



RAPPORT PROJET C++

MARCH 6

YNOV CAMPUS CASABLANCA

Réalisé par: Anass AJJA & Sanaa

GOULAHYANE & Gilchrist Randy

MAROC
ynov
CAMPUS

Sommaire

I.	Introduction	3
II.	Analyse des besoins	3
III.	Conception	4
1.	Classes	5
▪	La classe Article.....	5
▪	La classe Panier	5
2.	Fonctions et attributs	5
▪	Fonctions	5
▪	Attributs	5
IV.	Implémentation	6
	Captures d'écran	6
V.	Test et validation	11
VI.	Interface utilisateur	12
VII.	Conclusion	12
VIII.	Remerciements	13
IX.	References	13
X.	Annexe	15

I. Introduction

Le projet de **caisse automatique** vise à développer un logiciel simulant un système de caisse automatique pour un marché. Ce logiciel permettra aux utilisateurs de scanner des articles, de calculer le total des achats et de générer un reçu. L'objectif principal est de fournir une solution simple et efficace pour faciliter les transactions commerciales.

II. Analyse des besoins

Les exigences fonctionnelles du système comprennent :

- Capacité à scanner des articles à l'aide de codes-barres.
- Calcul automatique du total des achats, y compris les taxes.

-
- **Génération d'un reçu détaillant les articles achetés et le montant total.**
 - **Gestion des erreurs de saisie utilisateur pour assurer la fiabilité du système.**

Les exigences non fonctionnelles incluent :

- **Convivialité de l'interface utilisateur pour une utilisation intuitive.**
- **Fiabilité du système avec une gestion appropriée des erreurs.**
- **Performance optimale pour une réponse rapide aux interactions utilisateur.**

III. Conception

La conception du logiciel repose sur une architecture orientée objet.

1. Classes

Les classes principales sont Article et Panier.

- **La classe Article** représente un article avec des attributs tels que le code-barres, la désignation, le prix HT et l'heure de numérisation.
- **La classe Panier** gère une collection d'articles et fournit des fonctionnalités pour ajouter des articles, calculer le total des achats et générer un reçu.

2. Fonctions et attributs

- **Fonctions** : Implémentées pour effectuer des opérations telles que la validation des entrées utilisateur, le calcul des prix avec taxes et la génération de reçus.
- **Attributs** : Définis dans chaque classe pour stocker les données nécessaires, tels que le code-barres et le prix dans la classe Article.

IV. Implémentation

Le logiciel est implémenté en C++ en utilisant des concepts de programmation orientée objet. Les fonctionnalités telles que la validation des entrées utilisateur, le calcul des prix avec taxes et la génération de reçus sont réalisées à l'aide de méthodes dans les classes **Article** et **Panier**.

Captures d'écran

- Capture d'écran du class **Article**.

```

// Article class
class Article {
private:
    long long code_barre;
    std::string designation;
    double prixHT; // Price without tax
    std::string horaire_scan;

public:
    // Default constructor
    Article() {}

    // Parameterized constructor
    Article(int code, std::string name, double price, std::string time) : code_barre(code), designation(name), prixHT(price), horaire_scan(time) {}

    // Getter methods
    int getCodeBarre() const {
        return code_barre;
    }

    std::string getDesignation() const {
        return designation;
    }

    double getPrixHT() const {
        return prixHT;
    }

    // Calculate the price with tax
    double getPrixTTC() const {
        return prixHT * 1.2; // Add 20% tax
    }

    std::string getHoraireScan() const {
        return horaire_scan;
    }
};

```

- Capture d'écran du class **Panier**.

```

// Panier class
class Panier {
private:
    std::string date;
    std::vector<Article> articles;

public:
    // Default constructor
    Panier() {}

    // Setter for date
    void setDate(const std::string& date) {
        this->date = date;
    }

    // Getter for date
    std::string getDate() const {
        return date;
    }

    // Getter for articles
    const std::vector<Article>& getArticles() const {
        return articles;
    }

    // Check if the basket is empty
    bool estVide() const {
        return articles.empty(); // Return true if the basket is empty
    }
}

```

```

// Add an article to the basket and display the number of articles and total price
void ajouterArticle(const Article& article) {
    articles.push_back(article);
    std::cout << "\nArticle added: " << article.getDesignation() << std::endl; // Display the name of the added article
    std::cout << "Nombre d'articles in the basket : " << articles.size() << std::endl; // Number of articles in the basket
    std::cout << "Total price of the basket (TTC) : " << prixTotal() << " DH\n" << std::endl; // Total price of the basket
}

// Calculate the total price of the basket
double prixTotal() const {
    double total = 0.0;
    // Calculate the total price of the articles in the basket
    for (const auto& article : articles) {
        total += article.getPrixTTC();
    }
    return total;
}

// Print the receipt
void ticket() const {
    std::cout << BLUE << "\nMarjane market Receipt of " << date << " : \n";
    // Display the name and price of each article in the basket
    for (const auto& article : articles) {
        std::cout << "Article : " << article.getDesignation() << " | Price : " << article.getPrixTTC() << " DH\n";
    }
    std::cout << "Articles number : " << articles.size() << "\n";
    std::cout << "Total Amount : " << prixTotal() << " DH\n" << RESET;
}

```



```

// Calculate the duration of shopping
std::string dureeCourses() const {
    // Check if the basket is empty
    if (estVide()) {
        return "\033[31mNo articles scanned!\033[0m"; // Duration of shopping is 0 (empty basket)
    } else {
        std::string debut = articles.front().getHoraireScan();
        std::string fin = articles.back().getHoraireScan();

        int h1, m1, s1, h2, m2, s2;
        sscanf(debut.c_str(), "%d:%d:%d", &h1, &m1, &s1); // Extract the hours, minutes and seconds
        sscanf(fin.c_str(), "%d:%d:%d", &h2, &m2, &s2); // Extract the hours, minutes and seconds

        int debutSec = h1 * 3600 + m1 * 60 + s1; // Convert the time to seconds
        int finSec = h2 * 3600 + m2 * 60 + s2; // Convert the time to seconds
        int dureeSec = finSec - debutSec; // Calculate the duration in seconds

        int dureeH = dureeSec / 3600; // Convert the duration to hours
        dureeSec %= 3600; // Calculate the remaining seconds
        int dureeM = dureeSec / 60; // Convert the remaining seconds to minutes
        dureeSec %= 60; // Calculate the remaining seconds
        int dureeS = dureeSec; // Remaining seconds

        char buffer[100]; // Buffer to store the formatted string
        sprintf(buffer, "\033[34mShopping Duration : %02d:%02d:%02d\033[0m", dureeH, dureeM, dureeS); // Format
        return buffer;
    }
}
};

```

```

✓ bool isValidPrice(const std::string& price) {
✓     for (char c : price) {
        |     if (!isdigit(c) && c != '.') return false;
        |     }
        |     return true;
    }
}

✓ bool isValidCode(const std::string& code) {
✓     for (char c : code) {
        |     if (!isdigit(c)) return false;
        |     }
        |     return true;
    }
}

// Function to check if a string is a valid integer
✓ bool isValidNumber(const std::string& number) {
✓     for (char c : number) {
        |     if (!isdigit(c)) return false;
        |     }
        |     return true;
    }
}

```

- Capture d'écran de l'interface utilisateur montrant le processus de numérisation des articles.

```
PS C:\Users\luffy\OneDrive\Bureau\C C++> cd "c:\Users\luffy\OneDrive\Bureau\C C++\" ; i
f ($?) { g++ supermarket2.cpp -o supermarket2 } ; if ($?) { .\supermarket2 }
=====
                        Caisse automatique
=====
*****This console app is programmed by ASR Group*****

Purchase date: 2024-03-06

Enter the number of articles: 2

Enter the barcode for article 1: 218372464
Enter the name for article 1: Cocacola 2L
Enter the price HT for article 1: 16
The article Cocacola 2L is scanned at 14:04:51

Article added: Cocacola 2L
Nombre d'articles in the basket : 1
Total price of the basket (TTC) : 19.2 DH

If you want to exit, enter 00. To continue, enter Y or y key: Y

Enter the barcode for article 2: 18034729
Enter the name for article 2: Snickers
Enter the price HT for article 2: 7
The article Snickers is scanned at 14:05:03

Article added: Snickers
```

- Capture d'écran du reçu généré par le logiciel, affichant les détails des articles achetés et le montant total.

```
Marjane market Receipt of 2024-03-06 :  
Article :Coca2L | Price : 19.2 DH  
Article :Snickers | Price : 8.4 DH  
Articles number : 2  
Total Amount : 27.6 DH  
Shopping Duration : 00:00:15
```

V. Test et validation

Le logiciel est testé pour vérifier son bon fonctionnement conformément aux exigences spécifiées. Des tests unitaires sont réalisés pour chaque fonctionnalité, ainsi que des tests d'intégration pour s'assurer que toutes les parties du système fonctionnent correctement ensemble. Les résultats des tests confirment que le logiciel répond aux attentes et fonctionne comme prévu.

VI. Interface utilisateur

L'interface utilisateur du logiciel est conçue pour être conviviale et intuitive. Les utilisateurs peuvent facilement scanner des articles, visualiser le total des achats et générer des reçus. Des couleurs sont utilisées pour améliorer la lisibilité et guider l'utilisateur tout au long du processus.

VII. Conclusion

Le projet de caisse automatique a permis de développer un logiciel fonctionnel et efficace pour la gestion des transactions commerciales. Les défis rencontrés ont été surmontés grâce à une conception robuste et à une implémentation soignée. Des améliorations futures

peuvent être envisagées pour étendre les fonctionnalités et améliorer encore l'expérience utilisateur.

VIII. Remerciements

Je tiens à exprimer ma profonde gratitude à tout le groupe qui a contribué à la réalisation de ce projet de caisse automatique. Notre collaboration et notre travail d'équipe ont été essentiels à chaque étape du processus. Je tiens également à remercier MIFRAH Sara pour son leadership éclairé et ses précieux conseils tout au long du projet. Merci à tous pour votre engagement et votre expertise précieuse. Je suis reconnaissant envers chacun des membres du groupe pour leur soutien continu. Merci à tous pour votre dévouement et votre contribution précieuse.

IX. References

- Documentation C++ : <https://en.cppreference.com/w/>

-
- Tutoriels sur la programmation orientée objet :
<https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-cpp/>
 - Guides sur l'utilisation des bibliothèques standard C++ :
<https://www.cplusplus.com/reference/>

X. Annexe

- Diagramme de classe du système de caisse automatique.

