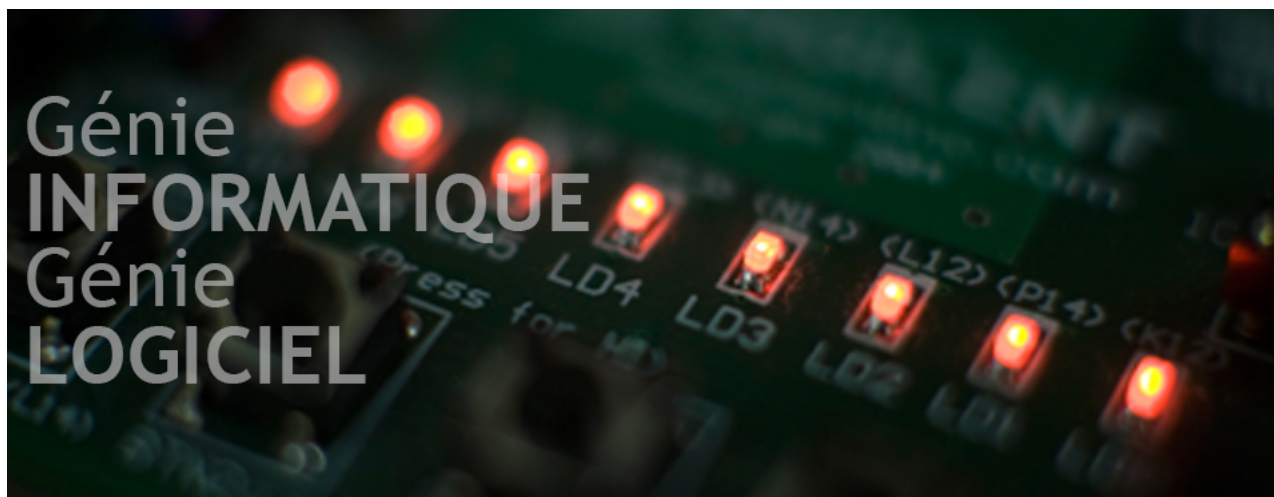




TRAVAIL PRATIQUE

MODELISATION ET PROGRAMMATION ORIENTEE-OBJET AVEC JAVA

«UN PETIT GESTIONNAIRE DE TACHE»



Realise par :
APEDO KODZO SITSOFE
Promo 21,
2016-2017

ENCADRANTS :

Mr. Ho Tuong Vinh

SOMMAIRE

INTRODUCTION

LES EXIGENCES

Les exigences fonctionnelles et diagramme de cas d'utilisation

Les exigences non-fonctionnelles

LA CONCEPTION

Diagramme de classes

Diagramme de séquence

Diagramme de cas d'utilisation

L'IMPLEMENTATION

Choix de la plateforme de développement et du langage

Choix de la base de données

LE TEST D'ACCEPTATION

CONCLUSION

ANNEXE

A INTRODUCTION

Le travail qui nous a été demandé concerne la conception d'une application de gestion des tâches d'un groupe de membres de personnes avec l'outil JAVA.

Le travail nous est donné avec l'exigence de l'utilisation du langage JAVA et aussi d'une base de données mais précisons aussi qu'il nous a été donné le libre choix d'utiliser la base de données que nous voulons pour la conception de l'application. L'application est sensée gérer des tâches et des personnes auxquelles seront allouées des personnes pour effectuer ces différentes tâches. De plus, l'application doit être conçue en langage JAVA avec une base de données et ensuite faire le test unitaire avec JUnit

B LES EXIGENCES DU PROJET

B - 1 Les exigences fonctionnelles

Selon le but visé par le programme, nous avons pour objectif :

CRÉER :

Nous devons créer un membre.

Le membre doit avoir comme attribut, un identifiant et un nom.

Ensuite créer une tâche avec comme attributs, l'identifiant, le nom de la tâche, la description et le statut de la tâche. Le statut de la tâche est sensée donner l'état de la tâche, si elle est allouée à un membre, si elle est en cours ou terminée. Aussi il faudrait vérifier si le membre ou la tâche n'existe pas avant d'en créer.

RECHERCHER :

Nous devons rechercher un membre ou rechercher une tâche. Cette fonction de recherche de tâche ou de membre est sensée nous permettre de savoir si l'élément recherché est dans la base, savoir sa position afin d'effectuer des opérations ou de voir son état. La recherche de la tâche ou du membre peuvent toutefois passer par leurs noms ou leur identifiant.

MODIFIER :

Pour cette option du programme, elle nous permettra de modifier le nom d'une tâche ou d'un membre, modifier les identifiants également de ces tâches et de ses membres et bien entendu, il faut vérifier l'existence de l'élément avant par la recherche

SUPPRIMER :

La suppression d'une tache ou d'un membre peut se faire par le nom ou l'identifiant, mais pour supprimer un élément, il faudrait préalablement vérifier son existence dans la base avant de pouvoir le supprimer.

ASSIGNER :

L'assignation d'un membre ou plusieurs membres a une tache comme l'assignation d'une ou plusieurs taches a un membre se gèrent dans le programme par les noms des tache ou des membres. La possibilité est donnée dans les deux sens pour les assignations

Diagramme de cas d'utilisation :

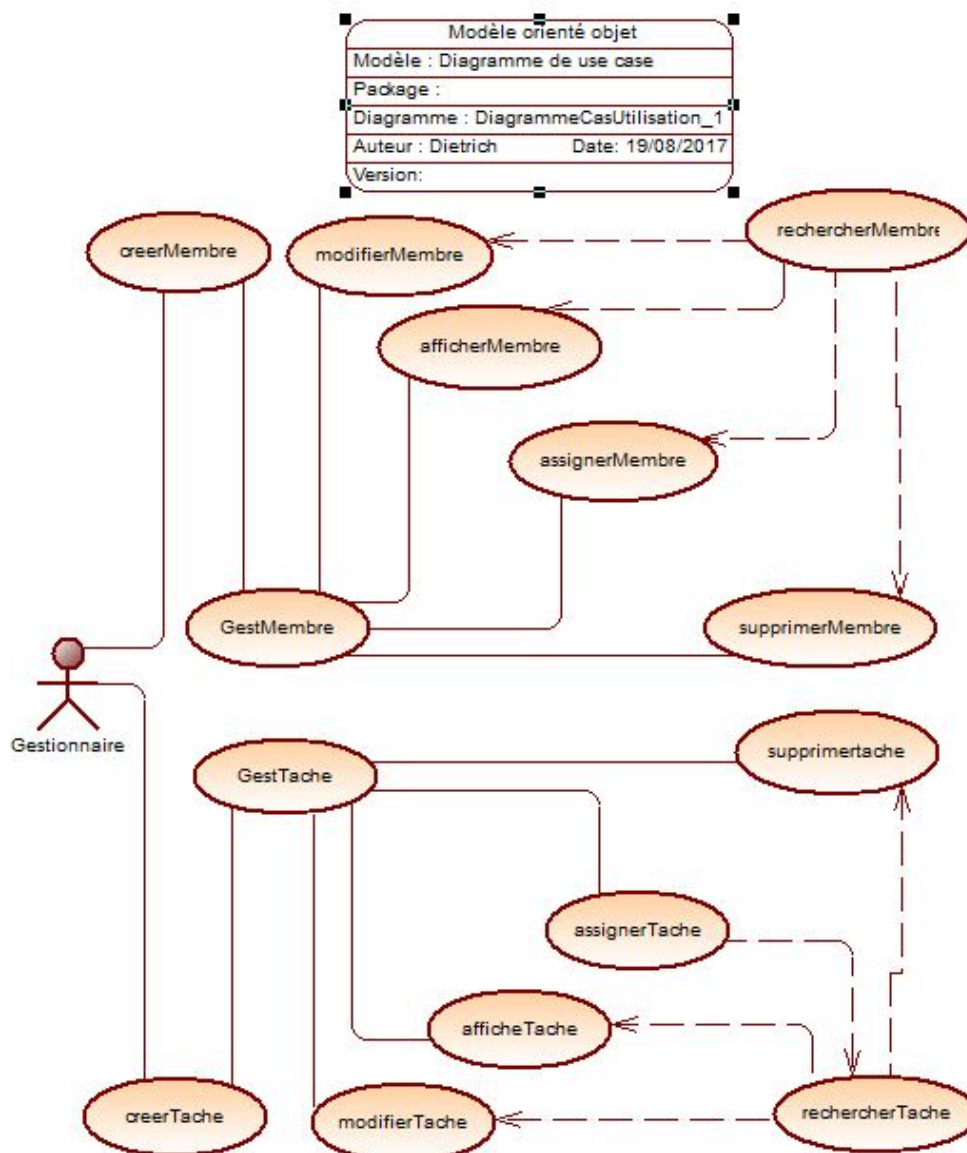


FIGURE 1: Diagramme de cas d'utilisation

B - 2 Les exigences non-fonctionnelles

Une exigence non-fonctionnelle est une exigence qui caractérise une propriété (qualité) désirée du système telle que sa performance, sa robustesse, sa convivialité, sa maintenabilité, etc. L'utilité d'un système logiciel est déterminée par ses exigences fonctionnelles et ses caractéristiques non-fonctionnelles.

Nous présentons, ci-dessous, le tableau des exigences non-fonctionnelles de notre application :

Description	Exigences fonctionnelles requises
Le système doit être disponible durant toutes les opérations	Toutes les exigences fonctionnelles
Les données doivent être disponibles et accessibles à tout moment où l'utilisateur en a besoin	Toutes les exigences fonctionnelles
En cas de défaillance du système, les données ne doivent pas subir un changement	Toutes les exigences fonctionnelles
Le système ne doit pas mettre un long temps avant de donner une réponse à l'utilisateur	Toutes les exigences

FIGURE 2: Tableau des exigences non-fonctionnelles

C CONCEPTION

C - 1 Diagramme de classe

Nous avons donc les diagrammes comme suit :

Diagramme de classe :

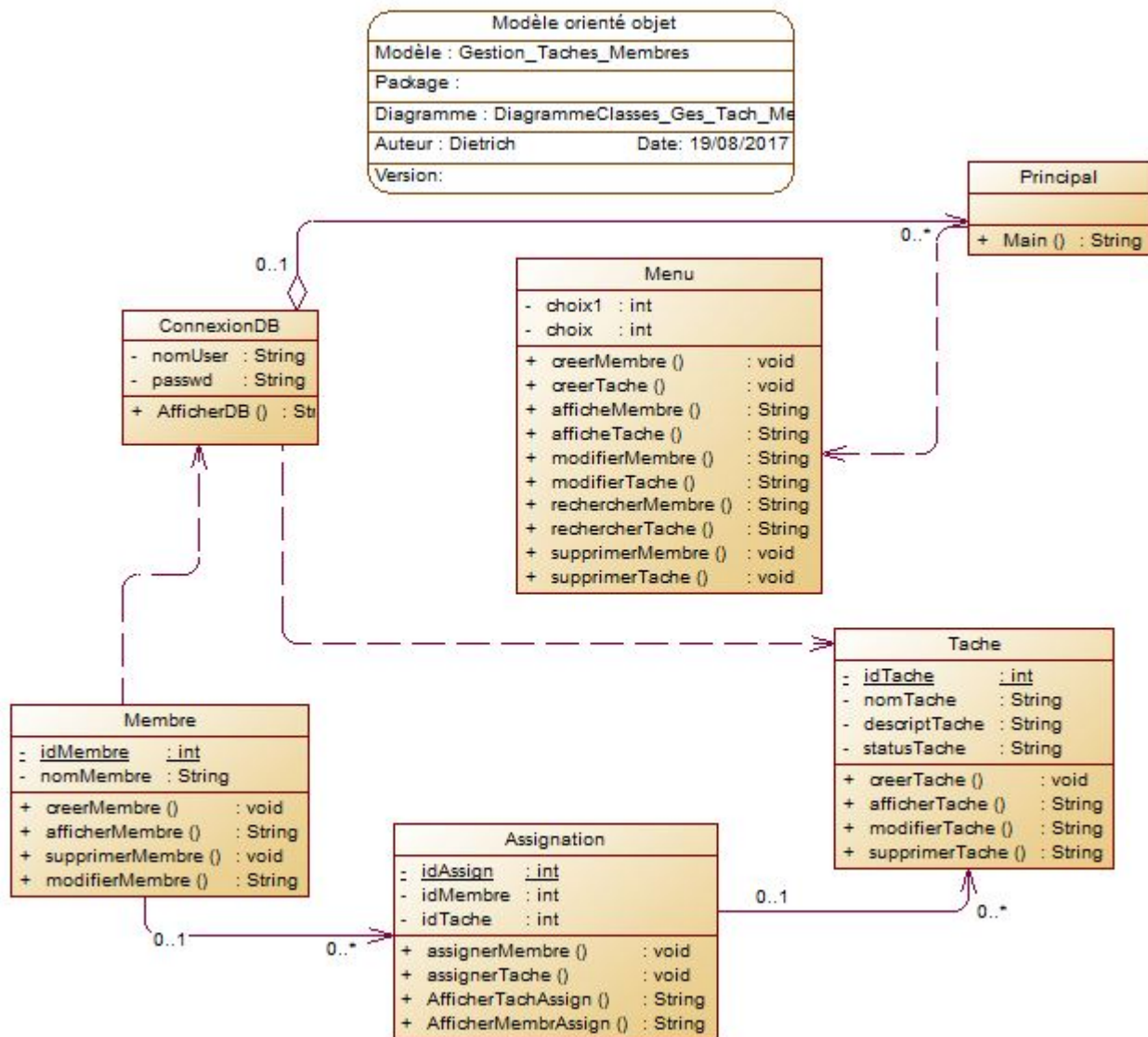


FIGURE 3: Diagramme de classe

Dans la réalisation de notre système, nous avons juger bon de définir trois classe a savoir :

- La classe Membre
- La classe Tache
- La classe Assignation

A une personne peut être assigner une ou plusieurs taches, dont des taches de différentes statut a savoir

- statut :initialiser
- statut :en cours
- statut :fini

C - 2 Diagramme de séquence

Dans le diagramme de séquence, comme de son rôle, il présente la succession chronologique des opérations réalisées par un acteur, ce qui fait de lui un des diagrammes d'interaction.

La création d'un membre :

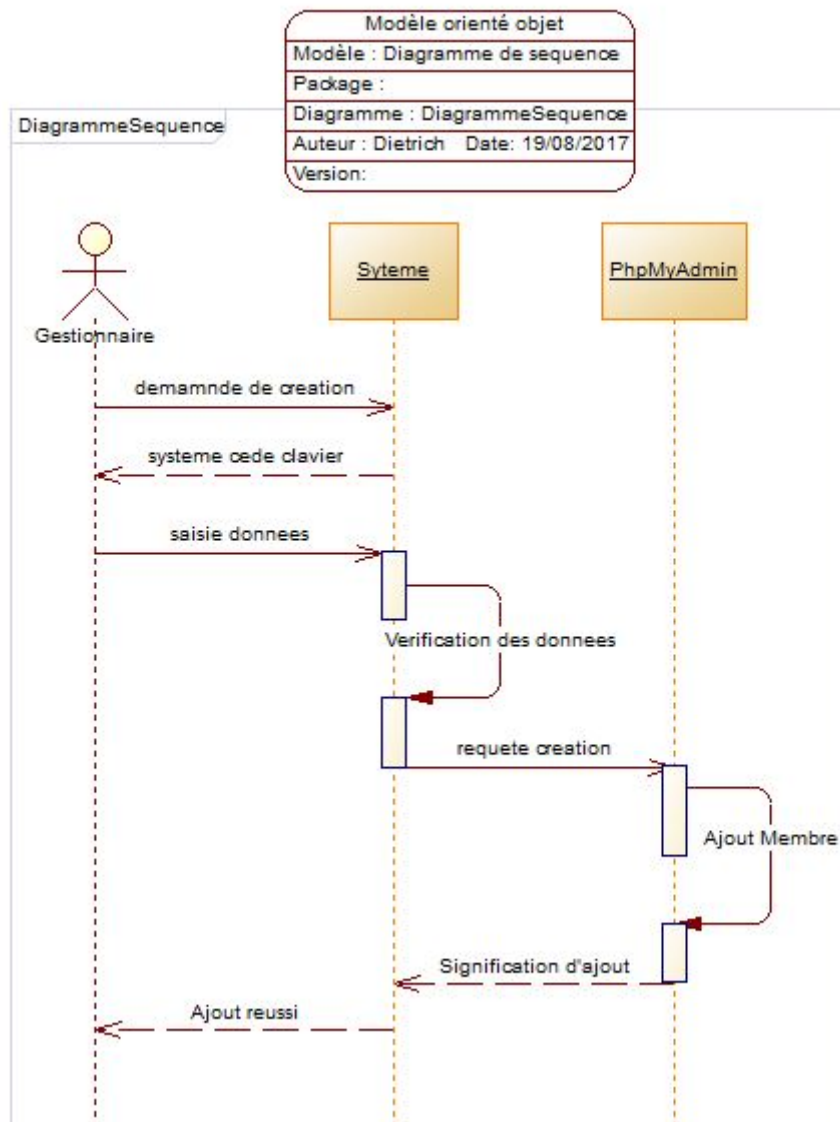


FIGURE 4: Diagramme de séquence pour création d'un membre

D IMPLEMENTATION

D - 1 Choix de l'IDE et du Langage de programmation

Vu la facilité, l'adaptabilité et de son utilisabilité, nous avons choisi Netbeans IDE qui est un environnement de développement intégré permettant de développer dans

une multitude de langages de programmation dont Java. Nous avons choisit comme langage de programmation Java, qui est un langage multi-plateforme, portable, modulable, structuré, orienté objet et dont la maintenance est beaucoup plus facile.

D - 2 La base de données choisie

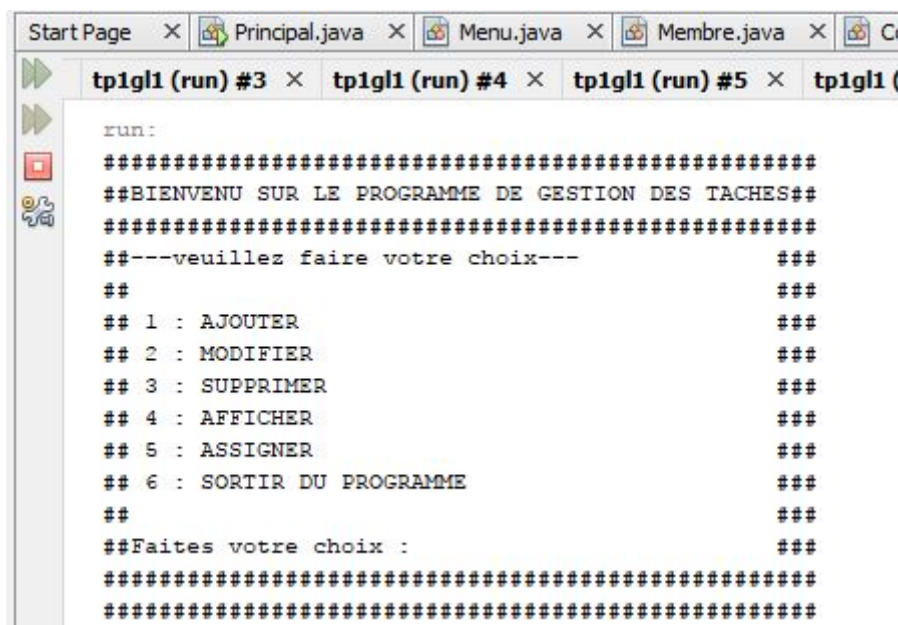
Nous avons choisi comme base de données MySQL de WAMPSERVER 3.0.6 car il est un outil adapté permettant de traiter, de gérer un grand volume de données et d'organiser correctement ces données. Il est donc l'un des plus connus et des plus utilisés des Systèmes de Gestion de Bases de Données Relationnelles (SGBDR) et accessibles par la plupart des langages de programmation à savoir : Java, PHP, C, C++, Python, etc.

E LE TEST D'ACCEPTATION

Nous démontrerons quelques grandes fonctionnalité de notre programme avec des captures d'écran pour montrer le fonctionnement.

Nous commencerons par l'affichage de notre Menu nous permettant de surfer dans le programme


Test1 du Menu :



```
run:
#####
##BIENVENU SUR LE PROGRAMME DE GESTION DES TACHES##
#####
##---veuillez faire votre choix---###
##                                   ###
## 1 : AJOUTER                      ###
## 2 : MODIFIER                     ###
## 3 : SUPPRIMER                    ###
## 4 : AFFICHER                     ###
## 5 : ASSIGNER                     ###
## 6 : SORTIR DU PROGRAMME          ###
##                                   ###
##Faites votre choix :               ###
#####
#####
```

FIGURE 5: Test fonctionnement du Menu

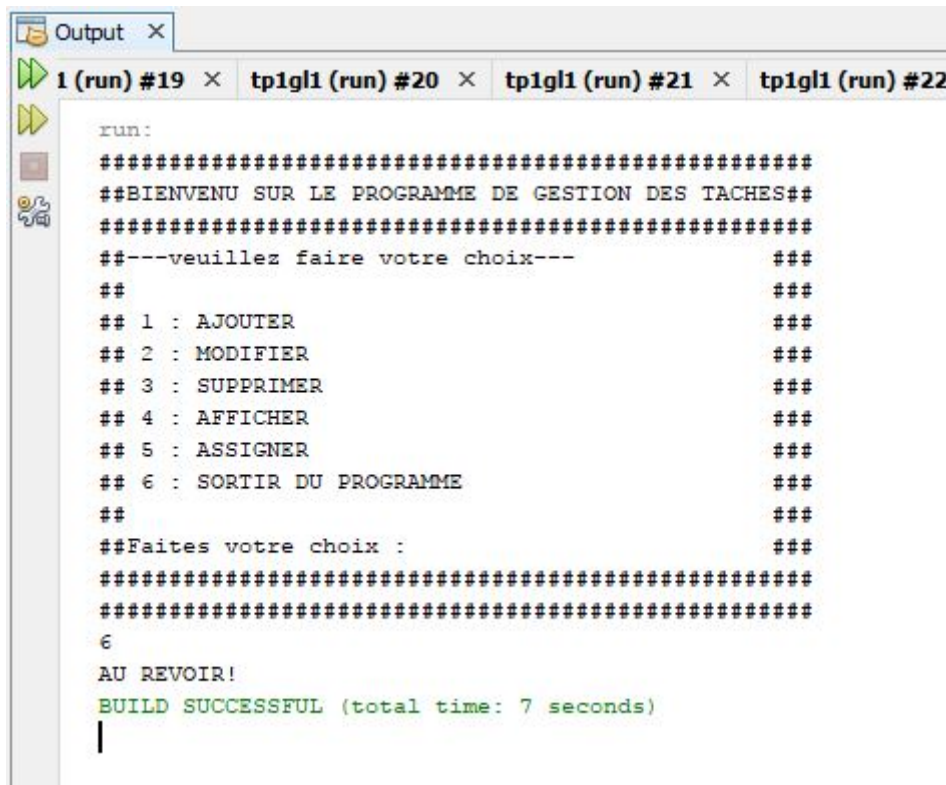
Test2 du Menu :



```
tp1.Principal > main > while (true) > switch (choix1) > case 7: >
Output x JUnitTestMembre.java x
x tp1gl1 (run) #41 x tp1gl1 (run) #42 x tp1gl1 (run) #43 x t
run:
#####
##BIENVENU SUR LE PROGRAMME DE GESTION DES TACHES##
#####
##---veuillez faire votre choix---###
#####
## 1 : AJOUTER###
## 2 : MODIFIER###
## 3 : SUPPRIMER###
## 4 : AFFICHER###
## 5 : ASSIGNER###
## 6 : SORTIR DU PROGRAMME###
#####
##Faites votre choix :###
#####
#####
7
ERREUR DE SAISIE!
BUILD SUCCESSFUL (total time: 5 seconds)
|
```

FIGURE 6: Contrôle des erreurs de choix au Menu

Sortir du programme :



```
Output x
1 (run) #19 x tp1gl1 (run) #20 x tp1gl1 (run) #21 x tp1gl1 (run) #22
run:
#####
##BIENVENU SUR LE PROGRAMME DE GESTION DES TACHES##
#####
##---veuillez faire votre choix---###
#####
## 1 : AJOUTER###
## 2 : MODIFIER###
## 3 : SUPPRIMER###
## 4 : AFFICHER###
## 5 : ASSIGNER###
## 6 : SORTIR DU PROGRAMME###
#####
##Faites votre choix :###
#####
#####
6
AU REVOIR!
BUILD SUCCESSFUL (total time: 7 seconds)
|
```

FIGURE 7: Sortir du programme

Test choix au sous-menu :

```
2

---veuillez faire votre choix---
1 : pour tache
2 : pour Membre
choix : 3
Faites le bon choix! : 
#####
##BIENVENU SUR LE PROGRAMME DE GESTION DES TACHES##
#####
##---veuillez faire votre choix---      ###
##                                     ###
## 1 : AJOUTER                         ###
## 2 : MODIFIER                        ###
## 3 : SUPPRIMER                       ###
## 4 : AFFICHER                        ###
## 5 : ASSIGNER                        ###
## 6 : SORTIR DU PROGRAMME             ###
##                                     ###
##Faites votre choix :                  ###
#####
#####
|
```

FIGURE 8: Test fonctionnement du sous-menu

Test d'ajout de membre :

```
#####

1

---veuillez faire votre choix---
1 : pour tache
2 : pour Membre
choix : 2
connexion etablie
Entrer le nom du membre : Dr_QUANG
ajout reussi
deconnexion reussie

BASE DE DONNEE :

connexion etablie

|idMembre|          |MEMBRE|
1          cephas
2          apedo
3          rubens
4          bombia
5          PERRQULT
6          olivier
7          Diet
8          Dr VINH
9          Dr_QUANG

#####
##BIENVENU SUR LE PROGRAMME DE GESTION DES TACHES##
```

FIGURE 9: Test d'ajout d'un membre

Test d'ajout de tache :

```

Output X JUnitTestMembre.java X
lg11 (run) #35 X tp1gl1 (run) #36 X tp1gl1 (run) #37 X tp1gl1 (run) #38 X tp1gl1 (run)
#####
1
---veuillez faire votre choix---
1 : pour tache
2 : pour Membre
choix : 1
driver OK
connexion effectuer

BASE DE DONNEE :

driver OK
connexion effectuer

|idTache|      |nomTache|      |statusTache|      |descripTache|
1          jhv          kjh          lk
2          luhipjoi      lljuttrder      ljpgtrdtre
3          lop          lopi          lpoid
Entrez le nom de la tache : laver
Entrez la description de la tache : carreaux
Entrez le status de la tache : en cours
ajout reussi
deconnexion reussie

BASE DE DONNEE :

driver OK
connexion effectuer

|idTache|      |nomTache|      |statusTache|      |descripTache|
1          jhv          kjh          lk
2          luhipjoi      lljuttrder      ljpgtrdtre
3          lop          lopi          lpoid
4          laver          carreaux          en cours

```

FIGURE 10: Test d'ajout test

Test de modification de membre :

```
#####
2

---veuillez faire votre choix---
1 : pour tache
2 : pour Membre
choix : 2
connexion etablie
Entre le nom du membre a modifier: Dr VINH
Entre le nouveau nom du membre modifier: Mr HO
modification effectue avec succes

BASE DE DONNEE :

connexion etablie

|idMembre|          |MEMBRE|
1          cephas
2          apedo
3          rubens
4          bombia
5          PERRQULT
6          olivier
7          Diet
8          Mr HO
9          Dr_QUANG
```

FIGURE 11: Test de modification de membre

Test d'assignation d'un membre a une tache :

```

##Faites votre choix :                                     ###
#####
#####
5
1 : Modifier l'assignation d'une tache
2 : Assigner une tache a un membre
choix : 2
driver OK
connexion reussi

LES TACHES :
|ID TACHES|      |NOM DES TACHES|      | DESCRIPTION|
1          jhv          kjh
2          luhipjoi      lljuttrder
3          lop           lopi
4          laver         carreaux
driver OK
connexion effectuer
ENTRER IDENTIFIANT DU MEMBRE : 3
ENTRER IDENTIFIANT DU TACHE : 4
Assignation reussie
deconnexion reussie
#####
##BIENVENU SUR LE PROGRAMME DE GESTION DES TACHES##
#####
##---veuillez faire votre choix---                        ###
##                                                         ###
## 1 : AJOUTER                                             ###
## 2 : MODIFIER                                            ###
## 3 : SUPPRIMER                                           ###
## 4 : AFFICHER                                            ###
## 5 : ASSIGNER                                             ###
## 6 : SORTIR DU PROGRAMME                                ###
##                                                         ###
##Faites votre choix :                                     ###
#####
#####

```

FIGURE 12: Test d'assignation d'un membre a une tache

Test de JUnit pour le menu :

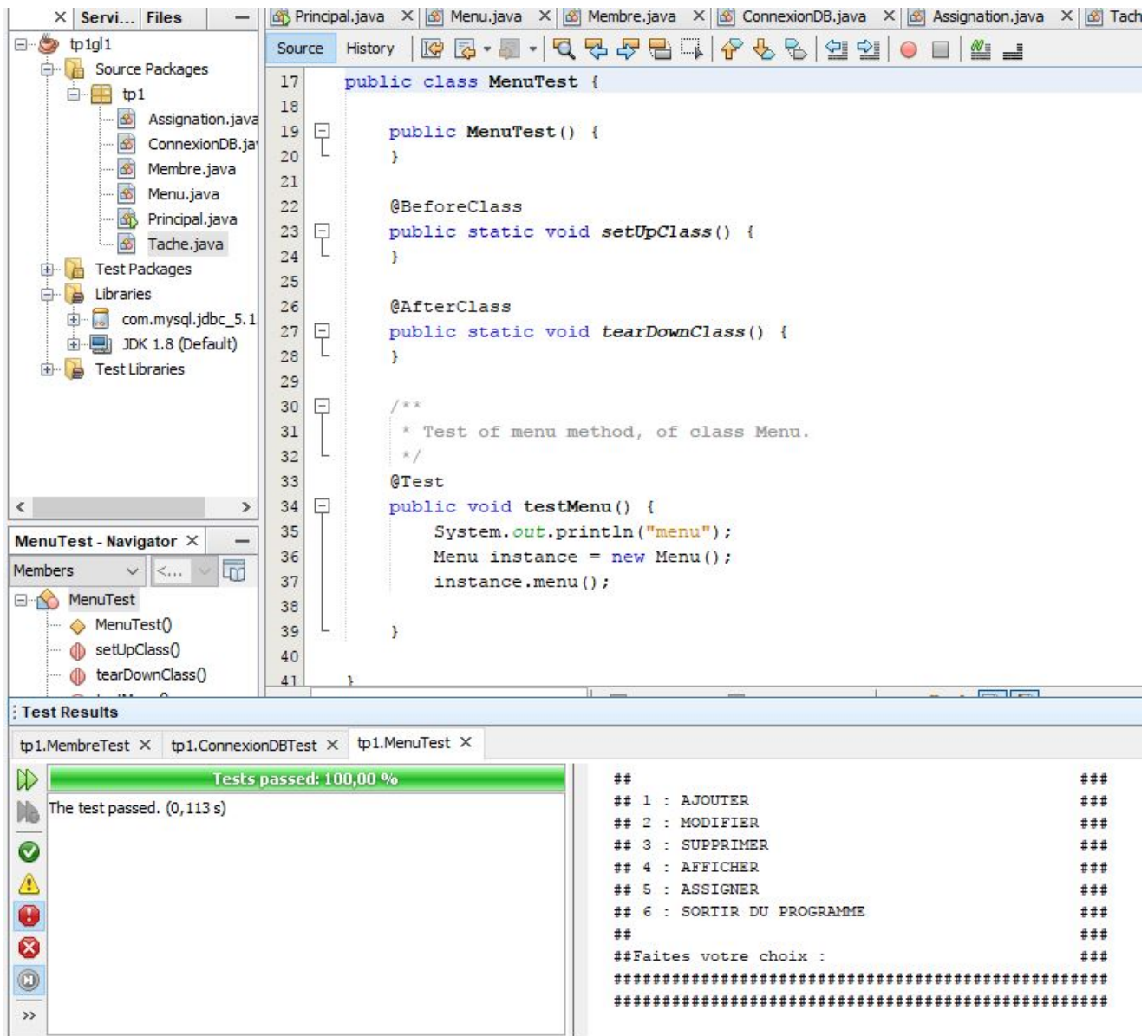


FIGURE 13: Test de JUnit pour le menu

Test JUnit pour la connexion a la base de données :

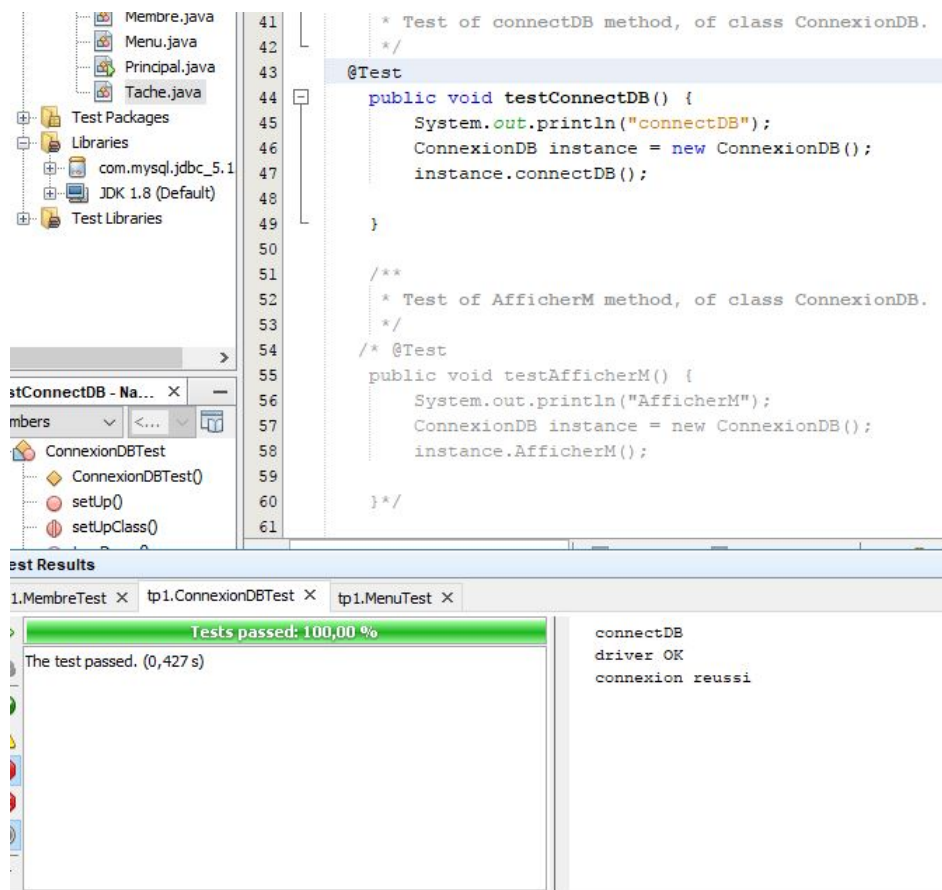


FIGURE 14: Test JUnit pour la connexion a la base de donnée

F CONCLUSION

L'application que nous avons développée pour la gestion des tâches et des membres permet de créer, de supprimer, d'afficher, d'assigner et de modifier des tâches et aussi des membres. . Nous avons utilisé UML pour modéliser notre système et le représenter avec le logiciel POWER AMC et utiliser le langage de développement Java, dans un environnement de développement intégré NetBeans IDE, pour développer notre application. Comme Framework, nous avons utilisé JPA 2.1 et une base de données MySQL. Pour tester le bon déroulement et fonctionnement de notre application, nous avons utilisé JUnit 4 et nous confirmons que ce programme marche bien et répond aux différentes exigences et spécifications données. Ce programme prend en charge plusieurs numéros de téléphone et adresses d'un contact, cependant, nous pourrions l'améliorer pour gérer des groupes de contacts. Ce programme nous a permis de prendre en main l'utilisation de l'outil de développement qui est Netbeans IDE

G ANNEXE

Principal.java

```
package tp1;

import java.util.Scanner;

public class Principal {

    private static Scanner sc;

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Membre mbre=new Membre();
        Tache tach = new Tache();
        ConnexionDB cnx = new ConnexionDB();
        sc = new Scanner(System.in);
        Assigination as=new Assigination();

        Menu men = new Menu();

        OUTER:
        while (true) {
            men.menu();
            int choix1 = sc.nextInt();

            switch (choix1) {
                case 5:
                    System.out.println(" ");
                    System.out.println("1 : Modifier l'assignation d'une tache ");
                    System.out.println("2 : Assigner une tache a un membre ");
                    System.out.print("choix : ");
                    int choixassign = sc.nextInt();
                    switch (choixassign) {
                        case 1:
                            cnx.connectDB();
                            //cnx.affficherM();
                            cnx.afficherT();
                            as.connectDB();
                            as.afficherTachAssign();
                            as.afficherAssign();
```

```

        as.modification();
        break;
    case 2:
        cnx.connectDB();
        // cnx.afficheBD();
        cnx.afficherT();
        as.connectDB();
        //as.afficherTachAssign();
        //as.afficherAssign();
        as.assig nation();
        break;
    default:
        System.out.println("Faites le bon choix : ");
        break;
    } break;
case 6:
    System.out.println("AU REVOIR!");
    //System.exit(0);
    break OUTER;

case 7:
    System.out.println("ERREUR DE SAISIE!");
    //System.exit(0);
    break OUTER;

default:
    System.out.println(" ");
    System.out.println("---veuillez faire votre choix---");
    System.out.println("1 : pour tache ");
    System.out.println("2 : pour Membre");
    System.out.print("choix : ");
    int choix = sc.nextInt();
    if (choix1==1 && choix==1){
        tach.connectDB();
        tach.afficherTacheDB();
        tach.ajouter tache();
        tach.afficherTacheDB();

    }
    else if( choix1==1 && choix==2){
        mbre.connectDB();
        mbre.ajouter();
        mbre.afficherDB();
        tach.afficherTacheDB();

    }
    else if (choix1==2 && choix==1){
        tach.connectDB();
        tach.modifier();

```

```

        mbre.selectionner();
        mbre.afficherDB();
        tach.afficherTacheDB();

    }
    else if (choix1==2 && choix==2){
        mbre.connectDB();
        mbre.modifier();
        mbre.afficherDB();
        tach.afficherTacheDB();

    }
    else if (choix1==3 && choix==1){
        tach.connectDB();
        tach.supprimer();
        tach.selectionner();
        mbre.afficherDB();
        tach.afficherTacheDB();

    }
    else if (choix1==3 && choix==2){

        mbre.connectDB();
        mbre.supprimer();
        mbre.afficherDB();
        tach.afficherTacheDB();

    }
    else if (choix1==4 && choix==1){

        tach.connectDB();
        tach.selectionner();
        mbre.afficherDB();
        //tach.afficherTacheDB();

    }
    else if (choix1==4 && choix==2){
        tach.connectDB();
        mbre.connectDB();
        //mbre.afficherDB();
        tach.afficherTacheDB();

    } break;
    }
}
}
}

```

Menu.java

```
package tp1;

public class Menu {

    public void menu(){

        System.out.println("
        #####");
        System.out.println("##BIENVENU SUR LE PROGRAMME DE GESTION
        DES TACHES##");
        System.out.println("
        #####");
        System.out.println("##---veillez faire votre choix--- ##");
        System.out.println("## ##");
        System.out.println("## 1 : AJOUTER ##");
        System.out.println("## 2 : MODIFIER ##");
        System.out.println("## 3 : SUPPRIMER ##");
        System.out.println("## 4 : AFFICHER ##");
        System.out.println("## 5 : ASSIGNER ##");
        System.out.println("## 6 : SORTIR DU PROGRAMME ##");
        System.out.println("## ##");
        System.out.println("##Faites votre choix : ##");
        System.out.println("
        #####");
        System.out.println("#####");
    }

}
```

Membre.java

```
package tp1;

import java.sql.*;
import java.util.Scanner;

public class Membre {

    private int idMembre;
    private String nomMembre;
```

```

private String sql;

String url="jdbc:mysql://localhost/tp1gl";
String user="root";
String password="";
Statement st = null;
ResultSet rs = null;
Connection cn = null;

Scanner sc = new Scanner(System.in);
public int getIdMembre() {
    return idMembre;
}
public void setId(int id, int idMembre) {
    this.idMembre = idMembre;
}
public String getNom() {
    return nomMembre;
}
public void setNom(String nom) {
    this.nomMembre = nom;
}

public void connectDB(){
    //s System.out.println("driver OKi");
    try{

        Class.forName("com.mysql.jdbc.Driver");

        cn = DriverManager.getConnection(url,user,
            password);
        st = cn.createStatement();
        System.out.println("connexion etablie");
    }catch (Exception e){
        System.out.println("delai depasser");
    }
}

public void modifier(){
    System.out.print("Entre le nom du membre a modifier: ");
    nomMembre=sc.nextLine();
    System.out.print("Entre le nouveau nom du membre modifier: ");
    String newNamMembre=sc.nextLine();
    try{
        sql = "UPDATE membre SET nomMembre = '"+newNamMembre +"'
            WHERE nomMembre ='"+nomMembre+"'";
        st.executeUpdate(sql);
        System.out.println("modification effectue avec succes");
    }catch(Exception e){

```

```

        System.out.println("echec de modification");
    }finally{
        try{
            cn.close();
            st.close();
        }catch (Exception e){
            System.out.println("deconnexion echoue");
        }
    }
}

public void supprimer(){
    System.out.println("");
    System.out.print("Entrer l'identifiant du membre supprimer : ");
    idMembre=sc.nextInt();
    try{
        sql ="DELETE FROM membre WHERE membre.idMembre='"+
            idMembre+"'";
        st.executeUpdate(sql);
        System.out.print("membre supprime");
    }
    catch(Exception e){
        System.out.print("echec de suppression de ce membre");
    }
}

public boolean ajouter(){

    System.out.print("Entrer le nom du membre : ");
    nomMembre=sc.nextLine();
    try{
        sql = "INSERT INTO Membre(idMembre,nomMembre)
            VALUES ('"+ idMembre + "','"+nomMembre+"')";
        st.executeUpdate(sql);
        System.out.println("ajout reussi");
    }catch (SQLException e){
        System.out.println("error d'ajout de membre");
    }finally{
        try{
            cn.close();
            st.close();
            System.out.println("deconnexion reussie");
        }catch (SQLException e){
            System.out.println("deconnexion echoue");
        }
    }

    return true;
}

```



```

public void selectionner(){

    System.out.println("");
    System.out.println("Recherche de toutes les taches assignees un membre
        : ");
    System.out.print("Saisir l'id du membre : ");
    idMembre=sc.nextInt();
    System.out.println("Resultat de la recherche : ");
    try{
        sql= "SELECT nomTache, nomMembre FROM tache, membre,
            assignation WHERE assignation.idMembre = membre.idMembre
            AND assignation.idtache = tache.idTache AND assignation.
            idMembre='"+idMembre+"'";
        rs= st.executeQuery(sql);
        System.out.println("|TACHES|\t\t|MEMBRE|");
        while (rs.next()){
            System.out.println(rs.getString("tache.nomTache"
                )+"\t\t"+ rs.getString("membre.nomMembre"));
            //System.out.print(rs.getString("membre.nommembre")+ "\t");
            //System.out.print(rs.getString("gestiontachetp1.tache.nomtache ")
                + "\t");
            //System.out.println("");
        }

        System.out.println("####FIN####");
    }catch (Exception e){
        System.out.println("taches assignees introuvables");
    }finally{
        try{
            cn.close();
            st.close();
            System.out.println("deconnexion reussie");
        }catch (Exception e){
            System.out.println("deconnexion echoue");
        }
    }

}

}

public void afficherDB(){

    System.out.println("");
    System.out.println("BASE DE DONNEE : ");
    System.out.println(" ");
    connectDB();
    try{

        sql= "SELECT * FROM Membre";
        System.out.println(" ");
    }
}

```

```

        rs= st.executeQuery(sql);
        System.out.println(" ");
        System.out.println("|idMembre|\t\t|MEMBRE|");
        while (rs.next()){

            System.out.println(rs.getInt("idMembre")+"\t\t"+
                rs.getString("nomMembre"));

        }

        System.out.println(" ");
    }catch (SQLException e){
        System.out.println("echec");
    }
}

}

```

ConnexionDB.java

```

package tp1;

import java.sql.*;

import java.util.Scanner;

public class ConnexionDB {
    String url="jdbc:mysql://localhost/tp1gl";
    String user="root";
    String password="";
    String sql;
    Connection cnx = null;
    Statement st = null;
    ResultSet rs = null;
    ResultSet rsTache = null;
    ResultSet rsMembre = null;

    Scanner sc = new Scanner(System.in);

    public void connectDB(){

        try{

```

```

        Class.forName("com.mysql.jdbc.Driver");
        System.out.println("driver OK");

        cnx = DriverManager.getConnection(url,user,
            password);
        st = cnx.createStatement();
        System.out.println("connexion reussi");
    }catch (ClassNotFoundException | SQLException e){
        System.out.println("erreur connexion");
    }
}

```

```

public void AfficherM(){
    System.out.println("");
    System.out.println("####AFFICHAGE DE LA BASE DE DONNEES####");
    System.out.println("");

    try{
        sql= "SELECT * FROM Membre;";
        //st = cnx.createStatement();
        //rs= st.executeQuery(sql);
        System.out.println("MEMBRES : ");
        System.out.println("|IDENTIFIANT DES MEMBRES|\t|
            NOM DES MEMBRE|");
        while (rs.next()){
            System.out.print("\t");
            System.out.print(rs.getInt("idmembre")+ "\t")
                ;
            System.out.print("\t\t\t");
            System.out.print(rs.getString("
                nomMembre")+ "\t");
            System.out.println("");
        }

    }
    catch (SQLException e){
        System.out.println("erreur de l'affichage");
    }
}

```

```

public void afficherT(){
    System.out.println("");
    try{
        sql= "SELECT * FROM tache ";
        rs= st.executeQuery(sql);
        System.out.println("LES TACHES : ");
        System.out.println("|ID TACHES|\t|NOM DES TACHES|\t|
            DESCRIPTION|\t\t\t\t\t| STATUS");
        while (rs.next()){

```

```

        System.out.print("\t");
        System.out.print(rs.getInt("idTache")+"\t");
        System.out.print(rs.getString("nomTache")+"\t\t\t");
        ;
        System.out.print(rs.getString("descripTache")+"\t\t\t\t\t\t");
        System.out.print(rs.getString("statusTache")+"\t\t");
        ;
        System.out.println(" ");
    }

}

}
catch (SQLException e){
    System.out.println("erreur de l'affichage");
}
}

}

```

Assignation.java

```

package tp1;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class Assignation {
    String url="jdbc:mysql://localhost/tp1gl";
    String user="root";
    String password="";
    Connection cn = null;
    Statement st = null;
    ResultSet rs = null;
    int idassign;
    int idTache;
    int idMembre;
    String sql;
    String sqlTache;
    String sqlMembre;
    Scanner sc = new Scanner(System.in);
}

```

```

public void connectDB(){

    try{

        Class.forName("com.mysql.jdbc.Driver");
        System.out.println("driver OK");

        cn = DriverManager.getConnection(url,user,
            password);
        st = cn.createStatement();
        System.out.println("connexion effectuer");
    }catch (ClassNotFoundException | SQLException e){
        System.out.println(" Erreur de la connexion");
    }

}

public void afficherTachAssign(){
    System.out.println("");
    System.out.println("*****LES TACHES ASSIGNEES A CHAQUE MEMBRE
        *****");
    try{
        sql= "SELECT tache.nomTache , membre.nomMembre FROM tp1gl.
            tache, tp1gl.membre, tp1gl.assignment WHERE membre.idMembre
            = assignment.idMembre AND tache.idTache = assignment.idTache "
            ;
        rs= st.executeQuery(sql);
        System.out.println("|Les Taches|\t\t|Les Membres|");
        while (rs.next()){
            System.out.println(rs.getString("nomTache")+"\t\t"
                "+ rs.getString("nomMembre"));

        }
        System.out.println("");
        System.out.println("#####FIN DE LA RECHERCHE#####");
    }catch (SQLException e){
        System.out.println("taches assignes introuvables");
    }

}

public void afficherAssign(){
    System.out.println("");
    System.out.println("###Tache assigner a membre###");

    try{
        sql= "SELECT assignment.idtache, assignment.idmembre FROM
            gestiontachetp1.assignment ";
        rs= st.executeQuery(sql);
        System.out.println("|TACHES|\t|MEMBRES|");
    }
}

```

```

        while (rs.next()){
            System.out.println(rs.getInt("assignation.idtache")
                               +"\t\t"+ rs.getInt("assignation.idmembre"));

        }
        System.out.println("");

    }
    catch (SQLException e){
        System.out.println("taches assignes introuvables");
    }
}

public void modification(){

    System.out.print("Entrer l'identification du membre modifier: ");
    idMembre = sc.nextInt();

    System.out.print("Entrer le nouvel identification correspondant du membre
        modifier: ");
    int newIdMembre = sc.nextInt();
    try{
        sql = "UPDATE assignation SET idmembre = '"+newIdMembre+"'
            WHERE nom = '"+idMembre+"'";
        st.executeUpdate(sql);
        System.out.println("modification d'assignation effectue avec
            succes");

    }catch(SQLException e){
        System.out.println("echec de modification d'assignation");
    }finally{
        try{
            cn.close();
            st.close();
        }catch (SQLException e){
            System.out.println("deconnexion echoue");
        }
    }
}

}

public void assignation(){

    System.out.print("ENTRER IDENTIFIANT DU MEMBRE : ");
    idMembre = sc.nextInt();
    System.out.print("ENTRER IDENTIFIANT DU TACHE : ");
    idTache = sc.nextInt();
    try{
        sql = "INSERT INTO tp1gl.assignation VALUES ('"+
            idassign + "','"+idMembre+"','"+idTache+"
            ')";
    }
}

```

```

        st.executeUpdate(sql);
        System.out.println("Assignment reussie");
    }catch (SQLException e){
        System.out.println("Echec d'assignation");
    }finally{
        try{
            cn.close();
            st.close();
            System.out.println("deconnexion
                reussie");
        }catch (SQLException e){
            System.out.println("deconnexion
                echoue");
        }
    }
}
}
}

```

Tache.java

```

package tp1;

import java.sql.*;

import java.util.Scanner;

public class Tache {
    private int idTache;
    private String nomTache;
    private String descripTache;
    private String statusTache;
    private int idMembre;
    String sql;

    String url="jdbc:mysql://localhost/tp1gl";
    String user="root";
    String password="";
    Connection cn = null;
    Statement st = null;
    ResultSet rs = null;
    Scanner sc = new Scanner(System.in);

    public Tache( String statusTache, String descripTache, String

```



```

nomTache){
    super();
    this.nomTache = nomTache;
    this.descripTache = descripTache;
    this.statusTache = statusTache;
}

public void connectDB(){

    try{

        Class.forName("com.mysql.jdbc.Driver");
        System.out.println("driver OK");

        cn = DriverManager.getConnection(url,user,
            password);
        st = cn.createStatement();
        System.out.println("connexion effectuer");
    }catch (ClassNotFoundException | SQLException e){
        System.out.println(" Erreur de la connexion");
    }

}

public Tache() {
    super();
    // TODO Auto-generated constructor stub
}

public int getIdTache() {
    return idTache;
}
public void setIdTache(int idTach) {
    idTache = idTach;
}
public String getNom() {
    return nomTache;
}
public void setNom(String nomTach) {
    this.nomTache = nomTach;
}
public String getDescription() {
    return descripTache;
}
public void setDescription(String descript) {
    this.descripTache = descript;
}

```

```

public String getStatus() {
    return statusTache;
}
public void setStatut(String status) {
    this.statusTache = status;
}

public String getNomTache() {
    return nomTache;
}

public void setNomTache(String nomTache) {
    this.nomTache = nomTache;
}

public int getIdmembre() {
    return idMembre;
}

public void setIdmembre(int idMembre) {
    this.idMembre = idMembre;
}

public void modifier(){
    System.out.println("old Tache");
    String oldNamTache=sc.nextLine();
    System.out.println("new Tache");
    String newNamTache = sc.nextLine();
    System.out.println("old description");
    String oldDescript = sc.nextLine();
    System.out.println("new description");
    String newDescript = sc.nextLine();
    System.out.println("old status");
    String oldStatus = sc.nextLine();
    System.out.println("new Tache");
    String newStatus=sc.nextLine();
    try{

        sql = " UPDATE tache SET nomTache='"+newNamTache+"',
            descriptTache='"+newDescript+"',statusTache='"+newStatus
            +" WHERE nomTache='"+oldNamTache+"'AND descriptTache='
            "+oldDescript+"'AND statusTache='"+oldStatus+"'";
        st.executeUpdate(sql);
        System.out.println("status modifier");
    }catch (SQLException e){
        System.out.println("modification echoue !");
    }finally{
        try{

```

```

        cn.close();
        st.close();
        System.out.println("deconnexion");
    }catch (SQLException e){
        System.out.println("erreur deconnexion");
    }
}

}

public void supprimer(){
    System.out.println("");
    System.out.println("id de Tache a supprimer : ");
    idTache=sc.nextInt();
    try{
        sql ="DELETE FROM tache WHERE idTache='"+nomTache+"'";
        st.executeUpdate(sql);
        System.out.println("Suppression reussi");
    }
    catch(SQLException e){
        System.out.println("echec de suppression ");
    }
}

```

```

public void ajoutertache(){

    System.out.print("Entrer le nom de la tache : ");
    nomTache=sc.nextLine();
    System.out.print("Entrer la description de la tache : ");
    descripTache=sc.nextLine();
    System.out.print("Entrer le status de la tache : ");
    statusTache=sc.nextLine();
    //connectDB();
    try{
        sql = "INSERT INTO tache VALUES ('"+idTache+"','"+
            nomTache+"','"+descripTache+"','"+
            statusTache+"')";
        st.executeUpdate(sql);
        System.out.println("ajout reussi");
    }catch (SQLException e){
        System.out.println("error d'ajout de tache");
    }finally{
        try{
            cn.close();

```

```

        st.close();
        System.out.println("deconnexion reussie");
    }catch (SQLException e){
        System.out.println("deconnexion echoue");
    }
}
}

```

```

public void selectionner(){
    System.out.println("");
    System.out.println("RECHERCHE DE TACHE SUIVANT STATUS");
    System.out.println("Veuillez entrer le status des taches : ");
    statusTache=sc.nextLine();
    System.out.println("Resultat de la recherche : ");
    try{

        sql= "SELECT nomTache ,statusTache FROM tache WHERE
            statusTache ='"+statusTache+"' ";

        rs= st.executeQuery(sql);
        System.out.println("|TACHES|\t\t|STATUS|");
        while (rs.next()){
            System.out.println(rs.getString("nomTache")+"\t\t"
                + rs.getString("statustache"));
        }
        System.out.println("");

        catch (SQLException e){
            System.out.println("taches not found");
        }finally{
            try{

                cn.close();
                st.close();
                System.out.println("reussite");
            }catch (SQLException e){
                System.out.println("echec deconnexion");
            }
        }
    }
}

```

```

public void afficherTacheDB(){

```

```

System.out.println("");
System.out.println("BASE DE DONNEE : ");
System.out.println(" ");
connectDB();
try{

    sql= "SELECT * FROM tache";
    System.out.println(" ");
    rs= st.executeQuery(sql);
    System.out.println(" ");
    System.out.println("|idTache|\t|nomTache|\t|statusTache|\t|
        descripTache|");
    while (rs.next()){

        System.out.println(rs.getInt("idTache")+"\t\t"+ rs.
            getString("nomTache")+"\t\t"+ rs.getString("
            descripTache")+"\t\t"+ rs.getString("statusTache"));

    }

    System.out.println(" ");
}catch (SQLException e){
    System.out.println("echec");
}

}

}

```