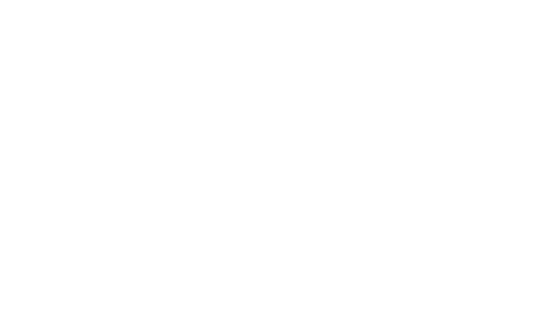


|  |
| --- |
| RAPPORT PROJET C++ |
|  |
| March 6  YNOV CAMPUS CASABLANCA  Réalisé par: Anass AJJA & Sanaa GOULAHYANE & Gilchrist Randy |



# Sommaire

[I. Introduction 3](#_Toc160628437)

[II. Analyse des besoins 3](#_Toc160628438)

[III. Conception 4](#_Toc160628439)

[1. Classes 5](#_Toc160628440)

[ La classe Article 5](#_Toc160628441)

[ La classe Panier 5](#_Toc160628442)

[2. Fonctions et attributs 5](#_Toc160628443)

[ Fonctions 5](#_Toc160628444)

[ Attributs 5](#_Toc160628445)

[IV. Implémentation 6](#_Toc160628446)

[Captures d'écran 6](#_Toc160628447)

[V. Test et validation 11](#_Toc160628448)

[VI. Interface utilisateur 12](#_Toc160628449)

[VII. Conclusion 12](#_Toc160628450)

[VIII. Remerciements 13](#_Toc160628451)

[IX. References 13](#_Toc160628453)

[X. Annexe 15](#_Toc160628454)

# 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.

# 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.

# Conception

La conception du logiciel repose sur une architecture orientée objet.

## 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.

## Fonctions et attributs

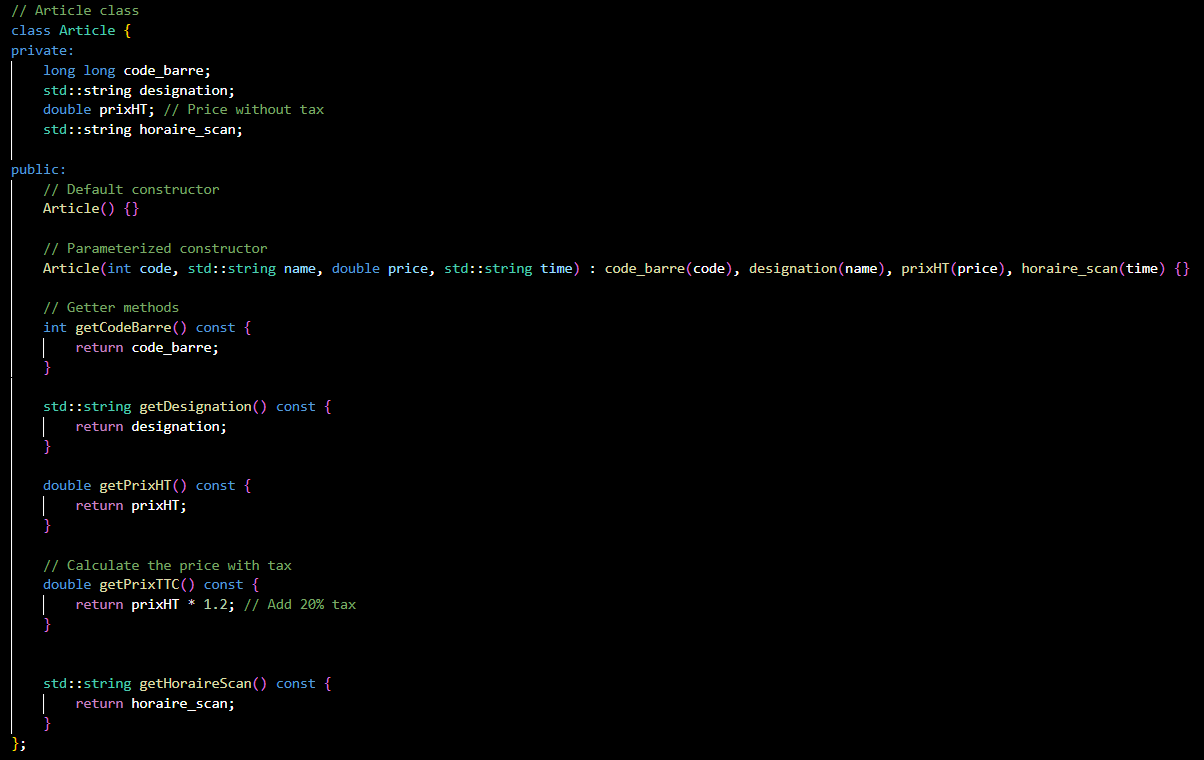
* Fonctions : Implémentées pour effectuer des opérations telles que la validation des entrées utilisateur, le calcul des prix avec taxes et la génération de reçus.
* Attributs : Définis dans chaque classe pour stocker les données nécessaires, tels que le code-barres et le prix dans la classe Article.

# 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.



* Capture d'écran du class Panier.

A screenshot of a computer program

Description automatically generated

A computer screen shot of a program code

Description automatically generated

A computer screen shot of text

Description automatically generated

A computer screen shot of code

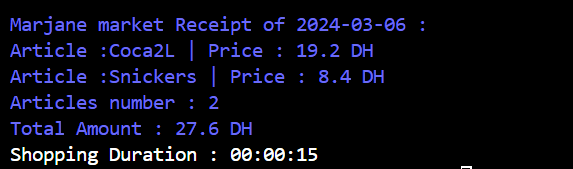
Description automatically generated

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

A screenshot of a computer screen

Description automatically generated

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



# 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.

# 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.

# 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.

# 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.

# 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/>

# 

# Annexe

* Diagramme de classe du système de caisse automatique.

