Application d'automatisation du contrôle des sorties hors de l'établissement

WEBER Benjamin

31/08/2018

Sommaire

Partie I

Abrégé

1.1 Objectif de l'établissement

Mettre en place un système permettant l'automatisation du suivi personnalisé des élèves à la sortie de l'établissement. En d'autres termes, automatiser la vérification de l'emploi du temps des élèves à la sortie de l'établissement en tenant compte des absences et punitions non clôturées propre à chaque élève.

1.2 Travail réalisé

- Création d'un cahier des charges : diagramme du besoin, diagramme pieuvre
- Proposition de solutions techniques répondant au cahier des charges : proposition d'une solution technique permettant l'identification rapide des élèves à la sortie par l'intermédiaire de la technologie RFID, réalisation de diagrammes SYSML représentant le fonctionnement global du logiciel
- Réalisation du logiciel :
  - Création d'un serveur local à l'aide du protocole de communication HTTP
  - Création d'une base de données locale SQL
  - Gestion sécurisées des mots de passes au sein du serveur
  - Gestion des droits d'accès au serveur
  - Création d'une interface utilisateur permettant la communication avec le serveur et la manipulation des données présentes dans la base de données
  - Création d'une charte graphique pour l'interface client
  - Création de l'interface client
    - Utilisation des langages HTML, CSS, JavaScript, jQuery
    - Utilisation de requêtes POST
    - Utilisation de requêtes AJAX
    - Création de contenu dynamique
- Proposition d'utilisation d'un ordinateur embarqué à la sortie de l'établissement
- Élaboration du dispositif à la sortie de l'établissement
- Rédaction d'un code dans l'optique d'être complété par un autre programmeur
- Présentation du logiciel aux utilisateurs

Partie II

Présentation de l'application

1.3 Objectif

1.4 Base de données

L'application dispose d'une base de données locale au serveur (afin d'éviter l'utilisation d'un tiers service d'hébergement en ligne pouvant poser des problèmes de sécurité). La configuration de cette base de données est donnée figure 1.4.

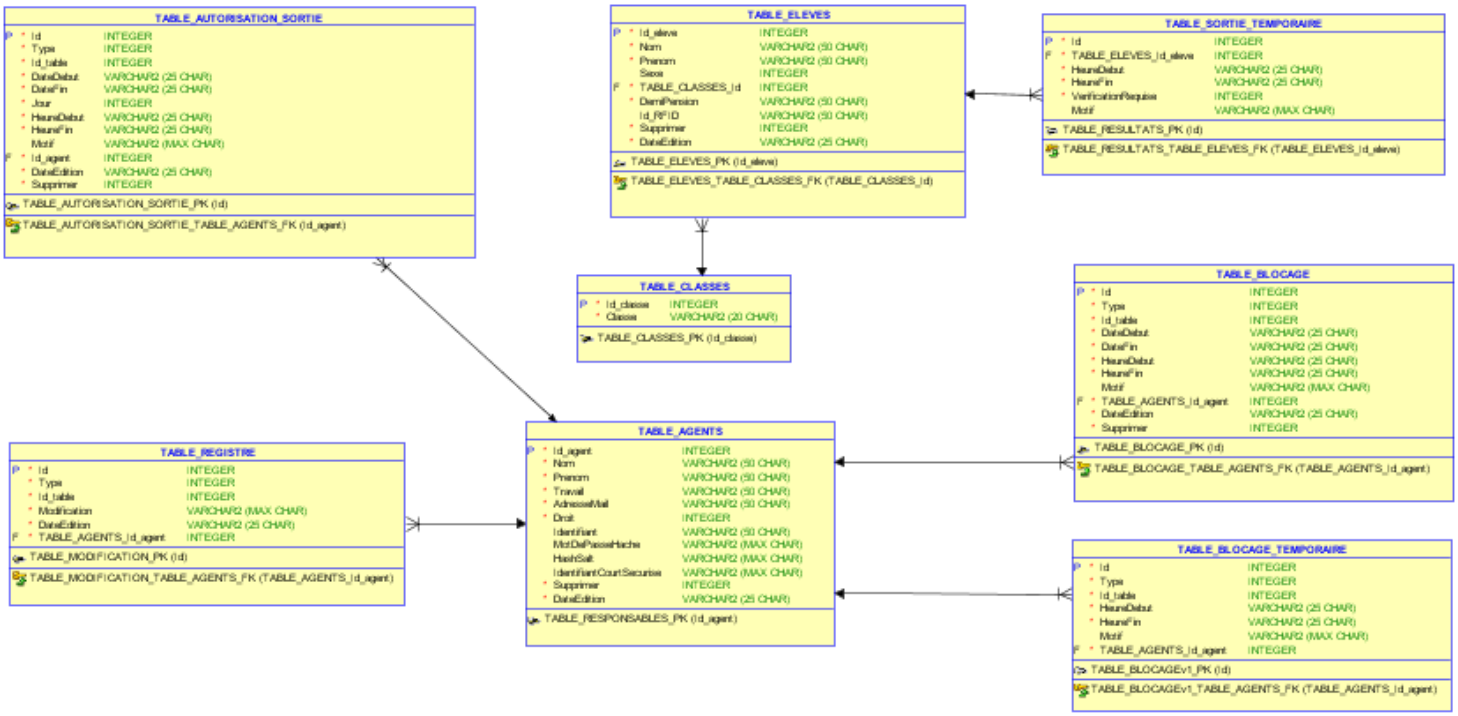


Figure 1.1: Configuration de la base de données utilisée par l'application

1.4.1 Configuration

Table élèves :

Cette table contient toutes les informations des élèves nécessaires au fonctionnement de l'application. Dans les faits, la raison d'être de cette table est de permettre une correspondance entre un numéro d'identification (comme un identifiant RFID) et un élève. S'ajoute à cela le fait que nous devons récupérer en ligne des informations sur les élèves. Ce faisant, il est nécessaire de disposer d'une table permettant de faire la correspondance interne entre un élève et son identifiant RFID et la correspondance externe entre un élève de la base de données locale et la source de données externe. Cette table est composée des colonnes suivantes :

Nom de la colonne	Représentation informatique	Description
Id_eleve	integer	Clé primaire de la table. Son unicité dans la table permet de l'utiliser comme identifiant interne dans la base de données (comme clé étrangère)
Nom	varchar(50 char)	Nom de l'élève utilisé par la source de données externe
Prénom	varchar(50 char)	Prénom de l'élève utilisé par la source de données externe
Id_classe	integer	Représente l'identifiant de la classe de l'élève (clé étrangère)
Sexe	integer	Représente le sexe de l'élève utilisé par la source de données externe. En faisant toute abstraction du genre de la personne cette valeur indique si la personne dispose des organes génitaux féminin ou masculin. Comme la plus part des sources de données externes utilisent des représentations textuelles du sexe (comme F pour le sexe féminin et M pour le sexe masculin), le parti a été pris de convertir cette dernière en valeur entière alors indépendante de toute représentation culturelle
DemiPension	varchar(50 char)	Cette colonne représente les jours de demi-pension de l'élève enregistrés en sous la forme d'un tableau d'entiers converti en texte. Chaque case du tableau représente un jour de la semaine, en sachant que la case d'indice 0 est celle représentant dimanche. Une valeur est donné à ces cases en fonction que l'élève est demi pensionnaire ou non le jour correspondant. Le parti pris a été de stocker l'intégralité du tableau dans une même colonne pour, d'une part, faciliter l'accès à ces informations (puisque tout est localisé au même endroit), et d'autre part, le tableau étant de très faible taille il ne prendra qu'une infime partie de la mémoire vive lors de l'analyse du texte pour le convertir en tableau. De plus, lorsque les jours de demi-pension d'un élève sont modifiés, la totalité de la valeur est remplacée (sans y opérer des modifications intermédiaires)
Id_RFID	varchar(50 char)	Identifiant RFID associé à l'élève
Supprimer	integer	Lorsqu'une entrée est supprimée de la base de données par un utilisateur, en réalité cette entrée n'est pas directement supprimée. L'entrée est uniquement mise à jour en changeant la valeur de cette colonne. C'est seulement au bout d'une certaine période qu'elle sera définitivement supprimée de la base de données. Cette variable représente donc l'état de suppression de l'entrée correspondante (supprimée ou non supprimée).
DateEdition	varchar(25 char)	Représente la date et l'heure de la dernière mise à jour de l'entrée correspondante. La partie pris a été de convertir toutes les variables de représentation temporelle en texte avant de les sauvegarder dans la base de données et ce afin d'éviter toute conversion automatique susceptible de dépendre de l'ordinateur utilisé

Table classe :

Cette table contient toutes les représentations textuelles des classes utilisées par l'établissement. Par exemple : "4A", "6b", "Classe 1", ...
Cette table est composée des colonnes suivantes :

Nom de la colonne	Représentation informatique	Description
Id_classe	integer	Clé primaire de la table servant d'identifiant interne de la classe dans la base de données
Classe	varchar(20 char)	Représentation textuelle de la classe

Table agent :

Cette table contient toutes les informations propres aux agents utilisant l'application. Dans le détail, chaque personne doit s'authentifier avant de pouvoir utiliser l'application et ce dans le but d'attribuer la responsabilité des actions effectuées sur l'application à une personne physique. Par exemple, si un agent décide de laisser sortir un élève en utilisant l'application, il sera inscrit dans la base de données que cet agent a laissé sortir cet élève.
Cette table est composée des colonnes suivantes :

Nom de la colonne	Représentation informatique	Description
Id_agent	integer	Clé primaire de la table utilisée comme identifiant interne de l'agent dans la base de données
Nom	varchar(50 char)	Nom de l'agent
Prénom	varchar(50 char)	Prénom de l'agent
Travail	varchar(50 char)	Fonction de l'agent dans l'établissement
AdresseMail	varchar(50 char)	Adresse e-mail fournit par l'agent. Cette adresse mail est utilisée pour envoyer les informations de connexion à l'agent (comme l'identifiant et le mot de passe). L'envoi du mot de passe par e-mail permet d'éviter les intermédiaires : seul l'agent concerné connait son mot de passe et aucun autre membre de sa hiérarchie. De manière étendue, elle est également utilisée pour notifier l'agent d'un changement effectué sur son profil utilisateur
Droit	integer	Tous les agents ne sont pas égaux devant l'application : certains peuvent effectuer des opérations refusées à d'autres. Pour délimiter les opérations accessibles par chaque agent on leurs attribuent un niveau d'accréditation sous la forme d'un entier
Identifiant	varchar(50 char)	Identifiant de l'agent nécessaire à son authentification. Cette identifiant doit être unique pour chaque agent
MotDePasseHache	varchar(MAX char)	Les mots de passes des agents sont salés puis hachés avant d'être sauvegardés dans la base de données
HashSalt	varchar(MAX char)	Représente le sel utilisé pour haché le mot de passe de l'agent
IdentifiantCourtSecurise	varchar(MAX char)	Dans l'établissement, il est très fréquent qu'une session ouverte soit utilisée par plusieurs agents (afin d'éviter de se déconnecter pour se connecter de nouveau avec de nouvelles informations de connexion). Pour pallier ce problème, une fois une session ouverte la manipulation d'informations sensibles (les informations concernant la sortie des élèves) devra être finalisée par l'envoi d'un code d'identification au serveur; en sachant que le code d'identification est unique pour chaque agent. Cela permet d'identifier un agent depuis n'importe-quelle session ouverte. De par sa faible taille le risque de collision des fonctions de hachage est trop grand pour que l'on puisse envisager de hacher les codes d'identifications. Ainsi, ils sont chiffrés avant d'être sauvegardés dans la base de données

Table autorisation de sortie :

Un agent peut décider de laisser sortir exceptionnellement un élève de l'établissement. Toutes les autorisations de sorties sont enregistrées dans cette table.
Cette table est composée des colonnes suivantes :

Nom de la colonne	Représentation informatique	Description
Id	integer	Clé primaire de la table
Type	integer	Dans le détail, il est possible de donner une autorisation de sortie soit à un élève soit à une classe toute entière. Si le destinataire de cette autorisation de sortie est susceptible de changer, la structure d'une autorisation de sortie reste la même dans les deux cas. Afin d'éviter d'avoir à créer deux tables ayant la même structure, on utilise cette variable pour indiquer si l'autorisation de sortie concerne une classe ou un élève
Id_table	integer	Représente la clé primaire de l'élève ou de la classe dans sa propre table
DateDebut	varchar(25 char)	Représente la date de commencement de l'autorisation de sortie
DateFin	varchar(25 char)	Représente la date de fin de l'autorisation de sortie
Jour	ineteger(25 char)	Représente le jour de la semaine pendant lequel l'autorisation de sortie est effective (par exemple l'autorisation est effective tous les lundis ou le mardi de telle heure à telle heure du... au..., etc.)
HeureDebut	varchar(25 char)	Représente l'heure de commencement de l'autorisation de sortie
HeureFin	varchar(25 char)	Représente l'heure de fin de l'autorisation de sortie
Id_agent	integer	Représente l'identifiant interne de l'agent ayant donné cette autorisation de sortie. Si un agent décide de l'enlever, la variable Supprimer change de valeur mais l'identifiant interne de l'agent reste celui de l'agent ayant donnée l'autorisation. Une fois l'entrée mise à jour, on enregistre dans une autre table (table registre) la modification effectuée, à savoir la suppression de l'autorisation de sortie, accompagnée de l'identifiant de l'agent ayant supprimé cette autorisation

Table blocage :

Un agent peut également décider d'interdire la sortie à un élève. La composition de cette table est la même que la table dédiée aux autorisations de sorties à la différence qu'une interdiction de sortie ne peux pas être programmée à l'avance car elle doit rester un acte ponctuel et exceptionnel. Ce faisant l'interdiction de sortie est effective le jour actuel uniquement. Autrement dit, la date de commencement et la date de fin de l'interdiction de sortie sont égales à la date actuelle.

Table registre :

Cette table enregistre toutes les modifications opérées sur la base de données
Cette table est composée des colonnes suivantes :

Nom de la colonne	Représentation informatique	Description
Modification	varchar(MAX char)	Description de la modification effectuée

Table des sorties journalières :

Jusqu'à présent les tables de la base de données été dédiées à une utilisation locale des ressources de l'établissement. A contrario, cette table a pour objectif de stocker des informations sur les élèves récupérées depuis une source de données externe (en ligne); puisque ces données n'ont d'existence qu'en ligne.

Cette source externe contient, notamment, les heures de sorties habituelles des élèves ainsi que les professeurs absents et les cours supprimés. L'application actualisera régulièrement cette table en effaçant l'intégralité des données avant de la peupler à nouveau. Elle contiendra alors les créneaux horaires pendant lesquels un élève est autorisé à sortir.
Cette table est composée des colonnes suivantes :

Nom de la colonne	Représentation informatique	Description
HeureDebut	varchar(25 char)	Heure de commencement où l'élève est autorisé à sortir
HeureFin	varchar(25 char)	Heure de fin où l'élève est autorisée à sortir
VerificationRequise	integer	Même si les emplois du temps des élèves spécifient qu'ils peuvent sortir, dans certains cas il faut qu'un agent vérifie s'ils peuvent sortir (par exemple s'il faut demander la permission des parents avant de laisser sortir l'élève). Cette variable indique si une vérification manuelle est nécessaire avant de laisser sortir l'élève ou non

1.5 Utilisation de la base de données

1.5.1 Maintenance

Au bout d'un certain temps les entrées inactives (comme une autorisation de sortie dont la date de fin est dépassée depuis longtemps ou alors une entrée dont la date d'édition est dépassée) sont définitivement supprimées de la base de données. Pour se faire à chaque démarrage de l'application, le logiciel vérifie le jour durant lequel il a nettoyé la base de données des entrées inactives. Si ce jour est assez loin dans le temps alors il recommencer le nettoyage de la base de données.

La table des élèves est mises à jour à l'aide d'un fichier fourni par l'établissement que nous nommerons fichier d'états des étudiants. Ce fichier comporte le nom, prénom, classe, sexe et les jours de demi-pension de chaque élève de l'établissement. Un élève est identifié par son nom, prénom, classe et sexe. Si un élève existe déjà dans la base de données alors seuls les jours de demi-pension sont mis à jour, sinon une nouvelle entrée est créée dans la base de données.

1.5.2 Manipulation du temps

L'application doit gérer des déplacements humains. Autrement dit, les horaires sauvegardés ne correspondront pas exactement aux horaires réels des élèves. Par exemple, un élève peut sortir quelques minutes avant ou après l'heure officielle. Pour pallier ce problème, ce sont les heures officielles qui sont sauvegardées mais elles ne sont pas utilisées telles quelles : on leurs soustrait une petite quantité avant de les comparer à l'heure actuelle. Ce fonctionnement est illustrée figure 1.5.2

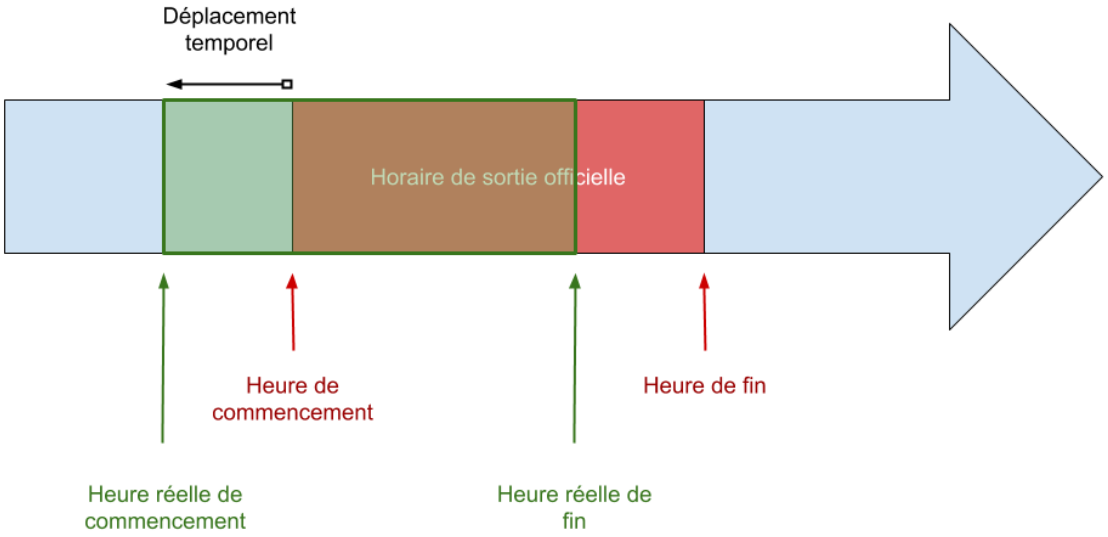


Figure 1.2: Principe du déplacement temporel effectué pour comparer les heures

1.6 Protocole de communication

L'application est construite comme un serveur local permettant l'échange de données avec tout autre appareil connecté au même réseau que le serveur. Lorsqu'un ordinateur tente d'établir une connexion avec le serveur, le serveur envoie en retour les pages HTML à afficher, les scripts à charger, les images à afficher, etc.

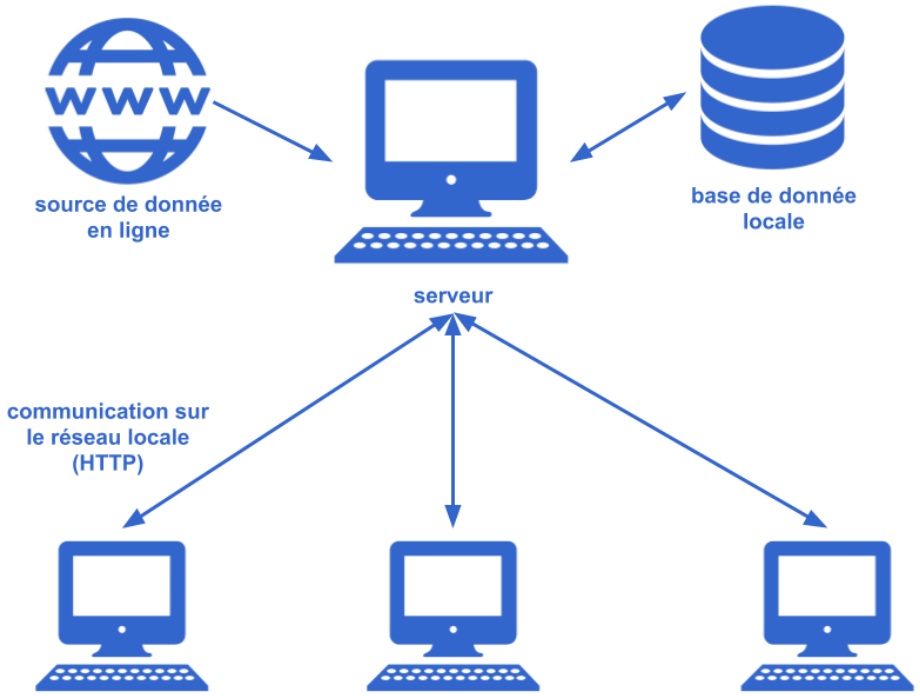


Figure 1.3: Fonctionnement de la communication avec l'application

Cette construction permet de garder le contrôle sur la base de données en limitant l'accès direct

à cette dernière : dans ce cas, seul le serveur est capable d'accéder et de modifier la base de données. Tous les autres appareils ne font qu'envoyer une requête pour effectuer certaines opérations. Ces requêtes ne comportent pas en elles-mêmes l'instruction permettant de modifier la base de données mais l'action que le serveur doit effectuer. Par exemple, si le poste client demande au serveur d'ajouter une entrée à la base de données, il enverra une requête ayant pour unique contenu son objet "ajouter une entrée à la base de données". Ensuite le serveur exécute l'instruction programmé dans le cas où il reçoit une requête ayant cet objet.
Cette configuration permet, en autre, de limiter les demandes non autorisées : seules les requêtes ayant été au préalable enregistrées peuvent être exécutées.

Puisque la communication entre le serveur est les clients se fait au sein du réseau de l'établissement la sécurité est minimale : les échanges ne sont pas chiffrés comme ce serait le cas avec un protocole de communication HTTPS. En particulier les informations d'authentification des agents sont envoyées en texte clair. En conséquence, une utilisation du serveur sur un réseau wifi est à proscrire : pirater un réseau filaire est compliqué car il est nécessaire de dérouter le flux réseau en provenance des ordinateurs avant de les renvoyer au serveur. Ce qui n'est pas le cas d'un réseau sans fil où une fois le chiffrement enlevé n'importe quel individu peut analyser le trafic.

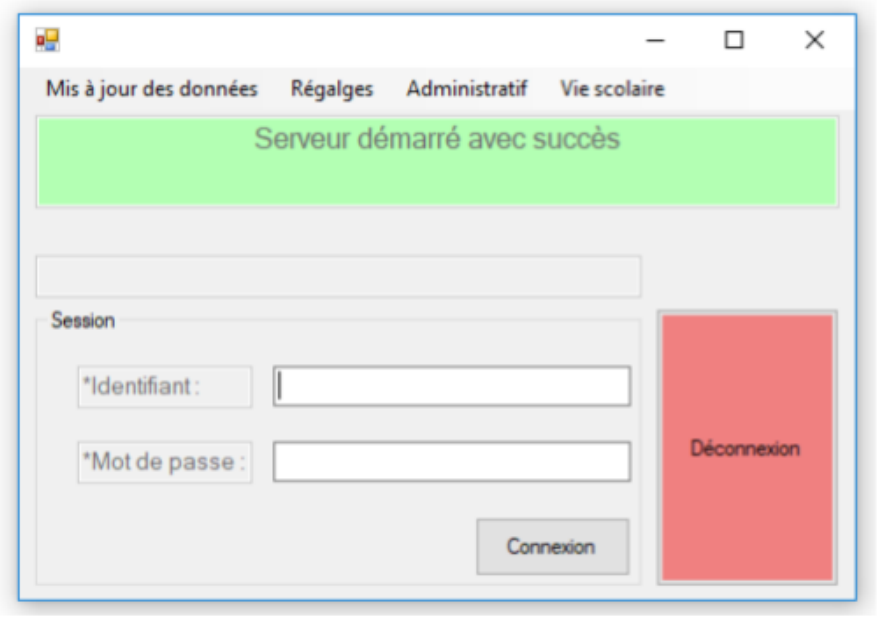
Partie III

Fonctionnement de l'application

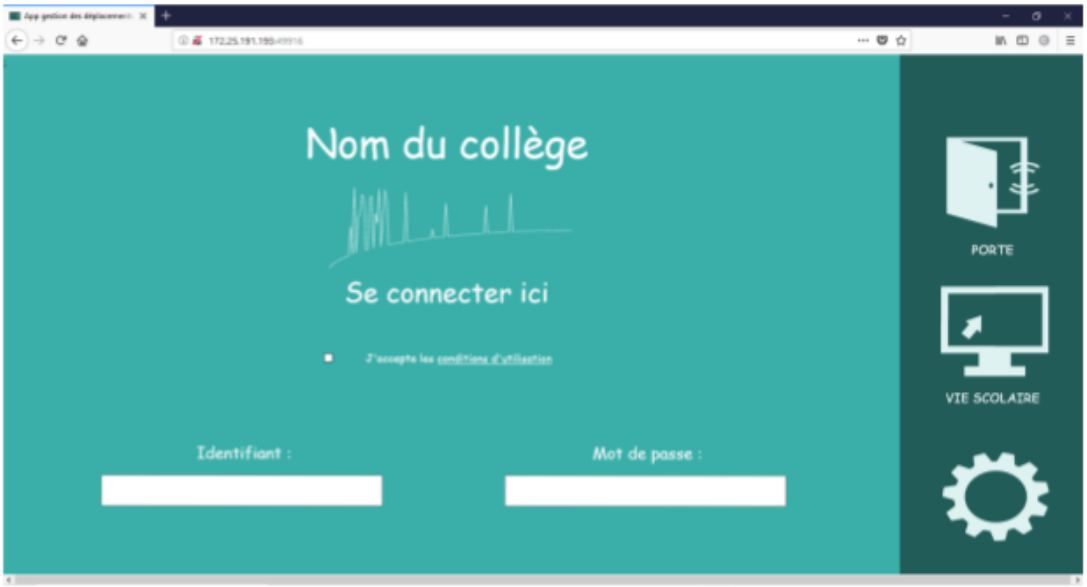
1.7 Structure de l'application

L'application se divise en deux parties comme l'illustre la figure 1.7 :

- l'interface serveur uniquement accessible depuis la salle du serveur
- l'interface client accessible par tout le monde connecté au réseau filaire local



INTERFACE SERVEUR



INTERFACE CLIENT

Figure 1.4: Capture d'écran des deux interfaces

1.8 Fonctionnement de l'interface serveur

1.8.1 Niveaux d'accréditations

Le menu de l'interface serveur, représenté figure 1.8.1, n'est pas accessible en totalité selon le niveau d'accréditation de l'utilisateur.
Les niveaux d'accréditations sont représentés par une liste [accreditation_niveau_0, accreditation_niveau_1, ...].
Le niveau d'accréditation le plus important est "accreditation_niveau_0" représentant l'indice 0 du tableau des accréditations. Ainsi, pour déterminer si un utilisateur à le droit d'accéder à un paramètre ou non, il suffit de comparer la valeur entière de son niveau d'accréditation agent_right et la valeur entière de l'accréditation requise needed_right. Dans le cas où $\begin{cases} \text{agent\_right} \geq -1 \\ \text{agent\_right} \leq \text{needed\_right} \end{cases}$ alors l'agent à le niveau d'accréditation nécessaire pour accéder au paramètre demandé.

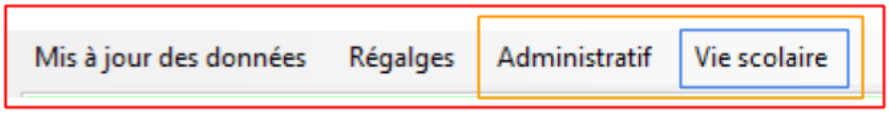


Figure 1.5: Menu de l'interface serveur

1.8.2 Gestion des sessions

Il existe deux types de sessions :

- la session physique permettant la connexion à l'interface serveur
- les sessions en lignes permettant l'envoi d'informations au serveur

Session physique

L'ouverture d'une session dans le cas physique et en ligne reste le même : l'application récupère l'identifiant et le mot de passe fourni par l'utilisateur. Ensuite, il cherche une correspondance avec l'identifiant fourni dans la table des agents. Puis, si la correspondance existe, il compare le mot de passe (une fois haché avec le sel sauvegardé) fourni et le mot de passe haché enregistré. Si les deux correspondent alors une session est ouverte. Ce qui signifie que toutes les opérations effectuées avec cette session seront sauvegardées avec l'identifiant de l'agent récupéré depuis la base de données.

Session en ligne

Les sessions en lignes fonctionnent légèrement différemment. En effet, dès qu'une connexion est établie avec le serveur, on enregistre l'adresse IP de l'appareil que l'on stocke dans une structure, ensuite enregistrée dans une liste statique, que nous nommerons liste clients. Remarquons qu'avant d'enregistrer l'adresse IP dans une nouvelle structure, on vérifie que cette adresse IP n'est pas déjà présente dans la liste clients.
Si la requête du client a pour objet la création d'une session, alors on récupère l'identifiant et le mot de passe (en clair) fourni dans la requête web. Puis, comme pour une session physique, on cherche une correspondance. Dès lors deux cas se présentent :

- il n'y a aucune correspondance (les informations d'authentification fournies sont incorrectes). Dans ce cas, on met à jour la structure contenant les informations du client. Au bout de trois tentatives consécutives de connexion (donc ayant échoué) la structure du client correspondant est mise à jour afin de spécifier que l'ordinateur du client n'a plus le droit de se connecter au serveur pendant un certain temps. Le client est alors temporairement bannit.
  Dans les faits, lorsque ce client tentera de se connecter, l'application vérifiera si le client est présent dans la liste clients et si oui, l'application vérifiera s'il est temporairement bannit. Si c'est le cas, alors l'application met immédiatement un terme à la communication avec le client.

- il y a une correspondance dans la base de données. Dans ce cas, on génère une suite de caractères alphanumériques qui sera envoyée comme cookie au client. Cette suite de caractères est ensuite sauvegardée dans le profil du client (dans la structure présente dans la liste clients). Lorsque ce même client enverra une requête quelconque l'application vérifiera que le cookie envoyé par le client est le même que celui stocké pour l'adresse IP de actuelle. Si c'est le cas, alors le client est reconnu par le serveur comme déjà authentifié. Sinon le client devra s'authentifier auprès du serveur. Ce fonctionnement est illustré figure 1.8.2.

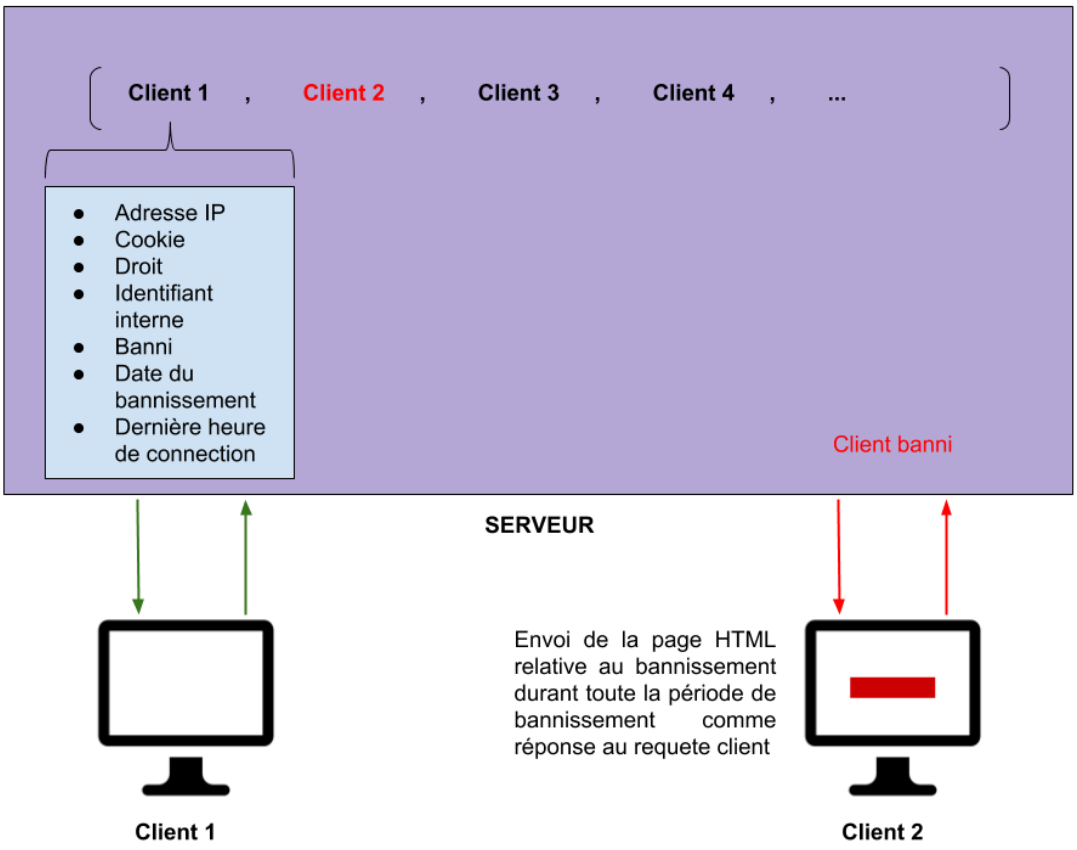


Figure 1.6: Fonctionnement des sessions en lignes

Ces structures ne sont pas supprimées de la liste client une fois que le temps d'inactivité d'une session ou le temps de bannissement d'une session est dépassé. Pour que ces informations soient supprimées de la liste clients il faut attendre qu'un client utilise le même poste informatisé pour envoyer une requête quelconque au serveur. Dès lors, l'application se rendra compte que le temps est dépassé et supprimera la structure correspondante pour la remplacer par une nouvelle.

1.8.3 Utilisation des mots de passes non protégés

A de nombreuses occasions, il devient obligatoire de manipuler les mots de passes non chiffrés (soit pour se connecter à un service externe, soit pour récupérer le mot de passe, non chiffré envoyé par un client). Pour ce faire, nous réservons un espace de la mémoire RAM pour stocker le mot de passe en clair. Une fois les opérations requises effectuées sur ce mot de passe, nous nettoyons l'espace mémoire en le remplaçant par du bruit.
Les comparaisons de mots de passes en mémoire RAM sont effectuées bit à bit à l'aide de pointer (pointant sur la case mémoire correspondant).

1.8.4 Gestion des erreurs

Toutes les erreurs n'entrainent pas forcément l'arrêt du processus en cours. C'est le cas pour certaines erreurs critiques ne devant pas être ignorées mais d'autres sont simplement traités en interne sans en notifier les utilisateurs.
Pour ce faire, les erreurs sont centralisées par un même processus : lorsqu'une erreur apparait dans tout le processus de l'application, on récupère les informations de relative à cette erreur que l'on enregistre ensuite dans un fichier texte dédié.

Lorsque la base de données n'est pas utilisée depuis un certain temps, quand une nouvelle requête est faite, la base de données doit être réactivée. Le temps de réactiver la base de données, le délai maximal autorisé pour une requête est dépassé, produisant alors une erreur. Si ce type d'erreur survient, on relance la requête. On recommence un certain nombre de fois. Au final, soit la requête réussi, soit le nombre maximal de tentatives pour une même requête est dépassé et dans ce cas l'application revoie une erreur.

1.8.5 Incorporation des lecteurs RFID

La solution technique retenue afin d'attribuer un identifiant à chaque élève est la technologie RFID. Il existe deux principaux types de lecteurs RFID :

- les lecteurs reliés en réseau
- les lecteurs connectés en USB

Les lecteurs reliés en réseau posent de nombreux problèmes. D'une part, le lecteur n'est jamais utilisé seul : il est couplé à l'interface client. Dès lors, soit le lecteur est relié en réseau directement au serveur. Ce faisait, il faudra faire la correspondance entre le lecteur et l'ordinateur utilisé pour recevoir les informations en retour. Soit, le lecteur est branché sur la prise réseau de l'ordinateur. Dès lors, l'ordinateur ne peut plus se connecter au serveur; sa prise réseau étant déjà utilisée. Toutefois, il serait possible d'utiliser une autre prise réseau sur l'ordinateur mais cela impliquera de devoir paramétrer le lecteur à chaque fois pour spécifier à l'interface client quel port de communication utilise le lecteur. En conclusion, ce type de lecteur est trop contraignant pour l'usage souhaité.

Intervient alors les lecteurs RFID reliés en USB. Ces lecteurs interagissent avec l'ordinateur comme une interface humaine, ou plus précisément comme un clavier USB. Ce qui a l'avantage indéniable de pouvoir être échangé à volonté sans avoir un quelconque paramétrage à faire. Néanmoins, cela amène une contrainte : l'identifiant RFID reçu par l'interface client dépend de l'état de la touche majuscule du clavier régulier. En effet, Le code lu et envoyé par le lecteur demeure toujours le même. La possible différence vient de l'interprétation qu'en fait l'ordinateur. Les informations envoyées par le lecteur USB correspondent à des chiffres codés par "D0" pour 0, "D1" pour 1, ... Lorsque la touche majuscule du clavier régulier est activée les codes ASCII envoyés par le lecteur RFID sont interprétés comme des entrées numériques. Lorsqu'elle ne l'est pas, les entrées sont interprétées comme des caractères. Par exemple, sur la plupart des claviers ne disposant pas de pavé numérique la touche correspondant au chiffre 1 et aussi celle du caractère '&'.
Pour pallier ce problème il a fallu intercepter le code ASCII de toutes les zones textes susceptibles de recevoir des identifiants RFID. Une fois interceptés, il suffit d'interpréter la valeur reçue comme un chiffre et d'afficher le résultat dans la zone correspondante.

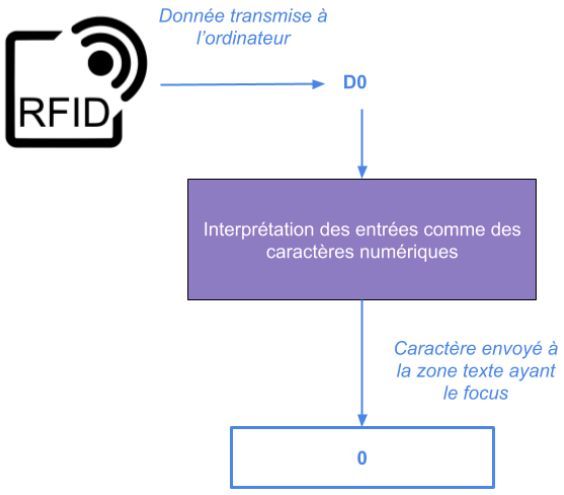


Figure 1.7: Gestion des zones textes recevant des entrées en provenance d'un lecteur RFID

1.8.6 Fonctionnement des prises de décision

Lorsqu'un élève demande à sortir de l'établissement le logiciel procède comme suit pour savoir s'il en a l'autorisation ou non :

- Si une interdiction de sortie est présente pour cet élève, il n'est pas autorisé à sortir
- Sinon si une autorisation de sortie est présente pour cet élève, il est autorisé à sortir sans vérification
- Sinon s'il s'agit d'un créneau régulier présent sur l'emploi du temps de l'élève, il est autorisé à sortir sans vérification. Toutefois, s'il s'agit d'un créneau récemment modifié dans l'emploi du temps de l'élève, il est autorisé à sortir après vérification.
- Sinon s'il est externe le jour correspondant et que l'horaire est celle de la pause méridienne, il est autorisé à sortir
- Sinon, il n'est pas autorisé à sortir

1.8.7 Interdiction de sortie automatique

A compté de 7 jours depuis la date actuelle si une absence non clôturée ou une punition non clôturée est présente sur la fiche de l'élève (et ce depuis le début de l'année) alors l'ordinateur met en place une interdiction de sortie pour cette élève.
Dans la base de données, il est enregistré un agent inaccessible par tout autre utilisateur et ne pouvant pas se connecter à l'application. Cet agent est celui utilisé par l'ordinateur lorsque qu'il prend la décision de mettre une interdiction de sortie.

1.9 Fonctionnement de l'interface client

1.9.1 Structure principale

La structure principale de l'interface client est illustrée figure 1.9.1. La page est divisée en 4 sections (Zone 1, 2,3 et Zone 4). Lorsqu'un des boutons du menu principal (Zone 3) est activé, le script principal charge le contenu des zones 1 et 2 et seulement ensuite il charge les scripts correspondants à ces zones.
Chaque script est enveloppé dans une structure dont seulement certaines fonctions sont accessibles depuis l'extérieur du scope de la structure. Et ce afin d'éviter les collisions entre plusieurs variables ayant le même nom dans différents scripts.

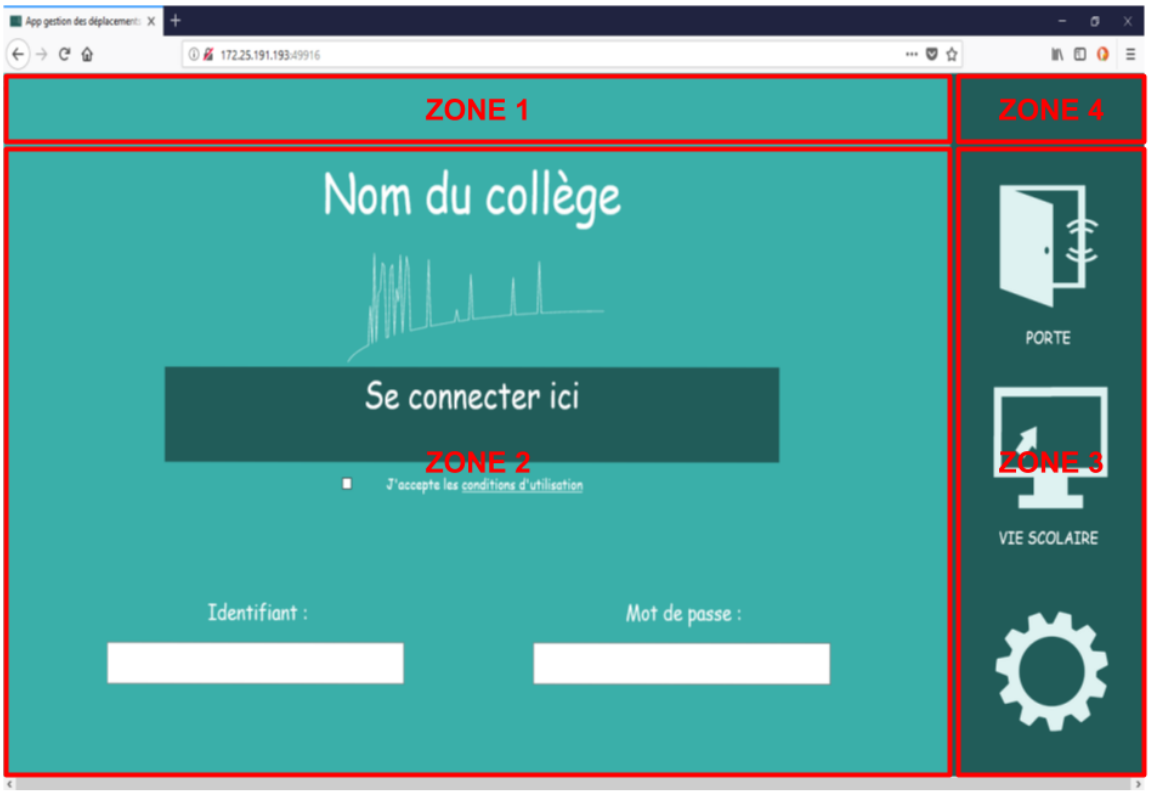


Figure 1.8: Structure principale de l'interface client

Une fois chargé, les scripts s'initialisent seuls. Il est donc indispensable que le contenu sur-lequel le script interagit existe a préalable. C'est la raison pour laquelle on attend que le contenu HTML soit chargé pour ensuite charger les scripts.
On aurait également pu charger les deux objets (le contenu HTML et le script correspondant) en même temps et initialiser le script seulement lorsque le contenu HTML a été chargé.

1.9.2 Structure vie scolaire

La structure dédiée à la vie scolaire sépare la Zone 2 en deux sous zones : les zones 2.1 et 2.2. Le contenu de la Zone 2.1 est chargé une seule fois au lancement de la section vie scolaire. Cette section permet de voir le profil de l'élève sélectionné.
La Zone 1 devient quant à elle un menu secondaire permettant de naviguer entre les options de la section vie scolaire. Lorsqu'un bouton de ce menu est activé, le contenu de la Zone 2.2 change. Une fois chargé, le script correspondant est chargé à son tour.

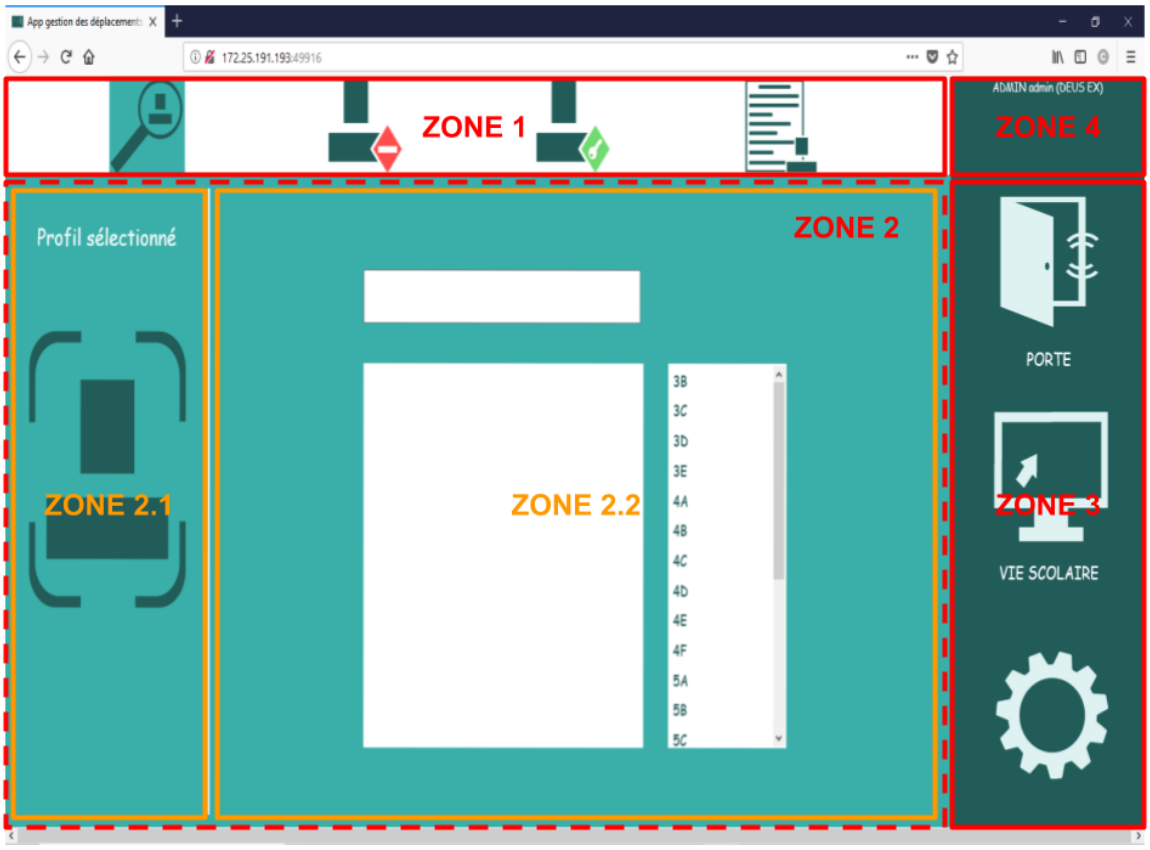


Figure 1.9: Structure relative à la vie scolaire

La section vie scolaire comporte également un objet crée dynamiquement : un tableau permettant d'ajouter ou du supprimer des autorisations de sortie. Ce tableau est créé dynamiquement car il doit accueillir des cellules pouvant être hors des cadres prédéfinis comme le montre la figure 1.9.2. Dans les faits, les contours du tableau sont de simples traits et les cellules sont des éléments indépendants positionnés sur le corps du tableau.

Figure 1.10: Création dynamique d'un tableau

La section vie scolaire concerne des données sensibles (des données pouvant interdire ou autoriser les élèves à sortir). Ce faisait, au premier chargement de l'interface client, un nouvel élément est créé et définit comme inactif (invisible pour l'utilisateur). Lorsque l'utilisateur décide d'envoyer ces données sensibles, cet élément devient actif, permettant alors à l'utilisateur d'entrer son code d'identification; comme l'illustre la figure 1.9.2.
Dans les faits, toutes les données concernant les sorties des élèves sont formatées par les scripts correspondant aux sections interdiction de sortie et autorisation de sortie. Lorsque l'utilisateur valide les modifications, ces données formatées sont envoyées au script en charge de la vérification du code d'identification et stockées temporairement. Une fois le code d'identification saisi, il est formaté et ajouté aux autres données temporairement sauvegardées. Ces données concaténées sont ensuite envoyées au serveur. C'est alors le script en charge de la vérification du code d'identification qui reçoit la réponse du serveur et la transmet au sous script correspondant.

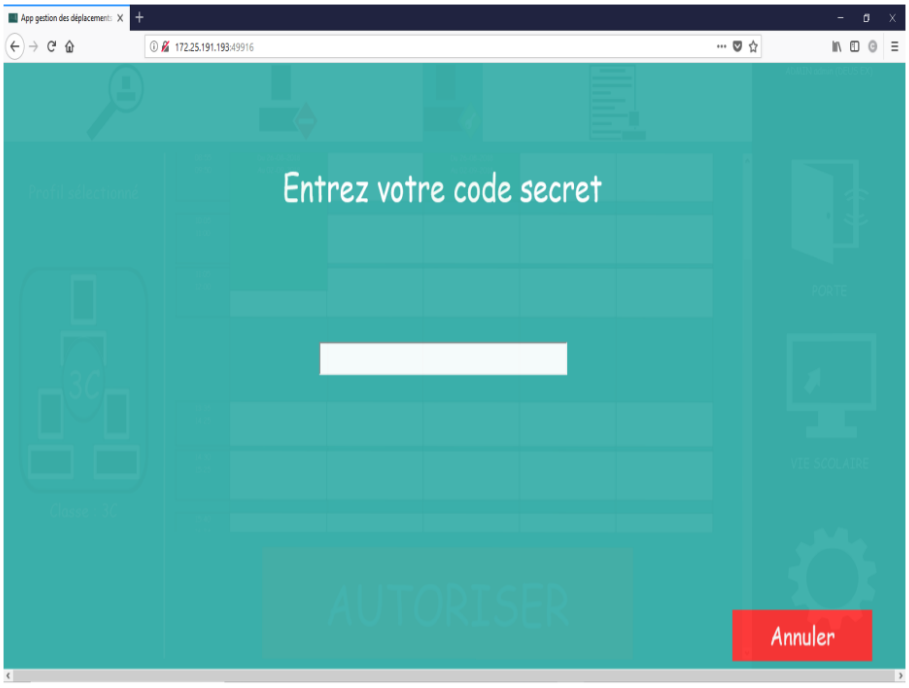


Figure 1.11: Validation du code d'identification dans l'interface client

1.9.3 Structure paramètres

La section paramètres transforme uniquement la Zone 1 en menu et conserve la Zone 2 entière.

1.9.4 Structure porte (sortie de l'établissement)

La section porte utilise la Zone 1 comme bandeau permettant d'afficher le résultat de la sortie de l'élève (si l'élève peut sortir ou non et si oui sous quelles conditions)

Partie IV

Fonctionnalités

1.10 Vie scolaire

1.10.1 Sortie non autorisée

Cet onglet permet de mettre à jour le son joué lorsqu'un élève tente de sortir alors qu'il n'en a pas l'autorisation.

Ce fichier peut également être envoyé au serveur depuis l'interface client.

1.11 Administration

1.11.1 Information sur l'établissement

Les informations enregistré au travers de cet onglet est ensuite envoyé au client lorsqu'il tente d'accéder à la section Autorisation de sortie. Les heures et les jours de cours sont utilisées pour créer un dynamiquement un tableau permettant de mettre en place des autorisations de sortie.
L'horaire de la pause méridienne n'est pas enregistré comme une heure de cours : elle permet de savoir sur quels créneaux horaires les élèves externes sont autorisés à sortir.

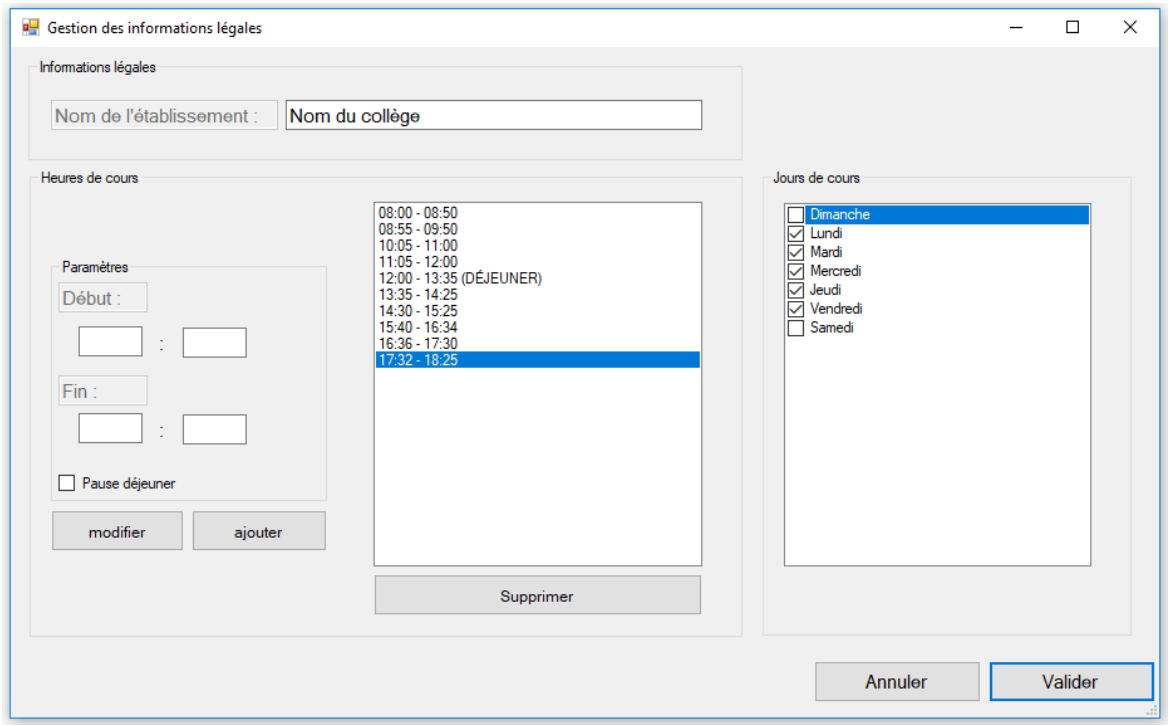


Figure 1.12:

1.11.2 Gestion des données élèves

Cet onglet permet de recherche les informations d'un élève au travers de son nom, son prénom, sa classe ou son identifiant RFID. Des lors, il est possible de modifier l'identifiant RFID associé à cet élève.
Dans le cas de redondance dans la base de données, il est possible de fusionner les informations de plusieurs étudiants. Cette fusion récupère l'identifiant RFID le plus récent, la photo de l'élève la plus récente et réunie ces informations dans le profil édité récemment. Il est à noter que les autorisations ou les interdictions de sortie ne sont pas transférées d'un élève à un autre pour des raisons de responsabilité.

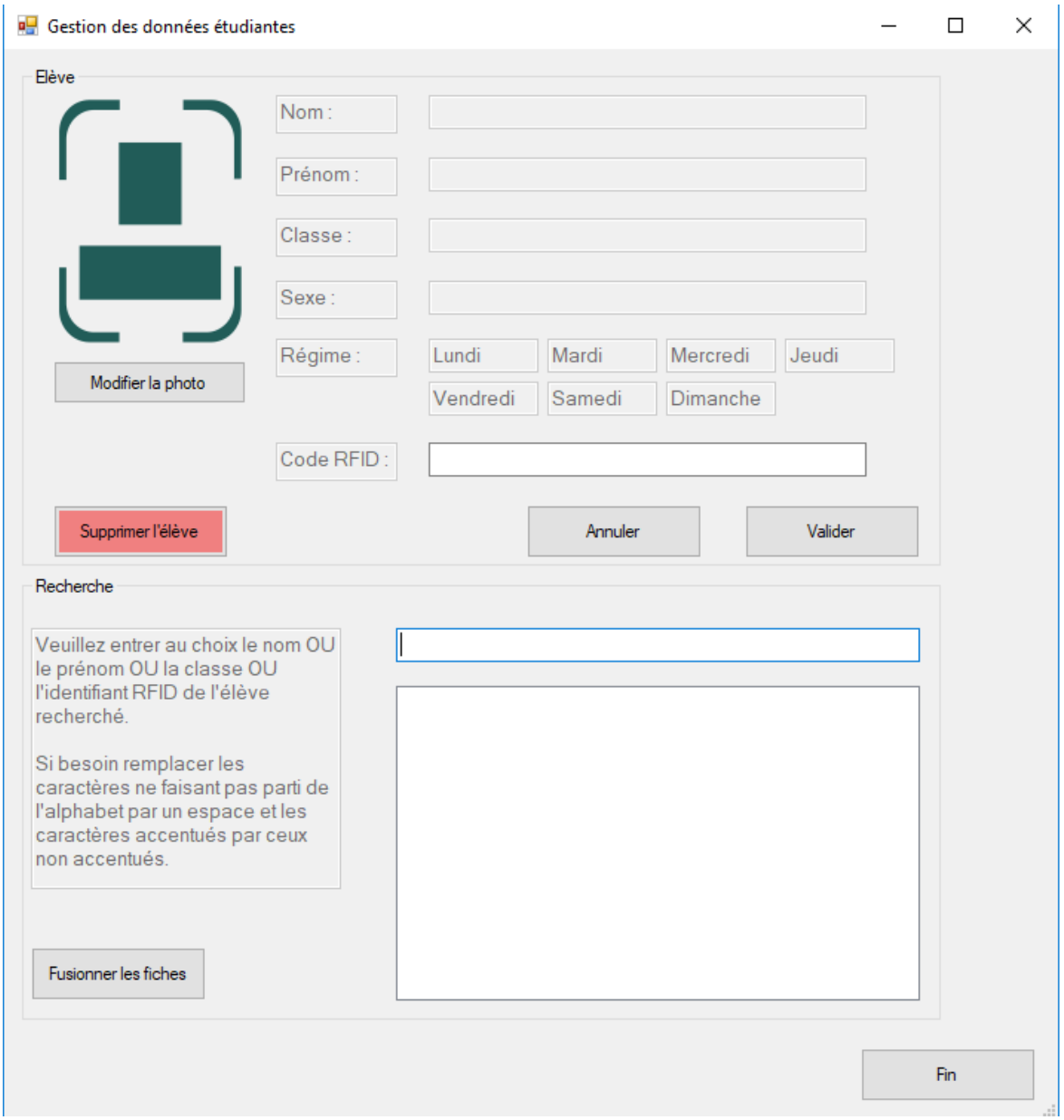


Figure 1.13:

1.11.3 Fichier d'états des étudiants

Afin de pouvoir lire le fichier des étudiants fourni selon le format CSV, il faut indiquer au logiciel à quoi correspond chaque colonne du fichier. Par exemple, la colonne 1 représente les noms des élèves. Lorsqu'un nouveau fichier est téléchargé, la base de données est immédiatement mise à jour avec ces nouvelles informations.

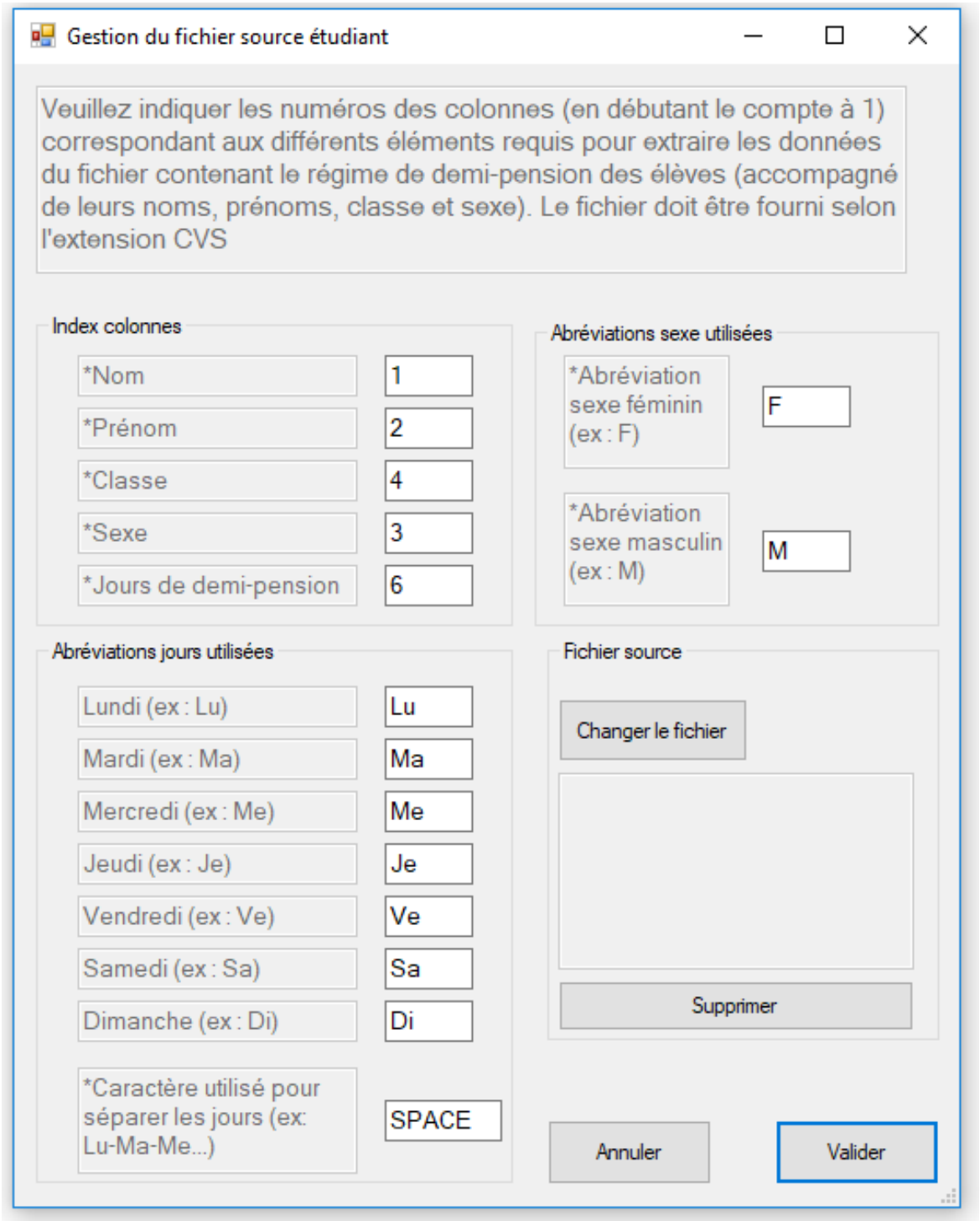


Figure 1.14:

Ce fichier peut également être envoyé au serveur depuis l'interface client.

1.11.4 Durée de la pause autorisant les sorties

Au bout de plusieurs heures consécutives durant lesquelles un élève n'a pas cours, il est, dans certain cas autorisé, à quitter l'établissement. Ce faisait lorsque l'application analysera l'emploi du temps de l'élève il faudra lui indiquer quelle est la durée minimale après laquelle un élève est autorisé à sortir.

1.11.5 Longueur de l'identifiant RFID

Tous les lecteurs RFID de l'établissement doivent lire le même nombre de caractères. Dès lors, il est facile de préciser ce nombre et ce afin de permettre à l'interface utilisateur de savoir quand est-ce qu'un identifiant RFID est complet. En effet, lorsqu'un élève se présente à la sortie de l'établissement, le lecteur RFID lit son identifiant et l'envoi à l'interface. Pour éviter à un agent de devoir valider l'identifiant RFID manuellement (en appuyant sur un bouton par exemple), l'interface compte le nombre de caractères reçus et une fois le compte arrivé à la longueur des identifiants RFID, l'interface envoi le résultat au serveur.

1.12 Réglages

1.12.1 Adresse IP du serveur

Permet de spécifier l'adresse IP du serveur ainsi que son port de communication

1.12.2 Fichier d'accès au serveur

Comme l'adresse IP du serveur (notamment le port de communication du serveur) est susceptible de changer au cours du temps, afin de faciliter l'accès à l'interface client, il est possible de préciser un chemin d'accès vers un dossier commun (accessible par tous les membres de l'établissement) dans lequel sera mis à jour un fichier HTML contenant un lien d'accès vers le serveur. Lorsque l'adresse IP du serveur change, ce fichier est mis à jour. L'utilité de ce fichier est représenté sur la figure 1.12.2.

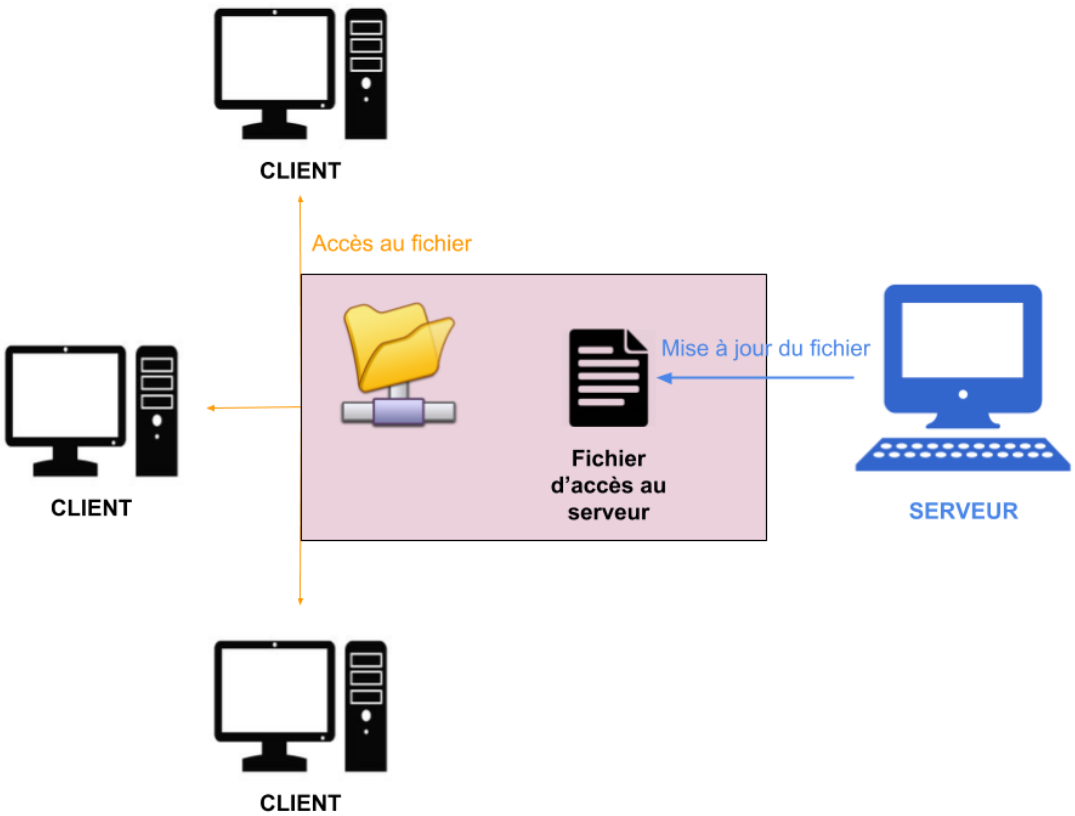


Figure 1.15: Principe d'utilisation du fichier d'accès au serveur

1.12.3 Gestion des adresses mail

Lorsqu'une erreur critique survient au sein du serveur, les personnes ayant renseigné leurs adresses e-mail dans cet onglet en seront notifiées.

1.12.4 Gestion du service d'envoi du courriel

Le serveur ne gère pas sa propre boite de réception en ligne, il utilise un service de courriel déjà existant. Ainsi, cette section permet de configurer l'accès à ce service.

1.12.5 Données de connexion au site web

Permet de préciser les paramètres d'accès à la source de données externe.

1.12.6 Gestion des droits d'accès

Permet la création de compte agent.

1.13 Mise à jour des données

1.13.1 Rafraichir la base de données

Permet de forcer la mise à jour de la base de données à partir du fichier d'états des étudiants fourni.

1.13.2 Purger la base de données

Supprime définitivement toutes les entrées des tables de la base de données (mise à part celles de la table relative aux informations des agents).

1.13.3 Redémarrer le serveur

Indique à Windows d'ouvrir un nouveau processus et ferme le processus actuel.