GL MIAGE - PROJET *2021*

Le but de ce projet est développer un application qui va permettre de faciliter la gestion d'un restaurant. Les objectifs du projet concernent la mise en pratique de ce que nous avons vu en cours:

- *travailler à plusieurs (groupe de 3) sur le développement d'une application,*
- *réussir à livrer un produit dans les temps,*
- *avoir le code le plus maintenable possible.*

Spécifications

On souhaite développer un application qui va permettre de faciliter la gestion d'un restaurant (service, cuisine, stocks, finances). Ainsi, les commandes des clients, le suivi des tables libres, la gestion en cuisine, la gestion des factures, la gestion des stocks seront entre autres facilitées.

Dans le détail, on souhaite que le personnel de service puisse saisir les commandes des clients via un terminal numérique, puis transmettre ces dernières à la cuisine. Le personnel de cuisine pourra visualiser les commandes entrantes et avertir le personnel de service concerné lorsque les plats sont prêts. Les serveurs pourront savoir en temps réel l'état d'avancement du repas de chaque table, et voir en temps réel l'état de l'occupation des tables du restaurant (par exemple à l'étage lorsqu'il y en a un), ce qui pourra leur permettre de savoir quelles tables sont propres, sales (à dresser pour un deuxième service) ou occupées. Les factures pourront être éditées rapidement, et les stocks de matières premières seront en permanence mis à jour en fonction des plats commandés et confectionnés par la cuisine. Les informations sont en permanence sauvegardées dans une base de données de façon à ce que la consultation soit possible. On pourra également avoir accès par exemple:

- à la popularité des plats,
- au temps de rotation moyen (le temps que les clients passent dans le restaurant),
- au temps de préparation moyen (temps écoulé entre le moment où la commande est passée et celui où elle est prête),
- au profit du déjeuner ou du dîner.

Pour ce faire, grâce à un terminal, le personnel du restaurant peut se connecter rapidement (chaque personne possède un identifiant) et effectuer la tâche souhaitée. Lorsqu'un serveur se connecte, il est accueilli par un écran d'état de l'étage dans lequel les tables qui lui sont attribuées sont colorées. Leurs tables sont colorées en fonction de leur statut: vert si la table est libre, jaune si elle est occupée, rouge si elle doit être débarrassée, orange si elle est réservée. À ce stade, un serveur peut sélectionner une table pour accéder aux informations de la table (nombre de couverts,...). Une fois qu'une table est sélectionnée, le personnel peut choisir parmi un certain nombre d'options. S'il choisit d'ajouter un article à la table, il se voit présenter les différentes catégories de plats proposés. Il peut alors sélectionner la catégorie appropriée, puis trouver l'article souhaité. Par exemple, si un client commande une sole meunière, le serveur doit se connecter, sélectionner la table et choisir "Ajouter un plat". Il sélectionnera ensuite "Poissons" dans la liste des catégories, puis le plat souhaité parmi ceux présentés. Il est ensuite renvoyé à l'écran de la table, où il peut choisir d'effectuer une autre tâche ou de se déconnecter. Pour chaque plat, si le plat n'est plus disponible, il n'apparaîtra pas dans les possibilités liées à la catégorie de plats. Cela évite au serveur d'avoir à vérifier avec la cuisine les commandes. Les commandes passées par le personnel de service sont, une fois validées, immédiatement affichées au personnel de cuisine par le biais d'une file d'attente, c'est-à-dire sur la base du premier entré, premier sorti. La seule exception à cette règle est pour les menus enfants, qui sont placés en tête de liste pour éviter les attentes trop longues.

Les rôles pris en charge par l'application sont les suivants : maître d'hôtel, serveur, assistant de service, cuisinier et directeur. Chaque compte utilisateur du système doit avoir ses propres privilèges. Tous les

écrans d'accueil personnalisés en fonction des rôles de chaque employé seront rafraîchis automatiquement en cas de besoin (lorsqu'une table est marquée prête; la commande d'une table est prête; le maître d'hôtel affecte un serveur à une table; ...). Le directeur peut gérer les employés (création, modification, suivi,...). Le cuisinier définit les plats à partir des matières premières. Si l'employé est un serveur, son profil contient également des informations sur les tables dont il est responsable. C'est le maître d'hôtel qui affecte les tables aux serveurs. Le directeur doit être en mesure de gérer d'autres aspects, comme la carte du jour, le suivi des stocks des matières premières et l'analyse des ventes. Les assistants de service récupèrent l'information leur permettant de savoir lorsqu'ils doivent desservir une table ou en dresser une nouvelle.

En plus de la coordination du personnel de service et de cuisine, l'application permet au directeur de connaître la recette quotidienne, hebdomadaire ou mensuelle. Des statistiques générées automatiquement permettent à la direction de voir quelle part des recettes provient de quel plat, c'est-à-dire quels sont les plats les plus populaires. Tout cela se fait automatiquement et reste à jour avec l'occupation du restaurant.

Objectifs et évaluation

Les objectifs principaux du projet sont les suivants:

- travailler à plusieurs (groupes de 3) sur le développement de l'application,
- réussir à livrer un produit fonctionnel dans les temps,
- avoir le code le plus maintenable possible (couplage/cohésion, principes SOLID, patterns,…),
- garantir le code produit grâce aux tests.

Voici les différents points qui serviront à l'évaluation de votre travail:

1/ réponse aux besoins exprimés

- *user stories*

2/ conception

- cohésion des composants, couplage des composants,
- architecture de l'application.

3/ réalisation

- tests unitaires, résultats des tests,
- cohésion des composants, couplage des composants,
- lisibilité/simplicité du code,
- structuration du projet.

4/ utilisation optimisée d'un SCM (Git) et de la méthode Scrum.

Travail à rendre

On vous demande de rendre votre projet en incluant les composants suivants:

- les sources, avec un lien vers votre dépôt (Github, Bitbucket, GitLab ou autre),
- le *backlog produit* et les *backlogs* précis de chaque sprint,
- les tâches, l'estimation, et leurs développeurs,
- les *burndown charts* à l'issue de chaque sprint,
- les tests unitaires,
- la version fonctionnelle obtenue à l'issue de chaque sprint.

Vous pouvez me contacter pour toute question.