#### Macintosh HD:Users:mosser:Documents:Resources:logos:Polytechnice.pngMacintosh HD:Users:mosser:Documents:Resources:logos:UNS.png

**D1.2 – Management (MGMT)**

**Projet de Fin d’Études**

**2013 – 2014**

[E]

SPFE-57 : Atelier IHM Gestion des emprunts de matériel

Participant(s) :

* Laurent, Alexis, alaurent@polytech.unice.fr, SI5 (IHM)
* Paeta, Suzy, paeta@polytech.unice.fr, SI5 (IHM)
* Roufast, Romain, roufast@polytech.unice.fr, SI5 (IHM)

Encadrant(s)

* Déry-Pinna, Anne-Marie, [pinna@polytech.unice.fr](mailto:pinna@polytech.unice.fr), I3S Rainbow
* Brel, Christian, [brel@polytech.unice.fr](mailto:brel@polytech.unice.fr) , I3S Rainbow

Coût du livrable : **???** heures

Budget total du projet : 948 heures

Table des matières

1. Description du Projet 3

2. Synthèse des résultats obtenus 4

Objectif #1 4

Objectif #2 4

Objectif #3 4

3. Implication des ressources 5

Ressource #1 – Alexis LAURENT 5

Ressource #2 – Suzy PAETA 5

Ressource #3 – Romain ROUFAST 5

Ressource #4 – Christian BREL 5

Ressource #5 – Anne-Marie DERY PINNA 5

4. Synthèse des livraisons 6

5. Suivi budgétaire 7

Consommation du budget 7

Synthèse 7

6. Suivi des lots 8

Lot #1 : Management 8

Lot #2 : Analyse utilisateurs et maquettes 9

Lot #3 : Conception 9

Lot #4 : Implémentation 10

Lot #5 : Tests utilisateurs 10

7. Synthèse & Retour d’expérience 11

Annexe 1. Compte rendu du premier coaching. 12

Annexe 2. Compte rendu du second coaching. 13

# Description du Projet

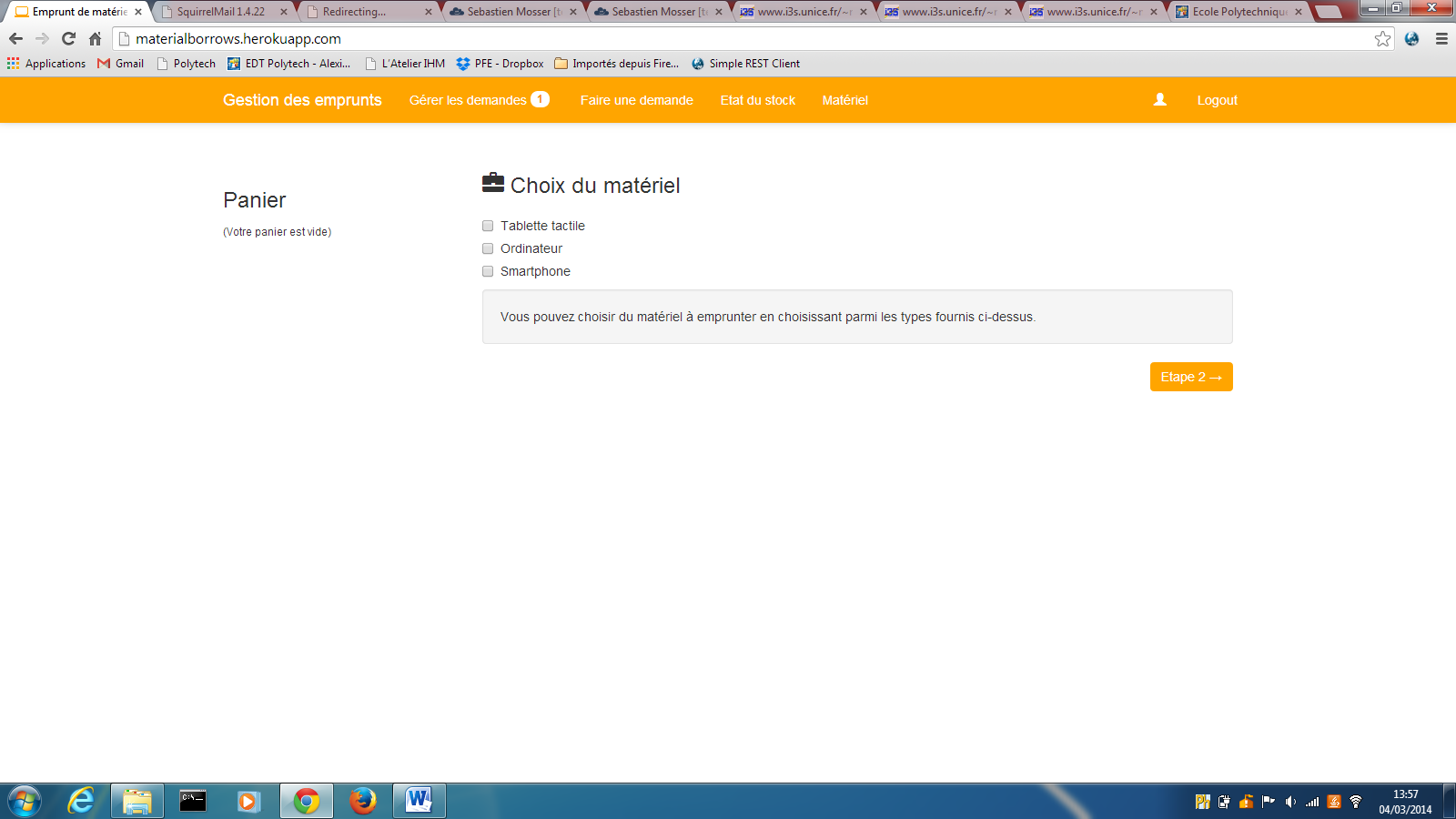
Notre projet s’intègre dans le cadre du Campus Sophia Tech. Il est issue d’un besoin concret : fournir une solution permettant de gérer les suivis des emprunts de matériel (ordinateurs, tablettes tactiles, Smartphones…), effectués par les enseignants et/ou les étudiants. Ces emprunts peuvent être nécessaires dans le cadre de travaux dirigés (pour les enseignants), ou pour le développement de projets (pour les étudiants). Pour l’heure, il n’existe pas de véritable système d’emprunts informatisé dans l’école, les responsables du matériel ont leur propre manière de gérer ce genre de situations décrite sur cette page : <http://atelierihm.unice.fr/emprunt-de-materiel-en-salle-ihm/>. Les systèmes déjà existants tendent à ne pas satisfaire les besoins, tant ceux-ci sont spécifiques en terme de besoins utilisateur.

Ce projet a deux buts précis. Le premier est de permettre aux emprunteurs d’effectuer très simplement, de manière rapide et à distance des demandes d’emprunt. Ils auront la possibilité de visualiser le stock disponible pendant une période spécifique. Les demandes seront ensuite directement traitées par les responsables.

Le second but de ce projet est de permettre aux responsables du matériel d’avoir un système automatisé et fiable. Ils auront ainsi plus de facilité à traiter les différentes demandes des emprunteurs. L’application leur permettra de gérer tout cela à distance, sans avoir à consulter les emprunteurs, sauf s’ils le désirent.

De plus, ils auront la possibilité de gérer le stock via l’application, ce dernier n’étant pas correctement gérer actuellement. Ils pourront ainsi ajouter, modifier ou supprimer du matériel. Ils pourront aussi fixer le stock disponible à l’emprunt.

Enfin, comme n’importe quel utilisateur de l’application, il leur sera possible d’effectuer une demande.



*Figure 1 : Page d’accueil de l’application – utilisateur ‘Admin’*

# Synthèse des résultats obtenus

[Public : Non-Spécialiste. Longueur : Entre 1 et 3 pages]

Tableau 1 – Synthèse des objectifs

|  |  |  |
| --- | --- | --- |
| # | Objectif | Statut |
| 1 | Trouver au moins une technologie adaptée pour répondre à l’homogénéité graphique et fonctionnelle, voulue sur plusieurs types de supports | Atteint |
| 2 | Etablir un système de privilèges | Atteint |
| 3 | Automatiser au maximum les entrées/sorties de matériel | Atteint |

## Objectif #1

Nous avons choisi d’utiliser le framework Ruby on Rails. Celui-ci est en effet adapté pour faire une application fonctionnelle sur plusieurs supports. Il nous permet d’une part de gérer un serveur avec une base de données et d’autre part de réaliser une interface client « responsive », qui s’adapte à toute taille d’écran grâce aux technologies du web (HTML et CSS).

Cette technologie répond donc à notre objectif. Toutefois, même s’il est possible de faire une application « responsive », nous n’avions pas le temps de faire une interface utilisable sur ordinateur et smartphone. De ce fait, en utilisant le système des KPI présenté par notre coach lors de notre premier entretien, nous avons décidé qu’il était prioritaire d’avoir une version web fonctionnelle. Ainsi, certaines parties de l’interface ne sont pas tout à fait « responsive ».

## Objectif #2

Ruby on Rails nous permet de gérer les sessions c’est-à-dire qu’à tout moment nous savons qui est connecté sur l’application. A chaque utilisateur enregistré dans la base de données est associé un privilège. Le langage nous permet d’effectuer une vérification avant d’appeler une fonction. De ce fait, nous avons très facilement implémenté une méthode vérifiant le statut de l’utilisateur courant avant de lui permettre d’accéder à certaines fonctionnalités de notre système.

Les responsables auront un statut responsable. Un utilisateur normal n’aura alors accès qu’aux fonctionnalités « Faire une demande » et « Gérer les demandes », tandis que les responsables pourront en plus accéder aux fonctionnalités de gestion du stock. Notre objectif a donc été correctement atteint.

## Objectif #3

Actuellement, les responsables n’ont pas de véritables moyens de connaître l’état du stock. Ils ne peuvent donc pas savoir quel est le stock emprunté, en réserve, disponible, etc. A cela s’ajoute le fait que lors d’un emprunt, le matériel emprunté n’est pas toujours enregistré.

Ainsi, nous avons implémenté différentes fonctionnalités pour pallier ce problème. Tout d’abord, nous avons une fonctionnalité qui permet de visualiser l’ensemble du stock. Le responsable aura une vision de chacun des matériels du stock, incluant leurs caractéristiques ainsi que leur état (emprunté, disponible, inutilisable,…).

Une seconde fonctionnalité permet à un responsable de visualiser le détail d’un matériel. Ainsi, en plus de ses caractéristiques, on y retrouve un historique des derniers emprunts effectués. Les responsables peuvent donc savoir qui sont les dernières personnes à avoir emprunté ce matériel.

Pour finir, pour donner le matériel ou récupérer le matériel d’un emprunt, il est nécessaire de passer par l’application. Le responsable a seulement besoin de scanner le matériel correspondant à l’emprunt et à la réception. L’application se charge ensuite de modifier son état en base données.

L’ensemble de ces fonctionnalités fournissent aux responsables une gestion complète du stock ainsi qu’une automatisation des entrées/sorties de matériel, garantissant l’intégrité du stock. Notre objectif a donc été atteint.

# Implication des ressources

## Ressource #1 – Alexis LAURENT

Alexis a participé à la phase d’analyse et de conception en amont du développement avec les autres membres du groupe. Une fois terminé, il s’est occupé de l’implémentation

## Ressource #2 – Suzy PAETA

## Ressource #3 – Romain ROUFAST

## Ressource #4 – Christian BREL

## Ressource #5 – Anne-Marie DERY PINNA

# Synthèse des livraisons

[Public : Non-Spécialiste. Longueur : 1 page]

Tableau 2 – Synthèse des livraisons

|  |  |  |  |
| --- | --- | --- | --- |
| Livrable | Nom du livrable | Prévu | Livré |
| 1.1 | Cahier des charges (DOW) | S4 | S4 |
| 1.2 | Rapport de management | S21 | S21 |
| 1.3 | Préparation de la soutenance | S21 | S21 |
| 2.1 | Analyse utilisateur | S8 | S8 |
| 2.2 | Maquettes | S8 | S8 |
| 3.1 | Cahier des charges fonctionnel | S8 (v1) - S20 (v2) | S21 (v1) |
| 4.1 | Code source | S19 (v1) - S20 (v2) | S21 (v1) |
| 5.1 | Rapport de retour sur tests utilisateurs | S20 (v1) - S21 (v2) | Pas livré |

\*En rouge *Date de fin* de la première itération et en vert *Date de fin* de la deuxième itération.

Les livrables associés aux lots 1 et 2, correspondant respectivement à la phase d’analyse et de conception ont été rendu dans les temps. En revanche, nous avons rencontré quelques difficultés pour la rédaction du Cahier des charges. Entre le tri des différentes fonctionnalités (primaires et secondaires) et la conception des diagrammes difficiles a fait que nous perdions trop de temps. Le problème venant du fait que nous avions du mal à décrire comment nous allions implémenter ces fonctionnalités, nous avons décidé de remettre sa rédaction à plus tard le temps de commencer la phase d’implémentation pour avoir une meilleure vision.

Lors de la phase d’implémentation, nous nous sommes rendu compte que nous n’avions pas correctement jugé le temps nécessaire à l’implémentation de nos fonctionnalités nous demandant plus de temps. Notre retard nous a empêchés de faire les tests utilisateurs prévus. Cela a aussi impacté sur la seconde itération qui n’a donc pas été effectué.

# Suivi budgétaire

[Public : Non-Spécialiste. Longueur : 1 page]

## Consommation du budget

[Sur la base des feuilles de temps, donnez un graphe cumulé de votre budget (temps) prévisionnel, ainsi que de votre consommation réelle. Chaque colonne représente le nombre d’heure **depuis le début du projet**.]

## Synthèse

[Donnez un paragraphe expliquant de manière synthétique les déviations entre votre budget prévisionnel et votre consommation réelle.]

# Suivi des lots

[Public : Spécialiste. Longueur : 1-2 pages par lot]

## Lot #1 : Management

[Pour chaque lot, reprenez du DoW les objectifs associés. Pour chaque tâche, rappelez de manière synthétique sa description, et expliquez clairement la différence prévu/consommé. Vous pouvez expliquer dans les termes des spécialistes de votre domaine. Identifiez clairement les liens entre objectifs du lot, résultats obtenus, en fonction de vos critères de succès.]

[Pour chaque livrable, donnez une description synthétique de son contenu, sa date planifiée et sa date de livraison réelle. Expliquez les déviations de livraison du point de vue client.]

## Lot #2 : Analyse utilisateurs et maquettes

## Lot #3 : Conception

## Lot #4 : Implémentation

## Lot #5 : Tests utilisateurs

# Synthèse & Retour d’expérience

[Public : Non-Spécialiste. Longueur : 2 pages max.]

[Décrivez de votre point de vue le déroulement de votre projet. Identifiez ce qui s’est bien passé, ce qui s’est mal passé, et ce qui aurait pu mieux se passer. Pour chaque point faible identifié, donnez des pistes pour pallier le problème dans un futur projet.]

Durant notre projet nous avons rencontré certaines difficultés mais nous avons aussi appris beaucoup de choses.

Premièrement nous avons choisi un projet qui implique une participation des utilisateurs. Pour commencer notre projet nous avons effectué des analyses utilisateurs pour identifier leur besoin. Ces analyses se sont très bien déroulées, les utilisateurs se sont prêter au jeu et nous avons pu en tirer les fonctionnalités qu’ils désiraient avoir sur leur application. Ceci est un point positif concernant notre démarche mais il nous a tout de même couté beaucoup de temps au début de notre projet. De plus nous avions décidé d’un certain nombre des fonctionnalités qui seraient primordiales et celles qui seraient secondaires et qui pourraient être implémentées ultérieurement. Cependant nous en avons choisi trop et nous n’avons pas réussi à tout implémenter. Nous pensons en parti que ceci est dû au choix d’une technologie que nous ne connaissions pas du tout avant de commencer le projet.

Ce qui s’est très bien passé dans le projet est la communication dans l’équipe et la répartition des tâches. Grâce au temps que nous avons passé à analyser les besoins des utilisateurs et à la conception qui s’est faites en équipe nous avons beaucoup réfléchit aux différentes parties du projet (back-end et front-end). Ce temps que nous avons passé à réfléchir tous les trois nous a permis de cerner le projet dans sa globalité et donc au moment de la répartition des tâches chacun de nous connaissait bien le contexte et ce que les autres membres de l’équipe attendaient de notre part. La délimitation des tâches de chacun était donc bien claire.

## Annexe 1. Compte rendu du premier coaching.

Nous avons eu quelques remarques de la part du coach, essentiellement sur des points purement syntaxiques. Notre démarche centrée utilisateur n’a pas été totalement comprise à la suite de la lecture du DoW. En effet, nous avons identifié lors de la conversation, quelques réflexions qui nous ont laissé penser que le point de vue de l’utilisateur final n’est pas forcément pris en compte dans le suivi des projets étudiés au quotidien en entreprise. Nous avons dû redéfinir à l’oral nos deux itérations et les expliquer. Aucune remarque négative n’a été faite quant à la gestion du budget dans le projet. Toutefois, notre coach en a profité pour nous présenter quelques méthodologies/outils utiles dans le suivi d’un projet : *profiling*, *KPI (Key Performance Indicator)*, gestionnaire de versions, *sprints*, *pitch elevator, Test Driven Development*.

Tout d’abord, nous avons identifié que la notion de profiling mise en avant par le coach s’apparente à celle des personas dans la démarche centrée utilisateur. Les personas vont représenter un type particulier de personne, auquel nous associons un nom. Ainsi, lors de la description des scénarios, nous retrouvons ces types de personnes en écrivant « **AS** Persona\_Name ». Cette formule va définir un acteur dudit scénario.

Dans un second temps, concernant les KPI, ceux-ci permettent de chiffrer, à tout moment du projet, la consommation du budget et l’effort restant à faire. Une valeur leur est attribuée en début de projet dans le but d’avoir une estimation de la répartition du budget. Par exemple, pour une application devant être développée sur smartphone, ordinateur et tablette, que doit-on privilégier ? Quel est l’ordre d’importance ? A 50% de la phase de développement dans le projet, nous devons être en mesure de chiffrer l’avancement en fonction de l’importance du support et ainsi déterminer un retard éventuel.

Notre coach nous a conseillé d’utiliser un gestionnaire de versions (ex. Git) dans le but de conserver une trace des documents en cas de besoin. Les différentes versions du code source doivent aussi être enregistrées. Nous ne savions pas qu’il fallait placer cette information dans le DoW mais étions déjà au courant.

Avant l’entretien, nous n’avions jamais entendu parler de la notion de sprint. Celle-ci peut être définie comme étant composée de plusieurs phases qui sont : la planification, la mise en œuvre et démonstration. Un second sprint doit prendre en compte les résultats obtenus lors du premier sprint. Dans notre DoW nous avons identifié trois sprints :

* Analyse Utilisateurs (Planification de l’analyse, Mise en œuvre des entretiens, Retours sur les entretiens + maquettage)
* Première itération (Conception, Implémentation, Test Utilisateurs)
* Deuxième itération (Conception, Implémentation, Test Utilisateurs)

Nous avions identifié ces phases bien avant l’entretien, mais la formalisation en tant que « sprint » nous était inconnue.

Le *pitch elevator* définit un concept pour convaincre une personne en deux minutes sur une idée, un projet... Celui-ci dépend grandement de l’interlocuteur ; l’adaptation du discours en fonction de la personne en face est nécessaire.

Pour finir, le « Test Driven Development » est une méthode de travail qui met en avant une écriture des tests avant de développer le code de l’application.

De manière générale, notre coach a accentué l’importance qu’il faut apporter aux tests durant les phases d’implémentation. Nous répondons à ce besoin, dans notre DoW, grâce à une tâche dédiée pour les tests fonctionnels et un lot consacré uniquement aux tests et retours des utilisateurs. Entre autres, notre analyse de la gestion des risques a été appréciée par le coach (cause, impact, évitement, conséquences, résolution, probabilité, description).

## Annexe 2. Compte rendu du second coaching.

Durant ce second entretien le coach nous a fait remarquer 2 points importants sur notre projet.

Le premier est qu’il manque une automatisation des tests. A l’heure actuelle nous ne pouvons pas tester de manière automatique le projet. Lors de l’ajout d’une fonctionnalité nous ne savons donc pas si cela a un impact sur le code existant et précédemment testé.

Le deuxième point important était qu’il fallait que nous sortions une version alpha fonctionnelle et sécurisée le plus rapidement possible même s’il manquait des fonctionnalités. En effet au moment de ce rendez-vous nous avions une version assez avancée de notre projet mais elle n’était pas fonctionnelle de bout en bout.