REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU
• X •ΘΛ•ΣX [://:V •X[•Λ[•O
FACULTE DU GENIE ELECTRIQUE
ET INFORMATIQUE

Mémoire de fin d'études

En vue de l'obtention du diplôme de Master en informatique

« Option : Réseaux, Mobilité et système Embarqué »

Thème

Réalisation d'une application mobile d'enrichissement de contenu HTML5 sur la plateforme Pixappy

Réalisé et présenté par :

Mr: ***KESRAOUI SMAIL***
Mr: ***LYAZIDI REZKI***

Dirigé par :

Mr: DAOUI MEHAMMED

2014/2015

Remerciements

C'est avec un grand plaisir que nous réservons ces quelques lignes en signe de gratitude et de profonde reconnaissance à tous ceux qui, de près ou de loin, ont contribué à la réalisation et l'aboutissement de ce travail.

Nous tenons tout d'abord à remercier notre promoteur Monsieur DAOUI MEHAMMED pour son soutient, son sérieux, sa gentillesse et surtout pour son aide précieuse tout au long de l'élaboration de ce travail.

Nos vifs remerciements s'adressent à tous les membres de l'équipe Symphotech Solution pour leur encadrement, assistance, soutien, leur disponibilité et leurs précieux conseils durant la période du stage(particulièrement Massinissa Mansouri).

Nous tenons aussi à exprimer l'honneur qui nous est fait par les membres du jury, en acceptant de juger notre travail.

Nous nous acquittons, enfin, volontiers d'un devoir de gratitude et de remerciements à tous les enseignants du département informatique pour la qualité de l'enseignement qu'ils ont bien voulu nous prodiguer durant nos études afin de nous fournir une formation efficiente.

Dédicaces

Je dédie notre travail à ma petite famille :

Mes parents et mes frères : (Mourad et Nadir).

A mes petites cousines Lyna et Anya ainsi que leur parents.

A tous mes amis dont (Hammid, Nabil , Ghiles, Oussama, Massi , Fatma , Cylia et Tiziri).

A toutes les personnes qui ont participé de près ou de loin à l'aboutissement de ce travail.

Et surtout à mon binôme Rezki sans qui ce mémoire n'aurait pas pu être effectué.

Smail

Dédicaces

Je dédie notre travail à toute ma famille :

Surtout mes parents et mes frères.

A tous mes amis de la faculté et d'ailleurs.

A toutes les personnes qui ont participé de près ou de loin à l'aboutissement de ce travail.

Et surtout à mon binôme Smail sans qui ce mémoire n'aurait pas pu être effectué.

Mille pardon pour l'ingratitude des mots et mille excuses pour tous les oubliés.

Rezki

SOMMAIRE

Chapitre 3 : Analyse et conception

Partie1 : Analyse

Partie2 : Conception

Chapitre 4 : Implémentation

Conclusion Générale

Liste des figures

CHAPITRE I

CHAPITRE II

CHAPITRE III

CHAPITRE IV

Liste des tableaux

CHAPITRE I :

CHAPITRE III

CHAPITRE IV

Introduction Générale

Les progrès conjoints de la microélectronique, des technologies de transmission sans fil et des applications embarquées ont permis de produire à coût raisonnable des terminaux mobiles de haute technologie comme les Smartphones et les tablettes PC.

Dans ce contexte, un grand besoin de logiciels pour Smartphone se manifeste de plus en plus. Les utilisateurs de ces systèmes voudront bien les exploiter au maximum de leurs performances à travers les applications qu'ils téléchargent sur leurs équipements.

Dans le cadre de notre stage, il nous a été demandé de faire la conception, le développement et L'intégration d'une application embarquée Android qui permet aux internautes d'enrichir du contenu web à partir de leur Smartphone. Cette application sera connectée à la plateforme de création de contenu rich media interactif *www.pixappy.com.*

Mis à part le développement proprement dit de l'application, la première étape consistait à nous familiariser avec l'environnement Android, puis de choisir les outils conviviaux et envisageables à l'aboutissement du projet. Par la suite, nous entamerons la modélisation et le développement de l'application.

Ce rapport peut ainsi être subdivisé en quatre chapitres :

Le premier chapitre « ***Présentation du projet*** » est consacré à la présentation de notre projet, des objectifs principaux. Nous présenterons la société *Symphotech.qui est notre organisme d'accueil de notre stage.*

Dans le second chapitre « ***Les différents systèmes d'exploitations mobiles*** »nous allons présenter les systèmes d'exploitation conçus pour fonctionner sur un appareil mobile.

Dans le troisième chapitre***« Analyse et Conception »*** nous élaborons une conception détaillée des cas d'utilisation, les diagrammes de séquence, ainsi que le diagramme de classe complet.

Enfin dans le dernier chapitre nous commencerons d'abord par présenter l'environnement matériel et logiciel, puis, l'état de réalisation.

Présentation et Objectif

I- Introduction :

Dans ce chapitre, nous commençons par présenter l'organisme d'accueil où le stage s'est déroulé : au sein de l'équipe Web de la société **Symphotech Solutions** à Draa-Ben-Khedda Tizi-Ouzou. Ensuite nous détaillerons la description du sujet, les objectifs ainsi que le travail demandé.

II- Présentation de l'organisme d'accueil :

1. Présentation générale :

Symphotech est une boite de développement informatique, société de services et d'ingénierie informatique qui propose des solutions informatiques et web innovantes répondant aux besoins de tout type d'entreprise. Elle est aussi spécialisée dans la création de sites internet et la conception de logiciels spécifiques. Symphotech Solutions SARL a créé la plateforme *Pixappy* pour le compte de la société « **Lyxeo SAS** ».

La société Lyxeo SAS est une société spécialisée dans les solutions Web ayant comme domaine d'expertise :

- les Sites E-commerce (pour commercialiser des produits en ligne),
- les applications de paiement (liée à la détention d'un compte bancaire),
- le paiement Paypal, Oscommerce (application Web de commerce électronique),
- Prestashop (application open source permettant de créer une boutique en ligne dans le but de réaliser du commerce électronique) … etc.

2. Présentation de la plateforme Pixappy:

Pixappy[1] est une plateforme web en mode graphique qui permet de créer du contenu *rich media* interactif en html5 sans besoin d'écrire du code. Elle permet aussi d'améliorer la façon dont la communication numérique est effectuée et offre une plus grande flexibilité dans la création de projets, le contenu peut ensuite être partagé sur les réseaux sociaux (Facebook, twitter) ou bien envoyé à des amis et contacts.

3. LA STRUCTURE DE LA PLATEFORME « Pixappy » :

Pixappy est aujourd'hui une plateforme web en mode SaaS ***(Software as a Service)*** à laquelle on peut se connecter gratuitement via un compte avec un Login/Password.
La plateforme est adaptée pour les PC et elle est composée de 2 modules distincts :

[1] www.pixppy.com

- Un module backoffice (CRM) qui permet de :
  - Gérer la liste de ses projets,
  - Créer un nouveau projet
  - Supprimer un projet
  - Dupliquer un projet
  - Visualiser son profile
  - Voir sa formule d'abonnement
- Un module Pixappy Designer qui permet de :
  - Visualiser un projet
  - Créer du contenu interactif en combinant des éléments de base de la plateforme à savoir des « Calques », « Widgets » et « Actions »
  - Exporter les contenu vers
    - Une iframe
    - Un répertoire html5
    - Une application Android

4. Eléments de base du *Pixappy Designer :*

Pour utiliser cette plateforme de manière efficace, il est primordial de maîtriser et de savoir utiliser les éléments de base du ***Pixappy Designer***, et qui constituent les briques de base de chaque création à savoir :

- **Les Calques :**

Ce sont, comme dans Photoshop, des zones qui permettent de recevoir des contenus multimédia (textes, photos des fichiers audio et vidéo), possédant des propriétés graphiques intrinsèques comme la taille et la position sur l'espace de travail, la couleur de fond, le contour ainsi que des attributs comme la visibilité, l'effet d'ombre, et la propriété d'overflow.

- **Les Widgets :**

Sont des éléments attachés aux calques et permettent grâce à leurs propriétés d'associer des effets dynamiques aux calques, afin de simplifier l'implémentation des projets. Un widget n'est spécialisé que dans un seul effet (visibilité/invisibilité, déplacement, ajout de texte, ajout d'image, ajout de lien vidéo…)

- **Les Evénements/Actions**

Les actions permettent de générer de l'interactivité entre les éléments du projet.

Un effet interactif est composé de 2 éléments :

1- Un évènement déclencheur qui peut être soit un événement DOM (comme le clic sur une zone de calque ou bien une action Widget).
2- Une action résultante associée à un Widget.

Chaque création est le résultat de l'association de ces trois éléments : Les calques, les Widgets et le couple Evénement/Actions.

5. Exemple d'une création Pixappy

La figure suivante illustre bien la dynamique que peut générer la plateforme pour une animation sur l'événement de noël par exemple. A l'événement clique sur un hotzone (calque) une suite d'actions s'enclenchent sur d'autres hotzones selon une séquence paramétrée. Par exemple le déplacement du père noël le long de l'image. D'autres événements, comme gratter un cadeau produit l'affichage d'une surprise.

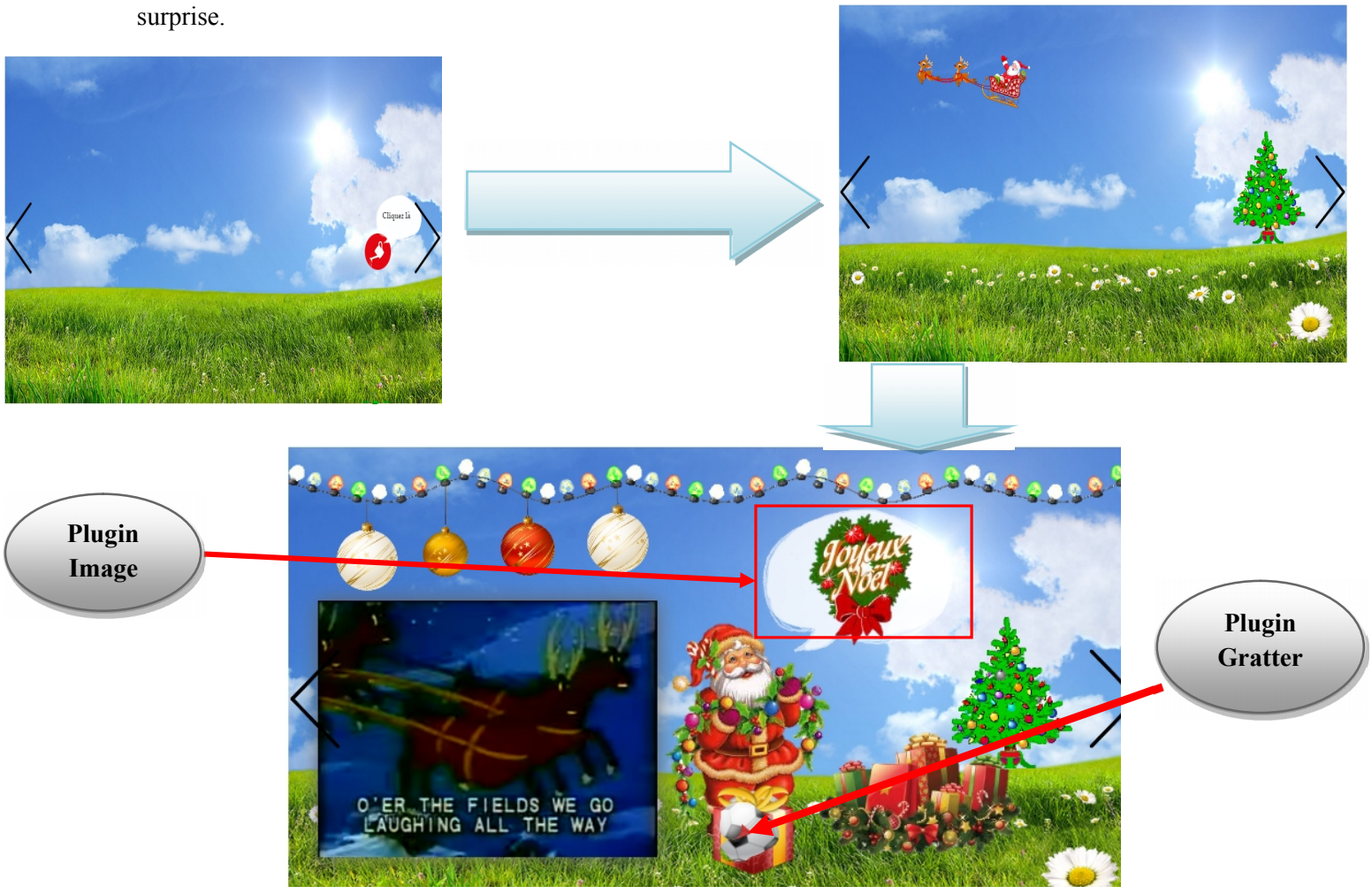


Fig. I.1.Exemple d'un projet pixappy

III. OBJETIF DE NOTRE TRAVAIL:

Aujourd'hui la plateforme Pixappy fonctionne exclusivement en mode web sous html5. Notre objectif est de créer une version mobile sous Androïd de la plateforme ***Pixappy Designer Mobile,*** qui nous permettra de réaliser des contenus *rich media* à l'aide d'un certain nombre d'actions directement à partir de son Smartphone.

L'application mobile pour Smartphones disposera des fonctionnalités qui permettront de :

- Se connecter à son compte sur la plateforme ***Pixappy***
- Consulter son profil (son compte)
- Visualiser les créations publiées sur la plateforme (mode visionneuse)
- Visualiser ses propres projets
- Créer un nouveau projet à l'aide du ***Pixappy Designer Mobile***
- Importer des photos directement de sa galerie photos ou prises directement à partir de son Smartphone

IV. Architecture de notre solution :

Pour que le client Android se connecte à une base de données MySQL, la solution la plus répondue est d'écrire et d'exécuter des scripts PHP à l'aide d'un Protocol http. Ensuite coder les données dans le format JSON, afin de récupérer les données par le client Android.

Le client Android, le serveur web et le serveur de base de données forment une architecture 3tiers. Cette architecture est illustrée dans le chapitre IV.

V. PLAN DE TRAVAIL :

Afin de mener à bien notre travail nous avons adopté le plan suivant :

- Analyser les besoins et rédiger un cahier des charges.
- Identifier la liste des fonctionnalités à implémenter sur la plateforme ***Pixappy Designer Mobile.***
- Définir l'ergonomie de l'application pour assurer une navigation simplifiée.
- Etablir un planning prévisionnel du projet avec les différentes phases.
- Concevoir l'architecture de l'application Android.
- Développer la solution technique.
- Intégrer le ***Pixappy Designer Mobile*** à la plateforme ***Pixappy Manager Mobile.***
- Rédiger le mémoire de fin de stage

VI. ETAPES DE REALISATION

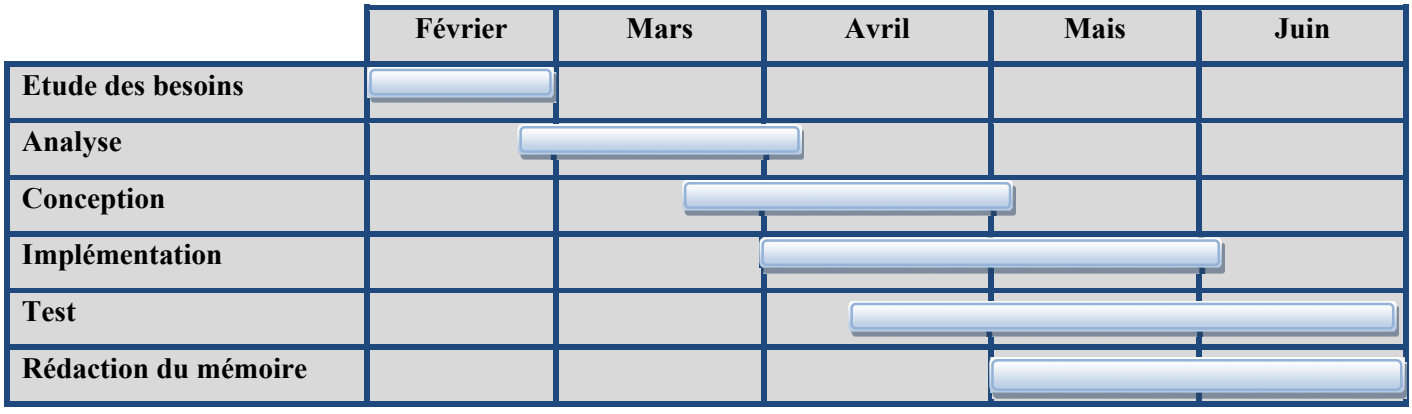
Les étapes de réalisation de notre projet, peuvent être décomposées en 05 phases comme suit :

PHASE	DESCRIPTION	COMMENTAIRE
PHASE 01	Etudes des besoins	La spécification des besoins doit décrire sans ambiguïté le système à développer, L'expression des besoins doit donc proposer ce que le système devra accomplir, non pas comment le faire.
PHASE 02	Analyse	Analyser les différents cas d'utilisation
PHASE 03	Conception	Concevoir le fonctionnement de notre système informatique et le choix du modèle architectural adéquat.
PHASE 04	Implémentation	La programmation et la réalisation en utilisant un environnement logiciel adéquat
PHASE 05	Test	Test du bon fonctionnement de l'application (en vérifiant les résultats souhaités)

Tab. I.1.Phases de réalisation

VII. PLANNING :

Pour finir notre travail dans les délais, nous avons commencé par la mise en place d'un chronogramme comportant la répartition des différentes tâches à réaliser au cours du temps. Le tableau ci-dessous illustre le chronogramme que nous avons suivi tout au long du cycle de développement de l'application.



Tab. I.2.Cycle de développement de l'application

VIII. Conclusion :

Dans ce chapitre, nous avons exposé le contexte du sujet de notre projet dont les étapes de réalisation seront décrites d'une manière détaillée dans les chapitres qui suivent et présenté l'entreprise au sein de laquelle notre stage a été effectué.

Systèmes d'Exploitation Mobiles

I. Introduction :

Un **système d'exploitation mobile** est un système d'exploitation conçu pour fonctionner sur un appareil mobile, Dans cette section, nous aborderons quelques plateformes existantes, Cela nous permettra d'avoir une idée assez générale des systèmes d'exploitation tournant sur mobiles, et d'essayer de faire le positionnement d'Android dans l'environnement des systèmes d'exploitation pour mobile.

II. Système d'exploitation des Smartphones :

Les 4 principaux systèmes sont Android, iOS, Windows phone et Blackberry OS, représentant la quasi-totalité des parts de marché.

1- iPhone OS

iPhone OS d'Apple (icône dans la figure II.1), est le premier OS pour téléphones tactiles qui a véritablement lancé la vague des Smartphones. Apparu sur le marché en 2007, il a innové dans un domaine qui n'était pas encore développé. Cependant, iOS n'a pas beaucoup évolué depuis sa création.iOS se caractérise par une interface peu chargée, qui permet l'exécution de tâches rapidement pour les utilisateurs réguliers, mais qui laisse peu de place à la personnalisation. Dans de nombreux cas on observe le positionnement de mêmes éléments qui diffère d'une application à l'autre. Une étude a montré que les utilisateurs sont fréquemment ralentis par ce manque d'homogénéité, ce qui entraîne un grand nombre d'erreurs.

Fig. II.1.Icone représentant le système iOS

2- BlackBerry :

Tout comme l'iPhone, le BlackBerry (voir figure II. 2) est aussi un téléphone très Utilisé. La fonction majeure qui a fait décoller le BlackBerry était le push mail. L'utilisateur n'a alors plus besoin de consulter périodiquement sa boîte pour vérifier s'il n'a pas de nouveaux messages. Ceux-ci lui parviennent directement comme un simple SMS. Cette fonctionnalité est assurée par les serveurs d'infrastructure du fabricant RIM (Research In Motion) avec un protocole propriétaire. Le mail est donc le point fort des BlackBerry qui a fait son succès auprès des cadres et dirigeants,

Fig. II.2.Exemplaire d'un téléphone utilisant BlackBerry OS

3- Symbian :

Symbian est très populaire (particulièrement en Europe) (voir figure II.3), c'est le leader mondial du système d'exploitation pour Smartphones Il offre une plateforme flexible, ce qui veut dire que les constructeurs de téléphones mobiles peuvent facilement y ajouter leurs technologies et infrastructures. De plus, il est soutenu par les grandes manufactures de l'industrie mobile comme Sony Ericsson, Motorola, et Nokia.Ce dernier étant constructeur numéro un des mobiles dans le monde et principal actionnaire de Symbian, cela garantit la position du Symbian dans le marché des systèmes d'exploitation pour les Smartphones. Néanmoins depuis quelques années, la part de marché de Symbian diminue à cause de concurrence avec d'autres plateformes.

Fig. II.3.Exemplaire d'un téléphone sous OS Symbian

4- Windows Mobile

Windows Phone (voir figure II.4), apparu en 2010, a su s'inspirer de ses concurrents et de son expérience passée dans le domaine du mobile. Proposant une interface simple et épurée ainsi que de multiples possibilités de personnalisation, il permet d'arriver rapidement à l'exécution d'une tâche ce qui a particulièrement séduit les utilisateurs. C'est d'ailleurs Windows Phone qui se trouve être l'OS mobile le plus adapté aux utilisateurs.

Fig. II.4. Icone représentant le système Windows Mobile

5- Android OS

Android est un OS basé sur linux conçu par Google pour les appareils mobiles. il est gratuit et complètement ouvert C'est-à-dire que le code source et les APIs sont ouvertes, il a été développé par l'Open Handset Aliance en 2007 et il est devenu une plateforme ouverte en 2008. Android a intégré plusieurs services de Google pour accéder rapidement aux services d'internet comme Gmail, YouTube, Google Talk, Google Calendar et Google Maps.

III .Comparaison entre les différents Systèmes d'exploitation mobiles :

Dans ce qui suit, nous donnons une comparaison entre les différents systèmes d'exploitation mobiles au niveau de leurs performances en premier lieu, puis de leur popularité et nous examinons les part de marché de chacun.

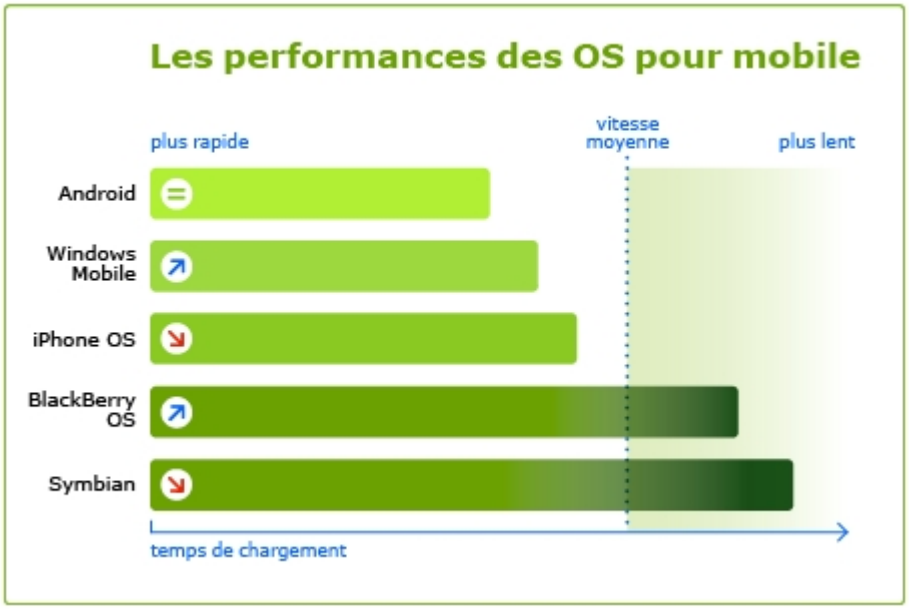


Fig. II.5. Performances des Systèmes d'Exploitation mobiles (Edouard Ricque)[1]

L'histogramme de la figure II.5 montre les différents systèmes d'exploitation mobiles comparés à base de leurs performances par temps de chargement, on remarque que le système Android présente le meilleur taux des performances d'un OS mobile par rapport aux autres systèmes, suivi du

[1] Edouard Ricque

système Windows Mobile. Ces performances expliquent les différentes parts acquises dans le marché du OS mobile. Le meilleur taux de performance réservé pour Android est confirmé par la grande part du marché qu'il réquisitionne illustrée dans l'histogramme de la figure II.6 :

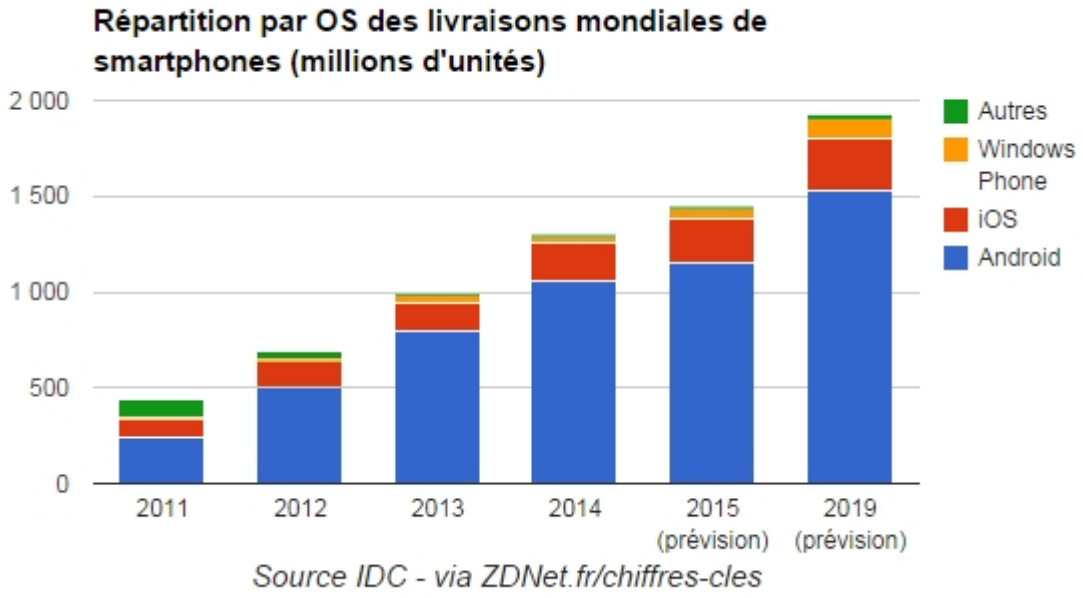


Fig. II.6.Parts de marché des Systèmes d'Exploitation mobile[2]

Selon l'histogramme sur les parts de marché des OS pour mobile, on remarque qu'Android est le plus commercialisé par rapport aux autres OS comme Windows Phonet iOS et depuis 2011 il garde toujours sa première place dans le marché mondial**.**

Vue que nous allons développer notre application sous Android intéressons nous à ce système plus en profondeur.

IV- Présentation de la plateforme Android:[3]

Cette partie présente Android sous un autre angle que celui de l'utilisateur, qui est celui d'une vue de l'intérieur, en explorant les aspects techniques internes.

1- Architecture d'Android :

La plate-forme Android est composée de différentes couches :

[2] Source IDC-via ZDNet.fr/chiffres-cles
[3] Cours Andoid 2014 .Mr M. Daoui

1.1 Application :

Le système Android peut exécuter un grand nombre d'applications incluant les clients de messagerie, les programmes de gestions d'SMS, les gestionnaires de rendez-vous, les navigateurs, …etc. Toutes ces applications sont écrites en Java.

1.2 Framwork Applicatif :

C'est un ensemble de fonctions permettant aux applications d'exploiter au maximum les ressources de l'équipement, de communiquer avec le système et les autres applications (Accès à divers informations, exécution en tâche de fond, afficher des notifications dans la barre d'état, déclencher des alarmes, …etc.). Ces fonctions sont les mêmes que celles utilisées par le système lui-même. Ce qui donne aux applications toute la puissance nécessaire.

1.3 Les Librairies :

C'est un ensemble de bibliothèques développés en C/C++. Ces librairies sont utilisées par divers composant d'Android dont le Framework applicatif.

Les composants majeurs de la plate-forme Android sont résumés sur le schéma suivant :

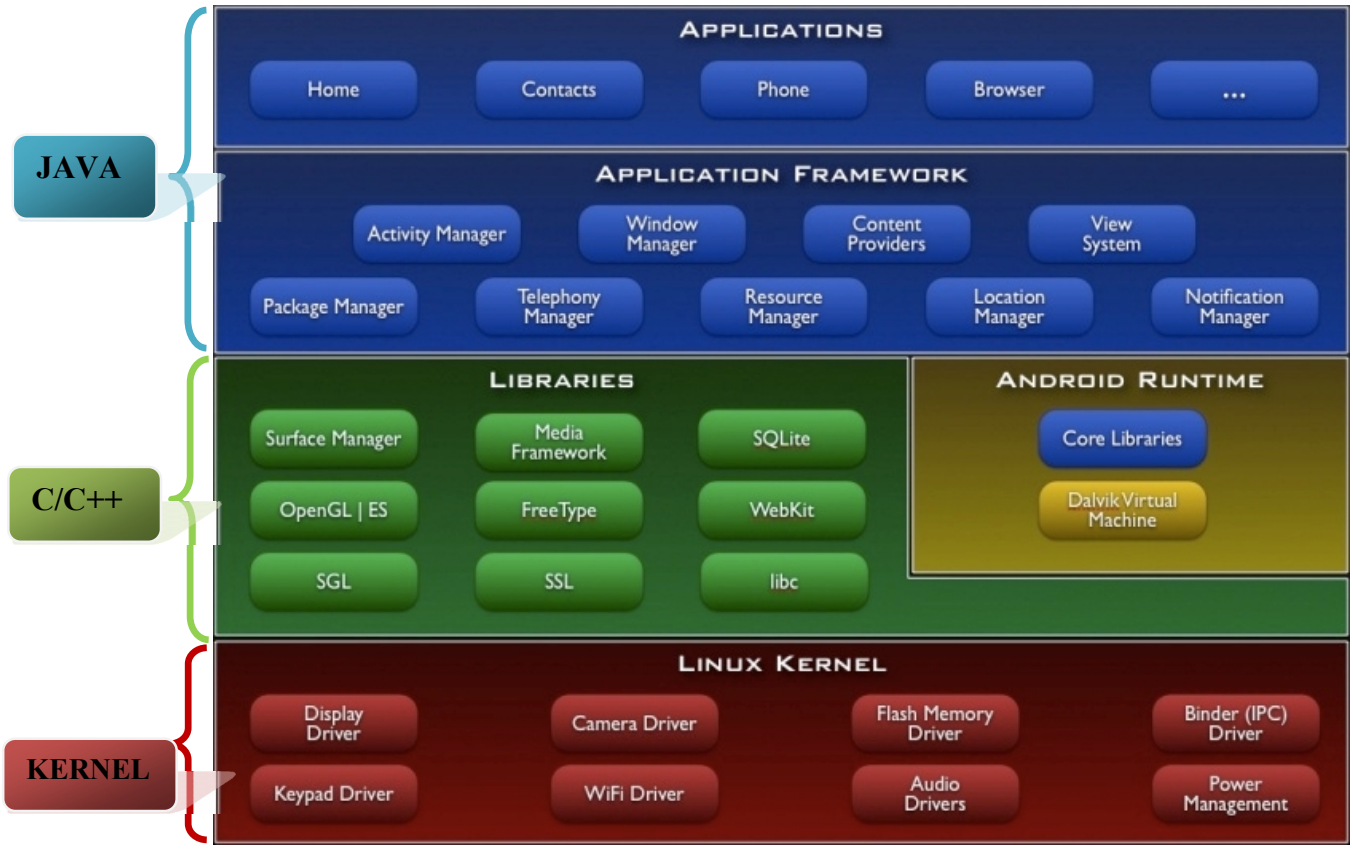


Fig. II.7. Architecture d'un système Android

1.4 Environnement d'exécution Android :

L'environnement d'exécution d'Android est la machine virtuelle ***Dalvik*** avec ses propres bibliothèques adaptées aux équipements mobiles, ce qui lui procure un défaut de calibrage sur des systèmes avec des ressources différentes. Dans la machine virtuelle Dalvik, les programmes sont écrits en Java et compilés avec les outils de java pour obtenir un ***byte code*** qu'on recompile encore avec un outil spécifique (dex) pour obtenir un code adapté à la machine Dalvic

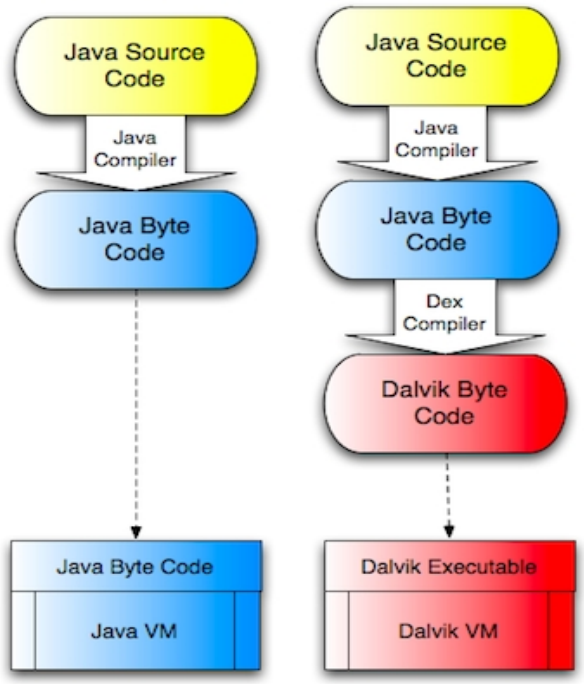


Fig. II.8.Différence entre la JVM et Dalvik

1.5 Le noyau linux :

Android repose sur le noyau Linux version 2.6 pour la gestion de la mémoire, des processus, et les pilotes des différents périphériques.

2- Les concepts d'une application Android :

Une application android est constituée d'un certain nombre de composants (Activity, service, Content Provider et Broadcast Receiver) :

2.1. L'activité (Activity) :

Une activité est un composant fournissant un écran avec lequel les utilisateurs interagissent avec l'application. Quand on exécute n'importe quelle application Android on remarque qu'une première fenêtre s'offre à nous avec laquelle il est possible d'interagir (introduire ou extraire des données), elle possède donc une interface graphique. Dans certains cas, en appuyant sur un bouton par exemple ou un TextView cela nous mène vers une autre activité, et ainsi de suite. Cela est possible

grâce au mécanisme d'Intent. Une application donc est un assemblage d'activités entre lesquelles il est possible de naviguer.

Une activité remplit tout l'écran, une application ne peut en afficher qu'une seule à la fois. De plus, elle contient des informations sur l'état actuel de l'application. Ces informations s'appellent le Context. Toute activité peut passer à tout moment en arrière plan dans une pile d'attente lorsque son exécution est interrompu comme lors d'un appel entrant.

2.1.1. Etat d'une activité :

Une activité peut se trouver dans trois états qui se différencient surtout par leur visibilité :

- *Active (Resumed) :* L'activité est visible en totalité. Elle est sur le dessus de la pile, c'est elle qui a le *focus*. C'est ce que l'utilisateur consulte et peut l'utiliser dans son intégralité et agir directement dessus.

- *Suspendue (Posed) :* L'activité est partiellement visible à l'écran. C'est le cas lors de la réception d'un SMS et qu'une fenêtre semi-transparente se pose devant l'activité pour afficher le contenu du message. Ce n'est pas sur l'activité suspendue qu'agit l'utilisateur. L'application n'a plus le focus, c'est l'application sus-jacente qui l'a. Pour que notre application récupère le focus, l'utilisateur devra se débarrasser de l'application qui l'obstrue, puis il pourra à nouveau interagir avec elle.

- *Arrêtée (Stopped) :* L'activité est tout simplement masquée par une autre activité, on ne peut plus la voir. L'application n'a évidemment plus le focus, et on ne peut pas agir dessus. Le système retient son état pour pouvoir reprendre, mais il peut arriver que le système tue l'application pour libérer de la mémoire système.

Les transitions d'états d'une activité sont captées par les méthodes suivantes :

onCreate () : est appelée au début de la création de l'activité et n'est appelé qu'une seule fois. Elle joue le rôle du constructeur en permettant d'initialiser des variables, affecter des listener...

onRestart () : appelée après un nouveau démarrage de la même activité (quand l'activité était arrêtée).

onStart () : l'activité va devenir visible. Cette méthode sert à lancer les animations, ou généralement tout ce qui est liée à l'affichage graphique, car elle est également appelée lors d'un retour de focus sur l'activité (dans ce cas onRestart est appelé avant).

onResume (): l'activité est maintenant visible, Cette méthode sera exécutée lorsque l'activité résume son exécution après la suspension (pause) et que l'activité commence à interagir avec l'utilisateur.

onPause () : méthode qui sert à arrêter une activité temporairement.

onStop () : l'activité ne sera plus visible, cachée par une autre activité qui est en premier plan. Une activité stoppée est aussi en vie, elle est encore en mémoire mais elle n'est pas rattachée au gestionnaire des fenêtres du système Android. Elle peut être tuée par le système Android en cas de besoin en mémoires.

onDestroy() : l'activité va être détruite. La destruction opère quand quelqu'un appelle cette méthode ou quand c'est le système qui décide de tuer l'activité pour économiser de l'espace.

2.1.2 Cycle de vie d'une activité :

Une activité n'a pas de contrôle direct sur son propre état, il s'agit plutôt d'un cycle rythmé par les interactions avec le système et d'autres applications. Voici un schéma qui représente ce que l'on appelle **le cycle de vie d'une activité**, c'est-à-dire qu'il indique les étapes que va traverser notre activité pendant sa vie, de sa naissance à sa mort.

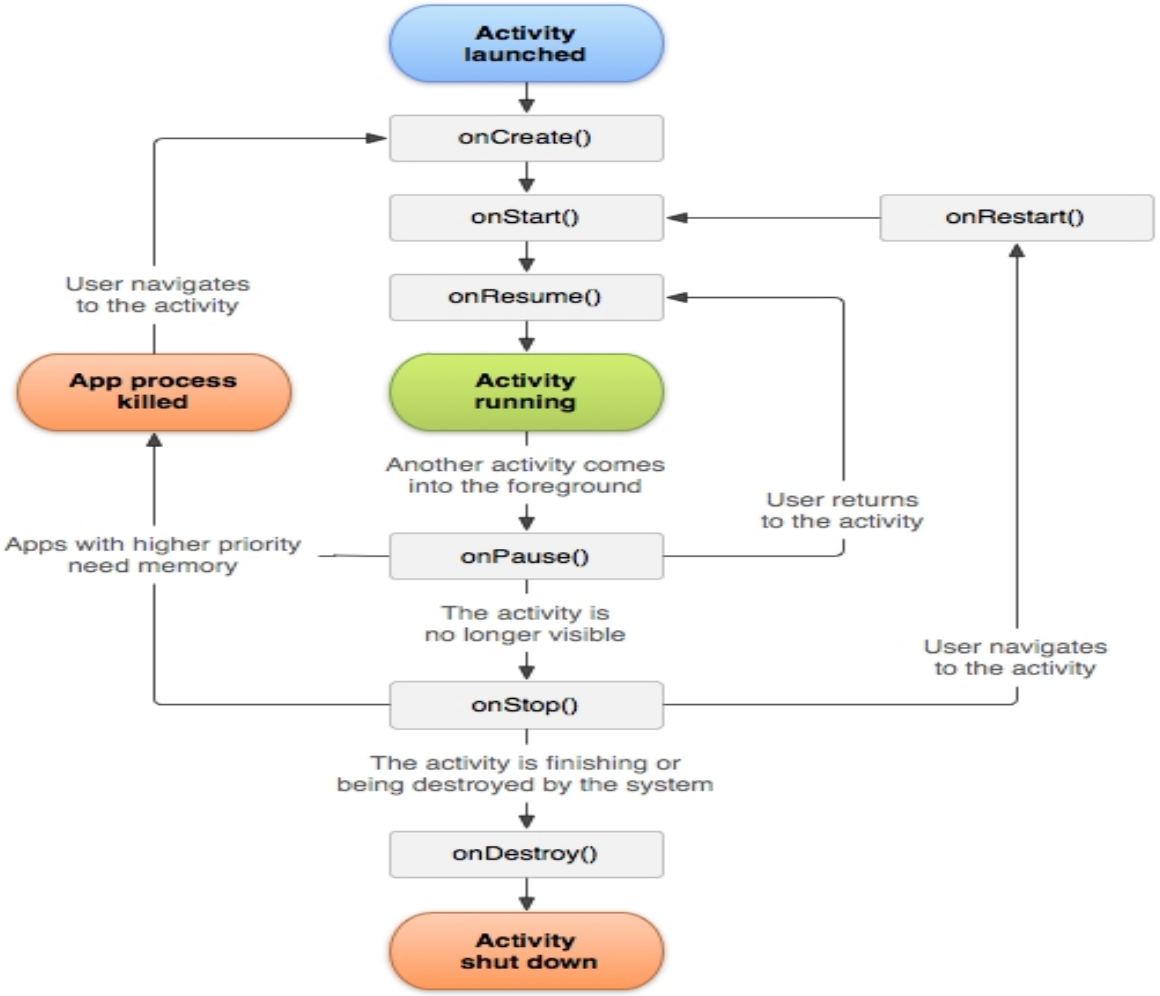


Fig. II.9.Cycle de vie d'une activité

2.2 Les Services :

Les services sont des tâches qui peuvent être lancées avec ou sans intervention de l'utilisateur. Elles s'exécutent dans le background de l'application et peuvent se terminer soit après la finalisation de la tâche, soit à travers une intervention externe. Les services représentent également une fonctionnalité d'une application exposée à d'autres applications. Il est important de mentionner que le service ne fournit pas d'interface graphique (User Interface).

Notre Player audio (lecteur de musique), par exemple, permet d'écouter la musique tout en consultant nos emails, etc.… Cette fonctionnalité n'est possible qu'à l'aide des Services.

2.3 Content Provider :

Les fournisseurs de contenu offrent un moyen de partager des données entre les applications qui peuvent être stockées sur le système de fichier local, sur une base de données SQLite ou le réseau (FTP, Web, etc.). Les autres applications peuvent accéder à ces données (ou les modifier si l'application le permet) en utilisant un content provider.

Un content provider est une classe qui implémente un ensemble de méthodes standards permettant à d'autres applications de consulter et modifier les données prises en charge par ce content provider .Par exemple, le système Android fournit un content provider fournissant la liste des contacts téléphoniques qu'une autre application peut utiliser .

2.4 Broadcast Receiver :

Les broadcaster sont les diffuseurs d'évènements/messages via des intentions (intentions dites broadcast pour les différentier des intentions plus classiques). Les messages ainsi diffusés pourront être réceptionnés par plusieurs applications, les applications qui se seront abonnées à ces broadcasts (diffusions).

2.5. Intent :

Trois composants d'application sont lancés par les Intents : Les activités, Les services, Les receveurs de diffusion. Les Intents sont l'une des pierres angulaires de la plateforme Android. Nous pouvons les comparer à des actions ou même à des intentions, ils permettent de dialoguer à travers le système à partir de canaux qui leurs sont dédiés. Quand le mobile reçoit un appel, la plateforme lance un Intent signalant l'arrivée d'un appel, de même pour un SMS. Nous les utiliserons dans notre application pour dialoguer à l'intérieur de celle-ci pour :

- Naviguer entre les activités.
- Surveiller les clicks sur l'AppWidget
- Lancer le service de téléchargement des données météorologiques

V. Historique d'Android[4]

1. Open Handset Alliance:

Afin de promouvoir ce nouveau système d'exploitation ouvert, Google a su fédérer autour de lui un consortium d'une trentaine d'entreprises. L'Open Handset Alliance (abrégé OHA) est un consortium Regroupant de grands constructeurs et développeurs de logiciels dans le but de développer des normes ouvertes pour les appareils de téléphonie mobile.

Ce consortium a été créé le 5 novembre 2007. Le premier standard annoncé a été Android, une plateforme pour appareils mobiles basée sur un kernel linux 2.6.

2. Bugdroïde :

Pour représenter Android la famille Google a utilisé un personnage nommé Bugdroid(voir la figure 10) qui est un petit rebot vert. Ce personnage est sous licence « creative commons by (3.0) » et peut donc être utilisé librement. Le site Engadget annonce que Bugdroid, le logo d'Android, serait en fait un personnage d'un jeu des années 1990 sur Atari Gauntlet, The Third Encounter.

[4] Mémoire réalisé par **Mlle BOUNOUAR Sonia et Mlle DAHMANI Kahina en 2013 page 9**

Fig. II.10.Personnage Bugdroïde

3. Historique des versions d'Android :

Android a débuté avec la sortie de la version 1.0 en septembre 2008. Android a connu plusieurs mises à jour depuis sa première version. Ces mises à jour servent généralement à corriger des bugs et à ajouter de nouvelles fonctionnalités. Dans l'ensemble, chaque version est développée sous un nom de code basé sur des desserts. Ces nom de codes suivent une logique alphabétique, en voici quelques-unes :

- **La version 2.3 (2.3.7)** :

  Gingerbread (Pain d'épice), sortie le 6 décembre 2010, dernière version dédiée uniquement aux Smartphones. Cette version est parfois utilisée sur de petites tablettes.

  Caractéristiques :

- Interface utilisateur mise à jour
- Support des grands écrans à résolutions extra-larges (WXGA et plus)
- Support de la VoIP et SIP
- Support des formats vidéo WebM/VP8, et l'encodage audio AAC
- Nouveaux effets audio tels que la réverbération, l'égalisation, la virtualisation du casque audio et accentuation des graves

- **La version *4.0.3*** :

  (Aussi appelée Ice Cream Sandwich) unifie le développement des interfaces Smartphone, tablette, télévision connectée et système embarqué. Avant la version 4.0, le

développement d'une application pour une tablette n'était pas compatible à un Smartphone à cause de la taille d'écran différente.

Il offre désormais la possibilité d'avoir les touches : retour, maison et applications récentes sur l'écran, éliminant ainsi le besoins de touches physiques.

Caractéristiques :

- Nouvelles APIs pour les développeurs (notification des mises à jour des contacts par les fournisseurs de services.
- Amélioration de l'accessibilité pour les lecteurs d'écrans
- Diverses modifications de l'interface de l'appareil photo
- Amélioration du calendrier (API diffuseurs de services)

- **La version 4.2**:

(Aussi appelée Jelly Bean) introduit photo sphère permettant une prise des photos à 360° type Street View, un système multi-compte sur tablette uniquement.

Caractéristiques :

- Corrige le bug qui ne faisait pas apparaître le mois de décembre dans la gestion des contacts
- Ajout du support pour les gamepads et joysticks Bluetooth

- **La version 4.4** : **(Aussi appelée Android KitKat**) :

C'est le nom d'une version du système d'exploitation Android qui a comme numéro 4.4. Son nom vient de la « tradition » de nommer les différentes versions d'Android par un nom de dessert ou pâtisserie et d'un partenariat avec Nestlé qui possède la marque de biscuits chocolatés Kit Kat. Pendant longtemps il avait été pensé que la version d'Android devant succéder à la 4.3 serait la 5.0 Key Lime Pie

Caractéristiques :

- Nouvelle interface translucide
- Enregistrement séquence vidéo de l'écran
- Amélioration du système de notification
- Gestion système des sous-titres
- Amélioration des performanc

- **La version 5.0** :

(Aussi appelée Lollipop), Son numéro de version et son nom de code ont été annoncés le 15 octobre 2014, pour une disponibilité publique le 3 novembre 2014. Les changements les

plus importants d'Android 5.0 sont sa disponibilité sur les nouvelles plateformes Android TV et Android Auto, ainsi que l'amélioration de l'autonomie de la batterie via le projet *Volta.*

Caractéristiques :

- Nouvelle interface / design ("Material design")
- Amélioration de la rapidité
- Amélioration de la gestion de la batterie

Version	Codename	API	Distribution
2.2	Froyo	8	0.4%
2.3.3 - 2.3.7	Gingerbread	10	7.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	6.4%
4.1.x	Jelly Bean	16	18.4%
4.2.x		17	19.8%
4.3		18	6.3%
4.4	KitKat	19	39.7%
5.0	Lollipop	21	1.6%

Data collected during a 7-day period ending on February 2, 2015.
Any versions with less than 0.1% distribution are not shown.

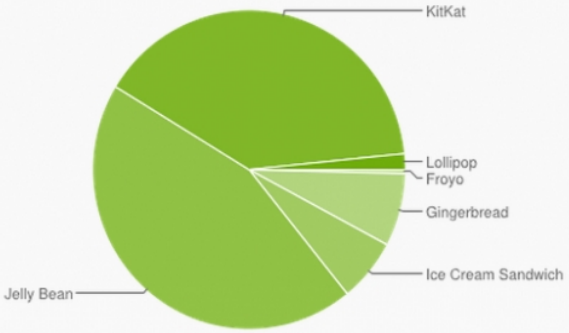


Fig. II.11. Répartition des versions d'Android

On remarque que la version KitKat (4.4.x) est la plus populaire avec plus de 39%.

VI- Différents équipements utilisant Android :

Le succès d'Android est indéniable, les fournisseurs l'ont bien compris, d'où l'apparition sur le marché de plus en plus d'appareils utilisant ce système d'exploitation :

1- Les Smartphones :

Est un téléphone intelligent, disposant d'un grand écran tactile, fonctionnant sur des réseaux haut débit et offrant de nombreuses fonctionnalités, dont la consultation du courrier électronique et la localisation par GPS. Le premier mobile commercialisé sous Android est le HTC Dream/G1 produit par la firme Taïwanaise HTC, lancé aux États-Unis le 22 octobre 2008.

Depuis le lancement en 2007 de l'iPhone d'Apple, la progression de leurs ventes est spectaculaire et elles ont dépassé en 2012 celles des téléphones mobiles classiques.

2- Les Tablettes :

En septembre 2010, plusieurs marques comme Samsung ont utilisé le système Android pour les tablettes (version 2.2 Fro Yo)

Android a été porté sur d'autres appareils comme la HP TouchPad, le portage a été réalisé début 2012 asé sur la version Ice Cream Sandwich, et fin 2011 avec la version 7 de CyanogenMod basé sur Gingerbread.

3- Les télévisions :

Le 5 avril 2010, la première télévision sous Android est dévoilée. Celle-ci est développée par l'entreprise suédoise People of Lava et se nomme Scandinavia. Elle possède les différentes applications connectant aux réseaux sociaux (facebook……), elle possède un navigateur Web ainsi qu'un client de messagerie électronique. (ENIE)

4- Montre :

Google a créé une version spécifique *Android Wear*, une version modifiée d'android dédiée au Smartphones, LG commercialise sa G Watch et Samsung une Gear Live, Motorola a aussi dévoilé sa montre sous le système Android.

5- Consoles de jeux vidéo :

Une console de jeux vidéo portable sous Android portant le nom GamePad est commercialisé depuis septembre 2012.

VII. Conclusion :

Ce chapitre nous a permis d'aborder les différents systèmes d'exploitation conçus pour fonctionner sur les appareils mobiles. Ainsi on a étudié le positionnement d'Android par rapport aux autres systèmes en spécifiant ses notions de bases, ses caractéristiques et ses possibilités d'utilisation.

Dans le chapitre suivant, nous présenterons une analyse détaillée pour les cas d'utilisation de notre système.

Analyse et Conception

Partie1 : Analyse

I. Introduction

L'analyse et la spécification des besoins représentent la première phase du cycle de développement d'un logiciel. Elle sert à identifier les acteurs réactifs du système et leurs associer chacun, l'ensemble d'actions avec lesquelles il intervient dans l'objectif de donner un résultat optimal et satisfaisant au client.

Ainsi, dans la premier partie de ce chapitre, nous commencerons en premier lieu par une spécification des besoins auxquels doit répondre l'application, passant ensuite à l'analyse de ces besoins à travers l'introduction des acteurs et les diagrammes de cas d'utilisation relatifs à ces acteurs.

II. Langage de modélisation :

Pour la modélisation objet, nous avons choisi le langage commun UML «Unified Modeling Language». En effet, UML est un langage de modélisation formel et normalisé, né de la fusion de plusieurs méthodes existantes. Il permet de modéliser informatiquement un ensemble d'éléments d'une partie du monde réel en un ensemble d'entités informatiques. Ces entités informatiques sont appelées objets. Ces objets sont décrit par des vues statiques et dynamiques, incluant un ensemble de diagrammes, qui collaborent pour représenter diverses projections d'une même représentation d'un système d'objets.

III. Spécification des besoins :

III.1. Les besoins fonctionnels :

Les services proposés par notre application se résument en quatre actions majeures :

- Se connecter à son compte sur la plateforme Pixappy
- Visualiser les créations publiées sur la plateforme
- Créer, gérer et visualiser les projets personnels
- Importer des photos directement de sa galerie

III.2.Les besoins non fonctionnels :

Les besoins non fonctionnels décrivent toutes les contraintes auxquelles est soumis le système pour sa réalisation et son bon fonctionnement :

Ergonomie et souplesse : L'application doit offrir une interface conviviale et ergonomique exploitable par l'utilisateur en envisageant toutes les interactions possibles à l'écran du support utilisé.

Rapidité : L'application doit optimiser les traitements pour avoir un temps de génération de schéma raisonnable.

Efficacité : L'application doit être fonctionnelle indépendamment de toutes circonstances pouvant entourer l'utilisateur.

Maintenabilité et scalabilité : Le code de l'application doit être lisible et compréhensible afin d'assurer son état évolutif et extensible par rapport aux besoins du marché.

IV. Analyse des besoins :

IV.1. Identification des acteurs :

Comme nous venons de le mentionner dans ce qui précède, «pixappy» est une application personnalisée et destinée à des utilisateurs qui vont être professionnels ou bien des amateurs possédant un compte.

Dans tous les cas, il y aura trois acteurs qui vont interagir avec l'application, et qui vont présenter leurs fonctionnalités et besoins que l'application doit satisfaire.

Un acteur est une entité externe (humain, logiciel, ou automate) qui interagit avec le système mais n'appartient pas à ce dernier. Il fournit de l'information en entrée et on en reçoit en sortie.

Les acteurs de notre système sont :

1- ***User :*** est un visiteur non inscrit.
2- ***Member :*** toute personne possédant un compte.
3- ***Le serveur :*** l'ordinateur hébergeant la plateforme pixappy

IV. 2.Spécification des tâches :

Les acteurs définis précédemment effectuent un certain nombre de tâches résumées dans le tableau ci-dessous :

Acteur	Tâches
- **User (non inscrit)**	**T0** : Lancer l'application **T1** : Visualiser les créations publiées sur la plateforme (mode visionneuse). **T2** : Rechercher des publications sur la plateforme.
- **Membre (possèdent un compte)**	**T0, T1** **T3** : Se connecter à son compte. **T4** : Créer un nouveau projet à l'aide du « Pixappy Designer Mobile » **T5** : Gérer ses projets (Modifier, Supprimer…) **T6** : Editer des calques pour pouvoir y travailler **T7** : Paramétrer un calque **T8** : Supprimer un calque **T9** : Ajouter une action à un Widget **T10** : Importer des photos directement de sa galerie photos ou prises directement **T11** : Publier un projet sur la plateforme **T12** : Paramétrer un projet **T13** : Visualiser un projet **T14** : Déconnecter

Tab. III.1 : « Spécification des tâches »

IV. 3.Spécification des scénarios :

Un scénario représente une succession particulière d'enchaînement s'exécutant du début à la fin du cas d'utilisation. Un ensemble de scénarios pour un cas d'utilisation représente tous ce qui peut arriver lorsqu'un cas d'utilisation est mis en œuvre.

Acteurs	Tâches	Scénarios
user (non inscrit sur Pixappy.com)	**T1**-Lancer l'application	**S0** : Cliquer sur l'icône pixappy
	T2 : Voire les publications existantes sur la plateforme.	**S1** : Cliquer sur la ligne représentant le projet à visionner.
	T3 : Rechercher des publications sur la plateforme.	**S2** : faire glisser le panel de gauche **S3**: Sélectionner la catégorie désirée (Home,Journey…)
Membre	**T1, T2, T3**	**Idem que visiteur**
	T4 : Se connecter à son compte.	**S0**, **S4** : Cliquer sur l'cône de « Connexion». **S5** : Saisir le Login et Password puis « **ok** »
	T5 : Créer un projet.	**S0, S4, S5** **S**6: Cliquer sur l'cône « Ajouter un projet ». **S7 :** Remplir le formulaire. **S8 :** Cliquer sur le bouton « Envoyer ».
	T6 : Gérer ses projets.	**S0, S4, S5** **S9**: Cliquer sur l'icône «Gérer les projets ». **S10** : Choisir l'opération à effectuer sur un projet en particulier (Modifier, Supprimer,…)
	T7 : Créer des calques pour pouvoir y travailler.	**S0, S4, S5,S9** **S11**: Cliquer sur le bouton « Modify » **S12**: Cliquer sur l'icône «Ajouter »ou glisser/déposer glisser directement un widget et l'introduire sur l'espace de travail.
	T8 : Paramétrer un calque pour obtenir l'effet désiré	**S0, S4, S5,S9,S11** **S13** : Sélectionner un claque désiré **S14**:Faire remonter le panel de paramétrage du bas
	T9 : Supprimer un calque	**S0, S4, S5,S9,S11- S12** **S15 :** Cliquer sur la corbeille de suppression
	T10 : Ajouter une action à un Widget	**S0, S4, S5, S9,S11** **S16** : Sélectionner un Widget et le faire glisser dans un calque **S17** : Cliquer sur l'icône d ajoute Actions **S18** : Sélectionner un événement déclencheur sur le Widget sélectionné et l'action à effectuer par le Widget destination
	T11 : Importer les ressources	**S0, S4, S5,S9,S11,S12** **S19** : Faire glisser le panel de droit **S20**:Cliquer sur « ressources »
	T12 : publier le projet sur la plateforme	**S0, S4, S5,S9,S11,S12,S19,S20** **S21** : Faire glisser le panel de droit **S22**:Cliquer sur « publicher »
	T12 : Paramétrer le projet	**S0, S4, S5,S9,S21** **S23** : Cliquer sur « Settings »
	T13 : Visualiser le rendue avant publication.	**S0,S2,S3,S4,S9,S10** **S24**: Choisir l'opération « Visualiser».
	T14 : Se déconnecter.	**S0,** **S25**Cliquer sue le bouton d'option de l'appareil **S26**: Choisir l'option «Logout ».

Tab. III.2 : « Récapitulatif des scénarios associé à chaque tâche ».

Description textuel de quelques cas d'utilisation :

IV.4.Les cas d'utilisation :

Une représentation d'un ensemble spécifique d'actions qui sont réalisées par le système et qui produit un résultat observable par un acteur en particulier. Elle permet de représenter l'ensemble des besoins et des exigences que notre application doit respecter.

Dans notre cas nous distinguons beaucoup de cas d'utilisation en voici certains :

IV.4.1. Cas d'utilisation s'identifier à l'espace Membre

Use case: s'identifier à l'espace « Membre ».

Scenarios: S0, S4, S5

Acteurs : Membre.

Résumé : cette fonctionnalité permet d'accéder un son propre compte « Membre ».

Description :

1- Lancer l'application.
2- Le système affiche l'activité principale.
3- Sélectionner le lien « Connexion ».
4- Le système affiche un formulaire que le « user » doit remplir.
5- Saisir Login et Password
6- Cliquer sur « OK » pour valider le formulaire.
7- le système affiche l'interface Membre ou un message d'erreur

Fig. III.1. Cas d'utilisation **« s'identifier à l'espace Member».**

IV.4.2. Cas d'utilisation ajouter un projet

Use case : Ajouter un projet.

Scenario: S0, S4, S5, S6, S7, S8.

Auteurs: Membre.

Résumé : Cette fonctionnalité permet aux acteurs (**Membres**) de créer et ajouter des projets dans son espace

Description :

1- Cliquer sur l'icône d'ajout(+)
2- Le système affiche un formulaire à remplir
3- Cliquer sur le bouton « Envoyer »
4- Le système ajoute le projet crée a sa liste des projets

Fig. **III** .2.Cas d'utilisation **« Ajouter un projet »**

IV.4.3. Cas d'utilisation Modifier un projet

Cas d'utilisation «Modifier un projet» :

Scenario: S0, S4, S5, S9, S11, S12.

Rôle : Membre.

Description :

1- Cliquer sur l'icône de gestion de projet
2- Choisir le projet à modifier et cliquer sur le bouton « Modify »
3- Le système affiche l'espace de travail « Workspace»
4- Modifier le projet selon les besoins
5- Cliquer sur l'icône de sauvegarde pour le mettre à jour dans la plateforme

Fig. III 3. Cas d'utilisation **« Modifier un projets ».**

IV.4.4. Cas d'utilisation Déconnexion

Cas d'utilisation «Sortir de l'espace membre » :

Use case: Se déconnecter.

Scenario: S0, S5, S26, S27.

Rôle : Membre.

Description :

1- Cliquer sur le bouton option de l'appareille
2- Choisir l'option « **Logout** ».
3- Le système affiche l'interface visiteur sinon renvoi un message d'erreur en cas de problèmes.

Fig. III.4. Cas d'utilisation **« Déconnexion ».**

IV. 5. Diagramme de cas d'utilisation :

UML définit une notation graphique pour représenter les cas d'utilisations. Cette notation est appelée **diagramme de cas d'utilisation** qui fait intervenir les acteurs ainsi que les relations qui existent entre eux et les cas d'utilisations eux-mêmes.

Les diagrammes de cas d'utilisations décrivent sous forme *d'actions* et de *réactions* le comportement fonctionnel de l'application modélisée.

Il existe trois types de relations standards entre cas d'utilisations qui sont proposés par UML :

- **« include»** le cas d'utilisation incorpore explicitement et de manière obligatoire un autre cas d'utilisation à l'endroit spécifié.
- **« extend »** le cas d'utilisation incorpore implicitement de manière facultative un autre cas d'utilisation à l'endroit spécifié.
- **Généralisation** : les cas d'utilisations descendant héritent des propriétés de leurs parents.

IV.5.1 Diagramme de cas d'utilisation « Membre»

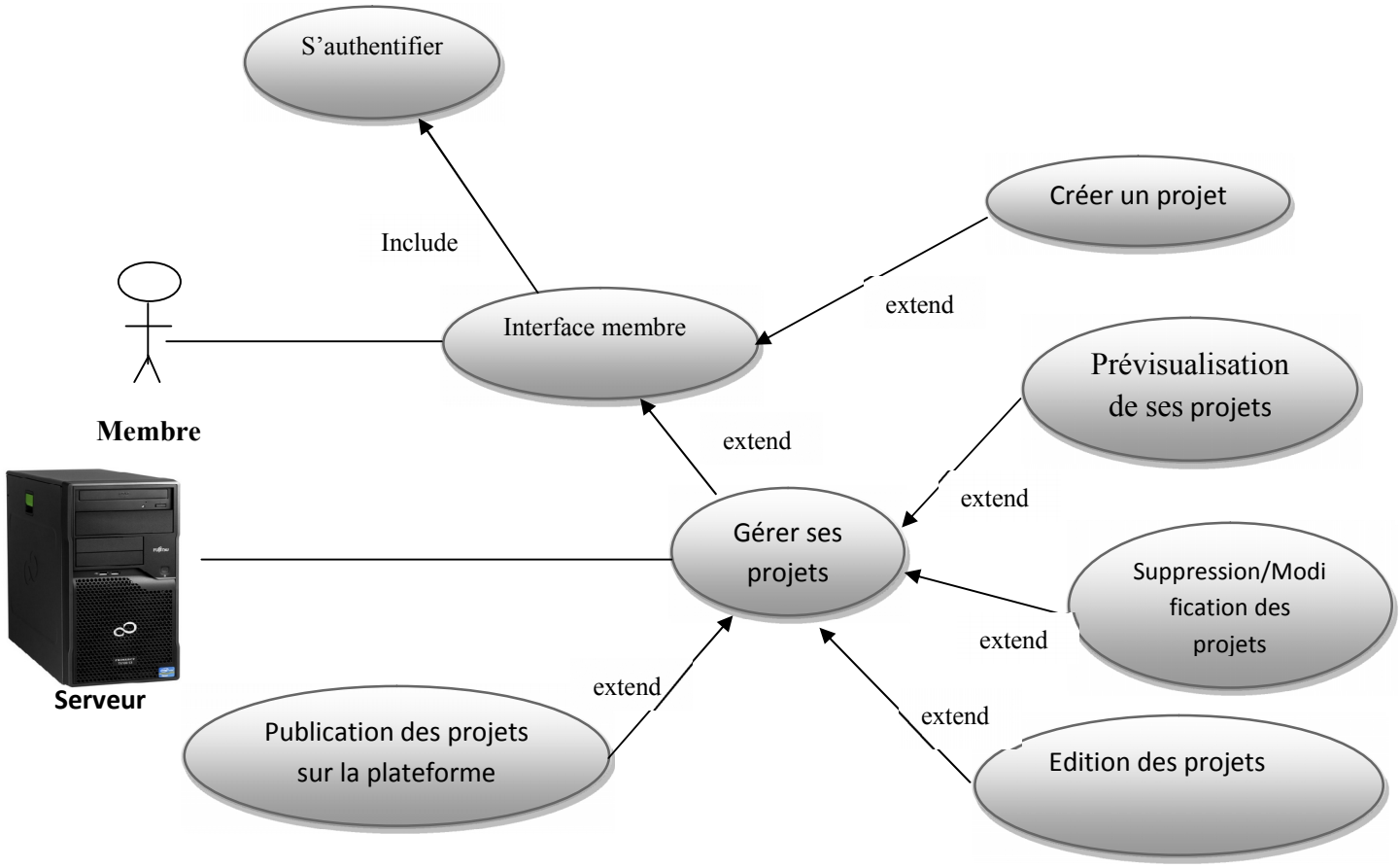


Fig. III .5« Diagramme de cas d'utilisation pour un membre ».

IV.5.2Diagrammes des cas d'utilisation pour le visiteur :

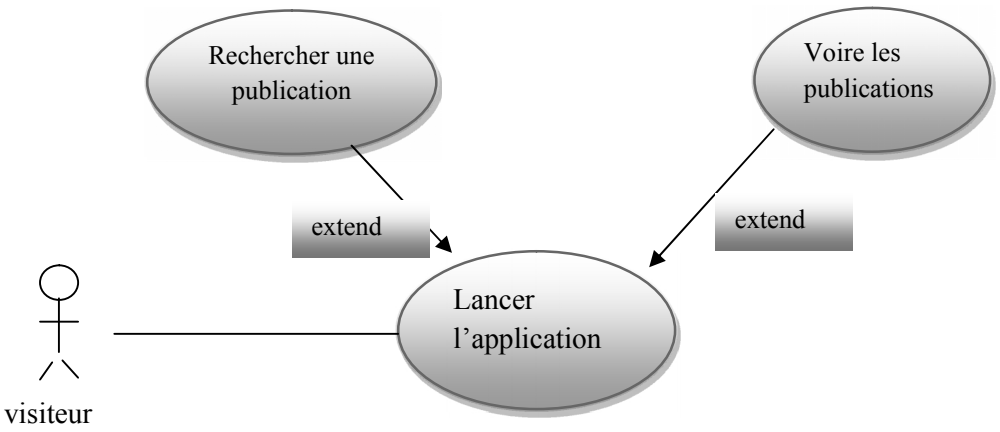


Fig. III.6 « Diagramme de cas d'utilisation pour un visiteur ».

Cette partie nous a permis de couvrir les différents besoins fonctionnels et non fonctionnels des acteurs de notre système.

Nous avons fourni une analyse plus détaillée de ces besoins grâce à un diagramme de cas d'utilisation relatif aux acteurs réagissant avec le système, qui sont les propriétaires du téléphone Android.

Dans la partie qui suit, nous présenterons la conception de l'architecture de notre système.

Partie2 : Conception

A travers cette partie, nous décrivons la conception effectuée pour réaliser convenablement le travail demandé.

Pour ce faire, nous choisissons de donner en premier lieu une idée sur les composants d'une application Android afin de comprendre sa structuration et par la suite nous détaillons la conception de notre application au moyen des diagrammes de classes et de séquences.

Après avoir parlé de la conception générale, nous choisissons dans un second lieu le modèle d'architecture adéquat pour la réalisation.

I. Conception générale :

I .1.Les composants d'une application Android

Une application Android consiste en un assemblage de composants liés via un fichier de configuration, qui présentent en quelque sorte les briques sur lesquelles se repose l'application. Ces concepts fondamentaux à préciser sont :

- Les vues

Les Views sont les composants basiques de l'interface graphique. Ce sont les éléments de l'interface que l'utilisateur voit et sur lesquels il agit. C'est de la classe View qu'héritent les widgets (exemple de composants graphiques tel que les boutons), les layouts (le plan sur lequel on organise et on place les composants graphiques) et tous les composants graphiques servant à la création d'une interface graphique interactive.

- Les contrôles

C'est bien la classe des composants graphiques cités dessus. Les contrôles sont tels que les boutons, les champs de saisie de texte, les cases à cocher, etc.

- Les activités

Une Activity représente la fenêtre qui sera affichée à l'utilisateur. C'est un ensemble de vues et de contrôles composant une interface logique. Elle permet également de gérer des fonctionnalités telles que l'appui sur une touche, l'affichage de messages, etc. Ce concept repose essentiellement sur l'interaction de l'utilisateur avec l'écran.

- Les ressources

Chaque application Android a ses propres fichiers ressources. C'est dans ces fichiers que seront puisés les textes, les images, les couleurs, etc.

- Le fichier de configuration

C'est fichier auto généré par l'application, qui lui est indispensable. C'est un fichier XML appelé « AndroidManifest» qui décrit le point d'entrée de l'application (le code à exécuter), les composants du projet ainsi que les permissions nécessaires pour l'exécution du programme.

Une illustration explicative de ces concepts est représentée par le schéma de la figure.

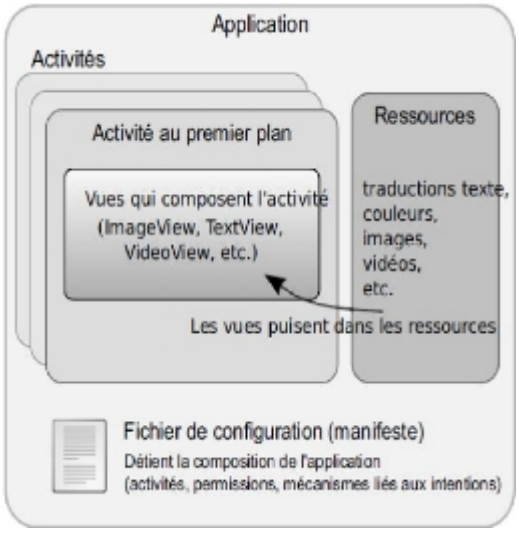


Fig. **II.7.** Composition d'une application Android.

II .Conception détaillée :

II.1.Diagramme de séquence :

Le diagramme de séquence représente la succession chronologique des opérations réalisées par les acteurs. Il montre les interactions entre les objets, en montrant les messages qu'ils échangent entre eux ordonnés dans le temps.

II.1.1 Authentification :

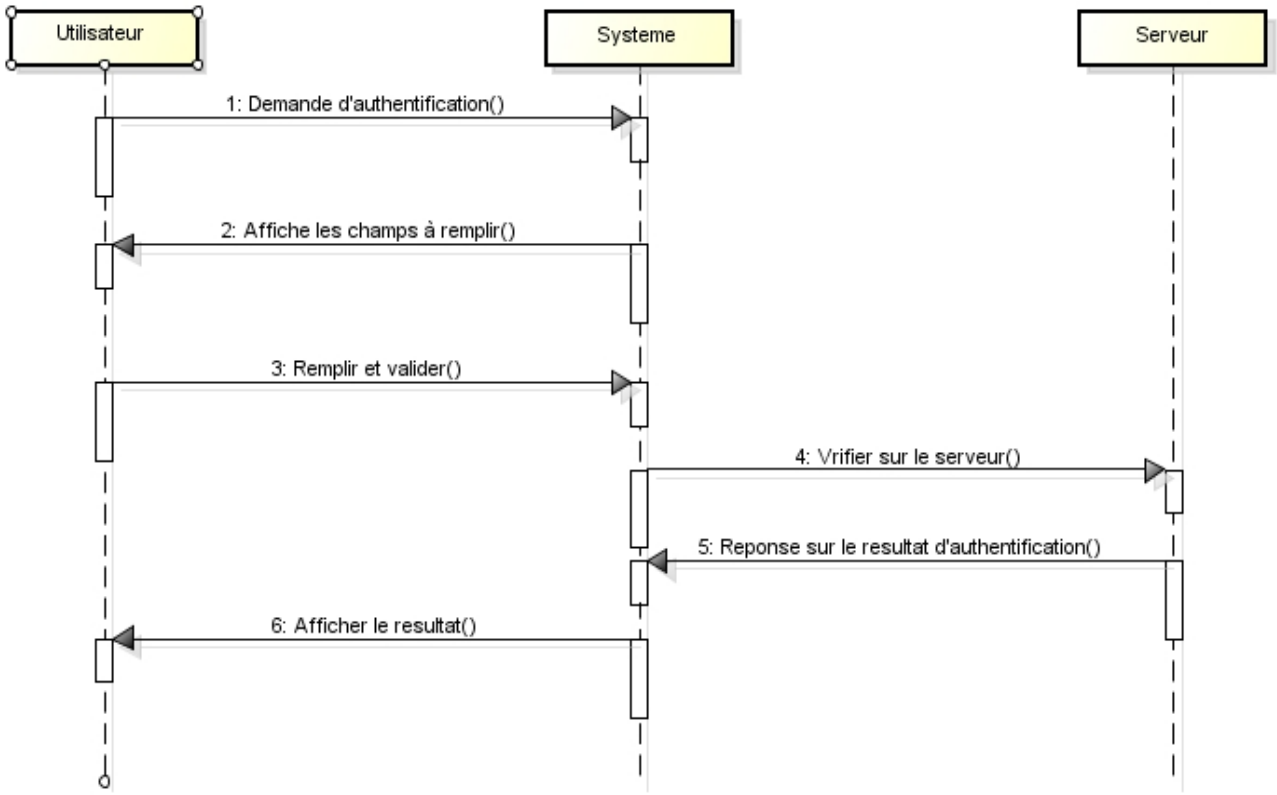


Fig. III.8: Diagramme de séquence du cas d'utilisation **«Authentification»**

II.1.2.Ajouter un projet :

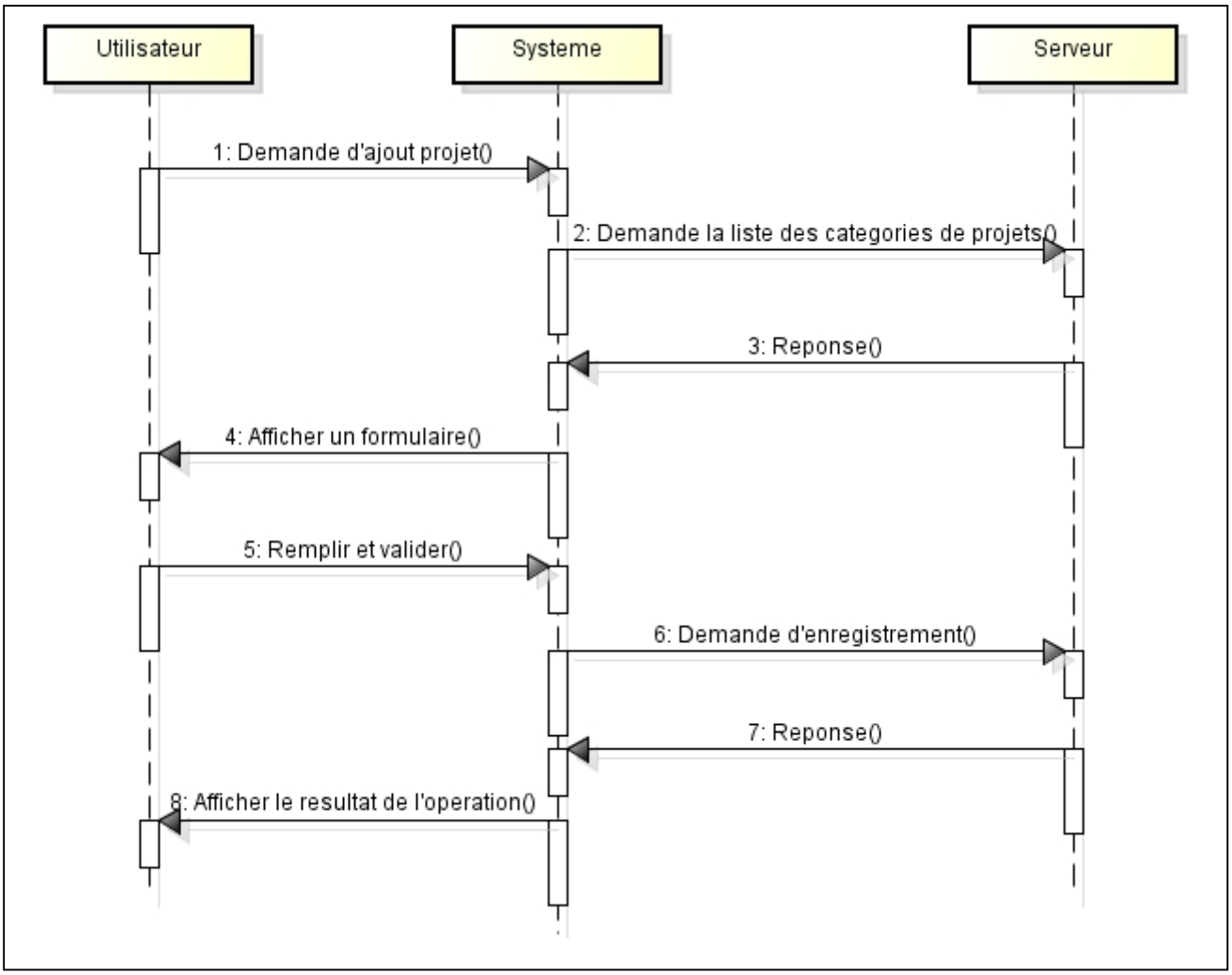


Fig. III.9: Diagramme de séquence du cas d'utilisation **«Ajouter un projet»**

II.1.3.Modifier un projet :

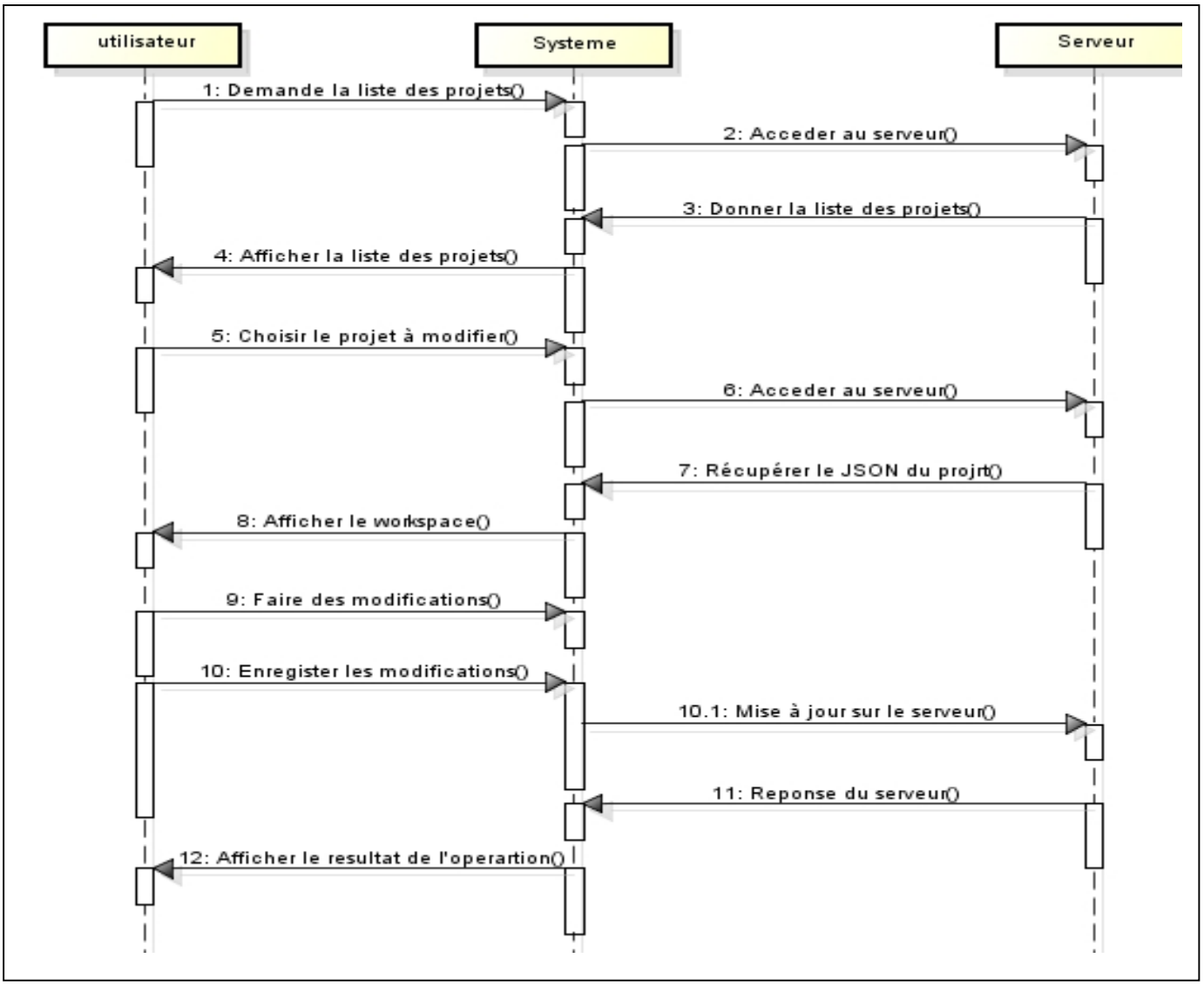


Fig. III.10: Diagramme de séquence du cas d'utilisation **«Modifier un projet»**

II.1.4.Déconnexion :

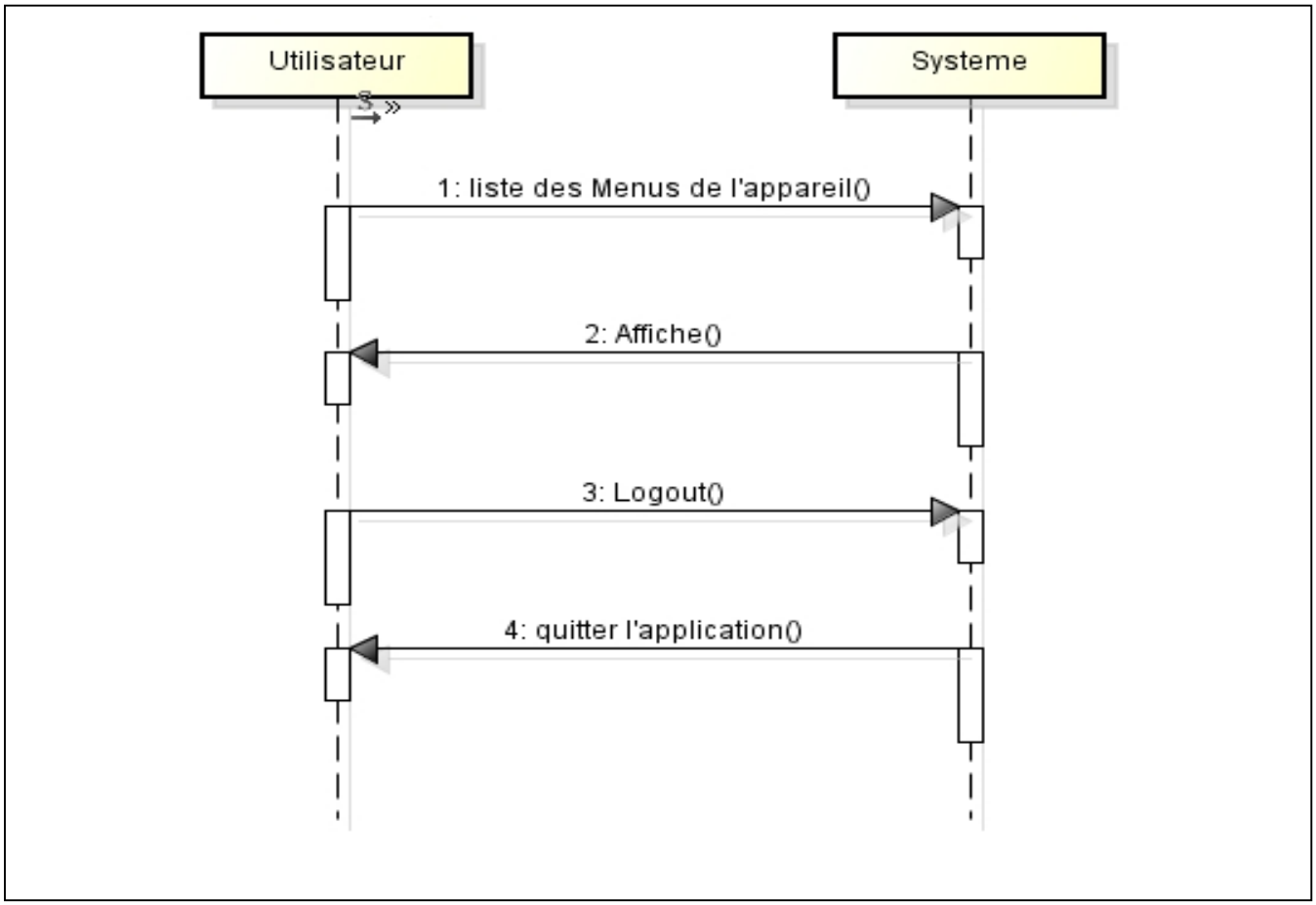


Fig. III.11: Diagramme de séquence du cas d'utilisation **«Déconnexion»**

III. Conception architecturale :

L'élément le plus important dans notre travail est la gestion des projets pixappy sur un équipement mobile, ce dernier se manifeste dans la classe projet. Cette classe véhicule les interfaces, les procédures et les données descriptives d'un projet. Elle est assez complexe pour la fragmenter en plusieurs classes interdépendantes. Pour cela nous avons adopté une architecture de type MVC qui sera présenté dans la section suivante.

III.1.Description du model MVC :

L'architecture MVC (modèle, vue et contrôleur) est une architecture à trois couches utilisée pour la programmation client/serveur et d'interface graphique.

C'est un modèle architectural très puissant qui intervient dans la réalisation d'une application. Il tire sa puissance de son concept de base qui est la séparation des données (modèle), de l'affichage (vue) et des actions (contrôleur).

C'est trois couches sont décrites comme suit:

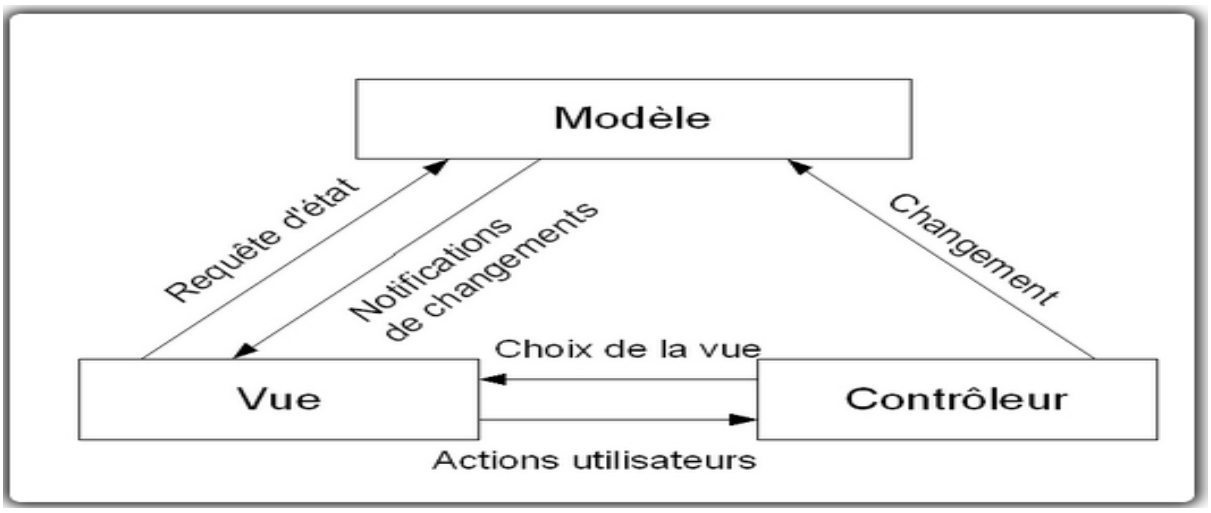


Fig III.12. Schéma de l'architecture MVC

Dans ce qui suit, nous présentons le diagramme de classe général de notre système:

IV. Diagramme de classe général :

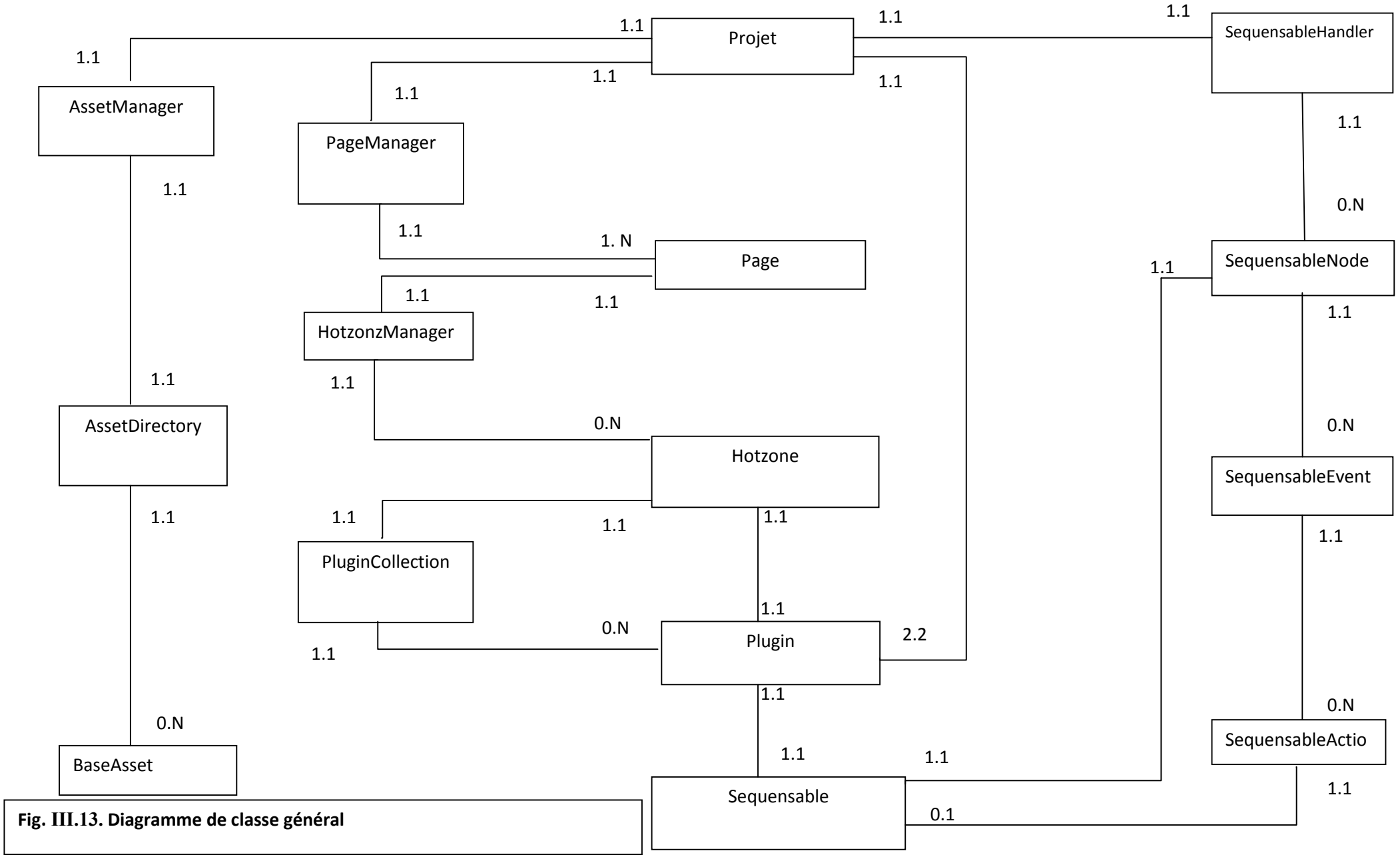


Fig. III.13. Diagramme de classe général

IV.1 .Interprétation :

Le diagramme ci-dessus illustre l'ensemble des classes implémentées dans notre système, dont la classe principale « Project » qui représente un ensemble d'information le concernant comme : la hauteur et la largeur du WorkSpace, D'autres classes s'associent à cette classe principale par un ensemble de cardinalité. On y trouve :

-Les classes qui étendent « Manager » et qui gèrent un ensemble d'objets différents, dont les fonctionnalités sont les même (Ajout, Suppression). Ces objets sont créés via les classes fille : AssetManager (s'occupant des répertoire contenant des ressources comme les images, les videos,…), PluginCollection(Se chargeant de la gestion des plugins contenus dans un Hotzone (Calque)), PageManager (S'occupant de la gestion des pages contenus dans une page d'un projet), HotzoneManager (S'occupant de la gestion des hotzones contenus dans une page d'un projet) et la classe SequensableHandler (Qui gère l'ensemble des sequensables d'un projet) : Un sequensable est la classe qui définit l'identifiant unique du plugin ainsi que les événements auxquels il répond et les actions qu'il peut effectuer comme un événement « Click » et une action de déplacement.

-AssetDirectory (Cette classe représente un répertoire contenant des assets (Ressources)), contenudans la classe AssetManager.

-BaseAsset : Représente le socle commun entre tous les assets(Ressources).

-SequensableNode : Le premier élément qui relie le plugin émetteur d'événements au plugin à l'écoute exécutant une action. Elle contient une liste de SequensableEvent.

-SequensableEvent : représente un événement spécifique déclenché par un plugin. Il contient une liste d'actions à exécuter lors de son déclenchement.

-SequensableAction : représente une action à exécuter.

Le diagramme suivant représente l'ensemble des classes qui étendent la classe *Manager :*

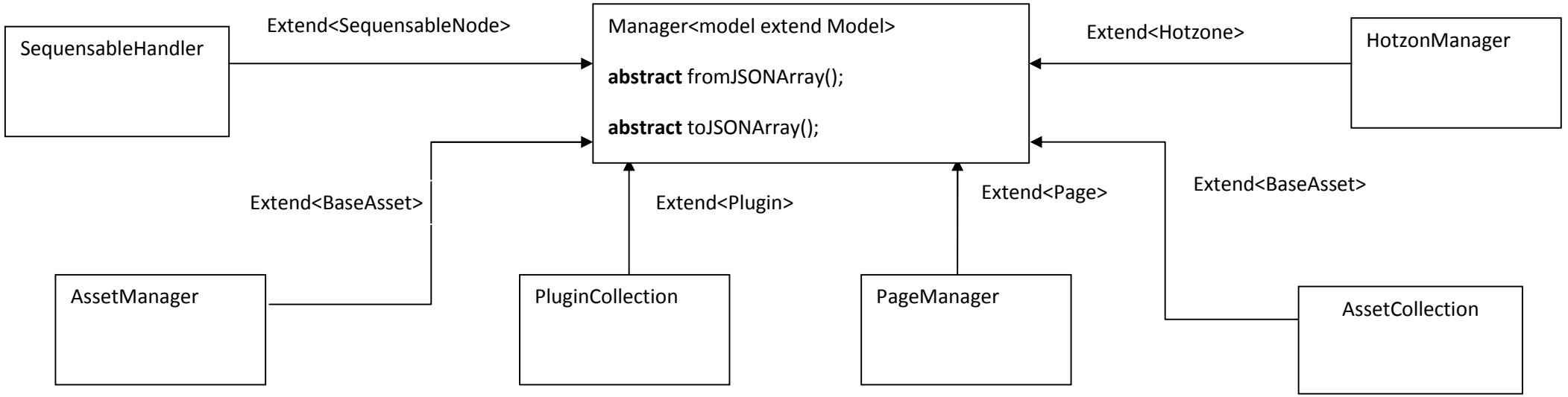


Fig. **III.14** : Schématisation de l'héritage entre la classe abstraite « Manager» et chacune de ses filles.

V. Conception graphique de l'interface utilisateur :

Dans ce qui suit nous allons présenter l'aspect graphique et ergonomique de l'interface utilisateur (IHM).

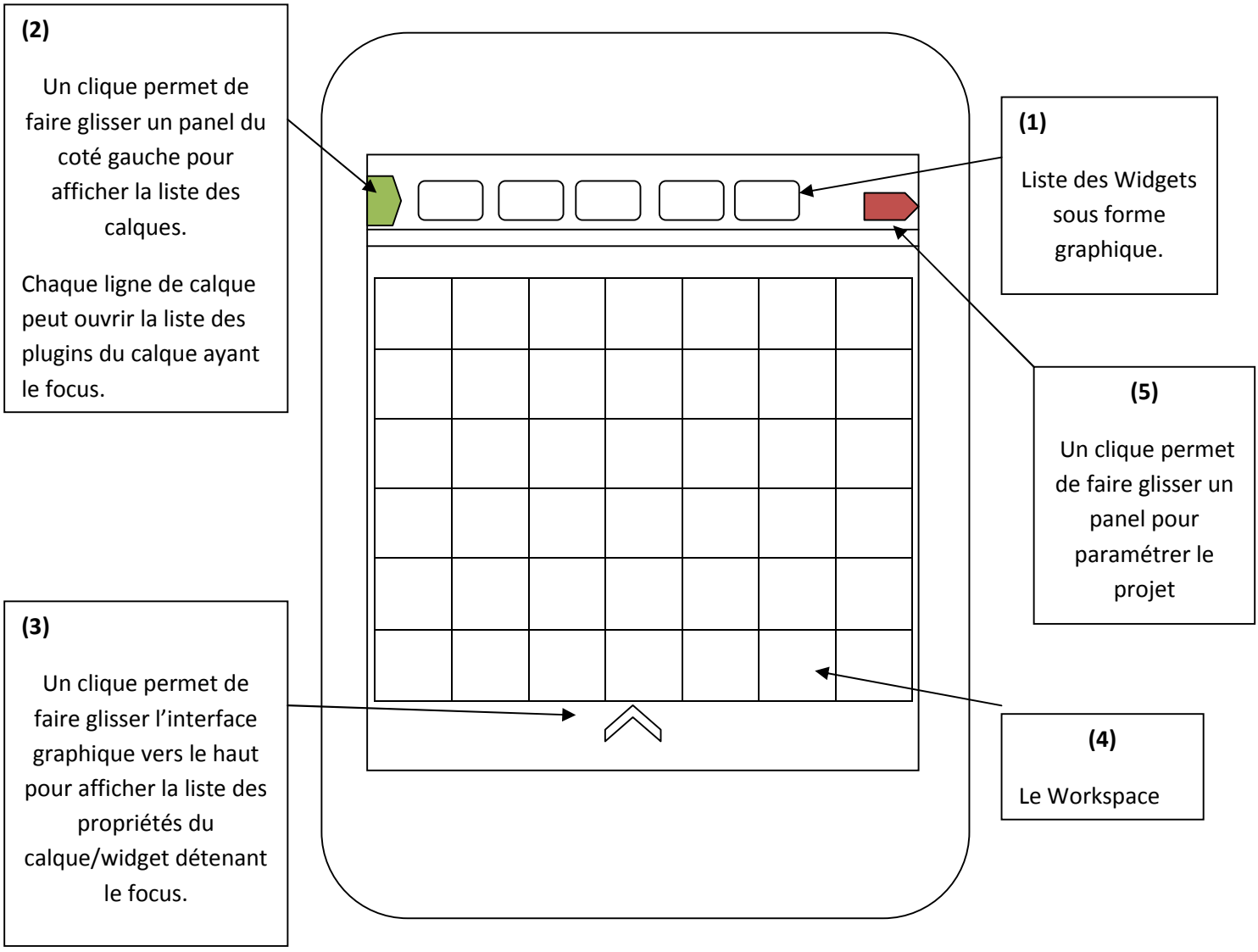


Fig. **III.15. Spécification du Designer Mobile**

VI. Fonctionnement des différents éléments du designer mobile :

(1) : C'est un layout à défilement horizontale contenant les différents widgets qu'on peut utiliser .Pour cela il suffit de faire un glisser/déposer de l'élément voulu vers le (4) Workspace pour l'intégrer a un calque existant ou en crée un nouveau.

(2) : Il permet la visualisation des calques présents dans (4) et un click sur l'un d'eux lui attribut le focus.

(3) : Il permet de changer les propriétés d'un calque ou d'un widget selon le type d'objet qui a le focus (la sélection d'un calque ou d'un widget au par avant lui attribut une couleur particulière).

(4) : C'est l'espace de travail du projet (la surface visible dans le rendu final).

(5) :il permet de paramétrer le projet(les ressources disponibles, paramétrage du Workspace….)

- **Liste des widgets implémenté :**

1. Show/Hiden
2. Avant/Après
3. Gratter
4. Rotation
5. Zoom
6. Image
7. Lien
8. Texte
9. Move

Show/hiden : L'utilisateur se verra afficher une photo initiale qui disparaitra lorsqu'il cliquera dessus pour laisser place à la seconde.

Avant/Après : L'utilisateur aura a sa disposition une barre horizontale pour afficher/masquer la photo masquer.

Gratter : En utilisant son doigt il fera apparaitre la photo en arrière plan morceau par morceau correspondant à zones touchées.

Rotation : Selon l'événement déclencheur le widget effectuera une rotation sur lui-même d'un degré donné.

Zoom : En touchant une zone, l'utilisateur fera apparaitre un rectangle montrant la partie correspondante de la photo en arrière plan. Déplacé son doigt sans relâcher fera déplacer le rectangle.

Image : Ce widget permet d'intégrer une image.

Lien: Il permet la réorientation vers une adresse web ou une boite E-mail.

Texte : Il permet d'insérer du texte dans différents formats.

Move : Il permet de faire un déplacement pour les hotzones qui le contient.

VII. Conclusion :

Après avoir présenté la conception globale et détaillée de notre système, nous sommes capables maintenant d'entamer la partie réalisation

Réalisation

I .Introduction :

Pour pouvoir mener à bien un projet informatique, il est nécessaire de choisir des technologies permettant de simplifier sa réalisation. Pour cela, après avoir compléter l'étude conceptuelle dans le chapitre précédent, nous allons aborder la partie implémentation dans ce qui suit. Nous commençons par présenter l'environnement matériel et logiciel, ensuite nous présenterons les différentes interfaces de l'application.

II. Environnement matériel et logiciel :

II.1. Environnement matériel

II.1.1Architecture matérielle :

L'architecture de notre application est à 3 niveaux (*architecture 3-tiers*), elle est partagée entre:

- Le client Android : Conteneur d'application et demandeur de ressources.
- Le serveur Web : Vue que les données serons communiquées entre deux environnements hétérogènes, le rôle principale du serveur web est de gérer la communication entre le client Android et le serveur de base de données,
- Le serveur de base de données fournis les données au serveur web.

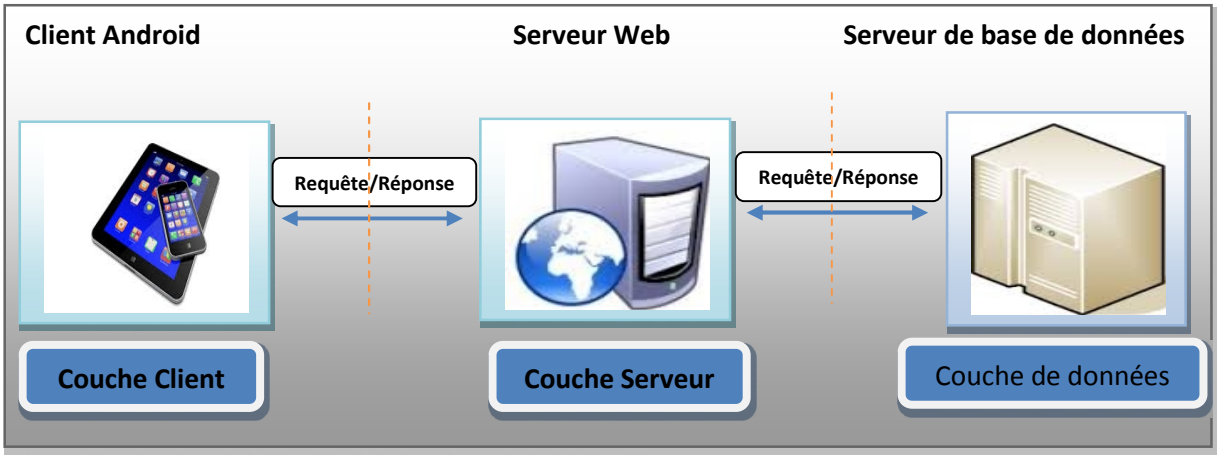


Fig. IV.1 : *Architecture matériel du système*

II.1.2.Matériels utilisés :

Pour la réalisation de l'application :

Un *pc portable* pour le développement ayant les caractéristiques suivantes :

- Intel core i3 2.10 GHz.
- 4 Go de mémoire vive.
- Windows 7 - 64 bits.

Un Smartphone Sony Xperia pour réaliser les tests.

II.1.3.Technologie :
Ci-dessous un tableau représentant les différentes technologies utilisées dans notre application :

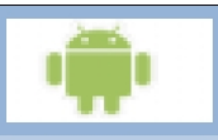
	Android Système d'exploitation open source pour Smartphones, PDA et terminaux mobiles.
	PHP Langage de scripts libre principalement utilisé pour produire des pages Web dynamiques.
	MySQL Système de gestion de base de données (SGBD).
	JSON (JavaScript Object Notation) Format de données textuel, générique, dérivé de la notation des objets du langage ECMAScript.

Tableau IV.1 : Les différentes technologies utilisées dans l'application

La méthode la plus répandue de se connecter à une base de données MySQL à distance à partir d'un appareil Android, est de l'effectuer à l'intermédiaire d'un serveur web. Le SGBD MySQL est habituellement utilisé avec le langage PHP, donc la façon la plus simple et la plus évidente est d'écrire des scripts PHP pour gérer la base de données et exécuter ces scripts en utilisant le protocole HTTP du système Android. Nous avons codé les données dans le format JSON (voir section IV-2).

L'architecture *3-tiers* du point de vue technologique, le client est la plateforme Android, le serveur web est le PHP et le serveur de bases de données est le MySQL.

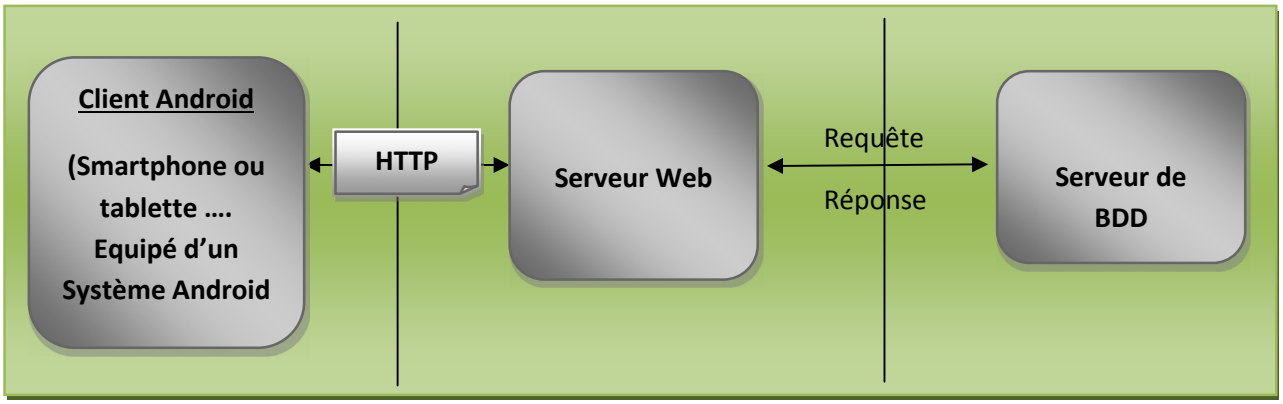


Fig. IV.2 : *Architecture 3-tiers du point de vue technologique*

II.2. Environnement logiciel :

Pour pouvoir réaliser une application dans des bonnes conditions il est de bien entendu de choisir son environnement de travail selon les besoins. Nous avons opté pour la réalisation d'une application mobile sous le système Android ce qui nous impose de travailler sous Android Studio avec le langage JAVA et le SDK Android pour son développement, afin d'obtenir un fichier à extension **.***apk* qui sera par la suite installé sur des terminaux mobiles de différents types fonctionnant avec le système Android 3.0 ou plus.

Le développement d'applications pour Android se fait entièrement en Java, ce dernier est un langage puissant orienté objet, utilisé très largement dans le monde du développement, en utilisant des IDE tels que : Eclipse, NetBeans et Android Studio.

II.2.1. JAVA :

Est un langage de programmation informatique orienté objet, La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows,
Mac OS ou Linux, avec peu ou pas du tout de modifications. Pour cela, diverses plateformes associées visent à garantir la portabilité des applications développées en Java.

II.2.2. XML :

XML (eXtensible Markup Language, soit « Langage de balisage extensible ») est un langage de balisage définissant un format universel de représentation des données, permettant de décrire la structure hiérarchique d'un document. Le fichier à extension .xml contient à la fois les données et les indications sur le rôle que jouent ces données, ces indications (ou balises) permettent de déterminer la structure des documents.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:orientation="vertical" >
   <TextView
      android:layout_width= " wrap_content"
      android:layout_height="wrap_content"
      android:text=  " Donnez  votre nom"/>
   <EditText
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:id="@+id/nom"/>
   <Button
      android:text= " OK"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/ok"/>
</LinearLayout>
```

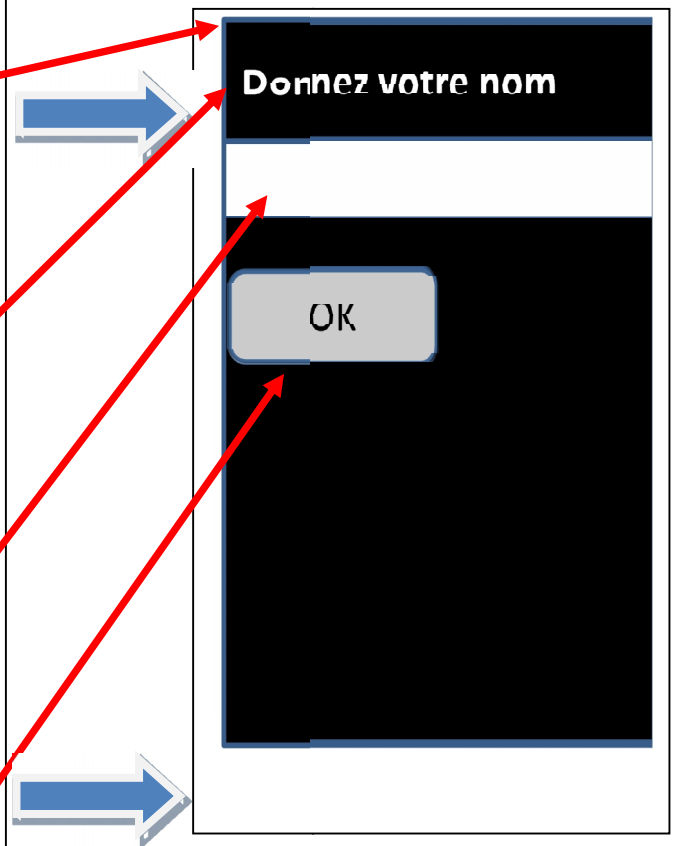


Fig. IV.3.Exemple d'un fichier XML

II.2.3.Android Studio

Android Studio est un environnement de développement des applications Android. Il est basé sur la version *IntelliJ*.

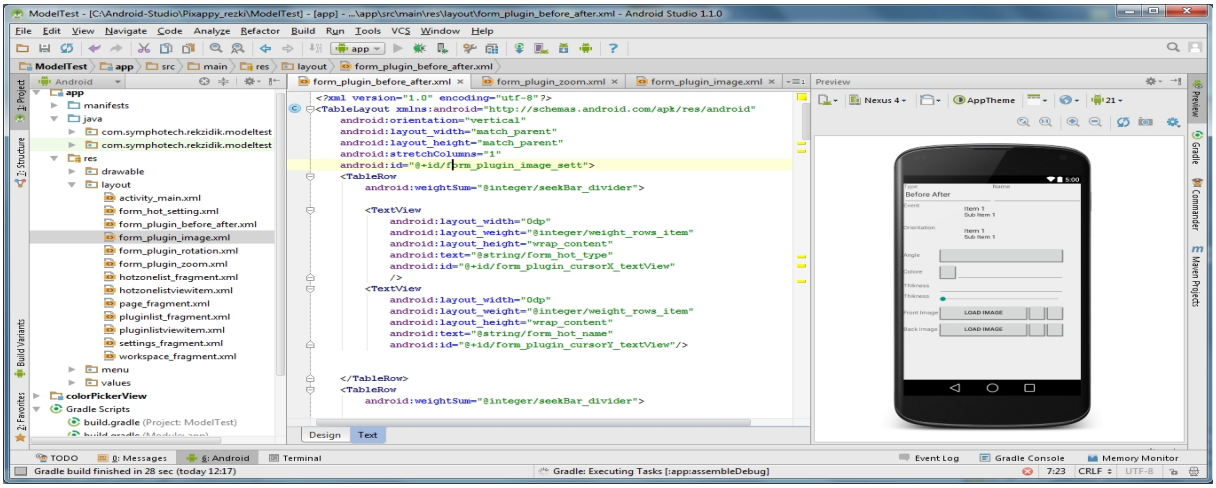


Fig. IV.4 : Interface de l'environnement Android Studio 1.1.0

II.2.4. SDK:

Signifie ***S**oftware **D**evelopment **K**it*, c'est un ensemble d'outils d'aide à la programmation pour concevoir des logiciels, jeux, applications mobiles, etc. pour un terminal et/ou un système d'exploitation spécifique. Un SDK contient du code**,** permettant de concevoir une interface ou une partie d'une interface numérique (web, mobile, jeux, logiciels de recherches, widget météo…). Ce code est conçu avec le langage de programmation correspondant au terminal (ordinateur, téléphone, tablette…) et au système de navigation ciblés. Par exemple, pour développer une application mobile (logiciel) pour iPhone (terminal mobile avec système d'exploitation iOS), il faut utiliser le SDK iOS mobile. Ce code est organisé sous forme de librairies (ou bibliothèques logicielles) c'est-à-dire des collections de fonctions prédéfinies, de points d'accès à du matériel et à des fonctionnalités système (ou natives) d'un terminal.

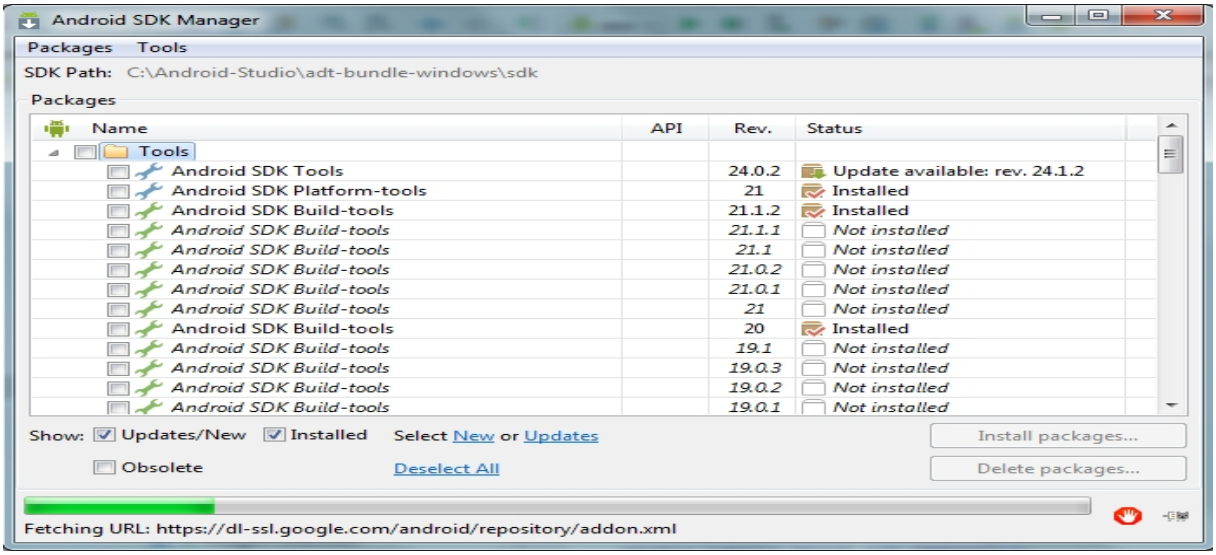


Fig. IV.5 : Interface de l'Android SDK Manager

II.2.5. L'émulateur :

L'Android Virtual Device, Emulator en anglais, aussi appelé AVD, est un émulateur de terminal, c'est-à-dire un logiciel qui fait croire à votre ordinateur qu'il est un appareil sous le système Android et pour chaque version d'Android est associée une version de l'émulateur . C'est la raison pour laquelle on n'a pas besoin d'un périphérique sous Android pour développer et tester la plupart de vos applications ! En effet, une application qui affiche un calendrier par exemple peut très bien se tester dans un émulateur, mais une application qui exploite le GPS doit être éprouvée sur le terrain pour que l'on soit certain de son comportement ou la gestion du Bluetooth.

Fig. IV.6 : Interface de l'émulateur

II.2.6. Git :

Git est un logiciel de gestion de versions, il est utilisé principalement par les développeurs. En effet, il est quasi exclusivement utilisé pour gérer des codes sources, car il est capable de suivre l'évolution d'un fichier texte ligne de code par ligne de code. Il est fortement conseillé pour gérer un projet informatique.

Ce type de logiciel est donc par conséquent deux utilités principales :

- **Suivre l'évolution d'un code source** : pour retenir les modifications effectuées sur chaque fichier et être ainsi capable de *revenir en arrière* en cas de problème.
- **Travailler à plusieurs** : Si deux personnes modifient un même fichier en même temps, leurs modifications doivent pouvoir être fusionnées sans perte d'information.

Quelques commandes de Git :

Git Commit : permet d'enregistrer la nouvelle version du code source.

Git Push: permet d'envoyer le code source sur le serveur pour l'enregistrer.

Git Pull: permet de télécharger les nouveautés du projet modifié depuis le serveur.

Git Status: permet de savoir l'état des fichiers modifiés (s'il y en a) par apport au dernier commit effectué.

III.2.7. Wireshark : est un logiciel de capture de trames, il permet d'analyser finement le trafic réseau, et de récupérer les paquets qui passent physiquement sur un réseau d'une façon graphique.

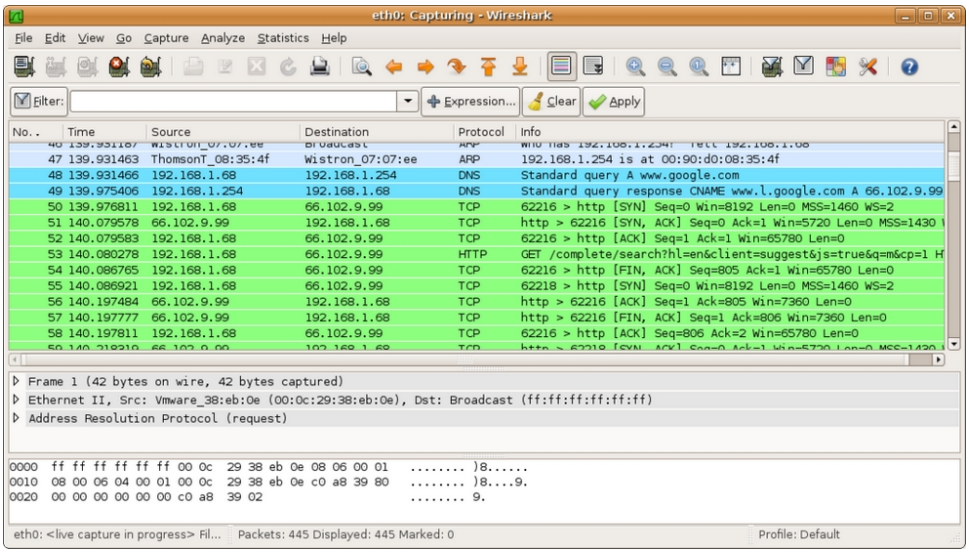


Fig. IV.7 : Interface du capteur de trames Wireshark

III. Protocole et format de donnes

III.1.Protocole de communication :

Dans notre projet, nous avons utilisé le protocole HTTP, afin de communiquer les données entre le ***Designer Mobile*** et le serveur web. En effet, Le HTTP est un protocole qui définit la communication entre un serveur et un client (facilite le dispatch des fonctions).

III.2.Format de données communiquées :

JSON (*JavaScript Object Notation*) est un format de données textuel, générique, dérivé de la notation des objets du langage *ECMAScript*. Il permet de représenter des informations structurées.

Un document JSON ne comprend que deux éléments structurels :

- des ensembles de paires nom / valeur.
- des listes ordonnées de valeurs.

Ces mêmes éléments représentent 3 types de données :

- des objets.
- des tableaux.
- des valeurs génériques de type tableau, objet, booléen, nombre, chaîne ou nulle.

Format JSON :

```
{
    "menu": {
        "id": "file",
        "value": "File",
        "popup": {
            "menuitem": [
                { "value": "New", "onclick": "CreateNewDoc()" },
                { "value": "Open", "onclick": "OpenDoc()" },
                { "value": "Close", "onclick": "CloseDoc()" }
            ]
        }
    }
}
```

Fig. IV.8. Format de données JSON

Le principal objectif du choix d'utilisation du JSON, dans notre application, est qu'il est simple à mettre en œuvre. Comme il présente les avantages suivants :

- Facile à apprendre, car sa syntaxe est réduite et non-extensible;
- Ses types de données sont connus et simples à décrire ;
- Peu verbeux et léger, ce qui le rend bien adapté aux terminaux mobiles contrairement au langage XML qui est très verbeux.

IV- Interface Homme/Machine de l'application:

Dans ce qui suit nous présenterons les différentes interfaces de l'application en citant les détails de chaque imprime écran.

Au premier lancement de l'application, il faut se connecter sur le site web pour créer un compte et l'activé depuis sa boite e-mail. L'utilisateur pourra ensuite accéder à toutes les fonctionnalités de l'application.

IV.1. Interface « Accueil et visionneuse »:

C'est l'interface principale, elle s'affiche lors du lancement de notre application, elle permet de se connecter à son propre compte et de visualiser les projets publiés sur la plate forme pixappy en mode visionneuse :

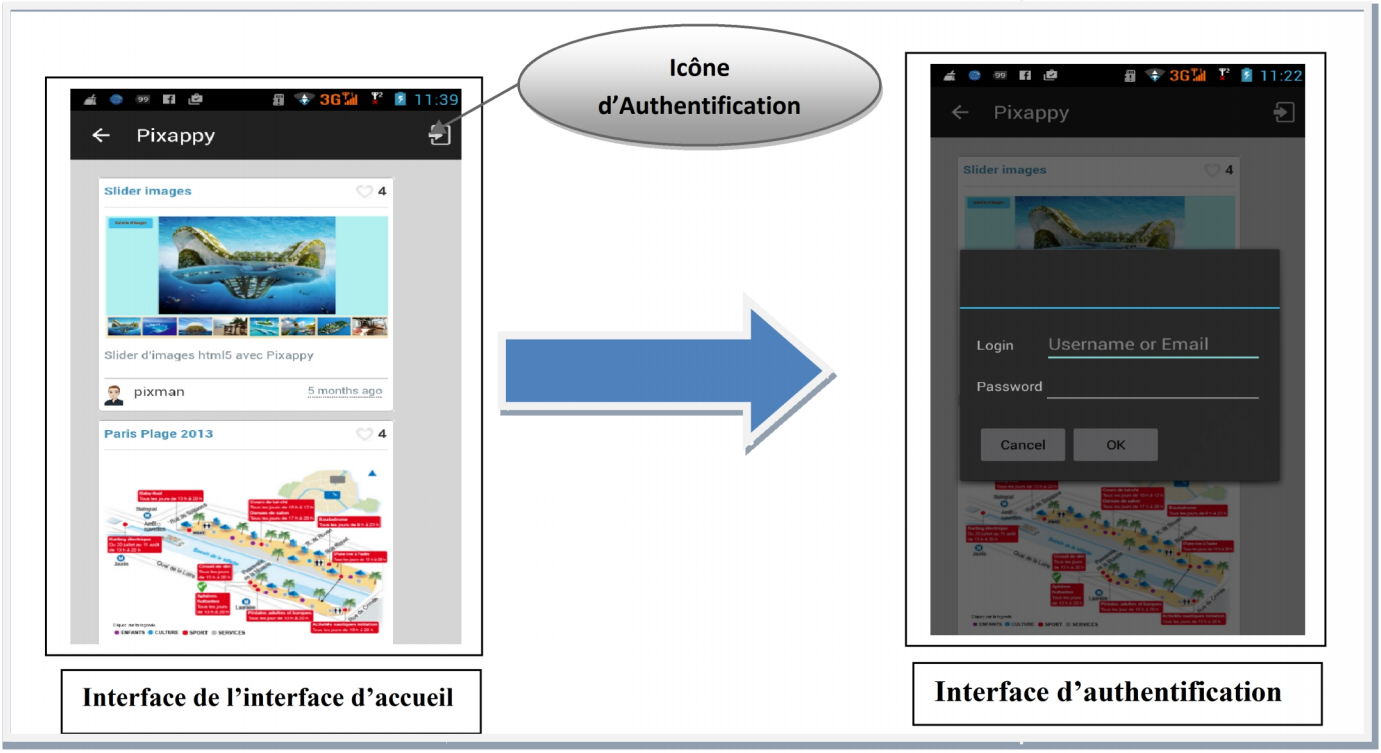


Fig. IV.9 : le passage de l'interface d'accueil à l'interface d'authentification

IV. 2. Création et gestion du projet :

Une fois connecté, l'utilisateur peut accéder à chaque fonctionnalité de l'application grâce aux différents onglets des interfaces de l'application

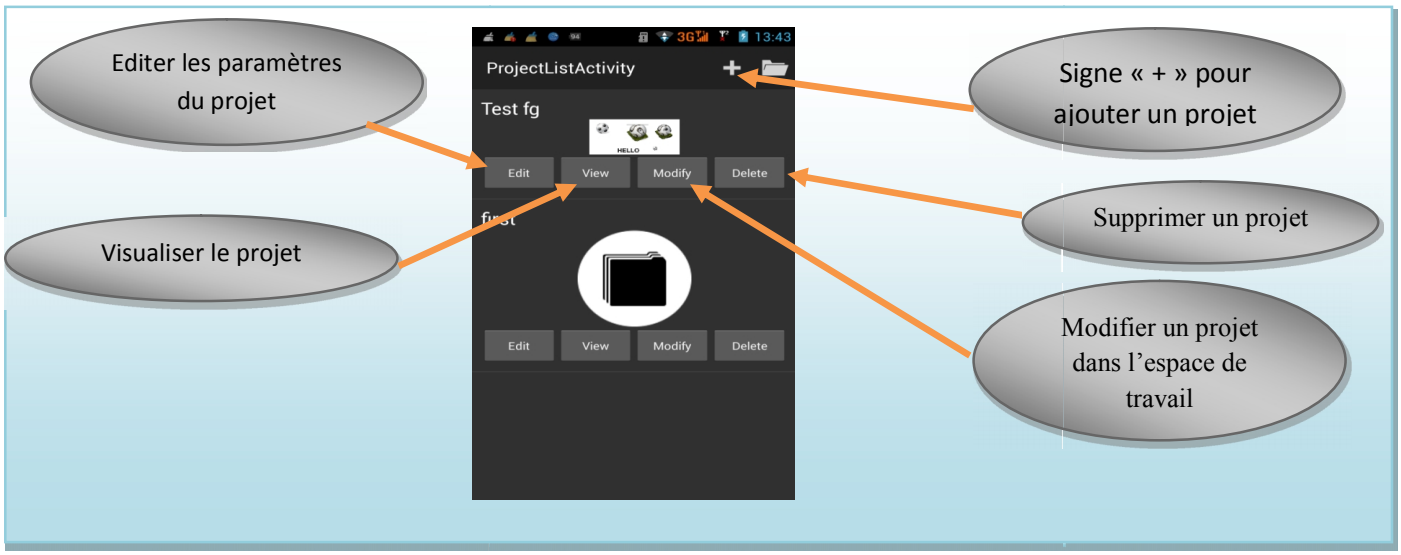


Fig. IV.10. Interface gestion de projet

Une fois cliquer sur le signe plus(+) une autre activité d'ajout prendra le focus (voir la figure IV.11) toute en permettant de remplir les champs pour paramétrer le projet.

pour ajouter un projet on doit remplir le formulaire de la figure ci-dessous

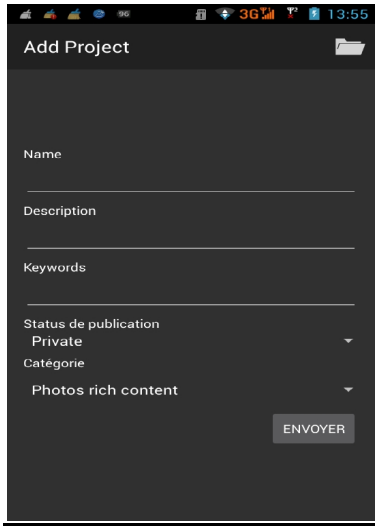


Fig. IV.11 : Interface d'ajout projet

IV. 3. Interface Espace de travail «workspace»

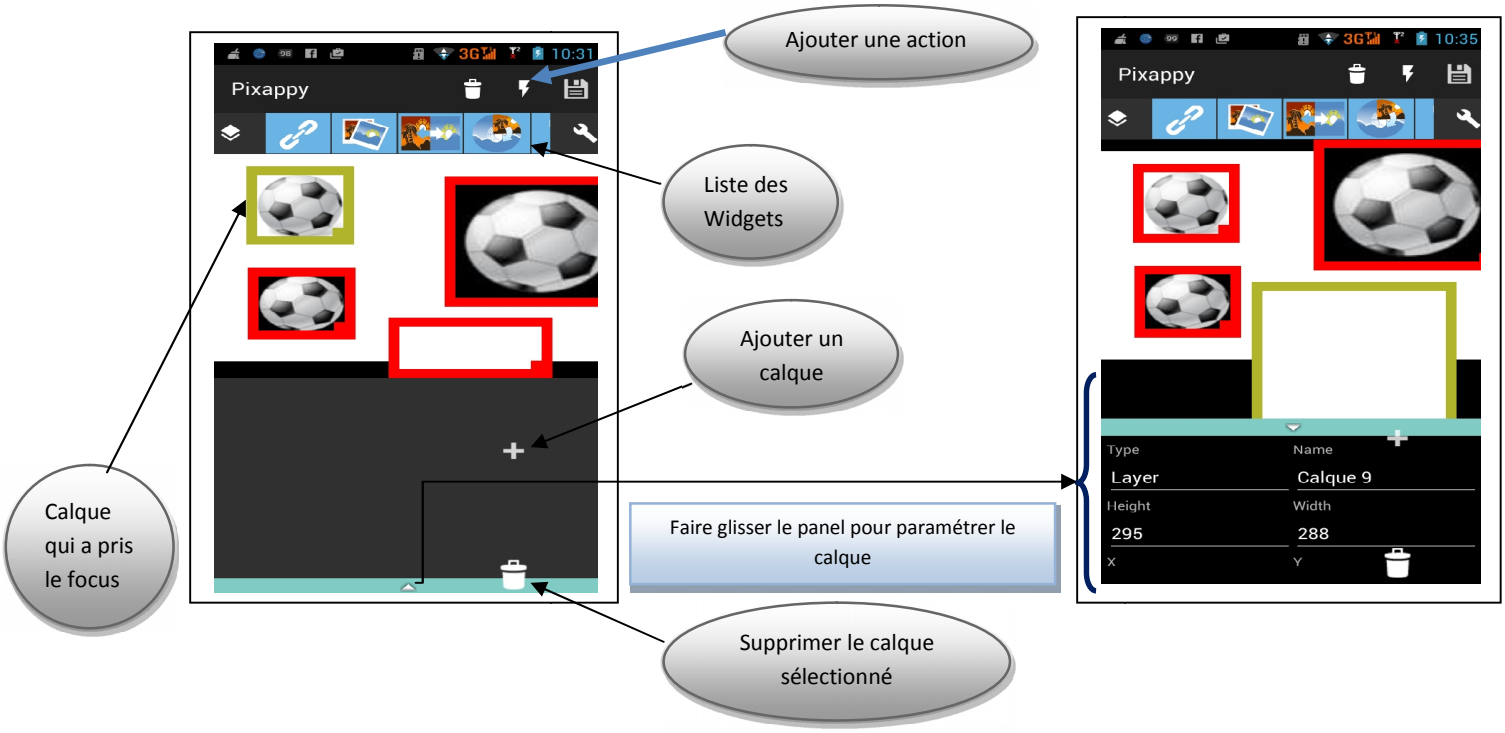


Fig. IV.12 : Interface Workspace

IV. 4. Interface « Ajouter une action à un calque » :

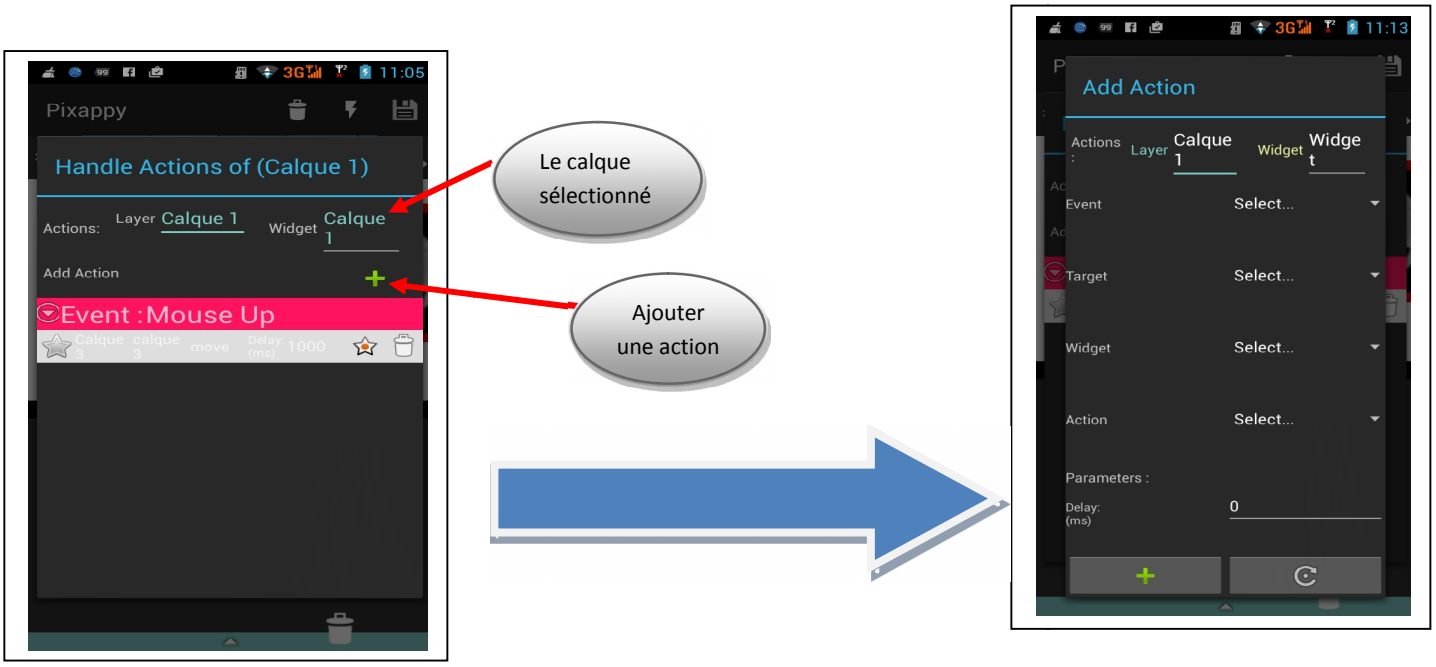


Fig. IV.13 : Interfaces d'ajout action pour un Widget

IV. 5. Interface « Importer des ressources » :

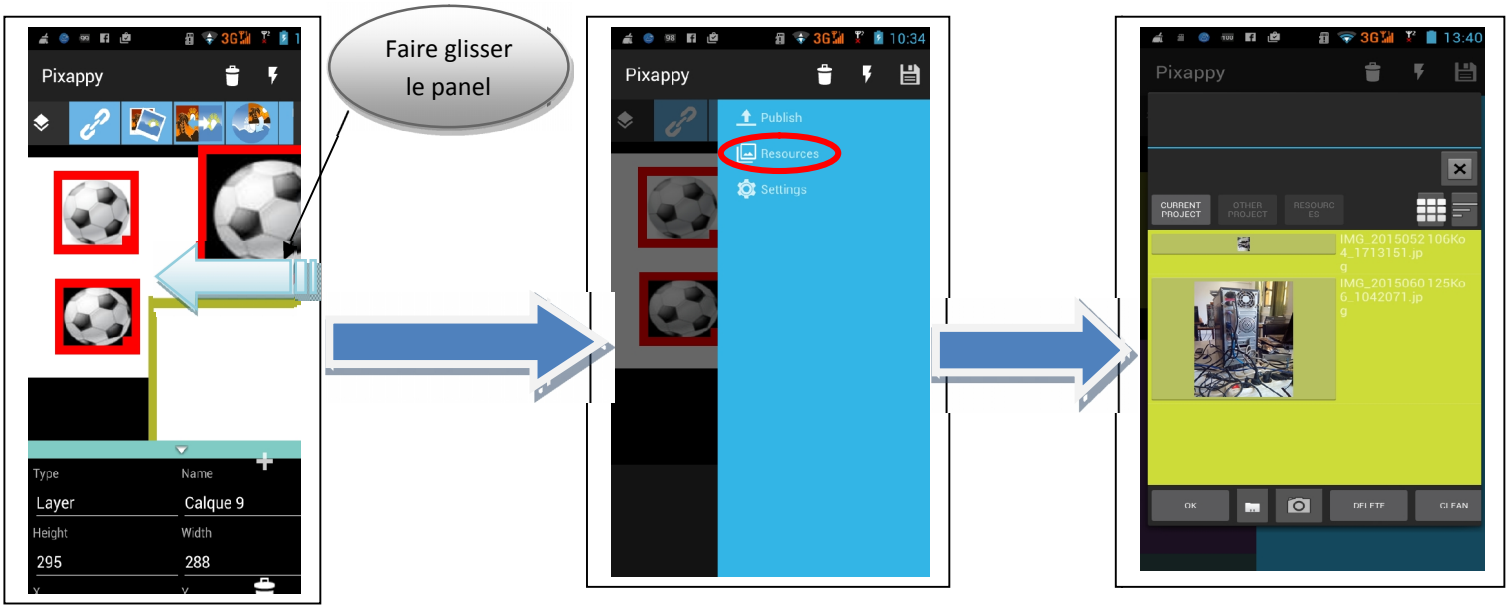


Fig. IV.14 : Interface « Importer des ressources »

V.6. Interface « Afficher la liste des plugins associés à un calque » :

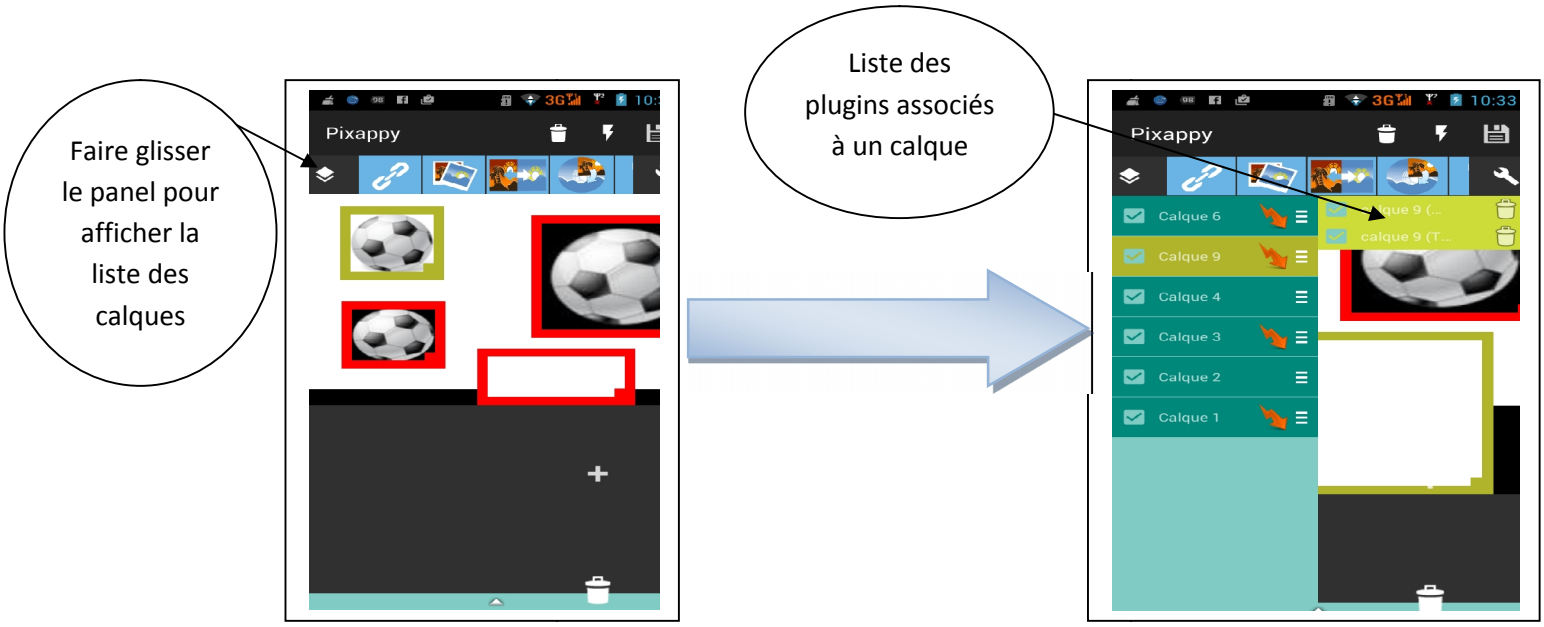


Fig. IV.15.Affichage de la liste des plugins associés à un calque

V. Conclusion :

Dans ce chapitre de réalisation, nous avons présenté les plates-formes matérielles et logicielles sur lesquelles nous avons développé notre projet, ainsi que les technologies employées. Nous avons, par la suite, présenté les interfaces les plus significatives de notre application.

Conclusion générale

L'élaboration de notre travail est dans le but de concevoir une application dédiée aux terminaux mobiles disposant de la plateforme Android. Cette application permet de réaliser des contenus *rich media* à l'aide d'un certain nombre d'actions directement à partir de son Smartphone.

Pour ce faire, nous avons recouru à différentes technologies, outils et orientation jugées nécessaires pour aboutir à l'objectif de notre application.

Dans ce rapport, nous avons détaillé les différentes étapes suivies pour une réalisation réussie du travail demandé.

Dans un premier lieu, nous avons commencé par introduire la société d'accueil et le cadre de l'élaboration du projet ainsi que la méthodologie suivie.

La deuxième étape dans la réalisation de notre projet était de faire une étude théorique sur les notions de base d'Android et son positionnement par apport aux autres OS mobiles.

Nous avons par la suite entamé l'analyse et la spécification des besoins en citant les besoins fonctionnels et non fonctionnels de notre projet et les modéliser par le diagramme des cas d'utilisation en indiquant le langage de modélisation utilisé.

Partant de la spécification, nous avons commencé la conception de l'application à travers les différents diagrammes UML à savoir les diagrammes de classes et les diagrammes de séquences.

Enfin, nous avons décrit, dans le chapitre de réalisation, l'implémentation de notre application tout en présentant quelques imprimes écran de certaines interfaces.

Ce stage s'est révélé bénéfique sur plusieurs points : il nous a permis de travailler sur une technologie pour terminaux mobiles et accroître nos connaissances dans le domaine de l'embarqué en abordant plusieurs aspects techniques d'Android.

Ce projet nous a également donné l'occasion de mieux connaître le milieu des sociétés de service informatique et de consolider nos expériences professionnelles.

En effet, Android offre l'opportunité de créer des logiciels mobiles innovants et révolutionnaires en encourageant les développeurs à puiser dans leur imagination et à mobiliser toutes leurs compétences pour exploiter au mieux les fonctionnalités de la plateforme.

Les perspectives

Pour améliorer de plus notre travail, nos perspectives à moyen ou court terme sont les suivantes :

- Supporter la totalité des plugins (Widget) de la plateforme.
- Ajouter la gestion et production de projet multipage.
- Supporter la fonctionnalité « Groupe » et « père/fils » (héritage) comme c'est déjà le cas sur la plateforme.
- Etendre l'importation d'asset aux ressources de la plateforme.
- Améliorer l'ergonomie du désigner en donnant la possibilité à l'utilisateur de la paramétrer.
- Sauvegarder sa liste des projets en local et lui donner la possibilité de travailler sans connexion.
- Implémenter une solution afin que l'utilisateur puisse s'inscrire sur la plateforme via l'application.

Webliographie

http://www.memoireonline.com/

http://www.developpez.com/

http://openclassrooms.com/

https://fr.wikipedia.org/wiki/Wikip%C3%A9dia:Accueil_principal

http://stackoverflow.com/

http://www.android.com/

Résumé du mémoire :

Notre Mémoire décrit les étapes de réalisation d'une application pour appareil mobile s'exécutant sous Android. L'application a pour but d'émuler (en partie) une plateforme de production de contenu rich media interactif (www.pixappy.com).