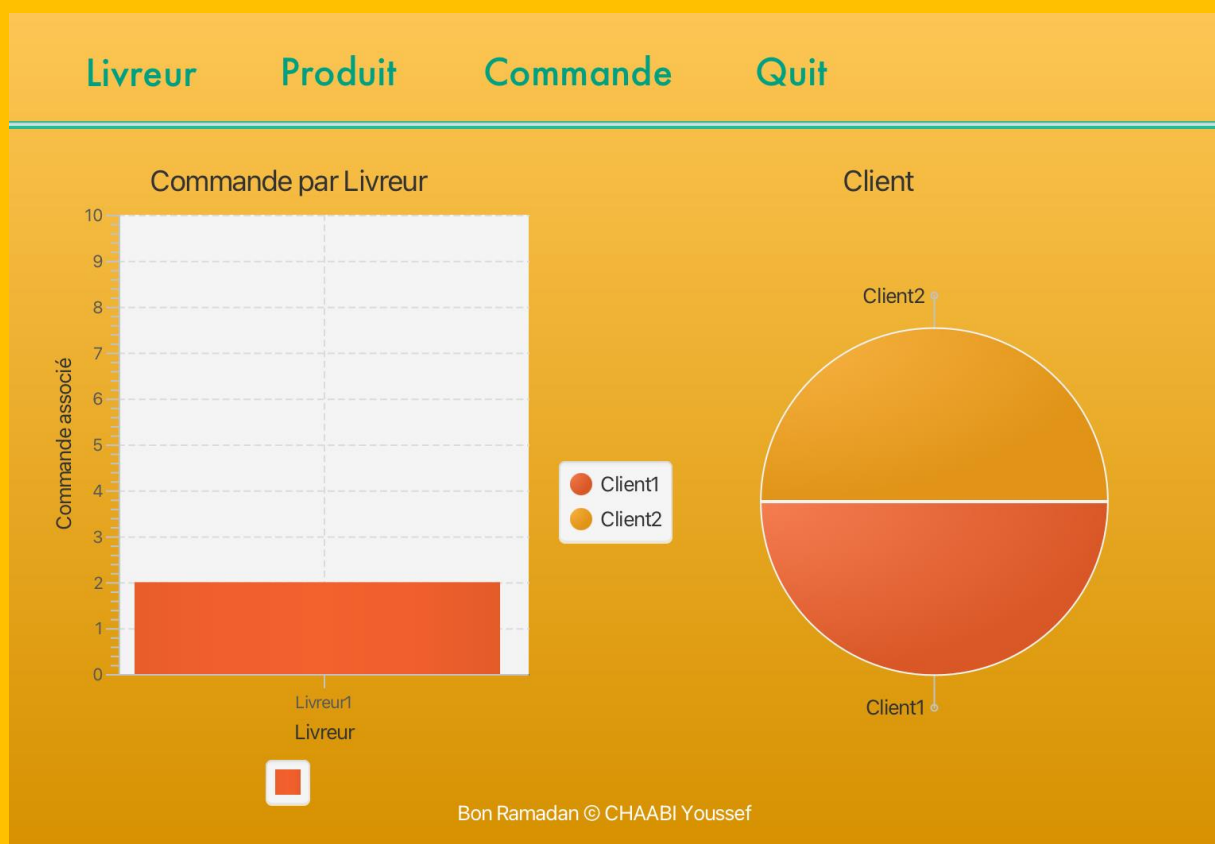


Compte rendu : Application desktop (Tracking des Livreurs ')

Encadrer par :
Prof. Lotfi EL AACHAK



Nom Complet : CHAABI Youssef

Introduction :

La programmation orientée objet (POO) est un paradigme de programmation qui permet de concevoir des programmes en se basant sur des objets, qui sont des entités qui représentent des choses ou des concepts du monde réel.

La POO se caractérise par les notions d'encapsulation, d'héritage et de polymorphisme. L'encapsulation permet de regrouper les données

et les méthodes qui agissent sur ces données dans une même entité, l'objet, pour en contrôler l'accès et garantir l'intégrité des données. L'héritage permet de créer des classes dérivées qui héritent des propriétés et des comportements d'une classe parente, facilitant ainsi la réutilisation et la modularité du code. Le polymorphisme permet de manipuler des objets de différentes classes de manière uniforme, en utilisant des méthodes ayant le même nom mais des comportements différents selon la classe.

Java est un langage de programmation orienté objet populaire qui met en œuvre ces concepts POO de manière stricte. En Java, toutes les entités sont des objets et les interactions entre ces objets sont basées sur les appels de méthodes. Java offre également un système de gestion automatique de la mémoire, grâce à la collecte des déchets, qui permet de simplifier la gestion de la mémoire et de prévenir les fuites de mémoire. Enfin, Java est également connu pour sa portabilité, qui permet d'exécuter du code Java sur différents systèmes d'exploitation et plates-formes matérielles sans nécessiter de modifications du code source.

Dans ce projet, notre objectif est d'élaborer une application de bureau en utilisant le concept de la programmation orientée objet et le langage Java, ainsi que d'autres outils. Cette application devra gérer les livreurs, les commandes et les produits, tout en offrant un tableau de bord générique.

Outils :

JDBC :

Le JDBC (Java Database Connectivity) est une API (Application Programming Interface) de Java qui fournit une interface commune pour les applications Java afin d'interagir avec les bases de données relationnelles. Le JDBC permet aux développeurs Java de se connecter à une base de données, d'envoyer des requêtes SQL et de récupérer les résultats de ces requêtes.

En Java, le JDBC est une bibliothèque standard incluse dans le kit de développement Java (JDK). Il fournit des classes et des interfaces pour établir une connexion à une base de données, créer des objets de requête, exécuter des requêtes SQL,

traiter les résultats et gérer les transactions. Les développeurs Java peuvent utiliser le JDBC pour accéder à une grande variété de bases de données, telles que MySQL, Oracle, Microsoft SQL Server, PostgreSQL, etc.

MySQL Server:

MySQL est un système de gestion de bases de données relationnelles open source, développé par Oracle Corporation. Il est l'un des systèmes de gestion de bases de données les plus populaires et largement utilisés dans le monde. Voici quelques avantages de MySQL par rapport à MariaDB :

La compatibilité avec un grand nombre d'applications : MySQL est un système de gestion de bases de données très mature, et il est utilisé par de nombreuses applications et frameworks.

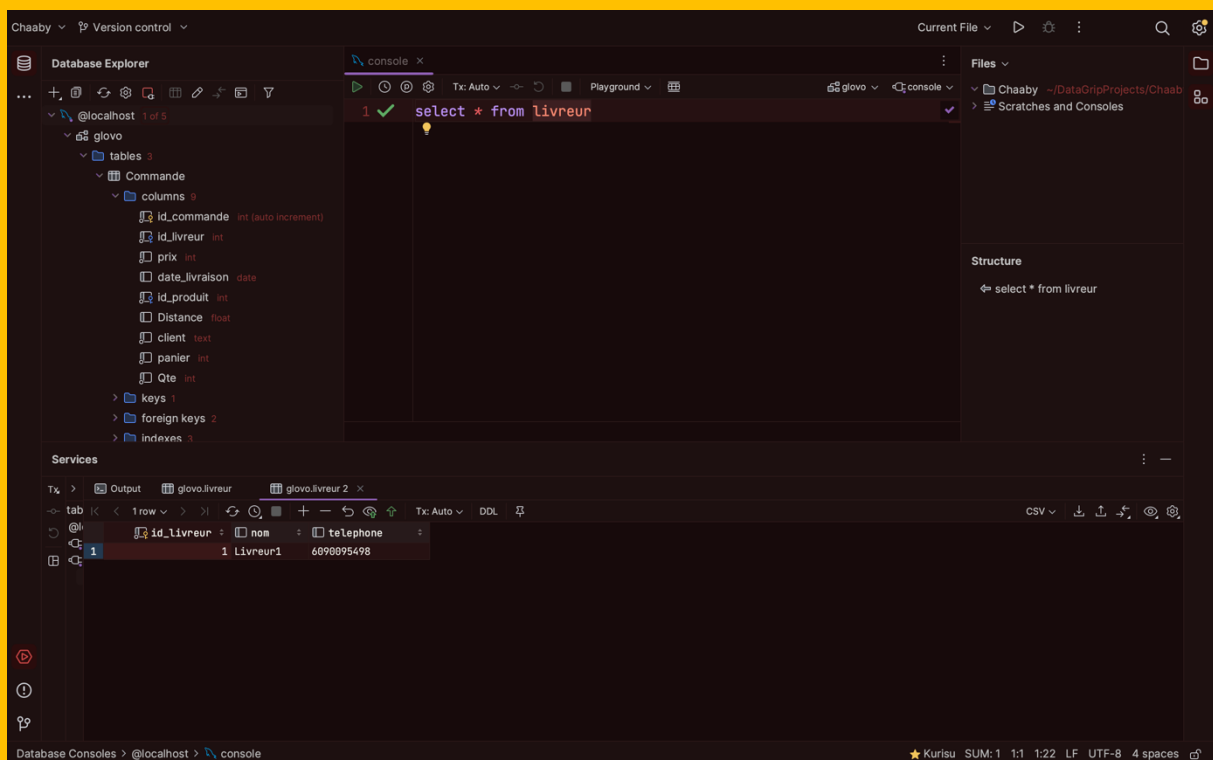
La prise en charge des clusters de haute disponibilité.

Une plus grande variété de moteurs de stockage : MySQL prend en charge un grand nombre de moteurs de stockage, notamment InnoDB,

MylSAM, Memory, CSV, Blackhole, Federated, et bien d'autres encore.

DataGrip :

JetBrains DataGrip est un outil puissant pour travailler avec les bases de données. Il prend en charge PostgreSQL, MySQL, SQL Server, Oracle, MongoDB, et beaucoup d'autres bases de données et sources de données DDL, et permet de voir la structure de vos bases de données et d'effectuer diverses tâches dans l'explorateur de bases de données.



IntelliJ :

IntelliJ IDEA est un environnement de développement intégré (IDE) pour la programmation en Java, Kotlin et d'autres langages de programmation. Il est développé par la société JetBrains et est disponible en version gratuite (Community Edition) et en version payante (Ultimate Edition) avec des fonctionnalités plus avancées.

Les avantages d'utiliser IntelliJ IDEA pour le développement en Java et JavaFX sont nombreux. En voici quelques-uns :

Productivité accrue : IntelliJ IDEA propose de nombreuses fonctionnalités qui permettent aux développeurs d'écrire du code plus rapidement et efficacement. Par exemple, il offre une complétion de code intelligente, une analyse de code en temps réel, des raccourcis clavier personnalisables et bien plus encore.

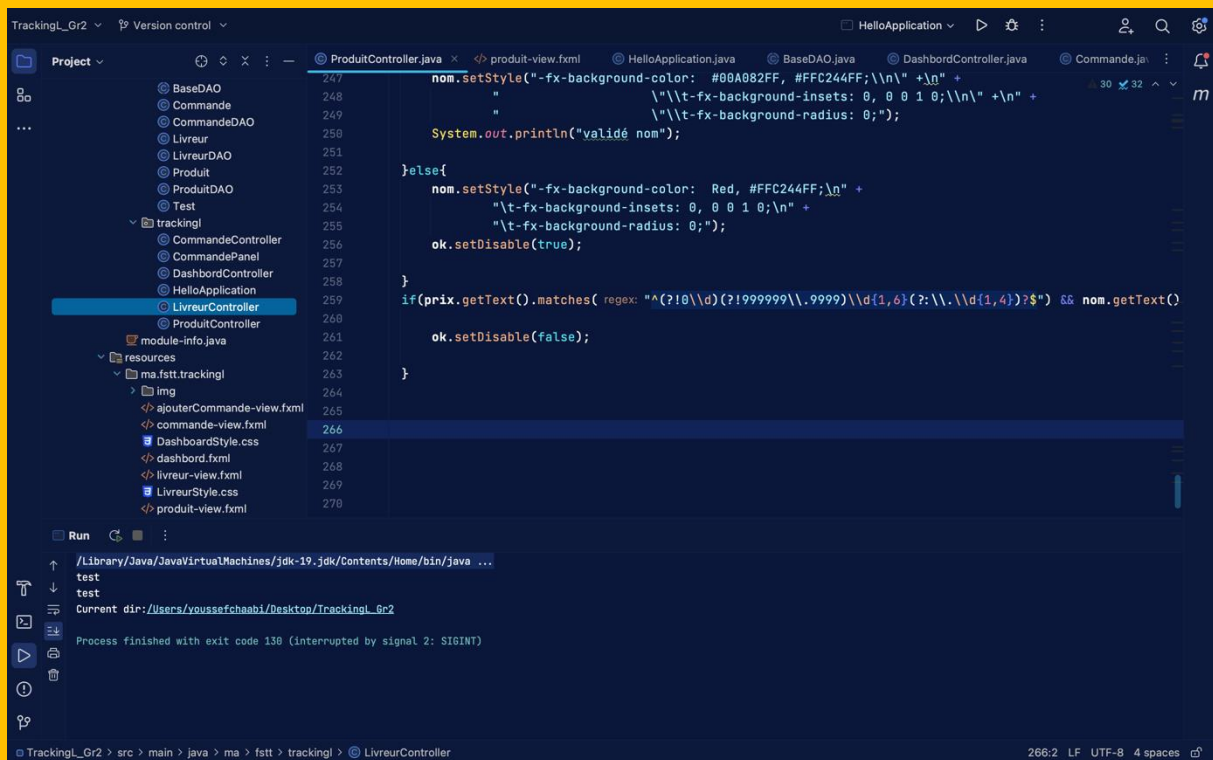
Débogage facile : L'IDE offre un débogueur puissant qui permet de déboguer rapidement et efficacement les erreurs dans le code. Il permet également de suivre les variables en temps réel

pour aider à comprendre les problèmes plus facilement.

Intégration de build : IntelliJ IDEA offre une intégration transparente avec les outils de build populaires tels que Maven et Gradle. Cela facilite la création, la compilation et l'exécution de projets Java et JavaFX.

Refactoring simplifié : L'IDE propose des fonctionnalités de refactoring puissantes qui permettent de renommer rapidement et facilement des classes, des méthodes et des variables, ainsi que d'autres opérations de réorganisation du code.

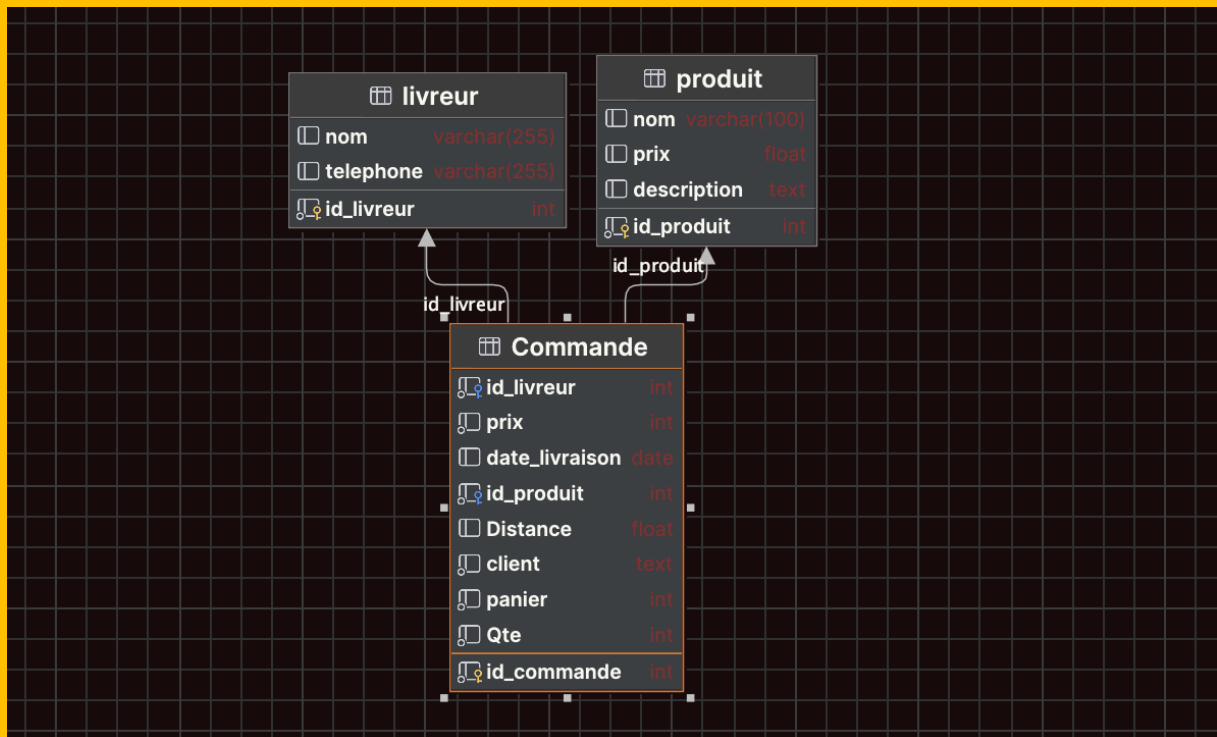
Support pour JavaFX : IntelliJ IDEA offre une prise en charge complète pour JavaFX, y compris la conception visuelle de l'interface utilisateur avec Scene Builder, la complétion de code pour les classes JavaFX, et bien plus encore.



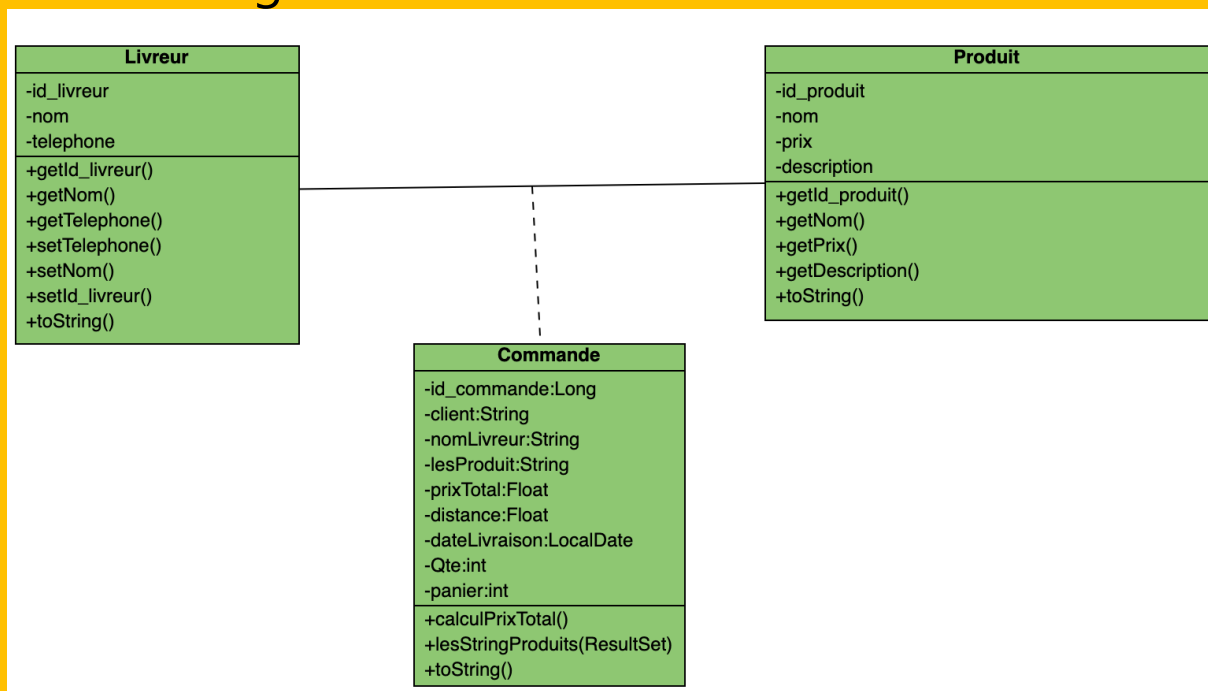
JavaFX :

JavaFX est une bibliothèque graphique intégrée dans le JRE et le JDK de Java. Oracle la décrit comme « The Rich Client Platform », c'est-à-dire qu'elle permet de réaliser des interfaces graphiques évoluées et modernes grâce à de nombreuses fonctionnalités, telles que les animations, les effets, la 3D, l'audio, la vidéo, etc. Elle a de plus l'avantage d'être dans le langage Java, qui permet de réaliser des architectures avec des paradigmes objet, et aussi de pouvoir utiliser le typage statique.

Conception :



et via les jointeurs, une correspondance est faite avec le diagramme de classe :

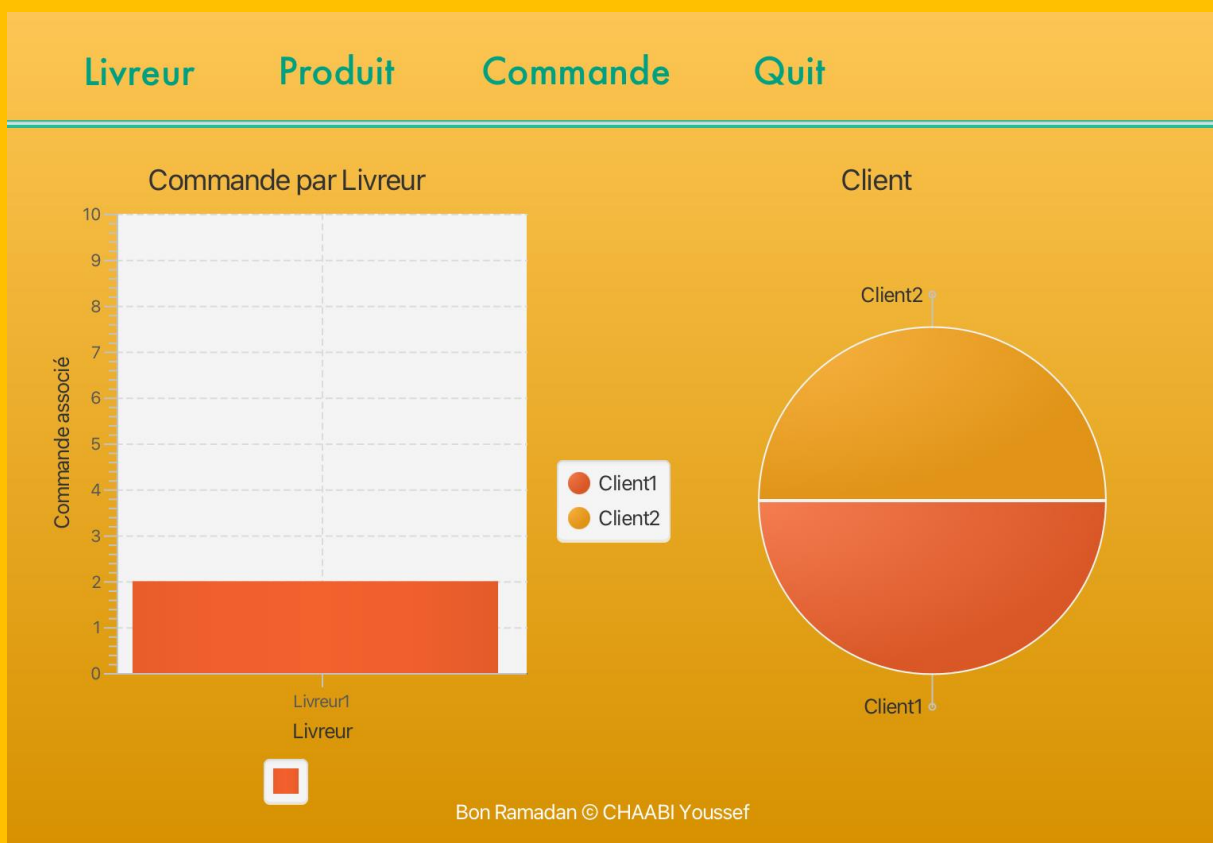


Cette correspondance m'a permis de manipuler des commande avec *plusieurs produits*.

Développement :

L'application contienne 5-6 différente scène dans un seul 'Stage' :

- Dashboard :



Cette scène contient un menu des scène CRUD « Livreur, Produit, Commande » et le bouton « Quit » ; en bas il y'a deux diagrammes de différentes type «PieChart» et «XYChart» qui représentent les pourcentages des clients fidèle et les commandes associé à chaque livreur.

- Livreur :

Back

Nom :

Telephone :

Save

ID	Nom Complet	Telephone	Modifier	Supprimer
94	Livreur 1	6090095498	Modifier	Supprimer

Validation (dans le cas d'un numéro de téléphone qui ne respecte pas un regex) :
Le bouton Save sera désactivé

Back

Nom :

TEST

Telephone :

08080808

Save

ID	Nom Complet	Telephone	Modifier	Supprimer
94	Livreur1	6090095498	Modifier	Supprimer

Ceci est appliqué dans tous les formulaires

- Produit :

Back

Nom Produit :

Prix :

Description :

Save

Nom Produit	Prix	Description	Modifier	Supprimer
Produit1	200.0	rien	Modifier	Supprimer

Back

Nom Produit :

Prix :

Description :

Save

Nom Produit	Prix	Description	Action
Produit 1	200.0	rien	<div>ModifierSupprimer</div>

Confirmation?

Are you sure you wanna delete this ?

AnnulerOK

[illegible]

Back

Autre Produit

Régénérer

Client :

TEST

Date de Livraison :

22/04/2023

Nom Produit :

Produit1

Quantite :

2

Nom Produit :

Produit2

Quantite :

3

Nom Livreur :

Livreur1

Distance :

3.2

Save

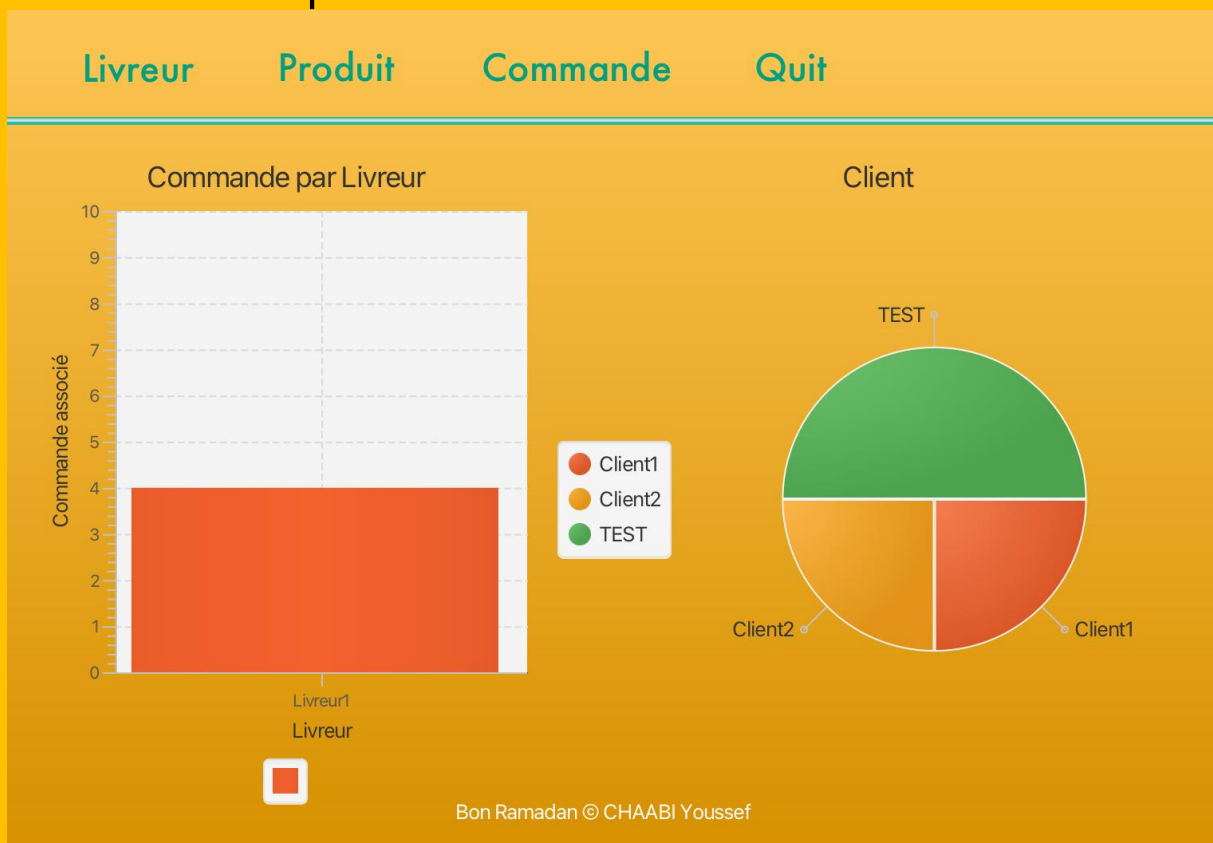
Avril

2023

Lun.	Mar.	Mer.	Jeu.	Ven.	Sam.	Dim.
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Possibilité d'ajouter plusieurs produits

Dashboard après la commande :



Difficultés rencontrer :

En réalisant ce projet on a passé énormément de temps à apprendre le JavaFX. Durant la création de cette application on a rencontré beaucoup de difficultés, voici quelques-unes des difficultés qu'on a rencontré en développant ce jeux vidéo :

- l'intégration des boutons dans les tableaux.
- la connexion à la Base des Données et la configuration du JDBC
- j'avais l'idées au début d'ajouter l'authentification avec la reconnaissance facial via la bibliothèque 'OpenCV', j'ai pas pu là configurer .
- Associer plusieurs produit à une seul commande

Conclusion :

Tout au long de la préparation de notre projet, nous avons essayé de pratiquer les connaissances requises durant notre cours du programmation orienté objet en Java, et aussi notre connaissance sur ce qu'on a trouvé concernant JavaFX, ce projet nous a permis d'entrer dans le monde de

développement d'application et acquérir une expérience concernant ce monde.