**Rapport de Stage**



*Mission : Développement d’une application Android*

FSA IUT de Valence

6, Rue Paul Henri Charles Spaak 51, rue Barthélémy de Laffemas

26000 Valence 26000 Valence

Département Informatique

**Rapport de Stage**



Maitre de stage : Jean-François COUPAT

Tuteur de stage : Sébastien JEAN

Remerciements

Je tiens à remercier toutes les personnes qui ont rendu possible ce stage. Tout d’abord, je remercie M. Jean-François COUPAT, mon maitre de stage qui m’a défini précisément les besoins de l’application et qui m’a suivi tout au long du stage.

Je remercie également Laurent DELOULME, le directeur des opérations France qui m’a admis en tant que stagiaire au sein de FSA, ainsi que Guy BONTEMPS et qui m’a donné des informations clés concernant la société. Mickael VARES m’a grandement aidé pour créer toutes les images des modes de marche. Je le remercie beaucoup pour sa sympathie et sa disponibilité tout au long de mon stage.

Je tiens aussi à remercier tous les employés de FSA Valence et tout particulièrement M. REBOUL pour l’accueil qui m’a été réservé et l’excellente ambiance qui a régnée tout au long du stage.

Je remercie enfin mon tuteur de stage M. JEAN qui a toujours suivi mon stage avec intérêt et bienveillance.

Ces dix semaines m’ont apporté énormément sur le plan professionnel évidemment mais aussi sur le plan humain, sur la gestion de projet, sur les transactions entre clients et vendeur mais aussi sur le fonctionnement d’une entreprise à l’internationale avec toutes les assistances, les réunions et les déplacements que cela implique.

Introduction

Pour conclure la formation de DUT informatique, j’ai été amené à faire un stage du 10 avril au 16 juin 2017. Ayant déjà eu l’occasion de développer sous Android durant mon projet tutoré, j’ai cherché un stage qui pourrait me permettre de continuer dans cette voie. C’est ainsi que j’ai choisi le sujet de stage de « Fabricom Systèmes d’Assemblage » (FSA) qui proposait d’adapter une serveur Web de suivi de production en une application mobile. Ce qui a pour but de pouvoir accéder de manière rapide, simple et ergonomique au suivi de la production depuis un simple smartphone.

Fabricom Systèmes d’Assemblage assemble et créer des chaines de production pour différents clients comme PSA , Renault… Devenue une entreprise multinationale, elle a des filiales en Roumanie, et, au Portugal par exemple. Du fait de la délocalisation de la production, le site de Valence est maintenant dédié à l’analyse, à la conception des projets et au service après-vente. Le seul endroit en France ou la production est toujours assurée est le site de Besançon. En effet, c’est ici que les solutions sont testés et évalués avant d’être déployés à l’étranger.

Chez FSA, chaque commande est unique et nécessite une demande précise du client qui donnera lieu à une étude de projet. De plus le faible nombre de client et le coté très concurrentiel du marché pousse FSA à proposer des solutions innovantes pour remporter les marchés. « Melodie Net » est donc un argument de poids dans les négociations. C’est pourquoi une application mobile basée sur « Melodie Net » permettrait de rendre l’outil plus complet, plus facile à utiliser et plus encore, plus moderne et intégrée dans la mode actuelle.

Encadré par Monsieur Jean-François COUPAT, responsable de l’outil « Melodie », nous avons dû définir ensemble les fonctionnalités attendues, mais également faire un choix quant aux technologies utilisées.

Basée sur le serveur web « Melodie Net », l’application Android que j’ai développé, a une importance fondamentale car elle est vouée à être utilisée dans de nombreux endroits à travers le monde. C’est pourquoi diverses contraintes ont été imposées et notamment la gestion de la langue. Toutes ces contraintes seront explicitées plus tard dans le rapport.

J’ai travaillé en collaboration avec Monsieur COUPAT. Je me suis chargé de la partie « Client », c’est-à-dire l’application Android et M. COUPAT de la partie « Serveur », c’est-à-dire le service Web sur lesquels les informations indispensables à l’application sont accessibles.

Alors quels ont été les besoins et les choix technologiques pour cette application et quels sont les étapes qui m’ont permis de réaliser une application mobile répondant à ces contraintes ?

Table des matières

[Remerciements 3](#_Toc483404538)

[Introduction 4](#_Toc483404539)

[Table des matières 5](#_Toc483404540)

[I) Présentation et objectifs généraux du projet 6](#_Toc483404541)

[A) Présentation de la société 6](#_Toc483404542)

[B) Contexte du projet et besoins du client 10](#_Toc483404543)

[C) Les besoins fonctionnels 11](#_Toc483404544)

[13](#_Toc483404545)

[13](#_Toc483404546)

[13](#_Toc483404547)

[13](#_Toc483404548)

[II) Conception de la solution 14](#_Toc483404549)

[A) L’élaboration du cahier des charges (modèle théorique de lappli) 14](#_Toc483404550)

[B) Explications de fonctionnement 14](#_Toc483404551)

[C) Le diagramme de classe(modèle pratique de lappli) 14](#_Toc483404552)

[III) Mise en œuvre de la solution 15](#_Toc483404553)

[A) Les outils technologiques choisis 15](#_Toc483404554)

[B) Les différentes phases de développement 15](#_Toc483404555)

[C) Les difficultés rencontrées et leurs solutions 16](#_Toc483404556)

[Conclusion 18](#_Toc483404557)

[Annexes 20](#_Toc483404558)

[1.1) Quelques exemples de produits fabriqués par les lignes de production FSA 20](#_Toc483404559)

[1.2) Clients de FSA 22](#_Toc483404560)

[1.3) L’application web « Melodie Net » 23](#_Toc483404561)

[1.4) Les diagrammes de l’application 24](#_Toc483404562)

[1.5) Les IHM de l’application 26](#_Toc483404563)

1. Présentation et objectifs généraux du projet
2. Présentation de la société



Chiffre d’Affaire : 22 606 800 €

**A.1) Historique :**

-La société Fabricom Systèmes d’assemblage a vu le jour en 1999 à l’issue du rachat de la société belge « Fabricom », crée en 1995, par GDF Suez.

-Par la suite, elle prend de l’importance en rachetant d’autres sociétés comme TEHNOMAT en Roumanie par exemple.

-En 2001, c’est la création de la coentreprise « JVS FABRICOM » à Sao Paulo au Brésil, en partenariat avec la société JVS EQUIPAMENTOS PARA AUTOMACAOINDUSTRIAL LTDA.

-En 2013, FSA s’internationalise encore plus avec la création de FSA ASSEMBLY INDIA PRIVATE LIMITED à Pune en Inde.

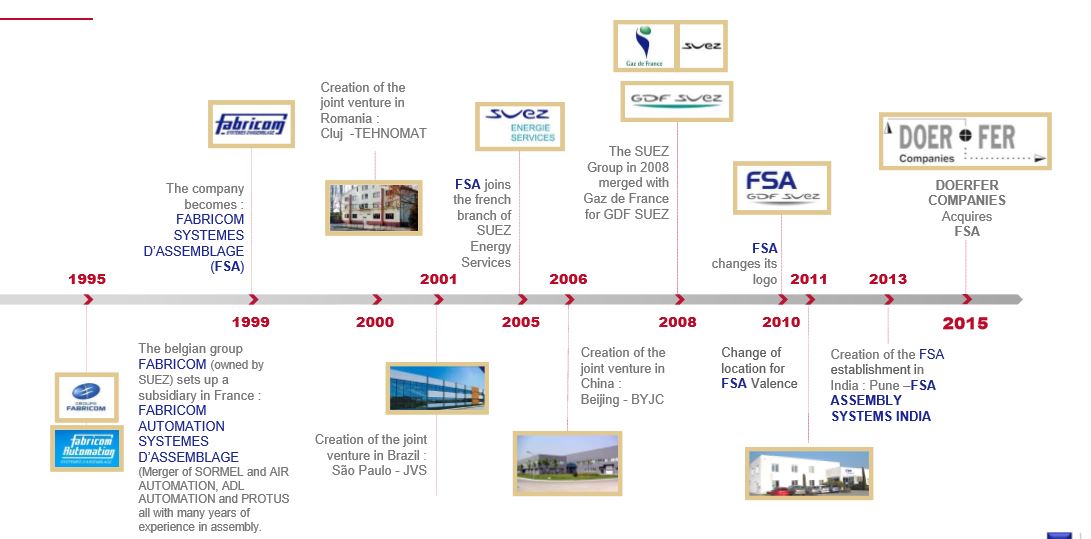


Figure 1 : Historique de la société

FSA (Fabricom Systèmes d’Assemblage) est une société au rayonnement mondial rachetée il y a peu par l’entreprise américaine DOERFER.

**A.2) Organisation au sein de l’entreprise :**

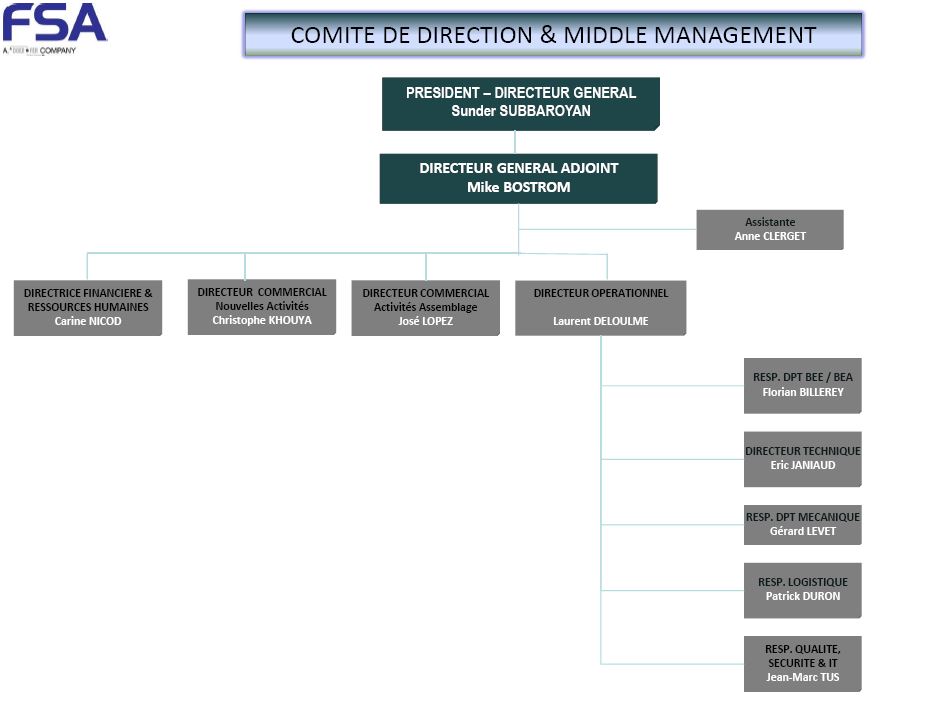
****

Figure 2 : Organigramme

**A.3) Marché :**

FSA a plusieurs particularités liées à son marché notamment. En effet, elle est capable de mener des projets de front partout sur la planète, ce qui est rare et très recherché par ses clients.

L’anglais est donc devenu un prérequis indispensable pour tous les membres de la société.

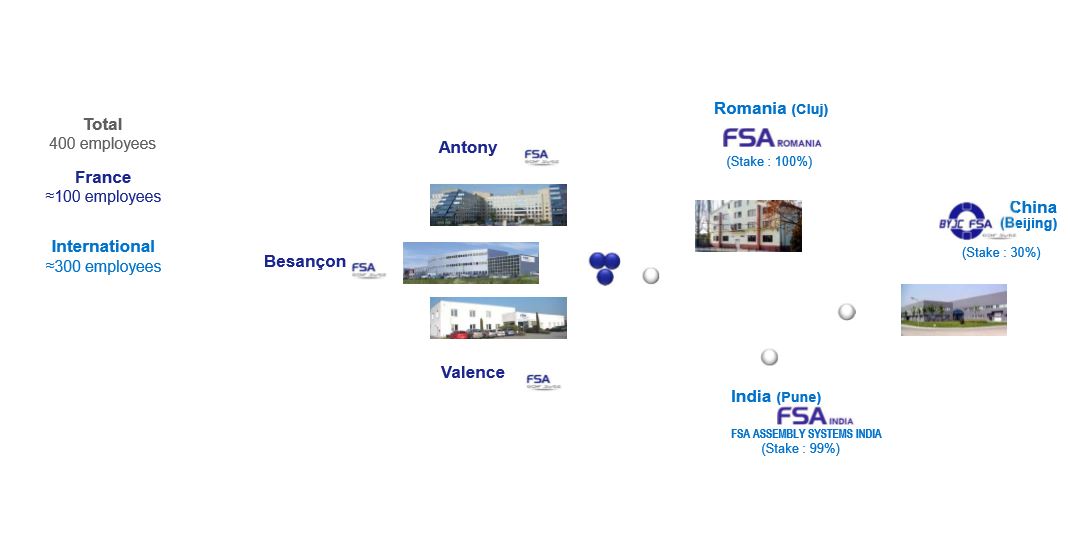


Figure 3 : implantation dans le monde

La vocation de la société est de concevoir et installer des chaines de montage principalement dans l’automobile. Historiquement, le site de Valence est né du rapprochement avec les équipementiers automobiles de la région et le site de Besançon de la proximité avec les grands constructeurs automobiles comme PSA.

FSA fonctionne par projets, c’est-à-dire qu’elle fonctionne en fonction des projets qui lui sont proposés. Ce fonctionnement lui permet d’adapter ses moyens au projet en cours, de constituer des équipes de projets constituée de chefs de projets, d’automaticiens et d’électroniciens. Avant d’en arriver là, chaque projet passe dans une phase de pré-étude ou FSA décide si elle peut s’occuper du projet et propose un chiffrage au client.

**A.4) Commercial :**

* 50 million d’euros de revenu

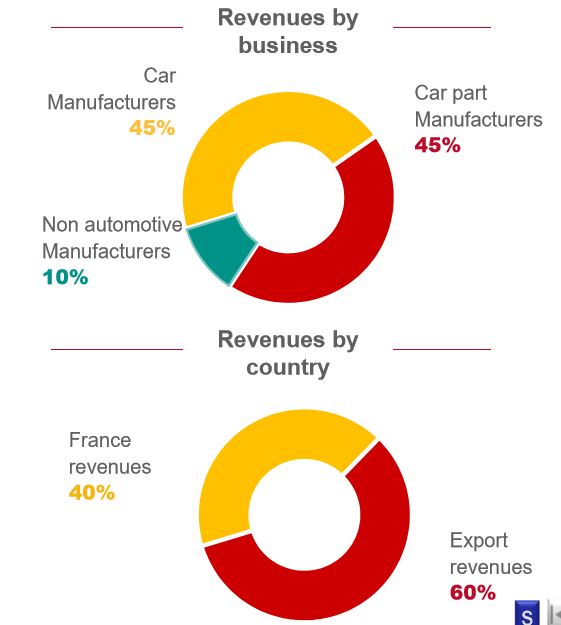


Figure 4 : répartition de la clientèle

FSA peut également être amené à fabriquer des produits quand on en trouve pas dans le commerce comme des presses électriques dans les années 1990, par exemple.

FSA a peu de clients mais il s’agit toujours de gros clients comme par exemple PSA, VALEO, RENAULT. D’où la concurrence très rude pour obtenir les marchés.

Vous pouvoir voir en annexe 1.1 quelques exemples de produits fabriqués par les lignes de Fabricom Systèmes d’Assemblage  et quelques clients de FSA en annexe 1.2.

1. Contexte du projet et besoins du client

L’application « MelodieNet » permet de suivre les lignes de production en temps réel. Développée par FSA, cette application web donne accès à de multiples interfaces dans lesquels la performance de production est détaillée par des chiffres et des Pareto.

Mon but durant ce stage est d’adapter les deux fonctionnalités essentielles de « MelodieNet » à savoir le « suivi machine » et le « suivi de production ». L’un correspondant au « mode de marches des cellules » sur « MelodieNet », l’autre au « bilan de production », lequel sera légèrement simplifié. (Voir annexe 1.1). L’objectif final étant de rendre l’état des lignes de production et leur productivité accessibles n’importe où grâce à un smartphone.

FSA étant une entreprise très active à l’international, l’une des premières contraintes pour l’application a été de pouvoir gérer plusieurs langues, au minimum le français et l’anglais.

De plus, elle doit permettre l’authentification d’utilisateurs en vérifiant les informations entrées dans la base de données.

Les informations des lignes de production affichées par l’application doivent être récupérées sur un Web Service dédié dont l’URL est modifiable à travers un écran de paramétrage de l’application.

Ce Web Service est de type « REST » et il exporte des données au format JSON.

1. Les besoins fonctionnels

L’une des contraintes sur l’application a été de maximiser sa simplicité d’utilisation, mais aussi de présenter les informations de manière lisible.

L’application devait impérativement être internationalisable. En effet le rayonnement international de Fabricom Systèmes d’Assemblage et plus précisément de DOERFER font de la gestion de la langue un paramètre fondamental au sein de l’application.

Au lancement de l’application, nous devrons être dirigé vers un écran d’authentification lequel nous permettra par la suite d’accéder au menu. L’enregistrement (=inscription) d’un utilisateur ne sera pas effectué car le client juge qu’il n’est pas nécessaire pour l’application mobile.

La base de l’application devra être un menu nous permettant de choisir entre trois écrans différents, à savoir le **« Suivi machines »**, le **« Suivi de la production »** et les **paramètres**.

Dans l’écran **« Suivi machine »,** il apparait important d’afficher le **numéro de cellule**, le **libellé de cellule** et enfin le code couleur nous permettant de connaitre **l’état de chaque cellule de la ligne**. Toutes ces informations seront présentées sous la forme d’un tableau que l’on pourra faire défiler si la place pour afficher les informations venait à manquer. De plus, l’affichage doit être « dynamique », c’est-à-dire qu’il doit permettre d’afficher les informations en temps réel par un simple rafraichissement.

Pour ce qui est de l’écran **« Suivi de production »,** il doit suivre le même principe que précédemment pour le **« Suivi machines»,** mais en affichant cette fois la référence de chaque produit mise en relation avec la quantités de produits corrects et la quantité de produits défectueux.

Dans les deux cas, le mobile devra envoyer une requête au service Web avec des paramètres qui permettront au service Web de répondre avec les informations souhaitées.

Enfin l’écran de **paramétrage** doit permettre, dans un premier temps, de modifier l’URL du service web dans l’application. Cette modification doit être sauvegardée dans l’application de manière à être la nouvelle URL même si on quitte l’application.

**Authentification par accès direct à la Base de données**

Application Android

Envoi d’une requête SQL

BDD

Melodie

Jtds-1.2.8 library

Envoi du résultat sous la forme d’un ResultTest

Importation

* **Charges d’utilisation :** 
  + S’authentifier
  + Paramétrage éventuel de l’application
  + Visualisation des informations concernant la production et les modes de marche
  + Rafraichissement des informations par « scrolling »
* **Charges du système :**
  + Charges d’utilisation
  + Simplicité d’utilisation
  + Se connecter au service Web pour afficher les informations sur machines et produits en temps réel.
  + Vérifier l’identifiant et le mot de passe en interrogeant le service Web

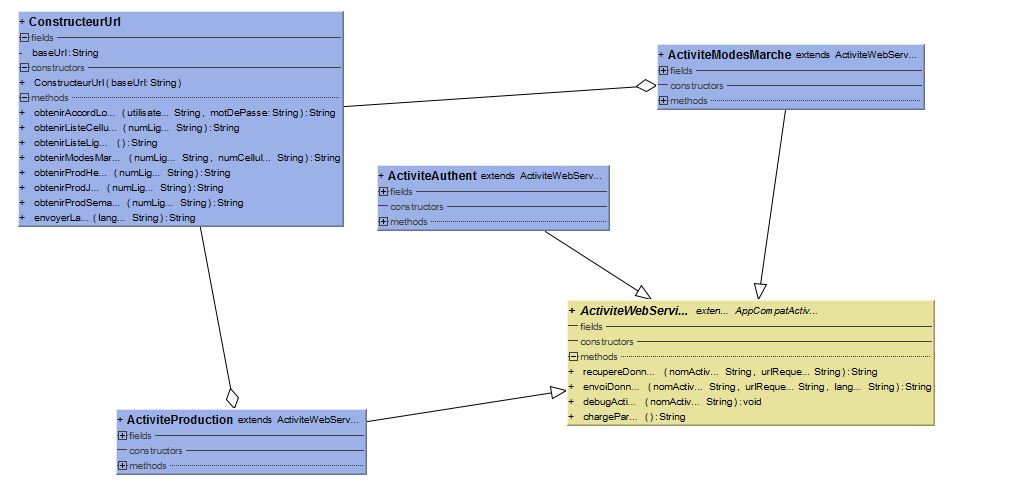
1506 lignes de java

1. Conception de la solution
2. L’élaboration du cahier des charges (modèle théorique de lappli)
3. Explications de fonctionnement

Nous avons choisi de développer l’application sur l’API 19 d’Android, la version 4.4 Kitkat. Donc tout appareil Android de version 4.4 ou supérieure est compatible avec l’application. Pour l’environnement de développement, nous avons choisi Intellij IDEA développé par JetBrains, sa polyvalence, sa simplicité et l’utilisation que nous en avions faite en TP d’Android nous ont orientés vers ce choix. Pour travailler en groupe, nous avons utilisé l’outil github. Nous utilisons le format JSON pour sauvegarder les constats. De cette manière, nous pouvons enregistrer des constats et récupérer des constats déjà enregistrés.

Le lien github du projet est le suivant : <https://github.com/Xenation/ProjetMusee-Montelimar.git>

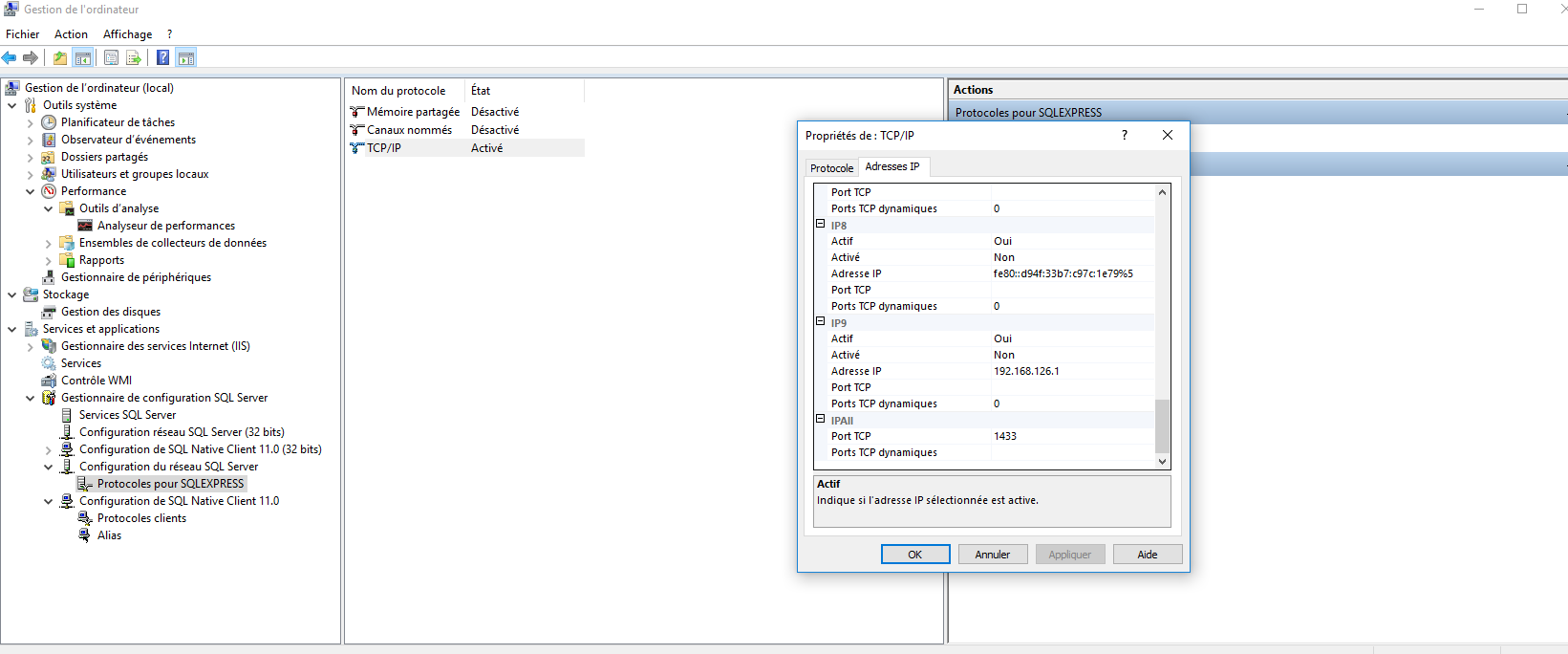
1. Le diagramme de classe(modèle pratique de lappli)



1. Mise en œuvre de la solution
2. Les outils technologiques choisis
3. Les différentes phases de développement

Les deux premiers jours du stage, j’ai réalisé des tests de non-régression sur l’application Web MelodieNet à la demande de mon maitre de stage. L

1. Les difficultés rencontrées et leurs solutions



* 27/04 Il fallait aller dans « Gestion de l’ordinateur » puis aller dans la configuration du réseau « SQL server » puis protocole pour SQLEXPRESS, TCP/IP et mettre à null la valeur de ports TCP dynamique et mettre 1433 dans la valeur port TCP (=statique). En effet cette configuration m’a permis de me connecter à la BDD ce qui plantait avant.
* 02/05 travail sur le hashage du mot de passe par le protocole MD5 lors de la connexion. Le problème est que le mot de passe entré par l’utilisateur que je hash via la méthode hashPassword de la classe Hash ne le crypte pas exactement comme il l’est dans la BDD.. On a « c2fca0234de447157701d0c2d64094 » au lieu de « c2fca02304de447157701d00c2d64094 » pour administrateur+. J’ai donc vérifié via http://md5encryption.com/ lequel était le bon. C’était celui de la base de données. Le soucis provenait de la :

StringBuffer sb = new StringBuffer();

for(byte b1 : byteTable){

sb.append(Integer.toHexString(b1 & 0xff).toString());

}

Il fallait remplacer ça par ça :

//Converts message digest value in base 16 (hex)

md5 = new BigInteger(1, digest.digest()).toString(16);

Le fait de remplacer le StringBuffer par un BigInteger a été déterminant.

* Problème : j’utilisais informUser = R.string.Password\_failure; pour récupérer une chaine dans les ressources au sein du code java, ce qui déclenchait une erreur car R.string.Password\_failure est considéré comme un entier par java. J’ai donc plutôt utilisé

informUser =getResources().getString(R.string.Password\_failure);

ce qui fait bien ce que je désirais, c’est-à-dire mettre le string « Password\_failure » dans informUser.

* Pas possible de se connecter (.connect) au WebService car il faut l’exécuter en tache de fond. Android le bloque par défaut pour ne pas qu’il bloque le thread UI
* Obtenir le string résultat en sortie d’une AsyncTask : 3 paramètres : paramètres, progression et résultat. Il faut donc tout d’abord mettre le résultat comme étant un String puis on fera un .get() à la suite de l’exécution de l’AsyncTask pour obtenir son résultat.

Conclusion

Ce projet a été divisé en plusieurs phases : la conception de l'application, le développement au cours desquels nous avons pris des réunions avec notre client et notre tuteur pour identifier nos taches.

Nous avons également gérer les coûts du projet comme dans le cas d'un projet en entreprise. La répartition des taches via Gantt Project a été décisive dans la gestion de notre temps. Elle nous a permis de maximiser le nombre de fonctionnalités de l'application mais aussi certaines fois de se concentrer sur l'essentiel, les fonctionnalités les plus importantes au détriment d'autres fonctionnalités secondaires et plus gourmandes en temps.

Le projet nous a fait aborder de multiples aspects de la programmation Android : Sur les

images tout d'abord, on nous avons réussi à utiliser l'appareil photo de la tablette pour prendre des photos que l'on a pu afficher dans l'application et modifier avec des calques par la suite.

Ces photos sont des composantes d'un constat. On peut en ajouter au cours de la création ou une fois le constat crée. De plus, nous avons pu apprendre à créer des formulaires complexes, à enregistrer les informations entrées et à les récupérer. Nous avons aussi appris à utiliser le design pattern « proxy » pour la gestion des droits d’accès lors de la connexion sur l’application.

Les perspectives d’évolution de l’application pourraient être l’ajout d’une fonctionnalité permettant de modifier un constat, de compléter un constat pré-rempli. Elles pourraient être également de récupérer les informations concernant l’œuvre

Enfin, pour conclure, nous avons beaucoup appris grâce à ce projet.

La découverte des activités du musée qui avait été l'une des raisons de notre choix, n'a pas été une déception, elle a été enrichissante pour tout le monde. Les expositions qu'il propose sont variées et les mettre en œuvre nécessite une organisation importante entre les deux partis que notre application vise, justement, à simplifier.

D'autre part, modéliser les besoins d'un client, lui poser les bonnes questions pour cerner le cadre de l'application a été une phase fondamentale dans le projet puisqu'elle nous a permis de mettre en application beaucoup de cours afin d'obtenir les informations utiles pour notre projet. Ces informations nous ont permis par la suite de produire la solution la plus proche et la plus adaptée possible aux besoins du client.

De plus, nous avons tous pu acquérir de solides compétences sur la programmation Android et les outils de développement associés. Chacun a des compétences différentes correspondant à son rôle dans le développement mais tout le monde a acquis cette base commune qui lui sera sans doute d'une grande utilité dans des projets Android futurs.

Annexes

* 1. Quelques exemples de produits fabriqués par les lignes de production FSA

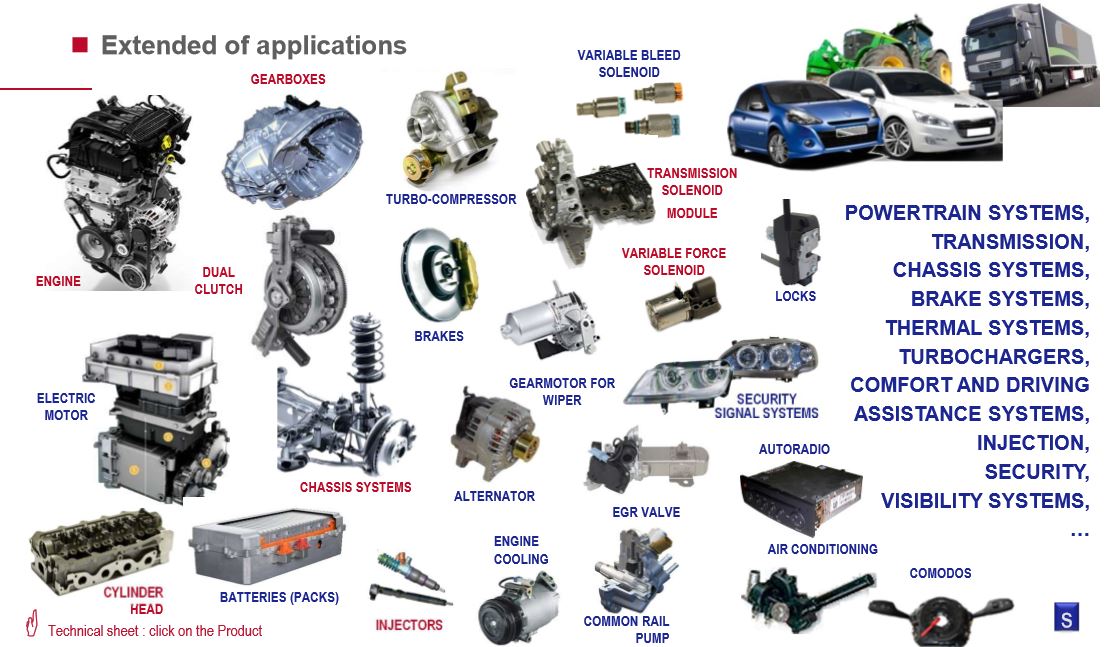


Figure 5 : Produits pour l'automobile



Figure 6 : Produits pour d'autres industries

* 1. Clients de FSA



Figure 7 : Clients

* 1. L’application web « Melodie Net »



Figure 8 Suivi des machines

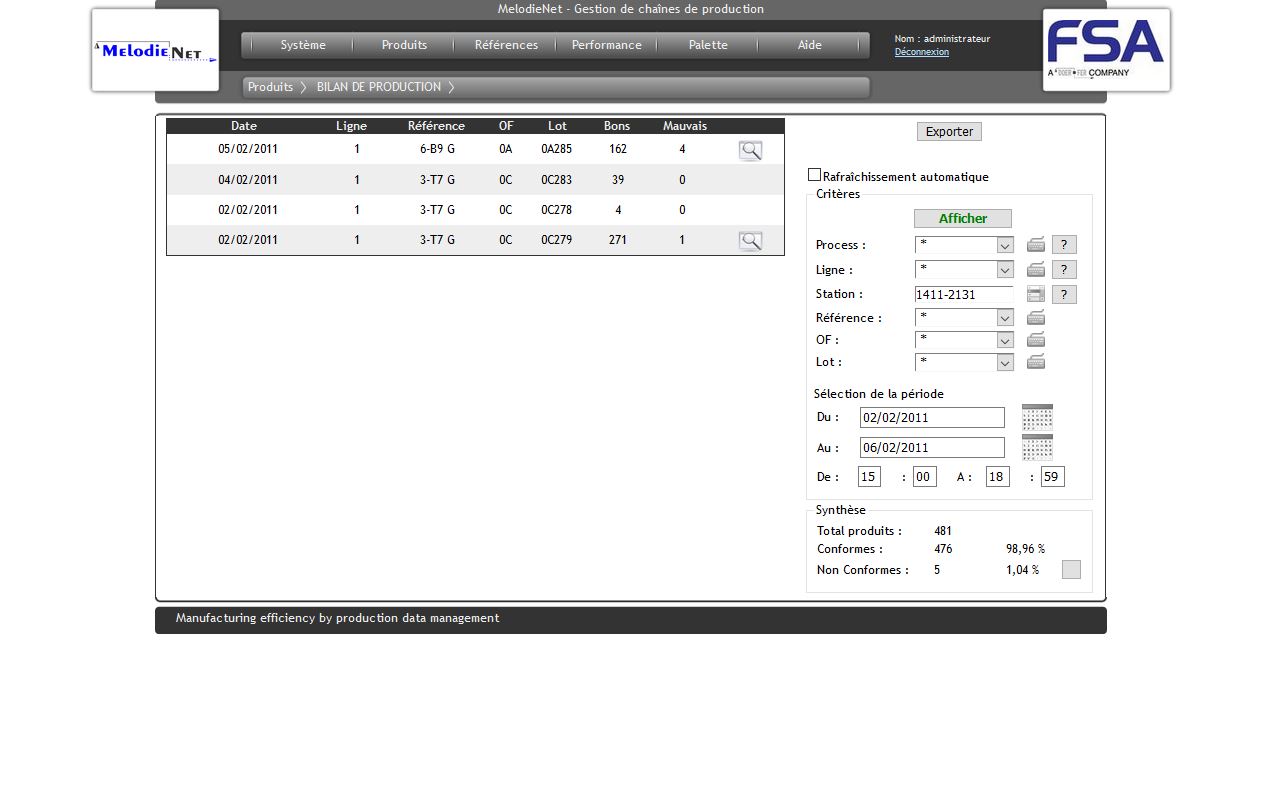


Figure 9 Suivi de la production

* 1. Les diagrammes de l’application

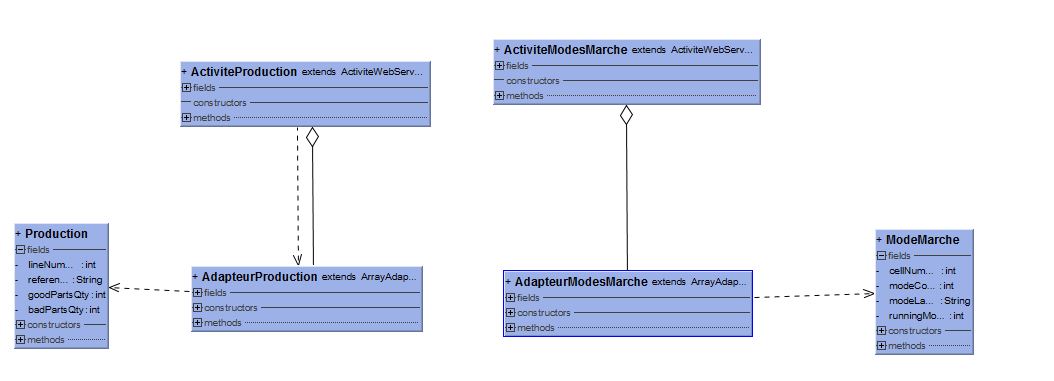


Figure 10 : diagramme de classe pour les listes

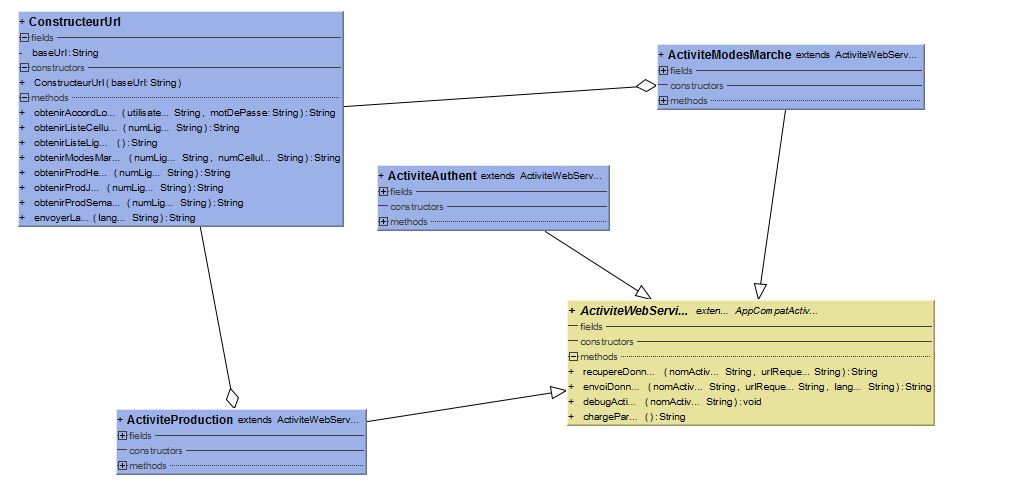


Figure 11 : diagramme de classe de communication avec le service Web

* 1. Les IHM de l’application

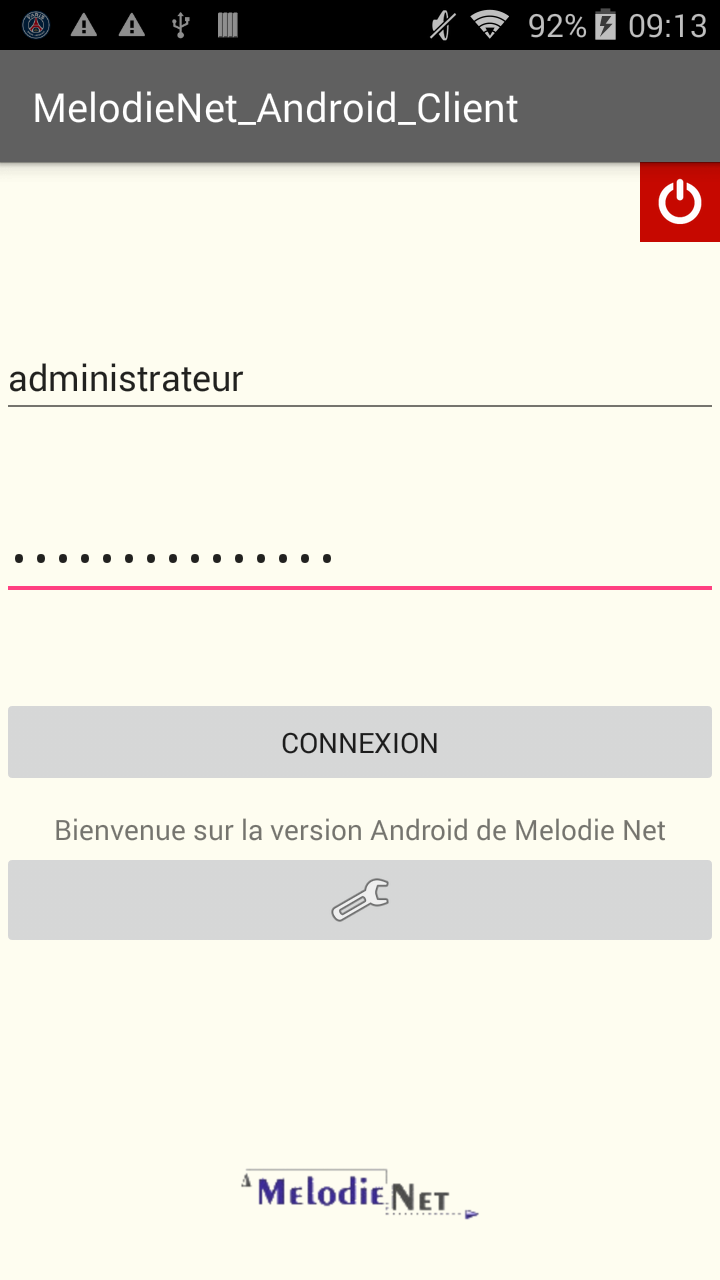
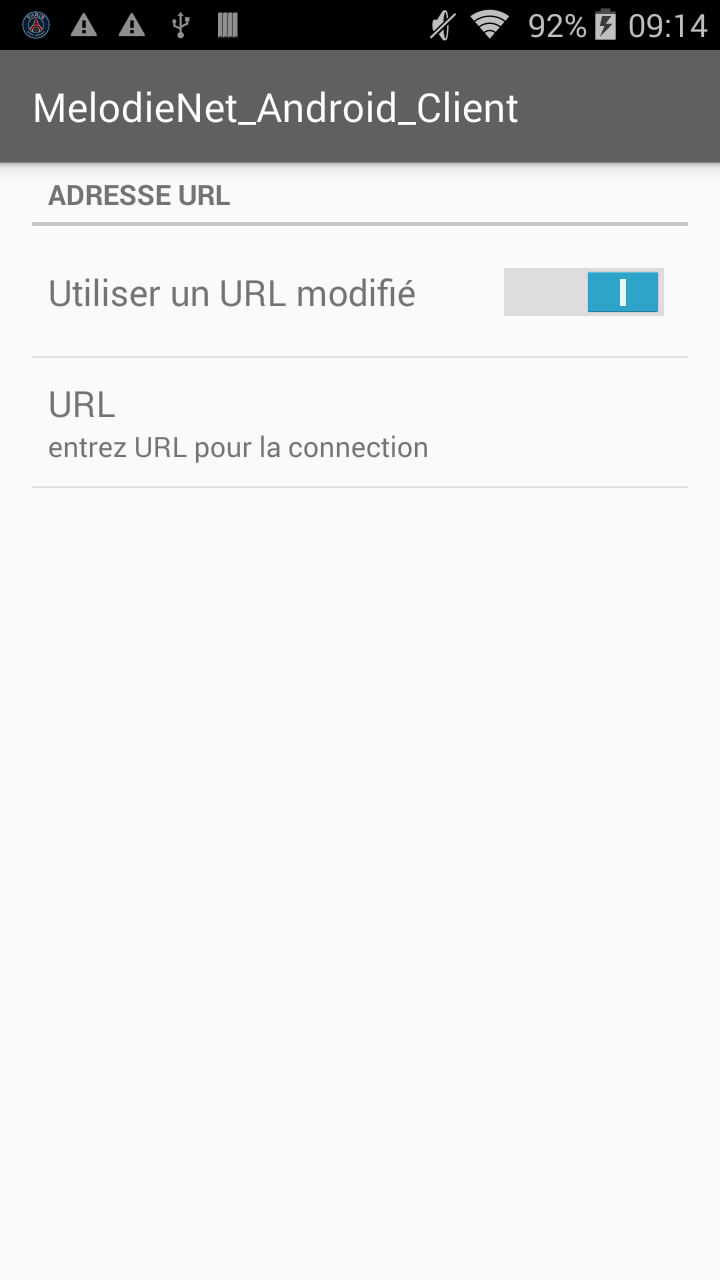
 

Figure 12 : Page de connexion Figure 13 : Vue des paramètres

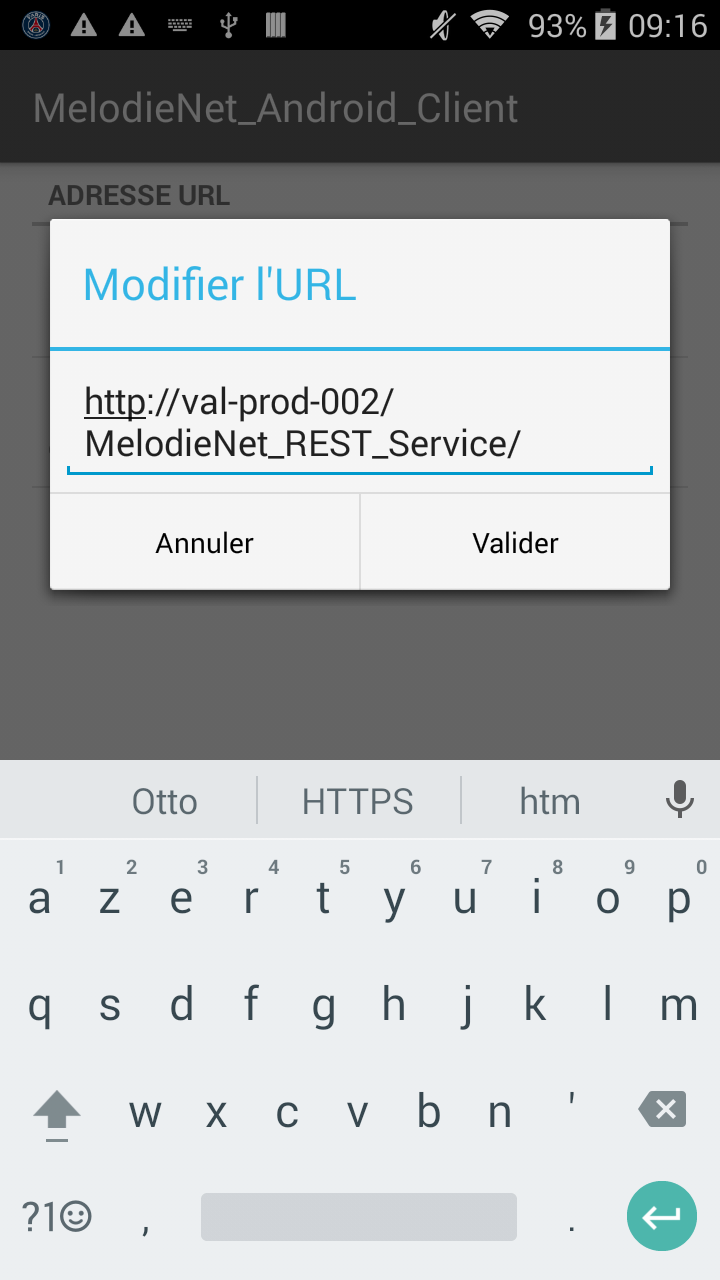
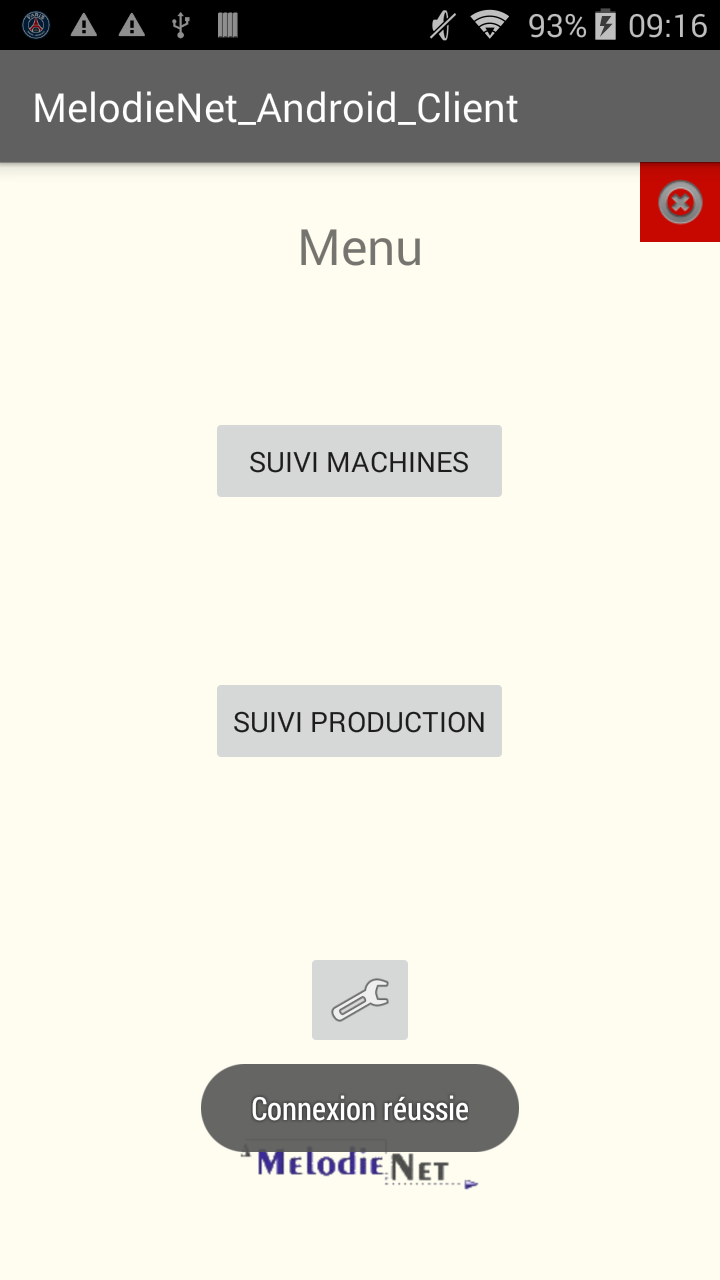
 

Figure 14 : modification de l'URL Figure 15 : menu de l’application

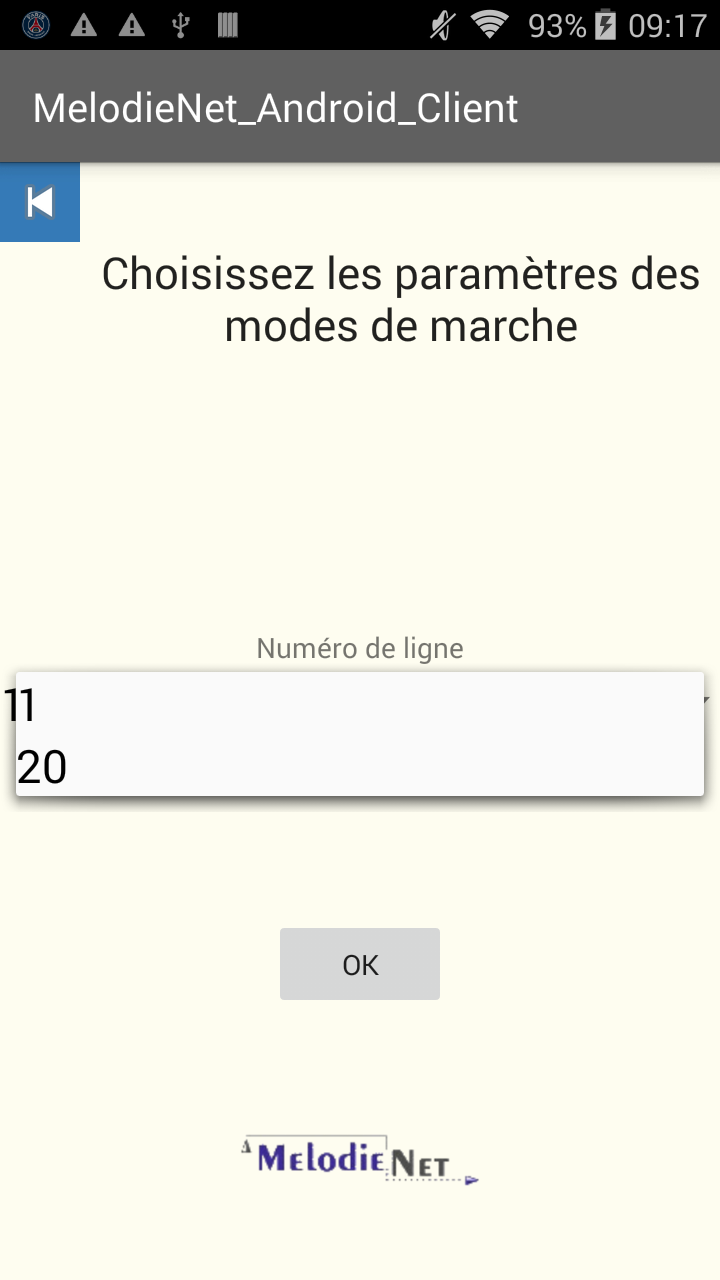
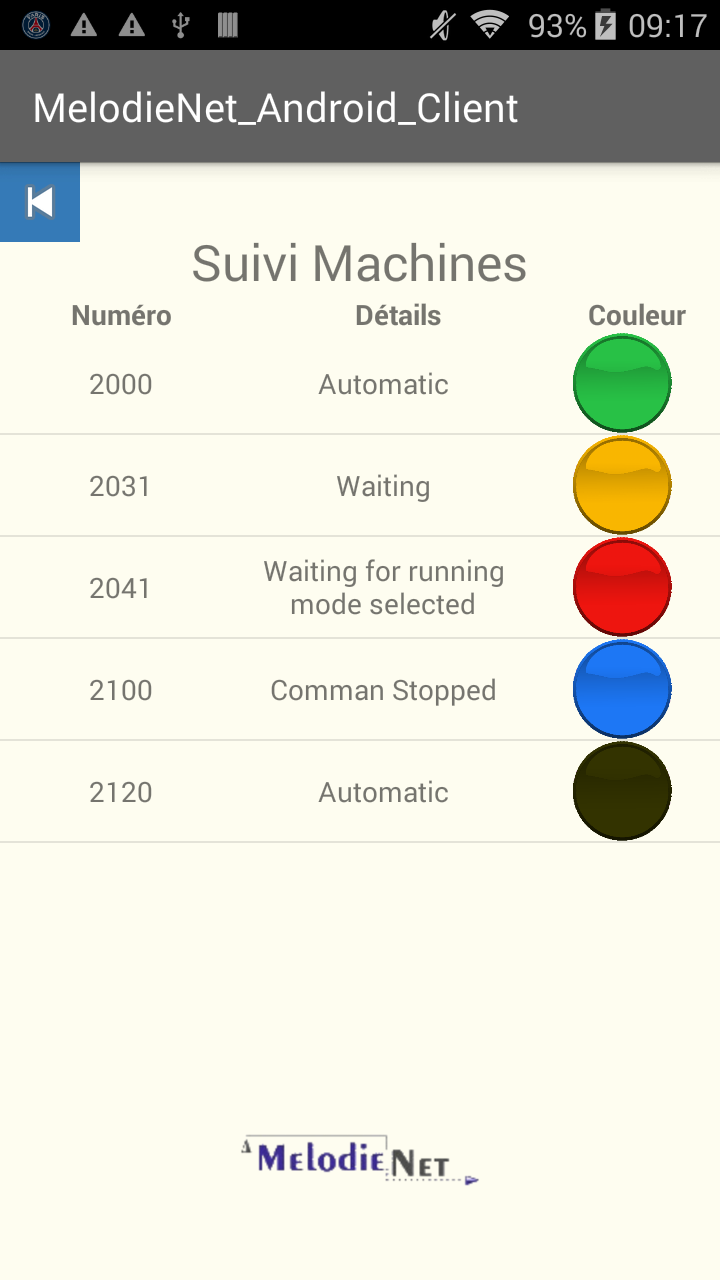
 

Figure 16 : paramètres des modes de marche Figure 17 : Suivi machine

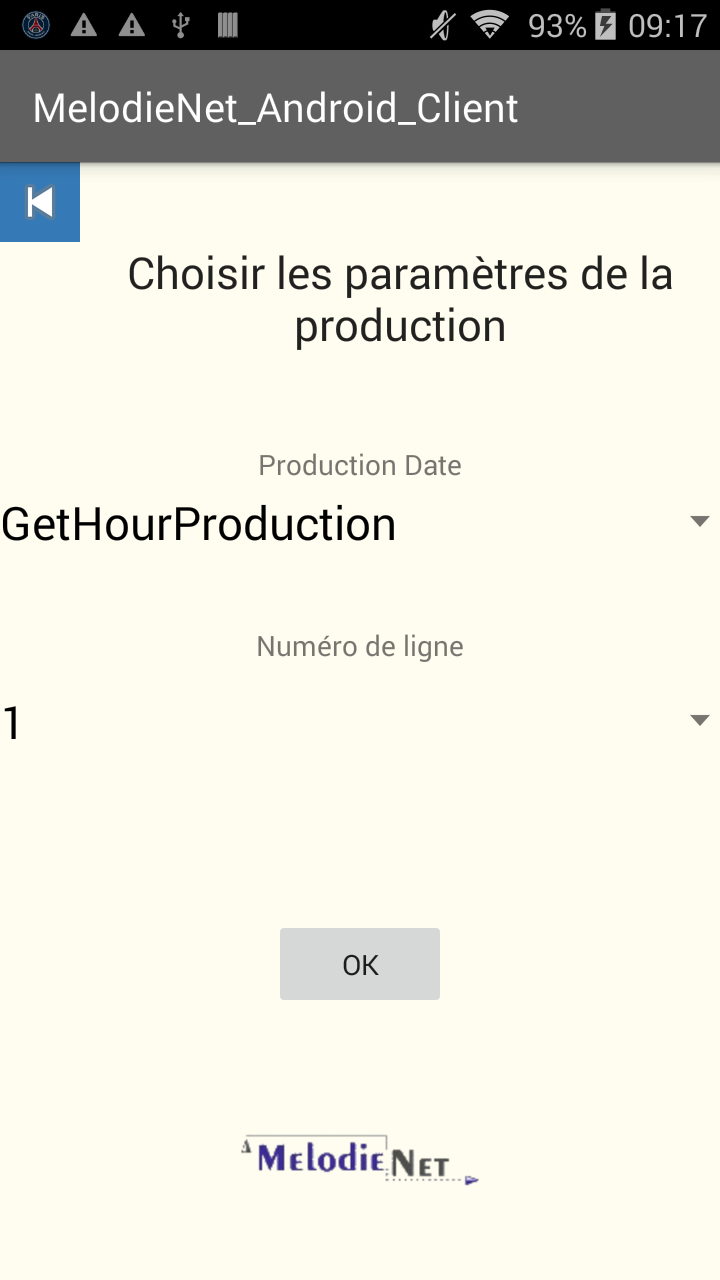
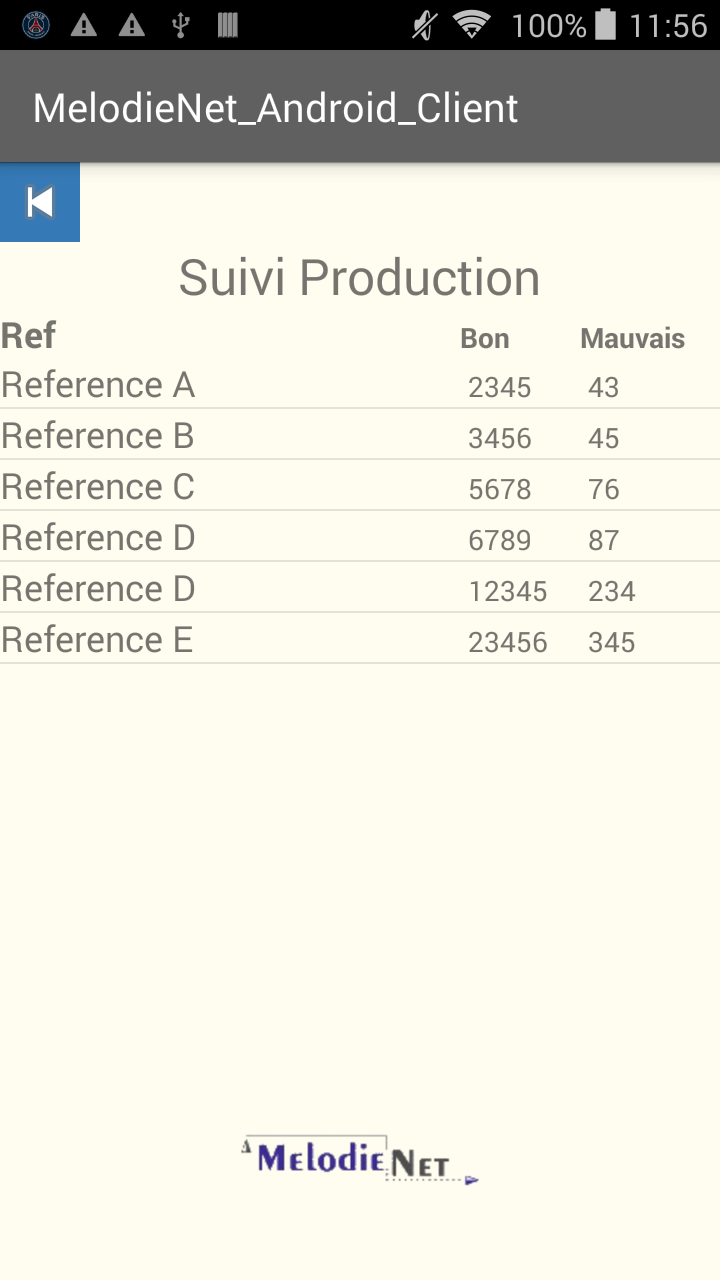
 

Figure 18 : paramètres de la vue production Figure 19 : Suivi de la production