**Thème du Projet : Gestion Fastfood / Restaurent**

**Introduction :**

Notre logiciel a pour but la gestion d'un établissement de restauration.

**Acteurs :**

* Un administrateur pour administrer le logiciel.
* Un caissier pour prendre les commandes et effectuer le paiement.
* Un cuisinier pour préparer les commandes.

Lors d’une prochaine évolution on intègrera deux acteurs supplémentaires :

* Un serveur pour prendre les commandes et effectuer le service à table.
* Un client régulier (on pourra éventuellement lui proposer des offres promotionnelles, d’enregistrer ses préférences, réservation de table,…)

**Interfaces :**

* Interface web pour l'administration.
* Interface web pour la caisse pour enregistrer les commandes et effectuer le paiement.
* Interface mobile pour la cuisine pour préparer directement les commandes.
* Interface web pour affiché les numéros de commande en cours de préparation.
* Interface mobile pour serveur pour prend la commande directement à table.

**Actions :**

* Rajouter des tables, des menus, fixer les prix…
* Prendre la commande du client.
* Encaisse la somme due.
* Valider la commande.
* Indiquer que la commande est prête.

**Technologies à utiliser :**

Ide :

* Eclipse OXYGEN - IDE for Java EE Developers

Coté serveur :

* Maven
* Hibernate
* Spring MVC
* Spring Security

SGBD:

* Mysql
* Client workbech ou phpmyadmin

 Coté Front :

* Angular 2.

Versionning :

* GIT
* On

Accueil Menu Produits Shopping carte

Menu: Liste des Menus Liste des produits par catégorie

Boutons: pour chaque produit ou menu ajouter à shopping cart

shopping cart : supprimer élément devant chaque élément / checkout->Page Identification/continuer commande

Identification: Nom Prenom adresse Tel mail mdp tous les champs sont requis/ si enregistrement/connexion Mel mdp

Gestion des roles si admin si Personnel si caissier pour le client caisse en mode appli web. si utilisateur pour le client interface commande.

Le diagramme de classe après analyse des besoins:MCD BD

Produit:            Menu               Utilisateur          Commande

-Category         Produits            Category           -Total

-Prix                                                                 -Produits

                                                                        -Utilisateur

1-     établire un cahier de change, un document word qui contiens des mockup avec des petites explication sur l'interface et les fonctionalitées, aucune notion technique mais juste l'interface et les fonctionalitées du projet,

2-     découpé le cahier de charge en version, et faire un découpage des versions a realisé, l'estimation en temps n'est pas vraiment importante dans ce genre de projets non benifique

3-     commençant par le premier version, et la faut faire une petite conception le minimum c'est le diagramme de classe pour le schema de la base de données

4-     choisir et decider que technologies / bibliotheque / SGBD faut travailler avec

5-     découpe le premier version en ensemble de taches independante avec le tehmps moyen de la tache entre 2h a 8h,

6-     met un ordre de prioritées des taches

7-     chaque dev prend une tache selon l'ordre de prioritée et il la il faut un outil de gention de ticket, trello peut faire l'affaire ou d'autre outil, et donc quand un dev prend un tiket il marque son nom dans le tiket pour que les autres ne font pas le meme travail

8-     une fois le dev termine le ticket il faut un push avec git dans une branche spéciale au ticket, le responsable de projet va lire son code et il demendara des modification jusqu'a ce que le tiket sera accepté

9-     une fois le ticket sera accepte le responsable de projet (admin) megera le ticket dans la branche principale du projet (master) et marquera que le tiket est terminé

10-  un fois tous les tickets sont terminé la version sort, et la c'est le role des testeurs qui testent l'ensemble des fonctionalitée et si y a des bug, ils creent des tikets,

11-  les devs prend des tikets de bug et de la meme meniere ils corrigent et les admin acceptent

12-  une fois tous les tests sont corrigé on passera de la version beta vers la version stable, et on commencera la prochaine version de la meme manière