

École Marocaine des Sciences de l'Ingénieur (EMSI)

Rapport de Projet de Fin d'Année

FIFA World Cup 2026 – Unity Hub

Système de reconnaissance hybride et contrôle d'accès multi-niveaux avec authentification forte (MFA)

Réalisé par :

- TAJIRI Wissal
- AMJAD Chaimaa

Encadré par :

- M. LARHLIMI Abderrahim

Soutenu le 02/01/2026, devant les jurys :

- M. LARHLIMI
- M. CHIBA

Filière : Ingénierie Informatique et Réseaux – 5ème année

Année universitaire : 2025–2026

Table des matières

Introduction Générale	6
1 Présentation du cadre de projet	7
1.1 Introduction	7
1.2 Étude de l'existant	7
1.2.1 Authentification par identifiant et mot de passe	8
1.2.2 Authentification à deux facteurs (2FA)	8
1.2.3 Systèmes de contrôle d'accès basés sur les rôles (RBAC)	8
1.2.4 Solutions biométriques existantes	9
1.3 Analyse critique de l'existant	9
1.3.1 Vulnérabilités de l'authentification traditionnelle	9
1.3.2 Limites des approches 2FA actuelles	9
1.3.3 Rigidité des modèles de contrôle d'accès	10
1.3.4 Fragmentation des solutions de sécurité	10
1.4 Solution proposée	10
1.4.1 Principe général	10
1.4.2 Avantages de la solution	11
1.4.3 Limites et contraintes	11
1.5 Choix du modèle de développement	12
1.5.1 Modèles envisagés	12
1.5.2 Justification du choix	12
1.6 Planning prévisionnel	13
1.7 Conclusion	13
2 Spécification des besoins	15
2.1 Introduction	15
2.2 Identification des acteurs	15
2.2.1 Acteurs primaires	16
2.2.2 Acteurs secondaires	17
2.3 Besoins fonctionnels	17
2.3.1 Authentification et gestion de session	17

2.3.2	Authentification multi-facteurs (MFA)	18
2.3.3	Gestion des niveaux d'accès	18
2.3.4	Gestion des ressources	18
2.3.5	Administration et supervision	19
2.4	Besoins non fonctionnels	19
2.4.1	Sécurité	19
2.4.2	Performance	20
2.4.3	Disponibilité et fiabilité	20
2.4.4	Évolutivité et maintenabilité	20
2.4.5	Ergonomie	20
2.5	Cas d'utilisation	21
2.5.1	CU1 : S'authentifier	21
2.5.2	CU2 : Enregistrer son visage	21
2.5.3	CU3 : Accéder à une ressource protégée	22
2.5.4	CU4 : Gérer les utilisateurs	22
2.5.5	CU5 : Configurer l'authentification OTP	23
2.5.6	CU6 : Consulter les journaux d'activité	23
2.6	Diagrammes de cas d'utilisation	23
2.6.1	Diagramme de cas d'utilisation global	24
2.6.2	Diagramme de cas d'utilisation MFA	24
2.7	Conclusion	26
3	Conception du système	27
3.1	Introduction	27
3.2	Vue globale de la conception	27
3.2.1	Démarche de conception adoptée	28
3.2.2	Découpage du système en composants	28
3.3	Modélisation dynamique	28
3.3.1	Diagramme de séquence : Authentification et décision MFA	29
3.3.2	Diagramme de séquence : Validation MFA via OTP	29
3.3.3	Diagramme de séquence : Configuration OTP	30
3.3.4	Diagramme de séquence : Accès refusé à une ressource	31
3.4	Modélisation statique	32
3.4.1	Présentation du diagramme de classes	32
3.4.2	Description des principales classes	33
3.4.3	Relations entre classes	35
3.5	Architecture du système	35
3.5.1	Architecture logicielle	35
3.5.2	Architecture matérielle	37

3.5.3	Protocoles et sécurisation des échanges	39
3.6	Conclusion	39
4	Réalisation du système	40
4.1	Introduction	40
4.2	Environnement matériel	40
4.2.1	Poste de développement	40
4.2.2	Configuration minimale requise	41
4.2.3	Contraintes liées à la biométrie	41
4.3	Environnement logiciel	41
4.3.1	Système d'exploitation	41
4.3.2	Outils de développement	42
4.3.3	Frameworks et technologies	42
4.3.4	Outils de test et de versionnement	42
4.4	Présentation des interfaces graphiques	43
4.4.1	Page d'accueil	43
4.4.2	Interface de connexion	44
4.4.3	Interface de validation MFA	44
4.4.4	Tableau de bord	45
4.4.5	Exigences de niveau d'accès	45
4.4.6	Interface Fan Zone	46
4.4.7	Interface Stadium Operations	47
4.4.8	Interface VIP Hospitality	47
4.4.9	Interface VIP Reservations	48
4.4.10	Interface Event Control Room	49
4.4.11	Interface Team Travel Coordination	49
4.4.12	Interface Documents Vault	50
4.4.13	Interface Predictions League	50
4.5	Conclusion	51
Conclusion Générale		52
Annexes		53
A1 – Table des acronymes	53	
A2 – Extraits techniques illustratifs	53	
A3 – Endpoints API principaux	55	
A4 – Checklist de conformité sécurité	56	

Table des figures

2.1	Diagramme de cas d'utilisation global du système	24
2.2	Diagramme de cas d'utilisation : Authentification multi-facteurs	25
3.1	Diagramme de séquence : Authentification et décision MFA	29
3.2	Diagramme de séquence : Validation MFA via OTP	30
3.3	Diagramme de séquence : Configuration de l'authentification OTP	31
3.4	Diagramme de séquence : Accès refusé à une ressource protégée	32
3.5	Diagramme de classes global du système	33
3.6	Architecture logicielle du système	36
3.7	Architecture matérielle de déploiement	38
4.1	Page d'accueil de la plateforme Unity Hub	43
4.2	Interface de connexion utilisateur	44
4.3	Interface de validation multi-facteurs (MFA)	44
4.4	Tableau de bord utilisateur	45
4.5	Interface d'affichage des exigences de niveau d'accès	46
4.6	Interface Fan Zone – Espace supporters	46
4.7	Interface Stadium Operations – Gestion opérationnelle	47
4.8	Interface VIP Hospitality – Services d'accueil privilégiés	48
4.9	Interface VIP Reservations – Gestion des réservations privilégiées	48
4.10	Interface Event Control Room – Centre de supervision événementielle	49
4.11	Interface Team Travel Coordination – Coordination des déplacements	49
4.12	Interface Documents Vault – Coffre-fort documentaire sécurisé	50
4.13	Interface Predictions League – Espace pronostics et classements	51

Liste des tableaux

1.1	Planning prévisionnel du projet	13
4.1	Outils de développement utilisés	42
4.2	Technologies utilisées par composant	42
4.3	Table des acronymes	53
4.4	Endpoints API principaux	56
4.5	Checklist de conformité sécurité	56

Introduction Générale

La Coupe du Monde FIFA 2026, co-organisée par les États-Unis, le Mexique et le Canada, représente un défi majeur en matière de sécurisation des accès aux systèmes d'information. Face à la sophistication croissante des cybermenaces et à la diversité des profils utilisateurs (spectateurs, personnel, administrateurs), les mécanismes d'authentification traditionnels se révèlent insuffisants. La problématique centrale de ce projet est la suivante : comment concevoir un système de contrôle d'accès hybride combinant authentification multi-facteurs, reconnaissance biométrique et gestion dynamique des niveaux d'autorisation, tout en préservant une expérience utilisateur fluide ?

Pour répondre à ce besoin, nous développons la plateforme **FIFA World Cup 2026 – Unity Hub**, un système de reconnaissance hybride intégrant une authentification forte (mot de passe, OTP, reconnaissance faciale) et un contrôle d'accès granulaire à quatre niveaux d'habilitation. L'architecture repose sur une séparation claire entre frontend React, backend Spring Boot et microservice biométrique FastAPI, garantissant modularité et évolutivité.

Ce rapport s'articule en quatre chapitres : présentation du cadre de projet et étude de l'existant (chapitre 1), spécification des besoins fonctionnels et non fonctionnels (chapitre 2), conception UML et architecture du système (chapitre 3), et réalisation technique avec présentation des interfaces (chapitre 4).

Chapitre 1

Présentation du cadre de projet

1.1 Introduction

La réussite d'un projet de développement logiciel repose en grande partie sur une compréhension approfondie du contexte dans lequel il s'inscrit, des solutions existantes qu'il entend dépasser, et des choix méthodologiques qui guideront sa réalisation. Ce premier chapitre vise précisément à établir ces fondations, en positionnant notre projet dans son environnement technologique et organisationnel.

Nous commençons par dresser un état des lieux des solutions actuelles de contrôle d'accès et d'authentification, en mettant en évidence leurs caractéristiques principales ainsi que leurs limitations dans un contexte à forte exigence sécuritaire. Cette étude de l'existant nous permet ensuite de formuler une analyse critique, identifiant les lacunes auxquelles notre projet propose de répondre.

Dans un second temps, nous présentons la solution envisagée — FIFA World Cup 2026 – Unity Hub — en détaillant son architecture fonctionnelle et ses apports par rapport aux approches traditionnelles. Nous justifions également le choix du modèle de développement retenu, avant de conclure par la présentation du planning prévisionnel qui structure les différentes phases du projet.

Ce chapitre constitue une base conceptuelle essentielle pour la suite du rapport.

1.2 Étude de l'existant

L'authentification et le contrôle d'accès constituent des préoccupations majeures pour tout système d'information moderne. Avant de présenter notre solution, il convient d'examiner les approches actuellement déployées dans l'industrie et d'en comprendre les mécanismes fondamentaux.

1.2.1 Authentification par identifiant et mot de passe

La méthode d'authentification la plus répandue demeure le couple identifiant/mot de passe. Ce mécanisme, hérité des premiers systèmes informatiques, repose sur un principe simple : l'utilisateur prouve son identité en fournissant une information secrète qu'il est censé être le seul à connaître. La plupart des applications web, des systèmes d'exploitation et des services en ligne continuent d'utiliser cette approche comme méthode d'authentification primaire.

D'un point de vue technique, les mots de passe sont généralement stockés sous forme de hachages cryptographiques (bcrypt, Argon2, PBKDF2) afin d'éviter leur exposition en cas de compromission de la base de données. Néanmoins, cette méthode présente des vulnérabilités intrinsèques liées au facteur humain : mots de passe faibles, réutilisation sur plusieurs plateformes, sensibilité aux attaques de type phishing ou ingénierie sociale.

1.2.2 Authentification à deux facteurs (2FA)

Face aux limites de l'authentification simple, l'industrie a progressivement adopté des mécanismes d'authentification à deux facteurs. Le principe consiste à combiner deux éléments de nature différente parmi les trois catégories classiques : ce que l'utilisateur sait (mot de passe), ce qu'il possède (téléphone, token matériel), et ce qu'il est (caractéristiques biométriques).

Les implémentations les plus courantes incluent l'envoi de codes par SMS, l'utilisation d'applications génératrices de codes temporels (TOTP) comme Google Authenticator ou Microsoft Authenticator, ou encore les clés de sécurité physiques conformes au standard FIDO2. Ces solutions renforcent significativement la sécurité en imposant la compromission de deux facteurs distincts pour réussir une usurpation d'identité.

1.2.3 Systèmes de contrôle d'accès basés sur les rôles (RBAC)

En parallèle des mécanismes d'authentification, les organisations déploient des systèmes de contrôle d'accès pour déterminer les ressources auxquelles chaque utilisateur peut accéder. Le modèle RBAC (Role-Based Access Control) constitue l'approche la plus répandue : les permissions sont attribuées à des rôles, et les utilisateurs se voient assigner un ou plusieurs rôles en fonction de leurs responsabilités.

Ce modèle offre une gestion centralisée et relativement simple des autorisations, particulièrement adaptée aux organisations avec des hiérarchies bien définies. Cependant, il manque de flexibilité pour gérer des contextes dynamiques où le niveau d'accès devrait varier en fonction de paramètres circonstanciels.

1.2.4 Solutions biométriques existantes

L'authentification biométrique s'est progressivement démocratisée, notamment grâce à l'intégration de capteurs d'empreintes digitales et de caméras pour la reconnaissance faciale dans les appareils mobiles. Des technologies comme Face ID d'Apple ou Windows Hello de Microsoft ont contribué à familiariser le grand public avec ces mécanismes.

Dans le domaine professionnel, des solutions telles que les systèmes de contrôle d'accès physique par reconnaissance faciale ou par empreinte palmaire sont déployées dans les environnements sensibles. Ces systèmes reposent généralement sur des algorithmes d'apprentissage profond capables d'extraire et de comparer des caractéristiques biométriques uniques.

1.3 Analyse critique de l'existant

L'examen des solutions présentées précédemment révèle plusieurs limitations qui justifient le développement d'une approche plus intégrée et adaptative.

1.3.1 Vulnérabilités de l'authentification traditionnelle

L'authentification par mot de passe seul présente des faiblesses documentées et récurrentes. Les études de sécurité montrent que la majorité des compromissions de comptes résultent de mots de passe faibles ou réutilisés. Les attaques par credential stuffing, exploitant des bases de données de mots de passe compromis, représentent une menace constante pour les systèmes reposant uniquement sur ce mécanisme.

Par ailleurs, les techniques de phishing se sont considérablement sophistiquées, rendant difficile pour les utilisateurs non avertis de distinguer une tentative d'hameçonnage d'une communication légitime. Dans un contexte d'événement international comme la Coupe du Monde, où des millions d'utilisateurs interagissent avec des systèmes numériques, cette vulnérabilité devient particulièrement critique.

1.3.2 Limites des approches 2FA actuelles

Si l'authentification à deux facteurs renforce indéniablement la sécurité, les implantations courantes présentent leurs propres limitations. L'envoi de codes par SMS, bien que largement déployé, est vulnérable aux attaques de type SIM swapping. Les applications TOTP, plus sécurisées, peuvent être contournées par des attaques en temps réel de type man-in-the-middle.

De plus, ces mécanismes sont souvent appliqués de manière uniforme, sans tenir compte du contexte de la connexion ni de la sensibilité des ressources sollicitées. Un utilisateur accédant à des informations publiques depuis un appareil reconnu subit les mêmes contraintes

d'authentification que celui tentant d'accéder à des données confidentielles depuis un terminal inconnu.

1.3.3 Rigidité des modèles de contrôle d'accès

Les systèmes RBAC traditionnels, bien qu'efficaces pour des organisations stables, peinent à s'adapter à des contextes dynamiques. Dans le cadre d'un événement comme la Coupe du Monde, les besoins d'accès évoluent rapidement : accréditations temporaires, niveaux de priviléges variables selon les phases de l'événement, et nécessité de réagir promptement aux incidents de sécurité.

La rigidité de ces modèles conduit souvent à une attribution excessive de permissions (principe du moindre privilège non respecté) ou, à l'inverse, à des blocages opérationnels lorsque les droits d'accès ne correspondent pas aux besoins réels des utilisateurs.

1.3.4 Fragmentation des solutions de sécurité

Un constat récurrent dans les organisations concerne la fragmentation des solutions de sécurité. L'authentification, le contrôle d'accès et la biométrie sont souvent gérés par des systèmes distincts, peu ou mal intégrés. Cette situation engendre des incohérences, des failles potentielles aux interfaces entre systèmes, et une expérience utilisateur dégradée par la multiplication des procédures d'identification.

1.4 Solution proposée

Face aux limitations identifiées, nous proposons le développement de la plateforme **FIFA World Cup 2026 – Unity Hub**, un système intégré de reconnaissance hybride et de contrôle d'accès multi-niveaux.

1.4.1 Principe général

Notre solution repose sur trois piliers fondamentaux intégrés au sein d'une architecture unifiée :

1. **Authentification multi-facteurs adaptative** : le système évalue dynamiquement le niveau de risque associé à chaque tentative de connexion et ajuste les exigences d'authentification en conséquence. Pour un accès à faible risque, une authentification simple peut suffire ; pour des ressources sensibles ou des connexions suspectes, des facteurs supplémentaires sont requis.
2. **Reconnaissance biométrique avancée** : l'intégration d'un module de reconnaissance faciale basé sur des algorithmes d'apprentissage profond permet une vérification

robuste de l'identité de l'utilisateur, difficilement falsifiable contrairement aux facteurs de connaissance ou de possession.

3. **Gestion granulaire des niveaux d'accès** : le système définit plusieurs niveaux d'habilitation (basique, étendu, sensible, administration), chacun associé à des exigences d'authentification spécifiques et à un périmètre de ressources accessibles.

1.4.2 Avantages de la solution

La plateforme Unity Hub présente plusieurs avantages significatifs par rapport aux approches existantes :

- **Sécurité renforcée** : la combinaison de multiples facteurs d'authentification, incluant la biométrie, élève considérablement le niveau de protection contre les tentatives d'intrusion et d'usurpation d'identité.
- **Adaptabilité contextuelle** : l'évaluation dynamique du risque permet d'ajuster les contraintes d'authentification au contexte, offrant un équilibre entre sécurité et ergonomie.
- **Architecture intégrée** : l'unification des fonctions d'authentification, de biométrie et de contrôle d'accès au sein d'une même plateforme élimine les incohérences et les vulnérabilités liées aux interfaces entre systèmes disparates.
- **Traçabilité complète** : la journalisation centralisée de tous les événements d'accès facilite l'audit de sécurité et la détection d'anomalies.
- **Modularité technique** : l'architecture en microservices (backend Spring Boot, service biométrique FastAPI, frontend React) favorise l'évolutivité et la maintenance du système.

1.4.3 Limites et contraintes

Nous identifions néanmoins certaines limites inhérentes à notre approche :

- **Dépendance matérielle** : la reconnaissance faciale requiert un équipement de capture d'image de qualité suffisante, ce qui peut poser des difficultés dans certains environnements.
- **Complexité d'intégration** : l'architecture distribuée, bien que modulaire, implique une coordination accrue entre les différents services et une gestion rigoureuse des communications inter-services.
- **Considérations relatives à la vie privée** : le stockage et le traitement de données biométriques soulèvent des questions de conformité réglementaire (RGPD) et de protection de la vie privée, nécessitant des mesures de sécurisation et d'anonymisation appropriées.

1.5 Choix du modèle de développement

Le choix du modèle de développement constitue une décision structurante pour la conduite du projet. Nous présentons ici les principales options envisagées et justifions notre choix.

1.5.1 Modèles envisagés

Plusieurs modèles de développement logiciel ont été considérés :

- **Modèle en cascade** : approche séquentielle où chaque phase (analyse, conception, développement, tests) doit être entièrement achevée avant de passer à la suivante. Ce modèle offre une structure claire mais manque de flexibilité face aux évolutions des besoins.
- **Modèle en V** : variante du modèle en cascade intégrant une correspondance entre phases de développement et phases de validation. Il renforce l'assurance qualité mais conserve la rigidité de l'approche séquentielle.
- **Modèle itératif incrémental** : approche cyclique où le système est développé par incrément successifs, chaque itération ajoutant de nouvelles fonctionnalités. Ce modèle permet une validation progressive et une adaptation aux retours d'expérience.
- **Méthodologies agiles** : famille d'approches (Scrum, Kanban, XP) privilégiant la collaboration, l'adaptation au changement et la livraison fréquente de fonctionnalités opérationnelles.

1.5.2 Justification du choix

Nous avons retenu une **approche agile de type Scrum**, adaptée aux spécificités de notre projet. Ce choix se justifie par plusieurs considérations :

1. **Nature évolutive des besoins** : dans le contexte d'un système de sécurité, les exigences peuvent évoluer rapidement en fonction des menaces identifiées et des retours opérationnels. L'approche agile permet d'intégrer ces évolutions sans remettre en cause l'ensemble du planning.
2. **Complexité technique** : l'intégration de technologies variées (biométrie, authentification forte, microservices) implique une part d'incertitude technique. Le développement itératif permet de valider progressivement les choix technologiques et d'ajuster l'architecture si nécessaire.
3. **Besoin de validation précoce** : la livraison incrémentale de fonctionnalités opérationnelles permet aux parties prenantes de valider régulièrement la conformité du système avec leurs attentes, réduisant ainsi les risques de divergence entre le produit final et les besoins réels.

4. **Contraintes temporelles** : le cadre académique du projet impose un calendrier constraint. L'organisation en sprints de durée fixe facilite le suivi de l'avancement et l'identification rapide des éventuels retards.

1.6 Planning prévisionnel

La conduite du projet s'organise autour de phases structurées, présentées dans le tableau 1.1. Ce planning prévisionnel fixe les jalons principaux et les livrables attendus à chaque étape.

TABLE 1.1 – Planning prévisionnel du projet

Phase	Description
Analyse	Étude de l'existant, recueil des besoins, identification des acteurs et des cas d'utilisation
Spécification	Formalisation des besoins fonctionnels et non fonctionnels, rédaction des spécifications
Conception	Modélisation UML (cas d'utilisation, séquences, classes), définition de l'architecture
Développement Backend	Implémentation des services Spring Boot, gestion des utilisateurs, authentification MFA
Développement Biométrie	Développement du service de reconnaissance faciale avec FastAPI et TensorFlow
Développement Frontend	Implémentation de l'interface utilisateur React + TypeScript
Intégration	Intégration des différents modules, tests d'intégration
Tests et Validation	Tests fonctionnels, tests de sécurité, correction des anomalies
Documentation	Rédaction du rapport, préparation de la soutenance

Nous observons que certaines phases se chevauchent volontairement, reflétant la nature itérative de notre approche de développement. Le développement parallèle des composants backend, biométrie et frontend optimise l'utilisation des ressources tout en permettant une intégration progressive.

1.7 Conclusion

Ce premier chapitre a permis de poser les fondations du projet FIFA World Cup 2026 – Unity Hub. L'étude de l'existant a mis en lumière les mécanismes d'authentification

et de contrôle d'accès couramment déployés, tandis que l'analyse critique a révélé leurs limitations face aux exigences d'un événement d'envergure internationale.

La solution proposée, combinant authentification multi-facteurs adaptative, reconnaissance biométrique et gestion granulaire des niveaux d'accès, répond à ces insuffisances en offrant un niveau de sécurité renforcé sans sacrifier l'ergonomie. Le choix d'une méthodologie agile garantit la flexibilité nécessaire pour s'adapter aux évolutions des besoins et aux contraintes techniques.

Le chapitre suivant sera consacré à la spécification détaillée des besoins, où nous présenterons les exigences fonctionnelles et non fonctionnelles du système, les acteurs impliqués et les principaux cas d'utilisation.

Chapitre 2

Spécification des besoins

2.1 Introduction

La spécification des besoins constitue une étape fondamentale dans tout projet de développement logiciel. Elle permet de formaliser les attentes des parties prenantes et de définir précisément le périmètre fonctionnel du système à concevoir. Cette phase d'analyse établit un référentiel commun entre les concepteurs et les utilisateurs finaux, réduisant ainsi les risques d'incompréhension et de divergence entre le produit livré et les besoins réels.

Dans le cadre du projet FIFA World Cup 2026 – Unity Hub, la spécification des besoins revêt une importance particulière en raison de la criticité des fonctions de sécurité à implémenter. Nous devons garantir que les exigences en matière d'authentification forte, de contrôle d'accès multi-niveaux et de reconnaissance biométrique sont exhaustivement définies et correctement comprises.

Ce chapitre présente successivement les acteurs interagissant avec le système, les besoins fonctionnels regroupés par domaine, les exigences non fonctionnelles, et enfin une description textuelle des principaux cas d'utilisation. Cette formalisation servira de base à la phase de conception présentée dans le chapitre suivant.

2.2 Identification des acteurs

L'identification des acteurs constitue le point de départ de l'analyse fonctionnelle. Un acteur représente toute entité externe — humaine ou système — interagissant avec l'application. Nous distinguons les acteurs primaires, qui utilisent directement les fonctionnalités du système, des acteurs secondaires, qui fournissent des services nécessaires au fonctionnement de l'application.

2.2.1 Acteurs primaires

Visiteur

Le visiteur désigne tout utilisateur accédant à la plateforme sans être authentifié. Son périmètre d'interaction est limité aux fonctionnalités publiques du système.

- **Rôle :** Utilisateur non authentifié découvrant la plateforme.
- **Interactions principales :**
 - Consulter la page d'accueil et les informations publiques.
 - Accéder au formulaire de connexion.
 - Initier une demande de création de compte.

Utilisateur authentifié

L'utilisateur authentifié représente une personne ayant validé le processus d'authentification multi-facteurs. Selon son niveau d'habilitation, il accède à un ensemble de ressources plus ou moins étendu.

- **Rôle :** Utilisateur disposant d'un compte validé et de droits d'accès correspondant à son niveau.
- **Interactions principales :**
 - S'authentifier via le processus MFA (mot de passe, OTP, reconnaissance faciale).
 - Consulter les ressources autorisées selon son niveau d'accès.
 - Gérer son profil et ses paramètres de sécurité.
 - Enregistrer ses données biométriques (visage).
 - Consulter l'historique de ses connexions.

Administrateur

L'administrateur dispose de priviléges étendus lui permettant de gérer l'ensemble des utilisateurs, des ressources et des paramètres de sécurité du système.

- **Rôle :** Responsable de la gestion et de la supervision de la plateforme.
- **Interactions principales :**
 - Gérer les comptes utilisateurs (création, modification, désactivation).
 - Attribuer et modifier les niveaux d'accès.
 - Configurer les ressources et leurs exigences d'authentification.
 - Consulter les journaux d'activité et les alertes de sécurité.
 - Définir les politiques de sécurité globales.

2.2.2 Acteurs secondaires

Service OTP

Le service OTP représente le composant générant et validant les codes à usage unique dans le cadre de l'authentification multi-facteurs.

- **Rôle** : Fournir un mécanisme de génération et de vérification de codes temporels.
- **Interactions principales** :
 - Générer un secret OTP lors de l'activation de la double authentification.
 - Valider les codes TOTP soumis par l'utilisateur.
 - Synchroniser les paramètres avec les applications tierces (Google Authenticator).

Service Biométrie

Le service biométrie assure les fonctions de reconnaissance faciale, de l'enregistrement des caractéristiques biométriques jusqu'à leur vérification lors de l'authentification.

- **Rôle** : Capturer, stocker et comparer les données biométriques faciales.
- **Interactions principales** :
 - Capturer et encoder les caractéristiques faciales lors de l'enrôlement.
 - Stocker les embeddings biométriques de manière sécurisée.
 - Comparer une capture en temps réel avec les données enregistrées.
 - Retourner un score de confiance au système principal.

2.3 Besoins fonctionnels

Les besoins fonctionnels décrivent les services que le système doit fournir aux utilisateurs. Nous les présentons par domaine fonctionnel, avec une numérotation permettant leur traçabilité tout au long du projet.

2.3.1 Authentification et gestion de session

BF1 Authentification par identifiant et mot de passe : Le système doit permettre à un utilisateur de s'authentifier en saisissant son adresse email et son mot de passe. Le mot de passe doit être vérifié contre une version hachée stockée en base de données.

BF2 Gestion des sessions : Le système doit créer une session utilisateur après authentification réussie, matérialisée par un token JWT. La durée de validité du token doit être configurable.

BF3 Déconnexion : Le système doit permettre à l'utilisateur de mettre fin à sa session de manière explicite, invalidant le token JWT associé.

BF4 Récupération de mot de passe : Le système doit proposer un mécanisme de réinitialisation de mot de passe via l'envoi d'un lien sécurisé à l'adresse email de l'utilisateur.

2.3.2 Authentification multi-facteurs (MFA)

BF5 Activation de l'authentification OTP : Le système doit permettre à l'utilisateur d'activer l'authentification par code à usage unique, en générant un secret compatible avec les applications TOTP (Google Authenticator).

BF6 Vérification du code OTP : Lors de la connexion, si l'OTP est activé, le système doit demander la saisie d'un code temporel et vérifier sa validité avant d'accorder l'accès.

BF7 Enrôlement biométrique : Le système doit permettre à l'utilisateur d'enregistrer son visage via une capture photographique, dont les caractéristiques seront encodées et stockées.

BF8 Vérification biométrique : Le système doit permettre de valider l'identité de l'utilisateur par reconnaissance faciale en comparant une capture en temps réel avec les données biométriques enregistrées.

BF9 Séquencement MFA adaptatif : Le système doit déterminer dynamiquement les facteurs d'authentification requis en fonction du niveau de sensibilité des ressources sollicitées et du profil de risque de la connexion.

2.3.3 Gestion des niveaux d'accès

BF10 Définition des niveaux : Le système doit supporter plusieurs niveaux d'accès hiérarchiques (LEVEL_1, LEVEL_2, LEVEL_3, ADMIN), chacun associé à des exigences d'authentification spécifiques.

BF11 Attribution des niveaux : Le système doit permettre à un administrateur d'attribuer un niveau d'accès à chaque utilisateur.

BF12 Modification des niveaux : Le système doit permettre la modification du niveau d'accès d'un utilisateur, avec prise d'effet immédiate.

BF13 Contrôle d'accès aux ressources : Le système doit vérifier, pour chaque ressource sollicitée, que le niveau d'accès de l'utilisateur est suffisant.

2.3.4 Gestion des ressources

BF14 Catalogue des ressources : Le système doit maintenir un catalogue des ressources accessibles, chacune associée à un niveau d'accès minimum requis.

BF15 Affichage conditionnel : Le système doit afficher à l'utilisateur uniquement les ressources correspondant à son niveau d'habilitation actuel.

BF16 Administration des ressources : Le système doit permettre à un administrateur d'ajouter, modifier ou supprimer des ressources du catalogue.

2.3.5 Administration et supervision

BF17 Gestion des utilisateurs : Le système doit permettre à un administrateur de lister, créer, modifier et désactiver des comptes utilisateurs.

BF18 Journalisation des accès : Le système doit enregistrer l'ensemble des tentatives de connexion (réussies ou échouées), avec horodatage, adresse IP et résultat.

BF19 Tableau de bord administrateur : Le système doit fournir une interface de supervision présentant les statistiques clés (utilisateurs actifs, tentatives de connexion, alertes).

BF20 Alertes de sécurité : Le système doit générer des alertes en cas de comportements suspects (tentatives multiples échouées, connexion depuis une localisation inhabituelle).

2.4 Besoins non fonctionnels

Les besoins non fonctionnels définissent les caractéristiques qualitatives du système, indépendamment des fonctionnalités spécifiques. Ils constituent des critères d'acceptation transversaux.

2.4.1 Sécurité

BNF1 Chiffrement des communications : Toutes les communications entre le client et les serveurs doivent être chiffrées via HTTPS (TLS 1.2 ou supérieur).

BNF2 Stockage sécurisé des mots de passe : Les mots de passe doivent être hachés avec un algorithme robuste (bcrypt) avant stockage.

BNF3 Protection des données biométriques : Les embeddings biométriques doivent être chiffrés au repos et leur accès restreint aux seuls processus de vérification.

BNF4 Protection contre les attaques courantes : Le système doit être protégé contre les injections SQL, les attaques XSS et CSRF.

BNF5 Limitation des tentatives : Le système doit implémenter un mécanisme de verrouillage temporaire après un nombre configurable de tentatives d'authentification échouées.

2.4.2 Performance

BNF6 Temps de réponse : Les opérations d'authentification standard doivent s'exécuter en moins de 2 secondes.

BNF7 Temps de vérification biométrique : La reconnaissance faciale doit produire un résultat en moins de 3 secondes.

BNF8 Capacité de charge : Le système doit supporter au minimum 100 utilisateurs simultanés sans dégradation notable des performances.

2.4.3 Disponibilité et fiabilité

BNF9 Taux de disponibilité : Le système doit viser un taux de disponibilité de 99% hors maintenance planifiée.

BNF10 Tolérance aux pannes : L'indisponibilité d'un service secondaire (biométrie) ne doit pas bloquer l'ensemble du processus d'authentification, mais proposer une alternative.

BNF11 Sauvegarde des données : Les données utilisateurs et journaux doivent faire l'objet de sauvegardes régulières.

2.4.4 Évolutivité et maintenabilité

BNF12 Architecture modulaire : Le système doit être conçu de manière modulaire, permettant l'ajout de nouveaux facteurs d'authentification sans refonte majeure.

BNF13 Documentation technique : Le code source doit être documenté, et une documentation d'API doit être maintenue à jour.

BNF14 Tests automatisés : Les fonctionnalités critiques doivent être couvertes par des tests unitaires et d'intégration.

2.4.5 Ergonomie

BNF15 Interface intuitive : L'interface utilisateur doit être claire et ne pas nécessiter de formation préalable pour les opérations courantes.

BNF16 Retour visuel : Le système doit fournir un retour visuel clair à chaque étape du processus d'authentification.

BNF17 Accessibilité : L'interface doit respecter les principes d'accessibilité (contraste suffisant, navigation clavier).

BNF18 Responsive design : L'application web doit s'adapter aux différentes tailles d'écran (desktop, tablette, mobile).

2.5 Cas d'utilisation

Les cas d'utilisation décrivent les interactions entre les acteurs et le système pour atteindre un objectif spécifique. Nous présentons ici une description textuelle des cas d'utilisation principaux. Les diagrammes UML correspondants seront présentés dans le chapitre de conception.

Les cas d'utilisation présentés ci-dessous illustrent les interactions principales entre les acteurs et le système dans des scénarios représentatifs.

2.5.1 CU1 : S'authentifier

Acteur principal Visiteur

Préconditions L'utilisateur dispose d'un compte actif dans le système.

Scénario nominal

1. Le visiteur accède à la page de connexion.
2. Le visiteur saisit son adresse email et son mot de passe.
3. Le système vérifie les identifiants.
4. Le système détermine les facteurs MFA requis selon le niveau de l'utilisateur.
5. Si OTP requis : le système demande le code, l'utilisateur le saisit, le système vérifie.
6. Si biométrie requise : le système demande une capture faciale, l'utilisateur se positionne, le système vérifie.
7. Le système génère un token JWT et redirige vers le tableau de bord.

Scénarios alternatifs

- Identifiants incorrects : le système affiche un message d'erreur et incrémente le compteur de tentatives.
- Code OTP invalide : le système demande une nouvelle saisie.
- Reconnaissance faciale échouée : le système propose une nouvelle tentative ou une méthode alternative.

Postconditions L'utilisateur est authentifié et dispose d'une session active.

2.5.2 CU2 : Enregistrer son visage

Acteur principal Utilisateur authentifié

Préconditions L'utilisateur est connecté et n'a pas encore enregistré de données biométriques.

Scénario nominal

1. L'utilisateur accède à la section de gestion de profil.

2. L'utilisateur sélectionne l'option d'enregistrement biométrique.
3. Le système active la caméra et affiche un aperçu.
4. L'utilisateur positionne son visage dans le cadre indiqué.
5. Le système capture l'image et extrait les caractéristiques faciales.
6. Le système stocke les embeddings biométriques.
7. Le système confirme l'enregistrement réussi.

Scénarios alternatifs

- Qualité insuffisante : le système demande une nouvelle capture.
- Visage non détecté : le système affiche des instructions de positionnement.

Postconditions Les données biométriques de l'utilisateur sont enregistrées.

2.5.3 CU3 : Accéder à une ressource protégée

Acteur principal Utilisateur authentifié

Préconditions L'utilisateur dispose d'une session active.

Scénario nominal

1. L'utilisateur consulte le catalogue des ressources.
2. L'utilisateur sélectionne une ressource.
3. Le système vérifie le niveau d'accès de l'utilisateur.
4. Le système affiche le contenu de la ressource.

Scénarios alternatifs

- Niveau insuffisant : le système affiche un message indiquant le niveau requis.

Postconditions L'utilisateur a consulté la ressource demandée.

2.5.4 CU4 : Gérer les utilisateurs

Acteur principal Administrateur

Préconditions L'administrateur est authentifié avec les priviléges appropriés.

Scénario nominal

1. L'administrateur accède au panneau de gestion des utilisateurs.
2. L'administrateur consulte la liste des utilisateurs.
3. L'administrateur sélectionne un utilisateur.
4. L'administrateur modifie les attributs (niveau d'accès, statut).
5. Le système enregistre les modifications.
6. Le système confirme la mise à jour.

Scénarios alternatifs

- Création d'utilisateur : l'administrateur remplit le formulaire de création.
- Désactivation : l'administrateur change le statut à inactif.

Postconditions Les modifications sont effectives immédiatement.

2.5.5 CU5 : Configurer l'authentification OTP

Acteur principal Utilisateur authentifié

Préconditions L'utilisateur est connecté et l'OTP n'est pas encore activé.

Scénario nominal

1. L'utilisateur accède aux paramètres de sécurité.
2. L'utilisateur sélectionne l'activation de l'OTP.
3. Le système génère un secret et affiche un QR code.
4. L'utilisateur scanne le QR code avec Google Authenticator.
5. L'utilisateur saisit le code affiché par l'application pour confirmer.
6. Le système valide et active l'authentification OTP.

Scénarios alternatifs

- Code de confirmation invalide : le système demande une nouvelle tentative.

Postconditions L'authentification OTP est activée pour l'utilisateur.

2.5.6 CU6 : Consulter les journaux d'activité

Acteur principal Administrateur

Préconditions L'administrateur est authentifié.

Scénario nominal

1. L'administrateur accède à la section des journaux.
2. Le système affiche les événements récents.
3. L'administrateur applique des filtres (date, utilisateur, type d'événement).
4. Le système affiche les résultats filtrés.

Postconditions L'administrateur a consulté les journaux souhaités.

2.6 Diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation offrent une représentation graphique synthétique des interactions entre les acteurs et le système. Ils permettent de visualiser l'ensemble des fonctionnalités offertes et leur accessibilité selon les profils utilisateurs.

2.6.1 Diagramme de cas d'utilisation global

Le diagramme de cas d'utilisation global présenté en figure 2.1 illustre l'ensemble des fonctionnalités du système et les acteurs qui y accèdent.

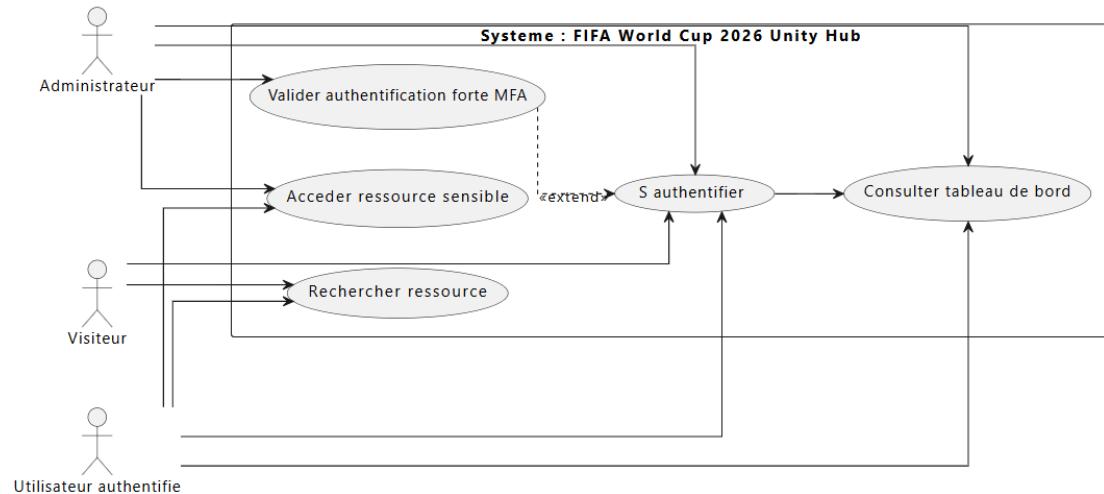


FIGURE 2.1 – Diagramme de cas d'utilisation global du système

Ce diagramme met en évidence la répartition des fonctionnalités entre les différents acteurs. Le visiteur peut accéder aux informations publiques et initier une connexion. L'utilisateur authentifié dispose des fonctionnalités de gestion de profil, de configuration MFA et d'accès aux ressources selon son niveau. L'administrateur bénéficie de priviléges étendus pour la gestion des utilisateurs et la supervision du système.

2.6.2 Diagramme de cas d'utilisation MFA

Le diagramme présenté en figure 2.2 détaille spécifiquement les cas d'utilisation liés au processus d'authentification multi-facteurs.

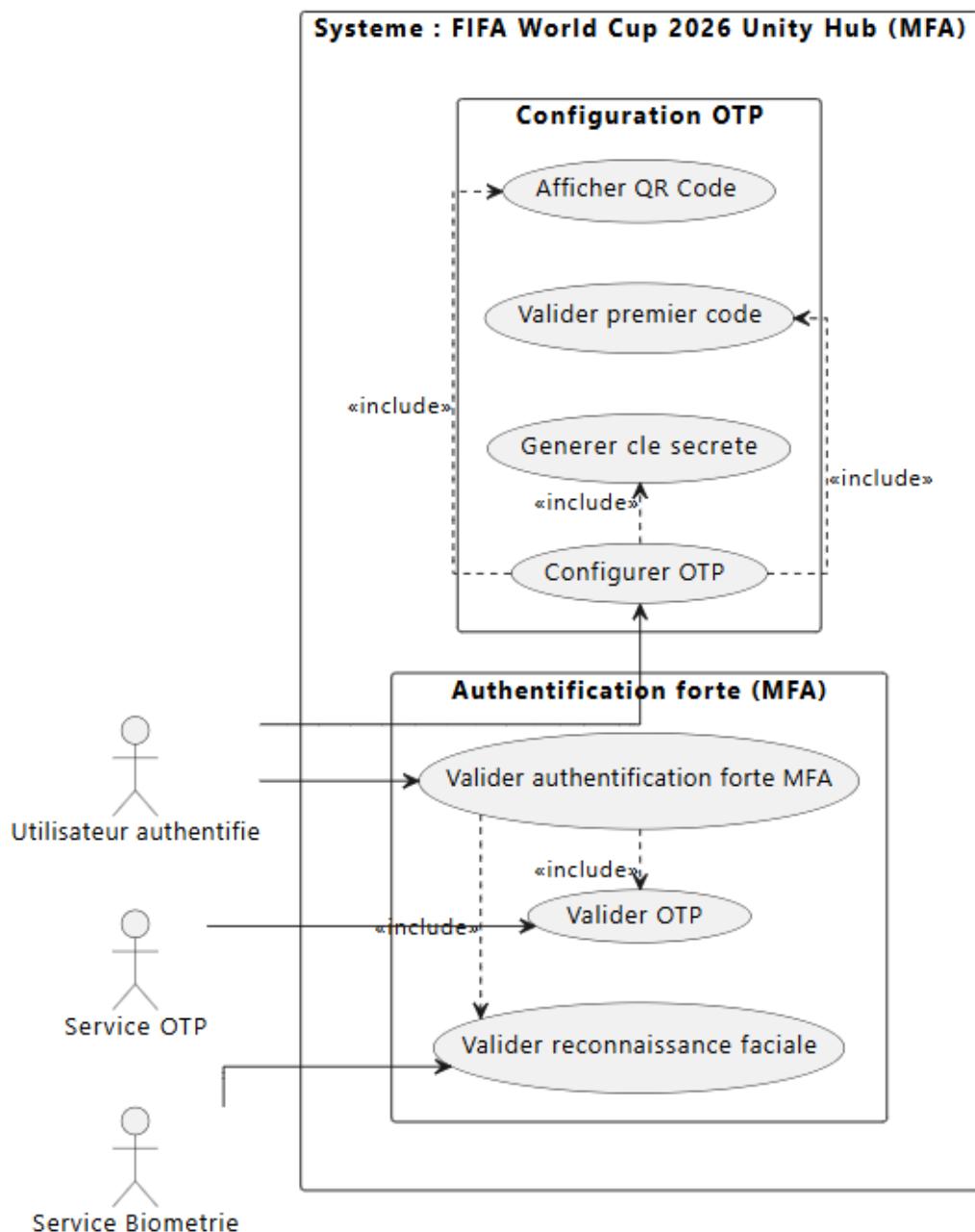


FIGURE 2.2 – Diagramme de cas d’utilisation : Authentification multi-facteurs

Ce diagramme illustre les différentes étapes et options du processus MFA : authentification par mot de passe, validation OTP, vérification biométrique, ainsi que les interactions avec les services secondaires (Service OTP et Service Biométrie). Il met en évidence les relations d’inclusion et d’extension entre les cas d’utilisation.

2.7 Conclusion

Ce chapitre a permis de formaliser l'ensemble des exigences du système FIFA World Cup 2026 – Unity Hub. Nous avons identifié cinq acteurs principaux et secondaires, chacun caractérisé par son rôle et ses interactions avec la plateforme. Les vingt besoins fonctionnels définis couvrent les domaines de l'authentification, de la gestion MFA, du contrôle d'accès multi-niveaux, de l'administration des ressources et de la supervision. Les dix-huit besoins non fonctionnels établissent les critères de qualité attendus en matière de sécurité, de performance, de disponibilité, d'évolutivité et d'ergonomie.

La description textuelle des six cas d'utilisation principaux fournit une vision concrète des interactions entre utilisateurs et système. Ces spécifications constituent le socle sur lequel reposera la phase de conception, présentée dans le chapitre suivant, où nous modéliserons la structure et le comportement du système à l'aide des diagrammes UML.

Chapitre 3

Conception du système

3.1 Introduction

La phase de conception constitue une étape charnière dans le cycle de développement logiciel, assurant la transition entre l'expression des besoins et leur implémentation technique. Elle vise à définir l'architecture du système, à modéliser son comportement dynamique et sa structure statique, et à établir les fondations sur lesquelles reposera la phase de réalisation.

Dans le cadre du projet FIFA World Cup 2026 – Unity Hub, la conception revêt une importance particulière en raison de la complexité des mécanismes de sécurité à implémenter. L'authentification multi-facteurs, la reconnaissance biométrique et la gestion granulaire des niveaux d'accès nécessitent une modélisation rigoureuse garantissant la cohérence et la robustesse du système final.

Ce chapitre présente successivement la démarche de conception adoptée, la modélisation dynamique à travers les diagrammes de séquence UML, la modélisation statique via le diagramme de classes, et enfin l'architecture logicielle et matérielle retenue. Cette formalisation s'appuie sur les spécifications établies au chapitre précédent et prépare la phase de réalisation détaillée dans le chapitre suivant.

3.2 Vue globale de la conception

La conception du système FIFA World Cup 2026 – Unity Hub s'inscrit dans une démarche structurée, guidée par les principes de modularité, de séparation des préoccupations et de réutilisabilité. Nous avons adopté une approche orientée objet, conformément aux pratiques actuelles de l'ingénierie logicielle.

3.2.1 Démarche de conception adoptée

Notre démarche de conception repose sur le formalisme UML (Unified Modeling Language), standard de facto pour la modélisation des systèmes orientés objet. Ce langage graphique permet de représenter différentes perspectives du système à travers des diagrammes complémentaires, facilitant la communication entre les membres de l'équipe et la validation des choix architecturaux.

Nous distinguons deux types de modélisation :

- **Modélisation dynamique** : elle décrit le comportement du système dans le temps, les interactions entre objets et les séquences d'opérations. Les diagrammes de séquence constituent l'outil privilégié pour cette modélisation.
- **Modélisation statique** : elle représente la structure du système, les classes qui le composent, leurs attributs, leurs méthodes et les relations qui les unissent. Le diagramme de classes en constitue la pièce maîtresse.

3.2.2 Découpage du système en composants

L'architecture du système s'articule autour de trois composants principaux, reflétant une séparation claire des responsabilités :

1. **Composant Frontend** : responsable de l'interface utilisateur, de la capture des interactions et de la présentation des informations. Ce composant communique avec le backend via des appels API REST.
2. **Composant Backend** : cœur métier du système, il gère l'authentification, le contrôle d'accès, la gestion des utilisateurs et la coordination des différents services. Il expose une API RESTful sécurisée.
3. **Composant Biométrie** : service spécialisé dans la reconnaissance faciale, isolé pour des raisons de performance et de maintenabilité. Il fournit des services d'enrôlement et de vérification des caractéristiques biométriques.

Cette organisation en composants faiblement couplés favorise l'évolutivité du système et permet une maintenance ciblée de chaque module.

3.3 Modélisation dynamique

La modélisation dynamique permet de représenter le comportement du système au fil du temps, en illustrant les échanges de messages entre les différents acteurs et composants. Les diagrammes de séquence UML constituent un outil privilégié pour cette représentation, car ils mettent en évidence l'ordre chronologique des interactions et les conditions de branchement.

3.3.1 Diagramme de séquence : Authentification et décision MFA

Le processus d'authentification constitue le cas d'utilisation central du système. La figure 3.1 illustre le flux complet d'authentification, depuis la saisie des identifiants jusqu'à la décision concernant les facteurs MFA requis.

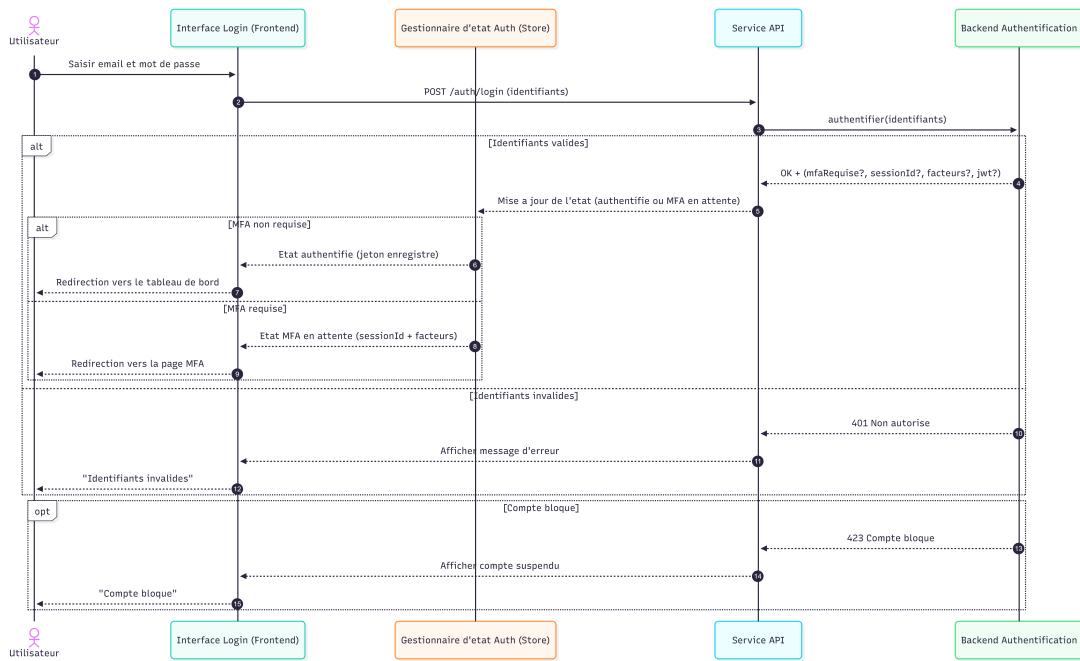


FIGURE 3.1 – Diagramme de séquence : Authentification et décision MFA

Ce diagramme met en évidence le séquencement des opérations lors de la connexion. L'utilisateur soumet ses identifiants au frontend, qui les transmet au backend. Après vérification des credentials, le backend détermine le niveau d'accès de l'utilisateur et évalue les facteurs MFA requis. Selon le niveau, le système peut exiger une validation OTP, une vérification biométrique, ou les deux. Cette logique adaptative constitue un élément différenciant de notre solution.

3.3.2 Diagramme de séquence : Validation MFA via OTP

La validation par code à usage unique représente un facteur d'authentification essentiel pour les niveaux d'accès intermédiaires et élevés. La figure 3.2 détaille les échanges lors de cette validation.

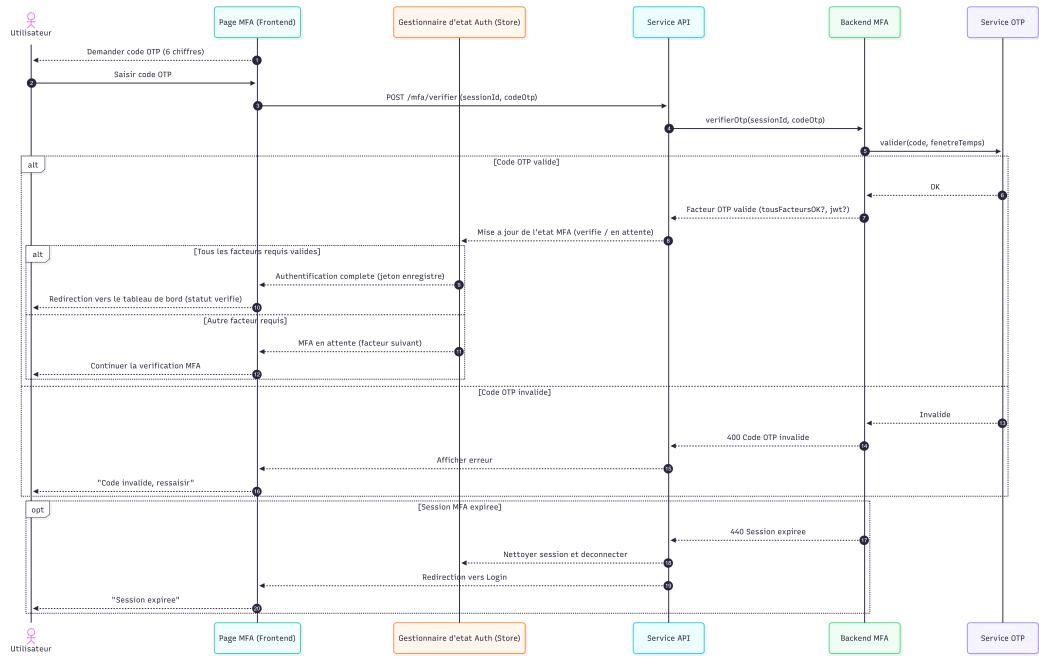


FIGURE 3.2 – Diagramme de séquence : Validation MFA via OTP

Le processus débute par l'affichage d'un formulaire de saisie du code OTP. L'utilisateur consulte son application d'authentification (Google Authenticator), saisit le code temporel et le soumet. Le backend vérifie la validité du code en recalculant la valeur attendue à partir du secret partagé et de l'horodatage actuel. En cas de succès, la session MFA est mise à jour avec la validation de ce facteur.

3.3.3 Diagramme de séquence : Configuration OTP

L'activation de l'authentification OTP par l'utilisateur constitue une opération sensible nécessitant une génération sécurisée du secret partagé. La figure 3.3 présente ce processus de configuration.

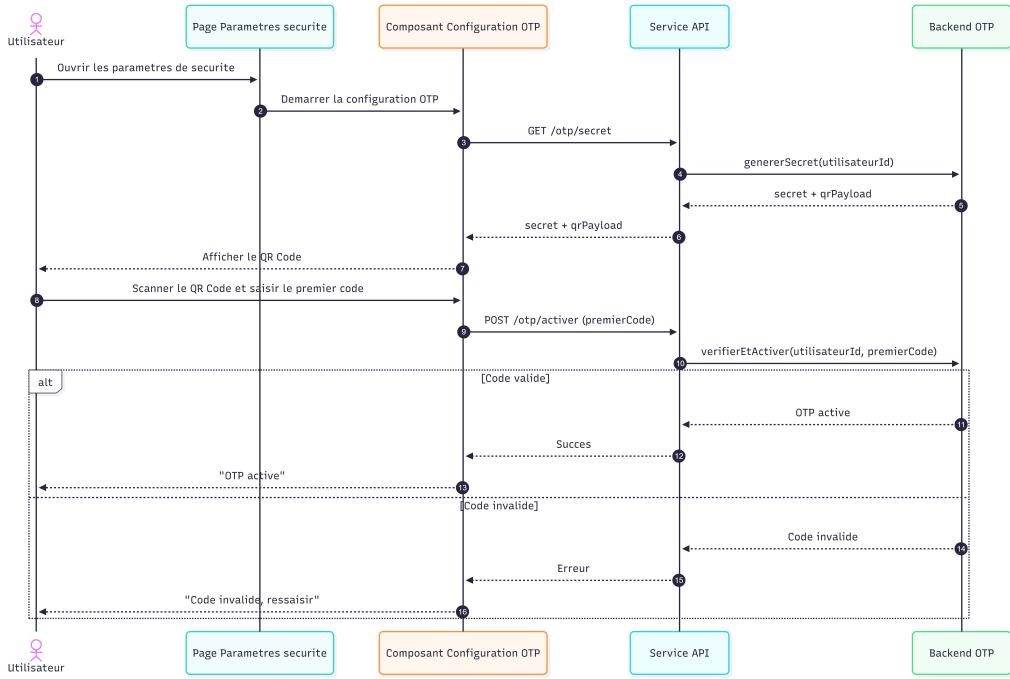


FIGURE 3.3 – Diagramme de séquence : Configuration de l'authentification OTP

L'utilisateur initie la configuration depuis son profil. Le backend génère un secret cryptographique aléatoire et le formate en URI compatible avec le standard TOTP. Cette URI est encodée sous forme de QR code et transmise au frontend pour affichage. L'utilisateur scanne ce QR code avec son application d'authentification, puis saisit un premier code pour confirmer la synchronisation. Une fois validé, le secret est associé au compte utilisateur et l'authentification OTP est activée.

3.3.4 Diagramme de séquence : Accès refusé à une ressource

Le contrôle d'accès aux ressources protégées constitue une fonction critique du système. La figure 3.4 illustre le scénario où un utilisateur tente d'accéder à une ressource requérant un niveau d'habilitation supérieur au sien.

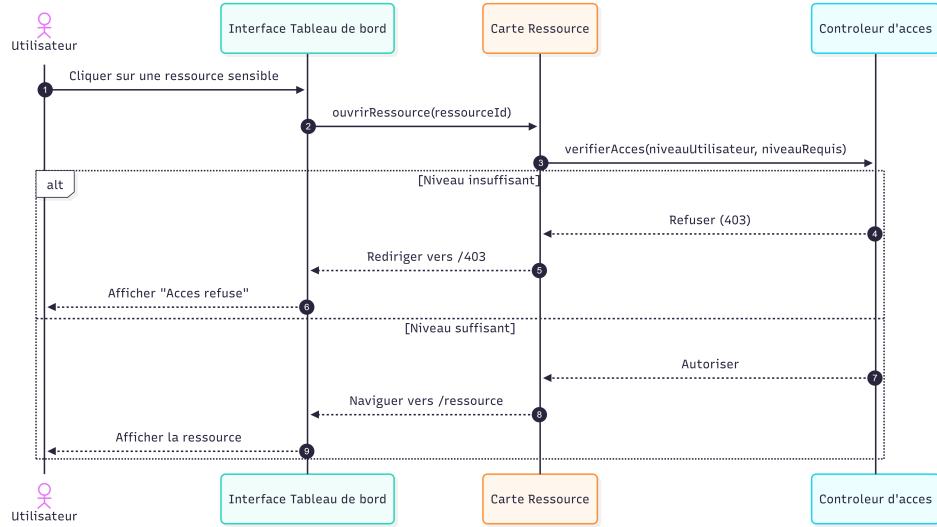


FIGURE 3.4 – Diagramme de séquence : Accès refusé à une ressource protégée

Ce diagramme montre la vérification effectuée par le backend lors de chaque tentative d'accès à une ressource. Le système compare le niveau d'accès de l'utilisateur authentifié avec le niveau minimum requis par la ressource. Si l'utilisateur ne dispose pas des habilitations suffisantes, un message d'erreur explicite est retourné, indiquant le niveau requis. Cet événement est par ailleurs journalisé pour permettre une traçabilité complète des tentatives d'accès.

3.4 Modélisation statique

La modélisation statique vise à représenter la structure du système indépendamment de son comportement temporel. Le diagramme de classes constitue l'outil central de cette modélisation, définissant les entités du domaine, leurs attributs, leurs méthodes et les relations qui les lient.

3.4.1 Présentation du diagramme de classes

Le diagramme de classes présenté en figure 3.5 offre une vue d'ensemble des principales entités du système et de leurs interconnexions.

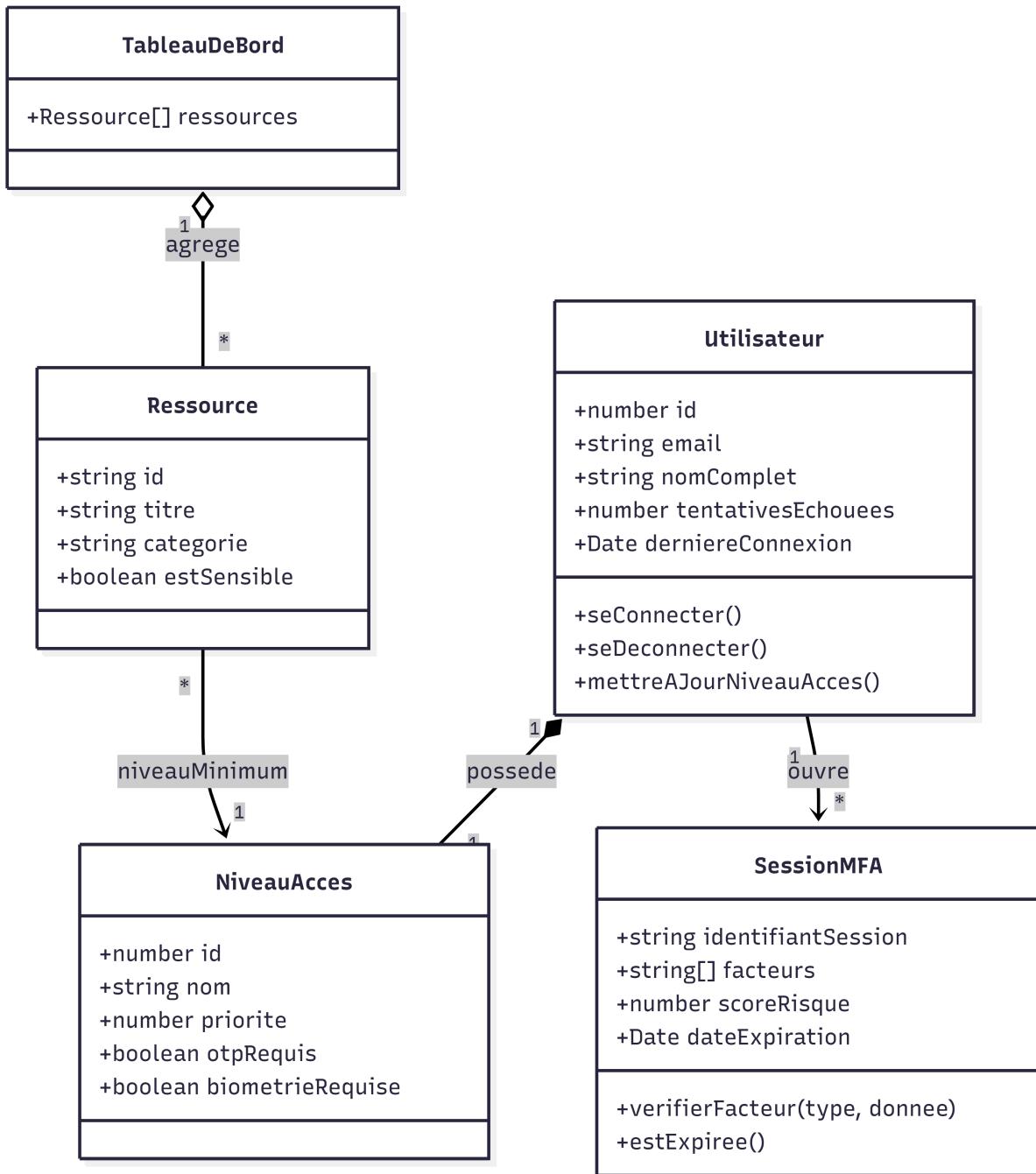


FIGURE 3.5 – Diagramme de classes global du système

Ce diagramme met en évidence l'organisation structurelle du système autour de plusieurs classes clés que nous détaillons ci-après.

3.4.2 Description des principales classes

Classe User

La classe `User` représente un utilisateur du système. Elle encapsule les informations d'identification (email, mot de passe haché), le niveau d'accès attribué, et les références

vers les données MFA associées. Cette classe constitue le pivot central du modèle, étant référencée par la plupart des autres entités.

- **Attributs principaux** : id, email, passwordHash, accessLevel, isActive, createdAt
- **Relations** : composition avec OtpConfig, association avec BiometricData, agrégation avec MfaSession

Classe AccessLevel

La classe **AccessLevel** modélise les différents niveaux d'habilitation du système. Chaque niveau est caractérisé par un identifiant, un libellé descriptif et la liste des facteurs MFA requis pour y accéder.

- **Attributs principaux** : levelCode, label, requiredMfaFactors, priority
- **Relations** : association avec User, association avec Resource

Classe Resource

La classe **Resource** représente une ressource protégée du système. Chaque ressource est associée à un niveau d'accès minimum requis, définissant ainsi les conditions d'habilitation nécessaires pour y accéder.

- **Attributs principaux** : id, name, description, resourceType, requiredAccessLevel
- **Relations** : association avec AccessLevel

Classe MfaSession

La classe **MfaSession** modélise une session d'authentification multi-facteurs en cours. Elle trace les facteurs déjà validés, l'horodatage de chaque validation et le statut global de la session.

- **Attributs principaux** : sessionId, userId, validatedFactors, createdAt, expiresAt, isComplete
- **Relations** : association avec User

Classe OtpConfig

La classe **OtpConfig** encapsule la configuration OTP d'un utilisateur, notamment le secret partagé et l'état d'activation.

- **Attributs principaux** : secret, isEnabled, activatedAt
- **Relations** : composition avec User

Classe BiometricData

La classe **BiometricData** stocke les données biométriques faciales d'un utilisateur sous forme d'embeddings (vecteurs de caractéristiques).

- **Attributs principaux** : embedding, capturedAt, isActive
- **Relations** : association avec User

Classe AccessLog

La classe **AccessLog** assure la journalisation des événements d'accès, permettant l'audit et la détection d'anomalies.

- **Attributs principaux** : logId, userId, eventType, timestamp, ipAddress, success, details
- **Relations** : association avec User

3.4.3 Relations entre classes

Le diagramme met en évidence plusieurs types de relations structurant le modèle :

- **Composition** : la relation entre User et OtpConfig indique que la configuration OTP n'existe pas indépendamment de l'utilisateur.
- **Association** : les relations entre User et Resource, ou entre User et AccessLog, représentent des liens sémantiques sans dépendance existentielle.
- **Extension future** : bien que non représentée dans ce diagramme, la conception permet une évolution vers une hiérarchie de rôles plus complexe.

3.5 Architecture du système

L'architecture du système définit l'organisation des composants logiciels et matériels, leurs interactions et les technologies utilisées. Nous présentons successivement l'architecture logicielle et l'architecture matérielle de déploiement.

3.5.1 Architecture logicielle

L'architecture logicielle du système FIFA World Cup 2026 – Unity Hub repose sur une organisation en couches et en services, favorisant la modularité et la maintenabilité. La figure 3.6 présente cette organisation.

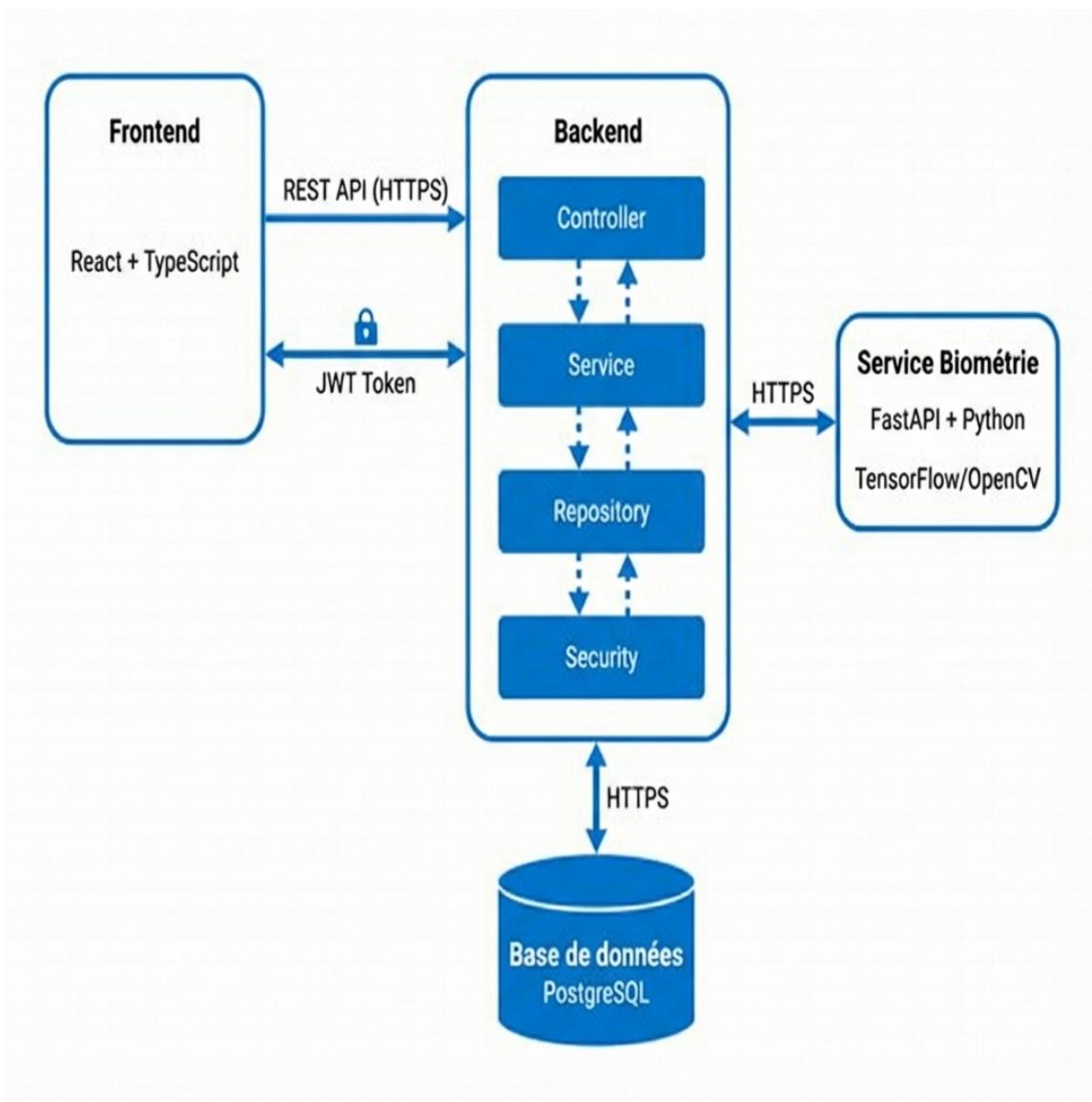


FIGURE 3.6 – Architecture logicielle du système

Cette architecture s'organise autour de trois blocs principaux interconnectés :

Frontend (React + TypeScript)

Le frontend constitue la couche de présentation du système. Développé avec React et TypeScript, il offre une interface utilisateur réactive et typée. Les principales responsabilités incluent :

- Affichage des interfaces d'authentification et de navigation
- Capture des interactions utilisateur et des données biométriques
- Communication avec le backend via des appels API REST
- Gestion de l'état applicatif côté client

Le choix de React se justifie par sa popularité, sa maturité et son écosystème riche. TypeScript apporte la sécurité du typage statique, réduisant les erreurs à l'exécution.

Backend (Spring Boot)

Le backend constitue le cœur métier du système. Développé avec Spring Boot (Java 17), il implémente la logique d'authentification, de gestion des utilisateurs et de contrôle d'accès. Son architecture interne suit le pattern MVC adapté aux API REST :

- **Couche Controller** : exposition des endpoints REST, validation des entrées
- **Couche Service** : logique métier, orchestration des opérations
- **Couche Repository** : accès aux données via JPA/Hibernate
- **Couche Security** : gestion des tokens JWT, filtres de sécurité

Spring Boot a été choisi pour sa robustesse, son intégration native avec Spring Security et la richesse de son écosystème.

Service Biométrie (FastAPI + Python)

Le service de reconnaissance faciale est isolé en un microservice autonome, développé avec FastAPI (Python). Cette séparation se justifie par :

- La spécificité des traitements d'intelligence artificielle (TensorFlow, OpenCV)
- Les besoins en ressources différents des autres composants
- La possibilité de faire évoluer ce service indépendamment

Ce service expose deux endpoints principaux : enrôlement (calcul et stockage des embeddings) et vérification (comparaison d'un visage capturé avec les données enregistrées).

3.5.2 Architecture matérielle

L'architecture matérielle décrit l'organisation physique du déploiement, les nœuds d'exécution et les protocoles de communication. La figure 3.7 présente cette organisation.

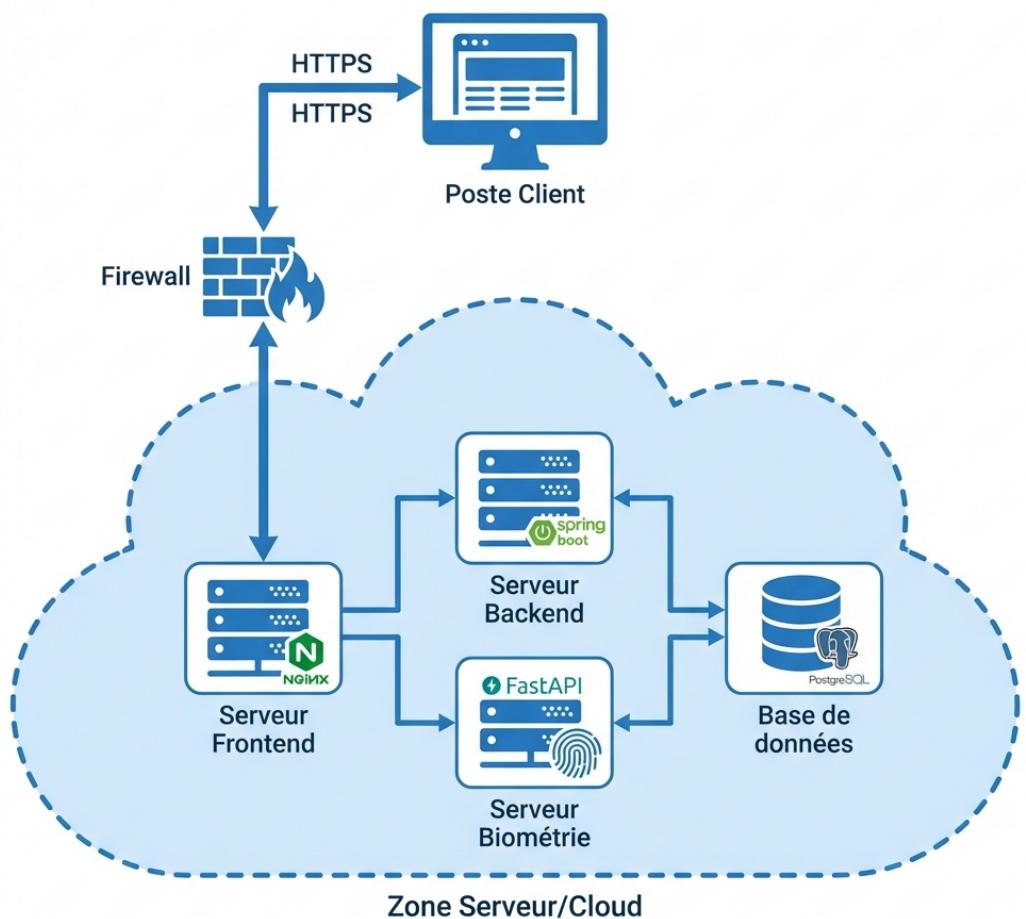


FIGURE 3.7 – Architecture matérielle de déploiement

L'architecture de déploiement s'articule autour des noeuds suivants :

Poste client

Le poste client exécute un navigateur web moderne accédant à l'application frontend. Les prérequis incluent une connexion internet stable et, pour les fonctions biométriques, une caméra accessible via l'API WebRTC du navigateur.

Serveur Frontend

Le serveur frontend héberge l'application React buildée sous forme de ressources statiques. Un serveur web (Nginx) assure la distribution de ces ressources et le routage des requêtes API vers le backend.

Serveur Backend

Le serveur backend exécute l'application Spring Boot. Il communique avec la base de données pour la persistance des informations utilisateurs et avec le service biométrie pour les opérations de reconnaissance faciale.

Serveur Biométrie

Le serveur biométrie héberge le service FastAPI. Il peut être dimensionné indépendamment pour absorber la charge des traitements d'intelligence artificielle, qui sont plus coûteux en ressources que les opérations classiques.

Serveur Base de données

Le serveur de base de données héberge le SGBD (H2 en développement, PostgreSQL en production). Il assure la persistance des données utilisateurs, des configurations MFA, des ressources et des journaux d'accès.

3.5.3 Protocoles et sécurisation des échanges

Les communications entre composants respectent les principes de sécurité suivants :

- **HTTPS** : toutes les communications client-serveur sont chiffrées via TLS
- **JWT** : les tokens d'authentification sont signés et vérifiés à chaque requête
- **CORS** : la politique CORS restreint les origines autorisées
- **Réseau interne** : les communications backend-biométrie transitent par un réseau privé

3.6 Conclusion

Ce chapitre a présenté la conception du système FIFA World Cup 2026 – Unity Hub à travers ses dimensions dynamique, statique et architecturale. La modélisation UML a permis de formaliser les principaux mécanismes d'authentification multi-facteurs et de contrôle d'accès, en mettant en évidence les interactions clés entre les différents composants du système.

Le diagramme de classes a structuré les entités centrales du domaine et leurs relations, tandis que les architectures logicielle et matérielle retenues assurent modularité, évolutivité et sécurité. Ces choix de conception constituent une base solide pour la phase de réalisation, présentée dans le chapitre suivant.

Chapitre 4

Réalisation du système

4.1 Introduction

La phase de réalisation constitue l'aboutissement concret du cycle de développement, où la conception théorique se matérialise en un système fonctionnel. Cette étape traduit les modèles UML et les spécifications architecturales en composants logiciels opérationnels, interfaces utilisateur et services interconnectés.

Dans le cadre du projet FIFA World Cup 2026 – Unity Hub, la réalisation s'est appuyée sur les choix technologiques validés lors de la conception : un frontend React avec TypeScript, un backend Spring Boot, et un service de reconnaissance faciale FastAPI. Le respect de ces orientations garantit la cohérence entre le système livré et les exigences définies aux chapitres précédents.

Ce chapitre présente successivement l'environnement matériel de développement, l'environnement logiciel utilisé, puis détaille les principales interfaces graphiques de l'application. Cette présentation permet d'illustrer concrètement les fonctionnalités implémentées et de démontrer la conformité du système avec les spécifications fonctionnelles.

4.2 Environnement matériel

Le développement du système a nécessité un environnement matériel adapté aux contraintes des différentes technologies mises en œuvre, notamment pour les traitements de reconnaissance faciale qui sollicitent davantage les ressources.

4.2.1 Poste de développement

Le développement a été réalisé sur un poste de travail présentant les caractéristiques suivantes :

- **Processeur** : Intel Core i7 (8 cœurs) ou équivalent AMD Ryzen

- **Mémoire vive** : 16 Go de RAM DDR4
- **Stockage** : SSD NVMe de 512 Go
- **Carte graphique** : GPU dédié compatible CUDA (pour l'accélération TensorFlow)
- **Périphériques** : Webcam HD (720p minimum) pour les tests de reconnaissance faciale

Cette configuration permet d'exécuter simultanément les trois composants du système (frontend, backend, service biométrie) tout en conservant des performances satisfaisantes lors des phases de développement et de test.

4.2.2 Configuration minimale requise

Pour le déploiement en environnement de production, les configurations minimales recommandées sont :

- **Serveur Backend** : 4 vCPU, 8 Go RAM, 100 Go stockage SSD
- **Serveur Biométrie** : 4 vCPU (GPU recommandé), 8 Go RAM, 50 Go stockage SSD
- **Serveur Base de données** : 2 vCPU, 4 Go RAM, 200 Go stockage SSD (extensible)
- **Poste client** : Navigateur moderne, webcam pour la biométrie, connexion internet stable

4.2.3 Contraintes liées à la biométrie

La reconnaissance faciale impose des contraintes matérielles spécifiques. La caméra du poste client doit offrir une résolution suffisante (720p minimum) et un bon comportement en conditions d'éclairage variables. Côté serveur, le service biométrie bénéficie significativement d'une accélération GPU pour le calcul des embeddings faciaux, bien qu'un fonctionnement sur CPU seul reste possible avec des temps de réponse légèrement allongés.

4.3 Environnement logiciel

L'environnement logiciel de développement regroupe l'ensemble des outils, frameworks et technologies utilisés pour la réalisation du système.

4.3.1 Système d'exploitation

Le développement a été effectué principalement sous Windows 11, avec une compatibilité vérifiée sous Linux (Ubuntu 22.04) pour le déploiement serveur. Les conteneurs Docker permettent d'assurer la portabilité des services entre environnements.

4.3.2 Outils de développement

Les outils suivants ont été utilisés pour le développement :

TABLE 4.1 – Outils de développement utilisés

Catégorie	Outil	Usage
IDE	Visual Studio Code	Développement frontend et Python
IDE	IntelliJ IDEA	Développement backend Java
Versionnement	Git	Gestion du code source
Plateforme Git	GitHub	Hébergement du dépôt
Conteneurisation	Docker	Isolation des services
Client API	Postman	Tests des endpoints REST
Navigateur	Google Chrome	Tests et débogage frontend

4.3.3 Frameworks et technologies

Le tableau 4.2 récapitule les principales technologies utilisées pour chaque composant du système.

TABLE 4.2 – Technologies utilisées par composant

Composant	Technologie	Version
Frontend	React	18.x
Frontend	TypeScript	5.x
Frontend	Vite	5.x
Frontend	TailwindCSS	3.x
Frontend	Axios	1.x
Backend	Spring Boot	3.2.x
Backend	Java	17
Backend	Spring Security	6.x
Backend	JWT (jjwt)	0.12.x
Backend	H2 / PostgreSQL	2.x / 15.x
Biométrie	FastAPI	0.100+
Biométrie	Python	3.11
Biométrie	TensorFlow	2.15
Biométrie	OpenCV	4.x

4.3.4 Outils de test et de versionnement

La qualité du code est assurée par des tests unitaires et d'intégration :

- **Backend** : JUnit 5 pour les tests unitaires, Mockito pour les mocks
- **Frontend** : Jest et React Testing Library pour les tests de composants
- **Biométrie** : Pytest pour les tests Python
- **API** : Collections Postman pour les tests d'intégration

Le versionnement du code suit le workflow Git Flow, avec des branches dédiées aux fonctionnalités, aux correctifs et aux versions stables.

4.4 Présentation des interfaces graphiques

Cette section présente les principales interfaces graphiques de l'application FIFA World Cup 2026 – Unity Hub. Chaque interface est décrite du point de vue de son rôle fonctionnel et de son ergonomie.

4.4.1 Page d'accueil

La page d'accueil constitue le point d'entrée de l'application. Elle présente la plateforme et oriente les visiteurs vers les fonctionnalités principales. La figure 4.1 illustre cette interface.

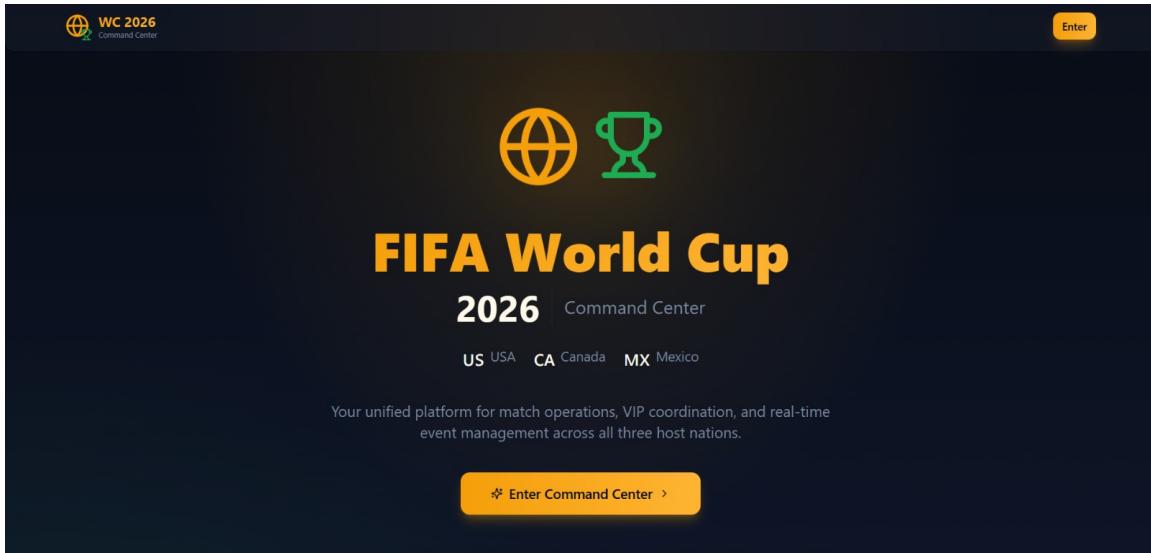


FIGURE 4.1 – Page d'accueil de la plateforme Unity Hub

Cette interface adopte une identité visuelle cohérente avec l'événement FIFA World Cup 2026, utilisant les couleurs officielles et une mise en page moderne. Un bouton d'appel à l'action invite l'utilisateur à se connecter ou à découvrir les fonctionnalités de la plateforme. La navigation principale permet d'accéder rapidement aux différentes sections publiques.

4.4.2 Interface de connexion

L'interface de connexion permet aux utilisateurs de s'authentifier auprès du système. Elle constitue la première étape du processus MFA. La figure 4.2 présente cette interface.

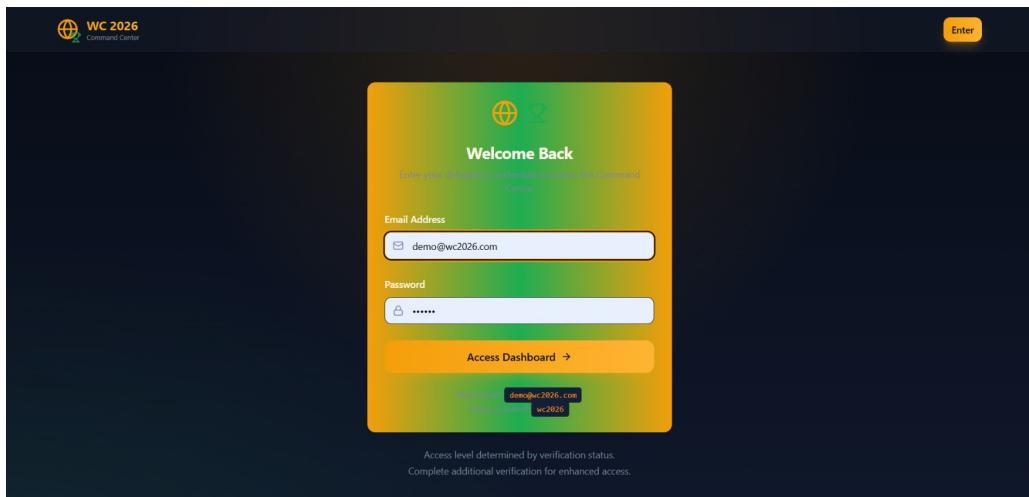


FIGURE 4.2 – Interface de connexion utilisateur

L'interface propose un formulaire épuré comprenant les champs email et mot de passe. Un lien vers la récupération de mot de passe est disponible pour les utilisateurs ayant oublié leurs identifiants. Les messages d'erreur sont affichés de manière claire en cas d'échec d'authentification, sans révéler d'information sensible sur la validité des identifiants.

4.4.3 Interface de validation MFA

Après validation des identifiants, l'utilisateur est dirigé vers l'interface de validation MFA si son niveau d'accès l'exige. Cette interface guide l'utilisateur à travers les étapes de vérification supplémentaires. La figure 4.3 illustre cette interface.

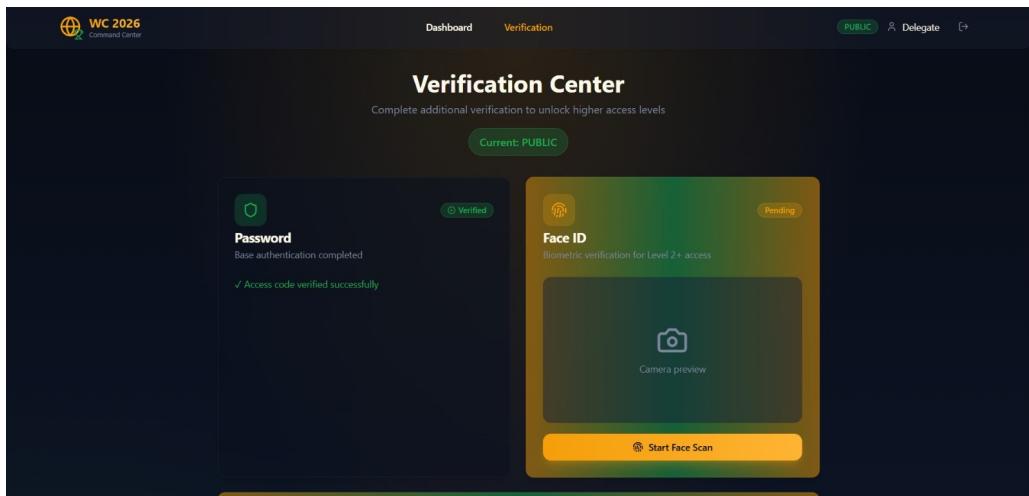


FIGURE 4.3 – Interface de validation multi-facteurs (MFA)

L'interface présente clairement les facteurs d'authentification requis et validés. Selon la configuration de l'utilisateur, elle peut demander la saisie d'un code OTP (avec un champ de saisie dédié) et/ou une vérification faciale (avec activation de la caméra et affichage d'un aperçu). Un indicateur de progression informe l'utilisateur des étapes restantes.

4.4.4 Tableau de bord

Une fois authentifié, l'utilisateur accède à son tableau de bord personnalisé. Cette interface centrale présente les informations et actions accessibles selon son niveau d'habilitation. La figure 4.4 présente le tableau de bord.

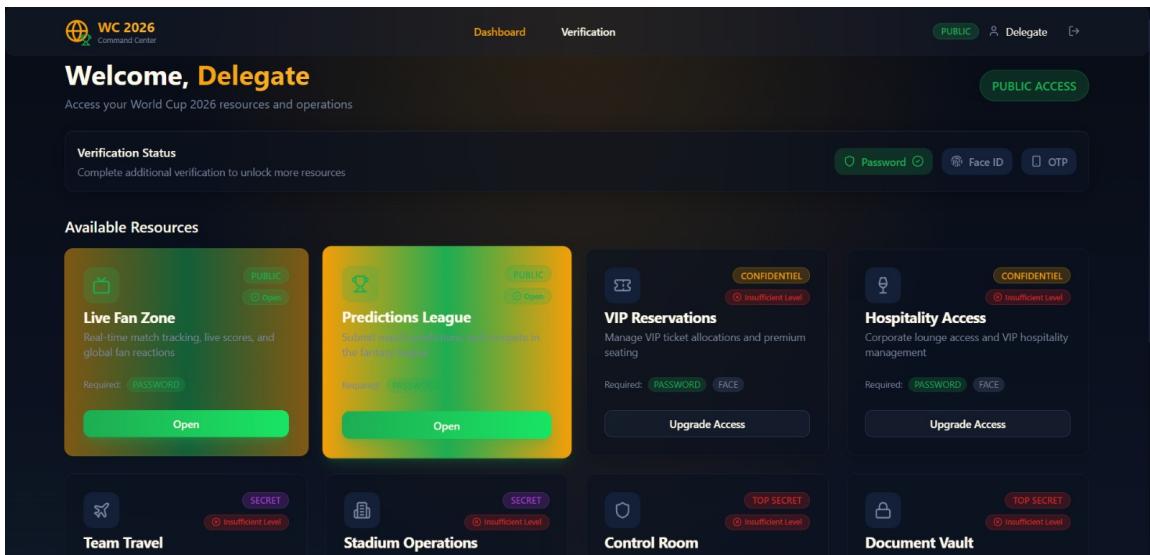


FIGURE 4.4 – Tableau de bord utilisateur

Le tableau de bord affiche le nom de l'utilisateur connecté, son niveau d'accès actuel et les ressources auxquelles il peut accéder. Des cartes thématiques permettent de naviguer vers les différentes fonctionnalités de la plateforme. Pour les administrateurs, des sections supplémentaires de gestion et de supervision sont visibles.

4.4.5 Exigences de niveau d'accès

Lorsqu'un utilisateur tente d'accéder à une ressource requérant un niveau supérieur au sien, une interface dédiée l'informe des exigences. La figure 4.5 illustre cette interface.

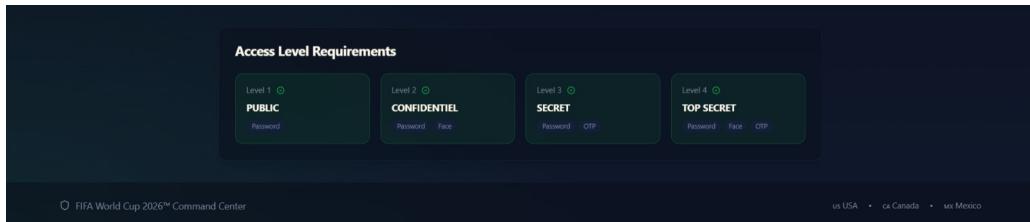


FIGURE 4.5 – Interface d'affichage des exigences de niveau d'accès

Cette interface affiche de manière explicite le niveau d'accès actuel de l'utilisateur et le niveau requis pour accéder à la ressource demandée. Elle détaille les facteurs d'authentification supplémentaires nécessaires et propose, le cas échéant, une option pour initier une élévation de privilège si l'utilisateur dispose des prérequis.

4.4.6 Interface Fan Zone

L'interface Fan Zone représente une ressource accessible aux utilisateurs de niveau intermédiaire. Elle illustre le contenu thématique lié à la Coupe du Monde. La figure 4.6 présente cette interface.

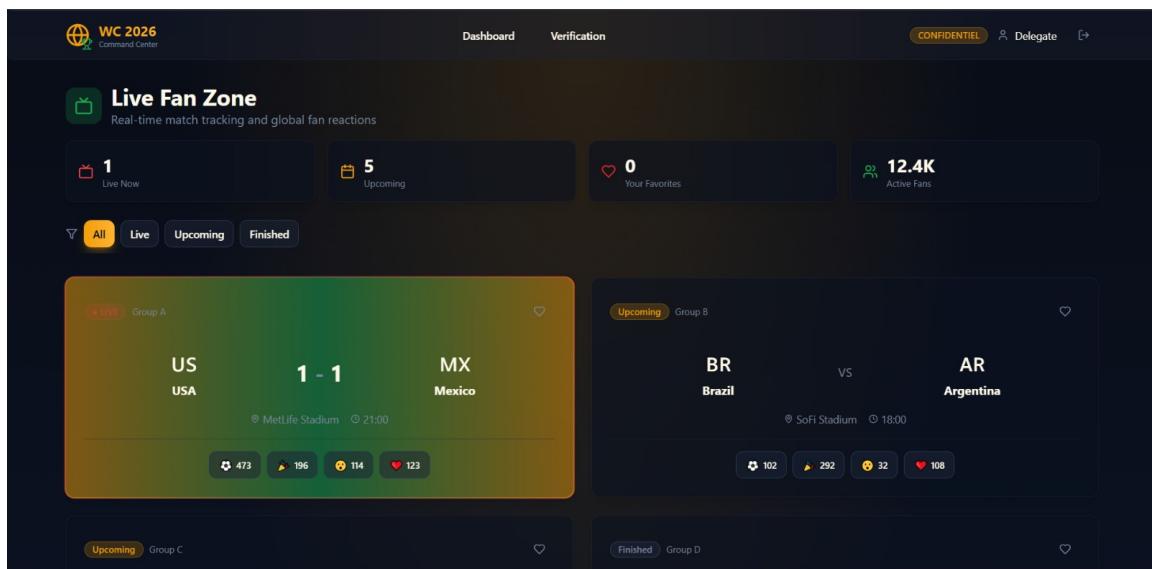


FIGURE 4.6 – Interface Fan Zone – Espace supporters

Cette interface propose des contenus exclusifs destinés aux supporters : actualités en temps réel, informations sur les matchs, statistiques et interactions communautaires. L'accès à cette zone requiert au minimum une authentification de niveau 2, incluant la validation OTP.

4.4.7 Interface Stadium Operations

L'interface Stadium Operations constitue une ressource sensible accessible uniquement aux utilisateurs disposant des plus hauts niveaux d'habilitation. La figure 4.7 illustre cette interface.

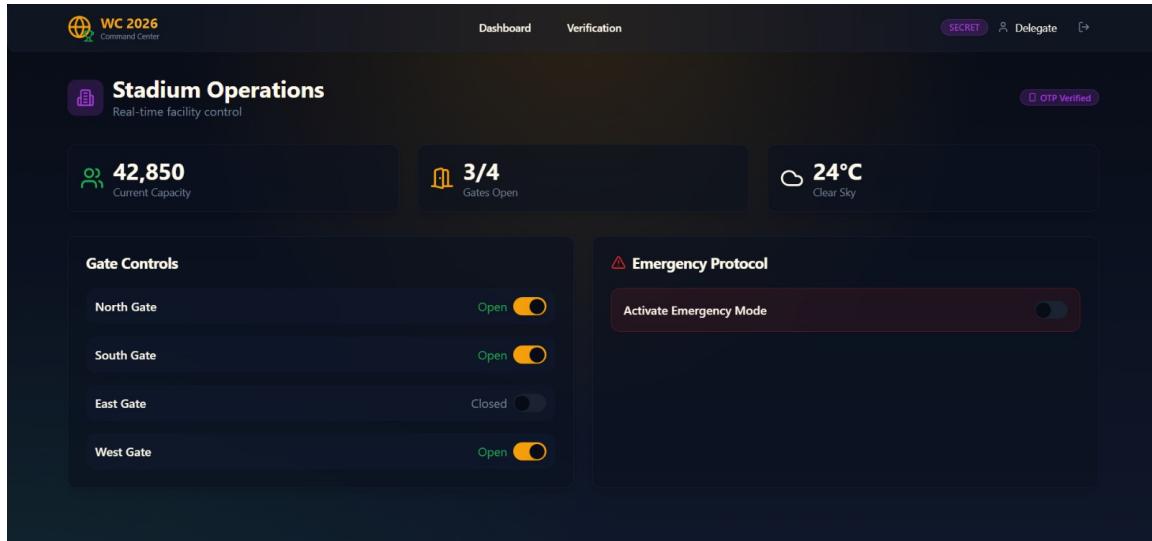


FIGURE 4.7 – Interface Stadium Operations – Gestion opérationnelle

Cette interface d'administration permet de gérer les opérations liées aux stades : contrôle des flux, gestion des accréditations, supervision des accès en temps réel. L'accès requiert une authentification de niveau 3 complète, incluant la vérification biométrique par reconnaissance faciale.

4.4.8 Interface VIP Hospitality

L'interface VIP Hospitality est dédiée à la gestion des services d'accueil haut de gamme pour les invités VIP de l'événement. La figure 4.8 présente cette interface.

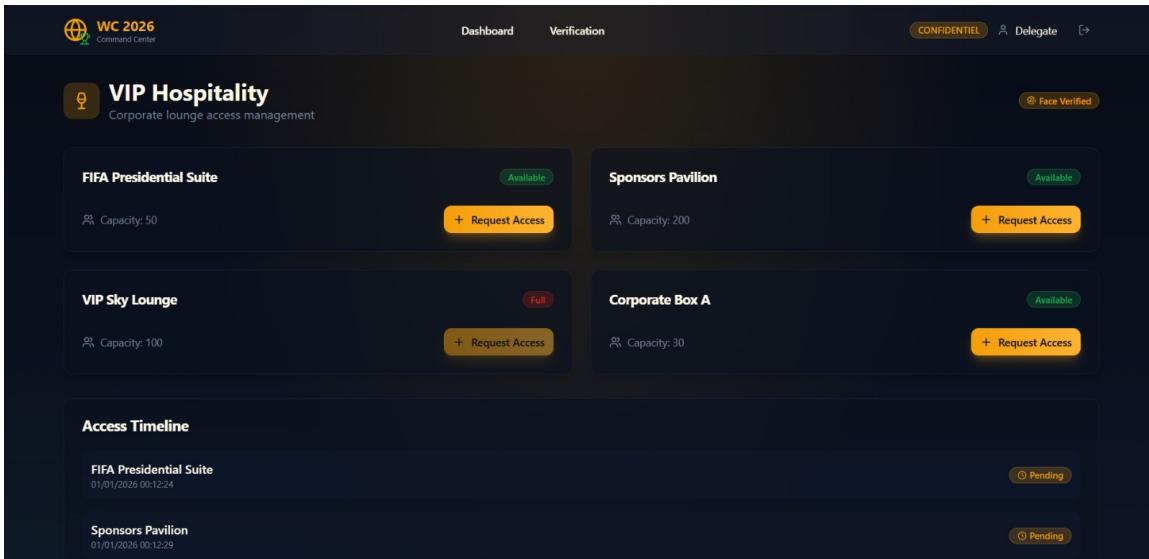


FIGURE 4.8 – Interface VIP Hospitality – Services d'accueil privilégiés

Cette interface permet de consulter et gérer les prestations exclusives réservées aux invités de marque : accès aux salons privés, services de restauration premium et accompagnement personnalisé. L'accès à cette ressource nécessite un niveau d'habilitation élevé, garantissant la confidentialité des informations relatives aux invités VIP.

4.4.9 Interface VIP Reservations

L'interface VIP Reservations centralise la gestion des réservations pour les espaces et services VIP. La figure 4.9 illustre cette interface.

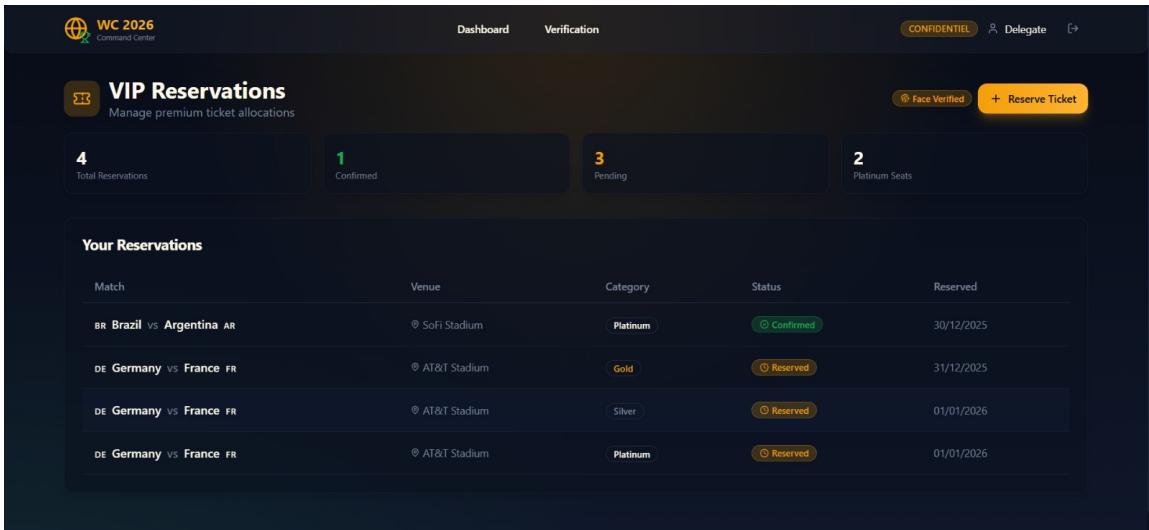


FIGURE 4.9 – Interface VIP Reservations – Gestion des réservations privilégiées

Cette interface affiche le calendrier des réservations, les disponibilités des espaces VIP et permet de créer ou modifier des réservations. Elle s'adresse aux responsables de

l'hospitalité et requiert une authentification multi-facteurs complète pour protéger les données sensibles des invités.

4.4.10 Interface Event Control Room

L'interface Event Control Room constitue le centre de supervision des événements en temps réel. La figure 4.10 présente cette interface.

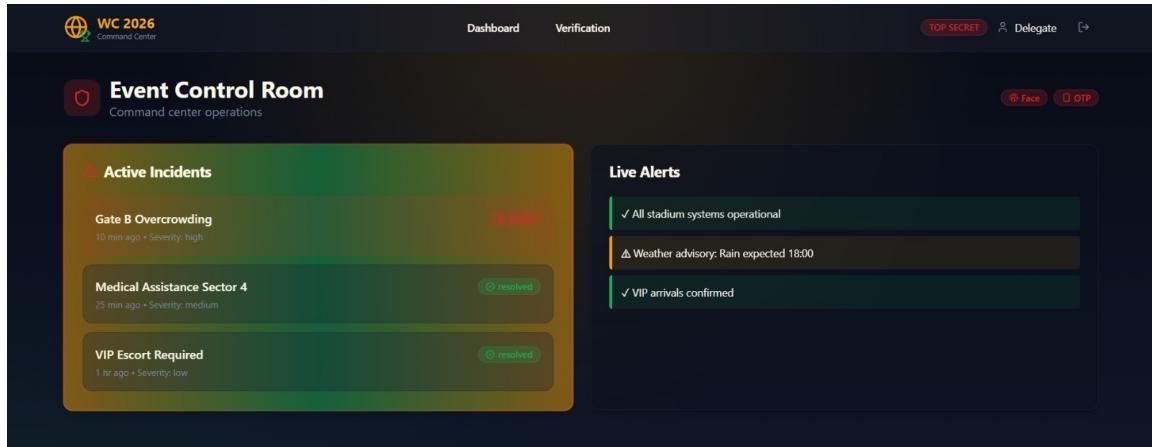
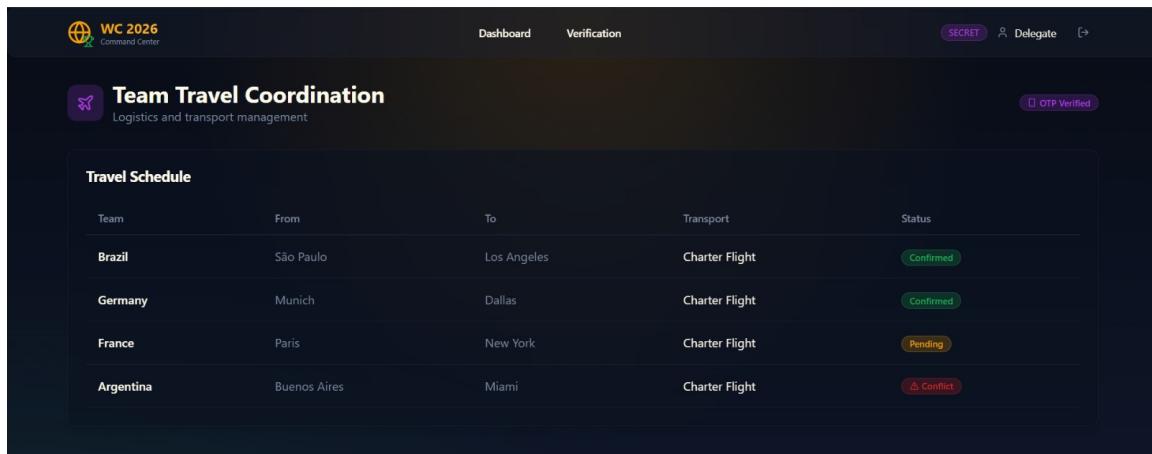


FIGURE 4.10 – Interface Event Control Room – Centre de supervision événementielle

Cette interface agrège les informations critiques des différents stades et sites de l'événement : flux de spectateurs, alertes de sécurité, statuts opérationnels. Elle permet aux responsables de prendre des décisions éclairées en temps réel. L'accès est strictement réservé aux administrateurs disposant du niveau d'accès maximal.

4.4.11 Interface Team Travel Coordination

L'interface Team Travel Coordination facilite la gestion logistique des déplacements des équipes participantes. La figure 4.11 illustre cette interface.



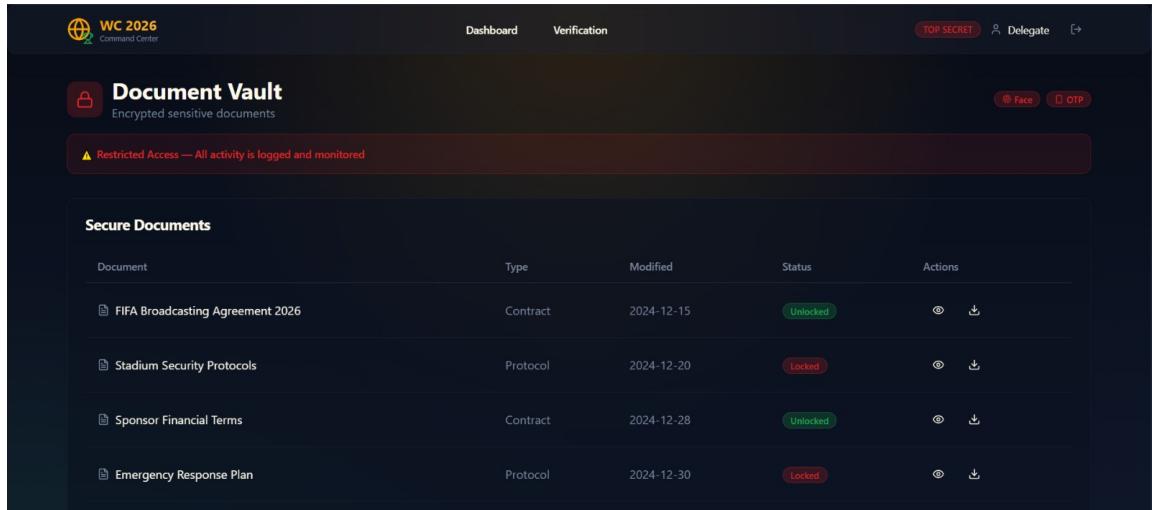
Team	From	To	Transport	Status
Brazil	São Paulo	Los Angeles	Charter Flight	Confirmed
Germany	Munich	Dallas	Charter Flight	Confirmed
France	Paris	New York	Charter Flight	Pending
Argentina	Buenos Aires	Miami	Charter Flight	Conflict

FIGURE 4.11 – Interface Team Travel Coordination – Coordination des déplacements

Cette interface permet de planifier et suivre les itinéraires des délégations officielles, de coordonner les transports et d'assurer la liaison avec les différents prestataires logistiques. Les informations sensibles relatives aux déplacements des équipes sont protégées par un contrôle d'accès renforcé.

4.4.12 Interface Documents Vault

L'interface Documents Vault offre un espace sécurisé pour le stockage et le partage de documents confidentiels. La figure 4.12 présente cette interface.



The screenshot shows the WC 2026 Command Center interface, specifically the Document Vault section. At the top, there are navigation links for Dashboard, Verification, and a red 'TOP SECRET' button. On the right, there are options for Delegate, Face, and OTP. The main area is titled 'Document Vault' with a subtitle 'Encrypted sensitive documents'. A warning message at the top states '⚠ Restricted Access — All activity is logged and monitored'. Below this, a table titled 'Secure Documents' lists five entries:

Document	Type	Modified	Status	Actions
FIFA Broadcasting Agreement 2026	Contract	2024-12-15	Unlocked	🔗
Stadium Security Protocols	Protocol	2024-12-20	Locked	🔗
Sponsor Financial Terms	Contract	2024-12-28	Unlocked	🔗
Emergency Response Plan	Protocol	2024-12-30	Locked	🔗

FIGURE 4.12 – Interface Documents Vault – Coffre-fort documentaire sécurisé

Cette interface permet d'accéder aux documents officiels, contrats et fichiers sensibles de l'organisation. Chaque document est classifié selon son niveau de confidentialité, et l'accès est conditionné par le niveau d'habilitation de l'utilisateur et la validation MFA appropriée.

4.4.13 Interface Predictions League

L'interface Predictions League propose un espace ludique de pronostics pour les utilisateurs de la plateforme. La figure 4.13 illustre cette interface.

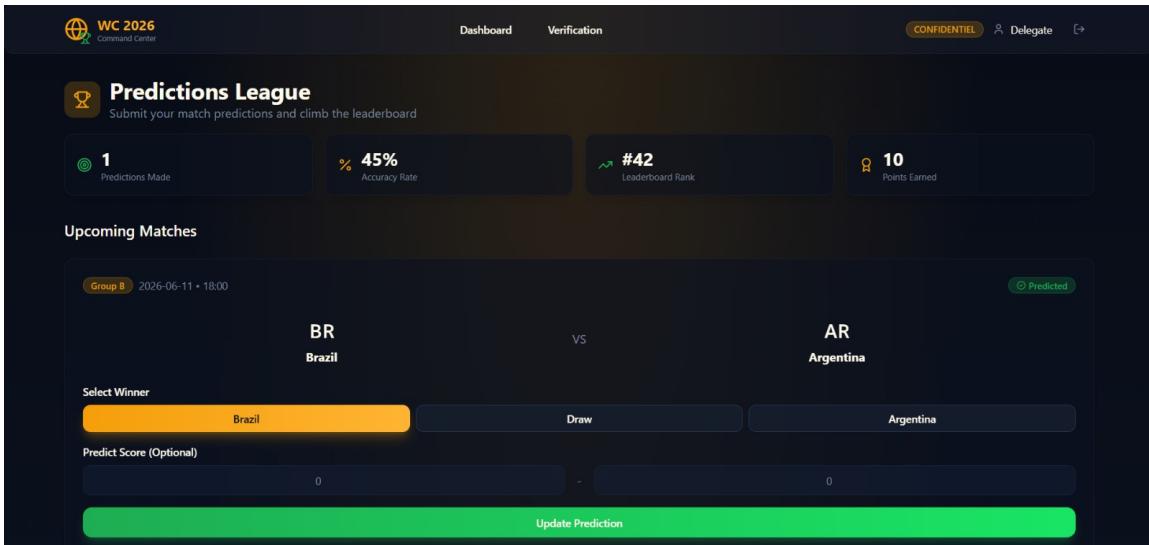


FIGURE 4.13 – Interface Predictions League – Espace pronostics et classements

Cette interface permet aux utilisateurs authentifiés de soumettre leurs pronostics sur les matchs à venir, de consulter les classements et de suivre leurs performances. Accessible dès le niveau d'accès basique, elle contribue à l'engagement des utilisateurs tout en démontrant la gestion différenciée des ressources selon les niveaux d'habilitation.

4.5 Conclusion

Ce chapitre a présenté la phase de réalisation du système FIFA World Cup 2026 – Unity Hub, en abordant les environnements matériel et logiciel ainsi que les principales interfaces utilisateur développées.

Les choix techniques retenus permettent l'exécution efficace des différents composants du système et garantissent la qualité, la maintenabilité et la cohérence de l'application. Les interfaces graphiques réalisées illustrent concrètement les fonctionnalités mises en œuvre, dans le respect des principes d'ergonomie et d'accessibilité.

La réalisation est conforme à la conception définie au chapitre précédent, confirmant la faisabilité technique des choix architecturaux et technologiques. La conclusion générale synthétisera l'ensemble des travaux réalisés et présentera les perspectives d'évolution du projet.

Conclusion Générale

Le projet FIFA World Cup 2026 – Unity Hub avait pour objectif de répondre aux défis de sécurisation des accès dans le contexte d'un événement international majeur. Face à l'insuffisance des mécanismes d'authentification traditionnels et à la diversité des profils utilisateurs, nous avons conçu un système intégrant authentification multi-facteurs adaptative, reconnaissance biométrique et contrôle d'accès granulaire.

Les travaux réalisés ont permis d'aboutir à une plateforme fonctionnelle répondant aux objectifs fixés. Le système implémente une authentification forte combinant mot de passe, code OTP et reconnaissance faciale, avec des exigences de vérification ajustées dynamiquement selon le niveau d'accès sollicité. L'architecture modulaire (frontend React, backend Spring Boot, microservice FastAPI) favorise la maintenabilité et l'évolutivité de la solution, tandis que la journalisation centralisée assure une traçabilité complète des événements d'accès.

Certaines limites sont néanmoins identifiées, notamment la dépendance aux conditions matérielles pour la reconnaissance faciale et les contraintes réglementaires liées au traitement des données biométriques. De plus, le système n'a pas été testé sous des charges représentatives d'un déploiement à grande échelle.

Parmi les perspectives d'évolution, l'intégration du standard FIDO2/WebAuthn et l'exploitation des journaux par des algorithmes d'apprentissage automatique pour la détection d'anomalies en temps réel constituent des axes prometteurs. Le projet Unity Hub constitue ainsi une base solide pour des développements futurs, illustrant la faisabilité d'une approche intégrée de la sécurité des accès.

Annexes

Ce document annexe regroupe des éléments complémentaires au rapport principal : table des acronymes utilisés, extraits techniques illustratifs et récapitulatifs de référence.

A1 – Table des acronymes

Le tableau 4.3 récapitule les principaux acronymes et abréviations utilisés tout au long de ce rapport.

TABLE 4.3 – Table des acronymes

Acronyme	Signification
API	Application Programming Interface (Interface de programmation applicative)
CORS	Cross-Origin Resource Sharing (Partage de ressources entre origines)
CSRF	Cross-Site Request Forgery (Falsification de requête intersite)
JWT	JSON Web Token (Jeton Web JSON)
MFA	Multi-Factor Authentication (Authentification multi-facteurs)
OTP	One-Time Password (Mot de passe à usage unique)
RBAC	Role-Based Access Control (Contrôle d'accès basé sur les rôles)
REST	Representational State Transfer (Transfert d'état représentationnel)
RGPD	Règlement Général sur la Protection des Données
TLS	Transport Layer Security (Sécurité de la couche transport)
TOTP	Time-based One-Time Password (Mot de passe temporel à usage unique)
UML	Unified Modeling Language (Langage de modélisation unifié)
XSS	Cross-Site Scripting (Script intersite)

A2 – Extraits techniques illustratifs

Cette section présente des extraits de code simplifiés illustrant les mécanismes clés du système. Ces extraits sont fournis à titre pédagogique et ne représentent pas le code de production dans son intégralité.

A2.1 – Configuration Spring Security (extrait simplifié)

L'extrait suivant illustre la configuration de base de Spring Security pour la gestion des filtres JWT et la protection des endpoints.

```
1 @Configuration
2 @EnableWebSecurity
3 public class SecurityConfig {
4
5     @Bean
6     public SecurityFilterChain filterChain(HttpSecurity http)
7         throws Exception {
8
9         http
10            .csrf(csrf -> csrf.disable())
11            .cors(cors -> cors.configurationSource(corsConfig()))
12            .sessionManagement(session ->
13                session.sessionCreationPolicy(STATELESS))
14            .authorizeHttpRequests(auth -> auth
15                .requestMatchers("/api/auth/**").permitAll()
16                .requestMatchers("/api/admin/**").hasRole("ADMIN")
17                .anyRequest().authenticated())
18            .addFilterBefore(jwtAuthFilter,
19                UsernamePasswordAuthenticationFilter.class);
20
21     }
22 }
```

Listing 4.1 – Configuration Spring Security pour JWT

A2.2 – Structure du payload JWT

Le jeton JWT généré après authentification complète contient les claims essentiels présentés ci-dessous.

```
1 {
2     "sub": "user@example.com",
3     "userId": 42,
4     "accessLevel": "LEVEL_2",
5     "mfaCompleted": true,
6     "validatedFactors": ["PASSWORD", "OTP"],
7     "roles": ["USER"],
8     "iat": 1735750000,
9     "exp": 1735753600
10 }
```

Listing 4.2 – Structure du payload JWT

A2.3 – Configuration des exigences MFA par niveau

L'extrait suivant illustre, sous forme de pseudo-configuration YAML, la correspondance entre les niveaux d'accès et les facteurs MFA requis.

```
1 access-levels:  
2   LEVEL_1:  
3     label: "Acces basique"  
4     required-factors:  
5       - PASSWORD  
6  
7   LEVEL_2:  
8     label: "Acces etendu"  
9     required-factors:  
10      - PASSWORD  
11      - OTP  
12  
13   LEVEL_3:  
14     label: "Acces sensible"  
15     required-factors:  
16       - PASSWORD  
17       - OTP  
18       - FACE_RECOGNITION  
19  
20   ADMIN:  
21     label: "Administration"  
22     required-factors:  
23       - PASSWORD  
24       - OTP  
25       - FACE_RECOGNITION
```

Listing 4.3 – Exigences MFA par niveau d'accès

A3 – Endpoints API principaux

Le tableau 4.4 récapitule les principaux endpoints exposés par l'API backend.

TABLE 4.4 – Endpoints API principaux

Méthode	Endpoint	Description
POST	/api/auth/login	Authentification (email + mot de passe)
POST	/api/auth/logout	Déconnexion et invalidation du token
POST	/api/mfa/verify-otp	Validation du code OTP
POST	/api/mfa/verify-face	Vérification biométrique faciale
GET	/api/mfa/status	Statut de la session MFA en cours
POST	/api/otp/setup	Génération du secret OTP (QR code)
POST	/api/otp/confirm	Confirmation de l'activation OTP
GET	/api/users/me	Profil de l'utilisateur connecté
GET	/api/resources	Liste des ressources accessibles
GET	/api/resources/{id}	Accès à une ressource spécifique
GET	/api/admin/users	Liste des utilisateurs (admin)
PUT	/api/admin/users/{id}/level	Modification du niveau d'accès
GET	/api/admin/logs	Consultation des journaux d'accès

A4 – Checklist de conformité sécurité

Le tableau 4.5 présente une checklist des mesures de sécurité implémentées ou recommandées pour le système.

TABLE 4.5 – Checklist de conformité sécurité

Mesure	Statut	Remarque
Communications HTTPS (TLS 1.2+)	✓	Obligatoire en production
Hachage des mots de passe (bcrypt)	✓	Facteur de coût ≥ 10
Tokens JWT signés (HS256/RS256)	✓	Clé secrète rotative
Expiration des tokens (TTL)	✓	1h par défaut
Protection CSRF désactivée (API stateless)	✓	Justifié par JWT
Politique CORS restrictive	✓	Origines whitelist
Rate limiting sur /auth/login	✓	5 tentatives / minute
Verrouillage temporaire après échecs	✓	15 min après 5 échecs
Journalisation des accès (AccessLog)	✓	Horodatage, IP, résultat
Chiffrement des embeddings biométriques	○	Recommandé
Rotation automatique des clés JWT	○	À implémenter
Audit de sécurité externe	○	Recommandé avant prod

Légende : ✓ = Implémenté ○ = Recommandé / À implémenter