Document Technique V0
SAE 2.01

réalisé par le Olivencia Eliot, Veron Victor, Larrose Thomas

Groupe : 4B

fait le 04/06/2025

Groupe 2A SAE Développement d'application . n Informatique [informatique] BUT Blagnac à Toulouse , 2024.
Français .tel-0XXXXXXX

En remerciant tous nos professeurs qui nous ont pour la plupart apporté une quelconque aide que ce soit par les Travaux Dirigé, ainsi que les Travaux Pratiques et Cours en Amphithéâtre et de cette manière apporté du savoir afin de pouvoir réaliser cette Étude d'opportunité .

SOMMAIRE

Table des matières

Auteur : Olivencia Eliot, Veron Victor, Larrose Thomas

Document Technique V0

Application de Gestion de Tournois

1. Présentation de l'application

L'application est conçue pour gérer des événements e-sportifs. Elle permet d'organiser les jeux, les tournois, les équipes, les joueurs et le personnel impliqué. La première version (V0) met en place la structure de base et la gestion des données.

1.1. Rôles et Entités

L'application organise les personnes qui l'utilisent et les différents types d'informations qu'elle gère. Les utilisateurs qui se connectent à l'application ont des rôles bien définis, chacun ayant des autorisations différentes :

- **ADMIN** : Ce rôle est attribué aux administrateurs du système. Il leur donne un accès complet et les privilèges pour gérer toutes les parties de l'application et ses données.
- **ORGANISATEUR** : Ce rôle est pour les personnes qui sont en charge des tournois. Elles peuvent créer, modifier, et superviser les tournois ainsi que gérer les éléments liés à leur déroulement.

Pour son fonctionnement, le système manipule et stocke plusieurs catégories principales d'informations, qui sont ses "entités" :

- **Les utilisateurs** : Cela regroupe tous les comptes des personnes qui se connectent à l'application, qu'ils aient le rôle d'administrateur ou d'organisateur.
- **Les jeux** : Ce sont les titres de jeux vidéo autour desquels les tournois sont organisés. L'application enregistre les informations clés de chaque jeu.
- **Les tournois** : Chaque événement compétitif est enregistré comme un tournoi, avec ses propres caractéristiques comme le nom, les dates, le lieu, et le format de compétition.
- **Les équipes et les joueurs** : L'application permet de suivre les groupes de participants (équipes) et les individus (joueurs) qui s'inscrivent aux différents tournois.

- **L'affectation du personnel aux tournois** : Ce système gère la manière dont les membres du personnel, tels que les arbitres ou d'autres assistants, sont assignés à des tournois spécifiques pour accomplir des fonctions particulières.

Diagramme de cas d’utilisation de la V0

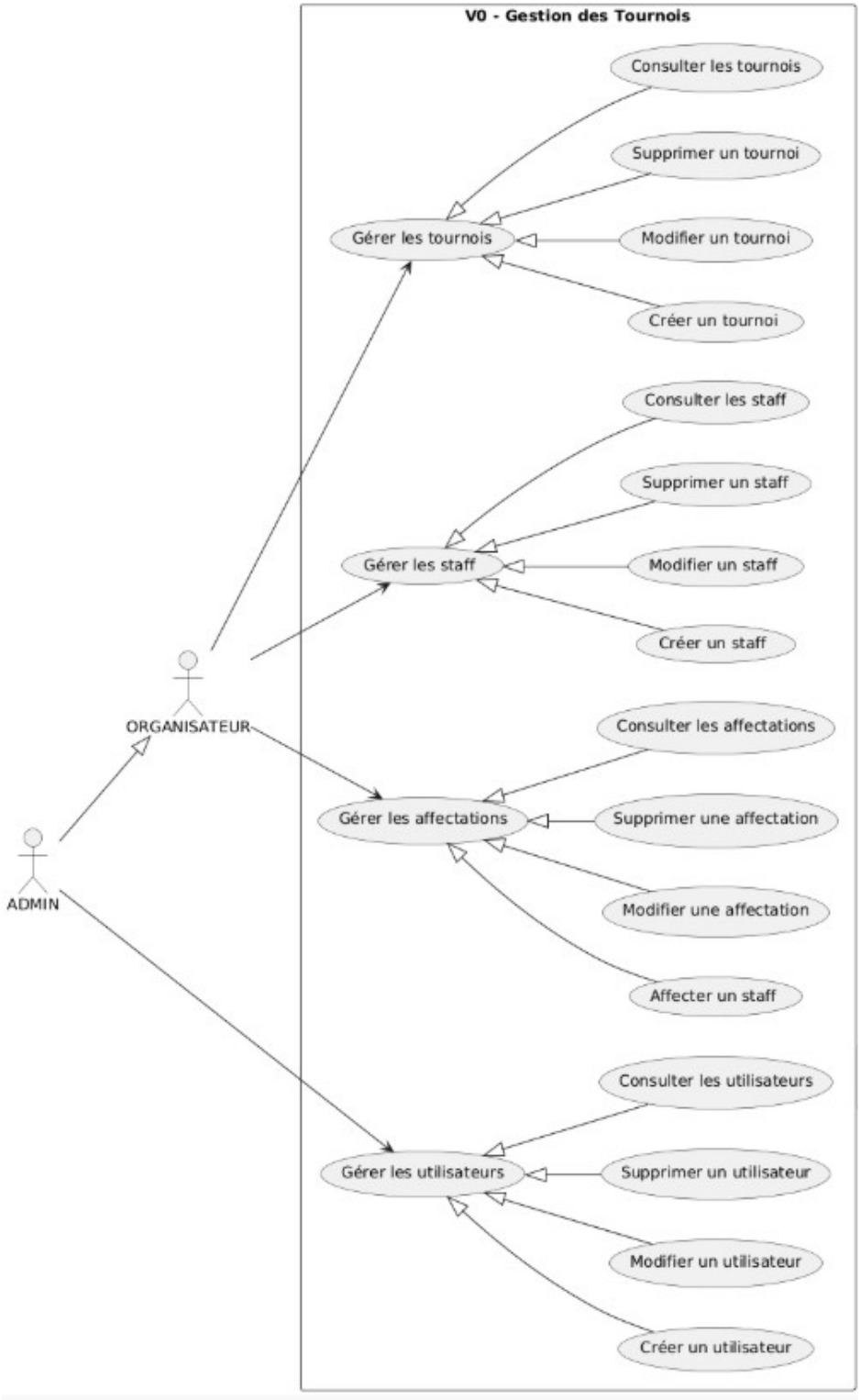


Diagramme use case V0

Ce diagramme de classes présente la structure essentielle des données de la V0 de l'application. Il modélise les entités clés (utilisateurs, équipes, tournois, jeux, staff) avec leurs attributs et met en évidence les relations qui les lient. C'est la fondation informationnelle sur laquelle l'application est construite.

2. Architecture

L'application est construite en plusieurs parties pour faciliter son développement et sa maintenance.

2.1. Structure Générale

L'application est divisée en couches :

- **Interface Utilisateur (UI)** : Ce qui est visible et interactif pour l'utilisateur.
- **Accès aux Données (DAO)** : Gère le stockage et la récupération des informations dans la base de données.
- **Base de Données** : Là où toutes les informations de l'application sont stockées.

2.2. Outils et Modules

L'application utilise des modules externes pour certaines fonctions :

- Des modules pour se connecter à la base de données.
- Des modules pour sécuriser les mots de passe (hachage).
- Des modules pour gérer les dates et heures.

2.3. Organisation du Code

Le code est organisé en dossiers (packages) pour une meilleure structure :

- fr.tournois : Le dossier principal de l'application.
- fr.tournois.model : Contient les définitions des informations (entités) de l'application.
- fr.tournois.dao : Contient les modules qui interagissent avec la base de données.
- fr.tournois.security : Contient les modules liés à la sécurité.
- fr.tournois.ui : Contient les modules de l'interface utilisateur.

3. Détail des Composants de la V0

Cette partie décrit les principaux modules de la V0.

3.1. Module Principal (fr.tournois)

- LiveTournois.java
- **Rôle** : Point de départ de l'application.
- **Description** : Lance l'affichage de l'interface graphique.

3.2. Modules de Données (fr.tournois.model)

Ces modules définissent les types d'informations utilisés dans l'application.

- Affectation.java : Informations sur l'affectation d'un membre du staff à un tournoi.
- Equipe.java : Informations sur une équipe (nom, joueurs).
- Inscription.java : Détails de l'inscription d'une équipe à un tournoi.
- Jeu.java : Informations sur un jeu vidéo.
- Joueur.java : Informations sur un joueur.
- Role.java : Liste des rôles possibles pour un utilisateur (ADMIN, ORGANISATEUR).
- Staff.java : Informations complètes sur un membre du personnel.
- Tournoi.java : Informations complètes sur un tournoi.
- Utilisateur.java : Informations de base d'un utilisateur (pseudo, mot de passe, rôle).

3.3. Modules d'Accès aux Données (fr.tournois.dao)

Ces modules gèrent les échanges avec la base de données pour chaque type d'information.

- AffectationDAO.java : Gère les opérations pour les Affectations.
- ConnectionManager.java : Gère la connexion unique à la base de données.
- DAOException.java : Une erreur spécifique pour les problèmes d'accès aux données.
- EquipeDAO.java : Gère les opérations pour les Equipes.
- JeuDAO.java : Gère les opérations pour les Jeux.
- JoueurDAO.java : Gère les opérations pour les Joueurs.

- StaffDAO.java : Gère les opérations pour les Staffs.
- TournoiDAO.java : Gère les opérations pour les Tournois.
- UtilisateurDAO.java : Gère les opérations pour les Utilisateurs.

```java
import fr.tournois.model.*;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;
import java.util.Optional;

public class AffectationDAO {
    private final Connection connection;

    private static final String FIND_BY_TOURNOI_AND_STAFF_QUERY =
    "SELECT a.*, t.*, s.* FROM Affectation a " +
    "JOIN Tournoi t ON a.id_tournoi = t.id_tournoi " +
    "JOIN Staff s ON a.id_staff = s.id_staff " +
    "WHERE a.id_tournoi = ? AND a.id_staff = ?";
    public AffectationDAO(Connection connection) {
        this.connection = connection;
    }

    public Affectation create(Affectation affectation) throws DAOException {
        validateAffectation(affectation);

        String sql = "INSERT INTO Affectation (id_affectation, id_staff, id_tournoi, role_specifique, date_debut, date_fin) " +
                     "VALUES (seq_affectation_id.NEXTVAL, ?, ?, ?, TO_DATE(?, 'DD/MM/YYYY HH24:MI:SS'), TO_DATE(?, 'DD/MM/YYYY HH24:MI:SS'))";
```

extrait de code de AffectationDAO

3.4. Modules de Sécurité (fr.tournois.security)

Ces modules gèrent les aspects de sécurité de l'application.

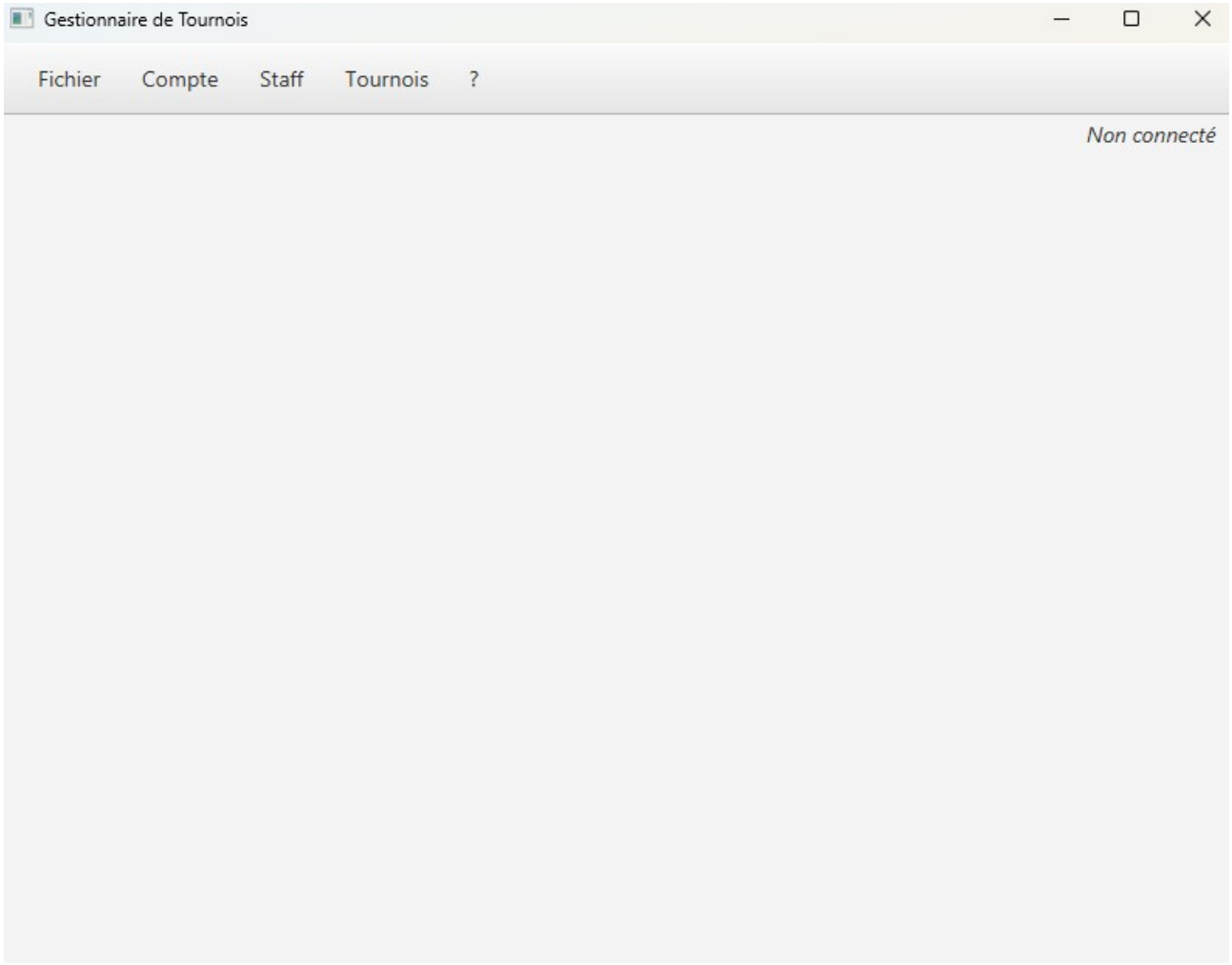
- PasswordHasher.java : Module pour hacher et vérifier les mots de passe de manière sécurisée.
- SecurityContext.java : Gère l'utilisateur connecté et ses droits d'accès.
- TestHash.java : Un module de test pour le hachage de mots de passe.

```java
public class PasswordHasher {
    private static final int COST = 12; // Coût de hachage (plus il est élevé, plus c'est sécurisé mais lent)

    /**
     * Hache un mot de passe en utilisant BCrypt
     * @param password Le mot de passe en clair
     * @return Le mot de passe haché
     */
    public static String hashPassword(String password) {
        return BCrypt.withDefaults().hashToString(COST, password.toCharArray());
    }

    /**
     * Vérifie si un mot de passe en clair correspond à un hash
     * @param password Le mot de passe en clair
     * @param hashedPassword Le hash du mot de passe stocké
     * @return true si le mot de passe correspond, false sinon
     */
    public static boolean verifyPassword(String password, String hashedPassword) {
        return BCrypt.verifyer().verify(password.toCharArray(), hashedPassword).verified;
    }
}
```

extrait de code module sécurité

3.5. Module d'Interface Utilisateur (fr.tournois.ui)

- **Rôle** : Contient l'interface graphique de l'application.
- **Description** : Ce module est responsable de l'affichage et des interactions avec l'utilisateur.



affichage au lancement de l'application

4. Développement et Outils Utilisés

- **Langage de Programmation** : Java.
- **Base de Données** : Une base de données relationnelle.
- **Librairies** : Des librairies pour la sécurité des mots de passe et la gestion des dates.
- **Contrôle de Version** : Git.