Caf

Café CDI 3

Phoumano BOUNMA, Philippe LY

École Normale Supérieure de Cachan

Formation CDI 3 - 2016

# SOMMAIRE

[SOMMAIRE 2](#_Toc444609843)

[REMERCIEMENTS 3](#_Toc444609844)

[INTRODUCTION 4](#_Toc444609845)

[PRÉSENTATION DU CONTEXTE 5](#_Toc444609846)

[LE PROJET 6](#_Toc444609847)

[Description du projet 6](#_Toc444609848)

[Description des besoins 6](#_Toc444609849)

[Front Office 6](#_Toc444609850)

[Back Office 6](#_Toc444609851)

[Méthode de gestion de projet 6](#_Toc444609852)

[Les choix techniques 6](#_Toc444609853)

[Les fonctionnalités 7](#_Toc444609854)

[Les maquettes d’interface 7](#_Toc444609855)

[Les données nécessaires et à mémoriser 7](#_Toc444609856)

[Les traitements 7](#_Toc444609857)

[La structure de l’application 7](#_Toc444609858)

[Les plans de tests 7](#_Toc444609859)

[Le déploiement 8](#_Toc444609860)

[La documentation 8](#_Toc444609861)

[Bilan du projet 8](#_Toc444609862)

[CONCLUSION 9](#_Toc444609863)

# REMERCIEMENTS

# INTRODUCTION

# PRÉSENTATION DU CONTEXTE

# LE PROJET

## Description du projet

Le projet Café CDI 3 est une simulation de machine à café. Il consiste à gérer les principaux mécanismes d’une machine à café : la préparation et distribution de boissons chaudes (café, thé, potage…), et surtout la gestion de la monnaie échangée entre la machine et le consommateur. C’est sur cette-dernière qu’est centré le sujet proposé par la formation continue. Néanmoins, le projet que nous allons réaliser ne se limite pas à cela et vise à simuler une machine à café aussi complète que possible.

## Description des besoins

### Front Office

L’énoncé du sujet donne une description succincte mais claire des besoins en ce qui concerne les fonctionnalités accessibles au consommateur. Celui-ci doit pouvoir sélectionner la boisson qu’il veut boire en appuyant sur un des boutons représentant les produits proposés par la machine. Au préalable il doit avoir introduit dans le monnayeur la somme requise pour obtenir ladite boisson, (dans notre cas, on suppose que chaque consommation coute 50 centimes). Un bouton d’annulation doit lui permettre de récupérer son argent s’il change d’avis avant d’avoir fait son choix.

### Back Office

Tout le reste se fait en arrière-plan, hors de la vue du client, en particulier tout ce qui concerne l’argent. Là encore, l’énoncé est très clair sur ce qui est demandé pour le monnayeur. Celui-ci gère la caisse de la machine, c’est-à-dire l’argent que celle-ci contient, ainsi qu’un compteur qui retient l’argent introduit depuis la dernière commande effectuée. Cet argent doit être contrôlé et la machine doit servir la boisson seulement s’il y a assez d’argent dans le compteur. Dans le cas contraire, la machine doit indiquer au client la somme qu’il lui reste à ajouter. Enfin, le monnayeur doit, le cas échéant, pouvoir calculer l’argent rendu avec les bonnes pièces.

Le programme développé doit gérer deux machines différentes.

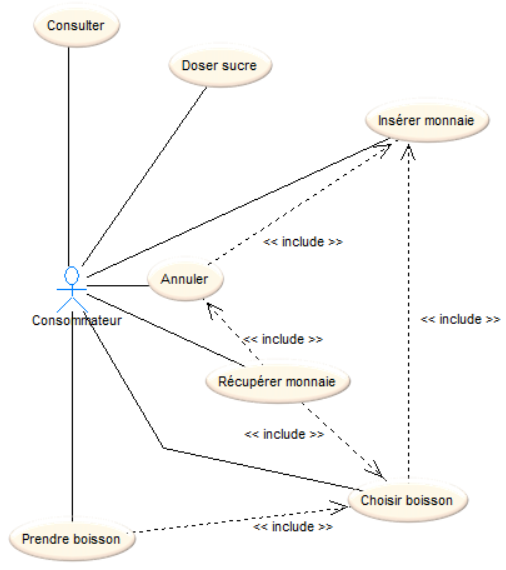
## Méthode de gestion de projet

MS Project [encore rien fait]

## Les choix techniques

L’application sera développée en langage C#.Net sous Visual Studio, un langage et un IDE que nous avons choisi simplement par familiarité (et surtout parce que le contexte actuel l’imposait). La gestion de version se fera avec GitHub. [à développer]

## Les fonctionnalités



## Les maquettes d’interface

|  |  |  |
| --- | --- | --- |
| **Cas d’utilisation** | Consulter | Doser sucre |
| **Acteur** | Consommateur | Consommateur |
| **Evénement déclencheur** | Néant | Néant |
| **Intérêts** | Connaître les boissons proposées par la machine | Choisir la quantité de sucre dans sa boisson |
| **Préconditions** | Machine en marche | Machine en marche |
| **Post-conditions** |  |  |
| **Scénario nominal** | 1. Le consommateur appuie sur une boisson |  |
| **Extensions** |  |  |
| **Contraintes** |  |  |

|  |  |  |
| --- | --- | --- |
| **Cas d’utilisation** | Insérer monnaie | Annuler |
| **Acteur** | Consommateur | Consommateur |
| **Evénement déclencheur** | Néant | Néant |
| **Intérêts** | Payer la boisson choisie | Annuler une commande avant d’avoir choisi |
| **Préconditions** | Machine en marche | De l’argent a été inséré dans le compteur |
| **Post-conditions** |  | L’argent contenu dans le compteur est libéré |
| **Scénario nominal** | 1. Le consommateur insère une pièce  2. Le monnayeur accepte la pièce | 1. Le consommateur appuie sur le bouton annuler  2. |
| **Extensions** |  |  |
| **Contraintes** | 2a. La pièce n’est pas acceptée : aller en 1  2b. La caisse est pleine : aller en 1 | 2a. Une boisson est en préparation. |

|  |  |  |
| --- | --- | --- |
| **Cas d’utilisation** | Récupérer monnaie | Prendre boisson |
| **Acteur** | Consommateur | Consommateur |
| **Evénement déclencheur** | Néant | Néant |
| **Intérêts** | Payer la boisson choisie | Commander la boisson ou annuler |
| **Préconditions** | Machine en marche | Machine en marche |
| **Post-conditions** |  |  |
| **Scénario nominal** | 1. Le consommateur insère une pièce  2. Le monnayeur accepte la pièce |  |
| **Extensions** |  |  |
| **Contraintes** | 2a. La pièce n’est pas acceptée : aller en 1  2b. La caisse est pleine : aller en 1 |  |

## Les données nécessaires et à mémoriser

Nous désirons faire en sorte que l’état des machines soit mémorisé lorsque l’application est fermée. Cependant, celle-ci ne fait intervenir qu’une quantité très faible de données. C’est la raison pour laquelle il a été décidé que ces données seraient conservées par sérialisation en fichier XML.

## Les traitements

[ex : Diagramme de séquence]

## La structure de l’application

[Diagramme de packages]

## Les plans de tests

## Le déploiement

[Diagramme de déploiement]

## La documentation

[Docu technique et docu d’utilisation]

## Bilan du projet

[Problèmes, solutions, améliorations pour l’avenir]

# CONCLUSION