Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement



***Rapport de Stage***

***Brevet de Technicien Supérieur 1ère année***

***Numérique, Option Informatique et Réseaux***

*Création d’un logiciel informatique pour*

*l’organisation d’une entreprise de déchets*

*informatiques (DEEE Recycling & Rachat Electronic Recycling)*

***Effectué à AGSI du :***

***16 mai 2022 au 24 juin 2022***

***Réalisé par :***

***DAUTEL ADRIEN***

***Encadrement : Anthony Guillot***

Une image contenant texte

Description générée automatiquement

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement



**Sommaire**

Page d’accueil **...........................................................p.1**

**I-Présentation de l’organisme d’accueil**

I.1 Présentation de la société AGSI ……………....................p.3

I.2 Présentation des matériels utilisés ………………………...p.4

I.3 Société DEEE Recycling & Rachat Electronic Recycling..p.5

**II-Présentation et Réalisation du projet**

II.1 Présentation du projet .....................................................p.5

II.2 Réalisation du projet .......................................................p.6

2.1 Analyse des besoins..................................................p.6

2.2 Création de mes besoins sur WinDev........................p.7

2.3 Création de mes boutons et interfaces........................p.8

2.4 Création d’une interface pour les interventions..........p.9

2.5 Réalisation d’un CRUD (Create, Read, Update, Delete) pour chaque zone de l’application ..........................................p.11

2.6 Procédure global et local.........................................p.14

2.7 Code et explication de la « Facture de Vente » .........p.15

2.8 Code de l’état pour le CERFA et la Facture ............p.17

2.9 Envoie des états par email ........................................p.20

**III-Conclusion**

III.1 Remerciements ...........................................................p.25

III.2 Annexe - Anglais ........................................................p.25

III.3 Annexe - Certificat de stage en entreprises ....................p.27

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

**I.1 Présentation de la société AGSI** 

La société AGSI (Anthony Guillot Solution Informatique) est une entreprise crée en 2001 situer à Louhans, 170 Rue des Vaux, 71500.

Une image contenant carte

Description générée automatiquementUne image contenant carte

Description générée automatiquement

AGSI est une société composée de Anthony Guillot et de son associés Geoffroy Vernet qui tient une société à Louhans spécialisée dans l’automatisme et l’électronique.

L’organigramme de cette société est donc la suivante :

Anthony Guillot

Gérant d’AGSI

Geoffrey Vernet

Associé d’AGSI

Adrien Dautel

Stagiaire

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

L’entreprise AGSI est spécialisée dans le secteur des systèmes et logiciels informatiques. Sur l'année 2021 elle réalise un chiffre d'affaires de 162 800 €.

Elle crée et développe des logiciels qui permettent aux entreprises de pouvoir gérer des stocks de marchandises, faire une caisse de paiement, faire une gestion de planning pour entreprise, etc... L’entreprise effectue ses logiciels pour toute sorte de secteurs, qu’ils soient industriels, commerciales, libérales, artisanales, etc...

**I.2 Présentation des matériels utilisés**

Pour le matériel à disposition, je possède un pc ASUS qui tourne sous Windows 7, qui coûte 400€ et d’un écran BENQ de 120€.

L’entreprise possède un serveur en local pour pouvoir procéder à des maintenances grâce au logiciel de gestion à distance VNC (Virtual Network Computing). Elle utilise une société française WinDev conçu par PC SOFT pour développer des applications, principalement orientés données pour Windows, Linux, .NET et Java. Cette application utilise son propre langage le WLangage.

La société AGSI est constitué d’un principal établissement situé au même endroit ou réside Anthony Guillot (170 Rue des Vaux, Louhans 71500)

Voici quelques images de l’entreprise et de son local

Une image contenant texte, équipement électronique

Description générée automatiquementUne image contenant texte, intérieur, encombré

Description générée automatiquement

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

**I.3 Société DEEE Recycling & Rachat Electronic Recycling**

La Société DEEE Recycling et Rachat Electronic Recycling sont 2 entreprises créé par le responsable Marc Delhaye en 2020. La société se situe à Chatenoy-le-Royal, 5 Rue de la Guerlande, 71880.

La société est divisée en deux parties, DEEE Recycling est l’entreprise qui va récupérer les déchets informatiques directement chez les particuliers, tandis que la société Rachat Electronic Recycling va revendre les composants électroniques.

Marc Delhaye est le seul au sein de ces entreprises, il se rémunère uniquement sur la valorisation des matériaux qu’il récupère.

**II.1 Présentation du projet**

Dans le contexte actuelle, Marc Delhaye fait face à une grande demande de composants électroniques, en effet beaucoup de matières premières proviennent d’Ukraine c’est-à-dire qu’il est très difficile pour les européens de l’ouest d’en bénéficier et donc de crée de nouveaux composants.

Face à cette demande grandissante Marc Delhaye s’est tourné vers AGSI pour pouvoir concevoir un logiciel lui permettant d’être plus rapide dans ces activités.

Le but du projet et de mon stage était donc de réaliser un logiciel pour tablette pouvant gérer les entrés et les sortis de stock. De plus il fallait pouvoir envoyer au client et aux acheteurs une facture d’achat et un CERFA (Centre d’Enregistrement et de Révision des Formulaires Administratif). Un CERFA est un formulaire administratif réglementé, c’est un document officiel dont un arrêté fixe le modèle. Lorsque des déchets informatiques changent de

lieux il est obligatoire pour Marc Delhaye d’en avoir un pour chaque déchet informatique.

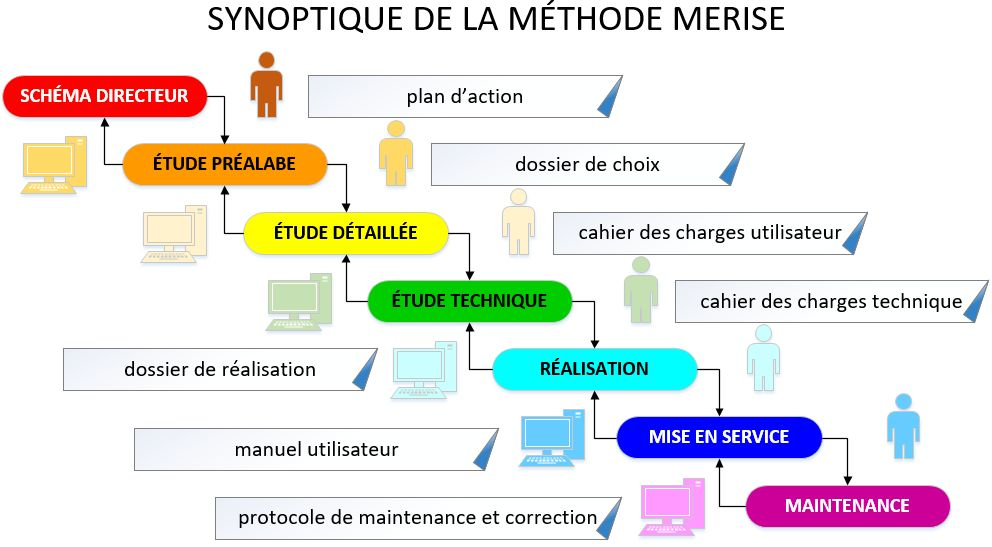
Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

**II.2 Réalisation du p****rojet**

**2.1. Analyse des besoins**

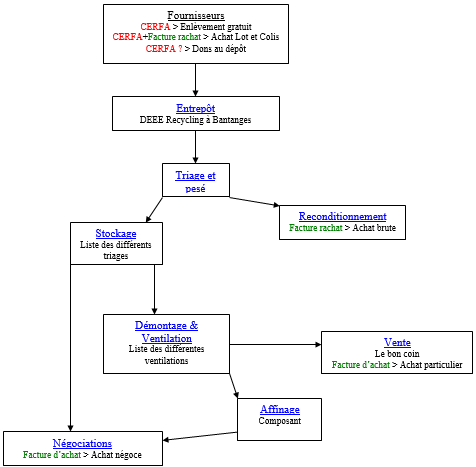
J’ai donc utilisé la méthode de développement MERISE



**ANALYSE**

**REALISATION**

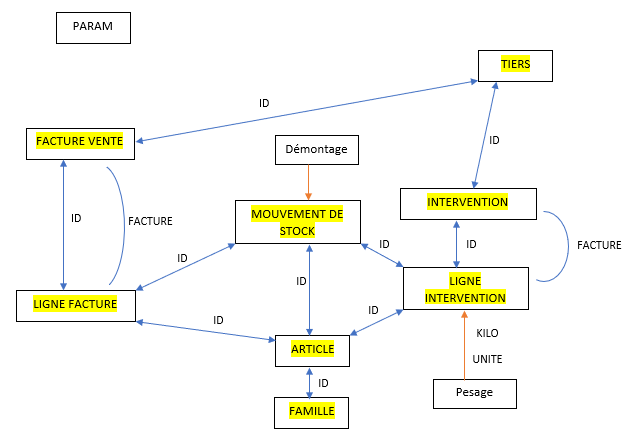
Pour la partie analyse j’ai d’abord fait une étude préalable avec toute les actions que l’application devra faire :



Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

Ensuite j’ai réalisé l’étude détaille en commençant à faire ma base de données en mettant les classes qui s’y trouverais et à faire le schéma du logiciel :



Chaque case surlignez en jaune correspond à une classe.

Enfin j’ai réalisé l’étude technique en précisant les éléments à mettre dans chaque classe, (ex : Dans la classe « TIERS » il y a IdTiers, nom, prénom, email, tél, etc.…)

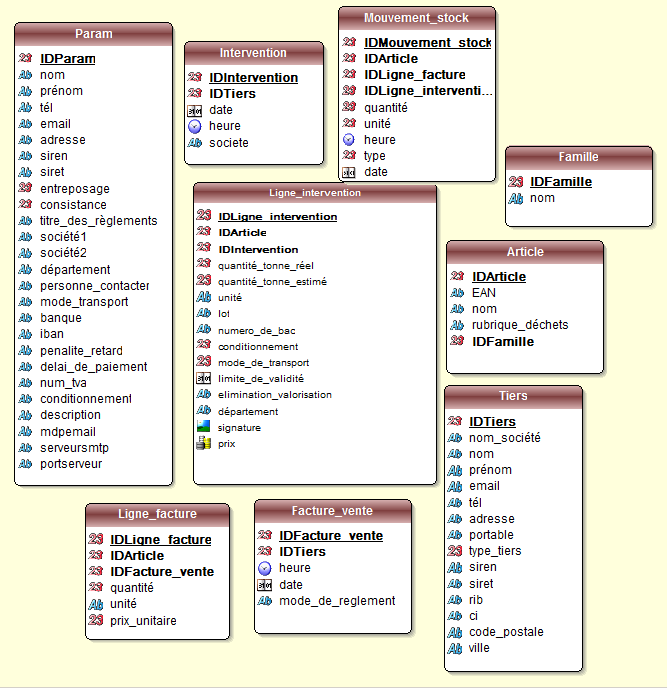
La classe « PARAM » n’est pas une classe comme les autres, en effet c’est une classe qui contient déjà les informations de Marc Delhaye et de ses entreprises. Ses informations sont remplies directement en brute dans la base de données pour faciliter les impressions de facture d’achat et de CERFA. Cependant c’est une classe qui doit exister puisque toutes les informations sont récupérées en base de données.

**2.2 Création de mes besoins sur WinDev**

Une fois que ma partie Analyse a été terminer je suis passé sur le logiciel de création WinDev.

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

J’ai donc créé mes classes avec tous les éléments et les ID de liens qui leurs correspondaient.

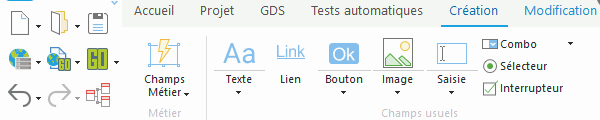
J’ai changé le type de chaque élément (numérique, texte, date, etc.…)

Tout d’abord je me suis intéressé à la première fenêtre lorsque j’allais ouvrir mon application, c’est-à-dire MENU.

**2.3** **Création de mes boutons et interfaces**

J’ai créé des boutons qui permettront de naviguer sur chaque classe.

Pour crée un bouton il suffit de déplacer dans l’interface de disposition « Bouton » sur la fenêtre.



Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

C’est le même procéder pour crée une fenêtre, on doit aller dans « Projet », puis « Nouveau » et enfin « Fenêtre ».

Après avoir mis tous les Boutons, je leur ai données un style, j’ai ajouté des images et leurs est donnée un meilleur rendu.Une image contenant texte

Description générée automatiquement

Nom du BOUTON

Le Libellé du bouton (ce qu’il y aura écrit dessus)

Ce logo donne l’accès au code de l’élément.

Image est pour enlever, ajoutée, crée où modifier une image qui apparaitra sur l’élément sélectionner

L’IHM est pour modifier l’apparence de l’élément sélectionner.

Liée l’élément avec un paramètre dans la base de données.

**2.4 Page Menu de l’application**

J’ai ajouté un Libellé pour le nom de l’entreprise, une image de fond neutre et un bouton Quitter qui arrêtera l’application

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

Une image contenant texte

Description générée automatiquementVoici donc à quoi ressemble l’écran d’accueil de l’application, (la fenêtre s’appelle donc FEN\_MENU) :

Comme je l’ai expliqué précédemment chaque classe a son bouton dans l’écran d’accueil c’est-à-dire qu’il ont aussi leurs fenêtre dédier.Une image contenant table

Description générée automatiquement

Pour ouvrir une fenêtre il faut aller dans le code du bouton (voir logo dans « Description d’un bouton » ). Puis dans une zone nommé « Clic sur BTN\_ (nom du bouton) » écrire tout simplement « Ouvre(FEN\_(nom de la fenêtre)) » ce qui ouvrira la fenêtre en question.

Une fois les boutons et les fenêtres crées, il faut que je mette en place mon système de mouvement de stock à l’aide des interventions.

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

Pour crée mon interface j’ai décidé de recenser toutes les interventions dans 2 tableaux. Sur ces 2 tableaux, un me servira pour les interventions c’est-à-dire avoir un ID d’intervention, un nom, une date, une heure, et la société (DEEE Recycling ou Rachat Electronic Recycling) qui s’en chargera. L’autre tableau sera en lien avec une intervention relié par son ID, j’ai donc nommé se tableau « Ligne d’intervention ». Voici à quoi ressemble la fenêtre d’intervention :

Une image contenant table

Description générée automatiquement

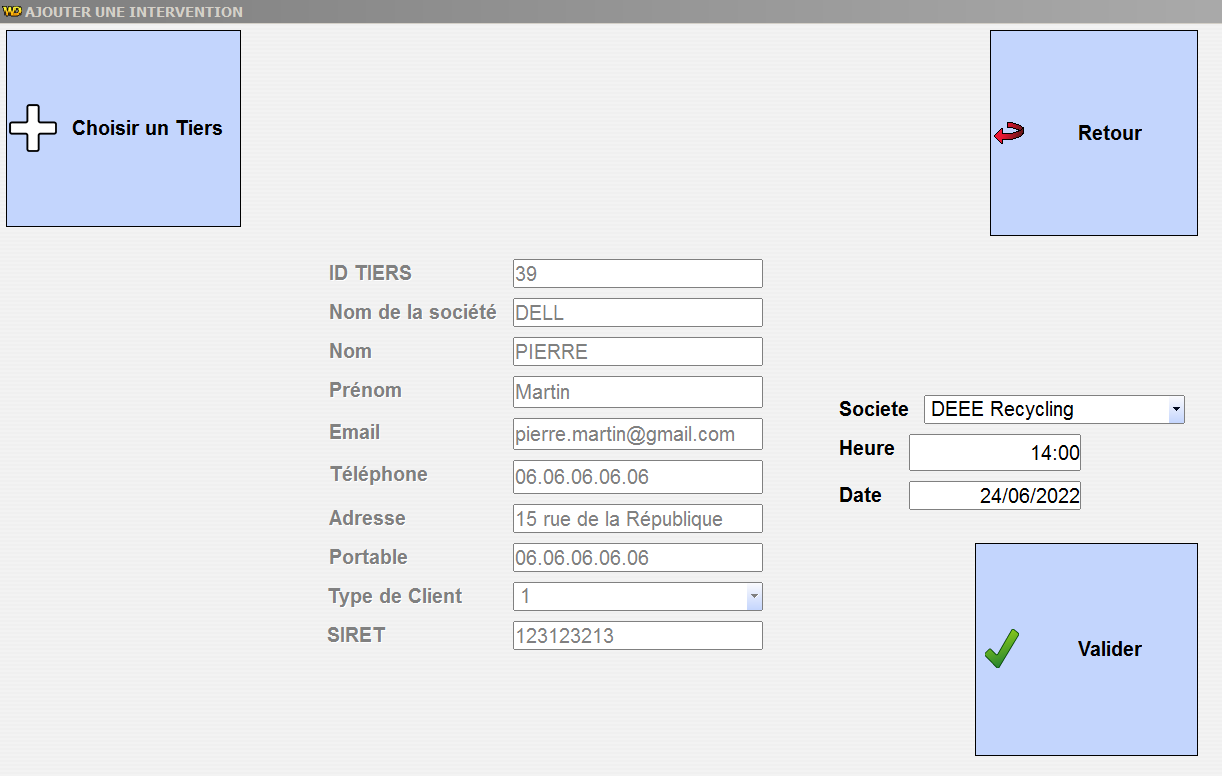
Comme vous pouvez le voir lorsque je clic sur une ligne dans le tableau intervention (à gauche) j’ai tout les objets informatiques qui ont été utilisé lors de cette intervention avec leurs caractéristiques (tableau à droite).

**2.5 Réalisation d’un CRUD**

Pour crée une intervention et une ligne d’intervention j’appuie sur le bouton « Création d’une intervention » ce qui m’amène dans une nouvelle fenêtre :

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement



La fenêtre s’appelle « AJOUTER UNE INTERVENTION », je dois donc choisir quelle société c’est charger de l’intervention, l’heure et la date.

Une image contenant table

Description générée automatiquementUne fois ces informations remplis je dois sélectionner un tiers qui sera associé à cette intervention. Ce bouton me ramène sur une autre fenêtre avec la liste de tous les tiers déjà enregistrer.

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

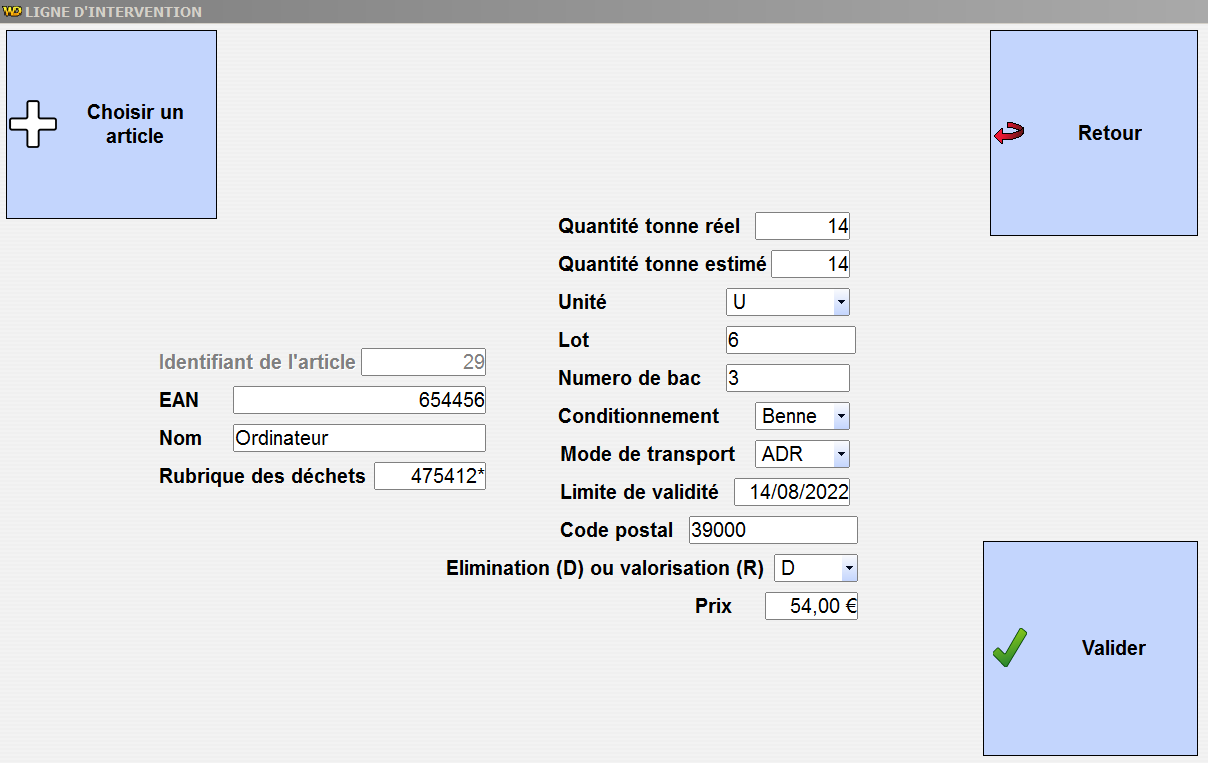
Je sélectionne une ligne puis je valide ce qui me remplira automatiquement les cases grisées dans ma fenêtre « AJOUTER UNE

INTERVENTION », le bouton Valider va donc enregistrer et mettre dans le tableau ma nouvelle intervention avec son tiers associé.

Maintenant que j’ai une intervention il me faut une ligne d’intervention avec les objets utilisés.

Une image contenant table

Description générée automatiquement

J’ai donc créé un bouton « Création d’une ligne d’intervention ». Cette fenêtre fonctionne exactement comme la fenêtre de création d’une intervention, elle m’ouvre une nouvelle fenêtre grâce à laquelle je pourrai remplir les informations de l’article :

Voila à quoi ressemble cette fenêtre.

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

**Remarque :** Sur chaque fenêtre dispose d’un bouton « Retour » ce qui permet à l’utilisateur de revenir sur la fenêtre précédente en cas de fausse manipulation, le code est simple il suffit d’écrire « Ferme » dans la zone « Clic sur BTN\_RETOUR »

**2.6 Procédure globales et locales**

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquementPour le code de toute ces actions j’ai beaucoup utilisé de procédure global, c’est-à-dire des variables avec un type définie (chaine de caractère, un entier, un booléen, etc.…) que je peux utiliser dans n’importe quelle fenêtre du projet. J’ai utilisé des procédures locales qui sont des morceaux de code qui sont assez longue à coder, ces procédures locales agissent comme des raccourcis sur le codage.

De plus j’ai utilisé la fonction *HLitRecherche* dans presque tout mes tableaux, fenêtre, bouton, etc... Cette fonction permet de se positionner dans le fichier de données pour effectuer une boucle de parcours parmi les enregistrements qui correspondent à une condition.

La fonction *HLitRecherche* ce défini comme cela :



J’ai pris le code de la « Facture de vente » et la « Ligne de Facture » (qui fonctionne exactement comme le code de l’ « Intervention » et la « Ligne d’Intervention »

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

**2.7 Code et explication de « Facture de Vente »**

Une image contenant texte

Description générée automatiquementVoici le code du tableau de la facture de vente :



Ici *gnFacturevente* est une procédure globale qui sert à utiliser la ligne sélectionnée.

La fonction *HLitRecherche* est utilisé ici pour récupérer dans la base de données « Facture de vente » l’ID qui correspond à l’ID associée dans la base de donnée « Ligne de Facture »

Puis j’utilise AFF\_LIGNEFAC ()  qui est une procédure locales que j’ai créé pour afficher les lignes dans le tableau « Ligne de Facture » (à droite)

Une image contenant table

Description générée automatiquement

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

Maintenant que nous avons l’« Intervention » et la « Facture de vente » je peux gérer le stock avec ma classe « Mouvement de Stock »

Une image contenant table

Description générée automatiquementVoici à quoi ressemble la fenêtre du « Mouvement de stock » :

Le « Mouvement de stock » est l’endroit de l’application ou tout est regrouper pour que Marc Delhaye puisse regarder les articles qu’il a vendu, acheter, démonter, valoriser, etc...

Lorsqu’une « Intervention » ou une « Facture de vente » est créé ou supprimer, celle-ci sera donc crée où supprimer dans le tableau « Mouvement de stock » grâce à son ID.

Une fois que tout fonctionne correctement il faut que je puisse envoyer les factures et les CERFA.

Pour le CERFA cela se passe dans l’« Intervention », tandis que la facture d’achat se passe dans la « Facture de vente ».Tout d’abord j’ai créé la facture et le CERFA à l’aide d’un état.Un état sur WinDev est un document que l’on peut créer et qui peut être imprimé, il peut être sous forme PDF, Word, Excel, etc...

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

**2.8 Code de l’état pour le CERFA et la Facture**

Pour les déchets un CERFA existe, le CERFA n° 12571 – Bordereau de Suivi des Déchets Dangereux (BSDD)

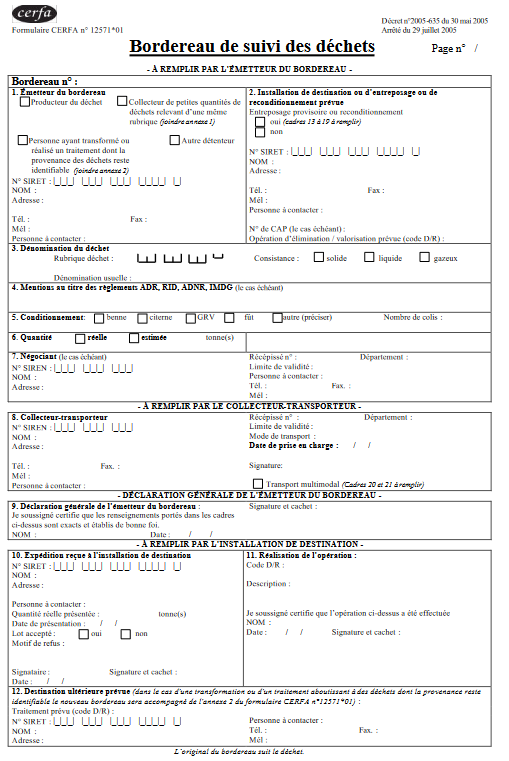
Ce CERFA contient deux pages c’est-à-dire qu’il me faut deux états.

Un état est constituer de forme géométrique, de texte et de rubrique, une rubrique permet d'éditer le contenu d'un fichier de données (elle agit comme un libellé mais avec des liaisons à la base de donnée).

Il est impossible de scanner le CERFA avec WinDev (version 10), donc j’ai dû le faire à la main. Comme je l’ai dit précédemment c’est un document officiel donc il faut que les champs soit au même endroit que sur l’original.

Voila ce que j’ai fait

Voici à quoi doit ressembler le CERFA (officiel)

 Une image contenant table

Description générée automatiquement

(CERFA rempli pour observer le rendu final)

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

Voilà une partie du CERFA de comment les rubriques, les textes, les cases, etc.… sont faites :

Une image contenant table

Description générée automatiquement

Chaque cases grises corresponds à une rubrique.

Une rubrique commence par « RUB\_ (nom de la rubrique) »

Une image contenant texte

Description générée automatiquementLe code de correspondant est le suivant, comme pour le code de l’« Intervention » et de la « Facture de vente » J’ai utilisé *HLitRecherche*

J’ai donc recherché d’abord les éléments qui se trouvaient dans la base de données grâce à l’ID, puis je les ai assignez un à un à chaque rubrique qui aura été créé pour l’occasion.

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

Pour l’état de la facture d’achat j’ai utilisé le même procéder (à l’aide des rubriques) que pour le CERFA.

Une image contenant table

Description générée automatiquementVoici à quoi ressemble la facture (je me suis inspiré des exemples trouvé sur internet pour crée mon modèle de facture) :

(Facture rempli pour observer le rendu final)

Le code de la facture est légèrement différent :

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquementUne image contenant texte

Description générée automatiquement

Je n’ai pas utilisé de *HLitRecherche*car je n’ai pas besoin de l’ID d’un article puisqu’il se peut que l’article soit utilisé plusieurs fois dans différentes factures. J’ai donc utilisé *HFiltre* et *HLitPremier* car c’est en fonction de la « Ligne de Facture » que l’article dépend.

J’ai donc ensuite été obliger de crée une boucle pour que tous les articles soient dans la facture (si il y en a plusieurs) car sinon le logiciel ne prendrait que le premier.

Et pour finir j’ai fait les calcule pour que le prix au total soit le bon.

**2.9 Envoie des états par email**

Une image contenant texte

Description générée automatiquementAprès avoir fait tous le CERFA et la facture d’achat il a fallu que je puisse les envoyer je suis donc retourné sur mes fenêtres d’« Intervention » et de « Facture de vente » pour pouvoir crée 2 bouton, le premier qui donnera un aperçu, et le deuxième pour pouvoir imprimer et/ou envoyer au destinataire grâce à un email que nous renseignerons via un ID de la classe « Tiers ». De plus je crée un troisième bouton dans la fenêtre « Intervention » qui servira pour la signature du CERFA.

Une image contenant texte

Description générée automatiquementVoici donc les boutons :

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

Pour avoir un aperçu j’ai utilisé la fonction *iAperçu* et *iImprimeEtat*, Une image contenant texte

Description générée automatiquement

(Exemple avec le CERFA, même procéder pour la facture)

La fonction *iAperçu* avec le paramètre *i100* donne un aperçu avec un zoom de 100%. Cependant l’aperçu ne dure que quelque seconde, j’ai donc dut utiliser la fonction *iImprimeEtat* qui va donner une impression dans le visualisateur de rapport et donc l’aperçu sera afficher jusqu’à ce que je fasse une action (imprimer, convertir le format, etc...).

Maintenant que j’ai vu comment serait le CERFA il ne me reste plus qu’a l’envoyer.

Voici le code du bouton en question :

Une image contenant texte

Description générée automatiquement

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

Une image contenant texte

Description générée automatiquementTout d’abord pour envoyer un mail je me suis connecter en programmation sur le mail de l’entreprise grâce à la fonction *EmailOUVRESessionSMTP*, cette fonction est de la syntaxe suivante :

J’ai créé une boucle pour tester la connexion au serveur si elle est vrai alors l’envoie peut marcher, sinon le logiciel m’affiche une alerte pour me dire que je n’est pas réussi à me connecter au mail.

J’ai donc utilisé le nom d’utilisateur, et le mot de passe (ici masqué en jaune) et j’ai défini le port SMTP (Simple Mail Transfer Protocol).

J’ai utilisé le type de variable *Email* qui m’a permis d’ajouter des propriétés spécifiques ce qui me permettra de modifier les effets de *Email*.

J’ai donc le Destinataire qui est une adresse email que j’utilise pour faire mes tests (si je reçois bien le CERFA enregistrer).

L’Expéditeur, ici c’est le mail de l’entreprise (je me suis connecté avec).

Le Sujet, qui est l’objet de l’email.

Le Message, qui est le texte que contient l’email.

Le NbDestinataire, qui est le nombre de personne à qui j’envoie l’email.

Ensuite j’ai *iDestination* (*iPDF*, « ... »), cette fonction sert à nous retrouver le nom d’un fichier grâce à son chemin et de le convertir en format PDF.

Avec la fonction *iImprimeEtat*, je laisse le format PDF actif, puis avec le *Email.Attache* je défini l’ordre et je récupère les PDF convertis à l’instant pour les mettre en pièces jointes.

Je fini par préciser le nombre de pièce jointe avec NbAttache.

J’ai de nouveau créé une boucle pour savoir si mon envoie à bien fonctionner

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

s

J’ai donc bien reçu les deux pièces jointes par programmation sur mon adresse mail.

**2.10 Signature du client**

Une image contenant texte

Description générée automatiquementPour pouvoir faire signer le client j’ai créé un bouton signature qui m’ouvre une fenêtre dans laquelle le client pourra signer avec son doigt ou un stylet grâce à la tablette :

Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

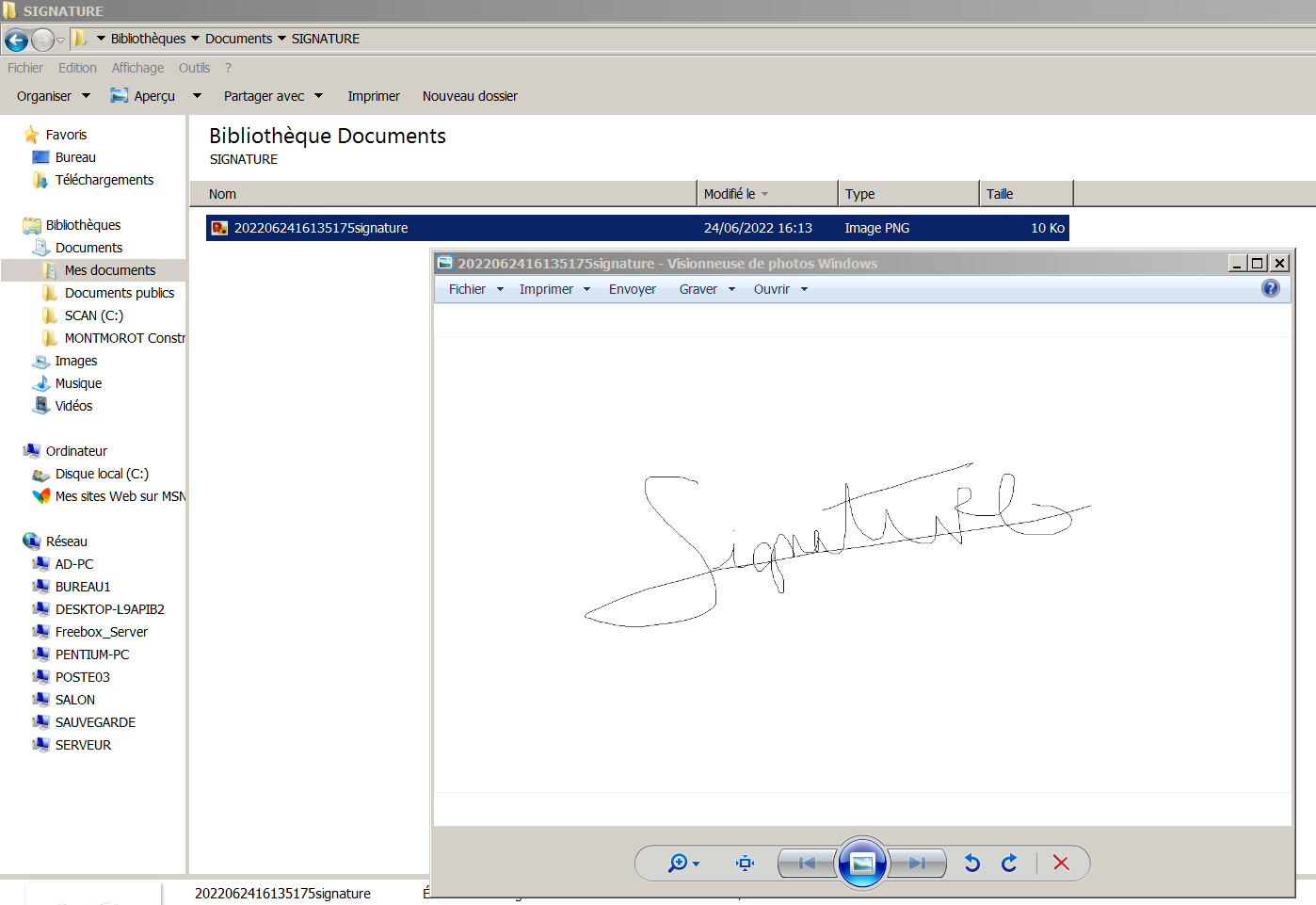
Pour pouvoir dessiner j’ai utilisé la fonction *dDebutDessin*, *dStylo* et *dFinDessin*. Ces fonctions permettent de dessiner dans un champ image ce que l’on veut.

Une fois que le client a fini de faire sa signature, il lui reste à appuyer sur le bouton « Valider » pour que l’image se sauvegarde dans un espace dans les fichiers de la tablette. Pour ce faire j’ai utilisé une procédure globale de type chaine et une fonction *dSauveImagePNG*, cette fonction sert a enregistrer une image à l’aide d’un chemin :



J’ai d’abord appelé l’image en question puis je lui est dit de s’enregistrer sur le chemin suivant. De plus j’ai décider de lui mettre la date et l’heure à laquelle la signature a été faite pour qu’il n’y est pas de doublons et donc de problème avec le CERFA selon l’ID du Tiers.

J’ai donc crée un nouveau dossier pour ranger toutes les signatures de chaque client qui est celui que j’ai renseigner dans mon code :



Une image contenant texte, horloge, périphérique, jauge

Description générée automatiquement

Maintenant que tout mon logiciel est réalisé je l’ai transférer sur une tablette pour ajuster les dimensions de l’écran.

Mon logiciel est terminé et l’installation chez le client s’est faite sans aucuns problèmes.

**III.1 Remerciements**

Je tiens tout d’abord à remercier Anthony Guillot , mon maître de stage, qui occupe la fonction de gérant au sein de la société AGSI, et qui m’a accueilli comme stagiaire dans son entreprise. Il m’a beaucoup appris sur la façon de travailler et sur ses missions au quotidien. Durant mon stage, j’ai eu la chance de pouvoir bénéficier des conseils de Geoffrey Vernet son associé et je voudrais les remercier pour le temps qu’ils m’ont accordé et leur aide précieuse.

**III.2 Annexe – Anglais**

Taizhou, Zhejiang Province, China. At a small factory workers melt down computer parts.... The eastern Chinese port city of Taizhou has now become the center for much of the illegal trade in e-waste (computer waste). 24 hours a day ships arrive in the city harbor carrying cargo of waste, including millions of computer parts. These parts flow out of the port in trucks into the city and hinterland where hundreds of tiny work shops break down the parts, melting off the precious metals and using acids to separate gold from circuits boards, mobile phones, monitors and other computer parts. Neighborhoods are filled with the noxious and toxic fumes of this unregulated industry and thousands of men, women and children are exposed to a toxic cocktail of fumes and dust released.

<https://stuartisett.photoshelter.com/image/I0000h1gpKWmkbco>

Atom is a free and open-source text and source code editor for macOS, Linux, and Microsoft Windows with support for plug-ins written in JavaScript, and embedded Git Control. Developed by GitHub, Atom is a desktop application built using web technologies. Most of the extending packages have free software licenses and are community-built and maintained. It is based on the Electron framework, which was developed for that purpose, and hence was formerly called Atom Shell. Electron is a framework that enables cross-platform desktop applications using Chromium and Node.js. Atom was initially written in CoffeeScript and Less, but much of it has been converted to JavaScript.

Atom was released from beta, as version 1.0, on June 25, 2015.Its developers call it a "hackable text editor for the 21st Century", as it is fully customizable in HTML, CSS, and JavaScript.

<https://en.wikipedia.org/wiki/Atom_(text_editor)>

## E-waste Mining

Even though the world knows that e-waste holds billions’ worth of components and metals, we don’t recycle as much as we can. We recycle 20% of electronic waste through the right channels. But now that private and public sectors are becoming more aware of their impact on the environment, things should change in the coming years, starting with mining.

E-waste mining doesn’t involve extracting ores buried underneath the ground. It’s the process of extracting valuable metals from existing gadgets. Instead of dumping old and obsolete devices in landfills, they can be used to create new ones. It’s beneficial to the environment, and it’s going to be great for the economy.

In a study published in *Environmental Science and Technology,* researchers from Macquarie Graduate School of Management and State Key Joint Laboratory of Environment Simulation suggested that e-waste mining could be as profitable as de novo mining.

https://securis.com/news/what-is-the-future-of-electronics-recycling/

**III.3 Annexe – Certificat de Stage**

**Une image contenant texte, reçu, document

Description générée automatiquement**