



Projet .NET

APPLICATION DE COMMANDE

Farah Zoghlami | Ayhann | Celian Picot | Tiona

Introduction

OBJET DU DOCUMENT

Ce document décrit les objectifs et les mises en œuvre de l'équipe pour développer ladite application.

OBJECTIF DU DOCUMENT

L'objectif du document est de mettre en lumière les périmètres, les atouts, les limites du logiciel ainsi que de révéler la gestion de projet de l'équipe.

Objectifs

SITUATION

En restauration, l'attente du client est une chose admise par tous. Celle-ci peut être stressante pour la personne qui attend, tout comme pour les serveurs et employés de l'établissement qui doit se démener pour servir et prendre commande au plus vite.

NATURE DE LA PRESTATION

Le logiciel est une application à installer sur tablette à chaque table pour commander des plats dans un restaurant afin de remédier, ou tout du moins réduire la pénibilité des personnes concernées. De plus, le client est autonome dans sa prise de commande.

ENVIRONNEMENT

La restauration rapide a déjà la main mise sur une technologie similaire. Des bornes sont disposées à l'entrée de l'établissement, permettant aux clients qui le souhaitent de commander leur nourriture sans ou peu de file d'attente.

Dans certains autres restaurants, après avoir passé commande, le client reçoit un objet connecté (un bipeur) et peut aller s'installer à table. Le bipeur, déclenché par le personnel, prévient le client lorsqu'il peut aller chercher ses plats.

Ses solutions permettent de réduire le stress des employés et des clients. Celles-ci sont cependant constituées de plusieurs étapes :

- Prise de commande (généralement debout).
- Attente à table.
- Aller récupérer son plat.

Nous pouvons d'ores et déjà dire que notre solution améliorera les systèmes déjà en place sur le marché de la concurrence.

Cibles

Les restaurants visés sont essentiellement les établissements de la gastronomie traditionnelle. La part de marché représente 63% dans l'Hexagone (source inkidata). Une cible donc assez importante malgré l'accroissement des franchises issues de la restauration rapide.

Périmètre du projet

LES MODULES

L'application comprend deux modules distincts :

- La commande des plats : le client sélectionne les plats qu'ils souhaitent
- L'administration de l'application : le restaurateur peut ajouter, modifier ou supprimer des plats.

LES FONCTIONNALITES

Module « Commande »

Les fonctionnalités mises en place côté client sont les suivantes :

- Sélectionner, supprimer des plats, des menus, des boissons.
- Filtrer selon les critères suivants : végétarien, sans gluten.
- Envoi des informations de commande.

Module « Administration »

Les fonctionnalités mises en place côté administration sont :

- Ajouter, modifier et supprimer les plats.
- Visionner les commandes effectuées par table.

BASE DE DONNEES

Dictionnaire de données

Le dictionnaire de données contient toutes les données relatives aux tables futures de la base de données.

Structure de la table clients

Colonne	Type	Null	Valeur par défaut
idClient	int(10)	Non	
nomClient	varchar(255)	Non	

Structure de la table commandes

Colonne	Type	Null	Valeur par défaut
idCommande	int(10)	Non	
nomCommande	varchar(255)	Non	
coutCommande	float	Non	
etatCommande	enum('annulé', 'payé', 'en cours')	Oui	NULL
id_panier	int(10)	Non	
id_client	int(10)	Non	
id_table	int(10)	Non	

Structure de la table consommations

Colonne	Type	Null	Valeur par défaut
idConso	int(10)	Non	
nomConso	varchar(255)	Non	
prixConso	float	Non	
detailConso	varchar(255)	Non	
id_type	int(10)	Non	

Structure de la table paniers

Colonne	Type	Null	Valeur par défaut
idPanier	int(10)	Non	
quantite	int(10)	Non	
id_conso	int(10)	Non	

Structure de la table tables

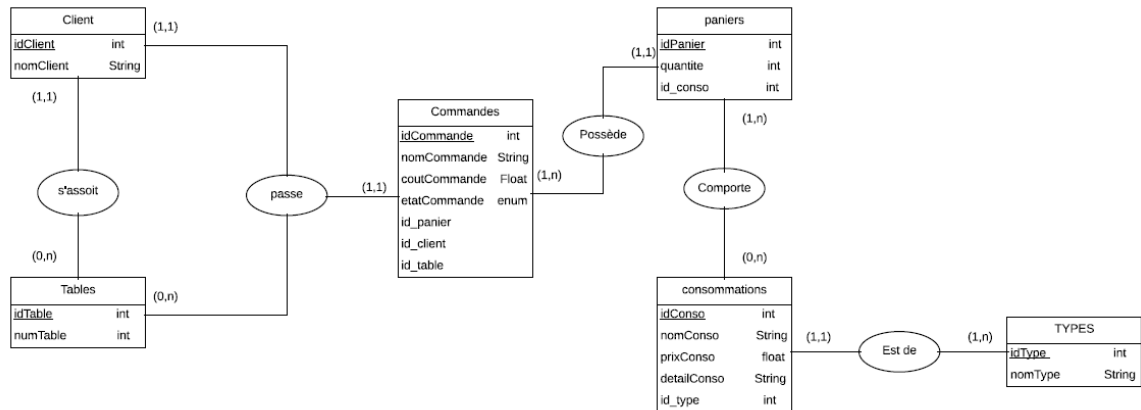
Colonne	Type	Null	Valeur par défaut
idTable	int(10)	Non	
numTable	int(10)	Non	

Structure de la table types

Colonne	Type	Null	Valeur par défaut
idType	int(10)	Non	
nomType	varchar(255)	Non	

Modèle Conceptuel de Donnée (MCD)

Le MCD est une représentation graphique de la base de données.



Gestion de projet

CHOIX DES OUTILS

Les outils utilisés par l'équipe projet sont représentés dans un tableau ci-dessous.

Langage	C#
IDE	Visual Studio
Base de données	PhpMyAdmin - mySql
Gestion d'équipe	Trello / GitHub

Mise en œuvre

EXEMPLE DE CODES

- Classe permettant la connexion à l'application.

```
1 référence
private void btnIdentifier_Click(object sender, EventArgs e)
{
    //Instanciation de la chaine de connexion
    string connexionString = "server=localhost;user=root;database=restoapp;persistsecurityinfo=True;port=3306;password=root;SslMode=None";
    MySqlConnection conn = new MySqlConnection(connexionString); //Implémentation d'un objet de type MySqlConnection
    try
    {
        conn.Open(); //Ouverture de la connexion
        string requete = "INSERT INTO clients (nomClient) VALUES (@nom)";
        MySqlCommand command = new MySqlCommand(requete, conn); //Implémentation d'un objet de type MySqlCommand
        command.Parameters.AddWithValue("@nom", tbxNom.Text); //Ajout de la valeur de la textBox au paramètre @nom
        command.ExecuteNonQuery(); //Execution de cette requête
        Main m = new Main(tbxNom.Text); //Ouverture d'une nouvelle fenêtre
        m.ShowDialog();
    }
    catch (MySqlException ex)
    {
        MessageBox.Show(ex.Message); //Si l'insert se passe mal on renvoie un message d'erreur
    }
}

1 référence
private void FrmIdentification_Load(object sender, EventArgs e) => btnIdentifier.Enabled = false; //Désactivation du bouton au démarrage

1 référence
private void btnClose_Click(object sender, EventArgs e) => this.Close(); //Fermeture de la fenêtre si on appuie sur la X

1 référence
private void tbxNom_TextChanged(object sender, EventArgs e)
{
    if (tbxNom.TextLength == 0)
    {
        btnIdentifier.Enabled = false; //Désactive le bouton si aucun identifiant n'est rentré
    }
    else
    {
        btnIdentifier.Enabled = true; //Active sinon
    }
}
```

- Page principale

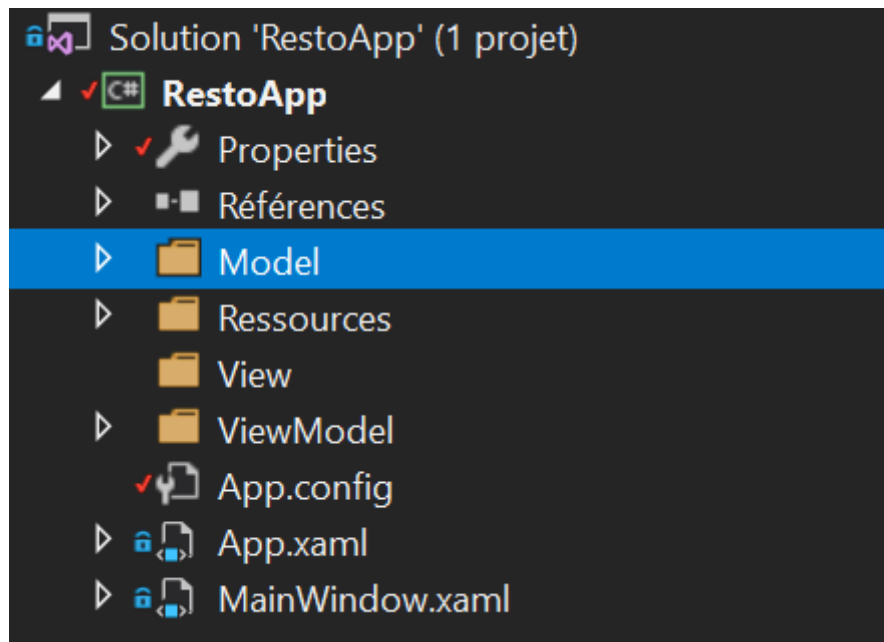
```
1 référence
private void Main_Load(object sender, System.EventArgs e)
{
    string connexionString = "server=localhost;user=root;database=restoapp;persistsecurityinfo=True;port=3306;password=root;SslMode=None";
    MySqlConnection conn = new MySqlConnection(connexionString);
    try
    {
        conn.Open();
        string requete = "SELECT nomConso, prixConso FROM consommations WHERE id_type=1"; //Recherche toutes les entrées
        MySqlCommand command = new MySqlCommand(requete, conn);
        MySqlDataReader reader = command.ExecuteReader(); //Lis les données
        while (reader.Read()) //Pour chaque entrée
        {
            lbEntree.Items.Add(reader.GetString(0) + " - " + reader.GetFloat(1) + "€"); //On ajoute le nom et le prix à une liste déroulante
        }
    }
    catch (MySqlException ex)
    {
        MessageBox.Show(ex.Message); //On renvoie un message d'erreur s'il n'y arrive pas
    }
    finally
    {
        conn.Close(); //Fermeture de la connexion
    }
    try
    {
        conn.Open();
        string requete = "SELECT nomConso, prixConso FROM consommations WHERE id_type=2"; //Recherche tous les plats
        MySqlCommand command = new MySqlCommand(requete, conn);
        MySqlDataReader reader = command.ExecuteReader(); //Lis les données
        while (reader.Read()) //Pour chaque plat
        {
            lbPlat.Items.Add(reader.GetString(0) + " - " + reader.GetFloat(1) + "€"); //On ajoute le nom et le prix à une liste déroulante
        }
    }
    catch (MySqlException ex)
    {
        MessageBox.Show(ex.Message); //On renvoie un message d'erreur s'il n'y arrive pas
    }
    finally
    {
        conn.Close(); //Fermeture de la connexion
    }
}

1klblAccueil.Enabled = false;
```

ARCHITECTURE

Le pattern choisi MVVM « Model View ViewModel » est une architecture beaucoup utilisée en C# dans la programmation objet.

La vue (View) est constituée des éléments graphique qui composent l'interface utilisateur.



Evolution du produit

COTE CLIENT

Plusieurs améliorations ont été proposé lors du projet. Du côté client, nous pourrions proposer une application complémentaire qui proposerait un compte client.

Celui-ci contiendrait les fonctionnalités suivantes :

- Des points de fidélités avec des avantages selon le nombre accumulé de visites.
- La possibilité pour le client de voir si une table est libre avant même d'arriver au restaurant.

COTE ADMINISTRATION

Une interface permettant de gérer la carte de son restaurant.

En effet, un tableau bord intuitif pour ajouter, supprimer, modifier des produits à la carte.

À la suite de cela, on peut réfléchir également à une gestion du stock.

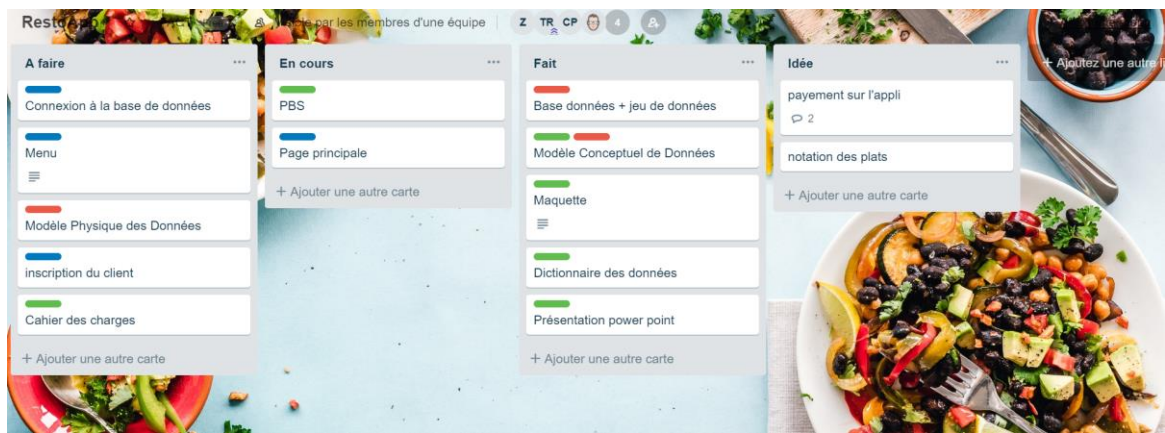
Ce qui permettra de savoir en temps réelle la disponibilité des plats.

On pourra également recueillir ses informations afin d'avoir une estimation plus précise pour passer une commande auprès des fournisseurs.

Gestion de projet

ORGANISATION DES TACHES

L'équipe projet se sert de l'outil Trello précédemment cité pour s'organiser dans le reste des travaux à fournir.



Les couleurs ont chacune leur signification :

- Le rouge représente les travaux concernant la base de données.
- Le vert désigne les tâches écrites à produire.
- Le bleu désigne le développement des modules de l'application.

ORGANISATION DE LA PRODUCTION

L'équipe possède un git commun pour que chacun puisse envoyer son travail au même endroit, et ainsi, constater le travail de chacun, pouvoir faire du code review, travailler en équipe en somme.

Deux branches distinctes ont été créés :

- DEV : le travail en développement et en test.
- MASTER : le produit avec des fonctionnalités qui fonctionnent et qui ont été revu et validé par toute l'équipe.