

Projet Fis'EAT

Barbary Théo
Thiry Basile



Sommaire

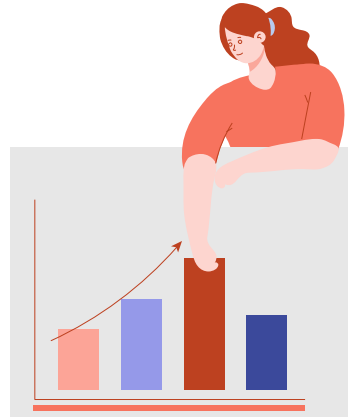
1 Choix de conception / Architecture

2 Répartition des tâches

3 Les options en plus

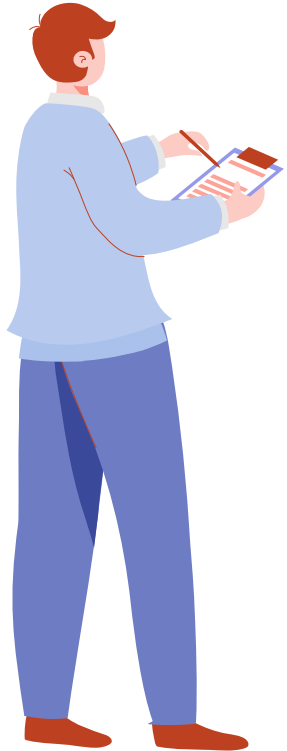
4 Analyse critique

5 Démonstration



Choix de conception et architecture

3 grandes parties



Questions :

- Quels éléments ajouter dans le modèle
- Comment intégrer l'aspect préparation des commandes
- Comment persister les données

Modèle

Cuisine

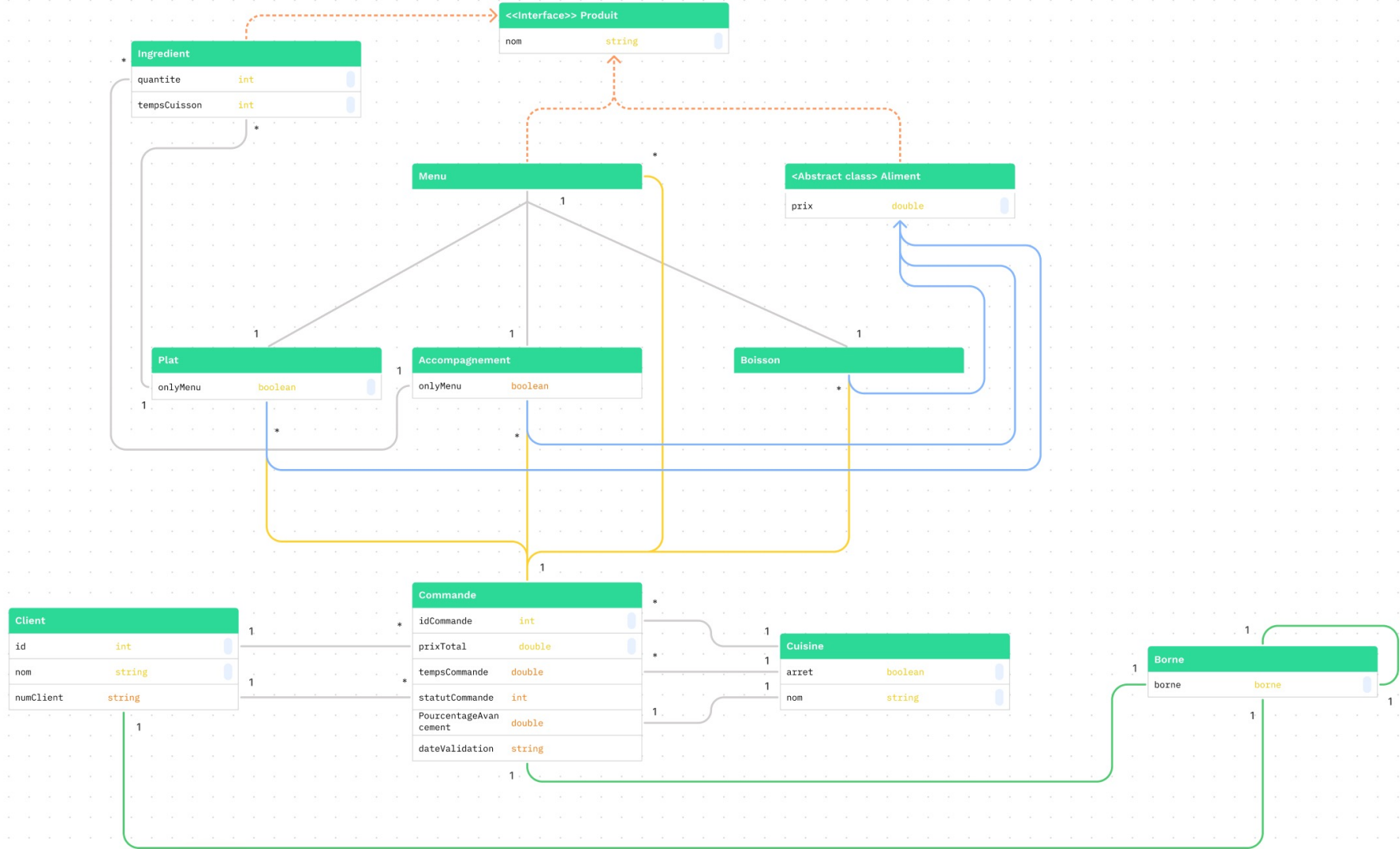
Persistence

Fichier JSON ?

Fichier binaire ?

BDD ?

Fichier binaire

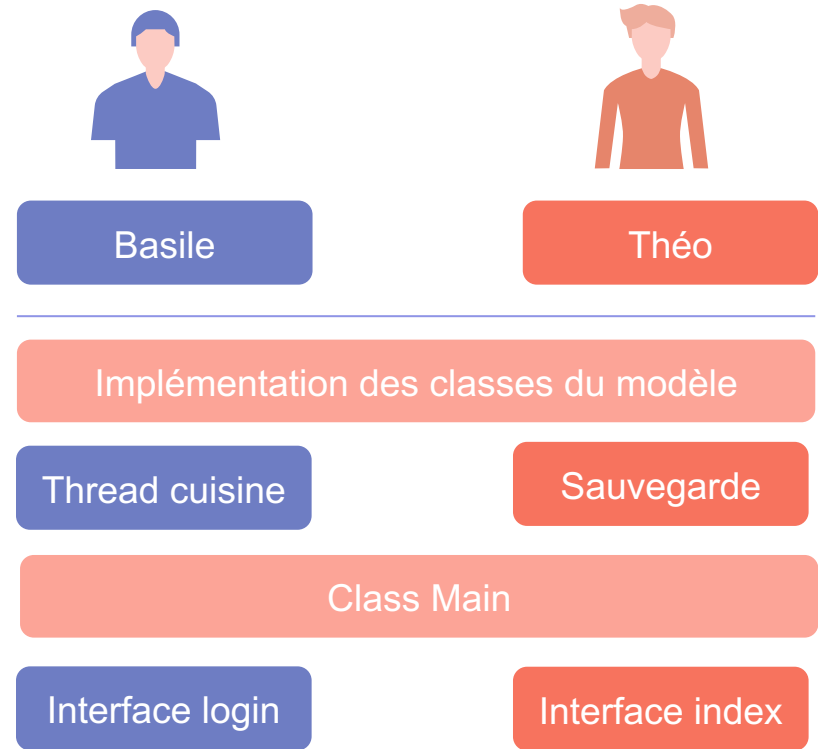


Répartition des tâches

Compétences rapport avec le projet :

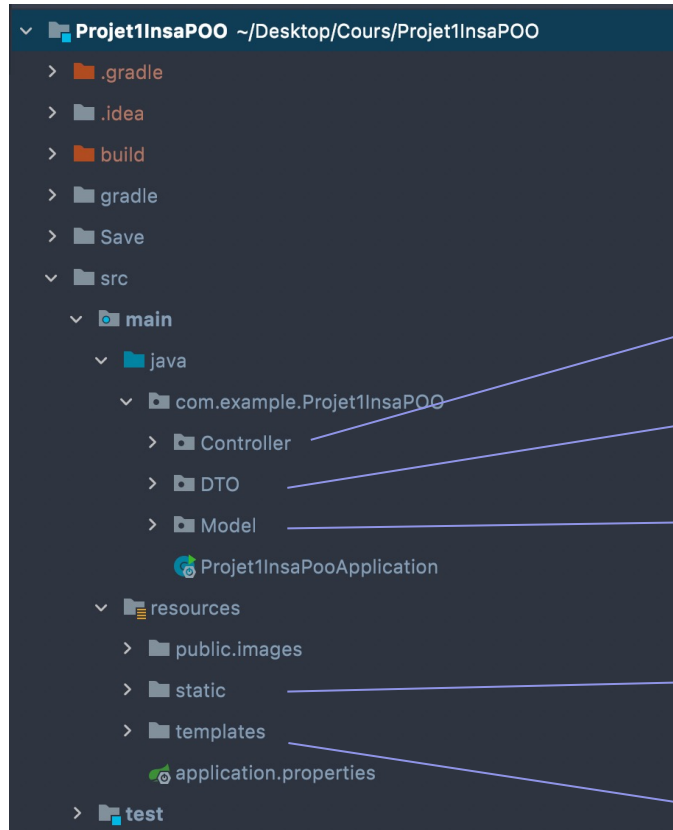
BARBARY Théo	THIRY Basile
Sauvegarde	Thread
Logique POO	Logique POO
Java	Java
Design Pattern	Java FX
Spring Boot	Gradle

Répartition des tâches en fonction de nos compétences :



Options supplémentaires

Utilisation de Spring Boot MVC



Controller : lien entre le front et le back

DTO : class permettant une transmission d'objet plus simplement

Fichier du model

Fichier css / js

Fichier html

Page de connexion

Vous avez un compte ?
Veuillez indiquer votre identifiant

login

Se connecter

Pas de compte ?
Inscrivez-vous en indiquant votre futur identifiant

Nom

Prénom

S'inscrire

Etat des commandes :

Commande : 0 à 100%

Page de principale

Bienvenue Barbary

Faites votre choix :

Commander un menu

Commander un plat seul

Commander une boisson seul

Commander un accompagnement seul

Historique des commandes

Se déconnecter

Commande actuelle :

Prix total : 0.0€

Modifier la commande

Payer la commande

Analyse critique

Analyse critique

Difficultés techniques

Plusieurs implémentations
des Thread

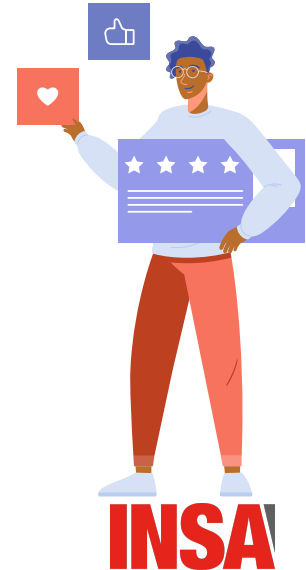
Affichage des cuisines

Bien optimiser le modèle

Difficultés d'organisation

Bien se répartir les tâches

Débattre sur l'implémentation
de la sauvegarde



Démonstration

