# AI-Enhanced Decentralized Identity Verification System

# Chapter 1: Introduction

## 1.1 Brief overview of the work:

The AI-Enhanced Decentralized Identity Verification System is a secure, privacy-focused digital identity platform that allows users to create, own, and verify their identities without relying on centralized authorities. The system leverages Self-Sovereign Identity (SSI) principles, blockchain-based trust, and AI-driven biometric and document verification to ensure authenticity while keeping personal data under user control.

Unlike traditional identity systems that store sensitive information on centralized servers, this platform stores identity credentials securely in user-controlled wallets and uses blockchain only for immutable cryptographic proofs. AI enhances the verification process through face recognition, liveness detection, and fraud prevention, enabling fast and reliable identity validation. The system is designed to support real-world use cases such as digital KYC, e-governance, education, and secure online access.

## 1.2 Project Objective:

- **User-Owned Digital Identity Management:** Design and implement a self-sovereign identity framework that allows users to create, own, and manage their digital identities through secure wallets, ensuring complete control over personal data without dependence on centralized authorities.
- **Decentralized Trust Using Blockchain:** Utilize blockchain technology to store immutable cryptographic proofs and credential hashes, enabling tamper-proof identity verification while avoiding on-chain storage of sensitive user information.
- **AI-Driven Identity Verification and Fraud Prevention:** Incorporate artificial intelligence techniques such as facial recognition, liveness detection, and document analysis to automate identity verification and prevent impersonation, forgery, and deepfake-based attacks.
- **Verifiable Credential Issuance and Validation:** Enable trusted issuers such as government bodies, educational institutions, or organizations to issue digitally signed verifiable credentials that users can securely store and selectively share with verifiers.
- **Consent-Based and Privacy-Preserving Verification:** Implement selective disclosure mechanisms that allow users to share only the required identity attributes during verification, ensuring privacy while maintaining verification accuracy.

- **Secure Off-Chain Data Storage:** Store identity documents and sensitive data in encrypted off-chain storage systems, using blockchain only as a verification and audit layer to ensure scalability and compliance with data protection standards.

- **Scalable Verification for Real-World Use Cases:** Support identity verification for domains such as banking, digital KYC, e-governance, education, and online platforms, enabling fast and reusable verification across multiple services.

- **Modular and Extensible System Architecture:** Design a modular architecture that separates identity management, AI verification, blockchain services, and storage layers, allowing future enhancements such as zero-knowledge proofs, cross-chain identity, or biometric extensions.

## 1.3 Project Scope:

The AI-Enhanced Decentralized Identity Verification System is a secure, web-based platform designed to enable privacy-preserving identity creation, management, and verification using Self-Sovereign Identity principles. The system allows users to own and control their digital identities while enabling trusted organizations to verify identity claims through blockchain-based proofs and AI-driven validation mechanisms. It eliminates centralized data storage and repetitive verification processes, making digital identity verification more secure, efficient, and scalable.

**Key Aspects of the Project Scope :**

- **Decentralized Identity Creation and Management:**
  - Enable users to generate Decentralized Identifiers (DIDs) and manage their digital identities through encrypted identity wallets.
  - Provide full user ownership and control over identity credentials without dependency on centralized identity providers.

- **Verifiable Credential Issuance and Storage:**
  - Allow trusted issuers such as governments, universities, or organizations to issue digitally signed verifiable credentials.
  - Store credentials securely in user-controlled wallets and encrypted off-chain storage systems.

- **Blockchain-Based Trust and Auditability:**
  - Use blockchain technology to store immutable cryptographic proofs, credential hashes, and revocation status.
  - Ensure tamper-proof verification and transparent audit trails without exposing sensitive personal data on-chain.

- **AI-Driven Identity Verification and Fraud Detection:**
  - Integrate AI techniques such as facial recognition, liveness detection, and document analysis for automated identity validation.

  - Detect fraudulent attempts including impersonation, forged documents, and deepfake-based attacks.

- **Consent-Based and Privacy-Preserving Verification:**
  - Implement selective disclosure mechanisms that allow users to share only the required identity attributes during verification.
  - Ensure explicit user consent before any credential is shared with a verifier.

- **Secure Off-Chain Data Storage:**
  - Store identity documents and biometric data in encrypted off-chain storage systems such as secure cloud storage or decentralized file systems.
  - Use blockchain solely as a verification layer to maintain compliance with privacy and data protection regulations.

- **Cross-Platform and Web-Based Accessibility:**
  - Develop the system as a browser-based application accessible across modern devices without additional installations.
  - Ensure a consistent and secure user experience for individuals, issuers, and verifiers.

- **Scalability and Extensibility:**
  - Design a modular architecture that supports integration of future technologies such as zero-knowledge proofs, cross-chain identity, and biometric extensions.
  - Enable the system to scale efficiently for large numbers of users, issuers, and verification requests.

## 1.4 Project Modules :

**1.4.1 User Authentication & Wallet Integration Module:** This module is responsible for secure user authentication using blockchain wallets. It enables users to connect their crypto wallets (such as MetaMask), authenticate via cryptographic signatures, and establish secure sessions. Nonce-based signature verification ensures resistance against replay attacks, while JWT-based session management maintains authenticated access. The module also handles wallet disconnection events and session expiration to maintain system security.

**1.4.2 Decentralized Identifier (DID) Management Module:** The DID Management Module handles the creation, registration, resolution, and lifecycle management of decentralized identifiers. It generates DIDs from wallet addresses, creates DID documents, registers them on the blockchain, and supports retrieval and updates. This module ensures that each user has a unique, verifiable, and blockchain-anchored digital identity with controlled status management.

**1.4.3 Document Management Module:** This module manages the secure handling of user identity documents such as passports and national IDs. It supports document upload, validation, encryption, decentralized storage using IPFS, retrieval, versioning, and deletion. Strong encryption ensures confidentiality, while IPFS integration guarantees tamper resistance and availability of identity documents.

**1.4.4 AI-Powered Verification Module:** The AI-Powered Verification Module performs automated identity verification using artificial intelligence techniques. It includes facial recognition, liveness detection, OCR-based data extraction, and document authenticity validation. A composite scoring mechanism evaluates multiple verification signals to determine identity validity, significantly reducing fraud and manual intervention.

**1.4.5 Credential Management Module:** This module manages the full lifecycle of verifiable credentials. It enables credential issuance, hashing, blockchain anchoring, retrieval, validation, revocation, and expiry handling. Credentials issued through this module are cryptographically secure, verifiable, and reusable across different services while remaining under user control.

**1.4.6 Selective Disclosure & QR Code Module:** The Selective Disclosure Module allows users to share only specific identity attributes required for verification. It supports QR-code–based credential sharing with time-limited access and detailed access logging. This module ensures privacy-preserving identity verification while enabling fast, offline, or device-to-device verification.

**1.4.7 Blockchain Integration Module:** This module provides the core blockchain interaction layer of the system. It manages smart contract deployment, transaction execution, event listening, gas optimization, and multi-network support. The module ensures reliable communication between off-chain services and on-chain identity registries.

**1.4.8 User Dashboard & Interface Module:** The User Interface Module provides a user-friendly web interface for identity management and verification workflows. It includes dashboards, credential viewers, verification wizards, activity logs, and settings management. This module ensures a smooth and intuitive experience for users interacting with decentralized identity services.

**1.4.9 Admin & Management Module:** The Admin Module enables system administrators to manage users, configurations, verification reviews, and support requests. It provides dashboards, reporting tools, and system controls required for operational oversight and maintenance.

## 1.5 Project Hardware/Software Requirements:

**Hardware Requirements :**

- **Device:** Desktop or laptop capable of running modern web applications and AI workloads.
- **Display:** Minimum 13” screen with 1366×768 resolution (Recommended: Full HD or higher).
- **CPU:** Minimum Intel Core i3 or equivalent (Recommended: Intel Core i5/i7 or AMD Ryzen 5/7).

- **RAM:** At least 4 GB (Recommended: 8 GB or more for smooth AI processing and multitasking).
- **Storage:** Minimum 20 GB free disk space for development tools, dependencies, and data storage.
- **Internet Connection:** Stable broadband connection (minimum 5 Mbps recommended for blockchain and API interactions).
- **Peripherals:** Webcam required for biometric verification (face recognition and liveness detection).

**Software Requirements :**

- **Operating System:** Windows 10/11, macOS (Catalina or later), or Linux (Ubuntu 20.04+).

- **Development Environment:**

  - **Programming Language:** JavaScript / TypeScript, Python
  - **IDE:** Visual Studio Code
  - **Version Control:** Git (optional but recommended)

- **Libraries and Tools:**

  - **Frontend:** React.js, Next.js, TailwindCSS
  - **Backend & APIs:** Node.js (Express / NestJS), REST APIs
  - **Blockchain Technologies:** Ethereum / Polygon (Testnet), Solidity
  - **Storage & Database:** MongoDB, IPFS

# Chapter 2: Literature review

## 2.1 Decentralized Identifiers and Self-Sovereign Identity

**Authors:** W3C Decentralized Identifier Working Group
**Published In:** W3C Recommendation (2022)

**Summary:** This specification introduces Decentralized Identifiers (DIDs) as a new type of identifier that enables verifiable, decentralized digital identity. The document defines DID syntax, DID documents, and resolution mechanisms that allow identities to be controlled by users rather than centralized authorities. It establishes the foundation for Self-Sovereign Identity (SSI) systems by removing dependency on central identity providers.

**Relevance:** This work forms the core theoretical foundation of the proposed system by defining how decentralized identities are created, resolved, and managed. The project directly adopts W3C DID standards to ensure interoperability and compliance with global identity frameworks.

## 2.2 Blockchain-Based Digital Identity Management

**Authors:** Ferdous, Chowdhury, and Alassafi
**Published In:** IEEE Access (2019)

**Summary:** The paper presents a comprehensive survey of blockchain-based identity management systems. It discusses how blockchain can be used to ensure immutability, transparency, and decentralized trust in identity verification while highlighting challenges such as privacy, scalability, and data storage limitations.

**Relevance:** This study supports the use of blockchain as a trust layer rather than a data storage mechanism. The proposed system follows this approach by storing only cryptographic hashes and verification proofs on-chain, addressing the privacy and scalability concerns discussed in the paper.

## 2.3 Verifiable Credentials for Secure Identity Sharing

**Authors:** Reed et al
**Published In:** IEEE Security & Privacy Magazine (2020)

**Summary:** This paper introduces the concept of verifiable credentials, where identity claims are digitally signed by trusted issuers and can be independently verified without contacting the issuer. It emphasizes selective disclosure and user consent as key mechanisms for privacy-preserving identity verification.

**Relevance:** The project adopts verifiable credentials as a primary method for identity proof exchange. The selective disclosure mechanism described in this work directly influences the design of the credential sharing and QR-based verification features of the system.

## 2.4 Biometric Authentication and Liveness Detection Using AI

**Authors:** Galbally, Marcel, and Fierrez
**Published In:** IEEE Transactions on Information Forensics and Security (2014)

**Summary:** This research explores biometric authentication techniques and the importance of liveness detection in preventing spoofing attacks such as photo, video, and replay attacks. It analyzes various face recognition and liveness detection methods used in secure identity verification systems.

**Relevance:** This paper justifies the integration of AI-based facial recognition and liveness detection in the proposed system to prevent impersonation and deepfake-based fraud during identity verification.

# Chapter 3: System Analysis & Design

## 3.1 Comparison of Existing Applications with your Application:

| Feature | Centralized KYC / National ID (e.g., banks, Aadhaar) | Existing SSI Platforms (Sovrin / Polygon ID / Civic) | AI-Enhanced DID System (proposed) |
|---|---|---|---|
| **Who controls identity** | Central authorities / provider-controlled (issuer + custodian). | User-first / holder-controlled DIDs. | User-owned wallets + DIDs (SSI). Users control credentials and sharing. |
| **Where data lives** | Personal data stored centrally (databases); documents often held by provider → higher breach risk. | Minimal on-chain; credentials held by user; some platforms use on-chain anchors or ZK proofs. | Off-chain encrypted storage for documents (IPFS/cloud); only hashes/proofs on-chain. |
| **Verification method** | Provider-driven KYC checks, manual or centralized automated checks. | Cryptographic VC verification (DID + VC), sometimes ZK proofs (Polygon ID). | VC verification + AI biometric/document checks (face match, liveness, OCR) before/alongside cryptographic checks. |
| **Privacy & selective disclosure** | Low - providers typically require full documents and store data. | High - selective disclosure is supported, especially with ZK approaches (Polygon ID). | High - selective disclosure + QR/time-limited proofs + ZKP-friendly design; user consent enforced. |
| **Fraud / spoof protection** | Relies on back-office checks; vulnerable to forged docs and centralized breaches. | Cryptographic integrity reduces some fraud; fewer active anti-deepfake measures out of the box. | Active AI fraud detection (liveness, forgery detection, composite scoring) + cryptographic proofs. |
| **Revocation / auditability** | Revocation handled internally (blacklists); audit trails centralized. | On-chain revocation/status models supported via registries/ledgers. | On-chain credential hash + revocation lists; auditable blockchain logs plus off-chain audit logs. |
| **Scalability & cost** | Scales but incurs heavy infra and compliance costs; central ops overhead. | Many SSI projects use testnets or permissioned ledgers to manage cost | Hybrid design: low on-chain storage (low gas), off-chain scalable storage |
| **Regulatory readiness** | Mature for compliance (banks, gov) but raises privacy concerns (centralized risk). | Emerging; some pilots and govt interest, but adoption varies. | Designed for compliance (consent, minimal on-chain PII); may require legal/regulatory onboarding for banks/gov. |

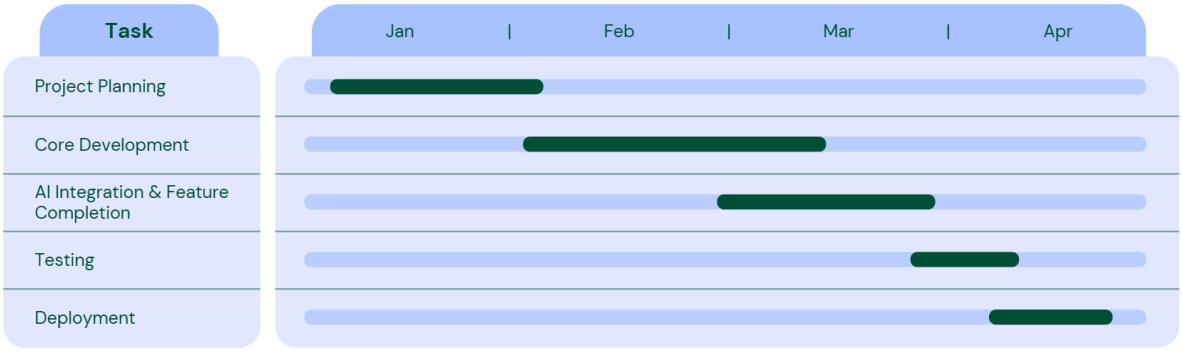
## 3.2 Project feasibility study :

**3.2.1 Technical Feasibility:** The proposed system is technically feasible as it is built using modern, well-supported technologies such as blockchain platforms, web frameworks, and AI libraries. Decentralized identity standards like DIDs and Verifiable Credentials are already defined by W3C and are supported by existing blockchain networks. AI-based face verification, liveness detection, and document analysis can be implemented using open-source machine learning frameworks. The modular architecture ensures scalability, easy maintenance, and smooth integration of future enhancements.

**3.2.2 Operational Feasibility:** The proposed AI-Enhanced Decentralized Identity Verification System is operationally feasible as it is designed to integrate smoothly into existing digital workflows without requiring major changes to user behavior or organizational processes. The system provides a user-friendly web interface that allows individuals to manage their identities and complete verification steps with minimal technical knowledge. Automated AI-based verification and blockchain-backed validation reduce manual intervention, making day-to-day operations efficient and reliable.

**3.2.3 Economic Feasibility:** The project is economically feasible as it relies primarily on open-source tools, blockchain test networks, and free or low-cost development platforms. There are minimal infrastructure costs during development, and cloud resources can be scaled based on usage. The system also has strong commercial potential in areas such as digital KYC, banking, e-governance, and education, making it viable for future real-world deployment.

## 3.3 Project Timeline chart:



## 3.4 Detailed Modules Description:

### 3.4.1 User Authentication & Wallet Integration Module

**Purpose:**

To authenticate users securely using blockchain wallets and establish trusted user sessions without traditional username–password mechanisms.

**Working Methodology:**

- The user connects a blockchain wallet (e.g., MetaMask) to the application.
- The system generates a unique nonce and requests the user to sign it using their wallet's private key.
- The signed message is verified on the backend to confirm wallet ownership.
- Upon successful verification, a JWT token is generated for session management.
- The module continuously monitors wallet connection status and invalidates sessions on disconnect.

**Expected Output:**

- Secure user login without passwords
- Verified wallet ownership
- Active authenticated user session

### 3.4.2 Decentralized Identifier (DID) Management Module

**Purpose:**

To create, register, manage, and resolve decentralized identifiers that uniquely represent users in the system.

**Working Methodology:**

- A DID is generated from the user's wallet address using standardized DID methods.
- A DID document containing public keys and service endpoints is created.
- The DID and its metadata are registered on the blockchain via a smart contract.
- DID resolution retrieves the DID document for verification or updates.
- DID status (active/inactive) is maintained to support revocation or suspension.

**Expected Output:**

- Unique blockchain-registered DID for each user
- Resolvable DID document
- Verified decentralized identity

### 3.4.3 Document Management Module

**Purpose:**

To securely handle identity documents required for verification while preserving privacy.

**Working Methodology:**

- Users upload identity documents such as passport or national ID.
- Files are validated for format and size.
- Documents are encrypted using AES-256 encryption.
- Encrypted files are stored in IPFS or secure off-chain storage.
- Document hashes are generated for integrity verification.
- Versioning allows tracking of document updates.

**Expected Output:**

- Securely stored encrypted documents
- IPFS content identifiers (CIDs)
- Tamper-proof document integrity

### 3.4.4 AI-Powered Verification Module

**Purpose:**

To automatically verify user identity and prevent fraud using artificial intelligence.

**Working Methodology:**

- Face verification compares live camera input with ID photo using deep learning models.
- Liveness detection ensures the user is physically present (blink, motion checks).
- OCR extracts textual data from identity documents.
- Document authenticity checks detect tampering or forgery.
- A composite verification score is generated based on all AI checks.

**Expected Output:**

- AI verification result (Pass / Fail)
- Confidence score for identity validity
- Fraud detection alerts

### 3.4.5 Credential Management Module

**Purpose:**

To issue, manage, validate, and revoke verifiable credentials.

**Working Methodology:**

- After successful verification, a verifiable credential is created.
- Credential data is hashed and digitally signed by the issuer.
- The hash is stored on the blockchain for immutability.
- Credentials are stored in the user’s wallet.

- Revocation and expiry mechanisms update credential status on blockchain.

**Expected Output:**

- Digitally signed verifiable credentials
- Blockchain-anchored credential proofs
- Valid or revoked credential status

### 3.4.6 Selective Disclosure & QR Code Module

**Purpose:**

To allow privacy-preserving sharing of identity attributes.

**Working Methodology:**

- Users select specific attributes to share (e.g., age, citizenship).
- A QR code is generated encoding the selected credential proof.
- QR codes are time-limited to prevent misuse.
- Verifiers scan and validate the QR code against blockchain proofs.
- All access events are logged for audit purposes.

**Expected Output:**

- QR-based credential sharing
- Partial identity disclosure
- Secure and auditable verification

### 3.4.7 Blockchain Integration Module

**Purpose:**

To provide decentralized trust and immutability using blockchain.

**Working Methodology:**

- Smart contracts are deployed for DID and credential registries.
- Transactions are executed for DID registration and credential anchoring.
- Event listeners monitor contract updates.
- Gas optimization techniques reduce transaction cost.
- Multi-network support allows testnet and mainnet usage.

**Expected Output:**

- Immutable blockchain records
- Trusted credential verification
- Transparent audit trail

### 3.4.8 User Dashboard & Interface Module

**Purpose:**

To provide an intuitive interface for users to manage identity and verification workflows.

**Working Methodology:**

- Dashboard displays user identity status and credentials.
- Verification wizard guides users through identity checks.
- Credential viewer allows inspection and sharing of credentials.
- Activity logs display verification history.
- Settings allow user preferences and security controls.

**Expected Output:**

- User-friendly web interface
- Clear identity and credential visibility
- Smooth verification experience

### 3.4.9 Admin & Management Module

**Purpose:**

To provide administrative control over the system.

**Working Methodology:**

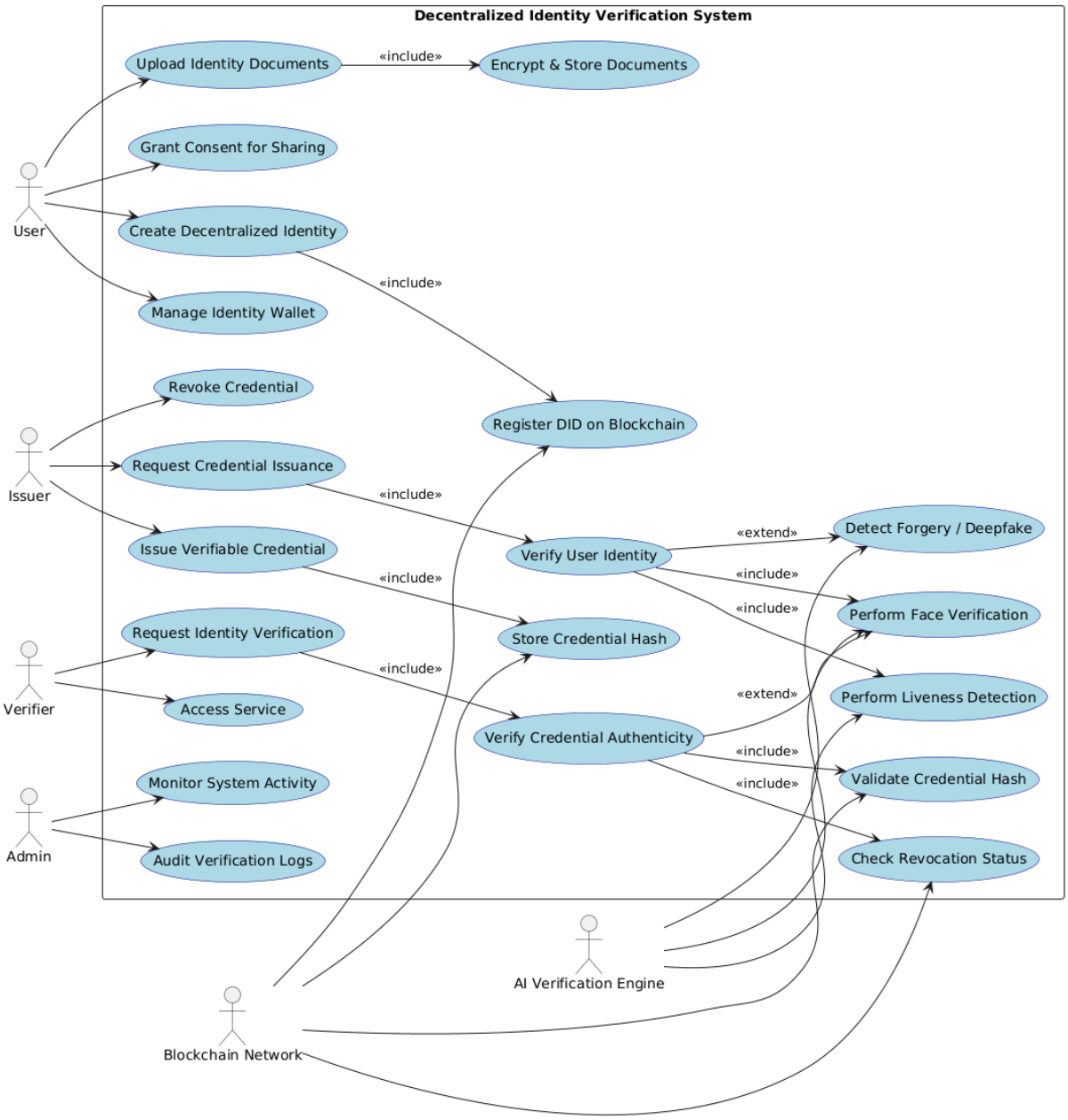
- Admin dashboard manages users and configurations.
- Verification reviews handle exceptional cases.
- Support tickets manage user issues.
- Reports summarize system usage and compliance.

**Expected Output:**

- Centralized system administration
- Effective system governance

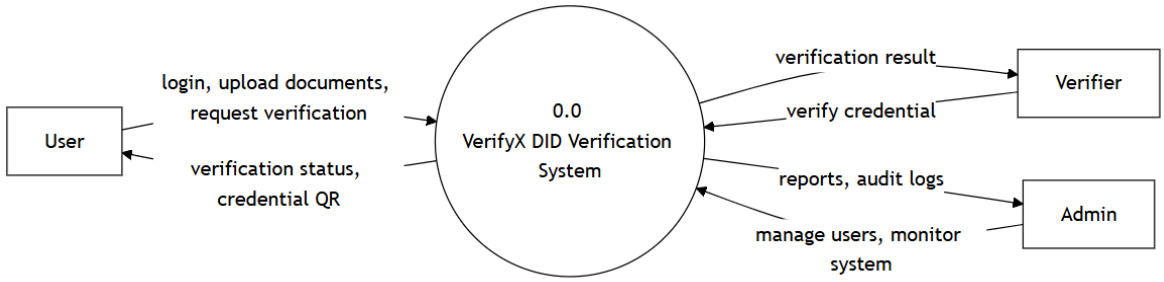
## 3.5 Project SRS:

### 3.5.1 Use Case Diagrams:

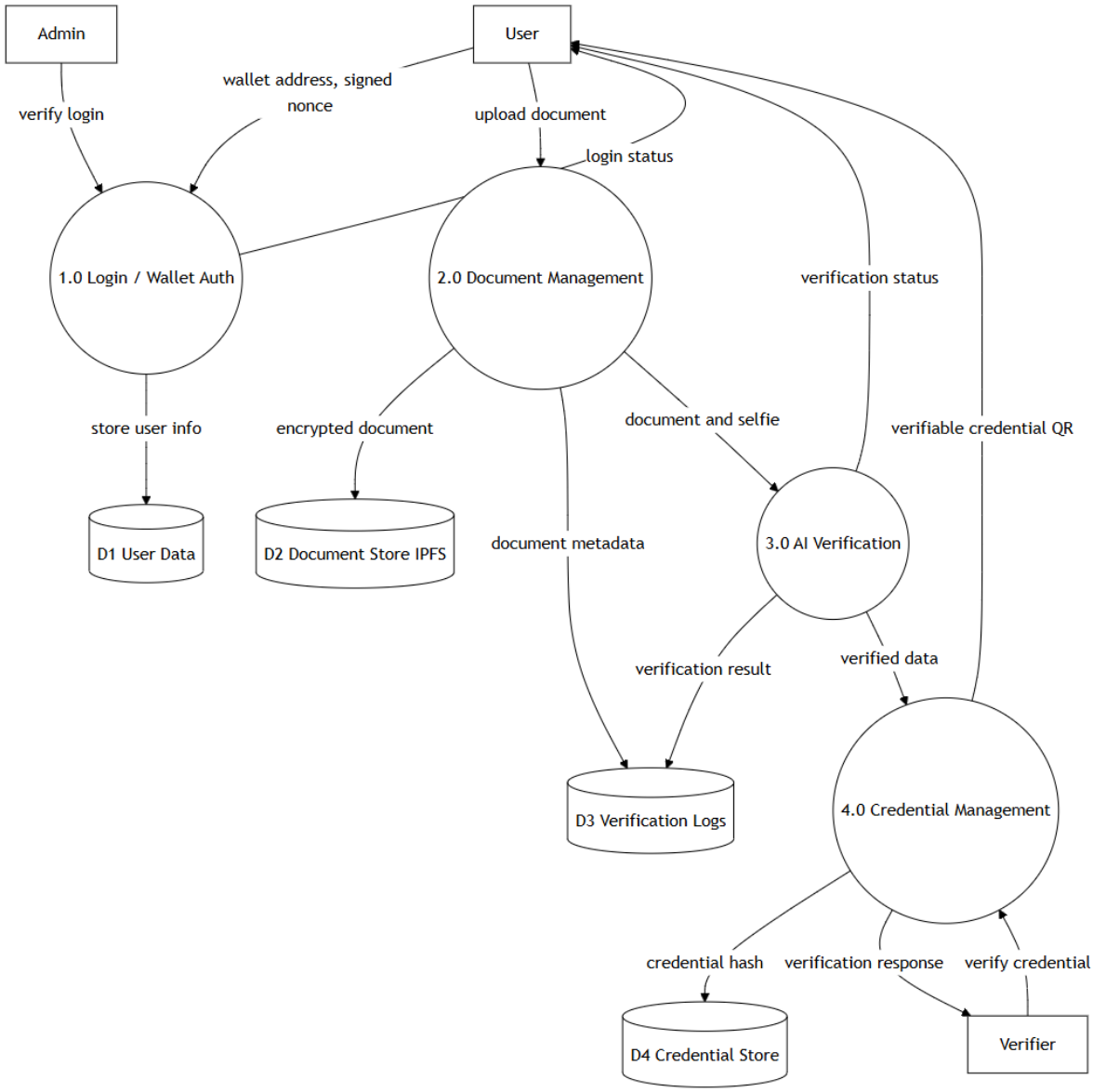


## 3.5.2 Data Flow Diagrams:

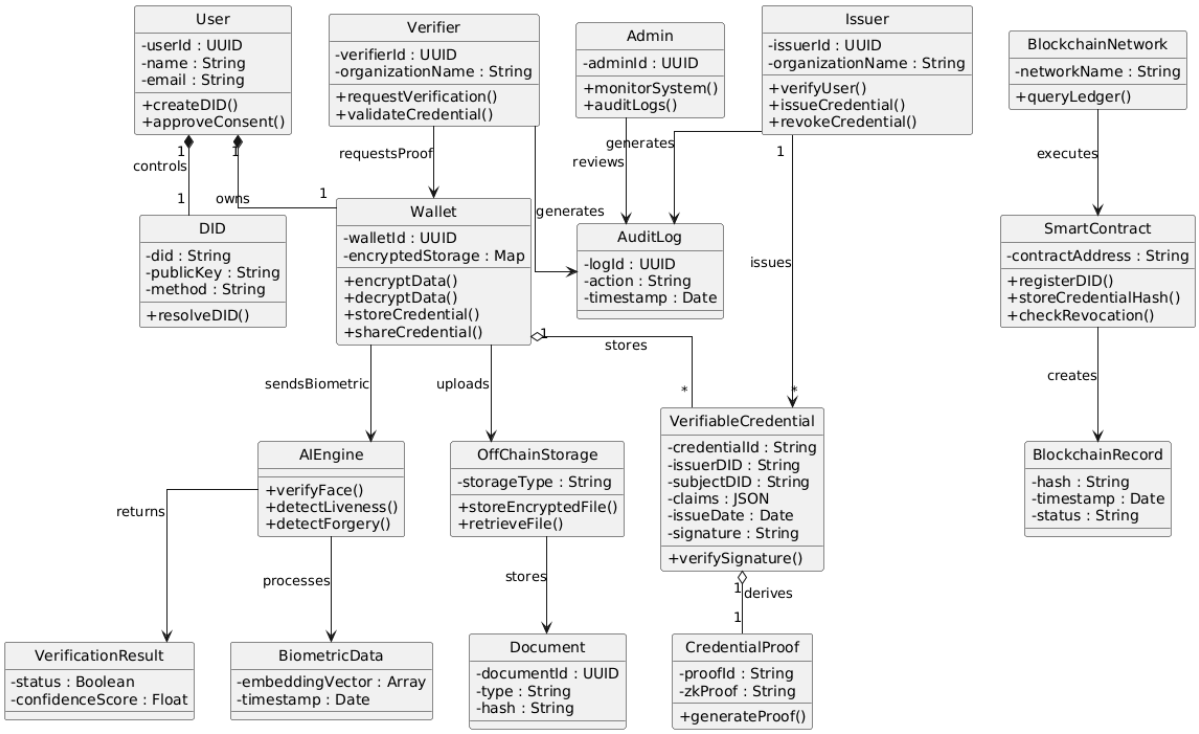
### DFD level 0:



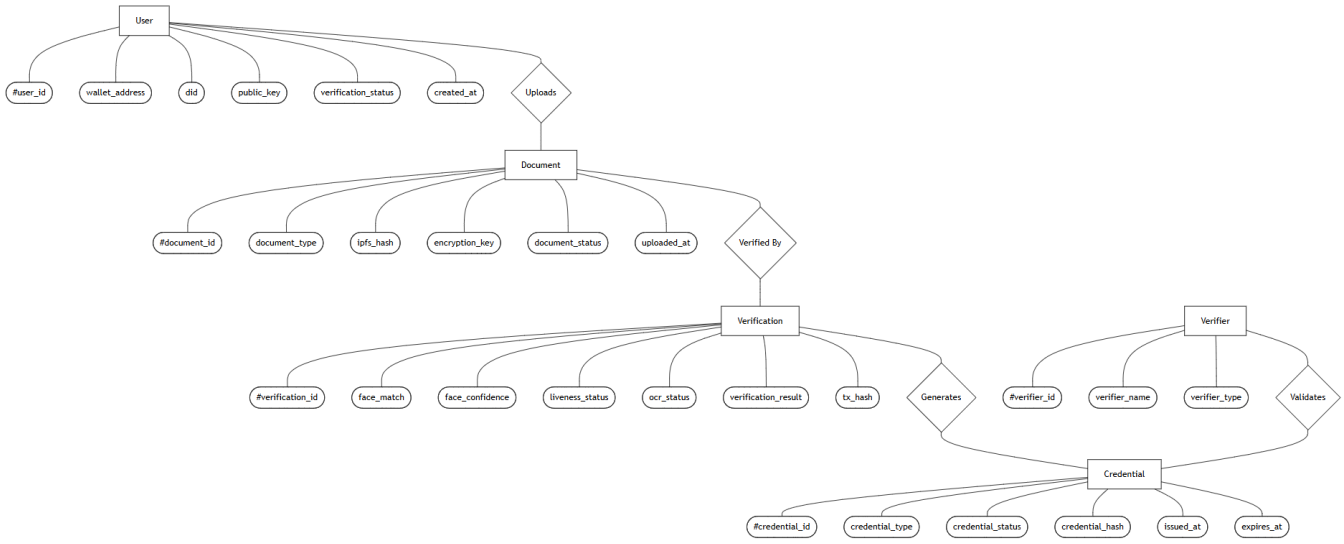

### DFD level 1:



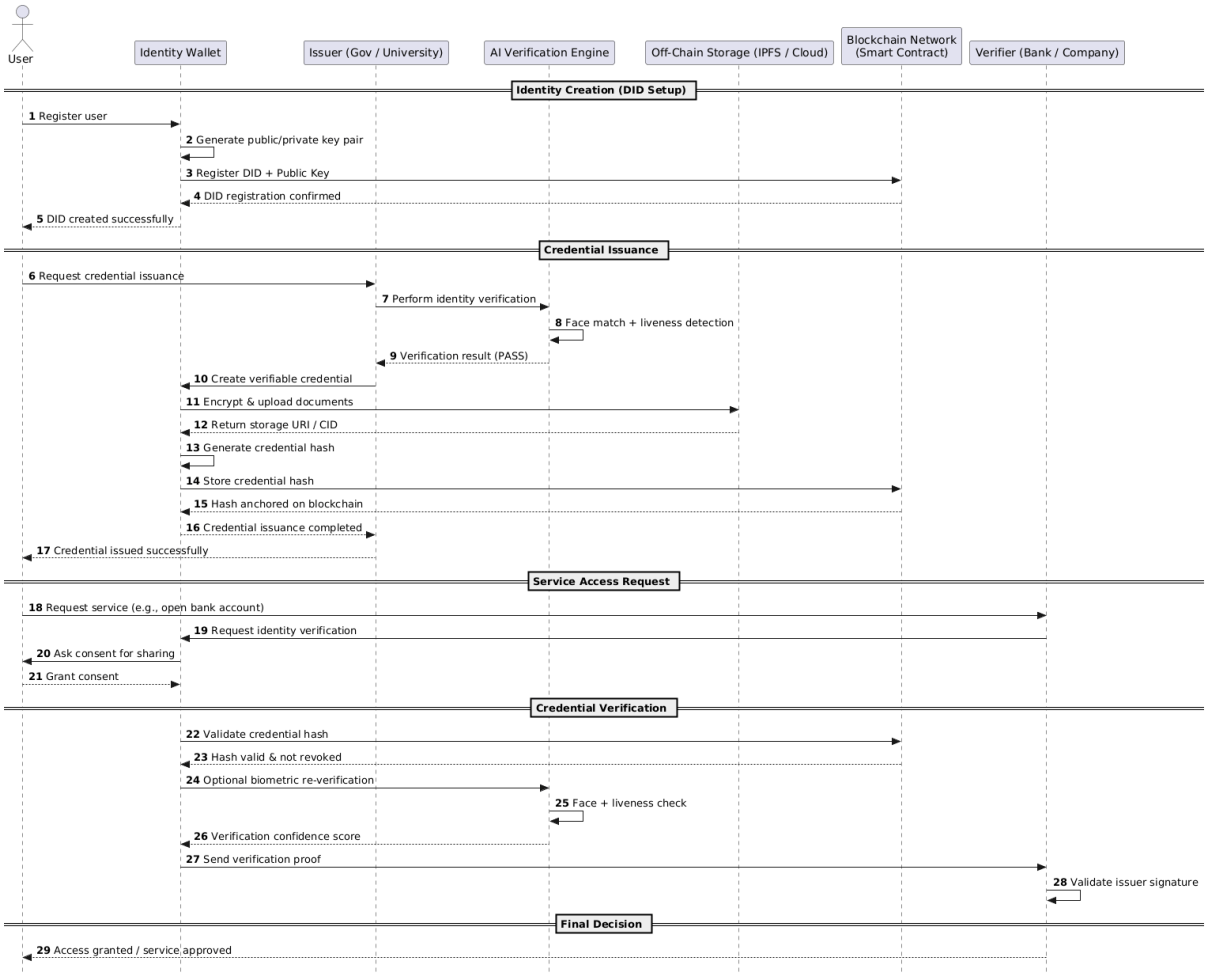
## 3.5.3 Class diagram:



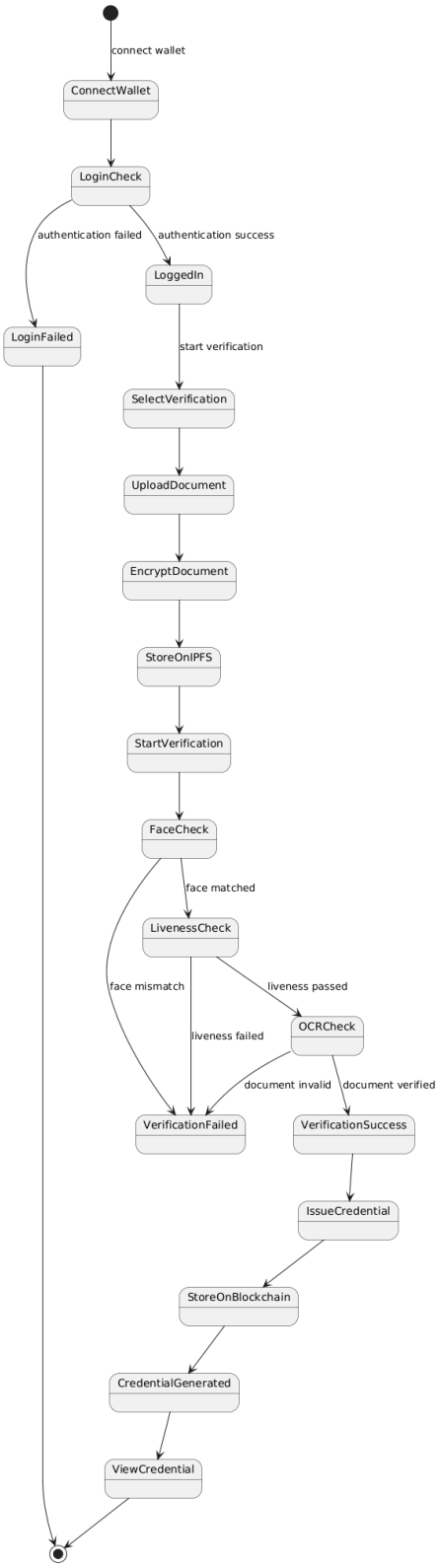
## 3.5.4 Entity Relationship Diagrams:

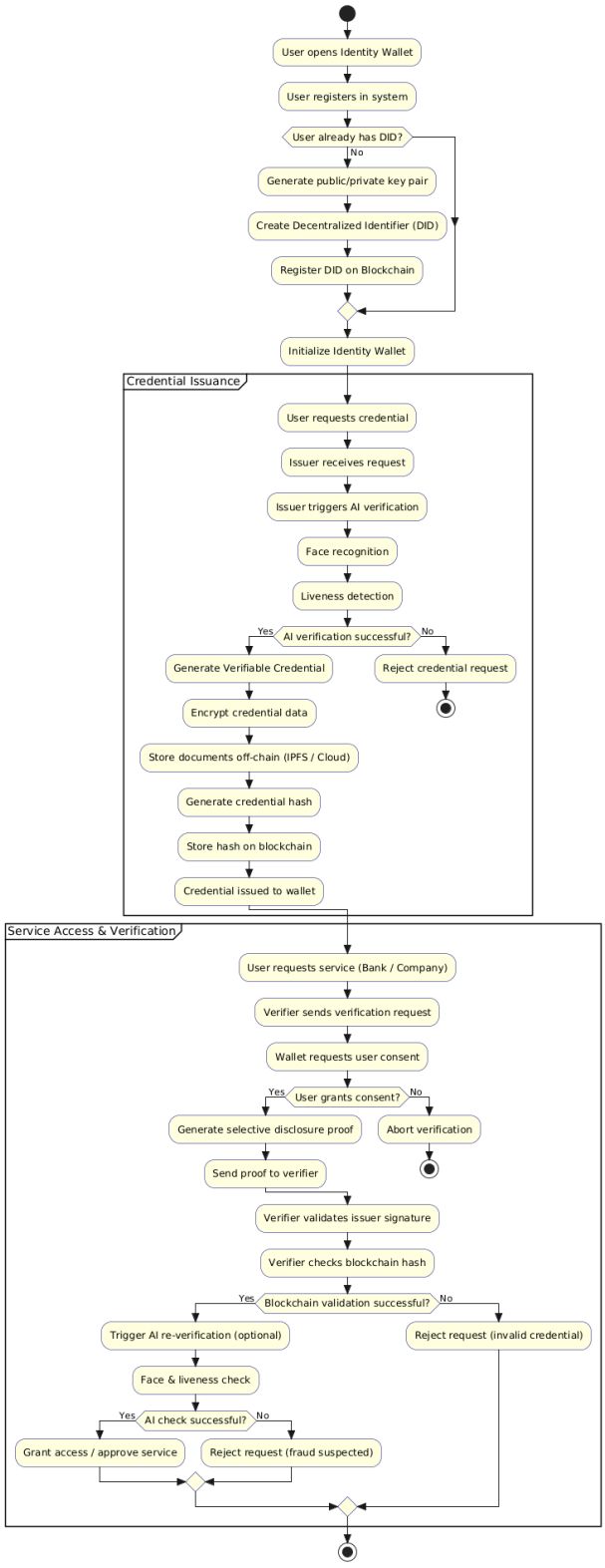


## 3.5.5 Sequence Diagrams:



## 3.5.6 State/Activity Diagram:



## 3.6 Data Dictionary:

### User Table (MongoDB)

| Field Name | Data Type | Description | Constraints |
|---|---|---|---|
| walletAddress | String | Blockchain wallet address of the user | Unique, Not Null |
| did | String | Decentralized Identifier associated with the user | Unique |
| publicKey | String | Public cryptographic key derived from wallet | Not Null |
| createdAt | Date | Account creation timestamp | Auto-generated |
| lastLogin | Date | Last successful login timestamp | Nullable |
| verificationStatus | String | Current identity status (pending, verified, rejected) | Enum |
| credentialIds | Array | List of credential IDs issued to the user | Nullable |

### Document Table (MongoDB)

| Field Name | Data Type | Description | Constraints |
|---|---|---|---|
| userId | ObjectId | Reference to User collection | Foreign Key |
| type | String | Document type (passport, driver_license, national_id) | Enum |
| ipfsHash | String | IPFS content identifier of encrypted document | Unique |
| encryptionKey | String | Encrypted AES key used for document encryption | Encrypted |
| status | String | Document status (uploaded, verified, rejected) | Enum |
| uploadedAt | Date | Document upload timestamp | Auto-generated |
| verifiedAt | Date | Document verification timestamp | Nullable |

### Verification Table (MongoDB)

| Field Name | Data Type | Description | Constraints |
|---|---|---|---|
| userId | ObjectId | Reference to User | Foreign Key |
| documentId | ObjectId | Reference to Document | Foreign Key |
| status | String | Verification result (success, failed, pending) | Enum |
| faceVerification.match | Boolean | Face match result | Not Null |
| faceVerification.confidence | Number | Face match confidence score | 0–100 |
| faceVerification.model | String | AI model used for face verification | Nullable |
| livenessDetection.isLive | Boolean | Liveness check result | Not Null |
| livenessDetection.confidence | Number | Liveness confidence score | 0–100 |
| livenessDetection.challenge | String | Challenge type used | Nullable |
| documentOCR.extracted | Boolean | OCR success status | Not Null |
| documentOCR.data | Object | Extracted document fields | Nullable |

| | | | |
|---|---|---|---|
| documentOCR.confidence | Object | Confidence per extracted field | Nullable |
| credentialHash | String | Hash of issued credential | Unique |
| blockchainTxHash | String | Blockchain transaction hash | Unique |
| createdAt | Date | Verification start time | Auto-generated |
| completedAt | Date | Verification completion time | Nullable |

**Credential Table (MongoDB)**

| Field Name | Data Type | Description | Constraints |
|---|---|---|---|
| userId | ObjectId | Reference to User | Foreign Key |
| verificationId | ObjectId | Reference to Verification | Foreign Key |
| type | String | Credential type (KYC, Identity, AgeProof) | Enum |
| status | String | Credential state (active, revoked, expired) | Enum |
| attributes | Array | Attributes included in credential | Nullable |
| hash | String | Credential hash stored on blockchain | Unique |
| blockchainTxHash | String | Blockchain transaction hash | Unique |
| issuedAt | Date | Credential issue timestamp | Auto-generated |
| expiresAt | Date | Credential expiration date | Nullable |

**DID Registry Contract**

| Field Name | Data Type | Description |
|---|---|---|
| controller | address | Wallet address controlling the DID |
| publicKey | string | Public key linked to the DID |
| createdAt | uint256 | DID creation timestamp |
| isActive | bool | DID active/inactive status |

**Credential Registry Contract**

| Field Name | Data Type | Description |
|---|---|---|
| credentialHash | bytes32 | Hash of verifiable credential |
| issuer | address | Issuer wallet address |
| subject | address | Credential owner wallet |
| issuedAt | uint256 | Issue timestamp |
| expiresAt | uint256 | Expiration timestamp |
| isRevoked | bool | Revocation status |