Journal de bord : Timbreuse

27/02/17 : Documentation sur les différents outils qui vont être utilisés comme NodeJS, Electron et les fonctionnalités de sqlite.

28/02/17 : Début de la mise en place de la base de données. Création de la fonction permettant de générer des bases de données de base.

01/03/17 : Suite de la mise en place de la base de données et début des fonctions de timbrage, c’est-à-dire les fonctions permettant de suivre les heures effectue par les élèves.

02/03/17 : Usinage de la boîte du serveur.

06/03/17 : Mise en place de la pile de requête des timbreuses permettant aux élèves de continuer à timbrer en cas de panne de réseaux. Création d’une documentation en HTML.

06/03/17 : Suite de la mise en place de la pile de requête. Mise en place d’un environnement de test crée avec des machines virtuelles basées sur le système d’exploitation Debian, proche de Raspbian, pour avoir des tests fiables

07/03/2017 : Début de la création de la fonction permettant d’exporter la base de données tout entière sur un clé USB externe.

09/03/2017 : Fin de la fonction d’export de la base de donné. Amélioration du système permettant que les timbreuses restent constamment connecter au serveur et qu’elles restent constamment connecte entre elles.

13/03/2017 : Mise en place de la fonction permettant aux timbreuses de copier leurs CSV de façon miroir lors du timbrage des élèves et réparation d’une erreur de conception sur la fonction permettant aux timbreuses d’être constamment connecté.

14/03/2017 : Fonction de fin de journée et amélioration de la configuration

16/03/2017 : Fin de la fonction de fin de journée. Début de réflexion sur comment appliquer le règlement de la timbreuse. Début de l’implémentation des détections de certaines erreurs utilisateurs.

17/03/2017 : Mise en place de la fonction de pause minimum. L’élève doit faire une pause toutes les quatre heures.

20/03/2107 : Début de la mise en place du système permettant de faire des demandes de congé. On peut créer des demandes de congé, les modifiés ainsi que les effacer. Reste à automatisé leurs mise en places. M.LOCATELLI demande un document décrivant les différents cas de figure **d’erreurs de timbrage ainsi** que les conséquences que cela implique. Un tel document a donc été écrit dans la journée.

21/03/2017 : Mise en place d’une base pour l’interface utilisateur, afin qu’il soit facile de créer des nouvelles pages (+modularité). C’est-à-dire, création d’une page (WEB) de base contenant un élément **iframe** (page web dans une page web) permettant d’afficher d’autre page web à l’intérieur de la page principale. Ça permet de garder une page de menu très modulable avec des pages annexes qui viennent s’y greffer.

23/03/2017 : À la suite d’une discussion avec M.LOCATELLI, le design suivant sera adopté pour la page principale de l’interface des professeurs : une liste sous forme de tableau avec possibilité d’afficher toutes les informations de l’élève dans une **nouvelle page**. Début de la création de cette la page d’accueil de l’interface des enseignants. Des problèmes sont rencontré quant à la communication en utilisant le protocole **IPC** entre le processus de rendue visuel et le processus de méthode pour le logiciel client.

24/03/2017 : Problème dans le processus de rendue visuel : plusieurs librairies reportent que la libraire ***jQuery***n’est pas définit. Il se trouve que il faut employer **la méthode de NodeJS** pour l’inclusion (*require*) plutôt que la méthode HTML (*<script src : ‘’…..’’></script>*). De plus, la communication entre le processus de rendue visuel et le processus de méthode a pu être rétablie en remplacent l’objet **iframe** par un objet **webview** (Protocole supporte par l’API du framework Electron pour des raisons de sécurité). Après la revue du projet avec M.LOCATELLI et M.CORNU, certains documents (cahier des charges, erreurs de timbrage, diagramme de Gantt) ont dû être corrigés. Finalement le document sur les **erreurs de timbrage** a été revu et validé par les deux professeurs. Un document sur le projet de déploiement a été écrit.

27/03/2017 : En début de journée, le **document sur le déploiement** de la timbreuse a été rendu à M.LOCATELLI. Une discussion a eu lieu en début d’après-midi concernant le risque sécuritaire qu’impliquent **les mots de passes navigant en clair** dans le système de la timbreuse. La fonction permettant de **se connecter** a été créée. Une **erreur** est toujours présente cependant, la fonction est inaccessible depuis la page web (Impossible d’appeler le script JS depuis la page HTML). Une **solution** sera trouvée dans la journée de demain.

28/03/2017 : Création de la fonction permettant de crée de nouveaux élèves. On peut remplir les informations directement sur le logiciel et l’élève sera crée correctement dans la base de donnée. Il sera possible par la suite de modifier les informations relatives à l’élève. Lors de la création d’un nouvel élèves on doit crée deux entité dans la base de donnée. La premiere est un « User » : dans cette table va etre stockées les informations personnelles de l’élèves (Nom, prénom, email). La deuxième entité est un « Student » elle contient les informations relatives à la quantité de travail de l’élève.

30/03/2017 : Création de la fonction permettant d’éffacer des élèves. Elles à la même problématique que la création étant donnée que les informations sont organisee dans deux tables différentes de la base de donnée. Rassemblement du matériel en préparation du premier déploiement qui approche. Mise en route d’un des deux raspberry pi, installation des divers logiciels, créations d’une image reinscriptible sur d’autre raspberry pi. Après avoir lancé l’application depuis le Pi, on sait que celui-ci est compatible.

31/03/2017 : Création de la fonction permettant de modifier des élèves déjà existant. Un problème de fuite de mémoire à été réglé. Il s’agissait d’une fonction qui ajoutait des listeners (fonction appelée si un evenement spécifique intervient) sans jamais effacer ceux qui n’ont pas été appelés.

Jalon du vendredi avec M.LOCATELLI. Apport de correctif sur le diagramme de Gantt, il manquait les dates de déploiement.

03/04/2017 : Réparation d’une erreur sur la fonction pour éditer les élèves. Il est en fait impossible d’exécuter plus de 1 niveau de sous-requête au sein d’une requête SQLite. Dernière préparation pour le déploiement du lendemain. Vérification des fonctions de base (tagger). Amélioration de la lecture du tag. L’UID ne sera plus lu dans une boite de texte mais directement commande clavier envoyées à la fenêtre. Préparations d’un tag par élèves pour pouvoir effectuer des tests. Discussion avec M.LOCATELLI sur la différence entre les circuits HC (transistors à effets de champ) et LS.

04/04/2017 : Préparation d’une interface pour les utilisateurs de la timbreuse (élèves) avec l’heure affichée de façon dynamique. Vérification du fonctionnement dans un environnement virtuelle. Déploiement en début d’après-midi. Les premiers tests montrent que les espaces ne sont pas supportés lors du timbrage. Les tags des élèves ne sont pas reconnus. Un formatage des tags lors de la création et de l’édition des élèves doivent être ajoutés.

06/04/2017 : Réparation d’un probléme concernant le timbrage. Lorsque les tag des utilisateurs etaient enregistré avec des espaces, le système ne les reconnaissait pas. Une améliorations de l’interface pour rendre celle-ci plus visible à eu lieu. Une subtilité du language ma oblige a commence à refaire toutes les fonctions utilisant le réseaux. En effet le système lit les requetes venant du reseaux non pas dans leurs totalité mais pendant un temps donné. Il est donc necessaire de crée des fonctions pour compiler ces requetes.

24/04/2017 : Réparer les erreurs décrites par le logiciel serveur de la Timbreuse pendant les vacances. Chercher pourquoi le Pi qui servait de client à griller. La théorie la plus probable est le soleil et un default de fabrication. Remplacement du pi par un nouveau. Amélioration de la fonction de timbrage pour que celle si utilise une pile de type FIFO (First In, First Out). Amélioration de l’interface des professeurs.