

1. Présentation du logiciel.....	2
2. Contexte.....	2
2. Architecture Matérielle et Logicielle.....	3
3. Modules Fonctionnels.....	4
1. Gestion des paiements.....	4
2. Signature cryptographique HMAC SHA256 chaînée.....	5
3. Archivage automatisé.....	5
4. Exportation des données pour l'administration fiscale.....	6
4. Modèles de données (modèles conceptuels).....	6
4.1 Commandes.....	6
4.2 Articles.....	8
4.3 Paiements.....	9
4.4 FermeturesAuto.....	10
4.5 Tracabilite.....	11
5. Modalités de stockages des données.....	11
6. Identification du système.....	12
Conformité au Référentiel de Certification v1.7.....	13

1. Présentation du logiciel

Nom du logiciel : caisse.enregistreuse.fr

Système de caisse en ligne pour assujettis à la TVA, conforme à l'article 88 de la loi de finances 2016 et au BOI-TVA-DECLA-30-10-30-20210519.

Caisse.enregistreuse.fr est une solution SaaS destinée aux professionnels enregistrant des paiements de clients particuliers. Accessible via <https://caisse.enregistreuse.fr>, elle garantit l'inaltérabilité, la sécurisation, la conservation et l'archivage des données d'encaissement.

2. Contexte

Garantir la conformité NF525 au cours des mises à jour et évolutions du logiciel, répondre aux obligations des éditeurs de logiciels de caisse conformément à la loi BOI-TVA-DECLA-30-10-30-20210519

- Enjeux internes à l'entreprise (ressources, processus, fournisseurs, sous-traitants)

Les enjeux internes font référence aux enjeux vis-à-vis de la gestion des ressources, qu'elles soient humaines ou matérielles, le processus de fabrication, mise à jour du logiciel en lui-même, de la maîtrise des fournisseurs ainsi que celle des sous-traitants.

Enjeu de gestion des ressources : De s'assurer en permanence de garantir la confidentialité des données enregistrées dans la base de données ainsi que la limitation des droits en lecture et en écriture en fonction de la confiance en l'individu. Il conviendra de restreindre au maximum à un minimum de personnes et uniquement au strict nécessaire des autorisations configurables pour ce qui est des accès aux sources de données du logiciel (base de données, fichier de sauvegarde).

Mais il faudra également s'assurer de garantir la sécurité des données vis-à-vis des risques humains. Il s'agira par exemple de limiter à uniquement certaines personnes sous réserve d'acceptation d'une clause de confidentialité l'accès aux données les plus sensibles.

Doit rester en permanence à l'esprit lors de la conception de nouvelles versions ou de mise à jour du logiciel.

Il sera également indispensable d'inclure cette réflexion lors de la mise en place d'un nouveau fournisseur ou d'un nouveau sous-traitant.

- Enjeux externes à l'entreprise (réglementaires ou économiques par exemple)

Conception Générale

La conformité de l'application dans son sens le plus strict et l'une des réflexions les plus importantes du logiciel.

La majeure partie de la clientèle de ce logiciel exige que ce logiciel se conforme en tout point aux réglementations en vigueur. La non-conformité du logiciel engendrerait un risque important de perte de clientèle, et également un risque encore plus grave vis-à-vis de la législation et de la responsabilité de l'entreprise mais représenterait de surcroît un manquement vis-à-vis de la confiance que l'utilisateur a en ce logiciel.

Il est indispensable pour ce logiciel de se conformer à la réglementation en vigueur autant vis-à-vis de ses obligations légales que vis-à-vis de ses obligations morales.

- Risques

Les risques encourus par la société à ne pas se conformer à la certification sont des risques majeurs. Le risque juridique incluant des peines de prison ainsi que des peines financières, le risque étant également la perte de la réputation du logiciel, mais aussi tout simplement également un risque important vis-à-vis de la finance de celui-ci car la non-conformité entraînerait très rapidement une très forte baisse du nombre de clients.

2. Architecture Matérielle et Logicielle

- **Serveurs** : Linux Ubuntu 22.04 (Digital Ocean, Europe)
- **Langages** : PHP 8.2
- **Base de données** : MySQL 8
- **Versoinnage** : SVN - svnserve, version 1.13.0 (r1867053)
- **Accès serveur** : Restreint aux administrateurs Net-Assembly

Versions de l'application : Il existe des versions de l'application destinées aux tablettes et smartphones Android, iOS, Windows Mobile, appareils SunMi, appareils Star Micronics, etc. Cependant chacune de ces applications intègre un conteneur web permettant d'afficher le contenu d'une page distante située à l'adresse caisse.enregistreuse.fr

Les seules fonctionnalités spécifiquement intégrées dans les applications Android ou iOS par exemple, sont les fonctionnalités indispensables et propres à l'architecture matérielle de l'appareil, comme par exemple des fonctionnalités de connexion aux imprimantes, de gestion des balances, connexion au lecteur de code barre, au tiroir-caisse, au terminaux de paiement, et qui sont appelées par le conteneur web via un connecteur.

Conception Générale

L'intégralité des fonctions de facturation, d'enregistrement de commande, de rapport, de connexion utilisateurs, de création de compte sont toutes réalisées par la même version de l'application disponible en ligne et éventuellement affichée dans un container, ce qui permet de ne maintenir qu'une seule version de l'application et de garantir que toutes les fonctionnalités soit bien identiquement implémentée sur toutes les plateformes où l'application est disponible (pas de nécessité de mise à jour sur l'appareil du client, garantie d'une seule version de l'application utilisée par l'intégralité des utilisateurs).

Une seule version de l'application est donc utilisée par l'intégralité des utilisateurs de celle-ci, et qui ne peut être autre que la dernière version, disponible en ligne, commune à tous les utilisateurs.

Ce mécanisme permet l'identification et la traçabilité de la distribution.

Le logiciel fonctionne dans le cloud uniquement, c'est à dire qu'il n'est pas accessible sans connexion internet.

3. Modules Fonctionnels

1. Gestion des paiements

Le logiciel permet d'enregistrer des commandes.

Il permet la gestion des ventes et des remboursements mais ne permet pas la gestion des achats puisqu'il est destiné à enregistrer des ventes à la manière d'une caisse enregistreuse.

Une commande peut être soit à l'état de devis soit à l'état de facture.

À l'état de devis une commande n'a pas de numéro de facture il n'est pas possible d'éditer un document sous forme de facture (uniquement des devis).

Une commande peut passer à l'état de facture par le biais d'un processus appelé "validation". La validation d'une commande permet de lui attribuer un numéro de commande qui sera un numéro séquentiel unique et sans trou, en même temps un saut cryptographique sera associé à la commande ainsi qu'une date de valeur définitive.

C'est uniquement au moment de la validation de la commande qu'il est possible d'éditer un ticket justificatif avec TVA ou d'une facture.

Chaque commande peut être soit une vente soit un remboursement.

La vente viendra augmenter le chiffre d'affaires de manière positive et le remboursement de manière négative.

Il est uniquement possible de modifier une commande lorsqu'elle se trouve en l'état de devis ; dès l'instant où un identifiant interne de facture est attribué à la commande, il

Conception Générale

devient impossible de modifier ni les articles qui composent cette commande ni le prix de la commande ou la date de valeur de la commande, ou le compte client affecté.

Il devient également impossible de supprimer cette commande : aucune fonctionnalité ne permettant la suppression d'une commande validée ne pourra être implémentée.

2. Signature cryptographique HMAC SHA256 chaînée

Quelque soit la manière de valider la commande une seule et unique fonction permet le passage de l'état de devis à l'état de facture dans le code source du logiciel.

Cette fonction effectue les opérations suivantes :

- attribution d'un numéro séquentiel unique et sans trou pour la facture (la séquence sera propre à la boutique et chaque boutique disposera de sa propre séquence unique et son trou)
- attribution définitive de la date de valeur comptable de la commande
- enregistrement en base de données de l'état "Validé"
- calcul d'un seul cryptographique HMAC SHA256 basé sur les informations de la commande (prix, TVA, etc), son identifiant interne, l'identifiant de la boutique, et le précédent so cryptographique enregistré pour la commande précédente portant le numéro de commande inférieur (cf dossier d'architecture technique)

3. Archivage automatisé

Le processus d'archivage automatisé dans ce logiciel n'est pas un processus de suppression des données et il ne rend pas les données inaccessibles.

La période avant l'archivage des données est fonction de la licence acquise par l'utilisateur.

Lorsque des données sont archivées, elles sont simplement déplacées dans une autre table de la base de données.

Le logiciel est conçu pour permettre la consultation des données archivées sans aucune différence avec les données non archivées, si ce n'est la nécessité d'utiliser un bouton permettant de passer des données non archivées aux données archivées.

A terme, l'archivage des données sera complètement transparent pour l'utilisateur.

4. Exportation des données pour l'administration fiscale

Dans le logiciel une page intitulée “Rapports” permet de télécharger différents rapports comptables sur différentes plages de dates avec différents formats.

Les formats peuvent être PDF, CSV, FEC, SAFT, HTML....

4. Modèles de données (modèles conceptuels)

4.1 Commandes

- id : identifiant interne
- datecreation : date de création de la commande
- datevalidation : date de validation (affectation du numéro de facture)
- datevaleur : date de valeur comptable
- idboutique : identifiant interne de l'établissement
- idcaisse : identifiant interne de l'établissement
- idutilisateur : identifiant interne de l'établissement
- idmodepaiement : identifiant interne de l'établissement
- prixinitial : avant TVA, avant réductions
- totalreduction : total des réductions appliquées
- totaltva : montant total de TVA
- prixfinal : prix final TTC
- titre : intitulé de la commande
- typetva : champ interne (champ non fiscal)
- nbarticles : nombre d'articles
- montantpaye : champ interne (champ non fiscal)
- idcompteclient : identifiant interne de l'établissement
- modepaiement : champ interne (champ non fiscal)
- terminee : champ interne (champ non fiscal)
- note : champ interne (champ non fiscal)
- notePublique : champ interne (champ non fiscal)
- pieceID : champ interne (champ non fiscal)
- remboursement : champ interne (champ non fiscal)
- idinterne : le numéro de facture attribué à la commande lors de sa validation
- choixlivraison : identifiant interne de l'établissement
- idtable : identifiant interne de l'établissement
- preparation : champ interne (champ non fiscal)
- alivrer : champ interne (champ non fiscal)
- livraisonTerminee : champ interne (champ non fiscal)

Conception Générale

- numcouverts : champ interne (champ non fiscal)
- pointsfidelite : champ interne (champ non fiscal)
- readyprepa : champ interne (champ non fiscal)
- dureePrepa : champ interne (champ non fiscal)
- dureePrepaOK : champ interne (champ non fiscal)
- numbippeur : champ interne (champ non fiscal)
- expediee : champ interne (champ non fiscal)
- factureEnvoyee : compteur d'impression/envoi de facture
- thehash : signature HMAC-SHA256 des champs fiscaux
- verifName : la signature recalculée lors de l'archive
- hashSource : la source du calcul de la signature
- validity : = "Valid HMAC SHA256" si la signature de la ligne correspond bien à la signature recalculée, sinon affiche "**** ERROR ****"

Format de la chaîne de caractères signée :

```
$c["datevalidation"]."_".$c["datevaleur"]."_".$c["idcaisse"]."_".importFloat($c["prixinitial"]).
"_".importFloat($c["totalreduction"])."_".importFloat($c["totaltva"])."_".importFloat($c["prix
final"])."_".$c["typetva"]."_".importFloat($c["nbarticles"])."_".$c["choixlivraison"]."_".$c["id"
]."_".$c["idinterne"]."_".$hash
```

4.2 Articles

- id : identifiant interne
- datecreation : date de l'ajout de l'article à la commande
- idboutique : identifiant interne de l'établissement
- idcommande : identifiant interne de commande
- idutilisateur : identifiant interne d'utilisateur
- idrayon : identifiant interne du rayon
- idplu : identifiant interne de l'article
- prixinitial : prix initial unitaire
- taux TVA : valeur du taux de TVA
- montantTvaDeductible : montant de TVA deductible
- taux TVA2 : montant de TVA en consommation sur place (champ non fiscal)
- tauxreduction : taux de réduction
- montantreduction : montant de réduction
- prixfinal : prix final TTC
- prixAchatPlu : prix achat (champ non fiscal)
- quantite
- nom : titre donnée à l'article
- image
- ventePartielle : champ interne (champ non fiscal)
- idpaiement : champ interne (champ non fiscal)
- preparation : champ interne (champ non fiscal)
- position : champ interne (champ non fiscal)
- idDeclinaison0-4 : champ interne (champ non fiscal)
- prixFinalHT : prix final HT
- thehash : signature de cette ligne
- verifName : la signature recalculée lors de l'archive
- hashSource : la source du calcul de la signature
- validity : = "Valid HMAC SHA256" si la signature de la ligne correspond bien à la signature recalculée, sinon affiche "***** ERROR *****"

Format de la chaîne de caractères signée :

```
$lasthash."_".$a["id"]."_".$a["datecreation"]."_".$a["idcommande"]."_".$a["idutilisateur"]."  
_".importFloat($a["prixinitial"])."_".importFloat($a["tauxTva"])."_".importFloat($a["montant  
TvaDeductible"])."_".importFloat($a["tauxTva2"])."_".importFloat($a["tauxreduction"])."_".i  
mportFloat($a["montantreduction"])."_".importFloat($a["prixfinal"])."_".importFloat($a["qu  
antite"])."_".$a["nom"]."_".importFloat($a["prixFinalHT"])
```

4.3 Paiements

- id : identifiant interne
- idboutique : identifiant interne de l'établissement
- idCommande : identifiant interne de la commande payée
- idVendeur : identifiant interne de l'utilisateur
- idCaisse : identifiant interne de la caisse
- idTypePaiement : identifiant interne de
- idModePaiement : identifiant interne de
- idUtilisateurCreditPaiement : donnée interne
- datePaiement : la date du paiement = date d'enregistrement du paiement
- montantPaye : le montant payé
- montantVerse : le montant utilisé pour payer le commande (si rendu de monnaie, différent)
- gocardlessID : donnée interne
- thehash : signature de cette ligne
- verifName : la signature recalculée lors de l'archive
- hashSource : la source du calcul de la signature
- validity : = "Valid HMAC SHA256" si la signature de la ligne correspond bien à la signature recalculée, sinon affiche "**** ERROR ****"

Format de la chaîne de caractères signée :

\$idcommande."_.\$idUtil."_.\$idCaisse."_.\$datePaiement."_.\$typeDeModesPaiement."
.".\$idModePaiement."_.\$importFloat(\$montantPaye)."_.\$importFloat(\$montantVerse)."_."
\$hash

Conception Générale

4.4 FermeturesAuto

La table qui enregistre les fermetures de caisse automatiques (quotidiennes, mensuelles, annuelles).

Le logiciel procède automatiquement à des clôtures de caisse automatique. Ces clôtures de caisse enregistrent pour chaque clôture : le niveau de la clôture, le total sur la période, le total historique, la date de début et la date de fin de la période.

L'événement qui déclenche la clôture de caisse automatique est la tentative par l'utilisateur d'enregistrer ou de modifier une opération de caisse comme par exemple la validation d'une commande, l'enregistrement d'un paiement pour une commande, etc. Cette clôture est effectuée de manière transparente sans que l'utilisateur soit averti de celle-ci et ne perturbe pas le processus de vente car elle est instantanée.

Cette clôture est réalisée à trois niveaux différents : quotidien, mensuel et annuel.

La clôture annuelle tient compte de la date de fin d'exercice de l'établissement tel que configuré dans la page configuration du logiciel.

- id : identifiant interne
- idboutique : identifiant interne de l'établissement
- dateStart : Date de début de période clôturée
- dateEnd : Date de fin de période clôturée
- idStart : Identifiant de la première commande de la période clôturée
- idEnd : Identifiant de la dernière commande de la période clôturée
- niveau : Niveau de clôture
- totalPeriode : Total du chiffre d'affaires sur la période clôturée
- totalHistorique : Total du chiffre d'affaires historique
- thehash : Y a-t-il un numérique de l'enregistrement
- thehash : signature de cette ligne
- verifName : la signature recalculée lors de l'archive
- hashSource : la source du calcul de la signature
- validity : = "Valid HMAC SHA256" si la signature de la ligne correspond bien à la signature recalculée, sinon affiche "***** ERROR *****"

Format de la chaîne de caractères signée:

```
$totaltsstart."_.$totaltsend."_.$totalidstart."_.$totalidend."_:importFloat($total)."_:importFloat($totalhistorique)."_.$lastHash
```

Conception Générale

4.5 Tracabilite

Cette table contient les données relatives à la traçabilité des opérations d'archivage. La purge n'étant pas effectuée dans le logiciel.

Les champs de l'archive sont :

- id : identifiant interne
- idboutique : identifiant interne de l'établissement
- idutilisateur : identifiant de l'utilisateur de l'établissement
- typeoperation : le type de l'opération (toujours 'archive')
- idcaisse : identifiant de la caisse
- dateoperation : la date de l'opération
- params : nom du fichier archivé
- idServer : l'adresse IP du serveur ayant réalisé l'opération tracée
- fileHash : signature HMAC-SHA256 du fichier archivé
- thehash : signature de cette ligne
- verifName : la signature recalculée lors de l'archive
- hashSource : la source du calcul de la signature
- validity : = "Valid HMAC SHA256" si la signature de la ligne correspond bien à la signature recalculée, sinon affiche "***** ERROR *****"

Format de la chaîne de caractères signée:

`$c["idutilisateur"]."_".$c["typeoperation"]."_".$c["dateoperation"]."_".$c["params"]."_".$c["fileHash"]."_".$hash`

5. Modalités de stockages des données

Les données sont stockées dans une base de données MySQL.

La base de données est hébergée sur un service cloud Digital Ocean, protégé par un accès administrateur sur un environnement Ubuntu, et la base de données est répliquée sur un serveur de réplication aux caractéristiques identiques au serveur maître.

Les copies de sauvegarde de la base de données sont effectuées à partir de la base de donnée répliquée, afin de ne pas perturber la base de donnée maître, et donc l'expérience utilisateur (sauvegarde sans interruption de service ou dégradation de service).

Afin de répondre aux problématiques de stockage des données, certaines tables ont été scindées en deux.

Il s'agit des tables d'enregistrement des **commandes** ainsi que des tables d'enregistrement des **articles**. Ces deux tables sont enregistrées sous forme de deux tables chacune.

Conception Générale

Les commandes de moins d'un an sont enregistrées dans la table "commandes", tandis que les commandes antérieures sont enregistrées dans la table "commandes_arch".

Les articles des commandes archivées sont enregistrés dans la table "articles_arch" tandis que les autres articles sont enregistrés dans la table "articles".

Lors de la consultation des données, la requête sera exécutée avec une **UNION** de données de ces deux tables.

Ce mécanisme permet de s'assurer que les tables commandes et articles restent avec une taille raisonnable (actuellement 1.5GO environ chacune, tandis que les tables d'archive font actuellement 10GO).

Le volume actuel des données correspond aux enregistrements réalisés dans le logiciel depuis février 2011 (soit actuellement plus de 14 années de données enregistrées).

Le logiciel peut fonctionner fluidement avec une table "commandes" qui peut aller jusqu'à 20GO.

La table "commande_arch", qui n'est interrogée que pour la génération d'archives, l'historique des commandes, ou l'export des données, réponds à des exigences de rapidité plus souples, et pourra également être stockée de manière compressée (ce qui n'est pas encore le cas), sans trop de perte de performances notable, ce qui laisse espérer des dizaines d'années de stockage de donnée sans avoir à apporter d'autres mécanismes d'économie de stockage.

Le logiciel fonctionne fluidement avec une table de données "archivées" allant jusqu'à 20GO.

Les données de caisse étant constituées quasiment exclusivement de données textuelles (sans contenu multimédia), elles sont fort propices à être compressées, avec des taux de compression très élevés, la solution que nous choisirons sera toujours orientée vers une compression, plutôt qu'une purge des données.

6. Identification du système

- identifiée dans SVN par numérotation (*révision SVN*)
- identifiée dans le code par un numéro de version incrémenté à chaque mise à jour

Le système d'encaissement est identifié par un numéro de version majeure, un numéro de version mineur et un numéro de révision inextricablement lié au système d'encaissement.

Ces numéros de version doivent être accessibles depuis l'interface utilisateur standard du système d'encaissement, via la page aide.

Conception Générale

Toute modification de code dans le périmètre fiscal ou paramétrage impactant le respect des exigences du référentiel doit entraîner une incrémentation du numéro de version majeure.

L'empreinte cryptographique de chaque version majeure doit être disponible à tout moment via un accès direct dans le logiciel.

Conformité au Référentiel de Certification v1.7

Inaltérabilité des données (Exigence IV.4 - Exigence 8)

Chaque transaction est immédiatement signée à l'aide d'un algorithme SHA256 chaîné, liant chaque écriture à la précédente. Toute modification a posteriori est techniquement détectable.

Sécurisation des données (Exigence IV.4 - Exigence 9)

Les données sont stockées sur des serveurs Linux sécurisés avec accès restreint par clé SSH. L'ensemble des communications passe par HTTPS/TLS. Les journaux d'activité sont conservés pendant 30 jours.

Enregistrement des données (Exigence IV.2 - Exigence 3)

Toutes les données d'encaissement, incluant date, heure, montant, mode de paiement, sont enregistrées immédiatement et de manière sécurisée.

Clôtures périodiques (Exigence IV.3 - Exigence 6)

Le système effectue les clôtures journalières, mensuelles et annuelles, tout en assurant la signature de chaque clôture.

Archivage des données (Exigence IV.5 - Exigence 10)

Les données peuvent être archivées mais cela ne nuit en rien à leur consultabilité.

Traçabilité des opérations (Exigence IV.7 - Exigence 15)

Toutes les actions critiques (encaissement, correction) sont journalisées et identifiées avec l'utilisateur concerné.

Conservation et Accès à l'administration fiscale (Exigence IV.9 - Exigence 19)

Un module dédié permet l'export conforme des données en format CSV, FEC, PDF, HTML, conforme aux attentes de l'administration fiscale.

Identification des versions majeures et mineures (Exigence IV.10 - Exigence 21)

Chaque version du logiciel est numérotée et enregistrée dans SVN. Tout changement de version majeure entraîne une nouvelle évaluation de conformité.

