

ECOLE PRATIQUE DES HAUTES ETUDES COMMERCIALES

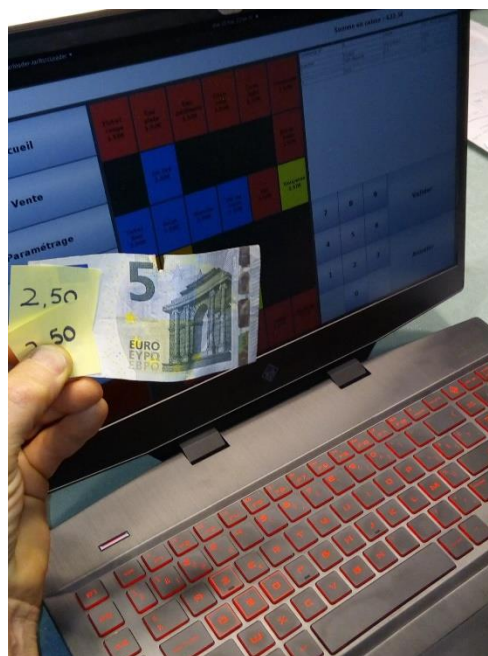


Avenue du Ciseau, 15
1348 Louvain-la-Neuve

Développement personnalisé d'un point de vente pour le judo club Athois

Travail de fin d'études présenté en vue de l'obtention du diplôme de bachelier en Informatique et Systèmes
orientation Technologie de l'Informatique

Gaëtan GORTZ



Rapporteur : Claude MASSON

Année Académique 2020 – 2021

Remerciements

Je tiens à remercier tout particulièrement le Judo Club Athois et les organisateurs du Grand Prix de Judo, Madame Barré, Madame Delhayé et Monsieur Provost pour l'idée et le temps qu'ils m'ont consacré pour me décrire leurs besoins ainsi que pour leurs retours.

De même, je tiens à adresser mes remerciements aux utilisateurs finaux qui ont également contribué à alimenter ces besoins et ces retours.

Je tiens aussi à remercier mes parents qui m'ont soutenu dans ce travail et sa relecture mais aussi dans ces trois années d'études. De même, je remercie toutes les personnes qui ont participé à la relecture de ce rapport de TFE.

Je tiens également à remercier Monsieur Van Dormael et Monsieur Noël que le sujet semblait intéresser et qui m'avaient donné leur accord pour devenir mon promoteur.

Et last but not least, mes plus vifs remerciements vont à Madame Masson, ma promotrice, qui a toujours pu se rendre disponible très rapidement pour répondre à mes questions et me faire des retours sur mes états d'avancement. Je lui en suis profondément reconnaissant.

Table des matières

Remerciements	2
Table des matières	3
Introduction	7
Besoins du client	8
Rôles	8
Écrans et fonctionnalités des users stories	8
US1 - Installation	8
US2 - Paramétrage	9
US3 - Accueil	10
US4 – Initialisation de la caisse	11
US5 – Vente de tickets	12
US6 – Achat de tickets	14
US7 – Impression de la vente	14
US8 – Mise en sécurité	14
US9 – Rapport mise en sécurité	16
US10 – Rapport des ventes de la journée	17
US11 – Imprimer le ticket d’une commande	18
US12 – Rapport fonds de caisse	19
Choix techniques	20
Choix du langage	20
Java	20
Python	20
Ruby	21
C#	21
Choix de la base de données	21
MySQL	21
HyperSQL	22
Conception	23
DBHelper	23

Déroulement	23
Conclusion	24
Interface vente.....	24
Déroulement	25
Conclusion	28
Interface fonds de caisse	28
Déroulement	28
Conclusion	29
Interface mise en sécurité	29
Déroulement	29
Conclusion	29
Interface paramétrage.....	29
Déroulement	30
Conclusion	30
Interface accueil.....	31
Déroulement	31
Conclusion	31
Interface rapports.....	31
Déroulement	32
Conclusion	32
Sécurité	34
Conclusion.....	35
Tests	36
Tests unitaires.....	36
Tests d'intégration par le client.....	36
Conclusion.....	37
Analyse personnelle.....	37
Analyse critique	37
Piste d'améliorations	38
Importance élevée	38
Importance moyenne.....	38
Importance faible	38
Annexes.....	39

Annexe I : Manuel d'utilisation.....	40
Installation du programme.....	40
Utilisation du programme	41
Annexe II : Users stories completes.....	51
Personnage.....	51
User Stories	51
US1 - Installation	51
US2 - Paramétrage	51
US3 - Accueil.....	53
US4 – Initialisation de la caisse	54
US5 – Vente de tickets	56
US6 – Achat de tickets.....	57
US7 – Impression de la vente.....	58
US8 – Mise en sécurité.....	58
US9 – Rapport mise en sécurité	61
US10 – Rapport des ventes de la journée	62
US11 – Imprimer le ticket d'une commande	63
US12 – Rapport fonds de caisse	64
Annexe III : Code source	66
Bibliographie	67
Choix technique	67
Comparaison autre software	67
Analyse de sécurité.....	67
Java	68
Installation.....	68
MVC	68
Interface graphique.....	68
Affichage des tableaux	68
Variables.....	69
Stockage des données.....	69
Affichage du fond d'écran Accueil	69
Impression rapports	69
Application exécutable.....	70

Base de données.....	70
MySQL	71
Derby	71
HSQLDB.....	71
Tests unitaires.....	71

Introduction

Dans ce TFE je vous parlerai du développement d'un point de vente pour le Judo Club Athois.

En effet, le Judo Club Athois souhaitait professionnaliser sa gestion de la comptabilité lors des Grands Prix. Les solutions imaginées les années précédentes demandaient un travail important. Qu'il s'agisse de l'utilisation d'une version démo d'un logiciel payant ou d'un logiciel freeware, ces solutions ne convenaient pas entièrement aux organisateurs.

- Logiciels payants :

L'obtention/achat d'une licence d'un logiciel payant n'est pas envisageable vu son coût élevé pour une seule utilisation annuelle.

L'utilisation d'une version 'démo' est limitée dans le temps, habituellement 1 mois maximum. Ceci implique de réinstaller chaque année l'application démo et de refaire complètement l'ensemble des paramètres. Souvent l'interface de paramétrage est peu conviviale et ne peut être réalisée que par une personne qui s'y connaît en informatique et qui comprend la logique sous-jacente.

- Logiciels gratuits :

Il existe quelques logiciels freeware tant pour Windows que pour Linux.

Ces logiciels impliquent eux aussi un paramétrage important mais celui-ci peut être conservé d'année en année.

Cependant ces logiciels ont été développés pour répondre au mieux à un maximum de besoins de leurs clients. Ils s'avèrent dès lors lourds et compliqués à utiliser.

C'est pourquoi, j'ai proposé aux organisateurs, dans le cadre de mon travail de fin d'étude, de concevoir une application de Point de Vente personnalisée afin qu'elle soit à la fois simple et conviviale pour l'utilisateur final mais également complète pour répondre aux besoins de statistiques des organisateurs.

Pour développer cette application, j'ai suivi les grandes phases d'un projet informatique. J'ai commencé par élaborer, avec les utilisateurs finaux, le cahier des charges. Partant de là, j'ai posé les grands choix techniques et enfin je me suis lancé dans le développement du code. Pour ce faire, j'ai commencé par l'écran le plus compliqué et le plus important c'est-à-dire "Vente". Quand cet écran a été entièrement développé et testé, j'ai développé les autres écrans en utilisant le même canevas. Au fur et à mesure qu'un nouvel écran était disponible, j'ai demandé aux utilisateurs de le tester afin d'obtenir leurs remarques.

Vous trouverez dans les pages ci-après une synthèse du cahier des charges, les choix techniques que j'ai été amené à faire, la conception proprement dite de l'application et enfin un chapitre concernant la sécurité.

Je terminerai ce rapport en faisant le bilan de mes apprentissages personnels et de l'application opérationnelle.

Besoins du client

Les besoins des organisateurs du Grand Prix ainsi que des principaux utilisateurs ont été analysés en 2 temps afin de comprendre ce qui était le plus important pour eux.

Tout d'abord lors du lancement de l'idée de créer une application de point de vente pour le Grand Prix de Judo où les grandes fonctionnalités ont été décrites.

C'est suite à cette première discussion qu'une des utilisatrices a dessiné les écrans tels qu'elle se les imaginait.

Avec les 2 utilisateurs principaux, sur base de ces écrans et en s'imaginant le déroulement d'un Grand Prix, depuis la préparation jusqu'au débriefing, j'ai décrit les fonctionnalités dans le détail, sous forme de user stories (US).

Le détail de toutes les user stories se retrouvent en annexe. Sont repris ici, les écrans et les grandes fonctionnalités.

Rôles

Pour cette application, j'ai examiné les fonctionnalités sous 3 angles de vue différents correspondant à 3 rôles :

- Les utilisateurs finaux : toutes les personnes qui seront amenées à gérer la caisse pendant la compétition.
- Les organisateurs du Grand Prix : les clients pour lesquels cette application est développée afin de leur donner une vue d'ensemble sur les ventes.
- Les consommateurs : les personnes qui viennent passer leurs commandes de tickets.

Écrans et fonctionnalités des users stories

US1 - Installation

En tant qu'utilisateur de l'application, je dois pouvoir installer et lancer facilement l'application à partir de n'importe quel PC.

Fonctionnalités

En préparation au Grand Prix du Judo, l'utilisateur reçoit un lien qui lui permet de télécharger un fichier exécutable sur son PC.

Par un double-clic sur le fichier exécutable, l'application s'ouvre et est totalement autonome.

Le lieu où se déroule le Grand Prix de Judo ne dispose pas d'un wifi stable, il faut donc pouvoir travailler en local tout au long de la journée.

L'utilisateur dispose d'un PC avec Windows 10 et d'un autre avec Linux (Ubuntu 20,04).

Il aimerait que l'application puisse tourner indifféremment sur l'un ou l'autre de ses PCs.

L'idéal serait aussi que l'application puisse être mise sur une tablette.

US2 - Paramétrage

En tant qu'organisateur du Grand Prix de Judo, je dois initialiser la liste des produits afin de visualiser les produits disponibles à la vente.

Écran

Les produits sont matérialisés par des tickets de couleur, la couleur étant fonction du prix.

Sur l'écran Paramétrage des Produits, je dispose de boutons que je peux paramétrer afin qu'ils représentent les caractéristiques des produits.

L'écran Paramétrage des Produits est divisé en 3 zones :

- sur la gauche, je garde le Menu de l'application
- au milieu, la plus grande partie de l'écran, je dispose de 36 boutons paramétrables (6 * 6)
- sur la droite, en haut, j'ai les paramètres du produit que je peux mettre à jour
- sur la droite en bas, j'ai un clavier numérique
- sur la droite, en bas, j'ai un bouton « Enregistrer » et un bouton « Annuler »



Figure 1 : Maquette interface de paramétrage

Fonctionnalités

Je choisis un bouton en cliquant dessus. Je peux introduire ses caractéristiques qui sont :

- la couleur du ticket
- le libellé du produit

- le prix de vente
- une librairie de photos

En cliquant sur le bouton « Enregistrer », le bouton produit est sauvé à la place définie et avec ses caractéristiques.

En cliquant sur le bouton « Annuler », les paramètres introduits ne sont pas enregistrés. Les paramètres sont réinitialisés.

De plus, si un produit ne devait plus être disponible en cours de journée, il faut pouvoir le cacher sans pour autant perdre ses caractéristiques.

US3 - Accueil

En tant qu'utilisateur de l'application, je souhaite qu'elle s'ouvre sur un écran d'Accueil.

Écran

L'écran d'Accueil est composé de 2 zones principales :

- sur la gauche, un Menu de l'application,
- sur le reste de l'écran, une photo du Grand Prix de Judo de la Ville d'Ath (idéalement, la même photo que celle reprise sur les affiches et documents du Grand Prix).



Figure II : Maquette interface accueil

Fonctionnalités

Le Menu de l'application reprend :

- Menu : permet de revenir sur l'écran d'accueil,
- Vente : permet d'aller sur l'écran où enregistrer les ventes des produits,
- Produits : permet d'aller sur l'écran où paramétrer les produits,
- Caisse : permet d'aller sur l'écran où initialiser le fonds de caisse en début de journée,
- Sécurité : permet d'aller sur l'écran où enregistrer les sorties de fonds lors de la mise en sécurité de l'argent
- Rapport : permet d'aller sur l'écran où visualiser et imprimer des rapports

Idéalement, un bouton « Aide » est disponible avec un manuel utilisateur succinct.

US4 – Initialisation de la caisse

En tant qu'utilisateur de l'application, je souhaite pouvoir introduire mon fonds de caisse en début de journée.

Écran

L'écran d'Initialisation de la Caisse est divisé en 3 zones :

- sur la gauche, le Menu de l'application – identique partout dans l'application,
- au milieu, je dispose de boutons représentant les billets et pièces en euros,
- sur la droite, en haut, va s'afficher ligne par ligne le nombre de billet / pièce de chaque type avec le total que cela représente ainsi que le total global,
- sur la droite en bas, j'ai un clavier numérique
- sur la droite, en bas, j'ai un bouton « Enregistrer » et un bouton « Annuler »

INITIALISATION DE LA CAISSE **JUDO CLUB ATHOIS**

Menu
Vente
Produits
Caisse
Sécurité
Rapport

200 € 100 € 50 €
20 € 10 € 5 €
2 € 1 €
0,50 € 0,20 € 0,10 €
0,05 € 0,02 € 0,01 €

5 * 20,00 € = 100,00 €
10 * 10,00 € = 100,00 €
20 * 5,00 € = 100,00 €
25 * 2,00 € = 50,00 €
50 * 1,00 € = 50,00 €
50 * 0,50 € = 25,00 €

TOTAL : 425,00 €

7 8 9
4 5 6
1 2 3
0 ,

Enregistrer
Annuler

Figure III : Maquette interface de fonds de caisse

Fonctionnalités

Pour initialiser le fonds de caisse, l'utilisateur indique le nombre et choisit en cliquant dessus le billet ou la pièce.

Pour indiquer le nombre, l'utilisateur utilise soit le pavé numérique, soit clique autant de fois que nécessaire sur le bouton représentant le billet ou la pièce.

Une ligne correspondant à ce billet / cette pièce s'affiche en haut à droite avec le nombre, la valeur faciale et le total de cette ligne.

Une ligne total global calcule au fur et à mesure le contenu du fonds de caisse. Cela permet à l'utilisateur de vérifier ce qu'il a introduit en le comparant aux montants renseignés sur l'enveloppe reçue avec l'argent du fonds de caisse.

Une fois l'introduction des différents billets et pièces du fonds de caisse vérifiée, l'utilisateur clique sur « Enregistrer » pour sauvegarder le fonds de caisse.

Le bouton « Annuler » ne sauvegarde pas le fonds de caisse et remet les lignes et totaux à zéro.

US5 – Vente de tickets

En tant qu'utilisateur de l'application, je veux enregistrer les différentes ventes de produits tout au long de la journée. Toutes les ventes enregistrées doivent être sauvegardées afin de pouvoir en faire un rapport en fin de journée.

Écran

L'écran de Vente des tickets est divisé en 3 zones :

- sur la gauche, le Menu de l'application – identique partout dans l'application,
- au milieu, je dispose des boutons représentant les produits (définis lors du paramétrage),
- sur la droite, en haut, va s'afficher ligne par ligne le nombre de produits achetés avec le total de la ligne
- dans ce même cadre, j'ai le total global de la vente ainsi que le montant reçu par le consommateur et la différence à lui rendre,
- sur la droite en bas, j'ai un clavier numérique
- sur la droite, en bas, j'ai un bouton « Enregistrer » et un bouton « Annuler »

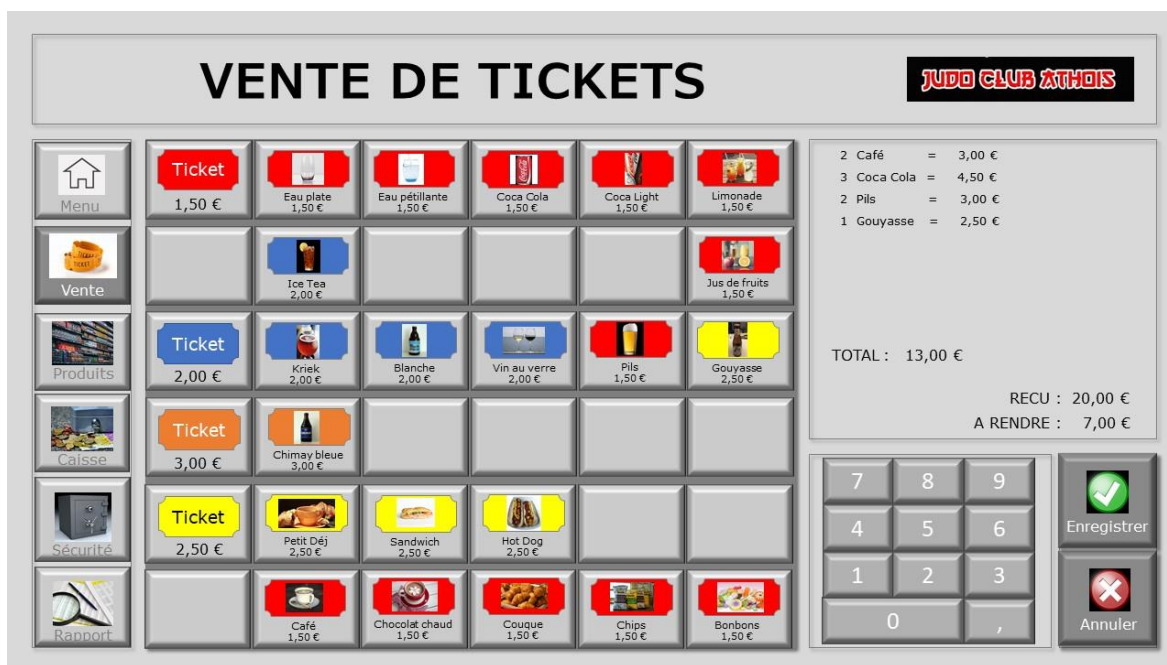


Figure IV : Maquette interface de vente

Fonctionnalités

Lorsqu'un consommateur se présente, j'enregistre sa commande en cliquant sur le bouton correspondant au produit désiré.

S'il s'agit d'une quantité élevée d'un même produit (par exemple un coach d'un club qui achète 20 tickets rouges pour l'ensemble de ses judokas), j'utilise le pavé numérique pour entrer le nombre désiré puis je clique sur le bouton correspondant au produit.

Le total global de la vente des produits est calculé au fur et à mesure de l'introduction de la commande. L'utilisateur peut ainsi annoncer sans risque d'erreur, le montant à payer.

Si la somme perçue par le client ne correspond pas exactement à la somme à payer, l'utilisateur peut introduire la somme reçue et l'application calcule la différence à rendre au consommateur.

Une fois toute la commande introduite, l'utilisateur clique sur « Enregistrer » pour sauvegarder cette commande.

Le bouton « Annuler » ne sauvegarde pas la commande et remet les lignes de produits et les totaux à zéro.

Toutes les commandes sauvegardées tout au long de la journée sont cumulées afin de connaître le nombre total de chaque produit vendu et le montant total que cela représente. Cette fonctionnalité sera utile pour les rapports de fin de journée.

US6 – Achat de tickets

En tant que consommateur, je souhaite ne pas avoir un trop long temps d'attente pour acheter mes tickets boissons / repas. De plus, le montant total de mes achats doit correspondre aux tickets reçus. Si je ne paie pas le prix juste, je veux recevoir la différence correcte entre le montant de mes achats et le montant donné.

Fonctionnalités

Au fur et à mesure que j'énonce ma commande, le vendeur de tickets l'introduit.

Je peux demander soit un certain nombre de tickets d'une couleur, soit commander un ou plusieurs produits précis.

Je ne dois pas suivre un ordre particulier, par exemple en fonction de la couleur.

Si je me trompe dans ma commande, il faut pouvoir l'annuler.

Quand ma commande est passée, je dois connaître le montant exact et correct à payer.

Si je ne dispose pas de la somme exacte, je dois être correctement remboursé de la différence.

Tout cela doit être très fluide et flexible.

US7 – Impression de la vente

En tant que consommateur, je suis parfois amené à demander un ticket de caisse afin de me faire rembourser les montants dépensés. Je peux soit le demander au moment de la commande, soit en fin de journée pour plusieurs commandes.

Fonctionnalités

Une fois ma commande passée, je demande un ticket reprenant le nombre et les produits achetés ainsi que le montant total payé. Ce ticket m'est délivré rapidement.

En fin de journée, je peux venir demander l'impression de mes différentes commandes faites en cours de journée. Ces tickets me sont délivrés rapidement.

US8 – Mise en sécurité

En tant qu'utilisateur de l'application, je veux introduire les mises en sécurité de l'argent disponible en caisse. Cela a pour effet de diminuer l'argent en caisse.

Écran

L'écran de Mise en Sécurité est divisé en 3 zones :

- sur la gauche, le Menu de l'application – identique partout dans l'application,
- au milieu, je dispose de boutons représentant les billets et pièces en euros,
- sur la droite, en haut, va s'afficher la date et l'heure de la mise en sécurité ainsi qu'un numéro séquentiel,
- sur la droite, en haut, je peux indiquer les personnes qui font la mise en sécurité,
- sur la droite, en haut, va s'afficher ligne par ligne le nombre de billet / pièce de chaque type avec le total que cela représente ainsi que le total global,
- sur la droite en bas, j'ai un clavier numérique
- sur la droite, en bas, j'ai un bouton « Enregistrer » et un bouton « Annuler »

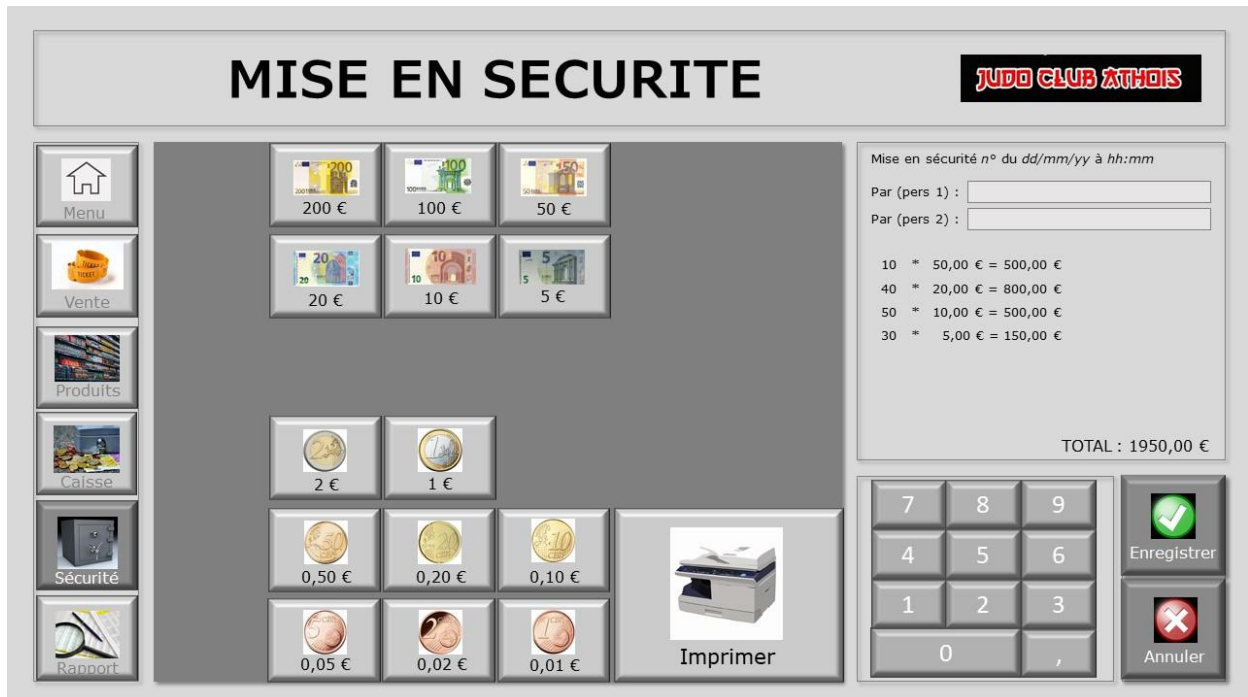


Figure V : Maquette interface de mise en sécurité

Fonctionnalités

Pour réaliser la mise en sécurité, l'utilisateur commence par introduire le nom des 2 personnes qui font cette mise en sécurité.

Ensuite, l'utilisateur indique le nombre et choisit, en cliquant, dessus le billet ou la pièce.

Pour indiquer le nombre, l'utilisateur utilise soit le pavé numérique, soit clique autant de fois que nécessaire sur le bouton représentant le billet ou la pièce.

Une ligne correspondant à ce billet / cette pièce s'affiche en haut à droite avec le nombre, la valeur faciale et le total de cette ligne.

Une ligne total global calcule au fur et à mesure le montant global de la mise en sécurité. Cela permet à l'utilisateur de vérifier l'argent qu'il remet dans l'enveloppe en le comparant au montant total de la mise en sécurité.

Une fois l'introduction des différents billets et pièces de la mise en sécurité vérifiée, l'utilisateur clique sur « Enregistrer » pour sauvegarder la mise en sécurité.

Le bouton « Annuler » ne sauvegarde pas la mise en sécurité et remet les lignes et totaux à zéro.

Un bouton « Imprimer » permet d'imprimer sur papier le contenu et le montant total de la mise en sécurité. Ce papier doit être glissé dans l'enveloppe contenant l'argent.

Une fois la mise en sécurité enregistrée, le montant total de ma caisse diminue du montant mis en sécurité.

US9 – Rapport mise en sécurité

En tant qu'utilisateur de l'application, je veux pouvoir imprimer une ou toutes les mises en sécurité de la journée via l'écran de Rapport.

Écran

L'écran de production de rapports comprend 3 zones :

- sur la gauche, le Menu de l'application – identique partout dans l'application,
- au milieu va s'afficher le résultat du rapport demandé,
- sur la droite, en haut, je dispose de boutons pour générer les différents rapports disponibles,
- sur la droite en bas, j'ai un bouton « Imprimer »
- sur la droite, en bas, j'ai un bouton « Enregistrer » et un bouton « Annuler »

PRODUCTION DES RAPPORTS

JUDO CLUB ATHOIS

Menu

Vente

Produits

Caisse

Sécurité

Rapport

Liste des ventes						
N°	Date	Heure	Nb	Produit	Total Ligne	Total Commande
1	25/04/2020	08:43	2	Petit Dej	5,00 €	
1	25/04/2020	08:43	1	Sandwich	2,50 €	
1	25/4/2020	08:43				7,50 €
2	25/04/2020	08:46	1	Café	1,50 €	1,50 €
3	25/04/2020	08:48	2	Café	3,00 €	
3	25/04/2020	08:48	1	Ice Tea	2,00 €	
3	25/04/2020	08:48				5,00 €
4	25/04/2020	08:51	1	Chocolat chaud	1,50 €	
4	25/04/2020	08:51	2	Café	3,00 €	
4	25/04/2020	08:51	1	Jus de fruits	1,50 €	
4	25/04/2020	08:51				6,00 €
5	25/04/2020	08:55	2	Coca Cola	3,00 €	
5	25/04/2020	08:55	2	Sandwich	5,00 €	
5	25/04/2020	08:55	1	Couque	1,50 €	
5	25/04/2020	08:55				9,50 €
6	25/04/2020	08:56	1	Petit Dej	2,50 €	2,50 €
7	25/04/2020	08:56	2	Jus de fruits	3,00 €	
7	25/04/2020	08:56	2	Petit Dej	5,00 €	
7	25/04/2020	08:56				8,00 €
8	25/04/2020	08:59	1	Eau pétillante	1,50 €	1,50 €

Liste des produits

Liste des ventes

Liste du Fond de caisse

Mises en sécurité

Imprimer

✓

Enregistrer

✗

Annuler

Figure VI : Maquette interface de rapports

Fonctionnalités

En cliquant sur le bouton correspondant au rapport mise en sécurité, l'application génère la liste des mises en sécurité effectuées jusque-là.

Chaque ligne de mise en sécurité reprend le numéro de la mise en sécurité, les dates et heures ainsi que le montant total de la mise en sécurité.

Je peux sélectionner/cliquer une ou plusieurs lignes de mise en sécurité.

En cliquant sur le bouton « Imprimer », les lignes sélectionnées seront imprimées.

En cliquant sur le bouton « Enregistrer », les rapports correspondant aux lignes sélectionnées seront sauvegardés sur le PC.

En cliquant sur le bouton « Annuler », les lignes sélectionnées seront décochées.

Le rapport qui est imprimé reprend pour chaque mise en sécurité :

- le numéro de mise en sécurité,
- les date et heure de mise en sécurité,
- les personnes ayant fait la mise en sécurité,
- le nombre des billets et pièces repris dans la mise en sécurité ainsi que le montant total de la ligne
- le montant global de la mise en sécurité

Pour quitter cet écran, je dispose du Menu qui me permet d'aller sur une autre fonctionnalité de l'application.

US10 – Rapport des ventes de la journée

En tant qu'utilisateur de l'application, je veux pouvoir imprimer toutes les ventes de la journée via l'écran de Rapport.

Fonctionnalités

En cliquant sur le bouton correspondant au rapport liste des ventes, l'application génère la liste de toutes les ventes effectuées jusque-là.

Cette liste est affichée à l'écran.

Chaque ligne reprend

- le numéro de la commande,
- les date et heure auxquelles la commande a été enregistrée,
- le nombre et le libellé du produit commandé ainsi que le total pour cette ligne.

Le total global de la commande est repris sur la dernière ligne de la commande.

En cliquant sur le bouton « Imprimer », la liste des ventes de la journée est imprimée.

En cliquant sur le bouton « Enregistrer », la liste des ventes de la journée est sauvegardée sur le PC.

En cliquant sur le bouton « Annuler », la partie centrale de l'écran est remise à blanc.

Le rapport qui est imprimé reprend chaque ligne affichée à l'écran avec toutes ses données.

De plus, une synthèse globale est effectuée. Elle reprend :

- le nombre total de commandes,
- le nombre total de chaque produit vendu avec son libellé,
- le montant total de la vente pour ce produit,
- le montant total de toutes les commandes de la journée.

Pour quitter cet écran, je dispose du Menu qui me permet d'aller sur une autre fonctionnalité de l'application.

US11 – Imprimer le ticket d'une commande

En tant qu'utilisateur de l'application, je veux pouvoir imprimer le ticket d'une commande via l'écran de Rapport.

Fonctionnalités

En cliquant sur le bouton correspondant au rapport ticket d'une commande, l'application me demande d'entrer le numéro de la commande. Une fois le numéro introduit, l'application génère la liste de tous les produits vendus pour cette commande.

Cette liste est affichée à l'écran.

Chaque ligne reprend

- le numéro de la commande,
- les date et heure auxquelles la commande a été enregistrée,
- le nombre et le libellé du produit commandé ainsi que le total pour cette ligne.

Le total global de la commande est repris sur la dernière ligne de la commande.

En cliquant sur le bouton « Imprimer », le ticket de la commande est imprimé.

En cliquant sur le bouton « Enregistrer », le ticket de la commande est sauvegardé sur le PC.

En cliquant sur le bouton « Annuler », la partie centrale de l'écran est remise à blanc.

Le rapport qui est imprimé reprend chaque ligne affichée à l'écran avec toutes ses données.

Pour quitter cet écran, je dispose du Menu qui me permet d'aller sur une autre fonctionnalité de l'application.

US12 – Rapport fonds de caisse

En tant qu'utilisateur de l'application, je veux pouvoir imprimer le nombre de billets et pièces introduits à l'initialisation via l'écran de Rapport

Fonctionnalités

En cliquant sur le bouton correspondant au rapport fonds de caisse, l'application génère le nombre de billets et pièces introduits lors de l'initialisation du fonds de caisse. Le résultat est affiché à l'écran.

Chaque ligne reprend le montant total pour ce billet / pièce.

Le montant total du fonds de caisse est également affiché.

En cliquant sur le bouton « Imprimer », le rapport est imprimé.

En cliquant sur le bouton « Enregistrer », le rapport est sauvegardé sur le PC.

En cliquant sur le bouton « Annuler », la partie centrale de l'écran est remise à blanc.

Le rapport qui est imprimé reprend :

- le nombre et le libellé des billets / pièces,
- le montant total pour ce billet / pièce,
- le montant global du fonds de caisse.

Pour quitter cet écran, je dispose du Menu qui me permet d'aller sur une autre fonctionnalité de l'application.

Choix techniques

Une fois les besoins bien cernés, j'ai pu commencer à me pencher sur les langages et outils que j'allais utiliser pour mener à bien ce projet. Étant donné que chaque élément dont nous avons parlé pouvait s'apparenter à un objet, je me suis tourné vers un langage orienté objet. Il fallait également un langage qui permette de faire une interface graphique. Les données devant être stockées et manipulées par après, une base de données relationnelle s'avérait nécessaire.

Choix du langage

Afin de réaliser ce projet, un langage permettant de traiter des objets, de faire une interface graphique simple et de traiter une base de données était requis. Plusieurs langages informatiques répondaient à ses contraintes :

- Java
- Python
- Ruby
- C#

Voyons dans un premier temps quelles sont les possibilités, avantages et inconvénients de chacun d'entre eux

Java

C'est le langage orienté objet le plus répandu dans le monde, il permet également de n'être codé qu'une fois pour être utilisé sur différentes plateformes qui supportent le Java (c'est une option émise par le client).

Java est également un langage facile à apprendre et nécessitant une grande rigueur dans son emploi, ce qui limite les erreurs. De plus, lorsqu'une erreur survient, la console d'erreur fournit suffisamment d'informations pour localiser et traiter l'erreur.

Java dispose de son propre 'garbage collector' pour libérer la mémoire utilisée, bien que souvent décrié pour sa consommation élevée de mémoire vive.

Du côté des inconvénients, on retrouve une esthétique sobre pour la GUI et un manque de performance dû à la compilation et au niveau d'abstraction.

Python

Python est un langage open-source interprété de haut niveau qui permet le travail en orienté objet. Il est simple d'utilisation car c'est un langage dynamique qui va s'occuper d'attribuer lui-même le type de variable. Il en ressort un langage légèrement plus lent à l'exécution. Il est portable car un même programme peut être exécuté sur différents systèmes d'exploitation. Sa syntaxe est très littéraire et il est donc simple à apprendre et le code est compréhensible.

Python dispose également d'une grande variété de services via les librairies et paquets ce qui permet de l'utiliser tant pour se connecter aux bases de données que pour créer des interfaces graphiques.

Tout comme Java, il gère lui-même ses ressources et ce n'est pas au programmeur de faire le nettoyage.

Ses avantages sont également ses inconvénients car un langage interprété est plus lent qu'un langage compilé ; un typage dynamique apporte plus d'erreurs lors de l'exécution et sa connectivité avec des bases de données est un peu sous-développé.

Ruby

C'est un langage où tout est considéré comme des objets, cependant il reste essentiellement un langage de script. C'est un langage très verbal et avec une bonne compréhension de l'Anglais, on peut rapidement comprendre ce que fait le code.

Cependant, c'est un langage assez lent et qui ne dispose pas d'IDE pour la réalisation d'interfaces graphiques.

C#

C'est un langage de haut niveau, purement orienté objet. De plus, il dispose et supporte le 'garbage collector'.

Cependant, il ne permet pas de créer des applications autonomes et est essentiellement prévu pour des applications Windows.

Au vu de cette comparaison entre les langages, Java et Python ressortent comme étant les plus appropriés pour ce projet. Étant donné que j'ai déjà travaillé en Java, c'est tout naturellement que je me suis tourné vers cette solution. J'ai décidé d'utiliser la toute dernière version stable du Java SE Development Kit afin de pouvoir bénéficier de toutes les fonctionnalités. De plus, en entreprise, ce sera peut-être mon environnement de travail. De même, l'IDE que j'utilise est Eclipse dans sa version de mars 2021.

Choix de la base de données

MySQL

Dans un premier temps, j'ai créé une base de données en MySQL via PHPMyAdmin. J'ai choisi de prendre une base de données MySQL car c'est une solution que j'ai utilisée lors de tous mes projets au sein de l'EPHEC et qui ne m'avait jamais posé de gros problèmes. De plus, d'un point de vue plus technique, MySQL reste l'un des gestionnaires de bases de données relationnelles les plus utilisés. En effet, il est fiable et propose des performances élevées. La communauté MySQL étant très

grande, il est également facile de trouver des réponses en cas de questions ou de l'aide pour lier mon code Java à ma base de données.

HyperSQL

Pendant le développement de mon application, il m'est apparu que la base de données devait être standalone et non plus lancée via un autre utilitaire. J'ai donc opté pour un gestionnaire de base de données codé en Java, HyperSQL¹. Aussi connu sous l'acronyme HSQLDB, cette solution me permet de lancer ma base de données depuis un fichier et de tout sauvegarder à l'intérieur de celui-ci sur l'ordinateur de mon client. Un atout de HyperSQL est qu'il est sous licence libre ce qui permet de l'utiliser sans restriction. HyperSQL ne fournit pas qu'une liaison vers la base de données, il permet également de se connecter à des bases de données locales ou externes.

¹ <http://hsqldb.org/>

Conception

DBHelper

En amont de l'implémentation de l'interface de Vente, afin de pouvoir accéder aux données stockées dans la base de données, il me fallait créer cette base de données et une classe qui puisse faire le lien. C'est le rôle de DBHelper.

Déroulement

Dans un premier temps, une base de données en MySQL a été créée via PHPMyAdmin. Cette base de données devait être lancée via XAMPP et tournait comme si elle était sur un serveur.

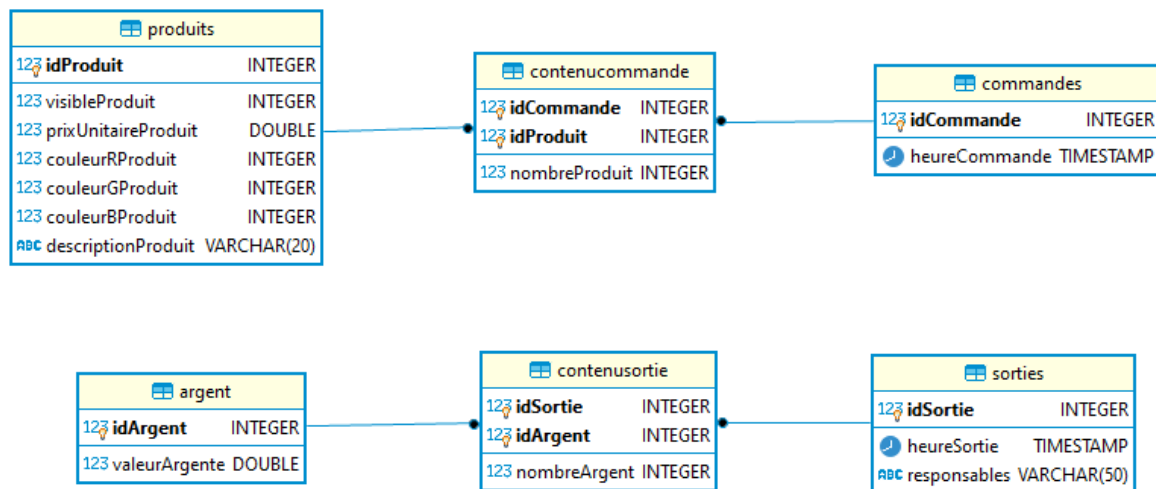


Figure VII : Schéma relationnel de base de données

Le schéma de la base de données comporte six tables. Trois sont consacrées aux produits, commandes et leurs contenus ; les trois autres sont consacrées à l'argent, les mises en sécurité, le fonds de caisse et leurs contenus. Afin d'éviter des erreurs d'introduction, des clés étrangères ont été définies entre les différentes tables.

Pour me connecter à cette base de données, j'ai utilisé les drivers MySQL² pour Java. Tout au long du développement de l'interface de Vente, c'était cette base de données distante qui était utilisée. Chaque fonction crée sa connexion vers la base de données puis exécute la requête sur celle-ci. Les résultats sont ensuite retournés et stockés dans l'objet qui avait fait appel à la fonction.

Problème : La base de données doit être lancée de manière distante et n'est pas intégrée à l'application.

Solution : Utiliser un gestionnaire de base de données écrit en Java.

² mysql-connector-java-8.0.18.jar

Pour ce faire, j'ai utilisé HyperSQL (HSQLDB) qui me permet de bénéficier d'une base de données en local sur l'ordinateur de la personne qui lance l'application. En effet, HSQLDB permet de se connecter à plusieurs types de bases de données et permet également d'en héberger une. Les données que je stockais devant être pérennes, j'ai choisi d'utiliser la solution de type "file" pour héberger et stocker ma base de données. Avec cette solution, HSQLDB va lancer en mémoire RAM la base de données sur base d'un fichier script préétabli ou sur base du fichier script comprenant les dernières modifications. À la fin de chaque requête, celle-ci est stockée en mémoire RAM et est également ajoutée au fichier contenant le script. Afin de ne pas avoir besoin de droits d'accès en écriture, le fichier script contenant la base de données est stocké, pour les utilisateurs Windows dans « C:\Utilisateurs\Public\posjudo\ » et pour les utilisateurs Linux dans les dossiers personnels.

Lorsque l'utilisateur voulait fermer le programme via la croix de fermeture, en haut à droite, la base de données ne se fermait pas correctement. En effet, pour éviter que des actions concurrentes ne se passent sur la base de données et endommagent celle-ci, HSQLDB crée un fichier lock pour signifier que la base est déjà utilisée par un processus. Pour pallier à ce problème, il faut fermer proprement la connexion à la base de données en exécutant la commande "SHUTDOWN ;". C'est pour cette raison que la croix de fermeture de la fenêtre a été désactivée et que pour quitter le programme, il faut appuyer sur le bouton "Quitter".

De plus, chaque fois que l'utilisateur clique sur le bouton "Quitter", une copie de la base de données dont le nom contient un timestamp est écrite. Ce qui permettra de revenir à une version antérieure en cas de nécessité.

Suite à la dernière démonstration et discussion avec le client, celui-ci souhaitant pouvoir voir à tout moment le montant de l'argent en caisse, j'ai ajouté une table mémoire qui contient une seule valeur qui est incrémentée à chaque "Vente" ou "Fonds de caisse" et décrétementée à chaque "Mise en sécurité". Pour changer la valeur de cette information, les fonctions setTotal(double d) et getTotal() sont utilisées.

Conclusion

Au travers de l'implémentation de ma solution, le type de base de données et la manière dont elle est stockée ont dû évoluer afin de pouvoir répondre au mieux à la demande de mon client : une application portable sur n'importe quel ordinateur.

Interface Vente

Afin de pouvoir répondre au plus vite aux besoins de mon client, j'ai commencé mon développement par la vue « Vente ».

Cette vue est la plus importante pour mon client car c'est celle qui lui permet d'introduire les différentes commandes dans la base de données afin d'avoir une idée des quantités de tickets vendus. Si le Grand Prix de Judo avait pu avoir lieu cette année (24-04-2021), cela aurait permis à mon client de bénéficier d'une première version minimale de ses besoins.

Déroulement

D'abord, je me suis attaqué à l'interface graphique de la vue Vente. Pour ce faire, je me suis basé sur les maquettes qui avaient été créées avec le client.

Problème : N'ayant que des notions limitées en design d'application en Java, l'agencement des différents menus et boutons ne fut pas simple.

Solution : Afin de pallier à ce problème, je me suis aidé de NetBeans³ qui permet de faire des interfaces graphiques en Java Swing en créant et déplaçant des éléments "préfabriqués". Par la suite, j'ai retravaillé le code par moi-même afin d'avoir une interface qui ressemblait au mieux à ce qui avait été convenu avec le client.



Figure VIII : Maquette interface de vente du client

³ <https://netbeans.apache.org/>

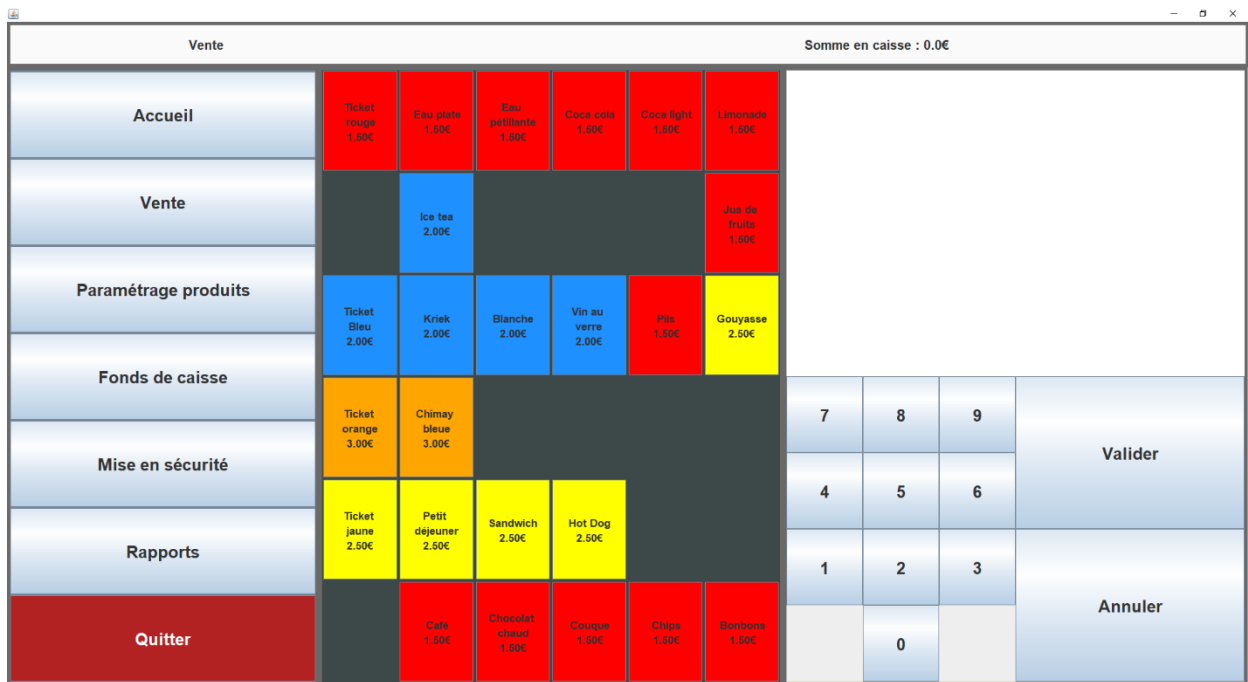


Figure IX : Capture d'écran de l'interface de vente finale

Lors de la création de mon interface graphique et afin de coller au mieux à la maquette, j'ai utilisé et testé plusieurs manières d'agencer mes différents panneaux (JPanel). La première manière appliquée pour diviser le menu latéral du corps de l'application est un BorderLayout. Celui-ci me permet de donner des dimensions minimales, maximales et préférées pour mes composants. Cependant cela réduit la portabilité de mon application sur des écrans plus petits.

Pour les sept boutons du menu latéral et tout l'intérieur du panneau de droite, j'utilise un GridLayout. Cette solution me permet de définir la taille de la grille de manière verticale et horizontale ainsi que la bordure entre les différents éléments. Cependant, tous les éléments auront la même taille. C'est pourquoi le pavé numérique et les boutons de Validation/Annulation peuvent sembler très grands par rapport aux dimensions des boutons de produits.

Une fois que mon interface graphique était créée, je me suis attaqué au contrôleur de la vue ainsi qu'au modèle sur lequel j'allais interagir. J'ai donc créé ces deux classes avec leurs constructeurs appropriés.

Le contrôleur prend en paramètre le bouton qui vient d'être pressé, la vente qui est en cours ainsi que, dans la suite du code, le tableau qui synthétise la commande.

Le modèle quant à lui se construit sur base d'informations pré-calculées comme l'ID de la vente qui est récupérée dans la base de données, l'heure du système pour horodater la vente et le contenu qui est une liste d'objets Produits dont le nombre de ventes est initialisé à zéro.

J'ai choisi d'initialiser par défaut le contenu de la commande avec une liste contenant tous les Produits plutôt que d'ajouter les produits vendus au fur et à mesure. Ce choix s'est fait car le temps nécessaire pour l'initialisation de la Liste avec tous les éléments à zéro est plus rapide que l'ajout incrémental des éléments. En effet, la seconde méthode implique de vérifier d'abord si le produit

est déjà dans la liste et l'incrémenter le cas échéant et sinon de l'ajouter à la liste. D'un autre côté, cette seconde solution est moins gourmande en mémoire vive car il y a moins d'objets stockés. La demande du client est de privilégier la vitesse d'encodage d'une commande plutôt que d'optimiser les ressources.

Dans le cas présent, les deux solutions n'ont qu'un impact mineur sur le temps d'exécution ou la quantité de mémoire vive utilisée. Si la liste de Produits s'avérait être beaucoup plus grande, un travail avec des structures chaînées et ordonnées permettrait d'avoir à la fois un temps de calcul moindre et une consommation de mémoire réduite.

Problème : L'objet Produit n'existe pas encore

Solution : Pour répondre au besoin du modèle Vente, j'ai créé l'objet Produit, d'abord de manière statique et ensuite de manière dynamique ; cet objet sera renommé par la suite en Paramétrage Produits.

Afin de pouvoir initialiser la Liste des Produits, on va créer la fonction `getProduits` dans la classe `DBHelper`. Une fois la Liste des Produits récupérée et initialisée à 0 produits vendus, la vente peut commencer.

L'utilisateur dispose de deux manières pour introduire le nombre de produits vendus :

- Soit x clics sur le bouton correspondant au produit.
- Soit utiliser le pavé numérique pour donner le nombre puis cliquer sur le produit souhaité.

Dans un premier temps, un `ActionListener` a été ajouté sur chaque bouton du panneau central. Le nom du produit, la vente en cours et le tableau de synthèse étaient envoyés vers le contrôleur. Le contrôleur, en fonction du nom du bouton cliqué pouvait déterminer l'action à effectuer.

Les deux actions possibles étaient soit d'ajouter le bouton cliqué au contenu de la vente, soit de valider ou annuler celle-ci.

Si la vente est validée, je sauvegarde les données de la vente dans la base de données au moyen de la fonction « `enregistrerCommande` ». Si le bouton cliqué était un bouton de ticket, j'incrmente l'attribut `vendu` du produit correspondant.

Si la vente est annulée, on revient à l'écran de vente et la liste des produits vendus est réinitialisée à zéro.

Par la suite, j'ai ajouté la prise en compte du pavé numérique en écoutant celui-ci et en stockant la valeur encodée au fur et à mesure dans une variable temporaire. Lorsqu'un bouton ticket était cliqué, j'incrémentais l'attribut `vendu` du produit correspondant à la valeur enregistrée.

De plus, suite à la dernière réunion avec mon client vers la fin du projet, il a demandé de connaître à tout moment le montant de l'argent en caisse. J'ai implémenté une fonction qui va faire la somme de la commande et ajouter celle-ci dans la mémoire via le `setTotal(double d)` et en faire l'affichage à l'écran.

Une fois que tout cela fonctionnait correctement, je me suis attaqué au tableau récapitulatif de la commande. Pour ce faire, j'utilise un `DefaultModelTable` qui va contenir les données à afficher et je le passe en argument à ma `JTable`. Ce `DefaultModelTable` est construit en lui injectant les données sous forme d'un tableau d'objets à deux dimensions. Pour connaître la taille en lignes, à donner à ce tableau, je fais appel à une fonction `getRow()` qui va calculer combien de produits ont un nombre de vente supérieur à zéro ; pour le nombre de colonnes c'est une valeur fixe. Une fois mon tableau à deux dimensions créé, il faut le remplir. Pour ce faire, je réalise les premières lignes à la main pour que la première contienne l'ID de la vente ainsi que l'horodatage, la seconde ligne est une ligne blanche de séparation et la troisième contient les titres des colonnes. Les lignes suivantes sont remplies via une boucle `forEach` qui va parcourir le contenu de la commande et n'afficher que les lignes qui ont des produits vendus. Pour finir ce tableau récapitulatif, j'affiche le total de tickets à donner ainsi que la somme totale de la commande qui sont calculés via des fonctions séparées. Ces fonctions sont testables via des tests unitaires `JUnit`.

Conclusion

Cette première interface fut l'une des plus compliquées et la plus longue à implémenter mais elle allait servir de base pour la suite du programme et le code qu'elle contenait devait être très rigoureux et correct pour pouvoir être ré-utilisé par après.

Interface Fonds de caisse

Une fois mon interface de Vente réalisée, ce sont les interfaces de "Fonds de caisse" et des "Mises en sécurité" qui devenaient les éléments les plus importants. J'ai décidé de commencer par l'interface "Fonds de caisse" car son implémentation allait pouvoir être réutilisée pour l'interface de "Mise en sécurité". Je me suis également basé sur le code de l'interface de "Vente" pour tout ce qui a trait à la vue.

Déroulement

Dans un premier temps, j'ai dupliqué la vue "Vente" pour la modifier et la faire correspondre à la maquette réalisée avec le client. J'ai commencé par changer les différents titres et valeurs des boutons. Tout comme pour l'interface de "Vente", j'ai commencé en hardcodant les valeurs de monnaies les plus couramment utilisées pour les boutons.

Une fois cela effectué, j'ai mis en place le contrôleur et le modèle du "Fonds de caisse". Tout comme la vente, un fonds de caisse dispose d'un ID et d'un horodatage. Cependant, les objets échangés ne sont plus des produits mais de l'argent, il m'a donc fallu créer cet objet.

L'objet "Argent" dispose de trois attributs qui sont son ID en base de données, sa valeur en euros et le nombre de fois qu'il a été introduit dans la caisse. Son constructeur ne comporte cependant que deux paramètres car par défaut le nombre de fois introduit est nul.

Une fois mon objet "Argent" créé, j'ai pu reprendre mon travail sur mon objet "Fonds de caisse" en utilisant celui-ci dans une liste. Cette fois-ci, le choix d'utiliser une liste de tous les objets "Argent" initialisés à zéro se justifie par la présentation demandée pour les rapports. En effet, sur le rapport,

même si aucun “Argent” d’un type n’a été introduit, sa ligne doit être présente afin d’avoir une uniformité dans les impressions de rapport.

Le résumé de ce qui est introduit au fur et à mesure dans la caisse est également représenté sous forme d’une JTable. Tout comme pour la synthèse de la vente, j’emploie un DefaultTableModel que j’alimente avec mon double tableau de données. A l’instar de la vente, toutes les lignes sont affichées même si aucune transaction de ce type n’a été effectuée. Cela me permet de me passer de la fonction comptant les lignes nécessaires pour la taille de mon double tableau. En effet, le nombre de formes que l’argent peut prendre est un nombre fixe.

Une fonction de calcul de la somme du fonds de caisse a également été implémentée, et est testable par des tests unitaires, afin de pouvoir en ajouter le montant à la variable en mémoire.

Conclusion

Cette interface fut moins compliquée que la précédente étant donné le ré-emploi de la vue “Vente”. Cependant, il a fallu que je crée l’objet “Argent” afin de pouvoir le stocker dans ma liste.

Interface Mise en sécurité

Ce fut l’interface la plus simple à réaliser car elle reprenait l’implémentation de l’interface “Fonds de caisse”.

Déroulement

J’ai repris l’implémentation de l’interface “Fonds de caisse”, puis j’ai fait les changements de nom des différents titres.

Les petites différences de cette interface vont se situer dans le code. En effet, lors de la mise à jour de la somme en caisse, je ne dois pas ajouter mais soustraire celle-ci de la somme en mémoire.

Aux derniers tests avec les utilisateurs finaux, j’ai regardé pour ajouter les noms des responsables de la mise en sécurité.

Conclusion

Comme annoncé, ce fut la vue la plus simple car elle reprenait la majorité de ses composants dans la vue fonds de caisse.

Interface Paramétrage produits

L’interface de paramétrage est le troisième point qui importe le plus aux organisateurs car au sein des logiciels qui avaient été testés jusque-là, ce paramétrage était long et complexe. Je me suis

donc tourné vers une version de paramétrage des produits la plus simple possible pour permettre à n'importe qui de le réaliser. J'ai donc isolé les quatre éléments importants pour ce paramétrage qui sont le prix, la description, la couleur du ticket et la possibilité de rendre le bouton visible ou non. Effectivement si un produit n'est plus disponible en cours de journée, le rendre invisible permet que les consommateurs en soient directement avertis lorsqu'ils passent une commande.

Déroulement

Afin de commencer, j'ai repris le menu latéral et la disposition du panneau des produits. Cependant, je ne pouvais plus hardcoder les valeurs mais je devais les chercher directement dans la base de données. Afin de limiter les requêtes vers la base de données, je n'en effectue qu'une seule et je crée une liste de Produits avec tous leurs paramètres. Afin de pouvoir faire le lien entre un bouton et un produit, le nom du bouton comprend l'ID du produit. J'assigne ensuite à chaque bouton les paramètres qui lui sont propres.

Lorsque l'on clique sur un bouton, les paramètres de celui-ci sont affichés dans le panneau de paramétrage sur la droite. On peut y retrouver l'ID du produit, sa description, son prix unitaire, les trois valeurs de la couleur RGB ainsi que la possibilité de rendre le bouton visible ou non. Dans un premier temps, pour uniformiser la vue, chaque paramètre est un champ de texte éditable. Quand on clique sur un bouton, ce sont les informations de ce bouton qui sont affichées, si le bouton n'a jamais été paramétré, les valeurs sont vides.

Lorsque l'utilisateur valide les modifications effectuées sur un bouton, les nouvelles valeurs sont récupérées dans les différents champs pour modifier le produit dans la liste et écrire cette modification dans la base de données. Les modifications sont immédiatement visibles à l'écran et à chaque lancement du programme, ce sera toujours la dernière version du tableau des produits qui sera chargée. En cas d'annulation des modifications, l'affichage se remet à zéro et rien n'est modifié en base de données.

Lors de la dernière démonstration, la manière d'encoder la couleur du bouton ne convenait pas entièrement au client car celui-ci ne connaît pas forcément les codes RGB des couleurs qu'il souhaite. Afin de pallier à ce problème, un tableau comprenant une légende avec quelques couleurs de base a été ajouté à côté des boutons de validation et d'annulation. Dans l'idéal, il aurait fallu avoir une palette de couleurs cliquable (cfr Pistes d'amélioration).

Conclusion

C'est une interface de base très importante pour garantir l'exactitude de la comptabilité car il faut que les données qui y sont introduites correspondent à la réalité du terrain. En simplifiant ce travail via une interface épurée et ne reprenant que les éléments indispensables, on diminue le risque d'erreurs lors du paramétrage.

Interface Accueil

L'interface du menu Accueil est l'une des dernières que j'ai réalisée. En effet, celle-ci permettait surtout d'afficher le menu latéral pour se rendre dans les différentes interfaces ainsi que le logo du Judo Club Athois.

Déroulement

Dans un premier temps, j'ai récupéré le menu latéral car celui-ci ne change pas. J'ai ensuite essayé de mettre l'image du Judo Club Athois dans le panneau de droite. Je suis arrivé à implémenter un code qui permet de lire une image située sur le disque dur de l'utilisateur.

Problème : Le temps nécessaire à la lecture et l'affichage du logo était de plus de 10 secondes. Cela me semblait inacceptable de devoir attendre aussi longtemps pour afficher l'image.

Solution : J'ai donc décidé de ne pas mettre l'image du Judo Club Athois mais plutôt un manuel succinct pour l'utilisateur afin que celui-ci puisse prendre rapidement en main l'application.

Ce manuel est stocké sur un JButton car ce type de bouton autorise d'afficher du texte en HTML, ce qui me permet de mettre en forme facilement les textes. J'ai également testé des JTextPane avec un setContentType en HTML mais le rendu ne correspondait pas à ce que je souhaitais. De plus, les JTextPane ne prennent que la place dont ils ont besoin et n'occupent pas donc pas tout le panneau.

Suite à des recherches supplémentaires⁴, je suis arrivé à insérer le fond d'écran d'accueil voulu par le client. Afin de garder le manuel succinct, j'ai modifié la photo afin qu'on y retrouve et le logo du Judo Club Athois et ce manuel en dessous. Cette fois-ci la photo n'est plus sauvée dans les fichiers de l'utilisateur mais directement dans le package Accueil, cela permet d'aller la rechercher beaucoup plus rapidement étant donné que son affichage est désormais immédiat.

Conclusion

Ces choix rendent mon application dédiée au Judo Club Athois. La diffusion éventuelle de l'application à d'autres clubs amis devra prévoir de changer la page d'Accueil.

Interface Rapports

C'est la dernière interface sur laquelle j'ai travaillé et celle qui me pose encore des problèmes. En effet, mon client souhaite pouvoir imprimer les différents rapports afin de pouvoir fournir des tickets de caisse aux consommateurs, joindre les rapports de mise en sécurité, imprimer le fonds de caisse reçu ainsi que la liste des produits vendus.

⁴ <https://memorynotfound.com/java-read-image-file-url-classpath-inputstream/>

Déroulement

Tout comme toutes les autres interfaces, le menu latéral ne change pas.

La vue centrale se compose d'une grande JTable qui va être alimentée par mon contrôleur en fonction des choix de rapport de l'utilisateur.

Le premier rapport que j'ai produit est le rapport global des ventes car celui-ci est le plus important en fin de journée pour savoir quels sont les totaux des différentes ventes. Cela permet aux organisateurs de prévoir les quantités nécessaires pour l'année suivante.

Afin de répondre au mieux aux demandes de rapports complémentaires, j'ai ajouté des champs éditables à côté des boutons de choix de rapport afin de pouvoir y encoder la date de la journée pour le rapport global des ventes, l'ID de la mise en sécurité ou du fonds de caisse ou du ticket. Quand on clique sur le bouton pour avoir le rapport, je récupère les informations liées et je les passe en argument lors de ma requête vers la base de données. La base de données va me créer un objet fonds de caisse, mise en sécurité ou commande sur base des informations, puis je vais pouvoir utiliser les fonctions d'affichages créées pour chacun de ces objets pour l'afficher dans le tableau de rapport.

Problème : Les constructeurs de mes objets étaient sans paramètres car l'initialisation se faisait sur base de valeurs précalculées ou nulles.

Solution : J'ai ajouté des constructeurs pour chacun de mes objets. Ces constructeurs prennent en paramètre les valeurs données, qui étaient récupérées grâce aux fonctions dans DBHelper.

La partie impression des rapports m'a posé beaucoup de soucis. Je n'ai en effet pas trouvé de méthodes permettant d'imprimer une table directement. À la suite de recherches, j'ai trouvé une solution temporaire mais qui n'est pas issue de ma réflexion. Dans un premier temps, le tableau rapport est transformé en une image grâce au code de `co2_gaz`⁵. Par la suite, cette image est imprimée⁶ sur l'imprimante configurée par défaut sur le système d'exploitation de l'utilisateur. Cette impression ne me plaisant pas, j'ai désactivé cette fonctionnalité.

En approfondissant mes recherches sur ce sujet, j'ai découvert l'existence de la librairie "iText" qui permet de créer des PDF personnalisés. Je pourrai approfondir cette piste afin de fournir des tickets propres au Judo Club Atois et qui reprendraient les données sous forme d'un tableau plus élégant.

Conclusion

L'interface de rapport permet de visualiser tous les rapports désirés mais ne permet malheureusement pas de les imprimer. Le client est donc déçu de ne pas pouvoir imprimer directement depuis l'application, mais de devoir passer par des captures d'écrans. Un travail sur

⁵ <https://codes-sources.commentcamarche.net/forum/affich-1162743-conversion-d-un-jpanel-en-pdf>

⁶ <https://stackoverflow.com/questions/10479621/how-to-print-image-in-java>

cette problématique est à envisager afin de corriger cela pour la prochaine édition du Grand Prix de Judo.

Sécurité

Pour l'analyse de sécurité, je vais examiner quelles sont les différentes données présentes dans mon application, quel est l'impact de la perte, du vol de ces données ainsi que la probabilité que cela arrive. Ensuite, en me basant sur le tableau de calcul des risques, j'évaluerai les risques encourus par les données.

IMPACT	IMPORTANT (7 à 9)	Moyenne	Importante	Critique
	MOYEN (4 à 6)	Faible	Moyenne	Importante
	FAIBLE (1 à 3)	Information	Faible	Moyenne
		FAIBLE (1 à 3)	MOYENNE (4 à 6)	IMPORTANTE (7 à 9)
		PROBABILITE		




Figure X : Tableau de calcul de risque

7

Les données qui sont stockées dans l'application quand on reçoit l'exécutable sont le prix, la description et la couleur des tickets du menu du Grand Prix de Judo de 2019. Ces données n'ont aucune valeur étant donné que ces menus sont visibles de tous.

Pendant l'événement, des données sur les ventes sont encodées. Celles-ci ne stockent que l'ID de la vente, son heure et son contenu, aucune donnée à caractère sensible et relative aux consommateurs n'est sauvegardée. De plus, ces données de vente sont sauvegardées en local sur l'ordinateur servant à la caisse et ne sont accessibles que par le propriétaire de l'ordinateur.

Les données relatives aux mises en sécurité sont jugées plus importantes surtout le nom des responsables de celles-ci. Cette donnée est stockée en clair et pourrait porter préjudice en cas de tentative d'escroquerie. La probabilité d'une escroquerie étant faible, l'analyse de sécurité en ressort un risque faible.

La somme en caisse est aussi stockée dans la base de données mais également visible directement sur l'écran de l'ordinateur. C'est une donnée qui pourrait amener les personnes à voler la caisse si le montant affiché est élevé. Cependant le vol de la caisse ne dépend pas de la sécurité

⁷ <https://www.vaadata.com/blog/wp-content/uploads/2018/02/Criticite%C3%A9-VF-1024x512.jpg>

de mon application mais de la sécurité de l'évènement. Les tentatives de vol étant nulles jusqu'à présent, on peut considérer la probabilité comme faible et donc le risque global est moyen.

Comme dit précédemment, toutes les données sont stockées en local sur l'ordinateur de la personne exécutant le programme. Cela permet d'empêcher toute personne extérieure de capter un éventuel flux de données. En cas d'ordinateur non sécurisé et volé, ce qui est peu probable, il est possible de retracer tout l'historique des ventes et des mises en sécurité. Ces données ne comportent qu'un faible intérêt mais leur perte pourrait impacter les statistiques pour les futurs Grand Prix de Judo. On peut donc qualifier cette perte d'une faille de sécurité moyenne. Une bonne solution serait de mettre en place des copies de la base de données sur d'autres supports.

Un dernier point de sécurité est l'exactitude des données encodées. Actuellement, seules les mises en sécurité et fonds de caisse requièrent la présence de deux personnes de l'organisation afin d'éviter toute tentative de vol. Cependant les noms encodés lors de ces actions peuvent l'être par n'importe qui. Un système de log in et de mot de passe pourrait certifier l'auteur des données encodées. Grâce au planning des bénévoles on peut déterminer qui tenait la caisse lors de ces différentes actions. La probabilité d'une fraude n'est pas nulle et les conséquences peuvent être graves, ce qui implique un risque moyen.

Conclusion

On peut conclure que le risque global est compris entre nul et moyen.

La bonne utilisation de l'ordinateur et son bon paramétrage, mise en veille automatique après x minutes d'inactivité, présence d'un mot de passe fort, ... sont les meilleurs garants de la sécurité des informations.

Les organisateurs font confiance aux personnes responsables de caisse pour la bonne gestion de celle-ci.

Tests

Tests unitaires

Afin d'assurer que la somme calculée pour chacune des commandes, mises en sécurité ou fonds de caisse soit correcte, des tests unitaires ont été implémentés. Chaque fonction de somme sera testée selon 3 cas :

- Toutes les valeurs sont nulles, la somme sera égale à 0
- Des valeurs quelconques, la somme sera différente de 0 et égale au total calculé à la main
- Des valeurs quelconques mélangées avec des valeurs nulles, la somme sera différente de 0 et égale au total calculé à la main

De plus, des tests sur le nombre total de tickets d'une commande, ainsi que sur le nombre de lignes à afficher, ont été effectués selon les mêmes trois cas.

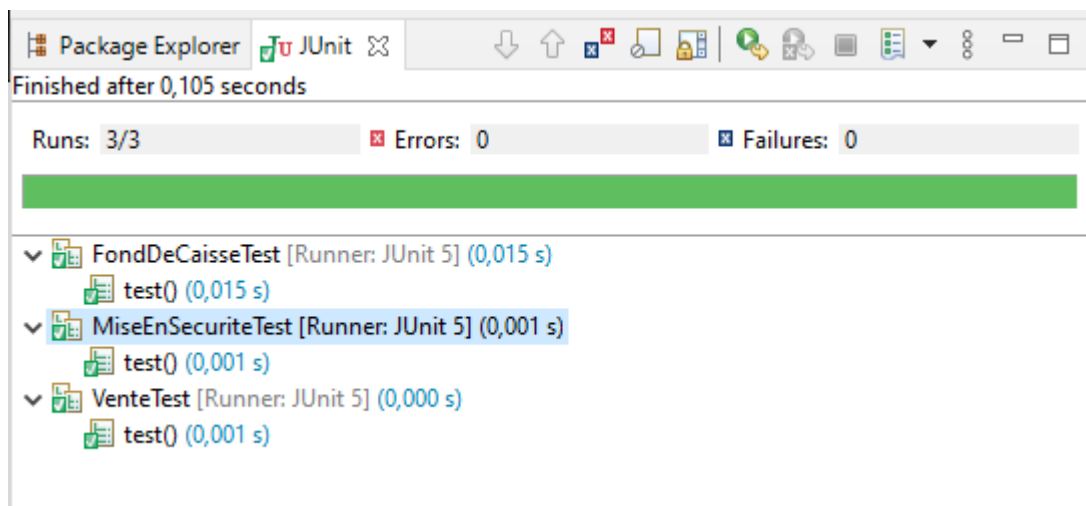


Figure XI : Résultat des tests unitaires

Tests d'intégration par le client

Lors des quelques démonstrations que j'ai pu faire aux clients et aux utilisateurs finaux, ceux-ci m'ont fait remarquer certains manquements ou demandé certaines modifications de traitement des données où j'avais pris une option différente.

On peut notamment évoquer mon choix d'initialiser la somme en caisse à 0 au lancement du programme, cependant le client m'a fait remarquer qu'en cas de panne de courant ou de fermeture du programme, il ne comptait pas réintroduire le fonds de caisse. Désormais, la somme en caisse n'est jamais réinitialisée, cela permet également à la fin de la journée, lors de la dernière mise en sécurité de s'assurer que tout l'argent a bien été sécurisé.

Conclusion

Analyse personnelle

Au cours des différentes phases de ce projet, ma motivation a fortement évolué. En effet, au moment de choisir le sujet de mon TFE, j'étais très enchanté et motivé à l'idée de pouvoir répondre aux besoins de mon club de judo. Tout au long de la phase préparatoire, jusqu'à la défense technique, écouter et prendre note des besoins de mon client me permettaient déjà d'imaginer comment j'allais m'y prendre.

Cependant, une fois que mon stage en entreprise a débuté, je ne trouvais plus le temps d'avancer sur mon TFE car n'avoir que quelques heures à y consacrer le soir était compliqué pour pouvoir bien avancer. Je n'avancais pas aussi vite que je le souhaitais. Le week-end, je disposais d'un peu de temps pour me consacrer aux différentes interfaces.

Une fois mon stage fini et son rapport bouclé, j'ai pu donner un grand coup d'accélérateur car je pouvais consacrer toute mon attention au développement. C'est à ce moment-là que la majorité des problèmes ont pu trouver une solution.

Maintenant que j'estime avoir une version suffisamment aboutie de mon programme pour la fournir à mon client en tant que première version du produit fini, je me sens soulagé et fier de voir qu'il peut faire tout ce qui m'a été demandé. Il reste quelques points à améliorer et qui sont développés dans l'analyse critique.

Analyse critique

Le programme que je fournis à mon client correspond à ses besoins et surtout corrige les différents problèmes qu'il rencontrait avec les anciennes solutions qu'il utilisait.

En effet, comparé à SmartCaisse⁸, le logiciel dont on utilisait une version démo, cette fois-ci mon client ne doit plus tout reparamétrer d'une année à l'autre mais uniquement faire les deux ou trois changements sur les produits en fonction du nouveau menu élaboré ou d'un changement de prix.

De même, comparé à Chromis⁹, le logiciel freeware qu'il avait vite abandonné, mon logiciel dispose d'une interface beaucoup plus sobre et moins complexe. On peut dès l'accueil retrouver un guide d'utilisation rapide pour aider les utilisateurs finaux à bien démarrer.

D'un point de vue plus technique, les différentes fonctionnalités demandées dans les User Stories ont été implémentées à 95%. Le layout des écrans/vues a été adapté suite à des contraintes techniques, mais on y retrouve tous les éléments souhaités. Le client a fait part de ses remarques sur certains éléments mais a accepté certaines modifications suite à des contraintes techniques.

Cette application respecte les contraintes techniques les plus avancées demandées par l'organisateur car son utilisation peut se faire tant sur un ordinateur Windows10 que sur un

⁸ <https://www.smartcaisse.fr/demo.html>

⁹ <https://chromis.co.uk/chromispos/>

ordinateur en Linux. Le logiciel dispose de sa propre base de données qui est sauvegardée en local et qui pourra permettre d'affiner les statistiques aux besoins. De plus, sa taille relativement légère permet de le stocker sur une clé USB et d'être transférable par mail vers d'autres clubs amis.

Piste d'améliorations

Importance élevée

- Développer l'impression de tous les rapports : fonds de caisse, mise en sécurité, ticket d'une commande et rapport des ventes. Dans les versions actuelles, les rapports sont visibles à l'écran.

Importance moyenne

- Calcul de la somme à rendre en fonction de la somme reçue juste après la validation d'une commande.
- Dans la commande en cours, développer la possibilité de changer le nombre de produits commandés dans une ligne.

Importance faible

- Ajout d'une icône (image) de chaque produit
- Ajout d'une icône d'imprimante sur le bouton « Imprimer »
- Tableau des couleurs plutôt que le code RGB
- Remplacement du « oui / non » par un switch
- Déplacement d'un bouton de produit par « drag & drop » dans l'interface de Paramétrage de produits
- Forme déroulante du tableau de synthèse des ventes en cas de vente de 22 produits différents ou plus (cas très rare)

Annexes

Annexe I : Autres schémas liés à la base de données

Schéma entités-associations

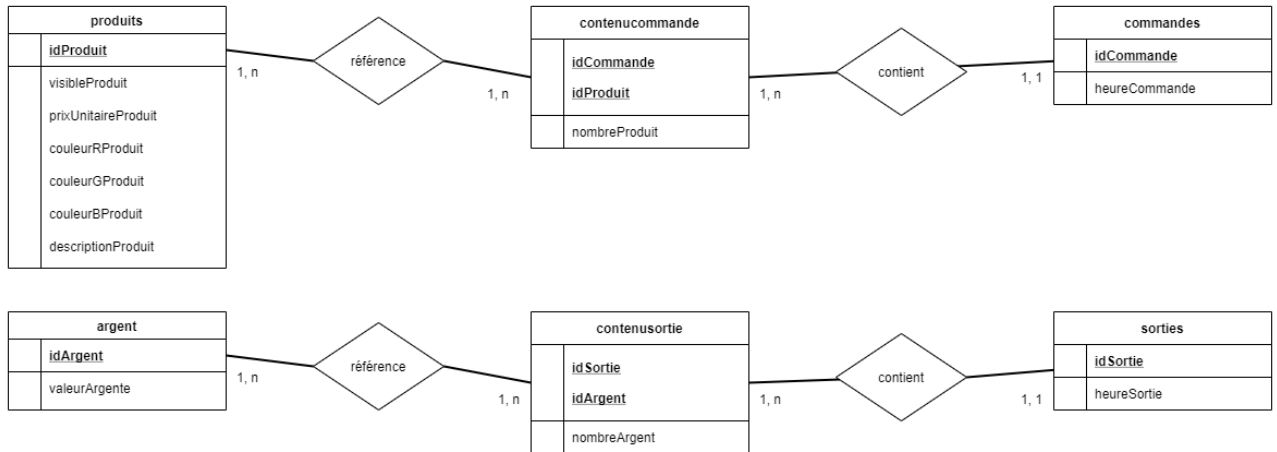
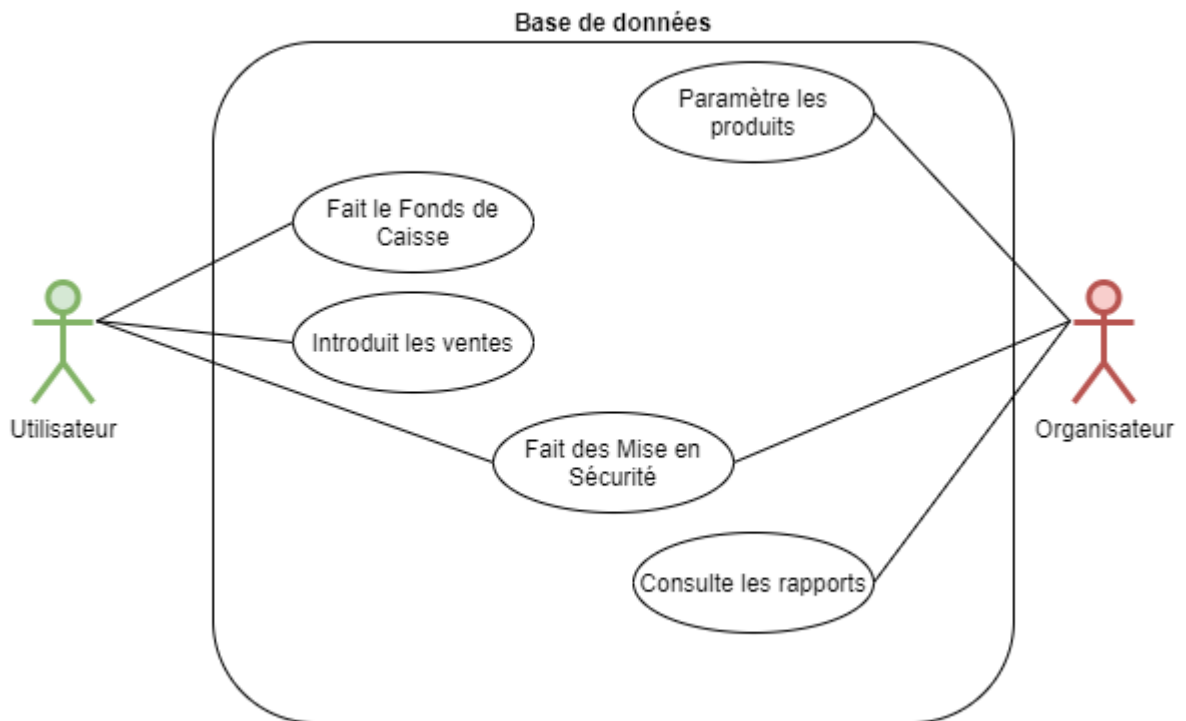


Schéma des cas d'utilisation



Annexe II : Manuel d'utilisation

Installation du programme

Une fois le programme téléchargé, celui-ci ne nécessite pas d'installation pour être utilisé. Cependant, il demande comme prérequis une version java supérieure à la 12. Vous pouvez installer la dernière version de java comme suit.

Pour la version Windows, vous pouvez retrouver l'exécutable [ici](#)

Pour la version linux, les lignes de commandes sont les suivantes :

```
$ sudo apt-get update
```

```
$ sudo apt-get install openjdk-16-jdk
```

Une fois Java installé, il ne vous reste plus qu'à lancer le programme par un double clic.

Vous pourrez trouver les fichiers de sauvegardes dans vos répertoires personnels.

Pour Windows: C:/Users/Public/posjudo

Pour Linux /home/NOMUTILISATEUR/

Au lancement du programme, vous arrivez sur l'écran d'accueil



Utilisation du programme

Menu de l'application

Le menu sur la gauche de l'écran vous permet de voyager dans les différents modules de l'application

Accueil	« Accueil » ramène sur l'écran principal
Vente	« Vente » permet d'enregistrer les commandes
Paramétrage produits	« Paramétrage produits » initialise ou modifie les produits disponibles à la vente
Fonds de caisse	« Fonds de caisse » initialise le fonds de caisse en début de journée
Mise en sécurité	« Mise en sécurité » permet d'enregistrer les sorties de caisse pour mettre l'argent en sécurité
Rapports	« Rapports » visualise à l'écran les différents rapports
Quitter	« Quitter » permet de quitter l'application en sauvegardant vos données

Paramétrer les produits

Une des premières actions à faire, est d'initialiser / modifier les produits que vous proposerez.

En cliquant sur le bouton « Paramétrage produits » vous arrivez à l'écran suivant.

Au centre de l'écran, vous disposez de 36 cases pour afficher les produits disponibles à la vente.

À droite de l'écran, vous trouvez les caractéristiques du produit qu'il faut introduire.

Paramétrage produits						Somme en caisse : 9.0€		
Accueil	Ticket rouge 1.50€	Eau plate 1.50€	Eau pétillante 1.50€	Coca cola 1.50€	0.00€	Limonade 1.50€	ID du bouton	
							Nom du produit	
Vente	0.00€	Ice tea Pêche 2.00€	0.00€	0.00€	Jus de pomme 2.00€	Jus de fruits 1.50€	Prix du produit	
							Composante R de la couleur du bouton	
Paramétrage produit	Ticket Bleu 2.00€	Kriek 2.00€	Blanche 2.00€	Vin au verre 2.00€	Pils 1.50€	Gouyasse 2.50€	Composante G de la couleur du bouton	
							Composante B de la couleur du bouton	
Fonds de caisse	Ticket orange 3.00€	Chimay bleue 3.00€	0.00€	0.00€	0.00€	0.00€	Bouton visible ?	
							Exemple de code RGB	Valider
Mise en sécurité							Rouge : 255 0 0	
							Bleu : 30 144 255	
Rapports	Ticket jaune 2.50€	Petit déjeuner 2.50€	Sandwich 2.50€	Hot Dog 2.50€	0.00€	0.00€	Jaune : 255 255 0	
							Orange : 255 165 0	
Quitter	0.00€	Café 1.50€	Chocolat chaud 1.50€	Couque 1.50€	Chips 1.50€	Bonbons 1.50€	Vert : 34 139 34	Annuler
							Brun : 160 82 45	
							Mauve : 148 0 211	
							Blanc : 255 255 255	

Ticket rouge 1.50€	Eau plate 1.50€	Eau pétillante 1.50€	Coca cola 1.50€	0.00€	Limonade 1.50€
0.00€	Ice tea Pêche 2.00€	0.00€	0.00€	Jus de pomme 2.00€	Jus de fruits 1.50€
Ticket Bleu 2.00€	Kriek 2.00€	Blanche 2.00€	Vin au verre 2.00€	Pils 1.50€	Gouyasse 2.50€
Ticket orange 3.00€	Chimay bleue 3.00€	0.00€	0.00€	0.00€	0.00€
Ticket jaune 2.50€	Petit déjeuner 2.50€	Sandwich 2.50€	Hot Dog 2.50€	0.00€	0.00€
0.00€	Café 1.50€	Chocolat chaud 1.50€	Couque 1.50€	Chips 1.50€	Bonbons 1.50€

Si des produits existent déjà, ils s'affichent ;
sinon la case est noircie.

Pour introduire un nouveau produit,
sélectionnez une des cases noircies.

Pour modifier un produit existant,
sélectionnez la case existante.

Par exemple, cliquez le 1^{er} bouton de la 2nd ligne et introduisez les caractéristiques du produit.

ID du bouton	21
Nom du produit	Ice tea pétillant
Prix du produit	2.0
Composante R de la couleur du bouton	30
Composante G de la couleur du bouton	144
Composante B de la couleur du bouton	255
Bouton visible ?	oui

« ID du bouton » confirme la case sélectionnée. Le premier chiffre renseigne la ligne, le second la colonne.

« Nom du produit » et « Prix du produit » permettent d'introduire le libellé et le prix.

« Composante R / G / B de la couleur du bouton » sont 3 paramètres qui permettent d'afficher la bonne couleur. Ceci est à mettre en lien avec la couleur des tickets.

Exemple de code RGB
Rouge : 255 0 0
Bleu : 30 144 255
Jaune : 255 255 0
Orange : 255 165 0
Vert : 34 139 34
Brun : 160 82 45
Mauve : 148 0 211
Blanc : 255 255 255

Pour vous aider à définir la bonne couleur, vous disposez d'une aide sous forme de tableau dans le bas de l'écran.

« Bouton visible » - indiquez « oui » si vous souhaitez que la case s'affiche ; « non » si vous voulez que la case soit noircie. Cela peut être utile si un produit n'est plus disponible en cours de journée, mais que vous ne voulez pas perdre ses caractéristiques. Dans l'écran de « Vente », le produit n'apparaît plus.

Valider
Annuler

« Valider » permet d'enregistrer les caractéristiques pour ce produit.

« Annuler » laisse les caractéristiques inchangées et désélectionne la case.

Enregistrer le fonds de caisse

Ensuite, vous devez initialiser votre fonds de caisse.

En cliquant sur le bouton « Fonds de caisse » vous arrivez à l'écran suivant.

Au centre de l'écran, vous disposez des boutons qui représentent les billets et pièces en euros.

A droite de l'écran, vous disposez d'un pavé numérique.

Fonds de caisse				Somme en caisse : 9.0€			
Accueil	200€	100€	50€				
Vente	20€	10€	5€				
Paramétrage produit							
Fonds de caisse	2€	1€		7	8	9	Valider
Mise en sécurité				4	5	6	
Rapports	0,50€	0,20€	0,10€	1	2	3	Annuler
Quitter	0,05€	0,02€	0,01€		0		

200€	100€	50€
20€	10€	5€
2€	1€	
0,50€	0,20€	0,10€
0,05€	0,02€	0,01€

Cliquez sur le bouton représentant le billet ou la pièce pour laquelle vous voulez introduire le fonds de caisse.

En cliquant plusieurs fois sur un même bouton, le nombre de billet / pièce de cette valeur s'incrémente.

Si vous devez introduire un nombre élevé, vous pouvez utiliser le pavé numérique. Dans ce cas, introduisez d'abord le nombre grâce au pavé numérique, puis cliquez sur le bouton représentant le billet / la pièce.

Par exemple : 10 (pavé numérique) billets de 20€ (bouton 20€)

Au fur et à mesure, les billets et pièces présents dans votre fonds de caisse s'affichent dans le tableau récapitulatif en haut à droite de votre écran.

Commande N°	3	Heure :	2021-05-26 17:02:37
Quantite	Produit	Prix unitaire	Total
0	0.01€		0.0
0	0.02€		0.0
0	0.05€		0.0
0	0.1€		0.0
0	0.2€		0.0
25	0.5€		12.5
20	1.0€		20.0
20	2.0€		40.0
10	5.0€		50.0
10	10.0€		100.0
10	20.0€		200.0
0	50.0€		0.0
0	100.0€		0.0
0	200.0€		0.0
...
Total :			422.5

Valider
Annuler

« Valider » permet d'enregistrer votre fonds de caisse.

« Annuler » ne change pas votre fonds de caisse.

Une fois validée, votre somme en caisse s'ajuste en ajoutant le total des billets / pièces introduits au montant déjà existant dans la caisse.

Somme en caisse : 551.5€

Enregistrer les ventes

Vos premiers clients arrivent et votre journée de vente peut commencer.

En cliquant sur le bouton « Vente » vous arrivez à l'écran suivant.

Au centre de l'écran, vous disposez des boutons représentant les produits disponibles.

A droite de l'écran, vous disposez d'un pavé numérique.

Vente						Somme en caisse : 551.5€				
Accueil	Ticket rouge 1.50€	Eau plate 1.50€	Eau pétillante 1.50€	Coca cola 1.50€		Limonade 1.50€				
Vente		Ice tea Pêche 2.00€			Jus de pomme 2.00€	Jus de fruits 1.50€				
Paramétrage produit	Ticket Bleu 2.00€	Kriek 2.00€	Blanche 2.00€	Vin au verre 2.00€	Pils 1.50€	Gouyasse 2.50€				
Fonds de caisse	Ticket orange 3.00€	Chimay bleue 3.00€								
Mise en sécurité							7	8	9	Valider
Rapports	Ticket jaune 2.50€	Petit déjeuner 2.50€	Sandwich 2.50€	Hot Dog 2.50€			4	5	6	
Quitter		Café 1.50€	Chocolat chaud 1.50€	Couque 1.50€	Chips 1.50€	Bonbons 1.50€	1	2	3	Annuler
								0		

Ticket rouge 1.50€	Eau plate 1.50€	Eau pétillante 1.50€	Coca cola 1.50€		Limonaade 1.50€
	Ice tea Pêche 2.00€			Jus de pomme 2.00€	Jus de fruits 1.50€
Ticket Bleu 2.00€	Kriek 2.00€	Blanche 2.00€	Vin au verre 2.00€	Pils 1.50€	Gouyasse 2.50€
Ticket orange 3.00€	Chimay bleue 3.00€				
Ticket jaune 2.50€	Petit déjeuner 2.50€	Sandwich 2.50€	Hot Dog 2.50€		
	Café 1.50€	Ghodelt chaud 1.50€	Couque 1.50€	Chips 1.50€	Bonbons 1.50€

Cliquez sur le bouton représentant le produit commandé par votre client.

En cliquant plusieurs fois sur un même bouton, le nombre pour ce produit s'incrémente.

Si vous devez introduire un nombre élevé, vous pouvez utiliser le pavé numérique. Dans ce cas, introduisez d'abord le nombre grâce au pavé numérique, puis cliquez sur le bouton représentant le produit.

Par exemple : 15 (pavé numérique) Coca cola (bouton Coca cola)

Au fur et à mesure, votre commande s'affiche dans le tableau récapitulatif en haut à droite de votre écran

Commande N°	3	Heure :	2021-05-26 17:13:27
Quantité	Produit	Prix unitaire	Total
15	Coca cola	1.5	22.5
2	Pils	1.5	3.0
1	Gouyasse	2.5	2.5
2	Café	1.5	3.0
20	Total :		31.0

Valider

Annuler

« Valider » permet d'enregistrer votre commande.

« Annuler » annule toute la commande.

Une fois validé, le total de votre commande s'ajoute à votre somme en caisse.

Somme en caisse : 582.5€

Faire les mises en sécurité

En cours de journée, la somme totale que vous avez en caisse peut devenir importante. Vous pouvez procéder à une mise en sécurité de l'argent et enregistrer cette sortie d'argent dans l'application.

En cliquant sur le bouton « Mise en sécurité » vous arrivez à l'écran suivant.
 Au centre de l'écran, vous disposez des boutons qui représentent les billets et pièces en euros.
 A droite de l'écran, vous disposez d'un pavé numérique.

Mise en sécurité				Somme en caisse : 582.5€				
Accueil	200€	100€	50€					
Vente	20€	10€	5€					
Paramétrage produit								
Fonds de caisse	2€	1€			7	8	9	Valider
Mise en sécurité	0,50€	0,20€	0,10€	4	5	6		
Rapports	0,05€	0,02€	0,01€	1	2	3	Annuler	
Quitter					0			

200€	100€	50€
20€	10€	5€
2€	1€	
0,50€	0,20€	0,10€
0,05€	0,02€	0,01€

Cliquez sur le bouton représentant le billet ou la pièce que vous voulez mettre dans votre mise en sécurité.

En cliquant plusieurs fois sur un même bouton, le nombre de billet / pièce de cette valeur s'incrémente.

Si vous devez introduire un nombre élevé, vous pouvez utiliser le pavé numérique. Dans ce cas, introduisez d'abord le nombre grâce au pavé numérique, puis cliquez sur le bouton représentant le billet / la pièce.

Par exemple : 25 (pavé numérique) billets de 10€ (bouton 10€)

Au fur et à mesure, les billets et pièces que vous mettez dans votre mise en sécurité s'affichent dans le tableau récapitulatif en haut à droite de votre écran.

Mise en sécurité N°	4	Heure :	2021-05-27 18:12:31
Quantite	Produit	Prix unitaire	Total
0	0.01€		0.0
0	0.02€		0.0
0	0.05€		0.0
0	0.1€		0.0
0	0.2€		0.0
0	0.5€		0.0
0	1.0€		0.0
20	2.0€		40.0
12	5.0€		60.0
25	10.0€		250.0
0	20.0€		0.0
0	50.0€		0.0
0	100.0€		0.0
0	200.0€		0.0
Total :			350.0

Valider

Annuler

« Valider » permet d'enregistrer votre mise en sécurité.

« Annuler » n'enregistre pas votre mise en sécurité et ne change pas votre somme en caisse.

Une fois validée, votre somme en caisse s'ajuste en déduisant le total des billets / pièces mis en sécurité du montant existant dans la caisse.

Somme en caisse : 232.5€

[Afficher les rapports](#)

En cliquant sur le bouton « Rapports » vous arrivez à l'écran suivant.

À droite de l'écran, vous disposez d'un sous-menu avec les 4 rapports disponibles :

- « Rapport vente » : toutes les ventes pour une date donnée
- « Rapport mise en sécurité » : le contenu d'une mise en sécurité
- « Rapport fonds de caisse » : le contenu d'une initialisation du fonds de caisse
- « Ticket N° » : le détail d'une commande vendue

Rapports		Somme en caisse : 232.5€	
Accueil		Rapport vente	2021-05-27
Vente			
Paramétrage produit		Rapport mise en sécurité	4
Fonds de caisse			
Mise en sécurité		Rapport fonds de caisse	3
Rapports			
Quitter		Ticket N°	3

Rapport Vente

Par défaut, la date reprise pour le rapport des ventes est la date du jour. Vous pouvez changer cette date si vous souhaitez obtenir le rapport des ventes d'une date précédente.

En cliquant sur le bouton « Rapport Vente » vous obtenez, au centre de l'écran, le résumé des ventes de cette journée.

Produit	Prix unitaire	Total vendu	Somme vendu
Ice tea Pêche	2.0	7	14.0
Jus de fruits	1.5	8	12.0
Gouyasse	2.5	4	10.0
Hot Dog	2.5	8	20.0
Café	1.5	4	6.0
Couque	1.5	7	10.5
Eau pétillante	1.5	4	6.0
Coca cola	1.5	8	12.0
Blanche	2.0	5	10.0
Ticket rouge	1.5	32	48.0
Ticket Bleu	2.0	7	14.0
Sandwich	2.5	6	15.0
Kriek	2.0	2	4.0
Ticket jaune	2.5	5	12.5
Bonbons	1.5	6	9.0
Eau plate	1.5	4	6.0
Pils	1.5	14	21.0
Chips	1.5	5	7.5
Jus de pomme	2.0	4	8.0
Limonade	1.5	6	9.0
Petit déjeuner	2.5	4	10.0
Ticket orange	3.0	1	3.0
Chimay bleue	3.0	2	6.0
Total		153	273.5

Rapport Mise en sécurité

Par défaut, le numéro repris concerne la dernière mise en sécurité. Vous pouvez changer ce numéro pour obtenir le rapport d'une mise en sécurité précédente.

En cliquant sur le bouton « Rapport Mise en sécurité » vous obtenez au centre de l'écran, votre dernière mise en sécurité.

Mise en sécurité N°	4	Heure :	2021-05-27 18:12:31.000000
Quantité	Produit	Prix unitaire	Total
20	2.0€		40.0
12	5.0€		60.0
25	10.0€		250.0
	Total :		350.0

Rapport Fonds de caisse

Par défaut, le numéro repris concerne le dernier enregistrement de fonds de caisse. Vous pouvez changer ce numéro pour obtenir le rapport d'une initialisation de fonds de caisse précédente. En cliquant sur le bouton « Rapport Fonds de caisse » vous obtenez au centre de l'écran, votre dernière initialisation de fonds de caisse.

Fonds de caisse N°	3	Heure :	2021-05-26 17:02:37.000000
Quantité	Produit	Prix unitaire	Total
25	0.5€		12.5
20	1.0€		20.0
20	2.0€		40.0
10	5.0€		50.0
10	10.0€		100.0
10	20.0€		200.0
	Total :		422.5

Ticket N°

Par défaut, le numéro repris concerne la dernière vente réalisée. Vous pouvez changer ce numéro pour obtenir le contenu d'une commande précédente. En cliquant sur le bouton « Ticket N° » vous obtenez au centre de l'écran, le détail de cette commande.

Commande N°	35	Heure :	2021-05-27 20:19:03.000000
Quantité	Produit	Prix unitaire	Total
2	Jus de fruits	1.5	3.0
3	Pils	1.5	4.5
1	Hot Dog	2.5	2.5
1	Chips	1.5	1.5
1	Bonbons	1.5	1.5
8	Total :		13.0

Quitter le programme

Grâce au bouton « Quitter », vous quittez l'application. Ceci a pour effet de fermer proprement tous les fichiers nécessaires et de sauvegarder vos données de la journée.

Annexe III : Users stories complètes

Personnage

- Consommateur
- Organisateur du Grand Prix de Judo de la Ville d'Ath
- Utilisateur

User Stories

Les besoins de l'organisateur du GP ainsi que des principaux utilisateurs ont été analysés en 2 temps. Tout d'abord lors du lancement de l'idée de créer une application de point de vente pour le Grand Prix de Judo où les grandes fonctionnalités ont été décrites.

C'est suite à cette première discussion qu'une des utilisatrices a dessiné les écrans tels qu'elle se les imaginait.

Avec les 2 utilisateurs principaux, sur base de ces écrans et en s'imaginant le déroulement d'un Grand Prix, depuis la préparation jusqu'au débriefing, j'ai décrit les fonctionnalités dans le détail, sous forme de user stories (US).

US1 - Installation

En tant qu'utilisateur de l'application, je dois pouvoir installer et lancer facilement l'application à partir de n'importe quel PC.

Description

En préparation au Grand Prix du Judo, l'utilisateur reçoit un lien qui lui permet de télécharger un fichier exécutable sur son PC.

Par un double-clic sur le fichier exécutable, l'application s'ouvre et est totalement autonome.

Le lieu où se déroule le Grand Prix de Judo ne dispose pas d'un wifi stable, il faut donc pouvoir travailler en local tout au long de la journée.

L'utilisateur dispose d'un PC avec Windows 10 et d'un autre avec Linux (Ubuntu 20,04).

Il aimerait que l'application puisse tourner indifféremment sur l'un ou l'autre de ses PCs.

L'idéal serait aussi que l'application puisse être mise sur une tablette.

Acceptation

Facilité d'installation sur un PC par un téléchargement de fichier

L'application tourne sans avoir besoin de wifi

Les données sont sauvegardées en local sur le PC

L'application tourne sans problème sur le PC en Windows 10

L'application tourne sans problème sur le PC avec Linux

Tâches techniques

Avoir une redirection automatique vers le bon fichier en fonction de l'OS

US2 - Paramétrage

En tant qu'organisateur du Grand Prix de Judo d'Ath, je dois initialiser la liste des produits afin de visualiser les produits disponibles à la vente.

Description

Les produits sont matérialisés par des tickets de couleur, la couleur étant fonction du prix.
Sur l'écran Produits, je dispose de boutons que je peux paramétrer afin qu'ils représentent les caractéristiques des produits.

L'écran de paramétrage des produits est divisé en 3 zones :

- sur la gauche, je garde le Menu de l'application
- au milieu, la plus grande partie de l'écran, je dispose de 36 boutons paramétrables (6 * 6)
- sur la droite, en haut, j'ai les paramètres du produit que je peux mettre à jour
- sur la droite en bas, j'ai un clavier numérique
- sur la droite, en bas, j'ai un bouton « Enregistrer » et un bouton « Annuler »

Je choisis un bouton en cliquant dessus. Je peux introduire ses caractéristiques qui sont :

- la couleur du ticket
- le libellé du produit
- le prix de vente
- une librairie de photos

En cliquant sur le bouton « Enregistrer », le bouton produit est sauvé à la place définie et avec ses caractéristiques.

En cliquant sur le bouton « Annuler », les paramètres introduits ne sont pas enregistrés. Les paramètres sont réinitialisés.

De plus, si un produit ne devait plus être disponible en cours de journée, il faut pouvoir le cacher sans pour autant perdre ses caractéristiques.

Maquette

PARAMETRAGE DES PRODUITS **JUDO CLUB ATHLOIS**

Navigation (Gauche) :

- Menu
- Vente
- Produits
- Caisse
- Sécurité
- Rapport

Zone Centrale (36 boutons) :

Row 1	Ticket 1,50 €	Eau plate 1,50 €	Eau pétillante 1,50 €	Coca Cola 1,50 €	Coca Light 1,50 €	Limonade 1,50 €
Row 2		Ice Tea 2,00 €				Jus de fruits 1,50 €
Row 3	Ticket 2,00 €	Kriek 2,00 €	Blanche 2,00 €	Vin au verre 2,00 €	Pils 1,50 €	Gouyasse 2,50 €
Row 4	Ticket 3,00 €	Chimay bleue 3,00 €				
Row 5	Ticket 2,50 €	Petit Déj 2,50 €	Sandwich 2,50 €	Hot Dog 2,50 €		
Row 6		Café 1,50 €	Chocolat chaud 1,50 €	Couque 1,50 €	Chips 1,50 €	Bonbons 1,50 €

Zone Paramètres (Droite) :

- Couleur :
- Libellé :
- Prix :
- Photo : [librairie_photos](#)

Zone Clavier (Droite) :

- 7 8 9
- 4 5 6
- 1 2 3
- 0 ,
- Enregistrer (✓)
- Annuler (✗)

Acceptation

Les 36 boutons sont cliquables et donnent accès à leurs caractéristiques si elles existent déjà. Si les caractéristiques n'existent pas encore, ce sont les valeurs initiales qui sont montrées.

Chacune des caractéristiques est accessible et modifiable.

Dès que le bouton « Enregistrer » est cliqué, le bouton sur l'écran du milieu se met à jour avec la bonne couleur, le libellé du produit et son prix ainsi que la photo choisie.

Pour quitter cet écran, je dispose du Menu qui me permet d'aller sur une autre fonctionnalité de l'application.

Si je choisis dans le Menu la fonctionnalité « Vente », je retrouve les boutons produits correctement paramétrés avec la couleur du bouton, le libellé du produit et son prix de vente ainsi que la photo.

Tâches techniques

Récupérer les paramètres dans la DB

Enregistrer les paramètres dans la DB

Avoir des champs pour le prix, la description, la couleur

US3 - Accueil

En tant qu'utilisateur de l'application, je souhaite qu'elle s'ouvre sur un écran d'Accueil.

Description

L'écran d'Accueil est composé de 2 zones principales :

- sur la gauche, un Menu de l'application,
- sur le reste de l'écran, une photo du Grand Prix de Judo de la Ville d'Ath (idéalement, la même photo que celle reprise sur les affiches et documents du Grand Prix).

Le Menu de l'application reprend :

- Menu : permet de revenir sur l'écran d'accueil,
- Vente : permet d'aller sur l'écran où enregistrer les ventes des produits,
- Produits : permet d'aller sur l'écran où paramétrer les produits,
- Caisse : permet d'aller sur l'écran où initialiser le fonds de caisse en début de journée,
- Sécurité : permet d'aller sur l'écran où enregistrer les sorties de fonds lors de la mise en sécurité de l'argent
- Rapport : permet d'aller sur l'écran où visualiser et imprimer des rapports.

Idéalement, un bouton « Aide » est disponible avec un manuel utilisateur succinct.



Acceptation

L'écran d'accueil reprend un Menu avec les fonctionnalités de l'application. Chaque bouton du Menu envoie vers le bon écran de l'application.

Tâches techniques

Récupérer la photo dans les fichiers
Créer un menu latéral comprenant des boutons redirigeant vers le bon écran.

US4 – Initialisation de la caisse

En tant qu'utilisateur de l'application, je souhaite pouvoir introduire mon fonds de caisse en début de journée.

Description

L'écran d'Initialisation de la Caisse est divisé en 3 zones :

- sur la gauche, le Menu de l'application – identique partout dans l'application,
- au milieu, je dispose de boutons représentant les billets et pièces en euros,
- sur la droite, en haut, va s'afficher ligne par ligne le nombre de billet / pièce de chaque type avec le total que cela représente ainsi que le total global,
- sur la droite en bas, j'ai un clavier numérique
- sur la droite, en bas, j'ai un bouton « Enregistrer » et un bouton « Annuler ».

Pour initialiser le fonds de caisse, l'utilisateur indique le nombre et choisit, en cliquant dessus, le billet ou la pièce.

Pour indiquer le nombre, l'utilisateur utilise soit le pavé numérique, soit clique autant de fois que nécessaire sur le bouton représentant le billet ou la pièce.

Une ligne correspondant à ce billet / cette pièce s'affiche en haut à droite avec le nombre, la valeur faciale et le total de cette ligne.

Une ligne total global calcule au fur et à mesure le contenu du fonds de caisse. Cela permet à l'utilisateur de vérifier ce qu'il a introduit en le comparant aux montants renseignés sur l'enveloppe reçue avec l'argent du fonds de caisse.

Une fois l'introduction des différents billets et pièces du fonds de caisse vérifiée, l'utilisateur clique sur « Enregistrer » pour sauvegarder le fonds de caisse.

Le bouton « Annuler » ne sauvegarde pas le fonds de caisse et remet les lignes et totaux à zéro.

Maquette

INITIALISATION DE LA CAISSE **JUDO CLUB ATHOIS**

Menu
Vente
Produits
Caisse
Sécurité
Rapport

Billets et pièces disponibles :

200 €	100 €	50 €
20 €	10 €	5 €
2 €	1 €	
0,50 €	0,20 €	0,10 €
0,05 €	0,02 €	0,01 €

Calculs et Total :

5	*	20,00 €	=	100,00 €
10	*	10,00 €	=	100,00 €
20	*	5,00 €	=	100,00 €
25	*	2,00 €	=	50,00 €
50	*	1,00 €	=	50,00 €
50	*	0,50 €	=	25,00 €

TOTAL : 425,00 €

Numérique : 7 8 9 4 5 6 1 2 3 0 ,

Enregistrer **Annuler**

Acceptation

Chacun des boutons représentant les billets et pièces est cliquable.

Les touches du pavé numérique sont cliquables et enregistrent le nombre.

Les lignes reprenant le nombre de tel billet / pièce et son total sont correctes en fonction de ce que l'utilisateur a introduit.

Le total global est la somme de tous les totaux ligne par ligne.

Dès que le bouton « Enregistrer » est cliqué, le fonds de caisse est sauvegardé.

Si le bouton « Annuler » est cliqué, le fonds de caisse n'est pas sauvegardé et les lignes dans le carré en haut à droite ainsi que le total global disparaissent.

Pour quitter cet écran, je dispose du Menu qui me permet d'aller sur une autre fonctionnalité de l'application.

Tâches techniques

Récupérer les différents billets possibles dans la base de données

Incrémenter à chaque clic le nombre du billet / pièce correspondant
Faire un tableau récapitulatif
Enregistrer le tableau et son horodatage

US5 – Vente de tickets

En tant qu'utilisateur de l'application, je veux enregistrer les différentes ventes de produits tout au long de la journée. Toutes les ventes enregistrées doivent être sauvegardées afin de pouvoir en faire un rapport en fin de journée.

Description

L'écran de Vente des tickets est divisé en 3 zones :

- sur la gauche, le Menu de l'application – identique partout dans l'application,
- au milieu, je dispose des boutons représentant les produits (définis lors du paramétrage),
- sur la droite, en haut, va s'afficher ligne par ligne le nombre de produits achetés avec le total de la ligne
- dans ce même cadre, j'ai le total global de la vente ainsi que le montant reçu par le consommateur et la différence à lui rendre,
- sur la droite en bas, j'ai un clavier numérique
- sur la droite, en bas, j'ai un bouton « Enregistrer » et un bouton « Annuler »

Lorsqu'un consommateur se présente, j'enregistre sa commande en cliquant sur le bouton correspondant au produit désiré.

S'il s'agit d'une quantité élevée d'un même produit (par exemple un coach d'un club qui achète 20 tickets rouges pour l'ensemble de ses judokas), j'utilise le pavé numérique pour entrer le nombre désiré puis je clique sur le bouton correspondant au produit.

Le total global de la vente des produits est calculé au fur et à mesure de l'introduction de la commande. L'utilisateur peut ainsi annoncer sans risque d'erreur, le montant à payer.

Si la somme perçue par le client ne correspond pas exactement à la somme à payer, l'utilisateur peut introduire la somme reçue et l'application calcule la différence à rendre au consommateur.

Une fois toute la commande introduite, l'utilisateur clique sur « Enregistrer » pour sauvegarder cette commande.

Le bouton « Annuler » ne sauvegarde pas la commande et remet les lignes de produits et les totaux à zéro.

Toutes les commandes sauvegardées tout au long de la journée sont cumulées afin de connaître le nombre total de chaque produit vendu et le montant total que cela représente. Cette fonctionnalité sera utile pour les rapports de fin de journée.

VENTE DE TICKETS

JUDO CLUB ATHOIS

Menu

Vente

Produits

Caisse

Sécurité

Rapport

Ticket

1,50 €

Eau plate

1,50 €

Eau pétillante

1,50 €

Coca Cola

1,50 €

Coca Light

1,50 €

Limonade

1,50 €

Ice Tea

2,00 €

Jus de fruits

1,50 €

Ticket

2,00 €

Kriek

2,00 €

Blanche

2,00 €

Vin au verre

2,00 €

Pils

1,50 €

Gouyasse

2,50 €

Ticket

3,00 €

Chimay bleue

3,00 €

Ticket

2,50 €

Petit Déj

2,50 €

Sandwich

2,50 €

Hot Dog

2,50 €

Café

1,50 €

Chocolat chaud

1,50 €

Couque

1,50 €

Chips

1,50 €

Bonbons

1,50 €

2 Café = 3,00 €

3 Coca Cola = 4,50 €

2 Pils = 3,00 €

1 Gouyasse = 2,50 €

TOTAL : 13,00 €

RECU : 20,00 €

A RENDRE : 7,00 €

789

456

123

0,

Enregistrer

Annuler

Acceptation

Chaque bouton représente un produit et est cliquable.

Les touches du pavé numérique sont cliquables et enregistrent le nombre de produit désiré.

Les lignes reprenant le nombre de tel produit et son total sont correctes en fonction de la commande introduite.

Le total global est la somme de tous les totaux ligne par ligne.

Les touches du pavé numérique permettent d'introduire la somme reçue du consommateur.

La différence à rendre s'affiche et est correcte (différence entre le montant reçu et la montant total).

Dès que le bouton « Enregistrer » est cliqué, la commande est sauvegardée.

Si le bouton « Annuler » est cliqué, la commande n'est pas sauvegardée et les lignes dans le carré en haut à droite ainsi que le total global disparaissent.

Pour quitter cet écran, je dispose du Menu qui me permet d'aller sur une autre fonctionnalité de l'application.

Tâches techniques

Récupérer les tickets paramétrés dans la base de données

A chaque clic incrémenter le nombre de produit vendu correspondant

Faire un tableau synthèse de la commande

Enregistrer la commande et son horodatage dans la base de données

US6 – Achat de tickets

En tant que consommateur, je souhaite ne pas avoir un trop long temps d'attente pour acheter mes tickets boissons / repas. De plus, le montant total de mes achats doit correspondre aux tickets reçus.

Si je ne paie pas le prix juste, je veux recevoir la différence correcte entre le montant de mes achats et le montant donné.

Page 57

Description

Au fur et à mesure que j'énonce ma commande, le vendeur de tickets l'introduit.

Je peux demander soit un certain nombre de tickets d'une couleur, soit commander un ou plusieurs produits précis.

Je ne dois pas suivre un ordre particulier, par exemple en fonction de la couleur.

Si je me trompe dans ma commande, il faut pouvoir l'annuler.

Quand ma commande est passée, je dois connaître le montant exact et correct à payer.

Si je ne dispose pas de la somme exacte, je dois être correctement remboursé de la différence.

Tout cela doit être très fluide et flexible.

Acceptation

Introduire ma commande se fait facilement et rapidement.

Dès la fin de ma commande, je connais le montant exact à payer.

Je reçois le montant du remboursement si je ne paie pas le montant exact de la commande.

Je peux annuler à tout moment ma commande pour autant qu'elle ne soit pas validée.

US7 – Impression de la vente

En tant que consommateur, je suis parfois amené à demander un ticket de caisse afin de me faire rembourser les montants dépensés. Je peux soit le demander au moment de la commande, soit en fin de journée pour plusieurs commandes.

Description

Une fois ma commande passée, je demande un ticket reprenant le nombre et les produits achetés ainsi que le montant total payé. Ce ticket m'est délivré rapidement.

En fin de journée, je peux venir demander l'impression de mes différentes commandes faites en cours de journée. Ces tickets me sont délivrés rapidement.

Acceptation

Demander et obtenir l'impression de ma / mes commandes se fait facilement et rapidement.

Chaque ticket imprimé reprend le nombre et le libellé de chaque produit avec son total ainsi que le total global de la commande.

US8 – Mise en sécurité

En tant qu'utilisateur de l'application, je veux introduire les mises en sécurité de l'argent disponible en caisse. Cela a pour effet de diminuer l'argent en caisse.

Description

L'écran de Mise en sécurité est divisé en 3 zones :

- sur la gauche, le Menu de l'application – identique partout dans l'application,
- au milieu, je dispose de boutons représentant les billets et pièces en euros,

- sur la droite, en haut, va s'afficher la date et l'heure de la mise en sécurité ainsi qu'un numéro séquentiel,
- sur la droite, en haut, je peux indiquer les personnes qui font la mise en sécurité,
- sur la droite, en haut, va s'afficher ligne par ligne le nombre de billet / pièce de chaque type avec le total que cela représente ainsi que le total global,
- sur la droite en bas, j'ai un clavier numérique
- sur la droite, en bas, j'ai un bouton « Enregistrer » et un bouton « Annuler »

Pour réaliser la mise en sécurité, l'utilisateur commence par introduire le nom des 2 personnes qui font cette mise en sécurité.

Ensuite, l'utilisateur indique le nombre et choisit, en cliquant dessus, le billet ou la pièce.

Pour indiquer le nombre, l'utilisateur utilise soit le pavé numérique, soit clique autant de fois que nécessaire sur le bouton représentant le billet ou la pièce.

Une ligne correspondant à ce billet / cette pièce s'affiche en haut à droite avec le nombre, la valeur faciale et le total de cette ligne.

Une ligne total global calcule au fur et à mesure le montant global de la mise en sécurité. Cela permet à l'utilisateur de vérifier l'argent qu'il remet dans l'enveloppe en le comparant au montant total de la mise en sécurité.

Une fois l'introduction des différents billets et pièces de la mise en sécurité vérifiée, l'utilisateur clique sur « Enregistrer » pour sauvegarder la mise en sécurité.

Le bouton « Annuler » ne sauvegarde pas la mise en sécurité et remet les lignes et totaux à zéro.

Un bouton « Imprimer » permet d'imprimer sur papier le contenu et le montant total de la mise en sécurité. Ce papier doit être glissé dans l'enveloppe contenant l'argent.

Une fois la mise en sécurité enregistrée, le montant total de ma caisse diminue du montant mis en sécurité.

MISE EN SECURITE



Menu



Vente



Produits



Caisse



Sécurité



Rapport



200 €



100 €



50 €



20 €



10 €



5 €



2 €



1 €



0,50 €



0,20 €



0,10 €



0,05 €



0,02 €



0,01 €



Imprimer

Mise en sécurité n° du dd/mm/yy à hh:mm

Par (pers 1) :

Par (pers 2) :

10	*	50,00 €	=	500,00 €
40	*	20,00 €	=	800,00 €
50	*	10,00 €	=	500,00 €
30	*	5,00 €	=	150,00 €

TOTAL : 1950,00 €

789

456

123

0,



Enregistrer



Annuler

Acceptation

Le numéro de mise en sécurité est automatiquement incrémenté de 1 par rapport à la mise en sécurité précédente.

Les date et heure de la mise en sécurité sont automatiquement reprises et correspondent au moment de la mise en sécurité.

Le nom des personnes effectuant la mise en sécurité est un texte libre.

Chacun des boutons représentant les billets et pièces est cliquable.

Les touches du pavé numérique sont cliquables et enregistrent le nombre de billets/pièces.

Les lignes reprenant le nombre de tel billet / pièce et son total sont correctes en fonction de ce que l'utilisateur a introduit.

Le total global est la somme de tous les totaux ligne par ligne.

Dès que le bouton « Enregistrer » est cliqué, la mise en sécurité est sauvegardée.

Si le bouton « Annuler » est cliqué, la mise en sécurité n'est pas sauvegardée et les lignes dans le carré en haut à droite ainsi que le total global disparaissent.

Le bouton « Imprimer » imprime directement les données de cette mise en sécurité.

Le nouveau montant de ma caisse correspond au montant avant mise en sécurité moins le montant mis en sécurité.

Pour quitter cet écran, je dispose du Menu qui me permet d'aller sur une autre fonctionnalité de l'application.

Tâches techniques

Récupérer les différents billets possibles dans la base de données

Incrémenter à chaque clic le nombre du billet / pièce correspondant

Faire un tableau récapitulatif

Enregistrer le tableau, son horodatage et les responsables

US9 – Rapport Mise en sécurité

En tant qu'utilisateur de l'application, je veux pouvoir imprimer une ou toutes les mises en sécurité de la journée via l'écran de Rapport.

Description

L'écran de production de rapports comprend 3 zones :

- sur la gauche, le Menu de l'application – identique partout dans l'application,
- au milieu va s'afficher le résultat du rapport demandé,
- sur la droite, en haut, je dispose de boutons pour générer les différents rapports disponibles,
- sur la droite en bas, j'ai un bouton « Imprimer »
- sur la droite, en bas, j'ai un bouton « Enregistrer » et un bouton « Annuler »

En cliquant sur le bouton correspondant au rapport mise en sécurité, l'application génère la liste des mises en sécurité effectuées jusque-là.

Chaque ligne de mise en sécurité reprend le numéro de la mise en sécurité, les dates et heures ainsi que le montant total de la mise en sécurité.

Je peux sélectionner/cliquer une ou plusieurs lignes de mise en sécurité.

En cliquant sur le bouton « Imprimer », les lignes sélectionnées seront imprimées.

En cliquant sur le bouton « Enregistrer », les rapports correspondant aux lignes sélectionnées seront sauvegardés sur le PC.

En cliquant sur le bouton « Annuler », les lignes sélectionnées seront décochées.

Le rapport qui est imprimé reprend pour chaque mise en sécurité :

- le numéro de mise en sécurité,
- les date et heure de mise en sécurité,
- les personnes ayant fait la mise en sécurité,
- le nombre des billets et pièces repris dans la mise en sécurité ainsi que le montant total de la ligne,
- le montant global de la mise en sécurité.

Pour quitter cet écran, je dispose du Menu qui me permet d'aller sur une autre fonctionnalité de l'application.

PRODUCTION DES RAPPORTS

JUDO CLUB ATHOIS


Menu


Vente


Produits


Caisse


Sécurité


Rapport

Liste des ventes						
N°	Date	Heure	Nb	Produit	Total Ligne	Total Commande
1	25/04/2020	08:43	2	Petit Dej	5,00 €	
1	25/04/2020	08:43	1	Sandwich	2,50 €	
1	25/04/2020	08:43				7,50 €
2	25/04/2020	08:46	1	Café	1,50 €	1,50 €
3	25/04/2020	08:48	2	Café	3,00 €	
3	25/04/2020	08:48	1	Ice Tea	2,00 €	
3	25/04/2020	08:48				5,00 €
4	25/04/2020	08:51	1	Chocolat chaud	1,50 €	
4	25/04/2020	08:51	2	Café	3,00 €	
4	25/04/2020	08:51	1	Jus de fruits	1,50 €	
4	25/04/2020	08:51				6,00 €
5	25/04/2020	08:55	2	Coca Cola	3,00 €	
5	25/04/2020	08:55	2	Sandwich	5,00 €	
5	25/04/2020	08:55	1	Couque	1,50 €	
5	25/04/2020	08:55				9,50 €
6	25/04/2020	08:56	1	Petit Dej	2,50 €	2,50 €
7	25/04/2020	08:56	2	Jus de fruits	3,00 €	
7	25/04/2020	08:56	2	Petit Dej	5,00 €	
7	25/04/2020	08:56				8,00 €
8	25/04/2020	08:59	1	Eau pétillante	1,50 €	1,50 €


Liste des produits


Liste des ventes


Liste du Fond de caisse


Mises en sécurité


Imprimer


Enregistrer


Annuler

Acceptation

Le bouton « Mise en sécurité » génère à l'écran la liste de toutes les mises en sécurité.

Les numéros, dates et heures de chaque ligne correspondent bien au moment où les mises en sécurité ont été faites.

Le montant total de chaque mise en sécurité est correct.

Les nombres, billets / pièces ainsi que le montant total de chaque ligne est correctement affiché à l'écran.

Le bouton « Imprimer » imprime directement le contenu montré à l'écran

Le bouton « Enregistrer » sauvegarde le rapport de mise en sécurité.

Le bouton « Annuler » réinitialise l'écran.

Tâches techniques

Récupérer toutes les mises en sécurité effectuées pour un jour donné

Afficher ces informations dans un tableau

US10 – Rapport des ventes de la journée

En tant qu'utilisateur de l'application, je veux pouvoir imprimer toutes les ventes de la journée via l'écran de Rapport.

Description

L'écran de production de rapports comprend 3 zones :

- sur la gauche, le menu de l'application – identique partout dans l'application,
- au milieu va s'afficher le résultat du rapport demandé,
- sur la droite, en haut, je dispose de boutons pour générer les différents rapports disponibles,
- sur la droite en bas, j'ai un bouton « Imprimer »
- sur la droite, en bas, j'ai un bouton « Enregistrer » et un bouton « Annuler »

Page 62

En cliquant sur le bouton correspondant au rapport liste des ventes, l'application génère la liste de toutes les ventes effectuées jusque-là.

Cette liste est affichée à l'écran.

Chaque ligne reprend :

- le numéro de la commande,
- les date et heure auxquelles la commande a été enregistrée,
- le nombre et le libellé du produit commandé ainsi que le total pour cette ligne.

Le total global de la commande est repris sur la dernière ligne de la commande.

En cliquant sur le bouton « Imprimer », la liste des ventes de la journée est imprimée.

En cliquant sur le bouton « Enregistrer », la liste des ventes de la journée est sauvegardée sur le PC.

En cliquant sur le bouton « Annuler », la partie centrale de l'écran est remise à blanc.

Le rapport qui est imprimé reprend chaque ligne affichée à l'écran avec toutes ses données.

De plus, une synthèse globale est effectuée. Elle reprend :

- le nombre total de commandes,
- le nombre total de chaque produit vendu avec son libellé,
- le montant total de la vente pour ce produit,
- le montant total de toutes les commandes de la journée.

Pour quitter cet écran, je dispose du Menu qui me permet d'aller sur une autre fonctionnalité de l'application.

Maquette

Cfr Maquette Rapport mise en sécurité

Acceptation

Le bouton « Liste des ventes » génère à l'écran la liste de toutes les ventes.

Les numéros de commande, dates et heures de chaque ligne correspondent bien au moment où les ventes ont été faites.

Le nombre et le libellé du produit vendu sont corrects pour chaque commande.

Le montant total de chaque produit vendu est correct ainsi que le montant total de la commande.

Le bouton « Imprimer » imprime directement le contenu montré à l'écran ainsi que la synthèse de la journée.

Le bouton « Enregistrer » sauvegarde le rapport de liste des ventes.

Le bouton « Annuler » réinitialise l'écran.

Tâches techniques

Récupérer toutes les ventes effectuées pour un jour donné

Afficher ces informations dans un tableau

US11 – Imprimer le ticket d'une commande

En tant qu'utilisateur de l'application, je veux pouvoir imprimer le ticket d'une commande via l'écran de Rapport.

Description

L'écran de production de rapports comprend 3 zones :

- sur la gauche, le menu de l'application – identique partout dans l'application,

- au milieu va s'afficher le résultat du rapport demandé,
- sur la droite, en haut, je dispose de boutons pour générer les différents rapports disponibles,
- sur la droite en bas, j'ai un bouton « Imprimer »
- sur la droite, en bas, j'ai un bouton « Enregistrer » et un bouton « Annuler »

En cliquant sur le bouton correspondant au rapport ticket d'une commande, l'application me demande d'entrer le numéro de la commande. Une fois le numéro introduit, l'application génère la liste de tous les produits vendus pour cette commande.

Cette liste est affichée à l'écran.

Chaque ligne reprend

- le numéro de la commande,
- les date et heure auxquelles la commande a été enregistrée,
- le nombre et le libellé du produit commandé ainsi que le total pour cette ligne.

Le total global de la commande est repris sur la dernière ligne de la commande.

En cliquant sur le bouton « Imprimer », le ticket de la commande est imprimé.

En cliquant sur le bouton « Enregistrer », le ticket de la commande est sauvegardé sur le PC.

En cliquant sur le bouton « Annuler », la partie centrale de l'écran est remise à blanc.

Le rapport qui est imprimé reprend chaque ligne affichée à l'écran avec toutes ses données.

Pour quitter cet écran, je dispose du Menu qui me permet d'aller sur une autre fonctionnalité de l'application.

Acceptation

Le bouton « Ticket d'une commande » génère à l'écran la liste de tous les produits vendus pour le numéro de la commande introduit.

Les date et heure correspondent bien au moment où la vente a été faite.

Le nombre et le libellé du produit vendu, ainsi que le montant total sont corrects.

Le montant total de la commande est correct.

Le bouton « Imprimer » imprime directement le contenu montré à l'écran.

Le bouton « Enregistrer » sauvegarde le ticket de cette commande.

Le bouton « Annuler » réinitialise l'écran.

Tâches techniques

Récupérer les données de la commande

Les enregistrer sous forme d'un tableau dans un PDF imprimable

US12 – Rapport fonds de caisse

En tant qu'utilisateur de l'application, je veux pouvoir imprimer le nombre de billets et pièces introduits à l'initialisation via l'écran de Rapport

Description

L'écran de production de rapports comprend 3 zones :

- sur la gauche, le menu de l'application – identique partout dans l'application,
- au milieu va s'afficher le résultat du rapport demandé,
- sur la droite, en haut, je dispose de boutons pour générer les différents rapports disponibles,
- sur la droite en bas, j'ai un bouton « Imprimer »

- sur la droite, en bas, j'ai un bouton « Enregistrer » et un bouton « Annuler »

En cliquant sur le bouton correspondant au rapport fonds de caisse, l'application génère le nombre de billets et pièces introduits lors de l'initialisation du fonds de caisse. Le résultat est affiché à l'écran.

Chaque ligne reprend le montant total pour ce billet / pièce.

Le montant total du fonds de caisse est également affiché.

En cliquant sur le bouton « Imprimer », le rapport est imprimé.

En cliquant sur le bouton « Enregistrer », le rapport est sauvegardé sur le PC.

En cliquant sur le bouton « Annuler », la partie centrale de l'écran est remise à blanc.

Le rapport qui est imprimé reprend :

- le nombre et le libellé des billets / pièces,
- le montant total pour ce billet / pièce,
- le montant global du fonds de caisse.

Pour quitter cet écran, je dispose du Menu qui me permet d'aller sur une autre fonctionnalité de l'application.

Maquette

Cfr Maquette Rapport mise en sécurité

Acceptation

Le bouton « Fonds de caisse » génère à l'écran la liste de tous les billets / pièces introduits lors de l'initialisation.

Le nombre et le libellé des billets / pièces, ainsi que le montant total sont corrects.

Le montant global du fonds de caisse est correct.

Le bouton « Imprimer » imprime directement le contenu montré à l'écran.

Le bouton « Enregistrer » sauvegarde le rapport de fonds de caisse.

Le bouton « Annuler » réinitialise l'écran.

Tâches techniques

Récupérer tous les fonds de caisse effectués pour un jour donné

Afficher ces informations dans un tableau

Annexe IV : Code source

Le code source se trouve sur Github dans le repository suivant :
<https://github.com/GGEphec/PoSJudo>

Annexe V : Java doc

La java doc est comprise dans le code source et est disponible dans le répertoire suivant :
<https://github.com/GGEphec/PoSJudo/tree/main/Code/doc>

Bibliographie

Choix technique

- ValueCoders, *Top Object-Oriented Programming Languages To Follow in 2020*, <https://www.valuecoders.com/blog/technology-and-apps/top-object-oriented-programming-languages-to-follow-in-2020/> consulté le : 20 janvier 2021
- Read Dive, *Top 5 Object-Oriented Programming Language*, <https://readdive.com/top-5-object-oriented-programming-language/> consulté le : 20 janvier 2021
- Google, *GUI python vs java*, https://www.google.com/search?q=GUI+python+vs+java&rlz=1C1DIMA_enBE783BE783&oq=GUI+python+vs+java&aqs=chrome..69i57j0i19i22i30j69i64.5527j0j7&sourceid=chrome&ie=UTF-8 consulté le : 20 janvier 2021
- University of Tennessee, *A Comparison of Programming Languages for Graphical User Interface Programming*, https://trace.tennessee.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1590&context=utk_chanhonoproj consulté le : 20 janvier 2021
- YouTube TechinBox, *Java MYSQL NetBeans POS inventory System Full Project*, https://www.youtube.com/watch?v=uHCR9btElyA&list=PLRmvSuBP-XfzI7PHoZ0e_xh05XZN3hGbM&index=2&ab_channel=TechinBox consulté le : 20 janvier 2021
- YouTube TutuFunny, *Inventory Management System step by step in Java*, https://www.youtube.com/watch?v=Ksf0383Hhdc&list=PLuji25yj7oIKWUxnb3GeRfqI9s5C26CyD&ab_channel=TutusFunny consulté le : 20 janvier 2021

Comparaison autre software

- Capterra, *7 Best Free and Open-Source POS (Point Of Sale) Software Tools*, <https://blog.capterra.com/the-top-6-free-and-open-source-pos-solutions/> consulté le : 20 janvier 2021
- SmartCaisse, *Caisse enregistreuse professionnelle certifiée*, <https://www.smartcaisse.fr/demo.html> consulté le : 26 mai 2021
- Chromis, *Chromis POS*, <https://chromis.co.uk/chromispos/> consulté le : 26 mai 2021

Analyse de sécurité

- Vaadata, *Comment évaluer les risques d'une faille de sécurité ?*, <https://www.vaadata.com/blog/fr/comment-evaluer-les-risques-dune-faille-de-securite/> consulté le : 26 mai 2021

Java

Installation

- Medium, *First steps of migrating from Java 8 to Java 11*,
<https://medium.com/@daniel.panagio18/first-steps-of-migrating-from-java-8-to-java-11-98ea9a5eafff> consulté le : 20 janvier 2021

MVC

- Stack Overflow, *swing - Java and GUI - Where do ActionListeners belong according to MVC pattern?*, <https://stackoverflow.com/questions/26517856/java-and-gui-where-do-actionlisteners-belong-according-to-mvc-pattern> consulté le : 21 mai 2021
- Sharpcorner, *Programming in Java Using the MVC Architecture*, <https://www.c-sharpcorner.com/UploadFile/201fc1/programming-in-java-using-the-mvc-architecture/> consulté le : 20 mai 2021

Interface graphique

- Objis, *Tutoriel Java SWING N°1 : votre première fenêtre SWING*, <http://objis.com/tutoriel-java-swing-n1-votre-premiere-fenetre-swing/> consulté le : 20 janvier 2021
- YouTube SL Coder Ego with Coding, *First GUI program- How to create a JFrame using eclipse(Programmatically)*,
https://www.youtube.com/watch?v=CVldNE9UVPs&ab_channel=SLCoderEgowithCoding consulté le : 6 mars 2021
- NetBeans, *Introduction to GUI Building*, <https://netbeans.apache.org/kb/docs/java/gui-functionality.html> consulté le : 6 mars 2021
- Coderanch, *Closing all open windows and opening new one*,
<https://coderanch.com/t/538905/java/Closing-open-windows-opening> consulté le : 21 mai 2021
- Dream.In.Code, *Change The Font Size Used In A Textfield – Java*,
<https://www.dreamincode.net/forums/topic/142986-change-the-font-size-used-in-a-textfield/> consulté le : 23 mai 2021
- Université Grenoble Alpes, *Swing : JPanel*, <https://miashs-www.un-ga.fr/prevert/Prog/Java/swing/JPanel.html> consulté le : 23 mai 2021
- Vaadin, *GridLayout with percentage column widths?*,
<https://vaadin.com/forum/thread/128956/gridlayout-with-percentage-column-widths> consulté le : 26 mai 2021

Affichage des tableaux

- Tutorialspoint, *How to create DefaultTableModel which is an implementation of TableModel*,
<https://www.tutorialspoint.com/how-to-create-defaulttablemodel-which-is-an-implementation-of-tablemodel> consulté le : 21 mai 2021
- Documentation Oracle, *DefaultTableModel (Java Platform SE 8)*,
<https://docs.oracle.com/javase/8/docs/api/index.html?javax/swing/table/DefaultTableModel.html> consulté le : 21 mai 2021

Variables

- Beginner's book, *Java - String substring() Method example*,
<https://beginnersbook.com/2013/12/java-string-substring-method-example/> consulté le : 20 mai 2021
- Attacomsian, *Remove the last character of a string in Java* ,
<https://attacomsian.com/blog/java-remove-last-character-of-string> consulté le : 20 mai 2021
- Stack Overflow, *exception - Try/catch in Java*,
<https://stackoverflow.com/questions/2068186/try-catch-in-java> consulté le : 22 mai 2021
- Codegrepper, *java round double to 2 decimal places*, <https://www.codegrepper.com/code-examples/java/java+round+double+to+2+decimal+places> consulté le : 23 mai 2021
- Stack abuse, *Using Global Variables/Constants in Java*, <https://stackabuse.com/using-global-variables-constants-in-java/> consulté le : 23 mai 2021
- Documentation Oracle, *DateTimeFormatter (Java Platform SE 8)*,
<https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html>
consulté le : 26 mai 2021

Stockage des données

- Stack Overflow, *How to make a new List in Java*,
<https://stackoverflow.com/questions/858572/how-to-make-a-new-list-in-java> consulté le : 19 mai 2021

Affichage du fond d'écran Accueil

- Stack Overflow, *java - How to add an image to a JPanel?*,
<https://stackoverflow.com/questions/299495/how-to-add-an-image-to-a-jpanel> consulté le : 24 mai 2021
- Documentation Oracle, *How to Use Editor Panes and Text Panes*,
<https://docs.oracle.com/javase/tutorial/uiswing/components/editorpane.html> consulté le : 26 mai 2021
- Memorynotfound, *Read an Image from File or URL or Class Path*,
<https://memorynotfound.com/java-read-image-file-url-classpath-inputstream/> consulté le : 26 mai 2021
- Stack Overflow, *java - Resize a picture to fit a JLabel*,
<https://stackoverflow.com/questions/16343098/resize-a-picture-to-fit-a-jlabel/16345968>
consulté le : 26 mai 2021
- DYclassroom, *How to read and write image file in Java - Image Processing Project*,
<https://dyclassroom.com/image-processing-project/how-to-read-and-write-image-file-in-java> consulté le : 26 mai 2021

Impression rapports

- Stack Overflow, *java - How to export a complete JTable component in pdf at a specific coordinate*, <https://stackoverflow.com/questions/12785264/how-to-export-a-complete-jtable-component-in-pdf-at-a-specific-coordinate> consulté le : 23 mai 2021

- Stack Overflow, *swing - How to print jpanel in java?*,
<https://stackoverflow.com/questions/30703329/how-to-print-jpanel-in-java> consulté le : 24 mai 2021
- Stack Overflow, *java - Why does the JTable header not appear in the image?*,
<https://stackoverflow.com/questions/7369814/why-does-the-jtable-header-not-appear-in-the-image> consulté le : 24 mai 2021

Application exécutable

- Java67, *How to Make Executable JAR file in Eclipse IDE - Java* ,
<https://www.java67.com/2014/04/how-to-make-executable-jar-file-in-Java-Eclipse.html>
consulté le : 19 mai 2021

Base de données

- JM Doudoux, *Les bases de données et Eclipse*,
<https://www.jmdoudoux.fr/java/dejae/chap031.htm> consulté le : 21 mai 2021
- Documentation Oracle, *Getting Started with Java DB*,
<https://docs.oracle.com/javadb/10.10.1.2/getstart/index.html> consulté le : 21 mai 2021
- Stack Overflow, *Java embedded database persistence*,
<https://stackoverflow.com/questions/2162804/java-embedded-database-persistence>
consulté le : 22 mai 2021
- Stack Overflow, *mysql - java - DataSource for standalone application - no application server* ,
<https://stackoverflow.com/questions/15588449/java-datasource-for-standalone-application-no-application-server> consulté le : 22 mai 2021
- Developper.com, *Working with Embedded Databases in Java*,
<https://www.developer.com/database/working-with-embedded-databases-in-java/> consulté le : 22 mai 2021
- DZone Database, *An Overview of 3 Java Embedded Databases* ,
<https://dzone.com/articles/3-java-embedded-databases> consulté le : 22 mai 2021
- Stack Overflow, *Java Embedded Databases Comparison*,
<https://stackoverflow.com/questions/462923/java-embedded-databases-comparison>
consulté le : 22 mai 2021
- Vogella, *Managing databases with Eclipse and the Database Tools* ,
<https://www.vogella.com/tutorials/EclipseDataToolsPlatform/article.html> consulté le : 22 mai 2021
- Google, *Connect Database to Java - Embedded SQL Tutorial*,
<https://sites.google.com/site/embeddedsqltutorial/step-3-connect-database-to-java>
consulté le : 22 mai 2021
- W3Schools, *SQL LIKE Operator*, https://www.w3schools.com/sql/sql_like.asp consulté le : 27 mai 2021
- W3Resource, *MySQL not equal to operator*,
<https://www.w3resource.com/mysql/comparison-functions-and-operators/not-equal-operator.php> consulté le : 27 mai 2021

- Java Code Geeks, *Getting column names of a database table*, <https://examples.javacodegeeks.com/core-java/sql/getting-column-names-of-a-database-table/> consulté le : 27 mai 2021

MySQL

- Nanyang Technological University, *Java Tutorial - An Introduction to Java Database Programming (JDBC) by Examples with MySQL*, https://www3.ntu.edu.sg/home/ehchua/programming/java/JDBC_Basic.html consulté le : 6 mars 2021

Derby

- Apache, *Getting Started with Derby*, <http://db.apache.org/derby/docs/10.15/getstart/getstartderby.pdf> consulté le : 21 mai 2021

HSQldb

- HyperSQL, *HyperSQL User Guide*, <http://hsqldb.org/doc/2.0/guide/guide.pdf> consulté le : 22 mai 2021
- Java Code Geeks, *JDBC HSQldb Tutorial*, <https://examples.javacodegeeks.com/enterprise-java/sql-enterprise-java/jdbc-hsqldb-tutorial/> consulté le : 22 mai 2021
- Stack Overflow, *persistence - Is there a way to persist HSQldb data?*, <https://stackoverflow.com/questions/4128753/is-there-a-way-to-persist-hsqldb-data/4128844> consulté le : 23 mai 2021
- Oligofren, *HSQldb not persisting in file mode*, <https://oligofren.wordpress.com/2011/03/15/hsqldb-not-persisting-in-file-mode/> consulté le : 23 mai 2021

Tests unitaires

- Stack Overflow, *How to run all Junit tests of a given package?*, <https://stackoverflow.com/questions/2036750/how-to-run-all-junit-tests-of-a-given-package/36702048> consulté le 27 mai 2021
- Junit, *User guide*, <https://junit.org/junit5/docs/current/user-guide/> consulté le 27 mai 2021
- JM Doudoux, *Chapitre JUnit*, <https://www.jmdoudoux.fr/java/dej/chap-junit.htm> consulté le 27 mai 2021