|  |  |
| --- | --- |
|  | PREVI : GESTION de maintenance préventive |
|  |  |
| 2017 - 2018 | Raphaël BLEU, Florent BERNET, Thibault DELAVOUX |

Tuteur universitaire : Guillaume PAQUETTE

Licence 3 professionnelle informatique – Université de Franche Comté



Remerciements

L’application développée dans le cadre de ce projet est le résultat du travail acharné de notre équipe, mais également de toute l’aide fournie par plusieurs acteurs extérieurs.

Nous tenons à remercier chaleureusement :

**Mr. Guillaume PAQUETTE**, encadrant pédagogique de ce projet, pour avoir accepté de tuteuriser un projet présenté par nos soins. Merci pour son implication et sa disponibilité tout au long du projet. Nos réunions régulières ont permis à l’équipe de garder au mieux le cap tout au long du développement. Son regard extérieur nous a permis de recarder l’application dans ses objectifs principaux et permettre d’en faire un outil le plus générique possible. Son exigence aura été pour nous une opportunité de nous dépasser et d’offrir un maximum de fonctionnalités à l’application.

**Mr. Raphaël KRUCIEN**, responsable Méthodes et Achats au pôle Maintenance à KEOLIS Besançon Mobilité, pour son regard professionnel nous ayant permis de cibler au mieux les solutions métiers à développer au cœur de l’application et de mettre l’accent sur les fonctionnalités principales.

**Mr. François BONNEVILLE**, enseignant à l’université de Franche Comté, pour son savoir nous ayant permis de cibler au mieux les technologies adaptées à nos besoins, et à leurs applications dans la mise en place de ce projet.

Table des matières

[Titre 2|deux 3](#_Toc533420261)

[Titre 3|trois 3](#_Toc533420262)

# Glossaire

L’application développée dans le cadre de ce projet est une solution métier, destinée au milieu très spécifique qu’est la maintenance. Ce rapport reprend donc plusieurs termes spécifiques que nous allons tâcher de définir ici. Certains de ces termes ne sont pas systématiques à toutes les organisations de maintenance et sont résultant d’habitudes d’une organisation précise. Nous utiliserons dans ce document, les termes utilisés par l’application à son origine, mais ceux-ci peuvent êtres redéfinis par les utilisateurs finaux dans leur pratique.

**Maintenance Préventive**

Ensemble d’actions à réaliser sur un système, visant à prévenir les défauts ou dysfonctionnement. La Maintenance préventive permet, par la réfraction, la révision ou le remplacement de limiter significativement la quantité et la gravité des problèmes rencontré par le système.

**Passation**

Une passation est une instance d’un ensemble d’opérations préventives appliquées à un équipement. Une passation s’établie sur une durée donnée et a donc un début et une fin. Une passation est donc l’application d’une visite préventive à un outil, à un moment donné.

**Visite**

Une Visite est un ensemble abstrait de fiches préventives, réunies ensembles pour êtres appliquées sur un ou plusieurs modèles.

**Fiche**

Une fiche est un ensemble d’opérations préventives à appliquer à un outil. Une fiche est rattachée à une ou plusieurs visites, ce qui déterminera les outils sur lesquels elle pourra être appliquée. Une fiche dans l’application Previ peut être considéré comme l’équivalent d’une feuille papier, relatant les opérations à effectuer par un acteur.

**Outil**

Un Outil est un objet réel sur lequel des opérations de maintenance préventive vont être effectuées. Très concrètement, un outil peut être un bus, une fraiseuse, un serveur informatique … etc.

**Modèle**

Un modèle est une entité abstraite qui permet de catégoriser les outils selon leurs caractéristiques communes. Chaque outil se verra attribuer des visites préventives définies en fonction du modèle auquel il appartient.

Chaque modèle est rattaché à une organisation, mais dans certains cas, un outil peut appartenir à plusieurs organisations. Pour différencier les appartenances, un modèle peut être défini plusieurs fois dans plusieurs organisations différentes.

**Organisation**

Une organisation renvoie à une structure distincte, à un Pôle, à un service ou a une personne de l’organisme utilisant l’application. Définir plusieurs niveaux d’organisation permet de séparer symboliquement les différents outils en fonction de leur appartenant. Selon les cas d’utilisation, il est possible qu’une seule organisation ai besoin d’être définie.

**GMAO**

Logiciel de Gestion de Maintenance Assisté par Ordinateur. Ensemble d’outils permettant de gérer et d’historiser informatiquement les actions réalisées par l’équipe de maintenance.

# Présentation du contexte

## La Maintenance Préventive

La maintenance préventive est mise en place dans toutes les structures professionnelles utilisant du matériel mécanique, électrique, électronique … etc. Elle a pour but de favoriser la longévité des équipements en limitant les pannes, leur durées et leur gravité. Le taux de maintenance préventive à injecter dans un système est un calcul prenant en compte le coût de celle-ci par rapport aux conséquences des pannes pouvant survenir.

La maintenance préventive s’oppose à la maintenance curative qui, elle, s’applique à la réparation ou la réflexion suite à une avarie. La maintenance corrective est donc l’application d’action sur un système défectueux pour le ramener à un état de fonctionnement dit « normal ».

La maintenance préventive est omniprésente dans l’industrie en représentant bien plus de la majorité du travail réalisé. Cette importance s’accroit avec la taille du service de maintenance, la taille du parc d’équipements à gérer et la criticité du bon fonctionnement de ces équipements. De plus, certaines normes réglementaires peuvent alourdir les opérations de maintenance préventives – par exemple les contrôles techniques obligatoires sur le matériel roulant. Dans l’industrie, notamment ferroviaire, il est admis régulièrement qu’un rapport efficace est de 80% de préventif pour 20% de curatif.

## Mise en place du projet

Si la maintenance préventive n’est pas forcément réglementaire, cela implique que les opérations ne sont pas toujours effectuées. Il devient donc difficile pour les organisations de maintenir une base documentaire saine quant aux actions réalisées sur les équipements et de favoriser une recherche efficace, quand celle-ci est gérée en version papier. C’est le cas de beaucoup d’entreprises, d’associations ou de particuliers n’ayant pas les moyens d’investir dans des logiciels de GMAO. De plus toute les GMAO n’incluent pas nécessairement d’outils de mobilité, ni de gestion de la maintenance préventive.

C’est sur cette base, que nous avons décidé de développer une application libre de droit permettant à toute organisation de gérer et d’historiser facilement leur maintenance préventive. Le projet Previ était né.

## Cahier des Charges

Etant à l’origine du projet, il a fallu dans un premier temps réfléchir au contenu de l’application afin de proposer un contenu utile et au plus près des attentes des professionnels qui seraient à même de l’utiliser. Nous avons donc sondé des chefs d’équipes et responsables de service de maintenance afin d’éclaircir au mieux les besoins.

L’application devait se décomposer en deux parties. Une partie d’administration permettant aux responsables de définir leurs actions préventives et de les appliquer sur des modèles, et une partie « Mobilité » pouvant être utilisée sur un équipement adapté pour permettre aux agents de réaliser simplement leurs passations et de les enregistrer. Les deux parties devaient partager une base de données commune et fonctionner ensembles sans pour autant être concurrentes. Ci-dessous, l’extrait du cahier des charges défini en début de projet.

### Previ – Administration

La première application accessible via navigateur WEB permettra de définir des visites préventives. Une visite préventive se constitue de plusieurs questions qui défilent qui correspondent à des contrôles (Ex. Vérifier le niveau d’huile) ou des opérations (Ex. Changer le filtre à particule). Cette première application permettra de rendre le logiciel flexible et utilisable par n’importe quel service Maintenance. En résumé, l’application se chargera donc de créer des OPERATIONS et des CONTROLES et des les ordonner dans des FICHES. Ces FICHES pourront être regroupées dans des VISITES qui, elles, seront disponibles dans la seconde application pour les passations.

Fonctionnalités :

* Authentification par Login/MDP de l’acteur avec des profils gérant leurs droits d’accès. (3 niveaux de droits : Administrateur, Chef d’équipe et Technicien). L’administrateur aura les pleins droits sur l’application et bénéficier de toutes les fonctionnalités. Un compte administrateur par défaut sera pré-implémenté sur l’application afin de faciliter l’intégration de l’application dans une nouvelle organisation.
* Gestion des utilisateurs. Création, modification et Suppression de ceux-ci (Administrateur uniquement)
* Créer des VISITES, constituées d’une ou plusieurs FICHES, elle-même constituées d’OPERATIONS (Administrateur et chef-d ’équipe).
* Recherche et consultation des visites préventives déjà réalisées avec export PDF possible (Touts profils).
* Personnalisation de la terminologie.

### Previ –Passation

La seconde application contiendra une version numérique des gammes préventives définies en amont dans la première par un chef d’équipe. Les agents auront accès aux VISITES déjà créées et afficheront une série d’écran présentant les OPERATIONS et CONTROLES à réaliser pour chaque FICHE. Les VISITES réalisées seront maintenues en base de données permettant un accès et une traçabilité des OPERATIONS et CONTROLES faits ou non faits et l’équipement (Véhicule, machine, structure passant la visite préventive). Enfin, ces visites pourront être exportées en format numérique (PDF) permettant leur impression.

Fonctionnalités :

* Authentification par Login/MDP de l’acteur
* Accéder à une VISITE et la réaliser selon l’outil sélectionné
* Mémoriser le/les acteurs ayant réalisé une visite.
* Sauvegarder les VISITES en cours et Clôturer une visite terminée.
* Permettre la navigation dans une visite en cours et la modification.
* Gestion de la mise en cache des informations saisies en hors connexion et synchronisation dès le retour d’une connexion.

# organisation

## Spécifications techniques et technologiques

### Previ - administration

L’application d’administration à été réalisée en WEB afin de permettre à l’utilisateur un accès permanent, et cela quel que soit l’endroit où le l’appareil utilisé. L’application utilise les technologies PHP, HTML5, CSS3 ainsi que les Framework Bootsrap4 et JQuerry.

Navigateur Web supportés : Chrome V 70.0.3538.77 / Firefox 63.0.1 / Vivaldi V2.1

### Previ - passation

L’application de passation est une Progressive Web App (PWA). Ce format permet une installation sur tout supports mobile et tablette avec tous les avantages d’une application native.

### Base de Données

La base de données utilisée est une base MySQL. Un script de création de la base est fourni avec l’application. Celui-ci génère automatiquement les tables nécessaires à la version de l’application correspondante.

## Gestion du Projet

Afin de respecter au mieux le cahier des charges, et maintenir une cohésion et une fluidité dans l’avancée du projet, nous avons dû en amont définir nos méthodes de travail, et ce sur plusieurs plans.

### Méthode Agile

Il fut décidé en début de projet de fonctionner selon la méthode agile SCRUM, qui fût légèrement adaptée pour s’adapter à notre rythme d’étudiant. L’idée étant au départ de définir des objectifs principaux, et de découper le travail à faire en sprints. Chaque fin de sprint débouchait sur une réunion pédagogique afin de faire le point sur le travail réalisé et d’établir les objectifs à accomplir pour le prochain Sprint. En interne, des échanges hebdomadaires ont eu lieu afin de faire le point sur les difficultés en cours, vérifier de la bonne avancée des opérations, réajuster au besoin et valider les tâches terminées ensembles.

### Tableaux

Afin de gérer plus efficacement les tâches au sein de l’équipe, nous avons opté pour un outil d’organisation : Monday. Sur cet outil, nous avons pu définir des tableaux pour les deux applications à réaliser et deux tableaux pour les Bugs s’y rapportant. Chaque tableau contenait les différents sprints réalisés et les tâches à y effectuer. Afin que chacun puisse avancer à son rythme, et éviter les interblocages, il à été décidé de ne pas pré-attribuer à l’avance les tâches. Il revenait donc à chacun la responsabilité de s’auto attribuer une tâche au moment où il entamait celle-ci.

### Planification et organisation calendaire

Afin de maintenir la cohérence dans l’avancée du projet, il a été défini dès le départ un ordre de réalisation des tâches majeurs, et une durée pour chacune. Nous avons donc prévu de démarrer par l’application d’administration et se lancer sur l’application mobile une fois la première terminée.

### Mise en place d’une base test

Si la majorité du développement c’est fait en « local » sur nos machines, une base de données Test partagée a été mise en place afin de s’approcher au mieux des conditions réelles d’utilisation. La base a été hébergée à XXXXXXXX sur un serveur XXXXXXX.

# Realisation

Si l’application complète est trop conséquente pour en détailler ici chaque aspect, plusieurs fonctionnalités ont néanmoins nécessité la mise en place de techniques et de technologies spécifiques intéressantes que nous allons détailler ici.

## Dashboard

L’application Administration a un statut borderline. C’est officiellement un site web, car développé comme tel, mais son aspect fonctionnel est celui d’un logiciel classique (afin de coller au mieux aux attentes et aux habitudes dans l’industrie). Toute l’application a donc été modelée pour fonctionner dans une seule fenêtre et le modèle de Dashboard (tableau de Bord) était parfait pour cela.

L’application charge au démarrage une page générique qui est celle du Dashboard, avec en son centre la zone de travail. Afin d’optimiser les temps d’exécutions et limiter les informations sur la bande passante, cette page n’est chargée qu’une seule fois, et elle rafraichit uniquement sa zone de travail en y chargeant le contenu souhaité. Pour se faire, nous avons implémenté la technologie Fetch qui est un descendant d’Ajax, permettant la récupération de ressources dur le réseau de manière Asynchrone. Fetch est une technologie très récente (à peine plus d’un an d’existence) et est donc moins bien supporté qu’Ajax qui est bien plus Ancien. Ce choix nous a valu de devoir exclure la possibilité d’utiliser l’application sur Internet Explorer, mais cela a d’autres avantages. Fetch est une technologie **native** aux navigateurs internet récents. Cela implique qu’aucune bibliothèque ou Framework spécifique n’est nécessité car déjà implémenté directement dans le navigateur. De plus, cette technologie est plus souple et consomme moins de ressource qu’Ajax, et son implémentation est plus simple. Ci-dessous un exemple d’implémentation :

// Insérer Bout de Code D’exemple ici

Fetch nous permet donc de manière simple d’actualiser une partie de la fenêtre uniquement et de faire des modifications en base de données sans recharger la page en cours. Ce choix a donc été fait en s’inscrivant dans notre logique d’optimisation des ressources et du temps d’exécution favorisant l’expérience utilisateur.

## Préférences

Afin d’être le plus générique possible, l’utilisateur à la possibilité de définir des paramètres de personnalisations. Il est par exemple possible de redéfinir la terminologie utilisée pour être en cohérence avec celle utilisée par l’utilisateur. Les termes Visites, Fiches, Opérations, Modèles, Outils et Organisations pourront être modifiés par un utilisateur Administrateur, et ce pour toute l’application.

// Insérer Capture d’écran de la modification

## Modularité du code

L’application dans sa globalité est vouée à être Open Source. La volonté de départ étant que chacun puisse contribuer à son évolution, ou simplement l’adapter aux besoins spécifiques d’une organisation. Afin de faciliter la collaboration entre les contributeurs, le code à été pensé le plus modulaire possible. Pour ajouter une nouvelle page à l’application, une seule ligne de code est à ajouter dans le code de base au niveau du menu, et cela réside dans un lien vers la page à charger. Une fois cette ligne insérer, la page PHP voulue sera appelée dans la zone de travail, et toutes les interactions de cette page seront chargées dans la zone de travail.

Toutes les fonctionnalités déjà implémentées dans la version de base de l’applications sont disponibles dans une bibliothèque PHP. A la création d’une nouvelle page, il est donc possible, par exemple, d’implémenter simplement des contenus modaux.

// Insérer ici code d’appel des pages modales

## Fenêtres modales

Pour renforcer l’expérience utilisateur, nous avons implémenté des fenêtres modales pour les modifications, ajouts et suppressions. Cela permet à l’utilisateur de ne pas quitter la page en cours en la voyant en transparence, et cela renforce l’idée d’agir sur l’élément sélectionné. D’autre part, cette méthode d’implémentation permet via Javascript d’envoyer des informations d’une fenêtre à une autre sans requêtes serveur, et donc améliorer la consommation de ressources et la fluidité d’utilisation ressenti par l’utilisateur (exemple avec les fenêtres de sélection ajoutant dynamiquement des lignes aux tableaux des fenêtres de modification et d’ajout).

Sur version mobile, les fenêtres modales se masquent en pages distinctes en prenant toute la taille de l’écran afin d’améliorer la lisibilité pour l’utilisateur.

// Insérer ici image de fenêtre modale

## Historisation

Comme décrit dans l’introduction, la maintenance préventive prends une place conséquente dans la vie d’une organisation de maintenance. Pour une organisation conséquente, la quantité de visites préventives par mois peut s’avérer extrêmement élevé. Afin que le cœur de l’application ne se retrouve pas ralentie par l’accroissement de la Base de Donnée et donc améliorer les temps d’exécutions, les visites préventives terminées sont historisées et placées dans des tables distinctes de celles contenant les visites en cours.

## Responsive

L’application Administration est entièrement responsive Design et ce dans une logique d’accès à l’application quelque soit l’endroit et la plateforme utilisée. Un administrateur peut donc à tout moment, depuis chez soi sur sa tablette ou depuis la salle de réunion sur son téléphone portable, accéder à toute ses informations, et permettre de modifier à la volée des éléments.

// Insérer Image Responsive

Autres sujets à traiter :

# bilan sur l’application

## Version 1

// a rédiger à la fin du projet

## A venir

L’application Previ, bien qu’implémentant toutes les fonctionnalités minimums pour son utilisation, reste encore à l’état embryonnaire. Beaucoup de fonctionnalités sont à ajouter dans les versions suivantes, et nous allons tâcher d’en énumérer plusieurs ici.

### Version 2

La version 2 de Previ prévoir d’implémenter :

* Page spécifique Statistique, très détaillée pour permettre aux agents Méthodes de suivre au mieux leur maintenance préventive et anticiper le futur (coût, temps, … etc).
* Assistant à la construction de requêtes et générateur de rendu, afin de permettre aux utilisateurs lambdas (sans connaissances en informatique) d’obtenir des information très spécifiques et personnalisées.

### Logiciel Libre

Comme nous l’avons défini dès le début de ce document, l’application est distribuée sous licence Libre et Open Source, permettant à quiconque le modifier et de l’adapter à des besoins spécifiques. Si nous allons continuer de faire vivre l’application, nous avons comme objectif de faire intervenir d’autres collaborateur afin d’amener l’application à son objectif final, en faire un acteur important en GMAO Libre.

### Implémentation

Si l’application n’est aujourd’hui pas implémentée dans un service de maintenance, c’est l’objectif principal à terme de l’équipe. La mise en place d’outils de communication seront mis en place prochainement afin de permettre une diffusion de l’outil le plus simplement possible. Nous avons dans l’espoir de convaincre un réseau d’entrer en phase de test d’ici la livraison de la version 2.