20/02/2018

Projet Mastermind

B2 Ingesup

Jonathan DALBERTO Mathias LAFONTAN Fanny LAJEUNESSE Simon ROBERT

Ynov aix-en-provence

Table des matières

[1. Présentation du projet 2](#_Toc506903520)

[a. Objectifs du projet 2](#_Toc506903521)

[b. Acteurs 2](#_Toc506903522)

[2. Spécifications 3](#_Toc506903523)

[a. Base de données 3](#_Toc506903524)

[i. Dictionnaire de données 3](#_Toc506903525)

[ii. Modèle conceptuel de données 3](#_Toc506903526)

[iii. Modèle logique de données 4](#_Toc506903527)

[b. Diagrammes de séquence 4](#_Toc506903528)

[i. Lancement du jeu 4](#_Toc506903529)

[ii. Inscription 5](#_Toc506903530)

[iii. Authentification 5](#_Toc506903531)

[iv. Afficher le profil 6](#_Toc506903532)

[v. Modifier le profil 6](#_Toc506903533)

[vi. Choix Joueur VS Ordinateur 7](#_Toc506903534)

[3. Spécification technique 7](#_Toc506903535)

[a. Langage et environnement de développement 7](#_Toc506903536)

[b. Outils 7](#_Toc506903537)

[Annexes 9](#_Toc506903538)

[Modèle métier 9](#_Toc506903539)

# Présentation du projet

## Objectifs du projet

L’objectif de ce projet est de réaliser le développement d’un programme Java reproduisant le fonctionnement du jeu Mastermind. Appliquer les principes de la programmation orientée objet en réalisant le design complet de l’architecture et en mettant en œuvre les méthodes de conception objet. Fournir une documentation exhaustive et cohérente rendant compte de la démarche adoptée et des différentes étapes d’analyse, de conception, de développement, de test et de déploiement de l’application.

Connecter Java à une base de données (ici, MySQL).

Implémenter quelques-uns des patrons de conception (design patterns) indispensables dans l’architecture objet.

## Acteurs

L’équipe projet est composée de quatre étudiants :

* DALBERTO Jonathan (développeur BDD)
* LAFONTAN Mathias (lead développeur)
* LAJEUNESSE Fanny (chef de projet, développeur BDD)
* ROBERT Simon (développeur)

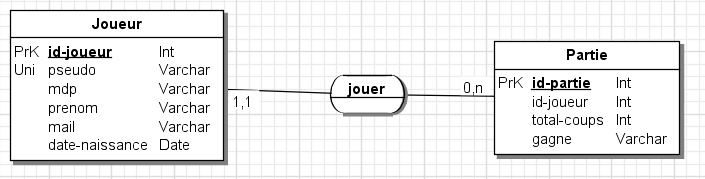
# Spécifications

## Base de données

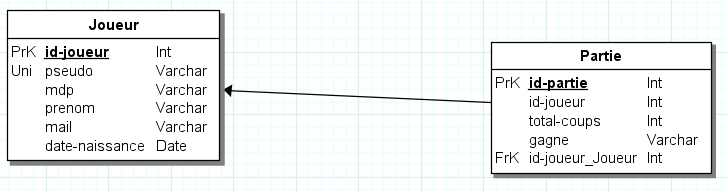
### Dictionnaire de données

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **NOM** | **DESCRIPTION** | **FORMAT** | **LONGUEUR** | **TYPE** | **GESTION** | **CALCUL** | **SOURCE** |
| **id\_joueur** | Identifiant du joueur | int | 10 | E | auto\_increment |  | B2 - projetDev -1718.pdf |
| **pseudo** | Pseudo | varchar | 25 | E | unique |  |  |
| **mdp** | Mot de passe | varchar | 8 | E | 1> et <8 |  |  |
| **nom** | Nom | varchar | 50 | E | N/A |  |  |
| **prenom** | Prénom | varchar | 50 | E | N/A |  |  |
| **date\_naissance** | Date de naissance | date |  | E | N/A |  |  |
| **mail** | Adresse mail | varchar | 50 | E | regex |  |  |
| **nb\_parties** | Nombre de parties jouées | int | 10 | C | count | count all games from id\_joueur |  |
| **nb\_gagne** | Nombre de parties gagnées | int | 10 | C | count | count all won games from id\_joueur |  |
| **nb\_perdu** | Nombre de parties perdues | int | 10 | C | count | count all lost games from id\_joueur |  |
| **meilleure\_partie** | Meilleure partie réalisée |  |  |  | N/A |  |  |
| **id\_partie** | Identifiant de la partie | int | 10 | E | auto\_increment |  |  |
| **nb\_coups** | Nombre de coups joués | int | 2 | E | <11 |  |  |
| **gagne** | Indique si la partie a été gagnée ou non | Bool |  | E | N/A |  |  |

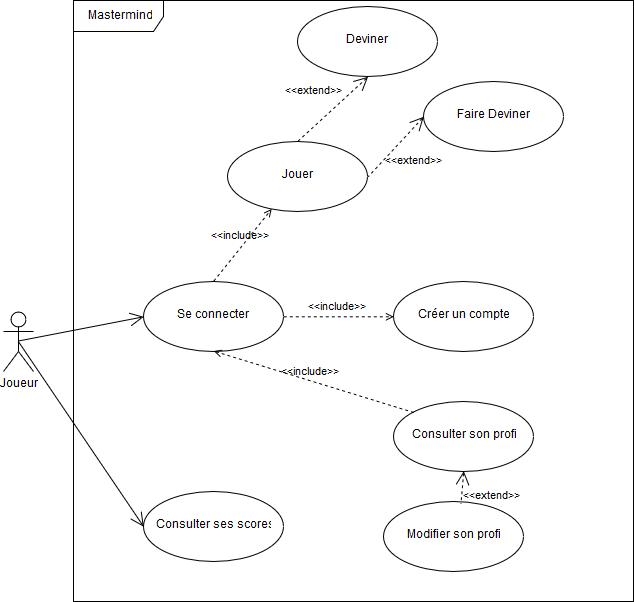
### Modèle conceptuel de données



### Modèle logique de données

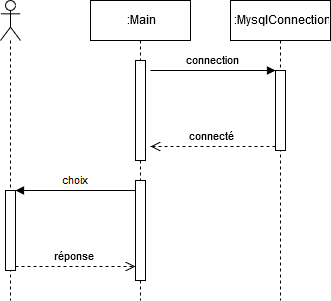


## Use Case

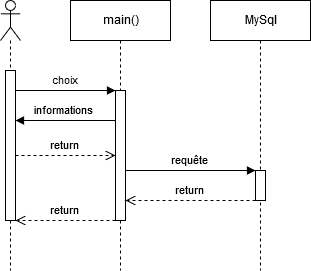


## Diagrammes de séquence

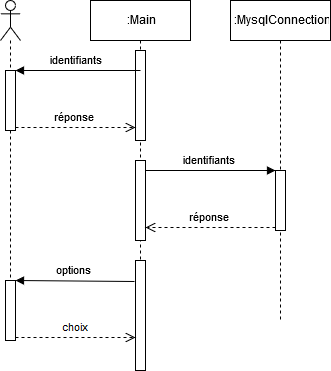
### Lancement du jeu



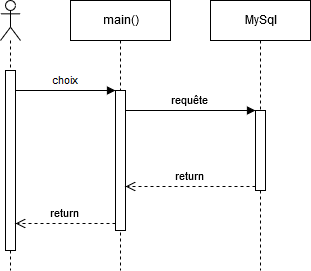
### Inscription



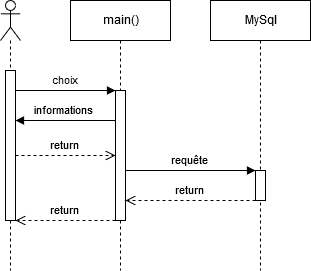
### Authentification



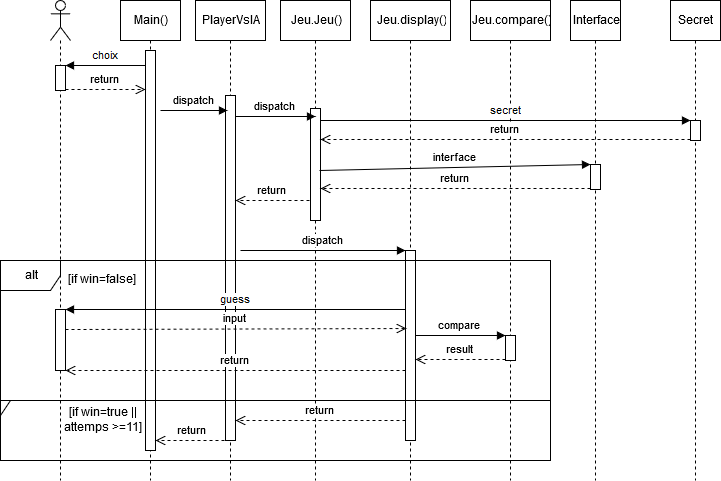
### Afficher le profil



### Modifier le profil



### Choix Joueur VS Ordinateur



## Algorithmes utilisés

### Joueur

Le joueur doit deviner la combinaison. Le programme récupère la combinaison donnée par le joueur (« guess ») et la compare à la combinaison (« secret ») qu’il a générée et stockée.

var guess;

guess = new Guess;

Secret = new Secret

Pour i allant de 0 à 5, avec un pas de 1 :

Est-ce que le chiffre à l’index i de guess existe dans secret ?

Si oui :

Est-ce que l’index est identique ?

Si oui :

BP++

Si non :

MP++

Si non : retour au début

* + 1. IA

# Spécification technique

## Langage et environnement de développement

Pour le développement de cette application, nous utiliserons le langage de programmation Java (version 1.8). Ce langage, étant orienté objet, est particulièrement adapté en raison de sa solidité et de sa maintenabilité. Il nous a été imposé dans le cadre du projet.

Pour la mise en place de la base de données, nous utiliserons le système de gestion open-source MySQL, lié à l’application Java grâce à l’interface JDBC (Java DataBase Connectivity).

La totalité du projet est déployée dans l’environnement Java 1.8, et nécessitera une machine virtuelle Java (JVM) dans une version 1.8 ou supérieure.

## Outils

Le développement se fera sur l’IDE IntelliJ (licence étudiante JetBrains).

Mise en place d’un répertoire Git pour la collaboration sur le code, ainsi que sur la documentation.

Mise en place d’un Google Drive pour le partage des informations et des divers documents, images…

Logiciel de modélisation JMerise (ou SQL Power Architect) pour le modèle de la base de données.

IntelliJ IDEA possède sa propre extension permettant de créer et/ou de générer les diagrammes de classe.

# Annexes

## Modèle métier