**INSTITUT UNIVERSITAIRE DE TECHNOLOGIE JLOGICIELS**de BLAGNAC-TOULOUSE II 5 route de Toulouse  
**Département Informatique** 31700 Cornebarrieu1, place Georges Brassens –BP 73-  
31703 BLAGNAC Cedex

Développement d’une application mobile dédiées au client de JLOGICIELS

Rapport de stage effectué du 3 avril au 23 juin 2017

Destinataires :

DELRIEU Jean-Luc  
LEGRAND Amélie

**Florent LABRUNE Année 2016-2017**

**INSTITUT UNIVERSITAIRE DE TECHNOLOGIE JLOGICIELS**de BLAGNAC-TOULOUSE II 5 route de Toulouse  
**Département Informatique** 31700 Cornebarrieu1, place Georges Brassens –BP 73-  
31703 BLAGNAC Cedex

Développement d’une application mobile dédiées au client de JLOGICIELS

Rapport de stage effectué du 3 avril au 23 juin 2017

Destinataires :

DELRIEU Jean-Luc  
LEGRAND Amélie

**Florent LABRUNE Année 2016-2017**

# REMERCIEMENT

Je tiens à remercier dans un premier temps, toute l’équipe pédagogique de l’IUT de Blagnac pour avoir proposé des enseignements et des cours de qualité tout au long de ces 2 années d’études, ainsi que pour leur écoute et leur attention.

Je tiens également à témoigner ma reconnaissance aux personnes suivantes qui ont fait partie intégrante de mon stage :

Monsieur Jean-Luc Delrieu, dirigeant de l’entreprise JLogiciels, pour avoir accepté de m’accueillir au sein de son entreprise pour mon stage de 3 mois, pour ses conseils avisés, son écoute et le temps qu’il m’a accordé.

Monsieur Alexy Delrieu, alternant en charge de l’application mobile, pour son accompagnement en début de stage et son aide tout au long de ce dernier.

Madame Amélie Legrand enseignante à l’IUT de Blagnac, pour m’avoir suivi pendant mon stage et pour ses conseils avisés

Merci à toute l’équipe de JLogiciels pour leur accueil, leur soutien et leur bonne humeur. Mention spéciale à Fabien Chabanon pour sa formation de qualité à la machine à café.

SOMMAIRE

[REMERCIEMENT 2](#_Toc485067630)

[INTRODUCTION 2](#_Toc485067631)

[L’entreprise JLOGICIELS 2](#_Toc485067632)

[I. Présentation de l’entreprise 2](#_Toc485067633)

[II. Activités de l’entreprise 2](#_Toc485067634)

[III. Organisation de l’entreprise 2](#_Toc485067635)

[Contexte du stage 2](#_Toc485067636)

[Analyse de la demande 2](#_Toc485067637)

[Environnements, démarche et choix techniques 2](#_Toc485067638)

[Conclusion 2](#_Toc485067639)

[A. Bilan professionnel 2](#_Toc485067640)

[B. Bilan personnel PPP 2](#_Toc485067641)

[Lexique 2](#_Toc485067642)

[Table des illustrations 2](#_Toc485067643)

[Annexe 2](#_Toc485067644)

[Résumé 2](#_Toc485067645)

[Abstract 2](#_Toc485067646)

# INTRODUCTION

Toute formation doit se terminer en une mise en pratique concrète des connaissances et des compétences. Le mieux est une confrontation à une situation réelle. Pour ceci, le stage est une épreuve parfaite. Il permet de **mettre en application ses connaissances et de les évaluer**. De plus, ce stage est pour moi une opportunité d'avoir une première approche professionnelle dans le domaine de l’informatique.

Pour ceci j'ai décidé de faire mon stage dans l’entreprise JLOGICIELS. **JLOGICIELS est un éditeur de logiciels précurseur dans le domaine des logiciels SaaS[[1]](#footnote-1) et spécialisé dans les CRM[[2]](#footnote-2) (GRC) et logiciels de gestion**. Actuellement, JLOGICIELS compte près de 5000 utilisateurs dans toute la France et les pays francophones, et répartis dans tous les secteurs d'activité, même si les métiers du bâtiment restent le secteur privilégier.

Dans une optique d’optimisation, l’entreprise cherche à offrir à ses clients des moyens simples, rapides et mobiles pour gérer leur logiciel et avoir une vue globale sur l’utilisation de ce dernier. **JLOGICIELS souhaite donc développer une application mobile** servant « d’écran de contrôle » à ses clients.

Dans un premier temps, nous allons analyser le contexte de l’entreprise ainsi que les besoins clients. Nous pourrons, par la suite, étudier les besoins pour définir les objectifs de l’application et les choix techniques qui en découlent. Après un développement complet de la solution mis en place, nous exposerons un bilan. Ce bilan, professionnel et personnel, sera à la fois porté sur le stage mais également sur ces deux années en IUT.

# L’entreprise JLOGICIELS

## Présentation de l’entreprise

Jean-Luc DELRIEU, créateur et gérant actuel de l’entreprise, a créé JLogiciels il y a maintenant 12ans en 2005. Mais pour revenir aux premiers logiciels qu’il a développés il faut remonter bien plus loin. En 1999 il créé son premier logiciel de gestion pour une entreprise détenue par des amis. Il développera ensuite une multitude d’autre logiciel sans étiquette jusqu’en 2005.

Une fois le label JLogiciels créé, il fonde l’entreprise du même nom qui se base à Cox. A partir de là, le rythme s’accélère et les commandes afflues, obligeant sont dirigeant à embaucher et déménager une première fois dans de locaux plus grand mais toujours à Cox.

Elle déménage ensuite à Cornebarrieu, où elle se situe maintenant depuis 4 ans. Le but étant de se rapprocher du cœur dynamique du grand Toulouse.

## Activités de l’entreprise

Ce graphe représente les différents parts que représente chaque activité au sein de l’entreprise JLogiciels.

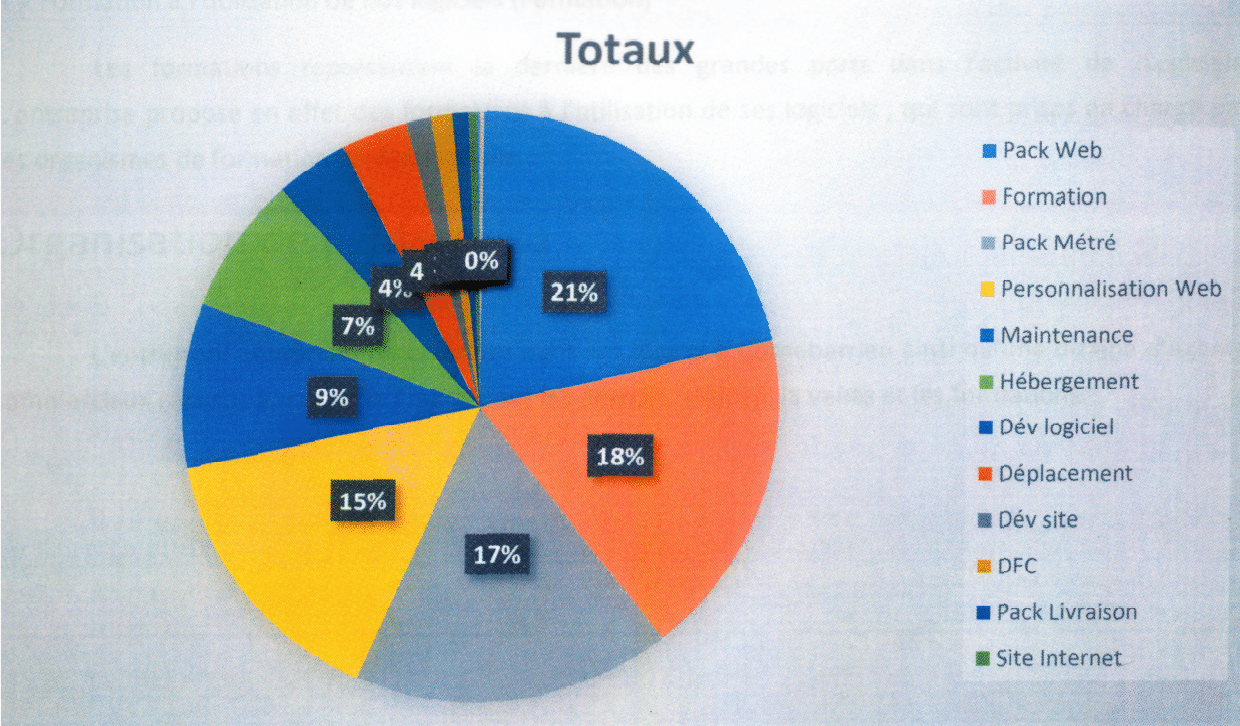


Figure 1 : Les différentes activités de JLOGICIELS

On peut voir que 4 secteurs sont majoritaires :

1. Les logiciels de gestion pour toutes les entreprises (environ 3000 utilisateurs) (Pack WEB)

Les logiciels de gestion représentent la majeure partie de l’activité de l’entreprise. Il permet aux entreprises de faciliter leur gestion commerciale en proposant des outils modernes et rapides en SaaS c’est-à-dire en ligne, accessible depuis n’importe où, sur n’importe quel poste, tablette ou smartphone, et sans aucune installation préalable.

1. Les logiciels de métré pour les négoces matériaux (environ 1500 utilisateurs) (Pack Métré)

Les logiciels métrés représentent eux aussi une grande partie de l’activité de l’entreprise. Ces logiciels sont plus axés sur les négoces matériaux et permettent de maîtriser le coût des futurs travaux. Par exemple, à partir d’un plan, il est possible de calculer l’ensemble des matériaux nécessaire à un chantier. Les principaux clients pour ces logiciels sont LEROY MERLIN, TOUT FAIRE, BIGMAT, GEDIMAT, etc.

1. Développement spécifique (Personnalisation Web)

La part du développement spécifique est à peu près égale à celle des logiciels métrés. JLogiciels propose en effet de personnaliser des logiciels existants voir même de créer des logiciels totalement nouveaux en fonction des demandes des clients.

1. Formation à l’utilisation des logiciels (Formation)

Les formations représentent la dernière des grandes parts dans l’activité de JLogiciels. L’entreprise propose en effet des formations à l’utilisation de ses logiciels ; qui sont prises en charge par les organismes de formation professionnelle.

## Organisation de l’entreprise

L’entreprise est composée de 7 personnes dans ses locaux à Cornebarrieu ainsi qu’une dizaine d’agents commerciaux répartis à travers la France pour les démonstrations, la vente et les formations

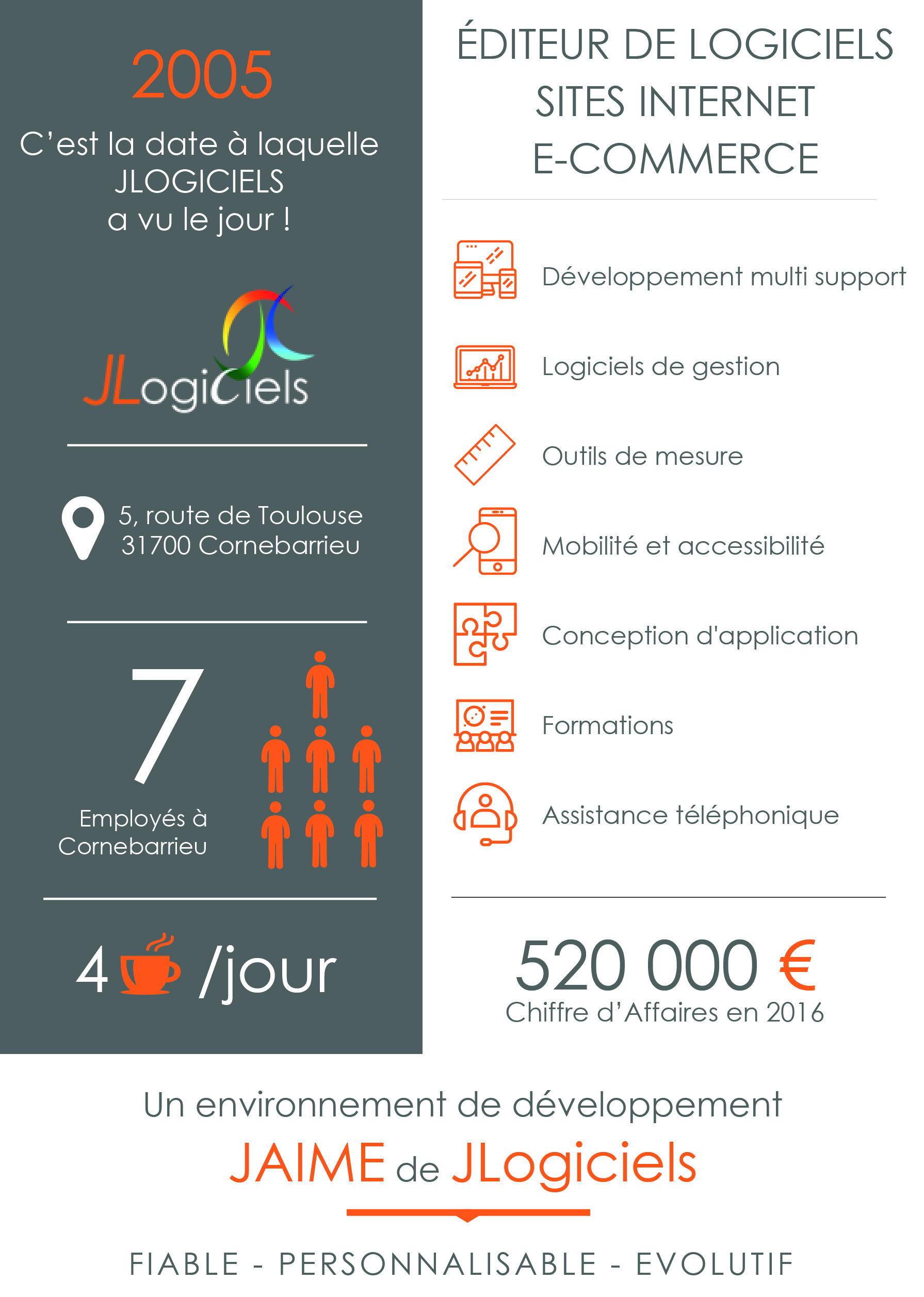


Figure 2 : Infographie sur l'entreprise JLogiciels

# Contexte du stage

Les logiciels de gestion (Pack WEB) est le secteur de l’entreprise qui comporte le plus d’utilisateurs. C’est aussi le domaine où la marge de progression en termes de nombre de clients est la plus grande.

La marge par utilisateur fait sur ce pack est très faible les prix de JLogiciels étant très compétitifs. Les bénéfices reposent donc sur la quantité, c’est pourquoi il est nécessaire de convaincre un maximum de nouveaux clients potentiels. Pour ceci l’entreprise a mis en place deux nouvelles stratégies venant s’ajouter aux deux déjà existantes.

La première déjà en place étant le fait que le logiciel soit un logiciel SaaS. Un logiciel dit « Software as a service » est en fait un logiciel en ligne. On y accède donc via un navigateur web quelconque et une adresse web précise (dans le cas des logiciels JLogiciels : <https://jaime.jlogiciels.fr>) contrairement à un logiciel classique qui se lance avec un fichier exécutable installer sur un pc. Les avantages de ce système sont nombreux. Les deux principaux étant qu’il n’y a besoin d’aucune installation préalable pour s’en servir et qu’on peut y accéder depuis n’importe quel appareil doté d’un navigateur web et d’une connexion internet.

La deuxième stratégie déjà en place est le prix qui est très compétitif, permettant à de petites entreprises comme des auto-entrepreneurs ou des PME d’avoir un logiciel de gestion professionnel à un prix tout à fait abordable.

L’une des deux nouvelles stratégies est l’évolution du pack web pour qu’il s’inscrive dans la nouvelle règlementation des logiciels de caisse et de facturation, qui sera mis en vigueur le 1er janvier 2018. Pour ceci le logiciel doit satisfaire des conditions d'inaltérabilité, de sécurisation, de conservation et d'archivage des données stipulées par la norme NF 525. Cette évolution, déjà en place, permet d’anticiper ce futur changement de législation et donc de viser des clients potentiels utilisant des logiciels bientôt plus aux normes.

La deuxième nouveauté est le développement d’une application mobile qui viendra s’ajouter au pack web. Permettant, ainsi, d’avoir une version simplifiée et légère de l’application en ligne sur smartphones. Cette application sera en permanence connectée à la même base de données que lorsqu’un utilisateur est connecté à la version web. Les utilisateurs auront donc accès à toutes leurs données depuis une application dédiée plus adaptée aux smartphones et tablettes. Offrant ainsi une expérience plus agréable que si l’utilisateur se connecté au logiciel depuis le navigateur de l’appareil. De plus en cas d’absence de connexion internet l’application sera utilisable en mode hors ligne et synchronisera ses données avec la base en ligne lorsqu’une connexion sera de nouveau disponible.

Mon stage s’inscrit donc dans le cadre de cette application mobile. Cette application est développée dans sa majeure partie par l’alternant Alexy Delrieu, expert en développement mobile. Une partie du développement de cette application me reviens, cette partie est appelée « Tableau de bord ». Ce dernier est en fait la page d’accueil de l’application. Le but de cette page est d’offrir à l’utilisateur une synthèse claire, simple et efficace sur ses données. L’utilisateur devra donc retrouver sur cette interface des valeurs, des statistiques, des graphiques et plein d’autres indicateurs pertinents qui lui sont réellement utiles.

La problématique est donc la suivante, comment offrir à chaque utilisateur un tableau de bord personnalisé qui lui correspond alors que tous les utilisateurs ont des besoins, des données et des profils différents ?

Résumons donc, actuellement JLogiciels proposent des logiciels de gestion en ligne à des centaines de client regroupant des milliers d’utilisateurs. Grace à un système de personnalisation, chaque client a un logiciel plus ou moins différent des autres. Pour rendre ce produit encore plus attractif, l’entreprise souhaite pouvoir proposer une application mobile en complément du logiciel en ligne utilisant la même base que la version web. Enfin dans cette application, en page d’accueil sera proposé un tableau de bord qui regroupera plusieurs indicateurs, informations et statistiques.

# Analyse de la demande

Le but de mon stage est donc de développer ce tableau de bord pour l’application mobile qui sera proposée aux utilisateurs du pack web. Cette interface offrira donc à l’utilisateur différent types de représentations des données :

* Du texte
* Des images
* Des valeurs calculées à partir des données de la base du client
* Des graphiques (histogrammes ou diagramme circulaire) à partir des données de la base du client

Cette interface devra également être adaptée à son utilisateur pour être la plus pertinente possible. En effet, un commerciale sera plus concernés par un certain type de données qu’un technicien, un comptable ou encore un responsable d’entreprise. Pourtant toutes ces personnes vont utiliser l’application de gestion. La demande et le besoin se transforme alors en une infinité de demandes et de besoins. La solution la plus simple serait de faire un tableau de bord statique, qui serait le même pour tout le monde, basé sur les mêmes données. Mais dans ce cas-là il perdrait toute pertinence et deviendrai inutile. A l’inverse on pourrait développer un tableau de bord spécifique à chaque utilisateur en fonction de son besoin mais la charge serait bien trop grande car il faudrait faire autant d’application que d’utilisateur.

Il y a une troisième solution, qui n’est pas de développer un ou plusieurs tableaux de bord, mais de développer un générateur de tableau de bord. Ce générateur serait un genre de machine à laquelle on dit ce que l’on aimerait comme tableau de bord et qui, en retour, nous le génèrerai. Ainsi un utilisateur pourrait décrire son tableau idéal à un développeur de JLogiciels qui traduirait ces spécifications en un langage que comprend le générateur. A son tour le générateur traduirai ces spécifications en l’interface souhaitai pour le tableau de bord.

Et c’est à moi que reviens le développement de cette machine « magique ». Ainsi, je ne me retrouve plus à développer une application pour les clients de JLogiciels mais un interprète de tableau de bord pour les développeurs de JLogiciels. Le schéma ci-dessous résume ce fonctionnement.

Figure 3 : Schémas représentant la conception d’un tableau de bord

**Programme**

**Développeur**

**Utilisateur**



Décris

Traduis

Génère

**Tableau de bord**

# Environnements, démarche et choix techniques

## Adaptation à l’environnement de travail

Mon stage étant le développement d’une application mobile, qui plus est, sur Android, les connaissances apportées par les cours de CDAM[[3]](#footnote-3) à l’IUT furent extrêmement utile. De plus, l’environnement utiliser pendant le stage était Android Studio, le même que lors des cours à l’IUT. Mais ce ne fut pas le seul cours très utile lors de mon stage. En effet, l’Android étant principalement basé sur le langage JAVA, un langage orientée objet, tous les cours de conception et de production d’applications orientées objet furent un avantage non négligeable.

Pour la réalisation du tableau de bord il est nécessaire d’avoir un outil pour dessiner des graphiques. Ce genre d’outil permettant de dessiner certaines chose est appelé « bibliothèque/librairie graphique ». Le système Android embarque avec lui une multitude de librairie graphique, mais malheureusement, aucune d’entre elles ne permet de dessiner des graphiques simplement. Heureusement il existe une multitude de projet open-source[[4]](#footnote-4) permettant ceci et qu’une application Android peut utiliser. La bibliothèque retenue est MPAndroidChart[[5]](#footnote-5) créée et développée par Philipp Jahoda disponible sur la plateforme GitHub[[6]](#footnote-6).

Les spécifications du tableau renseignées par les développeurs de JLogiciels étant stockées sur leurs serveurs, il faut également mettre en place un webservice[[7]](#footnote-7) qui fera la passerelle entre la base de données en ligne et l’application mobile. Ce webservice étant développé en PHP, les enseignements de l’IUT sont une fois de plus mis en application. En effet, le PHP est un langage WEB parfaitement adapté au développement d’un webservice qui a été vu en cours.

## Organisation

### PHASE DE TRAVAIL

L’organisation repose sur des phases courtes de travail itératives rappelant la méthode SCRUM. On construit ainsi l’application, petit à petit par des genres de sprints. Contrairement à la méthode SCRUM il n’y a pas de backlog du produit fini. Les backlogs de « sprints » sont eux par contre conservés.

Les phases de travail sont découpées en trois parties :

* La première est une phase de « conception » avec mon responsable. Nous discutons de la fonctionnalité à réaliser, son but, sa faisabilité, les atouts et les contraintes qu’elle offre.
* La deuxième phase est la phase de production de cette fonctionnalité.
* La troisième phase est une phase de « rendu ». Nous voyons si les objectifs ont été atteints, si des changements ou des évolutions sont possibles.

Cette organisation demande de développer de manière à pouvoir toujours retourner en arrière, ce qui apporte une grande modularité à l’application. Il faut garder en tête que n’importe quelle fonctionnalité produite peut être sujet à modification ou disparition dans de futures évolutions et que d’autres peuvent apparaître.

### DEROULEMENT DU DEVELOPPEMENT

Le développement est lui séparé en 5 parties distinctes (cf. Gant en annexe, figure 5). La première partie est l’étude de l’existant, je dois développer un module pour une application déjà en partie réalisée. De plus les statistiques que je dois faire se basent sur les données stockées dans l’application. **Il est donc indispensable que je comprenne et maitrise ce qui a déjà était fait** pour réaliser ma partie.

Les 4 parties suivantes sont la réalisation des différentes versions. Chacune de ces quatre parties représente une phase de travail décrite plus haut. A la fin de chaque partie, une version fonctionnelle est livrée. Le tableau de bord passera donc par 3 versions intermédiaires avant sa version finale. Voici les idées principales derrière chaque version :

1. Version la plus simple de l'application, elle permet uniquement d'afficher des diagrammes simple sans aucune personnalisation, aucune interaction et aucun paramétrage.
2. Elle permet au développeur de pouvoir paramétrer le style du tableau de bord, les couleurs, les tailles, les marges...
3. Elle permet des interactions entre le tableau de bord et son utilisateur, sur les graphiques il peut choisir la période à afficher (tous les graphiques étant temporels) ou l’échelle (mois, trimestre, année…). Il peut également afficher en plein écran les graphiques.
4. C’est la version finale. Elle ne touche pas le tableau de bord, cette version permet d’afficher des graphiques n’importe où dans l’application s’il y a la possibilité de faire des statistiques à partir des informations à l’écran (Ex : si l’utilisateur est sur une activité qui affiche des devis, il peut afficher un graphique à propos de ces devis).

Une semaine est réservée en fin de stage qui correspond à la semaine de retour d’Alexy Delrieu, l’alternant en charge du développement de l’application. Cette semaine permet de faire un point sur ce qui a été fait dans le code afin qu’il soit parfaitement harmonisé. Alexy doit maitriser la partie que je développe afin de pouvoir intervenir dessus après mon départ.

## Contraintes

### LE DEVELOPPEMENT MOBILE

Le développement mobile est un domaine assez particulier en informatique. A mi-chemin entre l’informatique « fixe » et l’informatique embarquée, il présente à la fois les avantages des 2 mondes mais aussi leurs inconvénients. Ces particularités sont tout autant de contraintes à prendre en compte pour l’application. De plus le tableau de bord étant entièrement personnalisable, il est impossible de prévoir le rendu final.

#### Un écran particulier

L’application devra répondre à une particularité propre au développement mobile, la taille de l’écran. En effet, la taille, la résolution et la densité[[8]](#footnote-8) sont totalement différents selon les appareils. Il faut donc réadapter le contenu pour l’afficher de la manière la plus lisible possible sans perte d’information pour chaque écran.

#### La mobilité

Surement la plus importante des particularités du développement mobile, la mobilité du périphérique nous oblige à prévoir le fait que l’utilisateur peut se trouver dans une zone ou l’accès internet est payant ou inexistant. Il faut donc trouver des alternatives lorsqu’Internet n’est plus disponible.

#### La puissance

Les performances offertes par nos smartphones aujourd’hui sont certes une prouesse technique, mais elles restent bien loin de celles d’ordinateurs ou serveurs. Le serveur utiliser par JLogiciels est équipé d’un bi Xeon et de 128Go de RAM, la puissance de calcul offerte est donc bien supérieure à celle de n’importe quel smartphone. Il faut donc éviter au maximum les calculs inutiles et optimiser les instructions.

#### La batterie

Un des points importants du développement mobile est la prise en compte de la consommation de la batterie. Cette contrainte change de manière profonde la façon dont fonctionnera l’application. On évitera notamment les longues taches en arrière-plan, on réveillera le processeur seulement quand c’est nécessaire et on privilégiera les fonctions consommant le moins d’énergie.

### Objectifs du tableau de bord

L’objectif principal de ce tableau de bord est de proposer une interface agréable, ergonomique qui apporte un vrai plus à l’application grâce à des indicateurs pertinents. Pour optimiser au maximum cette pertinence le tableau de bord doit pouvoir être différent pour chaque utilisateur. Comme dis précédemment, la solution choisis pour proposer ce niveau de personnalisation est de développer un générateur de tableau de bord.

Il faut voir ce générateur comme une boite de lego. Il offre aux développeurs de JLogiciels autant de briques qu’il souhaite pour construire un tableau de bord. Pour chaque brique, il peut choisir : sa forme, sa taille, sa couleur… Il a sa disposition 8 briques de type différentes : histogramme, camembert, valeur, formule, titre, commentaire, image et marge

#### Histogrammes et camemberts

Ces deux types sont transformés en interne par le générateur en un seul, les diagrammes. Pour ce type, le développeur doit obligatoirement préciser la table de la base de données client dont va se servir le générateur comme source de données pour créer le diagramme. Il doit également préciser l’attribut de date car tous les graphiques sont temporels avec des valeurs temporelles en ordonnée. Cette attribut date peut par exemple être la date de création si on interroge une table de devis. Il peut ensuite ajouter une clause « where », un attribut de cumulation ou un attribut de groupage, ainsi il peut créer la requête qu’il veut afin que les données utilisées par le graphique soient les plus pertinentes possibles. Enfin il peut choisir un titre pour le diagramme.

#### Valeurs

Les valeurs sont des indicateurs quantitatifs calculés à partir de la base de données. Comme pour les diagrammes le développeur doit préciser la table à interroger. Il peut également ajouter une clause « where » ou un attribut de cumulation. En revanche il est inutile dans le cas des valeurs de préciser un attribut de date ou de groupement le but étant d’avoir une valeur unique et non pas un groupe de valeur. En revanche il peut préciser un nom de variable qui sera utilisé dans le cas des formules. Le développeur peut également rajouter un titre qui permet d’ajouter du texte devant la valeur calculée.

#### Formules

Les formules fonctionnent comme les valeurs, la sortie est un indicateur numérique qu’on peut habiller d’un titre. Ce qui change est la méthode de calcul de cet indicateur. La base de données ne rentre pas en jeu dans ce cas-là, ce sont les variables dont on a parlé juste avant qui sont utilisées. Le développeur rentre une formule mathématique avec les noms de variables qu’il a définis et le générateur va venir remplacer les noms par les valeurs et calculer le résultat. Notez qu’on peut attribuer un nom de variable à une formule et donc l’utiliser dans une autre.

#### Titres et commentaires

Ici on fait une petite pause, il n’y a pas de base de données à interroger, pas de calcul à faire, pas de variables, juste du texte. Le développeur rentre un texte et le générateur l’affiche et c’est tout. Ces deux types sont d’ailleurs très proches, leur seule différence étant la mise en page du texte.

#### Images

Dans ce cas-là, le développeur a juste à donner le nom de l’image et c’est tout. Enfin c’est presque tout, pour que l’application retrouve cette image il faut qu’elle soit sur le serveur de JLogiciels dans un dossier précis. L’application récupère lors de la connexion d’un utilisateur toutes les photos contenues dans ce dossier et les stocks dans la mémoire interne du téléphone. Le générateur n’a alors qu’à se servir.

#### Marges

La marge est le type le plus simple, encore plus simple que titres et commentaires. Oui c’est possible, je vous l’assure. Simples zones vides, elles permettent d’aérer le tableau de bord en éloignant ses composants.

Toutes ces briques ont d’autres attributs qui leurs sont communs. On peut à tous paramétrer leurs dimensions (Hauteur, largeur) en pixel ou en % de la place disponible à l’écran. On peut également fixer une couleur de fond et une bordure avec son épaisseur et sa couleur. Il est également possible de mettre en formes tous les textes à afficher grâce à certaine balise HTML. Il est également possible d’attribuer un comportement à toutes les briques lorsque l’on « clique » dessus (sauf cas exceptionnel des graphiques). Ce comportement peut être de rediriger l’utilisateur au choix, vers une page de l’application ou vers une page web.

## Choix techniques

Dans l’esprit de cette personnalisation, rien n’est défini dans le code source de l’application. Tout le tableau de bord va se construire au moment du lancement de l’application en fonction des spécifications. Ces spécifications sont saisies par un développeur dans l’environnement de développement JAIME propre à JLogiciels. Cet environnement permet la communication entre les développeurs et le serveur de JLogiciels et sa base de données.

### ORGANISATION DE L’APPLICATION

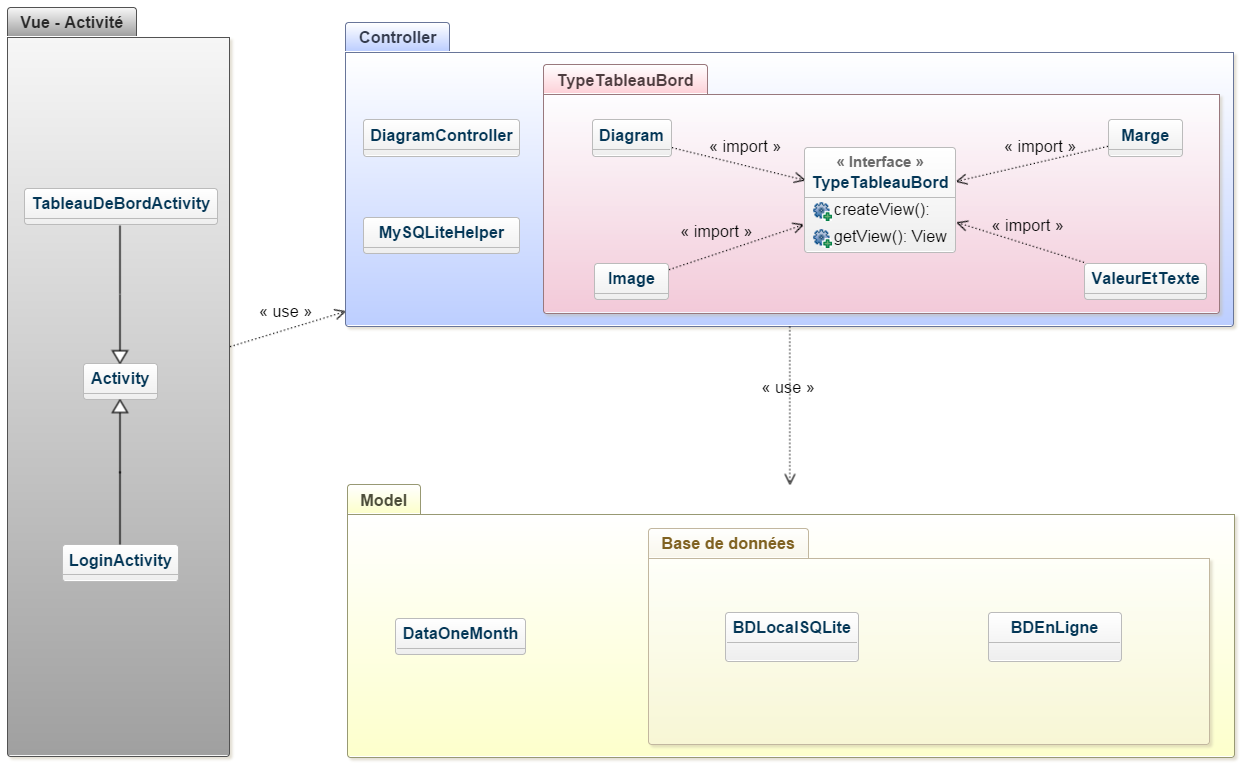


Figure 4 : Diagramme de classe MVC simplifié

Comme le montre ce diagramme, le tableau de bord respecte une architecture MVC[[9]](#footnote-9) (c’est le cas de toute l’application). Les activités sont les vues, dans les contrôleurs on retrouve un contrôleur pour la base de données, un contrôleur général pour le tableau de bord et un contrôleur pour chaque type décris précédemment. La base de données est utilisée comme model à quoi s’ajoute « DataOneMonth » qui est le format de données utilisé par les diagrammes.

Pour pallier le problème de performance évoqué plus tôt, toute interaction faite avec la base de données passe par son contrôleur « MySQLiteHelper ». Cette classe est une bibliothèque de fonctions autonomes qui assure la liaison entre la base et le reste de l’application. Dans cette classe, toutes les fonctions sont au maximum optimisées. Elles assurent également qu’il n’y est aucune redondance dans la base locale. Un autre contrôleur non représenter sur le schéma assure la synchronisation entre la base en ligne et la base locale.

Lorsque l’application est lancée, la première activité lancée est le tableau de bord. Avant d’appeler le générateur, une interface basique est créée proposant des boutons de navigation et un menu déroulant sur le côté. Une fois ceci fait l’activité appelle la fonction du contrôleur principal qui va générer le tableau de bord.

Le générateur complet représente environs 3000 lignes de code réparties sur une dizaine de classes. A cela s’ajoute une dizaine de lignes en PHP qui assure le webservice.

#### Le contrôleur principal

Son travail est simple, à partir des spécifications dans la base il doit répartir les données entre chaque contrôleur de type, récupérer les vues correspondantes et enfin créer la mise en page finale qui sera rendu à l’activité.

Pour chaque spécification d’un TypeTableauDeBord, le contrôleur général appelle le contrôleur de type correspondant et lui donne les données fournies par la base. Chaque contrôleur de type est stocké dans une même liste sauf les spécifications marqué du titre « INVISIBLE » et les formules. En parallèle il stocke toutes les variables dans une table d’association.

Une fois ceci fait le contrôleur calcul le résultat des formules à l’aide de la table d’association et les convertis en type valeur. Il ajoute ces valeurs dans la liste des contrôleurs.

Après ces deux traitements le contrôleur a la liste finale de tous les éléments à afficher. Il parcoure donc cette liste d’éléments et appelle la fonction « getView » commun à tous les types. La mise en page finale est simple, l’activité qui appelle le contrôleur doit lui préciser la largeur de la zone où sera affiché le tableau de bord en pixels. Ainsi le contrôleur met à la suite toutes les vues et lorsqu’une ligne est remplie, il passe à la suivante.

Le contrôleur principal sert également de bibliothèque de fonction générique utiliser par les « TypeTableauDeBord »

#### Le contrôleur des diagrammes

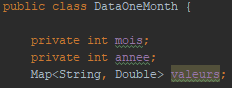
C’est le contrôleur le plus complexe. Tout d’abord penchons-nous sur les données qui lui sont transmises. Elles doivent respecter le format suivant :

Le controleur de la base de données assure le respect de ce format quand la base est interrogée pour un graphique. Le contrôleur va en premier lieux convertir cette chaine de caractère en une liste de « DataOneMonth » qui sera bien plus simple à manipuler par la suite.

Par la suite ces données sont converties en un histogramme ou un camembert selon le type choisis. Seul le cas de l’histogramme sera détaillé, les camemberts étant qu’une version simplifier des histogrammes.

La bibliothèque MPAndroidChart utilise un format de données bien précis qu’on ne détaillera pas non plus ici. Chaque diagramme à son échelle (mois, trimestre, semestre, année) et sa période propre. Nous allons donc nous attarder sur comment le contrôleur à partir d’une liste de « DataOneMonth » créer un tableau à 3 dimensions respectant le format suivant : [période][catégorie][valeur]. Tout ceci se passe dans la fonction

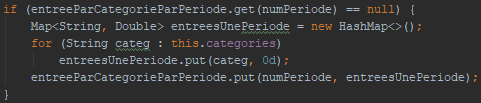
Tout d’abord regardons de plus près ce qu’est « DataOneMonth ».



C’est simplement un objet dans lequel est stocké le numéro de mois dans l’année, l’année et une table d’association stockant les valeurs liées aux différentes catégories dans le mois.

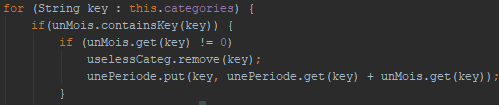
Dans le contrôleur, le tableau évoqué plus haut est représenté par la table d’association suivante :   
Une « SparseArray » étant une table d’association de type Clé (de type entier) ; Valeur (de type quelconque). Dans notre cas, la valeur est une autre table d’association Clé ; Valeur, la clé représentant une catégorie.

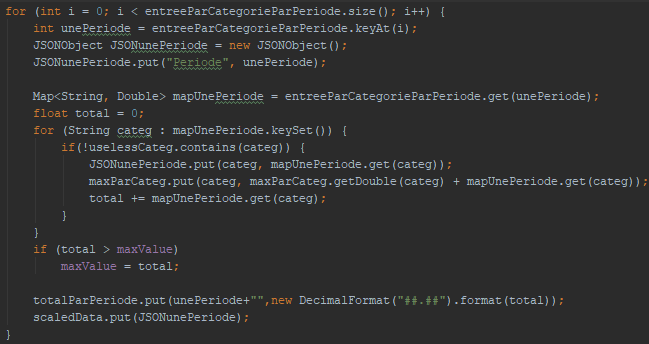
On parcourt tous les dates comprise dans le laps de temps désiré. Pour chaque date, on récupère le numéro de période associée.

Si jamais la période n’existe pas encore, on la créé et on initialise toutes ses valeurs à 0.

On récupère le mois correspondant à cette date dans la liste de « DataOneMonth » et la période associée.



Pour chaque catégorie contenue dans ce mois, on récupère la valeur associée qu’on ajoute à la valeur déjà contenu dans notre période.

Enfin on converti le tableau en une « JSONArray » qui est un type bien plus léger à manipuler pour la suite, le but étant toujours l’optimisation maximum. On profite également du parcours de toutes les valeurs pour calculer le total de chaque période et la période dont la valeur maximum est la plus grande.

Cette JSONArray est traitée dans différentes fonction pour créer un histogramme comme celui-ci grâce à la librairie MPAndroidChart :

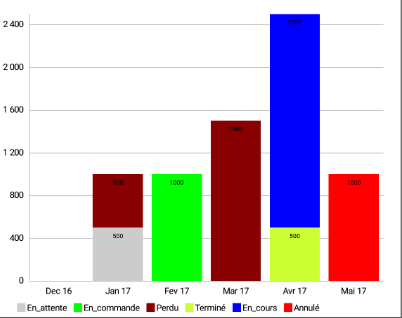


Figure 5: exemple de diagramme généré par l'application

Les contrôleurs de type ne servent pas qu’à créer les vues, ils accompagnent tout leur cycle de vie. Ainsi lorsqu’un utilisateur interagit avec un diagramme par exemple c’est son contrôleur qui intercepte cette interaction.

Par exemple, un utilisateur peut à tout moment changer la période et l’échelle d’un graphique à afficher. Pour ceci, il a à sa disposition des bandeaux déroulant

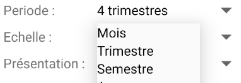


Figure 6: choix période et échelle

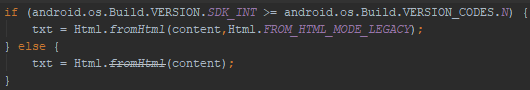
#### Le contrôleur des valeurs et des textes

Dans le cas général d’un texte simple, un nombre de paramétrage est transmis par le contrôleur général. Le contrôleur va créer une zone de texte qui respecte tous ces paramétrages (couleur et taille du texte, couleur de fond, couleur et taille d’une bordure…). Il va ensuite remplir cette zone avec le texte souhaité.

Ce contrôleur peut interpréter les balises html directement écrite dans le texte à afficher, évidemment ces balise sont supprimer avant d’afficher le rendu final. Pour interpréter les balises html on utilise la fonction « content » étant le texte à afficher ;

Le problème est que cette méthode est dite « deprecated » c’est-à-dire qu’elle sera supprimée dans les futures versions de java.

Pour contourner ce problème on regarde quelle version d’Android est installé sur le téléphones et on utilise la fonction qui correspond à la version. Voici donc le code qui en résulte :



#### Le contrôleur des images

Ce contrôleur récupère l’image dont le nom correspond à celui rentré par le développeur, pour ceci il ouvre le fichier à l’adresse donnée par le contrôleur principal

Ensuite il lit les données dans ce fichier et les transforme en une image grâce à la fonction suivante 

Puis il retourne cette image en fixant sa largeur et sa hauteur en fonction des spécifications.

#### Le contrôleur des marges

Il créé un rectangle d’une hauteur et d’une largeur fixé par le développeur. Ce rectangle sera colorié et encadré d’une bordure si le développeur à spécifier une couleur de fond et un encadrement.

# Rendu final

## Point de vue de l’environnement JAIME

C’est dans cet environnement que les développeurs vont venir rentrer toutes les spécifications pour le tableau de bord.

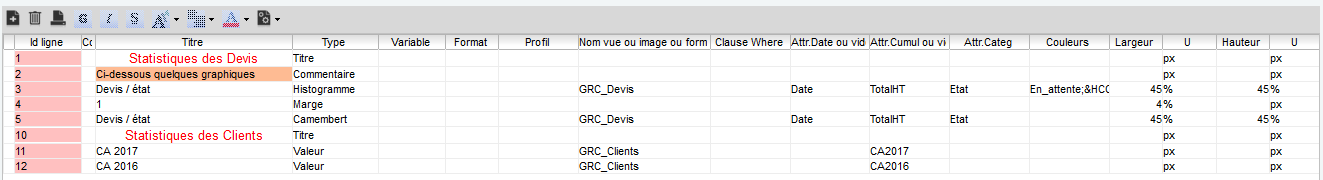
Voici un exemple de spécification d’un tableau de bord simple

Figure 7: Tableau de bord dans l'environnement JAIME

Notez que la couleur et le centrage dans la colonne « Titre » sont fait avec des balises html.

## Point de vue de l’application

Comme expliqué précédemment, toutes les informations contenues dans le tableau ci-dessus vont être téléchargé par l’application puis interpréter par le générateur et enfin afficher à l’écran.

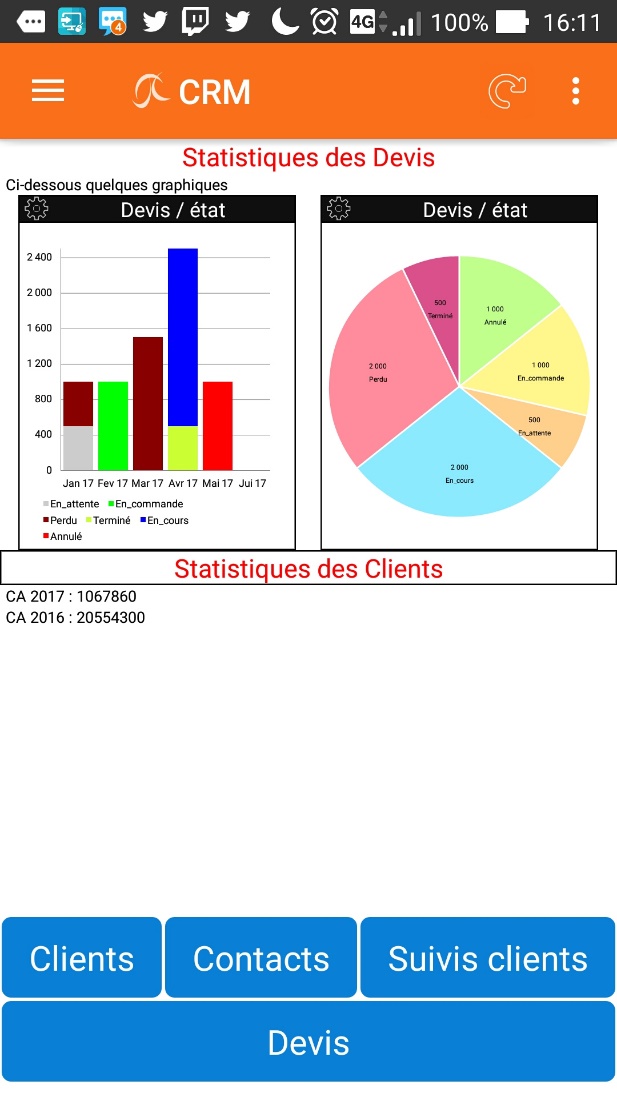
Voici le résultat obtenu lorsque l’application interprète le tableau ci-dessus

Figure 8 : Tableau de bord dans l'application

# Conclusion

## Bilan professionnel

Ce stage à JLogiciels m’a permis de renforcer mes compétences en développement Android. J’ai pu ainsi approfondir des notions vues en cours comme les layouts, les activités ou encore les requêtes internet mais aussi aborder de nouvelles notions telles que ‘utilisation de bibliothèques graphique importées, les tâches asynchrones, les bases de données SQLite, les préférences…

J’ai également pu comprendre les enjeux que représentent un développement mobile, notamment au niveau de la batterie, de la mobilité et de la discontinuité du réseau. Mais également les intérêts de proposer des applications mobiles en complément de logiciel dans le domaine professionnel.

En plus de tous ces apports techniques, le stage m’a aussi beaucoup appris sur le fonctionnement d’une entreprise. Son fonctionnement, les enjeux, la hiérarchie mais également que chaque entreprise a sa propre identité.

## Bilan personnel PPP

Certain dise que la curiosité est un vilain défaut. Pour moi, c’est ma plus grande qualité. Depuis toujours, aussi loin que je me souvienne, j’ai besoin de comprendre les choses qui m’entoure. C’est pourquoi en grandissant je me suis dirigé vers un cursus scientifique. J’ai « officiellement » rejoint le domaine scientifique en seconde.

En effet, j’avais choisis l’option de découverte SI (Sciences de l’ingénieur). Cette année-là j’ai découvert les bases de la mécanique, mais aussi de l’électronique et de l’informatique. Ce fut pour moi, une porte vers un savoir qui me permettrait de répondre à toutes les questions que je me posais. C’est donc tout naturellement qu’en première j’ai décidé de m’engager dans un baccalauréat scientifique, option sciences de l’ingénieure.

Lors de ces deux ans je me suis familiarisé avec ces différentes matières et plus particulièrement avec l’électronique et l’informatique. J’ai également eu l’occasion de réaliser deux projets en groupe, le projet personnel encadré et le projet personnel encadré, respectivement en première et en terminal. Ces deux projets avaient pour but de mettre en œuvres nos compétences dans les trois domaines cités plus haut. Bien évidemment, je me suis occupé de la partie informatique et électronique de ces projets.

J’ai également eu l’occasion de prendre l’option ISN en terminale qui m’a permis de vraiment travailler sur l’informatique pure. Dans cette matière j’ai découvert la programmation et j’ai réalisé un projet, une fois de plus en groupe. Ce projet m’a convaincu qu’il serait une bonne chose que de continuer mes études en informatique.

A la fin de ces 2 ans j’avais donc le choix entre ces deux domaines, l’informatique et l’électronique. J’ai postulé dans leur DUT respectif et j’ai été retenu dans les deux. J’ai donc choisis l’informatique car il était plus accessible pour moi, à la grande déception de mon professeur d’électrotechnique. Mais j’ai tout de même gardé en tête la possibilité de faire l’autre DUT, mais ceci, nous en parlerons plus tard…

Je me retrouvais donc en DUT informatique, un domaine aussi vaste qu’intéressant et des tonnes de compétences et connaissances à acquérir. Tout d’abord dans des domaines techniques. Lors de ces 2 ans j’ai appris la programmation, des bases au techniques plus complexes à travers notamment la programmation orientée objet, l’interface homme-machine ou encore la conception et la programmation objet avancée. Mais aussi comment fonctionne un ordinateur grâce aux cours d’architecture et de principe des systèmes d’exploitation. Nous avons également fait le tour des bases de données en les créant en cours de base données, les administrant en programmation et administration des BD et les utilisant en programmation web coté serveur.

J’ai également appris beaucoup dans d’autre domaine. Comme savoir présenter et mettre en avant sois même ou un projet en communication. Connaitre le monde économique dans lequel nous évoluons en économie et en droit. Apprendre à raisonner et à utiliser les maths dans l’informatique grâce aux différents modules de mathématiques tout au long de ces deux ans. Ou encore gérer un projet, une équipe, des ressources, le temps ou même le budget d’une entreprise à travers les cours de gestion.

Toutes ces compétences ont étés mis en œuvre lors de différent projet dans chaque matière indépendante ou dans de gros projet comme le PTUT. Il reste encore deux matières dont je n’ai pas parlé. La programmation d’application mobile, et l’informatique embarquée. Ces deux matières sont pour moi les deux domaines dans lesquels je voudrais poursuivre mes études.

Tout d’abord le mobile, dans ce domaine, j’aimerai faire une licence professionnelle APSIO à l’IUT de Blagnac, c’est une suite directe au DUT informatique dans laquelle il y a du développement objet, web mais surtout mobile. Ayant fait mon stage dans ce domaine ça serait pour moi le domaine privilégié au vu de mes connaissances et mes compétences.

Mais, car il y a toujours un mais, je n’ai toujours pas parlé de l’embarqué et j’avais également dis que je parlerai du deuxième DUT. Eh bien, c’est maintenant. L’embarqué fur la matière la plus captivante pour moi, c’était de la programmation, domaine que j’adore, mais au plus proche de la machine. Et c’est quelque chose qui m’a manqué ces deux dernières années. C’est pourquoi je souhaite, l’année prochaine, faire un DUT GEII (Génie Electronique et Informatique Industrielle) pour retrouver le domaine de l’électronique que j’avais quitté fin terminal. Ayant à présent le DUT informatique je pourrais faire GEII en année spéciale, une formation qui permet d’avoir un DUT en un an à Paul Sabatier à condition d’être déjà diplômé d’un Bac+2. Par la suite j’aimerai rejoindre la formation SRI de UPSSITECH pour ensuite travailler dans le domaine de la robotique. Un domaine pile à la croisée entre l’informatique et l’électronique, la perfection pour moi en somme.

# 

# Lexique

1. Software as a service.
2. Customer relationship management
3. Conception et Développement d’Applications Mobiles
4. Logiciel ou code libre de droits
5. <https://github.com/PhilJay/MPAndroidChart>
6. <https://github.com>
7. Protocole d'interface informatique de la famille des technologies web permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués
8. Nombre de pixels pour une zone physique de l’écran, elle est mesurer en DPI (dots per inche : points par pouce)
9. Model Vue Contrôleur

# Table des illustrations

Figure 1 : Les différentes activités de JLOGICIELS

Figure 2 : Infographie sur l'entreprise JLogiciels

Figure 3 : Schémas représentant la conception d’un tableau de bord

Figure 4 : Diagramme de classe MVC simplifié

Figure 5: exemple de diagramme généré par l'application

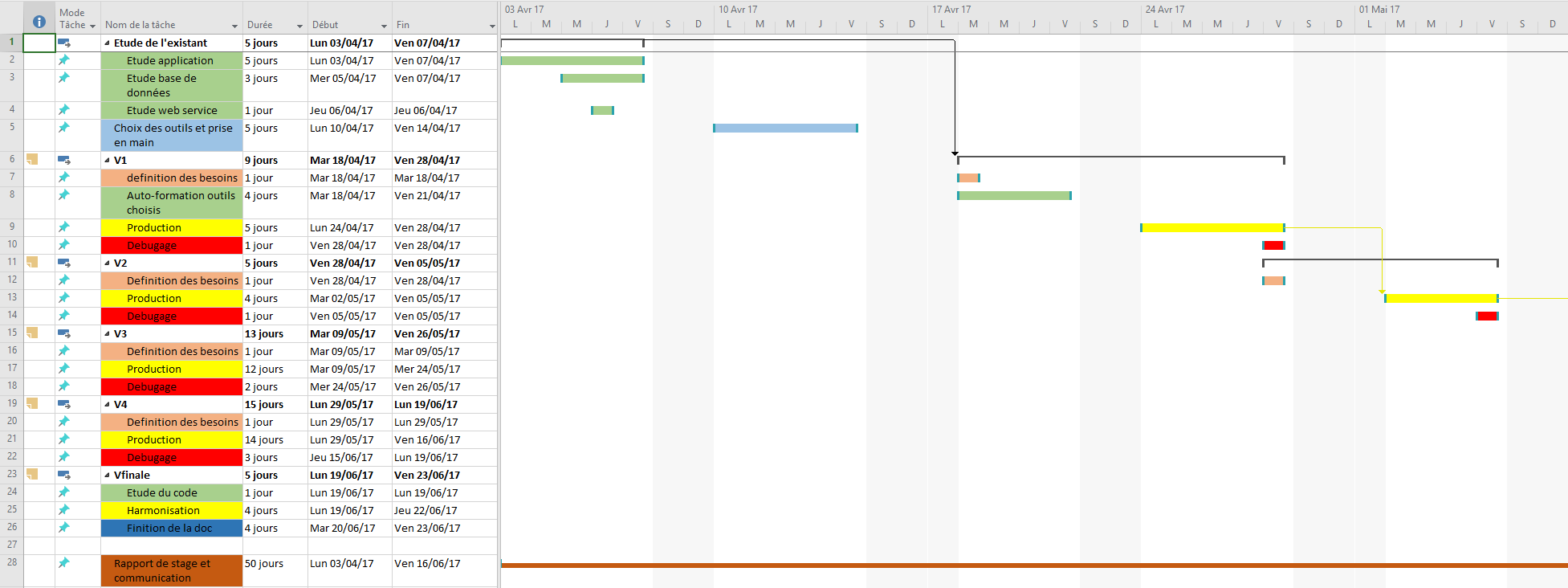
Figure 7: Tableau de bord dans l'environnement JAIME

Figure 8 : Tableau de bord dans l'application

Figure 9 : Gant prévisionnel

# 

# Annexe



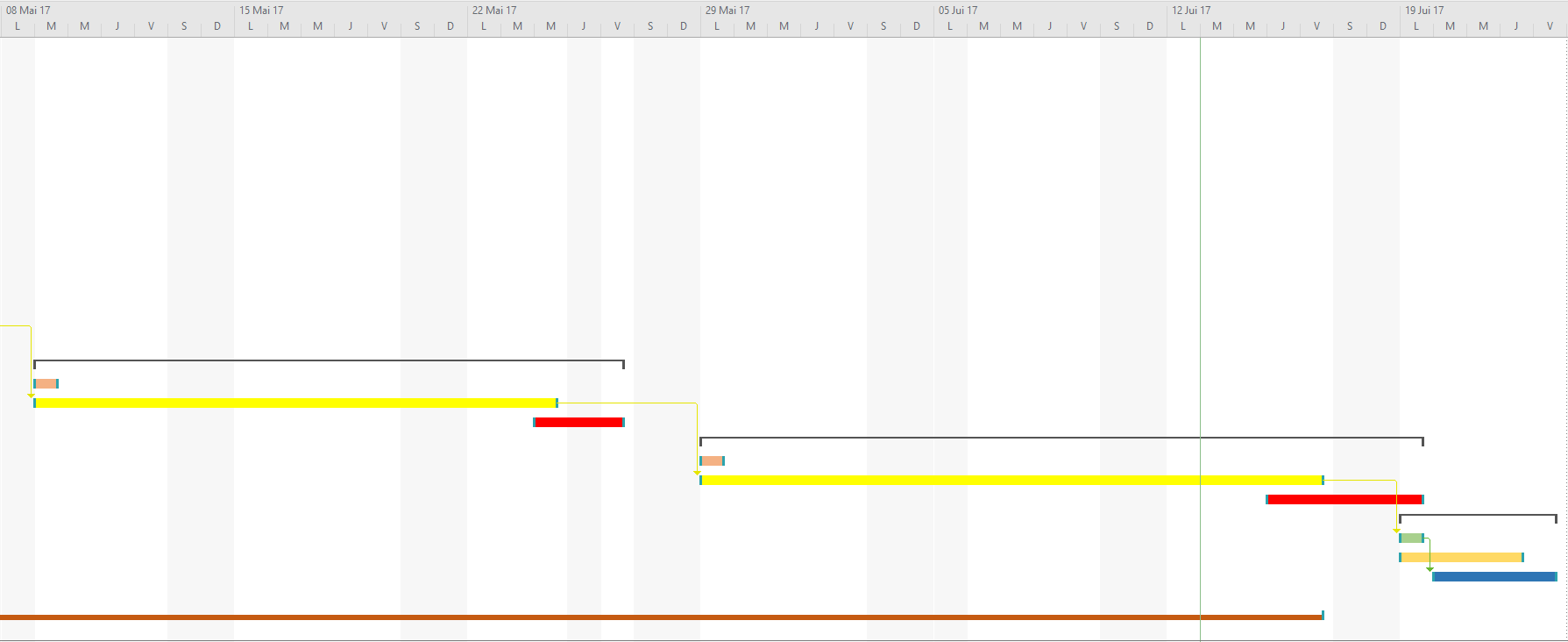


Figure 9 : Gant prévisionnel

# Résumé

Mon sujet de stage est de développer une application mobile pour JLogiciels. Cette entreprise développe principalement des logiciels de gestion pour les entreprises. Dans le but de se renouveler et de convaincre plus de clients, elle souhaite développer une application mobile qu’elle proposera en complément de ses logiciels.

Cette application sera une version plus légère des logiciels qu’il propose. Elle utilisera la même base de données que les logiciels et sera développer selon la même logique d’architecture. En plus des fonctionnalités offertes par le logiciel, l’application aura un tableau de bord qui offrira des indicateurs aux utilisateurs.

C’est à moi que reviens le développement de ce tableau de bord. Les utilisateurs de JLogiciels étant très variés, ce tableau de bord doit s’adapter à son utilisateur. La solution choisis est de développer un générateur de tableau de bord qui sera à l’avenir utilisé par les développeurs de JLogiciels pour créer n’importe quel tableau de bord.

Le générateur devra mettre à disposition des développeurs 8 briques élémentaires : histogrammes, diagramme circulaire, image, titre, commentaire, valeur, formule et marge. Chacune de ses briques doit être personnalisable. Dans cette personnalisation on retrouve, la taille, les couleurs, des bordures ou encore des comportements spécifiques.

Ce générateur sera implémenté directement dans l’application et se servira des données de cette dernière pour ses calculs. Il devra récupérer les spécifications données par les développeurs de JLogiciels directement sur leur serveur.

# Abstract

This internship happened in the company JLogiciels. This company is a precursor in the SaaS management software. It is like basical management software but which are online. It means they are accessible from any browser on any support. In recent years, the smartphones are more and more used to work. In this context, JLogiciels wants to develop an application which is a lighter version of the online software but allow consumers to use it offline. An employee is in charge of developping this application and the subject of the internship is a part of this development.

This part is called “Tableau de bord” or “dashboard” in English. This dashboard is the first activity displayed in the application. It allows to display graphs and statistics about a multitude of data. To develop this application the language used is Java for the algorithmic part and XML for the interfaces. The combination of this two languages is named Android. The first week was given to find a graphic bibliotheca because Android doesn’t offer natively something to create chart. The held solution is the library MPAndroidChart. The SQL language is also used for retrieve the data in the database of the company.

# 

JLogiciels souhaite développer une application mobile en complément des logiciels de gestion qu’il propose déjà. Dans cette application, un tableau de bord sera proposé, il regroupera plusieurs indicateurs statistiques et textuels. Je suis chargé de développer un générateur de tableau de bord qui sera utilisé par les développeurs de JLogiciels. Ce générateur sera implémenté directement dans l’application et se servira des données de cette dernière pour ses calculs. Il devra récupérer les spécifications données par les développeurs de JLogiciels directement sur leur serveur.

1. Software as a service. [↑](#footnote-ref-1)
2. customer relationship management [↑](#footnote-ref-2)
3. Conception et Développement d’Applications Mobiles [↑](#footnote-ref-3)
4. Logiciel ou code libre de droits [↑](#footnote-ref-4)
5. <https://github.com/PhilJay/MPAndroidChart> [↑](#footnote-ref-5)
6. <https://github.com> [↑](#footnote-ref-6)
7. Protocole d'interface informatique de la famille des technologies web permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués [↑](#footnote-ref-7)
8. Nombre de pixels pour une zone physique de l’écran, elle est mesurer en DPI (dots per inche : points par pouce) [↑](#footnote-ref-8)
9. Model Vue Contrôleur [↑](#footnote-ref-9)