

Projet Personnel Encadré (PPE 3&4) Application Backoffice SuperCar

BTS SIO – 2eme Année

Prof: Dr Didier Samfat

- Les **Projets Personnalisés Encadrés (PPE)** ont pour but de mettre en œuvre les savoirs et savoirs faire abordés dans les différents cours pour acquérir des compétences techniques globales mais aussi spécifiques à votre spécialisation SLAM.

Au niveau de ***l'examen***, les PPE ont une grande importance :

- Ils doivent vous permettre de renseigner votre **portefeuille de compétences**, au même titre que vos expériences en stage, en vue de l'épreuve E6 "***Parcours de professionnalisation***".
- Les PPE peuvent aussi être utilisés pour la préparation de l'épreuve pratique E4 "***Conception et maintenance de solutions informatiques***".
- Le PPE présenté ici s'inspire d'un contexte d'entreprise basé sur la réalité Mauricienne.
- En effet, l'équipe pédagogique en consultation avec les prestataires informatiques et concessionnaires de voitures, a conçu le contexte **SuperCar** qui est l'organisation cliente
- **MultiSys** est la Société de Services en Ingénierie Informatique (SSII), prestataire informatique chargée d'implémenter le Cahier des Charges de **SuperCar**.

Ce PPE a pour but de vous plonger dans un contexte professionnel, comme dans le cadre d'un stage:

- Un début d'application similaire en partie créée est alors présentée
- Il s'agira de modifier et d'enrichir le code pour répondre au **cahier des charges du client**
- Vous devez aussi faire de la **veille technologique** en matière de solution réseau et Progiciel de Gestion (ERP) en Open source et aussi sur les méthodes de développement **Java/MySQL**
- À partir de cet existant, plusieurs **Missions Professionnelles** vous sont confiées.
- Vous pouvez traiter ces missions en autonomie ou en mode projet avec un autre étudiant. Cependant, chaque étudiant devra réaliser ses propres travaux et **code sources**.
- Puisque vous partez d'un existant, des fichiers exemples en Java vous seront fournis.
- Il vous faudra alors comprendre le code, l'exécuter et **l'adapter** et le faire **évoluer** au contexte SuperCar
- Vous allez être pris en charge par le professeur qui va vous suivre exclusivement sur le travail en PPE et à la préparation de l'épreuve E4.

La **Partie 1** présente de façon détaillée le contexte de l'entreprise.

- C'est le genre d'information qui peut vous être communiqué lors d'un premier jour de stage.
- Vous devez commencer par lire attentivement cette partie et ne pas hésiter à prendre des notes, comme vous le feriez en stage, pour mémoriser les points essentiels qui pourront vous servir.
- En effet, le contexte est global et il ne faut pas se limiter à présenter uniquement les aspects qui peuvent vous servir.
- En revanche, vous devez en faire une lecture attentive et comprendre son contenu : si vous présentez ce contexte lors de l'épreuve E4, le jury peut très bien vous poser des questions sur le contexte.

La **Partie 2** présente du code source Java-MySQL d'une application gérant des étudiants dans une université.

- Votre but sera d'adapter ce code et de l'enrichir dans le cadre de SuperCar
- Utilisez UML pour modéliser les **cas d'utilisation** et les **classes** nécessaires au projet SuperCar
- Prenez le temps de bien étudier les différents aspects présentés avant de vous lancer dans les missions
- Prenez le soin de bien analyser les exemples ainsi que le cahier des charges de SuperCar
- Vous utiliserez aussi Windows Builder et **le package swing** vu en cours pour la partie graphique

La Partie 3 présente les *Missions Professionnelles* à réaliser.

- Dans un premier temps, la configuration nécessaire est présentée.
- Ensuite, chaque mission est expliquée, un peu comme elle le serait dans le cadre d'un stage.
- A vous ensuite de planifier et gérer votre travail sur les missions confiées.

La Partie 4 présente les tutoriels

- Ils vous aideront à installer les outils nécessaires à la réalisation des missions
- 2 exemples d'application sont donnés notamment pour l'utilisation du connecteur JDBC dans un environnement Java-MySQL
- Il est impératif de maîtriser cette partie pour réussir votre projet

Partie 1 :

Présentation du contexte

- Dans cette partie, le contexte de l'entreprise est présenté de façon détaillée, (comme un premier jour de stage).
- Vous devez en faire une lecture attentive.
- Tous les aspects ne vous seront pas forcément utiles, cependant il est important que vous ayez cette vision globale du contexte professionnel.
- N'hésitez pas à prendre des notes et poser des questions pour récupérer et comprendre les informations essentielles

- **MultiSys** est une SSI créée en 2005, dont l'expertise réside dans le développement de logiciels Backoffice, l'intégration de **Progiciel de Gestion** Open Source et des sites web entre autre
- Ils assurent aussi la ***maintenance évolutive*** des applications existantes et éditent de nouvelles versions logicielles en fonction du besoin de leurs clients.
- MultiSys est spécialisée dans les langages de programmation tels que **JAVA**, Python, C++, PHP, JavaScript et autres technologies comme Ajax, jQuery, Bootstrap.
- Au vu du contexte Mauricien elle propose MySQL comme base de donnée qui ne requiert aucune license commerciale leur conférant ainsi un avantage compétitif indéniable
- Ils ont aussi les compétences pour migrer leur développement vers la base de données **Oracle** si besoin est.
- Récemment ils ont recrutés de nouveaux **Experts** en Java capable de porter les applications multiplateformes notamment les mobiles **Android**
- Le Chef de projet (Prof) a déjà fait une analyse préalable et a nommé un Chef de Projet Junior (rôle à permuter entre étudiants) pour attribuer les tâches spécifiques à chaque membre de l'équipe en fonction du **Cahier Des Charges (CDC)** général.

- **SuperCar** existe depuis 2008. C'est un concessionnaire spécialisé dans la vente de voitures neuves de marques diverses.
- SuperCar importe des voitures de plusieurs pays: Japon, Singapore, Afrique du Sud, des Etats-Unis, Chine, d'Allemagne, Inde, France et Corée
- Une même marque peut provenir d'un ou plusieurs pays différents
- Les voitures sont stockées dans **4 entrepôts sous douane** dans les villes suivantes:
 - Port Louis
 - Baie du Tombeau
 - Phoenix
 - Plaisance
- Le siège social se trouve à **Pailles** intégrant les départements suivants:
 - Administration,
 - Marketing Ventes
 - Comptabilité et Finance,
 - Ressources Humaines
 - Garage & Réparation
- L'application actuelle utilisé par SuperCar est en **Cobol** et l'ancien prestataire informatique a mis la clé sous la porte !
- SuperCar souhaite:
 - Revoir son réseau (LAN & WAN)
 - Informatiser ses activités en intégrant de nouvelles procédures de travail
- Elle fait alors appel
 - A l'entreprise MultiSys SSII
 - Pour trouver les solutions adéquates au meilleur rapport qualité / prix
- Le Chef de Projet de MultiSys prend rendez-vous pour rencontrer le PDG **M. Maurice Beff** pour en savoir plus

Combien d'employés avez-vous ?

- Nous sommes 250 collègues

Comment sont structurés vos opérations?

- Nous avons un showroom au siège social où nous accueillons tous nos clients
- Les voitures sont stockées dans 4 entrepôts sous douane répartis à travers l'Ile

Quelles sont les nouvelles fonctionnalités que vous souhaitez implémenter ?

- Les clients viennent voir nos voitures au showroom et ont la possibilité de faire un test de conduite
- Par la suite le vendeur lui fait un devis en fonction de son choix (ex couleur et options)
- Après paiement on lui livre la voiture qui sort directement de l'entrepôt à son domicile
- Le vendeur reçoit alors une commission de Rs 3,500
- Nous souhaitons aussi avoir un système très sécurisé où chaque intervenant a accès qu'aux informations qui les concernent

Avez-vous un budget pour ce projet ?

- Oui, le Conseil d'Administration a approuvé un **budget de Rs5M**

Quels sont les problèmes avez-vous avec le système actuel ?

- Notre programme est vieux et n'est plus maintenu. Il tourne sur un seul PC dédié utilisé par 3 personnes pour la saisie des données
- Par ailleurs les entrepôts envoient un **fax** et la gestion du stock se fait manuellement ici par l'Administration (double saisie)
- Nous souhaitons donc avoir un **Progiciel de Gestion Intégré** (ERP) en réseau permettant aux entrepôts de mettre à jour directement le stock lorsqu'une voiture sort sous douane.
- Les vendeurs doivent aussi pouvoir se connecter au système et connaître l'état du stock à tout moment pour mieux guider le client

Qu'en est-il des autres départements ?

- **L'Administration** doit pouvoir consulter le stock pour commander des voitures si besoin
- Le **Comptable** doit connaître le montant de la commande pour paiement aux fournisseurs et aussi un rapport de vente journalier, hebdomadaire et mensuel.
- Les **Vendeurs** doivent savoir s'il y a des commandes de nouvelles voitures en cours
- Les **Ressources humaines** doivent pouvoir gérer les données personnelles des employés ainsi que la paye.

Quand souhaitez-vous être opérationnel sur le nouveau système ?

- **Il est impératif que « tout marche bien » dans 6 mois soit le 30 Avril 2021 au plus tard**

Le Directeur informatique de SuperCar donne la liste du matériel existant et son besoin futur

Le Siège Social est composé de 2 étages :

- Rez-de-chaussée (Showroom & Garage)
 - 30 PC, 20 Tablettes, 5 bornes Wifi Cisco-Meraki MR33, 5 imprimantes
 - 20 Téléphone IP et 8 Caméra IP
- 1^{er} étage (Admin, RH et Finance)
 - 50 PC, 20 x Laptop, 3 bornes Wifi Cisco-Meraki MR33, 5 imprimantes
 - 20 Téléphone IP et 10 caméra IP
- La salle informatique se trouve au 1^{er} étage
 - 1 x Core Switch Cisco Meraki MS250P - 48P
 - 2 x Access Switch Cisco Meraki MS390P - 48P
 - Routeur fibre optique Huawei fourni par MT
 - Firewall Layer 7 Cisco Meraki MX100
 - 1 Serveur pour l'application Cobol
- Dans le cadre de ce nouveau projet, on utilisera les interfaces Java Clients dites « lourd » sur les PC et laptop seulement
- Il est prévu ultérieurement de faire tourner l'app sur de SmartPhone et Tablette Android
- La « **Deadline** » pour le projet est **le 30 Avril 2021**

Actuellement dans les Entrepôts sous douane

- 1 PC, 1 Fax, 1 imprimante
- 25 employés par site

SuperCar veut :

- Un nouveau Progiciel de Gestion Intégré
- Un nouveau serveur en haute disponibilité pour héberger la solution complète
- Rajouter 15 PC au 1^{er} étage
- Rajouter 5 PC au Rez-de chaussé
- Rajouter 8 PC et 2 imprimantes par entrepôt
- Avoir un **WAN hautement sécurisé** entre les entrepôts et le siège social

Travail à faire:

- Redéfinir l'architecture global du réseau pour prendre en comptes ces nouveaux besoins et interconnecter tous les 4 entrepôts au siège social
- Soumettre un diagramme complet de réseau pour implémenter le nouveau système de gestion incluant les nouveaux
- Evaluer le coût total du nouvel hardware nécessaire et justifier chaque choix

Le personnel utilisant le système devra pouvoir :

- Accéder au système par un mécanisme d'authentification,
- Accéder à l'interface correspondant pour s'authentifier par login et mot de passe
- L'application doit permettre la **gestion des droits d'accès** (Autorisation, Privilège) en fonction du profil de l'utilisateur.

Toutes les données doivent être stockées sur un SGBD présent sur un serveur

- Elles doivent être protégé grâce à un accès sécurisé à Internet
- Elles doivent aussi être sauvegardées régulièrement (Backup). Il faudra décrire les procédures correspondantes qui seront implémentés
- Le SGBD doit être accessible depuis les postes distants (entrepôts)
 - Vous devez dimensionner et définir les spécifications du serveur à acheter
 - On utilisera des applicatifs Clients dits « **lourds** » par opposition au clients **légers** « Navigateur web »
 - Vous programmerez en Java Objet en utilisant les connecteurs correspondant pour MySQL

SuperCar fait appel à vous pour trouver et implémenter un **Progiciel de Gestion Intégré** (ERP).

- Il s'agit d'informatiser tous les processus de l'entreprise.
- Vous chercherez plusieurs solutions candidates en **Open Source Java** la plus adaptée pour la partie Finance RH, Relation Client (CRM) et ressources humaines
- Faites une **analyse** et **justifier** votre choix final
- On supposera que cette partie sera implémentée par une autre équipe et **l'intégration** de votre développement avec cet ERP ne fait pas partie de ce projet
- Vous pouvez aussi installer l'ERP sélectionner pour comprendre son fonctionnement.

Vous devez cependant définir et implémenter les modules pour gérer:

- Les stocks d'achat de nouvelles voitures,
- Les ventes avec rapport journalier, hebdomadaire et mensuel
- Les primes de ventes de chaque personne concernée
- Processus d'achat des voitures
- La gestion des employés ainsi que les salaires

Les Managers des entrepôts doivent pouvoir mettre à jour le stock directement depuis leur poste de travail:

- Une fois qu'une livraison a été effectuée
- Ou lorsqu'une ou plusieurs voitures neuves arrivent dans leur entrepôt respectif

Votre équipe doit prendre en considération les scénarios évoqués précédemment

- Définir la nouvelle architecture réseau pour relier les entrepôts au showroom (lignes Internet, routeur, switch, VPN etc..).
 - On suppose que le Showroom a déjà un réseau local de 150 PC répartis sur 2 étages et une fibre optique installée par **Mauritius Telecom** pour accéder à l'Internet.
 - Il faudra réaliser le schéma global des interconnexions
- Mettre en place un Système de Gestion de Base de données,
- Définir une interface pour Managers des 4 entrepôts pour la mise à jour,
- Définir une interface pour que l'Administration puisse passer de nouvelles commandes et faire le suivi des achats et des ventes
- Définir une interface pour les Vendeurs pour conclure la vente.
 - Le système devra calculer le nombre de ventes et le chiffre d'affaire pour une période donnée
 - Nais aussi les commissions correspondantes pour chaque vendeur
- Définir une interface pour les Ressources Humaines pour la gestion des employés et des salaires

Partie 2 :

UML et Fichiers Exemples

Java

- Dans cette partie, Il faudra utiliser UML en utilisant les Diagrammes de Cas d'Utilisation et de Classe
- Les codes sources d'un exemple de projet Java / MySQL (gestion d'étudiants) se trouvent dans le répertoire: *java-db-exemple*
- L'objectif pour vous est de faire tourner cette application et ensuite de l'étudier et de l'adapter à au contexte SuperCar
- Vous utiliserez aussi le package swing de Java vu en cours pour la partie graphique

Modélisation par UML

Diagramme de *Cas d'Utilisation*

- Faites un diagramme complet décrivant tous les cas d'utilisation pour un utilisateur donné
- Pour rappel l'exemple du cours vous est redonné ci-dessous:

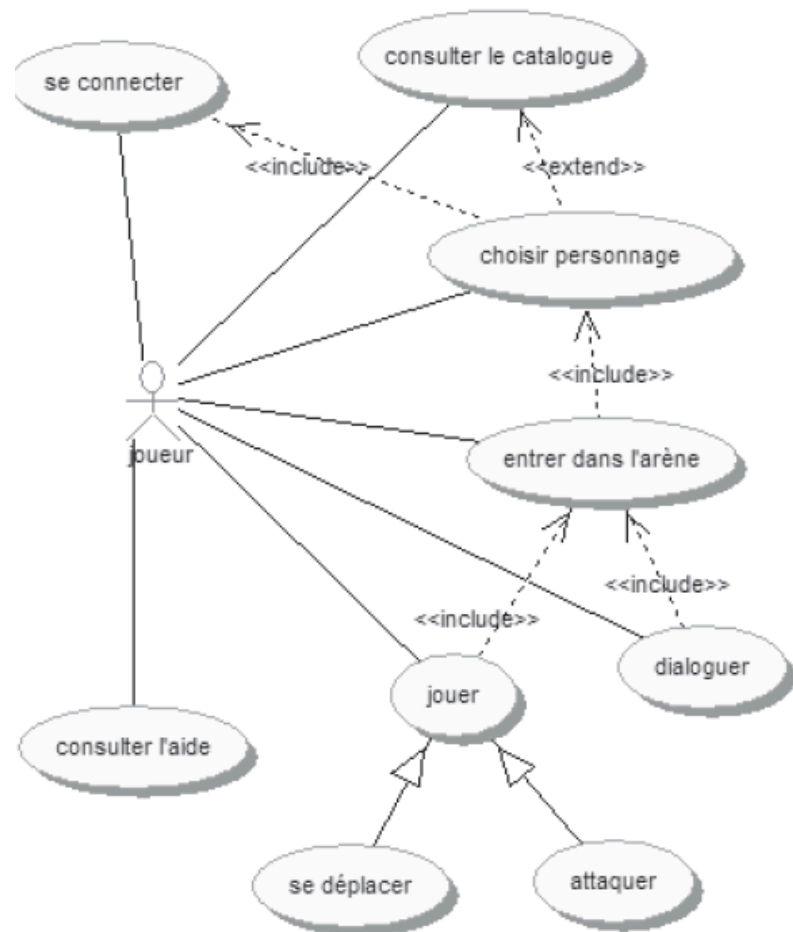
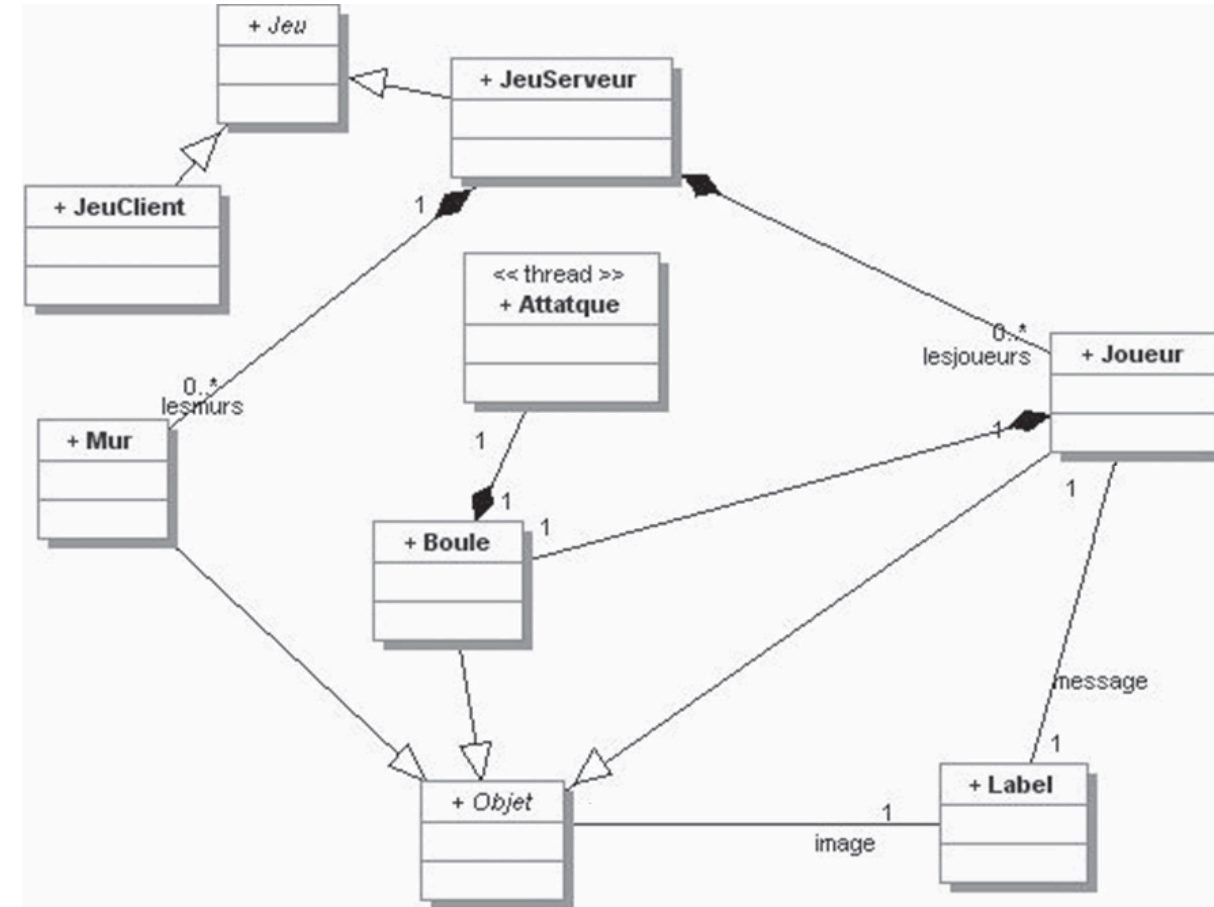


Diagramme de *Classes*

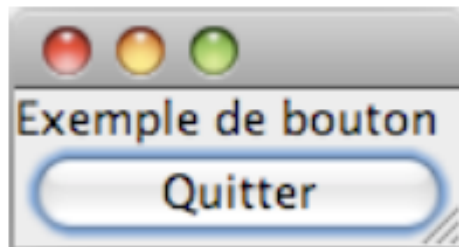
- Faites un diagramme complet décrivant toutes les classes dont vous aurez besoin pour SuperCar
- Pour rappel l'exemple du cours vous est redonné ci-dessous:



L'interface graphique

- La classe de container la plus simple est ***javax.swing.JPanel***
- Utiliser l'outil **WindowBuilder** (voir tutoriel en annexe) pour définir les composants qui constitueront la fenêtre de chaque interface graphique puis les rajouter suivant la disposition choisie.
- La disposition peut être spécifiée lors de l'appel au constructeur du panel:
`JPanel myPanel = new JPanel(new BorderLayout());`
- ou par un appel explicite à la méthode `setLayout`, par exemple :

```
myPanel.setLayout(new GridLayout(4,5));
```



Code java

Exemple: création d'une fenêtre en 4 étapes

```
import java.awt.*;  
import javax.swing.*;
```

1. Création d'une instance JFrame

```
public class ExBouton extends JFrame {
```

```
    public ExBouton() {  
        super();  
        JLabel label = new JLabel("Exemple de bouton");  
        JPanel panel = new JPanel(new BorderLayout());  
        panel.add(label, BorderLayout.NORTH);  
        JButton b = new JButton(" Quitter ");  
        panel.add(b, BorderLayout.SOUTH);  
        this.setContentPane(panel);  
        this.pack();
```

2. Ajout des composants au panel de la fenêtre

```
    }  
  
    public static void main(String[] args) {  
        ExBouton bouton = new ExBouton();  
        bouton.setVisible(true);
```

3. Calcul automatique de la taille de la fenêtre avec `pack()`

```
    }  
}
```

4. Affichage de la fenêtre par la méthode `setVisible()`.

- Un exemple vous est donné dans le répertoire compressé
 - *java-db-exemple.zip*
 - Ceci est un projet Eclipse avec MySql disponible sur GitHub a l'adresse
 - <https://github.com/samkeeleyong/java-db-example>
- Il faudra adapter les paramètres:
 - De connexion dans le fichier *dbconfig.properties*
 - Ensuite *cinfiguration*, pour exécuter le fichier `com.example.main.Main`
- Les étapes à suivre sont:
 1. Installer le projet
 2. Créer la base de donnée MySQL
 3. Modifier le fichier de propriétés de la base de données
 4. Exécuter le code
 5. Etudier le code
 6. Définir les Diagrammes de Class nécessaires pour SuperCar
 7. Adapter le code au contexte SuperCar

Le fichier ***Student.java*** décrit la classe étudiant

- Il faut s'en inspirer pour créer par exemples les classes:
 - Voitures
 - Employés
 - Ou autres qui vous seraient utiles

Le fichier ***DButil.java*** donne le code pour gérer la Base de Données Etudiant

- Il faut s'en inspirer pour créer les fonctions permettant de gérer
 - Voitures
 - Employés
 - Ou autres qui vous seraient utiles

Le fichier ***ConnectionFactory.java*** donne le code pour se connecter à une base de données

Le fichier ***QueryStatement.java*** donne le code pour faire des requêtes SQL

Le fichier ***main.java*** donne le code pour une interface utilisateur en mode texte

- Il faudra le modifier pour utiliser les méthodes de la classe swing pour réaliser l'interface graphique

Partie 3 :

Les Missions à réaliser

- Dans cette partie, nous définissons toutes les missions à réaliser en fonction des informations données précédemment
- Le résultat de chaque mission devra être décrit et figuré dans votre rapport final

MP1 - Etude du cahier de charges:

- Trouver l'ERP Open Source développé en Java qui répond au besoin (analyse et choix seulement)
- Le chef de projet Junior devra définir et répartir les tâches aux membres de l'équipe.

MP2 - Veille Technologique

- Bien articuler la démarche en structurant votre approche avec les modes **Push** ou **Pull** dans le cadre de ce projet.
- Ex: Java/MySQL et / ou tout **autre technologie** (ex hardware) dans le cadre de ce projet..

MP3 - Spécification du matériel & logiciel

- Vos préconisations de ce que le client doit acheter:
- Des nouvelles machines ou tout autre matériel nécessaires (incluant serveur web et serveur back-office)
- Eléments nouveaux de nœud de réseau pour les LAN et WAN (entre siège social les entrepôts)
- Justifiez chaque choix technique

MP4 - Topologie réseau et architecture client / serveur

- Faire un diagramme général du réseau complet de SuperCar
- Bien définir la topologie, l'emplacement de chaque matériel (surtout les serveurs)
- Décrire les installations étape par étape ainsi que la configuration

MP5 – Planifier de projet

- Utiliser le logiciel *ProjectLibre*
- Indiquer les coût de chaque ressources en journée-homme
- Inclure les rapport de ProjectLibre (Diagramme de Gant, Pert, l'équipe etc ..)
- Evaluer le coût du développement logiciel

MP6 - Coût total du projet

- Lister tous les besoins en matériel et logiciel
- Demander un devis à au moins 3 fournisseurs en fonctions des spécifications techniques
- Définir le CAPEX : dépenses d'investissement de capital (matériel et logiciel)
- Définir l'OPEX : dépenses d'exploitation annuelles (salaire, frais administratifs, service, maintenance, support et télécom)
- Définir le prix à facturer au client pour *le développement logiciel* et le *matériel* avec une marge de profit
- Calculer le cout de la maintenance annuel du logiciel pour Multisys

MP7 – Diagrammes UML

- Définir tous les Diagrammes de Cas d'Utilisation
- Définir tous les Diagrammes de Classes nécessaires

MP8 – Développement en Java

- Installer l'IDE Eclipse ainsi que les plug-in nécessaires
- Installer **WAMP** pour Windows ou **XAMP** pour MAC pour administrer les Bases de données avec *PhpMyAdmin*
- Installer les connecteurs JDBC
- Configurer le pare-feu pour le protocole TCP/IP pour des entrée et sorties et pour la connexion SQL entre serveur et client
- Configurer le serveur MySQL

MP9 – GIT

- Utiliser Egit (***Eclipse Git***) dans le cadre de votre projet
- Travailler en mode projet et gérer les différentes versions de votre code Java (***versioning***)
- Coordonner l'usage des différentes version entre membre de l'équipe

MP10 – Cybersécurité

- Définir les mécanismes de **sureté** & **sécurité** que vous allez mettre en place
- Détailler les moyens et services à mettre en œuvre au niveau matériel, réseau et logiciel.
- Bien documenter chaque approche ou méthode utilisée

MP11 – Réaliser un plan de E-formation

- En temps réel (**Moodle / BigBlueButton**) pour l' utilisateur final
- Mettre l'emphasis sur la pédagogie pour une prise en main rapide de votre logiciel

MP12 – Contrat de prestation de services

- Proposer une **contrat de développement** d'application incluant
- Coût de la version de développement et Coût du prix du support
- Maintenance annuelle
- Se rapprocher du Prof de droit si nécessaire

MP13 – La Documentation technique

- Faire un schéma de l'architecture des modules logiciel
- Rédiger un document qui permettrait à un nouveau collègue de comprendre ce qui a été fait

MP14 – LE RAPPORT FINAL

- Il devra décrire clairement et de manière concise toutes les missions abordés précédemment
- Inclure une documentation technique qui explique les différents modules et/ou code de votre programme
- L'objectif est de démontrer que vous avez bien maîtrisé le projet dans son ensemble ainsi que **les savoirs et savoirs-faire acquis ces 2 dernières années.**

Partie 4 :

Tutoriels d'installation

- Cette partie présente les tutoriels d'installation des outils nécessaires à la réalisation des missions et de la plateforme de développement
- Il s'agira dans un premier temps de tester un exemple MySQLTest1. Puis d'étudier le code avant de l'adapter au contexte SuperCar
- Les fichiers sources se trouvent dans le répertoire : **Exemple-MySQL-JDBC**
- On pourra utiliser l'outil WindowBuilder d'Eclipse pour la partie interface graphique

L'IDE Eclipse

Utilisation de l' IDE (Integrated Development Environment)

- Installation de l'environnement Java Entreprise

<https://www.java.com>

Télécharger la version Java JDK pour votre ordinateur (Mac ou Windows)

Lancer l'installation

- Installation de l'Editeur Eclipse

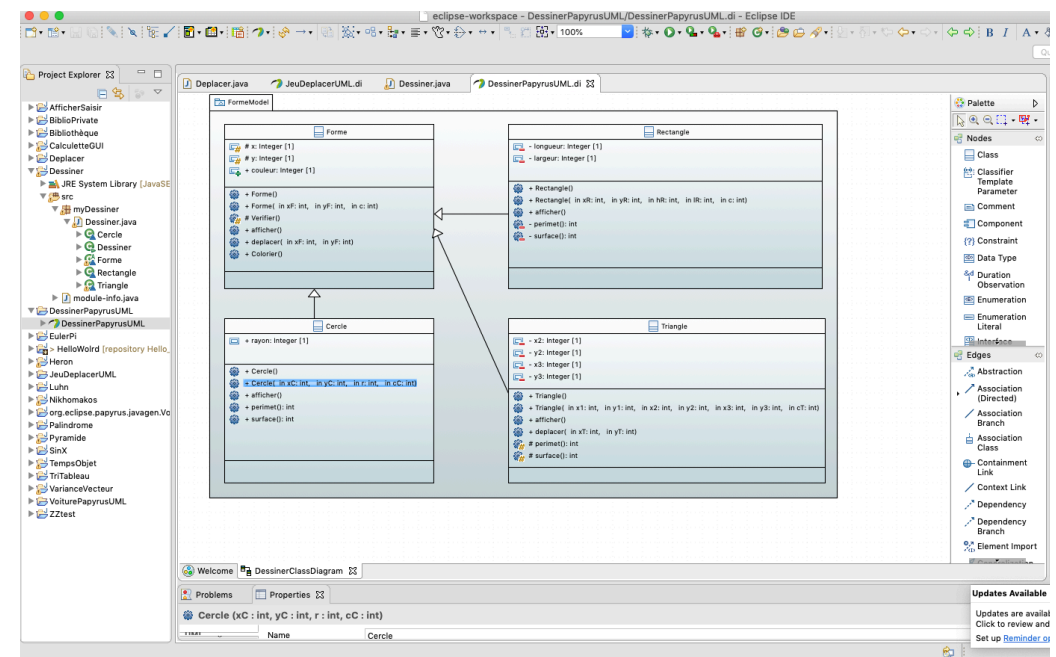
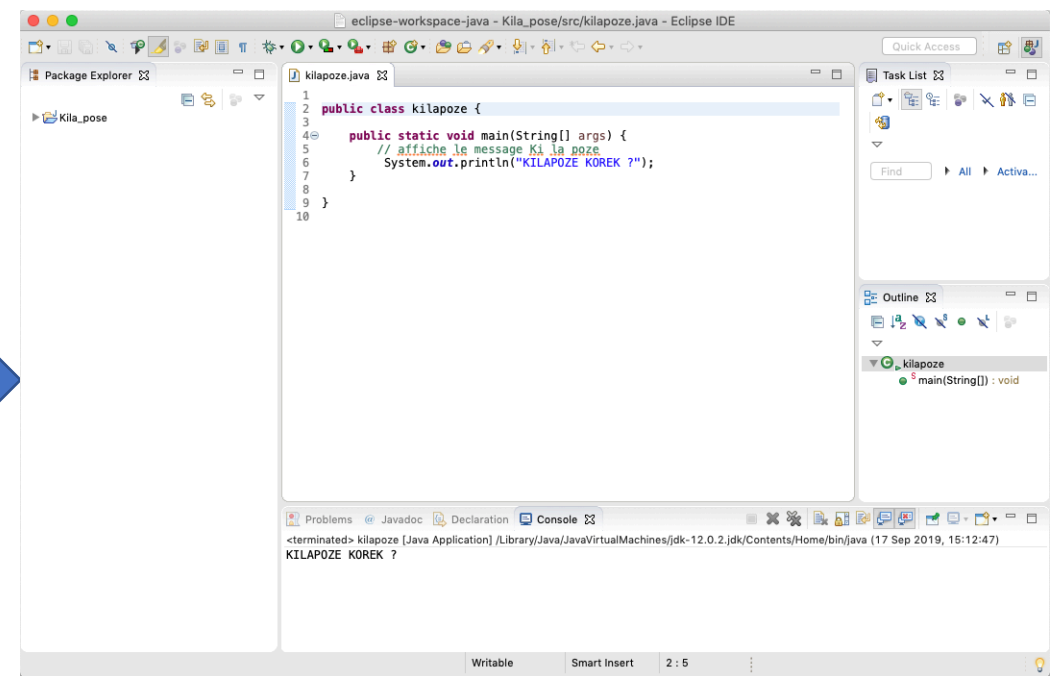
www.eclipse.org (pour java workspace)

Télécharger la version pour votre ordinateur (Mac ou Windows)

Lancer l'installation

Installer aussi les plug-ins Eclipse:

- **Egit** pour le versionning avec Github et
- **Papyrus** pour les diagrammes UML



Installation de la plate-forme

XAMPP est une distribution Apache entièrement gratuite et facile à installer contenant MySQL, PHP et Perl. Le packaging open source XAMPP a été mis au point pour être facile à installer et à utiliser:

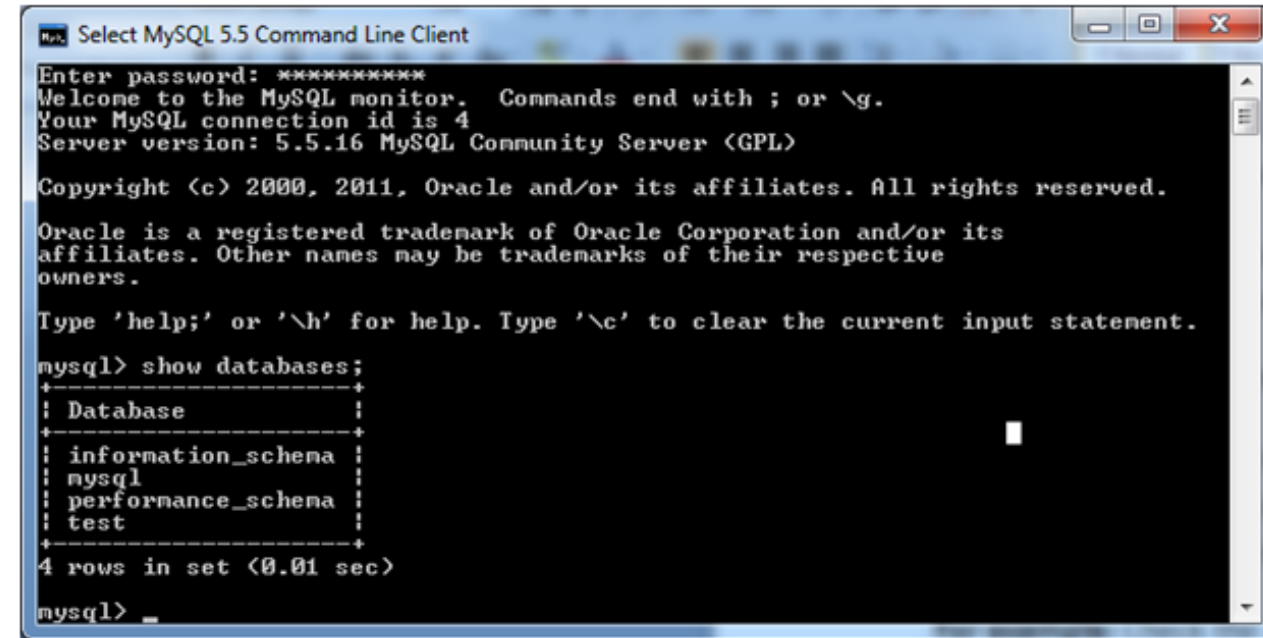
- Télécharger la version Windows ou Mac sur
- <https://www.apachefriends.org>

Dans les 2 cas:

- On installe les package Apache server, PhpMyAdmin, MySQL
- On utilisera principalement PHPmyAdmin pour administrer les BD MySQL

Vérifier l'installation en ouvrant la console MySQL Command Line Client

- Enter le mot de passe choisi
- Tester quelques commandes simple SQL
- Créer un BD simple (2 colonnes)



```
Select MySQL 5.5 Command Line Client

Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.5.16 MySQL Community Server (GPL)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| performance_schema |
| test       |
+-----+
4 rows in set (0.01 sec)

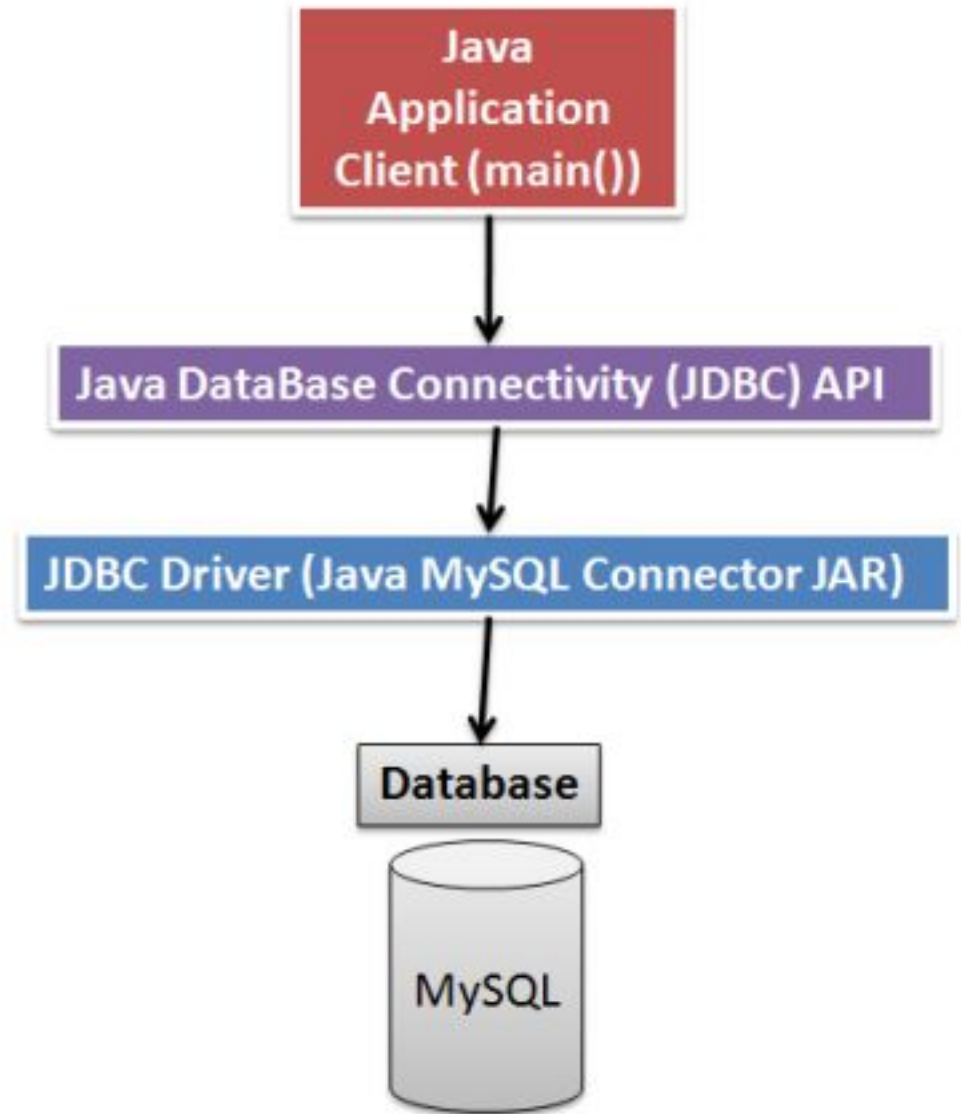
mysql> _
```

Grâce au connecteur **JDBC** on pourra accéder directement À MySQL depuis le code Java dans Eclipse

Les prochaines sections décrivent les installations et configurations nécessaires ainsi qu'une application à titre d'exemple

Accéder à une BD MySql depuis Java (Eclipse)

- Pour accéder à un serveur de BD depuis votre programme Java, il faut assurer les prérequis suivants:
- Le serveur de BD MySQL est installé avec le port **3306 ouvert** pour permettre aux PC d'y accéder.
- Si le SGBD tourne sur votre PC le host sera **localhost** au lieu de **nom.domain.com**
- Une librairie Java appelé **connector** enverra les commandes SQL depuis votre programme Java au serveur MySQL à travers le réseau
- Vous devez avoir déjà crée votre BD avec plusieurs tables se situant sur le serveur MySQL (utiliser PhpMyAdmin)
- On utilise l'API ou connecteur JDBC



1. Installer Eclipse si pas déjà fait
2. Télécharger JDBC (Java Database Connectivity) de: <http://www.mysql.com/products/connector>
 - Sélectionner: **Platform Independent**
3. Décompresser le fichier et assurer vous de l'existence d'un fichier jar similaire à
 - mysql-connector-java-5.1.34-bin.jar
4. Créer un nouveau projet java dans Eclipse: ex MySQLTest1
5. Faire coulisser le connecteur JDBC **mysql-connector-java-5.1.34-bin.jar** dans le projet Eclipse
 - Sélectionner **copy file** et non *link file*
6. Créer 2 classes Java : (voir les fichiers sources dans le répertoire Exemple-MySQL-JDBC)
 - Main.java et
 - MySQLAccess.java
7. Cliquer sur mysql-connector-java-5.1.34-bin.jar dans le projet pour le sélectionner
8. CTRL-click ou click droit pour ouvrir une fenêtre
 - Sélectionner **Build-Path** puis Configurer **Build-Path**
9. Dans la nouvelle fenêtre, sur le côté droit cliquer sur **Libraries**

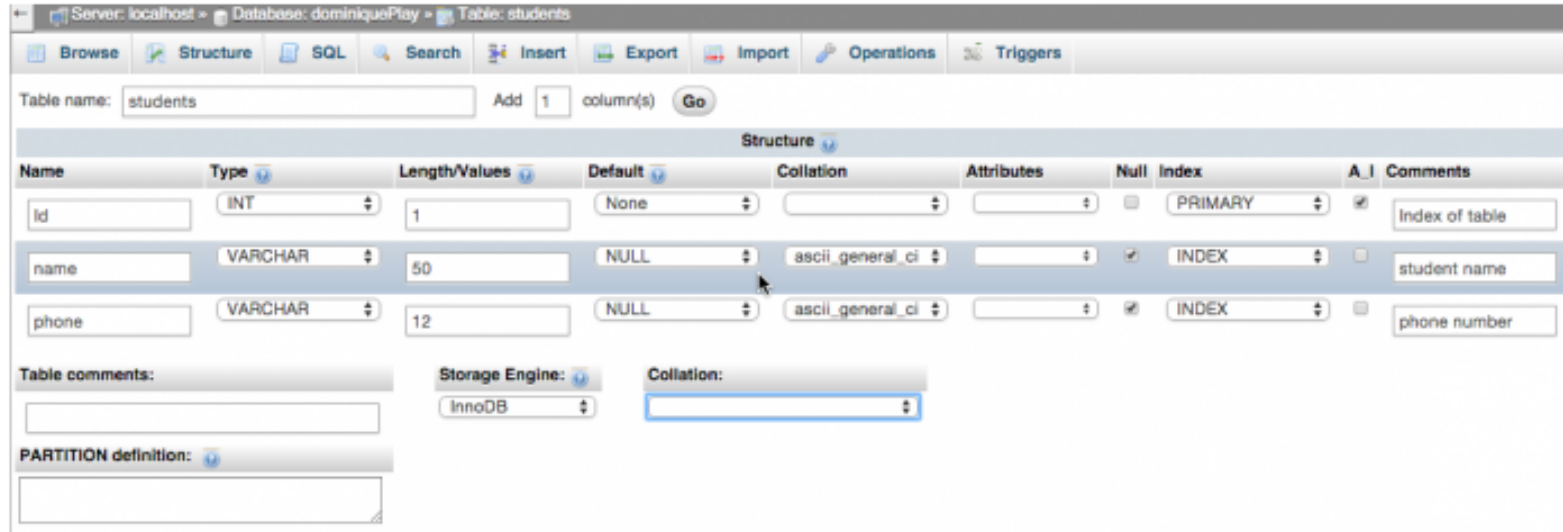
Les étapes d'installation (cont)

10. Cliquer sur **Add JARs**
11. Dans la fenêtre **JAR selection**, double-cliquer sur MySQLTest1
12. Sélectionner la librairie **mysql-connector-java-5.1.34-bin.jar**
13. Cliquer OK
14. Le connecteur MySQL est à présent disponible lors de la compilation du code Java
15. Sélectionner votre classe Main dans le projet et dans la fenêtre **Package Explore** l'exécuter comme une application
16. Si tout se passe bien la fenêtre de la console affichera les données récupérées du Serveur de Base de données

```
User: lars
Website: http://www.vogella.com
Summary: Summary
Date: 2009-09-14
Comment: My first comment
User: lars
Website: http://www.vogella.com
Summary: Summary
Date: 2009-09-14
Comment: My first comment
User: Test
Website: TestWebpage
Summary: TestSummary
Date: 3910-01-11
Comment: TestComment
The columns in the table are:
Table: comments
Column 1 id
Column 2 MYUSER
Column 3 EMAIL
Column 4 WEBPAGE
Column 5 DATUM
Column 6 SUMMARY
Column 7 COMMENTS
```


Création de la BD pour un deuxième exemple

- Utiliser PhpMyAdmin pour créer la table *students* qui doit contenir 3 colonnes

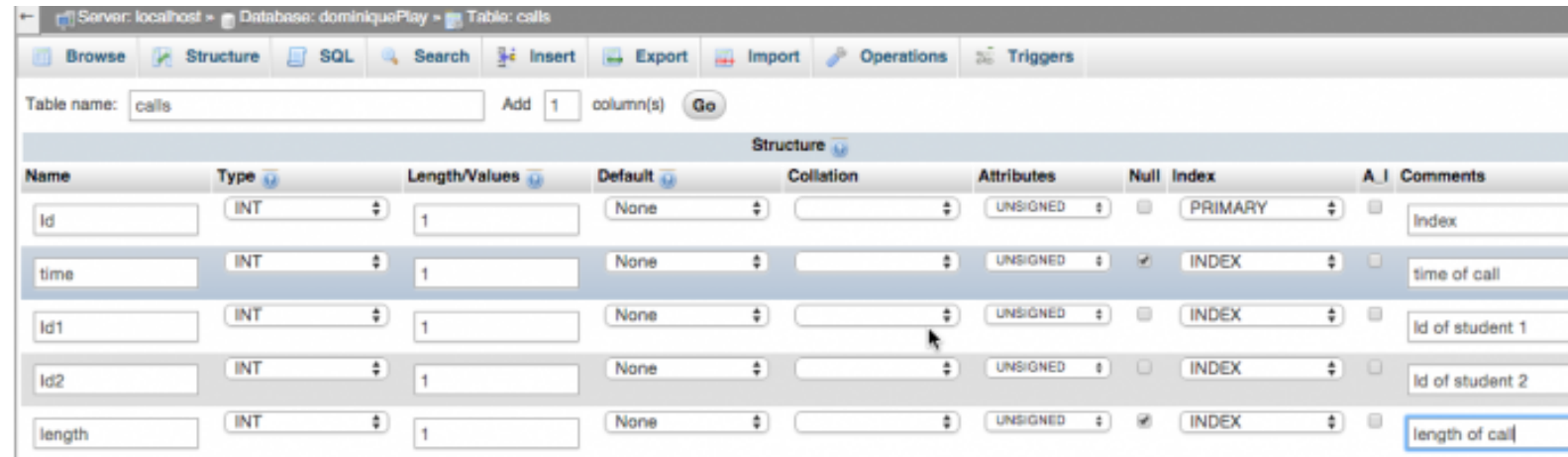


The screenshot shows the 'Structure' tab for a table named 'students' in a database named 'dominiquePlay' on 'localhost'. The table has three columns: 'id' (INT, 1, PRIMARY, Index of table), 'name' (VARCHAR, 50, INDEX, student name), and 'phone' (VARCHAR, 12, INDEX, phone number). The 'Storage Engine' is set to 'InnoDB' and the 'Collation' is 'ascii_general_ci'.

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A.I	Comments
id	INT	1	None				PRIMARY		Index of table
name	VARCHAR	50	NULL	ascii_general_ci			INDEX		student name
phone	VARCHAR	12	NULL	ascii_general_ci			INDEX		phone number

Remplir la table avec 3 étudiants et entrer les valeurs de votre choix

- Créer une autre table *calls* avec 5 colonnes pour enregistrer les appels téléphoniques



The screenshot shows the 'Structure' tab for a table named 'calls' in a database named 'dominiquePlay' on 'localhost'. The table has five columns: 'id' (INT, 1, PRIMARY, Index), 'time' (INT, 1, INDEX, time of call), 'id1' (INT, 1, INDEX, Id of student 1), 'id2' (INT, 1, INDEX, Id of student 2), and 'length' (INT, 1, INDEX, length of call).

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A.I	Comments
id	INT	1	None		UNSIGNED		PRIMARY		Index
time	INT	1	None		UNSIGNED		INDEX		time of call
id1	INT	1	None		UNSIGNED		INDEX		Id of student 1
id2	INT	1	None		UNSIGNED		INDEX		Id of student 2
length	INT	1	None		UNSIGNED		INDEX		length of call

Remplir la table avec 4 appels et entrer les valeurs de votre choix

Le répertoire Exemple-MySQL-JDBC contient les fichiers sources suivants :

- **Main.java** est le corps du programme principal
- **MySQLAccess.java** donne le code pour se connecter à la BD et faire des requêtes SQL

Le code Java qui permet d'accéder à la nouvelle BD se trouve dans le fichier

- **AcessStudentsCalls.java**

Le code Java pour lire les valeurs de la table **sudents** crée avant se trouve dans le fichier

- **MainStudentsCalls.java**

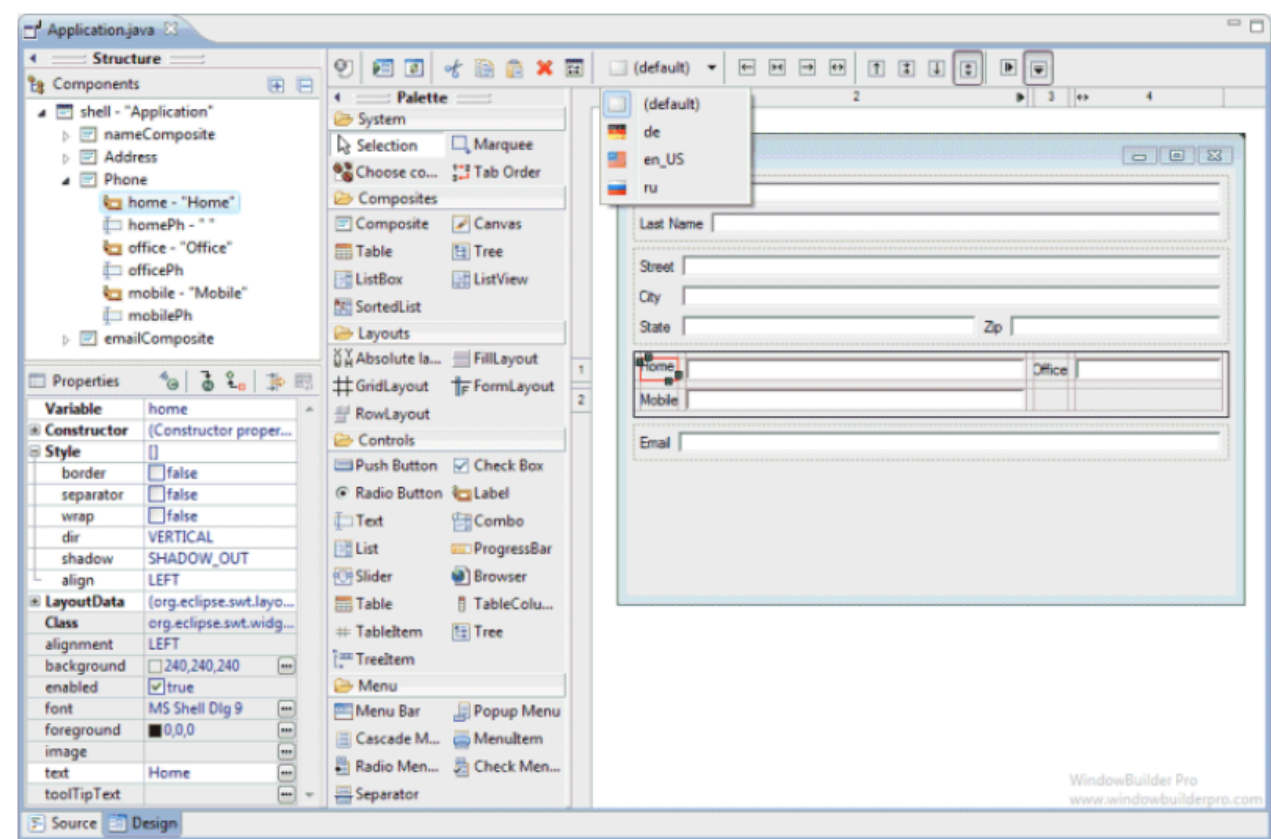
- Vous pouvez exécuter votre programme Java pour vérifier que vous obtenez bien les données entrées lors de la création de la table **students** dans PhpMyAdmin

Utilisation de WindowBuilder

- C'est un Designer de JAVA GUI
- Cet outil puissant permet de créer des interfaces graphiques complexes sans perdre trop de temps à produire des lignes de codes
- Il suffit de créer notre Interface en WYSIWYG et une fois complété, WindowBuilder va générer le code Java correspondant !!!

Pour réaliser les TP Java évènementiels vous devez installer WindowBuilder

- Eclipse -> help -> install new software -> General purpose tools -> Window builder
- Voir aussi les tutoriels pratiques sur youtube



Le Tutoriel se trouve à:

<https://www.eclipse.org/windowbuilder/>