
Software Requirements Specification

for

GenuineGrads: Blockchain- Based Certificate Verification Systems with Zero-Knowledge Proofs and NFTs

Version 1.0

Prepared by M.H.M.Shahadh - E2240185

CODL, University of Moratuwa

26th July 2025

EXECUTIVE SUMMARY

This document outlines the System Requirements Specification for GenuineGrads, a proposed blockchain-based system designed to mitigate academic certificate fraud and enhance verification processes within higher education. The current landscape is plagued by significant and costly issues related to credential misrepresentation and inefficient manual verification, as evidenced by numerous incidents in Sri Lanka and the broader South Asian region [1], [2], [3]. GenuineGrads directly addresses these challenges by offering a secure, efficient, and privacy-preserving platform for the issuance and verification of academic credentials.

GenuineGrads leverages the Solana blockchain [7] for its high throughput and low transaction costs, enabling the issuance of academic certificates as tamper-proof Non-Fungible Tokens (NFTs). The system utilizes the Metaplex Bubblegum v2 standard for on-chain minting and revocation functionalities, while off-chain metadata is securely stored on the InterPlanetary File System (IPFS). The architectural stack comprises a React.js frontend for user interaction, a Node.js/GraphQL backend for data management and API services, and Rust/Anchor smart contracts for on-chain logic and operations.

The platform is structured around three distinct user-facing portals: a University Portal for institutional users to manage templates, student enrollment, and NFT issuance; a Student Dashboard empowering graduates to view their credentials and generate Zero-Knowledge Proofs (ZKPs) of their achievements; and an Employer Verification Portal for authenticating NFTs and validating ZKP claims. A key innovation is the integration of ZKPs via Circom or Halo2 [8], which allows stakeholders, such as employers, to verify specific attributes (e.g., GPA ranges) without exposing sensitive underlying data. By automating verification and preventing credential tampering [3], [4], GenuineGrads is anticipated to significantly expedite credential checks, reduce fraud, and restore trust in academic qualifications, aligning seamlessly with Sri Lanka's national digital transformation strategies [5], [6].

TABLE OF CONTENTS

EXECUTIVE SUMMARY	ii
TABLE OF CONTENTS	iii
LIST OF FIGURES	vi
LIST OF TABLES	vii
LIST OF ABBREVIATIONS	viii
REVISION HISTORY	ix
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	2
1.4 Project Scope	3
1.5 References	4
2. Overall Description	5
2.1 Product Perspective	5
2.2 Product Features	6
2.3 User Classes and Characteristics	9
2.4 Operating Environment	9
2.5 Design and Implementation Constraints	11
2.6 User Documentation	12
2.7 Assumptions and Dependencies	13
3. System Features	15
3.1 University Registration & Authentication	15
3.1.1 Description and Priority	15
3.1.2 Stimulus/Response Sequences	15
3.1.3 Functional Requirements	16
3.2 Student Registration & De-duplication	16
3.2.1 Description and Priority	16

3.2.2 Stimulus/Response Sequences	17
3.2.3 Functional Requirements	17
3.3 Certificate Template Design	18
3.3.1 Description and Priority	18
3.3.2 Stimulus/Response Sequences	18
3.3.3 Functional Requirements	19
3.4 Bulk & Single Certificate Issuance	19
3.4.1 Description and Priority	19
3.4.2 Stimulus/Response Sequences	19
3.4.3 Functional Requirements	20
3.5 Badge & ZKP Commitment Generation	20
3.5.1 Description and Priority	21
3.5.2 Stimulus/Response Sequences	21
3.5.3 Functional Requirements	21
3.6 Certificate Revocation	21
3.6.1 Description and Priority	22
3.6.2 Stimulus/Response Sequences	22
3.6.3 Functional Requirements	22
3.7 Student Dashboard & ZKP Proof Generation	22
3.7.1 Description and Priority	22
3.7.2 Stimulus/Response Sequences	23
3.7.3 Functional Requirements	23
3.8 Employer Verification Portal	24
3.8.1 Description and Priority	24
3.8.2 Stimulus/Response Sequences	24
3.8.3 Functional Requirements	25
4. External Interface Requirements	26
4.1 User Interfaces	26
4.2 Hardware Interfaces	34
4.3 Software Interfaces	34
4.4 Communications Interfaces	35

5. Other Nonfunctional Requirements	37
5.1 Performance Requirements	37
5.2 Safety Requirements	38
5.3 Security Requirements	38
5.4 Software Quality Attributes	39
6. Other Requirements	40
7. Future Enhancements	41
Appendix A: Glossary	43
Appendix B: Analysis Models	46
1. Use case diagrams	46
2. Activity diagrams	47
3. Class diagrams	52
4. Sequence Diagrams	54
5. State diagram	56

LIST OF FIGURES

Figure 1 . University Admin Dashboard	26
Figure 2 . Student Management	27
Figure 3 . Students Bulk Registration	27
Figure 4 . Students Individual Registration	28
Figure 5 . Certificate Designer	28
Figure 6 . Issue Certificate	29
Figure 7 . Certificate Management	29
Figure 8 . Revoke Certificates	30
Figure 9 . University Analytics Page	30
Figure 10 . University Settings Page	31
Figure 11 . Student Dashboard	31
Figure 12 . Student View Certificate Page	32
Figure 13 . Student's Achievements Page	32
Figure 14 . Student Profile Mangement	33
Figure 15 . Student Verification Logs Page	33
Figure 16 . Certificate Verification Page (Public)	34
Figure 17 . University Admin Use Case Diagram	46
Figure 18 . Student Use Case Diagram	46
Figure 19 : Employer / Verifier Use Case Diagram	46
Figure 20 . Request ZKP Proof for Badge Claim Activity Diagram	47
Figure 21 . Issue NFT Certificate with Achievements Activity Diagram	48
Figure 22 . Revoke NFT Certificate by Burning Activity Diagram	49
Figure 23 . Verify NFT Certificate and ZKP Proof Activity Diagram	50
Figure 24 . Student Registration with Wallet Activity Diagram	51
Figure 25 . Private University Database Class Diagram	52
Figure 26 . Shared Centralized Database Class Diagram	53
Figure 27 . University Sequence Diagram	54
Figure 28 . Student Sequence Diagram	55
Figure 29 . Employer Sequence Diagram	55
Figure 30 . Certificate NFT Lifecycle State Diagram	56

LIST OF TABLES

Table 1 User Classed and Characteristics	9
--	---

LIST OF ABBREVIATIONS

ABBREVIATION	DEFINITION
API	Application Programming Interface
cNFT	Compressed Non-Fungible Token
GDPR	General Data Protection Regulation
GraphQL	Graph Query Language
HTTP	Hypertext Transfer Protocol
IPFS	Interplanetary File System
NFT	Non-Fungible Token
PDA	Program Derived Address
PII	Personally Identifiable Information
RLS	Row-Level Security
RPC	Remote Procedure Call
SAS	Solana Attestation Service
ZKP	Zero-Knowledge Proof

REVISION HISTORY

NAME	DATE	REASON FOR CHANGE	VERSION
M.H.M.Shahadh	25/07/26	Initial Creation	1.0

1. Introduction

This document presents a comprehensive Software Requirements Specification (SRS) for the GenuineGrads project, a cutting-edge platform designed to revolutionize academic credential verification. This specification outlines the functional and non-functional requirements necessary for the system's development, adhering to established industry standards for clarity and completeness.

1.1 Purpose

This SRS precisely defines the functional and non-functional requirements for GenuineGrads, a platform built on the Solana blockchain. Its core objective is to empower universities to issue academic certificates and achievement badges as Non-Fungible Tokens (NFTs). Concurrently, the system enables students and employers to verify these credentials on-chain and through the use of zero-knowledge proofs (ZKPs).

This dual focus on NFT-based certificates and zero-knowledge proofs is a fundamental design choice that addresses a critical challenge in digital credentialing. Public blockchain applications often face a tension between the inherent transparency of distributed ledgers and the necessity to protect sensitive personal data. By combining NFTs, which provide immutable and verifiable records on a decentralized ledger, with ZKPs, which allow for the verification of specific claims without revealing the underlying sensitive information, GenuineGrads offers a solution that is both highly verifiable and privacy-preserving. This strategic integration of technologies establishes a foundation for a robust and trustworthy system that respects individual data privacy while ensuring credential authenticity.

1.2 Document Conventions

To ensure clarity and consistent interpretation throughout this technical document, specific formatting conventions have been adopted. The document is formatted using Times New Roman font, size 12, with a line and paragraph spacing of 1.5. All major headings and sub-headings are rendered in a bold style. The text is presented in a normal style with a black color. These

conventions are designed to enhance readability and reduce ambiguity for all audiences engaging with the specification.

1.3 Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) is a foundational document for the GenuineGrads project, designed to be a comprehensive guide for all interested parties. The primary audiences for this document are:

- Project Supervisor/Stakeholders: This includes my university project supervisor and any other academic stakeholders who require a clear understanding of the project's scope, objectives, and high-level functionality.
- The Developer (Myself): This SRS is my primary roadmap for both developing and testing GenuineGrads. It ensures that the project's implementation and quality assurance efforts are aligned with all defined requirements and goals.
- Future Collaborators: Should the project be expanded upon in the future, this document will provide future developers with a detailed and professional overview of the system, its architecture, and its requirements.

For efficient navigation and comprehension, specific reading suggestions are provided:

- Project supervisor and stakeholders seeking a high-level overview of the product and its detailed feature specifications are advised to focus on Sections 2 (Overall Description) and 3 (System features).
- The developer (myself), as both the builder and the tester, will primarily use Sections 3 (System Features), Section 4 (External Interface Requirements) and 5 (Other Nonfunctional Requirements) as a guide for both the implementation and the derivation of test cases.

1.4 Project Scope

The GenuineGrads project is precisely defined by its core offerings and technological boundaries. The system will deliver three primary user-facing portals: a University Portal, designed for institutional registration, the creation of certificate templates, student enrollment, and the minting and revocation of NFTs; a Student Dashboard, enabling students to view their issued certificates and generate Zero-Knowledge Proofs (ZKPs) of their achievements; and an Employer Verification Portal, which facilitates the validation of NFT authenticity and ZKP claims.

From a technological standpoint, the project's scope includes the utilization of off-chain storage for metadata, specifically InterPlanetary File System (IPFS), and on-chain minting and burning mechanisms via Metaplex Bubblegum v2. The generation of zero-knowledge proofs will be powered by Circom and Halo2. The precise definition of the project's scope, including the specific technologies and their versions, is fundamental for effective expectation management and provides a solid foundation for technical implementation. This level of detail is crucial for development teams to align on tools and methods from the outset, thereby reducing ambiguity and minimizing potential rework. The explicit mention of specialized frameworks like Metaplex Bubblegum v2 and Circom/Halo2 also underscores the inherent complexity of the chosen technology stack, highlighting the necessity of such a detailed SRS to guide development and prevent uncontrolled expansion of features.

1.5 References

- [1] A. M. Abdullahi, U. Muhammad, and G. I. O. Aimufua, "Certificate Generation and Verification System Using Blockchain Technology and Quick Response Code," IOSR Journal of Computer Engineering, vol. 24, no. 1, pp. 37-47, 2022.
- [2] 'Faking Degrees: A Thriving industry in South Asia'. Accessed: Jul. 27, 2025. [Online]. Available: <https://www.dailymirror.lk/print/opinion/Faking-Degrees:-A-Thriving-industry-in-South-Asia/172-298260>
- [3] 'Digital Certificate Verification using Blockchain', SlideShare. Accessed: Jul. 27, 2025. [Online]. Available: <https://www.slideshare.net/slideshow/digital-certificate-verification-using-blockchain/254823961>
- [4] mintablelabrat, 'Should All University Certificates Be NFTs?', Mintology Blog. Accessed: Jul. 28, 2025. [Online]. Available: <https://blog.mintology.app/should-all-university-certificates-be-nfts/>
- [5] S. Kaaru, 'Sri Lanka forms committee to study digital currencies and blockchain', CoinGeek. Accessed: Jul. 28, 2025. [Online]. Available: <https://coingeek.com/sri-lanka-forms-committee-to-study-digital-currencies-and-blockchain/>
- [6] 'Sri Lanka sets budget for 2025 digital transformation strategy | Biometric Update'. Accessed: Jul. 28, 2025. [Online]. Available: <https://www.biometricupdate.com/202502/sri-lanka-sets-budget-for-2025-digital-transformation-strategy>
- [7] 'Visa Crypto Thought Leadership – A deep dive on Solana'. Accessed: Jul. 28, 2025. [Online]. Available: <https://usa.visa.com/solutions/crypto/deep-dive-on-solana.html>
- [8] 'Your transcript should belong to you: Using zero-knowledge to protect academic privacy'. Accessed: Jul. 28, 2025. [Online]. Available: <https://www.aleo.org/post/identity/your-transcript-should-belong-to-you-using-zero-knowledge-to-protect/>

2. Overall Description

This section provides a comprehensive, high-level overview of the GenuineGrads system, detailing its architectural context, primary features, target user groups, operational environment, and the key constraints and assumptions that influence its design and implementation.

2.1 Product Perspective

GenuineGrads is architected as a decoupled, multi-layered system, a design choice that inherently supports scalability, modularity, and maintainability. This architectural pattern is particularly well-suited for a system that anticipates high transaction volumes and evolving requirements within the dynamic blockchain ecosystem. The system is composed of several distinct and interconnected layers, each responsible for specific functionalities:

- **Blockchain Layer:** This foundational layer is built upon the Solana blockchain, leveraging the Anchor framework for the development of robust and secure on-chain NFT logic. Solana was chosen for its high throughput and low transaction costs, which are crucial for cost-effective issuance of thousands of certificates and frequent verifications.
- **Backend Layer:** Developed using Node.js and GraphQL, this layer serves as the orchestration hub for various critical operations. It manages Remote Procedure Call (RPC) interactions with the blockchain, processes webhooks (specifically from Helius.dev), handles database access, and orchestrates complex Zero-Knowledge Proof (ZKP) workflows. The integration of Helius for RPC and webhooks represents a strategic optimization for efficient and real-time blockchain interactions, effectively offloading some of the complex blockchain communication overhead from the core backend. Helius provides high-performance RPC and real-time data streaming capabilities, which allows the Node.js backend to interact with the Solana blockchain with reduced latency and improved reliability by leveraging a specialized third-party service. This demonstrates a practical approach to building a robust Web3 application that can handle anticipated transaction volumes.

- Front-end Layer: The user interface is developed using React/Next.js, complemented by ShadCN UI and TailwindCSS to ensure a responsive and intuitive user experience across various devices.
- Storage Layer: This layer employs a hybrid storage strategy. IPFS (InterPlanetary File System) or Arweave are utilized for the decentralized, immutable storage of off-chain metadata associated with the NFTs. This is further complemented by per-university private databases for sensitive, tenant-specific data, and a shared central database for global indices and de-duplication purposes. This multi-faceted storage approach balances decentralization and data immutability with the need for efficient querying and privacy for sensitive information.

The explicit choice of a decoupled architecture with distinct layers is a deliberate design for scalability, modularity, and maintainability. This architectural decision is paramount for a system that aims for widespread adoption and high transaction throughput in the blockchain space. The strategic integration of Helius for RPC and webhooks further underscores an optimization strategy for efficient and real-time blockchain interactions, effectively offloading some of the complex blockchain communication overhead from the core backend.

2.2 Product Features

The GenuineGrads system provides a comprehensive suite of features designed to support the entire lifecycle of academic credential issuance and verification, catering to the distinct needs of each user group. The feature set includes:

- For Universities (Issuing Institutions):
 - University Registration & Authentication: Enables institutions to register their profiles and securely authenticate their identity on the platform
 - Student Registration & Management: Supports both individual and bulk student registration via CSV, incorporating a de-duplication mechanism based on National Identity

Card (NIC) hash and wallet address to ensure unique entries across all universities.

- Certificate Template Designer: Offers a user-friendly drag-and-drop interface for designing customizable certificate layouts, including text fields, logos, seals, and dynamic QR code placement.
 - Bulk & Single Certificate Issuance: Facilitates the minting of compressed NFTs (cNFTs) for certificates, either individually or in large batches, utilizing Metaplex Bubblegum v2 for efficiency.
 - Achievement Badge Issuance & ZKP Commitment: Allows for the issuance of achievement badges and the generation of Zero-Knowledge Proof (ZKP) commitments off-chain for achievements requiring privacy, such as GPA thresholds.
 - Certificate Revocation: Provides administrators the capability to revoke issued certificates by burning the corresponding cNFTs via Metaplex Bubblegum v2.
 - Analytics Dashboard: Offers universities insights into certificate issuance volumes, student participation, and verification logs.
 - Student Onboarding via Email Invitations: Automates the process of inviting registered students to the platform via email.
 - Decentralized Wallet Integration: Enables universities to connect a blockchain wallet for secure NFT issuance.
- For Students:
- Student Profile Management: Allows students to manage their personal details and link their blockchain wallets.

- Request Certificates: Provides a mechanism for students to request certificates from universities, if such a feature is enabled by the institution.
 - View Issued Certificates: Enables students to view and manage their blockchain-based certificates (cNFTs).
 - Share Certificates: Facilitates easy sharing of credentials with employers or recruiters through direct links or QR codes.
 - Zero-Knowledge Proof (ZKP) Generation: Empowers students to generate privacy-preserving proofs of achievement, such as proving a GPA threshold without revealing the exact score.
 - NFT Certificate Storage: Ensures secure storage of academic certificates as NFTs in personal blockchain wallets, providing students with true ownership of their credentials.
- For Employers / Recruiters:
- Certificate Verification: Allows instant validation of credentials by scanning a QR code or entering a unique certificate ID.
 - Claim-Based Verification (ZKP): Enables verification of specific student achievements (e.g., "Has this student made the Dean's List?") without requiring access to full transcripts, preserving student privacy.

The system's detailed features show a thorough grasp of the entire credentialing process, from creating to verifying credentials, and are tailored to each user group's specific needs. By including both Achievement Badge Issuance & ZKP Commitment and Certificate Issuance, the platform goes beyond traditional degrees to offer more granular, privacy-focused credentialing. This design is forward-thinking and aligns with new trends in micro-credentials and verifiable claims, making it possible to verify individual skills or smaller accomplishments while still protecting user privacy.

2.3 User Classes and Characteristics

The GenuineGrads system is designed to serve distinct user classes, each with specific roles, responsibilities, and interactions. Clearly defining these user classes and their characteristics is fundamental for effective user experience design, precise access control, and the development of tailored interfaces and permissions.

User Class	Key Characteristics
University Admin	Manages institution profile, student data, certificate templates, issuance, revocation, and analytics
Student	Connects wallet, views issued certificates, requests ZKP proofs, and shares credentials.
Employer / Verifier	Scans QR codes or enters mint IDs to view certificate metadata, and uploads ZKP proofs for validation.
Regulatory Body	who oversees and regulate univerisities.

Table 1 User Classed and Characteristics

This table offers a clear, quick overview for all project stakeholders, defining each user's role, core responsibilities, and how they will interact with the platform. This structured format ensures the system effectively meets user needs by helping UI/UX designers create suitable interfaces, guiding developers on user stories, and assisting QA teams in developing role-specific test cases.

2.4 Operating Environment

The GenuineGrads system is designed to work in specific environments to ensure it is widely accessible and performs reliably. These environments include:

- Web: The system's web interfaces are designed to be compatible with modern browsers, specifically Chrome, Firefox, and Safari, supporting ES6+ standards to ensure a consistent and rich user experience.

- Mobile: A responsive design approach ensures that the system is fully functional and user-friendly on mobile devices. Wallet integration is facilitated through popular mobile-compatible Solana wallets such as Phantom, Solflare and Backpack, streamlining user interactions within the mobile ecosystem.
- Backend: The backend services are deployed as Docker containers on cloud Virtual Machines (VMs), providing portability and scalability. These services run on Node.js version 18 or higher, leveraging its asynchronous capabilities for efficient handling of requests.
- Blockchain: The core blockchain operations are conducted on Solana Devnet (for testing and development) and Solana Mainnet (for production). Interactions with the blockchain are efficiently managed through Helius webhooks, enabling real-time event processing and data synchronization.
- Storage: For persistent data storage, the system utilizes IPFS Gateways for decentralized content addressing and retrieval of off-chain metadata. Relational databases such as PostgreSQL or MySQL are employed for structured data storage, including user profiles and global indices.

Specifying the operating environment early in the development lifecycle is crucial for technology stack alignment, infrastructure planning, and setting realistic performance expectations. The explicit mention of specific popular Solana wallets, such as Phantom and Backpack, demonstrates a practical understanding of the Solana ecosystem's user interaction patterns and a commitment to user accessibility within that environment. This level of detail guides development teams in setting up their environments, choosing compatible libraries, and ensuring interoperability within the broader Solana ecosystem. It also implies specific compatibility testing requirements to ensure the system functions correctly across all stated environments.

2.5 Design and Implementation Constraints

The development and implementation of GenuineGrads are subject to several critical constraints that directly influence architectural and technical decisions. These constraints are essential for guiding the development team and ensuring the system meets specific non-functional requirements:

- Smart Contract Framework: All smart contracts developed for GenuineGrads must utilize the Anchor framework in conjunction with Metaplex Bubblegum v2. This constraint ensures adherence to a standardized, secure, and efficient development paradigm for Solana programs, leveraging existing battle-tested libraries for compressed NFTs.
- Database Security: The database schema must enforce Row-Level Security (RLS) for all multi-tenant private databases. This is a crucial security and multi-tenancy design constraint, ensuring strict data isolation where data belonging to one university cannot be accessed by another, which is paramount for a shared platform handling sensitive educational records.
- IPFS Upload Latency: The latency for uploading metadata to IPFS must not exceed 5 seconds per metadata batch. This performance target directly addresses a potential bottleneck in decentralized storage interactions, ensuring a responsive user experience during certificate issuance, especially for bulk operations.
- ZKP Generation Time: The off-chain generation of Zero-Knowledge Proofs (ZKPs) per badge must be completed within 30 seconds. This constraint highlights a strong focus on user experience and system responsiveness for computationally intensive cryptographic operations. Meeting this target is vital for the practical usability of the privacy-preserving features.

These constraints are crucial non-functional requirements that directly impact the system's architecture and how it's built. The specific goals for IPFS upload and ZKP generation show a

proactive effort to prevent potential slowdowns from decentralized storage and complex cryptographic tasks. This ensures the system is responsive and provides a good user experience.

2.6 User Documentation

Comprehensive and accessible user documentation is an integral part of the GenuineGrads project, designed to facilitate user adoption and maximize usability. The system will provide:

- In-app Guided Tours: Context-sensitive guided tours will be integrated directly within the application, particularly for complex features such as the certificate template designer and various dashboards.¹ These tours provide immediate, interactive assistance, significantly reducing the learning curve for new users and improving overall user satisfaction.
- Workshops and Awareness Programs: To address the unique complexities of a blockchain-based system, the project will offer workshops and educational programs. These sessions will focus on practical, hands-on training for system usage and, more critically, on the principles of secure digital asset management. This includes topics like safely managing cryptographic wallets, understanding private keys, and recognizing common security threats, thereby empowering users to interact with the system confidently and securely.

This focus on user documentation highlights the project's dedication to making the system both usable and easy to adopt. The project recognizes that a technically advanced system, which introduces new concepts like NFTs and ZKPs to potentially non-technical users such as university administrators, employers, and students, requires clear and accessible guidance. In-app guided tours are a modern approach to user experience, allowing users to get help without leaving the application. This directly improves their experience and helps ensure the system's successful adoption and efficient use.

2.7 Assumptions and Dependencies

The successful development and operation of GenuineGrads rely on several critical assumptions and external dependencies. Clearly stating these factors is essential for identifying potential risks and external influences that could impact project success:

- University Data Provision: It is assumed that participating universities will consistently supply valid National Identity Card (NIC) hashes and wallet addresses for student registration. This assumption is crucial for the student de-duplication process and the accurate association of certificates with individual students. Inconsistent or invalid data could lead to data quality issues and operational inefficiencies in the student registration and de-duplication processes.
- Helius RPC Quota Sufficiency: The project assumes that the Helius RPC (Remote Procedure Call) quotas will be adequate to accommodate the expected volumes of NFT minting transactions. This points to a potential scalability bottleneck if actual usage significantly exceeds projections, necessitating proactive monitoring and potential upgrades to Helius service plans to maintain performance.
- IPFS Node Uptime: A dependency exists on IPFS nodes maintaining an uptime of 99% or greater. This highlights the reliance on the decentralized storage network for metadata availability and underscores the need for robust pinning services to ensure the long-term accessibility of certificate metadata.
- Circom/Halo2 Circuit Artifacts: It is assumed that all Circom/Halo2 circuit artifacts required for Zero-Knowledge Proof generation and verification will be pre-compiled and properly versioned prior to their integration into the system. This indicates that the development and compilation of ZKP circuits are specialized tasks, potentially external to the core application development, and their readiness is a prerequisite for ZKP functionality.

Clearly stating these assumptions and dependencies is key for effective risk management. The project's reliance on external services like Helius and IPFS means their performance, uptime, and cost directly impact the system's stability and expenses. The assumption about university data quality highlights a dependency on client institutions; if their data isn't consistent, it could affect the student de-duplication process. These factors need to be continuously monitored with contingency plans in place to handle potential issues.

3. System Features

This section provides a detailed breakdown of the functional requirements for the GenuineGrads system, organized by core features. Each feature includes a description, its priority, typical stimulus/response sequences, and specific functional requirements that must be met during development.

3.1 University Registration & Authentication

This feature enables new universities to onboard onto the GenuineGrads platform and for their administrators to securely access their institutional portal.

3.1.1 Description and Priority

University administrators are required to register their institution and subsequently authenticate their identity through a Solana wallet signature. This process is deemed of High priority, as it forms the foundational step for any university to utilize the system's credentialing capabilities. The implementation of wallet-based authentication for universities is a critical design choice that leverages blockchain's native security primitives for identity verification, moving beyond traditional username/password systems. This approach inherently enhances security by decentralizing identity management and aligns strongly with the principles of Web3.

3.1.2 Stimulus/Response Sequences

- Stimulus: A university administrator navigates to the "Register University" page within the GenuineGrads University Portal. The administrator then completes the required form fields, including the university's name, domain, and their designated wallet address. Upon submission, a POST request is initiated to the /registerUniversity endpoint.
Response: The system processes the registration request. It stores the university's record in the shared central database and simultaneously invokes an on-chain Program Derived Address (PDA) registration for the university's designated mint authority. A success confirmation is returned to the administrator upon completion.¹

- Stimulus: For subsequent logins, the administrator connects their Solana wallet to the platform. The system presents a challenge message that the administrator must sign using their wallet. A POST request is then sent to the /auth/login endpoint.

Response: Upon successful verification of the wallet signature, the system issues a JSON Web Token (JWT) to the administrator, granting authenticated access to the university's portal.

3.1.3 Functional Requirements

- REQ-1: The system must accept all required university details and successfully create a corresponding entry in the shared registry database.
- REQ-2: The system must programmatically register an on-chain mint authority PDA for the newly registered university on the Solana blockchain.
- REQ-3: The system must securely authenticate university administrators by verifying their Solana wallet signatures against a cryptographic challenge.¹
- REQ-4: Upon successful authentication, the system must issue a JWT that is securely scoped to the authenticated university_id, ensuring proper authorization for subsequent actions.

3.2 Student Registration & De-duplication

This feature manages the enrollment of students into the system, ensuring data integrity and preventing duplicate entries.

3.2.1 Description and Priority

This feature allows university administrators to register students either individually or in bulk via CSV file uploads. A critical component of this process is the de-duplication mechanism, which checks the student's National Identity Card (NIC) hash and wallet address against a shared global index to prevent duplicate entries across all participating universities. This feature is assigned a High priority due to its fundamental role in maintaining accurate student records and ensuring the uniqueness of credentials.

3.2.2 Stimulus/Response Sequences

- Stimulus: A university administrator uploads a CSV file containing student data or manually fills out an individual student registration form. The backend system automatically hashes the provided NIC for de-duplication purposes.

Response: The backend system performs a lookup in the GlobalStudentIndex database, querying for existing records based on the generated nic_hash or the provided wallet_address. The query is structured as: `SELECT * FROM GlobalStudentIndex WHERE nic_hash=:hash OR wallet_address=:wallet;`

- Response (Conditional): If a matching record is found in the GlobalStudentIndex, indicating a duplicate, the system returns a duplicate error message to the administrator. Otherwise, if no duplicate is found, the student's profile is inserted into both the GlobalStudentIndex and the university's private student database, under the correct university_id.

Response: For bulk operations, the system returns a comprehensive success list of all successfully registered students, along with any specific failure feedback for individual rows that encountered errors during the process.

3.2.3 Functional Requirements

- REQ-5: The system must strictly enforce the uniqueness of both the NIC hash and the wallet address across all registered universities within the GlobalStudentIndex.

- REQ-6: The system must accurately write the full student profile details to the private database associated with the correct university_id.
- REQ-7: For bulk student registration operations, the system must provide clear, per-row success or failure feedback to the university administrator, indicating the status of each student record processed.

3.3 Certificate Template Design

This feature provides universities with the tools to create and manage the visual layout of their digital certificates.

3.3.1 Description and Priority

The system must provide a user interface that allows university administrators to design custom certificate layouts. This includes the ability to place and configure text fields, upload and position logos and seals, and designate areas for QR code placement. This feature is assigned a Medium priority, as it enables customization but is not critical for basic issuance functionality.

3.3.2 Stimulus/Response Sequences

- Stimulus: A university administrator accesses the certificate template designer interface. They then configure various design elements, such as adding text boxes, uploading images (logos, seals), and defining the position of dynamic QR codes. After completing the design, the administrator clicks a "Save" button.

Response: The frontend sends a POST request to the /certificate-template endpoint. This request includes a JSON object representing the template's layout configuration and any associated image assets.

Response: The backend processes the request by storing the template's JSON metadata in the university's private database. Concurrently, it uploads the image assets to IPFS

(InterPlanetary File System) and returns the Content Identifier (CID) for each uploaded asset to the frontend.

3.3.3 Functional Requirements

- REQ-8: The template designer must support intuitive drag-and-drop functionality for positioning various elements, including text fields and images.
- REQ-9: The system must automatically generate a QR-code placeholder within the template, which will dynamically link to the {mint_address} macro upon certificate issuance.
- REQ-10: The system must securely store the template's JSON configuration and the associated asset CIDs (Content Identifiers) in the database, enabling their reuse for future certificate issuances.

3.4 Bulk & Single Certificate Issuance

This feature manages the process of minting digital certificates as NFTs on the blockchain.

3.4.1 Description and Priority

This feature enables university administrators to mint compressed Non-Fungible Tokens (cNFTs) for academic certificates, either individually or in large batches. The minting process is conducted via Metaplex Bubblegum v2, embedding relevant student data and achievement details within the NFT metadata. This functionality is assigned a High priority due to its core role in the system's primary objective of issuing verifiable credentials.

3.4.2 Stimulus/Response Sequences

- Stimulus: A university administrator selects one or more students for whom certificates are to be issued and chooses a pre-designed certificate template. The administrator then initiates a POST request to the /issue-certificates endpoint.

Response: For each selected student, the backend system performs the following sequence:

it fetches the student's profile, retrieves the chosen certificate template, and compiles the student's achievement list. This data is then assembled into a comprehensive metadata JSON object. This metadata is subsequently uploaded to IPFS, and finally, the backend calls the Helius mint endpoint to initiate the cNFT minting process on the Solana blockchain.¹ The use of Metaplex Bubblegum v2 for cNFT minting allows for significant cost reduction and scalability compared to traditional NFTs, enabling the efficient processing of large batches of certificates.

- Response: Upon receiving a successful mint webhook notification from Helius, the system records the newly issued Certificate in the university's private database and logs the MintActivityLog in the shared central database, including on-chain transaction IDs and timestamps. Helius's webhook capabilities are crucial for real-time tracking of on-chain events, ensuring the backend system is immediately aware of successful minting operations.

3.4.3 Functional Requirements

- REQ-11: The system must reliably loop through and process batches of up to 1000 students per issuance operation without failure.
- REQ-12: The system must implement retry mechanisms for transient failures during the issuance process and accurately surface the final status (success/failure) for each certificate in the batch.
- REQ-13: The system must log all on-chain transaction IDs and their corresponding timestamps for every minted certificate, ensuring a complete audit trail.

3.5 Badge & ZKP Commitment Generation

This feature enables the creation of privacy-preserving achievement badges using Zero-Knowledge Proofs.

3.5.1 Description and Priority

For specific achievements that require privacy (e.g., demonstrating a GPA of ≥ 3.5 without revealing the exact GPA), the system must generate a Zero-Knowledge Proof (ZKP) commitment off-chain and securely store its hash. This feature is assigned a Medium priority, as it enhances privacy but is not essential for basic certificate issuance.

3.5.2 Stimulus/Response Sequences

- Stimulus: During the certificate issuance flow, if an achievement badge is configured with requires ZKP=true, the system automatically generates the necessary commitment inputs for the ZKP circuit. These inputs are then sent via a POST request to the ZKP-service/store-commitment endpoint. The integration with Circom/Halo2 circuits is fundamental here, as these frameworks are used to define the cryptographic logic for generating the ZKP commitments.

Response: The ZKP service processes the inputs and, upon successful generation, stores the derived commitment_hash in the corresponding Achievement record within the database. This hash serves as a compact, verifiable representation of the private claim.

3.5.3 Functional Requirements

- REQ-14: The system must seamlessly integrate with pre-compiled Circom/Halo2 circuits to accurately derive the ZKP commitment hash from the private inputs.
- REQ-15: The system must securely persist the ZKP commitment inputs in a dedicated, private ZKP store, ensuring their confidentiality until a proof is requested.
- REQ-16: The system must expose a /generate-zkp endpoint, allowing students to initiate the creation of ZKP proofs for their eligible achievements.

3.6 Certificate Revocation

This feature provides universities with the ability to invalidate previously issued certificates.

3.6.1 Description and Priority

University administrators must be able to revoke issued certificates by initiating the burning of the corresponding cNFTs via Metaplex Bubblegum v2. This feature is assigned a Medium priority, as it is essential for managing credential validity but is less frequent than issuance.

3.6.2 Stimulus/Response Sequences

- Stimulus: A university administrator selects a specific certificate for revocation within the University Portal. They then initiate a POST request to the /revoke-certificate endpoint.

Response: The backend system calls the burn() instruction on the Metaplex Bubblegum v2 smart contract for the selected cNFT. Upon receiving an on-burn webhook notification, the system updates the Certificate record in the private database by setting Certificate.revoked=true and inserts a corresponding record into the RevokedCertIndex in the shared database. This ensures that the certificate's status is reflected both on-chain (via burning the cNFT) and off-chain for quick lookups.

3.6.3 Functional Requirements

- REQ-17: The system must reject any revocation requests for cNFTs that do not exist or have already been burned, preventing erroneous operations.
- REQ-18: The system must log the reason for revocation and the timestamp of the revocation event, maintaining a clear audit trail for all invalidated certificates.

3.7 Student Dashboard & ZKP Proof Generation

This feature provides students with a personalized portal to manage their credentials and generate privacy-preserving proofs.

3.7.1 Description and Priority

Students must be able to view all their issued cNFTs (certificates and badges) and initiate the generation of Zero-Knowledge Proofs (ZKPs) for eligible achievements that require privacy. This feature is assigned a High priority, as it is central to the student's interaction with and benefit from the system.

3.7.2 Stimulus/Response Sequences

- Stimulus: A student logs into the Student Dashboard using their Solana wallet, which results in a JWT being issued for authentication.

Response: The student's frontend sends a GET request to the /certificates endpoint. The backend fetches the student's on-chain cNFTs via Helius's NFT API and returns a list of these NFTs along with their associated badge metadata to the student's dashboard.¹ Helius provides comprehensive APIs for querying and managing digital assets on Solana, enabling efficient retrieval of cNFT data.

- Stimulus: The student clicks a "Generate Proof" button for an eligible achievement. This triggers a POST request to the /generate-zkp endpoint.

Response: The backend retrieves the securely stored commitment inputs for the selected achievement, invokes the Circom/Halo2 ZKP generator service to compute the proof, and returns the generated proof JSON along with public signals to the student. The student is then able to download this proof.¹ The ability of Solana v1.16 to verify SNARK proofs efficiently via cryptographic syscalls allows for potential on-chain verification of simple ZKPs, enhancing trustlessness.

3.7.3 Functional Requirements

- REQ-19: All requests from the Student Dashboard must be authenticated via a valid JWT.
- REQ-20: The system must limit the generation of ZKP proofs to one per eligible achievement per student, preventing redundant or excessive proof generation.

- REQ-21: The system must store metadata related to generated proofs, including a verified flag, to track proof usage and validity.

3.8 Employer Verification Portal

This feature provides employers and other verifiers with a streamlined interface to validate academic credentials and privacy-preserving claims.

3.8.1 Description and Priority

Employers and other verifiers must be able to verify the authenticity of certificates (NFTs) and validate Zero-Knowledge Proof (ZKP) claims through a dedicated web user interface. This feature is assigned a High priority, as it is the primary means by which the system's core value proposition of verifiable credentials is realized for external parties.

3.8.2 Stimulus/Response Sequences

- Stimulus: An employer scans a QR code present on a certificate or manually inputs a certificate's mint ID into the Employer Verification Portal. This action initiates a GET request to the /verify-certificate?mint=... endpoint.

Response: The backend system fetches the certificate's metadata from Helius's NFT API and simultaneously checks the RevokedCertIndex to ascertain the certificate's current status. The system then returns the validity status (e.g., "Valid," "Revoked") and the associated metadata to the employer's interface.

- Stimulus: If a student has provided a ZKP, the employer uploads the proof JSON file or pastes the public inputs into the portal. This triggers a POST request to the /verify-zkp endpoint.

Response: The backend utilizes the Halo2 verifier (or Circom verifier) to validate the uploaded ZKP against its public inputs. The system then returns a boolean is_valid status, indicating whether the proof is cryptographically sound and the claim is true.

3.8.3 Functional Requirements

- REQ-22: The system must provide clear and unambiguous status indicators for certificate authenticity, displaying "Valid," "Revoked," or "Invalid Proof" as appropriate.
- REQ-23: The system must be capable of accepting ZKP JSON and public inputs via both file upload and direct text paste mechanisms in the Employer Verification Portal.

4. External Interface Requirements

4.1 User Interfaces

This section details the various interfaces through which the GenuineGrads system interacts with its users, underlying hardware, other software systems, and communication protocols. These interfaces are crucial for the system's interoperability and overall functionality.

The GenuineGrads system will feature three distinct user interfaces, each tailored to the specific needs and workflows of its primary user classes:

- University Portal: This interface will consist of responsive React pages, providing university administrators with comprehensive tools for institution registration, certificate template design, certificate issuance (both single and bulk), certificate revocation, and analytics reporting. The design prioritizes an intuitive drag-and-drop user interface to enhance usability.

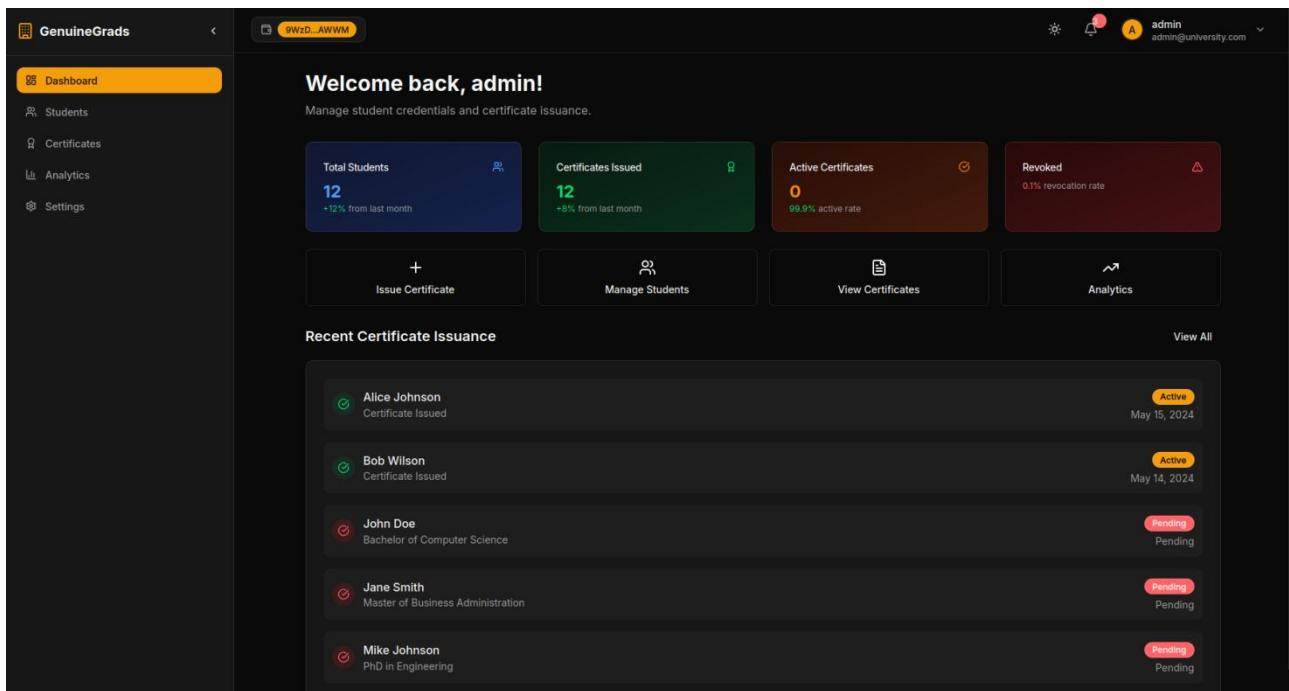


Figure 1. University Admin Dashboard

The screenshot shows the 'Students' section of the GenuineGrads application. On the left, a sidebar menu includes 'Dashboard', 'Students' (which is highlighted in orange), 'Certificates', 'Analytics', and 'Settings'. The main area is titled 'Students' and displays a search bar with placeholder text 'Search by name, NIC, or email...'. Below the search bar is a table titled 'Registered Students (12)'. The table has columns for Name, NIC, Email, Program, Date Added, Certificates, and Actions. Each row represents a student with their details and a three-dot menu icon.

Name	NIC	Email	Program	Date Added	Certificates	Actions
John Doe Dean's List 2023 +1 more	NIC123456789	john.doe@student.edu	Bachelor of Computer Scien...	Jan 15, 2024	3	...
Jane Smith Academic Excellence Award	NIC987654321	jane.smith@student.edu	Master of Business Adminis...	Feb 20, 2024	1	...
Mike Johnson Research Grant Recipient +1 more	NIC456789123	mike.johnson@student.edu	PhD in Engineering	Mar 10, 2024	2	...
Sarah Wilson Creative Writing Award	NIC789123456	sarah.wilson@student.edu	Bachelor of Arts	Jan 30, 2024	1	...
David Brown	NIC321654987	david.brown@student.edu	Master of Science	Feb 15, 2024	0	...
Alice Johnson Dean's List 2024 +1 more	NIC111222333	alice.johnson@student.edu	Bachelor of Computer Scien...	Mar 5, 2024	2	...
Bob Wilson Leadership Award	NIC444555666	bob.wilson@student.edu	Master of Business Adminis...	Mar 12, 2024	1	...
Carol Davis Research Excellence +1 more	NIC777888999	carol.davis@student.edu	PhD in Engineering	Mar 18, 2024	3	...
Eve Martinez Creative Excellence	NIC800111222	eve.martinez@student.edu	Bachelor of Arts	Mar 25, 2024	1	...
Frank Garcia						

Figure 2. Student Management

The screenshot shows the 'Bulk Upload Students' page. The sidebar menu is identical to Figure 2. The main area has a back link 'Back to Students' and a title 'Bulk Upload Students'. It includes a warning message: 'Upload multiple students using a CSV file' and 'Only students with connected wallets can be registered'. A large central box contains a 'Upload CSV File' section with a file input field labeled 'Choose a CSV file', instructions about required columns, and a 'Select File' button. To the right is a 'Upload Guidelines' box with rules for CSV files. Below these are 'Upload Statistics' and 'Quick Actions' sections.

Upload Guidelines

- CSV file must include: name, nic, email, program, walletAddress (required), achievements (optional)
- Achievements column is optional and can contain multiple values separated by commas.
- All email addresses must be in valid format.
- Maximum 100 students per upload for optimal performance.

Upload Statistics

Total Uploads	24
This Month	+8
Success Rate	98.5%
Avg. Students/Upload	15

Quick Actions

- Download Template
- Add Single Student

Figure 3. Students Bulk Registration

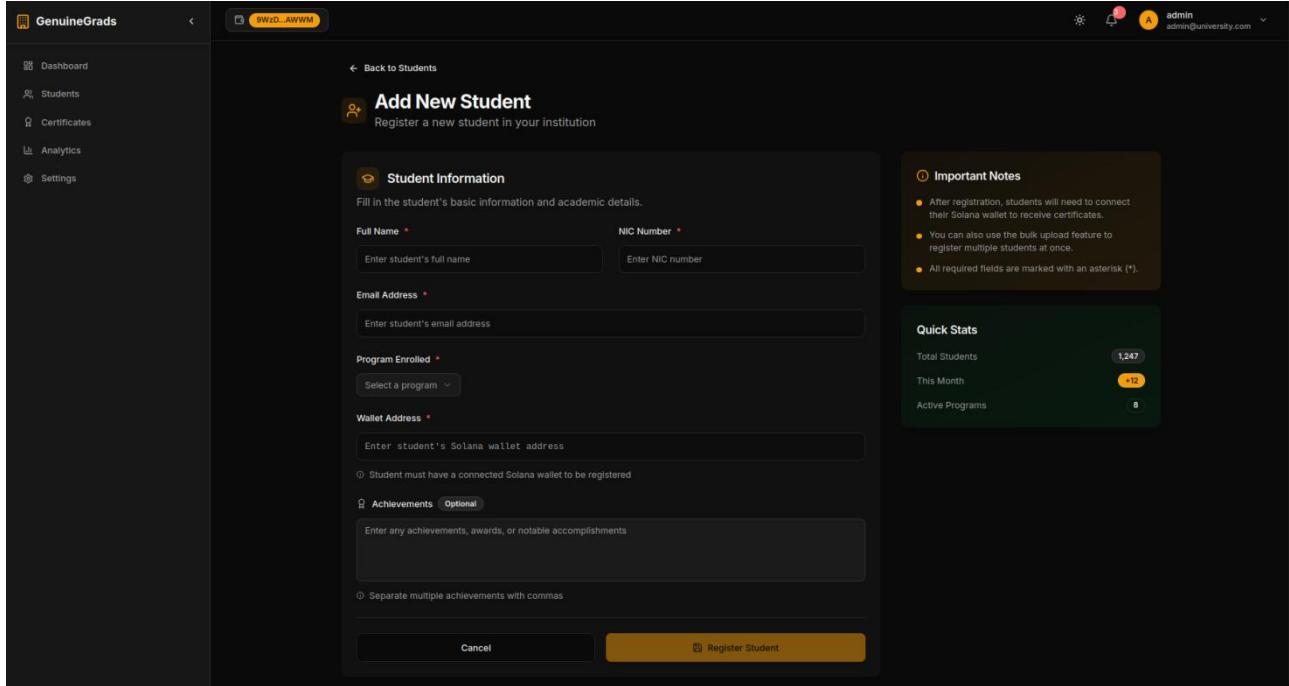


Figure 4. Students Individual Registration

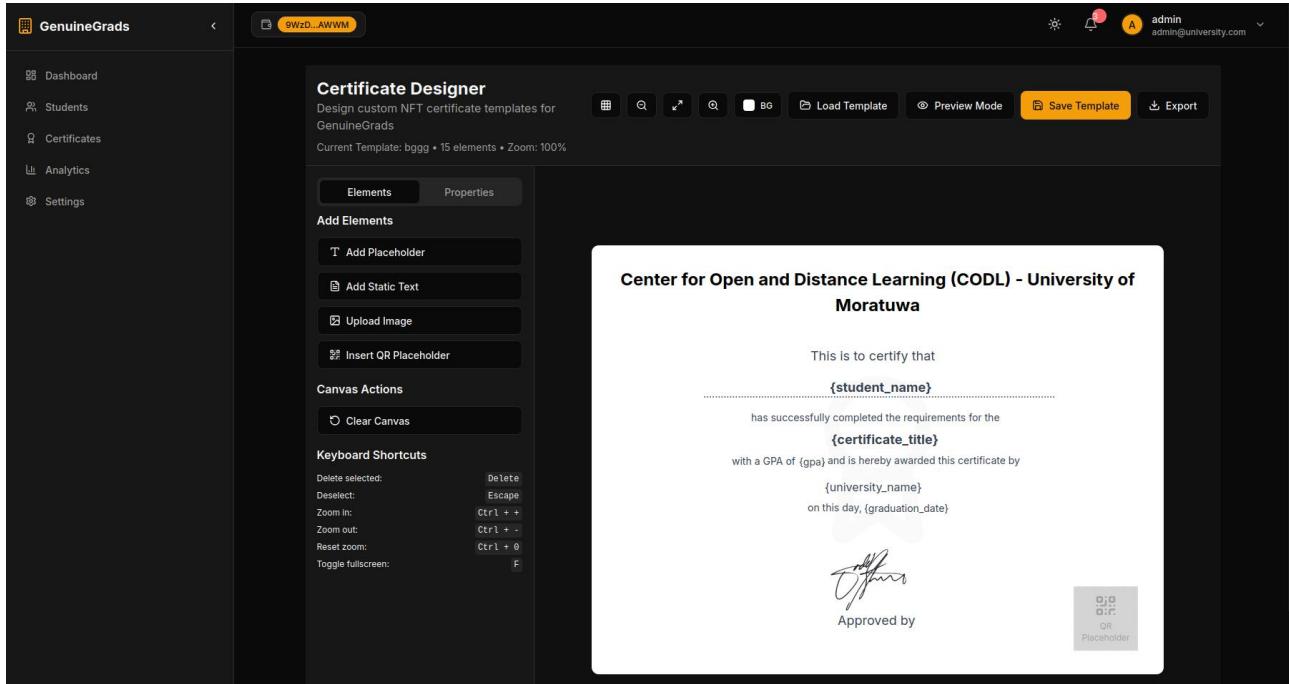


Figure 5. Certificate Designer

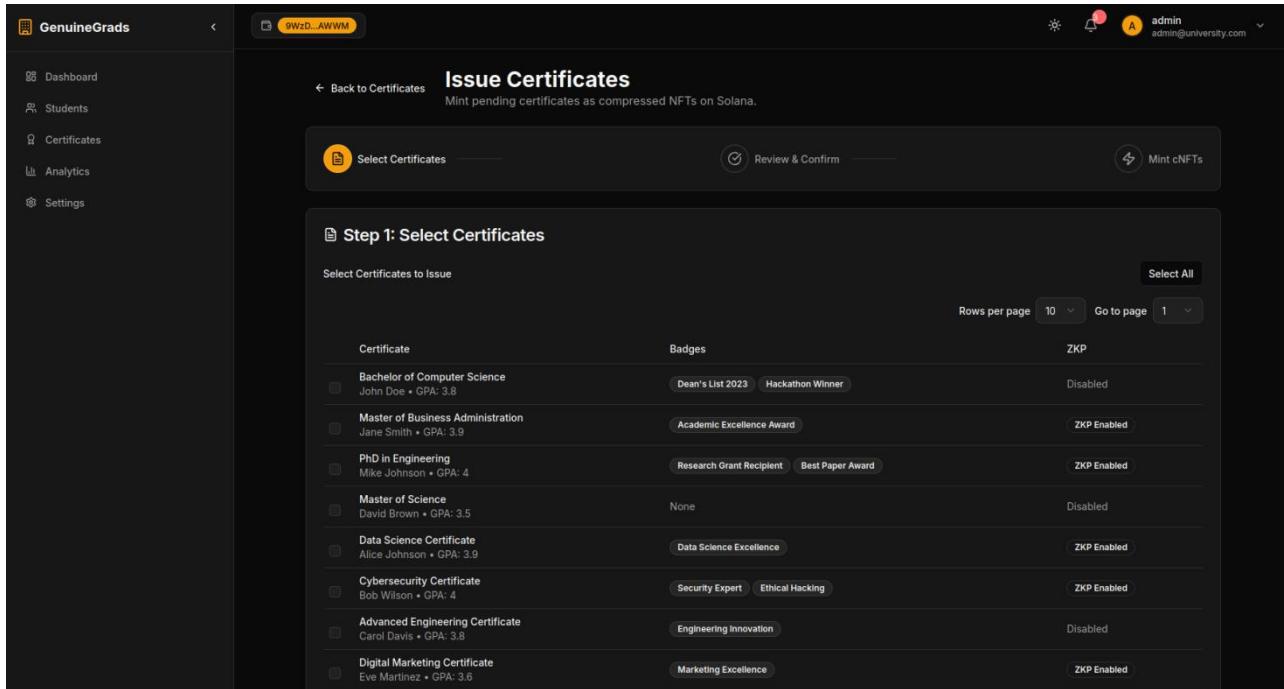


Figure 6. Issue Certificate

Certificates							
Manage issued certificates and their status.							
Certificates (12)		Actions					
Student	Certificate	GPA	Issue Date	Status	Badges		Actions
John Doe Student ID: stu-001	Bachelor of Computer Science	3.8	Pending	Pending	2 badges	...	
Jane Smith Student ID: stu-002	Master of Business Administration	3.9	Pending	Pending	1 badge	...	
Mike Johnson Student ID: stu-003	PhD in Engineering	4.0	Pending	Pending	2 badges	...	
Sarah Wilson Student ID: stu-004	Bachelor of Arts	3.7	Pending	Pending	1 badge	...	
David Brown Student ID: stu-005	Master of Science	3.5	Pending	Pending	None	...	
Alice Johnson Student ID: stu-006	Data Science Certificate	3.9	Pending	Pending	1 badge	...	
Bob Wilson Student ID: stu-007	Cybersecurity Certificate	4.0	Pending	Pending	2 badges	...	
Carol Davis Student ID: stu-008	Advanced Engineering Certificate	3.8	Pending	Pending	1 badge	...	
Eve Martinez Student ID: stu-009	Digital Marketing Certificate	3.6	Pending	Pending	1 badge	...	
Frank Garcia Student ID: stu-010	Web Development Certificate	3.7	Pending	Pending	2 badges	...	

Figure 7. Certificate Management

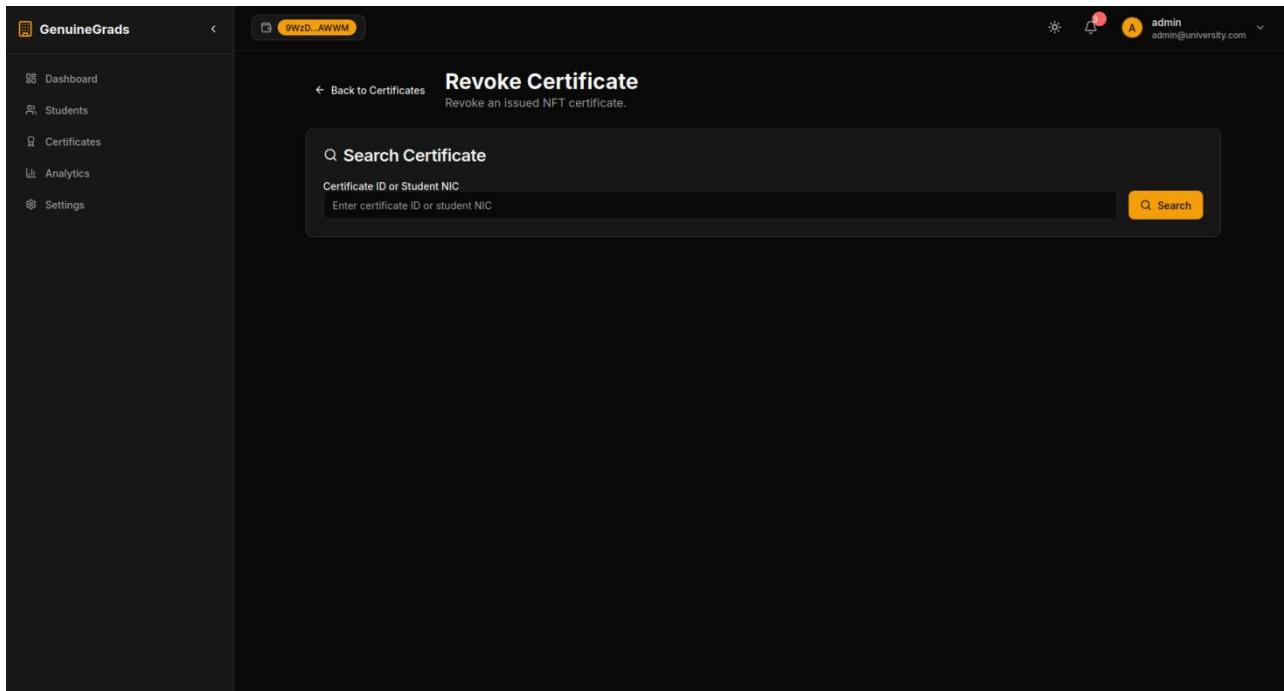


Figure 8. Revoke Certificates

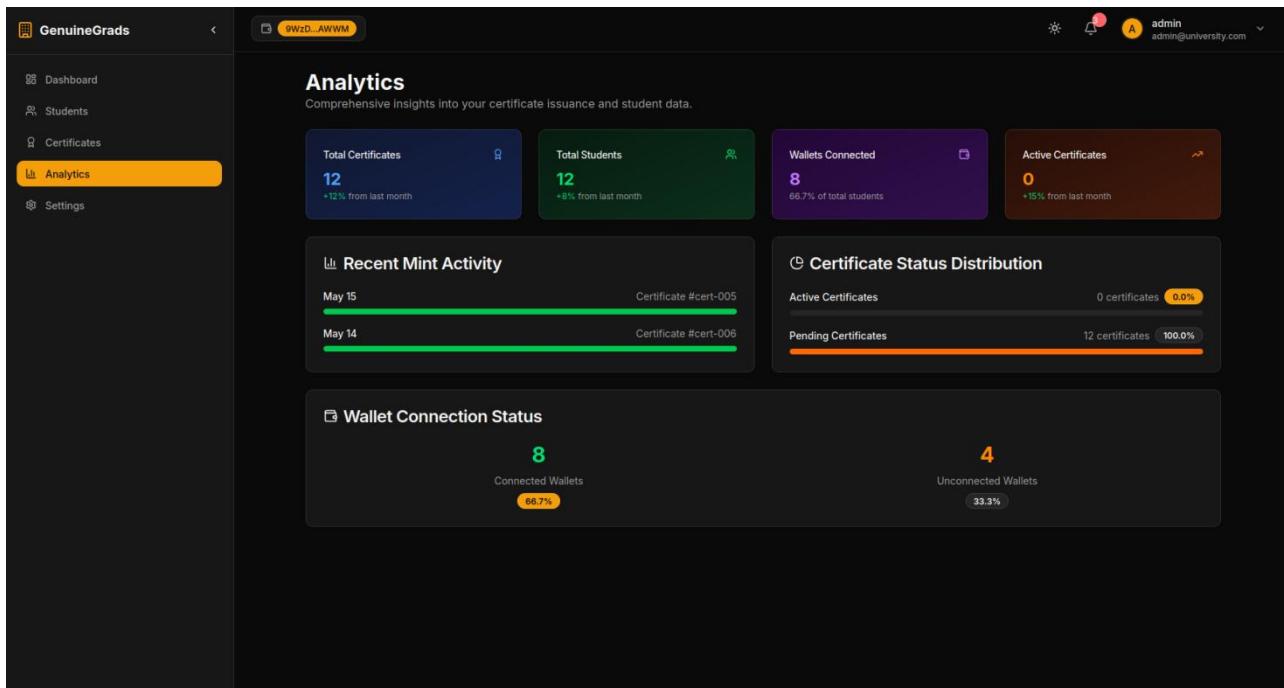


Figure 9. University Analytics Page

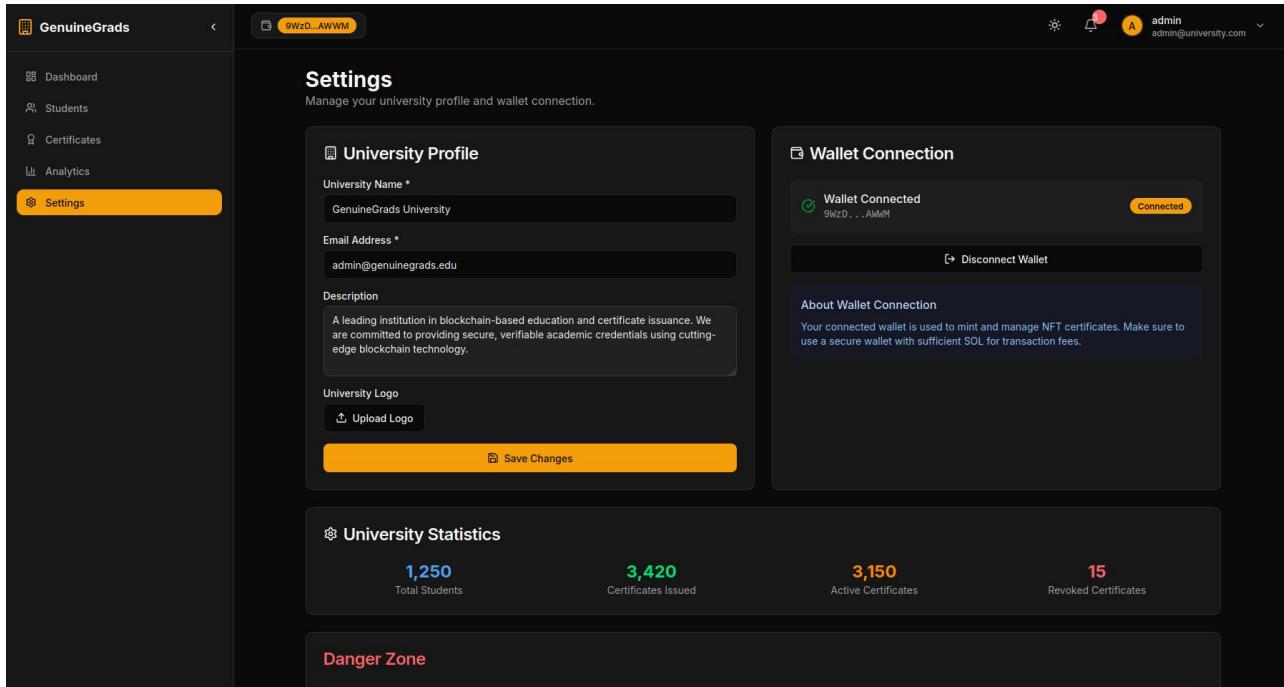


Figure 10. University Settings Page

- Student Dashboard: Designed as a centralized hub for students, this interface will feature a cNFT gallery to display issued certificates and badges. It will also include a proof console for generating Zero-Knowledge Proofs (ZKPs) and controls for sharing credentials with third parties.

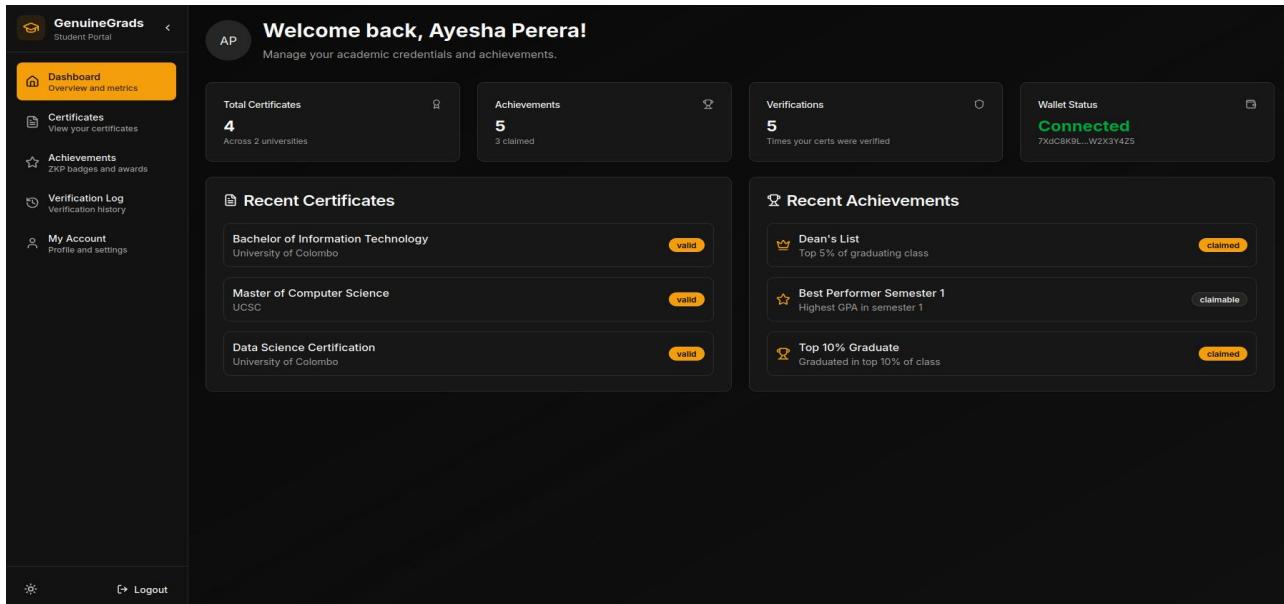


Figure 11. Student Dashboard

My Certificates

View and manage your academic certificates

Filters

Showing 4 of 4 certificates

Bachelor of Information Technology (Valid)

- University: University of Colombo
- Course: Computer Science
- GPA: 3.95
- Issued: 7/20/2024
- Asset ID: ASSET123...

Master of Computer Science (Valid)

- University: UCSC
- Course: Advanced Computing
- GPA: 3.88
- Issued: 12/15/2024
- Asset ID: ASSET456...

Data Science Certification (Valid)

- University: University of Colombo
- Course: Data Science
- GPA: 4.0
- Issued: 9/10/2024
- Asset ID: ASSET789...

Web Development Bootcamp (Valid)

- University: UCSC
- Course: Web Development
- GPA: 3.92
- Issued: 6/5/2024
- Asset ID: ASSET012...

4 Valid | 0 Expired | 0 Pending

Logout

Figure 12. Student View Certificate Page

My Achievements

ZKP badges and academic awards

Filters

Showing 5 of 5 achievements

Dean's List (Claimed)

- Status: Claimed
- Proof ID: ZKP_PROOF_001
- Copy Proof

Best Performer Semester 1 (Claimable)

- Status: Claimable
- Claim ZKP

Top 10% Graduate (Claimed)

- Status: Claimed
- Proof ID: ZKP_PROOF_002
- Copy Proof

Perfect Score (Claimed)

- Status: Claimed
- Proof ID: ZKP_PROOF_003
- Copy Proof

Innovation Award (Claimable)

- Status: Claimable
- Claim ZKP

3 Claimed | 2 Claimable | 0 Locked

Share All

Logout

Figure 13. Student's Achievements Page

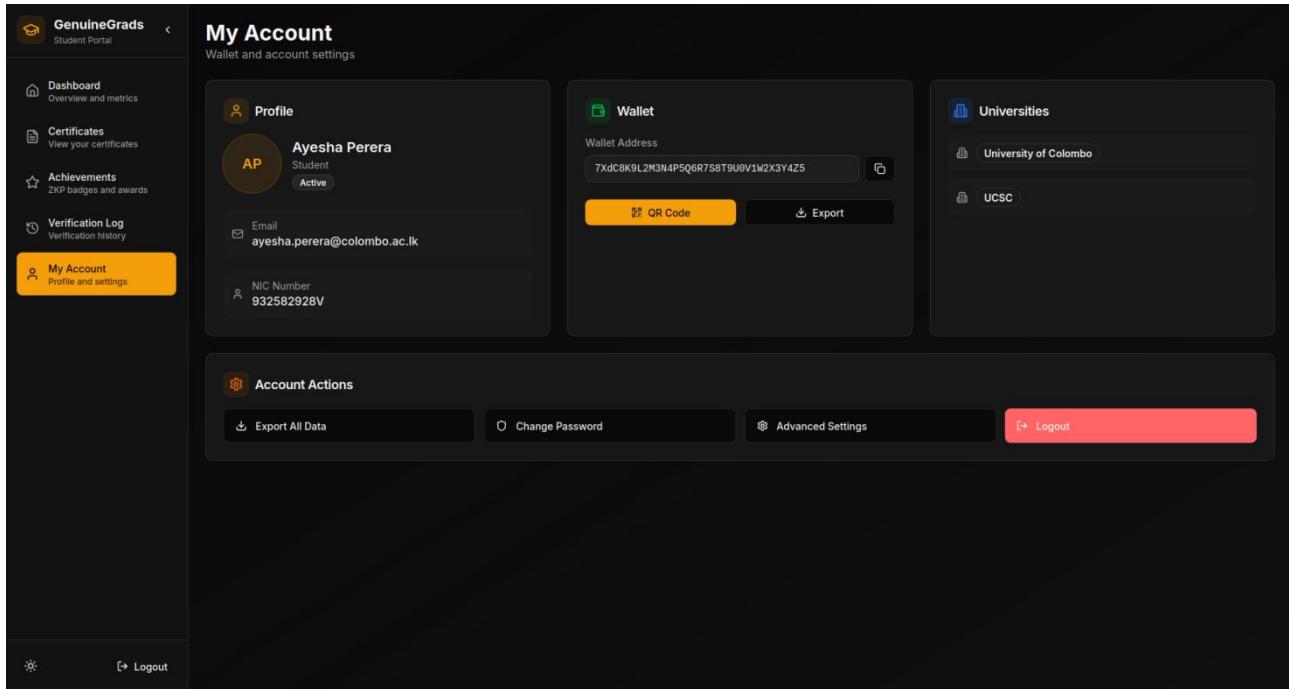


Figure 14. Student Profile Management

The screenshot shows the 'Verification History' page. The sidebar has 'Verification Log' selected. The main area shows summary statistics: 5 Total Verifications, 5 Successful, 0 Failed, and 0 Pending. Below this is a 'Filters' section with a search bar, status dropdown, time range, and a 'Clear Filters' button. A table titled 'Verification Events' lists 5 events with columns for Date, Asset ID, IP Address, Verified By, and Status (all marked as verified).

Date	Asset ID	IP Address	Verified By	Status
12/20/2024	ASSET123XYZ789	203.24.1.1	www.techcorp.lk	verified
12/18/2024	ASSET456ABC123	192.168.1.100	www.startup.io	verified
12/15/2024	ASSET7890DEF456	10.0.0.50	www.consulting.com	verified
12/12/2024	ASSET123XYZ789	172.16.0.25	www.bank.lk	verified
12/10/2024	ASSET012GHII789	8.8.8.8	www.agency.gov.lk	verified

Figure 15. Student Verification Logs Page

- Employer Portal: This portal is streamlined for verification purposes. It will incorporate a QR scanner widget for quick credential lookup, a display area for certificate metadata, and a dedicated proof verifier for validating ZKP claims.

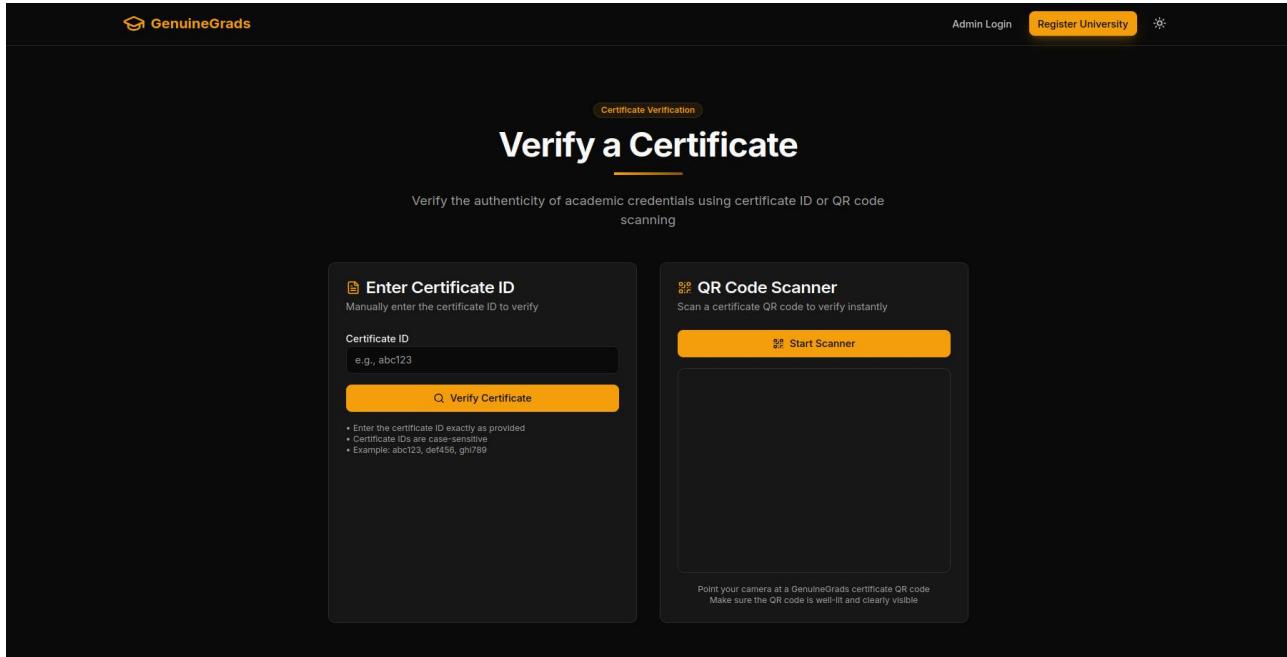


Figure 16. Certificate Verification Page (Public)

4.2 Hardware Interfaces

The GenuineGrads system is primarily a web-based application and does not require direct interfaces with specialized hardware. Its interactions are limited to standard user devices, including desktop computers and mobile devices, and their respective web browsers and blockchain wallet extensions (e.g., Phantom, Backpack). The responsive design ensures compatibility across a wide range of screen sizes and input methods.

4.3 Software Interfaces

The system's functionality relies on seamless integration with several key software interfaces:

- Helius NFT API: This API is crucial for querying on-chain cNFTs (Compressed NFTs) and for receiving real-time webhook notifications related to minting and burning events on the

Solana blockchain.¹ Helius provides high-performance RPC and data streaming capabilities, which are essential for efficient blockchain interactions.

- GraphQL API: The backend exposes a GraphQL API, which serves as the primary interface for frontend consumption of data and interaction with backend services. GraphQL's flexible querying capabilities enable efficient data retrieval, allowing frontends to request precisely the data they need, thereby optimizing network payloads.
- IPFS HTTP API: This API is utilized for uploading and retrieving metadata associated with certificates to the InterPlanetary File System (IPFS). IPFS ensures decentralized and immutable storage of certificate metadata, enhancing the robustness of the system.
- ZKP Generator: This component is exposed as a RESTful CLI (Command Line Interface) service, responsible for performing the computationally intensive task of generating Zero-Knowledge Proofs. Decoupling this as a separate service allows for specialized resource allocation and scalability for ZKP computation.

4.4 Communications Interfaces

All communication within the GenuineGrads system and with external services adheres to stringent security protocols:

- HTTP(s) Calls: All HTTP(s) calls, both internal and external, must be secured using TLS (Transport Layer Security). This ensures encrypted communication, protecting data integrity and confidentiality during transmission.
- Helius RPC (Remote Procedure Calls): The system uses Helius's RPC endpoints for all blockchain-related HTTP(s) calls. This is the primary method for fetching on-chain data and submitting transactions. All communication is secured using TLS to ensure encrypted data transmission, protecting integrity and confidentiality.

- Helius WebSockets: GenuineGrads utilizes Helius's WebSocket services for real-time, event-driven communication. This interface provides immediate feedback to users on critical blockchain activities, such as the status of minting and burning transactions, which significantly enhances the user experience.

5. Other Nonfunctional Requirements

This section outlines the nonfunctional requirements for the GenuineGrads system, which define the quality attributes and operational constraints beyond core functionality. These requirements are critical for the system's performance, reliability, security, and overall quality.

5.1 Performance Requirements

The GenuineGrads system is designed with specific performance targets to ensure a responsive and efficient user experience:

- **Page Loads:** All web pages within the University Portal, Student Dashboard, and Employer Verification Portal must load within 3 seconds on a 4G network connection. This target is crucial for maintaining user engagement and satisfaction.
- **Batch Minting:** The system must be capable of processing batch minting operations for up to 1000 students within a maximum of 10 minutes. This requirement is vital for the scalability and practicality of the system for universities issuing large numbers of certificates.
- **Proof Generation:** The off-chain generation of a Zero-Knowledge Proof (ZKP) per badge must be completed within 30 seconds. This target directly addresses the computational intensity of ZKP generation, ensuring that the privacy-preserving features remain practical and user-friendly.

These performance targets indicate a strong focus on user experience and system responsiveness, proactively addressing potential bottlenecks inherent in decentralized storage and computationally intensive cryptographic operations.

5.2 Safety Requirements

Safety requirements for GenuineGrads focus on protecting sensitive data and ensuring the integrity of the system's outputs:

- Data Privacy by Design: The system must be designed to ensure that no Personally Identifiable Information (PII) is stored in the shared central database.¹ This principle minimizes the risk of large-scale data breaches affecting sensitive student information.
- Template Designer Security: The certificate template designer must be architected in a manner that prevents the embedding or execution of malicious scripts.¹ This protects both university administrators and end-users from potential cross-site scripting (XSS) attacks or other forms of content-based vulnerabilities.

5.3 Security Requirements

Security is a top priority for GenuineGrads because it deals with sensitive academic information and relies on blockchain technology. The main security requirements are:

- Authentication: All API endpoints exposed by the backend must require robust authentication, either through a valid JSON Web Token (JWT) for session-based access or through a Solana wallet-signed authentication for direct blockchain interactions. This ensures that only authorized users or systems can access and manipulate data.
- Row-Level Security (RLS): Row-Level Security (RLS) must be strictly enforced on the shared central database to prevent unauthorized cross-tenant data access. This is a critical measure for data isolation, ensuring that one university's data cannot be viewed or modified by another.
- Input Validation: Comprehensive input validation mechanisms must be implemented across all user and API inputs to protect against common web vulnerabilities, including SQL injection and Cross-Site Scripting (XSS) attacks.

5.4 Software Quality Attributes

Beyond functional and performance aspects, GenuineGrads is committed to delivering a high-quality software product with the following attributes:

- Usability: The system aims for an intuitive and user-friendly interface, particularly highlighted by the drag-and-drop functionality in the template designer. This attribute ensures that users can efficiently and effectively interact with the system with minimal training.
- Reliability: Critical services within the GenuineGrads architecture are expected to maintain an uptime of 99.9% or greater. This high reliability target ensures continuous availability of core functionalities for all stakeholders.
- Maintainability: The system is designed with a modular decoupled, multi-layered architecture and versioned APIs to enhance its maintainability. This modularity simplifies updates, bug fixes, and the introduction of new features without disrupting the entire system.
- Portability: The system is designed to be deployable on major cloud providers, ensuring flexibility in infrastructure choices and ease of migration if required. This portability minimizes vendor lock-in and facilitates disaster recovery strategies.

6. Other Requirements

This section covers other vital requirements that go beyond standard functional and non-functional categories, focusing on governance, compliance, and data management.

- Governance: The system will implement a controlled and secure process for whitelisting new universities onto the platform. This ensures that only authorized and vetted institutions can join the GenuineGrads ecosystem, which is crucial for maintaining the integrity and trustworthiness of the network.
- Compliance: The GenuineGrads system must follow data-protection rules like GDPR, showing a commitment to strong privacy and data handling standards. This includes measures for data minimization, consent management, and protecting individual rights, ensuring all personal information is handled legally and ethically.
- Backup: The system will use a strong backup plan to protect data integrity and ensure it's always available. This includes daily snapshots of all databases and careful maintenance of IPFS pinned CIDs for all off-chain metadata. This multi-layered approach guarantees that all critical data can be restored if there is a system failure or data loss.

7. Future Enhancements

The initial release of GenuineGrads will focus on delivering the core functionalities outlined in this SRS. However, the architecture is designed to accommodate future enhancements that will further extend its capabilities and address evolving needs. These potential future features include:

- Solana Attestation Service (SAS) Integration: A potential future enhancement for the GenuineGrads system is the integration of the Solana Attestation Service (SAS). This would allow government regulatory bodies to officially attest to the registration and legitimacy of universities, creating a verifiable, on-chain record of accreditation. This strategic integration would prevent the registration of fraudulent institutions, significantly strengthening the platform's integrity and trustworthiness. Ultimately, this would expand the system's utility by building a trusted foundation for all credentials issued on the network.
- Multisig Wallet Authorization for Universities: A potential future enhancement is to implement multisignature wallet authorization for universities. This would enhance the security and integrity of the credentialing process by requiring that critical actions, such as issuing or revoking a certificate, be signed by multiple authorized wallets within the institution. This protocol is designed to protect against malicious actors by ensuring that no single individual can unilaterally issue or revoke a certificate, thereby safeguarding the trustworthiness and validity of the credentials on the platform.
- Decentralized Identity (DID) and Verifiable Credentials: While Zero-Knowledge Proofs are used for privacy-preserving claims, a future enhancement could involve integrating full W3C-compliant decentralized identity frameworks, such as DID-based credentials. This would provide a more robust and interoperable system for verifiable data, aligning the platform with broader industry standards.
- Bulk Verification for Employers: Currently, employers can only verify individual credentials. A future enhancement could be the development of a bulk verification tool,

allowing third parties to efficiently verify thousands of certificates at once. This would significantly improve the platform's utility for high-volume use cases.

- Multi-Blockchain Support: Initially, GenuineGrads is built exclusively on Solana. A future enhancement could involve exploring and implementing multi-blockchain support and cross-chain compatibility. This would expand the system's reach and flexibility by allowing it to operate on or interact with other blockchain networks.

Appendix A: Glossary

1. API

- Application Programming Interface
- A set of rules and tools that allows different software applications to communicate with each other. It defines the methods and data formats that applications can use to interact with each other.

2. cNFT

- Compressed Non-Fungible Token
- A type of NFT on the Solana blockchain that significantly reduces storage costs by using a Merkle tree to store token metadata off-chain.

3. GDPR

- General Data Protection Regulation
- A European Union law on data protection and privacy for all individuals within the EU and the European Economic Area.

4. GraphQL

- Graph Query Language
- A query language for APIs that allows clients to request exactly the data they need, nothing more.

5. HTTP

- Hypertext Transfer Protocol
- The foundational protocol for data communication on the World Wide Web, used to transmit web pages and other resources.

6. IPFS

- Interplanetary File System

- A decentralized protocol for storing and sharing files and data in a distributed peer-to-peer network.
7. **NFT**
- Non-Fungible Token
 - A unique and indivisible unit of data stored on a blockchain, used to represent ownership of a digital or physical item.
8. **PDA**
- Program Derived Address
 - A unique address on the Solana blockchain that is owned and controlled by a program, rather than a private key.
9. **PII**
- Personally Identifiable Information
 - Any data that can be used to identify, contact, or locate a specific individual, either directly or indirectly.
10. **RLS**
- Row-Level Security
 - A database feature that restricts access to individual rows of data in a table based on the user's role or other criteria.
11. **RPC**
- Remote Procedure Call
 - A protocol that allows a program to request a service from a program on another computer without needing to understand the network's details.
12. **SAS**
- Solana Attestation Service
 - A protocol that allows trusted issuers, such as government bodies, to create verifiable, on-chain credentials about off-chain data.

13. ZKP

- Zero-Knowledge Proof
- A cryptographic method where one party can prove to another that a statement is true, without revealing any information beyond the validity of the statement itself.

Appendix B: Analysis Models

1. Use case diagrams

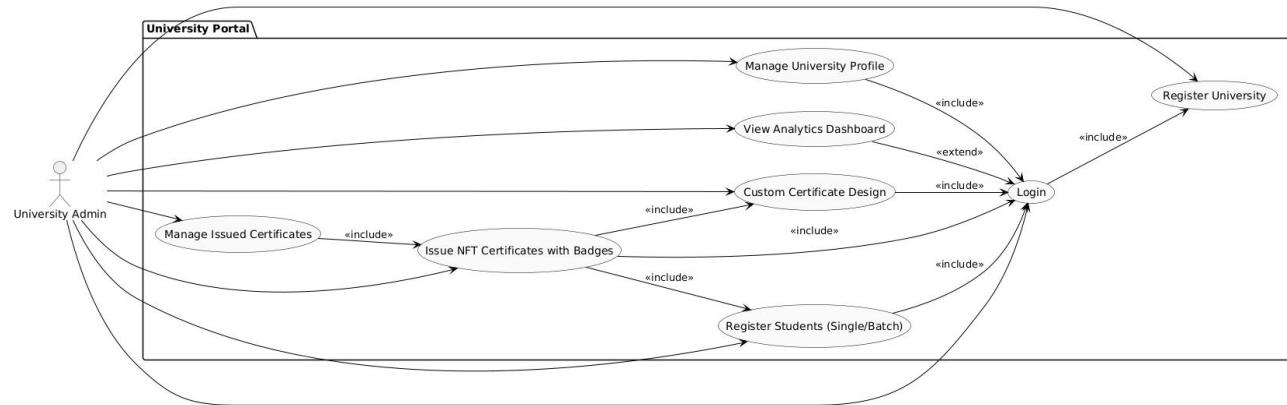


Figure 17. University Admin Use Case Diagram

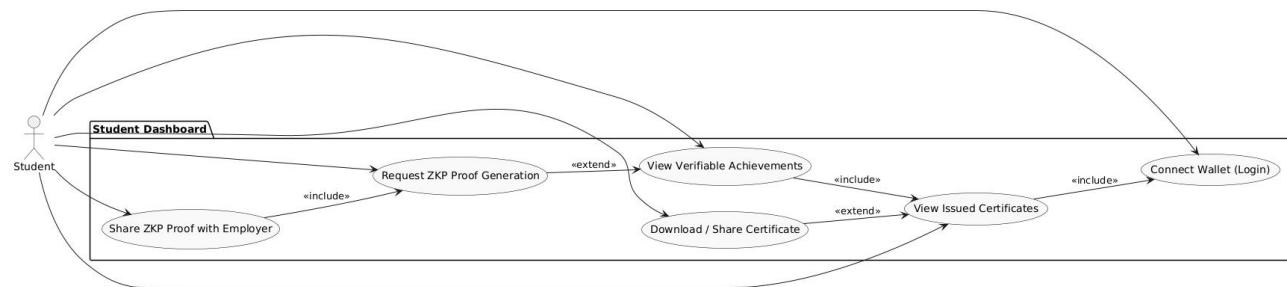


Figure 18. Student Use Case Diagram

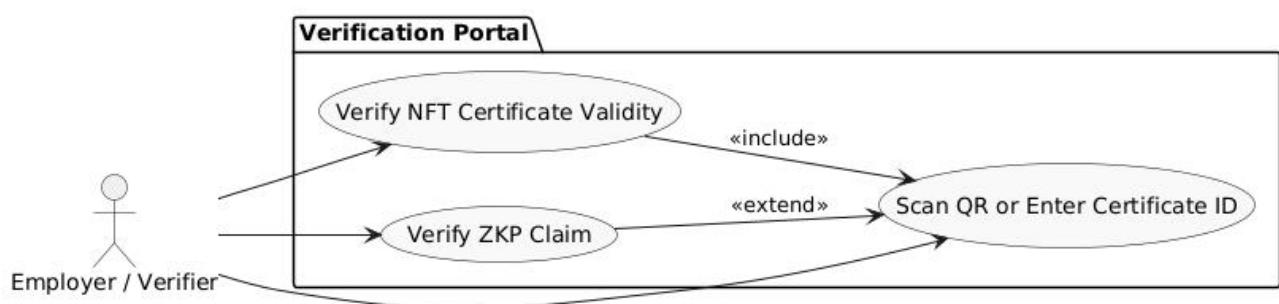


Figure 19: Employer / Verifier Use Case Diagram

2. Activity diagrams

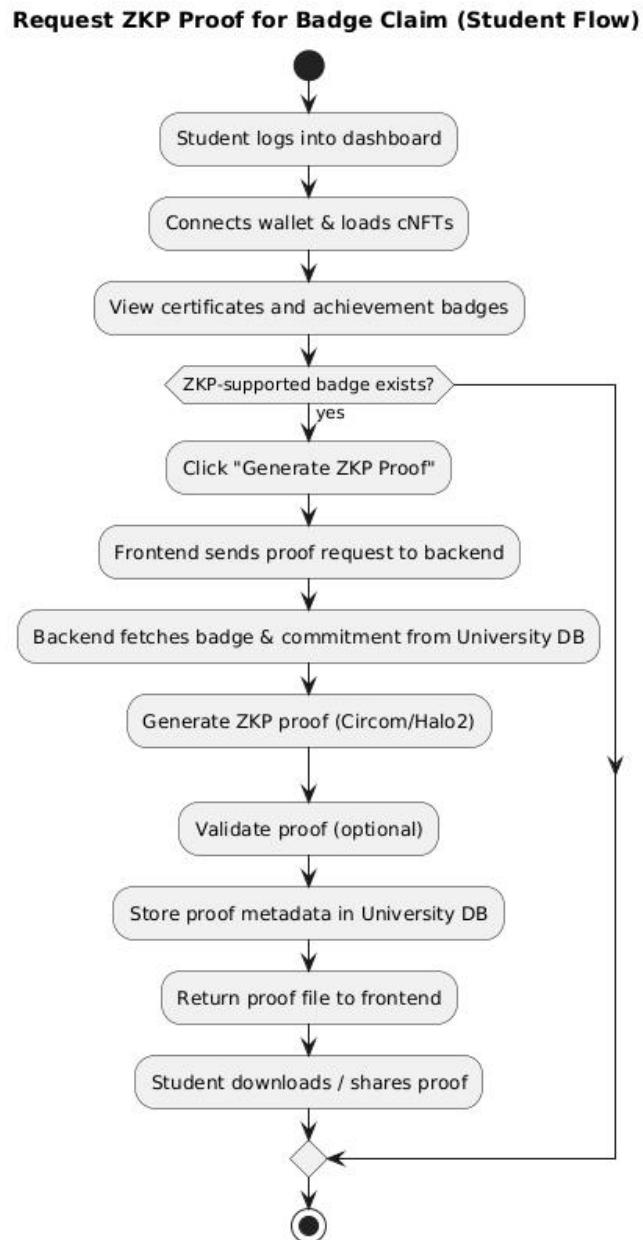


Figure 20. Request ZKP Proof for Badge Claim Activity Diagram

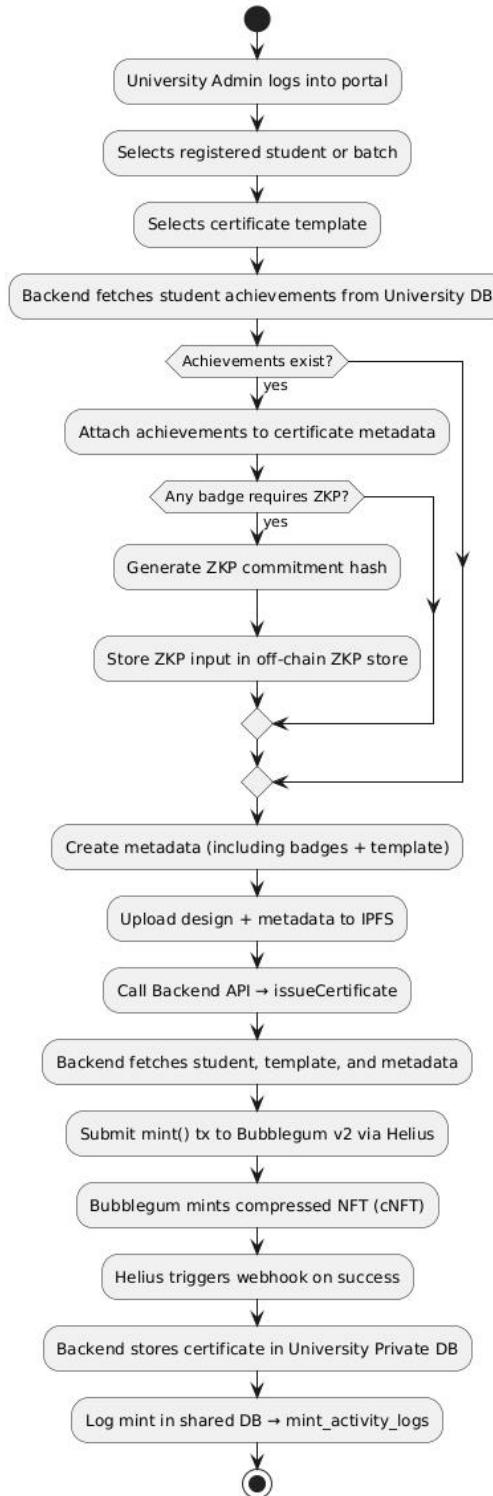
Issue NFT Certificate with Achievements


Figure 21. Issue NFT Certificate with Achievements Activity Diagram

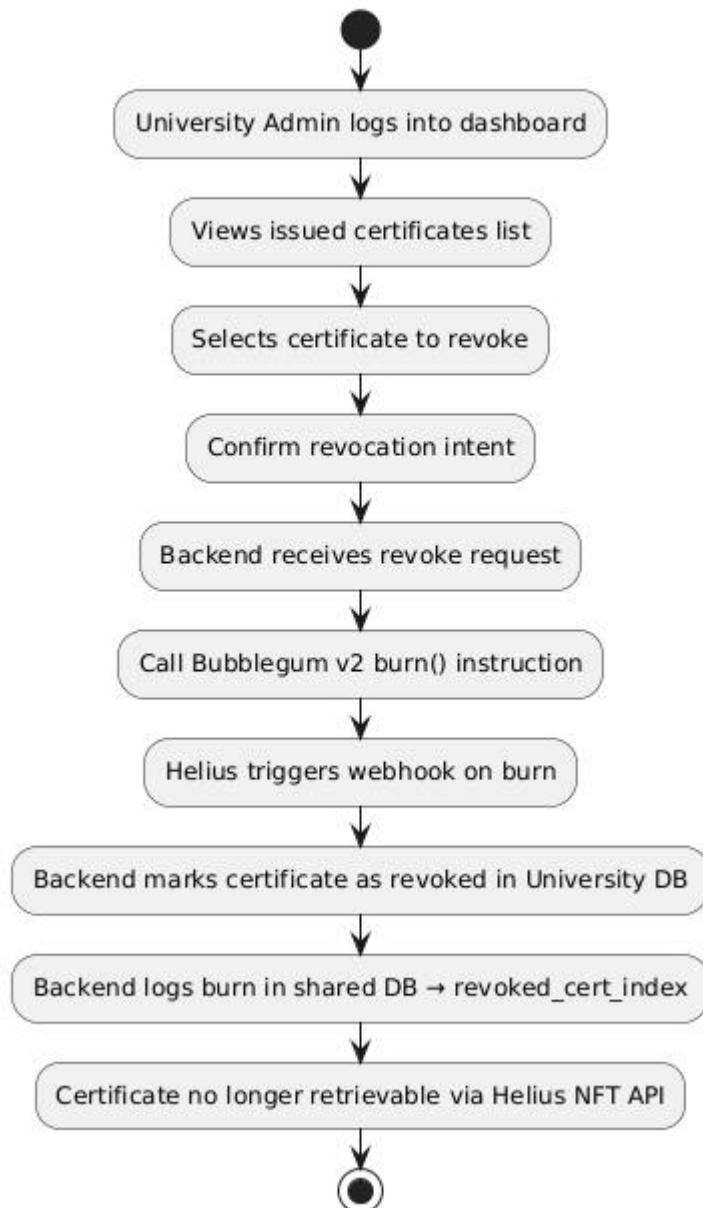
Revoke NFT Certificate by Burning (Admin Flow)

Figure 22. Revoke NFT Certificate by Burning Activity Diagram

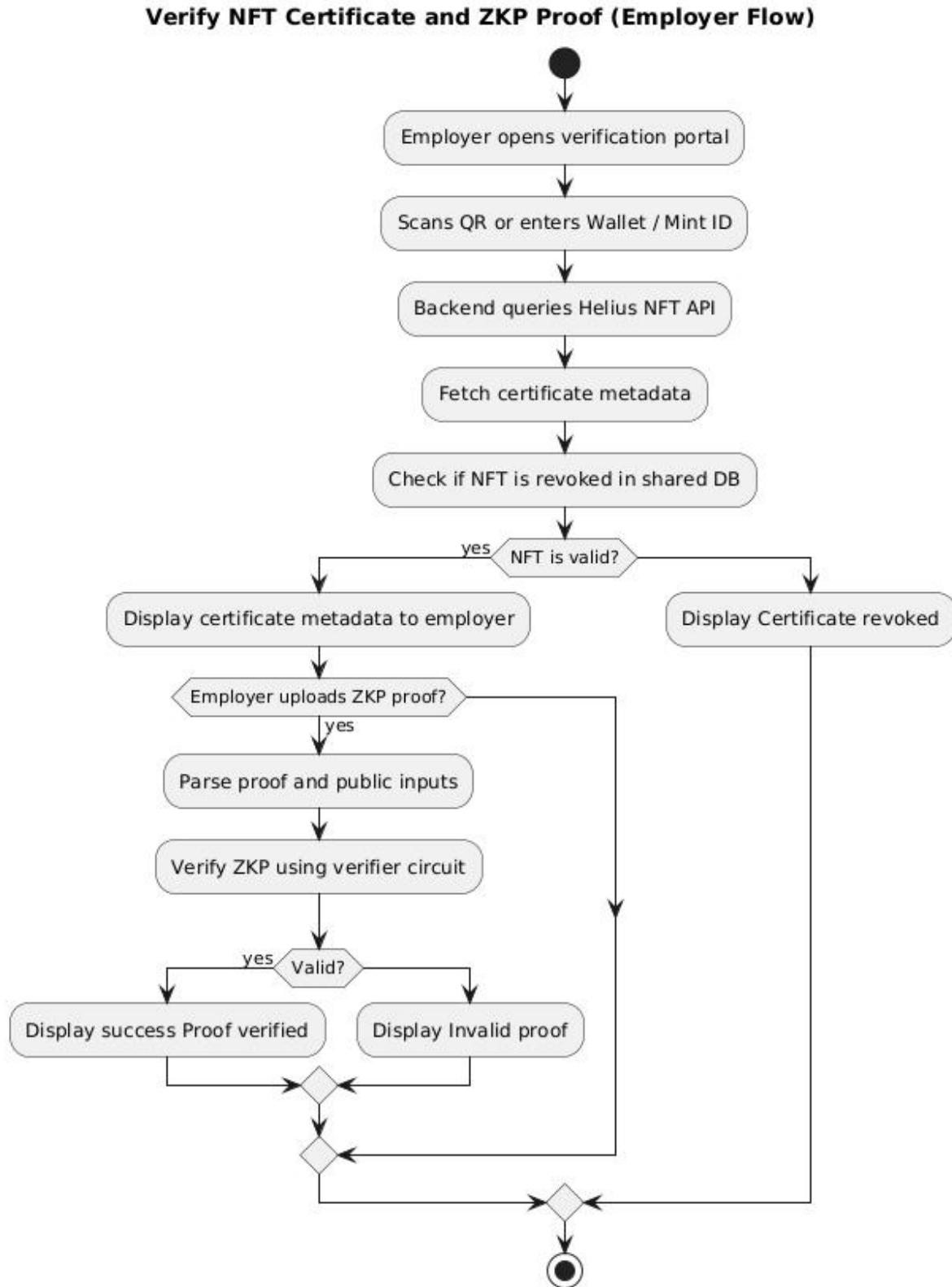


Figure 23. Verify NFT Certificate and ZKP Proof Activity Diagram

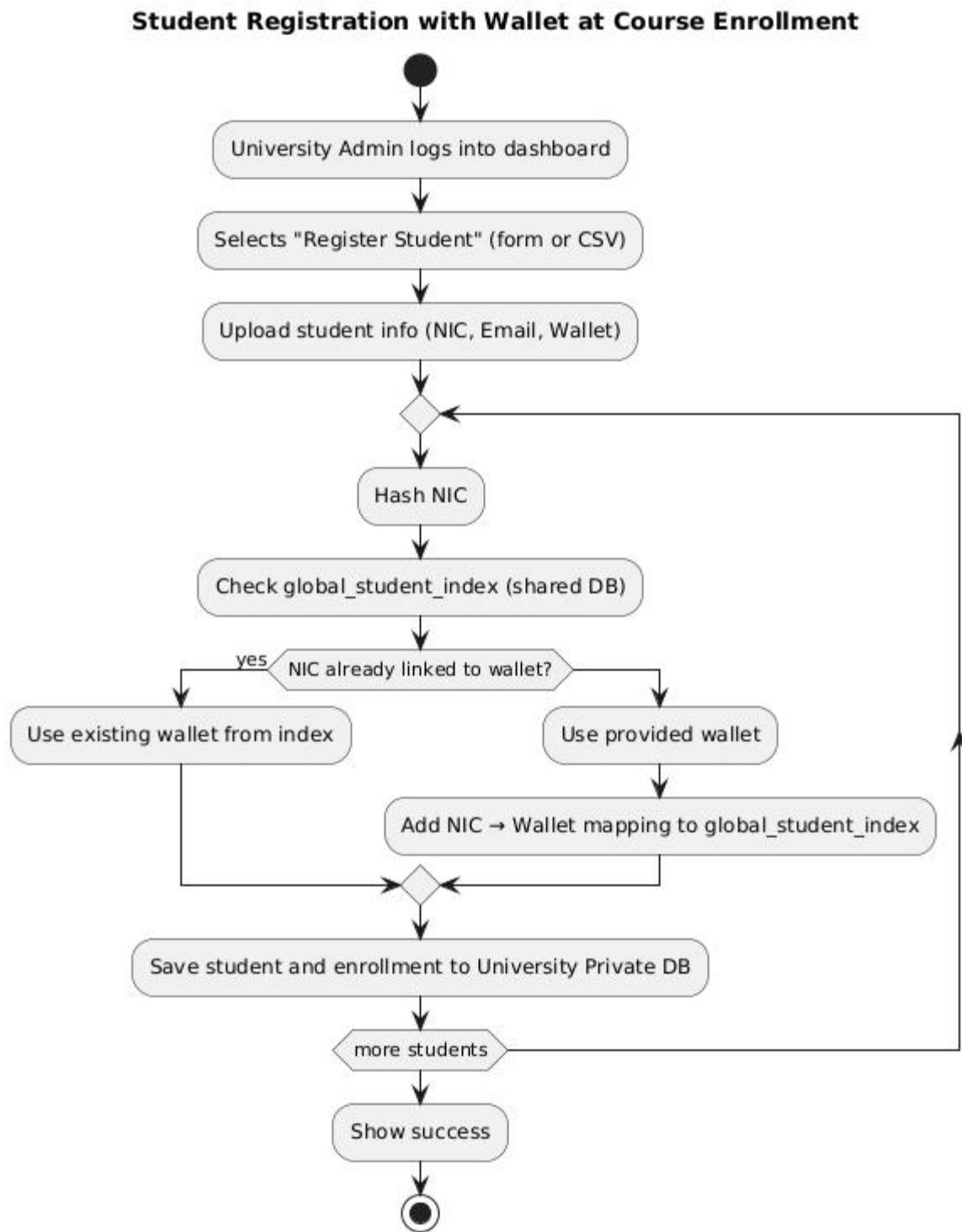


Figure 24. Student Registration with Wallet Activity Diagram

3. Class diagrams

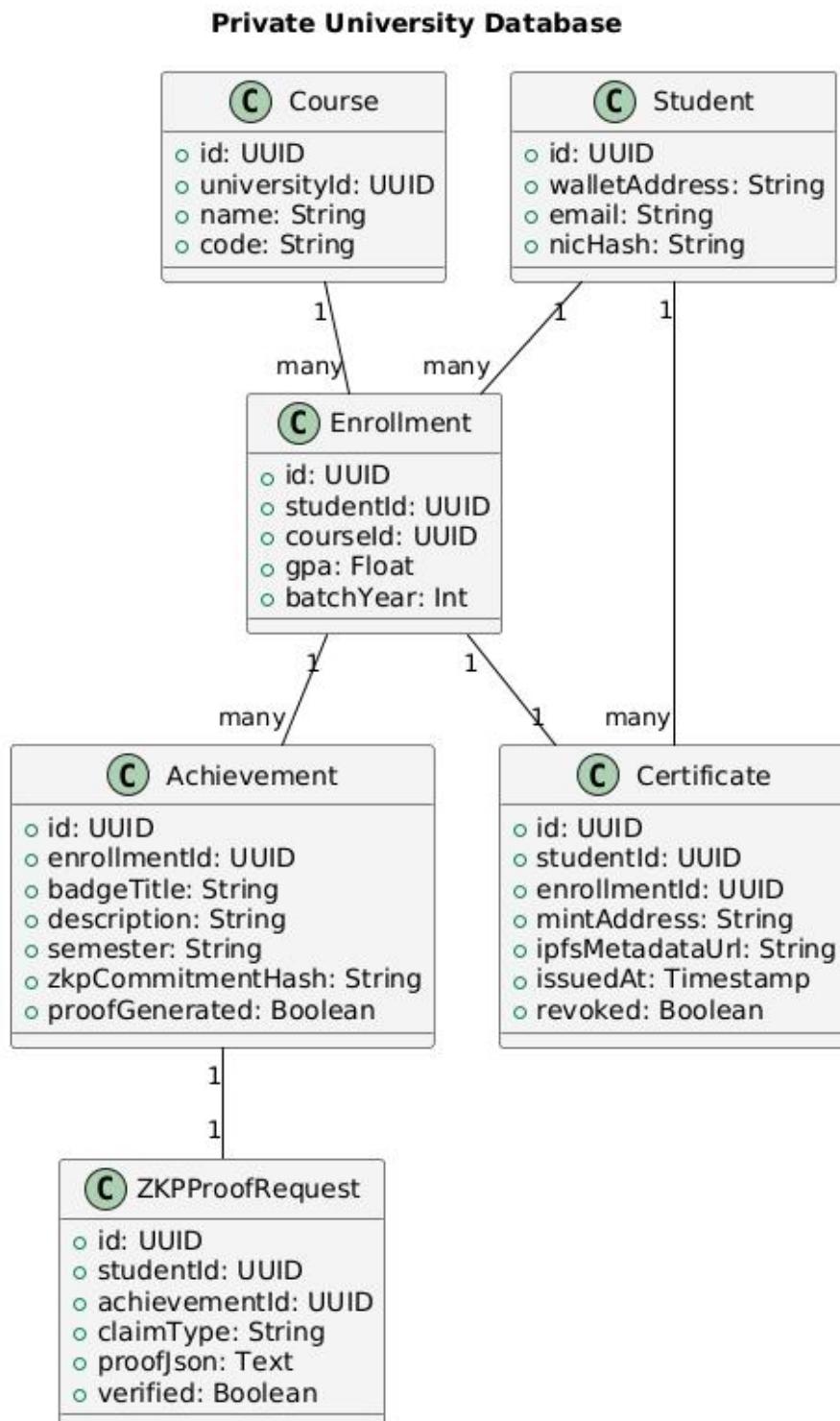


Figure 25. Private University Database Class Diagram

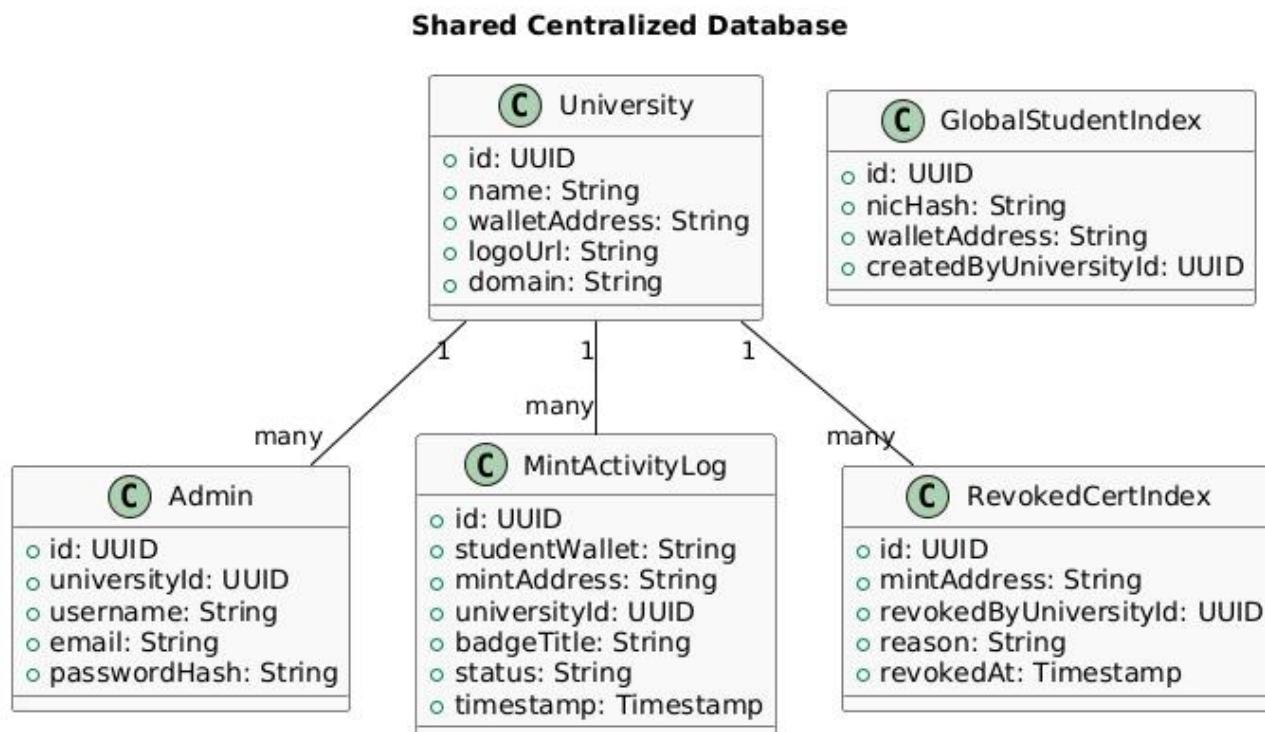


Figure 26. Shared Centralized Database Class Diagram

4. Sequence Diagrams

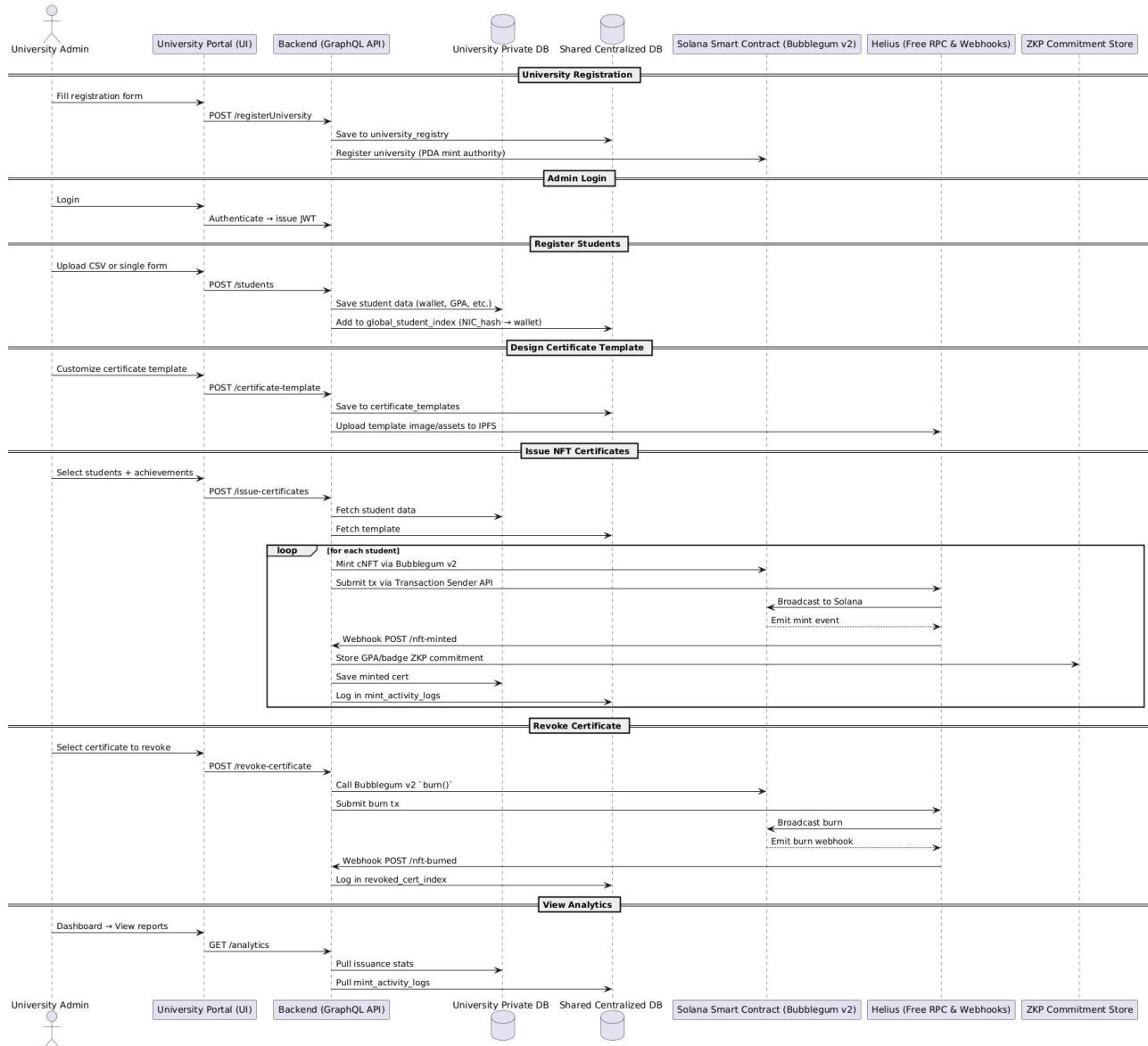


Figure 27. University Sequence Diagram

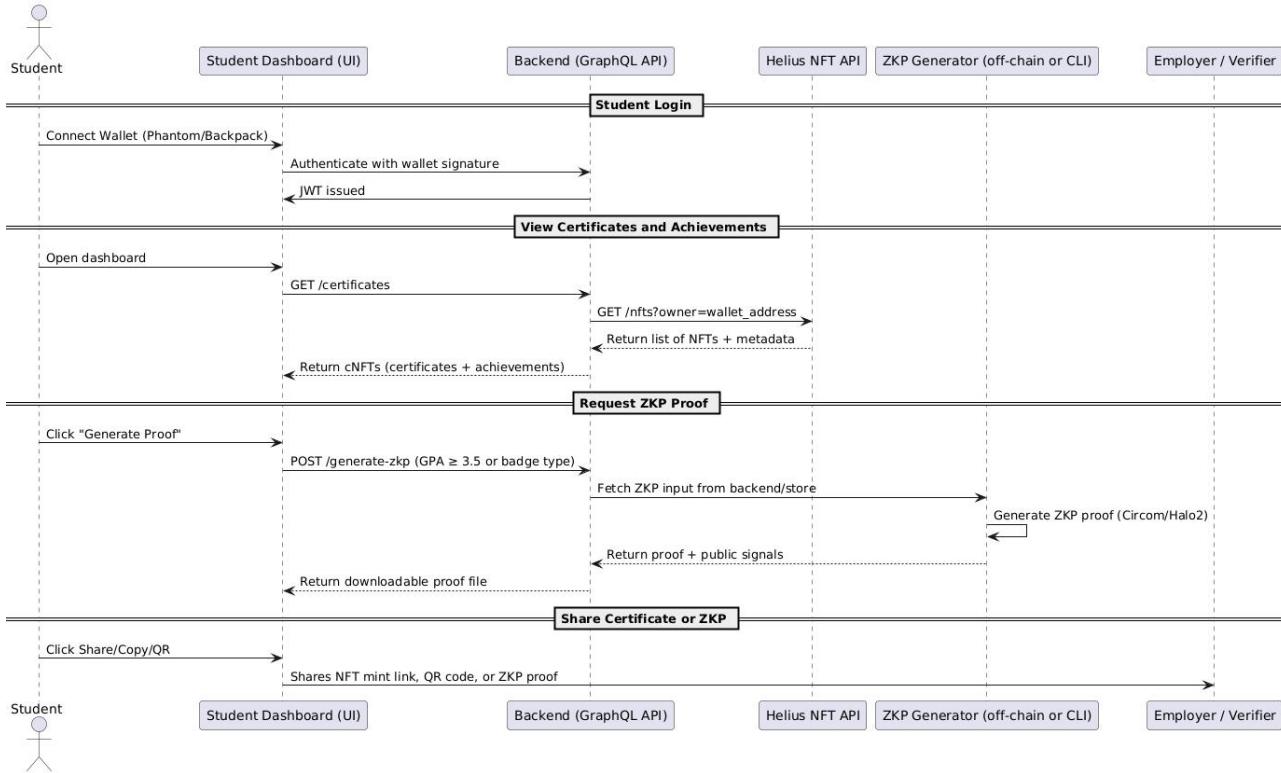


Figure 28. Student Sequence Diagram

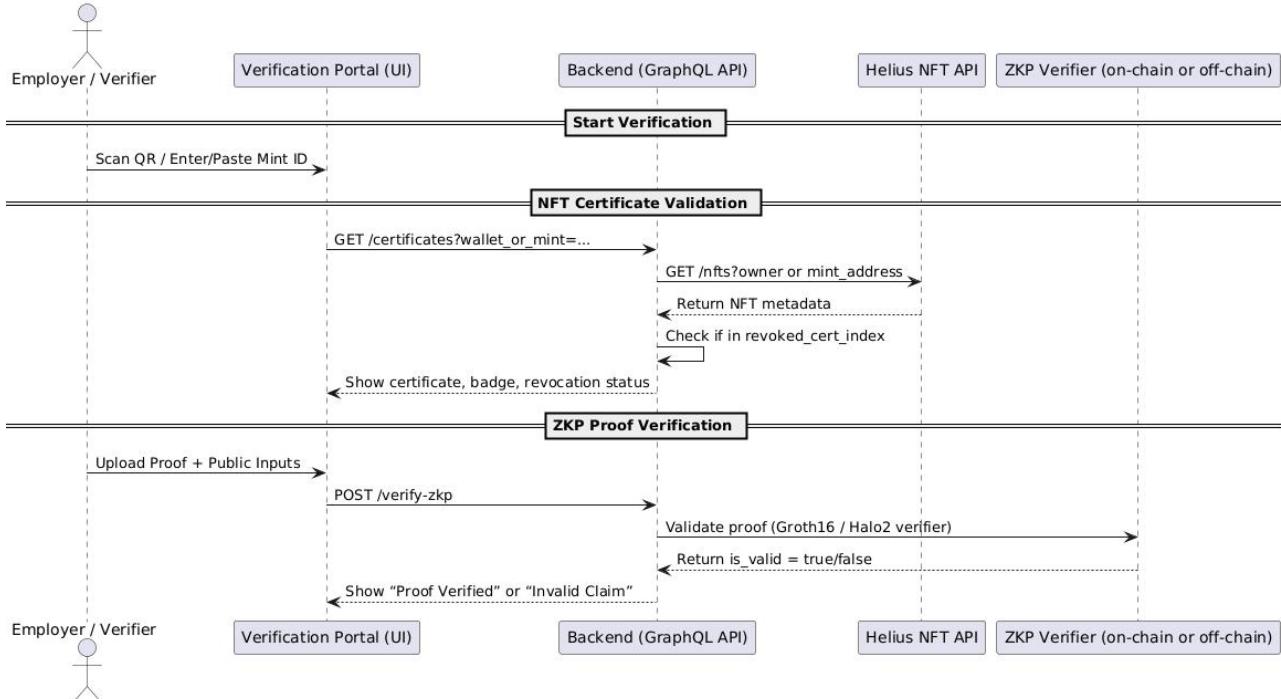


Figure 29. Employer Sequence Diagram

5. State diagram

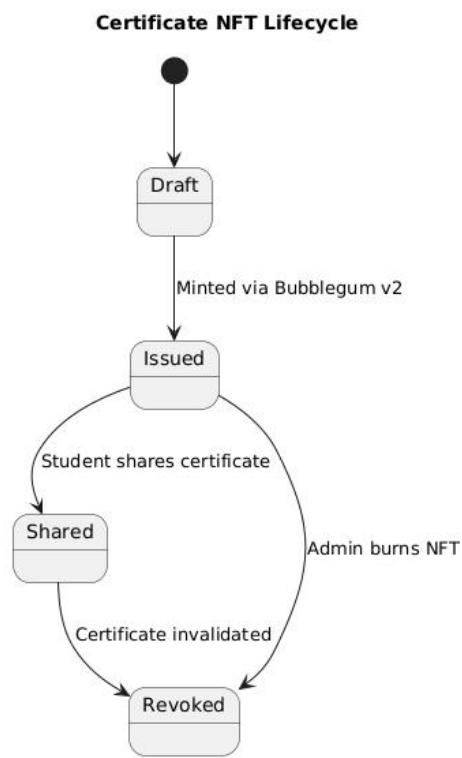


Figure 30. Certificate NFT Lifecycle State Diagram