IMPORTANT :
Le rafraîchissement des tables se fais toutes les **30** secondes. Pour modifier cette valeur cherchez la ligne :

```
timer.schedule(new Refresh(this.tables, v), 0, 30000);
```

Dans les fichiers **ControleurPaiment.java, ControleurReservation.java, ControleurTables.java**

Le mot de passe par défaut pour l'application de paiement est **root** .

Rapport du projet d'IHM

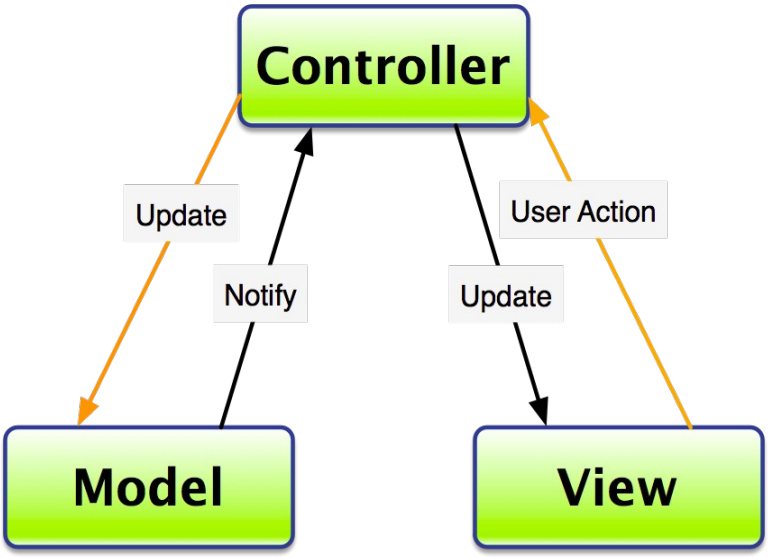
Présentation

Pour ce 3ème semestre nous avons dû réaliser trois applications pour la gestion d'un restaurant. Ces applications sont réalisées afin de mettre en pratique nos connaissances sur le modèle MVC ainsi que sur le Interfaces Hommes-Machines. Ces 3 applications sont organisées de la manières suivante :

- **Gestion des réservation** : la/le réceptionniste peux placer des réservations récupérées dans une base de données. Elle/Il peut aussi placer des nouveaux clients n'ayant pas de réservation. Pour placer les client l'utilisateur doit sélectionner des tables libres et valider sa sélection. Une fois la sélection validée les table sont notées comme occupées et dans le cas d'une réservation le nom du client est affiché.
- **Gestion du nettoyage des table** : le personnel de salle peux voir les tables à débarrasser et indiquer qu'elles ont été nettoyées. La table nettoyée est ensuite replacée a son emplacement initiale et indiquée comme libre.
- **Gestion du paiement** : le responsable du restaurant peux encaisser des clients en indiquant le montant à payer ainsi que le mode de paiement. Pour cela il doit sélectionner le groupe de table, entrer le montant, sélectionner un mode de paiement et valider. Une fois valider les tables sont à nettoyées.

Mes choix

Pour créer mon modèle MVC j'ai fais le choix de suivre ce schéma ci :



J'ai suivi ce schéma car le schéma qui est utilisé dans beaucoup de Framworks PHP comme Laravel qui m'est familier. J'ai choisi aussi d'organiser mon code de la manière suivant :

- Des fichiers principaux contenants chacun une méthode main() qui permet d'initialiser l'application en chargeant les vues, les contrôleurs et les modèles.
- Un fichier commun contenant la classe Refresh qui se charge de l'actualisation de l'état des tables.
- Un fichier commun à toutes les application qui contrôles les actions sur les tables.
- Les modèles qui se chargent de récupérer les données dans la base de données.
- Les contrôleurs qui servent à manipuler et à écouter les différents éléments des Vue.
- Les vues qui génèrent l'interface graphique de l'application.

Organisation

Application de Paiement :

Paiement.java contient la méthode main().
Refresh.java contient la méthode d'actualisation des tables.
Table.java afficheur d'une table.
Controleur.java classe mère des contrôleurs.
Histogramme.java afficheur d'histogramme.
ModeleTable.java modèle pour les tables.
ModelePaiement.java modèle pour le paiement et les statistiques.
ControleurPaiement.java contrôleur de l'interface.
ControleurTables.java contrôleur pour les tables.
VuePaiement.java vue du paiement.
VueStatistiques.java vue des statistiques.
VueLogin.java vue du login.

Application de Réservation :

Reservation.java contient la méthode main().
Refresh.java contient la méthode d'actualisation des tables.
Table.java afficheur d'une table.
Controleur.java classe mère des contrôleurs.
ModeleTable.java modèle pour les tables.
ModeleReservation.java modèle pour les réservations.
ControleurReservation.java contrôleur de l'interface.
ControleurTables.java contrôleur pour les tables.
VueReservation.java vue de la réservation.

Application de Nettoyage :

Nettoyage.java contient la méthode main().
Refresh.java contient la méthode d'actualisation des tables.
Table.java afficheur d'une table.
Controleur.java classe mère des contrôleurs.
ModeleTable.java modèle pour les tables.
ControleurTables.java contrôleur pour les tables.
VueNettoyage.java vue du nettoyage.