

Lycée Ste-Marie 84 Rue Saint-Désiré, 39000 Lons-le-Saunier 03 84 47 09 67 accueil@enseignementprive-lons.fr	BTS Systèmes Numériques Option A Informatique et Réseaux	<u>Session 2020</u>
--	---	----------------------------

Système de gestion de consignes à verrouillage par empreinte digitale

<i>Partenaire professionnel :</i>	<i>Étudiants chargés du projet :</i>		<i>Professeurs ou Tuteurs responsables :</i>
Raison Sociale	Noms	Prénoms	Noms Prénoms
Adresse	-		- ALLARD Morgan
	-		-
	-		-
	-		-
	-		-

Reprise d'un projet : ~~Oui~~ / Non

Présentation générale du système supportant le projet :

Il s'agit de proposer aux élèves un projet d'étude et de réalisation d'un système de consignes modulable, sécurisé et gérable à distance. L'utilité d'une consigne au sein d'un établissement scolaire est bien sûr évidente. Le plus de la proposition est l'abandon de clé de serrurier pour une clé biométrique basée sur l'empreinte digitale de l'utilisateur. Chaque système de consigne est extensible et l'on peut ajouter d'autres casiers. Chaque système est connecté au réseau local de l'école et une application Web d'administration permet de gérer les déblocage et statistiques d'utilisation à distance.

Analyse de l'existant :

Potentiellement toute armoire pourvue d'une serrure peut devenir une consigne avec une serrure biométrique. Le réseau local de l'école via le WIFI peut-être sollicité pour connecter n'importe où un système de consignes et la gestion par un navigateur Web est accessible par n'importe quel ordinateur également connecté.

Expression du besoin :

Il s'agit de modifier ou de créer, un meuble de rangement en ajoutant aux portes des gâches électriques et de relier celles-ci à un contrôleur de commande de type industriel ou bien d'en réaliser un de façon ad-hoc. Ce contrôleur doit simplement être compatible avec le protocole MODBUS. Chaque meuble équipé doit ensuite être relié à un bus de type RS485. Ce bus est pris en charge par le système de consignes.

Ce système permet de réaliser les fonctions de dialogue avec l'utilisateur, d'identification par l'empreinte digitale et les commandes du/des contrôleurs de casier qui ouvre(nt) et ferme(nt) les casiers. Il permet également de répondre aux sollicitations du système d'administration à distance via un micro-service de type REST et le réseau local. Ce système d'administration, doit permettre de gérer l'ensemble des systèmes de consignes connectés et d'agir dessus pour :

- avoir l'état d'utilisation d'un système de consignes
- débloquer le cas échéant, un casier à distance

Le système d'administration doit permettre également de générer des statistiques d'utilisation (localisation, durée moyenne d'utilisation d'un casier, taux de pannes) afin de pouvoir optimiser le placement et la quantité de casiers nécessaires.

Description structurelle du système :

Principaux constituants :	Caractéristiques techniques :
CTL-CAS - Contrôleur de casiers	<ul style="list-style-type: none"> • Pc embarqué type Raspberry Pi sous Linux • Écran LCD commandé par I2C • Encodeur rotatif avec bouton poussoir • Serrure d'administration • Lecteur d'empreinte digitale • RTC I2C • Transceiver RS485/USB • ModBus • Ethernet Wifi • Logiciel de communication et contrôle • Base de données NoSQL
CMD-CAS - Commande de casiers	<ul style="list-style-type: none"> • µcontrôleur type Arduino • carte de puissance pour la commande des gâches électrique pilotée par I2C • Transceiver RS485/UART • ModBus • Firmware de communication et de contrôle
ADM-CAS - Poste d'administration	<ul style="list-style-type: none"> • PC sous Windows • Serveur web Apache • Base de données MySQL • Ethernet

2-Spécifications

2-1 Diagramme de déploiement (figure 1)

Les éléments en jaune sont les éléments à réaliser les autres sont à intégrer. Il y a trois systèmes :

- Le système de contrôle des casiers
 - situé à côté des casiers et intégré dans un boîtier solide comportant les éléments d'interaction avec l'utilisateur
 - il relie le(s) système(s) de commande via le bus RS485
 - il est connecté au réseau local de l'école
- Le système de commande des casiers
 - situé dans le meuble des casiers et raccordé sur le bus RS485 relié au contrôleur de casiers, il relie chaque gâche à la platine de commande
 - le système est adressable et utilise ModBus pour communiquer.
- Le système d'administration
 - situé n'importe où sur un ordinateur relié sur le réseau de l'école, dans un endroit sécurisé.

2-2 Diagramme de cas d'utilisation (figure 2)

En rose les fonctionnalités liées au logiciel du contrôleur de casier. En cyan les fonctionnalités du site d'administration.

Contrôleur de casier

Utiliser un casier et s'identifier

NB : Un seul casier par utilisateur par système de contrôle est autorisé.

NB : Chaque action d'ouverture et de fermeture d'un casier est enregistrée et horodatée. Ceci afin de générer des statistiques d'utilisation.

NB : Les empreintes étant enregistrées dans le lecteur et leur nombre étant limité, un algorithme d'éviction sera utilisé (type LRU) en cas de saturation de l'espace mémoire.

Pour déposer :

1. L'utilisateur ouvre un casier libre en sélectionnant son numéro grâce à l'encodeur rotatif. Son numéro s'affiche sur l'écran.
2. L'utilisateur dépose un objet et referme le casier
3. L'utilisateur sélectionne l'option: sceller et est invité à présenter son index
 1. si l'empreinte est connue le message : « consigne scellée » est affiché
 2. si l'empreinte est inconnue les messages pour enregistrer l'empreinte sont affichés en séquence, puis le message « consigne scellée » s'affiche
4. la consigne est fermée jusqu'à présentation de l'empreinte ou déverrouillage administratif

Pour libérer :

1. L'utilisateur appose son empreinte sur le lecteur, si une consigne est scellée, celle-ci s'ouvre, sinon un message indique qu'il n'y a rien de scellé sur cette empreinte.
2. La consigne est libérée, l'utilisateur reprend son bien.

Débloquer un casier

Si la clé d'administration est tournée, des opérations de maintenance sont possibles comme débloquent un casier.

1. La clé est tournée, l'afficheur indique de sélectionner le numéro de casier
2. le casier s'ouvre
3. La clé est sortie, le mode administrateur est terminé.

NB: l'opération est enregistrée avec un intitulé : « déblocage »

Site d'administration

Authentification

Afin d'accéder à l'application, un compte d'administration par défaut et un mot de passe sont nécessaires.

Les opérations de connexion et déconnexion sont proposées.

Gérer les consignes et les statistiques

Les systèmes de contrôle de casiers étant connectés, un algorithme de scrutation permet de les lister dans l'application.

L'application propose une arborescence indiquant la composition de la consigne et de l'état des casiers.

Pour obtenir des informations de la part d'un contrôleur ou d'agir dessus, des requêtes GET/DELETE du microservice REST sont utilisées.

Les actions sont fonction de la nature des nœuds sélectionnés. Si plusieurs nœuds font parties de la sélection, l'action sera répétée.

Les actions sur un contrôleur de casiers sont :

1. **Télécharger les statistiques**
 1. télécharge les données présentes en base NoSQL et la valide si l'action s'est bien terminée.
2. **Afficher les statistiques**
 1. provoque le téléchargement des données (comme en 1.1) et l'affichage des synthèses de toutes les statistiques récoltées précédemment et inscrites dans la base de données MySQL (voir détails plus bas)
3. **Déverrouiller un casier**
 1. après sélection d'un casier sur un système, la commande d'ouverture du casier est envoyée. Une confirmation d'ouverture est renvoyée.
4. **Régler la date et l'heure**
 1. après sélection d'un système, permet de régler l'heure et la date à distance

NB : la configuration permet d'ajouter un contrôleur en fournissant son adresse IP et son emplacement. Lors de l'ajout, le nombre de casiers ainsi que les statistiques sont téléchargées.

NB : la configuration permet également de changer le mot de passe pour l'accès à l'interface.

Complément sur les statistiques élaborées.

Il s'agit de savoir :

- le taux d'utilisation global par emplacement
 - par mois
 - par semaine
- le temps moyen d'utilisation d'un casier (temps entre dépôt et retrait)
- le taux d'intervention (déblocage)
 - par mois

NB : Les informations seront présentées sous forme de diagrammes à barres et par données

textuelles.

Les données de base d'un contrôleur sont :

- emplacement
- nombre de casiers
- adresse IP
- nombre de casiers libres
- nombre de casiers scellés
- horodatage courant
- liste des événements d'ouverture et fermeture par casier

NB : les données échangées via le micro-service **REST** sont de type JSON.

Exemple d'interface envisagée :

Consigne Admin

Contrôleurs

Configuration

Contrôleurs

<input type="checkbox"/>	Etat	Emplacement	Dernier télécha...	Informations
<input type="checkbox"/>	En ligne	B320	11/11/2019 13:30	30% utilisé
<input checked="" type="checkbox"/>	Hors ligne	D320	11/11/2019 13:30	Intervention en cours
<input type="checkbox"/>	En ligne	D322	11/11/2019 13:30	75% utilisé

Télécharger statistiques

Tracer statistiques

Mettre à l'heure

Casiers

<input type="checkbox"/>	Numéro	Etat	Dernière utilisation	Taux d'utilisation	Taux d'intervention
<input type="checkbox"/>	1	Libre	11/11/2019 à 15h00	50%	0%
<input checked="" type="checkbox"/>	2	Scellé	01/11/2019 à 15h00	75%	0%
<input type="checkbox"/>	3	Libre	01/01/2019 à 15h00	10%	15%

Débloquer

Figure 1

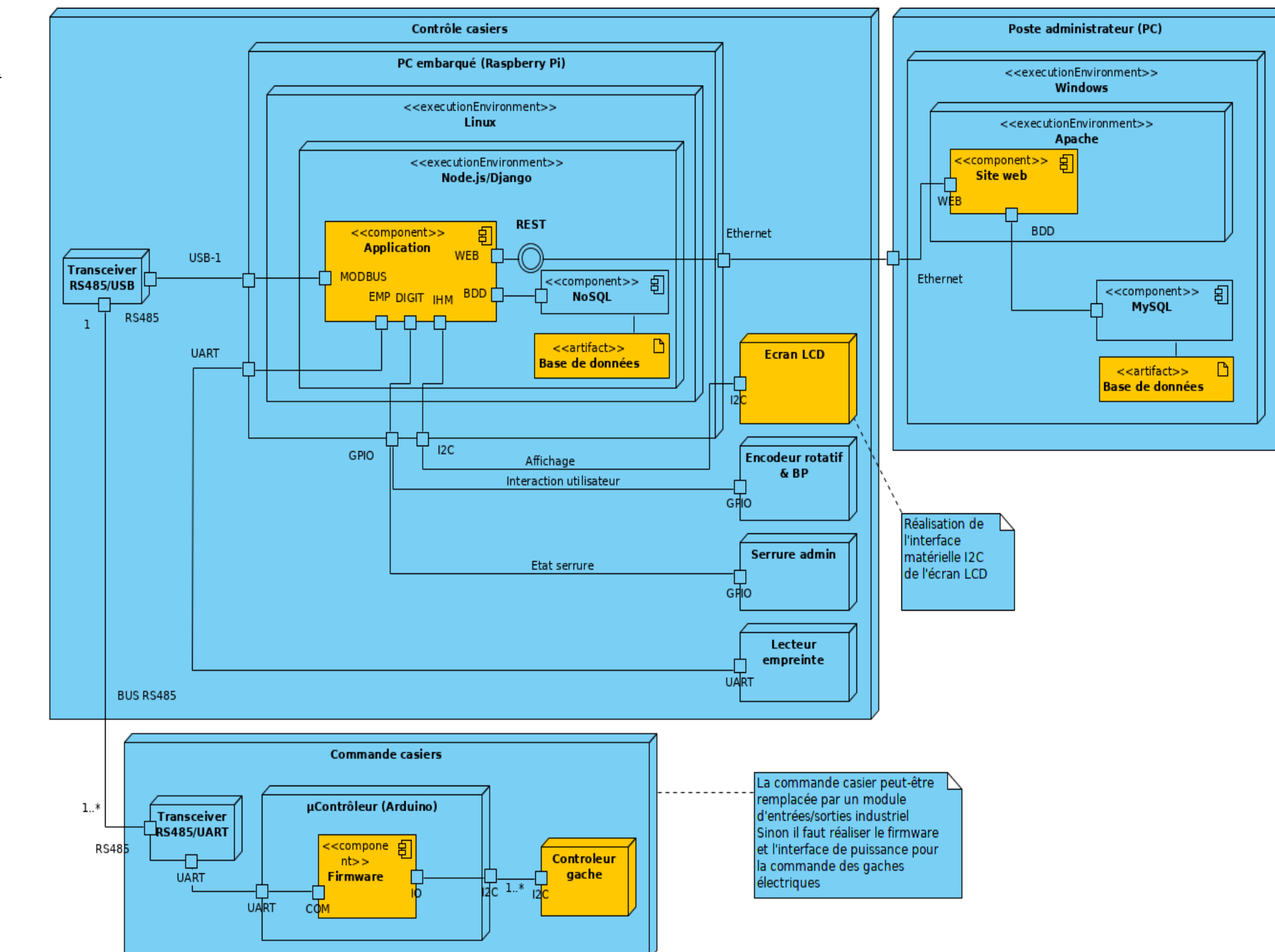
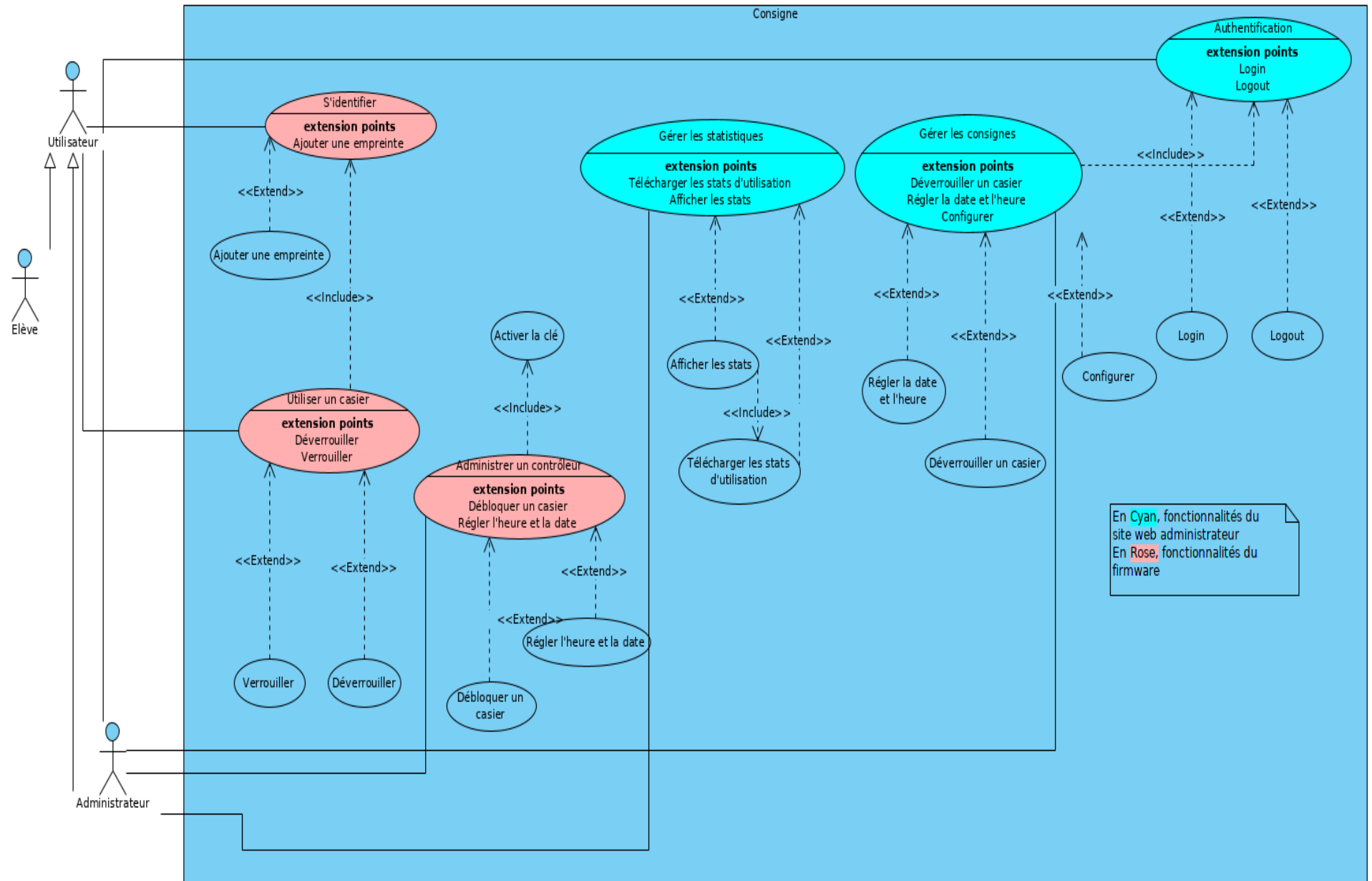
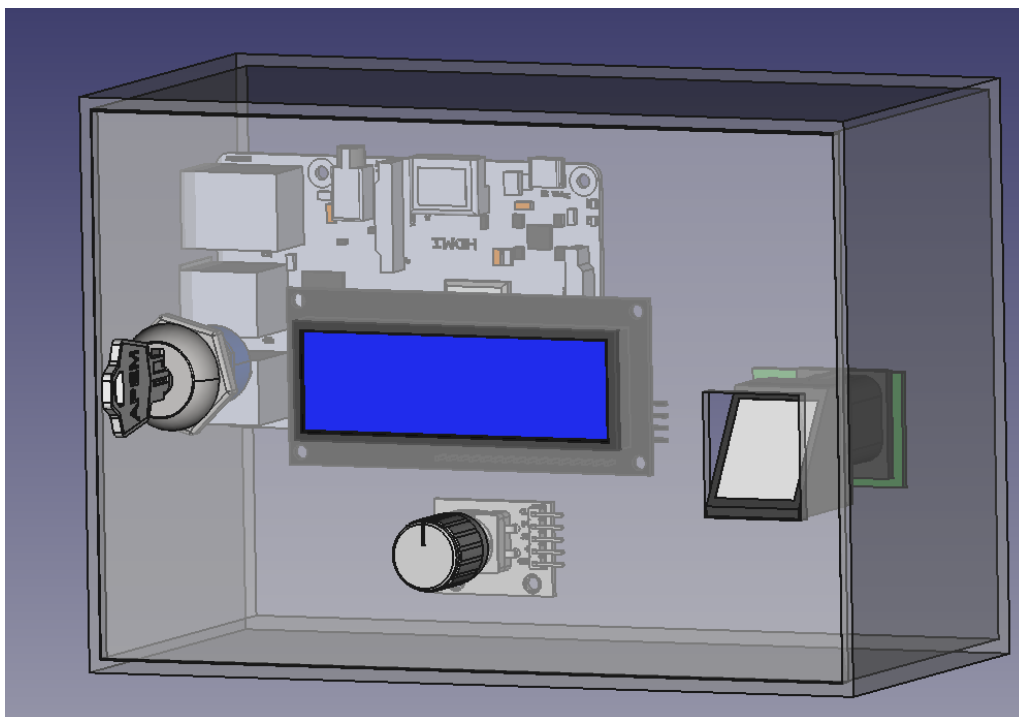


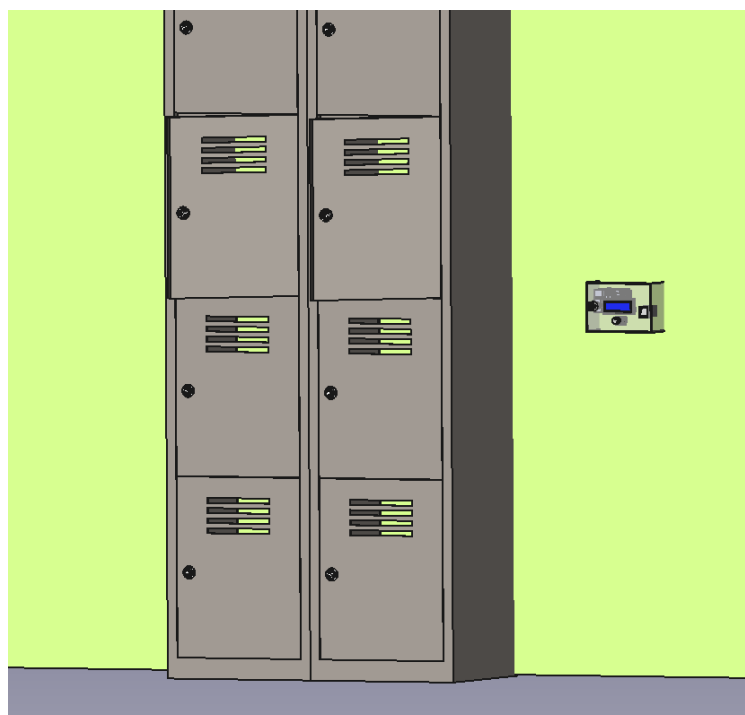
Figure 2



Vue générale



Boîtier du système de contrôle de la consigne



En situation avec les casiers

2-3 Contraintes de réalisation

Contraintes financières (budget alloué)

Le budget reste inférieur à 500€ si plusieurs éléments sont fournis par l'école.

Contraintes de développement (matériels et/ou logiciels imposés)

Les environnements sont Windows, Linux et firmware AVR. Les conteneurs logiciels sont Apache, Node.js (javascript) ou Django (Python), MySQL, NoSQL (node.js)

Le format des données échangées sont JSON, ModBus

Les matériels sont Raspberry PI, Arduino, et tous les éléments nécessaires au projet.

Contraintes de qualité (conformité, délais, ...)

Le délai de la réalisation est la durée du projet dans le cadre du BTS SN IR.

Le code doit répondre aux critères de qualité suivants :

- Entête de fichier précisant : auteur, date de création, de dernière modification, outils de production utilisés.
- Entête de fonctions précisant le rôle de la fonction et l'utilisation des paramètres.
- Tous les commentaires nécessaires à une bonne compréhension du code.

Livrables :

La documentation : doit être complète, homogène et non redondante. L'auteur de chaque page est identifiable. Elle comporte :

- **Une partie d'analyse** (spécifications) servant à décrire la vision et les contraintes du système à réaliser :
 - o Description du problème à traiter.
 - o Description des fonctionnalités et comportements.
 - o Description des données utilisées (format etc...)
 - o Description des interactions utilisateur avec définition des écrans.
 - o Description des contraintes.
NOTA : l'utilisation des diagrammes **UML** est conseillé afin de faciliter la compréhension.
- **Une partie conception** en relation avec les spécifications servant à dire comment celles-ci ont été réalisées/implémentées :
 - o Technologies utilisées (API, conteneurs logiciel, matériel etc ...)
 - o Configuration
 - o Algorithmes et données mis en œuvre (**UML**)
- **Un guide d'installation et d'utilisation.**

Le code : doit répondre aux critères de qualité suivants :

- Entête de fichier précisant : auteur, date de création, de dernière modification, outils de production utilisés.
- Entête de fonctions précisant le rôle de la fonction et l'utilisation des paramètres.
- Tous les commentaires nécessaires à une bonne compréhension du code.

Les tests fonctionnels : doivent permettre de contrôler le respect de l'aspect des écrans et des fonctionnalités issues des spécifications.

Pour chaque référence de spécification, il faut une fiche de test décrivant les opérations à mettre en œuvre afin de tester la réalisation de celle-ci.

NOTA : le code sera sauvegardé sur un serveur GitHub public

Contraintes de fiabilité, sécurité

Les appareillages électriques doivent être assemblés dans le respect des normes en vigueur.

2-4 Ressources mises à disposition des étudiants (Logiciels / Matériels / Documents)

Chaque étudiant dispose d'un ordinateur sous **Windows ou Linux (Ubuntu)**, doté de l'environnement de développement **Eclipse ou Visual Studio Code**. Il dispose de la suite **LibreOffice**. Pour la modélisation **UML**, l'environnement **Visual Paradigm** sera utilisé.

Pour le projet, les étudiants disposent du matériel suivant :

- D'une carte Raspberry PI (CTL-CAS - Contrôleur de casiers)
- D'une carte Arduino Uno (CMD-CAS - Commande de casiers)
- Un serveur PC sous Windows ou Linux (ADM-CAS - Poste d'administration)
- Transceiver RS485/USB
- Transceiver RS485/UART
- Lecteur d'empreinte digitale
- Afficheur LCD
- Encodeur rotatif
- Clé avec contact
- Fournitures électriques/électroniques (alimentations, composants ...)

3 – Répartitions des fonctions ou cas d'utilisation par étudiant

Fonctions à développer et tâches à effectuer

<p>Étudiant 1 - IR</p>	<p>Système contrôleur des casiers</p> <p>Matériel :</p> <ul style="list-style-type: none"> • Conception et réalisation de l'interface I2C pour l'afficheur LCD • <p>Logiciel :</p> <ul style="list-style-type: none"> • Développement du code permettant la gestion du module à distance <ul style="list-style-type: none"> ◦ sérialisation des données (REST, JSON) ◦ micro-services ◦ gestion de la base de données locale • Développement du code permettant la communication avec le module de commande <ul style="list-style-type: none"> ◦ protocole ModBus 	<p><i>Installation :</i></p> <ul style="list-style-type: none"> • Environnement de développement Visual Studio Code • Création du dépôt GitHub <p><i>Mise en œuvre :</i></p> <ul style="list-style-type: none"> • Raspberry PI • serveur Node.js • API express.js • API I2C • base de données NoSQL <p><i>Configuration :</i></p> <ul style="list-style-type: none"> • partage sur réseau pour le développement de l'application sur le raspberry Pi • Configuration du dépôt GitHub <p><i>Réalisation :</i></p> <ul style="list-style-type: none"> • Base de données NoSQL • Application Node.js <ul style="list-style-type: none"> ◦ partie API REST de l'application ◦ partie communication ModBus • Interface I2C pour l'afficheur <p><i>Documentation :</i></p> <ul style="list-style-type: none"> • Dossier de spécifications et conception du contrôleur de casiers : <ul style="list-style-type: none"> ◦ partie base de données ◦ partie API REST de l'application ◦ partie communication ModBus ◦ Schéma électronique de l'interface I2C pour l'afficheur ◦ Intégration de l'interface I2C • Dossier test • Manuel
<p>Étudiant 2 - IR</p>	<p>Système contrôleur des casiers</p> <p>Matériel :</p> <ul style="list-style-type: none"> • Intégration du lecteur 	<p><i>Installation :</i></p> <ul style="list-style-type: none"> • Environnement de développement Visual Studio Code <p><i>Mise en œuvre :</i></p>

	<ul style="list-style-type: none"> d'empreinte • Intégration de la clé d'administration • Intégration de l'encodeur rotatif • Intégration de l'affichage LCD <p>Logiciel :</p> <ul style="list-style-type: none"> • Développement du code permettant la gestion des périphériques (lecteur d'empreintes, clé, encodeur et afficheur) <ul style="list-style-type: none"> ◦ I2C, IO, UART • Développement du code d'Interaction Homme Machine (IHM) <ul style="list-style-type: none"> ◦ Cas : S'identifier ◦ Cas : Utiliser un casier ◦ Cas : Administrer un contrôleur 	<ul style="list-style-type: none"> • serveur Node.js • API <i>rpi-gpio, i2c-bus, raspi-serial, modbus-serial</i> <p>Configuration :</p> <ul style="list-style-type: none"> • partage sur réseau pour le développement de l'application sur le raspberry Pi • Configuration du dépôt GitHub <p>Réalisation :</p> <ul style="list-style-type: none"> • Application Node.js <ul style="list-style-type: none"> ◦ s'identifier ◦ utiliser un casier ◦ administrer un contrôleur • intégration des éléments matériels <p>Documentation :</p> <ul style="list-style-type: none"> • Dossier de spécifications et conception du contrôleur de casiers : <ul style="list-style-type: none"> ◦ s'identifier ◦ utiliser un casier ◦ administrer un contrôleur • Dossier test • Manuel
Étudiant 3 - IR	<p>Système de commande des casiers</p> <p>Matériel :</p> <ul style="list-style-type: none"> • Conception et réalisation du circuit de commande des gâches électriques. • Câblage et équipement des casiers (gâches) <p>Logiciel :</p> <ul style="list-style-type: none"> • Développement du firmware du système de commande des casiers <ul style="list-style-type: none"> ◦ Communication avec le contrôleur (ModBus) ◦ pilotage du circuit de commande (IO) 	<p>Installation :</p> <ul style="list-style-type: none"> • Environnement de développement Eclipse • Installation du plugin CDT-Arduino (eclipse) <p>Mise en œuvre :</p> <ul style="list-style-type: none"> • Arduino Uno • Électronique de commande • API <i>arduino-modbus</i> <p>Configuration :</p> <ul style="list-style-type: none"> • Configuration du dépôt GitHub • Apache • MySQL <p>Réalisation :</p> <ul style="list-style-type: none"> • Firmware du système de commande des casiers • Conception et réalisation du circuit de commande des gâches électriques <p>Documentation :</p> <ul style="list-style-type: none"> • Dossier de spécifications et conception du système de commande des casiers :

		<ul style="list-style-type: none"> ◦ partie commande ◦ partie communication ModBus • Dossier test • Manuel
Étudiant 4 - IR	<p style="text-align: center;">Système d'administration</p> <p>Logiciel :</p> <ul style="list-style-type: none"> • Installation, configuration et intégration des services et API (Apache, MySQL, PHP) • Conception et développement de la base de données. • Développement de l'application d'administration <ul style="list-style-type: none"> ◦ Cas : Gérer les statistiques ◦ Cas : Gérer les consignes ◦ Cas : Authentification ◦ Communication avec le contrôleur (REST) 	<p><i>Installation :</i></p> <ul style="list-style-type: none"> • Environnement de développement Eclipse • Installation du plugin php • Serveur Apache • Serveur MySQL <p><i>Mise en œuvre :</i></p> <ul style="list-style-type: none"> • <i>PHP</i> • <i>REST</i> • <i>HTML5, CSS3</i> • <i>MySQL</i> <p><i>Configuration :</i></p> <ul style="list-style-type: none"> • <i>Configuration du dépôt GitHub</i> <p><i>Réalisation :</i></p> <ul style="list-style-type: none"> • Application Web • Base de données <p><i>Documentation :</i></p> <ul style="list-style-type: none"> • Dossier de spécifications et conception du système d'administration : <ul style="list-style-type: none"> ◦ authentification ◦ gérer les consignes ◦ gérer les statistiques ◦ API REST ◦ Base de données • Dossier test • Manuel

4 - Exploitation pédagogique – Compétences terminales évaluées

Tâches	Revues	Contrats de tâche	Compétences	Candidat_1	Candidat_2	Candidat_3	Candidat_4
		Expression fonctionnelle du besoin					
T1.4	R2	Vérifier la pérennité et mettre à jour les informations.	C2.1	x	x	x	x
T2.1	R2	Collecter des informations nécessaires à l'élaboration du cahier des charges préliminaire.	C2.2	x	x	x	x
T2.3	R2	Formaliser le cahier des charges.	C2.3 C2.4	x	x	x	x
T3.1	R2	S'approprier le cahier des charges.	C3.1	x	x	x	x
T3.3	R2	Élaborer le cahier de recette.	C3.5	x	x	x	x
T3.4	R2	Négocier et rechercher la validation du client.	C2.4	x	x	x	x
		Conception					
T4.2	R3	Traduire les éléments du cahier des charges sous la forme de modèles.	C3.1 C3.3	x	x	x	x
T5.1	R3	Identifier les solutions existantes de l'entreprise.	C3.1 C3.6	x	x	x	x
T5.2	R3	Identifier des solutions issues de l'innovation technologique	C3.1 C3.6	x	x	x	x
T4.3	R3	Rédiger le document de recette.	C4.5	x	x	x	x
T6.1	R3	Prendre connaissance des fonctions associées au projet et définir les tâches.	C2.4 C2.5	x	x	x	x
T6.2	R3	Définir et valider un planning (jalons de livrables).	C2.3 C2.4 C2.5	x	x	x	x
T6.3	R3	Assurer le suivi du planning et du budget.	C2.1 C2.3 C2.4 C2.5	x	x	x	x
		Réalisation					
T7.1	R3	Réaliser la conception détaillée du matériel et/ou du logiciel.	C3.1 C3.3 C3.6	x	x	x	x
T7.2	RF	Produire un prototype logiciel et/ou matériel.	C4.1 C4.2 C4.3 C4.4	x	x	x	x
T7.3	RF	Valider le prototype.	C3.5 C4.5 C4.6	x	x	x	x
T7.4	RF	Documenter les dossiers techniques et de maintenance	C2.1 C4.7	x	x	x	x
T9.2	RF	Installer un système ou un service.	C2.5	x	x	x	x
T10.3	RF	Exécuter et/ou planifier les tâches professionnelles de MCO.	C2.5	x	x	x	x
T11.3	RF	Assurer la formation du client.	C2.2 C2.5	x	x	x	x
T12.1	RF	Organiser le travail de l'équipe.	C2.3 C2.4 C2.5	x	x	x	x
T12.2	RF	Animer une équipe.	C2.1 C2.3 C2.5	x	x	x	x

		Vérification des performances attendues						
T9.1	RF	Finaliser le cahier de recette.	C3.1 C3.5 C4.5	x	x	x	x	

5 – Planification des tâches du projet

Tâches		Prévisionnel		Réal		Semaines calendaires (rose revues et soutenance)																																													
		Début	Durée	Début	Durée	% accompli	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28																				
TG1	Prise en compte du cahier des charges et des critères de validation	4	1			0,00 %						Vacances d'hiver																																							
TG2	Prise en compte du matériel et des logiciels	4	3			0,00 %																																													
TG3	Rédiger les scénarios d'utilisation	4	3			0,00 %																																													
TG4	Prototyper les IHM	5	3			0,00 %																																													
TG5	Définition de l'architecture logicielle	6	6			0,00 %																																													
TG6	Gérer la planification	5	18			0,00 %																																													
TG7	Assurer la traçabilité des travaux	5	18			0,00 %																																													
Etudiant 1																																																			
TE1.1	Conception, réalisation et tests de la platine I2C pour l'afficheur	11	1			0,00 %																																													
TE1.2	Prise en main de l'API express et serveur Node.js	12	1			0,00 %																																													
TE1.3	Conception et réalisation de la base de données NoSQL	12	1			0,00 %																																													
TE1.4	Développement de l'API REST de l'application	13	2			0,00 %																																													
TE1.5	Prise en main de l'API modbus	15	1			0,00 %																																													
TE1.6	Développement de la communication modbus	16	2			0,00 %																																													
TE1.7	Intégration tests et documentation	20	4			0,00 %																																													
Etudiant 2																																																			
TE2.1	Prise en main du serveur Node.js et des API gpio, serial et i2c	11	2			0,00 %																																													
TE2.2	Prise en main du lecteur d'empreinte et de l'encodeur rotatif	12	1			0,00 %																																													
TE2.3	Développement de la partie « s'identifier »	13	1			0,00 %																																													
TE2.4	Développement de la partie « utiliser un casier »	14	2			0,00 %																																													
TE2.5	Développement de la partie « administrer un contrôleur »	15	3			0,00 %																																													
TE2.6	Intégration tests et documentation	20	4			0,00 %																																													
Etudiant 3																																																			
TE3.1	Conception, réalisation et tests de la platine de commande des gâches électriques	11	2			0,00 %																																													
TE3.2	Prise en main de l'API modbus	13	1			0,00 %																																													
TE3.3	Développement du firmware de communication et de commande	14	2			0,00 %																																													
TE3.4	Câblage et intégration des équipements	16	2			0,00 %																																													
TE3.5	Intégration tests et documentation	20	4			0,00 %																																													
Etudiant 4																																																			
TE4.1	Conception et réalisation de la base de données MySQL	11	1			0,00 %																																													
TE4.2	Développement de la partie « authentification »	12	1			0,00 %																																													
TE4.3	Développement de la partie « gérer les consignes »	13	2			0,00 %																																													
TE4.4	Développement de la partie « gérer les statistiques »	15	3			0,00 %																																													
TE4.5	Intégration tests et documentation	20	4			0,00 %																																													

6 – Conditions d'évaluation pour l'épreuve E62

6-1 Disponibilité des équipements lors de la soutenance finale

L'équipement sera-t-il disponible ? Oui

6-2 Atteinte des objectifs du point de vue client

- Utilisation d'un casier
 - dépôt
 - retrait
- Gestion des casiers disponibles (simulation d'occupation)
 - dépôt
 - retrait
- Déverrouillage d'un casier par l'administrateur (utilisation de la clé)
- Réglage de la date et de l'heure par l'administrateur (utilisation de la clé)
- Authentification pour l'accès à l'application d'administration
- Déverrouillage d'un casier à distance
- Réglage de la date et de l'heure à distance
- Téléchargement et affichage des statistiques d'utilisation

6-3 Atteinte des objectifs fixés aux candidats

Étudiant 1 :

- Affichage des informations sur l'écran LCD
- Commandes à distance fonctionnelles
 - réglage de la date et de l'heure
 - déverrouillage à distance
 - téléchargement des statistiques d'utilisation
- Enregistrement des statistiques d'utilisation (administration)

Étudiant 2 :

- Gestion des casiers
 - menus de sélection, affichage des informations et interaction avec l'utilisateur
 - comptabilité des ressources disponibles
- Gestion des empreintes
 - ajout et suppression (algorithme d'éviction)
 - reconnaissance
- Enregistrement des statistiques d'utilisation (usage courant)
- Communication modbus

Étudiant 3 :

- Gestion des gâches
 - fonctionnement des commandes de gâches
 - communication modbus
 - câblage et réalisation générale

Étudiant 4 :

- Interface homme-machine (esthétique et ergonomie)
- Administration
 - authentification
 - commandes à distance
 - déverrouillage
 - réglage de la date et de l'heure

- téléchargement des statistiques
- configuration
- affichage des statistiques d'utilisation

<i>Avis de la commission</i>

- Les concepts et les outils mis en œuvre par le candidat (1-2-3-4-5)... correspondent au niveau des exigences techniques attendu pour cette formation :

oui / à reprendre pour le candidat (1-2-3-4-5)

- L'énoncé des tâches à réaliser par le candidat (1-2-3-4-5)... est suffisamment complet et précis :

oui / à reprendre pour le candidat 1-2-3-4-5

- Les compétences requises pour la réalisation ou les tâches confiées au candidat (1-2-3-4-5) sont en adéquation avec les savoirs et savoir-faire exigés par le référentiel :

oui / à reprendre pour le candidat (1-2-3-4-5)

- Le nombre d'étudiants est adapté aux tâches énumérées :

oui / trop / insuffisant

Commentaires

Date :

Le président de la commission