Rapport itération n°3

Groupe 10

Refactoring MVC

Lors des itérations précédentes, notre application ne respectait pas correctement le pattern Model View Controller (*MVC*). Un large refactoring a donc été effectué pour résoudre ce problème.

Pour correctement séparer la vue du reste du code, toutes les interfaces ont été recréées en *FXML* à l'aide de *Scene Builder*.

Cette opération, coûteuse en temps, et considérée dans notre estimation, nous a permis de rendre le code plus clair. Nous avons ainsi créé un template (*GeneralView*) afin d'avoir un même menu sur toutes les pages internes de l'application.

Changement de base de données

Précédemment, la base de données utilisée était **SQlite**. Nous avons décidé de changer pour une base de données **H2**. Ce choix a été motivé par le fait que SQLite n'est pas officiellement supporté par **EclipseLink**, que nous avons utilisé pour sauvegarder nos objets. Nous avions notamment des erreurs avec les *ALTER TABLE* non supportées en SQLite et utilisée par EclipseLink pour ajouter les *FOREING KEYS*. Avec H2, tous ces problèmes ont été résolus.

Ajout des magasins sur la carte

L'ajout des magasins sur la carte depuis la base de données n'a pas posé beaucoup de problèmes. En effet, les magasins stockés dans la DB comportaient déjà les arguments latitude, longitude et horaire. Pour l'ajout par un utilisateur, nous avons rencontré un problème. En effet, nous n'avons aucun moyen de faire communiquer deux contrôleurs alors que le clic sur la carte (récupération des coordonnées) et la création du magasin se font dans deux contrôleurs différents. Nous avons donc utilisé une variable statique du contrôleur pour communiquer la position, ce qui pourrait poser problème si plusieurs fenêtres étaient créées en même temps.

Agrégation des markers et clic sur marker

Cette partie avait posé problème lors de l'itération précédente. Pour résoudre ce problème, nous avons changé les binômes assignés sur ce point. Ce nouveau binôme a trouvé une nouvelle source pour mieux comprendre la librairie **GMapFx**. La solution étant qu'il existe une classe du vue sur la carte qui implémente cela directement. Pour ce qui est du clic sur un *marker* (pour afficher de l'information), nous affichons une *InfoWindow* montrant le nom ainsi que l'horaire du magasin sélectionné.

Affichage du prix des produits de la liste de course

L'interface de la liste de course a été modifiée afin d'ajouter un menu déroulant (*combobox*), permettant de sélectionner un magasin précis de la DB. Une fois un magasin sélectionné, son stock est questionné afin de connaître la disponibilité du produit ainsi que son prix. Le prix adapté à la quantité souhaitée est affiché. Une autre possibilité pour l'utilisateur est de presser sur le bouton "*research shops*". Ce bouton déclenche l'ouverture d'un dialogue bloquant affichant l'ensemble des magasins contenant l'ensemble de la liste de course. Le prix de toute la liste de course est affiché à côté des magasins.

Gestion des recettes

Pour cette itération, il nous était demandé de réaliser (en partie) l'histoire numéro 3. L'utilisateur peut maintenant créer une recette. Nous avons choisi de sauvegarder les étapes de la recette comme un tableau de *String*. De cette manière, il est possible de ne modifier qu'une des étapes de la recette. L'interface de recette permet à l'utilisateur d'éditer des recettes déjà existantes, mais aussi de créer de nouvelles recettes "vierges" qu'il pourra modeler à sa guise. Celle-ci ce sauvegarde dynamiquement dans la base de données et au fur et à mesure de l'édition par l'utilisateur. Ainsi, pas de risque d'oublier de sauvegarder ou de voir une erreur interrompre le nouveau chef cuistot à l'oeuvre.