# Thème

Friterie

# Expression des besoins

|  |  |
| --- | --- |
|  |  |
| Gérer client | Client / (salarié) |
| Gérer des fournisseurs | Marchandise / Consommable |
| Gestion des planning | (Salariés) / Horaires ouvertures |
| Gérer les factures | Ticket client |
| Gérer entrée sortie | Gérer les stocks |

# Description

Application de gestion des commandes des clients et des stocks (entrées/sorties).

# Première ébauche

1er Etape : Login

2ème Etape :

* ngif employé**:**  Ecran de vente ( gestion des clients/commande )
* ngIf admin: Ecran administration (stock / marchandise / planning )

## Page employé

Saisie des commandes

* Formulaire de commande

Services utilisés

* Mise à jour du stock à encaissement
* Edition facture
* Total des produits/ paiement commande

## Page admin

Plusieurs onglets (gestion stock / gestion planning / bilan comptable : Statistiques (/ Gestion des salariés))

* Onglet Gestion Stock
  + Formulaire CRUD Stock
* Onglet Planning
  + Gestion des horaires d’ouverture (période de congés)
* Onglet Bilan Comptable
  + Chiffres d’affaires (journa/hebdo/mensuel/annuel)

# Architecture

src/

core/

personne/

personne.model

personne.service

use api.service

personne.module

marchandise/

marchandise.model

marchandise.service

use api.service

marchandise.module

stock/

stock.model

stock.service

use api.service

stock.module

planning/

planning.model

planning.service

use api.service

planning.module

api/

api.service

use httpClient

comptable/

comptable.service

user/

components/

login.component

userbar.component

user.model

user.service

user.module

core.module

client/

components/

client.list.component

client.component

client.popin

client.model

extend personnage.model

client.service

use personnage.service

client.module

client.route

ticketClient/

components/

ticketClient.list.component

ticketClient.component

ticketClient.popin

ticketClient.model

ticketClient.service

ticketClient.module

ticketClient.route

employé/

components/

employé.list.component

employé.component

employé.popin

employé.model

extend personnage.model

employé.service

use personnage.service

employé.module

employé.route

fournisseur/

components/

fournisseur.list.component

fournisseur.component

fournisseur.popin

fournisseur.model

extend personnage.model

fournisseur.service

use personnage.service

fournisseur.module

fournisseur.route

admin/

components/

admin.list.component

admin.component

admin.popin

admin.model

extend personnage.model

admin.service

use personnage.service

admin.module

admin.route

shared

components/

layout.component

modal.component

flex.dualcol.component

pipes/

date.pipe

directives/

shared.module

# Dictionnaire des données

## Class

Personne { id, nom, prenom, adress, tel, email, dateNaissance }

Client { super(), dateFidélisation } extends Personne

User { super(), role } extends Personne

Admin { super(), login, mdp } extends User

Employé { super(), login, mdp } extends User

# Question à poser

Comment lier les modules ensemble ?

Ebauche ok ?

Rôle pour le login ? ( Attribut ou class )