



[TIMBREUSE 2.0 – P1699]

Table des matières

1. Introduction	2
a. Définition du jargon	2
2. But du projet.....	3
a. Qu'est-ce qu'une timbreuse ?	3
b. Cahier des charges.....	3
i. Une redondance des informations.....	3
ii. Respect du règlement	4
iii. Interface et fonctionnalités.....	4
c. Acteurs du projet.....	5
i. Les élèves.....	5
ii. Les professeurs.....	5
iii. L'administrateur.....	6
3. Les outils.....	6
a. Choix des outils	6
b. Node JS.....	7
c. Electron	7
4. Fonctionnement.....	8
a. Le timbrage.....	8
iii. Rappel des règles	9
b. Le logiciel bureau.....	10
c. Fonctions importantes.....	10
i. Timbrage.....	10
ii. Fonction de fin de journée	11
iii. Service sans interruptions	12
5. Matériel	13
a. Le montage.....	13
6. Conclusion	14
a. Améliorations possibles	14
7. Bibliographie	14
8. Annexe.....	15
a. Les études.....	15
i. Node JS	15
ii. Electron.....	15
iii. Compatibilité avec les Raspberry Pi	15

iv.	Choix du port TCP	15
v.	Choix du moteur de base de données	15
vi.	Format de dates.....	15
b.	Les tests	15
i.	Export des CSV.....	15
ii.	Réplication des CSV sur les timbreuses.....	15
iii.	Timbrage hors connexion	15
iv.	Algorithme de formatage des secondes.....	15
v.	Procédure de test graphique	15
c.	Les tutoriels	15
i.	Professeurs.....	15
ii.	Administrateurs	15
iii.	Élèves	15

1. Introduction

Ce projet a été commencé durant l'année scolaire 2014-2015 par deux étudiants des classes du Module Complémentaire Technique (MCT), dans le cadre d'un grand projet. Le projet a initialement été créé en Java sur le [framework](#) Tomcat. Il était possible d'interagir avec les différentes fonctionnalités du logiciel depuis une interface web accessible depuis le réseau intranet du CPNV.

De décembre 2016 jusqu'en janvier 2017, M. Mark LOCATELLI m'a demandé de continuer le projet afin de corriger des problèmes qui étaient apparus après avoir poussé le système précédemment créé dans ses limites et de rajouter de nouvelles fonctionnalités. À l'aboutissement du projet, à cause notamment d'un manque de tests en conditions réelles et d'une trop grande confiance dans les tests réalisés au sein d'un environnement virtuel, de nouvelles erreurs sont apparues. L'erreur la plus critique concernait la fonction qui devait s'exécuter tous les soirs. Celle-ci manquait à s'exécuter, créant ainsi une instabilité générale de tout le système.

a. Définition du jargon

On appelle l'action de timbrer celle de faire lire son badge ou sa carte RFID (Radio Frequencies Identification), à une timbreuse esclave afin que celle-ci signale mon arrivée ou mon départ au système. On a deux types de timbreuses : les esclaves sur lesquelles les élèves peuvent timbrer et le serveur central où toutes les informations convergent.

Chaque élève se voit muni d'un badge ou d'une carte contenant un identifiant unique. On désignera cette identifiant comme « tag ». Les lecteurs de cartes dont sont équipées les timbreuses sont capables de lire le tag d'un élève et de le transmettre directement en sériel au système parent.

2. But du projet

a. Qu'est-ce qu'une timbreuse ?

La timbreuse est un système informatique dont le but est de suivre les allées et venues des élèves pour pouvoir suivre leurs rythmes et quantité de travail. Le logiciel est principalement destiné aux enseignants afin qu'ils puissent surveiller leurs élèves.

b. Cahier des charges

i. Une redondance des informations

Les professeurs voulaient avoir une sécurité de l'information accrue par rapport à la première version de la timbreuse. Dans le premier projet, toutes les informations étaient stockées dans une seule base de données sur le serveur central. Si celui-ci venait à tomber en panne, toutes les données auraient été définitivement perdues.

Donc, dans cette nouvelle édition de la timbreuse, la redondance des informations est un élément clé, comme expliqué dans le schéma suivant :

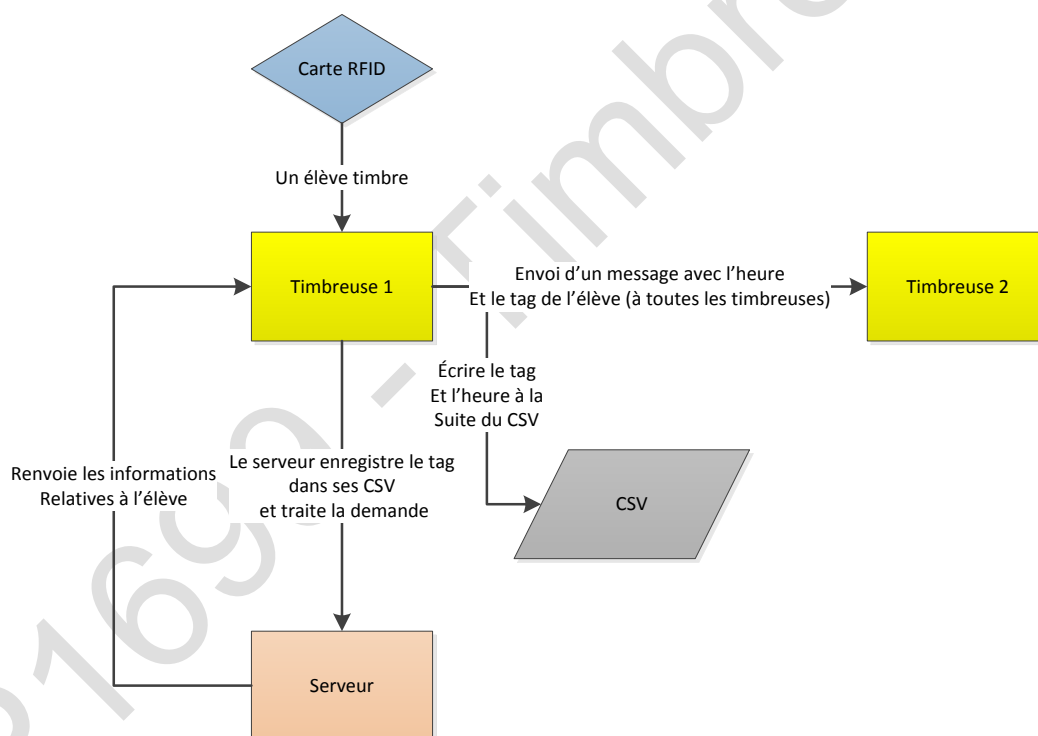


Schéma 1 Distribution des fichiers CSV entre les timbreuses

Lorsqu'un élève timbre sur l'une des timbreuses esclave à sa disposition, celle-ci va enregistrer dans un fichier CSV (Fichier contenant des informations sous forme d'un tableau où toutes les lignes sont séparées par des retours à la ligne, où les éléments d'une ligne sont séparés par des virgules) le tag de l'élève, ainsi que l'heure de son timbrage. Ces informations suffiraient à retracer l'activité d'un élève dans le cas où la base de données centrale viendrait à disparaître.

En plus d'enregistrer ce fichier CSV sur la timbreuse où l'élève a timbré, celle-ci l'enregistre aussi sur toutes les autres timbreuses connectées au réseau. Le serveur lui-

même, avant de traiter et d'enregistrer dans la base de données les informations, va enregistrer les données du tag dans des fichiers CSV.

Ensuite, ces fichiers CSV peuvent être extraits des timbreuses par les professeurs qui sont munis d'un tag avec les autorisations nécessaires. Ils branchent une clé USB sur la timbreuse et timbrent avec leur tag enseignant. Tous les CSV vont être exportés dans un dossier daté sur le support de stockage amovible. Seuls les administrateurs peuvent sortir les CSV stockés dans le serveur central. De plus, les CSV du serveur central sont accompagnés par une sauvegarde complète de la base de données au format CSV.

Enfin, les timbreuses sont munies d'un système de reconnexion automatique, qui va tenter de se reconnecter au serveur ainsi qu'aux autres timbreuses si celles-ci venaient à être déconnectées.

ii. Respect du règlement

À la différence de la version précédente, la nouvelle timbreuse tient compte des règles de temps fixées par le CPNV sur la quantité de travail. Le système vérifie la bonne application du règlement et applique des corrections automatiques aux réfractaires.

iii. Interface et fonctionnalités

Au profit d'une interface WEB, la timbreuse utilise maintenant un logiciel de bureau multiplateforme. En réalité chaque distribution du logiciel de la timbreuse contient tout le code nécessaire pour qu'elle puisse être exécutée en tant que serveur, esclave ou logiciel client. Cette sélection se fait par l'interface de commande du système d'exploitation utilisé, comme montré sur le schéma suivant :

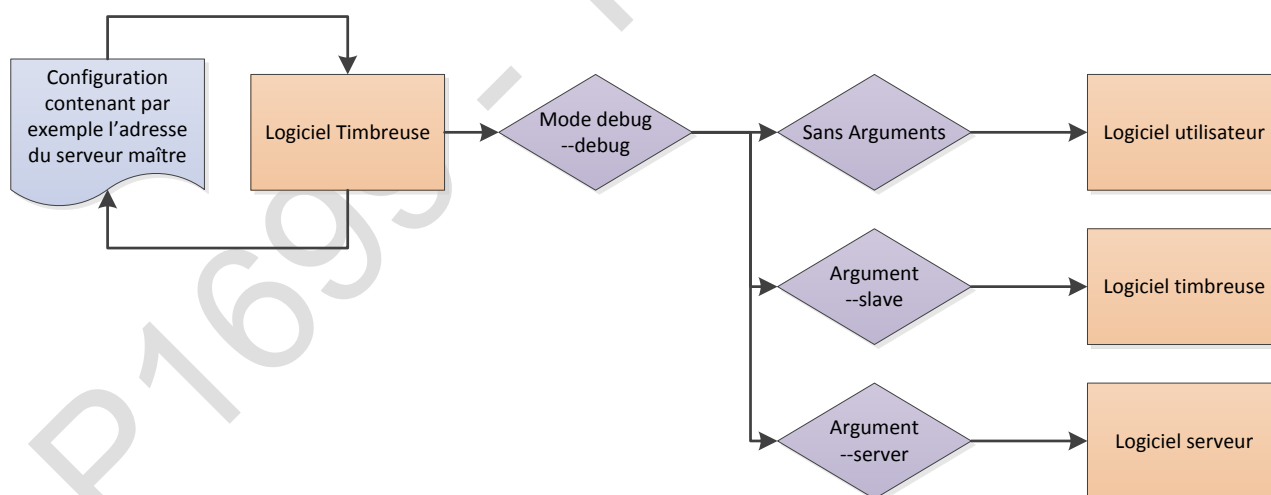


Schéma 2 Sélection du type de processus au lancement de la timbreuse

Tout d'abord, le logiciel de la timbreuse va charger le fichier de configuration situé à la racine de celui-ci. Il va ensuite analyser les paramètres qui ont été donnés au lancement du logiciel. Le mode « debug » est le seul à pouvoir être utilisé avec tous les modes en même temps. Il permet d'afficher plus d'informations par rapport au logiciel, mais aussi beaucoup d'informations inutiles, si ce n'est pour trouver un problème. Par défaut, le logiciel se lance en mode client. Le paramètre « slave » lance la timbreuse en mode esclave, respectivement le mode « server » permet de lancer la timbreuse comme un serveur.

Pour la liste des fonctionnalités complète, voir le [Cahier des Charges](#) en annexe.

c. Acteurs du projet

i. Les élèves

Les élèves sont les principaux acteurs du projet, étant donné qu'ils sont sensés, chaque matin et chaque soir, indiquer leurs allées et venues, grâce à leur tag. Ils peuvent ensuite consulter leur activité grâce au logiciel client de la timbreuse. Les élèves peuvent aussi consulter les dates de vacances qui ont été données par les professeurs. Finalement ils peuvent faire des demandes de congés directement depuis l'application. Celles-ci peuvent être acceptées, ou non, par leur professeur référent.

ii. Les professeurs

Si les élèves sont les acteurs principaux, les professeurs sont les plus importants. Sans les professeurs, la timbreuse n'aurait aucun intérêt, étant donné que son but fondamental est un but de surveillance de la quantité horaire de travail fournie par les élèves. Les professeurs peuvent modifier tout ce qui concerne les élèves. Voici la liste des informations modifiables sur le profil d'un élève par son professeur référent :

- Nom
- Prénom
- Nom d'utilisateur
- Date de naissance (Facultatif)
- Email (Facultatif)
- Tag
- Projet (Facultatif)

Les enseignants peuvent aussi modifier la différence de temps des élèves, signaler un étudiant comme malade, timbrer à la place de quelqu'un sans avoir besoin de sa carte, mettre en place des jours de congés et des vacances, mettre un élève en horaire bloqué et finalement créer des demandes de congés automatiques. Pour toutes ces options, le professeur peut laisser un commentaire sur son action qu'il pourra consulter plus tard pour se repérer dans ses propres actions.

1. Modification du temps

Les professeurs peuvent modifier le temps des élèves de trois façons différentes :

- « Set time » : Mettre le compteur d'un élève à une valeur précise.
- « Mod time » : Soustraire ou ajouter du temps au compteur d'un élève.
- « Reset » : Remet tout le compte d'un élève à zéro. Toutes les logs, demandes de congés et notification reliés à cet élève seront supprimés.

2. Horaire bloqué/fixe

Les professeurs peuvent choisir si un élève doit respecter l'horaire fixe ou l'horaire libre du MCT. Par défaut, un nouvel élève n'est pas en horaire bloqué. Un professeur peut choisir de mettre certains élèves en horaire bloqué, tandis que d'autres sont en horaire libre. Il est possible de modifier les horaires bloqués à partir du document de configuration *config.json*, situé à la racine du logiciel timbreuse du serveur. Pour de plus amples informations sur ce fichier de configuration, voir le document annexe sur la [configuration de la timbreuse](#).

3. Changer le statut d'un élève

Les professeurs peuvent timbrer à la place d'un élève en utilisant le logiciel client de la timbreuse. Ils n'ont pas besoins d'avoir accès au tag de l'élève. Ils ont aussi la possibilité de mettre un élève en mode absent. Le mode absent va faire en sorte que les élèves ne perdent pas le temps que la timbreuse retire aux autres élèves chaque soir. Le mode absent sera automatiquement désactivé la prochaine fois que l'élève timbrera.

4. Créer des congés

Les professeurs peuvent créer des congés soit sur une journée ou soit sur une plage de date. Durant la période de congés, les élèves peuvent venir timbrer, mais le temps obligatoire à faire normalement cette journée ne sera pas enlevé le soir.

5. Créer des demandes de congés

Les professeurs ont la possibilité de créer des demandes de congés pour un élève. Lors de l'exécution de la fonction de fin de journée (La fonction qui enlève le temps nécessaire à faire pour cette journée aux élèves) le serveur va calculer le temps raté par l'élève et faire la différence du temps à enlever par le temps d'absence justifiée. À noter que les élèves peuvent aussi faire des demandes de congés mais que, contrairement à celles des professeurs qui sont par défaut acceptées, celles des élèves doivent être acceptées par le professeur référant.

iii. L'administrateur

L'administrateur est une entité virtuelle qui peut contrôler les professeurs. L'administrateur peut créer de nouveaux professeurs (avec les mêmes options que lors de la création d'un élève, à la différence d'un champ supplémentaire pour un nom de classe), il peut modifier et/ou effacer des professeurs existants. Finalement, les administrateurs peuvent changer le professeur référant de chaque élève.

3. Les outils

a. Choix des outils

Le choix des outils a été sujet à des comparaisons avec les outils utilisés pour la version précédente de la timbreuse. Le détail de cette comparaison est visible dans le [document annexe](#) sur les choix et motivations dans le choix des outils. Les études sur le choix des framework et du matériel sont disponibles en annexe.

b. Node JS

Pour le logiciel de la timbreuse, l'utilisation de [Node JS](#) était intéressante car l'application utilise beaucoup de fonctionnalités réseaux qui sont nativement implémentées dans ce *framework*. Node JS est particulièrement utilisé dans les serveurs *HTTP*, ou sur des serveurs *TCP*, étant donné la capacité de ce *framework* à adapter l'utilisation des ressources aux nombre de connexions simultanées. De plus, Node JS est un *framework multiplateforme*, donc particulièrement adapté pour fonctionner à la fois sur les ordinateurs de bureau du MCT, mais aussi sur les [Raspberry Pi](#).

c. Electron

[Electron](#) est un *framework* additionnel à Node JS permettant de créer des interfaces graphiques à partir des technologies du WEB ([HTML](#), [Javascript](#) et [CSS](#)). Cet outil a été particulièrement utile pour créer le logiciel de bureau de la timbreuse, ainsi que l'interface des timbreuses esclave.

Ce framework crée deux processus parallèles lorsque l'application est créée. Le premier est le processus principal. Le processus principal va tourner en fond et ainsi gérer les opérations réseaux par exemple. Le deuxième processus va, lui, gérer la partie graphique de l'application. Les deux processus peuvent communiquer entre eux en utilisant le protocole de communication virtuel [IPC](#), comme montré dans le schéma suivant :

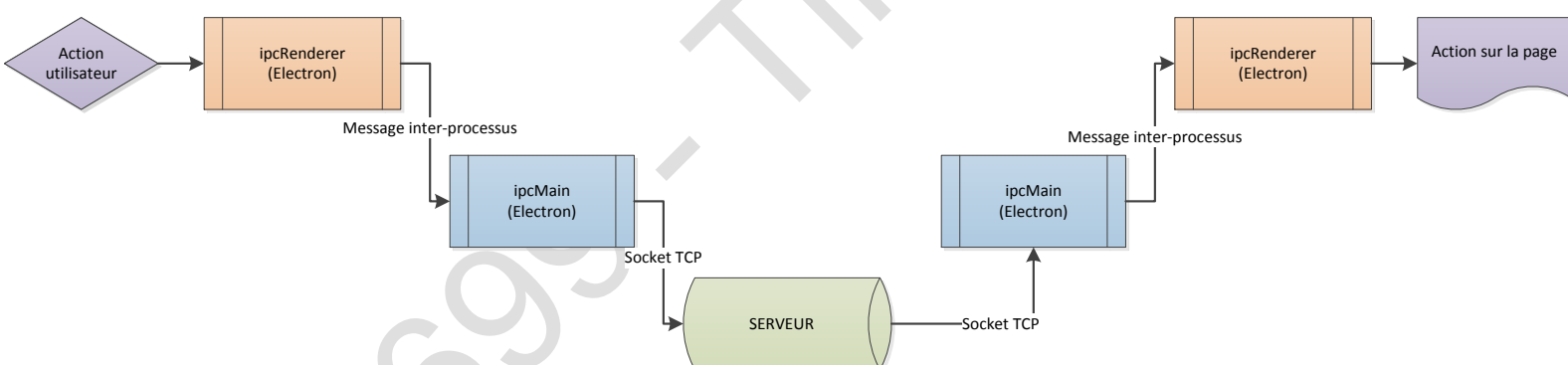


Schéma 3 Fonctionnement de la communication interprocessus dans le logiciel de la timbreuse

Dans le schéma ci-dessus, on peut voir le comportement du logiciel de la timbreuse lorsque que l'utilisateur fait une requête sur l'interface graphique de la timbreuse. La première étape est de formater et transférer les informations sur le processus principal d'*Electron*. Le processus principal va ensuite faire l'interface avec le serveur, via un socket [TCP](#) (à noter que si le socket est déconnecté, l'utilisateur est automatiquement déconnecté). Lorsque le serveur répondra à la requête, le processus principal va vérifier l'intégrité du message et va passer les paramètres de la réponse au processus principal, pour afficher le succès de l'opération, ou son échec.

4. Fonctionnement

a. Le timbrage

i. Déclenchement par l'utilisateur

Pour que l'élève timbre sur la timbreuse, le lecteur de carte RFID doit être connecté sur la timbreuse et le logiciel esclave doit être lancé. Lorsque l'élève passe son tag devant le lecteur, le système va lire son tag et effectuer toute la séquence décrite plus tôt dans la procédure de [redondance des informations](#).

ii. Résultat pour l'utilisateur

Après avoir timbré, l'élève voit apparaître un panneau donnant des informations par rapport à son compte, comme montré ci-dessous. Ces informations peuvent être effacées de l'écran si l'utilisateur touche avec son doigt l'écran de la timbreuse.

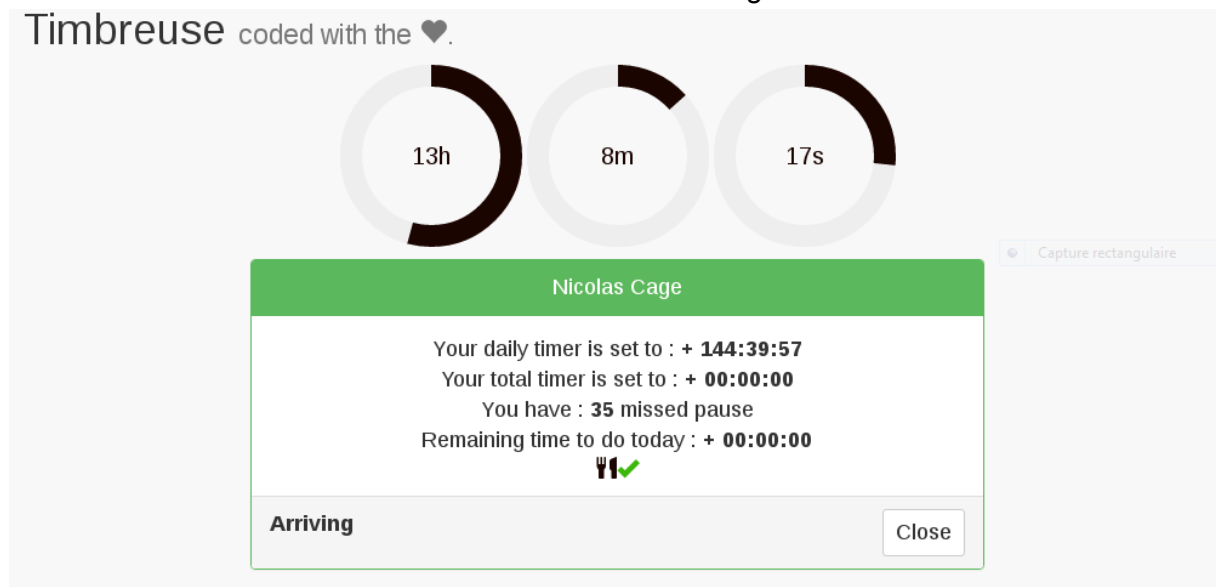


Figure 1 Affichage de la timbreuse esclave lorsqu'un élève a timbré

Dans la figure 1 ci-dessus, on voit plusieurs informations :

- L'heure actuelle, représentée par trois cercles centrés en haut de la page
- Le nom et prénom de l'élève en haut du panneau vert
- Le temps de l'élève pour aujourd'hui (Notez que les valeurs affichées sont fausses et n'ont pour but que d'afficher les différentes informations données par la timbreuse esclave)
- Le temps de l'élève total
- Le nombre de pauses ratées
- Le temps restant à faire aujourd'hui
- Si l'élève a mangé un symbole « Vue » vert est placé à côté de l'icône des services. Si l'élève n'a pas mangé, une croix rouge est placée à côté des services.
- Si l'élève arrive ou part (Dans le cas d'un élève qui arrive, le panneau est vert. Dans le cas d'un élève qui part le panneau est rouge)

iii. Rappel des règles

Le [règlement complet du MCT par rapport au temps de travail](#) est disponible en annexe. Les règles principales en vigueur au moment de l'écriture de ce rapport sont :

- Il est interdit :
 - D'arriver en retard
 - De partir avant l'heure obligatoire

- Si l'élève fait une pause de moins de 5 minutes, mais de plus de 20 secondes, la timbreuse lui enlèvera du temps automatiquement pour que sa pause dure 5 minutes.
- Si l'élève ne fait pas une pause de 20 minutes (block continu) toutes les 4 heures (block discontinu), la timbreuse lui enlèvera 20 minutes pour chaque pause ratée. A noter que les élèves peuvent rattraper leurs pauses ratées.
- Si un élève mange en moins de 30 minutes à midi, alors la timbreuse lui enlèvera 30 minutes en fin de journée.
- Si le dernier statut de l'élève à la fin de la journée, est « IN », alors l'élève se verra retirer tout le temps obligatoire de la journée et toutes les heures faites durant la journée seront ignorées.

Remarquez que toutes les infractions au règlement seront notifiées au professeur référant immédiatement et qu'il pourra consulter ces notifications dans le futur.

b. Le logiciel bureau

Des vidéo explicatives pour chacun des types d'utilisateurs (administrateur, professeurs, élèves) sont disponibles en [annexe](#).

c. Fonctions importantes

i. Timbrage

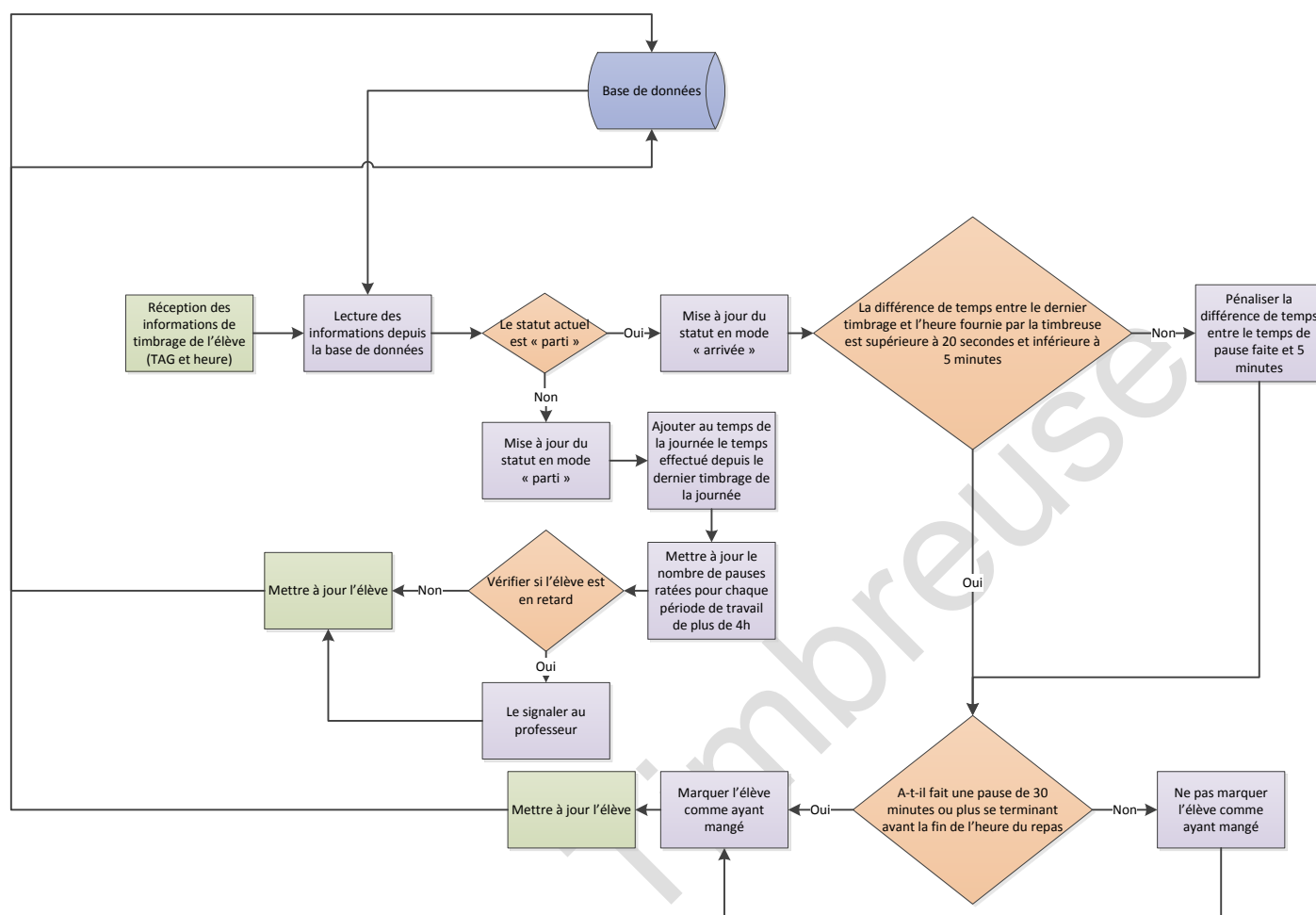


Schéma 4 Fonctionnement détaillé de la fonction de timbrage

Dans le schéma block ci-dessus, on peut voir comment fonctionne globalement la fonction de timbrage de la timbreuse. Le premier élément déclencheur est le tag d'un élève sur une timbreuse. Il va y avoir une lecture de la base de données de la part du serveur pour connaître toutes les informations relatives à l'élève. Le serveur va, par la même occasion, vérifié la bonne application des règles de timbrage et agir en conséquence. Après avoir mis à jour l'élève, les informations sont renvoyées à la timbreuse pour que les nouvelles informations soient affichées à l'écran. Il est à noter que l'action de timbrer ne vient pas forcément d'une timbreuse esclave, mais peut aussi venir d'un professeur avec un logiciel client.

ii. Fonction de fin de journée

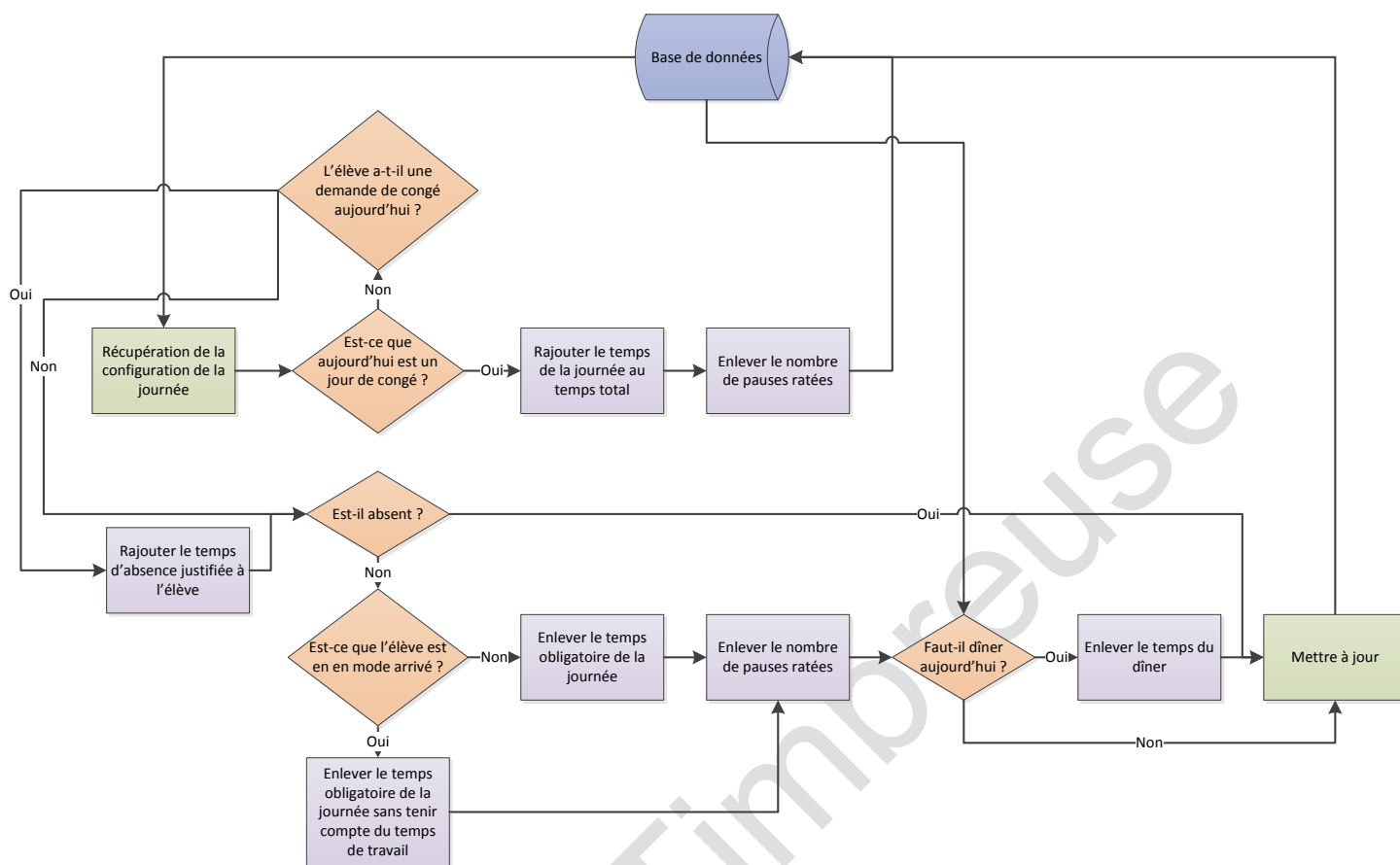


Schéma 5 Fonctionnement détaillé de la fonction fin de journée

Pour résumer le schéma ci-dessus :

- Si le jour d'aujourd'hui est un jour de congé, alors personne ne perd de temps
- Si l'élève traité est absent, alors il ne perd pas de temps
- Si l'élève traité a une demande de congé ce jour-là, alors le système lui rend un certain nombre d'heures par rapport à sa demande de congé.
- Si l'élève est en mode arrivée à la fin de la journée, il perd toutes ses heures faites ce jour-là, plus une pénalité d'un montant variable (la pénalité est égale au nombre d'heures à faire dans la journée)

iii. Service sans interruptions

Dans le cas où le serveur viendrait à s'éteindre, ou que la connexion viendrait à être déconnectée pour une raison ou pour une autre, les élèves devraient toujours être capables de timbrer. Pour ce faire, les timbreuses vont enregistrer dans une liste virtuelle tous les timbrages qu'elles n'ont pas réussi à distribuer. Une fois la connexion au serveur à nouveau active, toutes les commandes de timbrage seront envoyées au serveur. Cependant, il ne faut pas timbrer sur deux timbreuses différentes avec un même tag lorsque les timbreuses ne sont pas connectées. Le serveur va traiter les demandes séquentiellement par timbreuse, mais parallèlement entre les timbreuses. Des erreurs pourraient survenir lorsqu'un élève se met en mode départ avant de se mettre en mode arrivée, pouvant provoquer des pertes de temps pour les élèves, ainsi que des erreurs sur le système.

Lorsque les timbreuses sont déconnectées du serveur ou entre elles, elles vont tenter de se reconnecter au serveur et aux autres timbreuses toutes les 5 secondes.

5. Matériel

a. Le montage

Les timbreuses sont placées dans des boîtes en plastique (Voir la liste des pièces). Ces boîtes ont été usinées pour permettre de placer trois connectiques permettant de faire une interface entre l'intérieur de la boîte et l'extérieur. Les timbreuses fonctionnent sur des Raspberry Pi 3 model B avec un écran tactile de 7 pouces. M. Jean-Marc VULLIAMY à crée un panneau en PMMA et deux supports en métal. Ces supports servent à tenir l'écran collé à la plaque de PMMA. Finalement la plaque de PMMA vient se placer dans la boîte, comme un toit. Sur l'image ci-dessous on peut voir une représentation 3D de la timbreuse dans sa boîte :

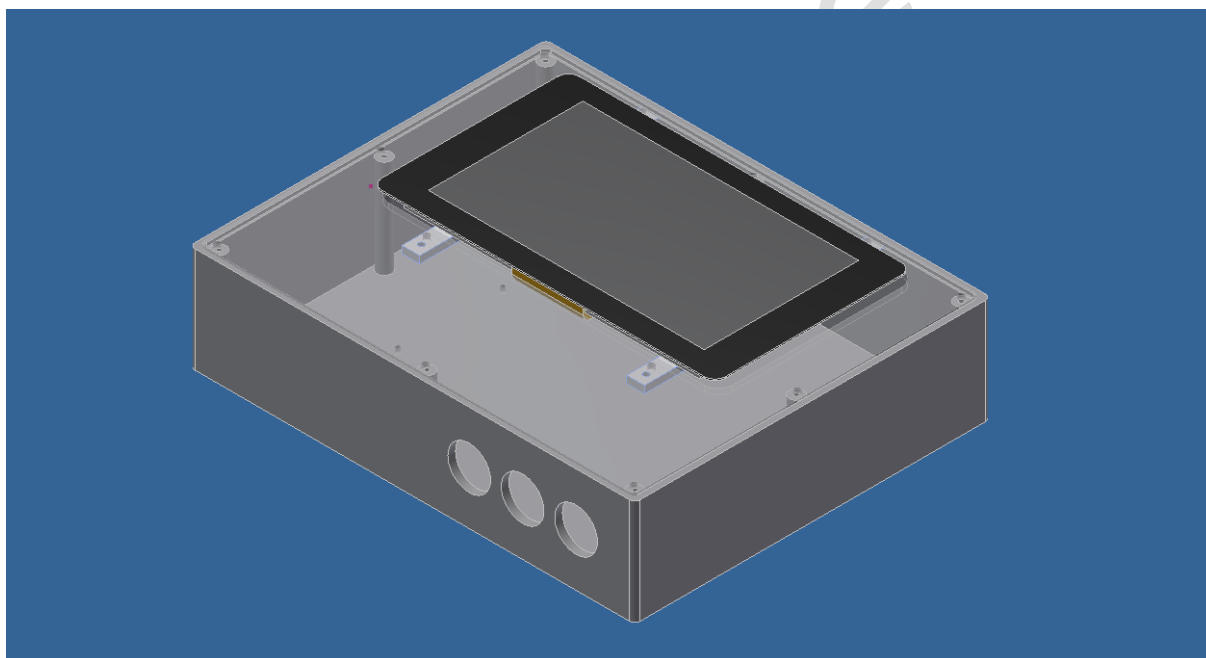


Figure 2 Montage de la timbreuse esclave et du serveur

b. Les appareils

La timbreuse utilise trois appareils pour pouvoir fonctionner correctement. Ceux-ci sont :

- Le Raspberry Pi 3 model b. On appelle généralement ce genre de système un SoC (System On Chips) qui signifie que cet appareil contient tous les éléments nécessaires pour pouvoir fonctionner de lui-même sans autres appareils (à part l'alimentation).
- L'écran 7 pouces pour le Raspberry Pi. L'écran est particulièrement utilisé pour que les utilisateurs puissent voir leurs informations de timbrage lorsqu'ils utilisent le système.

- Le lecteur de carte RFID. Cet élément permet au système de lire l'identifiant unique de la carte de l'élève

6. Conclusion

En conclusion, grâce à un déploiement précoce dans l'avancée du projet (Voir document sur le projet de déploiement en annexe), toutes les erreurs qui sont survenues ont pu être identifiées. Par rapport à la version précédente, de nouvelles fonctionnalités ont été implémentées, comme celles de pouvoir mettre les élèves en horaire bloqué, ou celles de sauvegarder et répliquer les données de timbrage sur de multiples supports pour avoir une certitude d'avoir au moins une sauvegarde des informations.

a. Améliorations possibles

Certaines fonctionnalités pourraient être améliorées pour pouvoir avoir un système plus solide et plus sécurisé :

- Encrypter toutes les communications sur le réseau
- Encrypter la base de données et demander un mot de passe (Un [code PIN](#) par exemple) au démarrage du serveur
- Authentifier les timbreuses auprès du serveur (Par exemple en encryptant l'heure contenue dans le message avec un mot de passe statique entre les appareils pour vérifier l'identité des appareils)
- Sécuriser les entrées utilisateurs sur l'interface de la timbreuse pour éviter toutes les attaques de type [XSS](#)
- Faire en sorte que tous les logs de l'élève ne soient pas envoyés en même temps, mais que seules celles qui vont être affichées soient envoyées au client
- Faire en sorte de ne pas télécharger les détails des élèves à chaque connexion des professeurs, mais télécharger seulement lorsque celui-ci veut les afficher afin de diminuer le trafic réseau
- Faire un test pour savoir si le réseau du CPNV ou la puissance des Raspberry Pi est la cause de la légère latence des requêtes réseaux de la timbreuse.

7. Bibliographie

- [Model 3D d'ensemble écran + Raspberry pi](#)
- [Lecteur RFID datasheet](#)
- [Raspberry pi comparaison](#)
- [Raspberry pi electrocomponent](#)
- [Combien d'Ampère peut fournir un Raspberry pi par ses USB 2.0](#)
- [Base d'icône de la timbreuse](#)
- [Framework Electron](#)

- [Fullcalendar librairie](#)
- [Jquery librairie](#)
- [Moment JS librairie](#)
- [Bootstrap librairie](#)
- [Bootstrap table librairie](#)
- [SB admin 2 bootstrap theme](#)

8. Annexe

a. Les études

- [Node JS](#)
- [Electron](#)
- [Compatibilité avec les Raspberry Pi](#)
- [Choix du port TCP](#)
- [Choix du moteur de base de données](#)
- [Format de dates](#)

b. Les tests

- [Export des CSV](#)
- [Réplication des CSV sur les timbreluses](#)
- [Exportation des CSV et de la base de données du serveur](#)
- [Timbrage hors connexion](#)
- [Algorithme de formatage des secondes](#)
- [Procédure de test graphique](#)

c. Les tutoriels

- Professeurs
- Administrateurs
- Élèves

d. [Cahier des charges](#)

e. [Journal de laboratoire](#)

P1699 - Timbreuse