GENUINEGRADS - BLOCKCHAIN-BASED CERTIFICATE VERIFICATE SYSTEM WITH COMPRESSED NFTS AND ZERO- KNOWLEDGE PROOFS

M.H.M.Shahadh

E2240185

Bachelor of Information Technology

Center for Open and Distance Learning

University of Moratuwa
Sri Lanka

08/2025

Declaration of the candidate & Supervisor

"I declare that this is my own work and this report does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my report, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works."

Signature: …….. Date: 26/08/2025

The above candidate has carried out work for the Bachelors of Information Technology Degree report under my supervision.

Name of the supervisor: Mr. Menan Velayuthan.

Signature of the supervisor: …….. Date: 26/08/2025

Acknowledgments

This project is being carried out under the supervision of professionals from the University of Moratuwa as part of my final-year project. I would like to express my sincere appreciation and gratitude for the guidance and support provided by all individuals involved in its development. In particular, the documentation and development of this project have been closely supervised by Mr. Menan Velayuthan, to whom I am especially grateful for his valuable contributions.

Abstract

GenuineGrads addresses the prevalent issue of academic credential fraud in South Asia by leveraging blockchain technology on the Solana network. The platform issues university certificates as compressed NFTs (cNFTs) using Metaplex's Bubblegum v2, which allows for cost-effective and scalable minting. These certificates are anchored on-chain with state compression (Merkle roots) to ensure integrity, while the detailed certificate metadata is stored off-chain on IPFS or Arweave. The system also includes an on-chain revocation mechanism via burnV2.

For efficient verification, GenuineGrads utilizes the Helius DAS (Digital Asset Standard) to query compressed assets through a unified API. This is complemented by Webhooks, which provide real-time eventing and monitoring. Additionally, the platform incorporates Zero-Knowledge Proofs (ZKPs) to enable selective and private disclosure of information, such as confirming a student's GPA meets a certain threshold, without exposing their full academic record. This approach aims to restore trust in the hiring and admissions processes by providing a secure and verifiable method for managing academic credentials.

By this interim milestone, all user interfaces are implemented in Next.js, diagrams are complete, and the MVP is in active development: university registration, bulk cNFT issuance, revocation (burn), and QR/ID-based verification against Bubblegum v2 and the DAS index. A decision analysis is underway between MPL- Bubblegum JavaScript SDK (fastest path) and Rust/Anchor CPI (hard-enforced on-chain policies) for production-grade issuance control. The design aligns with Solana’s state compression for scale [3], [21], Metaplex Bubblegum v2 lifecycle operations [2],[7],[13],[16], DAS for read/proof [5],[8],[19], and W3C VC 2.0 data model [18].

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
API	Application Programming Interface
BTC	Bitcoin
cNFT	Compressed Non-Fungible Token
CSV	Comma-Separated Values
DB	Database
DAS	Digital Asset Standard
DID	Decentralized Identifier
DNS	Domain Name System
EBSI	European Blockchain Services Infrastructure
ETH	Ethereum
ERD	Entity Relationship Diagram
EVM	Ethereum Virtual Machine
EU	European Union
GPA	Grade Point Average
GovTech SG	Government Technology Agency of Singapore
IPFS	InterPlanetary File System
JSON	JavaScript Object Notation
JS SDK	JavaScript Software Development Kit
KYC	Know Your Customer
MVP	Minimum Viable Product
NFT	Non-Fungible Token
NIC	National Identity Card
NEM	New Economy Movement blockchain
NAD	National Academic Depository (India)
MOHE	Ministry of Higher Education
OA	OpenAttestation
OSS	Open-Source Software

PII	Personally Identifiable Information
PoH	Proof of History (Solana)
QR	Quick Response (code)
RLS	Row-Level Security
RPC	Remote Procedure Call
SSO	Single Sign-On
UI	User Interface
UX	User Experience
W3C	World Wide Web Consortium
ZK	Zero-Knowledge
ZKP	Zero-Knowledge Proof

LIST OF APPENDICES

1. Introduction

1.1. Background

The prevalence of fraudulent academic credentials, such as fake degrees and forged documents, is a well-established global issue. A UNESCO/IIEP analysis indicates the existence of a worldwide market for "degree mills," which poses significant risks to the quality of the workforce and public safety [22]. In South Asia, reports suggest that fake degrees and ghostwriting services are readily available, a situation attributed to both regulatory shortcomings and strong demand [6]. Traditional methods for mitigating this problem, such as manual checks by registrars and evaluator networks, can reduce the risk of fraud. However, these methods are often slow and cannot be scaled efficiently to handle large numbers of verifications. Additionally, they frequently necessitate the over-disclosure of personal data, such as a full transcript, even when only a single credential needs to be validated. A truly effective solution must therefore integrate public verifiability with privacy-preserving selective disclosure and must be cost-efficient and scalable to accommodate national-level verification needs.

Blockchain technology creates publicly verifiable and tamper-proof records, and the Solana network enhances this with high-speed and low-latency performance. A key innovation is state compression, which drastically cuts storage costs. It works by storing a single cryptographic "fingerprint," or Merkle root, on the blockchain to represent a large amount of off-chain data. This allows for efficient verification of off-chain information against the on-chain root [3],[21]. This process is made practical through Metaplex Bubblegum v2, which uses this technique to create compressed NFTs (cNFTs) [16]. Tools like the Digital Asset Standard (DAS) and Webhooks further support this system by providing a unified way to read data and receive real-time updates [5],[8],[19].

1.2. Problem statement

Current certificate issuance/verification processes are slow, centralized, and lack robust privacy protections. Verifiers must either rely on easily falsified PDF documents or undertake time-consuming manual checks. Students are frequently required to overshare personal data to validate even simple claims, such as confirming they have a degree or that their GPA meets a certain threshold. Furthermore, issuing institutions lack affordable methods for bulk issuance and a universally verifiable system for revoking credentials.

1.3. Motivation & significance of project

This project aims to create a trustworthy and privacy-focused credential platform that would directly enhance the efficiency of the labor market and reduce the financial burden of fraud for both educational institutions and employers.

In Sri Lanka and across South Asia, this verifiable, low-friction system could streamline critical processes such as public-sector recruitment, applications for jobs abroad, and professional licensing. This aligns perfectly with the Sri Lankan government's recent initiative to digitalize the public sector. The platform would not only support this goal but also promote data minimization and transparent credential revocation.

Academically, this project offers a significant contribution by providing a clear implementation blueprint. It integrates several cutting-edge technologies, including Solana state compression, compressed NFT (cNFT) lifecycle management, Helius DAS/Webhooks, and Zero-Knowledge (ZK) predicate proofs, into a cohesive and standards-compliant credentialing system.

1.4. Aims and objectives

Aim :

The GenuineGrads project is building a secure, blockchain-based system for issuing and verifying academic credentials. Using Non-Fungible Tokens (NFTs) and Zero-Knowledge Proofs (ZKPs), the platform guarantees the authenticity of certificates while protecting user privacy. The system is designed to be highly scalable, leveraging Solana's compressed NFTs (cNFTs) for cost-effective storage. This architecture provides universities, graduates, and verifiers (such as employers) with an immutable record of academic achievements and a streamlined process for issuance, sharing, and validation. A key feature is the ability for instant revocation of credentials, providing a robust solution for managing the entire lifecycle of a certificate.

By creating a privacy-preserving credential verification system with standards-aligned presentations, GenuineGrads aims to eliminate certificate fraud, simplify the verification process, and give students greater control over their academic data. This innovative approach provides a powerful and secure solution for credential management on a national scale.

Objectives:

1. Build issuer, holder, and verifier portals.
2. Implement cNFT issuance on Bubblegum v2 and burnV2 revocation.
3. Integrate DAS for proof- backed verification.
4. Build ZK selective- disclosure predicates (e.g., GPA ≥ X)
5. Instant and trustless verification
6. Develop user portals for all stakeholders
7. Ensure scalability and performance

2. Literature review

How to Read This Section: We begin with an in-depth analysis of each system, examining dimensions such as architecture, issuance, identity and trust, storage, revocation, verification user experience, privacy and selective disclosure, scalability and operations, developer tooling, and adoption. We then synthesize these findings through comparative artifacts, including Table 1, which provides a feature-by-feature overview (with the full comparison available in Appendix F).

Systems examined

1. Blockcerts (MIT/Learning Machine / Hyland Credentials) — open standard for blockchain- anchored credentials [1].

2. OpenAttestation / OpenCerts (GovTech Singapore) — Ethereum- based verifiable document & education profile [9], [15].

3. EBSI Diplomas (European Blockchain Services Infrastructure) — EU VC- based credentialing framework and reference implementations [11].

4. Malaysia e- Scroll (NEM/Catapult) — nationwide university degree verification via QR on public ledger [12].

5. India DigiLocker + NAD — national digital document wallet and academic depository with API verification (non- blockchain core) [4], [14].

2.1. Blockcerts (Open Standard)

- Architecture & Issuance : Blockcerts packages credential data (JSON) and anchors a document hash on a public blockchain (Bitcoin/Ethereum). The issuer signs the credential; recipients hold it in a wallet. Verification

recomputes the hash and checks chain inclusion and issuer keys [1].

- Identity & Trust - Trust is established via issuer key management (published DID or public keys) and optional registries; there is no universal, on-chain issuer registry. Revocation lists can be hosted off-chain or anchored on-chain depending on deployment.

- Storage - Payloads live off-chain (wallet / storage URI), with hash anchoring on-chain. This is space-efficient but relies on off-chain availability for payloads.

- Revocation - Implemented via revocation lists or status flags. It is verifiable, but UX varies by implementer; propagation latency depends on where status is stored.

- Verification UX - QR + web verifier flows exist. Public verifiability is strong, but employers must trust the resolver service to fetch payloads.

- Privacy / Selective Disclosure - Redaction is app-layer; predicate proofs (e.g., GPA $\geq$ X) are not native. VC 2.0/BBS+ or SD-JWT can be layered, but this is outside core Blockcerts.

- Scalability & Ops - Scales well because only hashes are on-chain; issuance cost depends on chain fees. No native state-compression equivalent.

- Developer Tooling - Good open-source libraries and validators; ecosystem is stable.

- Adoption & Satisfaction - Early pilots (MIT, 2017) and vendor roll-outs show positive employer acceptance for QR/web verification. Privacy limitations remain a common critique.

Pros	Cons
Open, battle-tested	No native selective disclosure.
Chain-agnostic,	Revocation UX consistency varies.
Cheap on-chain use.	Requires off-chain payload availability.

Table 1: Pros and Cons of Blockcerts

2.2. OpenAttestation / OpenCerts (GovTech SG)

- Architecture & Issuance - Uses Ethereum smart contracts and Merkle roots for batches of documents. Documents are signed, anchored, and published; issuers/identities are registered through DNS-TXT or DID methods. OpenCerts is the education profile on OpenAttestation [9], [15].

- Identity & Trust - Stronger issuer identity via a verified registry (DNS ownership, DID). Clear document store and revocation models.

- Storage - Documents typically stored via object storage with signed URLs; hashes anchored on Ethereum. OA supports both wrapped documents and VC formats.

- Revocation - Clear revocation (e.g., document store revoke). Employers can instantly see revoked status if resolvers are up-to-date.

- Verification UX - Production-grade web verifiers; QR flows common in SG education ecosystem. Integrations exist for MOE institutions.

- Privacy / Selective Disclosure - Redaction or disclosure is document-level; predicate proofs not native. VC 2.0 + BBS+/SD-JWT can be layered with

additional infra.

- Scalability & Ops - Batch anchoring with Merkle roots reduces gas. Still subject to EVM fee dynamics; not optimized for Solana-style state compression.

- Developer Tooling - Excellent docs, schemas, reference verifiers, and a rich community.

- Adoption & Satisfaction - Gov-backed; high trust within SG, positive user satisfaction due to clarity and standardization.

Pros	**Cons**
Clear issuer identity model.	EVM fee exposure,
Mature tooling.	limited native selective disclosure
Government adoption.	operational complexity for non-SG adopters.

Table 2: Pros and Cons of OpenAttestation / OpenCerts

2.3. EBSI diplomas (EU VC 2.0 Stack)

- Architecture & Issuance - EBSI promotes W3C VC 2.0 credentials with EU governance. Issuers sign VCs that holders present to verifiers; distributed ledgers provide trust anchors for DIDs and status. It emphasizes standards over a single chain [11].

- Identity & Trust - Strong: issuer DIDs anchored to EU trust lists; conformity with eIDAS and governance policies. Status endpoints for revocation per VC specs.

- Storage - Holder-centric wallets store VCs; verifiers check signatures and status. No NFT anchoring; relies on VC status lists and registries.

- Revocation - Via StatusList or equivalent mechanisms; near-real-time, web-resolvable.

- Verification UX - Wallet-driven presentations (OID4VP) and QR; good user consent model (present just what's needed).

- Privacy / Selective Disclosure - Native BBS+ and SD-JWT support paths for field-level disclosure; predicate proofs require ZK integrations beyond baseline.

- Scalability & Ops - Web-scale issuance/verification; cost controlled by avoiding heavy on-chain data. No L1 anchoring of each credential.

- Developer Tooling - Growing OSS wallets, testbeds, conformance suites.

- Adoption & Satisfaction - EU pilots and national projects; strong policy legitimacy boosts acceptance.

Pros	**Cons**
Standards-first	No on-chain public record per-credential
Strong privacy and interop	Third-party verifiers must trust status endpoints and issuer governance.
Policy alignment.	

Table 3: Pros and Cons of EBSI diplomas

2.4. Malaysia e-Scroll (NEM/Catapult)

- Architecture & Issuance - Universities issue diploma records recorded on NEM (Catapult) chain. End-users scan QR to view credential details and chain proofs [12].

- Identity & Trust - University keys and a coordinating registry (MOHE/university portals). Public chain anchoring gives tamper evidence.

- Storage - Document metadata off-chain; transaction hashes on-chain. Reliant on portal availability for payloads.

- Revocation - University portals update status; verifiers re-query to see changes. On-chain signals vary by integration.

- Verification UX - Very simple QR; embraced by local employers. Clear "authentic/invalid" result page.

- Privacy / Selective Disclosure - Not native; reveals the full diploma record; no predicate proofs.

- Scalability & Ops - Adequate for national cohorts; cost profile depends on NEM fees.

- Developer Tooling - Public SDKs available; fewer modern integrations compared to OA/VC.

- Adoption & Satisfaction - Broad national rollout and awareness in MY; high verifier satisfaction from QR simplicity.

Pros	**Cons**
Nationwide visibility.	Limited privacy.

straightforward UX	Ecosystem lock- in.
public ledger anchoring	Unclear cross- border interop.

Table 4: Pros and Cons of Malaysia e- Scroll

2.5. India DigiLocker + NAD (non-blockchain baseline)

- Architecture & Issuance: Issuers (universities/boards) push verified documents to DigiLocker; NAD (National Academic Depository) acts as a trusted repository. Verifiers access via APIs/QR through government gateways [4], [14].

- Identity & Trust: Strong government KYC and issuer onboarding; trust rests on centralized governance and audit trails.

- Storage: Centralized storage of PDFs/XML; citizens control access via DigiLocker consents.

- Revocation: Repository status updates; immediate to verifiers who use APIs.

- Verification UX: Extremely simple in India; QR and web portals are ubiquitous; no wallet install needed.

- Privacy / Selective Disclosure: Document- level; no cryptographic selective disclosure or ZK predicates.

- Scalability & Ops: Web- scale; horizontally scalable government infra.

- Developer Tooling: Mature APIs and sandboxes; strong SSO integration with national IDs.

- Adoption & Satisfaction: Very high domestic adoption; employers appreciate speed and official provenance.

Pros	**Cons**
Massive adoption	Centralized trust.
Frictionless UX	No public audit trail.
Low verifier cost.	Limited privacy and cross- border interop.

Table 5: Pros and Cons of India DigiLocker + NAD

3. Project plan and initial design

3.1. Components of the system

3.1.1. Software component and tasks

- Frontend (Next.js/React)
  - University admin portal: Manage institution profile, register students, request certificate issuance (single or batch), certificate template designer, certificate revocation and analytics.

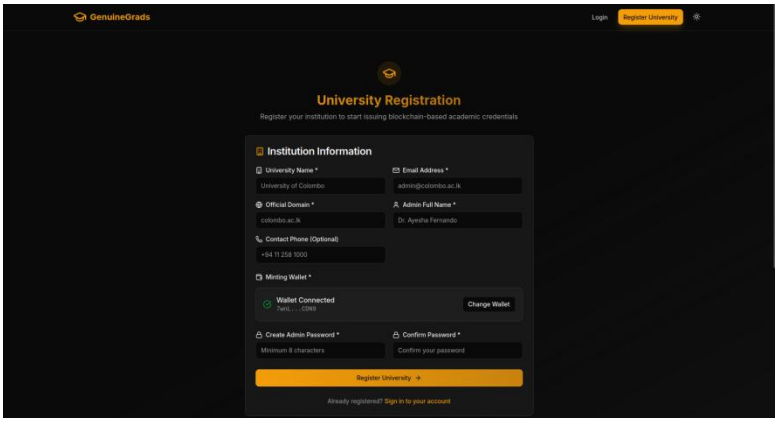

Figure 1: University registration page

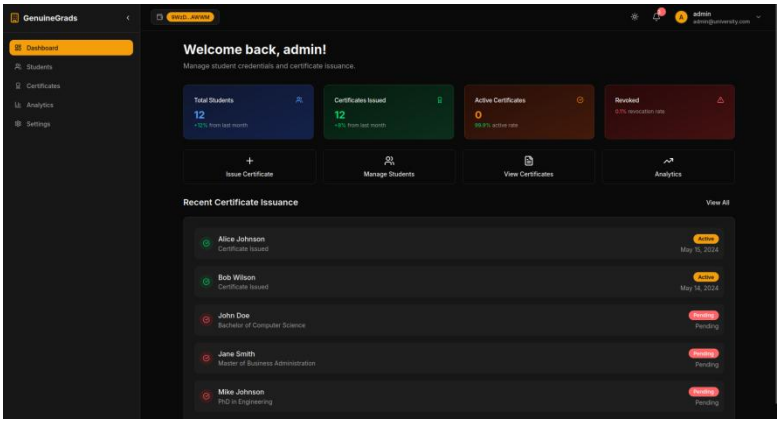

Figure 2: Univeristy dashboard

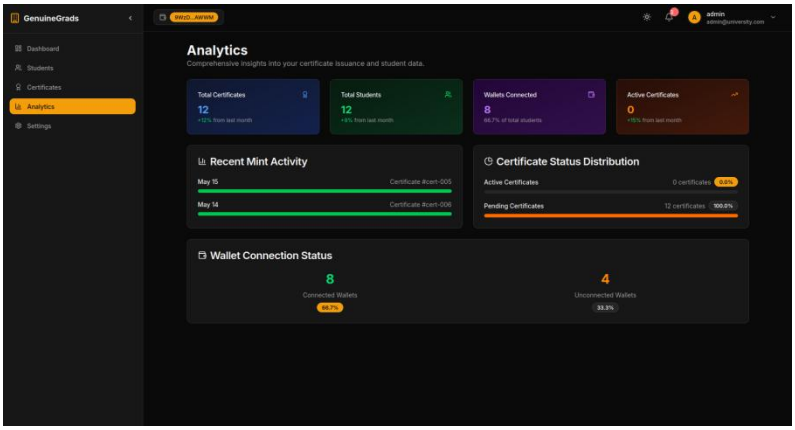

Figure 3: Analytics UI for Universities

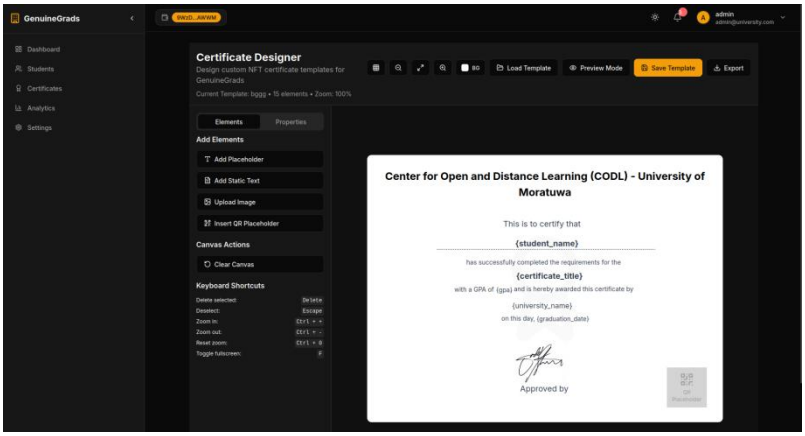

Figure 4: Certificate Designer

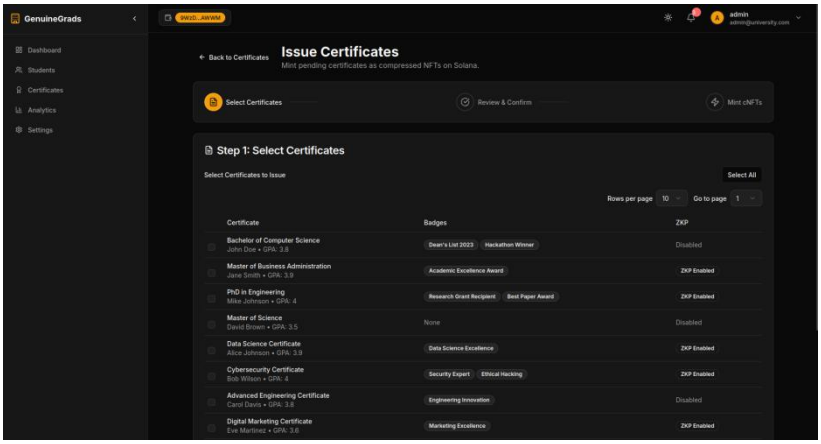

Figure 5: Issue Certificates UI

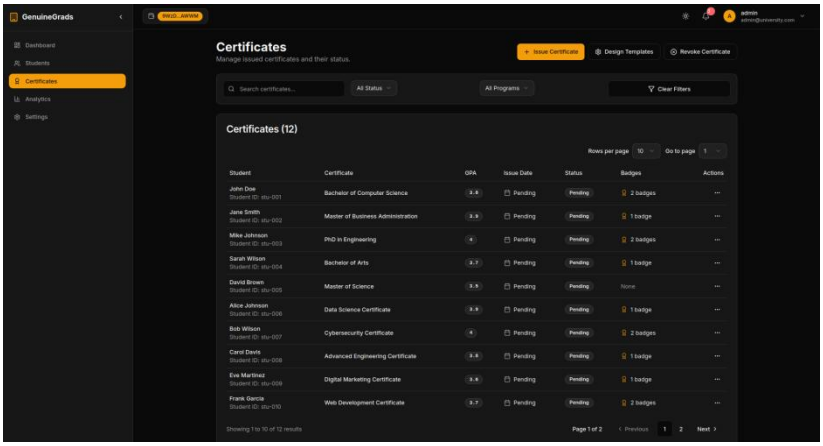

Figure 6: View Issued Certificates

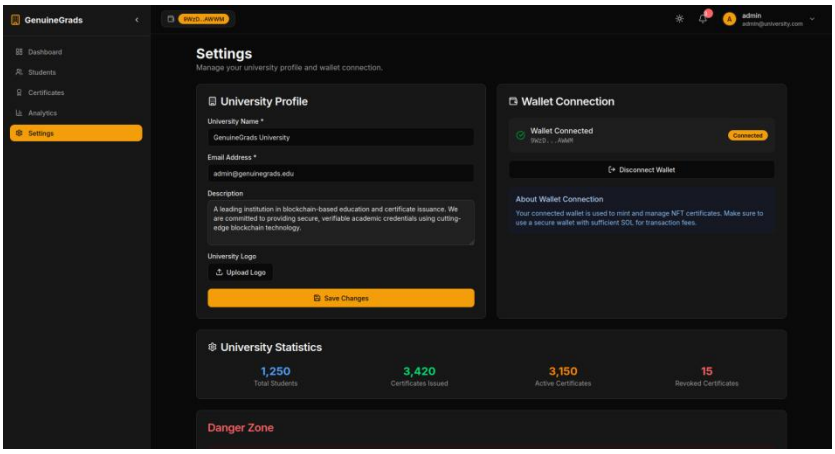

Figure 7: University settings page

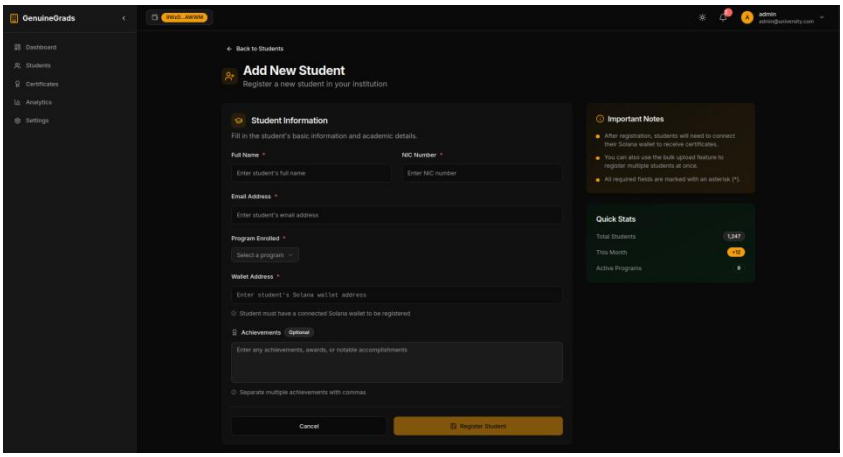

Figure 8: Single student registration

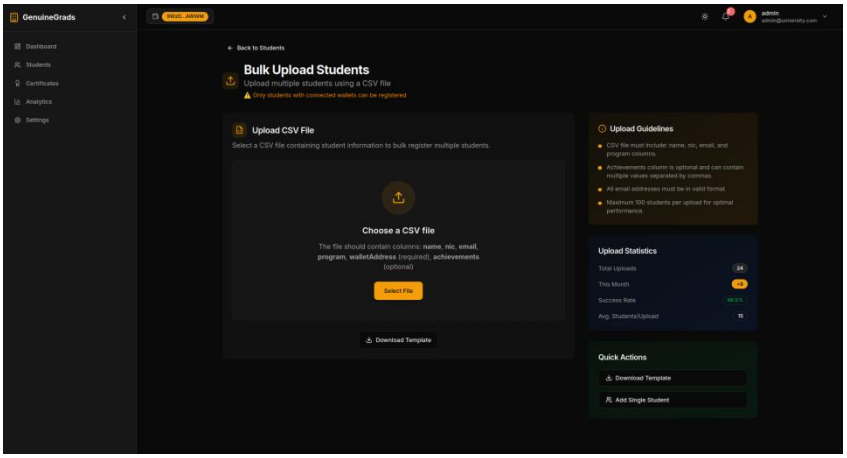

Figure 9: Bulk student registration

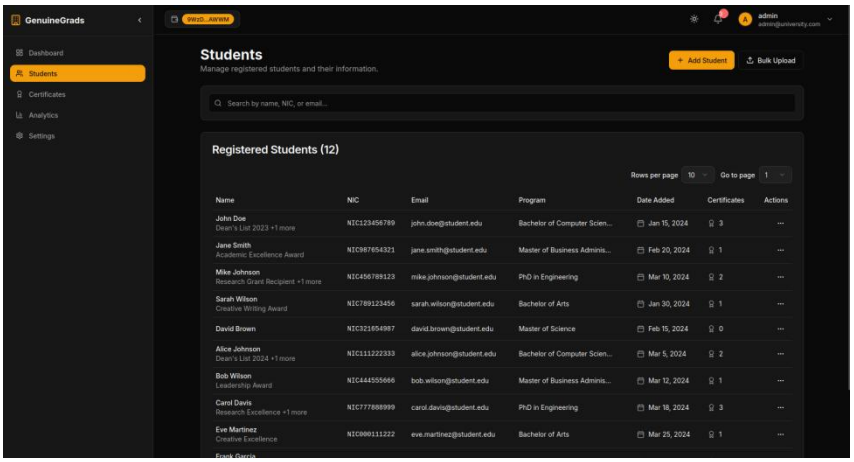

Figure 10: View registered students

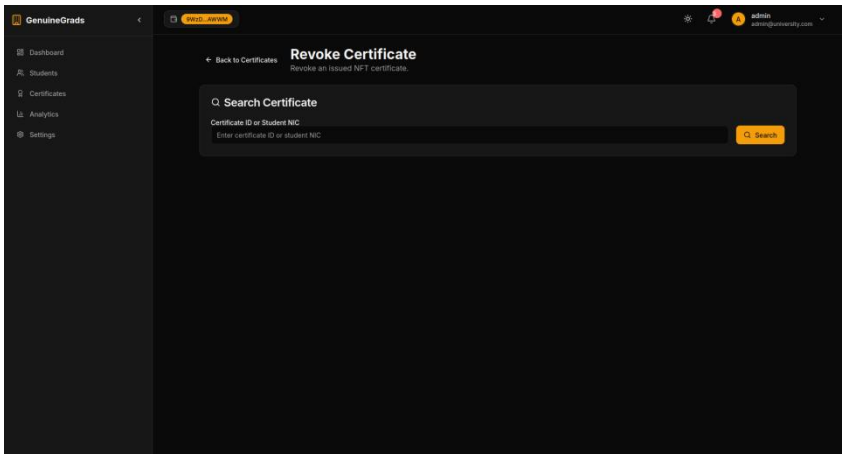

Figure 11: Certificate revocation page

- Student Dashboard: View and share issued NFT certificates, Manage achivements and ZKP generations, monitor verification logs.

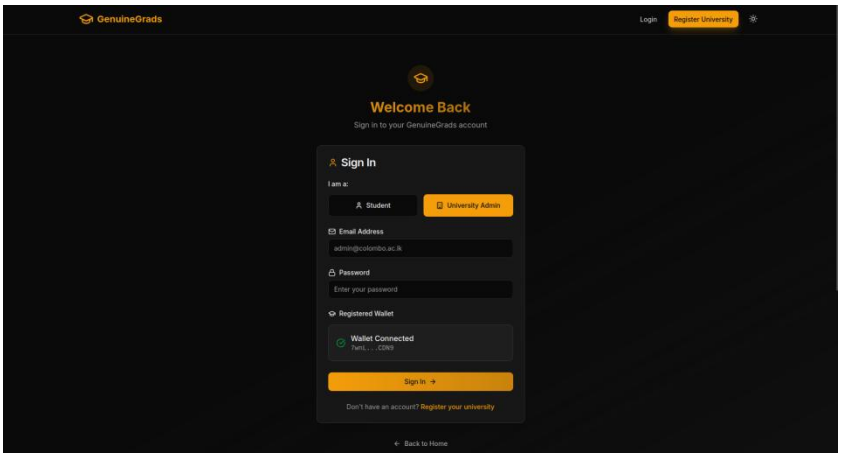

Figure 12: Login UI

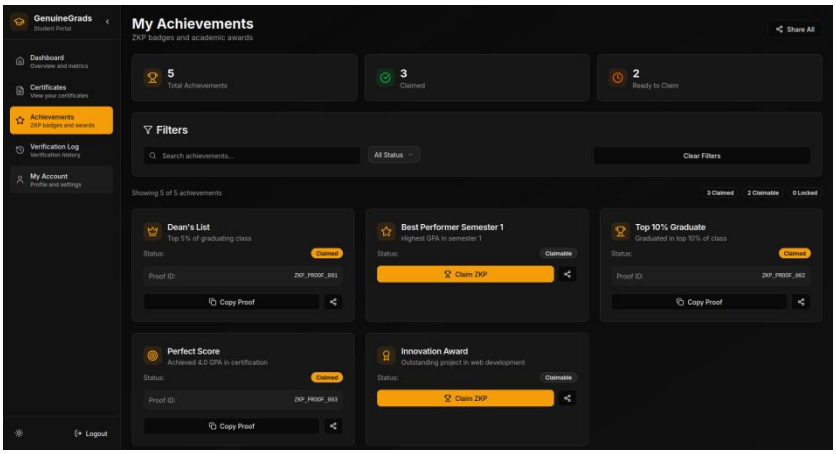

Figure 13: Achivement management UI

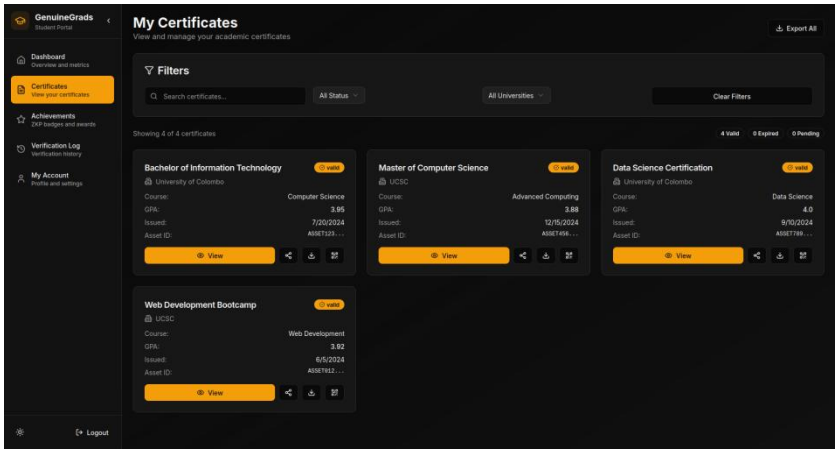

Figure 14: Manage issued certificates

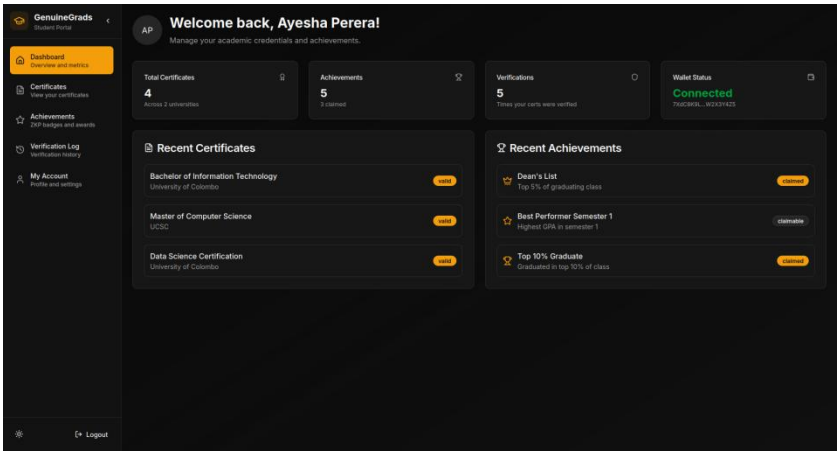

Figure 15: Student dashboard

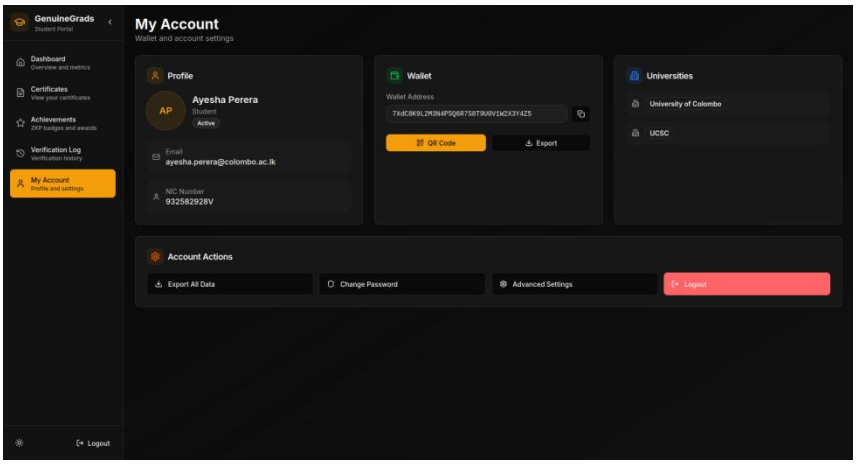

Figure 16: Student profile management

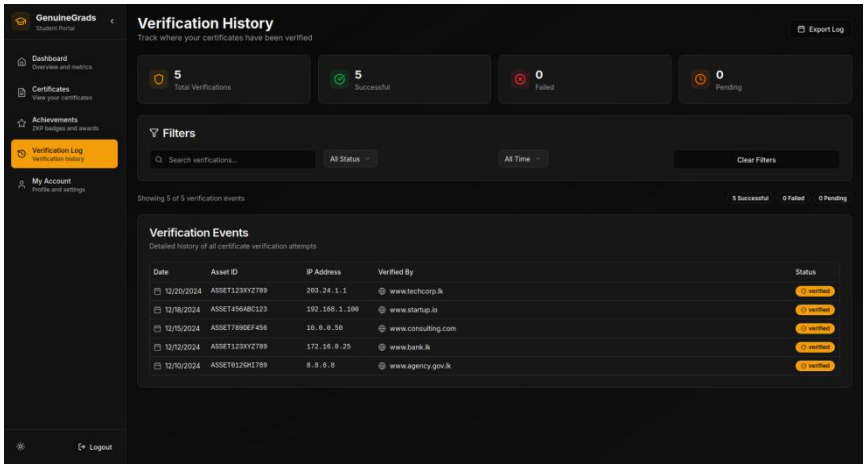

Figure 17: Verification history

- Employer/Verifier Interface: Verify certificates by scanning QR code or entering asset ID; check university registry and permit validity, Claim verification.

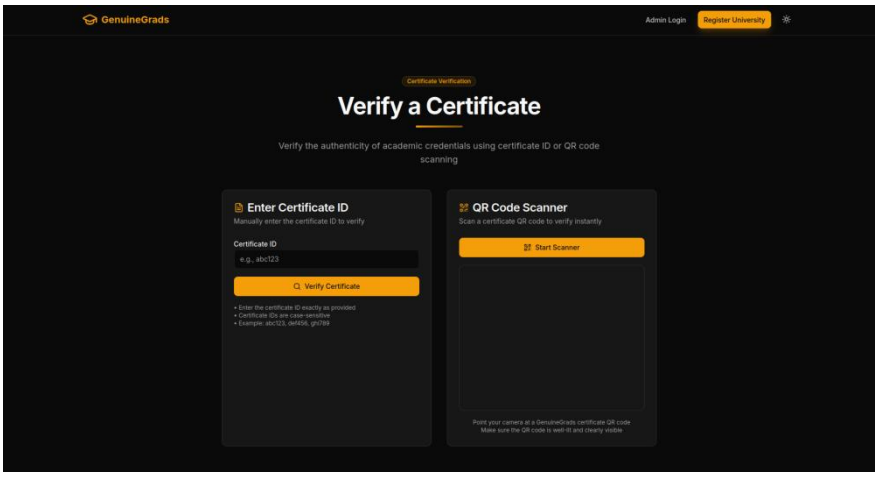

Figure 18: Verify certificates page

- Backend (Node.js + GraphQL)

  - Authentication & Session Management: Role-based access for university admins, students, and employers.

- Batch Certificate Preparation: CSV ingestion and validation; metadata assembly for multiple students.
- Metadata Storage: Upload and pin metadata JSONs to IPFS/Arweave.
- Blockchain Integration:
  - Permits: Calls Anchor program to create certificate permits (single or batched).
  - Bubblegum Minting: Uses Metaplex Bubblegum JS SDK for cNFT issuance, embedding permit references.
  - Verification: Queries Helius DAS API (getAsset, getAssetProof) and cross-checks on-chain permits + university registry.
- Webhook Receiver (Helius): Monitors certificate mints, updates indexes, triggers email/SMS notifications.

- On-chain Programs(Solana):
  - University Registry Module: Stores active universities authorized to issue certificates.
  - Certificate Permit Module: Creates lightweight permits (per student + metadata hash) before Bubblegum minting; supports revocation.
  - Compressed NFT Issuance/Revocation (Bubblegum v2 via Frontend/Backend): Certificates are minted as cNFTs; Merkle tree proofs are validated off-chain via DAS Api.

3.1.2. Database component

The GenuineGrads system follows a hybrid database model to balance privacy, scalability, and interoperability. Each participating university maintains a Private University Database that stores sensitive student and course information, while a Shared Centralized Database provides a global coordination layer for certificate issuance, revocation, and verification. This architecture enforces strict data segregation while allowing cross-institution interoperability.

3.1.2.1. Private university database

Each university instance maintains its own private schema. This ensures compliance with data protection regulations and allows institutions to maintain sovereignty over student records.

Key entities:

- Student – stores unique student identifiers, wallet addresses, hashed NIC values, and email for communication.
- Course – maintains course metadata per university.
- Enrollment – acts as a junction between students and courses, tracking GPA and batch year.
- Achievement – records additional recognitions such as semester awards or dean's list, linked to zero-knowledge proof (ZKP) commitments.
- Certificate – represents NFT-backed certificates issued via Solana. Stores blockchain metadata including mint address and IPFS/Arweave links.
- ZKPProofRequest – handles student-initiated proof requests for verifiable claims (e.g., GPA threshold, achievement badge) without exposing raw data.

3.1.2.2. Shared centralized database

The centralized layer provides a coordination hub across all universities, ensuring transparency and a unified verification process. It does not store detailed student academic data but only global identifiers and audit logs.

Key entities:

- University – registry of all participating institutions, including wallet addresses for on-chain authorization.
- Admin – university administrators with credentials to manage certificate issuance.
- GlobalStudentIndex – a de-duplicated index of students across universities, mapping hashed NICs to blockchain wallets.
- MintActivityLog – immutable log of certificate minting events, including status (pending, success, failure) for auditability.
- RevokedCertIndex – records certificates that have been revoked, ensuring verifiers can detect invalid credentials.

3.1.2.3. Advantages of the hybrid model

The hybrid database architecture of GenuineGrads leverages the strengths of both decentralized and centralized storage. By keeping sensitive academic records in the Private University Database and maintaining audit and verification data in the Shared Centralized Database, the system achieves an optimal balance between data privacy, scalability, interoperability, and trust. This design not only ensures that universities retain control over their students' personal information but also provides a unified global framework for certificate validation and fraud prevention.

Aspect	Private University Database	Shared Centralized Database
Data Privacy	PII stored locally under university control	Only anonymized/global references stored
Scalability	Scales horizontally per university	Handles global verification load

Auditability	Local academic records	Immutable minting & revocation logs
Interoperability	Supports ZKP & cNFT issuance per student	Provides a unified global verification interface
Fraud Prevention	Commitment hashes & local proof requests	Global revocation + NIC de-duplication

Table 6: Advantages of hybrid model

3.1.2.4. On-chain storage (Solana blockchain)

Purpose: Provides an immutable and publicly verifiable record of issued academic credentials.

Data Stored:

- Mint Address & Asset ID of each certificate NFT (cNFT).
- Merkle Tree Commitments (via Metaplex Bubblegum v2) for compression and proof generation.
- Hash Pointers linking to certificate metadata on IPFS/Arweave.
- Revocation Records (burn or status update transactions).

3.1.2.5. Off-chain metadata storage (IPFS/Arweave)

Purpose: Stores detailed certificate information that would be too costly or inefficient to keep on-chain.
Data Stored:

- Student certificate details (degree name, graduation date, faculty).
- Achievement badges (e.g., Dean's List, Semester Top Performer).
- Visual assets (certificate design templates, university logos).
- JSON metadata files following NFT metadata standards.

- Certificate template JSON.

3.1.3. Hardware component

None required beyond standard cloud instances/containers for API, DB, and queue. No specialized on-prem hardware.

3.2. Proposed methodology

This project will utilize agile development methodology to ensure iterative, flexible, and user-centric approach to build the application, This approach is well-suited to blockchain applications, where requirements may evolve during testing, and integration with emerging standards such as compressed NFTs (cNFTs) and Zero-Knowledge Proofs (ZKPs) requires iterative experimentation.

The methodology is structured into the following key aspects:

3.2.1. Iterative development and sprints

The development process will be divided into 2–3 week sprints, with each sprint delivering a functional increment of the system. At the end of every sprint:

- Demonstrations will be conducted with supervisor.
- Feedback will be incorporated into the subsequent sprint.
- A minimum viable prototype (MVP) will be prioritized, featuring university registration, certificate issuance as cNFTs, revocation, and verification.

3.2.2. Requirements refinement

User stories will guide the requirements, mapped to the three main actors:

- University Admins (certificate issuers),
- Students (recipients and sharers),
- Employers/Verifiers (verifying authenticity).

Each functional requirement from the SRS (e.g., “bulk upload of certificates”, “NFT verification via QR code”, “achievement badge issuance”) will be decomposed into sprint backlog items.

3.2.3. System architecture alignment

The methodology integrates with the pre-defined three-tier system architecture:

- Frontend (Next.js/React + ShadCN UI): Iteratively built with user feedback, ensuring responsive dashboards for universities, students, and verifiers.

- Backend (Node.js + GraphQL with Prisma + PostgreSQL): Developed in modular micro-services (authentication, metadata assembly, ZKP service, audit logging). Each module will be tested independently before full integration.

- On-Chain (Solana + Anchor + Metaplex Bubblegum v2): Anchor smart contracts will be developed incrementally, starting with university registration PDA, then certificate minting via Bubblegum CPI, and later advanced ZKP integration.

3.3. Technologies adapted

- Blockchain: Solana blockchain.
- Metaplex Bubblegum v2: cNFT lifecycle (mintV2, verify collection, burnV2).
- Helius DAS + Webhooks: getAsset, getAssetProof, event streaming.
- Next.js (React) + Tailwind + shadcn/ui.
- Node.js + GraphQL + Postgres (RLS).
- IPFS/Arweave for metadata & assets.
- ZK: Circom and Halo2 for predicate proofs.
- Rust/Anchor CPI (post-MVP) to hard-enforce mint policy.

A comparative analysis was carried out to evaluate the selected technologies against their closest alternatives (e.g., Solana vs. Ethereum, Bubblegum vs. standard NFTs, GraphQL vs. REST). The comparison considered user satisfaction, requirement satisfaction, scalability, cost efficiency, and ease of integration.

The full Comparison of Technologies Table is provided in Appendix A (Table A.1) for reference.

3.4. Test and deployment plan

3.4.1. Testing strategy

The GenuineGrads platform will adopt a multi-layered testing strategy to ensure correctness, reliability, and security of all components deployed on Solana Devnet.

1. Unit testing

    - Scope:
        - Smart contract instructions (register_university, Merkle tree setup).
        - Backend GraphQL resolvers and service functions.
        - Frontend React components (certificate dashboard, verification UI).
    - Tools:
        - Anchor Test Framework for program logic.
        - Mocha/ Chai for backend APIs.

2. Integration testing

    - Scope: Backend → Solana Devnet → Bubblegum v2 SDK → Helius RPC.
    - Approach (Manual):

- Test minting, verification, and burning of cNFTs.
- Validate Arweave/IPFS metadata retrieval.
- Simulate webhook events from Helius and confirm audit logging.

3. System testing (End-to-End)

- Scope: Complete workflow from university registration → student onboarding → certificate issuance → employer verification.
- Test Environment: Solana Devnet only (with Helius RPC).
- Expected Outcome: All functional requirements in SRS are satisfied.

4. Security & performance testing

- Validate role-based access (only registered universities can mint).
- Test Circom/Halo2 ZKP proof generation and verification flow.
- Perform load testing for bulk issuance (e.g., 5,000 certificates).

3.4.2. Deployment plan

1. Development Environment

- Local Solana Test Validator, Postgres in Docker, backend (Node.js/GraphQL), frontend (Next.js).
- Used for debugging and initial verification.

2. Staging/Devnet Environment

    - Hosted on DigitalOcean Droplets.
    - Blockchain interactions through Helius Devnet RPC + Webhooks.
    - Metadata pinned to Arweave testnet/IPFS sandbox.

3. Deployment Steps

    - Step 1: Apply Prisma migrations to Postgres database.
    - Step 2: Deploy Anchor program (registry + tree setup) to Solana Devnet.
    - Step 3: Deploy backend (GraphQL API) and frontend (Next.js) containers to DigitalOcean.
    - Step 4: Validate cNFT issuance and revocation via Bubblegum SDK + DAS queries.
    - Step 5: Monitor activity through Helius Webhooks and backend audit logs.

4. Rollback Plan

    - Frontend/Backend: Roll back to last stable Docker image.
    - Database: Restore from snapshots or backups.
    - Anchor Program: Re-deploy stable Devnet build with consistent program ID.

References

[1] ‘Blockcerts : The Open Standard for Blockchain Credentials’. Accessed: Aug. 26, 2025. [Online]. Available: https://www.blockcerts.org/

[2] ‘Burning Compressed NFTs | Bubblegum V2’. Accessed: Aug. 26, 2025. [Online]. Available: https://developers.metaplex.com/bubblegum-v2/burn-cnfts

[3] ‘Compressed NFTs’. Accessed: Aug. 26, 2025. [Online]. Available: https://solana.com/developers/courses/state-compression/compressed-nfts

[4] ‘DigiLocker: An Initiative Towards Paperless Governance’, DigiLocker. Accessed: Aug. 26, 2025. [Online]. Available: https://www.digilocker.gov.in/

[5] ‘Digital Asset Standard (DAS)’, Helius Docs. Accessed: Aug. 26, 2025. [Online]. Available: https://www.helius.dev/docs/api-reference/das

[6] ‘Faking Degrees: A Thriving industry in South Asia’. Accessed: Aug. 26, 2025. [Online]. Available: https://www.dailymirror.lk/print/opinion/Faking-Degrees:-A-Thriving-industry-in-South-Asia/172-298260

[7] ‘FAQ | Bubblegum V2’. Accessed: Aug. 26, 2025. [Online]. Available: https://developers.metaplex.com/bubblegum-v2/faq

[8] ‘getAssetProof’, Helius Docs. Accessed: Aug. 26, 2025. [Online]. Available: https://www.helius.dev/docs/api-reference/das/getassetproof

[9] ‘Getting started · OpenCerts’. Accessed: Aug. 26, 2025. [Online]. Available: https://opencerts.io/

[10] ‘In a Nutshell | ZK Compression’. Accessed: Aug. 26, 2025. [Online]. Available: https://www.zkcompression.com/learn/in-a-nutshell

[11] ‘Make information easy to verify and almost impossible to fake - EBSI -’. Accessed: Aug. 26, 2025. [Online]. Available: https://ec.europa.eu/digital-building-blocks/sites/spaces/EBSI/pages/710119737/Make+information+easy+to+verify+and+almost+impossible+to+fake

[12] N. S. Asia, ‘Malaysia’s Education Ministry uses NEM to set up a University Degree Verification System’, NEM Official. Accessed: Aug. 26, 2025. [Online]. Available: https://medium.com/@NEM_SEA/malaysias-education-ministry-uses-nem-to-set-up-a-university-degree-verification-system-db5447ded9c9

[13] ‘Minting Compressed NFTs | Bubblegum V2’. Accessed: Aug. 26, 2025. [Online]. Available: https://developers.metaplex.com/bubblegum-v2/mint-cnfts

[14] ‘National Academic Depository’. Accessed: Aug. 26, 2025. [Online]. Available: https://nad.gov.in/

[15] ‘OpenAttestation | OpenAttestation’. Accessed: Aug. 26, 2025. [Online]. Available: https://openattestation.com/

[16] ‘Overview | Bubblegum V2’. Accessed: Aug. 26, 2025. [Online]. Available: https://developers.metaplex.com/bubblegum-v2

[17] ‘Overview | Light Protocol’. Accessed: Aug. 26, 2025. [Online]. Available: https://docs.lightprotocol.com

[18] E. Barker, ‘Recommendation for key management: part 1 - general’, National Institute of Standards and Technology, Gaithersburg, MD, NIST SP 800-57pt1r5, May 2020. doi: 10.6028/NIST.SP.800-57pt1r5.

[19] ‘Solana Webhooks: Real-Time Blockchain Event Notifications’, Helius Docs. Accessed: Aug. 26, 2025. [Online]. Available: https://www.helius.dev/docs/webhooks

[20] A. Yakovenko, ‘Solana: A new architecture for a high performance blockchain’.

[21] ‘State compression’, Solana | News. Accessed: Aug. 26, 2025. [Online]. Available: https://solana.com/fi/news/tag/state-compression

[22] ‘The phenomenon and evolution of diploma mills’. Accessed: Aug. 26, 2025. [Online]. Available: https://etico.iiep.unesco.org/sites/default/files/2018-05/chapter1.pdf

[23] ‘The Verifiable Credentials 2.0 family of specifications is now a W3C Recommendation’, W3C. Accessed: Aug. 26, 2025. [Online]. Available: https://www.w3.org/news/2025/the-verifiable-credentials-2-0-family-of-specifications-is-now-a-w3c-recommendation/

[24] ‘Tower BFT | Agave’. Accessed: Aug. 26, 2025. [Online]. Available: https://docs.anza.xyz/implemented-proposals/tower-bft

Appendix A: Timeline – Gantt chart

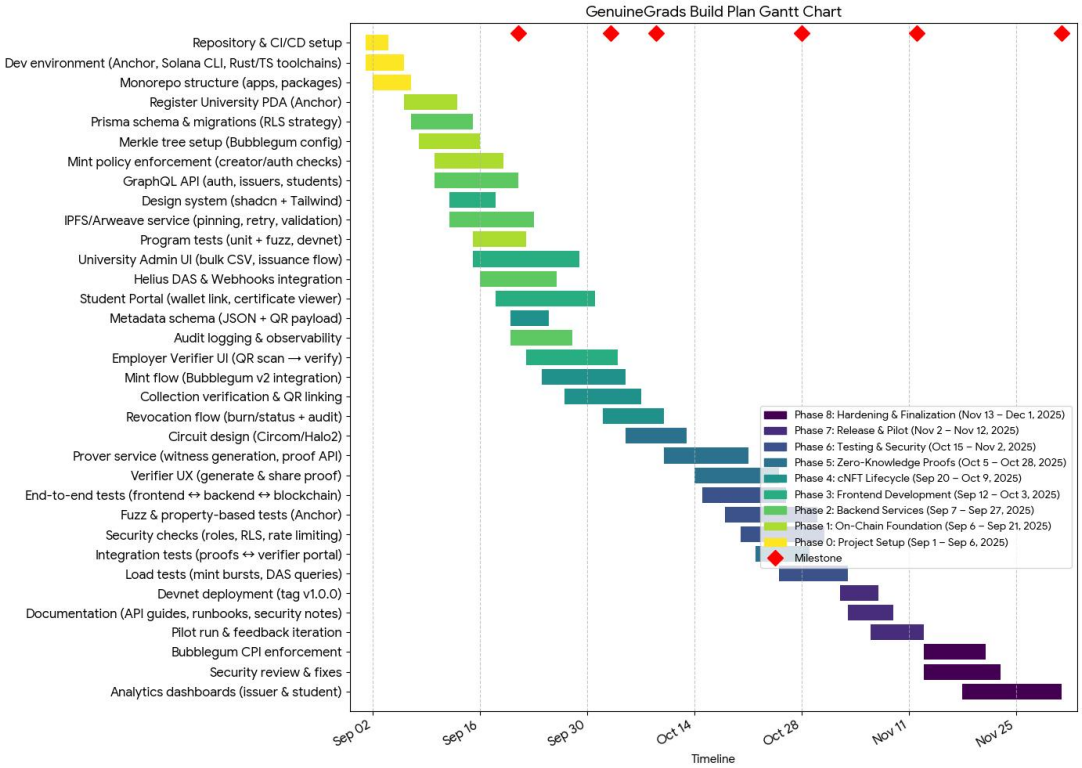

Figure 19: Gantt chart

Appendix B: SRS

"View the complete Software Requirements Specification (SRS) document here.

Appendix C: Use Case Diagrams

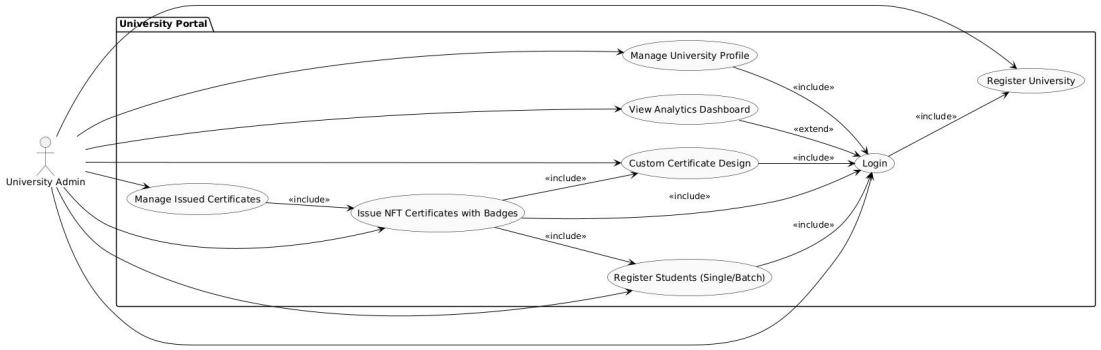

Figure 20: University use case diagram

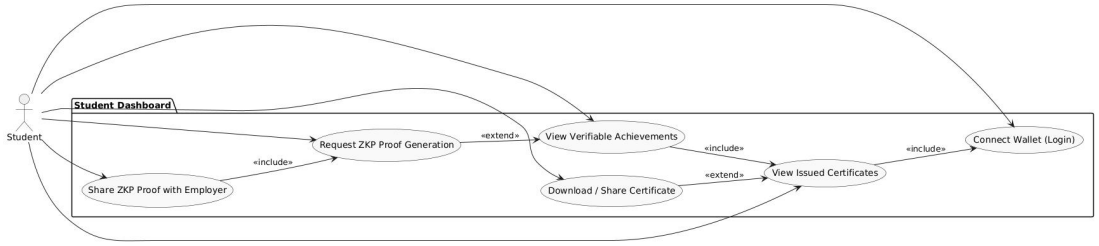

Figure 21: Student use case diagram

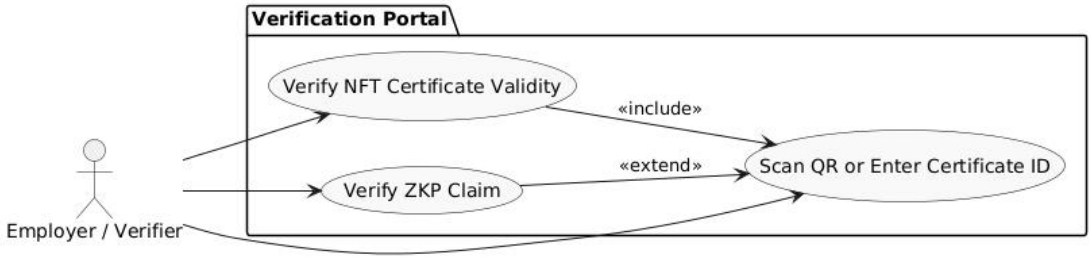

Figure 22: Employer/Verifier use case diagram

Appendix D: Activity Diagrams

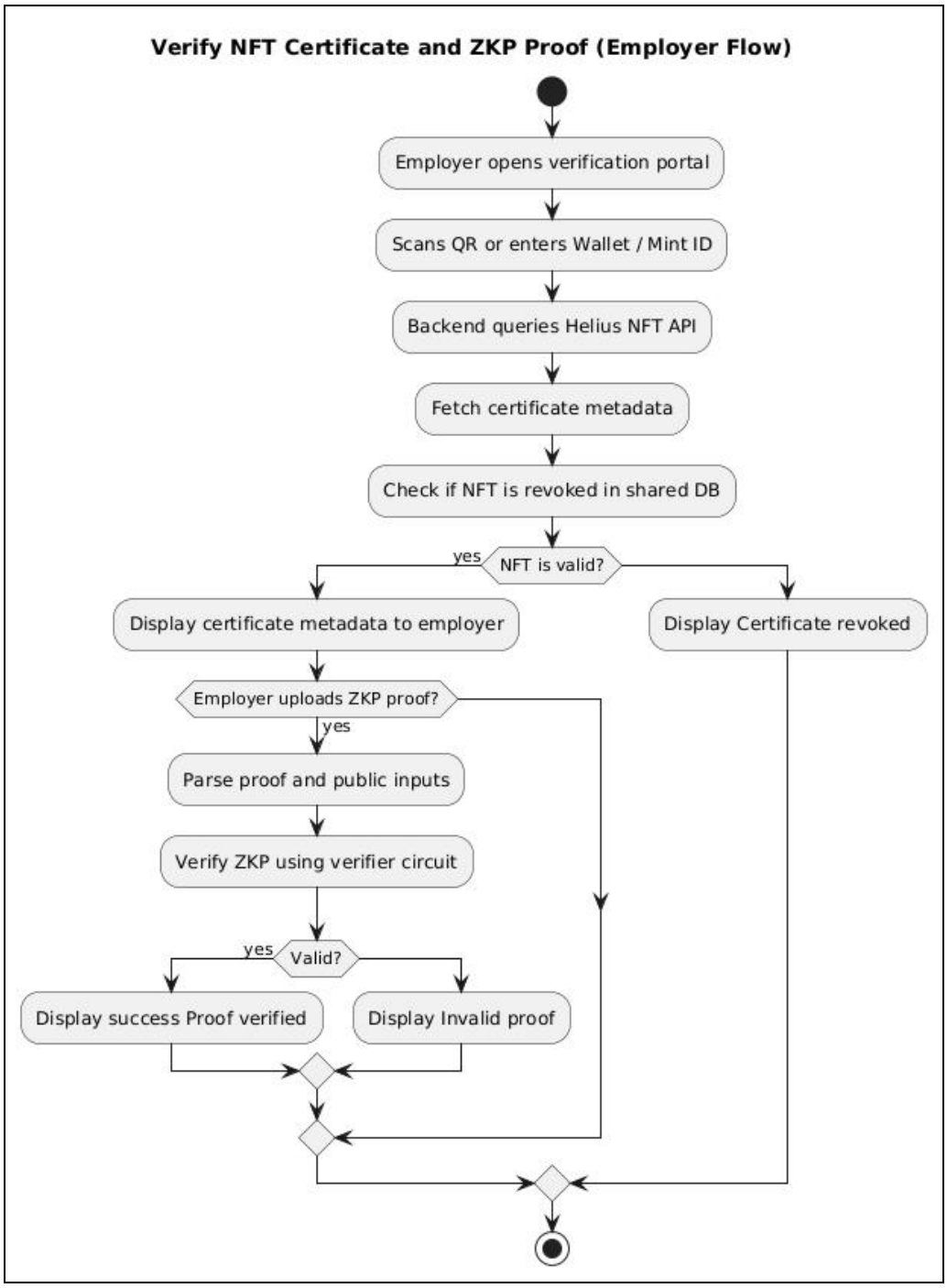

Figure 23: Verify certificate and zkp activity diagram

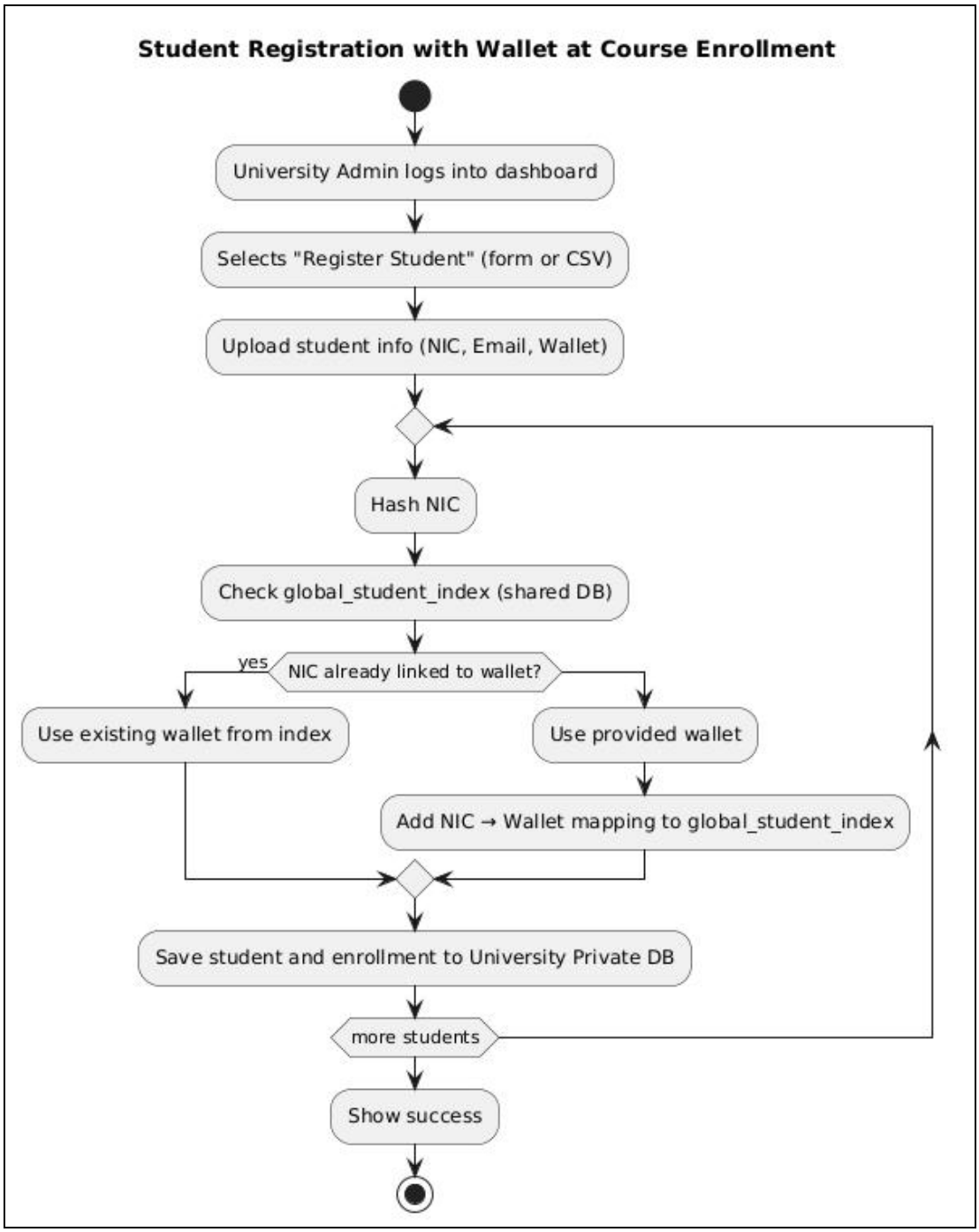

Figure 24: Student registration with wallet activity diagram

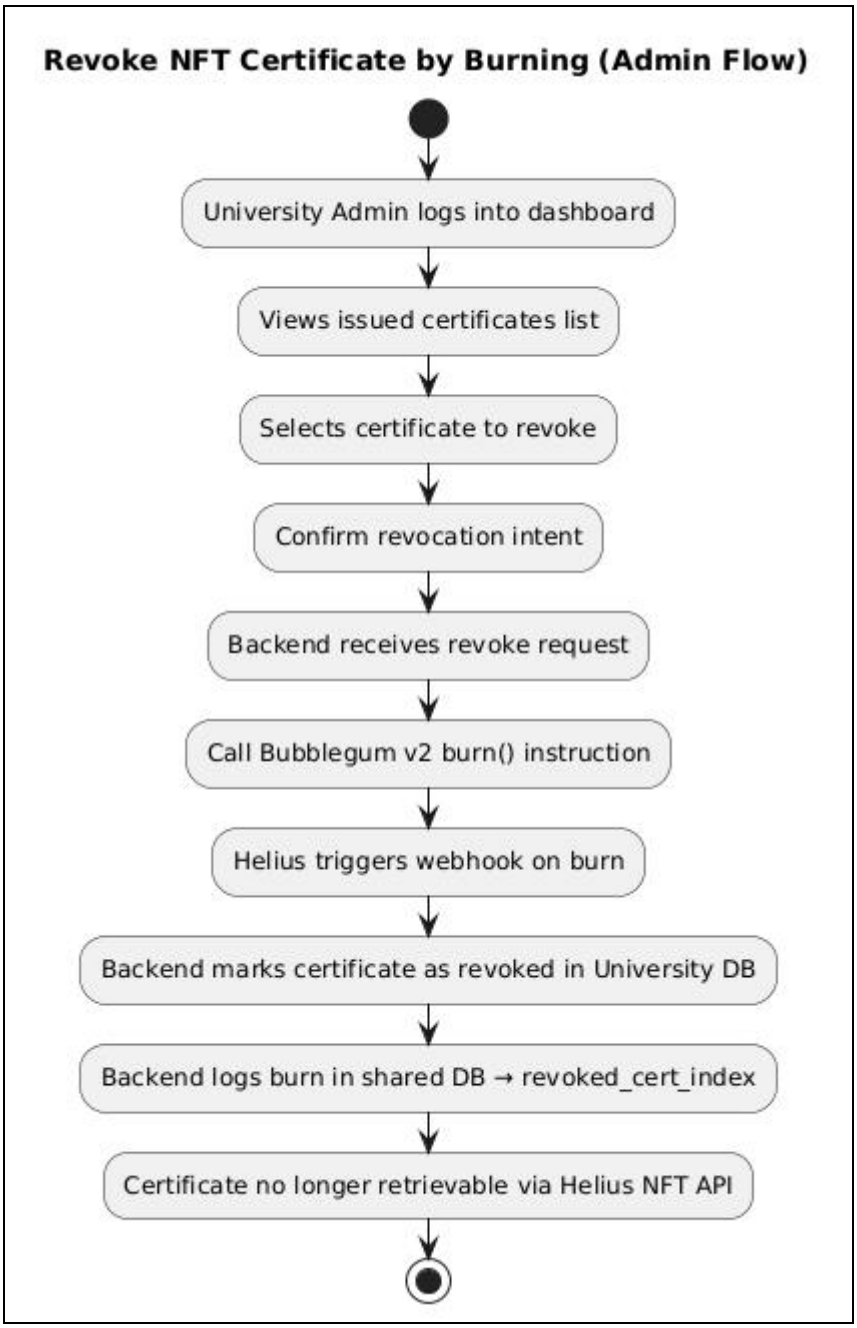

Figure 25: Revoke NFT activity diagram

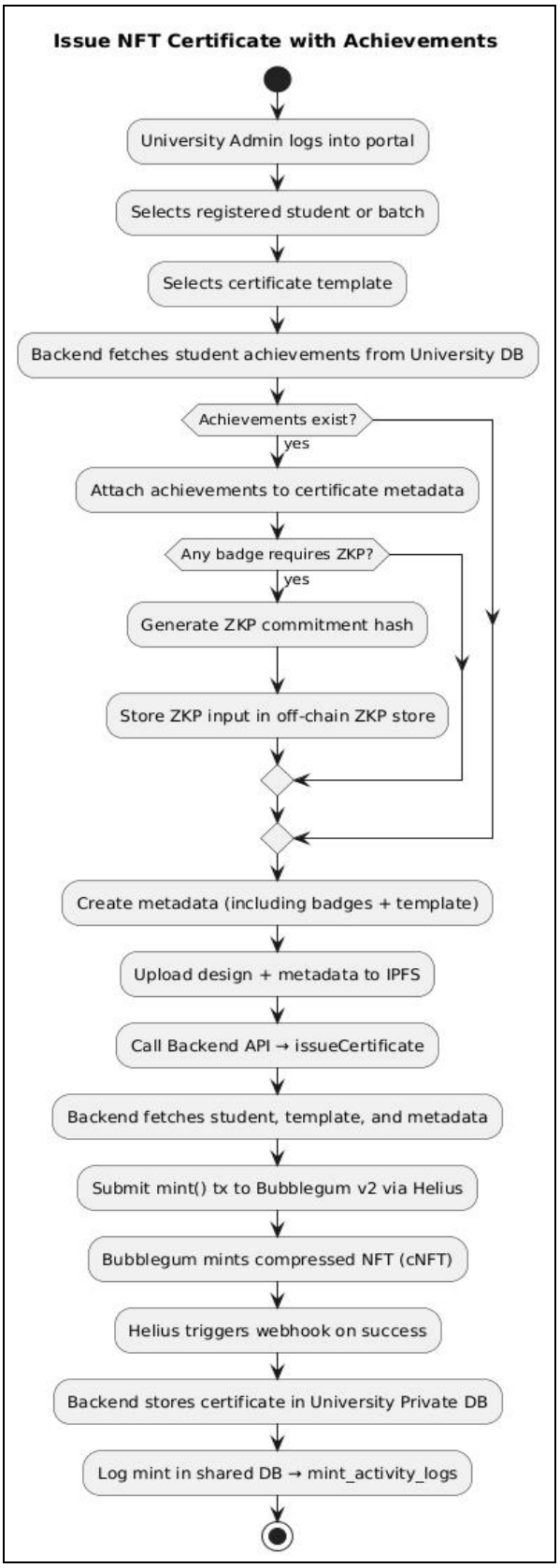

Figure 26: Issue certificate activity diagram

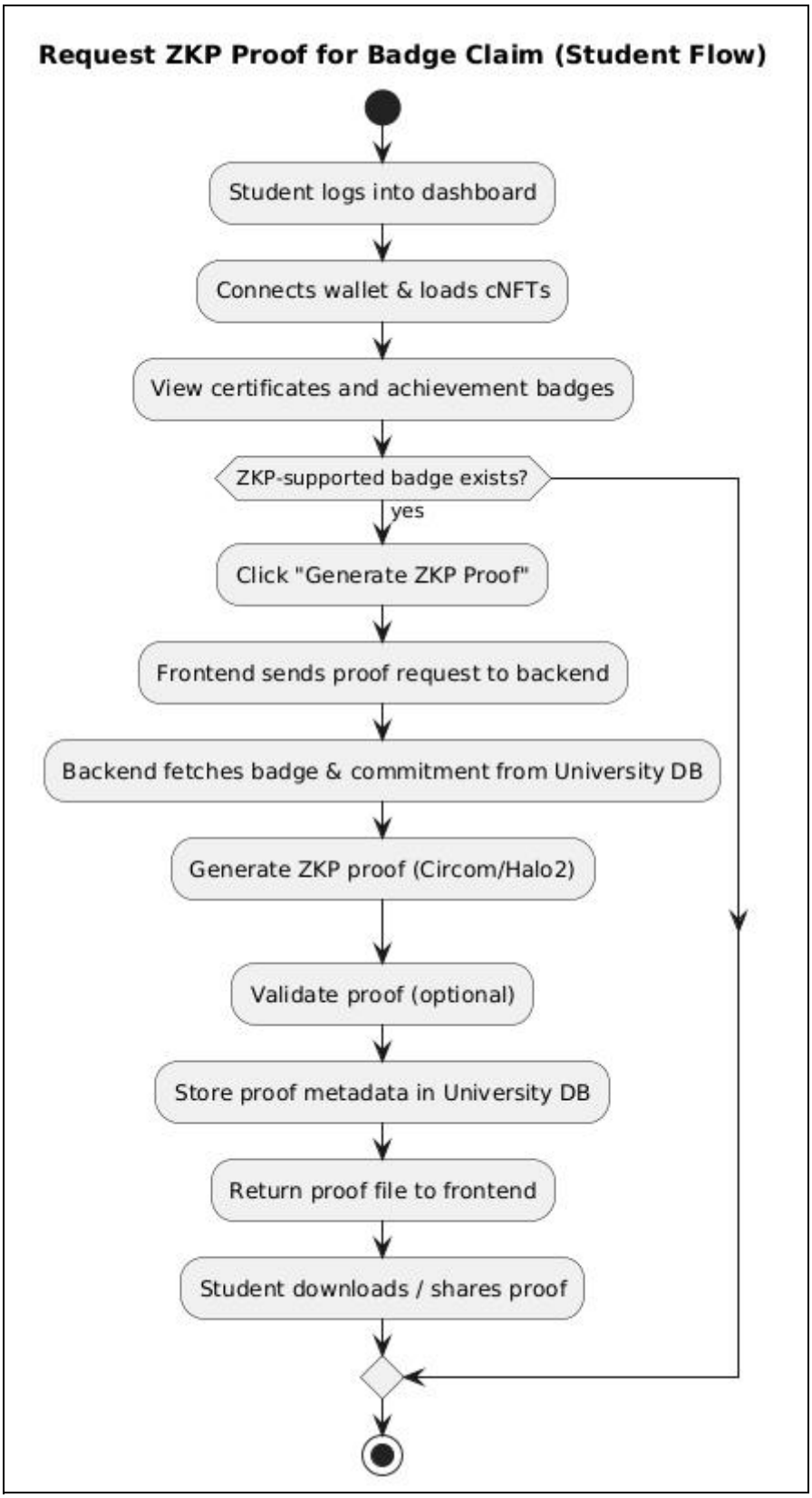

Figure 27: Request zk proof for badge claim

Appendix E: ERD

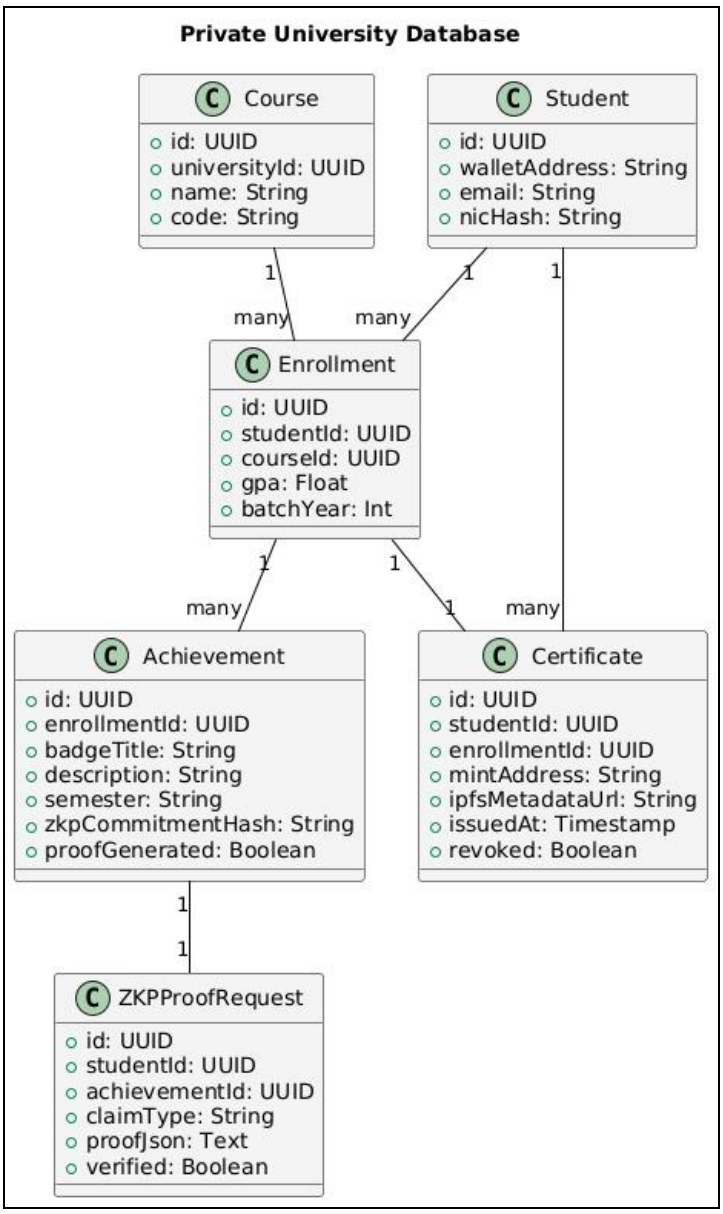

Figure 28: Private database ERD

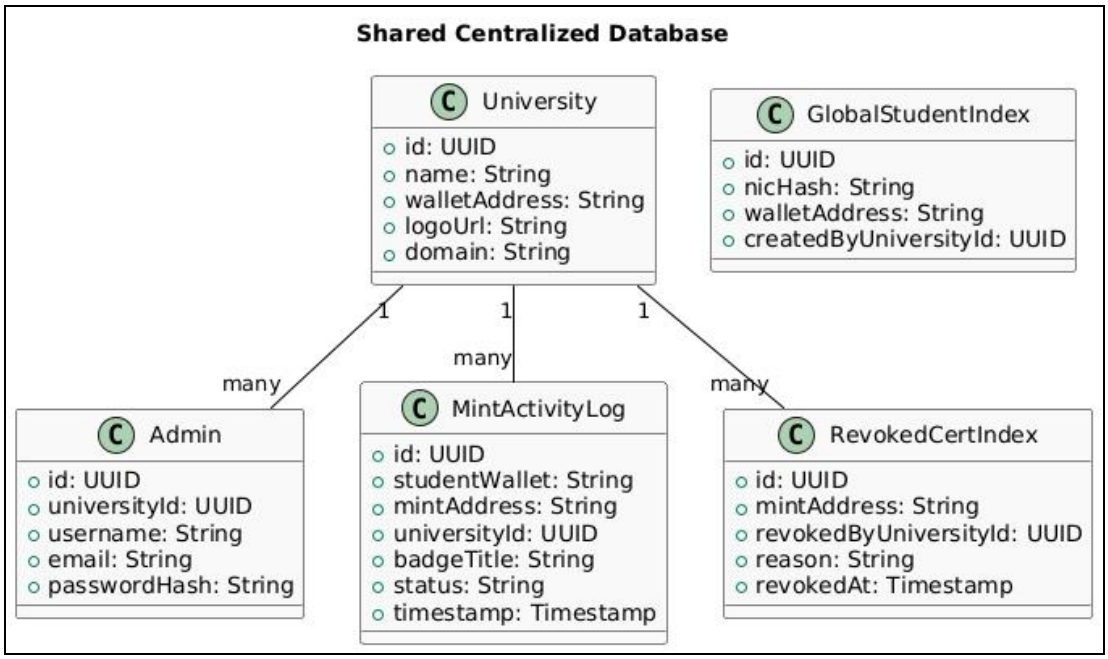

Figure 29: Shared database ERD

Appendix F: Feauture by feature comparison

Capability	Blockcerts	OpenAttestation/OpenCerts	EBSI Diplomas	e-Scroll	DigiLocker
Trust/Identity Model	Issuer keys/DIDs; optional registries	DNS/DID registry with governance	EU trust lists; eIDAS alignment	University keys; MOHE/university registry	Centralized gov onboarding & KYC
Anchoring	Hash on BTC/ETH	Hash/Merkle root on Ethereum	VC status lists; trust registries (not per-credential on-chain)	NEM/Catapult tx	Centralized repository
Storage of Payload	Off-chain; wallet/URI	Off-chain object store	Holder wallet	Off-chain portals	Central repo
Revocation Signal	Revocation list/status (varies)	Contract + document store revoke	VC StatusList endpoints	Portal status + on-chain refs	Repository status APIs
Verification UX	QR + web verifier	QR + production verifiers	Wallet presentation	QR web portal	web via DigiLocker
Selective Disclosure	Not native(can layer VC SD)	Not native(can layer VC SD)	Native SD (BBS+, SD-JWT	None	None
Predicate Proofs	No	No	No	No	No

(ZK)					
Scale/Cost	Good (hash only)	Good (Merkle batching; EVM fees)	Good (web-scale, low chain use)	Good (national size)	Excellent (web scale)

Table 7: Feature by feature comparison

Appendix G: Comparison of Technologies

Layer	Adopted Technology	Alternatives Considered	Advantages (Chosen)	Disadvantages (Chosen)
Blockchain	Solana (PoH + Tower BFT)	Ethereum, Polygon	High throughput, low fees	Less developer maturity vs. Ethereum
NFT Framework	Bubblegum v2 (cNFTs)	Standard NFTs (pNFTs)	Cost-efficient, scalable	Verification tools still evolving
Middleware	Helius DAS + Webhooks	QuickNode, Alchemy	Real-time indexing, rich APIs	Dependency on third-party infra
Backend	Node.js + GraphQL + Postgres RLS	REST + MongoDB	Flexible queries, RLS security	Learning curve for GraphQL
Storage	IPFS + Arweave	Filecoin, Centralized DB	Decentralized + permanent storage	Arweave has upfront costs
ZKPs	Circom + Halo2	Groth16, Plonk	Advanced recursive proofs	Higher proving costs
Frontend	Next.js + Tailwind + shadcn/ui	Vue, Angular	SSR, rich ecosystem	Slightly heavier build size

Table 8: Comparison of Technologies