

REPUBLIQUE DU SENEGAL



Un Peuple – Un But – Une Foi

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE ET DE L'INNOVATION

UNIVERSITE GASTON BERGER DE SAINT-LOUIS



U.F.R DE SCIENCES APPLIQUEES ET DE TECHNOLOGIE

SECTION D'INFORMATIQUE

MASTER SCIENCES ET TECHNOLOGIE, MENTION INFORMATIQUE

Rapport de Projet Génie Logiciel

SUJET : Conception et réalisation d'une application pour la gestion automatique des files d'attente à guichet multiple.

Présenté par :

Mouhamed Khoutbou Thiam P26 98

Encadreur :

Dr Awa Diatara

Année Universitaire 2020-2021

Table des Matières

Introduction

Chapitre 1 : Analyse des besoins et Conceptions

1.1 Analyse des besoins

1. Analyse fonctionnelle

1.2 Modélisation UML

1.2.1 Diagramme des cas d'utilisations

1.2.2 Diagramme des classes

1.2.3 Diagramme des séquences

Chapitre 2 : Réalisation du projet

1. Environnement de développement

2. Réalisation et Test

Conclusion

Introduction

Les files d'attente et les files d'attente excessives sont parmi les principales causes de mécontentement dans les banques, les magasins de détail, les hôpitaux et les institutions gouvernementales.

C'est dans ce sens, qu'on propose une application pour automatiser la gestion des files d'attente à multiple guichets.

En se basant sur les concepts de génie logiciel, de système d'informations et de développement d'applications et grâce à des informations collectées, nous avons élaboré notre vision de l'application, son architecture ainsi que les outils et technologies utiles.

Dans ce contexte, le travail réalisé s'étend sur deux chapitres. Nous allons d'abord faire l'analyse des besoins dans lequel nous identifions les acteurs et leur cas d'utilisation ainsi que la conception technique. Nous terminerons par présenter nos choix technologiques (environnement de développement et de Test) pour réaliser l'application.

Chapitre 1 : Analyse des besoins et conception

1.1. Analyse des besoins

Elle vise à comprendre les problèmes métiers posés. Elle détermine également les éléments intervenant dans le système, leur structure et leurs relations.

1. Analyse Fonctionnelle

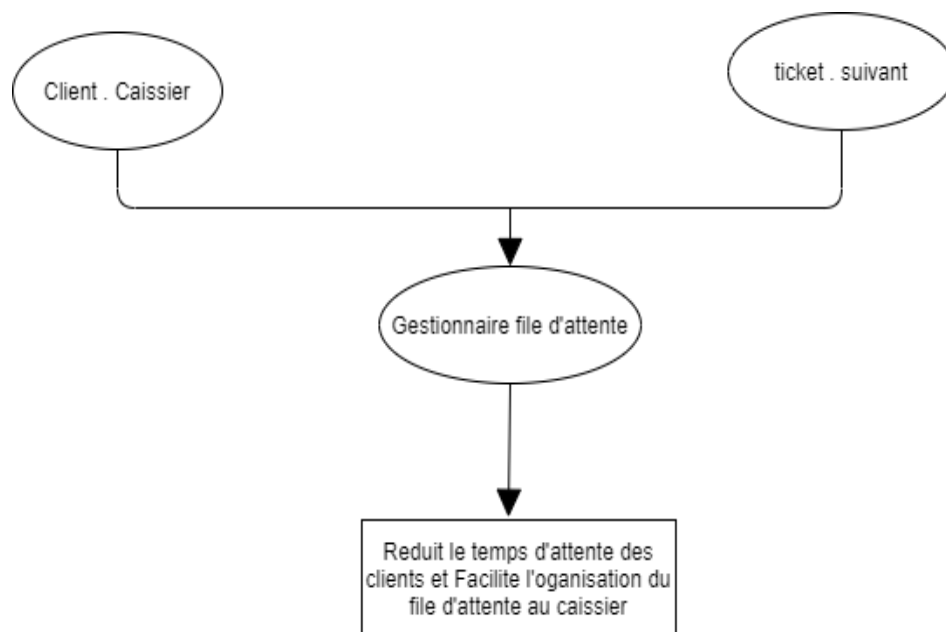


Figure 1 : Diagramme de bête à cornes

La bête à cornes permet d'exprimer la recherche du besoin.

Elle permet de répondre aux questions suivantes :

- A qui le système rend-il service ?
Dans notre cas Caissier et Clients.
- Sur quoi le système agit-il ?
Dans notre cas les Tickets et le Moniteur.
- Dans quel but ?
Dans notre cas génère des tickets et affiche le client appelé sur le moniteur.

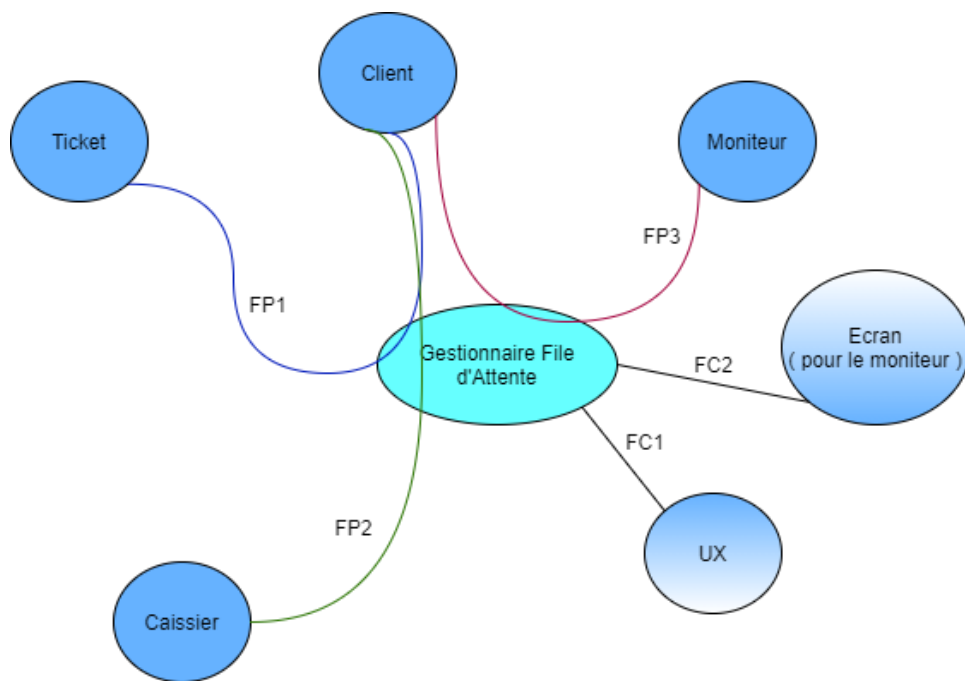


Figure 1 : Diagramme de Pieuvre

- Fonctions Principales :

FP1 : Générer un Ticket

FP2 : Appeler le client suivant dans la queue

FP3 : Afficher le client appelé

- Fonctions de Contraintes

FC1 : Une bonne User eXperience

FC2 : Un écran pour le moniteur

1.2. Modélisation UML

1.2.1. Diagramme des cas d'utilisations :

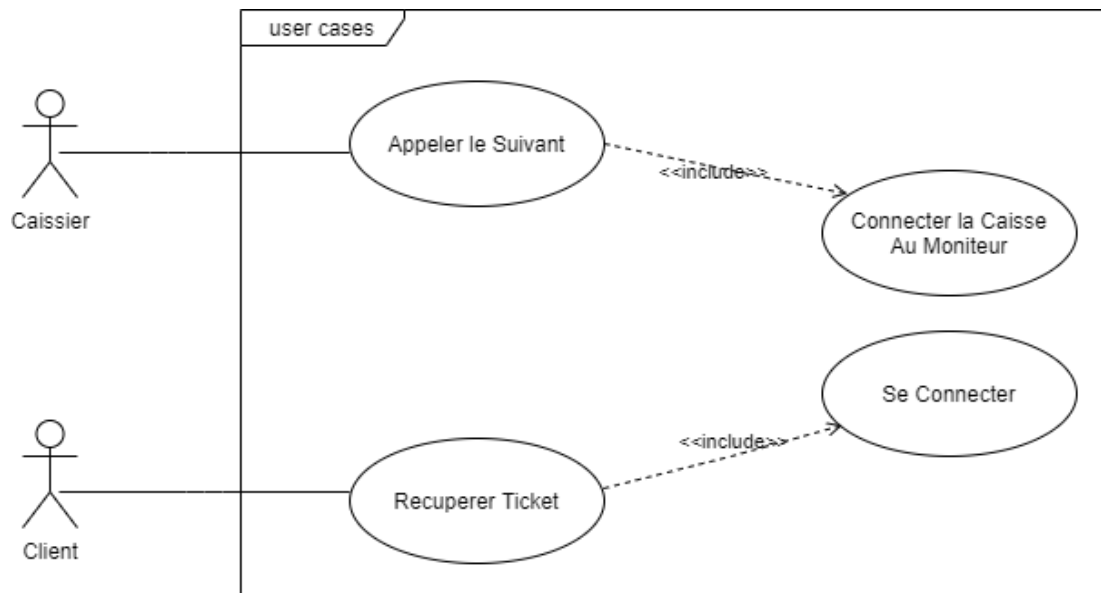


Figure : 3 Diagramme des cas d'utilisations

1.2.2. Diagramme des classes

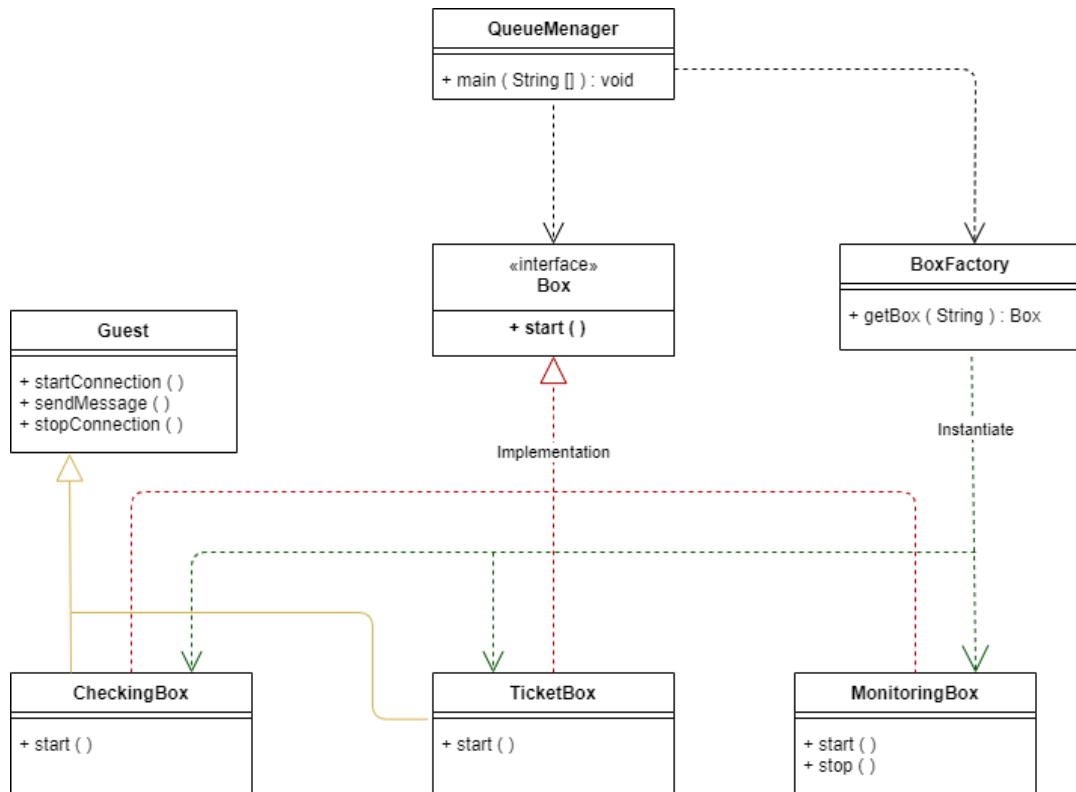


Figure 4 : Diagramme des classes

En regardant ce diagramme (Figure 4) on peut voir nettement qu'on a utilisé "**Factory**" comme design pattern pour la construction des Box.

1.2.3. Diagramme des séquences

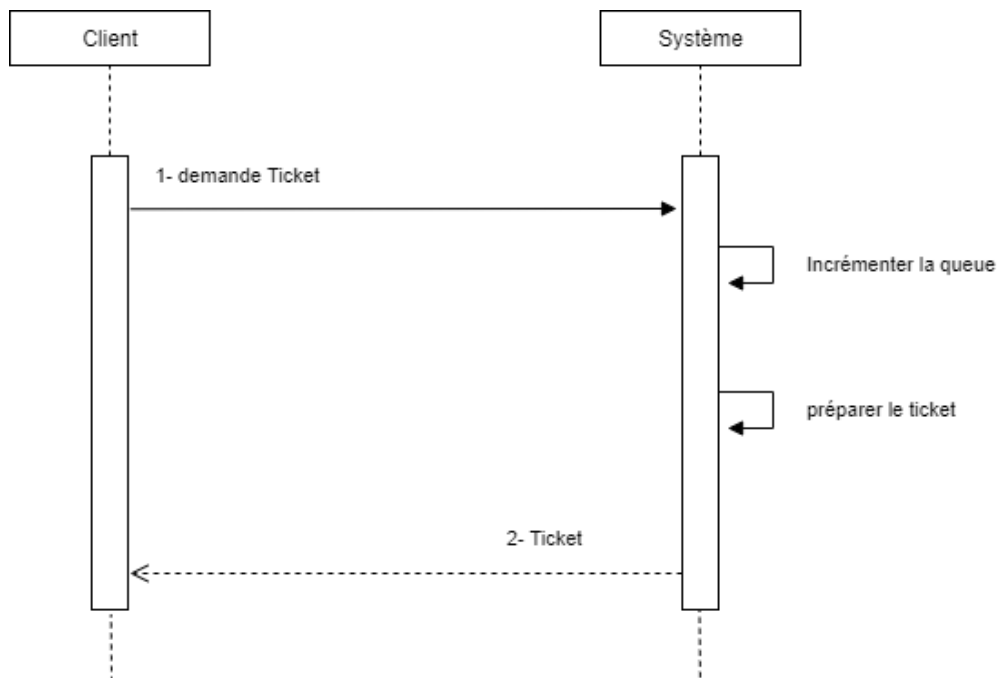


Figure 5 : Diagramme de Séquence de récupération de Ticket

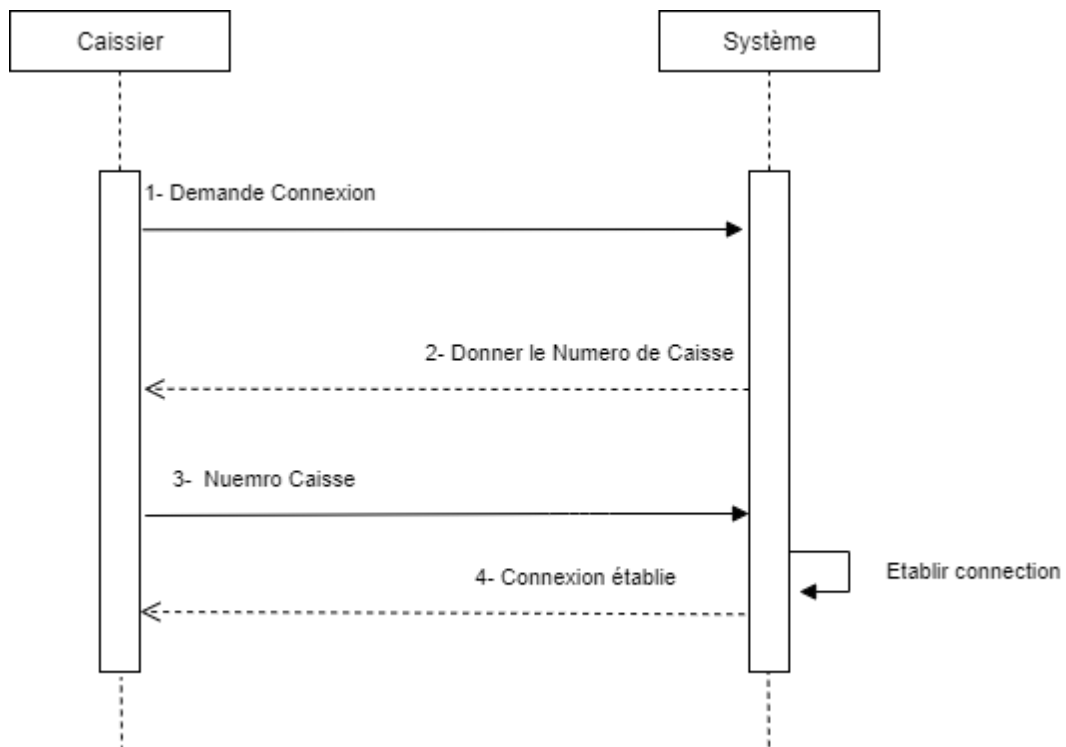


Figure 6 : Diagramme de Séquence de demande de connexion de la Caisse

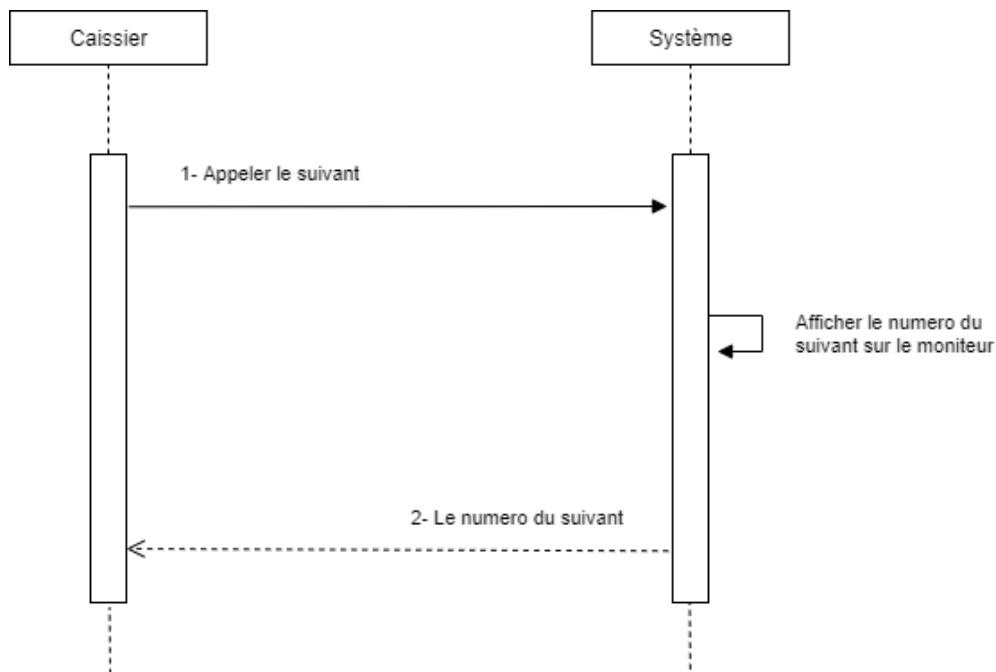


Figure 7 : Diagramme de Séquence d'Appel du suivant dans la queue

Chapitre 2 : Réalisation du projet

1. Environnement de développement

Pour réaliser l'application, nous avons eu recours aux logiciels suivants :

- **DrawIO** pour modéliser les diagrammes
- **Visual Studio Code** comme IDE
- **JUnit** pour les tests unitaires et tests d'intégration
- **Java** comme langage de programmation
- **Apache Maven** : Maven est un outil de gestion de projet logiciel. Basé sur le concept d'un modèle objet de projet (POM), Maven peut gérer la construction, le reporting et la documentation d'un projet.
- **Lanch4j** comme plugin Maven pour la génération automatique du fichier ".exe".

1. Réalisation et Tests

On a utilisé “**Factory**” comme design pattern pour la construction des Box.

Le code est bien commenté et structuré parce qu’on a pensé à son évolutivité et l’élargissement de l’équipe.

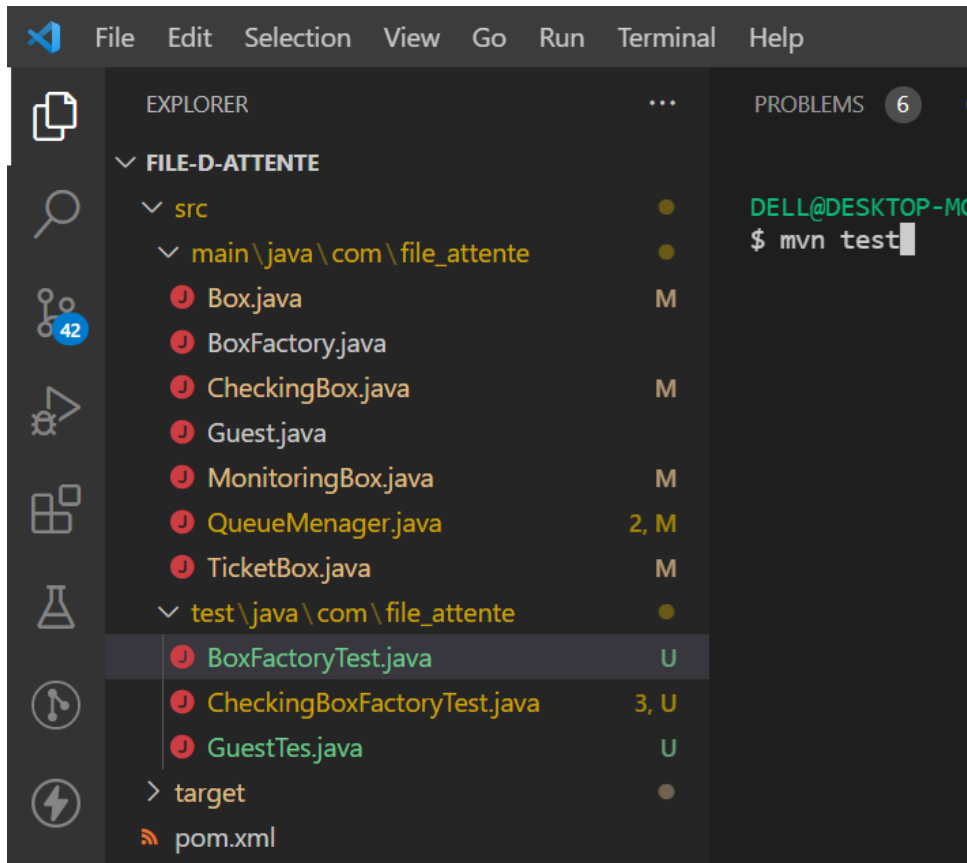


Figure 8 : Capture de structuration du code source

On a utilisé les **Sockets** pour assurer la partie réseau et la communication des modules. Pour s’assurer également que les modules peuvent être lancées dans des machines différentes et continuent de se communiquer.

Et on a choisi la “console” comme Interface Utilisateur pour diminuer le temps de réalisation.

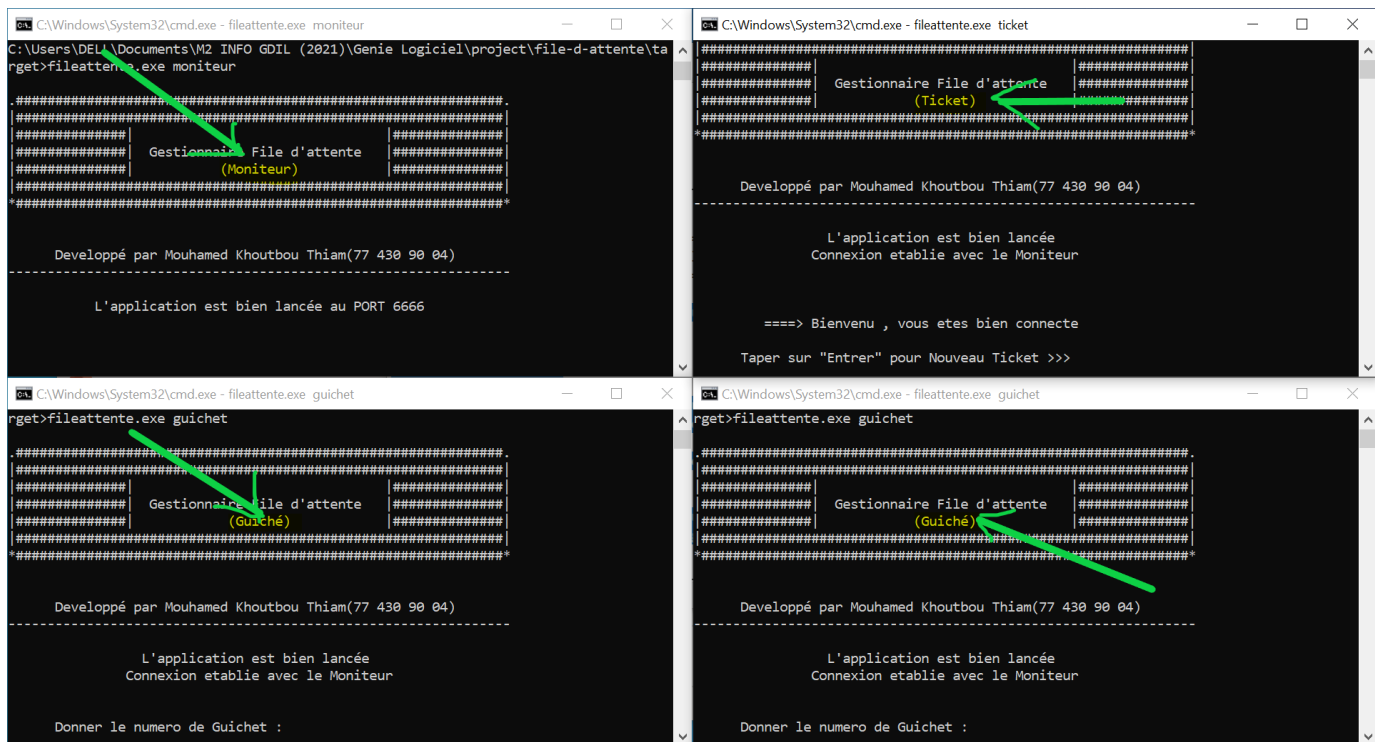


Figure 9 : Capture de Lancement des différents modules

Ici (Figure 9) on a lancé les 3 modules de l’application, le Box moniteur (en haut à gauche), le Box de ticket (en haut à droite) et 2 Box Guichet (en bas gauche et droite).

Et en fin on a écrit des classes de test pour les tests unitaires.

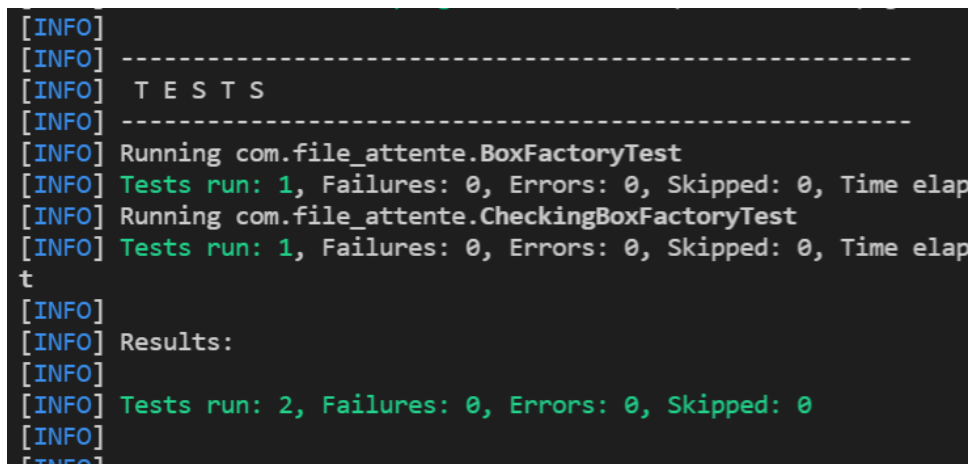


Figure 10 : Capture du résultat des tests unitaires

Conclusion :

L'objectif de notre projet de génie logiciel est de concevoir et d'implémenter une plateforme de gestion des files d'attente à multiple guichets dans les banques, les magasins de détail, les hôpitaux et les institutions gouvernementales.

Cette plateforme permettra entre autres aux clients de pouvoir récupérer un ticket sans aucune assistance. Elle permettra à ces derniers aussi de voir l'état d'avancement de la queue. Pour le cas du caissier, la plateforme leur offrira la possibilité d'appeler le client suivant sur la queue vers sa caisse.

La réalisation de ce projet nous a été d'une importance considérable, car elle nous a permis de suivre une méthodologie de travail Scrum. Elle nous a permis d'approfondir nos connaissances en informatique plus précisément sur la totalité des concepts du génie logiciel vus en classe