**Rapport de Stage**



Introduction

Pour conclure la formation de DUT informatique, j’ai été amené à faire un stage. 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’Asemblage qui proposer d’adapter une application 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 Système d’Assemblage assemble et créer des lignes 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.

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

Cette application 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, moi sur la partie « Client », c’est-à-dire l’application Android et M. COUPAT sur 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 comment ai-je réalisé une application mobile répondant à ces contraintes ?

Table des matières

[Introduction 2](#_Toc481593003)

[Table des matières 3](#_Toc481593004)

[I) Présentation et objectifs généraux du projet 4](#_Toc481593005)

[A) Présentation de la société 4](#_Toc481593006)

[B) Contexte du projet et besoins du client 4](#_Toc481593007)

[C) Les besoins fonctionnels 5](#_Toc481593008)

[II) Conception de la solution 9](#_Toc481593009)

[A) L’élaboration du cahier des charges (modèle théorique de lappli) 9](#_Toc481593010)

[B) Les IHM 10](#_Toc481593011)

[C) Le diagramme de classe(modèle pratique de lappli) 10](#_Toc481593012)

[III) Mise en œuvre de la solution 11](#_Toc481593013)

[A) Les outils technologiques choisis 11](#_Toc481593014)

[B) Les différentes phases de développement 13](#_Toc481593015)

[C) Les difficultés rencontrées et leurs solutions 13](#_Toc481593016)

[Conclusion 15](#_Toc481593017)

[Annexes 17](#_Toc481593018)

[1.1) L’application web « Melodie Net » 17](#_Toc481593019)

1. Présentation et objectifs généraux du projet
2. Présentation de la société
3. 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 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 machine » 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’URI du service web dans l’application. Cette modification doit être sauvegardée dans l’application de manière à être la nouvelle URI même si on quitte l’application.

* **Charges d’utilisation :** 
  + S’authentifier
  + Paramétrage éventuel de l’application
  + Visualisation des informations concernant la production
  + 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 se connectant à la base de données

Le projet a débuté avec une phase préalable de démarrage visant à définir les objectifs du projet et des membres de l’équipe. Cette première approche nous a amené à décrire le contexte du projet, les spécifications et à nous répartir les rôles. Nous avons également fait des calculs de coûts prévisionnels en tenant compte de nos ressources matérielles et humaines. Enfin, cette phase a été l’occasion pour nous de rencontrer Thomas WIERZBINSKI, le conservateur du musée d’art contemporain de Montélimar. Cette rencontre nous a permis de mieux comprendre les attentes liées au projet et les besoins du musée.

Pour mener à bien ce projet nous nous sommes attribué des rôles, en fonction des compétences de chacun. William MORDOHAY avait le rôle de chef de projet et avait pour mission de fixer les objectifs, de coordonner les actions de notre équipe et d’animer l’équipe de projet. Il était également le relais de l’information et le porte-parole du groupe.

Nicolas JOURDAN était le responsable Gantt, chargé de la conception et de la mise à jour du diagramme de Gantt qui répertoriait les tâches de tout le monde. Nicolas LOGUT était responsable du suivi des coûts, il avait pour mission de synthétiser les heures de travail en se basant sur le diagramme de Gantt préalablement établi et devait veiller au respect des coûts prévisionnels.

Alfred BRISAC et Emmanuel BOUREAU étaient responsables technique. Ils devaient garantir la fiabilité et la performance de notre application et devait s’assurer de la réalisation des objectifs techniques du projet.

Enfin, Andrey LAPCHIK était responsable de la documentation. Il devait concevoir une mise en forme pour tous les documents techniques du projet.

Un rôle précis était initialement attribué à tout le monde mais les tâches plus vastes ont été partagées pour que notre équipe soit polyvalente et acquiert plus d’expérience, notamment pour la partie développement.

Tout d’abord, la conception et l’implémentation du diagramme de classe a été réalisée par William MORDOHAY, Nicolas JOURDAN et Alfred BRISAC. La création et la modification des constats était assurée par William MORDOHAY, Nicolas JOURDAN et Nicolas LOGUT. L’affichage de la liste des constats et la gestion de la suppression de ces derniers étaient à la charge de William MORDOHAY et Andrey LAPCHIK.

De plus, Andrey LAPCHIK devait s’occuper de la validation du constat par signature et de la génération d’un document PDF. Enfin, la page de connexion, la gestion des droits d’accès, l’enregistrement des constats en JSON, la modification des images, la mise en relief des dégradations, ainsi que la prise de photo étaient à la charge des responsables techniques, Alfred BRISAC et Emmanuel BOUREAU.

La rédaction des documents était une mission partagée entre William MORDOHAY, Nicolas LOGUT et Andrey LAPCHIK. De plus, la rédaction du rapport de projet nécessitait beaucoup de temps en comparaison avec le développement, cela a donc mobilisé les trois rédacteurs de l’équipe simultanément.

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

Une fois que nous avions définis le cahier des charges nous avons pu démarrer la phase de production de l’application. Cette phase visait à mettre en place les différentes fonctionnalités de l’application. Nous avons donc attribué des tâches aux différents membres du groupe à partir de la semaine du 2 janvier. Cette semaine nous avons écrit les principales classes de l’application en nous basant sur le diagramme de classe.

La réalisation de la page de connexion s’est étendue de la semaine du 9 janvier à la semaine du 16 janvier. La gestion des droits a donc rapidement été mise en place. Nous avons réalisé la page de création de constat et ses fonctionnalités à partir du 9 janvier et nous avons l’avons achevé la semaine du 6 février. La modification de constat a été mise en place la semaine du 13 février et s’est terminée la semaine du 20 février. Le système de suppression de constat a été vite implémenté la semaine du 13 février. La programmation du menu principal comportant la liste des constats et de l’IHM correspondant a débuté la semaine du 9 janvier et a pris fin la semaine suivante.

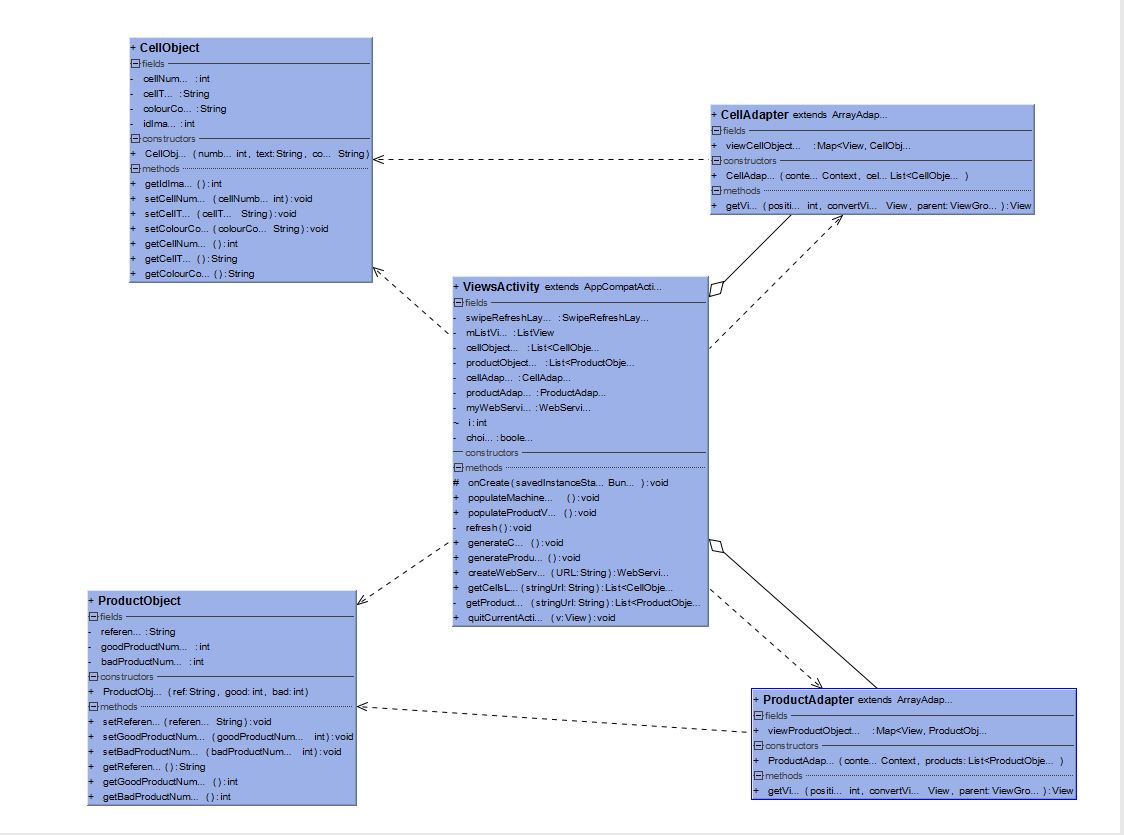
Une fois les principales fonctions réalisées nous nous sommes penchés vers les fonctionnalités plus spécifiques. Le développement du système de signature a commencé le 6 février et s’est étendu jusqu’au 20 février. La partie prise de photo et leur sauvegarde a débuté la première semaine de mars et c’est terminé la semaine du 13 mars. L’exportation des constats au format PDF a débuté la semaine du 13 février a été achevée la semaine du 6 mars. La mise en place du système d’annotation des dégradations a débutée le 6 mars et s’est terminée le 20 mars. Du point de vue du développement, la semaine du 3 avril était dédiée aux tests des différentes fonctionnalités.

1. Les IHM

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

Actuellement, nous avons la possibilité de définir les 3 différents mots de passe pour chaque utilisateur au premier lancement de l’application. Ensuite nous avons une page de connexion dans laquelle chaque mot de passe nous permet d’accéder à la grille de constats.

Sur cette grille, la sélection d’un constat permettra à l’utilisateur de consulter un constat dans lequel tous ses champs seront visibles. Dans cette vue, on pourra également ajouter des images à ce constat à l’aide d’un bouton « AddPhoto » ou encore cliquer sur le bouton de validation de constat qui nous demandera ensuite deux signatures : celle du préteur et celle de l’emprunteur. Ces deux signatures génèreront deux photos correspondant à chaque signature et sauvegardées en interne sur l’appareil Android./// +pdf ???

Hormis cela, nous pouvons supprimer chaque constat par l’appui sur le bouton « croix » du constat à supprimer. Une fenêtre s’ouvrira pour confirmer le choix et éviter les suppressions accidentelles. Après validation du choix, un message de type « toast » s’affichera à l’écran informant l’utilisateur de l’application de la suppression du constat. Le constat disparaitra de la grille de constat et sera également supprimé sur la tablette dans le fichier dans lequel il était stocké.

Sur la vue dans laquelle tous les constats sont rangés en grille, nous avons également la possibilité de créer un constat en cliquant sur le bouton « créer constat » en haut à droite de la vue. Cela nous emmènera sur une page nous demandant de sélectionner le type d’œuvre sur lequel porte le constat : Peinture, Dessin, Sculpture, Mobilier ou Autre.

Cela nous emmènera alors vers le formulaire adapté à l’œuvre choisie ou dans le cas de la sélection « Autre », vers un formulaire très court dont les champs sont personnalisables par l’utilisateur. Dans tous les cas, l’utilisateur pourra compléter les champs déjà prévus et en ajouter si besoin.

L’appui sur le bouton « Validate Report » permettra de créer un constat à partir des champs que l’utilisateur a rempli. Nous serons ensuite redirigés vers la consultation du constat dans lequel la visualisation des informations renseignées et l’ajout de photos au constat seront possible.

Lorsque nous choisirons d’ajouter une photo au constat, il suffira de cliquer sur le bouton correspondant qui ouvrira l’appareil photo de l’appareil Android et enregistrera la photo dans le constat après demande de confirmation à l’utilisateur. Cette photo sera rattachée au constat et permettra d’identifier l’œuvre auquel le constat se rattache si c’est la première photo prise.

Plusieurs photos peuvent être prises et les boutons « next » et « previous » permettent de passer d’une photo à l’autre.

Enfin, nous pouvons ajouter des calques à une image, mais aussi les cocher et les décocher afin de choisir si ils seront visibles ou non. On peut modifier des calques en appuyant sur le bouton « EDIT », ce qui nous permettra d’accéder au menu regroupant toutes les légendes du calque.

Dans ce menu des légendes, nous avons toujours la possibilité de changer le nom du calque auquel se rattache les légendes. Nous avons ensuite trois boutons : deux pour les différentes formes avec lesquelles on peut entourer une dégradation et un pour déplacer l’une des forme d’une légende sur l’image. Ces fonctionnalités ne seront disponibles qu’après avoir sélectionné une des légendes de la liste de légendes qui se situe dessous.

L’appui sur le carré de couleur d’une légende permettra de modifier la couleur de toutes les formes d’une légende.

Pour les légendes, on peut modifier le nom de chaque légende en appuyant sur un deuxième bouton « EDIT » après avoir sélectionner la légende à modifier.

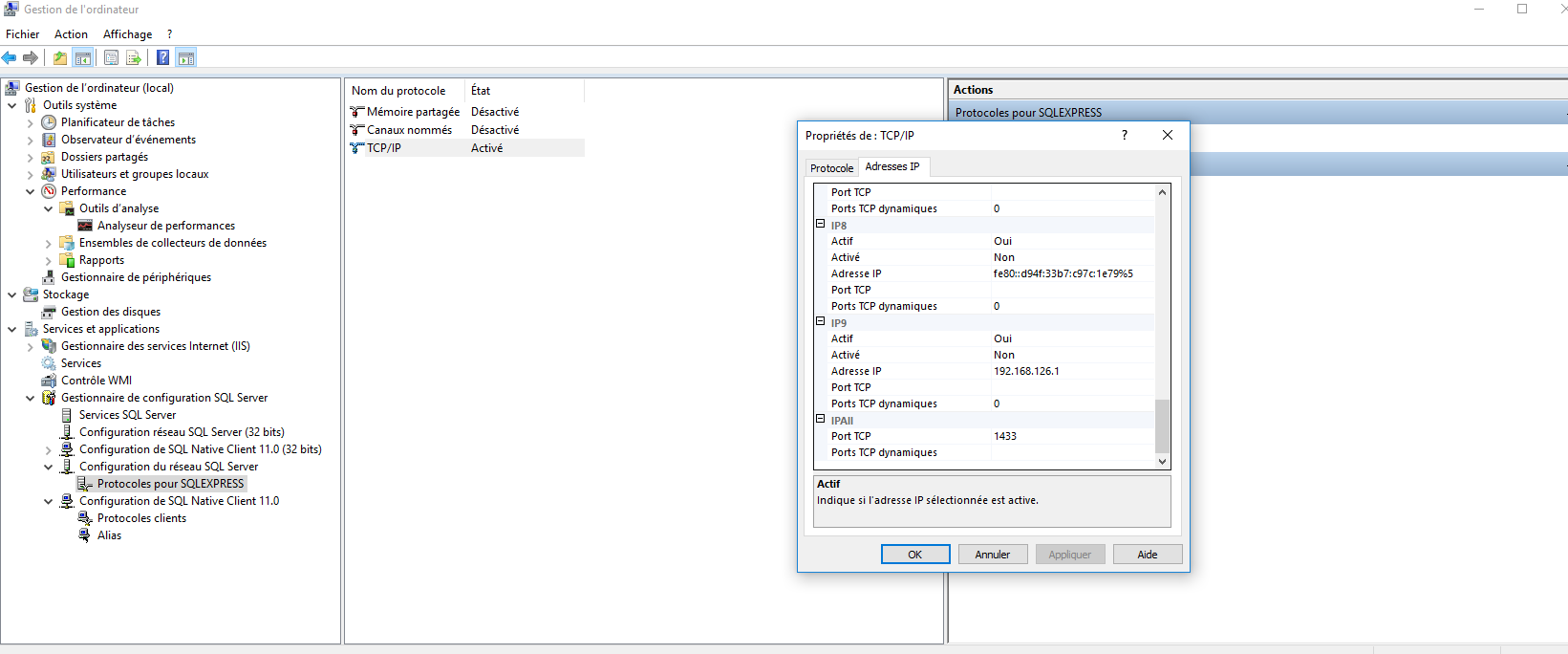
On peut donc déplacer les formes d’une légende, en créer, les placer, changer la couleur de ces formes et changer le nom de la légende.

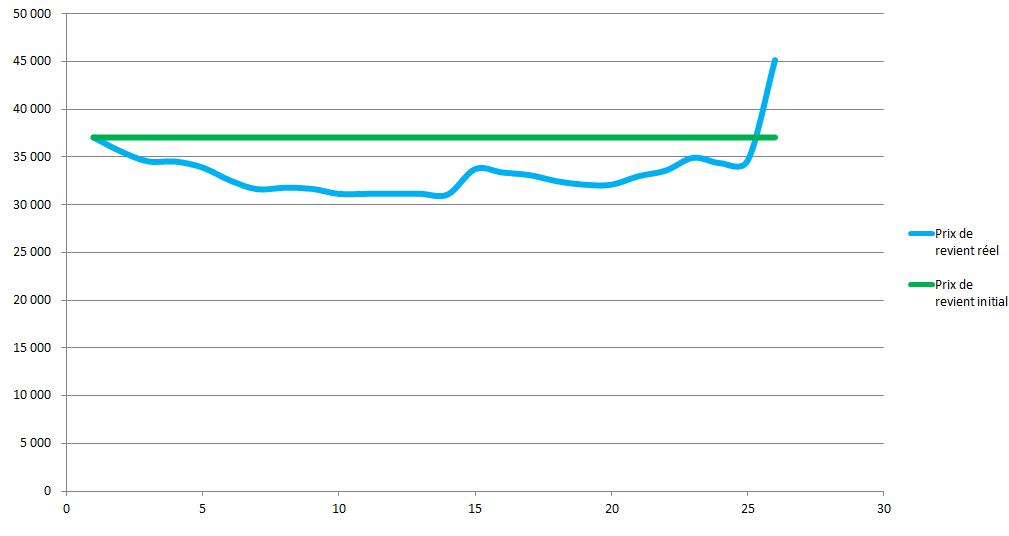
On peut également ajouter des légendes supplémentaires en cliquant sur le bouton « AJOUTER LEGENDE » qui nous demandera de donner un nom à la légende en cours de création et lui attribuera une couleur aléatoire et donc modifiable si besoin.

1. Les différentes phases de développement

L’état final prévu regroupait toutes ces fonctionnalités, en y ajoutant la modification de constats. La fonctionnalité permettant de générer des pdf n’a pas été implémentée faute de temps. Toutefois, la vue permettant d’annoter des dégradations sur l’image est bien plus complète que prévu. En ayant la possibilité de déplacer des formes déjà créés, de sauvegarder les dernières actions effectués sur l’image et de les refaire nous avons dépassé les attentes du client à ce niveau-là.

1. Les difficultés rencontrées et leurs solutions



Nous avons suivi l’évolution des coûts tout au long du projet en essayant de rester réguliers dans notre travail. Nous avons néanmoins constaté une baisse du coût de revient réel tout au long de la première moitié du projet. Cela s’explique simplement. En effet nous n’avons pas réalisé autant d’heure que nous en avions prévue au niveau des coûts prévisionnels. Ceci s’explique par le fait que nous n’avons pas eu besoin d’autant de temps pour être à jour sur notre travail et pour mener à bien toutes les tâches.

Nous constatons également une augmentation du coût de revient au niveau de la 14ème semaine de projet. En effet cette semaine était entièrement dédiée à l’avancement du projet, nous avons donc travaillé plus par rapport aux autres semaines. Cela donne un sens à cette remontée soudaine de la courbe du prix de revient réel. De plus, nous remarquons que la deuxième moitié du projet présente elle aussi une courbe en légère baisse. En effet il était plus difficile de consacrer du temps au projet en raison des nombreux travaux universitaires que nous avions à réaliser en parallèle.

Pour finir, en analysant la dernière semaine de projet nous notons une hausse significative du prix de revient réel. Cela est dû au fait que cette semaine était également consacrée entièrement au projet, nous avons donc travaillé beaucoup plus que les autres semaines. Ce pic de productivité a amené le prix de revient réel à dépasser le prix de revient initial de 8090€. Il n’y a rien à conclure, ces chiffres étaient de toute façon des prévisions.

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. L’application web « Melodie Net »



Figure 1 Suivi des machines

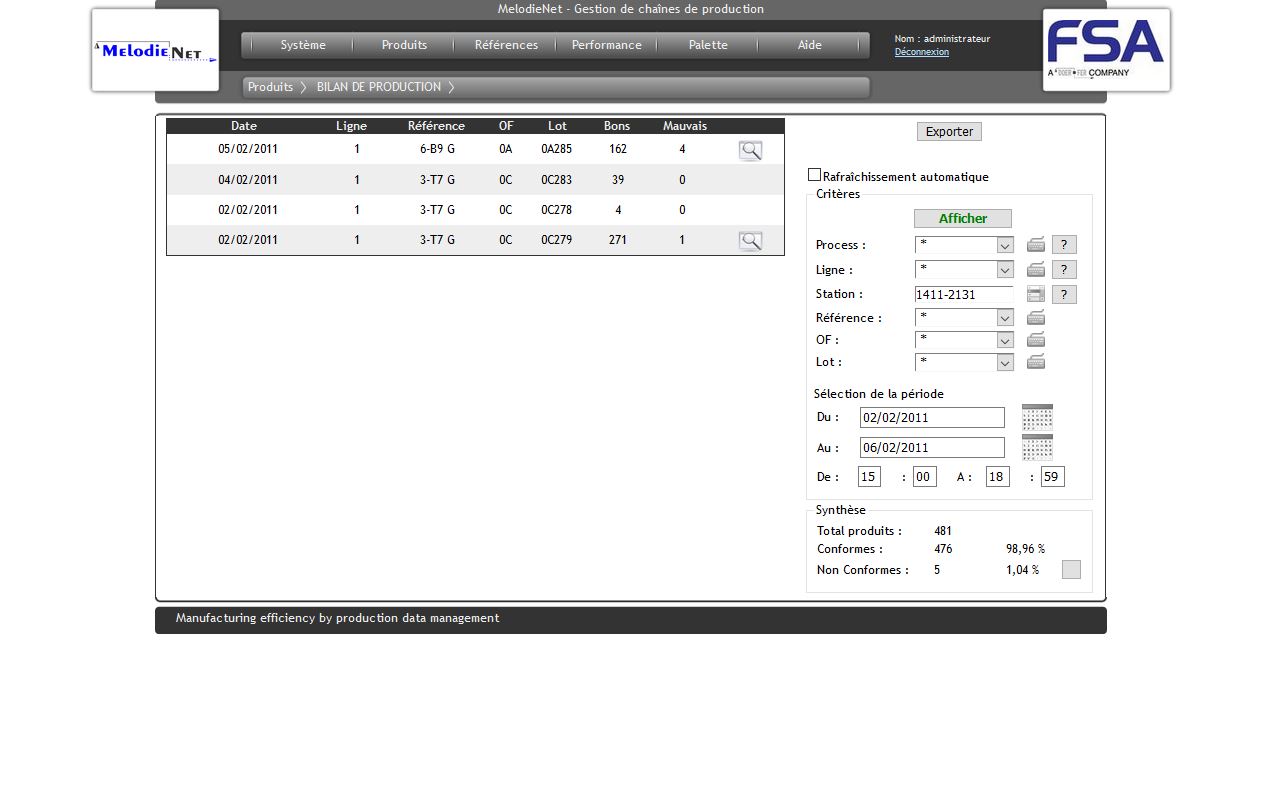


Figure 2 Suivi de la production