

2010 - 2011

Alexandre BOSLÉ
+00 33 (0)6 82 56 24 96
Erwan JEGOUIC
+00 33 (0)6 76 55 21 64
Abderahman KRIOUILE
+00 33 (0)6 13 15 33 81
Marcel Cyrille LAMENU
+00 33 (0)6 46 71 30 46



Tél. : 01 34 94 83 60

Projet Industriel E.S.I.A.L. - EBP Tableau de bord mobile

esial 

[RAPPORT FINAL]

Le rapport final fournit un rappel du contexte du projet, de la gestion de projet adoptée et des parties prenantes. Il vise toutefois principalement à donner un aperçu global et technique de l'évolution de l'ensemble du travail réalisé tout au long du projet, ainsi que les différentes difficultés rencontrées, les solutions fournies pour contourner les obstacles et les améliorations futures potentielles pouvant être articulées en option.

Table des matières

1. INTRODUCTION.....	2
1.1. HISTORIQUE	2
1.2. ENJEUX	2
2. EBP.....	3
2.1. PRESENTATION DE L'ENTREPRISE	3
2.2. MISE EN RELATION.....	3
3. « TABLEAU DE BORD MOBILE »	4
3.1. CAHIER DES CHARGES	4
3.1.1. DESCRIPTION DE LA DEMANDE	4
3.1.1.1. OBJECTIFS	4
3.1.1.2. PRODUIT DU PROJET	4
CARACTERISTIQUES MAJEURES DE L'APPLICATION	4
RESSOURCES EXPLOITEES PAR L'APPLICATION	6
MODULES OPTIONNELS DE L'APPLICATION.....	6
3.1.1.3. CRITERES D'ACCEPTABILITE ET DE RECEPTION	7
3.1.2. CONTRAINTES.....	8
3.1.2.1. CONTRAINTES DE COUT.....	8
3.1.2.2. CONTRAINTES DE DELAI	8
3.1.2.3. CONTRAINTES TECHNIQUES.....	9
3.1.3. PLANIFICATION	10
3.1.3.1. PLANIFICATION INITIALE.....	10
3.1.3.2. PLANIFICATION REVISEE	11
3.1.4. CONDUITE DU PROJET	12
3.1.5. RESSOURCES	13
3.1.5.1. RESSOURCES HUMAINES	13
COTE SERVEUR	13
COTE CLIENT	13
3.1.5.2. RESSOURCES LOGICIELLES.....	14
PLATE-FORME DE DEVELOPPEMENT	14
LANGAGE DE MODELISATION	14
BIBLIOTHEQUES EXTERNES	14
S.A.X. PARSER	15
KSOAP2.....	15
3.2. CONCEPTION	16
3.2.1. CAS D'UTILISATION	16
3.2.2. DIAGRAMMES D'ACTIVITES	17
3.2.2.1. CHOIX D'UN SERVICE.....	17
3.2.2.2. CHOIX DES DONNEES ET AFFICHAGE GRAPHIQUE	18
3.2.3. DIAGRAMMES DE SEQUENCES.....	19
3.2.3.1. AUTHENTIFICATION	19
3.2.3.2. AFFICHAGE GRAPHIQUE	20
3.3. DEVELOPPEMENT	20
3.3.1. SERVICE WEB.....	22
3.3.2. PARSING X.M.L.....	24
3.3.3. AFFICHAGE GRAPHIQUE	25
4. CONCLUSION	26

1. INTRODUCTION

1.1. HISTORIQUE

L'E.S.I.A.L., afin de renforcer les liens avec les entreprises industrielles, a mis en place un module intitulé « Projet Industriel », consistant à assigner un sujet de développement émis par le client, en l'occurrence, l'industriel, à un groupe d'étudiants qui devra, à partir des besoins du client, apporter une solution y répondant, en suivant scrupuleusement les différentes étapes de la gestion de projet.

C'est ainsi qu'au terme de la démarche de présentation des différentes thématiques, nous avons été affectés à l'entreprise EBP.

Le développement d'applications mobiles suit depuis quelques années une courbe exponentielle. Aussi, de cette tendance actuelle, le souhait pour un dirigeant d'entreprise de prendre connaissance, depuis son téléphone mobile, de l'évolution de son activité (chiffre d'affaires, stock, productivité, ventes sur une période donnée, etc.) découle-t-il logiquement.

Au vu de la rude concurrence existant dans le secteur prolifique de l'édition de logiciels, et de l'innovation permanente que celle-ci implique, il est en effet fondamental pour EBP de figurer au premier plan d'un tel domaine de la production informatique ; afin, d'une part, de satisfaire et de fidéliser ses clients potentiels, et, d'autre part, de démarcher de nouveaux clients.

1.2. ENJEUX

De nos jours, l'avancée de la technologie nous permet d'être connectés 24h / 24 grâce à une multitude d'outils de communication performants, mobiles, ergonomiques et miniaturisés, tels que *iPad*, *iPhone*, *Smartphone*, etc..

Dans ce contexte, et dans le cadre du projet industriel, il nous a été confié l'objectif de développer une application dont les différentes fonctionnalités permettront de visualiser, par le biais d'un téléchargement depuis une plate-forme distante, puis d'un affichage *ad hoc*, des données spécifiées par l'utilisateur, depuis un équipement mobile.

Du point de vue de l'utilisateur, l'application fonctionnera en temps réel, en ce sens qu'à un moment donné, les informations affichées par son appareil correspondront directement aux dernières versions des statistiques reflétant alors l'état courant du secteur de l'entreprise concerné.

Il nous revient ainsi d'élaborer une interface simple d'utilisation, conviviale, esthétique, et respectant un ensemble de normes structurelles ; en vue de satisfaire et fidéliser les clients en leur proposant un outil intuitif, portable, et efficace.

Le produit constituera, sur le plan commercial, un avantage de taille en ce sens qu'il permettra au client final de bénéficier d'un dispositif portable de suivi professionnel en temps réel, puis de le fidéliser quant à l'achat d'une licence de l'application.

2. EBP

2.1. PRESENTATION DE L'ENTREPRISE

Éditeur majeur de logiciels de gestion, EBP encourage les créateurs et les entrepreneurs à s'informatiser pour piloter leur activité commerciale, tout en se consacrant à leur cœur de métier.

Créée en 1984 par René Sentis, EBP Informatique s'est développée autour d'une idée simple : concevoir des outils de gestion informatique destinés aux P.M.E.-P.M.I., aux artisans et aux professions libérales.

Tout en conservant l'esprit des premiers logiciels voulus simples d'utilisation et performants, l'offre d'EBP s'est enrichie progressivement de solutions métiers spécifiques aux secteurs du bâtiment, de l'automobile, du commerce, de l'immobilier, de la restauration...

EBP compte aujourd'hui 300 salariés, dont plus de 110 développeurs et techniciens, et a réalisé un chiffre d'affaires de 27 M€ en 2009. Depuis 2002, le groupe EBP a une présence à l'international avec l'implantation de trois filiales en Belgique et en Espagne, et un réseau de distribution en Suisse. En Juin 2006, Itool Systems a rejoint le groupe EBP en tant que filiale A.S.P..

2.2. MISE EN RELATION

Le démarrage du projet fit l'objet de contretemps successifs, dans la mesure où il nous était alors difficile de joindre Alexandre DANVY, Directeur de Projets chez EBP, qui a entretemps quitté la société, lors de nos premières prises de contact téléphoniques.

Une rencontre avec l'entreprise fut finalement convenue sur le site de Rambouillet Vendredi 29 Octobre 2010. Au cours de cette réunion, à laquelle se sont rendus Erwan JEGOUIC et Marcel Cyrille LAMENU, la thématique du projet, consistant en le développement d'un « **Tableau de bord mobile - Indicateur de performances d'entreprise** », et plus précisément d'une application portable sous Android, pour la téléphonie mobile, nous fut présentée en détails.

Cet entretien nous a permis de délimiter le sujet, fixer les objectifs principaux à atteindre, les outils à utiliser ainsi que la marge de manœuvre qui nous sera accordée pour mener le projet à bien.

Début Janvier 2011, Pierre AUBIN, alors nouvellement responsable de la Direction de Projets, devint notre interlocuteur quant à l'encadrement et l'évolution du projet.

Eric NAVARRE, développeur au sein de l'équipe chargée du projet en interne, joua ensuite le rôle d'interlocuteur technique quant aux spécificités structurelles de l'application et des services auxiliaires à exploiter.

Ses multiples interventions et renseignements techniques nous ont permis d'avancer considérablement la phase de développement du projet, parallèlement à la construction du service web en interne ; puis de finaliser l'application dans un dernier temps.

3. « TABLEAU DE BORD MOBILE »

3.1. CAHIER DES CHARGES

3.1.1. DESCRIPTION DE LA DEMANDE

3.1.1.1. OBJECTIFS

La satisfaction de la demande implique de concevoir et développer une application intuitive, portable et efficace, permettant à un chef d'entreprise de consulter à tout moment, depuis une Plate-forme Mobile Web, en l'occurrence, son *Smartphone*, les critères chiffrés clés de sa société.

Il s'agit plus particulièrement de visualiser graphiquement en temps réel les différentes performances de son entreprise, les statistiques de son activité ou celle de ses clients, par le biais de la plate-forme Android.

Notre travail est quant à lui focalisé du côté client, et donc de l'application utilisée sur le *Smartphone*.

3.1.1.2. PRODUIT DU PROJET

Caractéristiques majeures de l'application

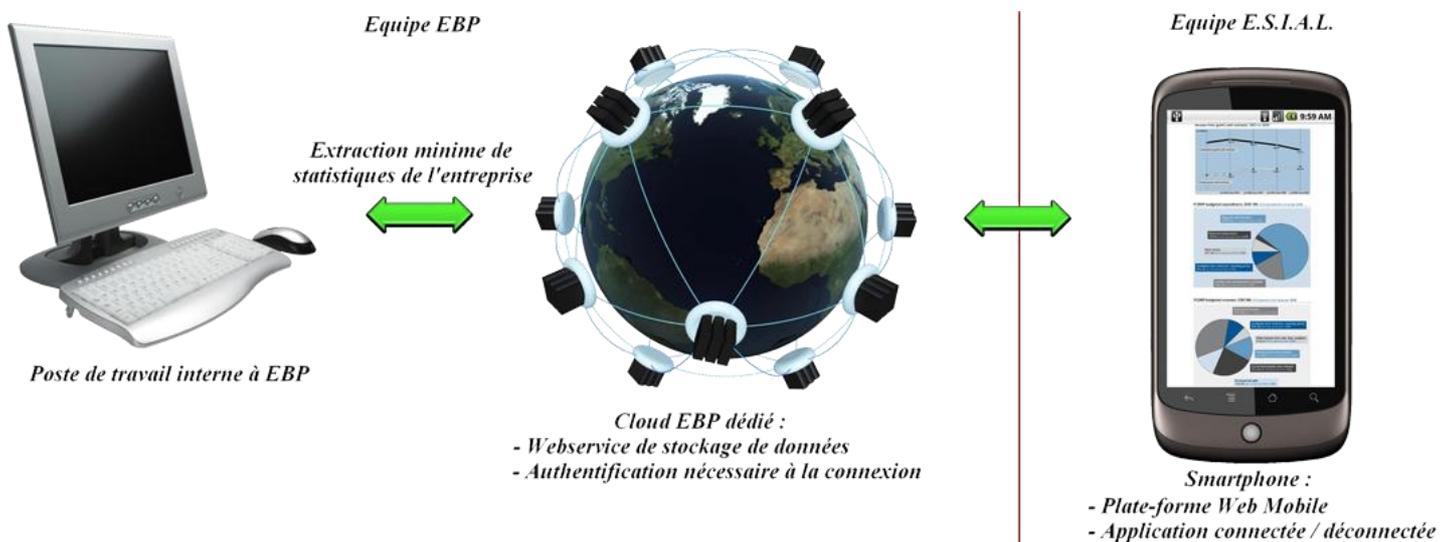


Figure 1 : Structure du projet

- La décision d'opter pour un *Smartphone* comme support de l'application est motivée par sa restriction à une lecture seule des données à afficher. La diffusion des informations propres à l'entreprise est ainsi limitée à une simple consultation. On parle effectivement de *Viewer*.
- Compte tenu de l'environnement matériel avec lequel elle interagit, il est essentiel pour l'application de sécuriser la communication des données de l'entreprise, à partir de la plateforme de stockage en ligne, ou service web, servant de base au *Cloud Computing* interne opté par l'entreprise, jusqu'au *Smartphone* utilisé, notamment en exigeant de l'utilisateur qu'il s'authentifie à sa connexion au service web. Des solutions alternatives grand public (potentiellement Windows Skydrive ou Google Docs) seront envisagées en option.
- Il est fondamental que les graphiques et tableaux affichés réagissent dynamiquement aux actions de l'utilisateur, en lui offrant un panel de fonctionnalités de consultation spécifique plus détaillée, selon les valeurs sélectionnées à partir de l'affichage par défaut.
- Le recours à un service web, que l'on détaillera davantage dans la section 3.2.1., s'explique par le fait qu'il s'agit d'un procédé de stockage simple, et à la portée de l'utilisateur lambda, dans la mesure où il comporte 3 fonctionnalités intuitives principales :
 - Fonction **put** : EBP se charge de déposer des fichiers contenant les valeurs chiffrées statistiques, à destination des clients possédant une licence d'utilisation de l'application
 - Fonction **get** : Le client récupère les fichiers situés sur le service web et correspondant aux données qu'il souhaite consulter
 - Fonction **list** : Le client consulte la liste de l'ensemble des fichiers disponibles sur le service web, distingués par secteurs, puis par types

Par ailleurs, celui-ci rend possible, d'un point de vue purement technique, une manipulation asynchrone de l'application, qui peut ainsi fonctionner que les acteurs en entrée et sortie soient connectés ou non, et, par conséquent, afficher indifféremment les données sélectionnées par l'utilisateur.

L'utilisation d'une telle plateforme de stockage, au demeurant accessible indifféremment à tout client doté d'une licence d'exploitation, constitue une approche nouvelle destinée à évaluer dans un premier temps la viabilité du processus de mise en ligne de données confidentielles, puis, dans un second temps, son acceptabilité par le client final une fois pourvu des ressources matérielles et logicielles requises pour le bon usage de l'application.

Ressources exploitées par l'application

La lecture des données à travers Android se fait à partir de fichiers X.M.L. disponibles en ligne depuis le service web avec lequel le *Smartphone* communiquera.

Ils stipulent le type de statistiques qu'ils contiennent, ainsi que le mode de visualisation graphique associé.

Les fichiers X.M.L. utilisés sont générés en interne par l'équipe de développeurs d'EBP située côté serveur, divers types d'exemples nous ayant été alors fournis afin de générer un module de traitement adapté.

La recherche des données via le service web par l'utilisateur est facilitée par le fait que celles-ci sont regroupées au sein d'une liste exhaustive fermée directement accessible au terme de l'authentification. Elles sont, par ailleurs, clairement identifiées nominativement par le biais des fichiers X.M.L. associés, selon leur nature, et le secteur concerné par les statistiques (chiffre d'affaires par client, chiffre d'affaires par article, etc.).

Modules optionnels de l'application

- Affichage dynamique des données (notifications de l'utilisateur d'éventuelles mises à jour des données consultées, de la progression visuelle du téléchargement des données en temps réel, rafraîchissement systématique de l'affichage au terme des mises à jour).
- Sauvegarde de l'état de l'application à sa fermeture, de manière à pouvoir le restituer tel quel à son prochain lancement.
- Simulation d'une présence permanente de l'application à travers une gestion multitâche de l'environnement.
- Libération des ressources mémoires utilisées lors du basculement de l'application en tâche de fond, afin de garantir une bonne fluidité d'utilisation du Smartphone en présence d'une instance de l'application.
- Aide contextuelle de l'application décrivant brièvement les fonctionnalités existantes.

3.1.1.3. CRITERES D'ACCEPTABILITE ET DE RECEPTION

Nous listons ci-après les aspects de l'application fondamentaux à la satisfaction potentielle de l'utilisateur final équipé d'un *Smartphone* muni de celle-ci :

- **Économe** du point de vue des ressources physiques à exploiter (consommation de la batterie, du processeur, des mémoires vive et physique, volume et importance des flux de données transférées puis manipulées, algorithmes, calculs et variables utilisés au sein du programme, etc.).
- **Rapide** (vitesse d'exécution, d'affichage, de rafraîchissement, de l'ordre de 10 ms, absence de ralentissements ou d'effets de bords impactant l'utilisation globale du *Smartphone*, etc.).
- **Fiable** (conformité de l'exécution aux commandes de l'utilisateur, et de l'affichage des données aux valeurs réelles).
- **Ergonomique** (esthétique, confort visuel, lisibilité des données, etc.).
- **Intuitive** (simplicité de navigation, d'utilisation des diverses fonctionnalités présentes, etc.).
- **Sécurisée** (chaque accès au service web est systématiquement accompagné d'une vérification des identifiant et mot de passe saisis initialement par l'utilisateur, les échanges d'informations émanant de l'entreprise sont ainsi gardés confidentiels à tout moment)

3.1.2. CONTRAINTES

3.1.2.1. CONTRAINTES DE COUT

Le projet industriel étant réalisé dans le cadre de notre scolarité à l'E.S.I.A.L., et découlant de ce fait d'un accord bilatéral entre cette dernière composante, et EBP; l'ensemble des contraintes budgétaires concernant la logistique et les moyens à mettre à notre disposition incombent directement à EBP.

Le déroulement du projet ne sera par conséquent soumis à aucune restriction de coûts de notre côté, dans la mesure où nous sommes bornés à la conception puis le développement de l'application.

3.1.2.2. CONTRAINTES DE DELAI

Le projet doit être clôturé fin Mars, de manière à remettre, d'une part l'ensemble de la documentation technique explicitant le projet ainsi que les descriptifs du sujet, en anglais et en français ; puis, d'autre part, présenter finalement l'ensemble du travail accompli lors de la soutenance finale parallèlement à la remise du produit livrable à l'industriel :

- **Jeudi 17 Mars 2011 : Remise du rapport final, des annexes techniques, du résumé, et de l'abstract**
 - Les éventuelles annexes techniques correspondent aux divers documents réalisés au cours du projet (spécification, conception, programmation, etc.).
 - Le résumé, d'environ 1 page, et l'abstract, d'environ ½ page, font apparaître les différents partenaires du projet, la chronologie du projet, les objectifs, les résultats obtenus, la démarche suivie, les acquis et l'éventuelle poursuite en stage.

- **Jeudi 24 Mars 2011 : Soutenance finale en français**

D'environ 50 minutes, elle présentera, parallèlement à la conduite du projet, l'ensemble des aspects et des fonctionnalités de l'application, les différentes batteries de tests effectuées, ainsi que les éventuelles difficultés rencontrées, les solutions adoptées afin de les surmonter, les modifications apportées, et enfin les possibles développements complémentaires.

3.1.2.3. CONTRAINTES TECHNIQUES

En complément des critères d'évaluation des performances inhérentes à l'application susmentionnés, le développement de l'application devra respecter certaines perspectives pratiques essentielles :

- L'application doit respecter l'intégrité structurelle d'Android, et donc mettre à contribution l'ensemble des dispositifs et fonctionnalités (qu'ils soient physiques ou logiciels) natifs et initialement intégrés, et ne pas tenter d'émuler des fonctions existant sur d'autres plates-formes.
- La visualisation graphique des statistiques, une fois récupérées sur le *Smartphone*, doit pouvoir fonctionner de manière autonome, c'est-à-dire indépendamment du contenu du service web, que l'utilisateur y soit encore connecté ou non. Elle requiert néanmoins que l'application demeure connectée à Internet, en raison des contraintes structurelles de la bibliothèque Google Charts utilisée.
- Les probables temps de latence engendrés par la consultation du service web depuis le *Smartphone* doivent être considérés au cours du développement, et, de ce fait, être gérés par l'application (notification de l'utilisateur des délais de téléchargement des données, rendu de la main à l'utilisateur lors du téléchargement, notification de la disponibilité des données une fois le téléchargement effectué, etc.).
- L'affichage doit prévoir d'associer et adapter automatiquement le type de graphiques ou de tableaux exploitables par Android aux types de données à afficher (lisibilité, niveau de détail, etc.).
- Le développement de l'application doit gérer de manière permissive les flux d'entrée, de façon à ne pas bloquer l'application en cas de conflits (notamment en ce qui concerne des données inexistantes, des types de données non gérés, la syntaxe du fichier X.M.L. d'entrée, etc.), mais permettre un affichage par défaut systématique notifiant, le cas échéant, l'utilisateur de la marche à suivre.
- Les données, une fois téléchargées par l'application, seront sauvegardées au sein des mémoires cache et locale du *Smartphone*. Toutefois, pour une meilleure disponibilité, il reste à déterminer s'il est plus judicieux de les stocker également sur la mémoire physique, une carte S.D. amovible, ou à distance.

3.1.3. PLANIFICATION

3.1.3.1. PLANIFICATION INITIALE

Conformément aux normes de la gestion de projet, nous avons initialement fixé une planification de la succession des tâches à accomplir au cours du projet, de façon à respecter les contraintes de délai précédemment abordées, tout en prenant en considération le stade, alors révolu, de prise de contact initiale.

Ces prévisions furent établies en partant de la date au plus tard, à partir de laquelle nous avons évalué régressivement le temps imparti à affecter, à l'aide d'estimations de la charge de travail (en heure * homme), à chaque stade :

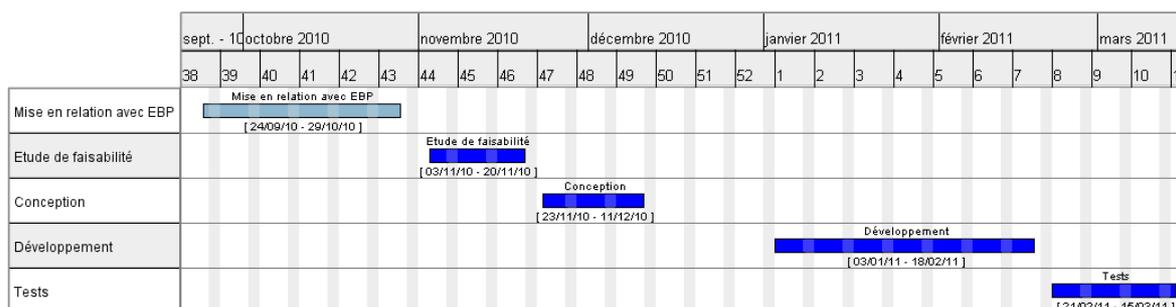


Figure 2 : Diagramme de Gantt initial

- Mise en relation avec EBP (50) : comparativement à la durée allouée sur le diagramme de Gantt, la durée considérée s'appuie sur les grandes difficultés de prise de contact initiale avec l'entreprise, seul le temps effectif utile ayant été ici comptabilisé.
- Etude de faisabilité (100) : cette phase tend à vérifier si le projet est faisable d'un point de vue technique dans notre cas.
Au cours de celle-ci ont été élaborés une note de cadrage visant à récapituler l'organisation du projet telle que l'avions assimilée lors de notre rencontre avec EBP à Rambouillet, puis le cahier des charges développé ici.
- Conception (300) : une fois l'étude de faisabilité terminée, nous avons pu aborder la conception. De cette partie dépendant toute la suite des événements, nous avons décidé d'y accorder autant de temps que nécessaire afin de ne rien laisser au hasard et ne pas prendre le risque d'en être alors pénalisés par la suite.
- Développement (250) : durée estimée en fonction des technologies à utiliser ainsi que de la complexité du sujet. Il y a pour nous une nécessité de réaliser des formations, car le développement sous Android, ainsi que les bibliothèques rattachées, est une compétence dont nous ne disposons pas. Nous sommes en revanche déjà tous familiarisés avec l'environnement eclipse, et le langage JAVA.
- Tests (300) : la phase de tests étant très importante pour le bon fonctionnement de l'application, nous avons choisi d'y consacrer une partie conséquente du projet.

3.1.3.2. PLANIFICATION REVISEE

Le déroulement du projet ayant néanmoins grandement évolué par rapport à nos prévisions, il fut nécessaire de modifier notre planification en conséquence :

Notre soutenance intermédiaire en anglais, ayant eu lieu Jeudi 16 Décembre 2010, nous a permis de présenter notre étude de faisabilité, à la lumière du cahier des charges fonctionnel fourni par EBP.

En obtenant l'aval d'EBP pour la poursuite du travail après leur avoir transmis les différents documents rédigés depuis le lancement du projet industriel, mais plus particulièrement, les divers diagrammes détaillant l'architecture de l'application élaborés jusqu'alors ; il nous fut possible de clôturer notre phase de conception et alors embrayer sur la plus consistante étape de développement.

Ayant formulé début Décembre le souhait d'être mis en relation avec l'équipe d'EBP chargée du développement en interne, mais étant alors sans nouvelle début Janvier, notre prise de contact téléphonique s'établit dès lors avec Pierre AUBIN, remplaçant Alexandre DANVY à la Direction de Projets d'EBP.

La communication reprit dès lors plus fréquemment, nous permettant ainsi, d'une part, d'être mis en relation avec Eric NAVARRE, développeur au sein de l'équipe ; et, d'autre part, de débiter à proprement parler le développement, moyennant quelques formations supplémentaires aux modules inhérents à **Android**, combinés avec l'environnement de travail **eclipse** pour lequel nous avons opté.

Notre avancement relatif à la gestion de la communication avec le service web ayant été modulé par les renseignements obtenus de l'équipe de développement interne, la phase de développement s'est finalement étendue outre mesure, ce pourquoi il nous a fallu opérer simultanément la finalisation de l'application parallèlement à la phase de tests fonctionnels.

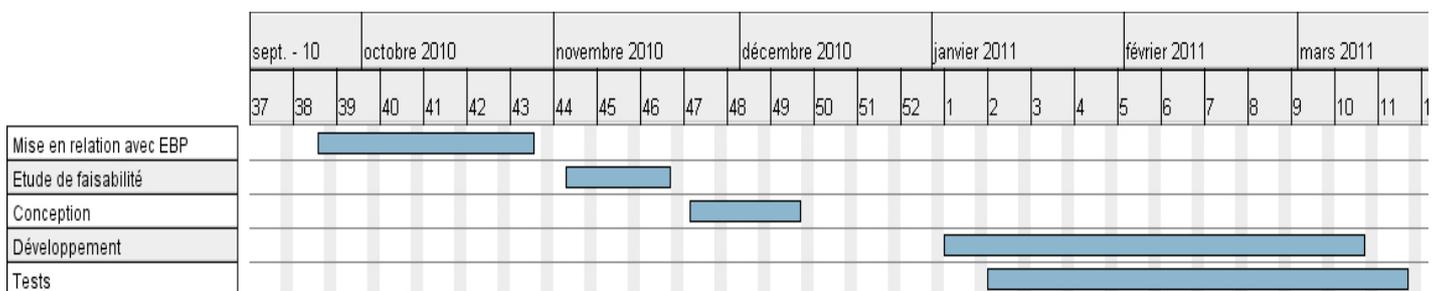


Figure 3 : Diagramme de Gantt révisé

3.1.4. CONDUITE DU PROJET

En ce qui concerne le fonctionnement de l'équipe de travail, nous avons réparti différentes tâches spécifiques à chaque membre, de la manière suivante :

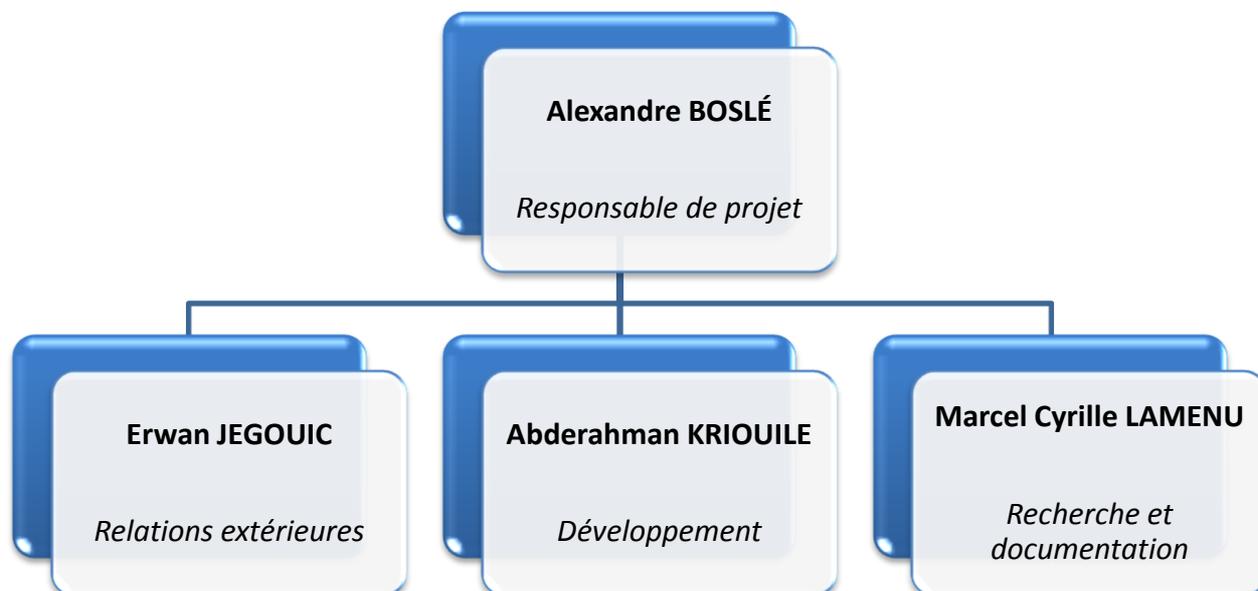


Figure 4 : Organigramme de l'équipe

- Responsable de projet :
 - Responsabiliser les membres de l'équipe
 - Gérer les membres de l'équipe, afin d'assurer la paix sociale
 - Veiller au bon déroulement du projet
- Relations extérieures :
 - Prendre contact avec les différentes parties prenantes
 - S'enquérir des disponibilités de chacun pour fixer les dates d'entretiens, éventuels déplacements sur sites ou événements officiels
 - Assurer une communication régulière de l'évolution du travail et des différents documents inhérents au projet
- Développement :
 - Répartir les tâches de conception et développement entre les différents membres de l'équipe
 - Gestion développement en parallèle des différentes parties du projet
 - Communiquer sur les choix techniques avec les développeurs EBP
- Recherche et documentation :
 - Rechercher les informations sur les technologies et les bibliothèques à utiliser
 - Les documenter
 - Les rendre accessible sur la forge afin que les autres puissent y avoir accès

Ces tâches sont toutefois des assignations principales, l'intégralité de l'équipe contribuant aux différentes phases de la réalisation du projet.

Nous fonctionnons par réunions hebdomadaires le Lundi à partir de 14h, au cours du créneau consacré au projet industriel.

À la fin de chaque réunion sont fixées le plan de la réunion suivante, les tâches à réaliser entretemps, ainsi que la répartition de celles-ci au sein de l'équipe.

Un compte-rendu est systématiquement réalisé à l'issue de chaque réunion puis mis en ligne sur une plate-forme commune au groupe, et hébergée sur la forge de l'école : <http://forge.esial.uhp-nancy.fr>.

Il est également transmis par courrier électronique, parallèlement aux éventuels documents supplémentaires produits, à l'ensemble des membres de l'équipe, ainsi qu'à nos encadrants universitaire et industriel.

Nous avons choisi de centraliser l'ensemble de nos données sur cette plate-forme, sur laquelle figure également l'intégralité de notre documentation, réalisée durant l'étude de faisabilité ainsi que la conception, et portant sur les technologies à mettre en œuvre.

3.1.5. RESSOURCES

3.1.5.1. RESSOURCES HUMAINES

Côté serveur

Une équipe dédiée au sein d'EBP est chargée du développement des aspects internes de l'application, sous la responsabilité du directeur de projets, Pierre AUBIN.

Eric NAVARRE, développeur de l'équipe en interne, s'est chargé de nous fournir les renseignements techniques requis tout le long du projet, à partir de Janvier 2011.

Côté client

Nous sommes chargés du développement de l'application Android destinée à être implantée sur le *Smartphone* de l'utilisateur final.

3.1.5.2. RESSOURCES LOGICIELLES

Le développement sera réalisé sous **eclipse** et **Android S.D.K.**, dans la mesure où nous avons tous préalablement travaillé au sein du premier à l'occasion de nombreux projets de développement, et où le second constitue une ressource officielle, et du reste prolifique en tutoriaux nous permettant d'appréhender plus aisément **Android**.

L'application développée n'est destinée au lancement que sur une cible bien précise : le système d'exploitation Android équipant le Smartphone de l'utilisateur final. Les outils de développement utilisés, en revanche, et notamment le langage **JAVA**, sont potentiellement exploitables par différents systèmes d'exploitation (Mac O.S., Linux, Windows, etc.), rendant ainsi le dispositif aisément portable sur d'autres plateformes, au besoin.

Plate-forme de développement

ANDROID doit son nom à la start-up du même nom spécialisée dans le développement d'applications mobiles, que Google a rachetée en Août 2005, nom qui vient lui-même d'« androïde », qui désigne un robot construit à l'image d'un être humain. Le logiciel, qui avait été surnommé gPhone par les rumeurs de marchés, est proposé de façon gratuite et librement modifiable aux fabricants de téléphones mobiles, ce qui facilite son adoption. Le gPhone a été lancé en Octobre 2008 aux États-Unis dans un partenariat de distribution exclusif entre Google et T-Mobile.

Android est un système d'exploitation fondé sur un noyau Linux, il comporte une interface spécifique, développée en JAVA, les programmes sont exécutés via un interpréteur J.I.T. ; interface outre laquelle il est toutefois possible de passer, moyennant un travail de portabilité alors plus conséquent.

Langage de modélisation

En ce qui concerne la conception, nous avons arrêté notre choix sur le langage **U.M.L.** pour les raisons suivantes :

- Langage intégrable dans eclipse, qui constituera notre plate-forme de développement JAVA pour Android.
- Le type de données manipulées en entrée de l'application consistera en des fichiers X.M.L. et non une base de données, ce pourquoi U.M.L. est plus approprié.

Bibliothèques externes

La réalisation de ce programme a nécessité l'utilisation de bibliothèques externes, notamment en ce qui concerne l'utilisation du service web ainsi que l'analyse syntaxique de fichiers X.M.L. contenant les données à afficher.

S.A.X. Parser

Simple A.P.I. for X.M.L., ou S.A.X., est une A.P.I. générale pour la lecture d'un flux X.M.L.. Il en existe notamment une implémentation dans le langage JAVA.

S.A.X. est une A.P.I. événementielle, le traitement étant réalisé au fil de la lecture du flux entrant. Si l'on considère le flux X.M.L. entrant, les événements sont caractérisés par la lecture de balises ouvrantes ou fermantes. Le traitement d'un flux X.M.L. par ce simple processus extrêmement puissant permet ainsi d'utiliser un fichier de configuration X.M.L., dont l'avantage est son aptitude contextuelle, laquelle est difficile à représenter dans un fichier de propriétés classique.

kSOAP2

kSOAP2-android est un projet en accès libre, ou *Open Source*, pour Android, qui fournit une solution simple et performante de la librairie *Simple Object Access Protocol*, ou S.O.A.P.. Il permet la transmission de messages entre objets distants, en autorisant un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur. Le transfert s'opère le plus souvent à l'aide du protocole H.T.T.P., un autre protocole tel que S.M.T.P. pouvant également assurer ce rôle.

Le protocole S.O.A.P. est composé de deux parties :

- une enveloppe, contenant des informations sur le message lui-même afin de permettre son acheminement et son traitement
- un modèle de données, définissant le format du message, c'est-à-dire les informations à transmettre

S.O.A.P. a été initialement défini par Microsoft et I.B.M., mais est devenu une référence depuis une recommandation du W3C, utilisée notamment dans le cadre d'architectures de type *Service Oriented Architecture* pour les services web.

Avantages :

Utiliser S.O.A.P. via H.T.T.P. facilite la communication et évite les problèmes de proxys et pare-feu par rapport à des technologies plus anciennes.

S.O.A.P. est notamment :

- Adaptable à différents protocoles de transport
- Indépendant de la plate-forme
- Indépendant du langage
- Extensible

Inconvénients :

En raison de la quantité d'informations qu'impose le format X.M.L., S.O.A.P. peut alourdir considérablement les échanges par rapport à des *middlewares* comme C.O.R.B.A. ou I.C.E., ce qui n'est pas forcément un handicap quand les volumes de données transités par S.O.A.P. sont faibles par rapport au volume total de données échangées.

3.2. CONCEPTION

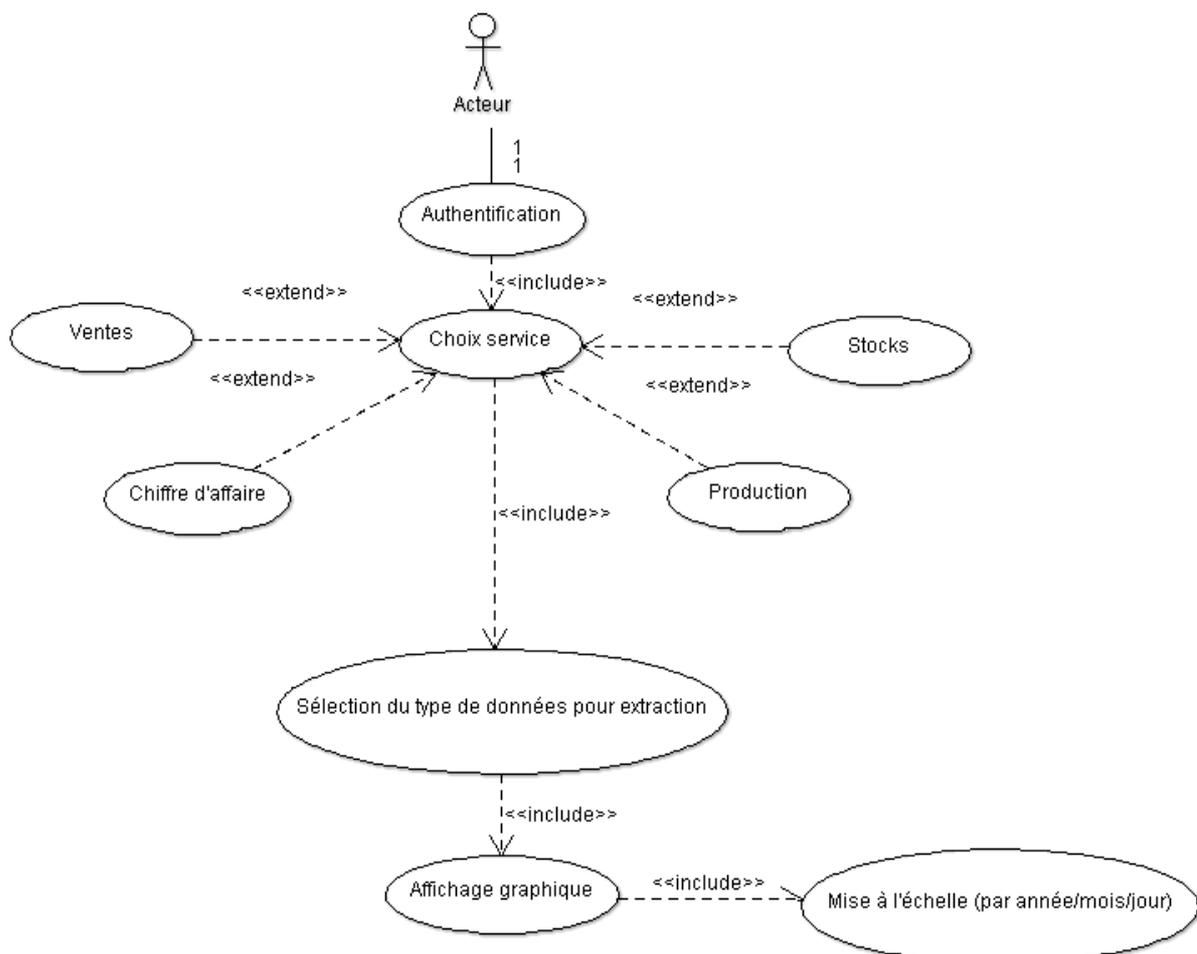
Les éléments relatifs à la conception de l'architecture globale du projet sont décrits ci-dessous. Nous n'avons cependant pas joint tous les diagrammes, certains atteignant un niveau de détail technique sujet à changement selon les évolutions optées par l'équipe de développement interne ; seuls les plus importants figurent ci-dessous.

3.2.1. CAS D'UTILISATION

Les cas d'utilisations permettent de décrire l'interaction entre l'acteur, i.e. l'utilisateur, et le système, i.e. l'application localisée sur le *Smartphone*. L'idée forte est de dire que l'utilisateur d'un système a un objectif. Le cas d'utilisation est une description des interactions qui vont permettre à l'acteur d'atteindre son objectif en utilisant le système.

L'opérateur doit initialement s'authentifier auprès du service web depuis l'application, pour ensuite pouvoir choisir une catégorie de statistiques (chiffre d'affaires, ventes, production, stocks, etc.).

Une fois la sélection faite, il est possible de l'affiner en choisissant le type précis des données à extraire. L'application se charge alors de l'affichage graphique, offrant à l'utilisateur deux options complémentaires : on peut alors redimensionner les graphiques obtenus (par année, mois ou jour) ; ou basculer d'une période à une autre.



3.2.2. DIAGRAMMES D'ACTIVITES

Les diagrammes d'activités présentent l'ensemble des algorithmes exploités par les fonctions principales de l'application. Ils mettent également en évidence la structure de chaque processus de base, de son lancement à son terme.

Nos deux diagrammes principaux illustrent les fonctionnalités permettant d'afficher l'interface de recherche, et les graphiques produits à partir des données choisies.

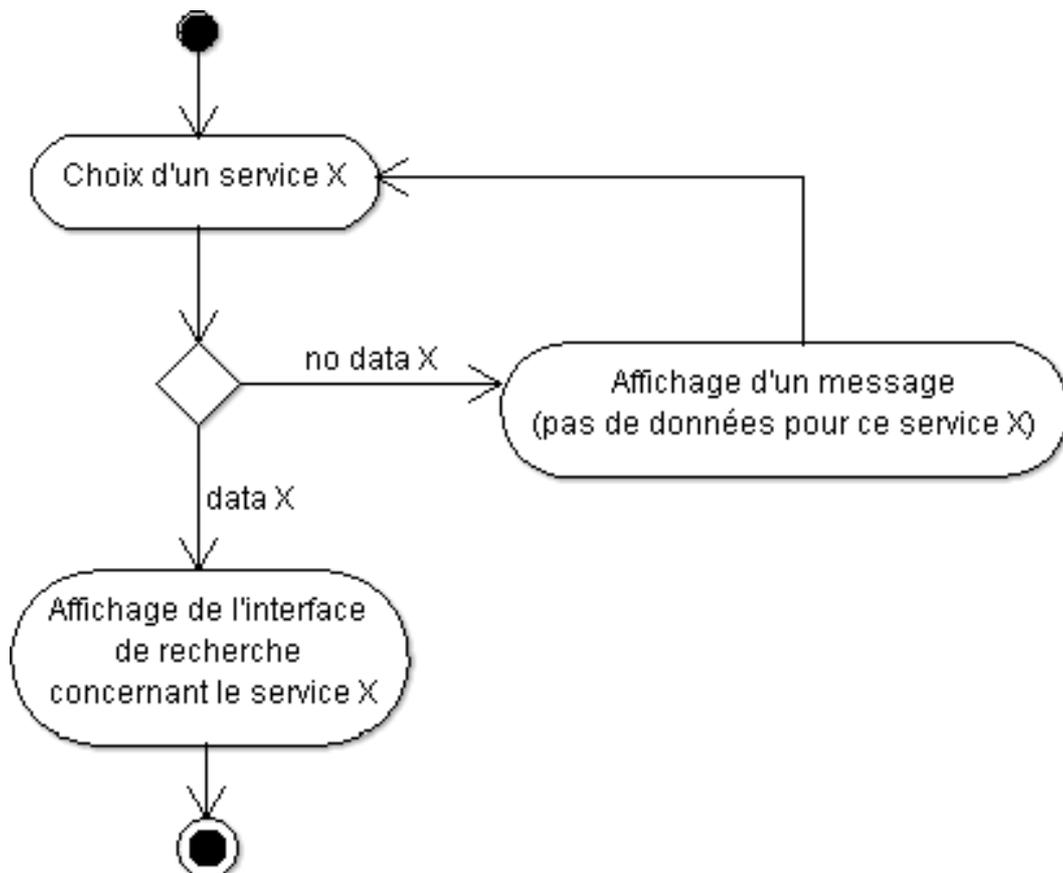
3.2.2.1. CHOIX D'UN SERVICE

L'algorithme de choix d'un service se modélise de la façon suivante :

Pour que l'utilisateur puisse accéder à l'interface de sélection, il lui faut tout d'abord choisir un service X donné.

Un test est alors réalisé par l'application afin de vérifier s'il existe des données correspondantes sur le service web.

L'application affiche, le cas échéant, directement l'interface de choix associée au service X. Dans le cas contraire, un message en notifie l'utilisateur.



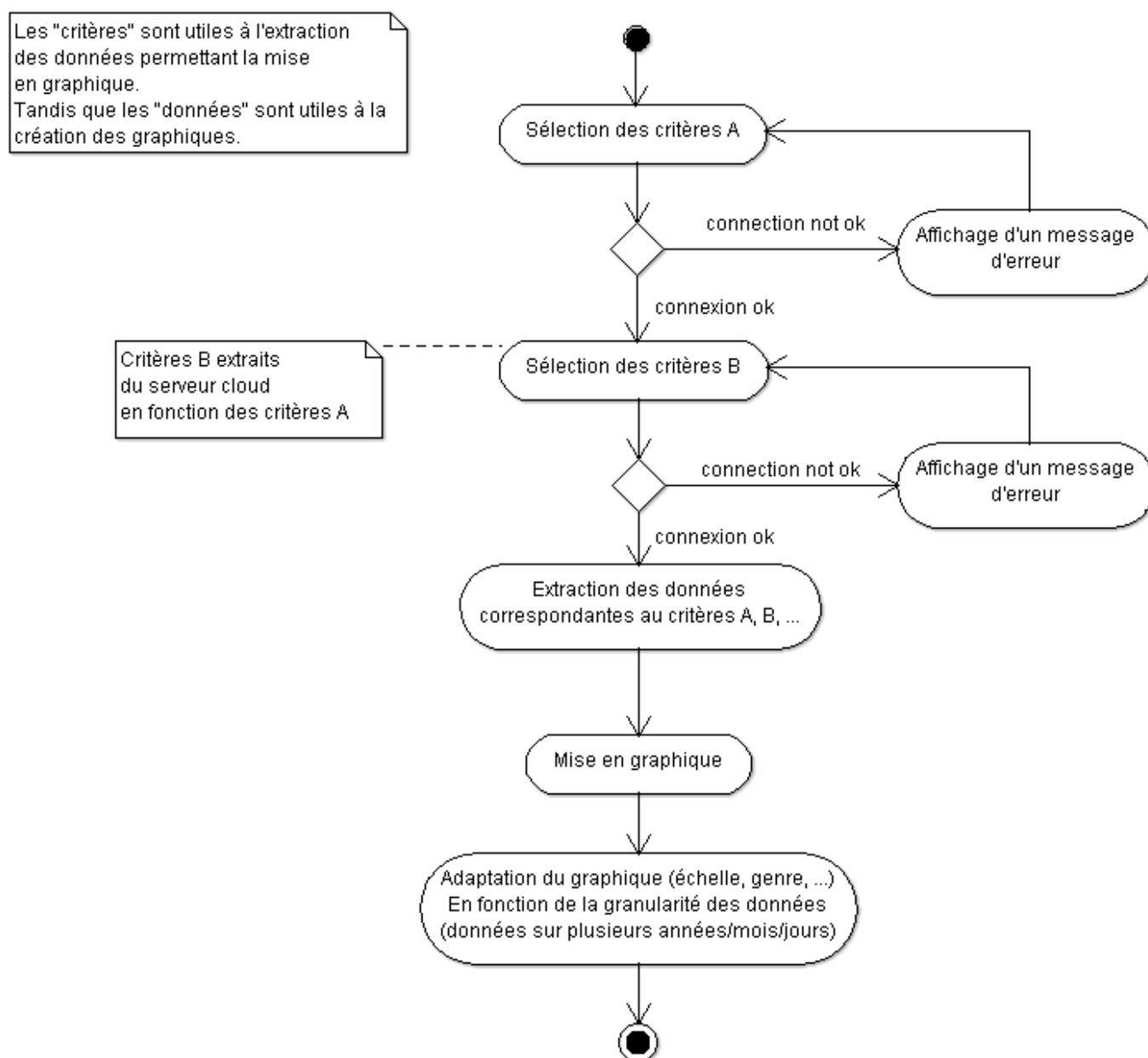
3.2.2.2. CHOIX DES DONNEES ET AFFICHAGE GRAPHIQUE

La sélection des critères de recherche se faisant par l'intermédiaire de listes, il est inutile pour l'utilisateur de saisir de données.

Ces listes contiennent les données extraites du service web en fonction du secteur sélectionné.

L'utilisateur affine sa recherche au fur et à mesure, en ce sens où, suite à chaque sélection, l'application tente de se connecter au service web, afin d'afficher les critères de choix suivants en conséquence.

Une fois tous les critères sélectionnés, les données sont extraites afin d'en produire les graphiques associés, qui peuvent alors être mis à l'échelle par l'utilisateur.



3.2.3. DIAGRAMMES DE SEQUENCES

Les diagrammes de séquences nous permettent de modéliser différents scénarii, et ceci du point de vue de plusieurs différents acteurs en interaction avec l'application.

3.2.3.1. AUTHENTIFICATION

Le diagramme d'authentification de l'utilisateur depuis son *Smartphone* est constitué de quatre entités distinctes :

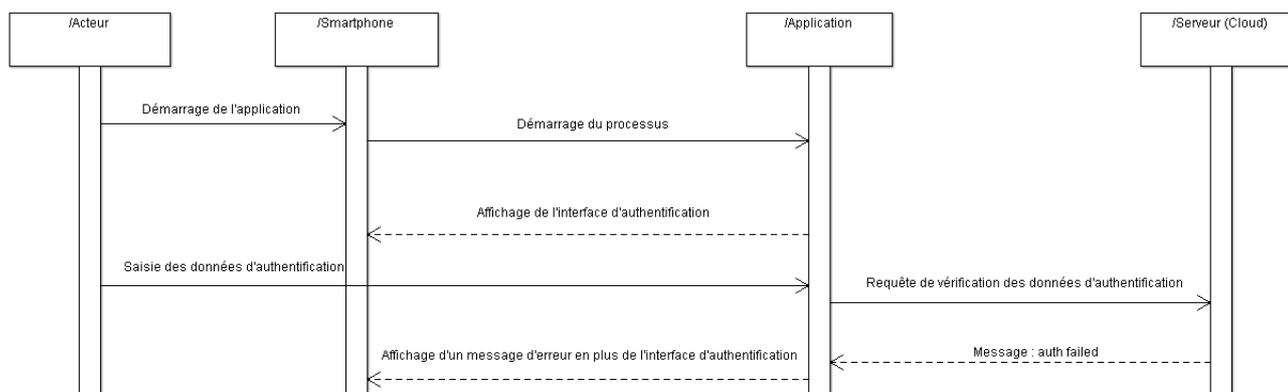
- L'opérateur
- L'application
- Le *Smartphone* sur lequel est lancée cette dernière
- Le serveur situé sur le service web contenant les données

L'opérateur commence par exécuter l'application depuis son téléphone, qui lance alors le processus d'authentification.

Lorsque le programme est actif, il vérifie si une connexion au service web est possible, auquel cas le serveur envoie un message de confirmation, et le programme peut ainsi afficher l'interface d'authentification.

L'opérateur doit alors saisir ses identifiant et mot de passé, que l'application doit valider.

S'ils sont corrects, le serveur confirme la requête de l'utilisateur, et le programme affiche enfin l'interface de recherche.



3.2.3.2. AFFICHAGE GRAPHIQUE

Seules les trois entités principales figurent sur le diagramme de visualisation graphique, afin de réduire la complexité du schéma :

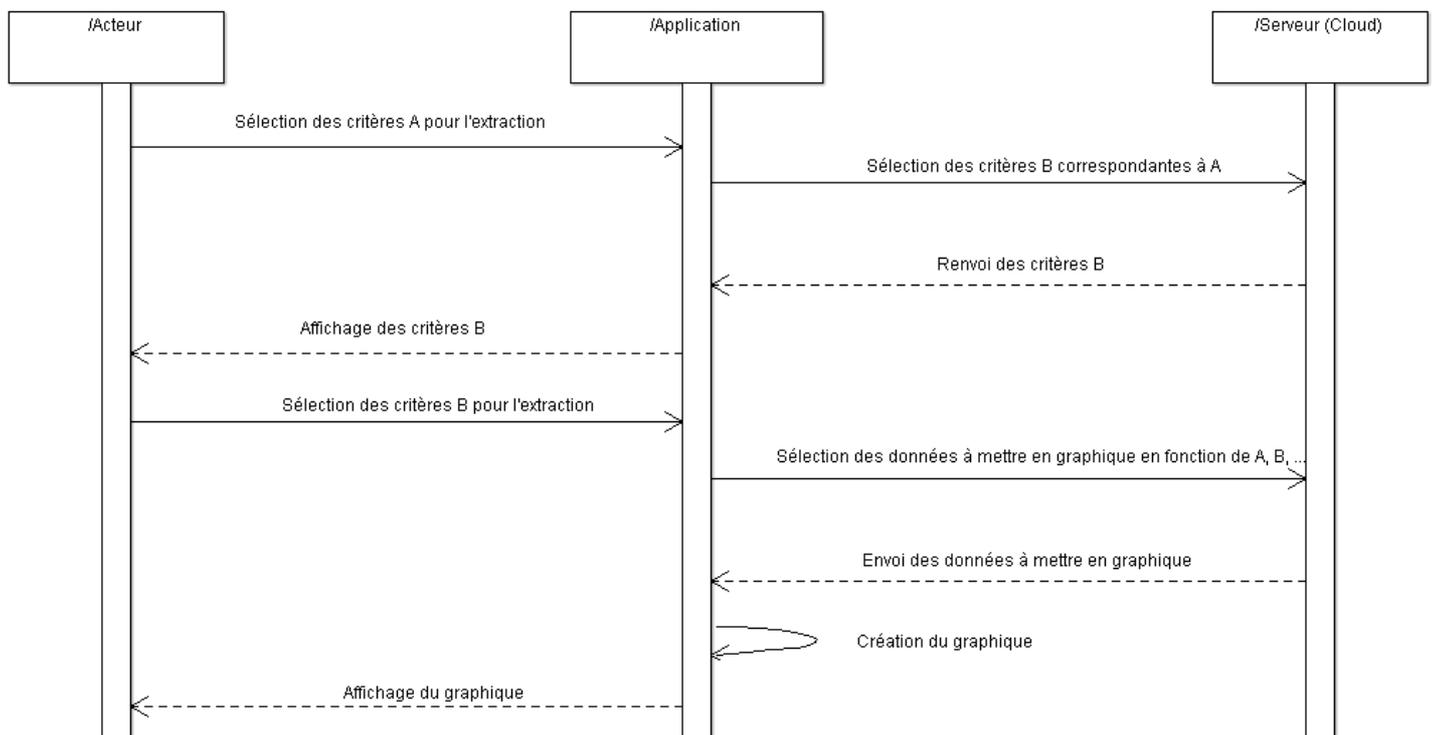
- L'opérateur
- L'application
- Le serveur

L'utilisateur doit tout d'abord choisir un service de manière à ce que l'application vérifie ensuite s'il existe des données correspondantes sur le serveur.

Si tel est le cas, le serveur confirme sa requête, et le programme affiche une liste déroulante contenant le premier type de critères disponibles pour l'extraction de données.

L'utilisateur sélectionne ainsi des critères de recherche de type A, l'application vérifiant alors s'il existe des critères de type B correspondants aux critères A.

Enfin, lorsque l'utilisateur a finalisé son choix de critères, l'application envoie une requête au serveur afin d'extraire les données sélectionnées, pour finalement élaborer les graphiques correspondants, et les afficher.



3.3. DEVELOPPEMENT

La phase de développement fut abordée avec la conception schématique de l'architecture, largement revue par la suite, des classes de l'application.

Une fois le modèle à disposition, nous avons isolé les modules indépendants à implémenter pour pouvoir les travailler séparément, à l'appui des bibliothèques externes auxquelles elles recourent, le cas échéant.

Le composant dédié à la lecture et le traitement des fichiers X.M.L. à lire en entrée fut le premier abordé, étant effectivement en amont de la transmission des données à afficher. Le développement fut réalisé dans un premier temps parallèlement à la production de modèles de fichiers fournis par EBP.

Les affichages de l'interface graphique et des données à traiter ont progressé en parallèle, le premier fixant en effet les critères d'affichage des fenêtres sur le Smartphone, le second fournissant des batteries de test au premier.

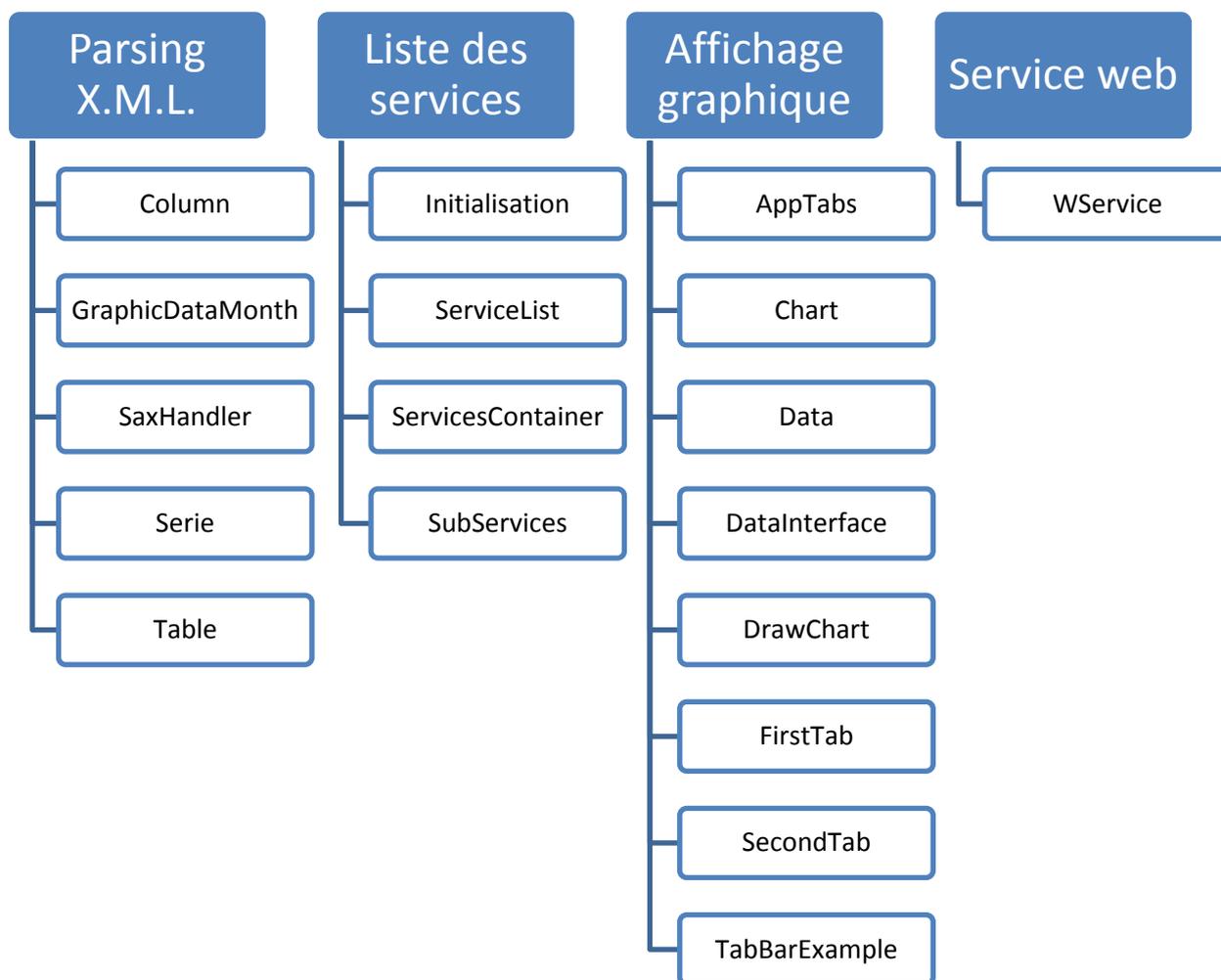


Figure 5 : Architecture des classes pour chaque module

3.3.1. SERVICE WEB

Un **service de la toile**, ou **service web**, comme on le désigne ici, est un programme informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués. Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications, ou machines, sans intervention humaine, et en temps réel.

Le concept a été précisé et mis en œuvre dans le cadre de *Web Services Activity*, au W3C, particulièrement avec le protocole S.O.A.P.. Associé avec les Échanges de Données Informatisés, ou E.D.I., le consortium e.b.X.M.L. l'a utilisé pour automatiser des échanges entre entreprises. Cependant, le concept s'enrichit avec l'approfondissement des notions de ressource et d'état, dans le cadre du modèle *Representational State Transfer*, ou R.E.S.T., et l'approfondissement de la notion de service, avec le modèle S.O.A..

Dans sa présentation la plus générale, un service web se concrétise par un agent, réalisé selon une technologie informatique précise, par un fournisseur du service. Un demandeur, à l'aide d'un agent de requête, utilise ce service. Fournisseur et demandeur partagent une même sémantique du service web, tandis qu'agent et agent de requête partagent une même description du service pour coordonner les messages qu'ils échangent.

Il existe plusieurs technologies derrière les services web :

- Les services web de type R.E.S.T. exposent entièrement ces fonctionnalités comme un ensemble de ressources identifiables, *Uniform Resource Identifier*, ou U.R.I., et accessibles par la syntaxe et la sémantique du protocole H.T.T.P.. Les services web de type R.E.S.T. sont donc basés sur l'architecture du web et ses standards de base : H.T.T.P. et U.R.I..
- Les services web *W.S.** exposent ces mêmes fonctionnalités sous la forme de services exécutables à distance. Leurs spécifications reposent sur les standards S.O.A.P. et W.S.D.L. pour transformer les problématiques d'intégration héritées du monde *middleware* en objectif d'interopérabilité.

Les standards *W.S.** sont souvent décriés, comme risquant de générer une course à la performance technologique. Toutefois leur robustesse dans le milieu des services entre professionnels est reconnue, et ils restent largement utilisés. Aussi préfère-t-on les faire évoluer.

Suite à nos différents entretiens téléphoniques avec Eric NAVARRE, qui nous ont permis d'échanger nos points de vue sur le fonctionnement du serveur et du traitement des données à afficher par le *Smartphone*, la structure du service web a été revue et globalement modifiée :

- La structure des fichiers X.M.L. correspondant aux données à afficher, et à lire en entrée du programme, nous a été communiquée par le biais de différents modèles desdits fichiers, nous présentant ainsi plusieurs approches de l'affichage des statistiques de l'entreprise.
- Le développement en interne des parties application et serveur du projet a été abordé, et se poursuit parallèlement à l'avancement du programme de notre côté, qui s'appuiera pour le moment sur des jeux de tests et des modèles de plate-forme distante locaux que nous développerons nous-mêmes.
- Les fonctionnalités de serveur de fichiers censées permettre de récupérer une grande variété de fichiers propres à l'entreprise, et non uniquement les fichiers X.M.L., envisagées par EBP ont finalement été abandonnées au profit d'une architecture de service web comprenant les trois fonctions principales suivantes :
 - **GetServiceDescriptors (user, pwd) :**

Récupérer la liste des différents services de l'entreprise dont il faudra pouvoir consulter les données, et qui sera retournée sous la forme d'un fichier X.M.L. contenant la liste de l'ensemble de ces services, caractérisés par les paramètres suivants :

 - Identifiant du service
 - Nom du service
 - Description du service (par exemple : "Gestion commerciale")
 - **GetDataDescriptors (user, pwd, serviceId) :**

Récupérer la liste des données disponibles pour l'un des services sélectionné, également retournée sous la forme d'un fichier X.M.L. contenant la liste des données en question, caractérisées comme suit :

 - Identifiant de la donnée
 - Nom de la donnée
 - Description de la donnée (par exemple : "Ventes de 2009")
 - **GetData (user, pwd, serviceId, dataId) :**

Récupérer les données précises à afficher, qui seront alors retournées sous la forme d'un fichier X.M.L. au même format que les fichiers d'exemple que nous a précédemment communiqués Eric NAVARRE.

- Les données que l'utilisateur souhaitera afficher ne seront pas filtrées sur le serveur en fonction de critères que celui-ci saisirait, mais seront disponibles d'entrée à partir d'une liste fermée qu'il pourra consulter sur la page d'accueil succédant à la page d'authentification.

Ces trois méthodes s'appuieront sur deux paramètres communs, à savoir le couple (identifiant, mot de passe), utilisé pour des raisons de sécurité, et permettant ainsi un accès limité aux statistiques de l'entreprise.

Ces paramètres à passer pour le test du service web sont respectivement :

- **user** : ESIAL
- **pass** : EBP2011

La classe `WebService` de l'application s'appuie sur la bibliothèque `kSOAP2`, et se charge de la gestion de la connexion du *Smartphone* au service web.

Elle se charge de comparer les couples (identifiant, mot de passe) avec le tableau des couples valides, et récupère les valeurs saisies pour l'authentification.

La section *Open Line* d'EBP nous a été également ouverte, afin d'accéder au Progiciel de Gestion Intégrée Ligne P.M.E. et au Module C.R.M., qui nous donnent un aperçu de la gestion de la chaîne commerciale de l'entreprise, et de la potentielle manipulation attendue du tableau de bord mobile

Bien que nous fournissant notamment une approche de la consultation des grandeurs statistiques d'EBP, et, en moindre mesure de l'interaction avec le service web, ces outils n'ont été consultés qu'à titre informatif et majoritairement d'un point de vue esthétique.

3.3.2. PARSING X.M.L.

La lecture des fichiers X.M.L. en entrée de l'application est réalisée à l'aide de l'A.P.I. `SAX`, et peut aussi bien prendre en paramètre un fichier X.M.L., local ou distant, dont on passe respectivement en paramètre le chemin dans le système de fichiers ou l'adresse U.R.L..

Centré principalement autour des bibliothèques `JAVAx.xml.parsers` et `org.xml.sax`, sur lesquelles il s'appuie, le module de *Parsing* permet l'analyse d'un fichier X.M.L. à travers la récupération de ses différentes balises et des données contenues à travers celles-ci, puis sa lecture sous **Android**, comme illustré ci-dessous pour un exemple arbitraire :

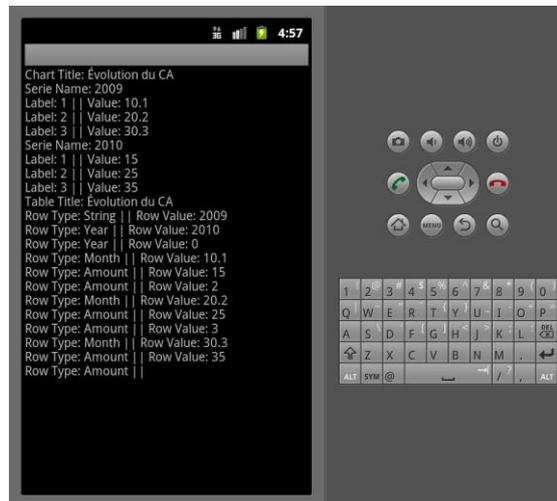


Figure 6 : Affichage du Parsing d'un fichier X.M.L. passé en paramètre

3.3.3. AFFICHAGE GRAPHIQUE

Conformément aux spécifications de fonctionnement de l'application, une fois les données récupérées une première fois auprès du service web et stockées sur le *Smartphone*, le programme est pleinement opérationnel, qu'il soit indifféremment connecté ou non à la plate-forme distante.

En revanche, l'affichage graphique des données sur le *Smartphone* s'effectue, quant à lui, à l'appui de l'A.P.I. **Google Charts**, et qui nécessite de la part du *Smartphone* d'être connecté sur Internet, de façon à pouvoir exploiter correctement les fonctionnalités de traitement graphique de l'A.P.I..

4. CONCLUSION

Nous sommes parvenus, en collaboration avec l'entreprise EBP, à réaliser un tableau de bord mobile fonctionnel, au terme de l'échéancier prévu pour le projet industriel.

À travers cette application, un utilisateur lambda est capable de suivre le pilotage des activités de son entreprise où qu'il soit, à condition de disposer d'un terminal mobile adéquat pouvant se connecter à un service web lui offrant, une fois l'authentification réussie, un large éventail de choix de services et d'informations.

Toutefois, afin d'améliorer l'application, nous aurions pu approfondir la description des services, et fournir à l'utilisateur des effets d'affichage dynamique correspondant à la mise à jour en temps réel du contenu du service web. Cependant, le temps (commencement tardif, attente de données techniques internes, etc.) et la complexité, du point de vue technique (découverte et utilisation de nouveaux outils, création de modules adéquats, etc.), de ces tâches ont été les principaux facteurs qui nous ont rebutés.

Ces fonctionnalités pourront cependant faire l'objet d'améliorations optionnelles, afin que tous les aspects de l'application puissent être réalisés, telle qu'elle a été présentée dans notre cahier des charges technique.

Ce projet nous a permis d'appliquer des connaissances techniques et managériales qui nous ont été inculquées au cours de ces trois années de formation à l'E.S.I.A.L., ainsi que de nous initier à avoir un aperçu sur l'entreprise et de la gestion d'un projet en groupe sur le long terme, dans l'optique de mener pleinement à bien notre stage de fin d'études.

Nous avons été confrontés à de nombreux problèmes, mais avons, dans la plupart des cas, abouti à une solution afin de les résoudre, ou d'apporter une correction alternative.

Par ailleurs, la conduite du projet nous a donné conscience de l'importance fondamentale à accorder, dès le commencement, à son approche sous l'angle de la gestion de projet. En effet, notre choix d'établir très tôt nos méthodes de travail, l'organisation de nos réunions, documentation, mise en commun des ressources, ainsi que l'affectation de rôles spécifiques à chacun, nous a permis de superviser le projet avec une plus grande facilité.

De plus, l'étude de ce sujet aura été une bonne occasion de découvrir et nous apporter de nouvelles connaissances quant à des outils dont nous ne soupçonnions pas nécessairement l'existence, et plus particulièrement concernant le domaine des systèmes d'exploitation utilisés par la téléphonie mobile, en l'occurrence, de type *Smartphone*, qu'outre le fait d'employer quotidiennement, nous sommes désormais capables d'appréhender d'un point de vue technique.

Nous souhaiterions enfin remercier l'ensemble de nos encadrants pour leur supervision de notre avancement tout au long du projet :

- Eric NAVARRE pour l'encadrement technique apporté
- Pierre AUBIN pour la reprise en main globale du projet au sein de l'entreprise
- Bertrand PÉTAT pour le suivi universitaire et les démarches administratives