**Collège de Rosemont**

**PROJET  
Le Cahier des Charges**

**Par  
  
gr. 445**

**Travail remis à M. Sylvain Labranche dans le cadre du cours  
DÉVELOPPEMENT DES APPLICATIONS INFORMATIQUES - PROJET  
420-B34-RO**

**Pour le sprint 2, à remettre le 23 décembre**

# Présentation du projet

## Demande du client

Pour ce sprint, Le client souhaite une base de donnés pour mieux gérer les feuilles de temps générés par le logiciel.

Le client veut aussi des comptes utilisateurs aussi stockés dans la base de données et que seulement un employé connecté puisse sauvegarder sa feuille de temps dans la base de données.

## Explication du projet

Pour notre projet, on choisit de faire une application web contenant les interfaces graphiques demandés, d’abord on a le menu qui permet de choisir si l’utilisateurs est un patron ou employé, dans le cas du patron l’application mène à une fenêtre de connexion ou il devra entrer son mot de passe et son nom d’utilisateur.

Après le login du Patron, on ouvre l’interface graphique qui permet de charger les feuilles de temps (Fichiers.JSON) et de les valider, après les avoir validés, Les feuilles de temps seront sauvegardés dans la base de donné. Si on a choisi la partie employée dans le menu principal, l’interface qui permet à l’employé d’entrer ses heures de travail s’affiche.

Dans l’interface employée, on a des champs de saisie pour entrer les informations tel que son numéro d’employé, l’année, le numéro de la semaine. Une espace pour entrer le code de projet et le temps accordé à ce projet pour chaque jour de la semaine ainsi que la possibilité de cocher la case d’une journée de maladie s’il y a lieu, ceci empêche la saisie du code de projet et le temps de cette journée et le mets à 999 et 420 respectivement.

Désormais, l’application compte avec un bouton d’auto-remplissage de feuille de temps, ce bouton copie les informations du Lundi et les mets dans toutes les journées de la semaine (Lundi-Vendredi)

L’application peut gérer maintenant le code de projets 996 et 997 pour un maximum d’une heure chaque pour les employés qui qualifient pour réclamer les heures de transport payés.

Le code projet 996 est pour les employés qui utilise leur voiture pour se rendre au travail et le code 997 et pour les employés qui se déplacent en transport commun.

Le code 995 est maintenant réservé pour les employées qui accouchent, l’employée en question doit marquer à tous les jours de cette semaine code 995, sauf le weekend, pour réclamer ses heures payées de maternité.

## Explication des fonctionnalités souhaitées

* La gestion d’auto-remplissage de la feuille de temps;
  + Lorsque l’employé utilise l’interface pour entrer son temps de travail, il peut remplir seulement le lundi et appuyer le bouton pour mettre ces informations dans le reste de la semaine (Mardi à Vendredi).
* La gestion de Heures de déplacement payés;
  + Pour les employés qui habitent loin du travail et font ses déplacements en voiture, ils ont le droit à 1 h de transports payé en écrivant le code projet à 996
  + Pour les employés qui habitent loin du travail et font ses déplacements en transport commun, ils ont le droit à 1 h de transports payé en écrivant le code projet à 997
  + Le système empêchera la saisie de plus de 60 minutes pour ces codes de projet.
* La gestion de journées de maternité;
  + Les employées qui accouchent peuvent réclamer une semaine de maternité en entrant le code projet 995 à chaque jour de la semaine
  + Le système empêchera la saisie de ce code les weekends

Répartition des tâches à chacun des membres de l’équipe

## Product owner

Sebastian est la Product Owner. Il a conçu avec le Scrum Master le document Technique et Le cahier de charge, il a fait les diagrammes de l’application tel que le Diagram UML.

## Scrum master

Anthony est la Scrum master. C’est lui qui a établi les user stories , il a aussi divisé les tâches pour le projet, il organise et prends en note la date des mêlées Scrum ainsi que les points importants abordés dans ces rencontres. Il a géré le back log tout au long du sprint 3 et a collaboré avec le reste de l’équipe pour le développement du projet Il a implémenté avec l’aide de Martin Vézina la base de donné avec l’entity Framework

## Équipe de développement

Pour le sprint 3, l’équipe de développement est composée par Guillaume Tremblay, Martin Forget et Martin Vézina, ils ont tous implémenté les nouvelles fonctionnalités que le client nous a présenté. Ils se sont occupés d’implémenter les 3 nouvelles fonctionnalités demandés.

Plus précisément, Martin Vézina a eu l’initiative de s’occuper de la base de donné originalement on le plan était de faire une base de donné NoSQL avec MongoDB, vu que cela aurait été simple d’intégrer les fichier JSON que l’application généré. Après il a trouvé des solutions avec Le scrum master pour implémenter la base de données avec L’entity Framework et Microsoft SQL manager.

Martin Forget et Guillaume Tremblay ont implémenté les fonctionnalités imposées par le client, soit Le bouton d’auto-remplissage la gestion des heures payés de transport et la gestion des jours de maternité payés.

Langages et logiciels utilisés

On s’est tous mis d’accord d’utiliser le langage C# pour tous les sprints à date de ce projet car c’est le langage dont on est le plus à l’aise. On utilise I’IDE Visual Studio et on se sert du Framework .net plus précisément ASP.NET pour faire une application web avec des interfaces graphiques. Pour implémenter la base de données on a utilisé l’entity Framework et Microsoft SQL Server. Étant donné que l’intégration est très simple au projet .NET

Pareil comme dans le sprint précédant, on continue à utiliser Jira pour la gestion du projet et pour entreposer notre code on utilise GitHub

* C#
  + C’est un langage avec lequel nous sommes familiers
* ASP.NET
  + C’est un Framework qui Inclus HTML, CSS et Javascript et facilite la conception d’une application web
* GitHub et Git
  + C’est le logiciel qu’on utilise pour la gestion de versions et pour collaborer tous dans le même projet.
* Jira
  + Utile pour la gestion du projet.
* SQL
  + On s’en sert lors de la création de tables pour la base de donnés du projet, à travers Microsoft SQL server manager

Échéancier du sprint

On fait nos mêlés Scrum en moyenne de 3 fois par semaine, 2 fois dans le temps alloué au cours du projet soit les mercredis et les vendredis, et une autre fois souvent les Lundis, pour discuter sur notre progrès faite durant la fin de semaine, quand on a besoin de rencontres additionnels le Scrum master organise une rencontre quand l’équipe au complet pourra être présent.

Chacun d’entre nous travail sur le projet quand on peut vu qu’on a un horaire d’étude très chargé et condensé, on a une bonne communication. Quand quelqu’un fini une tâche en particulier on se tient au courant à travers notre groupe de Discord, pour que le reste du groupe puisse analyser le code ou contribuer ou ajouter une modification avant de considérer la tâche comme faite.

Diagramme de fenêtres de l’application

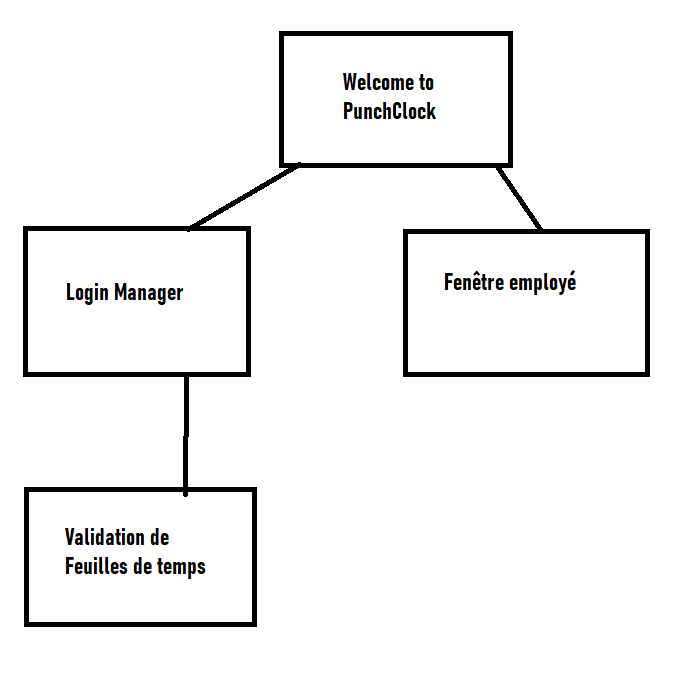
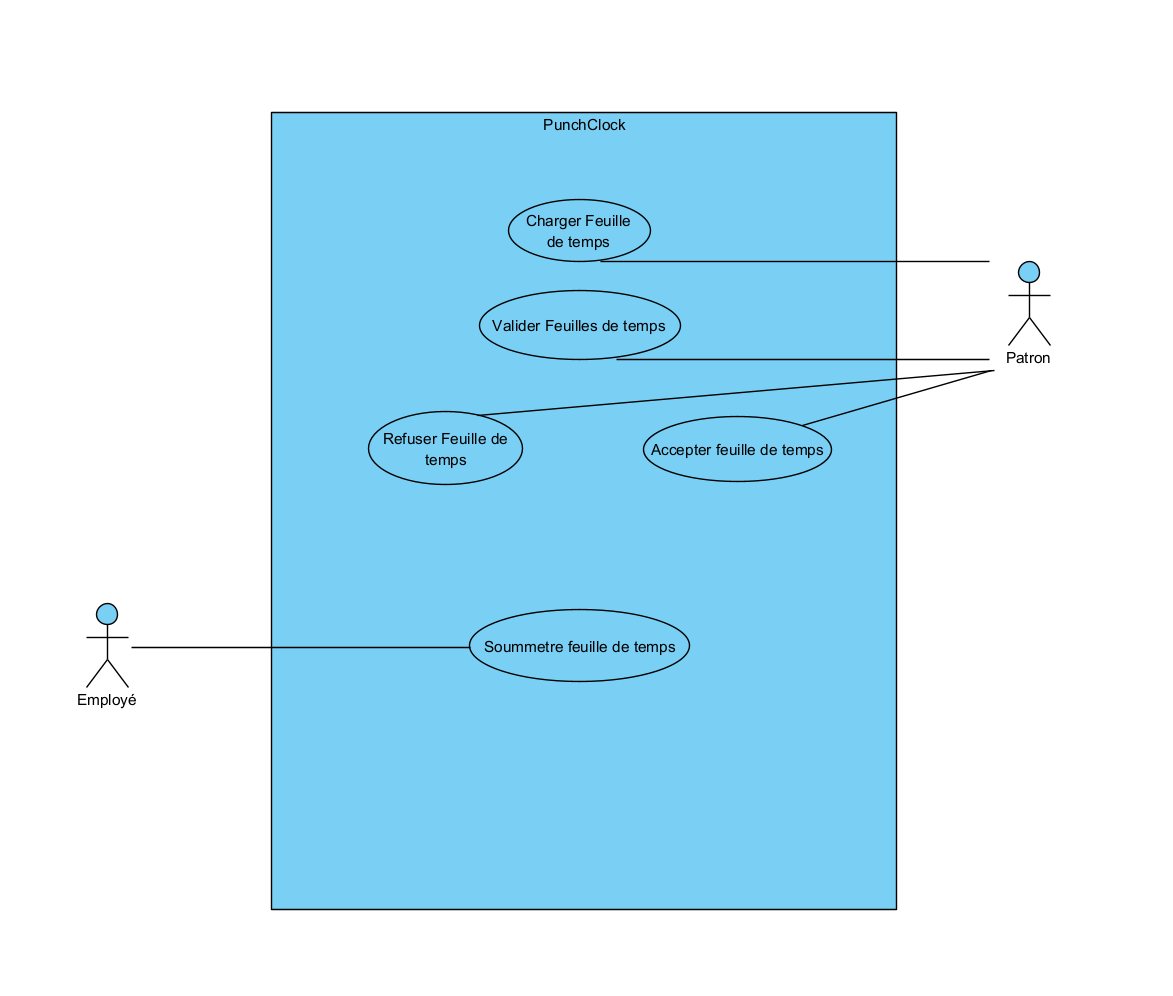


Diagram de cas d’utilisation et description narrative



**Nom :** Charger la feuille de temps(1).

**Acteurs :** Patron

**Objectif :** Charger une feuille de temps dans l’application provenant des documents sauvegardés dans l’ordinateur de l’utilisateur

**Règle d’initialisation :** Le patron ouvre son profil de ‘*Manager*’ dans l’application

**Scénario nominal :**

1. Le patron appui sur le bouton de chargement des feuilles de temps
2. Le patron choisi la feuille de temps qu’il souhaite faire valider

**Règle de terminaison :**La feuille est chargé dans l’application et est prête à être validée.

**Scénarios d’exception :**

1. Le système n’est pas fonctionnelle pour une raison donné Il n’ya pas de feuilles de temps sauvegardés dans l’ordinateur du patron
2. Interruption soudaine d’internet
3. Le patron ferme l’application abruptement pour une raison donnée

**Nom :** Valider feuille de temps (2).

**Acteurs :** Patron

**Objectif :** Valider la feuille de temps

**Règle d’initialisation :** Une fois la feuille de temps chargée, le patron appui sur le bouton valider

**Scénario nominal :**

1. L’application affiche toutes les informations de la feuilles de temps de l’employé
2. Le patron la lit et valide s’il respect les conditions du travail

**Règle de terminaison :** Le patron décide s’il accepte ou refuse la feuille de temps

**Scénarios d’exception :**

1. Le système n’est pas fonctionnel pour une raison donné

2. La feuille de temps ne respecte pas le format établi.

3. Le patron ferme l’application abruptement pour une raison donnée.

4. Interruption soudaine d’internet

**Nom :** Accepter la feuille de temps(3).

**Acteurs :** Patron

**Objectif :** Accepter la feuille de temps d’un employé

**Règle d’initialisation :** Le patron a pris une décision concernant la feuille de temps chargé préalablement.

**Scénario nominal :**

1. Le patron appui sur le bouton d’accepter la feuille de temps
2. Le système enregistre la feuille de temps dans la base de donné, dans la table de feuille de temps aprouvés

**Règle de terminaison :** Le patron a sauvegardé la feuille de temps avec succés

**Scénarios d’exception :**

1. Le système n’est pas fonctionnel pour une raison donné

2. Le dossier FeuillesTemps n’existe pas encore sur le disque c.

3. Le patron ferme l’application abruptement pour une raison donnée.

4. Interruption soudaine d’internet

**Nom :** Refuser la feuille de temps(4).

**Acteurs :** Patron

**Objectif :** Refuser la feuille de temps d’un employé

**Règle d’initialisation :** Le patron a pris une décision concernant la feuille de temps chargé préalablement.

**Scénario nominal :**

1. Le patron appui sur le bouton refuser feuille de temps
2. S’il n’existe pas encore l’application crée un nouveau dossier à l’intérieur du dossier FeuillesTemps appelée FeuillesRefuser
3. L’application sauvegarde la feuille de temps chargé dans ce dossier.

**Règle de terminaison :** Le patron a sauvegardé la feuille de temps avec succés

**Scénarios d’exception :**

1. Le système n’est pas fonctionnel pour une raison donnée.

2. Le dossier FeuillesTemps n’existe pas encore sur le disque c.

3. Le patron ferme l’application abruptement pour une raison donnée.

4. Interruption soudaine d’internet.

**Nom :** Soumettre une feuille de temps(5).

**Acteurs :** Employé

**Objectif :** Soumettre une feuille de temps de travail pour une semaine de l’année

**Règle d’initialisation :** L’employé a accédé à la fenêtre de remplissage de feuille de temps de l’application

**Scénario nominal :**

1. L’employé choisi premièrement le numéro de la semaine dont il souhaite remplir la feuille de temps.
2. L’application rempli automatiquement les champs des dates exactes des jours de la semaine que l’employé a choisi
3. S’il y a un jour férié dans cette semaine, le système rempli automatiquement le temps de cette journée à 420 minutes et le projet à 998.
4. L’employé entre son numéro d’employé.
5. L’employé entre ses minutes par projet, dans la section de chaque jour de la semaine et il les ajoute au système
6. S’il y a lieu l’employé peut cocher la case de journée de maladie et le système rempli le projet à 999 et les minutes à 420.
7. Une fois toutes les informations remplis, l’employé appui sur le bouton Subit, pour soumettre la feuille de temps
8. Le système va déterminer s’il l’employé a respecté toutes les normes concernant les politiques des heures de travail de l’entreprise
9. Si la feuille de temps n’est pas conforme l’application averti l’employé
10. Une fois toutes les champs requis soient conformes et remplis, l’application sauvegarde la feuille de temps dans la table de feuilles de temps à approuver par un manager

**Règle de terminaison :** La feuille de temps a été créé et sauvegardé avec succès.

**Scénarios d’exception :**

1. Le système n’est pas fonctionnel pour une raison donnée

3. L’employé ferme l’application abruptement pour une raison donnée.

4. Interruption soudaine d’internet