

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A. MIRA de Bejaia



Faculté de Sciences Exactes  
Département d'Informatique

# MÉMOIRE DE FIN DE CYCLE

En vue de l'obtention du diplôme de Master professionnel en génie logiciel

## Thème :

Conception et réalisation d'une application sous Android pour la consultation des états des navires au port de Bejaia

---

Travail réalisé par HITACHI Ghiles et KAIBA Yanis

Encadré par Dr. BRAHAMI Houda née ELBOUHISSI

Présenté devant le jury constitué de :

Président : M<sup>me</sup> LOUKADIR O

Examineur : M<sup>r</sup> ALLEM K

Examineur : M<sup>r</sup> CHENNA A

Promotion 2016 / 2017

---

# REMERCIEMENTS

Nos premiers remerciements sont adressés à nos familles et nos amis qui nous ont soutenu, non seulement durant la réalisation de ce travail, mais aussi tout au long de notre formation, et ce, depuis notre première année à l'université.

Nous remercions également nos enseignants à l'université de Bejaia pour leur contribution à notre formation.

Nous adressons évidemment nos plus sincères remerciements pour notre encadreur M<sup>me</sup> BRAHAMI pour sa disponibilité, ses précieux conseils et ses efforts remarquables à l'égard des étudiants, nous lui souhaitons par l'occasion une bonne continuation dans ce sens de responsabilité qui mérite un grand respect.

Nous tenons aussi à remercier les membres du jury pour avoir assisté à notre soutenance.

Nous vous remercions autant, vous, qui êtes en train de lire ses lignes.

---

# TABLE DES MATIÈRES

TABLE DES MATIÈRES	ii
LISTE DES FIGURES	vi
LISTE DES TABLEAUX	viii
LISTE DES ABRÉVIATIONS	ix
INTRODUCTION GÉNÉRALE	1
1. L'ENTREPRISE PORTUAIRE DE BEJAIA	2
1.1 Introduction .....	2
1.2 Présentation générale de l'organisme d'accueil .....	3
1.3 Création de l'EPB .....	3
1.4 Missions et Activités de l'EPB .....	4
1.4.1 Ses missions .....	4
1.4.2 Ses Activités .....	4
1.5 Présentation des différentes structures de l'entreprise .....	5
1.5.1 Organigramme.....	5
1.5.2 Présentation du centre informatique .....	6
1.5.3 Les missions du centre informatique.....	6
1.5.4 Organisation humaine du centre de système d'informations .....	6
1.6 La Problématique .....	7
1.7 Conclusion .....	8
2. LA TECHNOLOGIE DU MOBILE	9
2.1 Introduction .....	9

---

2.2 Les systèmes d'exploitation mobiles .....	10
2.2.1 Qu'est-ce qu'un système d'exploitation mobile? .....	10
2.2.2 Les différents systèmes d'exploitation mobiles .....	10
2.2.2.1 BlackBerry OS de RIM .....	10
2.2.2.2 Windows phone de Microsoft .....	11
2.2.2.3 iOS d'Apple .....	11
2.2.2.4 Android de Google .....	12
2.2.3 Étude comparative .....	12
2.2.3.1 Avantages et inconvénients de chaque OS .....	12
2.2.3.2 Les parts du marché .....	14
2.2.3.3 Quel système d'exploitation choisir ? .....	15
2.3 Les Applications mobiles .....	16
2.3.1 Qu'est-ce qu'une application mobile? .....	16
2.3.2 Types d'applications mobiles .....	16
2.3.2.1 Applications natives .....	16
2.3.2.2 Applications web mobiles .....	17
2.3.2.3 Applications hybrides .....	17
2.3.3 Quel type d'application mobile choisir ? .....	18
2.4 La plateforme Android .....	18
2.4.1 Historique .....	18
2.4.2 L'Architecture d'Android .....	20
2.5 L'application Android .....	21
2.5.1 Les types d'applications Android .....	21



---

2.5.2 Les composants d'une application Android .....	22
2.5.3 Le cycle de vie d'une activité .....	24
2.6 Conclusion .....	27
<b>3. CONDUITE DU PROJET</b> .....	<b>28</b>
3.1 Introduction .....	28
3.2 Le langage de modélisation unifié (UML) .....	29
3.2.1 Qu'est-ce que UML ? .....	29
3.2.2 Les diagrammes UML .....	29
3.2.2.1 Les diagrammes structurels (ou statiques) .....	29
3.2.2.1 Les diagrammes de comportements (ou dynamiques) .....	30
3.3 Le processus unifié (UP) .....	31
3.3.1 Qu'est-ce que UP ? .....	31
3.3.2 Les principes d'UP .....	32
3.3.3 La vie du processus unifié .....	33
3.3.4 Pourquoi UP ? .....	35
3.4 Gestion du temps .....	35
3.5 Conclusion .....	37
<b>4. RÉALISATION DU PROJET</b> .....	<b>38</b>
4.1 Introduction .....	38
4.2 La phase d'inception .....	39
4.2.1 Expression des besoins .....	39
4.2.1.1 Intitulé du projet .....	39
4.2.1.2 Objectif de l'application .....	39

---

4.2.1.3 Public visé .....	39
4.2.1.4 Identification des acteurs .....	39
4.2.1.5 Identification des besoins .....	40
4.2.2 Analyse des besoins .....	41
4.2.2.1 Diagramme de cas d'utilisation .....	41
4.2.2.2 Description textuelle des cas d'utilisation .....	41
4.2.3 Conception (initiale) .....	45
4.2.3.1 Vue globale de l'architecture du système .....	45
4.2.3.2 Diagrammes de séquence système .....	46
4.3 La phase d'élaboration .....	51
4.3.1 Conception (détaillée) .....	51
4.3.1.1 Diagrammes de séquence détaillés .....	51
4.3.1.2 Diagramme de classes .....	57
4.4 La phase de construction .....	59
4.4.1 Implémentation .....	59
4.4.1.1 Langages, outils et technologies de développement utilisés .....	59
4.4.1.2 Le pattern MVC et la structuration du code .....	63
4.4.1.3 Diagramme de déploiement .....	64
4.5 La phase de Transition .....	65
4.5.1 Tests .....	65
4.6 Conclusion .....	67
CONCLUSION GÉNÉRALE ET PERSPECTIVES .....	68
RÉFÉRENCES .....	70

---

---

# LISTE DES FIGURES

1. L'ENTREPRISE PORTUAIRE DE BEJAIA	2
1.1 Logo de l'EPB .....	3
1.2 Organigramme général de l'EPB .....	5
1.3 Organigramme du centre de système d'informations .....	7
2. LA TECHNOLOGIE DU MOBILE	9
2.1 Logo de BlackBerry .....	10
2.2 Logo de Windows Phone .....	11
2.3 Logo d'iOS .....	11
2.4 Logo d'Android .....	12
2.5 Pourcentage de vente des smartphones par OS .....	14
2.6 Architecture d'Android .....	20
2.7 Interaction entre composants d'une application .....	23
2.8 Cycle de vie d'une activité .....	25
3. CONDUITE DU PROJET	28
3.1 La vie du processus unifié .....	33
3.2 Diagramme de GANTT représentant la gestion du temps de notre projet .....	36
4. CONCEPTION ET RÉALISATION DU PROJET	38
4.1 Diagramme général de cas d'utilisation .....	41
4.2 Vue globale de l'architecture du système .....	46
4.3 Diagramme de séquence système du cas d'utilisation « S'authentifier » .....	47
4.4 Diagramme de séquence système du cas d'utilisation « Consulter la liste des navires en quai » .....	48

4. 5	Diagramme de séquence système du cas d'utilisation « Consulter la liste journalière des E/S des navires » .....	48
4. 6	Diagramme de séquence système du cas d'utilisation « Rechercher un navire » ..	49
4. 7	Diagramme de séquence système du cas d'utilisation « Consulter la liste des navires d'un client » .....	50
4. 8	Diagramme de séquence système du cas d'utilisation « Localiser un navire » ..	51
4. 9	Diagramme de séquence détaillé du cas d'utilisation « S'authentifier » .....	52
4. 10	Diagramme de séquence détaillé du cas d'utilisation « Consulter la liste des navires en quai » .....	53
4. 11	Diagramme de séquence détaillé du cas d'utilisation « Consulter la liste journalière des E/S des navires » .....	54
4. 12	Diagramme de séquence détaillé du cas d'utilisation « Rechercher un navire » ...	55
4. 13	Diagramme de séquence détaillé du cas d'utilisation « Consulter la liste des navires d'un client » .....	56
4. 14	Diagramme de séquence détaillé du cas d'utilisation « Localiser un navire » ..	57
4. 15	Diagramme de classes de notre solution informatique .....	58
4. 16	Diagramme de déploiement de notre solution informatique .....	64
4. 17	Organigramme illustré montrant le test d'authentification .....	65
4. 18	Organigramme illustré montrant le test de consultation de la liste des navires en rade .....	66

---

# LISTE DES TABLEAUX

2. LA TECNOLOGIE DU MOBILE	9
2.1 Avantages et inconvénients des OS mobiles .....	13
2.2 Récapitulatif des différentes versions d'Android .....	19
4. RÉALISATION DU PROJET	38
4.1 Description textuelle du cas d'utilisation « S'authentifier » .....	42
4.2 Description textuelle du cas d'utilisation « Consulter la liste des navires en quai » .....	42
4.3 Description textuelle du cas d'utilisation « Consulter la liste journalière des E/S des navires » .....	43
4.4 Description textuelle du cas d'utilisation « Rechercher un navire ».....	43
4.5 Description textuelle du cas d'utilisation « Consulter la liste des navires d'un client » .....	44
4.6 Description textuelle du cas d'utilisation « Localiser un navire ».....	45

---

# LISTE DES ABRÉVIATIONS

<b>API</b>	Application Programming Interface
<b>App</b>	Application
<b>CNAN</b>	Compagnie Nationale Algérienne de Navigation
<b>CRUD</b>	Create, Read, Update, Delete
<b>CSS</b>	Cascading Style Sheets
<b>EPB</b>	Entreprise Portuaire de Bejaia
<b>EPE</b>	Entreprise Publique Economique
<b>E/S</b>	Entrée/Sortie
<b>GPS</b>	Global Positioning System
<b>GUI</b>	Graphic User Interface
<b>HTC</b>	High Tech Computer
<b>HTML</b>	Hyper Text Markup Language
<b>HTTP</b>	Hyper Text Transfer Protocol
<b>IDE</b>	Integrated Development Environment
<b>IP</b>	Internet Protocol
<b>ISO</b>	International Standardization Organization
<b>JDK</b>	Java Development Kit
<b>JSON</b>	Java Script Object Notation
<b>LG</b>	Lucky Goldstar
<b>MLDR</b>	Modèle Logique de Données Relationnel
<b>MVC</b>	Model View Controller
<b>OHA</b>	Open Handset Alliance
<b>OMT</b>	Object Modeling Technique
<b>ONP</b>	Office National des Ports
<b>OOSE</b>	Object Oriented Software Engineering
<b>OS</b>	Operating System
<b>PDA</b>	Personal Digital Assistant
<b>PHP</b>	PHP Hypertext Preprocessor
<b>REST</b>	Representational State Transfer
<b>RIM</b>	Research In Motion

<b>SDK</b>	Software Development Kit
<b>SOAP</b>	Simple Object Access Protocol
<b>SOGEPORIS</b>	Société de Gestion des Participations de l'Etat pour Ports
<b>SoNaMa</b>	Société Nationale de Manutention
<b>SPA</b>	Société Par Actions
<b>SQL</b>	Structured Query Language
<b>SUN</b>	Stanford University Network
<b>TCP</b>	Transmission Control Protocol
<b>TI</b>	Texas Instruments
<b>UML</b>	Unified Modeling Language
<b>UP</b>	Unified Process
<b>URI</b>	Uniform Resource Identifier
<b>XML</b>	eXtensible Markup Language

---

# INTRODUCTION GÉNÉRALE

---

L'évolution de la technologie du mobile a remarquablement étendu les moyens d'accès à l'information : l'accès au contenu ne s'effectue plus de la même façon ni par les mêmes appareils d'il y a quelques années, l'information est devenue exploitable en tout lieu couvert par un réseau. Ces appareils, aujourd'hui qualifiés d'intelligents, sont le fruit de nombreuses années d'évolution. Les assistants personnels et les smartphones actuels fusionnent entre plusieurs possibilités d'utilisations rassemblés en un seul appareil. Avec cet exploit, et que ce soit en entreprises ou ailleurs, le nombre de leur utilisation continue sa croissance exponentielle notamment avec l'évolution des réseaux sans fil et l'apparition du système Android.

L'objectif de ce projet est de concevoir et d'implémenter une application mobile destinée à fonctionner sous Android. Elle sera accessible par des smartphones, par des tablettes ou par d'autres appareils Android. Son objectif est de fournir une plateforme disposante d'un ensemble de fonctionnalités et d'outils nécessaires qui permettent la consultation des états des navires pour le public visé, à savoir : les administrateurs du centre du système d'informations à l'entreprise portuaire de Bejaia et les clients de cette entreprise.

Le présent mémoire est structuré en quatre principaux chapitres. Après cette introduction, nous présenterons dans le premier chapitre l'entreprise portuaire de Bejaia, l'organisme d'accueil de notre stage. Puis, nous ferons un saut vers la technologie du mobile dans le deuxième chapitre où nous donnerons quelques généralités sur le sujet, nous allons présenter également la plateforme Android d'une manière détaillée. Après, nous exposerons dans le troisième chapitre notre conduite de projet et la manière dont nous avons géré le temps qu'on nous a donné pour le réaliser avant d'entamer le quatrième chapitre qui présentera l'analyse et la conception de notre application Android. Enfin, nous allons conclure ce mémoire par une conclusion générale qui synthétise tout le travail qui a été fait accompagnée de quelques perspectives.



---

# CHAPITRE 1

## L'ENTREPRISE PORTUAIRE DE BEJAIA

---

### 1.1 Introduction

Le port de Bejaia, premier port d'Algérie en marchandises générales, occupe une place très importante à l'échelle nationale avec sa position au cœur de l'espace méditerranéen. Ce qui n'est pas une nouvelle, en fait, Bejaia a toujours joué un grand rôle dans la transmission du savoir dans le bassin méditerranéen. Bougie a su attirer beaucoup de puissants marchands, depuis la Saldae romaine qui devient un port d'embarquement de blé du grenier de Rome. La réalisation des ouvrages actuels débuta en 1834, par la colonisation Française, elle fut achevée en 1987. C'est en 1960 qu'a été chargé le premier pétrolier d'Algérie [w1].

Aujourd'hui, le port de Bejaia est géré en grande partie par l'Entreprise Portuaire de Bejaia (EPB), cette entreprise nous a offert un stage de fin d'étude durant lequel il nous a été demandé de créer une application mobile sous Android pour la consultation des états des navires au niveau du port. Une application devant répondre à certains soucis que l'entreprise nous a fait comprendre et dont la contrariété majeure est la suivante :

- L'accès aux informations à l'intérieur de la firme est dépendant des postes de travail fixes.

Ce premier chapitre sera donc consacré à la présentation de l'organisme d'accueil. D'abord, nous allons présenter l'EPB, à savoir : ses missions, ses activités et ses différentes structures internes, pour mettre la lumière sur le département où se déroulera notre stage. Puis, nous exposerons la problématique qu'on nous a exprimé au niveau de l'EPB et qui nous a mené à nous lancer dans ce travail pour concevoir une solution adéquate. Enfin, nous achèverons ce chapitre par une conclusion dans laquelle nous allons expliquer la raison de notre choix de cette entreprise.

## 1.2 Présentation générale de l'organisme d'accueil

Le port de Bejaia, régi par l'EPB, joue un rôle très important dans les transactions internationales vu sa place et sa position géographique.

Aujourd'hui, il est classé premier port d'Algérie en marchandises générales et troisième port pétrolier. Il est également le premier port du bassin méditerranéen certifié ISO 9001 :2000 pour l'ensemble de ses prestations, et avoir ainsi installé un système de management de la qualité.



Figure 1.1 : Logo de l'EPB [w1].

Cela constitue une étape dans le processus d'amélioration continue de ses prestations au grand bénéfice de ses clients.

L'Entreprise portuaire a connu d'autres succès depuis, elle est notamment certifiée à la norme ISO 14001 :2004 et au référentiel OHSAS 18001 :2007, respectivement pour l'environnement et l'hygiène et sécurité au travail [w1].

## 1.3 Création de l'EPB

Le décret n°82-285 du 14 Août 1982 publié dans le journal officiel n° 33 porta création de l'Entreprise Portuaire de Bejaia ; entreprise socialiste à caractère économique ; conformément aux principes de la charte de l'organisation des entreprises, aux dispositions de l'ordonnance n° 71-74 du 16 Novembre 1971 relative à la gestion socialiste des entreprises et les textes pris pour son application à l'endroit des ports maritimes.

L'entreprise, réputée commerçante dans ses relations avec les tiers, fut régie par la législation en vigueur et soumise aux règles édictées par le susmentionné décret.

Pour accomplir ses missions, l'entreprise est substituée à l'Office National des Ports (ONP), à la Société Nationale de Manutention (SO.NA.MA) et pour partie à la Compagnie Nationale Algérienne de Navigation (CNAN). Elle fut dotée par l'Etat, du patrimoine, des activités, des structures et des moyens détenus par l'ONP, la SO.NA.MA et de l'activité Remorquage,

précédemment dévolue à la CNAN, ainsi que des personnels liés à la gestion et au fonctionnement de celles-ci.

En exécution des lois n° 88.01, 88.03 et 88.04 du 02 Janvier 1988 s'inscrivant dans le cadre des réformes économiques et portant sur l'autonomie des entreprises, et suivant les prescriptions des décrets n°88.101 du 16 Mai 1988, n°88.199 du 21 Juin 1988 et n°88.177 du 28 Septembre 1988.

L'Entreprise Portuaire de Bejaia ; entreprise socialiste ; est transformée en Entreprise Publique Economique, Société par Actions (EPE-SPA) depuis le 15 Février 1989, son capital social fut fixé à Dix millions (10.000.000) de dinars algériens par décision du conseil de la planification n°191/SP/DP du 09 Novembre 1988. Actuellement, le capital social de l'entreprise a été ramené à 1.700.000.000 DA, détenues à 100% par la Société de Gestion des Participations de l'Etat « Ports », par abréviation « SOGEPORTS » [w1].

## 1.4 Missions et Activités de l'EPB

### 1.4.1 Ses missions

Les missions de l'EPB se résument en quelques lignes :

- La gestion, l'exploitation et le développement du domaine portuaire sont les charges essentielles de la gestion de l'EPB, c'est dans le but de promouvoir les échanges extérieurs du pays.
- Elle se doit d'assumer la police et la sécurité au sein du pays.
- Elle est chargée des travaux d'entretien, d'aménagement, de renouvellement et de création d'infrastructures.
- L'EPB assure également des prestations à caractère commercial, à savoir ; le remorquage, la manutention et l'acconage. [w1]

### 1.4.2 Ses Activités

Les principales activités de l'entreprise sont :

- L'exploitation de l'outillage et des installations portuaires.
- L'exécution des travaux d'entretien, d'aménagement et de renouvellement de la super structure portuaire.
- L'exercice du monopole des opérations d'acconage et de manutention portuaire.

- L'exercice du monopole des opérations de remorquage, de pilotage et d'amarrage.
- La police et la sécurité portuaire dans la limite géographique du domaine public portuaire [w1].

## 1.5 Présentation des différentes structures de l'entreprise

### 1.5.1 Organigramme

L'EPB est organisée selon des directions fonctionnelles et opérationnelles dirigées par une direction générale qui est chargée de concevoir, coordonner et contrôler les actions liées à la gestion et au développement de l'entreprise. La figure suivante montre la structure générale de l'entreprise portuaire de Bejaia :

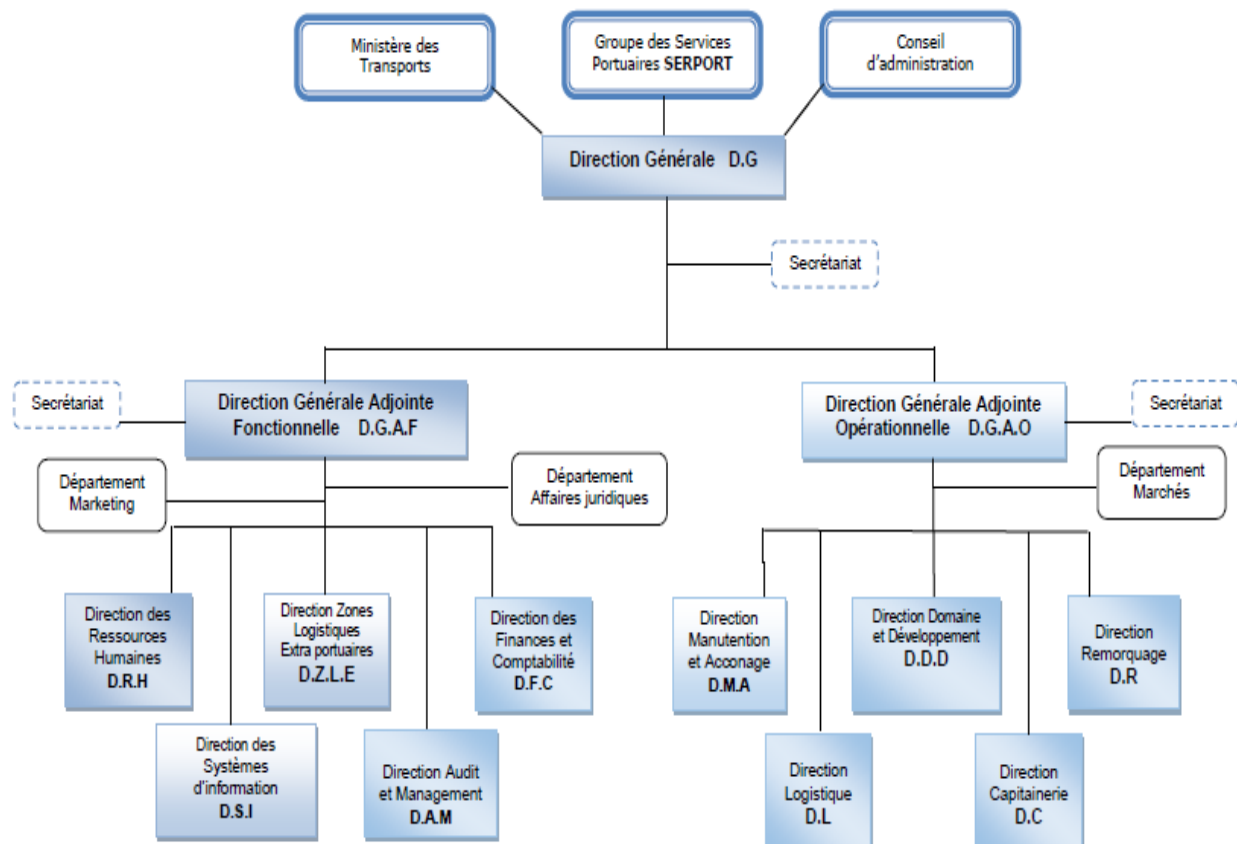


Figure 1.2 : Organigramme général de l'EPB [w1].

### **1.5.2 Présentation du centre informatique**

La structure informatique de l'EPB est un département rattaché à la direction générale adjointe ; il a été créé en 1989 et c'est à cette époque que les premières applications de l'entreprise ont vu le jour.

En 1995 la micro-informatique a été introduite à l'EPB et les premières applications sont écrites sous DBASE 5. A partir de 2001 l'entreprise portuaire a lancé un plan pour développer les applications métiers sous PHP et DELPHI 5 et comme système de gestion de bases de données MySQL [w1].

### **1.5.3 Les missions du centre informatique**

L'informatique a pour mission l'automatisation des métiers de l'Entreprise Portuaire de Bejaia, et cela en mettant en place les logiciels et l'infrastructure nécessaires pour la gestion du système d'information.

L'EPB déploie des systèmes d'informations pour accroître la productivité, automatiser les processus métiers et fournir un meilleur service aux clients. Ces systèmes intègrent de plus en plus des fonctionnalités réseau pour relier tous les utilisateurs à l'entreprise ou établir des liens avec la clientèle et les fournisseurs.

Le réseau apporte aujourd'hui une réelle valeur ajoutée en permettant d'intégrer de nouveaux partenaires, fournisseurs et clients [w1].

### **1.5.4 Organisation humaine du centre de système d'informations**

Le centre de système d'informations, se constitue de trois départements principaux : Le département « Infrastructures », le département « Programmes et méthodes » et le département « Génie logiciel » où se déroulera principalement notre stage. Chaque département regroupe un nombre de ressources humaines chargé de tâches spécifiques, La figure suivante illustre cette constitution :

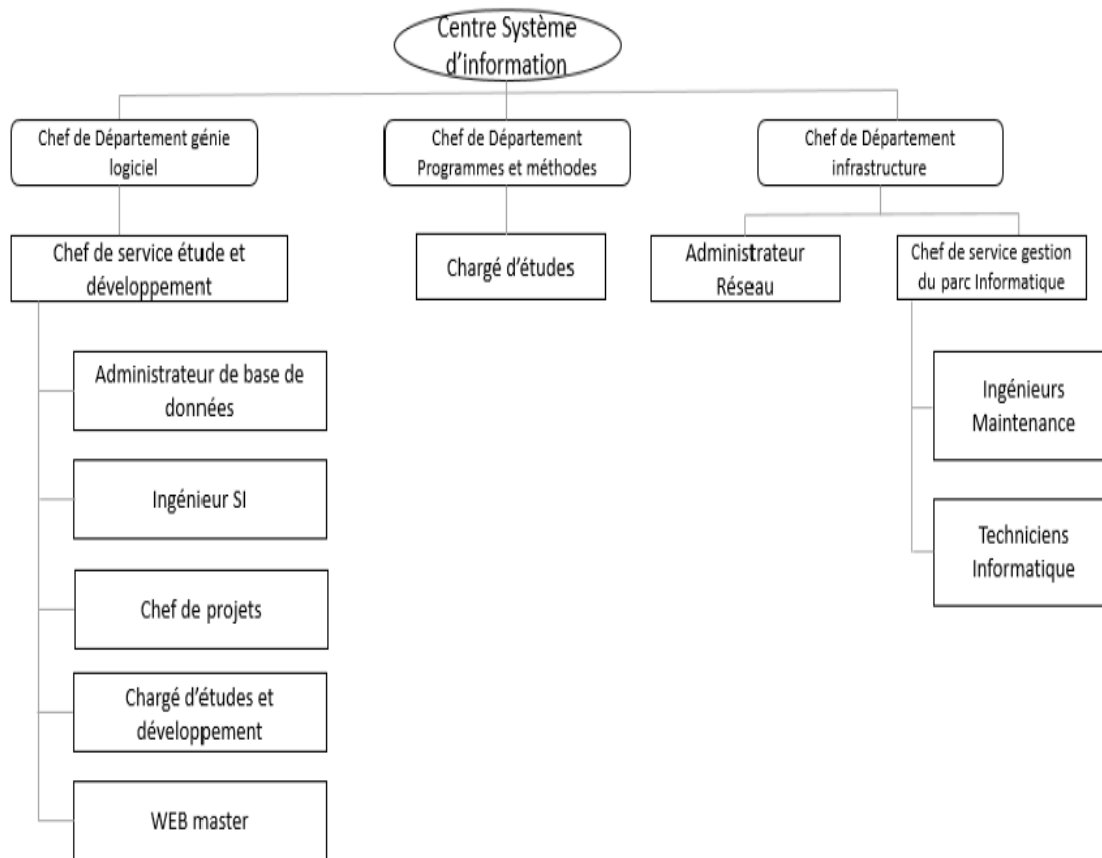


Figure 1.3 : Organigramme du centre de système d'informations [w1].

## 1.6 La Problématique

Avec les avancées de la technologie du mobile d'aujourd'hui, les entreprises ont souvent tendance à étendre leur gestion sur des terminaux mobiles. Cette pratique se révèle très efficace quant à l'accès aux ressources indépendamment du temps et de lieu.

Cependant, L'EPB qui dispose d'une application web locale pour l'accès en interne aux membres de l'entreprise et d'un site web pour l'accès en externe afin de visualiser l'état des navires, n'a pas une application mobile correspondante.

Cette réalité contraint les chargés d'administration de l'EPB à ne pouvoir utiliser que les postes fixes pour effectuer leurs missions, par conséquent, accomplir leurs tâches nécessite parfois des déplacements contrariants ce qui augmente les risques indésirables de retards de passage de l'information.

Par ailleurs, les clients de l'EPB qui ont des différents biens en rapport avec l'entreprise ne peuvent de leur côté qu'utiliser son site web pour consulter l'état des navires, comme effet, on peut envisager qu'une panne du site web en question risque de causer des problèmes indésirables à ces clients ou de les inciter à passer au contact directe avec l'entreprise pour une simple consultation.

De ces faits, pour améliorer la mobilité des différents acteurs à l'intérieur et à l'extérieur de l'entreprise et offrir ainsi plus de possibilités d'accès, il nous a été demandé de réaliser une application mobile sous Android pour remédier à ce problème, autrement dit, une application mobile qui permettra un accès à n'importe quel moment et à n'importe quel lieu dans l'entreprise indépendamment du poste où se trouve la machine d'accès au site local.

Cette solution qui devra reprendre quelques fonctionnalités disponibles sur le site web de l'EPB ne sera pas seulement destinée à un usage interne, mais aussi, les clients en relation avec l'entreprise doivent pouvoir y accéder en externe pour consulter l'état de leurs navires, l'accès ces clients sera donc moins restreint que par le seul biais d'un site web.

## 1.7 Conclusion

À travers ce chapitre, nous avons donné une vue générale sur l'EPB, l'organisme d'accueil de notre stage de fin d'étude. Nous avons également présenté la problématique et les raisons qui nous ont menés à réaliser ce projet.

Avec sa structure interne qui paraît bien organisée à notre opinion, nous avons vu par le passage que le port de Bejaia géré en grande partie par l'EPB occupe une place très importante au niveau national.

Par ailleurs, l'introduction de l'informatique à cette entreprise joue sans doute un rôle très important dans son efficacité. En ce qui nous concerne, le département de génie logiciel dont elle dispose semble être un atout pour un stage des plus appropriés : un stage dans le domaine de notre spécialité chez une entreprise qui ne manque pas de réputation.

Nous avons conclu ce chapitre, le chapitre prochain présentera une vue générale sur la technologie du mobile.

---

# CHAPITRE 2

## LA TECHNOLOGIE DU MOBILE

---

### 2.1 Introduction

De nos jours, Le marché des mobiles intelligents connaît une diversité de systèmes d'exploitation et d'appareils, une variété due à la rivalité des producteurs qui donnent de différents choix de systèmes d'exploitation aux consommateurs désirant se munir de leur propre dispositif. De ce fait, la guerre des mobiles passe principalement par l'affrontement des systèmes d'exploitation. Sur ce secteur, deux grandes sociétés mènent la danse en termes de parts de marché : Google et Apple avec environ 87,8% et 11,5% respectivement. Le premier, père d'Android, voit son système déployé sur des appareils venant de nombreuses marques. L'autre, avec iOS, est bien plus exclusif puisqu'il est réservé aux terminaux iPhone [w8].

Android et iOS ne sont pas seuls pour autant. D'autres challengers sont dans la partie et ils ne manquent pas d'intérêt. C'est le cas de Windows phone, le pendant mobile de Windows, ainsi que BlackBerry OS, le système d'exploitation des terminaux du constructeur du même nom.

Cependant, parmi toutes ces alternatives, on peut se poser les questions suivantes :

- Comment choisir le bon système d'exploitation ?
- Quels sont les avantages et inconvénients de chaque système ?

Pour répondre à ces deux questions qui ne se départagent pas, nous ferons dans ce chapitre un léger survole sur les systèmes d'exploitation mobiles les plus populaires, nous exposeront également leurs avantages et leurs inconvénients en vue d'une étude comparative de ces systèmes. Nous allons par la suite présenter de manière succincte les notions fondamentales de la plateforme Android, à savoir, un bref historique de ses versions existantes, son architecture et les bases d'une application Android. Puis, nous achèverons par une conclusion pour couronner tout ce qui a été dit.



## 2.2 Les systèmes d'exploitation mobiles

### 2.2.1 Qu'est-ce qu'un système d'exploitation mobile ?

Un système d'exploitation mobile (OS) est un logiciel qui permet aux téléphones intelligents, aux tablettes et aux autres dispositifs d'exécuter des applications et des programmes. C'est le système qui fait fonctionner l'appareil. Il commence généralement quand un appareil démarre, présentant un écran avec des icônes ou des tuiles qui présentent des informations et de fournir l'accès aux applications. Les systèmes d'exploitation mobiles gèrent également la connectivité réseau cellulaire et sans fil, ainsi que l'accès au téléphone [w2].

### 2.2.2 Les différents systèmes d'exploitation mobiles

Il existe plusieurs systèmes d'exploitation mobiles pour smartphones et tablettes, certains qui ne sont que sur des appareils d'occasion sont délaissés aujourd'hui, par exemple, le Symbian qui fut l'OS des produits Nokia se voit remplacé par Windows Phone. Voici donc les quatre OS les plus populaires :

#### 2.2.2.1 BlackBerry OS de RIM

C'est le système d'exploitation propriétaire de la société RIM (Research In Motion) destiné aux terminaux mobiles de sa gamme BlackBerry. Présent sur le marché depuis le début des années 2000, il était destiné au départ à un usage professionnel, le système d'exploitation a pris du retard dans le domaine smartphones aujourd'hui. La dernière tentative de BlackBerry porte le nom BlackBerry 10. Malheureusement elle s'est soldée par un échec et cet OS ne se trouve désormais que sur quelques smartphones d'occasion, ce qui fait que BlackBerry OS va bientôt tomber aux oubliettes. En fait, En 2016, BlackBerry s'est tourné vers Android pour ses nouveaux modèles et on peut considérer le système BlackBerry 10 comme enterré [w4].



Figure 2.1 : Logo de BlackBerry

### 2.2.2.2 Windows phone de Microsoft

Initialement le système d'exploitation pour mobile de Microsoft, s'appelait Windows Mobile. Ce système destiné aux professionnels était en déclin depuis la première version du début des années 2000 et Microsoft a donné comme successeur à la version 6.5, Windows Phone 7. Il s'agit en réalité d'un tout nouveau système d'exploitation, sorti en octobre 2010, destiné au grand public avec une interface nommée Metro. Windows Phone, désormais appelé Windows 10 Mobile au nom de la nouvelle version, utilise un système de tuiles dynamiques et son écran d'accueil se révèle très facilement personnalisable. Il profite d'un centre de notifications très complet, qui permet de consulter les dernières alertes, de gérer l'affichage ou d'activer la connexion Wi-Fi. Le magasin d'applications de Windows 10 Mobile s'est progressivement rempli depuis le lancement des premiers appareils et compte plus de 400 000 applications disponibles aux dernières nouvelles. Tout comme Android, Windows Phone est destiné à plusieurs machines et les principaux partenaires de Microsoft sont HTC, Samsung, LG et Nokia [w3].



Figure 2.2 : Logo de Windows Phone

### 2.2.2.3 iOS d'Apple

C'est un système d'exploitation mobile conçu par Apple, destiné uniquement aux appareils de la marque. Il est embarqué dans les iPhone, iPod touch et iPad toutes générations confondues. iOS, réputé par son efficacité et simplicité d'utilisation, a parcouru du chemin depuis la version 1.0 sortie en 2007 avec l'iPhone de première génération. Dernièrement, la version iOS 10 a été déployée, introduisant quelques changements dans les fonctionnalités, mais à l'ergonomie toujours très bonne. iOS dispose de l'interface la plus accessible, avec un système d'icônes et de dossiers aussi simple à prendre en main qu'efficace [w4].



Figure 2.3 : Logo d'iOS

iOS a été un moment leader sur le marché des OS mobiles mais étant cantonné aux appareils de la marque, il a été rattrapé et même dépassé à l'échelle mondiale par Android, il n'occupe aujourd'hui que 15% du marché, néanmoins, il conserve l'avantage dans le domaine professionnel grâce sa réputation en termes de sécurité [w3].

### 2.2.2.4 Android de Google

C'est un système d'exploitation mobile open source développé par Google, sortie officiellement en septembre 2008 en version 1.0. [01] L'expansion rapide d'Android s'explique par le fait que ce dernier n'est pas cantonné à certains appareils. Le système d'exploitation est aujourd'hui utilisé sur des centaines de machines de différentes marques, qu'il s'agisse de smartphones, tablettes, baladeurs et même téléviseurs. Les principaux constructeurs partenaires de Google sont : HTC, Samsung, Sony Ericsson, LG, Motorola.

De plus, les développeurs n'ont aucune contrainte de la part de Google pour leurs applications, ce qui les encourage à développer pour la plate-forme et en conséquence, le Play store, son magasin d'application est le plus fourni du marché évolue de manière exponentielle, Au même titre, les ventes explosent et Android détient désormais les plus grosses parts de marché à l'échelle mondiale avec environ 81,5% [w3].



Figure 2.4 : Logo d'Android

Les nombreux appareils qui en sont équipés ne disposent pas tous de la même version, il existe plusieurs versions d'Android, aussi bien pour les smartphones que pour les tablettes, Android a beaucoup évolué et aujourd'hui, le développement en est à la version 7 Nougat [w3].

## 2.2.3 Étude comparative

### 2.2.3.1 Avantages et inconvénients de chaque OS

En fait, Il existe plusieurs facteurs de comparaison entres les systèmes d'exploitation mobiles y compris la sécurité, la personnalisation, la fonctionnalité et le coût qui sont les plus pertinents. Le tableau suivant montre une comparaison générale basée sur ces facteurs [w4] :

OS	Avantages	Inconvénients
BlackBerry	<ul style="list-style-type: none"> <li>- Meilleur OS en termes de sécurité.</li> <li>- Parfait pour un usage professionnel.</li> <li>- Consomme moins d'énergie.</li> <li>- Consommation data relativement basse.</li> <li>- Appareils moins cher que certains des téléphones iOS et Android.</li> </ul>	<ul style="list-style-type: none"> <li>- Parc applicatif très réduit.</li> <li>- Inadapté pour un usage personnel.</li> <li>- Une navigation qui manque d'ergonomie et d'accessibilité.</li> <li>- Peu de choix de smartphones, uniquement ceux de la firme.</li> <li>- Avenir de BlackBerry incertain.</li> </ul>
Windows Phone	<ul style="list-style-type: none"> <li>- Un OS considéré très sûr.</li> <li>- Parfait pour un usage professionnel.</li> <li>- Interface simple à prendre en main.</li> <li>- Smartphones équipés d'une version Office portable et d'applications de communication telles que Skype d'où une bonne intégration des réseaux sociaux.</li> <li>- Les appareils ne coutent pas très cher avec un choix considérable de marques.</li> </ul>	<ul style="list-style-type: none"> <li>- Le store de Windows reste loin derrière iOS et Android.</li> <li>- Pas de multitâches.</li> <li>- Pas de compatibilité avec le format Flash.</li> <li>- Pas de transfert de fichiers via la technologie sans fil Bluetooth.</li> <li>- Pas de compatibilité avec Mac OS.</li> <li>- Pas très bon pour un usage personnel.</li> <li>- Incertitude concernant la viabilité de l'OS face à iOS et Android.</li> </ul>
iOS	<ul style="list-style-type: none"> <li>- Un OS simple d'usage, performant et sûr, jardin clos iOS signifie que les appareils ne peuvent exécuter que des applications qui sont pré-approuvées par Apple.</li> <li>- Bonne flexibilité entre l'usage professionnel et personnel.</li> <li>- Apps Store très riche avec plus d'un million d'applications approuvées.</li> </ul>	<ul style="list-style-type: none"> <li>- Un OS très fermé, par exemple, toute modification du système d'exploitation est quasiment impossible, impossible d'installer une application qui ne vient pas de l'Apps Store, personnalisation de l'interface impossible sans le JailBreaker qui fait perdre la garantie.</li> <li>- Peu de choix de smartphones et de tablettes, uniquement ceux de la firme qui au passage coûtent vraiment cher.</li> </ul>
Android	<ul style="list-style-type: none"> <li>- Un Os souple et s'adapte à toutes les utilisations, tout service Google inclus.</li> <li>- Bonne flexibilité entre l'usage professionnel et personnel.</li> <li>- Un vaste choix d'applications sur son Play Store qui est le plus riche.</li> <li>- Un OS Open source et très ouvert à la personnalisation, par exemple, possibilité d'installer une ROM modifiée, de personnaliser l'interface à sa guise.</li> <li>- Un large choix de marques d'appareils.</li> <li>- Autre que la haute gamme, les appareils Android sont beaucoup moins chers.</li> </ul>	<ul style="list-style-type: none"> <li>- Android a une réputation relativement faible pour la sécurité, par exemple, pas de contrôle sur les applications de Play Store, donc plus de risques de nuisance.</li> <li>- Les nombreuses versions d'Android et les différents appareils laisse que certaines applications ne sont pas adaptées à tous les dispositifs (comme les problèmes de résolution ou de performances).</li> <li>- Publicités agaçantes, les applications avec des annonces sur l'écran.</li> </ul>

Tableau 2.1 : Avantages et inconvénients des OS mobiles [w3], [w4], [w5], [w6], [w7]

### 2.2.3.2 Les parts du marché

Le marché des smartphones a connu au départ une domination de différents OS, à une époque révolue, le Symbian était l'OS de référence, iOS a connu aussi un grand succès depuis le lancement de l'iPhone en 2007, étant lors unique dans son genre, facile à utiliser et très attirant, il a dominé le marché un certain moment, du moins jusqu'à peu après la sortie d'Android en 2008, ce dernier l'a rattrapé et même dépassé aujourd'hui comme le montre la figure suivante [w8] :

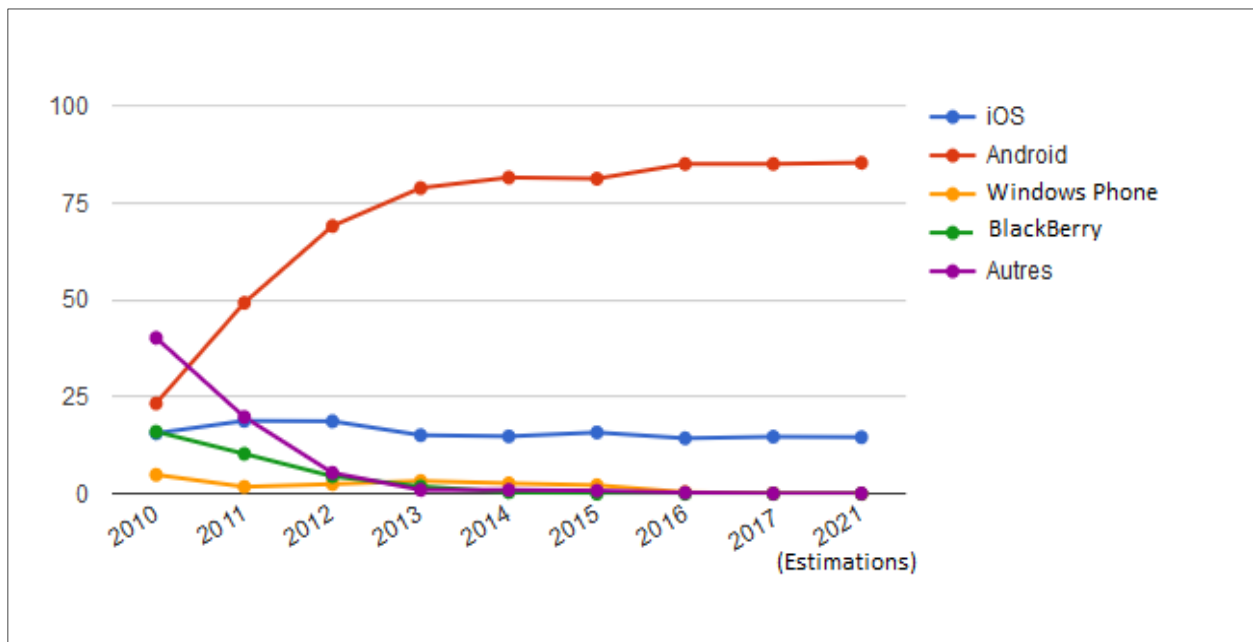


Figure 2.5 : Pourcentage de vente des smartphones par OS [w8].

Au troisième trimestre 2016, 373.2 millions de smartphones ont été vendus. Cependant, BlackBerry et les autres systèmes d'exploitation (dont Symbian) ne font presque plus partie des ventes conclues dans le marché mondial.

Windows Phone, quant à lui, fait naufrage avec une chute continue, de 2.5% en 2015 à 0.7% au début du premier trimestre 2016, il faut dire qu'en l'espace d'un an, la livraison de smartphones sous Windows a fondu de 8.2 en 2015 à 1.48 millions au troisième trimestre 2016.

L'iPhone par contre participe avec une part plus élevée, mais il n'est pas tiré d'affaire pour autant, depuis le début de l'année, l'iPhone est en recul, il descend à 11.5% avec environ 42.9 millions d'unités vendues au dernier trimestre 2016.

Alors que tous les OS déclinent, Android vole la vedette avec une part de 87.8%, à savoir presque 327.7 millions d'unités vendues au dernier trimestre 2016, ce qui laisse à estimer que sur 10 smartphones, près de 9 sont sous Android [w8].

### 2.2.3.3 Quel système d'exploitation choisir ?

Choisir son système est crucial, puisque c'est dans cette interface que l'on passe le plus clair de son temps. Afin de cerner au mieux les modèles qui conviendront le plus à ses besoins, plusieurs critères sont à prendre en compte [w6] :

- Besoins et utilisations de la personne ou des employés en mobilité pour une entreprise.
- Budget de la personne ou de l'entreprise alloué à l'équipement mobile.
- Avantages et inconvénients des différents systèmes d'exploitation existants, pour ce dernier critère, il n'est, parfois, pas très évident d'en choisir un facilement car à chaque un des avantages et des inconvénients.

Pour une utilisation personnelle, ceux qui cherchent la simplicité à tout prix peuvent se tourner vers iOS ou Windows 10 Mobile, les plus débrouillards leur préféreront Android qui offre des possibilités de personnalisation quasi infinies[w3]. Cependant, les dispositifs iOS sont coûteux, donc ne sont pas à la disposition de tout le monde, quant à ceux de Windows Phone, leurs inconvénients les laisse défavorisés de la plupart des utilisateurs, de plus, le fait est que la plupart préfèrent les possibilités de personnalisation qu'offre Android, ce qui explique en partie la domination de ce dernier sur le marché, à ce sujet, les gens ont tendance à suivre la majorité, du coup Android.

Pour une utilisation professionnelle, BlackBerry a connu une bonne expérience en entreprise, mais aujourd'hui dépassé par ses concurrents, ce qui est le même malheur pour Windows Phone. iOS reste un bon partenaire pour l'usage en entreprise grâce à sa sécurité, en particulier pour les entreprises désirant d'un maximum de contrôle sur leur mobilité. Mais reste toujours, quand on veut le meilleur rapport qualité-prix, on se tourne toujours vers Android qui est aussi très présent en entreprise. Si l'on regarde de plus près, les inconvénients d'Android sont plus ou moins négligeables face aux avantages qu'il apporte.

D'un point de vue global, il est clair qu'Android reste le favori de tous, en fait, il s'adapte à toutes les utilisations, il a donc une bonne flexibilité entre un usage personnel et professionnel.

Pour un programmeur, Android est de proche la plateforme la plus captivante, avec des langages de programmation faciles à prendre en main, principalement le langage Java. Toutefois, les autres ne sont pas négligées pour autant, mais avec le Play Store d'Android qui est très ouvert contrairement à l'Apps Store d'Apple qui ne donne pas toujours son accord pour les nouvelles applications, les développeurs fuient vers Android. De plus, une application Android est destinée naturellement à fonctionner sur un plus grand nombre d'appareils qu'une autre application sur une autre plateforme.

Dans notre cas, en raison de ces différents arguments, nous avons opté dès le départ pour le développement d'une application sous Android, ce qui est d'ailleurs le choix de l'organisme d'accueil de notre stage à l'EPB.

## **2.3 Les Applications mobiles**

### **2.3.1 Qu'est-ce qu'une application mobile ?**

Une application mobile est un programme téléchargeable de façon gratuite ou payante et exécutable à partir du système d'exploitation d'un smartphone ou d'une tablette. Au début 2016, les utilisateurs de smartphones avaient en moyenne 28 applications mobiles installées sur leurs téléphones [w9].

### **2.3.2 Types d'applications mobiles**

Techniquement parlant, il y a trois types d'application mobile que tout utilisateur peut rencontrer :

#### **2.3.2.1 Applications natives**

On parle d'applications natives quand on développe une application pour une plateforme spécifique, c'est-à-dire que si l'on veut créer une application qui sera disponible pour Android ou pour iOS, il faut développer dans le langage par défaut utilisé par les plateformes cibles. Ces plateformes retirent 25% du prix de vente pour une application native payante [w11].

Le développement de l'application native nécessite le recours à la mémoire du smartphone sans omettre les options reliées au système d'exploitation en question, on peut accéder à toutes les fonctionnalités que celle-ci offre comme les montres connectées ou la localisation

GPS. De cette façon, le résultat se résume dans l'aboutissement à des applications mobiles avec des fonctions plus professionnelles, développées et performantes.

Un des points négatifs de ce genre d'applications est que pour assurer un usage plus exponentiel de ces applications mobiles, il faut penser à lancer la même application mobile compatible à tout système d'exploitation mobile, il faut donc développer une version différente pour Android et pour iOS ou pour autre plateforme. Le temps de développement est ainsi multiplié vue que chaque plateforme possède son langage et nécessite un apprentissage important pour avancer [w10].

### **2.3.2.2 Applications web mobiles**

Les applications web mobile, appelées aussi web App, fonctionnent très bien sur les smartphones ou les tablettes, de différents OS, sont accessibles dans les navigateurs web. Ces applications conçues avec HTML et CSS sont des sites web dédiés spécifiquement pour fonctionner dans les tailles d'écran des mobiles [w11].

Certains développeurs conçoivent une seconde version de leur site web pour que celle-ci soit adaptée à des écrans plus petits. On parle de la technique de responsive design, Les éléments et l'organisation de l'écran sont retaillés en fonction de la taille de l'écran du navigateur [w15].

Peu importe la marque du mobile ou son OS, on peut accéder à l'application web par le biais de son navigateur et donc pas besoin de la télécharger. Vu qu'elle ne tient pas en compte les divergences persistantes entre les systèmes d'exploitation et les marques de smartphones, l'application web manque d'ergonomie et de plus elle ne se sert pas de la mémoire du smartphone ce qui la place en infériorité par rapport à l'application native [w11].

### **2.3.2.3 Applications hybrides**

Il s'agit d'une application mobile qui fusionne entre les caractéristiques de web App et celles de l'application native, elle contient une instance de navigateur isolé, qu'on appelle aussi WebView, en plus d'un outil d'application qui peut communiquer entre la plateforme native et la WebView. Ce qui signifie que ce type d'application peut tourner sur un mobile et avoir accès à celui-ci, donc recourir par exemple à la caméra ou au GPS [w10][w11].



Le plus connu des outils permettent la communication entre les WebView et les plateformes natives est Apache Cordova. Les frameworks les plus utilisés pour ce type d'applications, en plus de Cordova, sont :

- JQuery Mobile qui est l'adaptation de jQuery pour mobile ;
- Phone Gap qui permet de développer rapidement des applications mobiles à l'aide de HTML, CSS et JavaScript [w10].

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

### **2.3.3 Quel type d'application mobile choisir ?**

S'il faut choisir entre les trois types d'application mobile, il serait préférable de tenter pour les applications natives donnant des résultats satisfaisants, nous avons donc opté pour le développement d'une application native sous Android.

Par contre, si l'on est tenté par les applications destinée à différentes plateformes, on peut concevoir des applications web mobiles, mais celles-ci ne seront exécutées que depuis un navigateur web. En revanche, les applications hybrides, quant à elles, n'arrivent pas à satisfaire les attentes des consommateurs comme le fait celles natives [w11].

## **2.4 La plateforme Android**

Android est le système d'exploitation mobile le plus répandu. Développé par Google qui est le principal acteur, cependant, d'autres acteurs comme les membres de l'Open Handset Alliance (OHA) collaborent au développement.

### **2.4.1 Historique**

Android n'a pas démarré avec Google comme on pourrait le croire, mais il s'agissait initialement d'une entreprise américaine du nom de Startup Android Inc. créée en 2003, elle s'est fixé pour objectif de créer un OS plus intelligent pour téléphone mobile, son projet était inconnu du grand public et elle a fait face aux difficultés financières.

Android a été racheté par Google en 2005. Après la création du consortium OHA en 2007 regroupant plus de 80 acteurs industriels en télécom, composants électroniques et logiciels

(Google, Samsung, TI, LG, Bouygues, T-Mobile, ARM, Intel, Toshiba, Acer, eBay, Accentuer ...etc.), Google annonça la première version officielle du système en septembre 2008, Android 1.0 Apple Pie [01].

Voici un tableau récapitulatif des différentes versions d'Android :

Nom	versions	Première sortie	API <sup>1</sup>	Distribution
Apple Pie	1.0	23 septembre 2008	1	-
Banana Bread	1.1	9 février 2009	2	-
Cupcake	1.5	30 avril 2009	3	-
Donut	1.6	15 septembre 2009	4	-
Eclair	2.0 / 2.0.1 / 2.1	26 octobre 2009	5 / 6 / 7	-
Froyo	2.2	20 mai 2010	8	-
Gingerbread	2.3 / 2.3.3 / 2.3.4 / 2.3.5 /2.3.6 / 2.3.7	6 décembre 2010	9 / 10	0.9 %
Honeycomb	3.0 / 3.1 / 3.2 / 3.2.1 /3.2.2 /3.2.4 /3.2.6	22 février 2011	11 / 12 /13	-
Ice Cream Sandwich	4.0 / 4.0.1 / 4.0.2 / 4.0.3 /4.0.4	18 octobre 2010	14 / 15	0.9 %
Jelly Bean	4.1 / 4.1.1 / 4.1.2 / 4.2 / 4.2.1 /4.2.2/ 4.3	9 juillet 2012	16 / 17 / 18	10.1 %
KitKat	4.4 / 4.4.1 / 4.4.2 / 4.4.3 / 4.4.4	31 octobre 2013	19	20.0 %
Lollipop	5.0 / 5.0.1 / 5.0.2 / 5.1 / 5.1.1	17 octobre 1014	21 /22	32.0 %
Marshmallow	6.0 / 6.0.1	21 avril 2015	23	31.2 %
Nougat	7.0 / 7.1 / 7.1.1	22 aout 2016	24 / 25	4.9 %

Tableau 2.2 : Récapitulatif des différentes versions d'Android. [02], [w18], [w20]

<sup>1</sup> **API** est l'abréviation d'**A**pplication **P**rogramming **I**nterface, ou « interface de programmation d'application » en français, un ensemble de règles à suivre pour pouvoir dialoguer avec d'autres applications. Dans le cas de Google API, il permet en particulier de communiquer avec Google Maps [03].

## 2.4.2 L'Architecture d'Android

Android est en fait une plateforme qui inclut un système d'exploitation basé sur le noyau Linux, des middlewares (couches logicielles tierces) et des applications clés. Globalement divisée en quatre zones comme le montre le graphique suivant :

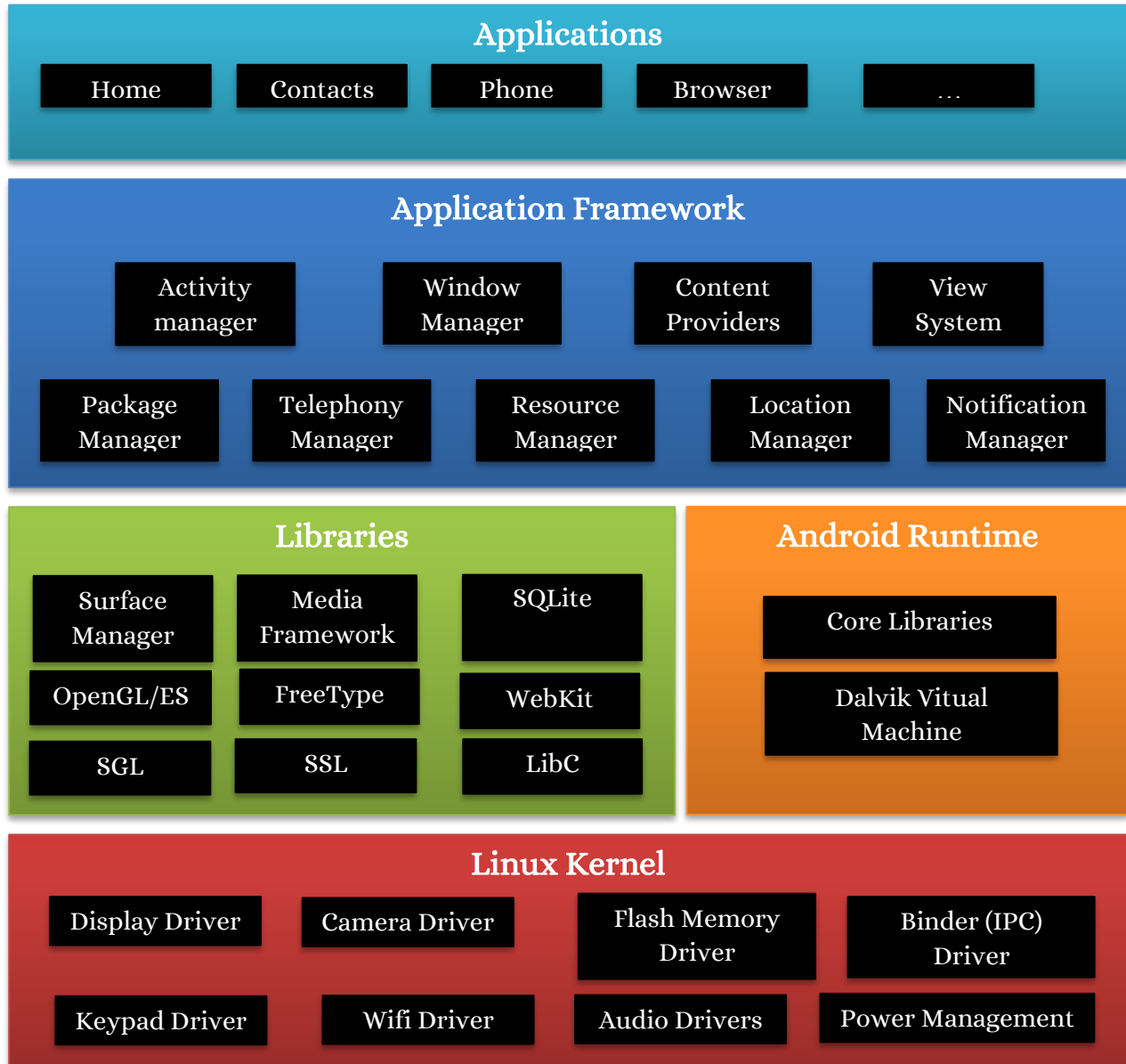


Figure 2.6 : Architecture d'Android [01]

Ces éléments peuvent être décrits comme :

- Applications : Le projet Open Source Android contient plusieurs applications par défaut comme le navigateur, l'appareil photo, la galerie, la musique, le téléphone et plus encore.
- Application Framework : Une API qui permet aux applications Android d'interagir avec le système Android.
- Libraries and runtime : Les bibliothèques pour de nombreuses fonctions communes (par exemple : le rendu graphique, le stockage de données, la navigation sur le Web, etc.) de l'Application Framework et du moteur Dalvik, ainsi que le noyau de bibliothèques Java pour exécuter des applications Android.
- Linux kernel : La couche de communication avec le matériel sous-jacent, un noyau linux modifié et optimisé pour des systèmes avec ressources limitées (Faible processeur, mémoire, batterie, ...).

Le noyau Linux, les bibliothèques et le moteur d'exécution sont encapsulés par l'Application Framework. Le développeur d'applications Android travail généralement avec les deux couches supérieures pour créer de nouvelles applications Android [w19].

## 2.5 L'application Android

Les applications Android sont écrites principalement avec le langage JAVA. Le SDK (Sun Development Kit) regroupe un ensemble d'outils permettant de faire ainsi.

Toutefois, il est possible d'écrire des applications Android en utilisant le langage C/C++ qui seront exécutées directement par le système d'exploitation Linux embarqué. Android fournit le kit de développement NDK (Native Development Kit) pour le développement en C/C++.

### 2.5.1 Les types d'applications Android

Il existe différents types d'application Android qui rentre tous dans la catégorie d'applications natives :

- Application de premier plan : c'est une application qui est utilisable uniquement lorsqu'elle est visible et mise en suspens lorsqu'elle ne l'est pas.

- Application d'arrière-plan (Services) : N'interagit pas avec l'utilisateur, elle s'exécute en tâche de fond.
- Intermittente : c'est une application qui présente une certaine interactivité mais effectue l'essentiel de sa tâche en arrière-plan. Ces applications notifient l'utilisateur lorsque cela est nécessaire.
- Widget : ces applications représentées sous forme d'un widget de l'écran d'accueil.

En ce qui nous concerne, nous allons créer une application Android de premier plan.

### 2.5.2 Les composants d'une application Android

Les applications Android sont constituées de composants à couplage. Avant de parler de ces composants, il faut savoir qu'ils sont liés par un Manifest qui décrit chacun d'eux ainsi que les interactions entre eux. En fait, chaque projet Android contient un fichier au format XML nommé Manifest (AndroidManifest.xml).

Le manifeste permet de définir la structure et les métadonnées d'une application, ses composants et ses prérequis. Il contient des balises pour chacun des composants qui constituent l'application, il permet aussi de spécifier les différentes options pour les projets, comme le matériel nécessaire pour les faire fonctionner, certains paramètres de sécurité ou encore des informations plus ou moins triviales telles que le nom de l'application ainsi que son icône. Il est donc de priorité de savoir comment manipuler ce fichier [03].

Les composants d'une application Android sont :

- Activités (Activity) : Couche de présentation de l'application, ce sont les programmes qui sont une activité pour l'utilisateur. Ils possèdent donc un GUI<sup>2</sup> ;
- Les fournisseurs de contenus (Content providers) : Partage de contenus entre applications, ils offrent un niveau d'abstraction pour l'accès à toutes les données stockées sur le terminal. Il est encouragé d'ouvrir ces données aux autres applications. Les données sont identifiées au moyen d'URI (Unified Ressources Identifier) ;

---

<sup>2</sup> **GUI** : Graphic User Interface, une interface graphique d'interaction Homme-Machine (IHM)

- Les intentions (Intents) : Framework de communication inter applications, ce sont des messages systèmes qui servent de support événementiel pour permettre le dialogue entre applications. On répond et on envoie des intentions qui lancent ou communiquent avec les activités ;
- Services : Les composants qui tournent en arrière-plan, ils ont une durée de vie infinie (contrairement aux activités et aux fournisseurs de données). Il faut les voir comme des démons qui tournent en tâche de fond ;
- Les receveurs de diffusions (Broadcast receivers) : Consommateurs des messages diffusés par les intents ;
- Widgets : Composant d'application visuelle ;
- Notifications : Framework de notifications aux utilisateurs.

La figure suivante montre l'interaction entre certains composants d'une application :

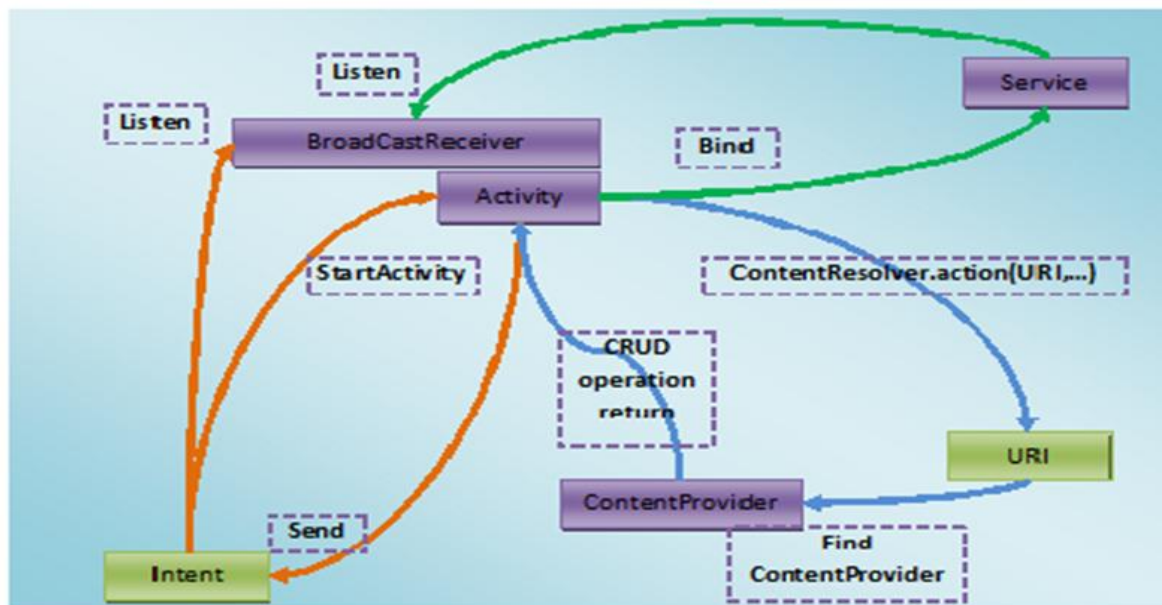


Figure 2.7 : Interaction entre composants d'une application [w21]

Une activité envoie des intentions au système qui identifie l'activité associée et soit la lance (startActivity), soit lui parle au moyen du BroadcastReceiver. Un BroadcastReceiver est une interface qui est implémentée par l'activité écoutante.

Une activité se lie à un service par la méthode Bind et soit interagit directement avec les méthodes du service, soit écoute les événements envoyés par celui-ci.

Une activité peut demander, via un ContentResolver, au système d'effectuer une action CRUD sur des données identifiées par une URI. Le système trouve le Content Provider qui gère ses données et lui demande d'effectuer l'action demandée. Le résultat de l'action est renvoyé à l'activité. Un service peut aussi utiliser ce mécanisme pour effectuer des actions CRUD sur des données [w21].

### 2.5.3 Le cycle de vie d'une activité

Comme l'activité est le composant de l'application qui interagit avec l'utilisateur, il est important de savoir comment la gérer, on parle du cycle de vie d'une activité.

Le système, pour des raisons de priorisation d'activités (coup de téléphone par exemple est prioritaire), peut tuer une activité quand il a besoin de ressources. Pour cette raison, aucune activité ne peut penser pouvoir vivre jusqu'au bout de son traitement, elle doit être considérée comme un humain. Elle peut avoir un accident, être à l'hôpital et/ou mourir [w22].

Une activité possède quatre états que sont :

- « Active » : l'activité est lancée par l'utilisateur, elle s'exécute au premier plan ;
- « En Pause » : l'activité est lancée par l'utilisateur, elle s'exécute et est visible, mais elle n'est plus au premier plan. Une notification ou une autre activité lui a volé le focus et une partie du premier plan ;
- « Stoppée » : l'activité a été lancée par l'utilisateur, mais n'est plus au premier plan et est invisible. L'activité ne peut interagir avec l'utilisateur qu'avec une notification ;
- « Morte » : l'activité n'est pas lancée.

Le schéma suivant indique le cycle de vie d'une activité et les méthodes appelées lors des changements d'état :

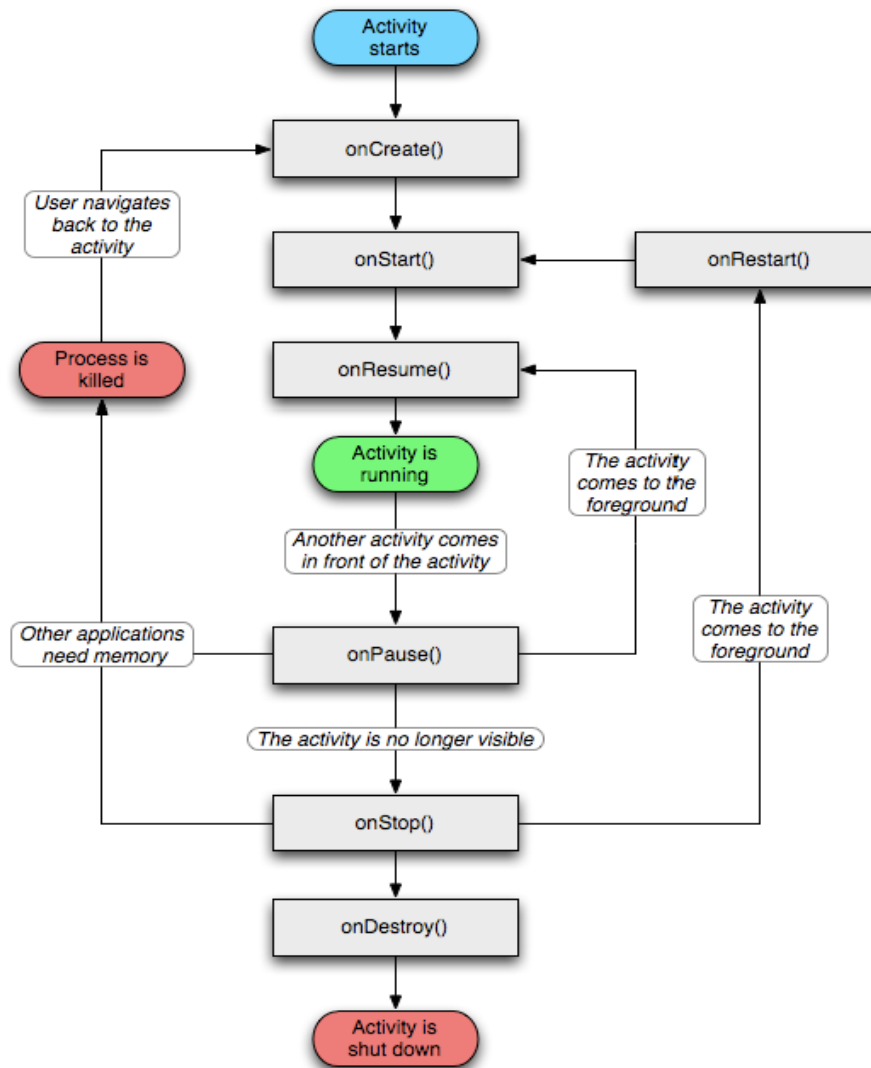


Figure 2.8 : Cycle de vie d'une activité [w22]

Ainsi, les différentes méthodes sont présentées dans la page suivante.

- La méthode `onCreate` est appelée :
  - Au premier lancement de l'activité ;
  - Si l'activité est ressuscitée, le bundle passé en paramètre sera celui sauvegardé par `onSaveInstanceState()` ;
  - Si l'état du terminal change et que l'activité est associée à cet état (passage du mode portrait au mode paysage).



Elle configure les IHM et tous les traitements d'initialisation qui ne sont effectués qu'une seule fois au lancement de l'activité.

- La méthode `onDestroy` est appelée lors de la mort de l'activité (soit naturelle, soit par le système). Parfois, l'urgence du système détruira l'activité sans même appeler `onDestroy`. Cette méthode doit libérer les ressources allouées dans la méthode `onCreate`. La seule méthode qui est appelée avec certitude avant la destruction de l'activité est `onPause`.
- Les méthodes `onStart`, `onRestart` et `onStop` n'ont pas grand intérêt.
- Les méthodes `onPause` et `onResume` sont celles dans lesquelles l'activité doit sauvegarder ses états et les restituer.
- La méthode `onResume` est appelée immédiatement avant que l'activité ne passe au premier plan. De ce fait, c'est le bon endroit pour reconstruire les données affichées par l'IHM et mettre celle-ci à jour, quitte à lancer un thread de reconstruction des données.

Inversement, la méthode `onPause` est la seule par laquelle il est certain que l'application passera avant de se faire détruire. Il convient donc de sauvegarder l'état de l'application dans cette méthode.

Il existe deux autres méthodes un peu particulières qui n'apparaissent pas dans le cycle de vie :

- `onSaveInstanceState()` ou `onRetainNonConfigurationInstance()` qui sont déclenchées lorsqu'il faut mémoriser l'état d'une activité suite à une relance de celle-ci. Les causes de cette relance pouvant être un manque de mémoire alors que l'activité est en pause, un changement de la configuration comme par exemple l'écran (rotation, ouverture d'un clavier par exemple) forçant à redessiner les widgets de l'activité dans la nouvelle résolution,

Toute la subtilité du mécanisme réside dans le choix fait par le développeur des données à sauvegarder. Il ne doit pas vouloir y mettre la terre entière et être précis dans ce qu'il choisit de stocker et de restaurer. [w22]

## 2.6 Conclusion

Actuellement, Les mobiles intelligents sont capables de servir de manière très étendue, entre la communication, le divertissement, l'organisation et la gestion, ils deviennent des outils indispensables pour une vie plus simple, à la fois personnelle et professionnelle.

Cependant, il faut dire que ses mobiles intelligents, ainsi que les technologies qui les accompagnent ont parcouru un chemin considérablement long pour en arriver jusqu'à leur état actuel. Aujourd'hui, les tablettes intelligentes et les smartphones, aussi séduisants que profitables, savourent le fruit de longues années d'évolution et se présentent aux consommateurs sous différentes allures, un large choix de modèles et de systèmes d'exploitation très convaincants.

Sur ce point, la connaissance de ces systèmes d'exploitation est nécessaire pour les développeurs, du moins, pour un choix de la plateforme adéquate. Néanmoins, Android qui s'adapte à toutes les utilisations reste la vedette entre tous. Un OS étant ouvert à la personnalisation et à la contribution en matière de programmation des applications, attire la plupart des consommateurs et des développeurs.

Nous avons vu également qu'Android utilise un système d'exploitation Linux embarqué tout au bas niveau qui est la couche de communication avec le matériel, enveloppée d'autres couches et de bibliothèques jusqu'aux applications tout en haut. Cette architecture permet la programmation d'application Android non seulement en langage par défaut qui est le Java, mais aussi en langage C ou C++.

En parlant d'application, nous avons présenté les différents composants d'une application Android et la manière dont les applications sont gérées en connaissant le cycle de vie d'une activité.

Nous avons clos ce chapitre, Dans le prochain, nous allons exposer la conduite de notre projet.

---

# CHAPITRE 3

## CONDUITE DU PROJET

---

### 3.1 Introduction

Comme la réalisation de n'importe quel type de projet, un projet informatique nécessite une certaine planification, généralement une phase d'analyse suivi d'une étape de conception puis de réalisation. On parle de trois étapes indispensables faisant partie de tout type de processus de développement.

Par ailleurs, la construction d'une maison nécessite des plans à différents niveaux (vision extérieure, plan des différents étages, plans techniques...), tout comme la réalisation d'une application informatique, elle est basée sur plusieurs diagrammes pour les différents aspects.

Dans le but de mener à terme notre projet, nous avons choisi de nous reposer sur le processus unifié UP qui est accompagné d'un bon nombre d'outils divers de modélisation que propose UML.

En fait, UP est un processus générique assez compliqué vue qu'il est destiné aux projets de taille et de risques importants. Toutefois, il est possible d'adapter UP pour des projets relativement simples en respectant les étapes les plus pertinentes à son projet. Grâce au principe de développement par itérations que respecte UP, les risques et les changements qui apparaissent pendant le développement seront traités au fur et à mesure de ces itérations.

## 3.2 Le langage de modélisation unifié (UML)

### 3.2.1 Qu'est-ce qu'UML ?

Le sigle UML se lit en Anglais Unified modeling Language, ce qui se traduit en Français par le langage de modélisation unifié. Développé 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.

Selon l'OMG, UML est un langage visuel dédié à la spécification, la construction et la documentation des artefacts d'un système logiciel. Il hérite de plusieurs méthodes telles qu'OMT (Object Modeling Technique) et OOSE (Object Oriented Software Engineering) et BOOCH. Les principaux auteurs de la notation UML sont Grady BOOCH, Ivar JACOBSON et Jim RUMBAUGH [04].

Cependant, UML n'est pas une méthode, mais seulement un langage graphique qui permet de modéliser et de communiquer les divers aspects d'un système d'information. N'ayant pas un cycle de vie propre, UML doit être associé à un processus de développement.

### 3.2.2 Les diagrammes UML

UML propose des diagrammes pour la représentation des différents aspects du système, un diagramme est une représentation graphique, qui s'intéresse à un aspect précis du modèle et utilise la sémantique. La combinaison de plusieurs modèles permet une représentation complète du système dans tous ses aspects.

UML dans sa version 2 propose treize diagrammes qui peuvent être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux grands ensembles [05].

#### 3.2.2.1 Les diagrammes structurels (ou statiques)

Ces diagrammes, au nombre de six, ont vocation à représenter l'aspect statique d'un système :

- **Diagramme de classes** : Ce diagrammes représente la description statique du système en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements. C'est le diagramme pivot de l'ensemble de la modélisation d'un système ;
- **Diagramme de composants** : Ce digramme représente les différents constituants du logiciel au niveau de l'implémentation d'un système ;
- **Diagramme d'objets** : Ce diagramme permet la représentation d'instances des classes et des liens entre instance ;
- **Diagramme de structure composite** : Ce diagramme permet de décrire la structure interne d'un ensemble complexe composé par exemple de classes ;
- **Diagramme de déploiement** : Ce diagramme permet de représenter l'architecture physique supportant l'exploitation du système. Cette architecture comprend des nœuds correspondants aux supports physiques (serveurs, routeurs...) Ainsi que la répartition des artefacts logiciels (bibliothèques, exécutables...) Sur ces nœuds. C'est un véritable réseau constitué de nœuds et de connexion entre ces nœuds qui modélise cette architecture ;
- **Diagramme de paquetages** : Ce diagramme donne une vue d'ensemble du système structuré en paquetage. Chaque paquetage représente un ensemble homogène d'éléments du système (classe, composants...).

### 3.2.2.1 Les diagrammes de comportements (ou dynamiques)

Ces diagrammes représentent la partie dynamique d'un système réagissant aux évènements et permettant de produire les résultats attendus par les utilisateurs. Sept diagrammes sont proposés par UML :

- **Diagramme de cas d'utilisation** : Ce diagramme est destiné à représenter les besoins des utilisateurs par rapport au système. Il constitue un des diagrammes les plus structurants dans l'analyse d'un système ;
- **Diagramme d'état-transition** : Ce diagramme montre les différents états des objets en réaction aux évènements.
- **Diagramme d'activités** : Ce diagramme donne une vision des enchaînements des activités propres à une opération ou à un cas d'utilisation. Il permet aussi de représenter les flots de contrôle et les flots de données ;

- **Diagramme de séquence** : Ce diagramme permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets ;
- **Diagramme de communication (collaboration)** : Ce diagramme est une autre représentation des scénarios de cas d'utilisation qui met plus l'accent sur les objets et les messages échangés ;
- **Diagramme global d'interaction** : Ce diagramme fournit une vue générale des interactions décrites dans le diagramme de séquence et des flots de contrôle décrits dans le diagramme d'activités ;
- **Diagramme de temps** : Ce diagramme permet de représenter les états et les interactions d'objets dans un contexte où le temps a une forte influence sur le comportement du système à gérer.

### 3.3 Le processus unifié (UP)

#### 3.3.1 Qu'est-ce qu'UP ?

Pour commencer, un processus définit une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. L'objet d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles.

Le processus unifié UP (Unified Process) est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques.

C'est un processus générique pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles d'entreprises. UP est donc un ensemble de principes génériques adapté en fonction des spécificités des projets [w23].

L'objectif d'UP est de maîtriser la complexité des projets informatiques en diminuant les risques. Il répond aux préoccupations suivantes :

- QUI participe au projet ?

- QUOI, qu'est-ce qui est produit durant le projet ?
- COMMENT doit-il être réalisé ?
- QUAND est réalisé chaque livrable ?

### 3.3.2 Les principes d'UP

UP est basé sur les trois principes suivants :

- **Il est itératif et incrémental** : L'itération est une répétition d'une séquence d'instructions, un traitement ou une procédure qui se répète un nombre de fois fixé à l'avance ou tant qu'une condition définie n'est pas remplie.  
Une itération prend en compte un certain nombre de cas d'utilisation et traite en priorité les risques majeurs. Elle désigne la succession des étapes de l'enchaînement d'activités, tandis qu'un incrément correspond à une avancée dans les différentes phases de développement.
- **Il est centré sur l'architecture** : Dès le démarrage du processus, on aura une vue sur l'architecture à mettre en place, on parle de tout ce qui doit être réalisé dans le projet. L'architecture d'un système logiciel peut être décrite comme les différentes vues du système qui doit être construit. L'architecture logicielle équivaut aux aspects statiques et dynamiques les plus significatifs du système. Elle émerge des besoins de l'entreprise, tels qu'ils sont exprimés par les utilisateurs et autres intervenants et tels qu'ils sont reflétés par les cas d'utilisation.

Elle subit également l'influence d'autres facteurs :

- La plateforme sur laquelle devra s'exécuter le système ;
- Les briques de bases réutilisables disponibles pour le développement ;
- Les considérations de déploiement, les systèmes existants et les besoins non fonctionnels (performance, fiabilité, sécurité...).

Les cas d'utilisation doivent une fois réalisés, trouver leur place dans l'architecture. Celle-ci doit prévoir la réalisation de tous les cas d'utilisation. Elle se dévoile peu à peu, au rythme de la spécification et de la maturation des cas d'utilisation jusqu'à ce qu'elle soit jugée stable.

- **Il est piloté par les cas d'utilisation** : Le but principal d'un système informatique est de satisfaire les besoins du client. Les cas d'utilisation permettent d'illustrer ces besoins, ils dictent donc les fonctionnalités complètes du système. [w23]

### 3.3.3 La vie du processus unifié

UP gère le processus de développement par deux axes.

- **L'axe vertical** représente les principaux enchaînements d'activités, qui regroupent les activités selon leur nature. Cette dimension rend compte l'aspect statique du processus qui s'exprime en termes de composants, de processus, d'activités, d'enchaînements, d'artefacts et de travailleurs.
- **L'axe horizontal** représente le temps et montre le déroulement du cycle de vie du processus ; cette dimension rend compte de l'aspect dynamique du processus qui s'exprime en terme de cycles, de phases, d'itérations et de jalons.

UP répète un certain nombre de fois une série de cycle qui s'articule autour de quatre phases : inception, élaboration, construction et transition.

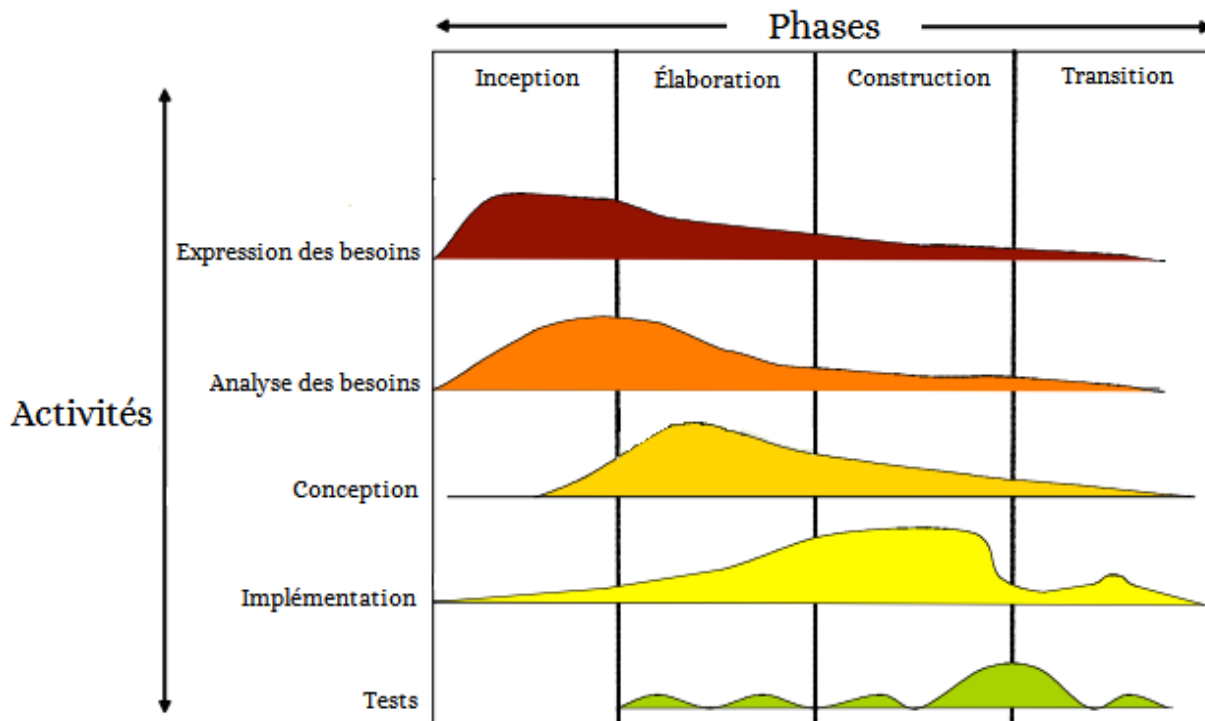


Figure 3.1 : La vie du processus unifié [w23]



Dans chaque phase, une ou plusieurs itérations peuvent avoir lieu, chaque itération passe par les différentes activités de la phase. Toutefois, une itération nécessite un plan d'itération qui regroupe cet ensemble d'activités.

À la fin de chaque itération, des livrables sont réalisés (appelés aussi artefacts), ces livrables peuvent être des documents (diagrammes, plans d'itération ...) ou des prototypes exécutable (maquettes IHM, version beta...).

La réalisation d'un livrable marque généralement la fin d'une phase, à la fin de chaque phase on prévoit des jalons, ces jalons correspondent à des étapes d'évaluation de la phase terminée, on détermine ainsi si l'on doit lancer la phase suivante ou refaire le travail de la phase terminée.

L'avancée d'un ou de nombreux cas d'utilisation regroupés dans les différentes phases de développement est appelée : incrément.

Pour mener efficacement un tel cycle, les développeurs ont besoins de toutes les représentations du produit logiciel [w23] :

- Un modèle de cas d'utilisation ;
- Un modèle d'analyse : détailler les cas d'utilisation et procéder à une première répartition du comportement ;
- Un modèle de conception : finissant la structure statique du système sous forme de sous-systèmes, de classes et interfaces ;
- Un modèle d'implémentation : intégrant les composants ;
- Un modèle de déploiement : définissant les nœuds physiques des ordinateurs ;
- Un modèle de test : décrivant les cas de test vérifiant les cas d'utilisation ;
- Une représentation de l'architecture.

Ces modèles sont assurés par l'utilisation du langage UML.

### 3.3.4 Pourquoi UP ?

UML facilite la compréhension et l'adoption d'une méthodologie orientée objets du développement logiciel qui s'appuie sur la modélisation des objets du monde réel. En fait, UML rend facile la tâche de traduire les besoins des clients en un modèle facilement interprétable puis assure l'utilisation de ce modèle pour bâtir une conception indépendante des langages de programmation, organisée autour de ces objets. Raison pour laquelle UML est un atout pour le développement des logiciels informatiques de qualité.

Néanmoins, UML n'est pas une méthode comme mentionné précédemment : S'il permet de modéliser un système, il ne définit pas le processus d'élaboration des modèles. D'où, les questions suivantes s'imposent :

- Dans quel ordre doit-on utiliser les diagrammes ?
- À quel moment de la conception d'un système doivent-ils intervenir ?

UP répond bien à ces questions. C'est un processus de développement moderne, itératif, efficace sur des projets informatiques de toutes tailles. Très complet, il couvre l'ensemble des étapes, depuis le lancement du projet jusqu'à la livraison de la solution, ainsi, il impose l'utilisation d'un ou plusieurs digramme(s) UML nécessaire(s) pour chaque 'une de ses étapes. Il est donc un processus parfaitement adapté pour l'utilisation d'UML, idéal pour une bonne conduite de projets de développement de logiciels informatiques.

## 3.4 Gestion du temps

Malgré le peu de temps que nous avons pour réaliser notre projet, nous avons essayé de gérer ce temps d'environ deux mois en respectant le processus unifié.

La réalisation d'un tel processus au complet est quasiment impossible pendant une aussi courte durée. Néanmoins, nous avons programmé une planification initiale pour gérer la conduite de ce projet en respectant les activités (tâches) les plus pertinentes dans UP pour chaque phase.

Nous avons donc planifié une période d'environ deux semaines pour la phase d'inception, deux autres semaines pour la phase d'élaboration et presque deux semaines et un demi pour les phases de construction et de transition, voir environ sept semaines pour tout le projet.

Ces planifications initiales sont évidemment plus ou moins précise, en fait, elles peuvent varier selon les risques qui sont imprévisibles et qui peuvent apparaitre lors de la réalisation du projet : par exemple, une itération pour corriger un cas d'utilisation après une mal interprétation d'un besoin peut causer un décalage dans cette planification.

Nous allons illustrer ce planning avec le diagramme de Gantt, couramment utilisé en gestion de projet, c'est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet. La colonne de gauche du diagramme énumère toutes les tâches à effectuer, tandis que la ligne d'en-tête représente les unités de temps les plus adaptées au projet (jours, semaines, mois ...). Chaque tâche est matérialisée par une barre horizontale, dont la position et la longueur représentent la date de début, la durée et la date de fin [w24].

La figure suivante montre notre planning initial pour la gestion du temps de notre projet :

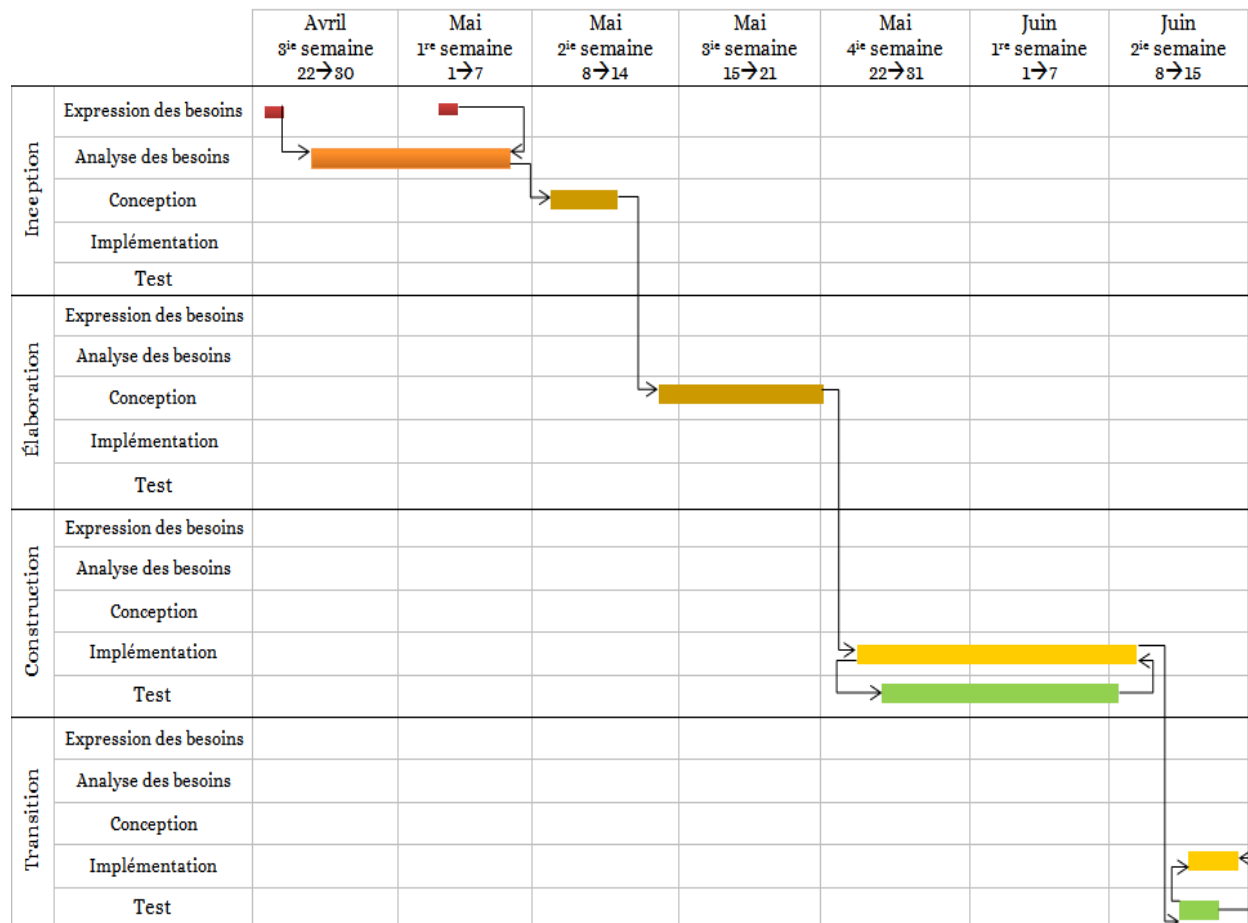


Figure 3.2 : Diagramme de GANTT représentant la gestion du temps de notre projet

### **3.5 Conclusion**

Nous avons présenté dans ce chapitre notre conduite de ce projet : le processus unifié UP que nous avons utilisé avec le langage de modélisation unifié UML.

Nous avons également présenté notre planification du temps pour le réaliser, grâce à un diagramme de GANTT, nous avons illustré le délai de réalisation de chaque tâche que nous avons initialement planifié.

Nous avons clos ce chapitre. Dans le prochain chapitre nous allons présenter la conception et la réalisation de notre application Android.

---

# CHAPITRE 4

## CONCEPTION ET RÉALISATION DU PROJET

---

### 4.1 Introduction

Se laisser guider par le processus unifié lors du développement d'un logiciel informatique promet une production de qualité vu qu'il touche à tous les aspects du système à concevoir. Ce processus est guidé par les cas d'utilisation, du coup, une bonne interprétation des besoins du client est une nécessité incontestable pour se lancer dans la réalisation de son projet. Toutefois, l'assurance d'une telle tâche commence avant tout par une bonne expression de ces besoins et une bonne représentation graphique non seulement de ces besoins, mais aussi de tous les aspects du système.

Ce chapitre présentera ce que nous avons réalisé tout au long de notre projet suivant le processus UP, il exposera donc toute la documentation dégagée lors des quatre phases, principalement à travers quelques diagrammes UML.

En premier lieu, La phase d'inception qui mettra en évidence les objectifs de notre application Android, ses acteurs ainsi que leurs besoins. Puis, La phase d'élaboration qui consiste en la description de la solution informatique proposée en se servant des résultats de la phase d'analyse. Ensuite, La phase de construction, qui quant à elle, présente la transformation de la solution informatique de la phase d'élaboration en un système réparti en une application Android côté client et un script PHP coté serveur. Enfin, la phase de transition qui consiste à effectuer des tests de fonctionnement de notre application Android.

## 4.2 La phase d'inception

### 4.2.1 Expression des besoins

Dans cette partie, nous présentons les informations et les besoins que nous avons recueilli à l'EPB.

#### 4.2.1.1 Intitulé du projet

Notre projet se résume en la conception et l'implémentation d'une application sous Android pour la consultation des états des navires à l'EPB. L'application sera nommée « EPB ».

#### 4.2.1.2 Objectif de l'application

Elle a pour objectif de fournir une plateforme disposante d'un ensemble de fonctionnalités et d'outils nécessaires qui permettent la consultation des états des navires pour le public visé.

#### 4.2.1.3 Public visé

L'application Android que nous allons réaliser s'adresse essentiellement aux administrateurs du centre du système d'informations à l'EPB et les clients en relation avec l'entreprise : les agents de transportation propriétaires des navires et les clients réceptionnaires de ces navires.

#### 4.2.1.4 Identification des acteurs

Les acteurs principaux qui interagissent avec l'application sont les utilisateurs suivants :

- **Administrateurs** : Employés chargé de la gestion du système informatique au centre du système d'informations à l'EPB ;
- **Clients** : on distingue deux types de clients :
  - Agents : propriétaires des navires chargés de transportation ;
  - Réceptionnaires : Clients finaux aux quels la cargaison du navire est destinée.

#### 4.2.1.5 Identification des besoins

Les besoins aux quels notre application doit répondre sont :

- Les utilisateurs doivent pouvoir visualiser l'état et les informations des navires consignés à l'EPB (un navire peut être en quai, en rade ou bien attendu) ;
- Les utilisateurs doivent pouvoir localiser un navire donné (un administrateur peut localiser tous les navires tandis qu'un client ne peut localiser que les navires le concernant) ;
- Les utilisateurs doivent pouvoir consulter la liste journalière d'entrées et sorties des navires consignés à l'EPB ;
- Les clients doivent pouvoir effectuer une recherche d'un de leurs navires pour visualiser son état (un agent peut effectuer une recherche d'un navire dont il est propriétaire et un réceptionnaire peut effectuer une recherche d'un navire dont il attend l'arrivée) ;
- Les administrateurs doivent pouvoir effectuer une recherche d'un navire d'un client donné pour visualiser son état.

Aux besoins cités précédemment, on peut associer les exigences non fonctionnelles suivantes :

- Garantir un accès sécurisé aux utilisateurs afin de les différencier et de protéger les données, et cela par un moyen d'authentification ;
- Contrairement à l'administrateur qui peut consulter et localiser n'importe quel navire, un client ne peut consulter que la liste des navires le concernant, par conséquent, ne peut localiser que ces navires ;
- L'application doit fonctionner sous réseau ;
- L'application doit avoir une bonne ergonomie : cohérences des fonctionnalités, interface simple et facile à manipuler par les utilisateurs.





– **Le cas d'utilisation « S'authentifier » :**

Cas d'utilisation	S'authentifier.
Acteur	Utilisateur : administrateur ou client : agent ou réceptionnaire.
Objectif	Accès sécurisé de l'utilisateur aux fonctions qui lui sont réservées.
Pré-condition	/
Scénario nominal	<ul style="list-style-type: none"> <li>– L'utilisateur accède à l'application ;</li> <li>– Le système affiche l'interface d'authentification (formulaire) ;</li> <li>– L'utilisateur saisi son identifiant, son mot de passe et valide ;</li> <li>– Le système vérifie l'existence du compte ;</li> <li>– Le système donne l'accès à l'interface d'accueil.</li> </ul>
Alternatives	Si les informations saisies sont invalides (champs non remplis), inexistantes, ou une erreur s'est produite, le système affiche un message d'erreur approprié et réaffiche l'interface d'authentification.
Options	/

Tableau 4.1 : Description textuelle du cas d'utilisation « S'authentifier »

Pour les cas d'utilisation : « Consulter la liste des navires attendus », « Consulter la liste des navires en rade » et « Consulter la liste des navires en quai », le traitement que le système effectuera avec chacun d'entre eux est pratiquement le même, il suffit donc d'illustrer un seul des trois cas, nous avons choisi le troisième :

– **Le cas d'utilisation « Consulter la liste des navires en quai » :**

Cas d'utilisation	Consulter la liste des navires en quai.
Acteur	Utilisateur : administrateur ou client : agent ou réceptionnaire.
Objectif	Visualiser les informations des navires en quai.
Pré-condition	L'utilisateur s'authentifie.
Scénario nominal	<ul style="list-style-type: none"> <li>– L'utilisateur demande l'interface de consultations de la liste des navires en quai depuis le menu ;</li> <li>– Le système affiche l'interface demandée qui contient la liste des navires en quai;</li> </ul>
Alternatives	Si les informations sont inexistantes (aucun navire n'est en quai) ou une erreur s'est produite, le système affiche un message d'erreur approprié et

	réaffiche l'interface d'accueil.
Options	L'utilisateur peut localiser un navire de la liste en appuyant sur le bouton de localisation accompagné.

Tableau 4.2 : Description textuelle du cas d'utilisation « Consulter la liste des navires en quai »

– **Le cas d'utilisation « Consulter la liste journalière des entrées/sorties des navires » :**

Cas d'utilisation	Consulter la liste journalière des entrées/sorties des navires.
Acteur	Utilisateur : administrateur ou client : agent ou réceptionnaire.
Objectif	Visualiser la liste des entrées/sorties des navires effectuées pendant la journée.
Pré-condition	L'utilisateur s'authentifie.
Scénario nominal	<ul style="list-style-type: none"> <li>– L'utilisateur demande l'interface de consultation de la liste journalière des entrées/sorties des navires depuis le menu ;</li> <li>– Le système affiche l'interface demandée qui contient la liste des entrées/sorties des navires ;</li> </ul>
Alternatives	Si les informations sont inexistantes (aucun navire n'a transité le jour même) ou une erreur s'est produite, le système affiche un message d'erreur approprié et réaffiche l'interface d'accueil.
Options	/

Tableau 4.3 : Description textuelle du cas d'utilisation « Consulter la liste journalière des E/S des navires »

– **Le cas d'utilisation « Rechercher un navire » :**

Cas d'utilisation	Rechercher un navire.
Acteur	Utilisateur : administrateur ou client : agent ou réceptionnaire.
Objectif	Visualiser les informations d'un navire donné et son état.
Pré-condition	L'utilisateur s'authentifie.

Scénario nominal	<ul style="list-style-type: none"> <li>– L'utilisateur demande l'interface de recherche d'un navire depuis le menu ;</li> <li>– Le système affiche un formulaire de recherche ;</li> <li>– L'utilisateur saisi le nom du navire et valide ;</li> <li>– Le système vérifie l'existence du navire ;</li> <li>– Le système affiche les informations du navire.</li> </ul>
Alternatives	Si les informations saisies sont invalides (champs non remplis), inexistantes, ou une erreur s'est produite, le système affiche un message d'erreur approprié et réaffiche le formulaire de recherche.
Options	L'utilisateur peut localiser le navire en appuyant sur le bouton de localisation accompagné.

Tableau 4.4 : Description textuelle du cas d'utilisation « Rechercher un navire »

– **Le cas d'utilisation « Consulter la liste des navires d'un client » :**

Cas d'utilisation	Consulter la liste des navires d'un client.
Acteur	Administrateur.
Objectif	Visualiser la liste des navires d'un client donné.
Pré-condition	L'administrateur s'authentifie.
Scénario nominal	<ul style="list-style-type: none"> <li>– L'administrateur demande l'interface appropriée depuis le menu ;</li> <li>– Le système affiche un formulaire de recherche ;</li> <li>– L'administrateur saisi l'identifiant d'un client ;</li> <li>– Le système vérifie l'existence du client et des navires le concernant ;</li> <li>– Le système affiche la liste des navires du client.</li> </ul>
Alternatives	Si les informations saisies sont invalides (champs non remplis), inexistantes, ou une erreur s'est produite, le système affiche un message d'erreur approprié et réaffiche le formulaire de recherche.
Options	L'administrateur peut localiser un navire de la liste en appuyant sur le bouton de localisation accompagné.

Tableau 4.5 : Description textuelle du cas d'utilisation « Consulter la liste des navires d'un client »

– **Le cas d'utilisation « Localiser un navire » :**

Cas d'utilisation	Localiser un navire.
Acteur	Utilisateur : administrateur ou client : agent ou réceptionnaire.
Objectif	Visualiser la position du navire sur une carte.
Pré-condition	L'utilisateur s'authentifie.
Scénario nominal	<ul style="list-style-type: none"> <li>– L'utilisateur clique sur le bouton de localisation d'un navire ;</li> <li>– Le système fait appel au service Google maps ;</li> <li>– Le système affiche la carte avec la position du navire dessus.</li> </ul>
Alternatives	<ul style="list-style-type: none"> <li>– Si l'utilisateur est un administrateur, le système lui affiche la carte avec la position du navire dessus ;</li> <li>– Si le client est un client, le système lui affiche la carte uniquement s'il est propriétaire du navire ou s'il attend son arrivée ;</li> <li>– Si une erreur s'est produite, le système affiche un message d'erreur approprié et réaffiche l'interface supérieure.</li> </ul>
Options	/

Tableau 4.6 : Description textuelle du cas d'utilisation « Localiser un navire »

## 4.2.3 Conception (initiale)

### 4.2.3.1 Vue globale de l'architecture du système

Lors de l'utilisation d'UP, il est indispensable de définir une architecture du système à suivre dès le départ. Dans cette partie, nous illustrons l'architecture globale de la solution informatique que nous envisageons. Le système utilisera une architecture 3-tier et le service web Google maps (défini à la page 62) pour compléter ses tâches et répondre aux cas d'utilisation. Cette architecture définit un client qui se connecte à un serveur via le protocole TCP/IP (internet ou réseau local) en utilisant des requêtes HTTP, ce serveur étant la couche métier, interroge à son tour la base de données MySQL et renvoi les résultats encodés au

format JSON au client via le même protocole. Ce moyen est en fait souvent utilisé lors de la programmation des applications client/serveur sous Android. La figure suivante montre cette architecture :

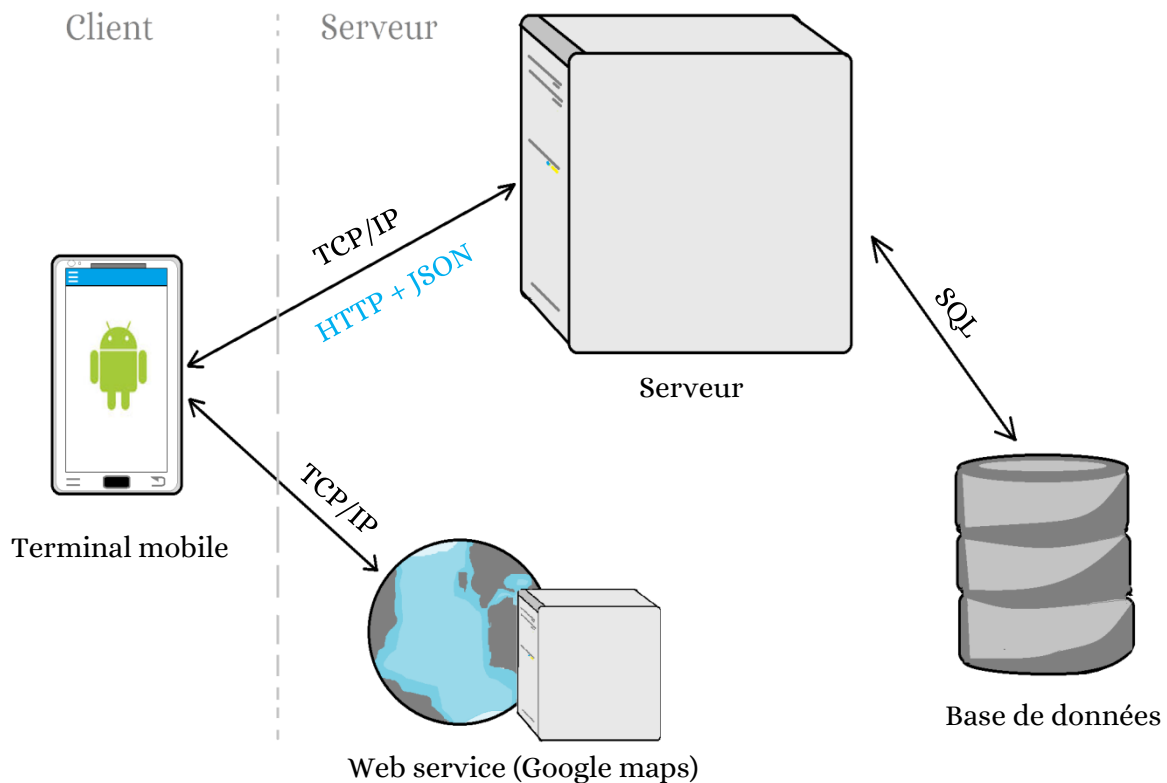


Figure 4.2 : Vue globale de l'architecture du système

#### 4.2.3.2 Diagrammes de séquence système

Dans cette partie, nous exposons les solutions systèmes pour les cas d'utilisation précédents, ces solutions ; en forme de diagrammes de séquences ; décrivent le comportement du système lors de l'exécution du scénario de chaque cas d'utilisation. Le système, ici, inclus tous les composants de l'architecture précédente.

– Le cas d'utilisation « S'authentifier » :

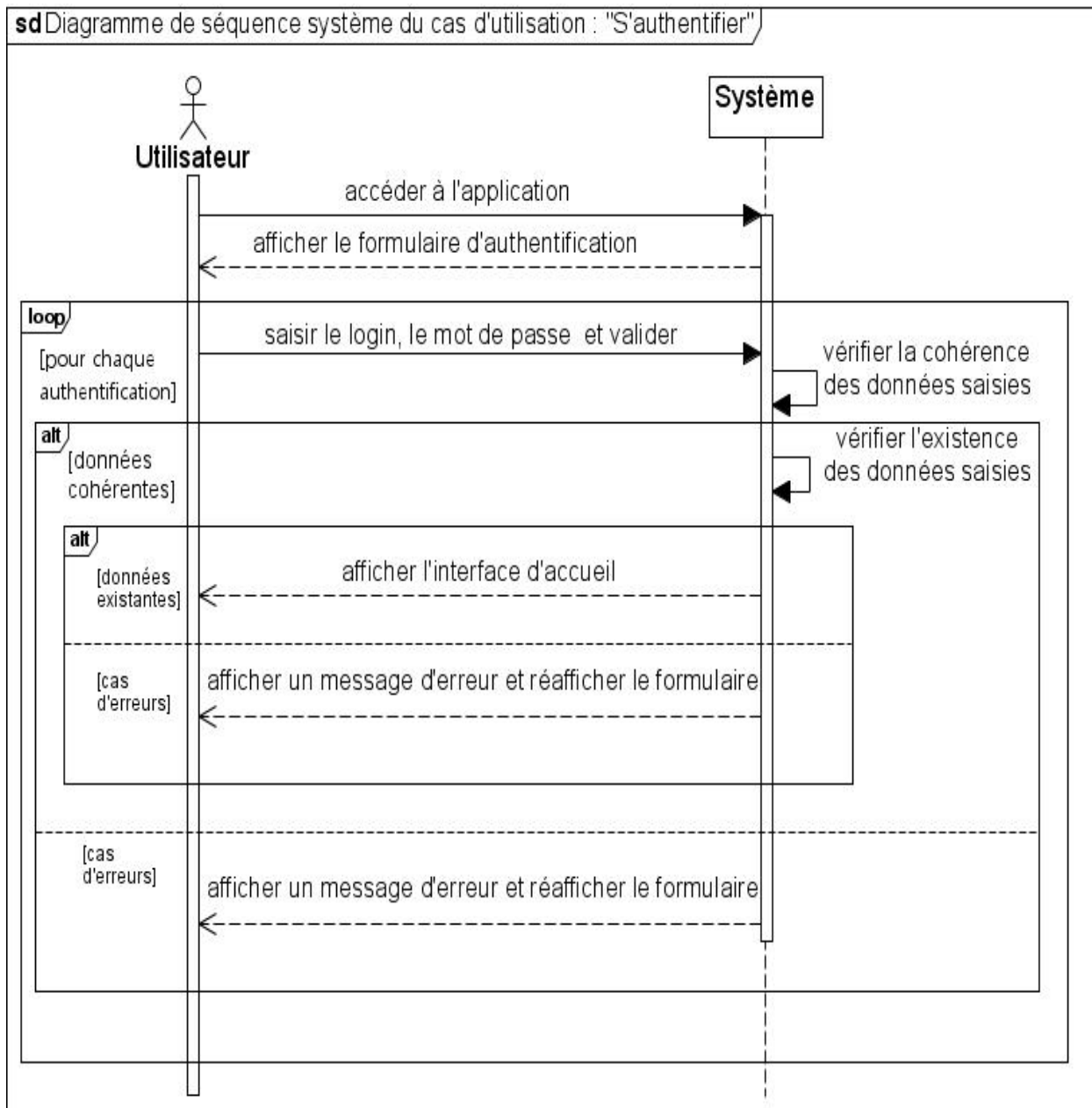


Figure 4.3 : Diagramme de séquence système du cas d'utilisation « S'authentifier »

- Le cas d'utilisation « Consulter la liste des navires en quai » :

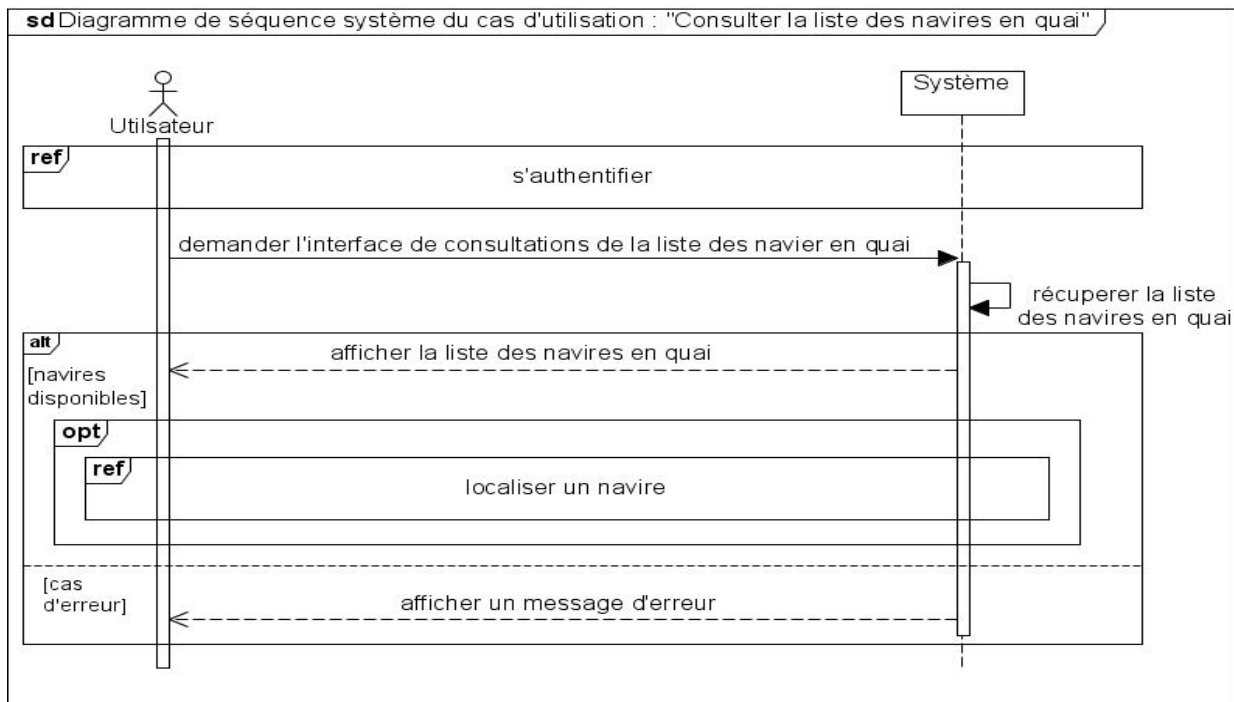


Figure 4.4 : Diagramme de séquence système du cas d'utilisation « Consulter la liste des navires en quai »

- Le cas d'utilisation « Consulter la liste journalière des entrées/sorties des navires » :

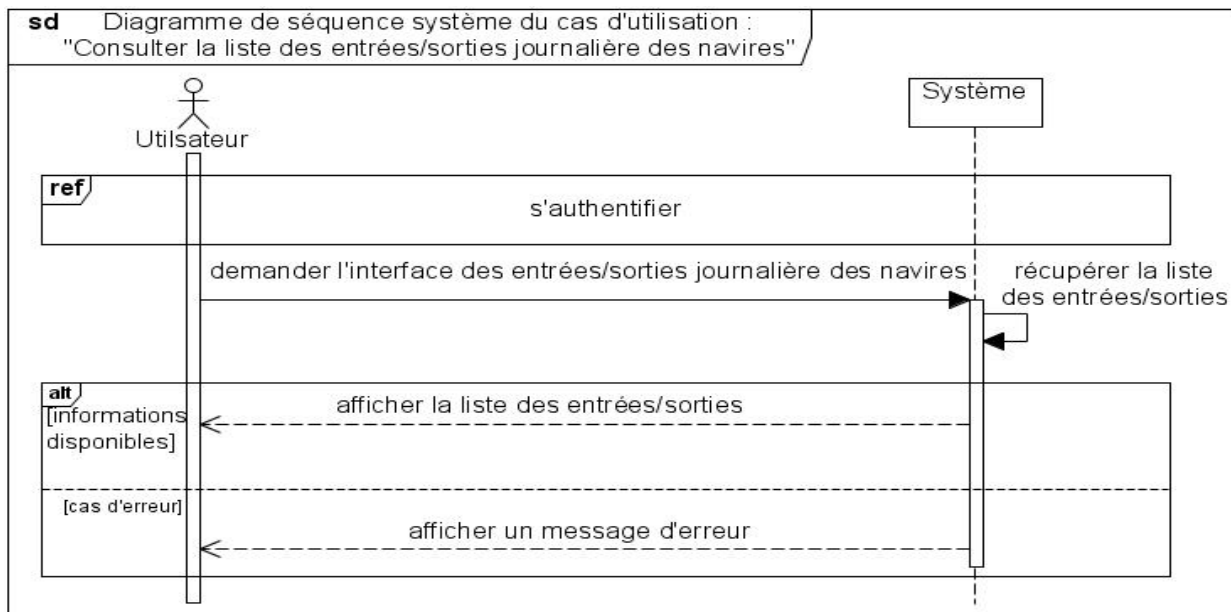


Figure 4.5 : Diagramme de séquence système du cas d'utilisation « Consulter la liste journalière des E/S des navires »

– Le cas d'utilisation « Rechercher un navire » :

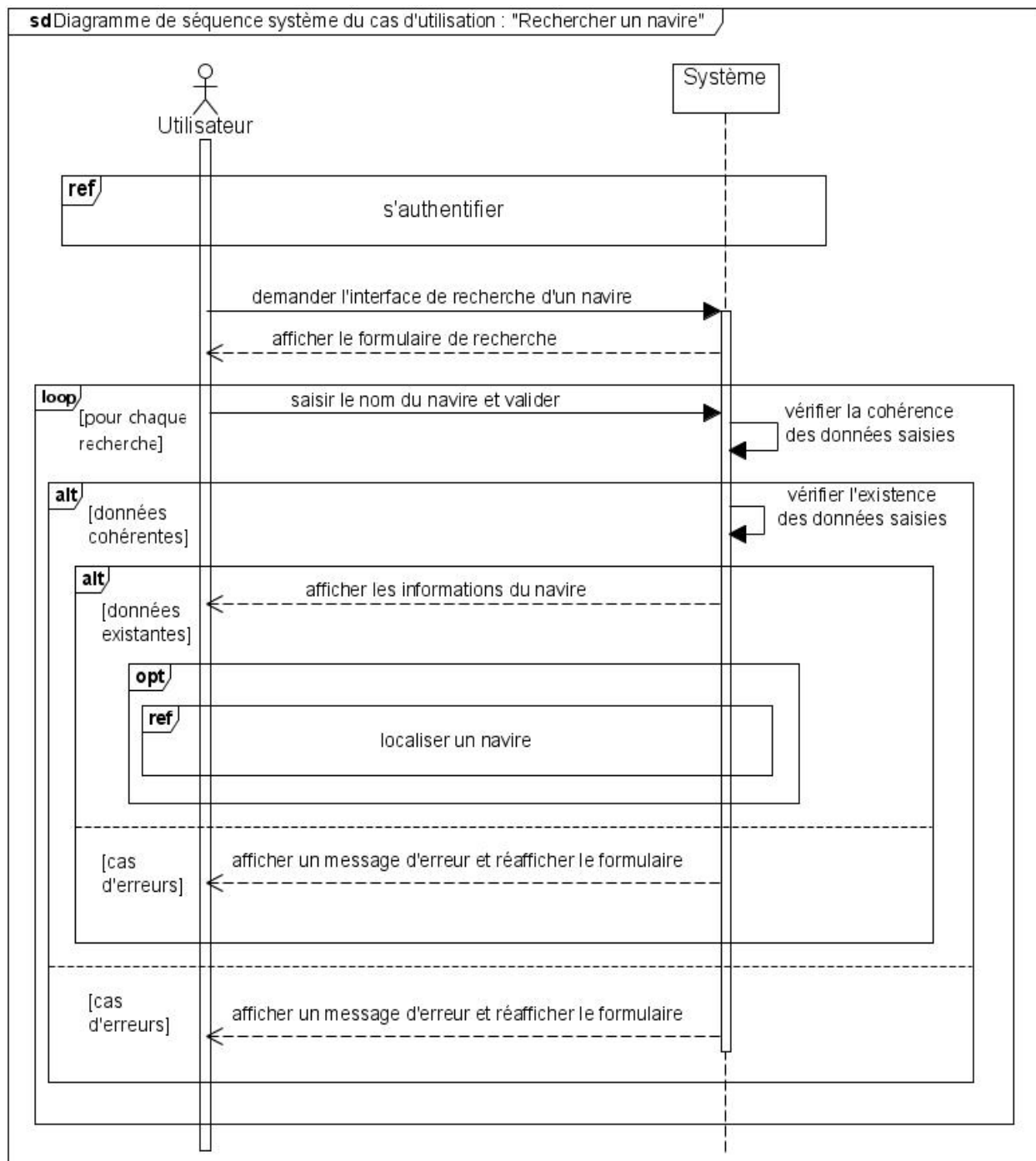


Figure 4.6 : Diagramme de séquence système du cas d'utilisation « Rechercher un navire »



- Le cas d'utilisation « Consulter la liste des navires d'un client » :

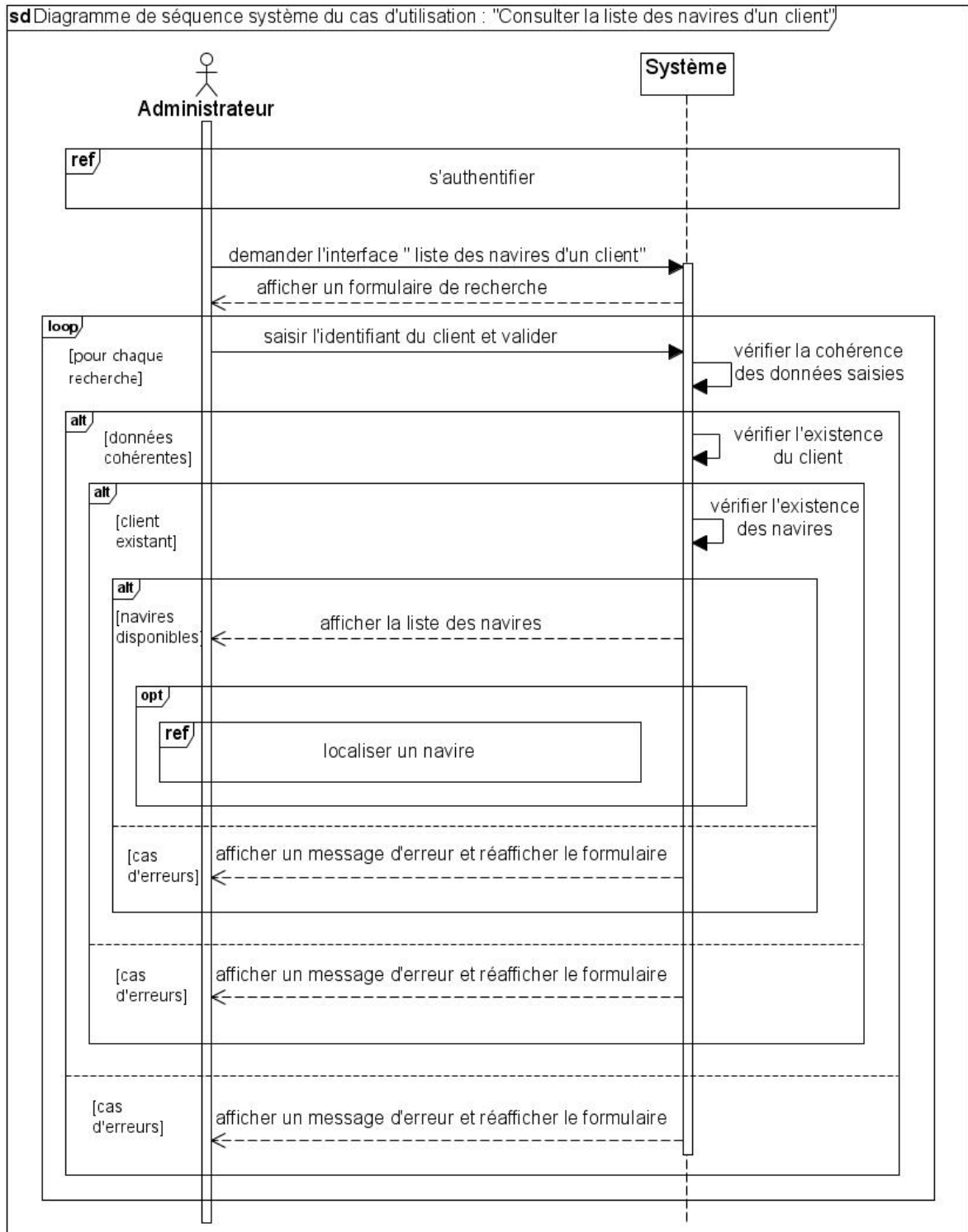


Figure 4.7 : Diagramme de séquence système du cas d'utilisation « Consulter la liste des navires d'un client »

– Le cas d'utilisation « Localiser un navire » :

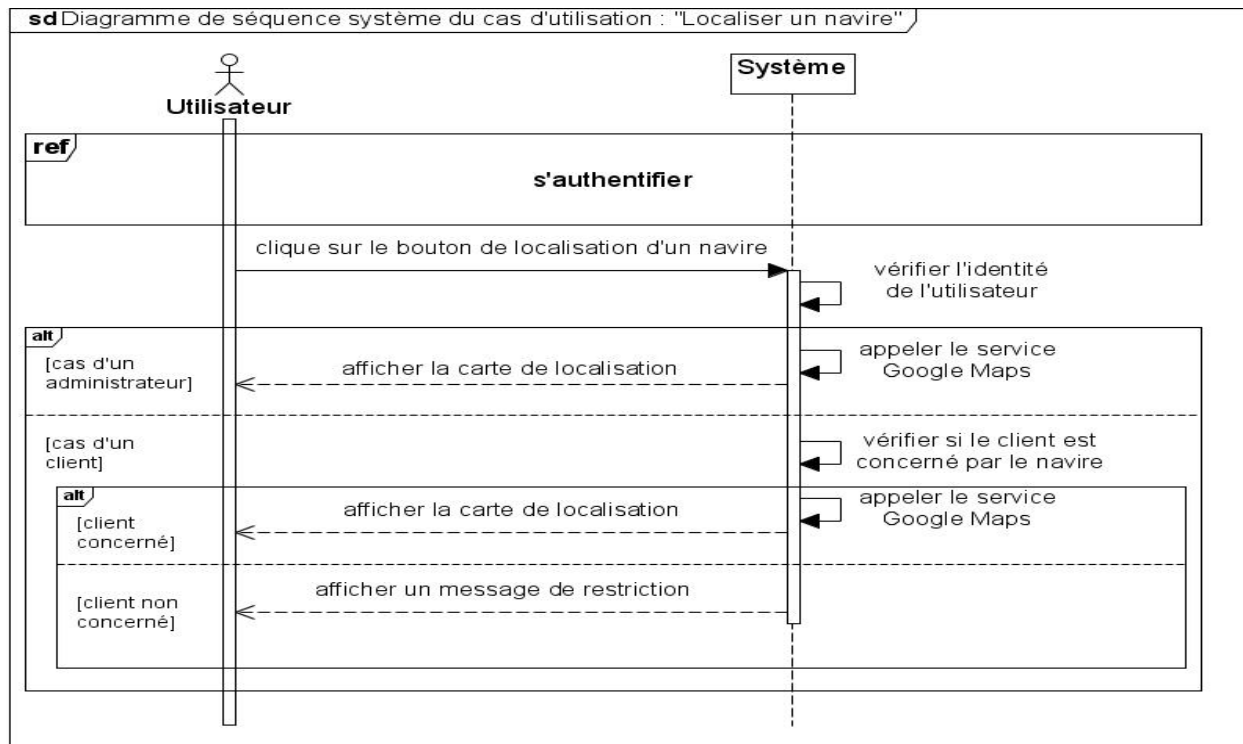


Figure 4.8 : Diagramme de séquence système du cas d'utilisation « Localiser un navire »

## 4.3 La phase d'élaboration

### 4.3.1 Conception (détaillée)

#### 4.3.1.1 Diagrammes de séquence détaillés

Dans cette partie, nous détaillons les diagrammes de séquence systèmes des cas d'utilisation précédents, cet enrichissement qui consiste à diviser le système à des composants internes en suivant l'architecture globale définie préalablement est l'une des philosophies du processus UP, cela nous permettra d'avancer vers la solution finale prête à implémenter tout en évitant de dérailler de l'architecture. Ici, la vue (fichier .XML) et son contrôleur (fichier .Java) se trouvent dans le côté client (terminal mobile) tandis que le modèle (Une classe dans un fichier .PHP) et le contrôleur qui écoute les requêtes HTTP (fichier .PHP) se trouvent dans le côté serveur avec la base de donnés. Voici les diagrammes de séquence détaillés des cas d'utilisation :

– Le cas d'utilisation « S'authentifier » :

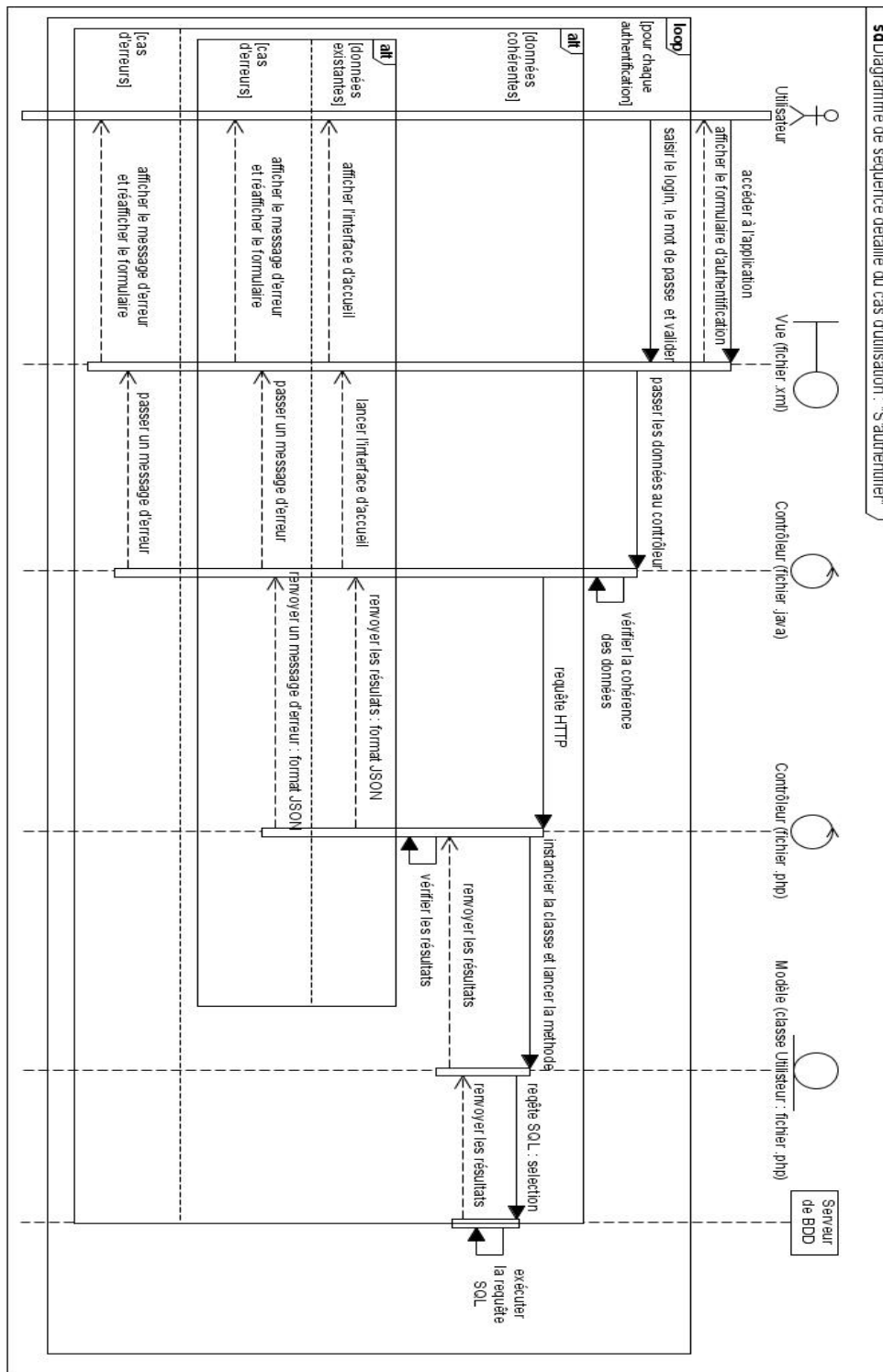


Figure 4.9 : Diagramme de séquence détaillé du cas d'utilisation « S'authentifier »

– Le cas d'utilisation « Consulter la liste des navires en quai » :

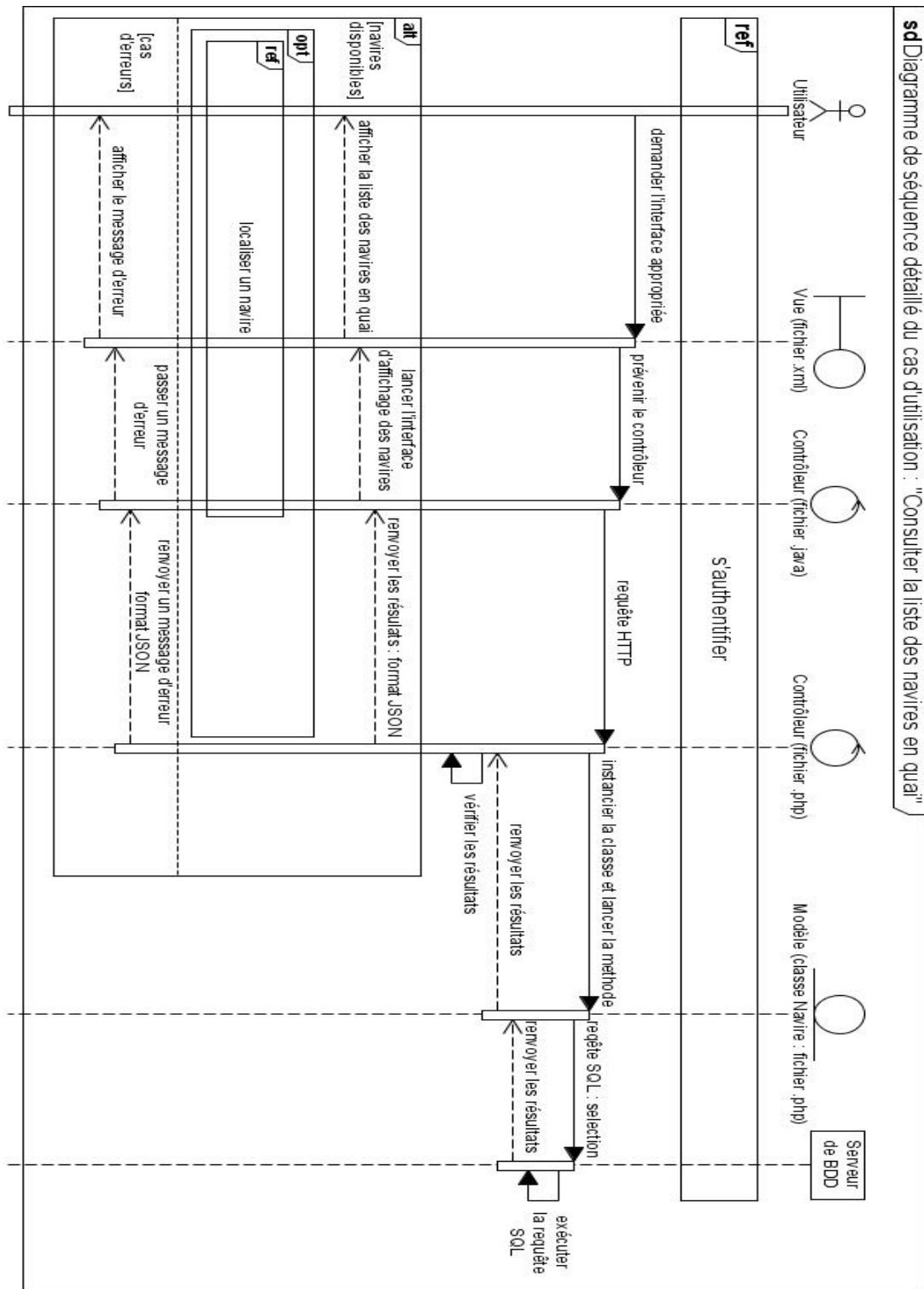


Figure 4.10 : Diagramme de séquence détaillé du cas d'utilisation « Consulter la liste des navires en quai »

- Le cas d'utilisation « Consulter la liste journalière des entrées/sorties des navires » :

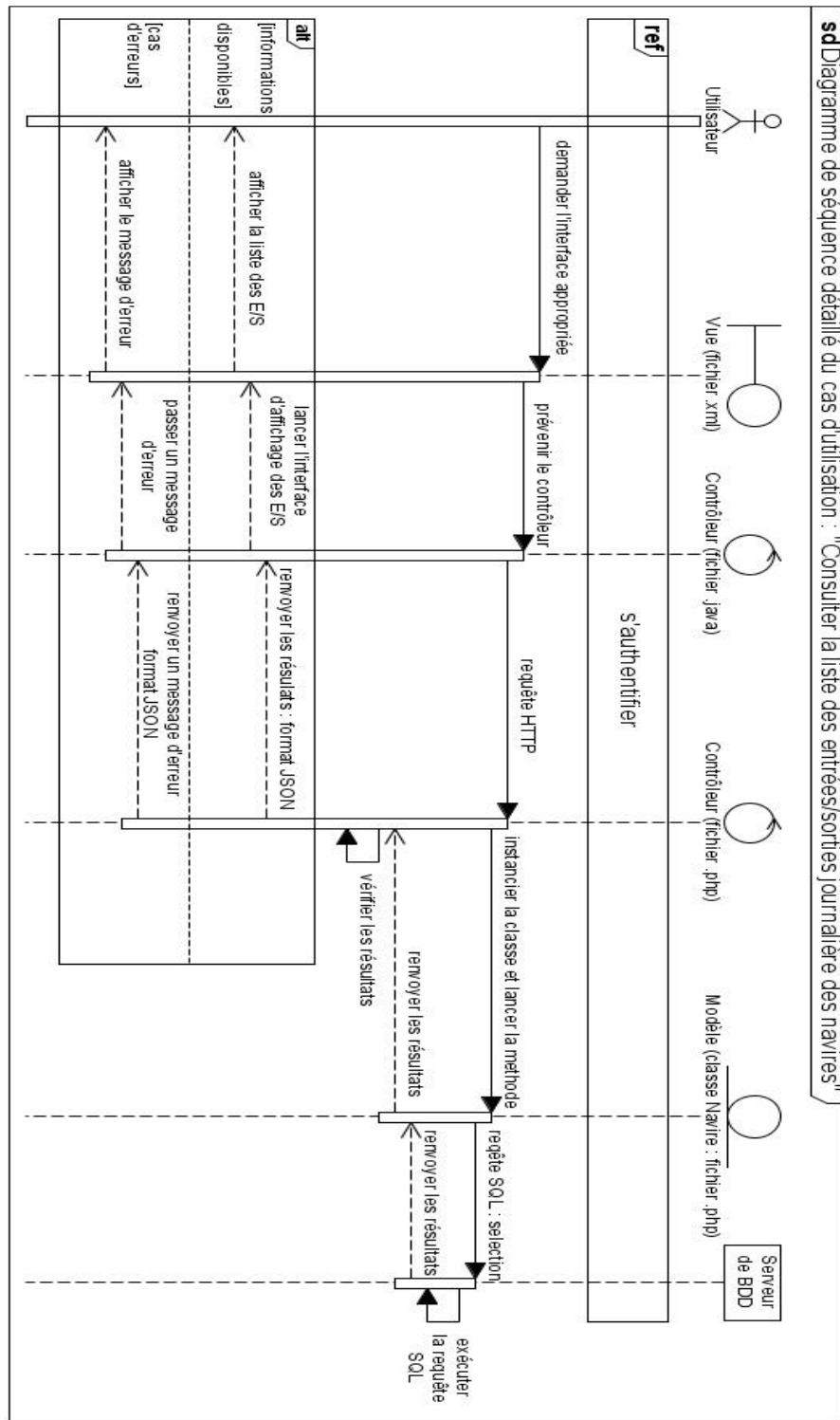


Figure 4.11 : Diagramme de séquence détaillé du cas d'utilisation « Consulter la liste journalière des E/S des navires »

– Le cas d'utilisation « Rechercher un navire » :

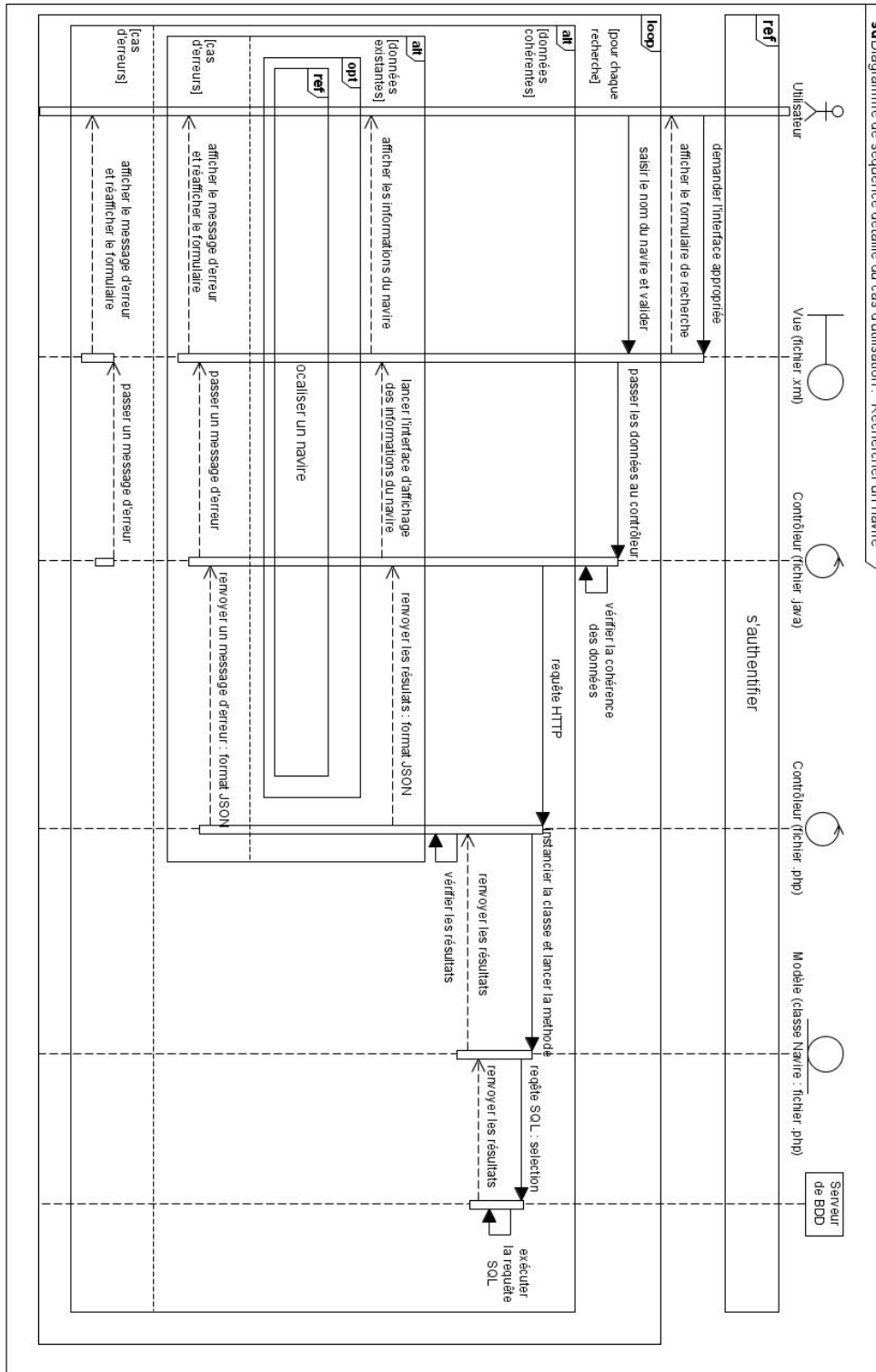


Figure 4.12 : Diagramme de séquence détaillé du cas d'utilisation « Rechercher un navire »



– Le cas d'utilisation « Localiser un navire » :

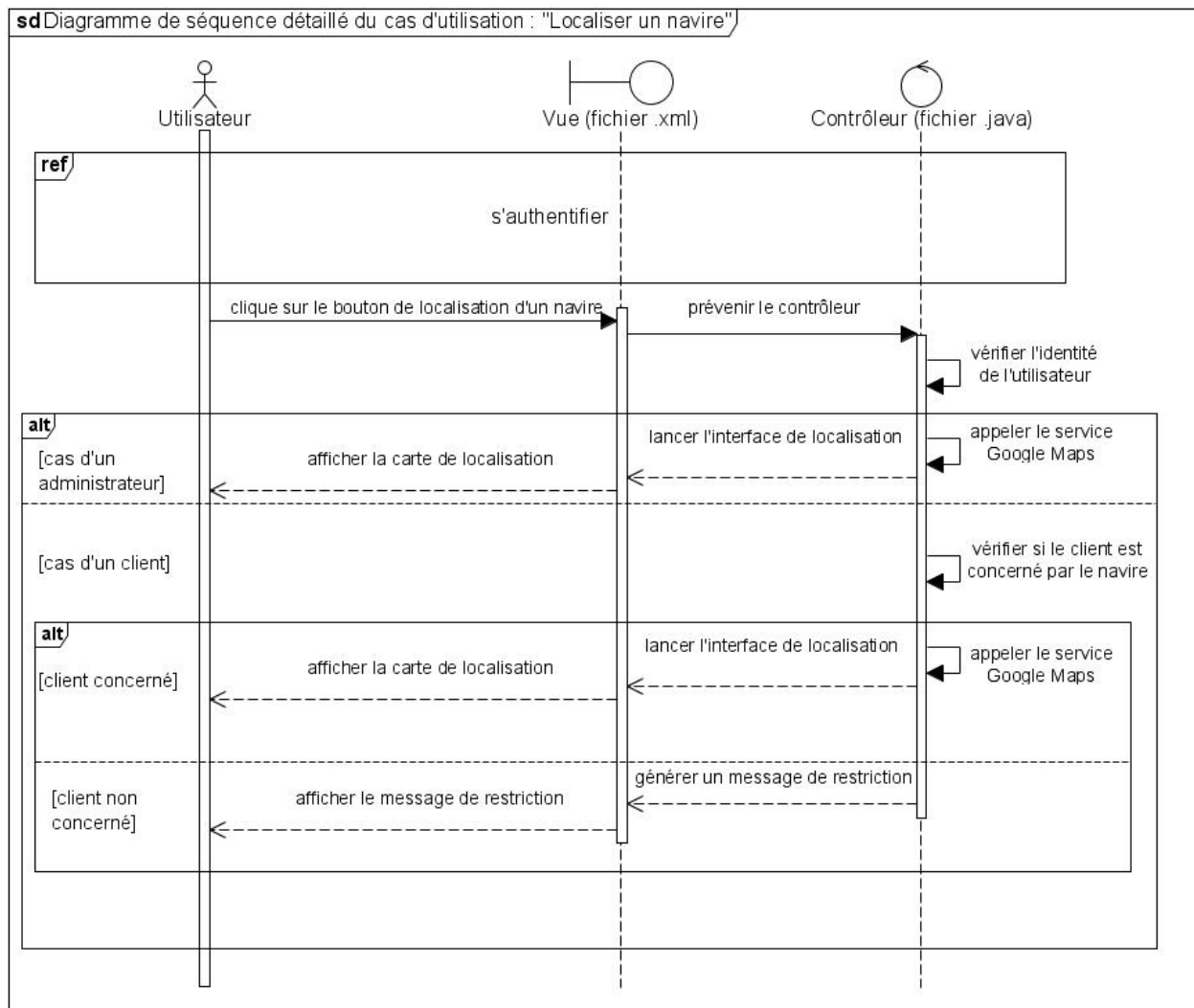


Figure 4.14 : Diagramme de séquence détaillé du cas d'utilisation « Localiser un navire »

#### 4.3.1.2 Diagramme de classes

Dans cette partie, nous entamons le diagramme de classes de notre solution informatique, chaque classe reflète directement une entité du monde réel avec les informations que contient chaque entités (attributs) et les relations entre elles. Nous avons également introduit quelques méthodes à certaines de ces classes pour effectuer les traitements nécessaires.

Voici le diagramme de classes de notre solution informatique :



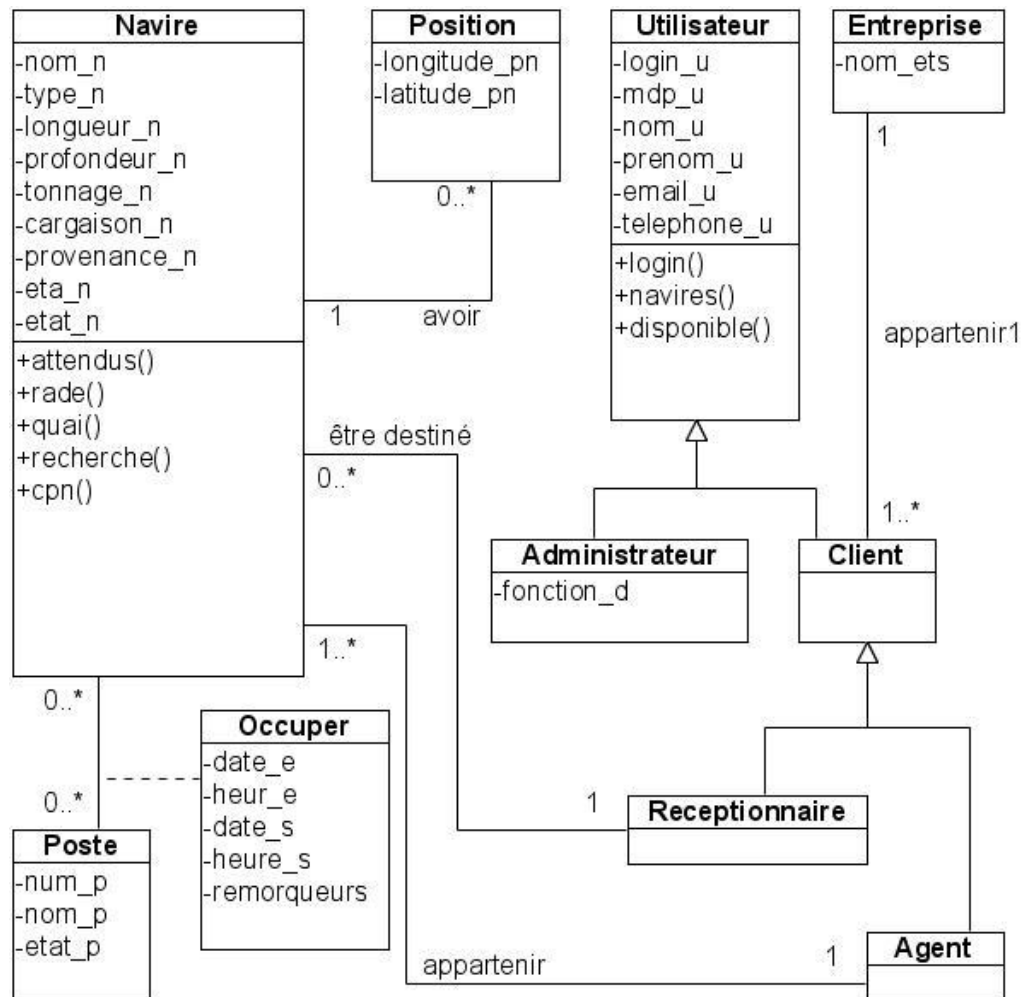


Figure 4.15 : Diagramme de classes de notre solution informatique

L'application des règles de passage à un modèle logique de données relationnel équivalent nous ont permis d'obtenir le MLDR suivant :

- Administrateur (**login\_d**, mdp\_d, nom\_d, prenom\_d, email\_d, telephone\_d, fonction\_d) ;
- Entreprise (**nom\_ets**) ;
- Receptionnaire (**login\_r**, mdp\_r, nom\_r, prenom\_r, email\_r, telephone\_r, nom\_ets\_r#) ;
- Agent (**login\_a**, mdp\_a, nom\_a, prenom\_a, email\_a, telephone\_a, nom\_ets\_a#) ;
- Navire (**nom\_n**, type\_n, longueur\_n, profondeur\_n, tonnage\_n, cargaison\_n, provenance\_n, eta\_n, etat\_n, login\_r#, login\_a#) ;
- Position (**longitude\_pn**, **latitude\_pn**, nom\_n#) ;
- Poste (**num\_p**, nom\_p, etat\_p) ;
- Occuper (**nom\_n#**, **num\_p#**, **date\_e**, **heur\_e**, date\_s, heur\_s, remorqueurs).

## 4.4 La phase de construction

### 4.4.1 Implémentation

#### 4.4.1.1 Langages, outils et technologies de développement utilisés

– **Le langage Java :**

Java est un langage de programmation orienté objet, développé par SUN Microsystems et destiné à fonctionner dans une machine virtuelle, de ce fait, il permet de créer des logiciels compatibles avec des nombreux systèmes d'exploitation dont Android.

Java est non seulement un langage de programmation puissant conçu pour être sûr, inter-plateformes et international, mais aussi un langage de développement qui est continuellement étendu pour fournir des nouvelles caractéristiques et des bibliothèques permettant de gérer de manière élégante des problèmes traditionnellement complexes dans les langages de programmation classiques, tels que le multithreading, les accès aux bases des données, la programmation réseau, l'informatique répartie...etc.[T1].

Pour pouvoir programmer en java, on a besoin d'une machine virtuelle et d'un ensemble de bibliothèques de références, SUN Microsystems offre dans ce but le JDK (Java Development Kit), un outil devant être installé sur la machine de programmation.

– **Le langage XML :**

XML est l'acronyme de « eXtensible Markup Language », ce qui signifie en français « langage de balisage extensible ». Il s'agit d'un langage de création de fichiers de données ayant la particularité de permettre la création et la définition de balises à la demande (d'où l'origine de l'expression « eXtensible »). XML a été conçu pour permettre les transferts d'informations d'une plateforme à une autre sans avoir à développer les interfaces.

Comme XML est un langage de balisage qui permet une bonne structuration, Android convertit automatiquement les interfaces créées avec ce langage en classes Java, c'est tout l'intérêt d'avoir fait Android créer les interfaces avec le langage XML. Cela offre une souplesse de développement, facilite la modification du code et assure la séparation entre la présentation et le comportement des composants de l'application [w25].

– **Le langage PHP :**

PHP est l'acronyme de « Php Hypertext Preprocessor », il signifiait à l'origine « Personal Home Page » quand il a été créé par Rasmus LERDORF en 1994 pour ajouter quelques fonctionnalités aux pages personnelles HTML [07].

PHP est un langage de script qui s'exécute au côté serveur, son rôle est principalement de créer des pages web dynamiques et interactives. Mais cela ne l'empêche pas d'être utilisé pour une autre fin.

– **Le format JSON :**

JSON est l'acronyme de « JavaScript Object Notation », ce qui se traduit par « Notation Objet issue de JavaScript », c'est un format léger d'échange de données. Il est facile à lire ou à écrire pour des humains, aisément analysé ou généré par des machines, basé sur un sous-ensemble du langage de programmation JavaScript (JavaScript Programming Language).

JSON est un format texte complètement indépendant de tout langage, mais les conventions qu'il utilise seront familières à tout programmeur habitué aux langages descendant du C (comme le Java). Il ne comprend que deux éléments structurels : des ensembles de paires nom / valeur et des listes ordonnées de valeurs. Ces mêmes éléments représentent trois types de données :

- Des objets ;
- Des tableaux ;
- Des valeurs génériques de type tableau, objet, booléen, nombre, chaîne ou nul [T1].

– **La plate-forme de développement WampServer :**

WampServer est un environnement de travail local destiné pour le développement web, gratuit, créé par Roumain BOURDON, il est composé de trois éléments permettant de simuler un vrai serveur web [w28] :

- Un serveur web « APACHE » ;
- Un serveur de script « PHP » ;
- Un serveur de bases de données « MySQL ».

WampServer est accompagné d'un outil pour gérer les bases de données nommé PHPMAdmin.

– **L'environnement de développement intégré (IDE) Android studio :**

Google a créé un IDE complet pour la création d'application mobile Android nommé Android Studio, annoncé le 15 mai 2013. Il est open source et disponible gratuitement, permettant de réaliser des projets sur différents types de support Android (tablettes, smartphones, téléviseurs, gadgets électroniques...etc) [w26].

Principalement utilisé pour éditer des fichiers Java étant le langage par défaut d'une application Android native ainsi que des fichiers de mise en page XML avec la possibilité de visualiser le rendu et les manipuler en utilisant une interface graphique.

Comme Android studio utilise le langage Java, une installation préalable du JDK est requise. Un autre prérequis est le SDK (Software Development Kit) Android.

– **L'outil de développement logiciel SDK :**

L'outil le plus important est le SDK Android. Ce kit de développement inclut les utilitaires nécessaires pour aider les développeurs Android : les différents APIs développés par Google à la fois pour contrôler les fonctions de l'appareil et l'intégration des services web (tel que Google maps), un émulateur complet pour tester des applications, c'est un dispositif virtuel pour exécuter les applications, et prend en charge la connexion de données, le WiFi, le GPS, la fonctionnalité tactile pour l'écran, et toutes autres technologies destinée aux smartphones [w26].

Nous avons installé le SDK Android avec toutes les API de Google depuis l'API 7 (Android 2.1 Eclair) jusqu'à l'API 25 (Android 7.1.1 Nougat), cela nous donne un large choix de plateformes auxquelles notre application pourrait être destiné. Nous avons choisi de faire cibler notre application aux terminaux mobiles supportant au moins l'API 19 (Android 4.4 KitKat) qui possède une bonne mise à jour des services Google, cela permettra une couverture d'environ 80% des appareils Android.

– **Le service Google maps :**

Google Maps est un exemple de la technologie des services web. Par définition, un service web est un système logiciel identifié par un URI, dont les interfaces publiques sont définies et décrites en XML. Sa définition peut être découverte [dynamiquement] par d'autres systèmes logiciels. Ces autres systèmes peuvent ensuite interagir avec le service web et le consommer d'une façon décrite par sa définition en utilisant des messages XML transportés par des protocoles internet (principalement un appel de méthodes du service)[08].

Il existe deux types de services web :

- Services Web SOAP : Utilisent le protocole SOAP (Simple Object Access Protocol) et le langage XML, d'après les statistiques, les services web de ce type sont estimés couvrir 15% de la totalité des services web ;
- Services Web RESTful : Utilisent l'approche REST (Representational State Transfer), basés sur les méthodes standards du protocole HTTP (GET, PUT, POST, DELETE) ainsi que JSON or XML. Ils occupent 85% de la totalité des services web, de ce fait, nombreux acteurs fournissent des services de ce genre, tels que Amazon, Facebook, Yahoo, Twitter, Google...etc [09].

Google Maps est un service web RESTful gratuit de cartographie en ligne créé par Google et lancé en 2004 aux États-Unis. C'est un service qui permet, à partir de l'échelle d'un pays, de zoomer jusqu'à l'échelle d'une rue, il est utilisé essentiellement pour se localiser ou localiser un endroit ou une cible sur une carte [w29].

Il existe également plusieurs versions mobiles de Google Maps, qui utilisent les réseaux des téléphones notamment 3G pour charger les cartes de la même manière que sur la version internet, Google est allé même à accompagner son système Android par l'application « Maps ».

Toutefois, pour consommer ce service lors de la programmation d'une application Android, comme dans notre cas, il faut accéder à une URI générée par Android Studio (plus exactement par l'API Google) avec un compte Gmail, cette URI fait transférer avec une requête HTTP des informations relatives au projet Android et la machine de développement qui le contient. Suite à quoi, le site de Google ciblé par l'URI génère une clé à introduire dans le code, cette clé est supposée être unique pour chaque projet Android utilisant Google maps.

#### 4.4.1.2 Le pattern MVC et la structuration du code

Pour organiser le contenu de notre solution, nous avons utilisé le pattern MVC (« Modèle Vue Contrôleur » en français ou « Model View Controller » en anglais). MVC est un pattern (modèle) de stylisme en anglais dit « design pattern », c'est une simple bonne pratique d'organisation du code source lors de l'implémentation d'une application.

MVC impose une certaine structuration :

- Tout ce qui concerne l'interaction avec l'utilisateur et la présentation (affichage) des données doit être contenu dans la couche nommée « Vue » ;
- Tout ce qui concerne le traitement, le stockage et la mise à jour des données de l'application doit être contenu dans la couche nommée « Modèle » ;
- Tout ce qui concerne l'écoute et le contrôle des actions de l'utilisateur et la relation entre et la vue et le modèle doit être contenu dans la couche nommée « contrôleur » [06].

En suivant ces principes, notre application Android comprend :

- Des composants graphiques : ce sont des fichiers .XML qui représentent les interfaces graphiques de l'application, ils sont disponibles dans le répertoire du chemin : « app > res > layout » dont le répertoire racine de l'application est « app ». Ces fichiers correspondent à la couche « Vue » ;
- Des composants de contrôle : ce sont des fichiers .Java qui représentent les écouteurs des interfaces graphiques de l'application, ce sont les piliers de l'application car ils définissent toutes les actions à faire, ils sont disponibles dans le répertoire du chemin : « app > java > dz.portdebejaia.www.epb » dont le répertoire racine de l'application est « app » et le dernier répertoire est une combinaison à choisir lors de la création de l'application. Ces fichiers correspondent à la couche « Contrôleur » ;

À chaque composant graphique .xml est associé un contrôleur .Java, l'association de ces deux fichiers constitue ce qu'on appelle une « activité (Activity) » Android.

Comme notre application contient également une partie serveur avec laquelle communique la partie client, elle comprend aussi :

- Un fichier de contrôle coté serveur nommé « Main » : c'est un fichier .PHP, il sera déployé sur le serveur afin d'écouter les requêtes HTTP émises par l'application Android, interroger le modèle pour effectuer le traitement adéquat et renvoyer les résultats encodés au format JSON vers le client. Ce fichier correspond à la couche « Contrôleur » côté serveur ;
- Un fichier de classes coté serveur nommé « Model » : c'est un fichier .PHP, il sera déployé sur le serveur et contient les classes définies dans le diagramme de classes, ces classes, une fois instanciés par le contrôleur, peuvent à leur tour interroger la base de données et récupérer les informations. Ce fichier correspond à la couche « Modèle ».

Cette structuration (la partie serveur surtout) est mieux illustrée dans le diagramme de déploiement.

#### 4.4.1.3 Diagramme de déploiement

Le diagramme de déploiement suivant montre la distribution de différents composants de notre solution informatique (composants, artefacts et base de données) sur les différents terminaux matériels (nœuds sous forme de cubes), les dépendances entre les composants (flèches discontinues) et les relations (connexions) entre les terminaux (droite continue) :

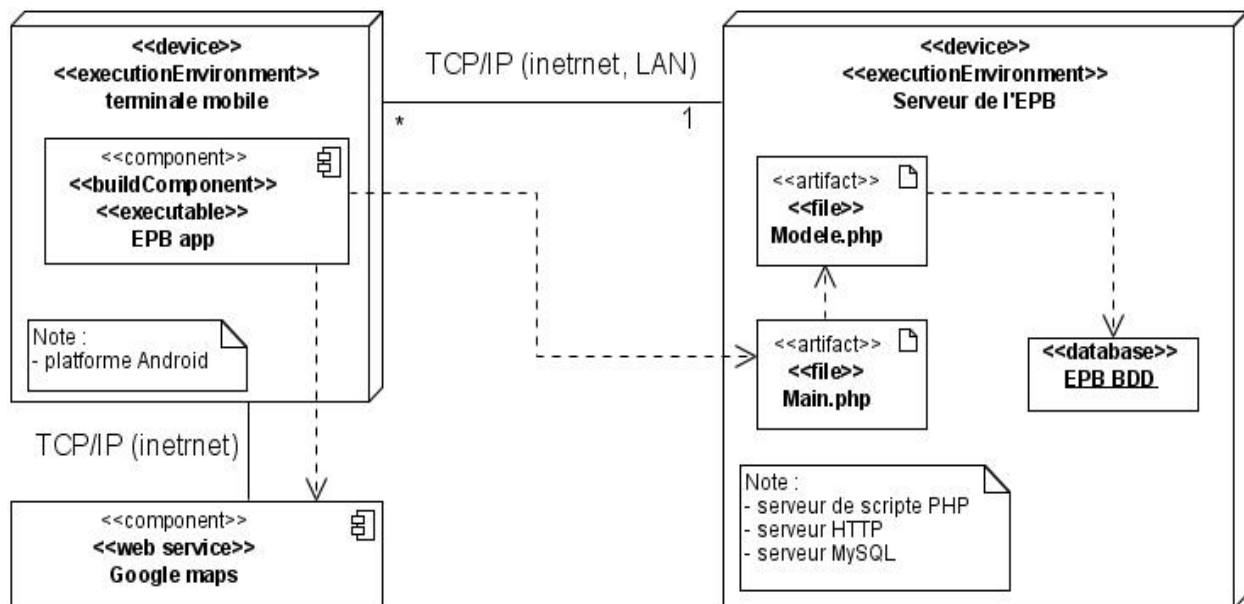


Figure 4.16 : Diagramme de déploiement de notre solution informatique

## 4.5 La phase de Transition

### 4.5.1 Tests

Voici quelques interfaces montrant certains des tests de fonctionnalités de l'application :

– **Test d'authentification :**

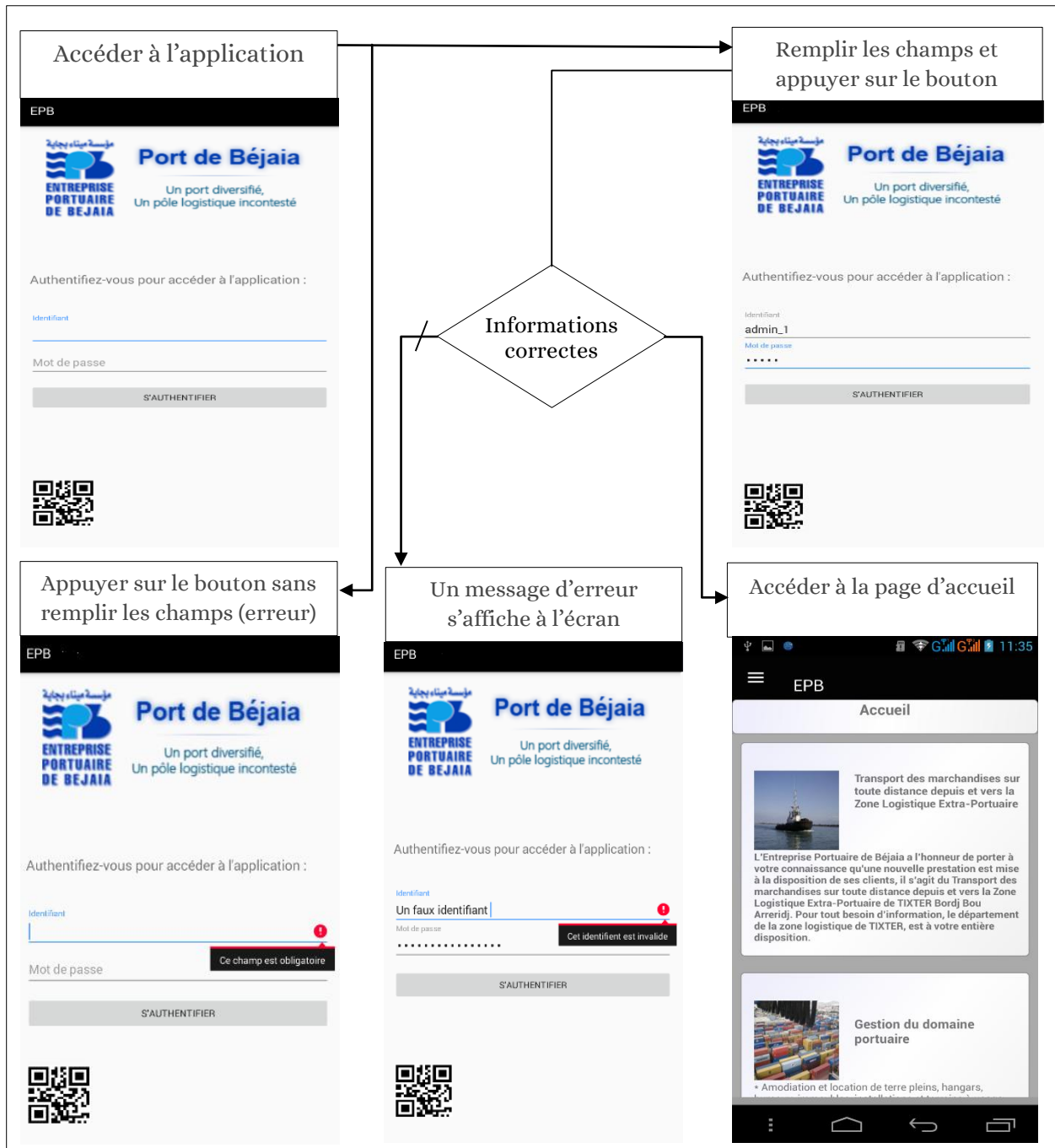


Figure 4.17 : Organigramme illustré montrant le test d'authentification



– Test de consultation de la liste des navires en rade (cas d'un administrateur authentifié) :

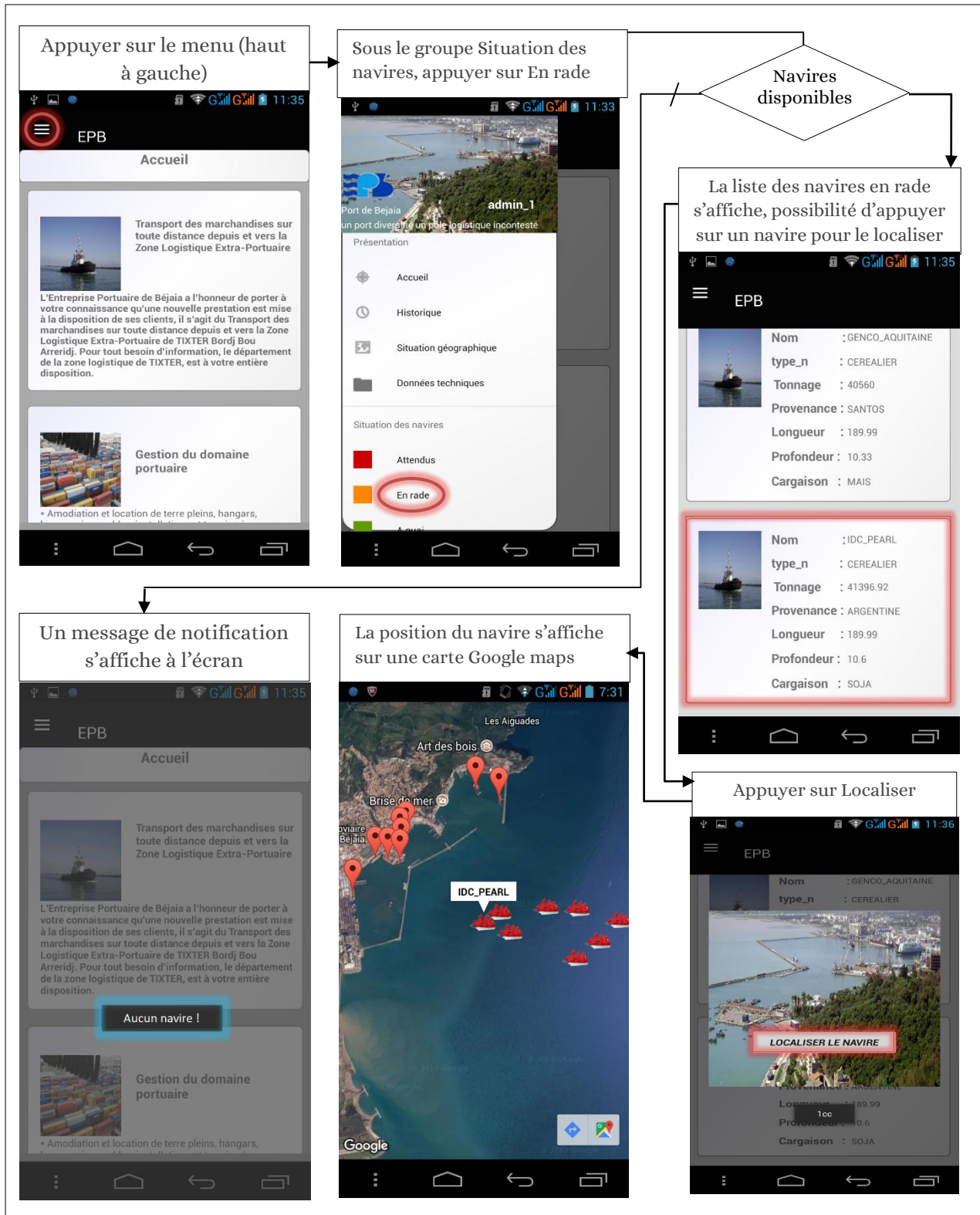


Figure 4.18 : Organigramme illustré montrant le test de consultation de la liste des navires en rade

## 4.6 Conclusion

Dans ce chapitre, nous avons présenté la conception et la réalisation de notre projet, ces deux mots peuvent résumer implicitement les quatre phases du processus UP dont nous avons suivi les activités les plus pertinentes.

Ces activités sont refaites de manière itérative jusqu'à ce qu'on obtient un résultat satisfaisant en se rapprochant de l'architecture souhaité qui devient de plus en plus stable en avançant dans le procédé d'UP, certaines des activités sont évidemment plus fréquentes dans des phases que d'autres, par exemple l'expression et l'analyse des besoins sont essentiellement plus utilisées dans la phase d'inception que dans les trois restantes et plus utilisées que les autres activités durant cette phase. Ces activités sont d'ailleurs les plus importantes sur lesquelles repose tout le processus.

Nous avons également exposé à travers quelques diagrammes UML et d'autres encore l'essentiel de notre travail de manière brève depuis l'expression des besoins jusqu'aux tests de fonctionnement de notre application, en passant par l'analyse des besoins, la conception et l'implémentation suivant les quatre phases.

Nous avons pareillement présenté les outils de travail et les langages de programmation sur lesquels nous nous sommes basés pour implémenter au mieux notre conception. Sur ce point, on conclut que le langage Java et XML sont des prérequis nécessaires pour développer une application Android native, le JSON quant à lui, il est indispensable pour assurer la communication entre les différents langages des composants du système distribués sur une architecture client-serveur : on parle de communication entre le langage Java et le langage PHP assuré par le format JSON. La maîtrise d'un langage côté serveur comme PHP est également nécessaire pour le développement d'une application Android client-serveur.

D'un autre côté, l'utilisation d'un patron de conception (design pattern) n'est pas obligatoire. Néanmoins, structurer son code à l'aide d'un modèle de stylisme tel que MVC n'est pas à négliger, en fait cette pratique facilite la compréhension des composants de l'application et suit une certaine logique pour mieux se retrouver dans le code, son intérêt est nettement plus visible lorsqu'un développeur essaie de comprendre le travail d'un autre ou parfois même son propre travail.

---

# CONCLUSION GÉNÉRALE

## ET PERSPECTIVES

---

Avec les avancées de la technologie du mobile, la mobilité des systèmes d'informations en entreprises devient une nécessité. Cette pratique se révèle très efficace quant à l'accès aux informations indépendamment du temps et de lieu : On peut désormais effectuer ses tâches sans l'obligation d'être physiquement présent à un endroit ou un poste donné.

Cette mobilité des systèmes d'informations est devenue concrète grâce aux possibilités qu'apportent les mobiles intelligents, principalement par le biais des applications web mobiles ou d'applications natives. Toutefois, pour le développement des applications natives, on se confronte au fait que ces appareils sont disponibles sous plusieurs plateformes, quand on revient au choix de la plateforme, la question risque de ne plus se poser : Android, un système libre et open source, avec sa domination sur le marché demeure le premier choix des utilisateurs. D'après les recommandations de l'entreprise portuaire de Bejaia, qui consiste à favoriser les logiciels et solutions libres, nous avons opté pour le système Android afin de développer une application mobile native dédiée à la consultation des états des navires pour cette entreprise, son but est d'améliorer la mobilité intérieure et étendre l'accès des clients à l'extérieur de la firme.

Pour réaliser cette application, nous avons utilisé le processus de développement UP qui s'appuie sur le formalisme UML, nous avons pu, à travers les quatre phases de ce processus, arriver à notre objectif.

Pendant les phases d'inception et d'élaboration, nous avons analysé et conçu notre solution basée sur une architecture client/serveur à 3 tiers, nous avons utilisé les diagrammes de cas d'utilisation, de séquence et de classes.

Pendant la phase de construction, nous avons implémenté le code avec le langage Java côté client et le langage PHP côté serveur en utilisant le pattern MVC tout en respectant l'architecture définie. La communication entre ces deux langages différents est assurée avec des requêtes HTTP et l'encodage des réponses du serveur au format JSON qui sont reconnaissables par les deux langages. Nous avons également illustré le déploiement des composants de notre solution sur les différents terminaux physiques avec le diagramme de déploiement.

Pendant la phase de transition, nous avons exécuté des tests sur notre application Android pour bien assurer le fonctionnement de notre solution, nous avons illustré quelques-uns de ces tests avec des captures d'écrans des interfaces de notre application.

Pour conclure, ce projet a fait l'objet d'une expérience intéressante qui nous a permis d'exploiter nos compétences, d'élargir nos connaissances et de découvrir un nouveau domaine de programmation pour nous : celui des applications mobiles.

Cependant, notre travail tel qu'il est, possède des insuffisances. Nous avons donc quelques perspectives en tête pour des futures améliorations, à savoir :

- Enrichir l'application avec l'ajout des fonctionnalités de gestion (ajout, modification et suppression) des données du système d'information pour l'administrateur ;
- Enrichir l'application avec l'ajout de fonctionnalités de présentations dynamiques (mises à jour régulièrement) telles que les données techniques de l'entreprise, ses chiffres et bilans d'activité, statistiques...etc ;
- Enrichir l'application avec l'ajout des fonctionnalités se basant sur d'autres services web telle que la météo ;
- Ajouter d'autres acteurs comme le transitaire, généraliser l'administrateur...etc.

---

# RÉFÉRENCES

---

## Ouvrages :

- [01]. James Douglass LEFRUIT, Android Introduction générale, INRIA Grenoble, date non mentionnée.
- [02]. Jean-François LALANDE, Développement Android, INSA Centre Val de Loire, 2016.
- [03]. Frédéric ESPIAU, créez des applications pour Android, openclassrooms, 2013.
- [04]. Christine SOLNON, Modélisation UML, INSA de Lyon, 2013.
- [05]. D.GABAY & J. GUBAY, UML2 Analyse et conception, Paris : du nord, 2008.
- [06]. CYSBOY, Apprenez à créer des applications web avec JEE, openclassrooms, 2011.
- [07]. Jean ENGELS, PHP5 cours et exercices, Eyrolles, 2009.
- [08]. Dr BRAHAMI Née EL BOUHISSI Houda, Cours de services web, Université de Bejaia, 2017.
- [09]. Mickaël BARON, Web services REST, ENSMA France, 2011.

## Thèses :

- [T1]. Aymen HEDIDAR, conception et réalisation d'une application mobile M-Banking, Université virtuelle de Tunis, 2012.

## Web (dernière visite : le 15/06/2017) :

- [w1]. [www.portdebejaia.dz/index.php/fr](http://www.portdebejaia.dz/index.php/fr)
- [w2]. [searchmobilecomputing.techtarget.com/definition/mobile-operating-system](http://searchmobilecomputing.techtarget.com/definition/mobile-operating-system)
- [w3]. [www.tomsguide.fr/article/guide-smartphone-acheter,2-1219-3.html](http://www.tomsguide.fr/article/guide-smartphone-acheter,2-1219-3.html).
- [w4]. [www.cbronline.com/news/mobility/security/ios-vs-android-vs-windows-phone-vs-blackberry-which-mobile-os-is-right-for-your-enterprise-mobility-programme-4857822](http://www.cbronline.com/news/mobility/security/ios-vs-android-vs-windows-phone-vs-blackberry-which-mobile-os-is-right-for-your-enterprise-mobility-programme-4857822)
- [w5]. [www.techseo.xyz/2016/08/avantages-et-inconvenients-android.html](http://www.techseo.xyz/2016/08/avantages-et-inconvenients-android.html)
- [w6]. [www.futur.fr/actualites/278](http://www.futur.fr/actualites/278)
- [w7]. [www.tienregion.com/actualites/windows-phone-iphone-android-blackberry/](http://www.tienregion.com/actualites/windows-phone-iphone-android-blackberry/)

- 
- [w8]. [www.zdnet.fr/actualites/chiffres-cles-les-os-pour-smartphones-39790245.htm](http://www.zdnet.fr/actualites/chiffres-cles-les-os-pour-smartphones-39790245.htm)
- [w9]. [www.definitions-marketing.com/definition/application-mobile/](http://www.definitions-marketing.com/definition/application-mobile/)
- [w10]. [www.openclassrooms.com/courses/developpez-une-application-mobile-multi-plateforme-avec-ionic/qu-est-ce-que-ionic](http://www.openclassrooms.com/courses/developpez-une-application-mobile-multi-plateforme-avec-ionic/qu-est-ce-que-ionic)
- [w11]. [www.taktilcommunication.com/blog/applications-mobile/definition-typologie-applications-mobiles.html](http://www.taktilcommunication.com/blog/applications-mobile/definition-typologie-applications-mobiles.html)
- [w12]. [www.commentcamarche.net/contents/1123-telephonie-mobile-3g-et-4g-expliquees](http://www.commentcamarche.net/contents/1123-telephonie-mobile-3g-et-4g-expliquees)
- [w13]. [www.gurau-audibert.hd.free.fr/josdblog/2012/05/les-differents-reseaux-mobiles](http://www.gurau-audibert.hd.free.fr/josdblog/2012/05/les-differents-reseaux-mobiles)
- [w14]. [www.google.dz/search?q=les+r%C3%A9seaux+mobiles&rlz=1C1CHBD\\_enDZ740DZ740&oq=les+r%C3%A9seaux+mobiles](http://www.google.dz/search?q=les+r%C3%A9seaux+mobiles&rlz=1C1CHBD_enDZ740DZ740&oq=les+r%C3%A9seaux+mobiles)
- [w15]. [www.easyfleet.net/front/index.php/lexique.html](http://www.easyfleet.net/front/index.php/lexique.html)
- [w16]. [fr.organilog.com/454-fonctionnement-geolocalisation-mobile](http://fr.organilog.com/454-fonctionnement-geolocalisation-mobile)
- [w17]. [www.memoireonline.com/03/12/5461/m\\_Interconnexion-entre-deux-reseaux-cellulaires-des-normes-GSM-par-faisceau-hertziens-cas-de-CCT-et0.html](http://www.memoireonline.com/03/12/5461/m_Interconnexion-entre-deux-reseaux-cellulaires-des-normes-GSM-par-faisceau-hertziens-cas-de-CCT-et0.html)
- [w18]. [www.socialcompare.com/fr/comparison/android-versions-comparison](http://www.socialcompare.com/fr/comparison/android-versions-comparison)
- [w19]. [mathias-seguy.developpez.com/tutoriels/android/installation-eclipse-environnement-travail](http://mathias-seguy.developpez.com/tutoriels/android/installation-eclipse-environnement-travail)
- [w20]. [developer.android.com/about/dashboards/index.html](http://developer.android.com/about/dashboards/index.html)
- [w21]. [mathias-seguy.developpez.com/tutoriels/android/premiers-pas](http://mathias-seguy.developpez.com/tutoriels/android/premiers-pas)
- [w22]. [mathias-seguy.developpez.com/tutoriels/android/comprendre-cyclevie-activite](http://mathias-seguy.developpez.com/tutoriels/android/comprendre-cyclevie-activite)
- [w23]. [sabricole.developpez.com/uml/tutoriel/unifiedProcess](http://sabricole.developpez.com/uml/tutoriel/unifiedProcess)
- [w24]. [www.gantt.com/fr/](http://www.gantt.com/fr/)
- [w25]. [www.iphonedveloppeur.fr/20/04/fonctionnement-dune-application-iphone](http://www.iphonedveloppeur.fr/20/04/fonctionnement-dune-application-iphone)
- [w26]. [supinfo.com/articles/single/1641-presentation-android-studio](http://supinfo.com/articles/single/1641-presentation-android-studio)
- [w27]. [android-sdk.fr.uptodown.com/windows](http://android-sdk.fr.uptodown.com/windows)
- [w28]. [www.wampserver.com/2011/11/11/presentation-wampserver](http://www.wampserver.com/2011/11/11/presentation-wampserver)
- [w29]. [google-latlong.blogspot.de/2012/12/google-maps-you-learn-we-listen.html](http://google-latlong.blogspot.de/2012/12/google-maps-you-learn-we-listen.html)
-

---

# RÉSUMÉ

L'évolution de la technologie du mobile a remarquablement étendu les moyens d'accès à l'information : elle est devenue exploitable en tout lieu couvert par un réseau. En tirant profit de cette évolution, les entreprises ont souvent tendance à étendre leur gestion sur des terminaux mobiles. Dans ce but, l'entreprise portuaire de Bejaia voulait se doter de cette pratique qui se révèle très efficace quant à l'accès aux informations indépendamment du temps et de lieu. Notre travail consiste donc à concevoir et implémenter une application Android dédiée à la consultation des états des navires au port de Bejaia, cette application devra reprendre quelques fonctionnalités disponibles sur le site web de la firme.

## Mots clés

Application mobile, Android, UP, UML, service web, Java, XML, MVC, HTTP, JSON, PHP, SQL.

---

# ABSTRACT

The mobile technology evolution has remarkably extended the information access faculties: it became exploitable in every place that is covered by network. Taking advantage from this evolution, companies have usually a tendency to extend their management on mobile terminals. In this purpose, the Bejaia port company wanted to take in this practice which reveals to be very effective in the means of accessing information independently from time and space. Our work is then to design and implement an Android application that will be intended to the checking of ship's situations on the port of Bejaia; this application will have to retake some functions available on the company's website.

## Keywords

Mobile application, Android, UP, UML, web service, Java, XML, MVC, HTTP, JSON, PHP, SQL.

---