# Projet IAP

Gérer une entreprise



Ahmad AL-HARIRI --- BETTAYEB Yacine

Groupe 105



# Table des matières

Présentation du projet	3
Organisation des tests de l'application	5
Bilan de validation	.6
Bilan de projet	7
Preuve de validation	8
Annexes	10
Trace d'exécution du sprint 6R	10



#### Présentation du projet

L'objectif du projet est de programmer un interpréteur de commandes permettant d'ajouter des spécialités à l'entreprise avec leur coût horaire, d'ajouter des travailleurs pour chaque spécialité déclarer, d'ajouter des clients et des commandes, des taches, de distribuer les taches automatiquement aux travailleurs ayant le moins d'heure de travail et de donner une facture pour chaque commande.

Cette application doit interpréter 14 commandes. Celles-ci sont représenter par des chênes de caractère non accentuées de format spécifiés dans ce document. Le caractère séparateur entre a commandé et les différents champs d'information est le caractère 'espace'. La taille maximale des chaînes de caractère est de 35.

#### Les commandes sont les suivantes :

#### C1. Commander demande de l'aide

Elle sert à afficher toutes les commandes qu'on puisse faire dans l'interpréteur, la syntaxe est : « -help ».

#### C2. Commande de nettoyer l'affichage dans l'interpréteur

Elle sert à supprimer toute l'écriture (affichage) dans l'interpréteur, la syntaxe est « clear ».

### C3. Commande d'ajouter des spécialités

Elle sert à ajouter des spécialistes (compétence) à l'entreprises, la syntaxe est « developpe » suivie de nom de la spécialités « nom » suivie de prix horaire de la spécialité n « entier » inférieur ou égale à 10000.

#### C4. Commande d'ajouter des travailleurs

Elle sert à ajouter des travailleurs ayant des compétences dans l'entreprise, la syntaxe est « embauche » suivie de nom de travailleur « nom1 » suivie de nom de la spécialités « nom2 ».

### C5. Commande de définition des clients

Elle sert à définir des clients dans l'entreprise, la syntaxe est « démarche » suivie de nom de client « nom »

#### C6. Commande de définition d'une commande



Elle sert à insérer une commande pour un client choisi, la syntaxe est « commande » suivie de nom de la commande « nom1 » suivie de nom de client « nom 2 ».

#### C7. Commande de définition d'une tache

Elle sert à ajouter une tache pour une commande choisi à une spécialité choisie, la syntaxe est « tache » suivie de nom de la commande « nom1 » suivie de nom de la spécialité « nom2 » suivie de nombre d'heure de travaille « entier » inférieur à 10000.

#### C8. Commande de définition de progression d'une tache

Elle sert à ajouter le nombre d'heure de travail effectué (l'avancement de travailleur pour chaque tache), la syntaxe est « progression » suivie de nom de la commande « nom1 » suivie de nom de la spécialité « nom » suivie de nombre d'heure effectué « entier »

Si la commande est suivie de mot « passe » on le système réaffecte la tâche à un travailleur de même spécialité possédant le minimum heure de travail.

#### C9. Commande d'affichage des spécialités déjà déclarer

Elle sert à afficher la liste des spécialités, dans l'ordre de leur déclaration, ainsi que le coût associé, la syntaxe est « spécialités »

#### C10.Commande d'affichage des travailleurs

Elle sert à afficher la liste des travailleurs compétant pour la spécialité choisie, par ordre de déclaration des travailleurs, la syntaxe est « travailleurs » suivie de nom de la spécialité « nom ». Si la commande est suivie par « tous » à la place de nom de la spécialité, dans ce cas on affiche la liste de tous les travailleurs pour toutes les spécialistes. Les spécialités sont affichées dans l'ordre de leur déclaration ; pour une même spécialité, les travailleurs sont affichés par ordre de déclaration.

# C11. Commande d'affichage des commandes

Elle sert à afficher la liste des commandes effectuées par le client choisi, selon leur déclaration, la syntaxe est « client » suivie par le nom de client « mot » si la « client » est suivie par « tous » on affiche la liste de toute les commandes, par ordre de déclaration de client, et de commande.

## C12. Commande d'affichage d'état d'avancement



Elle sert à afficher la description de l'état d'avancement des commandes et des tâches associés, par ordre de commandes, et de spécialités, la syntaxe est « supervision ».

#### C13. Commande d'affichage des charges de travailler

Elle sert à afficher le nombre d'heure restantes pour l'accomplissement de chacune des tâches qui lui sont affectées, les tâches sont affichées par ordre de commande, et de spécialités, la syntaxe est « charge » suivie par le nom de travailleur « mot ».

#### C14. Commande de sortie du programmer

Elle sert à fermer la console, la syntaxe est : « interruption »

# Organisation des tests de l'application

Le développement de l'application s'est fait à l'aide d'une méthodologie de développement sûre : le développement agile. Nous procédions pas à pas, par Sprints, chacun représentant un incrément de fonctionnalité de l'application.

Nous avions 6 Sprints définis. Pour le développement de chacun d'entre eux, il y avait 4 étapes clefs :

- l'analyse fonctionnelle ;
- -la spécification;
- -le codage;
- -la validation.

Chaque Sprint#n (hormis le premier) est codé en s'appuyant sur le Sprint#n-1. Il reprend ses structures ainsi que les fonctions auxquelles il faisait appel tout en modifiant ce qui est nécessaire et en ajoutant de nouvelles commandes. Afin de tester la qualité de notre code, nous possédions deux fichiers textes pour chaque Sprint#n : un jeu de données test (inSpn.txt) ainsi que la sortie escomptée (outSpn.txt).

Si l'application fonctionnait comme elle le devait, en entrant le inSpn.txt dans la console, nous nous retrouvions avec une sortie correspondant à outSpn.txt.

Pour vérifier si la sortie et le outSpn sont les mêmes, il est possible de les lire soimême ou plutôt d'utiliser le site DiffChecker qui compare les différences entre deux



textes (au fur et à mesure que l'application se complexifie, il sera préférable d'opter pour cette solution).

Si aucune différence n'est observée entre la sortie obtenue (run) et le outSpn, alors le Sprint#n est dit 0-défaut et est donc validé. Il est alors possible de passer au suivant.

# Bilan de validation

#### Sprint0:

Le sprint 0 avait pour but de coder la fonction « main » et la commande « interruption » et le début de la fonction « traite developpe »

#### Sprint 1:

le sprint 1 avait pour but de coder plusieurs fonction « traite\_embauche » « traite\_client » « traite\_commande » …, de même structure que la fonction « traite\_developpe » de sprint0. Ce sprint ne gère aucune donnée, c'est un code d'affichage uniquement.

#### Sprint2:

Le sprint 2 avait pour but de mémoriser quelques données (les spécialités et le coût horaire, les travailleurs, les clients) dans leurs structures. Puis de modifié quelque affichage concernant les spécialités, les travailleurs, les clients. Nous avons travaillé sur la version **release** où plusieurs travailleurs peuvent avoir plusieurs compétences.

#### Sprint3:

Le sprint 3 avait pour but de mémoriser les commandes, les tache, la progression de chaque tache (le mot **passe** n'est pas implémenté). Puis de modifiés quelques affichages concernant les clients et l'état de tache pour chaque commande (**supervision**).

#### Sprint4:

Le sprint 4 avait pour but d'affecter les taches aux travailleurs automatiquement. Nous avons travaillé sur la version release où il faut affecter les taches au première travailleur ayant la même compétence et possédant le moins nombre d'heure à faire pour compléter toutes ses taches. Et dans le deuxième temps, il faut traiter l'affichage de mot clé « charge »

#### Sprint5:



Le sprint 5 avait pour but de crée une facture pour chaque commande et une facture générale lorsque toutes les taches se termine. Nous avons créé la fonction « traite\_facture » qui gère entièrement la facture et détecte la fin de tous les commandes.

#### Sprint6:

Le sprint6 avait pour but d'implémenter le mot clé « passe »

# Bilan de projet

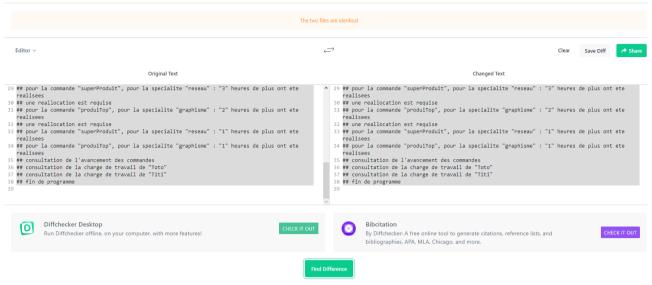
Ce premier projet de programmation nous permet d'approfondir nos connaissances et de nous entraîner à programmer. C'est l'entrée dans le domaine informatique. Le projet doit être réalisé en équipe de deux, ce qui nous a permis de nous entraider tout en travaillant en autonomie. Dans notre équipe, aucune de nous ne savait programmer en « C » donc le début fût compliqué. Nous avons réussi à valider le sprint 1, la partie la plus ardue du projet. Après plusieurs essais, nous avons enfin réussi à comprendre la mécanique de codage de l'application. Chaque sprint était compliqué pour nous, parfois nous ne voyions pas les erreurs dans notre codage et craignions que le programme ne fonctionne pas.

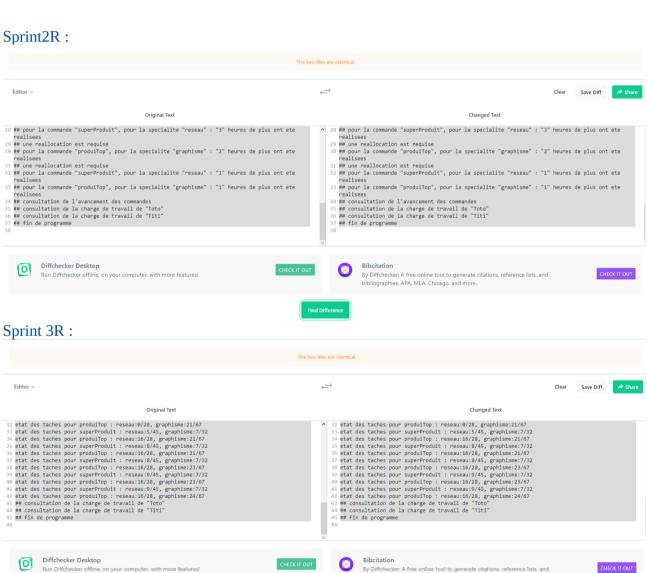
Malgré notre travaille intensive qui durait toute la journée et jusqu'au 2 heures du matin parfois. Nous avons réussi à valider le sprint 6 release en utilisant le jeu d'essai que le prof nous a donné. Le jour où l'enseignant teste le plus haut niveau atteint afin de le valider, nous pensions avoir validé parfaitement le sprint 6r même avec un essaie plus compliqué, mais on a trouvé quelques erreurs d'affichage et une autre erreur dans l'affectation des charges à chaque travailleur, nous avons bien réussi à corriger les erreurs d'affichage très vite, mais l'erreur d'affectation n'a pas été résolu vite le jour d'après « le Vendredi » on a trouvé la petite erreur dans le programme, c'était le nom d'une variable dans une boucle for, nous avons renvoyé notre code au professeur en lui indiquant la cause de l'erreur et on a validé le sprint6 release. C'est un projet qui, afin d'être réussi, demande beaucoup de concentration, de réflexion, de stratégie, de patience et de motivation.



#### Preuve de validation

#### Sprint 1R:

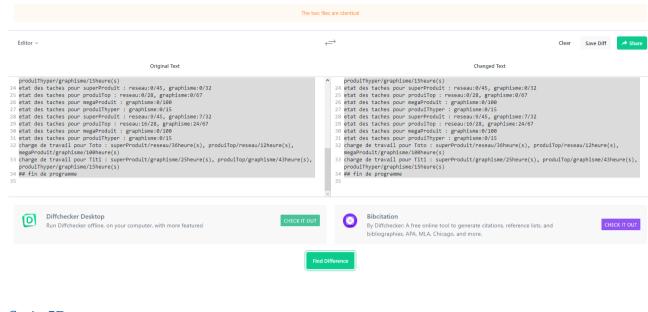




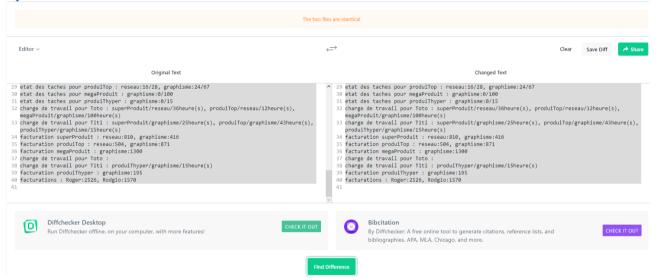
bibliographies. APA, MLA, Chicago, and more.



#### Sprint4 R:

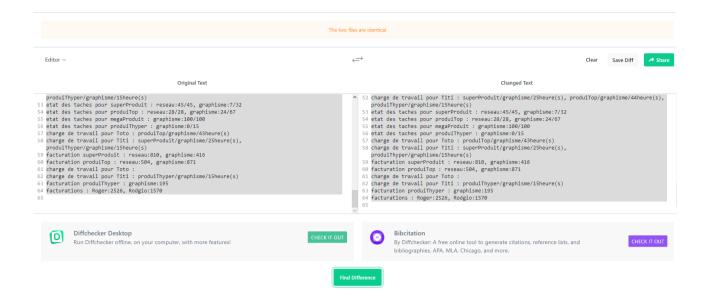


#### Sprint5R:



#### Sprint6R:





#### Annexes

# Trace d'exécution du sprint 6R

#### Entrée:

developpe reseau 18 developpe graphisme 13 specialites

embauche Toto reseau embauche Titi graphisme embauche Toto graphisme travailleurs tous

demarche Roger demarche Rodgio client tous

commande superProduit Roger commande produiTop Rodgio commande megaProduit Roger commande produiThyper Rodgio client tous supervision

tache superProduit reseau 45 charge Toto charge Titi tache superProduit graphisme 32 charge Toto P a g e - 10-|13



charge Titi
tache produiTop graphisme 67
charge Toto
charge Titi
tache megaProduit graphisme 100
charge Toto
charge Titi
tache produiThyper graphisme 15
charge Toto
charge Titi
tache produiTop reseau 28
charge Toto
charge Titi
supervision

progression superProduit reseau 5 progression superProduit graphisme 7 progression produiTop graphisme 21 progression produiTop reseau 16 supervision charge Toto charge Titi progression superProduit reseau 3 passe supervision charge Toto charge Titi progression produiTop graphisme 2 passe supervision charge Toto charge Titi progression superProduit reseau 37 progression produiTop reseau 12 progression megaProduit graphisme 100 supervision charge Toto charge Titi progression produiTop graphisme 1 passe supervision charge Toto

progression superProduit graphisme 25 progression produiTop graphisme 43 charge Toto charge Titi

progression produiThyper graphisme 15

interruption

charge Titi

#### Sortie:



```
la specialite reseau peut etre prise en charge par : Toto
la specialite graphisme peut etre prise en charge par : Toto, Titi
le client Roger a commande :
le client Rodgio a commande :
le client Roger a commande : superProduit, megaProduit
le client Rodgio a commande : produiTop, produiThyper
etat des taches pour superProduit :
etat des taches pour produiTop :
etat des taches pour megaProduit :
etat des taches pour produiThyper :
charge de travail pour Toto : superProduit/reseau/45heure(s)
charge de travail pour Titi :
charge de travail pour Toto : superProduit/reseau/45heure(s)
charge de travail pour Titi : superProduit/graphisme/32heure(s)
charge de travail pour Toto : superProduit/reseau/45heure(s)
charge de travail pour Titi : superProduit/graphisme/32heure(s), produiTop/graphisme/67heure(s)
charge de travail pour Toto : superProduit/reseau/45heure(s), megaProduit/graphisme/100heure(s)
charge de travail pour Titi : superProduit/graphisme/32heure(s), produiTop/graphisme/67heure(s)
charge de travail pour Toto : superProduit/reseau/45heure(s), megaProduit/graphisme/100heure(s)
charge de travail pour Titi: superProduit/graphisme/32heure(s), produiTop/graphisme/67heure(s),
produiThyper/graphisme/15heure(s)
charge de travail pour Toto: superProduit/reseau/45heure(s), produiTop/reseau/28heure(s),
megaProduit/graphisme/100heure(s)
charge de travail pour Titi: superProduit/graphisme/32heure(s), produiTop/graphisme/67heure(s),
produiThyper/graphisme/15heure(s)
etat des taches pour superProduit : reseau:0/45, graphisme:0/32
etat des taches pour produiTop: reseau:0/28, graphisme:0/67
etat des taches pour megaProduit : graphisme:0/100
etat des taches pour produiThyper : graphisme:0/15
etat des taches pour superProduit : reseau:5/45, graphisme:7/32
etat des taches pour produiTop : reseau:16/28, graphisme:21/67
etat des taches pour megaProduit : graphisme:0/100
etat des taches pour produiThyper : graphisme:0/15
charge de travail pour Toto: superProduit/reseau/40heure(s), produiTop/reseau/12heure(s),
megaProduit/graphisme/100heure(s)
charge de travail pour Titi: superProduit/graphisme/25heure(s), produiTop/graphisme/46heure(s),
produiThyper/graphisme/15heure(s)
etat des taches pour superProduit : reseau:8/45, graphisme:7/32
etat des taches pour produiTop: reseau:16/28, graphisme:21/67
etat des taches pour megaProduit : graphisme:0/100
etat des taches pour produiThyper : graphisme:0/15
charge de travail pour Toto: superProduit/reseau/37heure(s), produiTop/reseau/12heure(s),
megaProduit/graphisme/100heure(s)
charge de travail pour Titi: superProduit/graphisme/25heure(s), produiTop/graphisme/46heure(s),
produiThyper/graphisme/15heure(s)
etat des taches pour superProduit : reseau:8/45, graphisme:7/32
etat des taches pour produiTop: reseau:16/28, graphisme:23/67
etat des taches pour megaProduit : graphisme:0/100
etat des taches pour produiThyper : graphisme:0/15
charge de travail pour Toto : superProduit/reseau/37heure(s),
Page - 12-13
```

specialites traitees: reseau/18, graphisme/13

produiTop/reseau/12heure(s), megaProduit/graphisme/100heure(s)

charge de travail pour Titi : superProduit/graphisme/25heure(s), produiTop/graphisme/44heure(s),

produiThyper/graphisme/15heure(s)

facturation megaProduit : graphisme:1300

etat des taches pour superProduit : reseau:45/45, graphisme:7/32 etat des taches pour produiTop : reseau:28/28, graphisme:23/67

etat des taches pour megaProduit : graphisme:100/100 etat des taches pour produiThyper : graphisme:0/15

charge de travail pour Toto :

charge de travail pour Titi : superProduit/graphisme/25heure(s), produiTop/graphisme/44heure(s),

produiThyper/graphisme/15heure(s)

etat des taches pour superProduit : reseau:45/45, graphisme:7/32 etat des taches pour produiTop : reseau:28/28, graphisme:24/67

etat des taches pour megaProduit : graphisme:100/100 etat des taches pour produiThyper : graphisme:0/15

charge de travail pour Toto : produiTop/graphisme/43heure(s) charge de travail pour Titi : superProduit/graphisme/25heure(s),

produiThyper/graphisme/15heure(s)

facturation superProduit : reseau:810, graphisme:416 facturation produiTop : reseau:504, graphisme:871

charge de travail pour Toto :

charge de travail pour Titi : produiThyper/graphisme/15heure(s)

facturation produiThyper : graphisme:195 facturations : Roger:2526, Rodgio:1570

