

Système de Billetterie et Accréditation Biométrique

pour Grands Événements (Cas : Mondial 2030)

Réalisation : Dounia Chraoua

Promotion : 4IIR-G2

Mémoire de projet / Rapport de réalisation

Table des matières

Résumé	4
Abstract	5
1 Introduction	6
1.1 Contexte	6
1.2 Objectifs	6
2 Description du projet	7
2.1 Vue d'ensemble	7
2.2 Technologies utilisées	7
3 Analyse des besoins	8
3.1 Acteurs	8
3.2 Cas d'utilisation principaux	8
4 Conception	9
4.1 Diagramme de classes	9
4.2 Description des classes (attributs)	9
5 Architecture et base de données	13
5.1 Architecture technique	13
5.2 Migration et XAMPP	13
6 Conclusion	14

7 Bibliographie

15

Table des figures

4.1	Diagramme de classe du système – représentation synthétique des entités et relations	9
-----	--	---

Résumé

Ce rapport présente la conception, le développement et la mise en œuvre d'un **Système de Billetterie et Accréditation Biométrique** destiné aux grands événements, illustré par le cas pratique du *Mondial 2030*. L'objectif principal du système est d'améliorer la sécurité et la fluidité d'accès par l'intégration de la reconnaissance faciale et des empreintes digitales au processus de billetterie, tout en conservant un historique détaillé et des capacités d'analytique (analytics). Le projet a été réalisé en Java, avec une base de données MySQL gérée via XAMPP. Le rapport couvre l'analyse du besoin, la conception (diagramme de classes), l'implémentation, la migration de la base de données et les résultats obtenus.

Abstract

This project report describes the design and implementation of a **Ticketing and Biometric Accreditation System** for large events (case study : World Cup 2030). It integrates facial recognition and fingerprint verification into ticketing to provide secure, rapid entry and analytics. Implemented in Java with a MySQL database using XAMPP, the report details requirements, design (class diagram), implementation and migration, and presents evaluation results.

1. Introduction

1.1. Contexte

L'organisation de grands événements sportifs ou culturels nécessite des solutions robustes pour gérer l'afflux massif de spectateurs tout en garantissant la sécurité. Les systèmes classiques de billetterie confrontent des problèmes de fraude, de falsification et d'engorgement aux points d'accès. L'utilisation de la biométrie (reconnaissance faciale et empreintes digitales) intégrée à la billetterie permet d'authentifier les détenteurs de billets de façon fiable et rapide.

1.2. Objectifs

Les objectifs principaux du projet sont :

- Concevoir une application de billetterie couplée à un module d'accréditation biométrique.
- Permettre l'enregistrement sécurisé des données biométriques (visage, empreinte).
- Offrir une interface d'accès aux agents (validation de billet) et générer des logs d'accès.
- Fournir des rapports analytiques (nombre d'entrées, tentatives de fraude, taux de validation).

2. Description du projet

2.1. Vue d'ensemble

Le système se compose de plusieurs modules :

1. Module de billetterie : gestion des événements, génération et vente de billets.
2. Module biométrie : capture et stockage des embeddings faciaux et des templates d'empreintes.
3. Module d'accréditation : vérification à la porte via périphériques (gate devices).
4. Module de logs et analytics : historique des accès, échecs, statistiques.

2.2. Technologies utilisées

- Langage : **Java** (backend).
- Base de données : **MySQL** (migration et gestion avec **XAMPP**).
- Front-end : Angular
- Bibliothèques : bibliothèques de traitement d'images / reconnaissance faciale (selon l'implémentation choisie), bibliothèques de lecture d'empreintes, JDBC pour accès DB.

3. Analyse des besoins

3.1. Acteurs

- Spectateur / Utilisateur final
- Vendeur / Agent billetterie
- Agent de contrôle / Portier
- Administrateur du système

3.2. Cas d'utilisation principaux

- Inscription du spectateur et enregistrement biométrique
- Achat / génération et envoi du billet
- Validation à l'entrée (scan QR + vérification biométrique)
- Gestion des logs et génération des rapports

4. Conception

4.1. Diagramme de classes

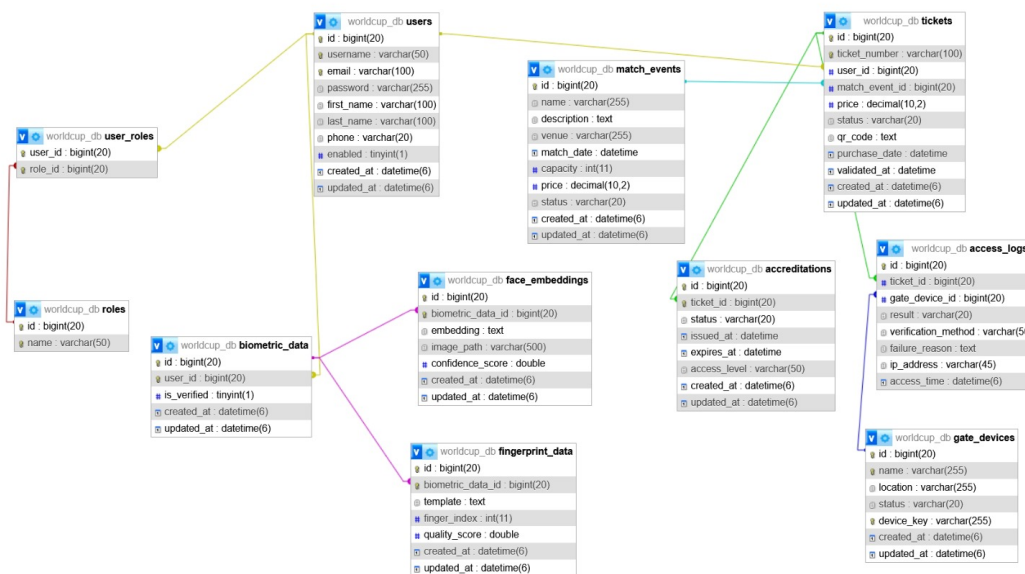


FIGURE 4.1 – Diagramme de classe du système – représentation synthétique des entités et relations

4.2. Description des classes (attributs)

Ci-dessous, la liste des classes principales retenues pour le modèle orienté objet / base de données, avec leurs attributs (une ligne = attribut) :

Classe User

- id : long
- username : String
- email : String
- passwordHash : String
- firstName : String
- lastName : String

- phone : String
- enabled : boolean
- createdAt : DateTime
- updatedAt : DateTime

Classe Role

- id : long
- name : String

Classe UserRole (association)

- userId : long
- roleId : long

Classe BiometricData

- id : long
- userId : long
- isVerified : boolean
- createdAt : DateTime
- updatedAt : DateTime

Classe FaceEmbeddings

- id : long
- biometricDataId : long
- embedding : Text
- imagePath : String
- confidenceScore : double
- createdAt : DateTime
- updatedAt : DateTime

Classe FingerprintData

- id : long
- biometricDataId : long
- template : Text
- fingerIndex : int
- qualityScore : double
- createdAt : DateTime
- updatedAt : DateTime

Classe MatchEvent

- id : long
- name : String
- description : Text
- venue : String
- matchDate : DateTime
- capacity : int
- price : Decimal
- status : String
- createdAt : DateTime
- updatedAt : DateTime

Classe Ticket

- id : long
- ticketNumber : String
- userId : long
- matchEventId : long
- price : Decimal
- status : String
- qrCode : Text
- purchaseDate : DateTime
- validatedAt : DateTime
- createdAt : DateTime
- updatedAt : DateTime

Classe Accreditation

- id : long
- ticketId : long
- status : String
- issuedAt : DateTime
- expiresAt : DateTime
- accessLevel : String
- createdAt : DateTime
- updatedAt : DateTime

Classe GateDevice

- id : long
- name : String
- location : String

- status : String
- deviceKey : String
- createdAt : DateTime
- updatedAt : DateTime

Classe AccessLog

- id : long
- ticketId : long
- gateDeviceId : long
- result : String
- verificationMethod : String
- failureReason : Text
- ipAddress : String
- accessTime : DateTime

Ces classes correspondent directement aux tables de la base MySQL utilisées dans le projet. Elles peuvent être mappées via JPA/Hibernate (si utilisé) ou via DAO/JDBC selon l'architecture retenue.

5. Architecture et base de données

5.1. Architecture technique

Le projet adopte une architecture en couches :

- Présentation : pages web / console d'administration
- Service / Business logic : Java (logique métier)
- Persistance : MySQL (via XAMPP pour le développement local)

5.2. Migration et XAMPP

Pour le développement local et la phase de migration, XAMPP a été utilisé pour héberger MySQL et phpMyAdmin. Les étapes effectuées :

1. Installation XAMPP et démarrage des services Apache et MySQL.
2. Création d'une base 'worldcup_{db}'(*ounomdsir*).
2. Exécution des scripts SQL de création de tables (DDL) via phpMyAdmin ou via ligne de commande.
3. Import initial des données de test (fixtures).
4. Configuration du fichier de connexion JDBC dans l'application Java (URL, utilisateur, mot de passe).

6. Conclusion

Le système développé montre la faisabilité d'une billetterie couplée à des mécanismes biométriques pour sécuriser et accélérer l'accès aux grands événements.

7. Bibliographie