**Rapport du projet** : Réalisation d’une application java pour la gestion

Administrative d’un collège

**Réalisé par :** Aziz ROCHDI

Anass OULED BEN TAHAR

Abdoulaye CAMARA

Noureddine ELALJA

**Encadré par** : Mr. Said ELKAFHALI

Table des matières

[Remerciement**:** 3](#_Toc50141206)

[Introduction générale 4](#_Toc50141207)

[Chapitre 01 Cahier de charge 5](#_Toc50141208)

[Chapitre 02 : modélisation de l’application 5](#_Toc50141209)

[2.1-authentification : 5](#_Toc50141210)

[2.2-affectation : 5](#_Toc50141211)

[2.3-gestion d’absences : 5](#_Toc50141212)

[2.4 gestion d’élèves : 6](#_Toc50141213)

[2.5 Modèle logique de données MLD ; 7](#_Toc50141214)

[Chapitre 03 : réalisation 7](#_Toc50141215)

[3.1 présentation du langage java 8](#_Toc50141216)

[3.1.1 bref historique : 8](#_Toc50141217)

[3.1.2 caractéristiques de java : 8](#_Toc50141218)

[3.2 présentation du MySQL 9](#_Toc50141219)

[3.2.1 petite définition : 9](#_Toc50141220)

[3.2.2 caractéristiques de MySQL : 9](#_Toc50141221)

[3.3 Description de l’application : 9](#_Toc50141222)

[3.3.1 page d’authentification : 9](#_Toc50141223)

[3.3.2 page d’accueil 10](#_Toc50141224)

[3.3.3 la page affectation : 11](#_Toc50141225)

[3.3.4 Gestion d’absence 12](#_Toc50141226)

[3.3.5 gestion d’élève 13](#_Toc50141227)

[Conclusiongénérale 15](#_Toc50141228)

# 

# Remerciement**:**

*Avant de commencer à élaborer nous voulons bien, présenter tous nos remerciements à notre professeur monsieur Said ELKAFHALI, qui a nous donné la possibilité de vivre une très belle expérience, en préparant cette application, qui a été vraiment une occasion pour mettre en pratique tous ce qu’on a vu en théorie.*

# 

# Introduction générale

Dans le cadre des mini projets réalisés au sein de **FST SETTAT** nous étions amenés à réaliser une application java qui va permettre la gestion administrative d’un collège pour remédier les différents obstacles liés au mode traditionnel du gestion

Ce rapport présente l’ensemble des étapes suivies pour réaliser l’application, il contient trois chapitres principaux :

Le premier chapitre est consacré à la description de cahier de charge et à l’objectif du projet.

Le deuxième chapitre a pour but la modélisation de l’application.

Le dernier chapitre consiste à la présentation des technologies utilisés pour le développement de cette application ainsi que sa réalisation

Finalement nous terminons notre rapport par une conclusion générale.

# Chapitre 01 Cahier de charge

L’objectif de notre application est de fournir une solution adaptative à la gestion administrative d’un collège en traitant les points suivants :

* La gestion des affectations des cours et des salles.
* La gestion des absences en permettant d’ajouter une absence et d’afficher les absences d’une classe donnée.
* La gestion des élèves en donnant la possibilité d’ajouter un élève et de le transférer d’une classe a une autre.

# Chapitre 02 : modélisation de l’application

Les cas d’utilisations

## 2.1-authentification :

Lorsque l’utilisateur demande l’accès à l’application, id doit avoir le nom d’utilisateur et un mot de passe valides.

Si le login et/ou le mot de passe ne sont pas corrects l’application va afficher un message d’erreur, sinon elle va afficher la page d’accueil, dans la quelle on peut choisir une option parmi les options disponibles.

## 2.2-affectation :

Cette option permit d’effectuer les affectations des cours aux classes en précisant le classe et le cours concernées et date du cours.

Si les informations sont correctes on va afficher un message pour confirmer l’opération sinon un message qui précise l’erreur commit va s’afficher.

## 2.3-gestion d’absences :

Ce cas comporte deux scénarios :

* Ajouter une absence : permit d’affecter les absences aux élèves qui ont étés absents en précisant le nom et le prénom d’élève et sa classe ainsi que la date de l’absence.
* afficher les absences : permit à l’administration de voir les absences d’une classe donnée, en sélectionnant une classe et en saisissant une date.

S’il y’a des erreurs on aura des messages qui les mentionnent

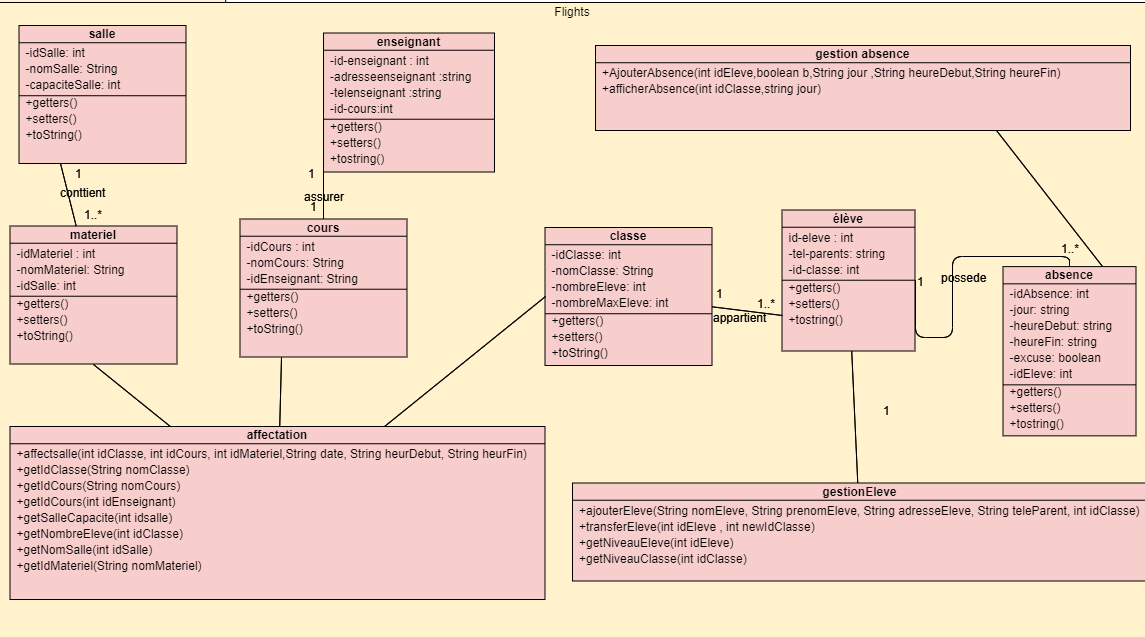
Sinon l’opération va s’effectuer.

## 2.4 gestion d’élèves :

Ce cas d’utilisation donne deux possibilités :

* Ajouter un élève qui permit d’ajouter un élève en saisissant son nom et son prénom et le numéro de téléphone de ses parents, et en choisissant une classe qui a encours de places libres et qui est correspondant avec son niveau.
* Transférer un élève cette option va permettre de transférer un élève d’une classe à une autre en sélectionnant l’élève consterné, la classe de départ et la classe d’arrivé parmi les classes qui ont encours de places libres.

## 2.5 Modèle logique de données MLD ;



# Chapitre 03 : réalisation

Dans ce chapitre nous présentons l’environnement logiciel du projet ainsi que les interfaces de l’application des différents scénarios d’utilisation.

Pour l’implémentation nous avons utilise le langage de programmation java et l’environnement de gestion de la base de données MYSQL.

## 3.1 présentation du langage java

### 3.1.1 bref historique :

 Java est une langage de programmation orienté objet créé par James Gostling et Patrick Naughton, employés de Sun Microsystemes avec le soutien de Bill Joy (cofondateur de Sun Microsystemes) en 1982,présenté officiellement le 23 mai 1995

Au sunWord.

La société Sun a été rachetée en 2009 par la société Oracle qui détient et maintient désormais java.

### 3.1.2 caractéristiques de java :

Java a plusieurs caractéristiques par rapport aux autres langages de programmation nous présentons dans la suite les plus importants :

**Oriente objet**

Comme la plupart des langages récents, Java est orienté objet. Chaque fichier source contient la définition d'une ou plusieurs classes qui sont utilisées les unes avec les autres pour former une application. Java n'est pas complètement objet car il définit des types primitifs (entier, caractère, flottant, booléen,...).

**Indépendant de la machine**  
C'est peut-être un des plus gros points fort de Java. Un source Java compilé donne des *"bytes-codes"* sorte de pseudo-assembleur qui s'adresse à une *machine virtuelle Java*. Cela signifie que pour exécuter ces bytes-codes, il faut un interpréteur java qui simule cette machine virtuelle. Les logiciels de navigation compatibles Java intègrent un tel interpréteur. Les applications ou applets Java sous leur forme "bytes-codes" sont donc indépendantes de la machine physique.

**Multi-thread**  
Une application Java peut lancer plusieurs tâches ou processus indépendants qui s'exécutent simultanément. Java gère un mécanisme de moniteurs, qui permet de synchroniser ces processus ou *"threads"*.

**Sécurisé**  
Assurer la sécurité des programmes qui circulent sur le réseau, est un problème majeur. Java a intégré, dès la conception, plusieurs mécanismes de sécurité visant à [rendre](http://www.brunette.brucity.be/NewBrunette/files/Java/TutJava/securite.htm) les programmes fiables et à éliminer les risques de virus ( [vérification](http://www.brunette.brucity.be/NewBrunette/files/Java/TutJava/security.gif) du "bytes codes", pas de manipulation pointeurs ).

**Simple**  
Plus simple que le C++, Java n'a pas de pointeurs - arithmétique sur les pointeurs, pas de fonctions. Il n'intègre ni l'héritage multiple, ni la surcharge d'opérateurs.

## 3.2 présentation du MySQL

### 3.2.1 petite définition :

 **MySQL**est une [**base de données**](http://www.mosaiqueinformatique.com/formation-base-de-donnees/formation-access.html) relationnelle libre qui a vu le jour en 1995 et très employée sur le Web, souvent en association avec PHP (langage) et Apache (serveur web). MySql fonctionne indifféremment sur tous les systèmes d'exploitation (Windows, Linux, Mac OS notamment).

Le principe d'une base de données relationnelle est d'enregistrer les informations dans des **tables**, qui représentent des regroupements de données par **sujets**(table des clients, table des fournisseurs, table des produits, par exemple). Les tables sont reliées entre elles par des **relations**.

### 3.2.2 caractéristiques de MySQL :

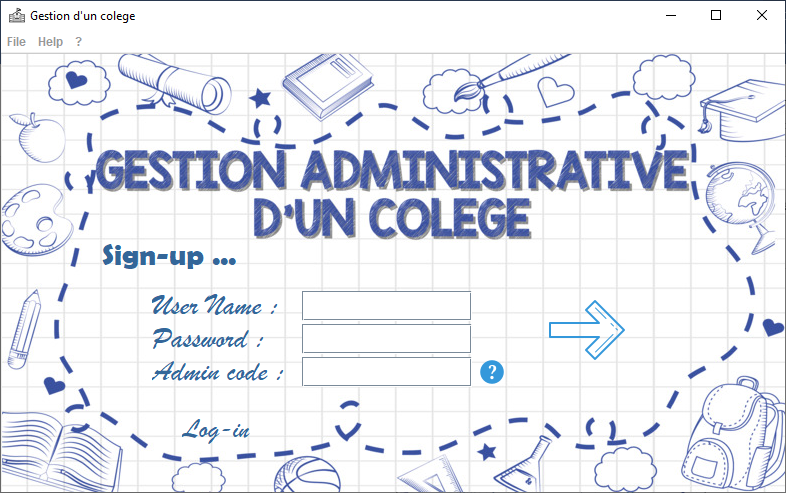
* + - De nombreux langages de programmation peuvent avoir accès à MySql
    - Il fonctionne avec une multitude de systèmes d’exploitation
    - Il utilise le langage de requête SQL
    - Il propose de nombreux moteurs de base de données

## 3.3 Description de l’application :

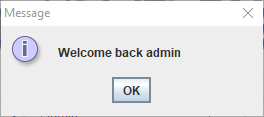
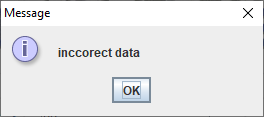
Dans cette section nous allons présenter les différents page et scenarios de l’application.

### 3.3.1 page d’authentification :

C’est la première page à rencontrer qui permet à un utilisateur soit à s’identifier s’il possède un compte ou de le créer s’il ne le possède pas.



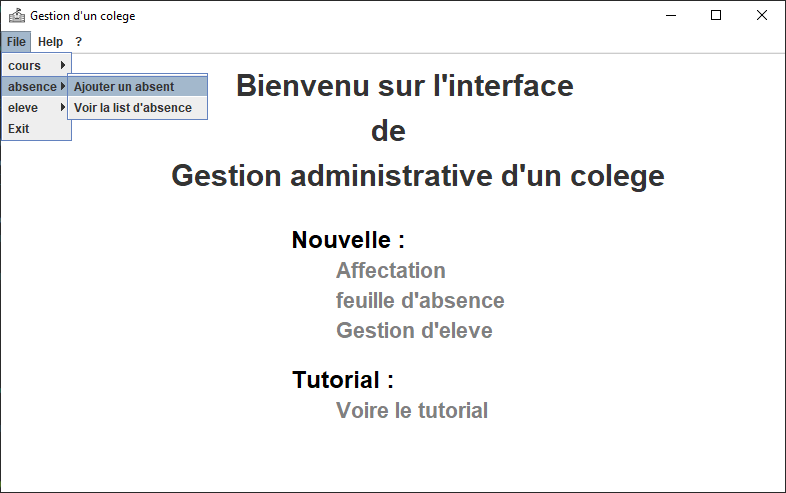
Si les informations saisis par l’utilisateur ne sont pas correctes une fenêtre d’erreur va s’afficher sinon on va afficher un message qui indique que les informations sont valides



### 3.3.2 page d’accueil

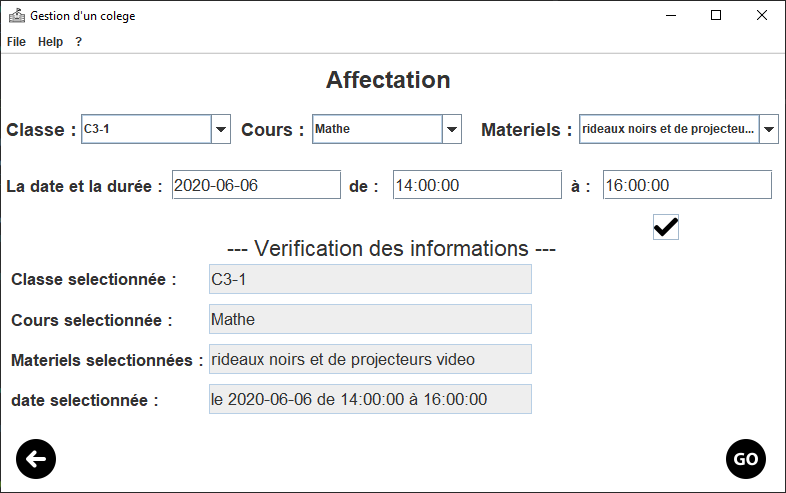
En cliquant sur le bouton OK dans la fenêtre « welcome back admin « on va se trouver devant la page d’accueil qui va nous permettre d’accéder au reste de l’application soit en utilisant le menu bar soit en cliquant sur boutons :

* Affectation : pour accéder à la page d’affectation des cours.
* Feuille d’absence : pour ajouter ou voir les absences.
* Gestion d’élève : pour ajouter ou transfèrer un élève.

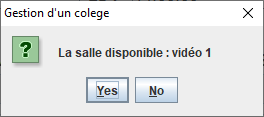


### 3.3.3 la page affectation :

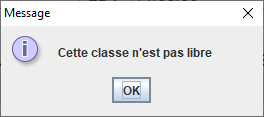
Cette page permet d’affecter un cours à un classe on en entrant les informations demandées.



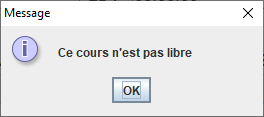
Pour valider l’opération on clique sur le bouton « **✓** », si y’a pas des erreurs on va avoir un message pareil au message suivant :



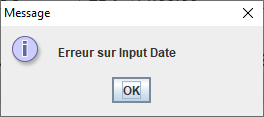
Sinon au cas d’erreur on va se trouver devant un parmi les messages suivants :



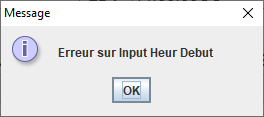
Si la classe sélectionner a un autre cours dans le même heure



Si le prof qui assure ce cours n’est pas disponible



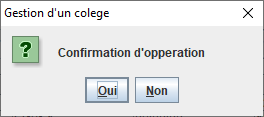
Si on n’a pas respecté le format « yyyy-MM-dd », ou la si date saisi n’est dans l’intervalle de l’année scolaire ou bien on a saisi une date inférieure à la date actuelle.



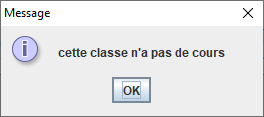
Si on n’a pas respecter le format « HH:mm:ss », ou si Si l’Heure début supérieur à l’heure fin

### 3.3.4 Gestion d’absence

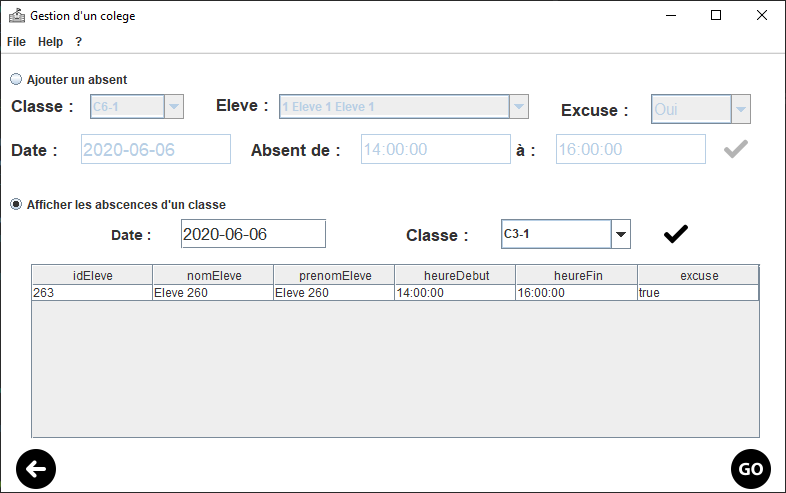
Cette page permet soit d’ajouter une absence soit les afficher en remplissant les champs demandés, et en cliquant sur le bouton « **✓** » correspondant à chaque cas et puis le bouton « Go », si y’a pas des erreurs on aura le message de confirmation suivant pour le cas d’ajout des absences.



Sinon on peut avoir une erreur tel que le suivant

  Si la classe ou il appartient l’élève n’a pas de cours dans la période choisie

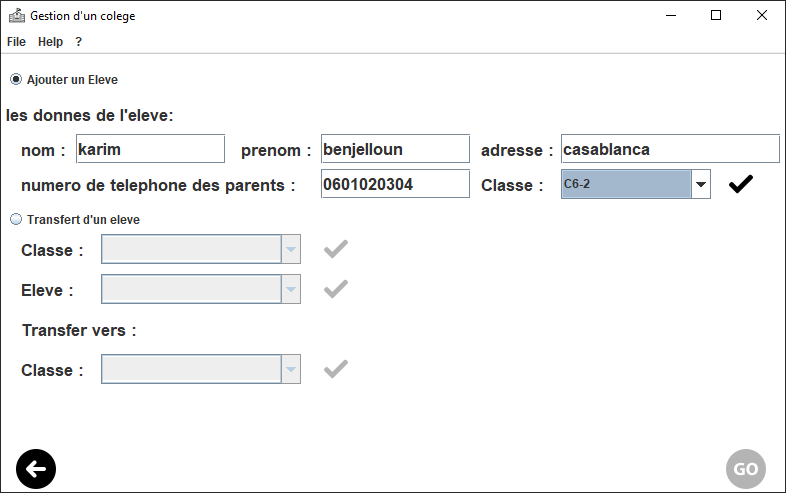
Et pour le cas d’affichage des absences on va avoir la table en bas dans la fenêtre suivante.



### 3.3.5 gestion d’élève

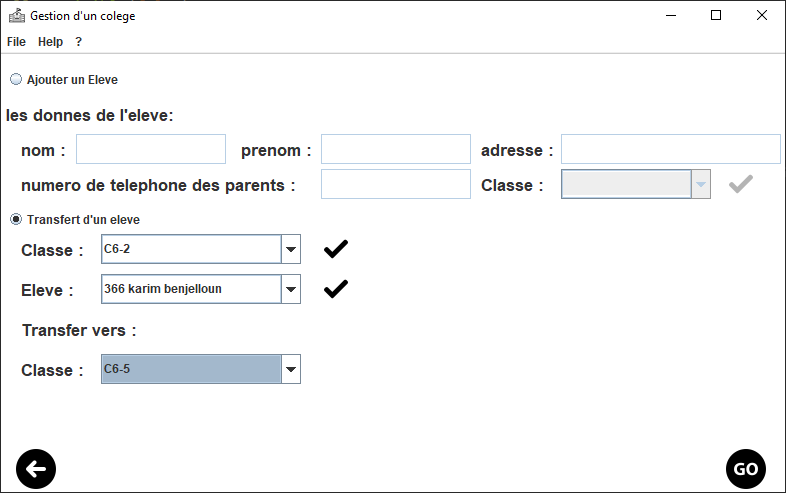
Cette page est divisée aussi en deux partis une partie pour ajouter un élève en haut, et une partie pour le transférer en bas voici les deux cas d’utilisations.

-le cas d’ajout : permet d’ajouter un élève, en saisissant son prénom, son nom, son adresse et le numéro de téléphone de ses parents, après on choisi une classe qui correspondant a son niveau.si il y’a une erreur on aura message qui l’indique, sinon on aura un message de confirmation

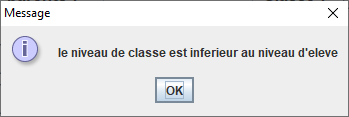
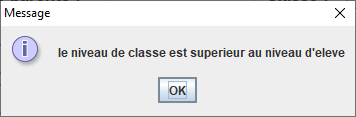


-le cas de transfert : permet de transférer un élève d’une classe à une autre en choisissant le

Classe de départ, l’élève concerné et la classe de destination à chaque choix on clique sur le bouton « **✓** » correspondant.



Lors de l’ajout d’élève on peut avoir l’erreur suivant qui veut dire que la classe de départ et la classe n’ont pas le même niveau.



# **Conclusion générale**

Au cours de la réalisation de cette application nous avons acquérir beaucoup de compétences personnels et techniques, et elle a été une occasion pour mettre en pratique tous ce qu’on a vu dans le module de ‘’programmation oriente objet avec java ‘’, elle a nous donnée également la possibilité de vivre notre première expérience de réaliser une telle application.