

**Programmation orienté objet avancée  
*Dossier d’énoncé du travail de Java***

*Informatique de gestion*

*UE 230 : Projet informatique intégré*

Vous devez remplir chacune des zones bleues encadrées dans le document.

# Identification des étudiants du groupe

## Etudiant 1

Prénom et nom :

|  |
| --- |
| Hugo Janssens |

Bloc (1, 2 ou 3) :

|  |
| --- |
| **2** |

Groupe (A, B, C …) :

|  |
| --- |
| **A** |

## Etudiant 2

Prénom et nom :

|  |
| --- |
| Wangi Weber |

Bloc (1, 2 ou 3) :

|  |
| --- |
| **2** |

Groupe (A, B, C …) :

|  |
| --- |
| **C** |

# Domaine d’application

## Sujet

|  |
| --- |
| **Agenda** |

## Description du domaine d’application

|  |
| --- |
| Notre domaine d’application est la gestion d’un agenda. Chaque agenda est personnel à un utilisateur et permet à ces derniers de créer ou de participer à des événements. Ces événements peuvent rester personnels à l’utilisateur qui les a créés ou bien être disponibles pour plusieurs utilisateurs à la fois.  Les utilisations principales de notre programme seront : l’affichage de la liste, la création, suppression et modification des événements auxquels notre utilisateur participe ou organise. Il pourra également les trier par type d’événement afin de pouvoir les différencier facilement.  Pour chaque événement de nombreuses informations peuvent/doivent être enregistrées comme le prix, les dates de début ou de fin, si l’événement est privé ou non, le nombre de personnes maximum, etc. Nous pouvons également enregistrer l’adresse de l’événement et stocker certaine dans elle-même si elles sont récurrentes.  Chaque utilisateur a un type, qui lui donne accès à des promotions pour chaque événement. Celles-ci seront donc similaires pour des utilisateurs d’un même type. |

# Base de données

## Schéma entités-associations

|  |
| --- |
|  |

## Schéma des tables

|  |
| --- |
| Image |

## Documentation des tables

Pour chacune des tables, remplissez les zones ci-dessous.

*N.B. Dans la description des colonnes, placez une croix dans "Oblig" si la colonne est obligatoire et placez une croix dans "ID" si la colonne est identifiante à elle seule (clé primaire ou secondaire (unique)). S’il s’agit d’une colonne clé étrangère, placez une croix dans "FK" et précisez le nom de la table référencée.*

Nom de la table **n°1**:

|  |
| --- |
| **Event** |

Clé primaire :

|  |
| --- |
| **id** |

Description des colonnes :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nom de colonne | Type(longueur) | Oblig | ID | FK | Table référencée |
| Id | Int | X | X |  |  |
| Title | Varchar | X |  |  |  |
| Description | Text | X |  |  |  |
| AdditionnalInformations | Text |  |  |  |  |
| IsImportant | Boolean | X |  |  |  |
| StartDate | Date avec heure | X |  |  |  |
| EndDate | Date avec Heure | X |  |  |  |
| Price | Int | X |  |  |  |
| ParticipantNbMax | Int |  |  |  |  |
| IsPrivate | Boolean | X |  |  |  |
| fk\_creator | int |  |  | x | User |
| fk\_eventType | int | x |  | x | EventType |
| fk\_address | int | x |  | x | Adress |

Nom de la table **n°2** :

|  |
| --- |
| **User** |

Clé primaire :

|  |
| --- |
| **id** |

Description des colonnes :

| Nom de colonne | Type(longueur) | Oblig | ID | FK | Table référencée |
| --- | --- | --- | --- | --- | --- |
| Id | Int | X | X |  |  |
| FirstName | Varchar | X |  |  |  |
| LastName | Varchar | X |  |  |  |
| fk\_userType | int | x |  | x | UserType |

Nom de la table **n°3** :

|  |
| --- |
| **UserType** |

Clé primaire :

|  |
| --- |
| **id** |

Description des colonnes :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nom de colonne | Type(longueur) | Oblig | ID | FK | Table référencée |
| Id | Int | X | X |  |  |
| Name | Varchar | X |  |  |  |

Nom de la table **n°4** :

|  |
| --- |
| **Promotion** |

Clé primaire :

|  |
| --- |
| **id** |

Description des colonnes :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nom de colonne | Type(longueur) | Oblig | ID | FK | Table référencée |
| Id | Int | X | X |  |  |
| ReductionPourcent | Int | X |  |  |  |
| fk\_userType | int | x |  | x | UserType |
| fk\_event | int | x |  | x | Event |

Nom de la table **n°5**:

|  |
| --- |
| **Address** |

Clé primaire :

|  |
| --- |
| **id** |

Description des colonnes :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nom de colonne | Type(longueur) | Oblig | ID | FK | Table référencée |
| Id | Int | X | X |  |  |
| Label | Varchar |  |  |  |  |
| NumberStreet | Text | X |  |  |  |
| AdditionnalInformations | Text |  |  |  |  |
| fk\_locality | int | x |  | x | Locality |

Nom de la table **n°6** :

|  |
| --- |
| **Locality** |

Clé primaire :

|  |
| --- |
| **id** |

Description des colonnes :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nom de colonne | Type(longueur) | Oblig | ID | FK | Table référencée |
| Id | Int | X | X |  |  |
| ZipCode | Int | X |  |  |  |
| Name | Varchar | X |  |  |  |

Nom de la table **n°7** :

|  |
| --- |
| **EventType** |

Clé primaire :

|  |
| --- |
| **id** |

Description des colonnes :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nom de colonne | Type(longueur) | Oblig | ID | FK | Table référencée |
| Id | Int | X | X |  |  |
| Name | Varchar | X |  |  |  |

Nom de la table **n°8** :

|  |
| --- |
| **Participation** |

Clé primaire :

|  |
| --- |
| **id** |

Description des colonnes :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nom de colonne | Type(longueur) | Oblig | ID | FK | Table référencée |
| Id | Int | X | X |  |  |
| fk\_event | int | X |  | X | Event |
| Fk\_user | int | X |  | X | User |

# Fonctionnalités

## CRUD : Listing, insertion, suppression, modification

Vous devez prévoir les fonctionnalités complètes d’administration d’une table de la base de données. Les opérations CRUD (**C**reate, **R**ead, **U**pdate et **D**elete) doivent être faites sur une table qui contient :

* Des **colonnes de types différents** (au moins une colonne de type **texte**, au moins une colonne de type **numérique**, au moins une colonne de type **date** et au moins un **booléen**) ;
* Plusieurs **colonnes facultatives**;
* Au moins une **colonne clé étrangère** vers une autre table de la base de données.

*N.B. 4 Fonctionnalités seront donc implémentées sur une même table, à savoir le listing du contenu de la table (dans une JTable), un formulaire d’insertion d’une nouvelle ligne dans la table, la suppression d’une ou plusieurs lignes choisie(s) par l’utilisateur et la possibilité de modifier une ligne de la table (en remplissant d’abord les champs du formulaire d’insertion avec les valeurs de la ligne choisie, en permettant ensuite à l’utilisateur de modifier les champs du formulaire et enfin de sauver la ligne ainsi modifiée).*

Nom de la table pour les opérations CRUD :

|  |
| --- |
| **Event** |

*Vérification du respect des contraintes :*

Colonne de type date :

|  |
| --- |
| **StartDate, EndDate** |

Colonne de type booléen :

|  |
| --- |
| **isPrivate, isImportant** |

Colonnes facultatives :

|  |
| --- |
| **ParticipantNbMax** |

Colonne clé étrangère (précisez vers quelle table) :

|  |
| --- |
| **id (vers TypeEvent), id (vers User “creator”),**  **id (vers User “participant” ), id (vers Address)** |

## Recherches

### Recherche 1

Titre de la recherche :

|  |
| --- |
| **Les événements en fonction du type d'événement** |

Objectif de la recherche :

|  |
| --- |
| **Obtenir la liste des événements correspond à un type précis.** |

Jointure entre les tables suivantes (**au moins 3**) :

|  |
| --- |
| **Table 1 : EventType**  **Table 2 : Event**  **Table 3 : User**  **Table 4 : UserType** |

Description des entrées (introduites par l’utilisateur) :

|  |  |
| --- | --- |
| Critère de recherche | Format (type de composant Swing) |
| Sélectionne un type d'événement | Combo-box proposant la liste des noms des types d'événements existant dans la bd |
|  |  |
|  |  |

Sorties (dans une JTable) :

|  |  |
| --- | --- |
| Nom de la colonne | Provenant de la table (nom de la table) |
| title | Event |
| startDate | Event |
| endDate | Event |
| firstName | User |
| lastName | User |
| name | UserType |

### Recherche 2

Titre de la recherche :

|  |
| --- |
| **Recherche d'événement entre deux dates données** |

Objectif de la recherche :

|  |
| --- |
| **Trouver tous les événements situés entre une date de début et une date de fin définies par l’utilisateur.** |

Jointure entre les tables suivantes (**au moins 3**) :

|  |
| --- |
| **Table 1 : Event**  **Table 2 : Address**  **Table 3 : Locality** |

Description des entrées (introduites par l’utilisateur) :

|  |  |
| --- | --- |
| Critère de recherche | Format (type de composant Swing) |
| startDate | JXDatePicker de date |
| endDate | JXDatePicker de date |
|  |  |

Sorties (dans une JTable) :

|  |  |
| --- | --- |
| Nom de la colonne | Provenant de la table (nom de la table) |
| titre | Event |
| startDate | Event |
| isPrivate | Event |
| CompleteAddress (numberStreet, zipCode, name) | Address  Locality |

### Recherche 3

Titre de la recherche :

|  |
| --- |
| **Quelles sont les réductions disponibles pour un type d’utilisateur** |

Objectif de la recherche :

|  |
| --- |
| **Afficher l’ensemble des promotions par type d’utilisateur** |

Jointure entre les tables suivantes (**au moins 3**) :

|  |
| --- |
| **Table 1 : UserType**  **Table 2 : Promotion**  **Table 3 : Event**  **Table 4 : EventTYpe** |

Description des entrées (introduites par l’utilisateur) :

|  |  |
| --- | --- |
| Critère de recherche | Format (type de composant Swing) |
| le type d’utilisateur | Combo box proposant la liste des noms des types d’utilisateur existant dans la bd |
|  |  |
|  |  |

Sorties (dans une JTable) : **le tout trié par date**

|  |  |
| --- | --- |
| Nom de la colonne | Provenant de la table (nom de la table) |
| reduction | Promotion |
| titre | Event |
| startDate | Event |
| endDate | Event |
| prixCalculé | Event & Promotion |
| name | EventType |

## Tâche métier

Vous devez implémenter au moins une fonctionnalité relative à votre projet autre que les précédentes (exemples : planification, règles métier, statistiques…). Cette tâche sera implémentée dans la couche métier (calculs, règles métier…). L’exécution de cette tâche métier peut nécessiter éventuellement l’affichage de plusieurs écrans successifs, l’obtention d’informations auprès de l’utilisateur ou des accès en lecture et/ou écriture à la base de données.

Description de la tâche métier :

|  |
| --- |
| Notre tâche métier consistera par le biais de plusieurs écrans à l’affichage de nouvelles informations calculées sur base de celles présentent dans la BD. Les 3 écrans par lesquels nous devront passer pour atteindre cet objectif sont les suivants :  1) Premier écran : combo-box  Combo Box proposant la liste des firstname + lastname (+ Id entre parenthèses) disponibles. L’utilisateur choisit la personne de son choix pour avoir accès à l’ensemble des événements qu’il a créé.  2) Deuxième écran :  Jtable présentant la liste des événements de l’utilisateur sélectionné.   * La personne qui utilise le logiciel choisi un événement dans la JTable.   3) Troisième écran  Affichage des statistiques :  -Nombre de participants  -Prix total  -Total des promotions |

Entrées (saisies auprès de l’utilisateur) :

|  |
| --- |
| 1er étape (1er écran) : Sélection dans une combo-box d’un utilisateur  (2eme écran) Affichage des évènements qu’il a créés.  2eme étape (2eme écran) : Sélection d’un événement appartenant à l’utilisateur choisi. |

Sorties (affichage) :

|  |
| --- |
| (3eme et dernier écran)  - Affichage du nombre de participants à l'événement.  - Affichage de la somme totale en fonction de tous les participants et leurs promotions.  - Affichage de la somme totale sans les promotions.  - Affichage de la somme totale des promotions de tous les participants. |

## Thread

Description du processus parallèle (autre que l’affichage de l’heure) :

|  |
| --- |
| **On créera une animation graphique sur un des coins de l’écran. Celle-ci représentera un calendrier dont les jours et les mois défilent chronologiquement. L’animation sera visible tout au long de l’utilisation du programme.** |