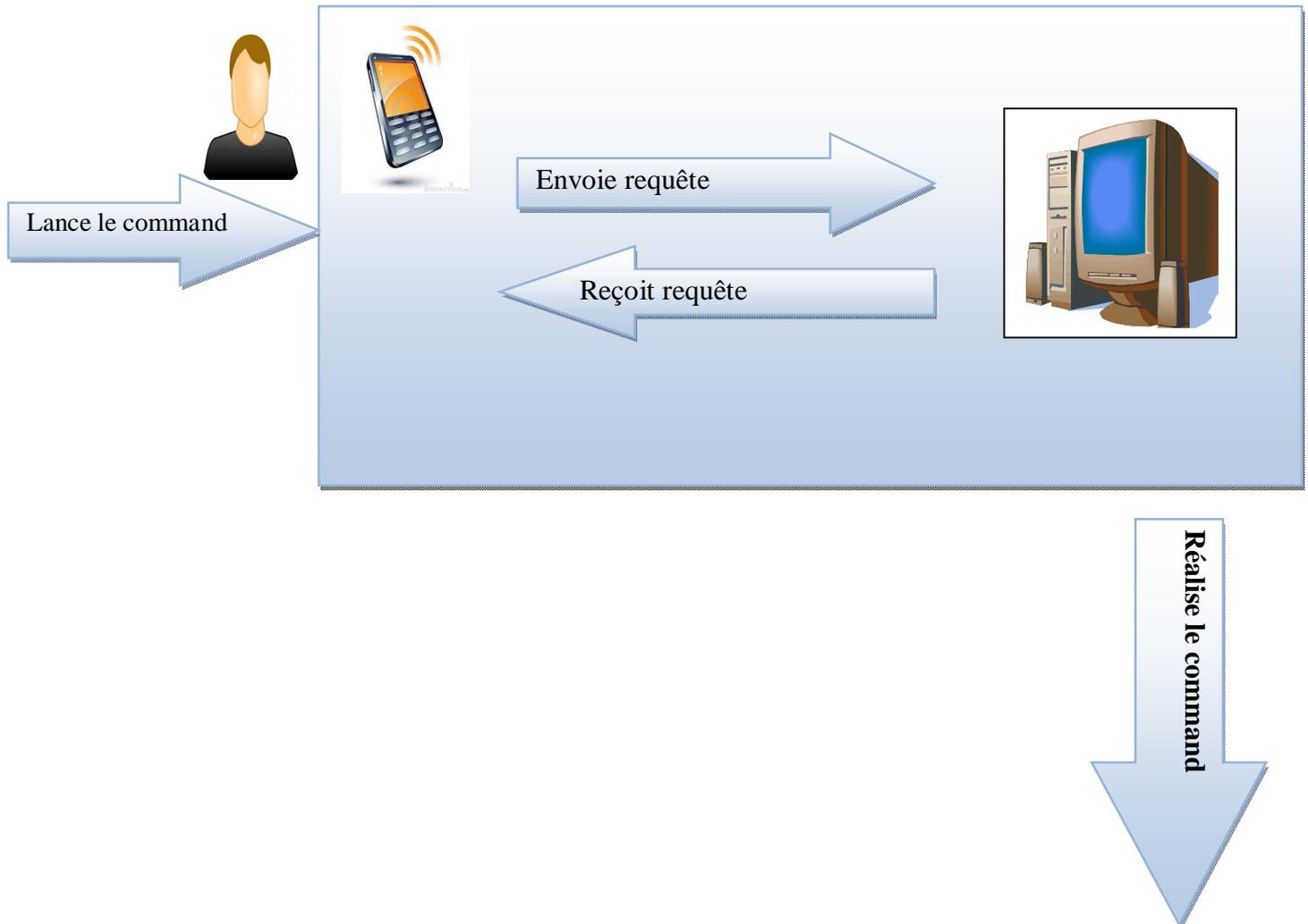
Les servies présentés précédemment sont fourni par une interface d'utilisateur qui compose des différents parties

### Etape 2 : La conception

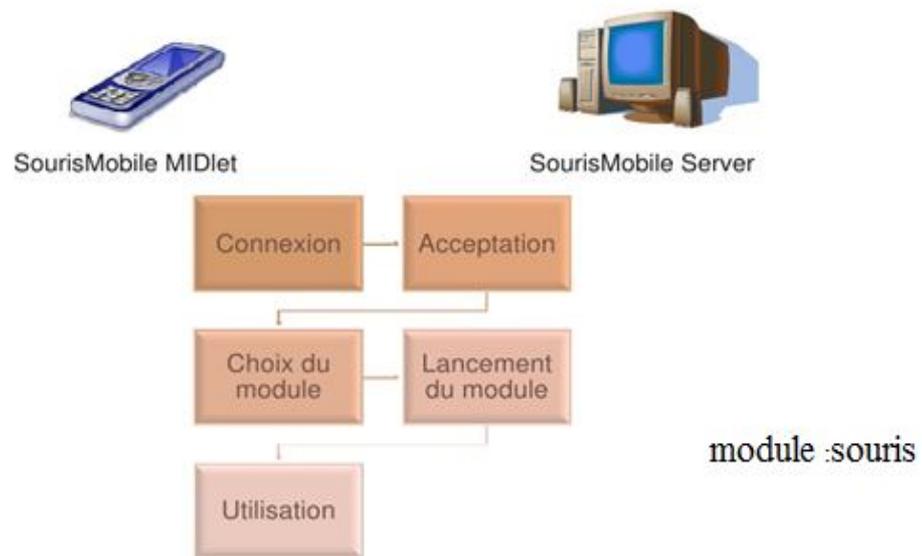
L'étape de conception est un processus itératif à plusieurs niveaux, elle compose d'une conception globale et une autre détaillée :

### 1. Conception globale

Notre application peut être spécifiée par le schéma suivant:



**Figure 17 : Architecture générale du système**



**Figure 18 : Architecture générale deuxième du système**

**a) L'utilisateur :**

L'utilisateur est la personne qui donnait des ordres au téléphone mobile pour faire des procédures nécessaires

**b) Client :**

Le téléphone mobile Bluetooth marche comme un Client pour Lance la connexion de l'ordinateur.

**• Interface graphique :**

Ce Client met à la disposition des utilisateurs une interface graphique qui facilite la communication avec le Server. Cette facilité est due à la simplicité de ce type d'interface. A travers cette interface, il est facile d'effectuer les commandes voulus.

**c) Bluetooth :**

La technologie Bluetooth pour faire la connexion entre téléphone mobile et l'ordinateur

**d) Server :**

L'ordinateur Bluetooth marche comme un serveur pour attendre la connexion de téléphone portable.

**• Interface graphique :**

Ce Server met à la disposition des utilisateurs une interface graphique qui facilite la communication avec le Client. Cette facilité est due à la simplicité de ce type d'interface. A travers cette interface, il est facile d'effectuer les commandes voulus.

## 2. Conception détaillée :

### a) L'utilisateur : ouvrir/ appuie

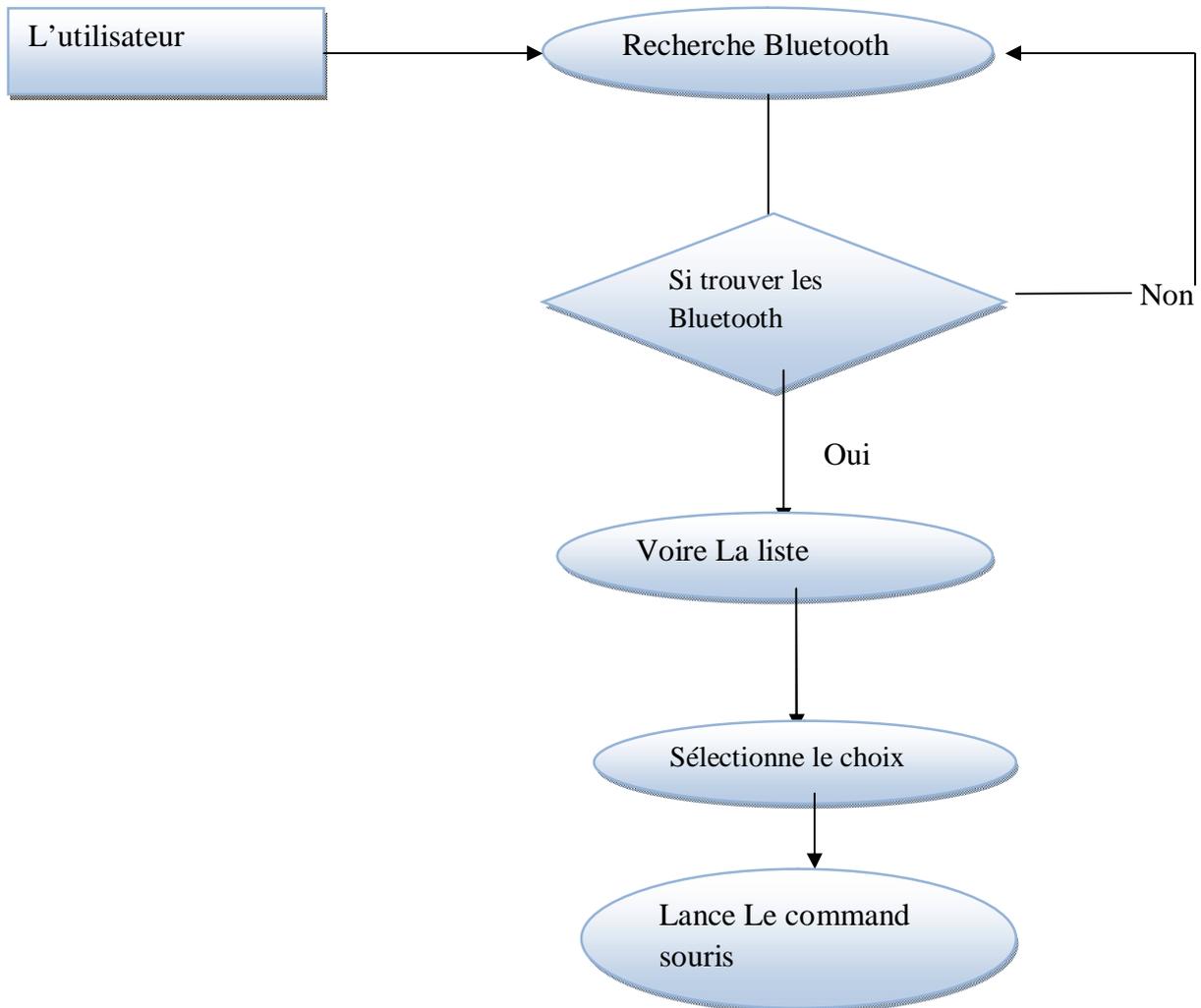


Figure 19 : Schéma de L'utilisateur

- **Recherche Bluetooth :**

Trouver des Bluetooth la première étape réalisée par l'utilisateur lorsque l'utilisateur appuie sur le choix de recherche qui existe dans le téléphone mobile(Client).

- **Voire la liste :**

Si le Client trouve Bluetooth alors le client crée un liste contient les noms des appareils (ordinateur) qui sont trouver donc l'utilisateur voir la liste.

L'utilisateur sélectionne sur le nom de l'ordinateur qui veut contrôler à distance.

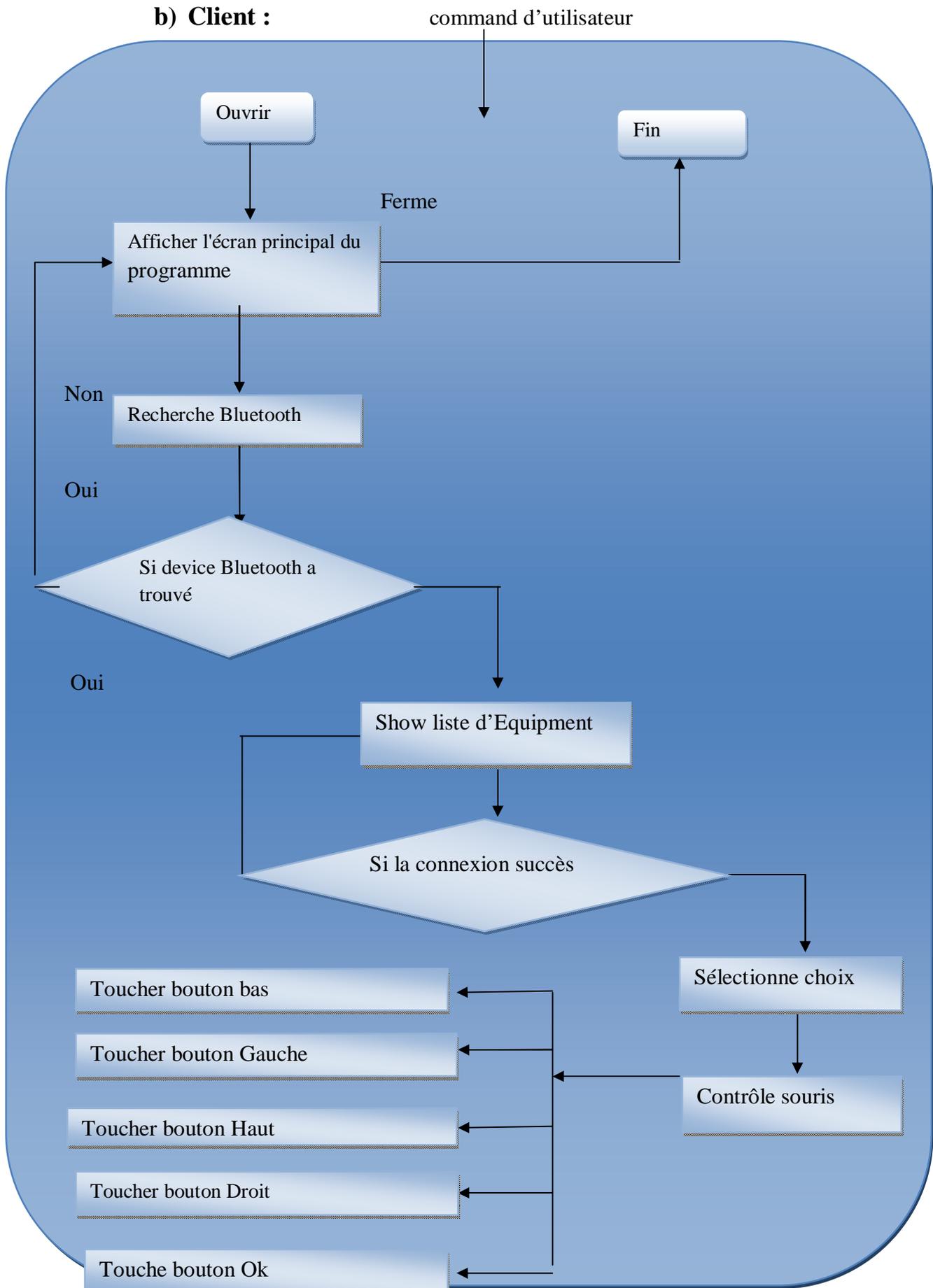
- **Sélectionne le choix :**

L'utilisateur Sélectionne sur le choix souris

- **Lance le command :**

L'utilisateur contrôle l'ordinateur à distance appuie : Droite, gauche, Haut bas....

**b) Client :**



**Figure 20 : Schéma pour le terminal mobile.**

- a) Quand l'utilisateur lance ou clique sur le bouton Rechercher, le téléphone portable
- b) Le Client établit une communication avec un serveur.
- c) Le Client envoie
- d)
- e) messages à la recherche d'appareils à proximité.
- f) Bluetooth va détecter les ordinateurs qui l'attendent.
- g) Les Serveurs qui sont à l'écoute de ces messages répondent au client.
- h) Cela permet au client de créer (affiche) une liste des serveurs potentiels qui peuvent être utilisés.
- i) l'utilisateur choisisse l'ordinateur pour le contrôler à distance.
- j) Contrôle l'ordinateur à distance

**c) Bluetooth recherche des ordinateurs :**

Bluetooth est une technologie de communication qui permet d'envoyer et de recevoir des données sans employer de fils. En utilisant les fonctions Bluetooth, vous pouvez configurer une connexion sans fil entre les appareils Bluetooth.

**Établissement de la connexion****Echange de données****Terminaison de connexion**

Lorsque toutes les communications entre les appareils est terminée, ils terminent la connexion d'une manière ordonnée. Cela permet de s'assurer que chaque appareil est prêt à communiquer avec ses pairs, ou avec un autre dispositif à une date ultérieure.

## d) Server

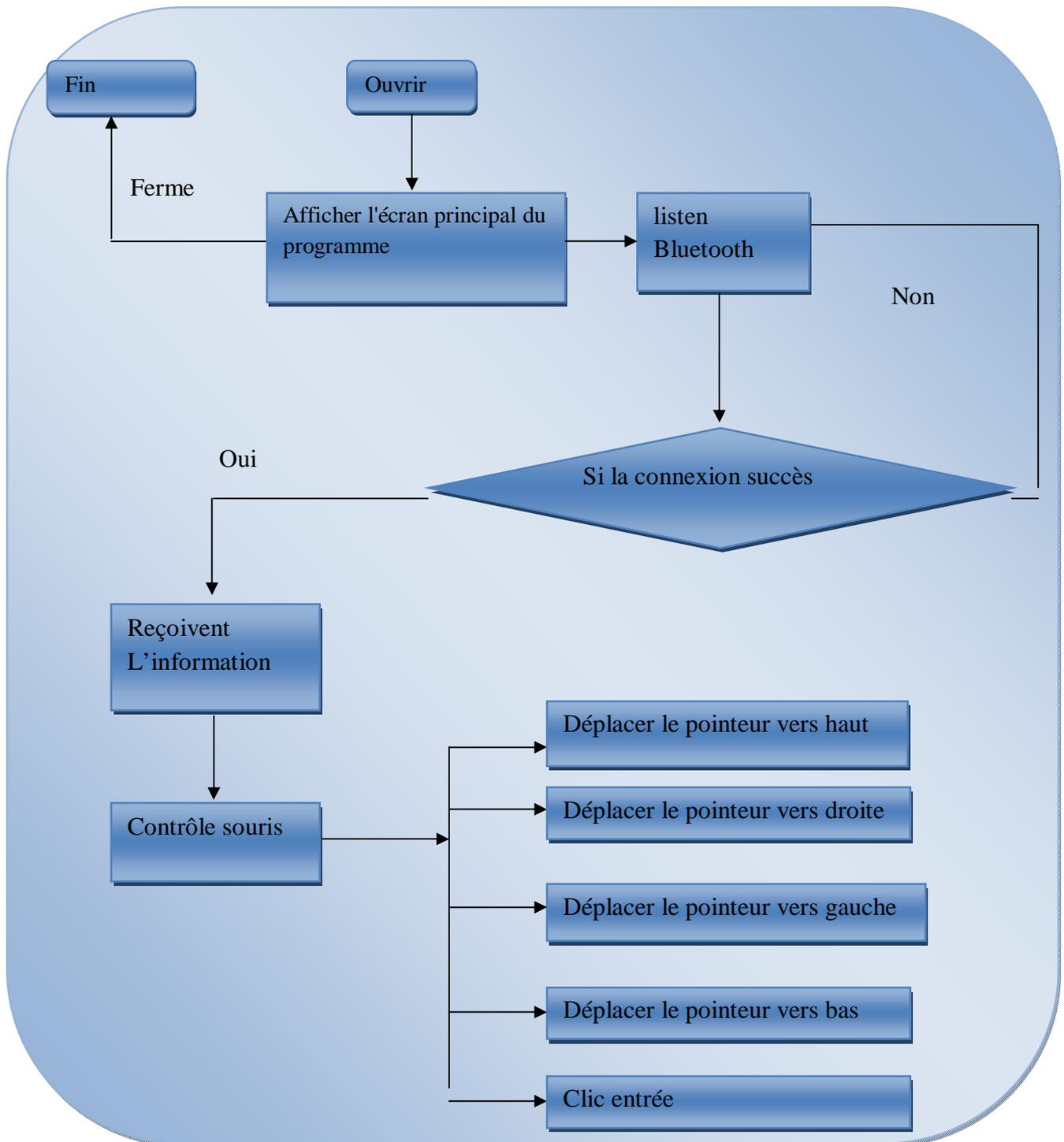


Figure 21 : Schéma pour le terminal PC.

### ➤ **Contrôle de l'ordinateur à distance**

L'ordinateur attende toujours des données à partir de téléphone portable.

L'utilisateur sélectionne un ordinateur dans la liste des ordinateurs trouvés.

L'utilisateur du téléphone portable touche un bouton sur le mobile phone.

Le sous-système qui marche sur téléphone portable va vérifier bouton est bouton défini pour but de contrôle à distance.

Si oui, il va envoyer de donnée vers l'ordinateur.

Le sous-système à côté de l'ordinateur reçoit de donnée qui est le code de bouton que l'utilisateur a touché. Il va le comparer avec les codes définis de contrôle pour contrôler de façon automatique la souris

### 3. Introduction UML

UML (Unified Modeling Language) est un langage de modélisation orientée objet développée en réponse à l'appel à propositions lancé par l'OMG (Object Management Group) dans le but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets [16]. Il est hérité de plusieurs autres méthodes telles qu'OMT1 (Object Modeling Technique) et OOSE2

(Object Oriented Software Engineering) ET Booch. Les principaux auteurs de la notation UML sont Grady Booch, Ivar Jacobson et Jim Rumbaugh.

UML est utilisé pour spécifier un logiciel et/ou pour concevoir un logiciel. Dans la spécification, le modèle décrit les classes et les cas d'utilisation vus de l'utilisateur final du logiciel. Le modèle produit par une conception orientée objet est en général une extension du modèle issu de la spécification [28].

Il enrichit ce dernier de classes, dites techniques, qui n'intéressent pas l'utilisateur final du logiciel mais seulement ses concepteurs. Il comprend les modèles des classes, des états et d'interaction. UML est également utilisée dans les phases terminales du développement avec les modèles de réalisation et de déploiement.

UML est un langage utilisant une représentation graphique. L'usage d'une représentation graphique est un complément excellent à celui de représentations textuelles. En effet, l'une comme l'autre sont ambiguës mais leur utilisation simultanée permet de diminuer les ambiguïtés de chacune d'elle. Un dessin permet bien souvent d'exprimer clairement ce qu'un texte exprime difficilement et un bon commentaire permet d'enrichir une figure. Il est nécessaire de préciser qu'un langage tel qu'UML ne suffit pas à produire un développement de logiciel de qualité à toute seule. En effet, UML n'est qu'un formalisme, ou plutôt un ensemble de formalismes permettant d'appréhender un problème ou un domaine et de le modéliser, ni plus ni moins. Un formalisme n'est qu'un outil. Le succès du développement du logiciel dépendant évidemment de la bonne utilisation d'une méthode comme UML mais il dépend surtout de la façon dont on utilise cette méthode à l'intérieur du cycle de développement du logiciel.

UML 2.0 comporte ainsi treize types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système. Ils se répartissent en deux grands groupes :

#### Diagrammes structurels ou diagrammes statiques

- diagramme de classes (Class diagram)
- diagramme d'objets (Object diagram)
- diagramme de composants (Component diagram)
- diagramme de déploiement (Deployment diagram)
- diagramme de paquetages (Package diagram)
- diagramme de structures composites (Composite structure diagram)

#### Diagrammes comportementaux ou diagrammes dynamiques

- diagramme de cas d'utilisation (Use case diagram)
- diagramme d'activités (Activity diagram)
- diagramme d'états-transitions (State machine diagram)
- Diagrammes d'interaction (Interaction diagram)
- diagramme de séquence (Sequence diagram)
- diagramme de communication (Communication diagram)
- diagramme global d'interaction (Interaction overview diagram)
- diagramme de temps (Timing diagram)

Ces diagrammes, d'une utilité variable selon les cas, ne sont pas nécessairement tous produits à l'occasion d'une modélisation. Les plus utiles pour la maîtrise d'ouvrage sont les diagrammes d'activités, de cas d'utilisation, de classes, d'objets, de séquence et d'états-transitions. Les diagrammes de composants, de déploiement et de communication sont surtout utiles pour la maîtrise d'œuvre à qui ils permettent de formaliser les contraintes de la réalisation et la solution technique.

### **3.1 Les diagrammes structurels**

#### **3.1.1 Les diagrammes de classe**

Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes. Outre les classes, ils représentent un ensemble d'interactions et de paquetages, ainsi que leurs relations. Les classes sont les descripteurs d'un ensemble d'objets qui ont une structure, un comportement et des relations similaires. Les classes sont représentées par des rectangles compartimentés. Les classes sont reliées l'une à l'autre avec plusieurs formes : les associations (une classe associée une autre classe), la dépendance (une classe dépend ou utilise une autre classe), la spécialisation (une classe est une spécialisation d'une autre classe) ...etc. [28]

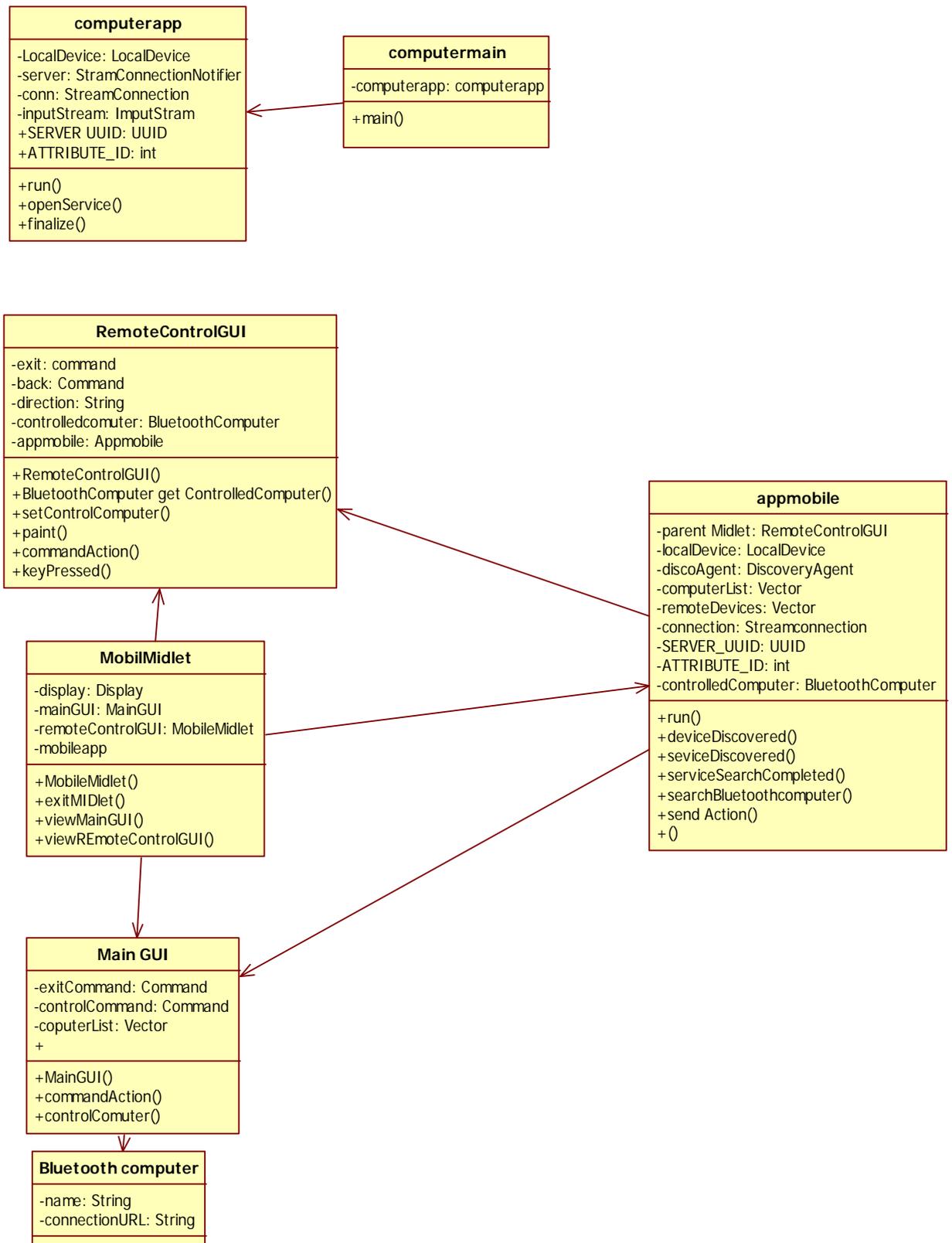


Figure 22 : Les diagrammes de classe

### 3.1.2 Le diagramme de packages

En UML, un diagramme de paquetages décrit comment un système est organisé en groupes logiques appelés paquetages tout en montrant les dépendances entre ces paquetages. En effet, Les packages (paquetages en français) permettent de regrouper et d'isoler des classes, des associations, et éventuellement d'autre packages. Un package regroupe le plus souvent un ensemble d'entités qui correspondent à une fonctionnalité bien définie. Bien souvent, c'est cette fonctionnalité qui définira le nom du package. [28]

Un package n'introduit pas de sémantique particulière. Le système à concevoir est représenté par un package racine

### 3.1.3 Le diagramme de structure composite

C'est un nouveau diagramme introduit-en UML2. Il permet de présenter la décomposition hiérarchique d'un objet, un cas d'utilisation, une collaboration, une activité ou une classe, en un ensemble de structures internes. Ces structures internes sont des ensembles d'éléments interconnectés avec leurs relations et leurs modes de communication. Ce diagramme permet de réduire la complexité des objets en donnant une vue détaillée sur l'architecture interne de ces objets durant l'exécution du système. [29]

### 3.1.4 Le diagramme de composant

Les diagrammes de composants décrivent les composants et leurs dépendances dans l'environnement de réalisation. Les diagrammes de composants sont des vues statiques de l'implémentation des systèmes qui montrent les choix de réalisation. En général ils ne sont utilisés que pour des systèmes complexes. Un composant est un élément physique qui représente une partie implémentée d'un système. Un composant peut être du code (source, binaire ou exécutable), un script, un fichier de commande, un fichier de données, une table, etc. il peut égaleriser un ensemble d'interfaces qui définissent alors le comportement offert à d'autres composants. Les services sont implémentés par les éléments du composant. En outre, chaque composant peut posséder des attributs et des opérations. Un composant est éventuellement connecté à d'autres composants (via des dépendances ou des compositions). [17]

### 3.1.5 Le diagramme de déploiement

Un diagramme de déploiement est un graphe composé de nœuds interconnectés par des liens de communication. Ce diagramme montre la disposition physique des différents matériels « Les Nœuds » qui entrent dans la composition d'un système et la répartition des instances de composants, processus et objets qui <<vivent>> sur ces matériels. Il peut également donner la structure d'une plateforme technique, mais aussi de spécifier la localisation des nœuds constitués par des unités distribuées, de préciser où se trouvent les processus et de montrer comment les objets se créent et se déplacent dans une architecture distribuée.

### **3.1.6 Le diagramme d'objets**

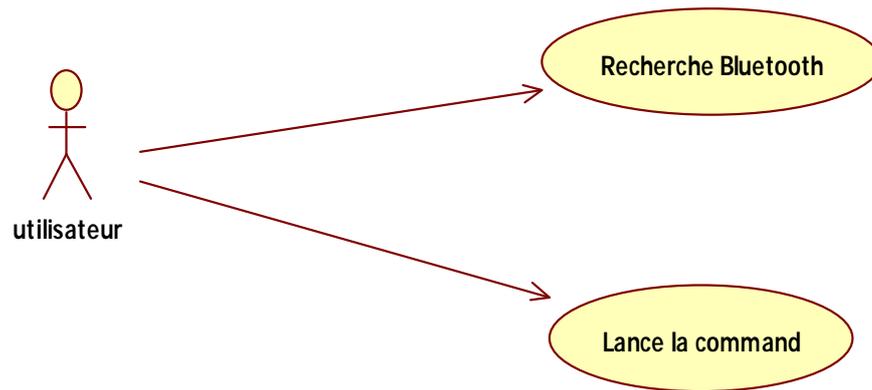
Le diagramme d'objets, dans le langage de modélisation de donnée UML, permet de représenter les instances des classes, c'est-à-dire des objets. Comme le diagramme de classes, il exprime les relations qui existent entre les objets, mais aussi l'état des objets, ce qui permet d'exprimer des contextes d'exécution. En ce sens, ce diagramme est moins général que le diagramme de classes.

Les diagrammes d'objets s'utilisent pour montrer l'état des instances d'objet avant et après une interaction, autrement dit c'est une photographie à un instant précis des attributs et objet existant. Il est utilisé en phase exploratoire.

## **3.2 Les diagrammes comportementaux**

### **3.2.1 Le diagramme de cas d'utilisation**

Les cas d'utilisations constituent le concept principal de la méthode OOSE (Object Oriented Software Engineering) d'Ivar Jakobson, un des pères d'UML. Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde la fonctionnalité du système en unités cohérentes, les cas d'utilisation, ayant un sens pour les acteurs. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique. [28]



**Figure 23 :Le diagramme de cas d'utilisation**

Le diagramme de cas d'utilisation se compose d'acteurs un cas d'utilisation. Les traits entre les acteurs et les cas d'utilisation représentent les interactions.

### 3.2.2 Le diagramme global d'interaction

Comme son nom indique, le diagramme global d'interaction donne une vue globale sur les interactions des objets actifs du système. Il regroupe des diagrammes d'activité et des diagrammes de séquence. Il est présenté soit comme un diagramme d'activités dont les activités sont remplacées par des diagrammes de séquence. Ou comme un diagramme de séquence contenant des notations du diagramme d'activité pour montrer le flux des activités.

### 3.2.3Le diagramme de temps

Le diagramme de temps est une nouveauté apparue dans UML2.0. Il permet de présenter l'interaction entre les objets actifs et leurs changements d'état sur un axe de temps. L'axe X présente le temps et contient des unités temporelles et l'axe Y montre les objets et leurs états.

### 3.2.4Le diagramme d'activité

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation [30].

Les diagrammes d'activités sont relativement proches des diagrammes d'états-transitions dans leur présentation, mais leur interprétation est sensiblement différente. Les diagrammes d'états-transitions sont orientés vers des systèmes réactifs, mais ils ne donnent pas une vision satisfaisante d'un traitement faisant intervenir plusieurs classeurs et doivent être complétés, par exemple, par des diagrammes de séquence. Au contraire, les diagrammes d'activités ne sont pas spécifiquement rattachés à un classeur particulier. On peut attacher un diagramme d'activités à n'importe quel élément de modélisation afin de visualiser, spécifier, construire ou documenter le comportement de cet élément.

La différence principale entre les diagrammes d'interaction et les diagrammes d'activités est que les premiers mettent l'accent sur le flot de contrôle d'un objet à l'autre, tandis que les seconds insistent sur le flot de contrôle d'une activité à l'autre.

Dans la phase de conception, les diagrammes d'activités sont particulièrement adaptés à la description des cas d'utilisation. Plus précisément, ils viennent illustrer et consolider la description textuelle des cas d'utilisation. De plus, leur représentation sous forme d'organigrammes les rend facilement intelligibles et beaucoup plus accessibles que les diagrammes d'états-transitions. .

#### **a. Action**

Une action correspond à un traitement qui modifie l'état du système. Cette action peut être appréhendée soit à un niveau élémentaire proche d'une instruction en termes de programmation soit à un niveau plus global correspondant à une ou plusieurs opérations.

#### **b. Transition et flot de contrôle**

Dès qu'une action est achevée, une transition automatique est déclenchée vers l'action suivante. Il n'y a donc pas d'événement associé à la transition. L'enchaînement des actions constitue le flot de contrôle.

#### **c. Activité**

Une activité représente le comportement d'une partie du système en termes d'actions et de transitions. Une activité est composée de trois types de nœuds :

- Nœud d'exécution (action, transition),

- Nœud de contrôle (nœud initial, nœud final, flux de sortie, nœud fourche, nœud de jonction, nœud de fusion-test, nœud de test-décision, pin d'entrée et de sortie),
- Nœud d'objet.

Une activité peut recevoir des paramètres en entrée et en produire en sortie.

#### **d. Le nœud fourche (Fork)**

Un nœud fourche (fork) permet à partir d'un flot unique entrant de créer plusieurs flots concurrents en sortie de la barre de synchronisation.

#### **e. Nœud de jonction (Join)**

Un nœud de jonction (join) permet, à partir de plusieurs flots concurrents en entrée de la Synchronisation, de produire un flot unique sortant. Le nœud de jonction est le symétrique du nœud de fourche.

#### **f. Nœud de test-décision**

Un nœud de test-décision permet de faire un choix entre plusieurs flots sortants en fonction des conditions de garde de chaque flot. Un nœud de test-décision n'a qu'un seul flot en entrée. On peut aussi utiliser seulement deux flots de sortie : le premier correspondant à la condition vérifiée et l'autre traitant le cas sinon.

#### **g. Nœud de fusion-test**

Un nœud de fusion-test permet d'avoir plusieurs flots entrants possibles et un seul flot sortant. Le flot sortant est donc exécuté dès qu'un des flots entrants est activé.

#### **h. Pin d'entrée et de sortie**

Un pin d'entrée ou de sortie représente un paramètre que l'on peut spécifier en entrée ou en sortie d'une action. Un nom de donnée et un type de donnée peuvent être associés au pin. Un paramètre peut être de type objet.

#### **i. Flot de données et nœud d'objet**

Un nœud d'objet permet de représenter le flot de données véhiculé entre les actions. Les objets peuvent se représenter de deux manières différentes : soit en utilisant le pin d'objet soit en représentant explicitement un objet.

### **j. Partition**

UML permet aussi d'organiser la présentation du diagramme d'activité en couloir d'activités. Chaque couloir correspond à un domaine de responsabilité d'un certain nombre d'actions.

Les flots d'objets sont aussi représentés dans le diagramme. L'ordre relatif des couloirs de responsabilité n'est pas significatif.

### **k. Représentation d'actions de communication**

Dans un diagramme d'activité, comme dans un diagramme de temps, des interactions de communication liées à certains types d'événement peuvent se représenter. Les types d'événement concernés sont :

- signal,
- écoulement du temps.

### **3.2.5 Diagramme d'états-transitions**

Un diagramme d'états-transitions présente un automate à états finis. Il permet ainsi de décrire les changements d'états d'un objet ou d'un composant.

Un état se caractérise par sa durée et sa stabilité.

Une transition représente le passage instantané d'un état vers un autre.

Une transition est déclenchée :

- soit par un événement.
- soit automatiquement lorsque aucun événement déclencheur est spécifié.

### **3.2.6 Diagramme de séquence**

IL permet d'étudier les interactions entre les objets et constitue une partie dynamique de système d'information. IL montre d'une façon séquentielle les envois des messages qui interviennent entre les objets, il peut également montrer les flux des données :

### **3.2.7 Les diagrammes de communication**

Qui donne une représentation spatiale des objets, des liens des interactions, il est équivalent au diagramme de séquence

#### **4. Points forts et points faibles**

##### **4.1 Les points forts d'UML**

UML est un langage formel et normalisé

- gain de précision
- gage de stabilité
- encourage l'utilisation d'outils

UML est un support de communication performant

- Il cadre l'analyse.
- Il facilite la compréhension de représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel.

##### **4.2 Les points faibles d'UML:**

La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.

la nécessité de s'accorder sur des modes d'expression communs est vitale en informatique. UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.

Le processus (non couvert par UML) est une autre clé de la réussite d'un projet.

l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue

Les auteurs d'UML sont tout à fait conscients de l'importance du processus, mais l'acceptabilité industrielle de la modélisation objet passe d'abord par la disponibilité d'un langage d'analyse objet performant et standard.

## 5. Conclusion

Nous avons présenté dans cette partie les différentes phases pour aboutir à la réalisation de notre application et qui peuvent être schématisées par le schéma décrit ci-avant. En suite une interprétation de ses vues de conception, en terme des diagrammes et des schémas. Et les résultats et les implémentations de ces derniers sont donnés dans le chapitre suivant.

## Chapitre VI

# L'implémentation

## 1. Introduction

Cette partie présente le dernier volet de ce rapport. Elle a pour objet d'exposer le travail réalisé.

Dans la phase de l'implémentation, il y a deux problèmes principaux : programmation pour la communication entre le téléphone portable et l'ordinateur ; et la programmation pour le contrôle de la souris. Ils sont présentés dans cette partie.

D'abord, nous commençons par la présentation de l'environnement logiciel et technologique utilisé.

## 2. Environnement de développement partie client: NetBeans

On a utilisé l'environnement NetBeans pour simplifier le travail afin de développer notre application mobile. L'environnement NetBeans adopte l'IDE (*Integrated Development Environment*) à Java gratuitement. L'IDE est un logiciel open source développé par SUN Microsystems utilisé pour exécuter, compiler. Un IDE est dédié à un seul langage de programmation. On peut également trouver dans un IDE un système de gestion de versions et différents outils pour faciliter la création de l'interface graphique GUI.

Par rapport aux autres environnements comme Eclipse, Wireless Toolkit... le choix a été adopté par rapport à son intégrité de bibliothèques de langage J2ME, faciliter l'interface de développement et sa flexibilité.

## 3. Langage de programmation : Java

Dans le domaine mobile il y a plusieurs systèmes d'exploitation utilisés dans les téléphones portables afin d'adopter notre application aux différents systèmes mobiles on a choisi le langage Java, comme dernier a été utilisé dans la majorité des systèmes mobiles et adapté à n'importe quel système d'exploitation.

La particularité principale de Java est que les logiciels écrits dans ce langage sont très facilement portables sur plusieurs systèmes d'exploitation tels qu'UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications.

C'est la plate-forme qui garantit la portabilité des applications développées en Java. Ce dernier permet de développer des applications client serveur. Java a donné naissance à un système d'exploitation (JavaOS), à des environnements de développement (éclipse/JDK), des machines virtuelles (MSJVM, JRE) applicatives multi plates-formes (JVM), une déclinaison pour les périphériques mobiles/embarqués (J2ME).

Aujourd'hui, Java (sous la forme de JavaME) trouve une nouvelle niche dans la création d'applications RIA (Rich Internet Applications), des applications qui proposent des fonctionnalités, notamment des interfaces, plus évoluées à la fois sur Internet et sur les téléphones portables.

#### **4. Technologie adoptée : RME**

J2ME (Java 2 Micro Edition) ou Java ME est le Framework Java spécialisé dans les applications mobiles. Des plates-formes Java compatibles avec J2ME sont embarquées dans de nombreux téléphones.

Les téléphones se caractérisent par leur fragmentation qui se traduit sur Java ME : les caractéristiques des mobiles sont différentes d'un constructeur à un autre rendant ainsi complexe et coûteux le développement d'applications mobile.

L'architecture J2M

**Les profile** : Ils permettent à une certaine catégorie de terminaux d'utiliser des caractéristiques communes telles que la gestion de l'affichage, des évènements d'entrées/sorties (pointage, clavier, ...) ou des mécanismes de persistance (Base dedu JCP (Java CommunityProcess))

Les machines virtuelles : En fonction de la cible, la machine virtuelle pourra être allégée afin de consommer plus ou moins de ressources (KVM, CVM, ...)

**Le système d'exploitation** : L'environnement doit s'adapter au système d'exploitation existant (Windows CE, Palm Os, SavaJe, ...)

## 5. Les configurations

Les configurations définissent les caractéristiques de bases d'un environnement d'exécution pour un certain type de machine possédant un ensemble de caractéristiques et de ressources similaires. Elles se composent d'une machine virtuelle et d'un ensemble d'API de base. Deux configurations sont actuellement définies :

- CLDC (Connected Limited Device Configuration)
- CDC (ConnectedDevice Configuration)

La CLDC 1.0 est spécifiée dans la JSR 030 : elle concerne des appareils possédant des ressources faibles (moins de 512 Kb de RAM, faible vitesse du processeur, connexion réseau limitée et intermittente) et une interface utilisateur réduite (par exemple un téléphone mobile ou un PDA "entrée de gamme"). Elle s'utilise sur une machine virtuelle KVM. La version 1.1 est le résultat des spécifications de la JSR 139 : une des améliorations les plus importantes est le support des nombres flottants.

La CDC est spécifiée dans la JSR 036 : elle concerne des appareils possédant des ressources plus importantes (au moins 2Mb de RAM, un processeur 32 bits, une meilleure connexion au réseau), par exemple un set top box ou certains PDA "haut de gamme". Elle s'utilise sur une machine virtuelle CVM.

## 6. Les profils

Les profils se composent d'un ensemble d'API particulières à un type de machines ou à une fonctionnalité spécifique. Ils permettent l'utilisation de fonctionnalités précises et doivent être associés à une configuration. Ils permettent donc d'assurer une certaine modularité à la plateforme J2ME. Il existe plusieurs profils tels que MIDP 1.0 , WMA , RMI ...

MIDP est un profil standard qui n'est pas défini pour une machine particulière mais pour un ensemble de machines embarquées possédant des ressources et une interface graphique limitée.

Sun a développé un profil particulier nommé KJava pour le développement spécifique sur Palm. Ce profil a été remplacé par un nouveau profil nommé MIDP for Palm OS.

Cette multitude de profils peut engendrer un certain nombre de problème lors de l'exécution d'une application sur différents périphériques car il n'y a pas la certitude d'avoir à disposition les profils nécessaires. Pour résoudre ce problème, une spécification particulière issue des travaux de la JSR 185 et nommée « Java Technology for the Wireless Industry (JTWI) » a été développée. Cette spécification impose aux périphériques au minimum : CLDC 1.0, MIDP 2.0, Wireless Messaging API 1.1 et Mobile Media API 1.1. Son but est donc d'assurer une meilleure compatibilité entre les applications et les différents téléphones mobiles sur lesquelles elles s'exécutent.

## 7. Midlet

MIDlets sont gérés par le logiciel de gestionnaire d'applications intégré à l'appareil en raison d'une application en cours d'exécution peut être interrompue à tout moment par des événements extérieurs, comme les appels téléphoniques entrants. Le gestionnaire d'application contrôle les activités des MIDlets multiples dans l'environnement Java ME sur un dispositif et peut sélectionner les MIDlets sont actives à un moment donné en démarrant et en leur faisant une pause individuellement.

Il ya trois états possibles dans un MIDlet du cycle de vie:

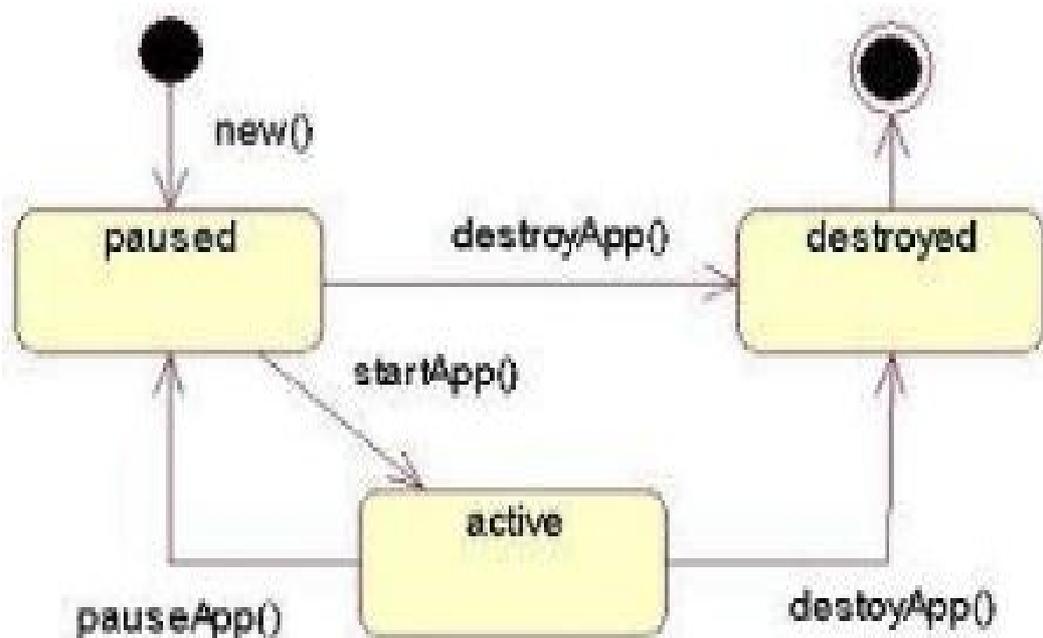


Figure 24 : Cycle de vie d'une MIDLET

Le gestionnaire d'application crée MIDlets, généralement en réponse à une demande des utilisateurs, dans la pause de l'État. À un certain moment après la création, le MIDlet est activé et ses changements d'état de pause à active. Lorsque le MIDlet soit mis fin à ses changements d'état de destruction de soit active ou en pause.

## 8. Communication entre téléphone portable (Client) et l'ordinateur (Server):

Chaque application Bluetooth se compose de 5 parts suivants : l'initialisation de stack, la gestion de l'équipement, la découverte de l'équipement, la découverte du service, la communication. Les APIs de chaque part sont utilisées dans le système que je construis. La partie suivant va présenter comment utiliser ces APIs dans ce système.

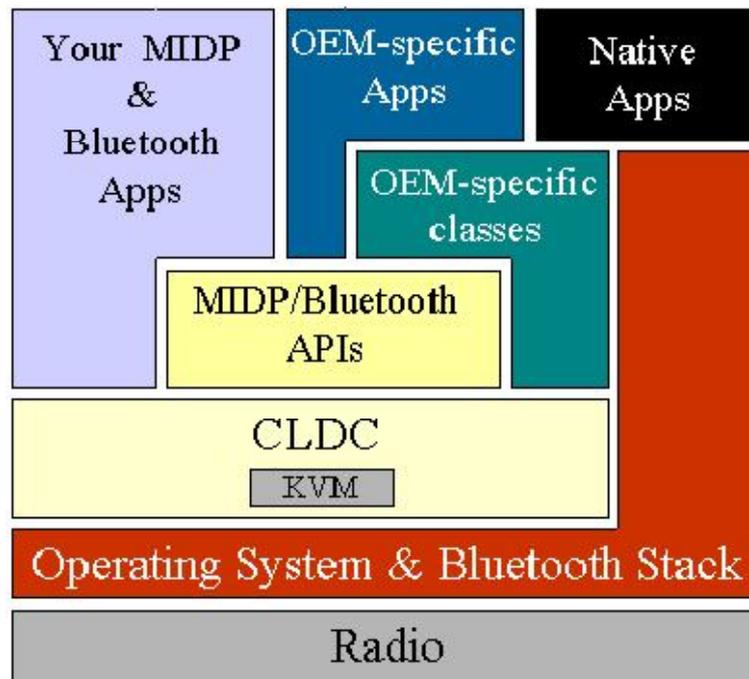


Figure 25 : APIs de Bluetooth en architecture de J2ME

Dans ce système le téléphone portable est programmé comme client, cependant l'ordinateur est programmé comme serveur.

### 8.1 Initialisation de stack

Stack de Bluetooth est responsable de contrôler l'équipement Bluetooth, on doit donc initialiser le stack avant que vous puissiez faire toute autre chose.

```
...  
  
// set the port number  
  
BCC.setPortNumber("COM1");  
  
// set the baud rate  
  
BCC.setBaudRate(50000);  
  
// set the connectable mode  
  
BCC.setConnectable(true);  
  
// set the discovery mode to Limited Inquiry Access Code  
  
BCC.setDiscoverable(DiscoveryAgent.LIAC);  
  
...
```

## 8.2 La gestion de l'équipement

Les API de Java pour Bluetooth fournissent deux classes `LocalDevice` et `RemoteDevice` qui sont destinées à la gestion de l'équipement. On utilise la classe `LocalDevice` pour recueillir des informations concernant cet équipement telle que : type d'équipement, type de services fournis. Par exemple :

```
...

// retrieve the local Bluetooth device object

LocalDevice local = LocalDevice.getLocalDevice();

// retrieve the Bluetooth address of the local device

String address = local.getBluetoothAddress();

// retrieve the name of the local Bluetooth device

String name = local.getFriendlyName();
```

Cependant, on utilise la classe `RemoteDevice` pour collecter des informations de l'équipement à distance (dans le cadre du signal Bluetooth). Par exemple :

```
...

// retrieve the device that is at the other end of

// the Bluetooth Serial Port Profile connection,

// L2CAP connection, or OBEX over RFCOMM connection

RemoteDevice remote =

RemoteDevice.getRemoteDevice(

javax.microedition.io.Connection c);

// retrieve the Bluetooth address of the remote device

String remoteAddress = remote.getBluetoothAddress();

// retrieve the name of the remote Bluetooth device

String remoteName = local.getFriendlyName(true);
```

```
...
```

La classe de RemoteDevice fournit également des méthodes pour authentifier, autoriser, ou chiffrer des données transférées entre les équipements locaux et à distance.

### 8.3 La découverte de l'équipement

J2ME fournit la classe DiscoveryAgent et interfaceDiscoveryListener pour trouver les équipements et les accéder. L'équipement Bluetooth peut utiliser la classe DiscoveryAgent pour obtenir la liste d'équipements accessibles.

La méthode de DiscoveryAgent.startInquiry place le dispositif dans un mode d'enquête. Par exemple :

```
...  
  
// retrieve the discovery agent  
  
DiscoveryAgent agent = local.getDiscoveryAgent();  
  
// place the device in inquiry mode  
  
booleancomplete = agent.startInquiry();
```

Quand l'enquête est accomplie ou annulée, DiscoveryListener.inquiryCompleted est appelé. On utilise la méthode DiscoveryAgent.retrieveDevices pour avoir la liste d'équipement, par exemple :

```
...  
  
// retrieve the discovery agent  
  
DiscoveryAgent agent = local.getDiscoveryAgent();  
  
// return an array of pre-known devices  
  
RemoteDevice[] devices =  
  
agent.retrieveDevices(DiscoveryAgent.PREKNOWN);  
  
...
```

#### 8.4 La découverte du service

Avant qu'un service puisse être découvert, il doit d'abord être enregistré sur le serveur. Le serveur est responsable de :

- Créer un service record qui décrit le service offert
- Ajouter un service record à la base de données de découverte du service du serveur (SDDB), il est visible et disponible pour les clients potentiels
- Enregistrer les mesures de sécurité de Bluetooth liées au service
- Accepter les connexions en provenance du client.
- Mettre à jour un service record dans la base de donnée SDDB.
- Supprimer un service record

Pour créer un nouveau service record qui représente les service , on appelle `Connector.open` avec un argument d'URL de connexion de serveur :

```
...  
  
StreamConnectionNotifier service =  
  
(StreamConnectionNotifier) Connector.open("someURL");  
  
Obtenir le service record créé par l'équipement de serveur :  
  
ServiceRecordsr = local.getRecord(service);  
  
Indiquez que le service est prêt à accepter une connexion du client :  
  
StreamConnection connection =  
  
(StreamConnection) service.acceptAndOpen();
```

Quand le serveur est prêt à quitter, fermer la connexion et enlever le service record :

```
service.close();  
  
...
```

### 9. La communication : transmission entre le téléphone portable et l'ordinateur

Pour que l'équipement local utilise le service sur l'équipement à distance, tous les deux doivent partager un protocole commun de communication.

Le serveur (ordinateur) doit :

- Construire l'URL qui indique comment connecter à serveur et le stocke dans service record
- Rendre le service record disponible au client
- Accepter la connexion à partir du client

- Envoyer et recevoir des données du et au serveur

Pour établir la connexion au serveur, le client (téléphone portable) doit :

- Initialiser la découverte de service pour rendre le service record
- Construire la connexion URL utilisant le service record
- Ouvrir une connexion au serveur
- Envoyer et recevoir des données du et au serveur

### **10. Programmation pour le contrôle automatique de la souris**

Pour contrôler de façon automatique la souris, on utilise la classe `java.awt.Robot`. à chaque fois l'ordinateur reçoit la donnée qui est le code de bouton touché à partir de téléphone portable, il va analyser cette donnée pour avoir une action correspondante sur la souris.

#### **Interfaces application mobile (partie clients) :**

1. Select recherche
2. En attente
3. Affiche liste de pc
4. Select pc qui je veux contrôler
5. En attente
6. Select souris
7. contrôler

#### **Interfaces d'application (partie PC) :**

Frame En attente les commandes de client.

### **11. Conclusion**

Dans ce chapitre, nous avons présenté les détails d'implémentation de l'application de cette application.



## **Conclusion général**

Le projet single sign on est permet de l'utilisateur contrôle l'ordinateur a distance a partir téléphone mobile

On a choisi ce thème pour l'intérêt qu'il port dans des domaine réseaux informatique et aussi pour enrichir nos connaissance dans l'espoir de les appliquer dans la vie professionnelle.

Sa préparation nous a beaucoup appris ,puisque'elle a permis de valider les connaissances acquises sur le plan théorique au niveau pratique.

Durant notre recherche on a remarqué le projet « « n'est pas riche en documentation spécialement dans ce qui relève de la pratique.

Le champ d'action de notre sujet, étant très vaste, nous n'avons pas la prétention de l'avoir totalement épuisé, mais nous espérons avoir répondu aux questions que nous étions en train de nous poser, dans le temps et dans l'espace, laissant l'occasion à d'autres personnes pouvant traiter les autres aspects de la question.

### **Perspectives**

Le travail qu'on a réalise peut être amélioré et enrichi d'en faire un système me plus performant.

Parmi les perspective a rendre en compte pour améliorer le fonctionnement du système nous citons notamment :

L'intégration de plus de command d'applications(gestion des touches manquantes du clic droit déplacement de fenêtres.....).

La mise en place du clavier.



## Bibliographie

- [1][http://fr.wikipedia.org/wiki/Application\\_mobile](http://fr.wikipedia.org/wiki/Application_mobile)- Consulté le 20/3 /2013
- [2]<http://www.definitions-webmarketing.com/Definition-Application-mobile>- Consulté le 25/ 3/2013
- [3]<http://www.mobileenfrance.com/2009/08/18/application-embarquee-ou-service-mobile-%E2%80%93-partie-1-definitions/>- consulté 20/3/2013
- [4] <http://fr.clever-age.com/veille/blog/comment-aborder-un-projet-de-mobilite.html>  
consulté 2/04/2013
- [5] conception&développement d'application pour Smartphone&Tablettes
- [6]<http://olivierguillet.com/2012/02/les-differents-types-dapplications-mobiles-natives-web-apps-hybrides-flash/>- Consulté le 28 / 3/2013
- [7]<http://www.journaldugeek.com/2013/01/24/facebook-et-google-au-top-des-applications-mobiles/>-  
consulté le 19/04/2013
- [8]<http://www.commentcamarche.net/contents/wireless/wlintro.php3> - Consulté le 28/ 3/2013
- [9]<http://www.commentcamarche.net/contents/wireless/wwan.php3>- Consulté le 28/3 /2013
- [10] <http://www.distrimedia.fr/technologies-gsm-cellulaire.html>- Consulté le 28 /3 /2013
- [11]<http://www.commentcamarche.net/contents/wireless/wman.php3>-Consulté le 28 /3 /2013
- [12]<http://www.commentcamarche.net/contents/wimax/wimax-intro.php3>-Consulté le 28 /3 /2013
- [13]<http://www.commentcamarche.net/contents/wireless/wlan.php3>-Consulté le 28/3 /2013
- [14]<http://www.commentcamarche.net/contents/wifi/wifiintro.php3>-Consulté le 28/ 3/2013
- [15]<http://www.commentcamarche.net/contents/wireless/wpan.php3>-Consulté le 28 / 3/2013
- [16]<http://www.commentcamarche.net/contents/bluetooth/bluetooth-intro.php3>-Consulté le 28/3 /2013

- [17] [http://www.maisondugsm.com/a/encyclopedie/definition/4/systeme\\_d\\_exploitation.html](http://www.maisondugsm.com/a/encyclopedie/definition/4/systeme_d_exploitation.html)-  
Consulté le 30 / 3/2013
- [18] [http://fr.wikipedia.org/wiki/Syst%C3%A8me\\_d'exploitation\\_mobile](http://fr.wikipedia.org/wiki/Syst%C3%A8me_d'exploitation_mobile)- Consulté le 3/ 4/2013
- [19] <http://www.commentcamarche.net/contents/1092-systeme-d-exploitation#top>- Consulté le 3/  
4/2013
- [20] [http://fr.wikipedia.org/wiki/IOS\\_\(Apple\)](http://fr.wikipedia.org/wiki/IOS_(Apple))- Consulté le 13/ 4/2013
- [21] <http://fr.wikipedia.org/wiki/Android>- Consulté le 14/ 4/2013
- [22] [http://fr.wikipedia.org/wiki/Windows\\_CE](http://fr.wikipedia.org/wiki/Windows_CE)- Consulté le 13/ 4/2013
- [23] <http://www.techno-science.net/?onglet=glossaire&definition=7738>- Consulté le 13/ 4/2013
- [24] [http://fr.wikipedia.org/wiki/Symbian\\_OS](http://fr.wikipedia.org/wiki/Symbian_OS)- Consulté le 8/ 4/2013
- [25] [http://fr.wikipedia.org/wiki/.NET\\_Compact\\_Framework](http://fr.wikipedia.org/wiki/.NET_Compact_Framework)- Consulté le 13/ 4/2013
- [26] [http://msdn.microsoft.com/fr-fr/library/9s7k7ce5\(v=vs.90\).aspx](http://msdn.microsoft.com/fr-fr/library/9s7k7ce5(v=vs.90).aspx)- Consulté le 8/ 4/2013
- [27]Outils et méthodes pour le développement des application embarquées NGUYEN Van Tien
- [28]Rum Baugh J., Jacobson I,Booch G. «Uml 2.0 guide de référence.»  
CampusPress, 2004.
- [29] Roques, P. & Vallée, F, «Uml en action (2 ed.) », Eyrolles, 2003
- [30][JosephGabay ,David Gabay ,«uml 2 analyse et conception mise en œuvre  
Guidée avec études de cas» DUNOD ,2008



**Résumé :** L'essor des appareils mobiles, avec leur performance est croissant, a favorisé la multiplication de leurs usages et multiples ces domaines. Est devenue une nécessité dans notre vie quotidienne pour tous les segments de la société, C'est ainsi que le développement des applications mobiles est apparu et se positionne actuellement comme une activité majeure du génie logiciel. Cependant les développeurs sont confrontés à certaines difficultés dans la conception de telles applications à cause des limitations qu'imposent les architectures matérielles des appareils mobiles, la capacité de stockage est limitée, la capacité de traitement est limitée, de l'hétérogénéité des technologies et des protocoles qui influence directement la structure de l'application. et l'objectif de cette étude est Connaître les applications mobiles et leurs caractéristique et leur les technologies et aussi ses systèmes d'exploitation.

**Mots-clés:** les applications de téléphonie mobile, J2ME, équipement mobile, UML et Bluetooth.

**ملخص:** صعود الأجهزة النقالة مع أدائها المتنامي، وهذا لكثرة استخداماتها و تعدد مجالاتها. فأصبحت تشكل ضرورة حتمية في حياتنا اليومية ولجميع فئات المجتمع، هكذا برز تطوير التطبيقات النقالة ووضعنا نفسها بوصفها نشاطا رئيسيا في هندسة البرمجيات. ومع ذلك، والمطورين يواجهون بعض الصعوبات في تصميم مثل هذه التطبيقات نظرا للقيود التي تفرضها بنية هذه الأجهزة، سعة التخزين محدودة، قدرة معالجتها محدودة، عدم تجانس التقنيات والبروتوكولات التي تؤثر بشكل مباشر على بنية التطبيق، والهدف من هذه الدراسة معرفة التطبيقات النقالة وخصائصها وتقنياتها وكذلك أنظمة التشغيل الخاصة التي تتعلق بها.

**الكلمات المفتاحية:** تطبيقات الهاتف النقال، J2ME، المعدات المنقولة، UML وبلوتوث.

**Summary:** The rise of mobile devices, with their performance is growing, has favored the multiplication of their uses and multiple these areas. Has become a necessity in our daily life for all segments of society, this is how the mobile application development has emerged and is positioning itself as a major activity in software engineering. However, developers are confronted with some difficulties in the design of such applications due to limitations imposed by the hardware architecture of mobile devices, the storage capacity is limited, and the processing capacity is limited, the heterogeneity of technologies and protocols which directly influences the structure of the application. And the objective of this study is to know the mobile applications and their characteristics and their technologies and also its operating systems.

**Keywords:** mobile applications, J2ME, mobile equipment, UML and Bluetooth.

