COLLÈGE MONTMORENCY

Travail pratique #2

Pondération : 18%
Travail en équipe de 2

Toutes les spécifications du projet doivent être respectées

Dates des remises

Livrable	*Contenu*	*Date de remise*	Dossier de remise	%
#0	• Le cadre général de l'application incluant une planification du projet en deux livrables. • Le modèle logique de la base de données.	Vendredi 12 avril 2019 avant 23 h 55.	Tp2\cadre général de l'application	20/100
#1	• Le premier livrable selon la planification du projet. • Journal des historiques de EGit (log) en format txt	Mardi 23 avril 2019 avant 23 h 55.	Tp2\livrable1	40/100
#2	• Le deuxième livrable selon la planification du projet. • Journal des historiques de EGit (log) en format txt	Mardi 30 avril 2019 avant 23 h 55	Tp2\livrable2	40/100
Le « workSpace » du projet doit être nommé : TP2_VosNoms_livrable_N. Celui-ci doit être compressé en format compressé (.ZIP ou .RAR), puis déposé dans Moodle dans le dossier attitré selon le livrable. N représente le numéro du livrable (0,1 ou 2)				
Pénalité de retard pour chaque livrable : 5% par jour de retard				

Programmation d'une application de bibliothèque de musique connectée avec une base de données MySQL ou SQLite

Buts :

- Établir le cadre général de l'application.
- Établir un échéancier de développement de l'application.
- Concevoir l'interface graphique de l'application à partir des spécifications fonctionnelles.
- Programmer l'interface graphique de l'application.
- Utilisation des composants Swing complexes (table et liste).
- Modéliser les données de l'application.
- Programmer les modules non graphiques de l'application.
- Utilisation de l'API JDBC pour interagir avec une base de données (MySQL ou SQLite)
- Produire l'aide en ligne.
- Produire la documentation relative à l'application.

Travail à faire :

Vous devez programmer une application Java graphique d'une bibliothèque de musique (artiste, album, chansons, genre) qui interagit avec une base de données de type MySQL ou SQLite. Votre programme doit être réalisé selon le modèle MVC (modèle vue contrôleur), et respecter l'ensemble des spécifications du projet énoncées ici-bas.

Spécifications fonctionnelles :

L'application Bibliothèque de musique implique les tâches suivantes :

1. Authentification par un nom d'usager et un mot de passe.
2. La base de données doit avoir les tables suivantes :
    a. Les informations décrivant l'artiste sont:

Numéro (identifiant)
Nom de l'artiste
Membre : champ indiquant si l'artiste est membre de l'association des artistes (Oui/Non)
Photo de l'artiste

    b. Un album est décrit par les informations suivantes:

Numéro
Titre
Genre : choisissez en 5 parmi les suivants : (Folk, Jazz, Classique, Hip-Hop, Pop, Rock, Alternative, Indie Rock, Franco)
Année de la sortie de l'album
Image de la couverture de l'album
Numéro de l'artiste principal : Un album appartient à un seul artiste principal, mais un artiste peut posséder plusieurs albums.

3. La gestion des artistes qui comporte les traitements suivants :
    a. Ajouter un artiste.
    b. Afficher un artiste et tous ses albums (voir la figure 2)
    c. Modifier un artiste.
        i. Tous les champs de l'artiste sont modifiables sauf le numéro de l'artiste qui est un identifiant.
    d. Supprimer un artiste sélectionné dans la liste.
        i. Un artiste ne peut être supprimé s'il possède au moins un album. Dans ce cas, l'utilisateur est informé par le biais d'un message d'erreur que la suppression ne peut être effectuée. L'utilisateur doit confirmer la suppression de l'artiste par le biais d'une boite de dialogue.
    e. Rechercher un artiste avec le champ suivant: **nom de l'artiste**.
        i. La recherche doit pouvoir ce faire même si le champ est connu partiellement.
            1. Exemple : nom artiste = jos, cela affichera tous les enregistrements dont le nom contient la chaîne « jos ».
4. Rendre disponible l'aide en ligne de l'application.

Spécifications concernant les interfaces graphiques :

A. *Spécifications générales*

- Toutes les fenêtres doivent avoir un titre significatif et la même icône symbolisant l'application (significative).
- On doit pouvoir retourner au menu précédent ou annuler l'opération en cours à tout moment. Adaptez les interfaces graphiques pour le faire.
- L'application doit avoir au minimum 5 fenêtres.
    - Une fenêtre d'identification de l'utilisateur.
    - Une fenêtre de choix des traitements (menu).
    - Une fenêtre de Gestion des artistes.
    - Une fenêtre de Gestion des albums.
    - Une fenêtre pour afficher l'aide en ligne.
- Une description de ces fenêtres est fournie ici-bas.

B. *Descriptions des fenêtres de l'application*

Fenêtre d'identification de l'utilisateur

Elle permet d'accéder à l'application. Le mot de passe ne doit pas s'afficher. La figure #1 montre une maquette de votre fenêtre.

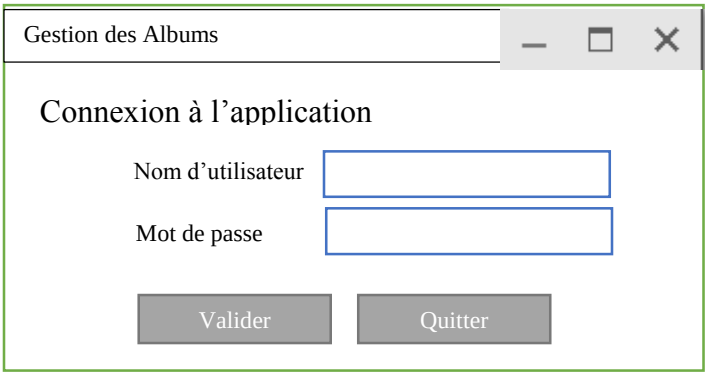


Figure 1

Fenêtre Choix des traitements

Elle affiche un menu permettant le choix des traitements à effectuer. Ces choix sont :

- Artistes : ouvre la fenêtre de gestion des artistes.
- Albums : ouvre la fenêtre de gestion des albums.
- Quitter : permet de quitter l'application après une demande de confirmation.

Vous devez faire la conception de cette fenêtre vous-mêmes.

Fenêtre Gestion des artistes

Elle affiche la liste des artistes dans une table ainsi qu'un menu offrant les traitements possibles liés aux artistes : Ajout, Modification, suppression et recherche. Voici la maquette de cette fenêtre. Cette fenêtre doit s'adapter à la taille de l'écran.

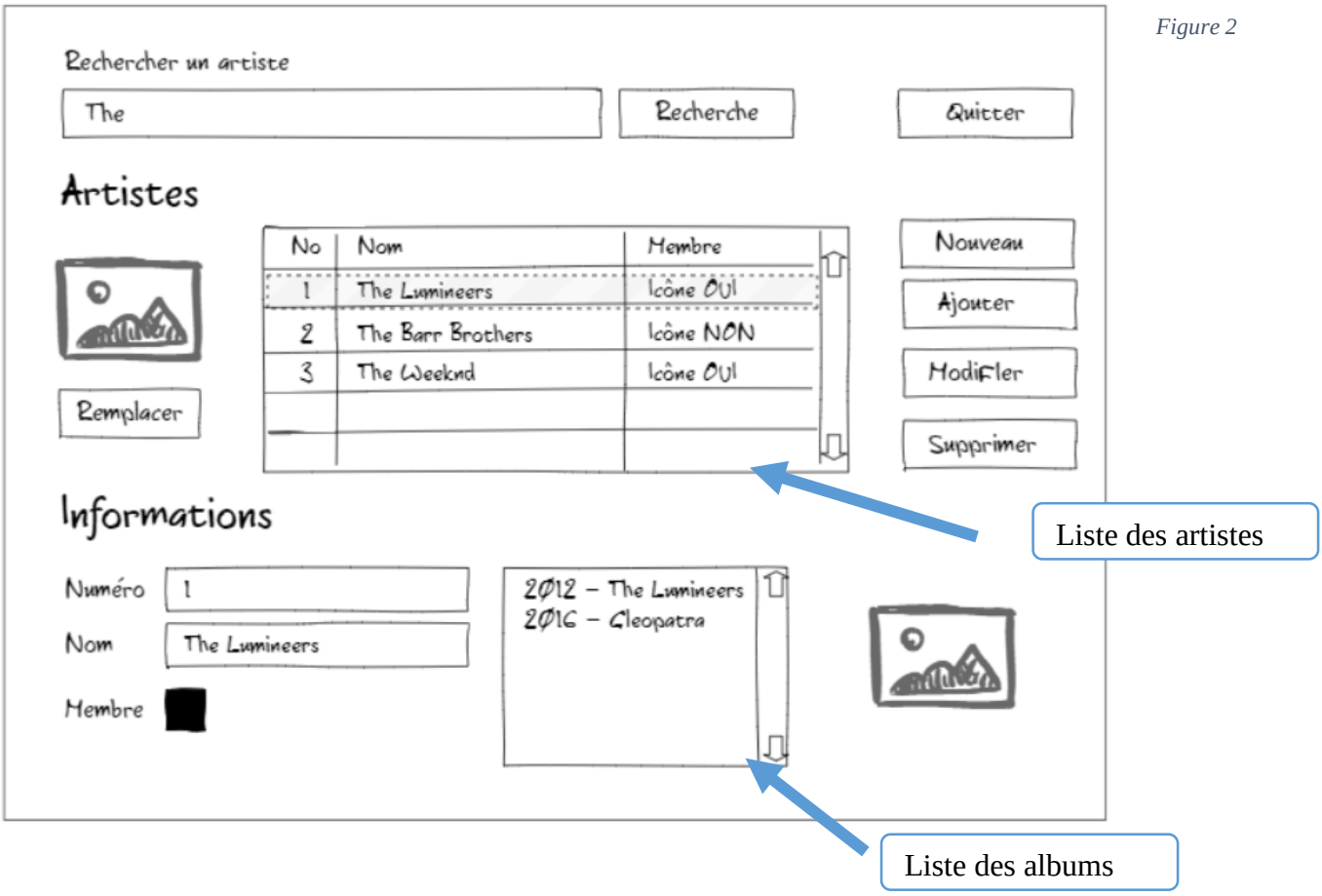


Figure 2

Liste des artistes

- Les artistes sont affichés dans un composant `JTable` muni d'une barre de défilement.
  - Le champ membre est sous forme d'icône symbolisant l'affiliation à l'association des artistes.
  - Le champ numéro est affiché en gras et centré.
  - En sélectionnant un artiste de la table, ses informations sont affichées dans le formulaire plus bas:
    - Photo de l'artiste dans un composant `Jlabel`;
      - Permettre le remplacement de l'image de l'artiste par l'utilisateur quand on modifie ou un ajoute un artiste ou un album.
    - Numéro et nom dans une zone de texte non éditable;
    - Champ membre dans une case à cocher.
    - Albums de l'artiste dans un composant `JList`;
      - On veut afficher uniquement le titre de l'album et l'année de sa sortie en ordre croissant de l'année. Redéfinissez la méthode `toString()` de la classe Album.
    - Photo de l'album de l'artiste dans un composant `Jlabel` lorsqu'on sélectionne un album dans la liste;

Boutons

- Le bouton Nouveau permet de créer un nouvel artiste.
    - Tous les champs de la section informations seront à vide sauf le Numéro d'artiste qui lui, s'auto-incrémentera automatiquement selon le dernier numéro de la base de données
    - On appuie sur le bouton Ajouter pour ajouter l'artiste dans la base de données
    - Une image par défaut sera ajoutée.
- En double cliquant sur une ligne de la table, on active la modification de l'artiste correspondant. On peut alors modifier le nom et/ou le champ membre qui apparaissant dans le formulaire de modification.
    - Par la suite, on appuie sur le bouton Modifier pour sauvegarder la modification dans la base de données.
- Le bouton supprimer permet de supprimer un artiste et ses albums de la base de données

Fenêtre Aide en ligne

Un document expliquant la réalisation de cette fenêtre vous sera remis dans Moodle ultérieurement.

BONUS (10 points)

GESTION DES ALBUMS

La gestion des albums comporte les traitements suivants :

a. Ajouter un album.
b. Afficher la liste des albums.
c. Modifier un album.
d. Supprimer un album.
    i. L'utilisateur doit confirmer la suppression de l'album par le biais d'une boite de dialogue.
e. Rechercher des albums à partir d'un ou de plusieurs champs suivants : **titre, années de sortie, artiste ou genre.**
    i. Ces champs peuvent être saisis totalement ou partiellement.

Fenêtre Gestion des albums

Vous devez faire la conception de cette fenêtre vous-mêmes.

Assurez-vous d'appliquer le même patron que la fenêtre de la figure #2 pour l'interface graphique ce cette fenêtre. C'est-à-dire :

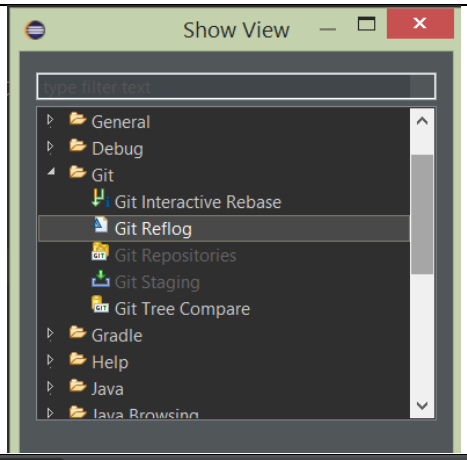
- Cette fenêtre doit s'adapter à la taille de l'écran.
- Être capable de faire une recherche par titre d'album.
- Être capable d'ajouter, de modifier ou de supprimer un album d'un artiste.
    - Le numéro d'album n'est pas modifiable et s'auto-incrémente automatiquement selon le dernier numéro de la base de données si on ajoute un album.
- Lors de la saisie d'un artiste pour un album, le client aimerait faire le choix dans une fenêtre comportant une table de tous les artistes simplement en double cliquant sur la ligne correspondante.

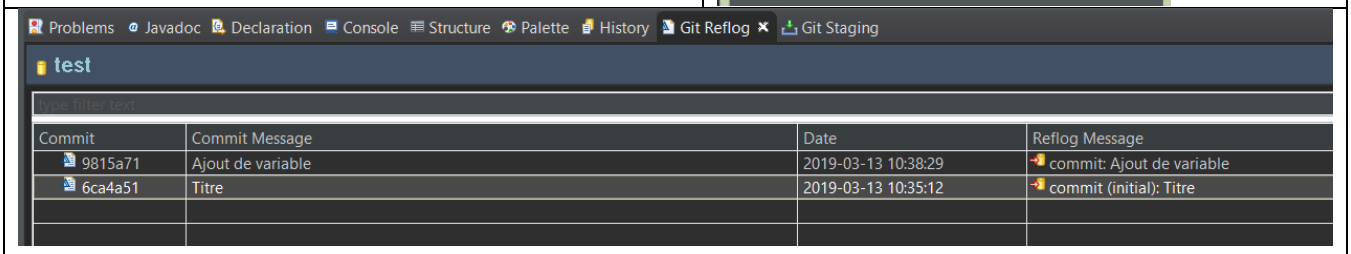
ANNEXE

Historique de Git dans Eclipse

Affichage de l'onglet du journal de référence

Dans Windows >>> Others >>> Git >>> Git Reflog

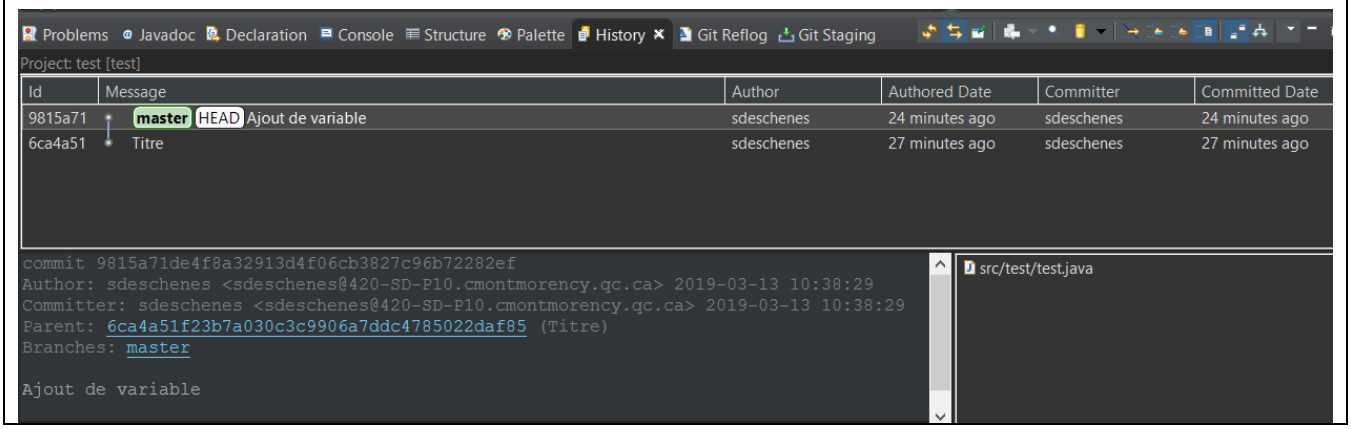




Par la suite, c'est possible de cliquer sur un des commit pour voir les modifications.

Affichage de l'onglet d'historique

Clic droit de la souris sur le projet relié à Git >>> Teams >>> Show in History

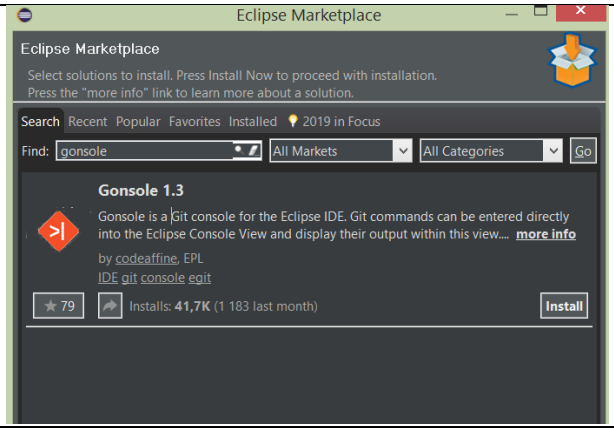


Ajout de la console Git à Eclipse

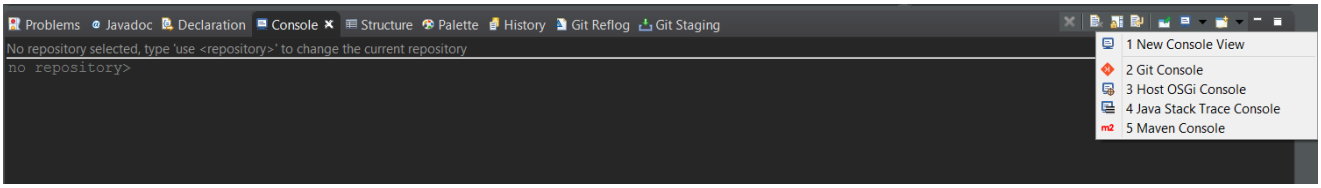
On recherche dans le « Marketplace » le Gonsole et on l'installe

Assurez-vous de confirmer les fonctionnalités voulues et les licences.

Acceptez le contenu non signé



Par la suite, allez dans l'onglet Console et cliquez sur le bouton Open Console. Par la suite, sélectionnez Git Console



Ensuite, vous devez vous connectez à votre dépôt en utilisant la commande `use <repository>`

Pour afficher le journal, faites la commande `log`

Par la suite, vous pouvez faire un copier-coller dans un fichier texte.