TECHNICAL UNIVERSITY OF MOLDOVA
FACULTY OF COMPUTERS, INFORMATICS, AND MICROELECTRONICS
DEPARTMENT OF SOFTWARE ENGINEERING AND AUTOMATION
SOFTWARE ENGINEERING STUDY PROGRAMME

INTERNSHIP REPORT

BlockSign - Securing Documents With Blockchain

Team No. 1

Team members:

Student 1: Alexei Pavlovschii, FAF-231 _____

Student 2: Alexandru Bujor, FAF-231 _____

Student 3: Gabriel Moraru, FAF-232 _____

Student 4: Filip Obrijan, FAF-232 _____

Student 5: Vladimir Vitcovschii, FAF-231 _____

Mentor: Andrei Poștaru _____

Submission date: 01/10/2025

Chișinău, 2025

ABSTRACT

The project titled **"BlockSign - Securing Documents With Blockchain"** was developed by students Alexei Pavlovschii, Alexandru Bujor, Gabriel Moraru, Filip Obrijan, and Vladimir Vitcovschii from the Technical University of Moldova.

This project comprises 5 chapters: Problem Framing, Domain Analysis, Solution Proposal, System Design and Practical Implementation, as well as Introduction, Conclusions and Bibliography.

It addresses the growing need for secure, transparent, and verifiable document management systems in modern organizations. Traditional approaches often rely on centralized storage and password-based authentication, which are vulnerable to tampering, unauthorized access, and human error. Our solution proposes a **blockchain-anchored mechanism** for document signing and verification, ensuring both the integrity and authenticity of documents.

The system implements a **passwordless authentication** scheme based on cryptographic key pairs, reducing risks associated with compromised credentials. Document integrity is preserved by calculating a SHA-256 hash, while signatures from all involved participants are verified using the **Ed25519 algorithm**. Once finalized, a document's hash can be anchored to a blockchain network, providing immutable proof of its existence and content at a given time.

From an architectural perspective, the back-end is developed in Node.js with Express.js and Prisma ORM on top of a PostgreSQL database. Email-based workflows, including OTP verification and document notifications, ensure usability and trust between system participants. The modular design of the application allows for future integration with cloud storage providers and expanded blockchain support.

Through this project, the students combined concepts from distributed systems, **cryptography**, and software engineering to design and implement a practical, real-world solution. The outcome is a **secure and scalable platform** that demonstrates how blockchain technology can enhance document authentication and verification processes in both academic and enterprise contexts.

Keywords: Blockchain, Cryptography, Documents, Security.

Content

INTRODUCTION **5**

1 PROBLEM FRAMING **6**

1.1 Problem Description 6

1.2 Problem Statement 9

2 Domain Analysis **10**

2.1 Target Audience 10

2.1.1 Individual Users 10

2.1.2 Educational Institutions 10

2.1.3 Small and Medium Enterprises (SMEs) 11

2.1.4 Government Institutions 11

2.1.5 Law Firms and Notary Services 11

2.1.6 Financial and Banking Sectors 12

2.1.7 Healthcare Organizations 12

2.2 Market Size and Growth 12

2.2.1 Key Competitors and Solutions 13

2.2.2 Trends and Opportunities 14

2.2.3 Challenges and Barriers 15

2.3 Technical Research 15

2.3.1 Blockchain-Based Notarization 16

2.3.2 Electronic Signatures with Multi-Factor Authentication 17

2.3.3 Electronic & Remote Notarization 17

2.3.4 Advanced Cryptographic Techniques 18

2.3.5 Hybrid and Scalable Architectures 19

3 Solution Proposal **20**

4 System Design **22**

4.1 Technical Requirements 22

4.1.1 Functional Requirements 22

4.1.2 Non-Functional Requirements 24

4.2 Behavioral Modeling 25

4.2.1 Use Case Diagrams 25

4.2.2 Sequence Diagrams 28

4.3 Structural Modeling 31

4.3.1 Class Diagram . . . 31
4.4 Figma User Interface Mockups . . . 33
5 Practical Implementation . . . 44
5.1 Back-end architecture . . . 44
5.1.1 Application Layer . . . 44
5.1.2 Database Layer . . . 44
5.1.3 Security Measures . . . 45
5.1.4 Authentication and Tokens . . . 45
5.1.5 Email Service . . . 46
5.1.6 Cryptographic Module . . . 46
5.1.7 Document Workflow . . . 47
5.1.8 Extensibility . . . 47
5.2 Front-end UI . . . 48
5.2.1 Frontend Architecture . . . 48
5.2.2 Landing Page and User Onboarding . . . 49
5.2.3 Registration Flow and User Identity . . . 50
5.2.4 Authentication System and Security Features . . . 52
5.2.5 User role specified UI . . . 53
5.2.6 Additional registration steps . . . 55
5.2.7 Document Management Interface . . . 58
5.2.8 Responsive Design and User Experience . . . 60
5.2.9 Client-Side Security Implementation . . . 60
5.2.10 Internationalization and Localization . . . 61
5.2.11 Feature Accessibility by User Role . . . 61
5.2.12 Technology Integration and Performance . . . 62
CONCLUSIONS . . . 63

INTRODUCTION

In the digital era, organizations and individuals are increasingly reliant on electronic documents for communication, collaboration, and decision-making. The authenticity, integrity, and accessibility of these documents have therefore become critical factors in building trust and ensuring compliance with regulatory standards. Traditional paper-based approaches to document management are costly, time-consuming, and vulnerable to fraud or human error. At the same time, centralized digital systems often face challenges related to security, data breaches, and limited transparency.

This project addresses these challenges by designing and implementing a secure, scalable, and user-friendly system for document registration, signing, and verification. The system relies on public-key cryptography (Ed25519) to provide a passwordless authentication mechanism that ensures only authorized users can access or sign documents. A carefully designed workflow guides users through registration and approval, while administrators maintain oversight and control of pending regstration requests.

The central contribution of this work is the integration of digital signatures and cryptographic payloads into the document lifecycle. Each document is uniquely identified by its SHA-256 hash, which guarantees integrity and allows participants to independently verify that the file they review is identical to the file stored in the system. By requiring each participant to sign the canonical payload, the system ensures non-repudiation and establishes a verifiable chain of consent. Once all signatures are collected, the document status transitions to signed, and participants are notified accordingly.

In addition to the cryptographic foundation, the system emphasizes practical usability. Users interact with the platform through intuitive registration and login flows, email-based verification, and role-based access controls. Administrators can approve or reject registration requests, while users can create documents, tag participants by username, and track their signature status. For future development, the platform is designed to integrate with decentralized storage or blockchain anchoring, extending its guarantees of persistence and transparency.

Overall, this project demonstrates how modern cryptographic primitives and structured workflows can be combined to deliver a robust document signing and verification solution. It lays the foundation for further research into decentralized trust, compliance with international standards, and seamless integration with existing enterprise systems.

1 PROBLEM FRAMING

1.1 Problem Description

Securing documents—whether physical or digital—remains a persistent, multi-dimensional problem that affects governments, enterprises, and individuals.

Physical Documents: Physical records (e.g., passports, certificates, contracts) have long been protected via signatures, seals, stamps, and locked storage, yet they remain vulnerable to forgery, misuse, and destruction (Figure 1.1). Law-enforcement bodies warn that advances in consumer printing and imaging have lowered the barrier to document counterfeiting **interpol˙identity˙nodate**. Border and customs activity shows the problem is active and measurable: in one U.S. port alone, officials intercepted more than 6,800 fraudulent or stolen documents in FY2023—a 219% year-over-year increase—demonstrating both scale and growth **us˙customs˙and˙border˙protection˙cincinnati˙nodate**. In addition to fraud, disasters can irreparably damage archives; professional guidance exists precisely because fires, floods, and collapses have destroyed unique holdings (e.g., the 2009 Cologne City Archive collapse that obliterated a major European archive) **international˙council˙on˙archives˙emergency˙nodate**, **geoengineerorg˙cologne˙nodate**.

Figure 1.1 - Signature Forgery Impact on Important Processes

Digital Documents: Digitization improved accessibility and scale but introduced remote theft, silent manipulation, mass leakage, and extortion (Figure 1.2). The scale is well documented: Verizon's 2024 DBIR analyzed 30,458 incidents with 10,626 confirmed breaches, noting increased vulnerability exploitation via web applications **verizon˙2024˙nodate**. Financial impact remains high: IBM's 2025 study reports a global average breach cost of USD 4.4M (down from USD 4.88M in 2024, still historically elevated), underscoring that document exposure is expensive even when containment improves **ibm˙cost˙nodate**,

ibm˙cost˙nodate-1. Ransomware and data-theft extortion continue to pressure organizations across critical sectors, with complaints to the FBI's IC3 rising 9% in 2024 and losses hitting $16.6B across cyber and scam crimes **reuters˙˙fbi˙complaints˙nodate**.

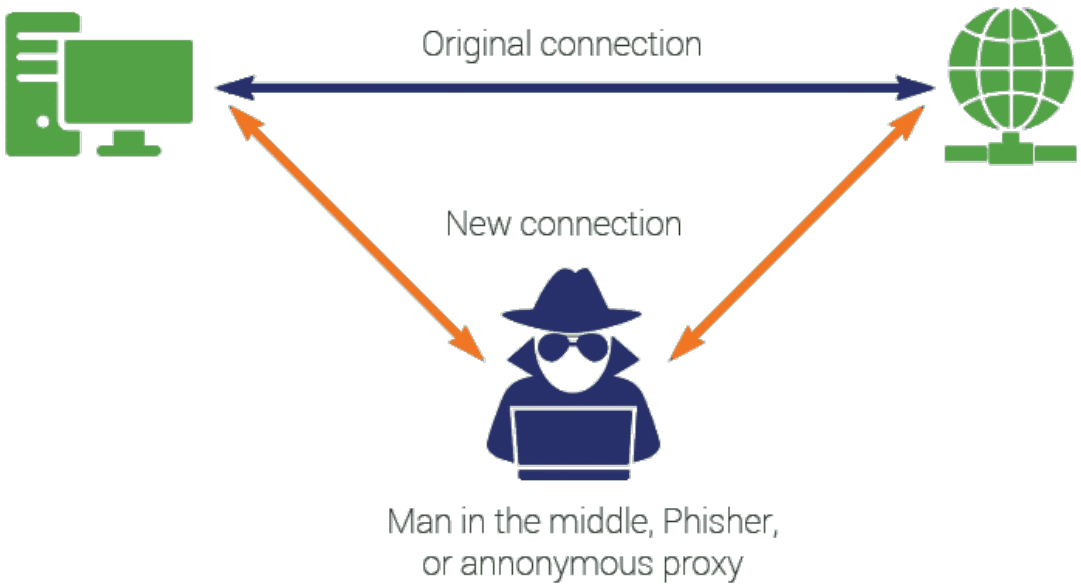

Figure 1.2 - Man In The Middle Attack

Trust and authenticity weaknesses in current ecosystems: Even when digital signatures and PKI are used, centralized trust anchors can fail. The DigiNotar breach (2011) led to hundreds of fraudulent certificates for prominent domains, breaking the authenticity guarantees expected from TLS/PKI and forcing browser vendors to distrust the CA entirely—an instructive "single point of failure" for trust **wired˙diginotar˙nodate**. The episode shows that document authenticity that depends on a compromised authority can be globally undermined.

Human and process error remain a major source of breaches: Beyond external attackers, misconfigurations, errors, and insider misuse contribute materially to breaches. Verizon's 2024 DBIR details the role of internal actors and errors across sectors; in healthcare, for example, internal actors feature far more prominently than elsewhere, reversing earlier trends **hipaa˙journal˙verizon˙nodate**. This reflects a broad, persistent problem: authorized access used improperly can compromise sensitive documents at scale.

Regulatory pressure - confidentiality, integrity, accountability: Regulatory frameworks require clear safeguards for documents that include personal data. The GDPR mandates "appropriate technical and organizational measures" and emphasizes data integrity and accountability **eur-lex˙gdpr˙nodate**. In the EU, eIDAS sets legal scaffolding for electronic identification and trust services, defining requirements for

trustworthy digital interactions (e.g., signatures, seals, timestamps) **eur-lex˙eidas˙nodate**. Non-compliance introduces legal and financial risks on top of the technical ones.

Common vulnerability patterns in applications that handle documents: Applications that store, view, transfer, or sign documents routinely exhibit high-impact weaknesses. The OWASP Top 10 highlights recurring problems such as Broken Access Control (ranked #1 in 2021), Cryptographic Failures, Injection, and Security Misconfiguration (Figure 1.3); notably, 94% of tested apps exhibited some form of access control weakness in the dataset behind the 2021 list **owasp˙top˙nodate**, **owasp˙a01˙nodate**. These patterns map directly to risks for document confidentiality, integrity, and availability.

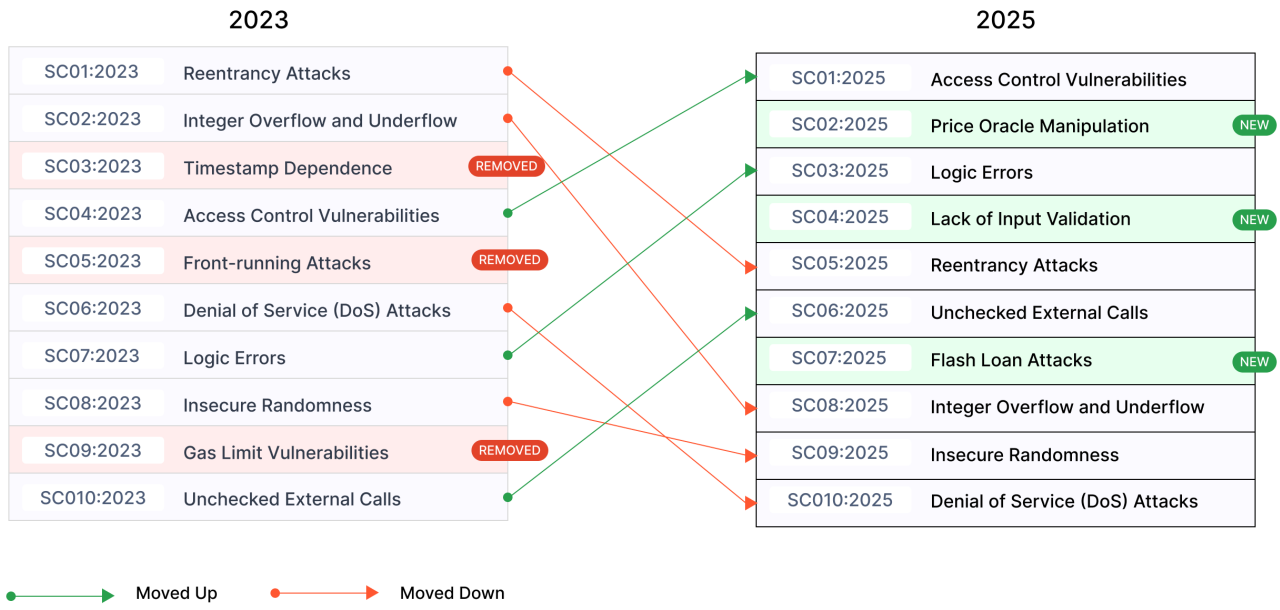

Figure 1.3 - OWASP Top 10 (2023 vs 2025)

The Core Problem: Taken together, these issues demonstrate that document security is a multi-faceted challenge:

- Physical documents face forgery and catastrophic loss risks despite traditional controls **interpol˙identity˙nod us˙customs˙and˙border˙protection˙cincinnati˙nodate**, **international˙council˙on˙archives˙emergency˙n geoengineerorg˙cologne˙nodate**.
- Digital documents face breach, extortion, and manipulation at global scale, with material financial impact **verizon˙2024˙nodate**, **ibm˙cost˙nodate**, **ibm˙cost˙nodate-1**, **reuters˙˙fbi˙complaints˙nodate**.
- Trust infrastructures (e.g., CAs) can become single points of failure **wired˙diginotar˙nodate**.
- Human/organizational errors and app-level vulnerabilities remain prevalent **hipaa˙journal˙verizon˙nodate**, **owasp˙top˙nodate**, **owasp˙a01˙nodate**.
- Regulatory frameworks demand provable safeguards and accountability **eur-lex˙gdpr˙nodate**, **eur-lex˙eidas**

These facts collectively demonstrate that ensuring authenticity, integrity, confidentiality, availability, and accountability for documents is an unresolved, real-world problem spanning both physical and digital realms.

1.2 Problem Statement

Existing mechanisms for securing physical and digital documents remain vulnerable to forgery, loss, unauthorized access, manipulation, and systemic trust failures, while regulatory obligations require stronger assurance and accountability. The problem is to guarantee authenticity, integrity, confidentiality, availability, and verifiability of documents over time and across domains, despite evolving threats, human error, and infrastructural weaknesses (Figure 1.4).

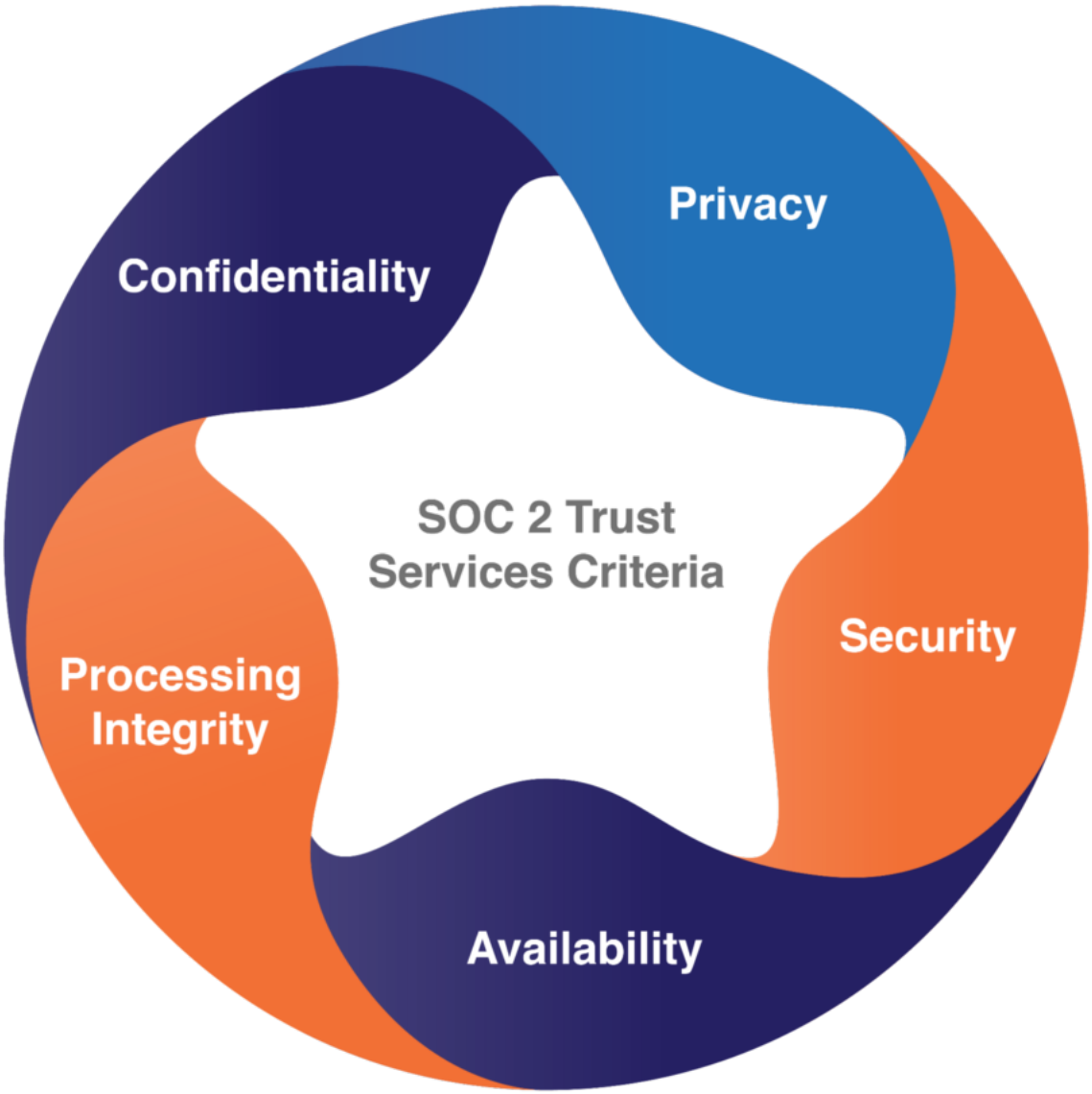

Figure 1.4 - Service Organization Controls (SOC) security principles standard

2 Domain Analysis

2.1 Target Audience

BlockSign is designed to serve a diverse range of users who require secure, verifiable, and legally recognized methods of document handling. By leveraging blockchain technology, BlockSign ensures authenticity, immutability, and compliance across various sectors. The primary audiences are described in this section.

2.1.1 Individual Users

In an era where personal data misuse and identity theft are prevalent, individuals can utilize BlockSign to notarize personal documents such as wills, rental agreements, and contracts. This ensures their documents remain immutable and verifiable over time, providing peace of mind and legal assurance **world˙economic˙fo**

2.1.2 Educational Institutions

Educational institutions require secure systems for managing sensitive records, including diplomas, transcripts, and certificates (Figure 2.1). Cases of diploma fraud have increased globally, undermining student mobility and employer trust. Blockchain notarization of educational credentials ensures their authenticity across borders, which is particularly important in student exchange programs and international hiring **unesco˙credentials˙nodate**.

Figure 2.1 - Digital Certificates

2.1.3 Small and Medium Enterprises (SMEs)

SMEs often lack access to enterprise-grade solutions like DocuSign or Adobe Sign due to cost constraints. BlockSign offers a cost-effective and transparent notarization platform, enabling startups, freelancers, and growing firms to establish trust with partners, clients, and regulators without heavy infrastructure investments **european˙comission˙digital˙nodate**.

2.1.4 Government Institutions

Government agencies responsible for issuing and verifying official records (e.g., passports, permits, licenses) face persistent threats from forgery and manipulation of physical and digital documents (Figure 2.2). These institutions require trustworthy digital notarization solutions to comply with regulations such as the EU's eIDAS Regulation and the General Data Protection Regulation (GDPR) **european˙comission˙eidas˙noda**

Figure 2.2 - Customs and Border Protection officer verifying a travel document

2.1.5 Law Firms and Notary Services

Law firms and notary services handle contracts, affidavits, and property transactions, all of which require high trust and legal enforceability. Traditional notarization can be time-consuming and limited by geographic restrictions. Remote Online Notarization (RON), already gaining traction in the United States and parts of Europe, demonstrates that secure video-based validation combined with blockchain immutability can streamline legal workflows while retaining compliance **ncfa˙canada˙future˙nodate**.

2.1.6 Financial and Banking Sectors

Financial institutions, including banks and insurance companies, must guarantee the validity of signed contracts, loan agreements, and client identities (Figure 2.3). Financial institutions are among the industries with the highest breach costs, averaging USD 4.44 million per incident **accenture˙state˙nodate**.

Figure 2.3 - Personal Loan Agreement

2.1.7 Healthcare Organizations

The healthcare sector is increasingly dependent on electronic records and digital consent forms. The sector faces both internal and external threats: 70% of healthcare breaches involve insiders misusing access privileges **kadir˙canoz˙use˙nodate**. For hospitals, clinics, and research institutions, BlockSign offers a mechanism to notarize patient records, safeguard sensitive data, and ensure compliance with confidentiality obligations under GDPR.

2.2 Market Size and Growth

According to industry reports, the global market for digital signatures was worth about USD 7.47 billion in 2023, and it's expected to soar to around USD 37.79 billion by 2029, expanding at an annual growth rate of 31%.

Other estimates are even more bullish: one study puts the 2023 market at USD 4.6 billion with a forecast to USD 43.5 billion by 2030, representing a 37.9% CAGR between 2023 and 2030 (Figure 2.4) **grand˙view˙horizon˙global˙nodate**. Another source projects growth from USD 3.2 billion in 2021 to

USD 48.4 billion by 2028, at 35.4% CAGR (Figure 2.5) **acumen˙digital˙nodate**. Despite slight variations in numbers, the consensus is clear: the market is booming.

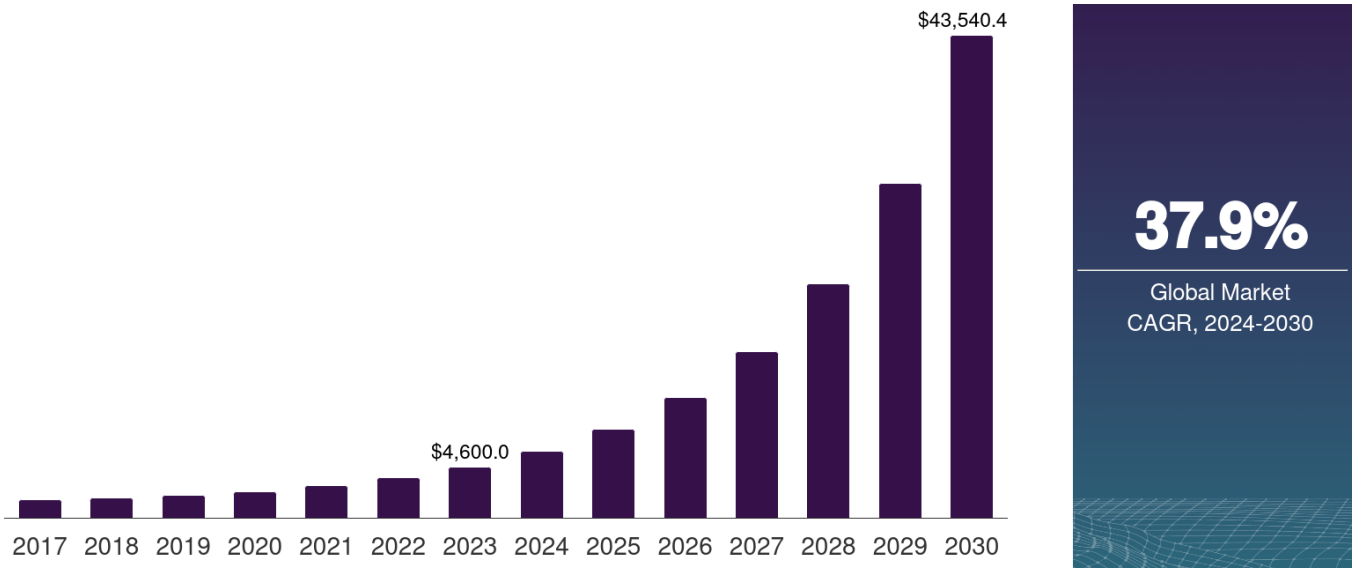

Figure 2.4 - Global Digital Signature Market Size 2017-2030 (Horizon)

Blockchain-related identity services and notarization tools are also on the rise, particularly in industries where trust, compliance, and speed are critical: banking, finance, legal, and healthcare. These solutions help reduce fraud, accelerate cross-border processes, and ensure compliance with regulations like GDPR, eIDAS in Europe, or ESIGN and UETA in the United States.

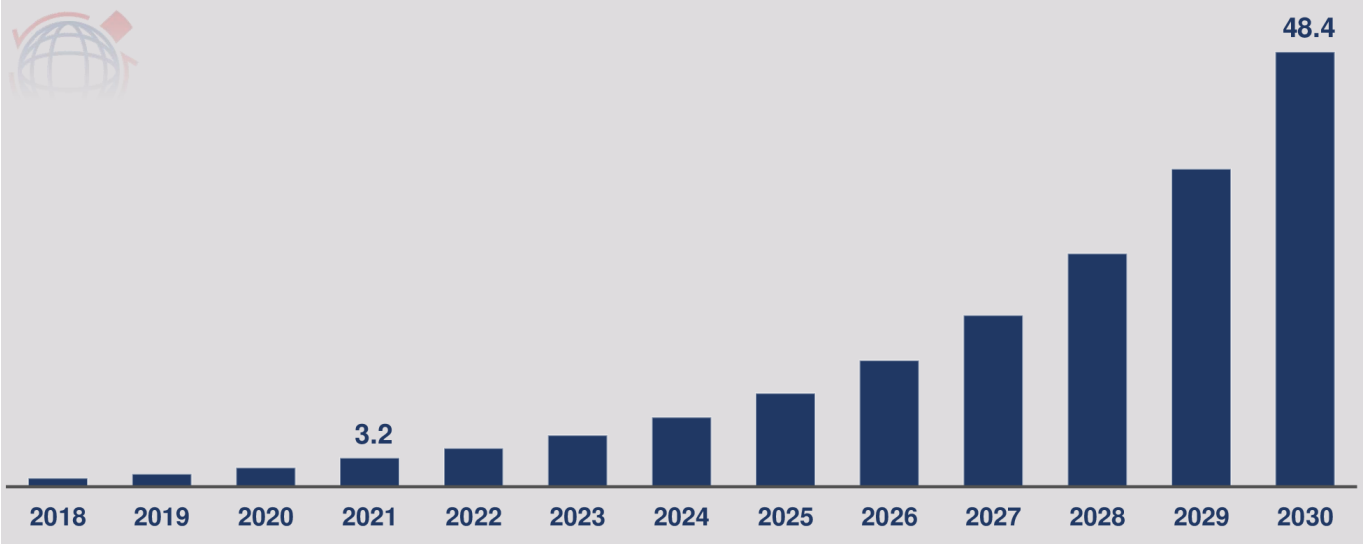

Figure 2.5 - Digital Signature Market 2018-2030 (Acumen)

2.2.1 Key Competitors and Solutions

Several big players are driving the digital signature space:

- **DocuSign** and **Adobe Sign** are trusted and widely used, especially by enterprises, but still rely on

centralized infrastructure.

- **Notarize**, **NotaryCam**, and **SignNow** specialize in Remote Online Notarization (RON), often using video identity checks to meet legal and compliance needs.
- On the blockchain side, platforms like **DoxyChain** and **Blocknotary** offer immutable timestamping and decentralized storage.
- Emerging frameworks like **KILT Protocol** and **Concordium** support decentralized identity and can integrate into notarization workflows.

These players show how mature the e-signature market is becoming—and how blockchain is carving out its place as a serious value-add technology (Figure 2.6) **electro˙iq˙e-signature˙nodate**.

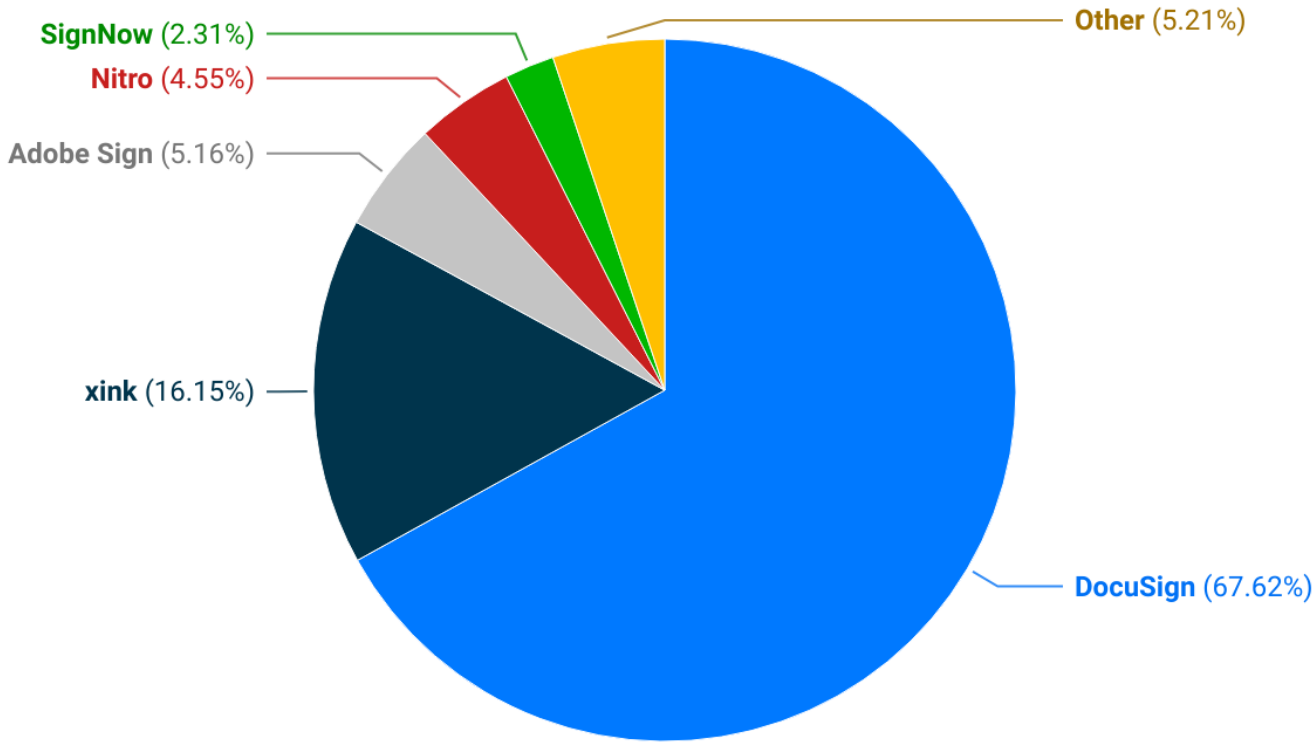

Figure 2.6 - Market Share of Leading Electronic Signature Companies

2.2.2 Trends and Opportunities

A few clear trends are shaping the future of digital signatures:

1. **Trustless, decentralized verification** – Blockchain-based notarization helps remove single points of failure tied to central Certificate Authorities (CAs).
2. **Remote and cross-border notarization** – With more people working globally, demand increases for legally recognized remote notarization systems.
3. **AI and automation integration** – Automated document checks, fraud detection, and smart contract workflows are on the rise.
4. **Quantum-safe cryptography** – Some platforms are starting to prepare for future cryptographic threats

to ensure long-term document integrity.

5. **Hybrid architectures** – Combining public and permissioned blockchains helps balance trust, scalability, and cost-efficiency.

These trends create fertile ground for innovation, especially for solutions that balance security, legal compliance, and user-friendliness (Figure 2.7).

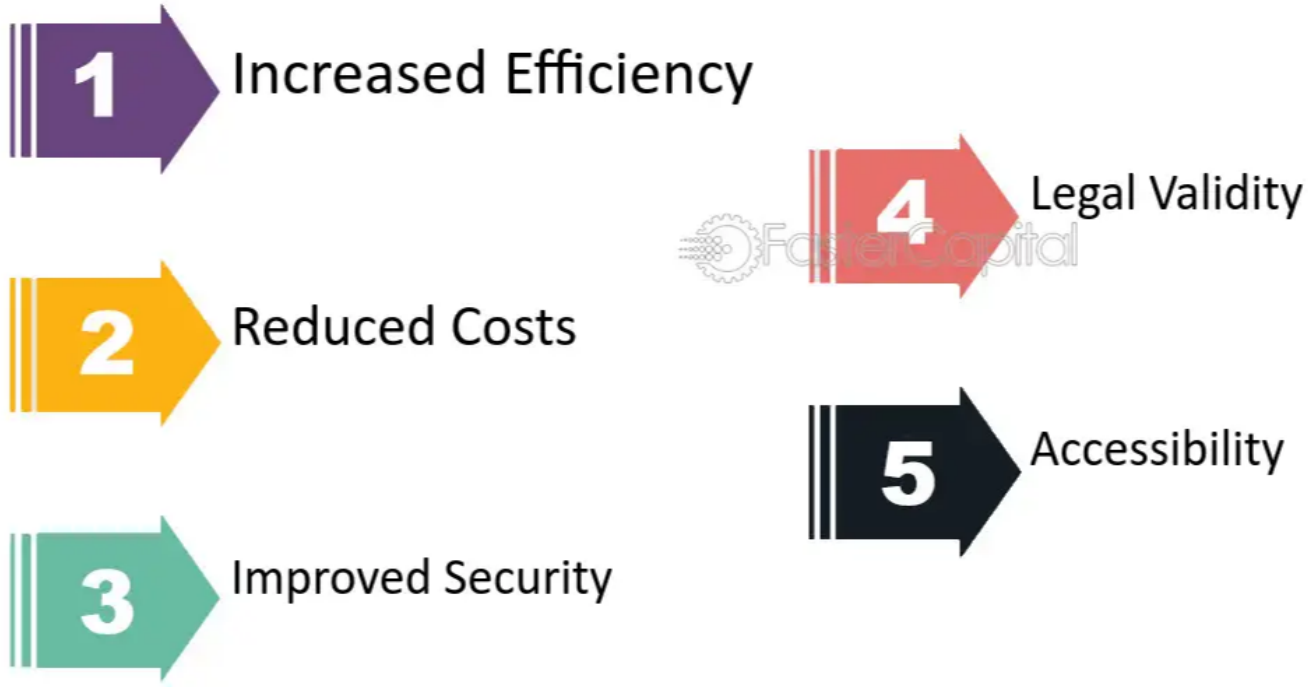

Figure 2.7 - Benefits of Digital Signatures

2.2.3 Challenges and Barriers

However, adoption still faces some challenges:

- Legal frameworks vary widely across countries, slowing global standardization and interoperability.
- Many institutions remain comfortable with traditional notarization; change can feel risky.
- Balancing high security with simplicity is tough—systems that are too complex won't encourage adoption.
- Public blockchains can struggle with scalability and can be costly for frequent use.

2.3 Technical Research

This section describes core technologies potentially applicable for implementing the secure document notarization idea. For each, it will be described how it operates, its benefits, and real-world usage.

2.3.1 Blockchain-Based Notarization

Document notarization on blockchain typically involves computing a cryptographic hash of the file and storing it timestamped on a decentralized ledger. Anyone can later rehash the document to verify integrity without revealing the document's content **doxychain˙power˙nodate**. Some valuable benefits of this approach are:

- **Immutability and transparency:** Hash entries cannot be altered after being written to the chain and are visible across the network (Figure 2.8).
- **No central point of failure:** Decentralized consensus removes reliance on a single trusted authority **deeksha˙uikey˙blockchain-based˙nodate**.
- **Timeproof:** Timestamps on transactions provide verifiable evidence of document existence at a given time **doxychain˙power˙nodate**.

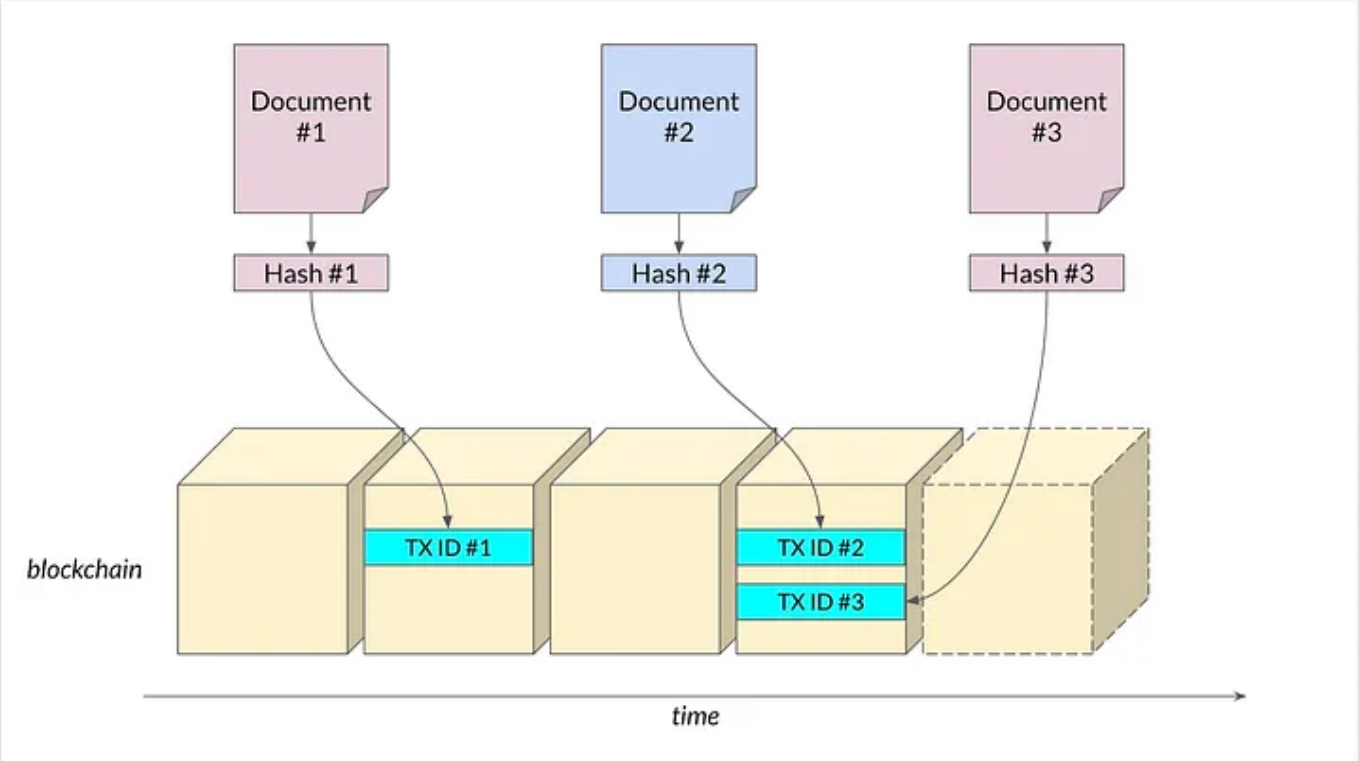

Figure 2.8 - Blockchain Notarization

Recent academic research explores moving from simple "proof of existence" to full document validation, integrating smart contracts, Zero-Knowledge Proofs (ZKPs), AI-based content analysis, and quantum-resistant cryptography—to ensure privacy, detect fraud, and future-proof security **tejas˙chandrakant˙m**
Hybrid blockchain architectures are also being proposed to improve scalability and cost-efficiency, balancing public and permissioned ledger benefits **domenico˙tortola˙scalable˙nodate**.

2.3.2 Electronic Signatures with Multi-Factor Authentication

Combining digital signatures with MFA requires users to authenticate via multiple channels—e.g., password plus a one-time code or biometric—before signing or notarizing a document (Figure 2.9) **sutisoft˙role˙no** Several useful features and benefits are:

- **Significantly enhanced security:** Microsoft reports that MFA can block 99.2% of account compromise attacks **pingidentity˙eight˙nodate**.
- **Fraud reduction:** Additional authentication steps make forging or stealing identities substantially harder **sutisoft˙role˙nodate**, **uplevel˙systems˙top˙nodate**.
- **Regulatory compliance:** MFA aids in meeting stringent identity verification requirements in sectors like finance and legal services **sutisoft˙role˙nodate**.

Figure 2.9 - Multi-Factor Authentication

Remote Online Notarization (RON) platforms integrate MFA to verify signer identity and prevent unauthorized access—combining this with tamper-evident seals and audit trails strengthens legal validity and trust **pronotary˙how˙nodate**.

2.3.3 Electronic & Remote Notarization

E-Notary systems use digital signatures, cryptographic seals, and a public key infrastructure (PKI) to notarize documents electronically. Remote Online Notarization (RON) adds a layer of video verification and identity validation via secure video + digital credentialing (Figure 2.9) **wikipedia˙enotary˙nodate**. The benefits include:

- **Tamper-evident documents:** Digital signatures and seals prevent unnoticed alterations.
- **Cost savings and efficiency:** Eliminates paper handling and on-site meetings; notarizations can be

completed online, rapidly.

- **Audit trails:** Many systems log each step of the notarization process, boosting transparency and legal compliance.

Figure 2.10 - Remote Notarization

For example, Notario.org offers an online notarization platform supporting identity verification via video, digital certificates, and legal compliance across several countries (Spain, USA, Germany, etc.) **wikipedia˙notario˙nodate**.

2.3.4 Advanced Cryptographic Techniques

Zero-Knowledge Proofs (ZKPs) enable one party to prove a statement is true without revealing the underlying data. In blockchain notarization, ZKPs can allow verification of document integrity or content compliance without exposing sensitive data (Figure 2.9) **wikipedia˙privacy˙nodate**. The two benefits of this technique are:

- **Privacy preservation:** Keeps personal or confidential document content hidden while still proving authenticity.
- **Regulatory advantages:** Complies with data protection regulations by limiting exposure of sensitive data.

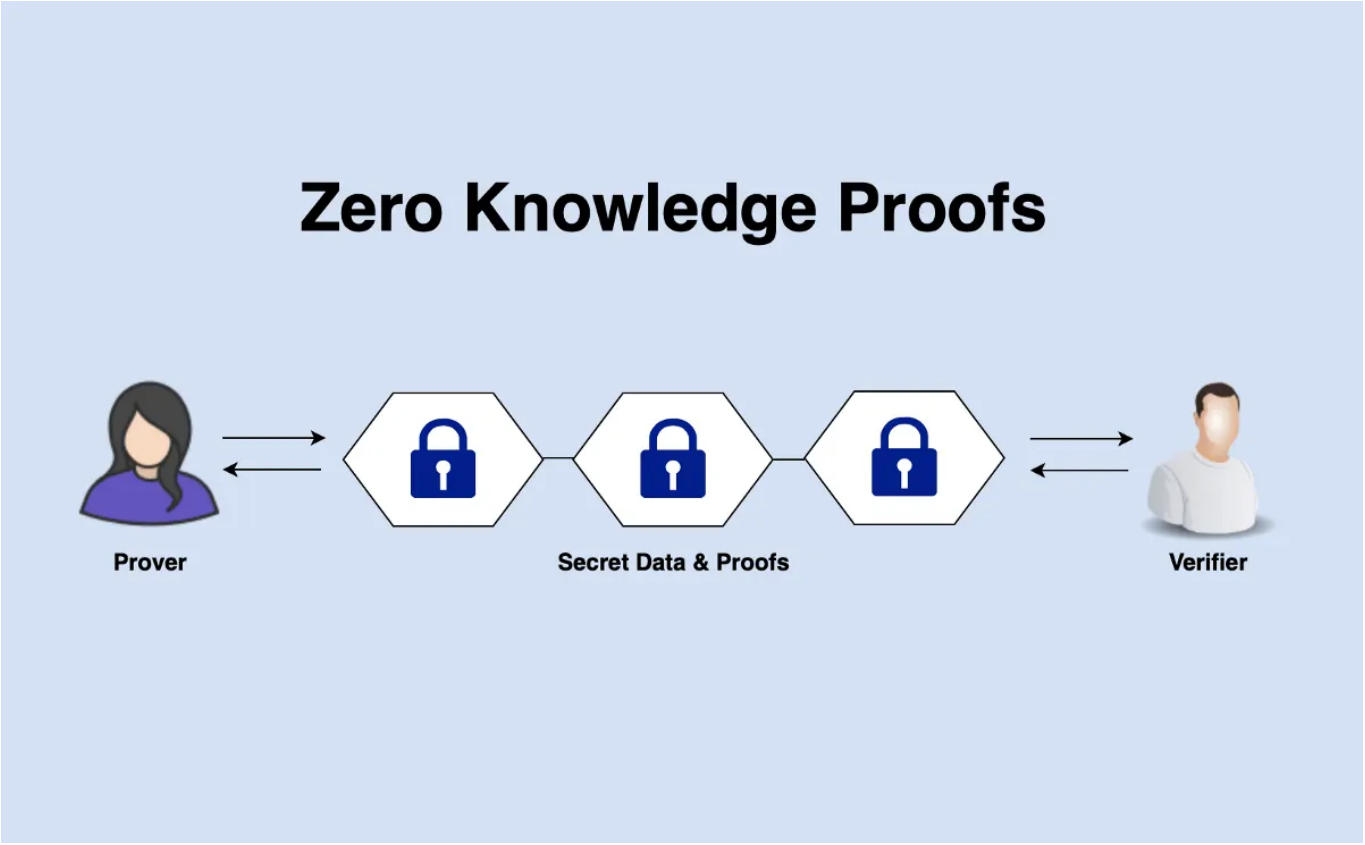

Figure 2.11 - Zero-Knowledge Proof Mechanism

With the potential arrival of quantum computing, future document notarization systems must adopt quantum-safe cryptographic algorithms to remain secure long-term. Research highlights this as an evolving domain in notarization systems to guard against future computational threats.

2.3.5 Hybrid and Scalable Architectures

To address limitations of public blockchains (e.g., transaction fees, latency) and private blockchains (e.g., trust, centralization), hybrid DLT (Distributed Ledger Technology) designs are emerging. They combine local/private storage for efficiency with public chains for decentralization and transparency. Benefits include:

- **Cost optimization:** Keeps frequent notarizations low-cost by batching hashing operations off-chain while anchoring checkpoints on-chain.
- **Scalability:** Manages large volumes efficiently while maintaining tamper evidence and verifiability.

3 Solution Proposal

Based on defined problems and domain research, it is evident that existing document management approaches face serious limitations. Physical records remain vulnerable to forgery, loss, and destruction, while digital records suffer from risks such as unauthorized modification, cyberattacks, and dependence on centralized authorities. Current protections, including digital signatures and certificate-based systems, mitigate some threats but are not sufficient to guarantee authenticity, long-term verifiability, and compliance with modern security standards. At the same time, regulatory frameworks such as GDPR and eIDAS demand solutions that ensure confidentiality, integrity, and accountability. These findings highlight the need for a system that not only secures documents but also provides transparency, privacy, and user trust.

The proposed solution is a blockchain-based notarization platform that enables the creation, signing, and verification of documents in a secure and transparent manner. The platform leverages blockchain to store only the cryptographic hash of documents, ensuring immutability and tamper resistance without exposing sensitive content. This approach allows users to instantly verify document authenticity by comparing a locally computed hash with the one recorded on the blockchain, making the verification process fast, reliable, and independent of intermediaries.

The architecture of the system is structured around four main components:

1. **Client Applications (Web and Mobile)** – provide users with an accessible interface to create, sign, and verify documents. Both platforms support multilingual functionality (English, Russian, Romanian) and integrate usability features such as search, filtering, and history tracking.
2. Authentication and Identity Management – user authentication is handled through Google OAuth2 and local registration flows, with strong password policies and multi-factor authentication (MFA) applied during sensitive operations such as signing. Identity verification relies on a one-time check of personal identifiers (e.g., IDNP), which are not stored long term, ensuring compliance with privacy regulations. Session management is secured with JWT tokens.
3. Application Server – coordinates document workflows, including creation, pending approvals, and signature collection. The server computes document hashes, manages participant approvals, and interacts with both the blockchain network and the database. It also enforces security controls such as input validation, secure APIs, and proper session handling.
4. Blockchain and Database Layer – the blockchain serves as a public, immutable ledger for document hashes, timestamps, and participant references, ensuring integrity and transparency. The database stores only operational metadata and user account information, excluding sensitive identity data. Together, these layers guarantee both system efficiency and data protection.

Functionally, the platform supports a full document lifecycle: registration and identity verification, creation and upload of documents, participant assignment, approval workflows, signing, and final notarization on the blockchain (Figure 3.1). Each step is backed by security controls, ensuring that fraudulent activity, tampering, or unauthorized access is either prevented or immediately detectable.

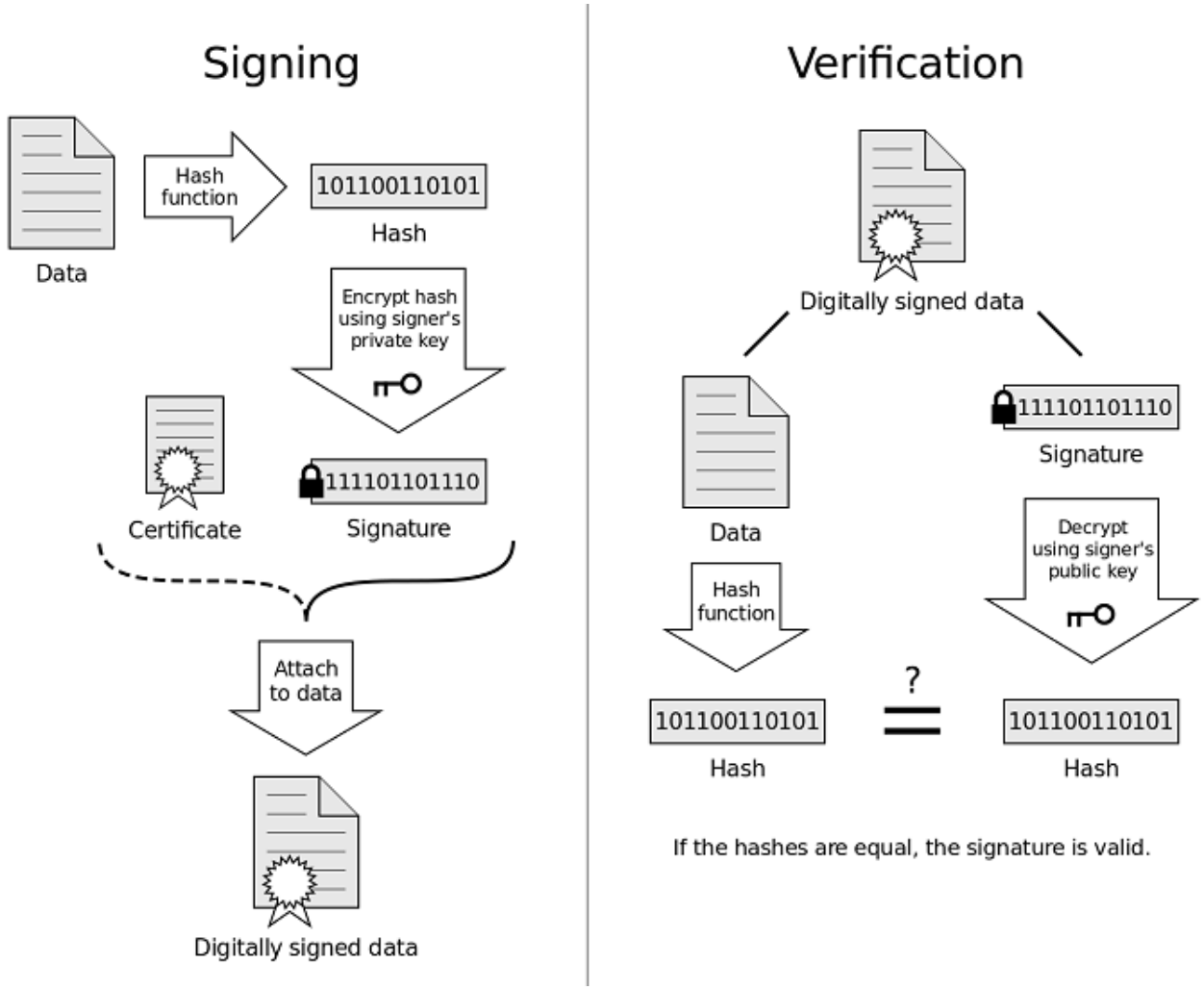

Figure 3.1 - Core Signing and Verifying Logic

By combining usability, regulatory compliance, and cryptographic trust mechanisms, the solution directly addresses the shortcomings identified in traditional document management. It provides a system that ensures authenticity, integrity, transparency, and privacy, offering users a verifiable and tamper-proof method of managing critical documents.

4 System Design

4.1 Technical Requirements

This section describes the architectural backbone of the project, mainly Functional and Non Functional requirements. It describes key actors, functionalities accessed by them, as well as other technical specifications which will be considered in the project practical implementation part (Figure 4.1).

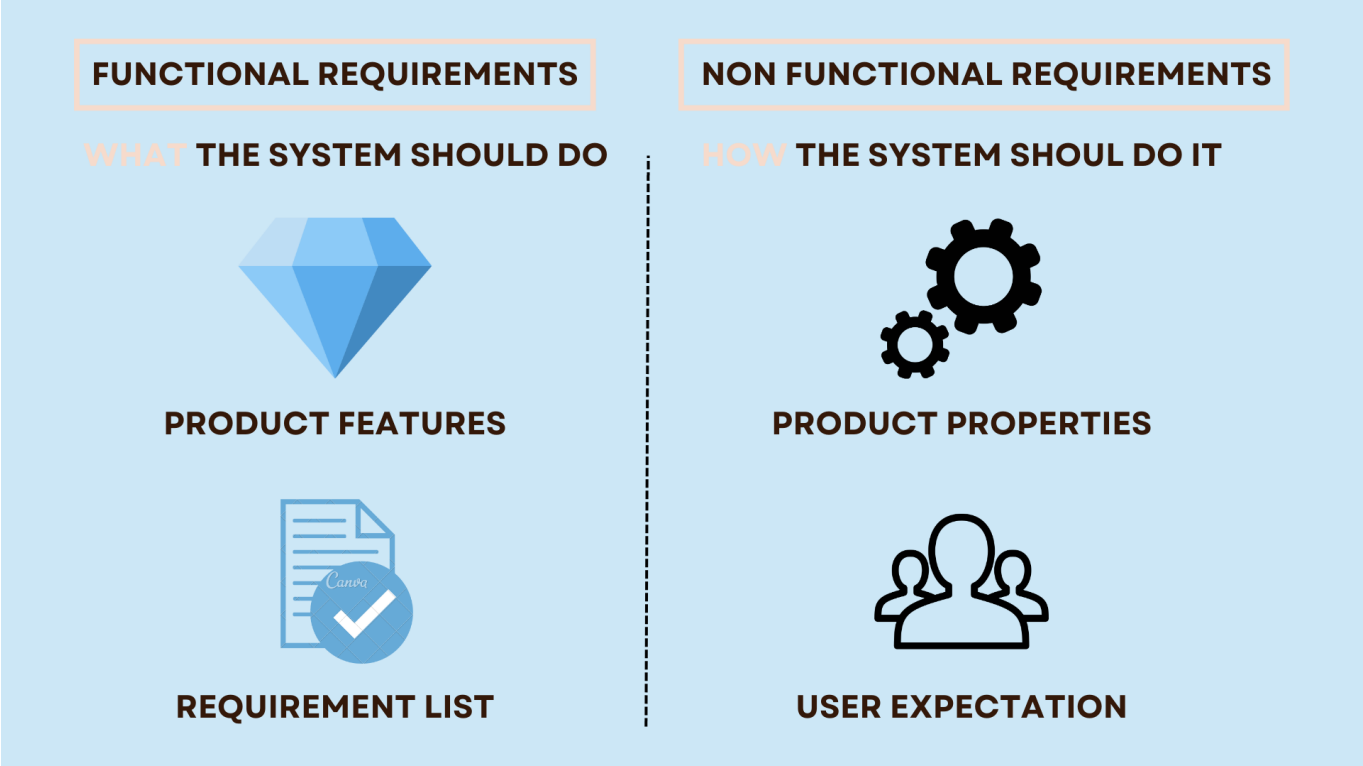

Figure 4.1 - Functional And Non-Functional Requirements

4.1.1 Functional Requirements

The application is designed around three primary user roles:

1. **Unauthorized users**, who may access only the main page and perform registration or login.
2. **Authorized users**, who gain full access to document-related services such as creation, signing, verification, and account management.
3. **Administrators**, who are responsible for managing the user database and can access limited user information when necessary.

The system incorporates a wide set of functions to support these roles. First, it provides multilingual support, ensuring that the entire interface is available in English, Russian, and Romanian. The main page allows navigation to registration, login, and account pages, as well as a mail integration for sending quick requests to support.

On the documents page, authorized users may create new documents by uploading files through a drag-and-drop interface, specifying the parties involved, and submitting them for signing. Pending documents appear in the personal cabinet of each participant until all signatures are collected. At the same time, users can verify documents by uploading a file, generating its hash, and comparing it with blockchain records. If no match is found, a notification is displayed.

The account page contains a full history of signed and pending documents. Each entry provides a preview, the date, file size, hash, status, and the option to download. Account management includes editing personal details, changing passwords, enabling multi-factor authentication, verifying a phone number, and connecting an authenticator app.

Additional functionality is provided through search and filters, which allow users to locate documents based on keywords, sides, or metadata. Pending documents can be opened in a PDF viewer for review and signed electronically. Once the final party signs, the system hashes the document and records it immutably on the blockchain.

Finally, the system enforces secure authorization and registration flows. Users may authenticate through Google OAuth 2.0 or via local registration (Figure 4.2).

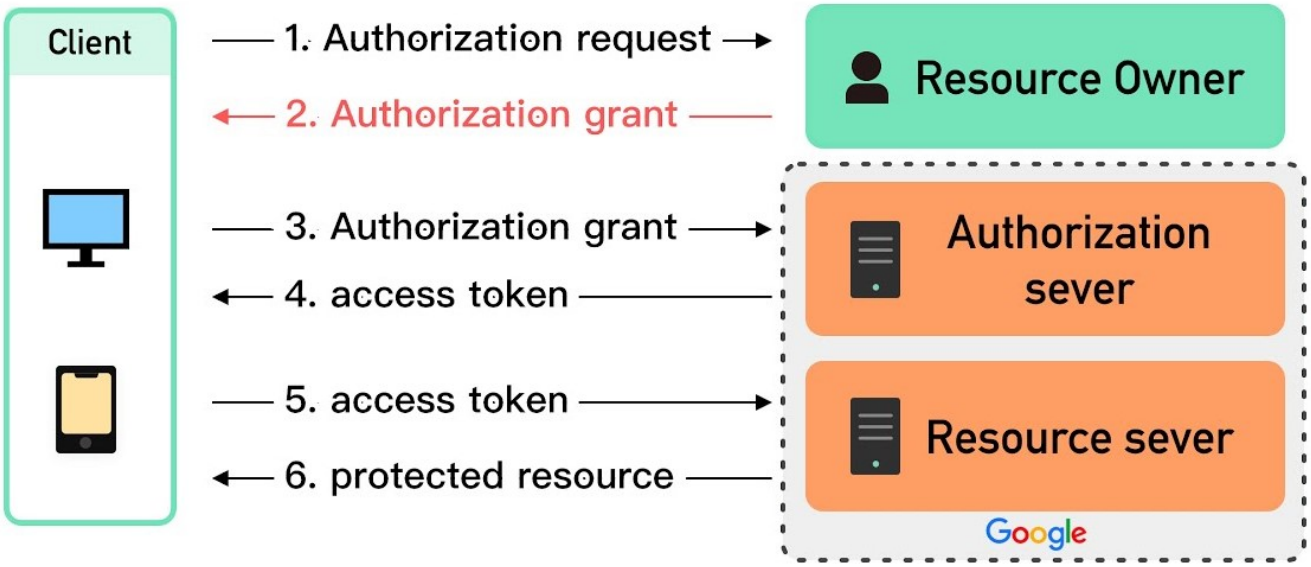

Figure 4.2 - OAuth Mechanism Description

Registration follows a structured, multi-step process: (1) account creation with password and phone validation, (2) email verification through a six-digit code, and (3) submission of identity data (IDNP, birth date, selfie) for administrative approval. A password recovery mechanism is also implemented, allowing users to initiate a reset request by email, verify the reset code, and set a new password meeting complexity requirements .

4.1.2 Non-Functional Requirements

Beyond functionality, the application is governed by several non-functional requirements that ensure its performance, usability, and security.

From a performance perspective, documents themselves are never stored on the blockchain. Instead, only their final cryptographic hash is recorded. This guarantees integrity verification without exposing sensitive content or overloading the ledger.

Regarding usability, the system is designed as both a web application and a mobile application, ensuring broad accessibility. To support a diverse user base, the entire interface is available in three languages: English, Russian, and Romanian.

In terms of security, the platform adopts multiple protective measures:

1. **Sensitive data minimization** – personal identifiers such as IDNP are used solely during registration for verification and are not retained long-term.
2. **Multi-factor authentication (MFA)** – required for signing documents, adding an extra layer of identity assurance.
3. **JWT tokens** – used to secure session management and provide controlled access to the personal cabinet (Figure 4.3).

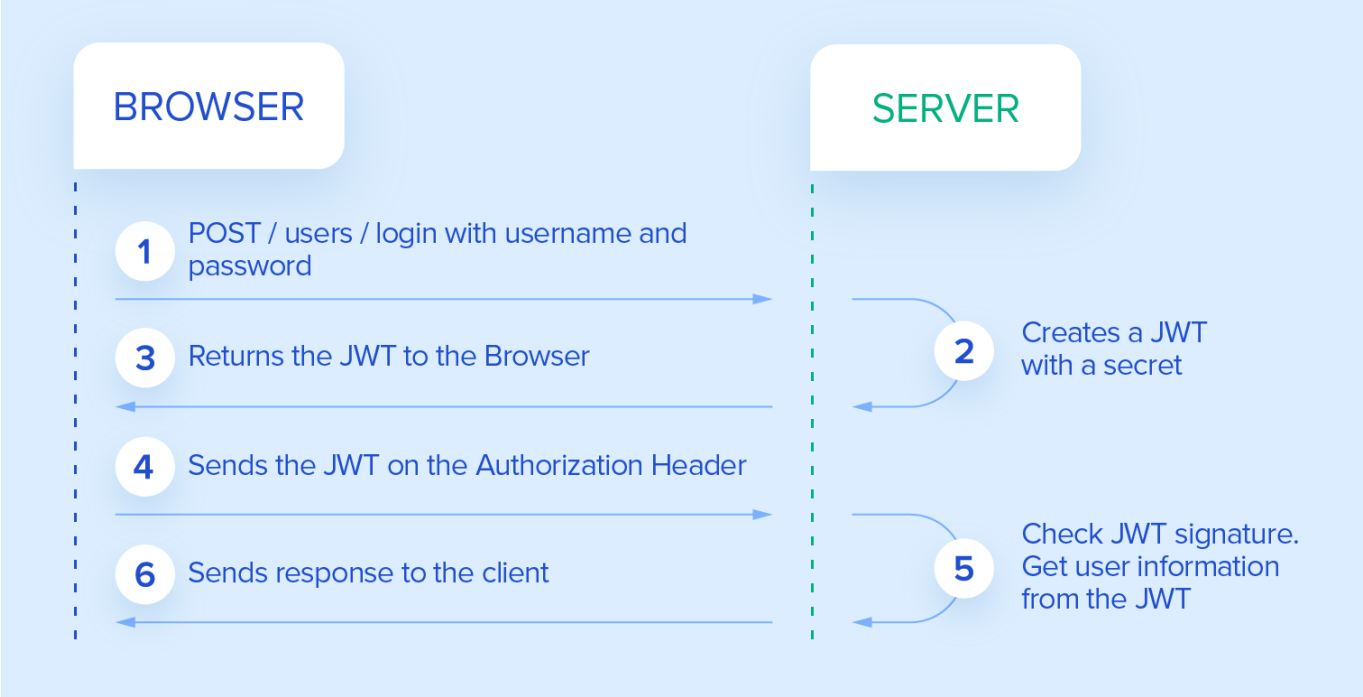

Figure 4.3 - JWT Token Usage Mechanism

Finally, the system adheres to privacy and compliance standards. The blockchain stores only document hashes, with no personal data or file content, ensuring confidentiality and compliance with data protection principles.

4.2 Behavioral Modeling

To capture the dynamic aspects of the system and illustrate how users and components interact, several UML diagrams were created. These diagrams help visualize workflows, user interactions, and message flows between system elements, making the architecture more comprehensible and easier to validate against requirements.

4.2.1 Use Case Diagrams

The use case diagrams represent the primary interactions between different user roles and the system, highlighting the functionalities available to each category of user. There are four main use cases described:

1. **Registration process:** The Figure 4.4 illustrates the registration process for an unauthenticated user. The actor initiates the flow by selecting the option to register into the system. The process includes several mandatory steps such as filling in registration data, providing personal information, verifying the email, and awaiting administrator approval. An optional extension allows the user to authenticate through Google OAuth as part of the email verification step. The diagram highlights the dependencies between activities using ≪include≫ and ≪extend≫ relationships, showing that registration is a structured, multi-step procedure designed to ensure security and identity verification.

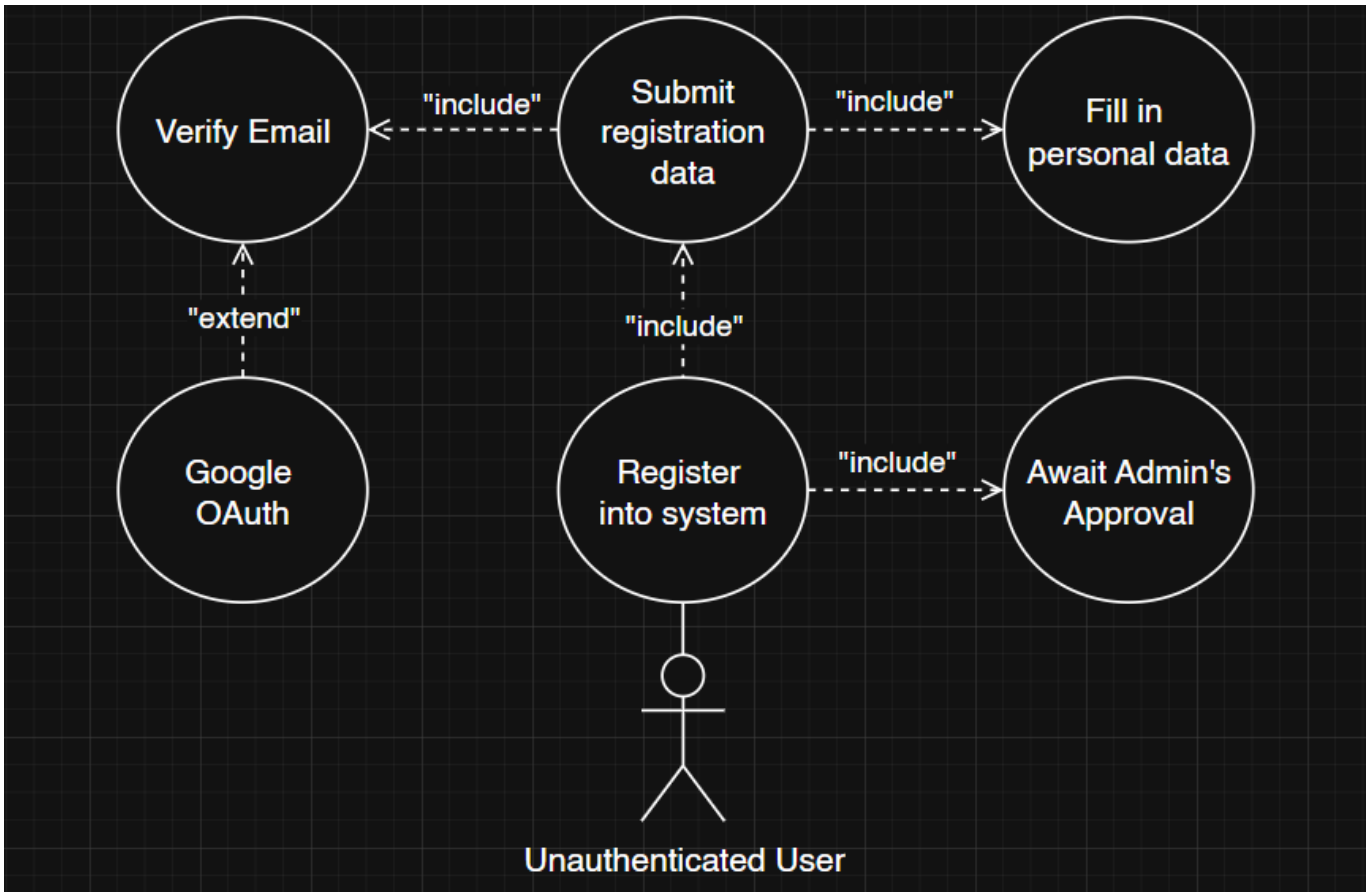

Figure 4.4 - UML Registration Use Case

2. **Document Creation:** The Figure 4.5 represents the process of creating a document within the system by an authenticated user. The actor begins by initiating document creation, which includes multiple dependent steps. The user must upload the document, after which the system checks whether it already exists on the blockchain. The process requires the creator's approval, the specification of participant IDs, and the subsequent approval of all tagged participants. Once these steps are completed, the creation is confirmed, finalizing the process and preparing the document for secure handling. The diagram emphasizes the structured and multi-party approval workflow that ensures document authenticity and prevents duplication.

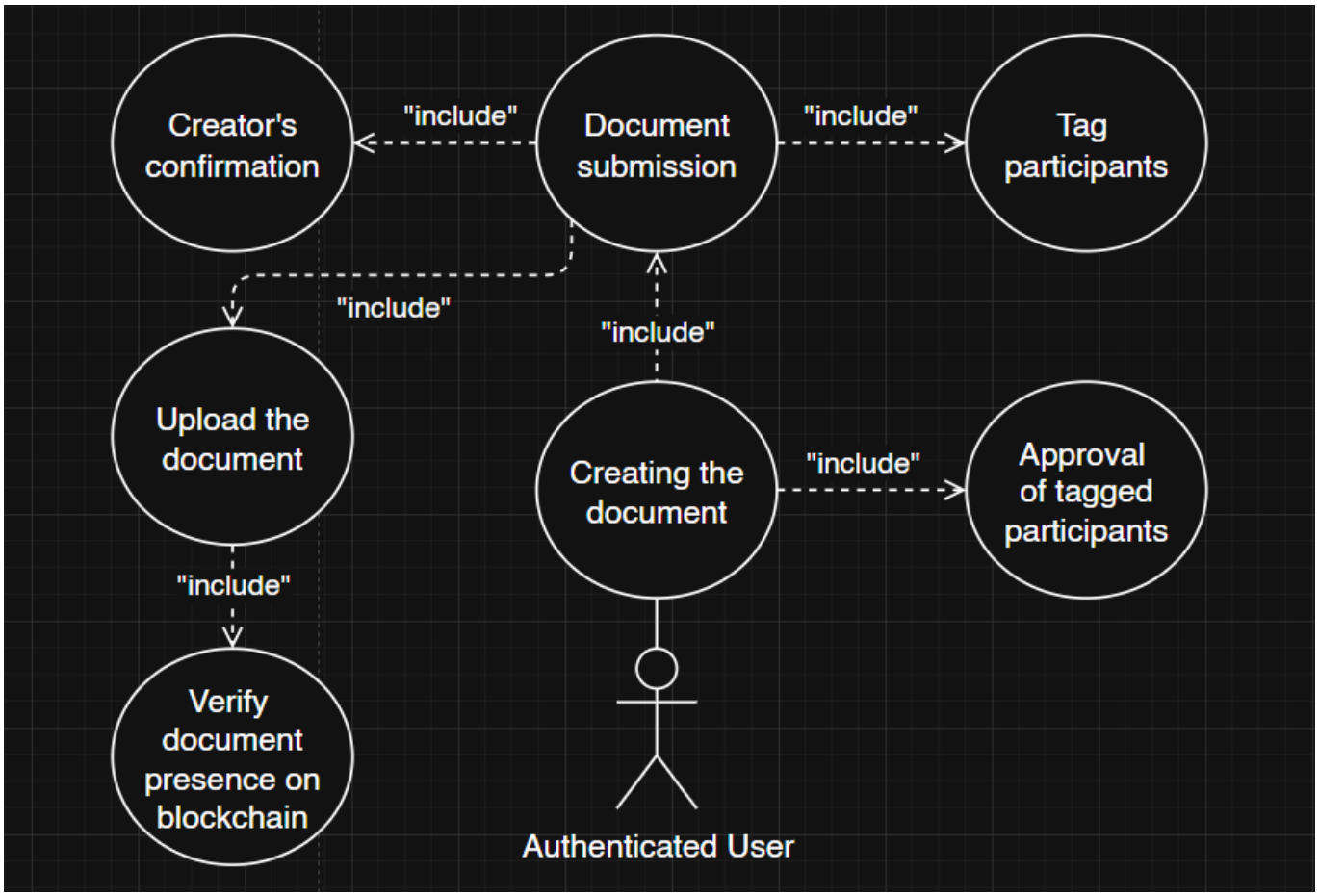

Figure 4.5 - UML Document Creation Use Case

3. **Document Verification:** The Figure 4.6 depicts the process of verifying a document by an authenticated user. The actor initiates the verification by uploading the document and providing the participants' identifiers. The system then calculates the cryptographic hash of the uploaded file and performs a check against existing blockchain records. If a corresponding hash is found, the document is confirmed as authentic and unaltered; otherwise, the system indicates that no match exists. This diagram highlights the structured verification workflow, where hashing and blockchain lookups ensure document integrity and transparency in the verification process.

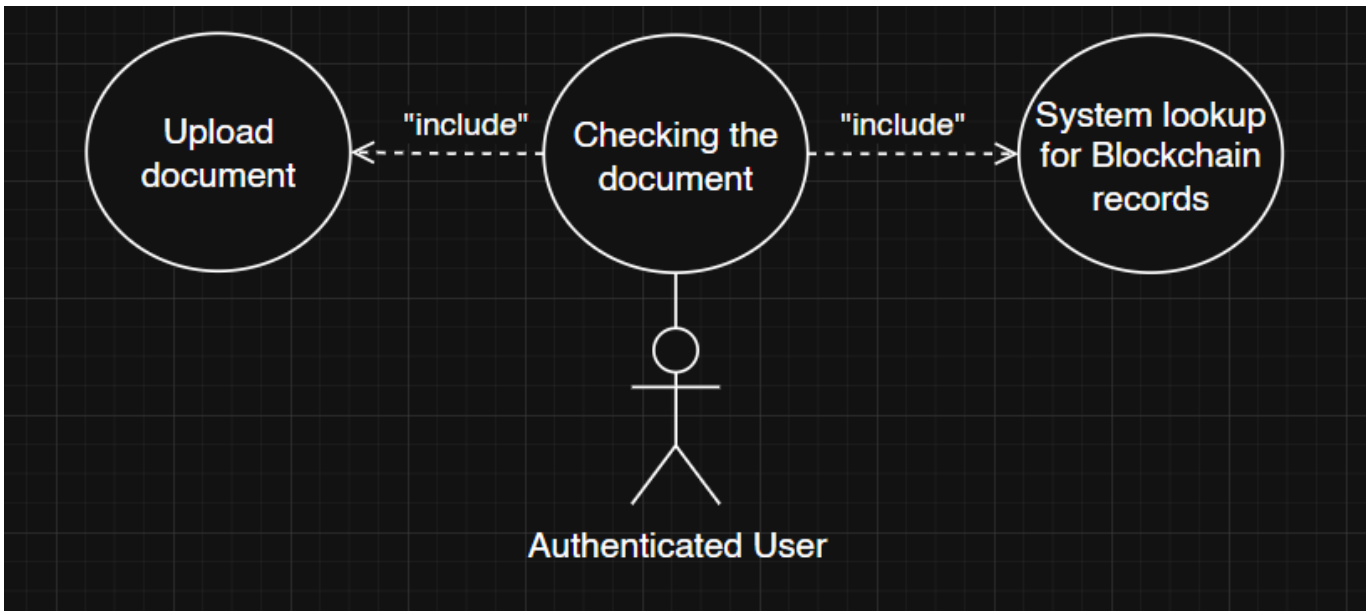

Figure 4.6 - UML Document Verification Use Case

4. **Account Approval:** The Figure 4.7 models the administrator's role in approving user registration. The administrator actor is responsible for validating new accounts by checking the uniqueness of the user and comparing the submitted registration data against records provided by the identity provider (IDP). The process also includes requesting personal data through the IDP to confirm authenticity before final approval. This ensures that only legitimate users, whose identities can be verified and who have not previously registered, are granted access to the system. The diagram emphasizes the administrator's critical role in safeguarding trust, preventing duplicate accounts, and maintaining the integrity of the registration process.

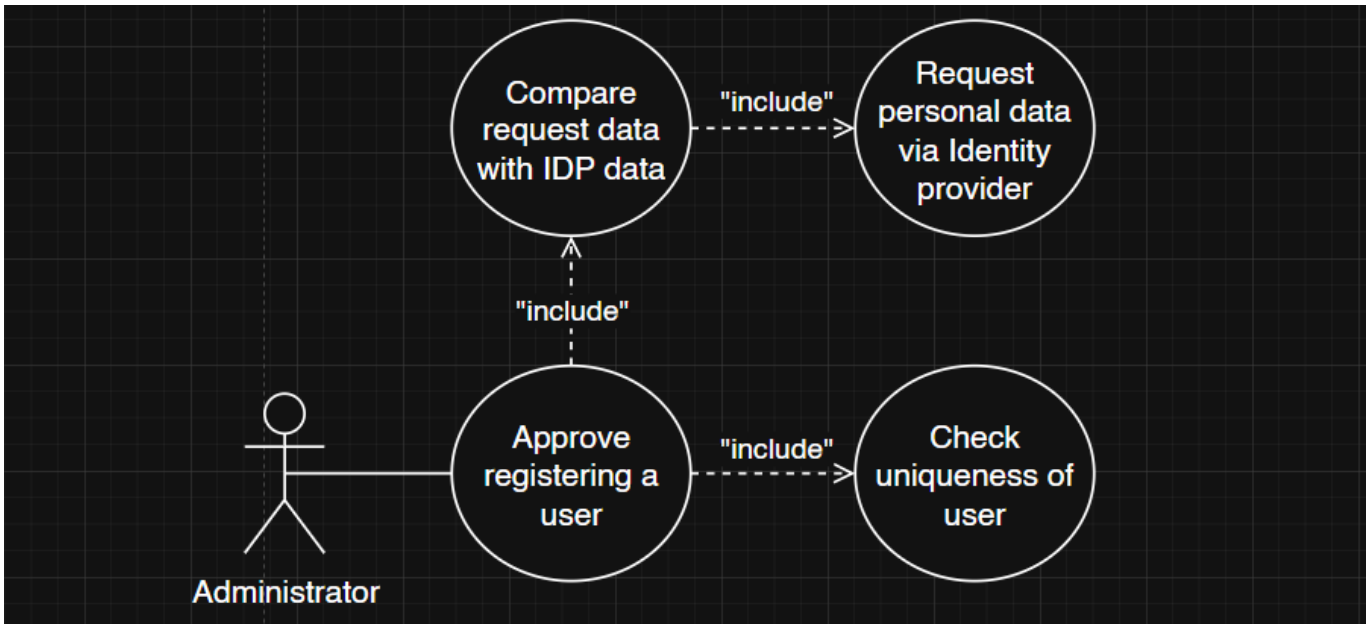

Figure 4.7 - UML Account Approval Use Case

4.2.2 Sequence Diagrams

The sequence diagrams illustrate the chronological flow of actions and messages for key scenarios, showing how the system components collaborate to achieve the intended operations.

1. **Registration process:** The Figure 4.8 illustrates the registration workflow for an unauthenticated user. The process begins when the user initiates registration, which triggers the client interface to start the registration procedure and confirm the initiation with the server. The client then renders the registration window where the user provides personal data. At this point, the server initiates an email verification by sending a code to the user's email account. The client displays this code input field, and the user submits the verification code to confirm the email. The server validates the code and returns a response, which the client displays as a status message. Once verified, the user submits the full registration data, and the server saves the request, confirms the creation, and the client interface updates the UI to reflect successful registration. This sequence diagram emphasizes the interaction between the user, client UI, server, and email service to ensure secure and validated registration.

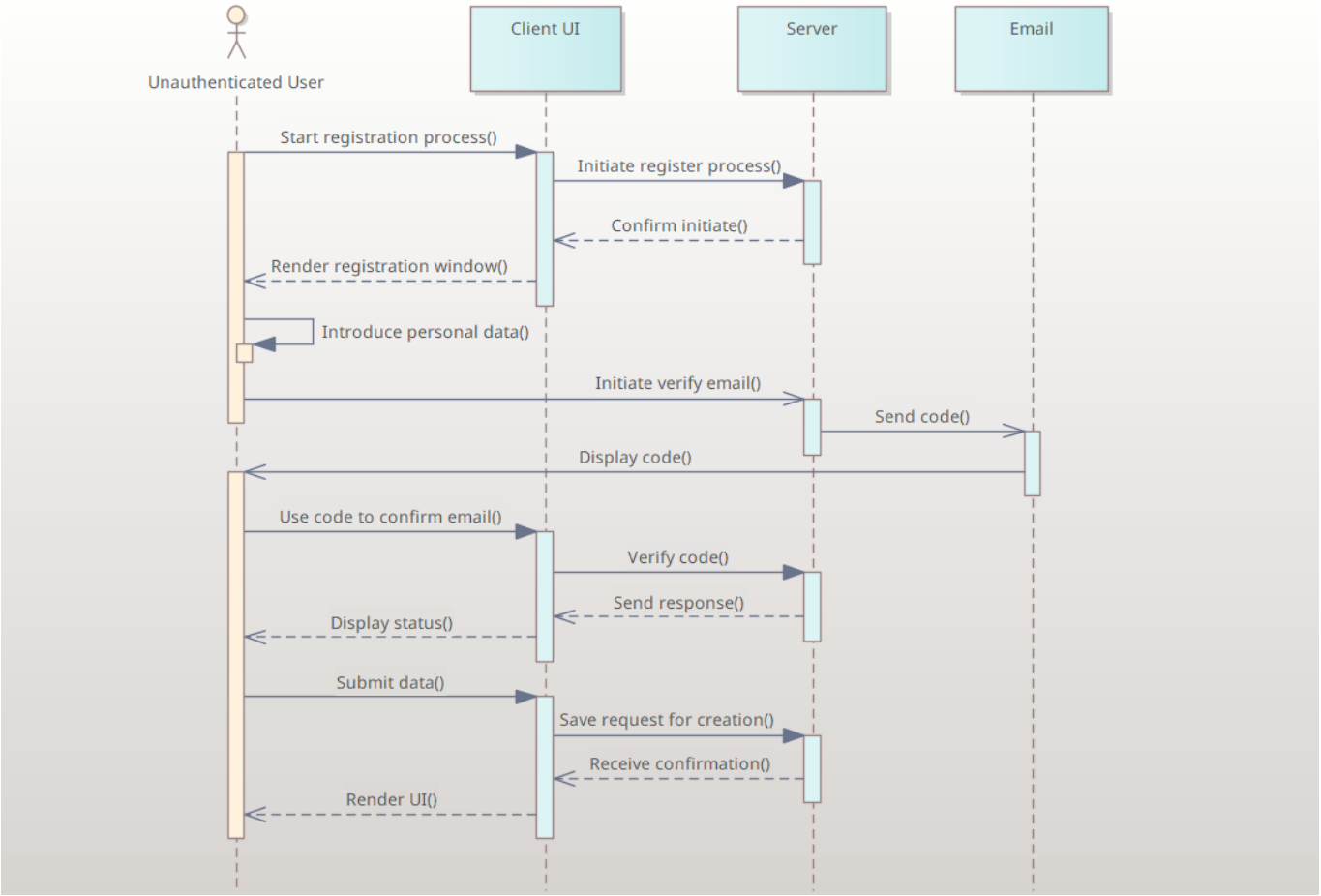

Figure 4.8 - UML Registration Sequence

2. **Document Creation:** The Figure 4.5 describes the process of document creation by an authorized user. The workflow starts when the user initiates the creation process through the client, which renders

the UI. The user uploads the document, provides the participants' IDs, and confirms the action. The client then sends a request for document creation to the server, which computes the hash of the file and checks the blockchain for existing records. If no duplicate is found, the server proceeds by distributing the document to all listed participants for approval. Once the approvals are collected, the server creates a new immutable record on the blockchain and confirms its creation. At the same time, a reference to the new document is stored in the database, linking it to the user's account. Finally, the confirmation is returned to the client, which updates the user interface with the new document's details. This sequence diagram highlights the coordinated interaction between client, server, blockchain, and database to ensure document integrity, approval, and traceability.

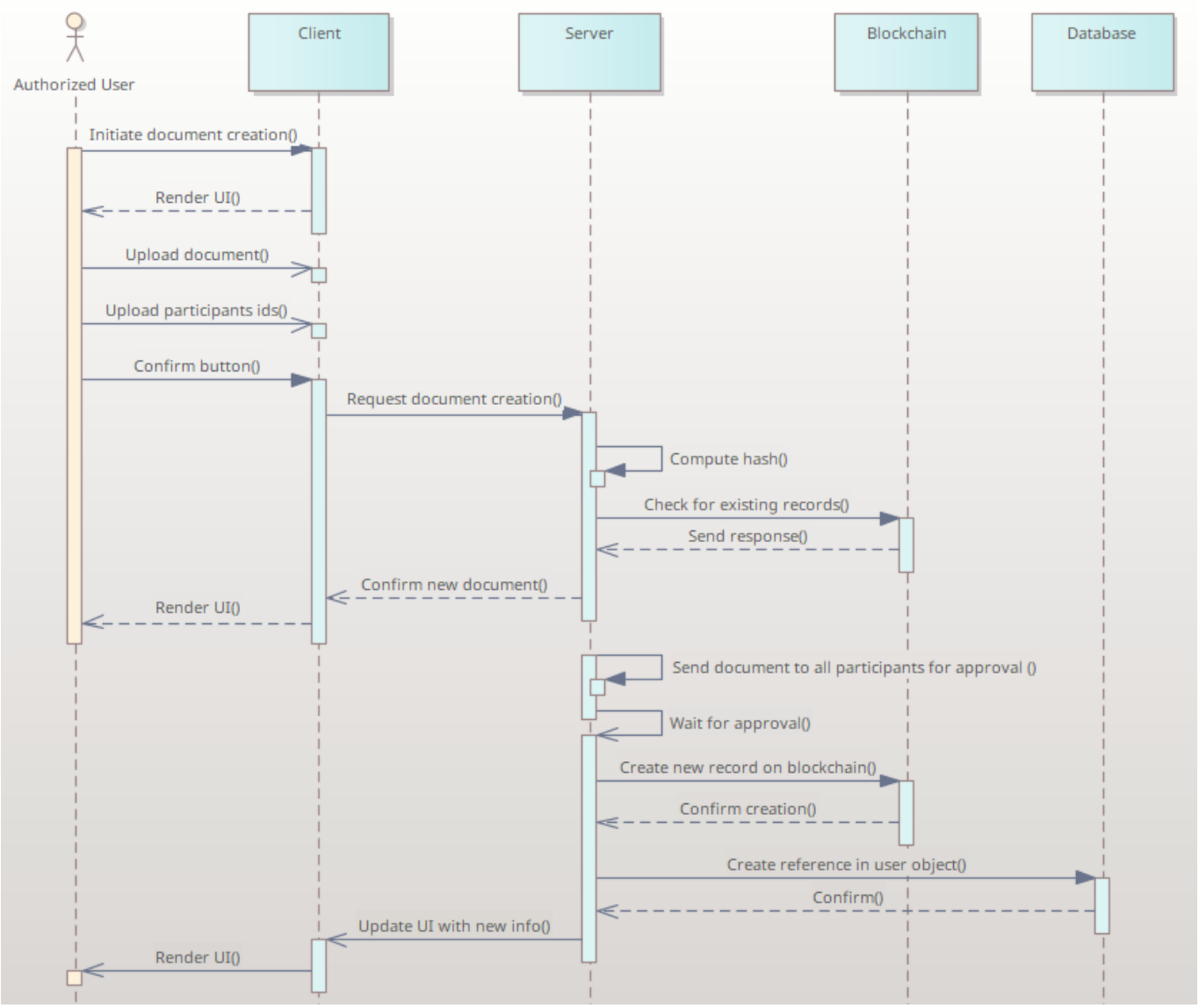

Figure 4.9 - UML Document Creation Sequence

3. **Document Verification:** The Figure 4.10 shows the workflow for document verification by an authorized user. The process begins when the user initiates the verification through the client interface,

which renders the UI. The user uploads the document and enters the participants' IDs before confirming the action. The client then sends a verification request to the server. The server computes a cryptographic hash of the uploaded file and queries the blockchain to check whether a matching record already exists. The blockchain responds with the result of the lookup, which the server forwards back to the client. Finally, the client displays the verification outcome to the user. This diagram highlights the critical interaction between user actions, system components, and blockchain infrastructure to ensure document authenticity and integrity.

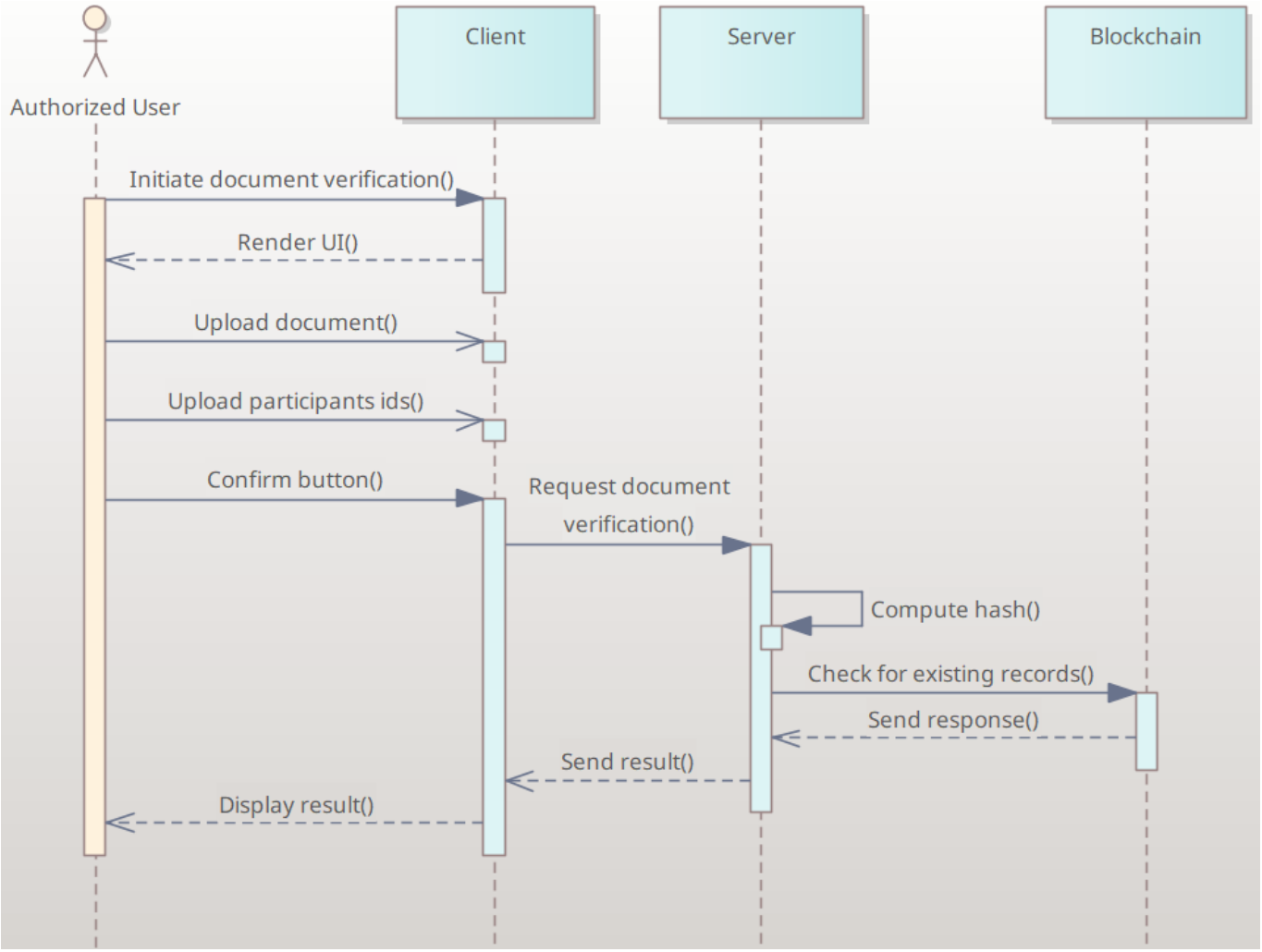

Figure 4.10 - UML Document Verification Sequence

4. **Account Approval:** The Figure 4.11 illustrates the user account approval process managed by the administrator. The flow begins when the administrator requests pending registration data for review, which the server provides to the admin panel. To validate the registration, the server queries the Identity Provider (IDP) for personal data associated with the submitted IDNP and returns the response. The administrator then compares the provided information against the IDP data. If the data is valid, the administrator approves the account creation. The server processes this approval by generating a new

user object in the database, after which a confirmation of account creation is returned to the administrator. This diagram highlights the administrator's critical role in ensuring user identity verification and the coordinated interaction between the server, identity provider, and database during registration approval.

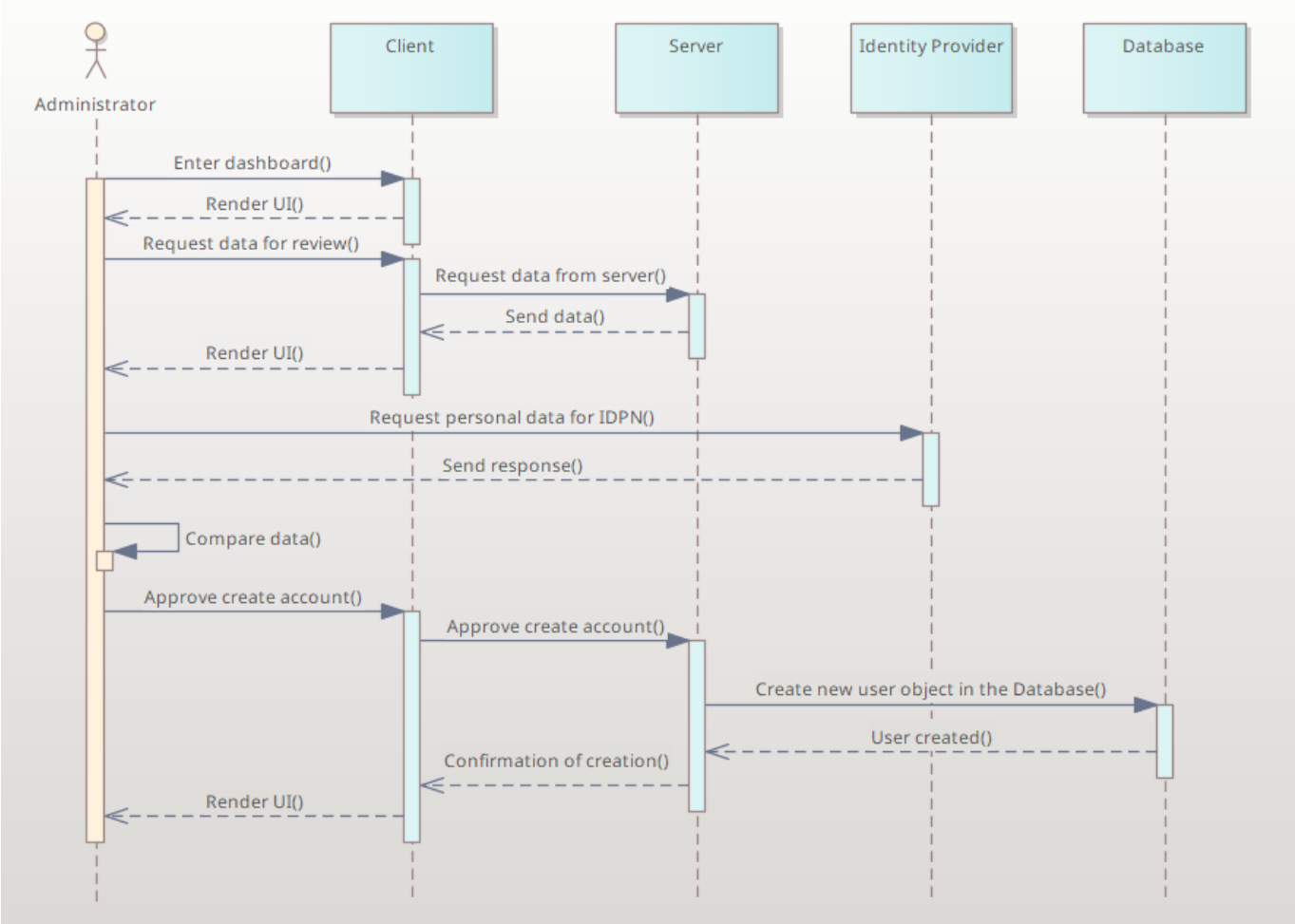

Figure 4.11 - UML Account Approval Sequence

4.3 Structural Modeling

This section presents the structural view of the system, focusing on the static aspects of its architecture. Structural modeling helps define the main components, their responsibilities, and the relationships between them.

4.3.1 Class Diagram

The Figure 4.12 models the core entities and relationships within the document notarization system. At the center is the User class, which contains attributes such as ID, full name, email, phone, status, role, and timestamps, alongside methods for document creation, signing, declining, and verification. Each user is assigned a Role, which defines permissions and enforces access control.

The Document class represents notarized files with properties like ID, title, owner, status, storage

reference, size, and creation details. Its methods manage the signing workflow, including adding participants, submitting for signing, and finalizing signatures. A document is linked to multiple Participants, each of whom has a role, status, and the ability to either sign or decline. These actions generate corresponding Signatures, which record details such as participant ID, signature type, reference, and timestamp.

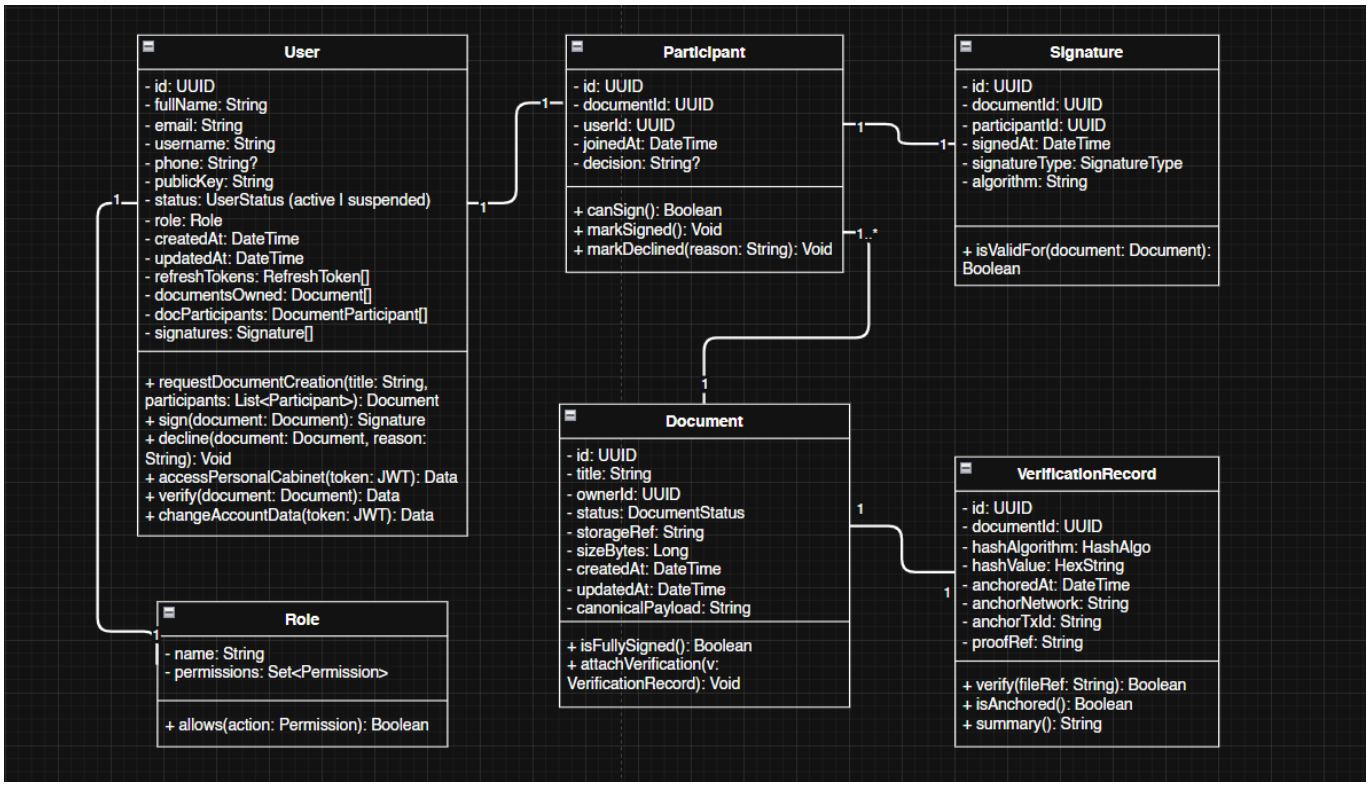

Figure 4.12 - UML Core Domain Class Model

Additionally, the system maintains VerificationRecords, which store blockchain-related data like hash values, algorithms, anchoring timestamps, and network references. These records provide functions to verify integrity, check anchoring, and generate summaries.

Altogether, this diagram highlights how users, roles, documents, participants, signatures, and verification records interact. It reflects a cohesive domain model where documents are securely managed, signed by multiple parties, and validated against blockchain anchors to ensure integrity and trustworthiness.

4.4 Figma User Interface Mockups

The landing page (Figure 4.13) is the first page seen by the user when entering the site, so it gives the user a clear description of the functionality the service provides. In addition to the general introduction, the landing page highlights the main features of the platform such as secure digital document signing, document verification, and account management. The page also provides quick navigation buttons for login and registration, along with a concise call-to-action section that encourages new users to get started immediately. The design focuses on clarity, minimalism, and trust-building by including branding and possibly short explanations of how the service ensures security and authenticity.

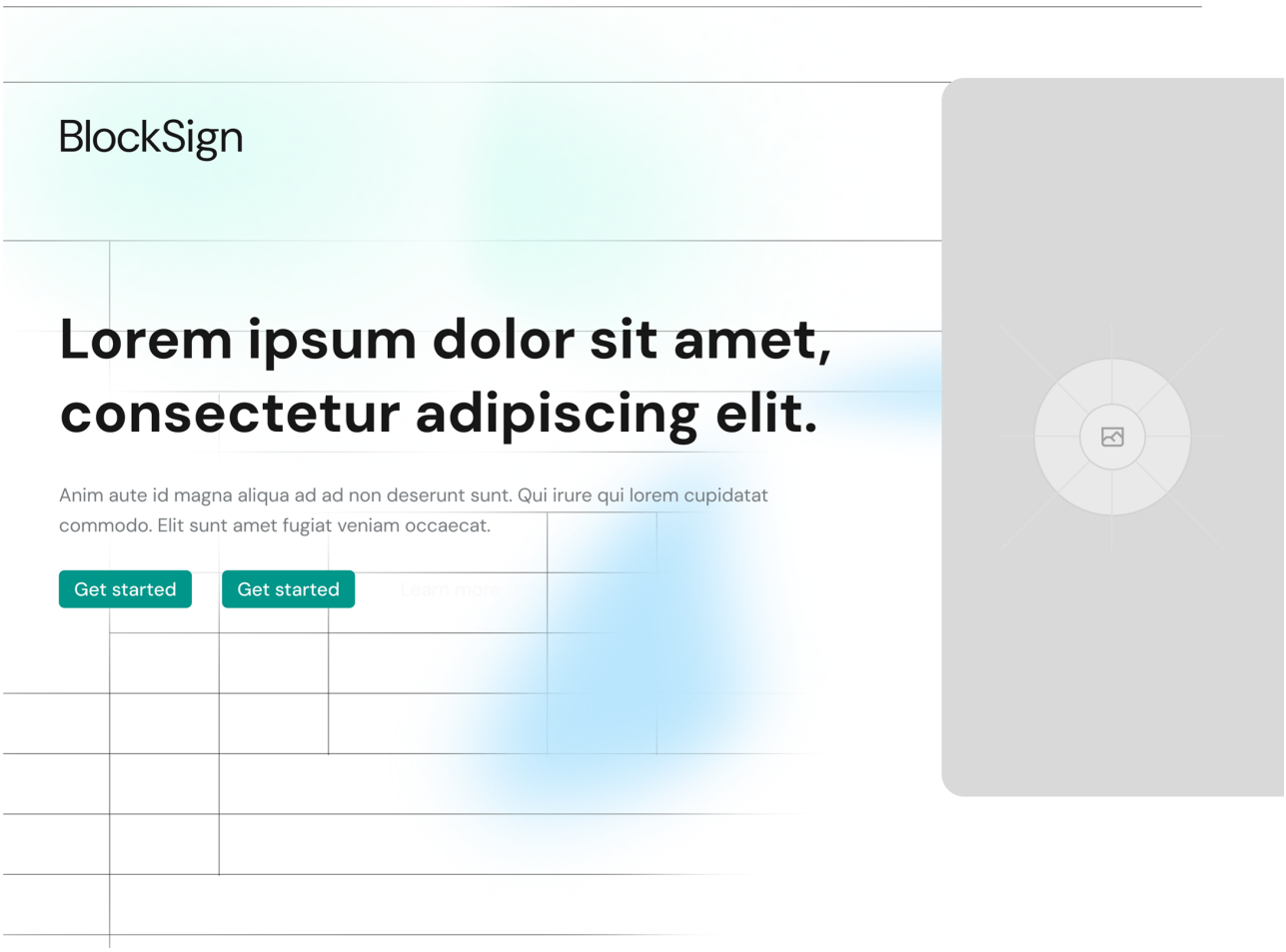

Figure 4.13 - Landing page

The login page (Figure 4.14) has the possibility of entering via email and password, as well as entering through existing accounts in services such as Google. It also contains a “Forgot password?” option to help users recover their credentials securely. The design maintains a simple, uncluttered layout to make the login process quick and intuitive. Security measures such as preventing brute-force attempts or notifying users of suspicious logins are typically included in the backend to protect user data.

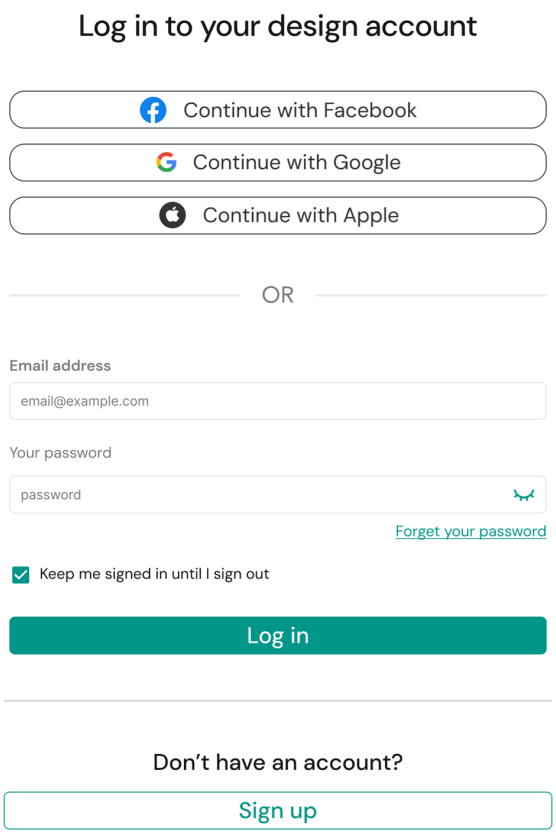

Figure 4.14 - Login page

The registration page (Figure 4.15) provides the user with the possibility to create an account in the service and lets the use of other services such as Google to provide the information needed to service. The form includes basic personal data such as full name, email, and password, while enforcing password strength requirements. The integration with third-party authentication providers ensures faster registration and reduces errors during manual data entry.

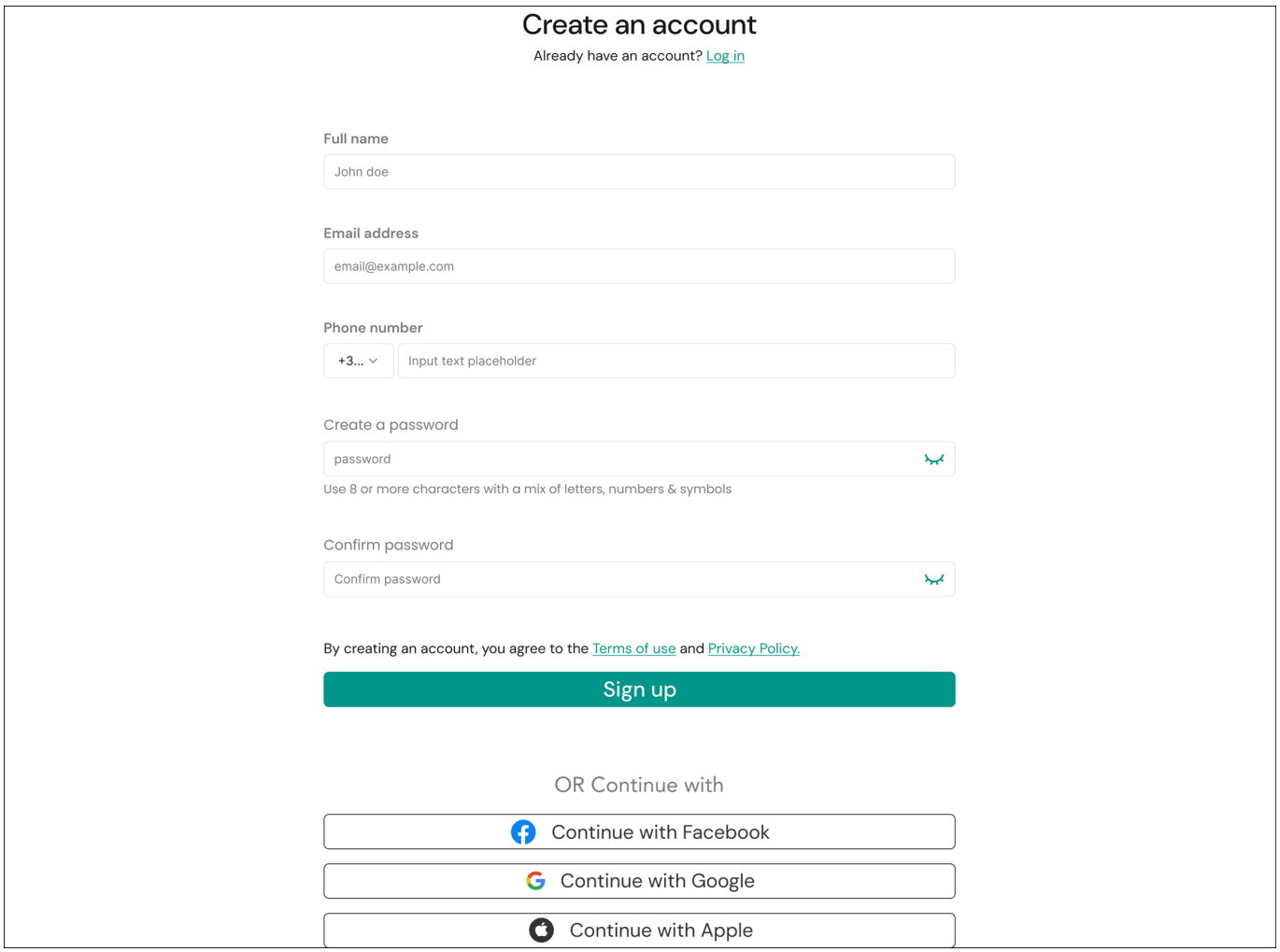

Figure 4.15 - Register page - User info

The second step of registration (Figure 4.16) is a confirmation of the user's email. An email with a verification link or code is sent to the address provided. This step ensures the validity of the email and protects against fake accounts. Without confirmation, the user cannot proceed to the next steps, which increases platform reliability and security.

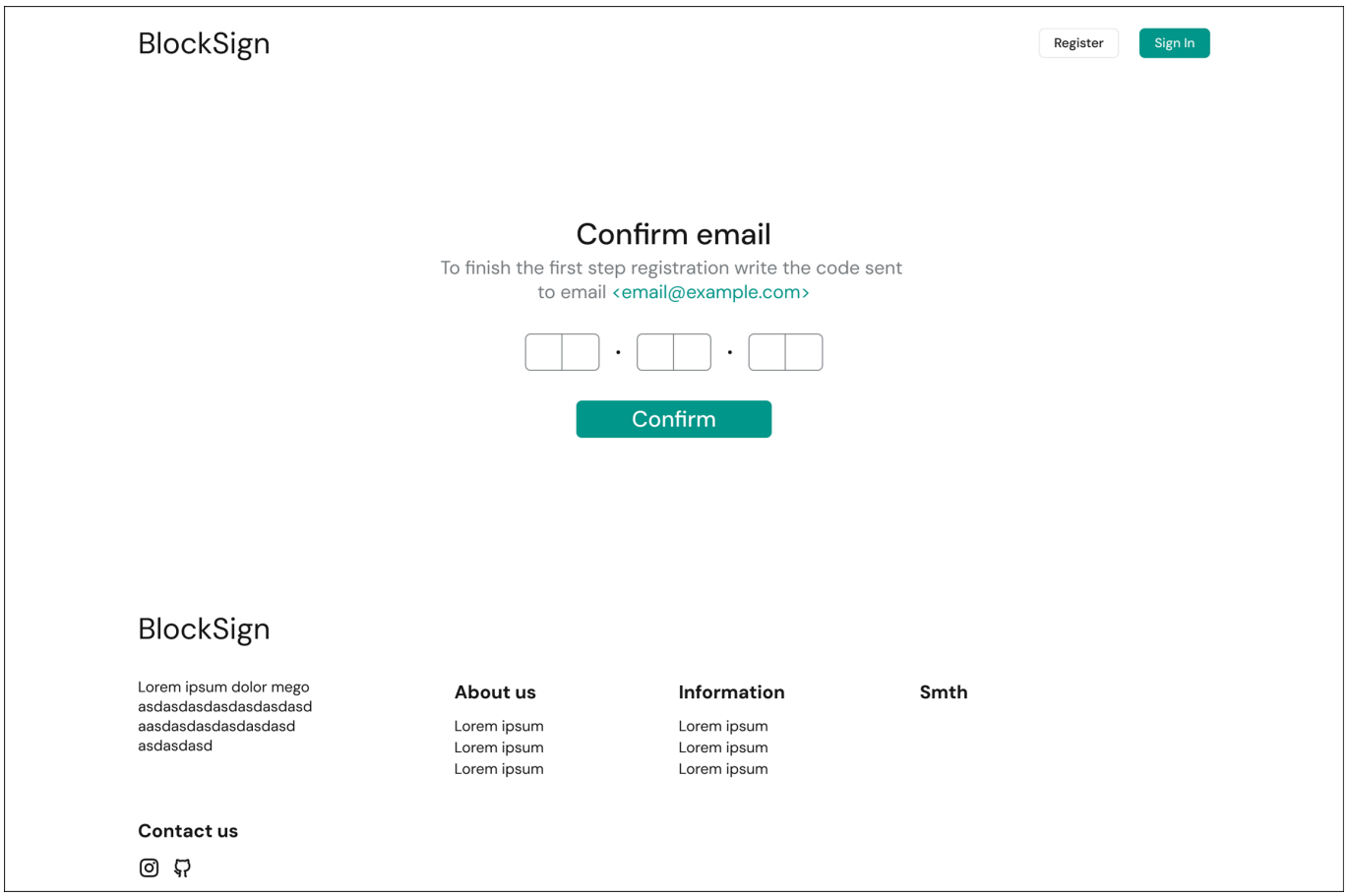

Figure 4.16 - Register - Verifying the email

The third step of registration (Figure 4.17) is a confirmation of identity using sensitive personal data that is not stored or used after verification. This step may include providing a government-issued ID or other verification methods, processed through a secure third-party provider. The system uses these checks only to validate the authenticity of the user, ensuring compliance with regulations while safeguarding privacy.

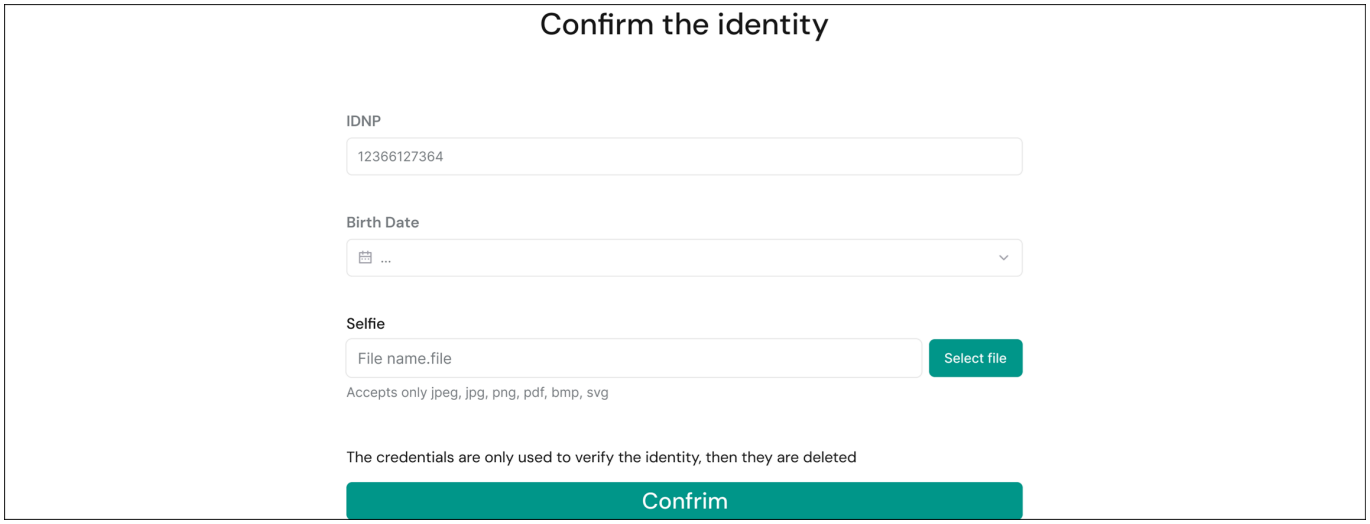

Figure 4.17 - Register - Confirming the identity

Then the user sees the successful finish (Figure 4.18) screen and is redirected to his account. This page confirms that the registration has been completed, and it often includes a “Go to Dashboard” or “Start Now” button. It reassures the user with a confirmation message and sometimes provides short onboarding tips or links to documentation.

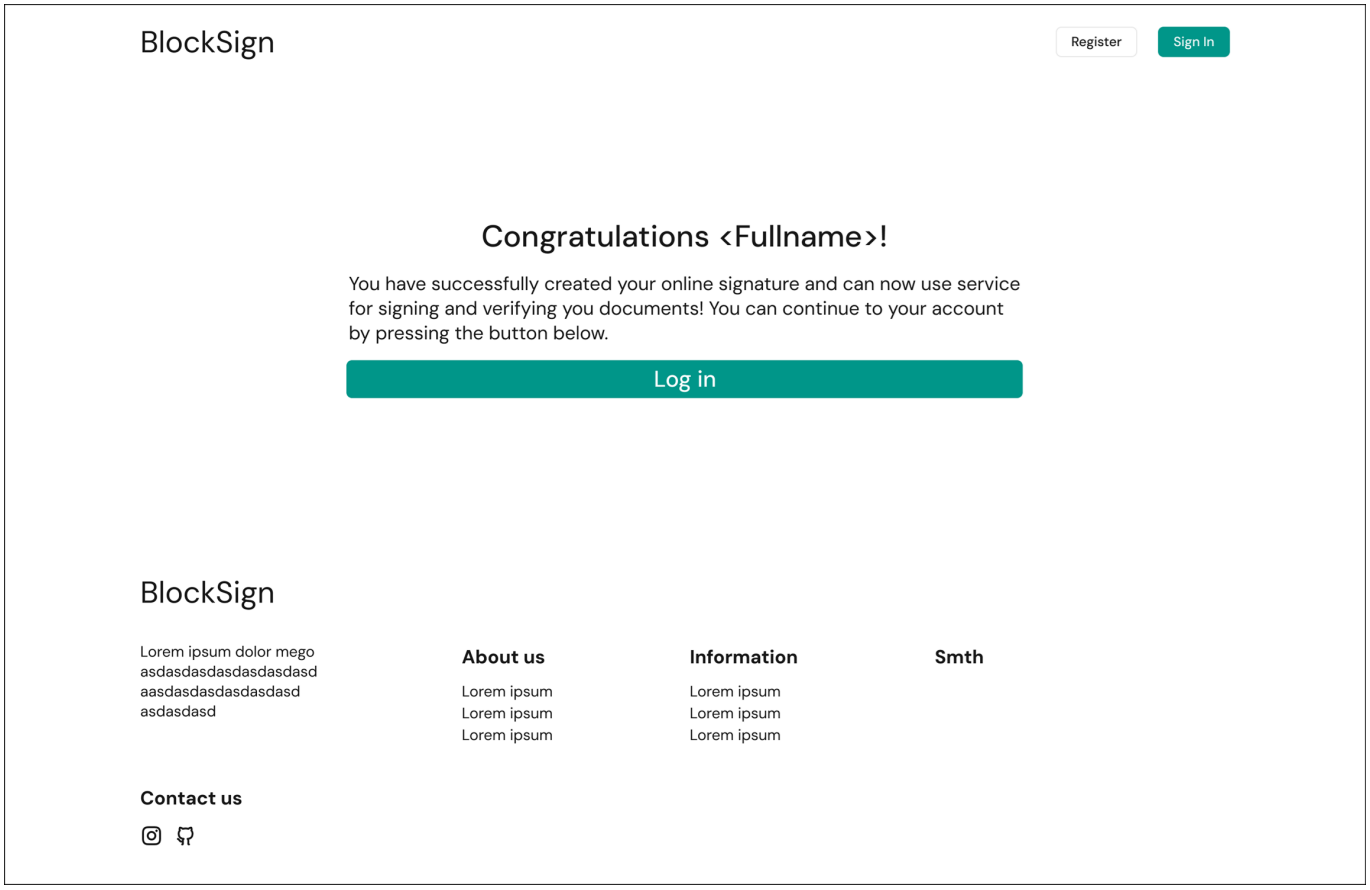

Figure 4.18 - Register - Successful finish

When authorized, the user gets access to the main features of the platform: documents (Figures 4.19, 4.22), and account settings (Figures 4.23, 4.24, 4.25).

The documents page has 2 main functionalities: uploading (Figure 4.19) and creating a session of signing the document, and verifying (Figure 4.22) whether a file with the given users was ever signed. This structure makes the page a central hub for all document-related operations.

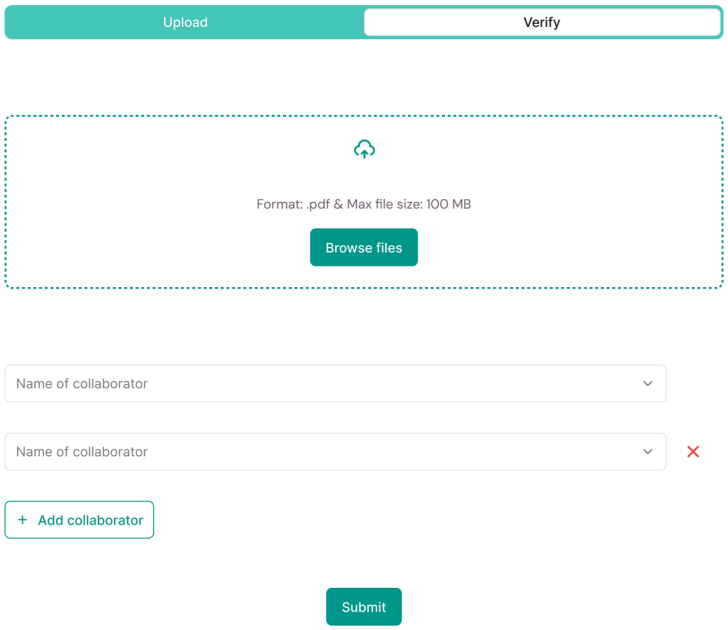

Figure 4.19 - Documents page - Verify an existing document

The results of verifying might be only “exists” (Figure 4.20) or “does not exist” (Figure 4.21). When a document is confirmed as existing, metadata such as the signing parties, date, and document ID may be shown. If the document is not found, the system suggests uploading it for signing or checking the entered details again.

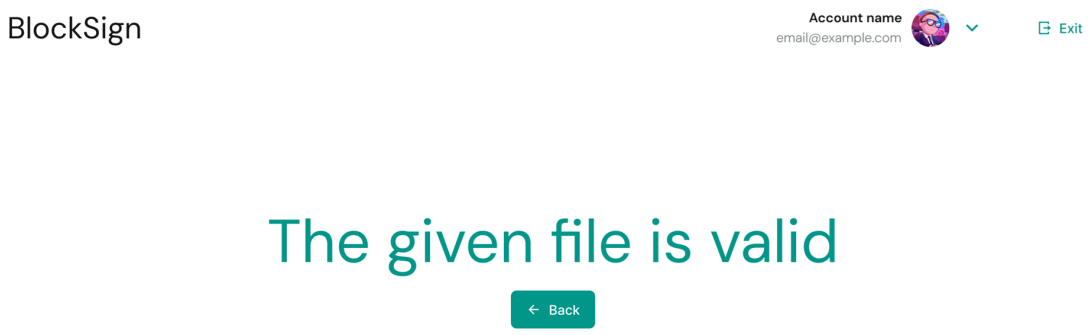

Figure 4.20 - Documents page - Existing instance of document

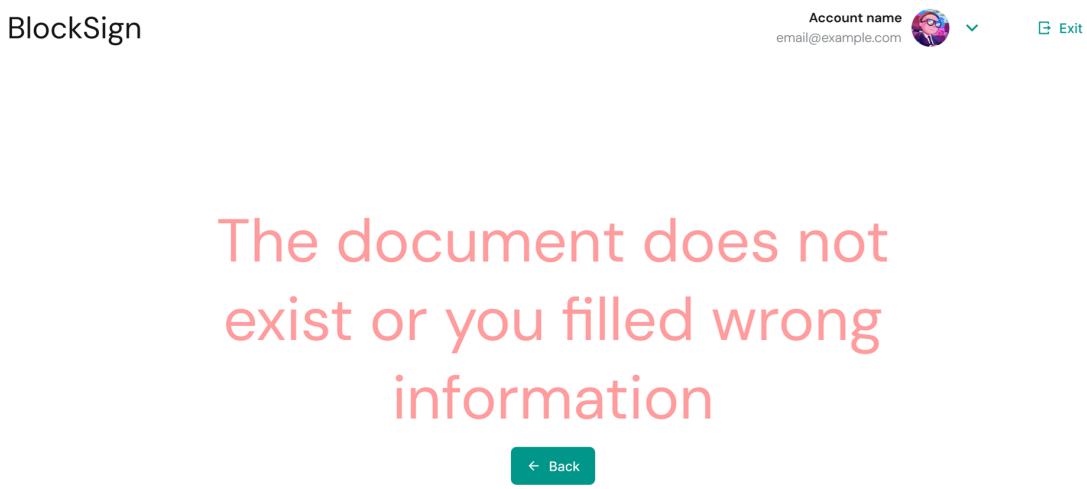

Figure 4.21 - Documents page - Not existing instance of document

On the upload option (Figure 4.22) the session of signing the document is created by adding the sides of the signing via account name/id. The user can invite multiple parties, define the order of signing, and set additional security measures such as signing deadlines. The interface ensures that uploaded files are stored securely and encrypted until the signing process is complete.

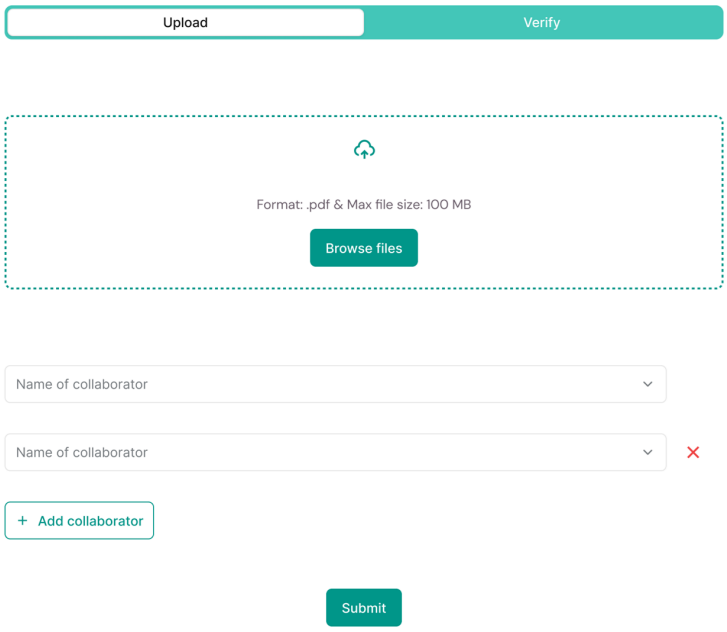

Figure 4.22 - Documents page - Upload and sign the document

The account page gives the ability to manage the user account (Figure 4.23) and protection settings (Figure 4.24) as well as view and sign documents (Figure 4.25). This separation into tabs makes navigation straightforward and intuitive.

The account tab (Figure 4.23) allows customization of the previously introduced personal information, including name, email, and profile image. It also enables linking or unlinking third-party login services.

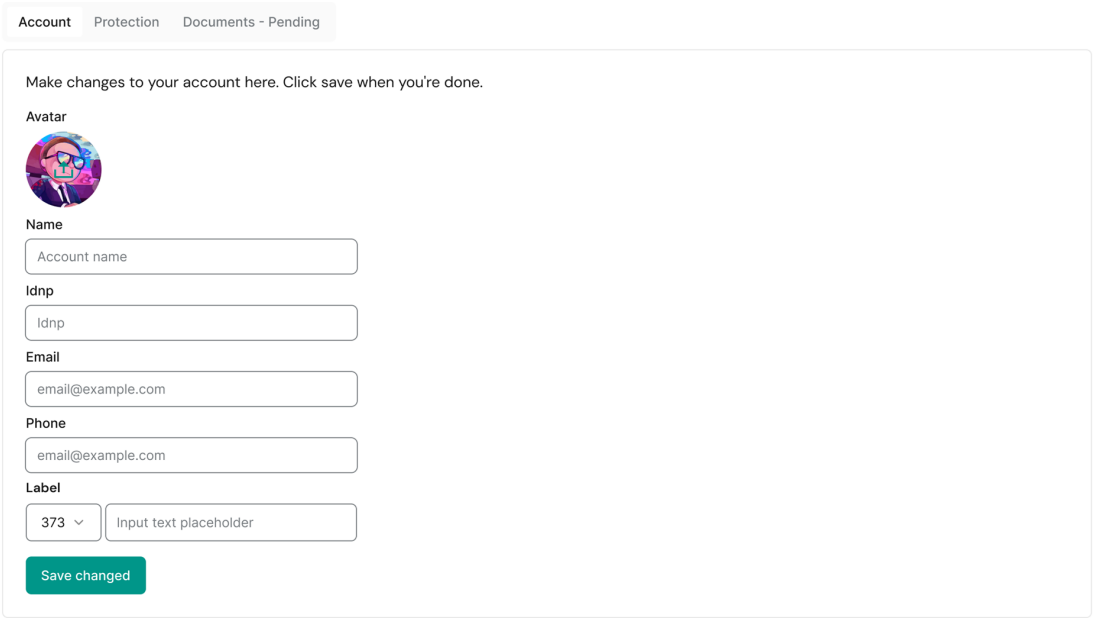

Figure 4.23 - Account page - Possibility to modify personal info

The protection tab (Figure 4.24) provides the possibility to change password and enable multi-factor authentication. It may also include login history, trusted devices, and alerts for unauthorized access. This improves transparency and enhances user trust in the platform.

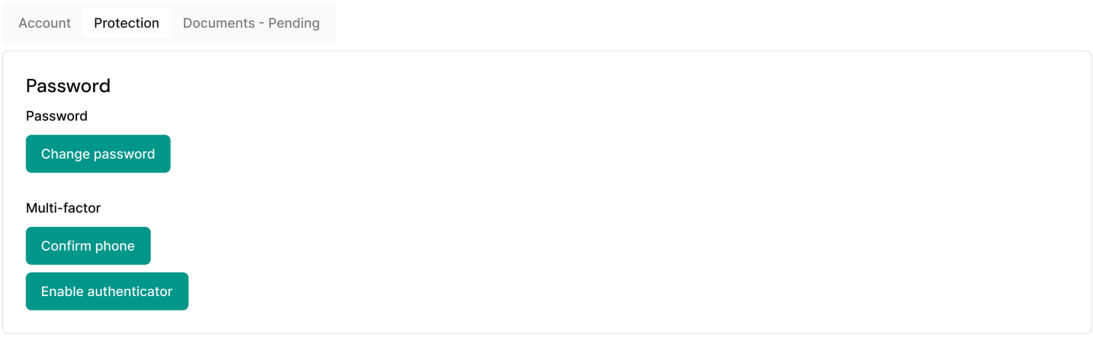

Figure 4.24 - Account page - Possibility to modify the protection methods

The documents tab (Figure 4.25) gives the user a list of their documents and pending requests. Users can view details such as status (signed, pending, declined), participants, and timestamps. From this tab, the user can also directly open a document for review or initiate a new signing process.

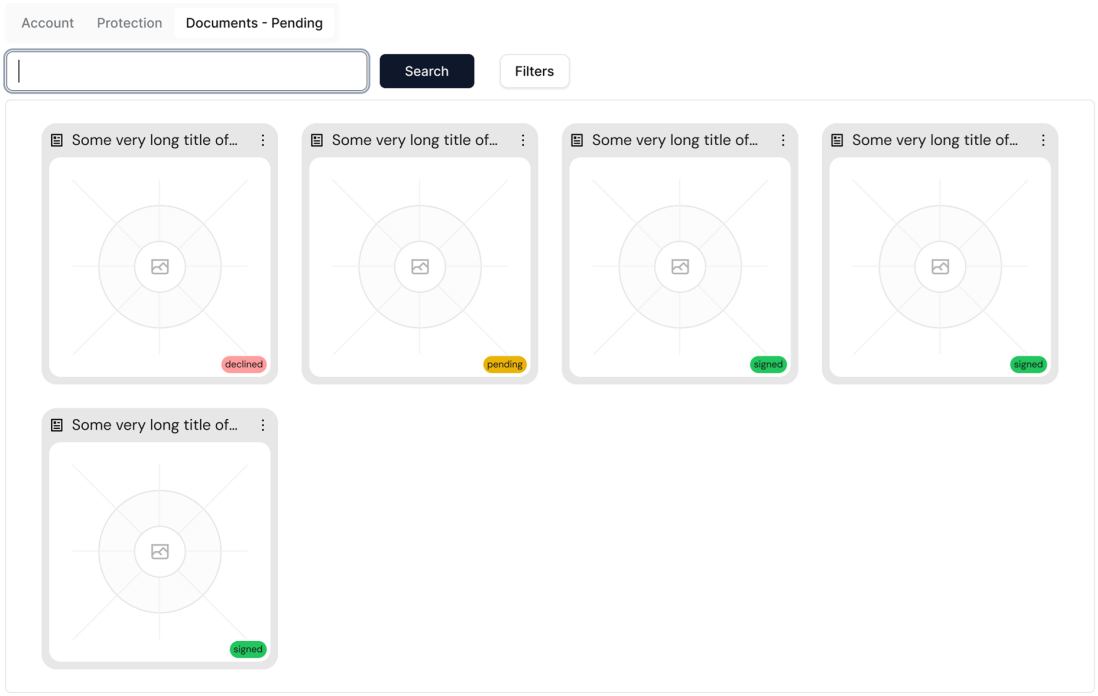

Figure 4.25 - Account page - View of the documents

5 Practical Implementation

The Project repositories are hosted on GitHub, are public and can be accessed by following these references: Back-end Core Logic**alexei˙pavlovschii˙blocksign˙nodate** and Front-end User Interface**vladimir˙vitcov**

5.1 Back-end architecture

The back-end of the system has been developed with a focus on scalability, modularity, and security. Its purpose is to provide all the critical functionality required for user registration, authentication, document management, and secure verification of signatures (Figure 5.1).

Figure 5.1 - Main Project Routers

The implementation relies on Node.js with the Express.js framework, while data persistence is handled by PostgreSQL through the Prisma ORM.

5.1.1 Application Layer

The server is structured around separate route modules, each of which is responsible for a specific part of the functionality. For example, the auth.routes.ts file manages the passwordless login process using challenge–response verification, while registration.routes.ts covers the steps of email OTP verification, registration requests, and account completion with a public key. The administrator's responsibilities, such as reviewing and approving registration requests, are implemented in admin.registration.routes.ts. Finally, user-oriented functionality, including profile management and document operations, is centralized in user.routes.ts. This modular design improves maintainability and makes it easier to extend the system in the future.

5.1.2 Database Layer

The database schema is implemented using Prisma ORM on top of PostgreSQL. Entities such as User, RegistrationRequest, Document, DocumentParticipant and Signature are modeled explicitly with for-

eign keys and unique constraints to guarantee consistency (Figure 5.2).

```
model User {
  id                String     @id @default(uuid())
  email             String     @unique
  fullName          String
  username          String     @unique
  publicKeyEd25519  String?
  keyCreatedAt      DateTime?
  phone             String?    @db.VarChar(32)
  role              UserRole   @default(USER)
  status            UserStatus @default(ACTIVE)
  createdAt         DateTime   @default(now())
  updatedAt         DateTime   @updatedAt

  refreshTokens     RefreshToken[]        @relation("UserRefreshTokens")
  documentsOwned    Document[]            @relation("UserDocumentsOwned")
  docParticipants   DocumentParticipant[] @relation("UserDocumentParticipants")
  signatures        Signature[]           @relation("UserSignatures")

  createdFromRequestId String?
}
```

Figure 5.2 - User Database Model

The use of Prisma Client enables type-safe queries and simplifies operations like migrations, seeding, and testing.

5.1.3 Security Measures

The architecture incorporates several security mechanisms. Authentication challenges are valid only for five minutes and become invalid once used. Refresh tokens are stored in secure cookies to reduce exposure. The use of Ed25519 ensures modern cryptographic security, while SHA-256 guarantees the integrity of documents. Finally, file validation ensures that only PDF documents are accepted.

5.1.4 Authentication and Tokens

Authentication in the system is fully passwordless and relies on public/private key pairs. Users prove their identity by signing challenges provided by the server. Successful verification grants them access tokens and refresh tokens in the form of JWTs.

Access tokens are short-lived and returned in the response body, while refresh tokens are long-lived and stored securely in HTTP-only cookies. Middleware functions such as requireUser and requireAdmin check the validity of tokens and enforce role-based access control.

5.1.5 Email Service

The system integrates Nodemailer as an email delivery mechanism configured with SMTP credentials for the project's domain. Emails are used in two important contexts. First, during registration, users receive a six-digit OTP code for email validation. Second, during document workflows, participants receive notifications when they are asked to review and sign a document, and again when the document has been finalized (Figure 5.3).

```
await sendEmail(recipients.join(','),
  `Document to review & sign: ${doc.title}`,
  `<p>You have a new document to review and sign: <b>${doc.title}</b>.</p>
   <p>Verify its SHA-256 hash matches the payload shown in the app before signing.</p>`,
  { attachments: [{ filename: `${doc.title}.pdf`, content: req.file.buffer, contentType: 'application/pdf' }] }
);
```

Figure 5.3 - Email Sending Mechanism

5.1.6 Cryptographic Module

Security of authentication and documents is achieved with the Ed25519 digital signature scheme, implemented via the @noble/ed25519 library (Figure 5.4).

```
const mnemonic = generateMnemonic(wordlist, 128); // 12 words (128 bits).
const passphrase = ''; // optional user-provided extra password (BIP-39 optional)

const seed = mnemonicToSeedSync(mnemonic, passphrase); // Buffer/Uint8Array

// Derive Ed25519 SLIP-0010 key at your fixed path
const PATH = "m/44'/53550'/0'/0'/0'"; // BlockSign path
const { key: privateKey } = derivePath(PATH, Buffer.from(seed).toString('hex'));

const publicKey = await ed.getPublicKeyAsync(privateKey);
```

Figure 5.4 - Key Pairs Generation Mechanism

A dedicated crypto utility handles key pair generation, signing, and verification of messages. In addition, SHA-256 is used for hashing documents, ensuring their integrity before they are accepted or signed.

5.1.7 Document Workflow

The MVP implementation of the document signing process follows a well-defined lifecycle. When a user creates a document, they upload the file, calculate its SHA-256 hash, and sign a canonical payload. The payload and initial signature are stored in the database.

Participants selected by the creator receive an email with the PDF attached and the corresponding hash. Each participant is expected to calculate the hash independently, compare it with the payload, and if it matches, sign the document (Figure 5.5).

```
const canonical = buildCanonicalPayload({
  sha256Hex: body.sha256Hex,
  title: body.docTitle,
  participants: users.map(u => ( u.username! )),
});

const ok = await ed.verifyAsync(
  Buffer.from(body.creatorSignatureB64, 'base64'),
  Buffer.from(canonical, 'utf8'),
  Buffer.from(creator.publicKeyEd25519, 'hex')
);
```

Figure 5.5 - Build Paylaod and Verify Signature

Their signatures are then added to the database. Once all required signatures are collected, the document status changes to signed, and a notification is sent to all involved parties.

5.1.8 Extensibility

The modular design of the back-end makes it adaptable to future improvements. Currently, documents are distributed as email attachments, but the system is ready for integration with external storage providers such as AWS S3. The ChainAnchor entity in the future versions of database will anticipate the anchoring of finalized document hashes on blockchain networks. Similarly, while usernames are currently used to tag participants, the design leaves room for later integration with decentralized identity systems.

5.2 Front-end UI

The front-end of the BlockSign system represents a modern, secure, and user-centric web application built using cutting-edge technologies. The implementation emphasizes client-side security, responsive design, and intuitive user experience while maintaining the highest standards of cryptographic security.

5.2.1 Frontend Architecture

The front-end is built using Next.js 14 with React.js as the core framework, leveraging TypeScript for type safety and enhanced developer experience. The application utilizes Tailwind CSS for styling, providing a consistent and responsive design system across all components (Figure 5.6).

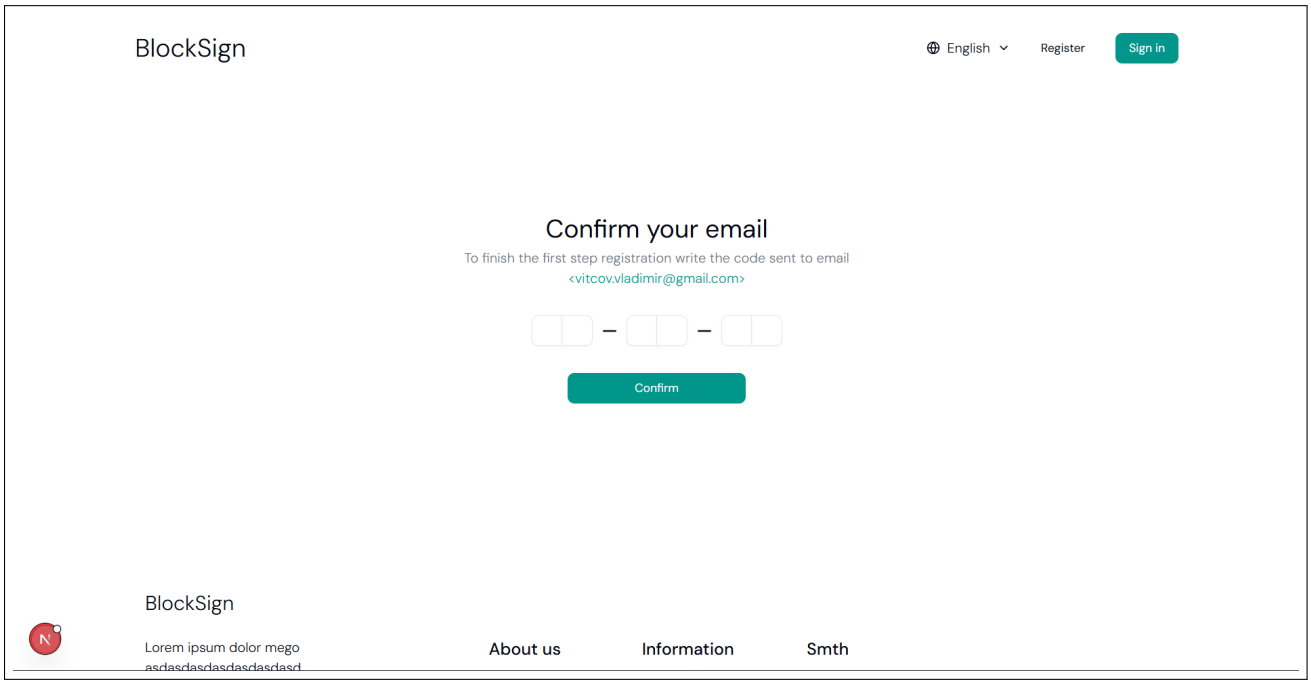

Figure 5.6 - Frontend Technology Stack Implementation

The architecture follows a modular component-based approach with clear separation of concerns:

- **Presentation Layer**: React components with Tailwind CSS styling
- **State Management**: React Context API for global state management
- **Security Layer**: Client-side cryptographic operations and secure key management
- **Communication Layer**: HTTP clients for API communication with the backend
- **Internationalization**: Multi-language support with i18next

The front-end implementation prioritizes client-side security, ensuring that sensitive information such as private keys and mnemonic phrases never leave the user's browser. This approach aligns with the zero-trust security model and provides users with complete control over their cryptographic materials.

5.2.2 Landing Page and User Onboarding

The landing page serves as the primary entry point for users, featuring a clean and professional design that communicates the platform's value proposition effectively. The page is structured to provide comprehensive information about BlockSign's capabilities while maintaining an elegant and accessible design.

Landing Page Design and Layout

The landing page follows modern web design principles with a focus on clarity, accessibility, and conversion optimization. The page is divided into several key sections that progressively introduce users to the platform's features and benefits.

The header section features the BlockSign logo and navigation menu, providing immediate access to registration and login functionality. The hero section contains a compelling value proposition with clear call-to-action buttons that guide users toward registration or learning more about the platform.

The design emphasizes trust and security through the use of professional typography, consistent color schemes, and carefully selected imagery that reinforces the platform's reliability and sophistication.

Feature Presentation

The landing page includes dedicated sections that highlight BlockSign's key features:

- **Passwordless Authentication**: Emphasis on modern security through cryptographic keys
- **Document Integrity**: Explanation of cryptographic verification and tamper-proof signatures
- **Multi-party Signing**: Collaborative document workflows with multiple participants
- **Regulatory Compliance**: Adherence to international digital signature standards
- **User Privacy**: Client-side cryptography and zero-trust architecture

Educational Content

The landing page serves an educational purpose by explaining digital signature concepts to users who may be unfamiliar with cryptographic authentication. This includes simplified explanations of:

- How digital signatures work and why they're more secure than traditional methods
- The benefits of passwordless authentication using mnemonic phrases
- Document integrity verification and tamper detection
- Legal validity and compliance aspects of digital signatures

Trust Indicators

The page incorporates various trust indicators to build user confidence:

- Security certifications and compliance badges
- Transparent explanation of the platform's security architecture
- Clear privacy policy and data handling practices

- Educational resources about digital signature technology

The user onboarding process is designed to be intuitive and secure, guiding users through each step of the registration and setup process with clear instructions and visual feedback. Progressive disclosure techniques are used to avoid overwhelming new users while ensuring they understand the security implications of their actions.

5.2.3 Registration Flow and User Identity

The registration process implements a multi-step verification system that ensures user authenticity while maintaining privacy. The flow begins with basic user information collection and email verification through OTP (One-Time Password) codes.

Users receive an email containing a six-digit verification code that must be entered to proceed with the registration (Figure 5.7).

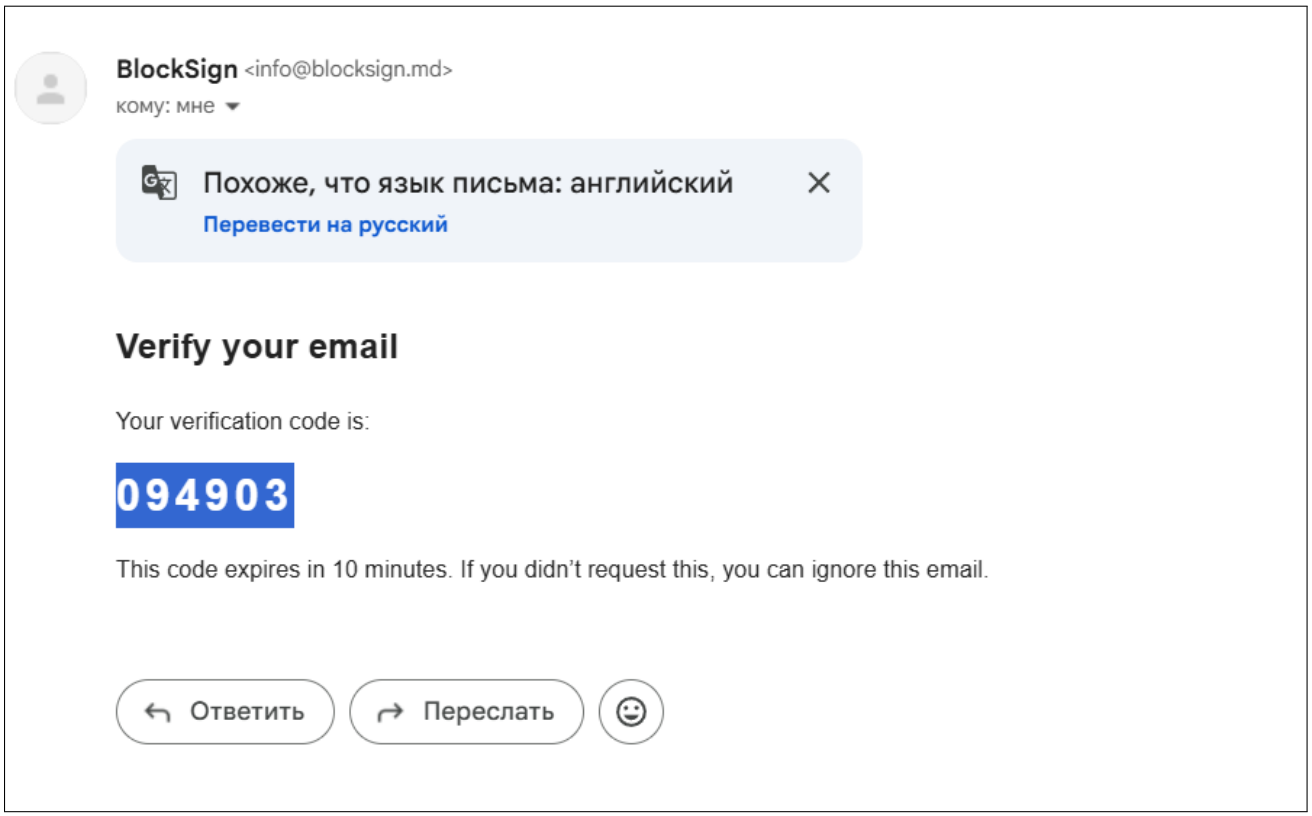

Figure 5.7 - Email Verification Code

The registration form captures essential user information including full name, email, phone number, and IDNP (personal identification number) for identity verification purposes (Figure 5.8).

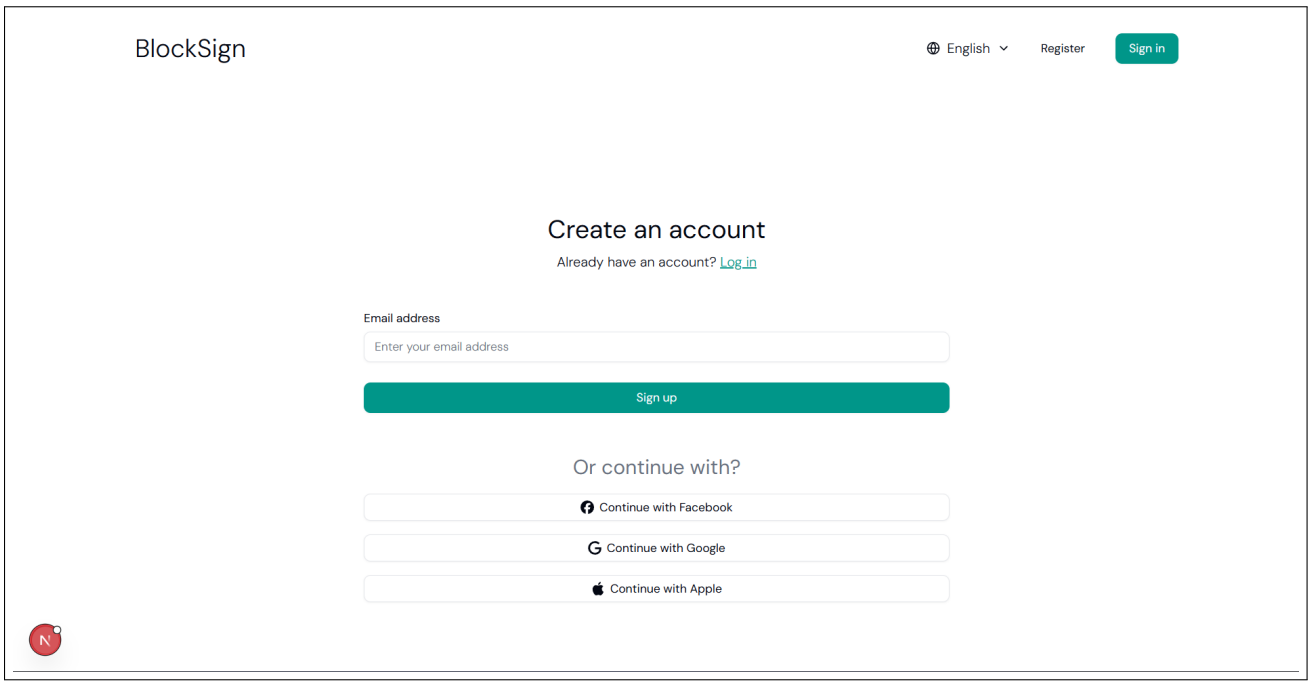

Figure 5.8 - User Registration Form

Upon successful submission, users see a confirmation page indicating that their registration request has been submitted for administrative review (Figure 5.9).

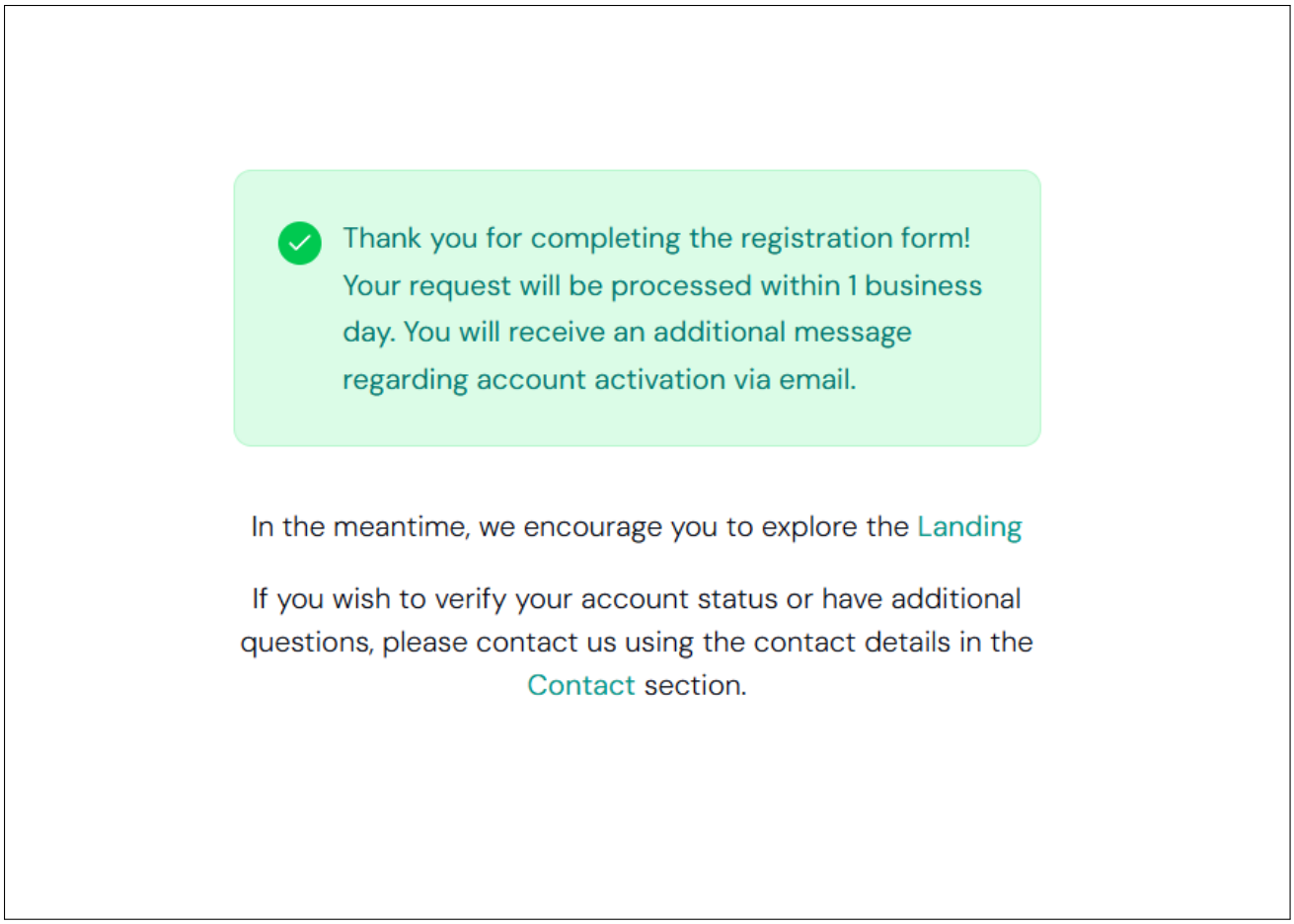

Figure 5.9 - Registration Success Confirmation

Once the administrator approves the registration request, users receive a notification email to complete their account setup.

5.2.4 Authentication System and Security Features

The authentication system implements a passwordless approach using cryptographic key pairs and mnemonic phrases. This section details the user interface components that facilitate secure authentication while maintaining usability.

Mnemonic-Based Authentication

After registration approval, users are guided through a secure key generation process. The system generates a 12-word mnemonic phrase that serves as the master key for the user's cryptographic identity. This approach ensures that users maintain complete control over their authentication credentials without relying on traditional password-based systems.

The mnemonic phrase generation interface provides clear instructions for users to securely store their recovery phrase (Figure 5.16). The interface emphasizes the importance of keeping the phrase secure and warns users about the consequences of losing access to it.

Users are presented with additional security options, including the ability to set up a PIN for convenient access while maintaining the security of the mnemonic phrase (Figure 5.17). This dual-layer approach balances security with user convenience.

PIN-Based Quick Access

For enhanced user experience, the system offers an optional PIN-based authentication method that allows users to access their accounts without entering the full mnemonic phrase for routine operations (Figure 5.18). The PIN is stored locally and encrypted using the user's public key, ensuring that it cannot be used without the corresponding private key.

Secure Login Interface

The login interface supports both mnemonic phrase authentication and PIN-based access. Users can choose their preferred authentication method based on their security preferences and usage patterns (Figure 5.10).

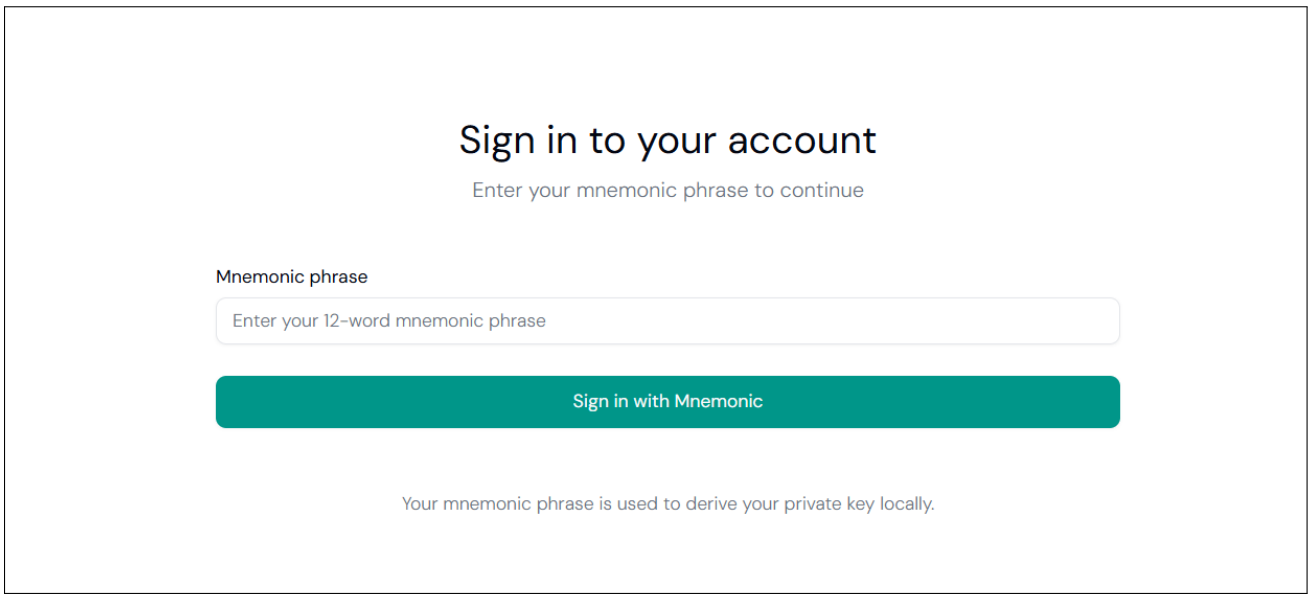

Figure 5.10 - Mnemonic-Based Login Interface

The interface provides clear feedback during the authentication process and includes security features such as automatic session timeout and secure token management.

5.2.5 User role specified UI

The authorized user is represented by the possibility to view his account info in the header as well as having acess to the authorized only pages (Figure 5.11).

Figure 5.11 - Header Account User

Also there is some difference for the admin header, it has additional button to navigate to the admin console (Figure 5.12).

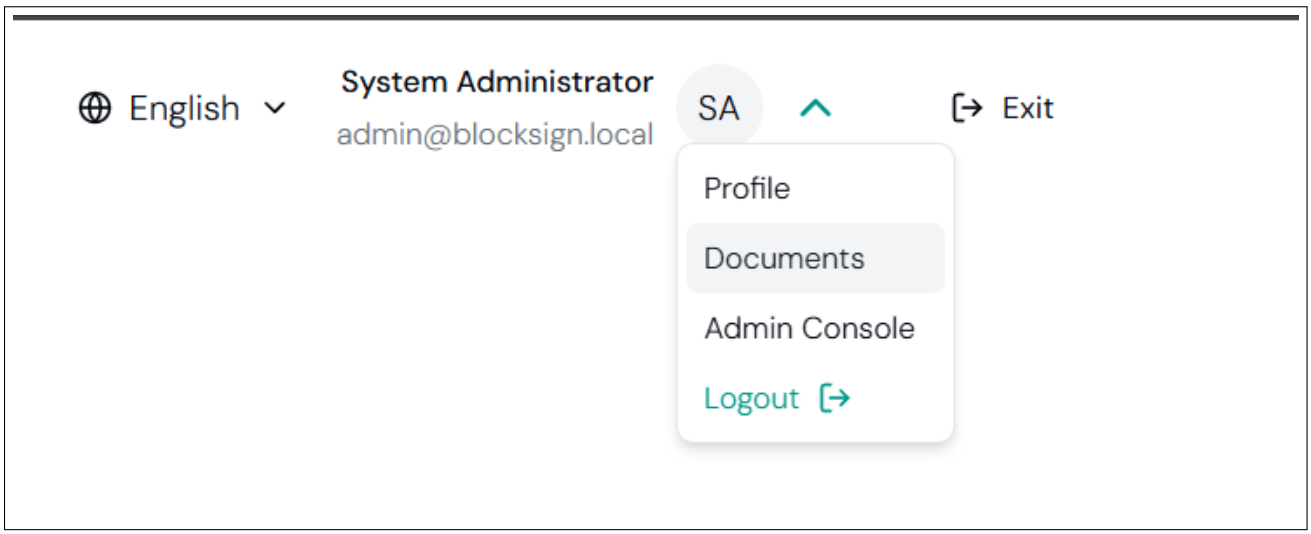

Figure 5.12 - Header Account Admin

Navigating to the admin console will lead to the following page (Figure 5.13):

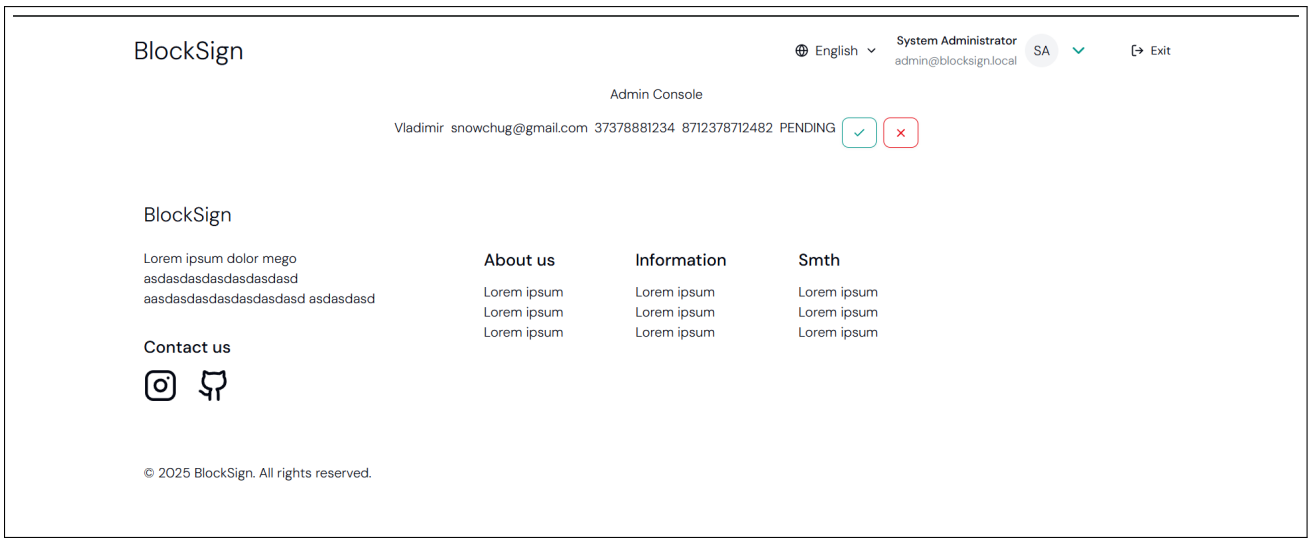

Figure 5.13 - Admin Console

On this page the admin can confirm or reject the registration requests. The requests are listed in the table with the following information:

- Full Name
- Email
- Phone
- idnp
- Action

If there are no requests, the admin will see the following page (Figure 5.14):

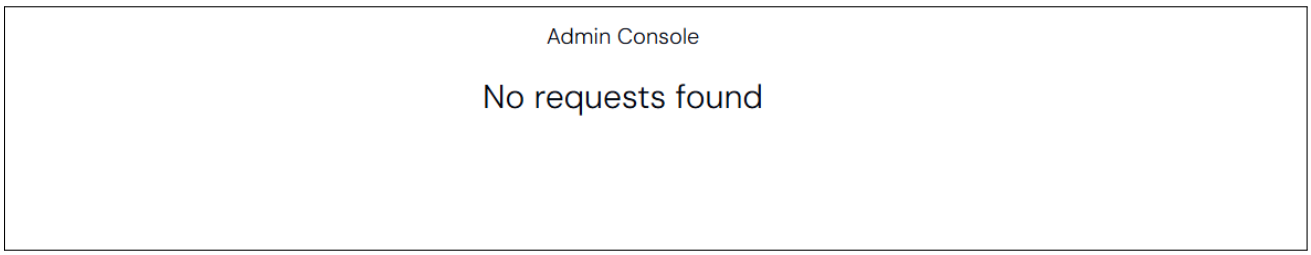

Figure 5.14 - Admin Console Empty

5.2.6 Additional registration steps

Then an email with the finish of the registration process is sent to the user (Figure 5.15).

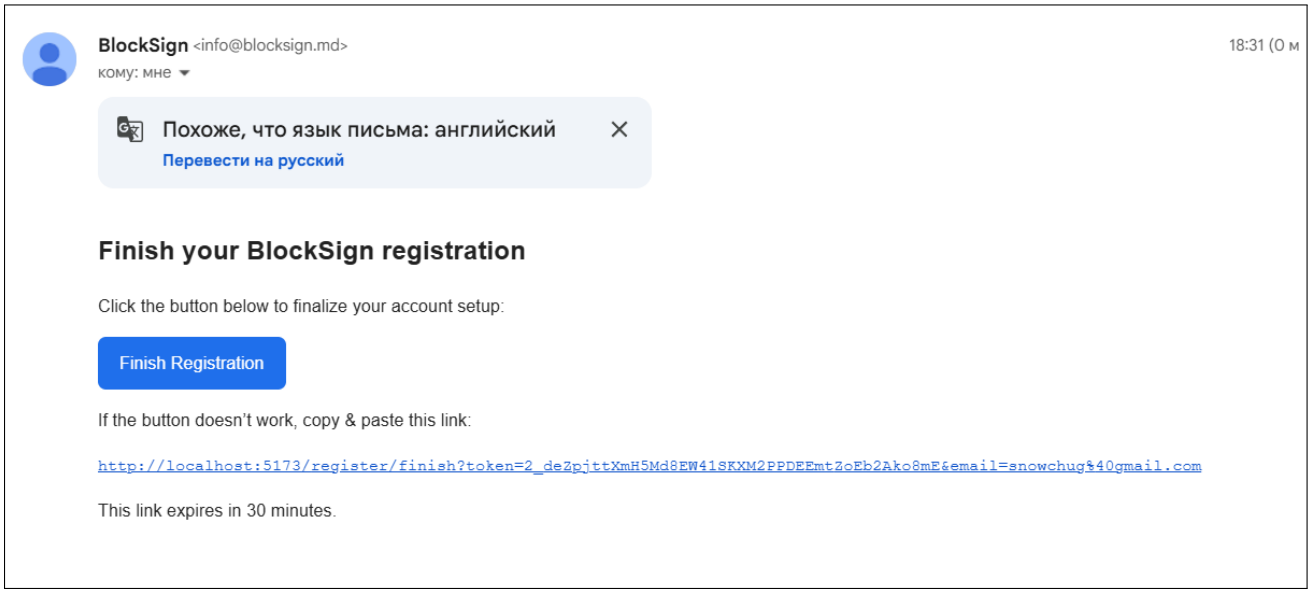

Figure 5.15 - Finish Registration Mail

Clicking the button or link will lead to the following page (Figure 5.16):

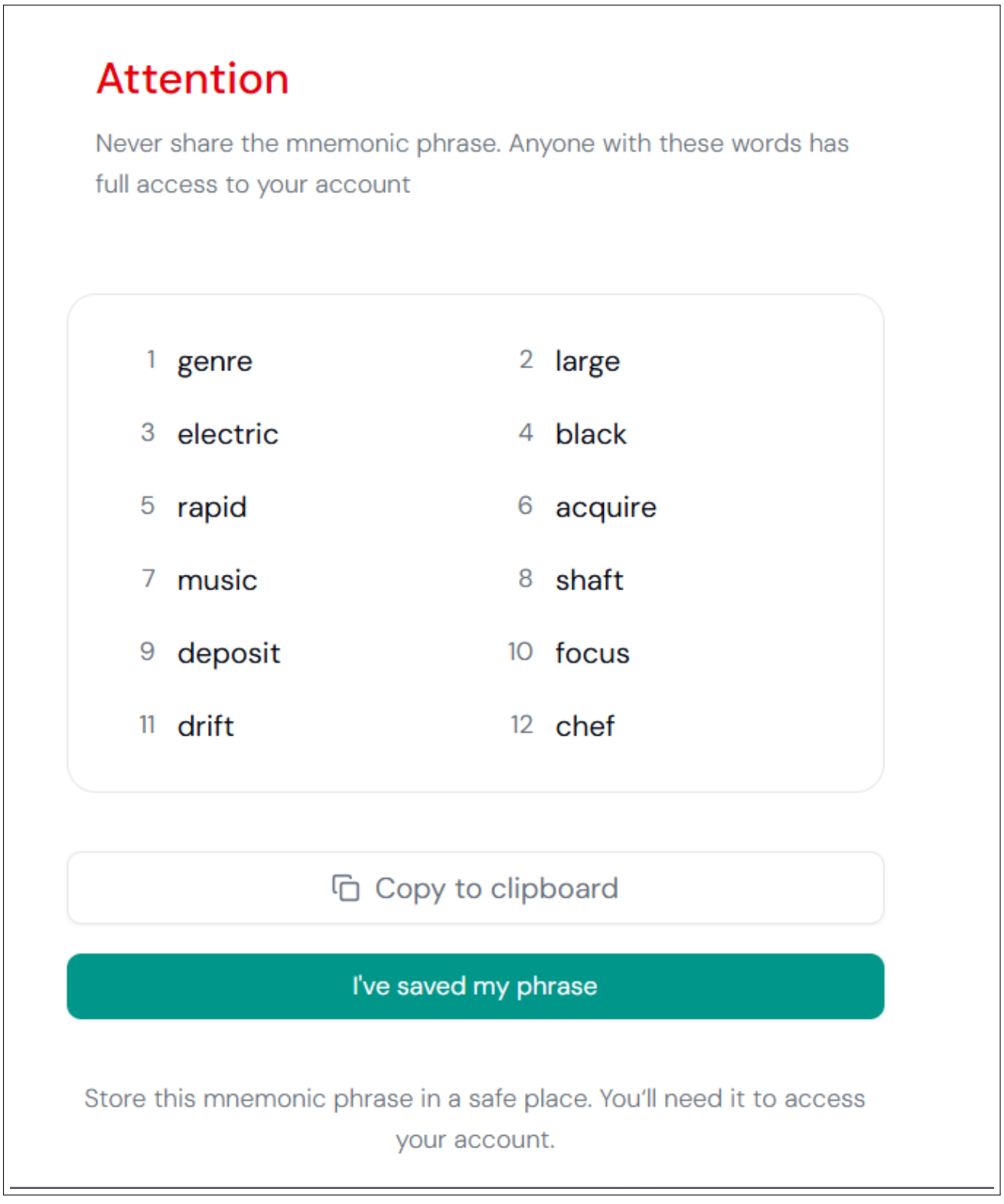

Figure 5.16 - Mnemonic Phrase

Here the user can see his mnemonic phrase and store it in a secure way (Figure 5.16). The phrase is generated on each refresh, but is valid only after clicking the button continue otherwise it is not valid anywhere. Then the user can choose to set a pin or remain only with mnemonic phrase (Figure 5.17).

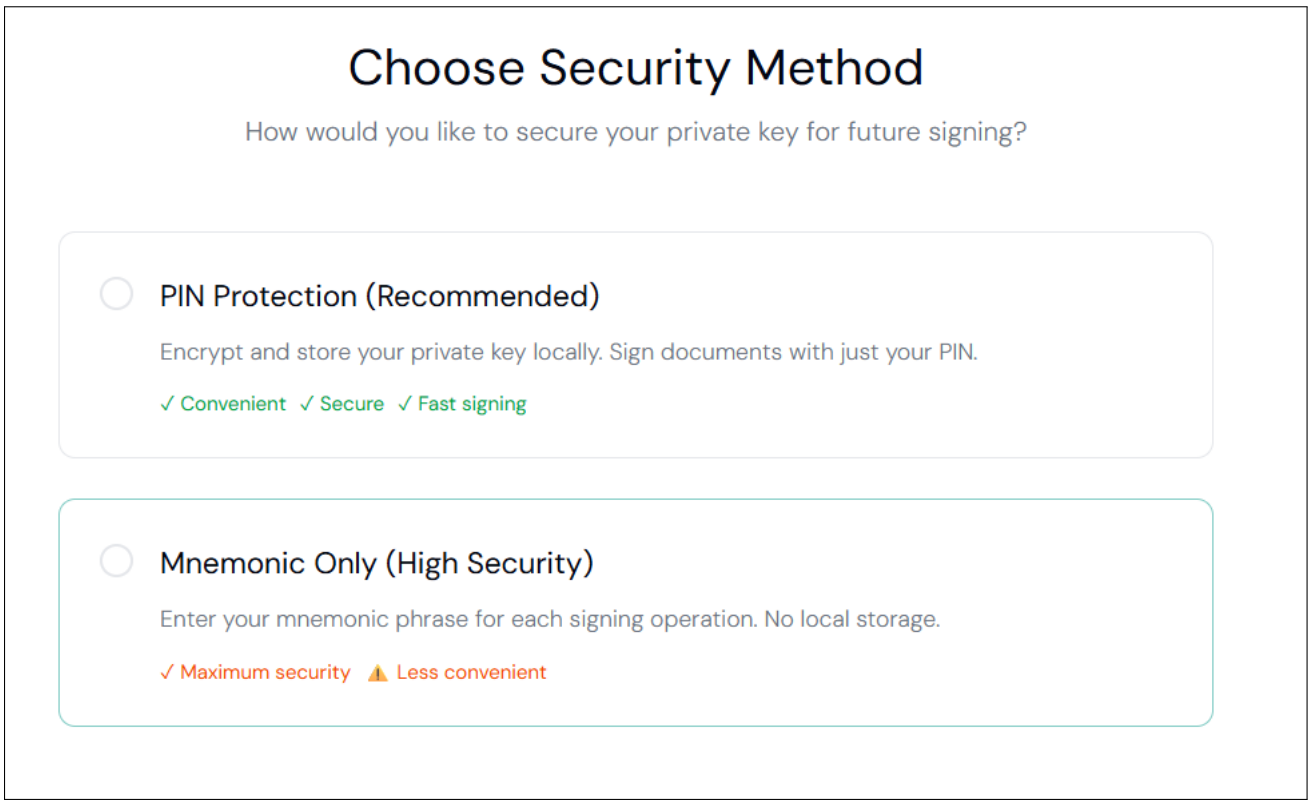

Figure 5.17 - Security Method Selection

Then the user can set a pin for the site (Figure 5.18). The pin is a shorter way to access the account and site functionality to avoid using the mnemonic phrase every time.

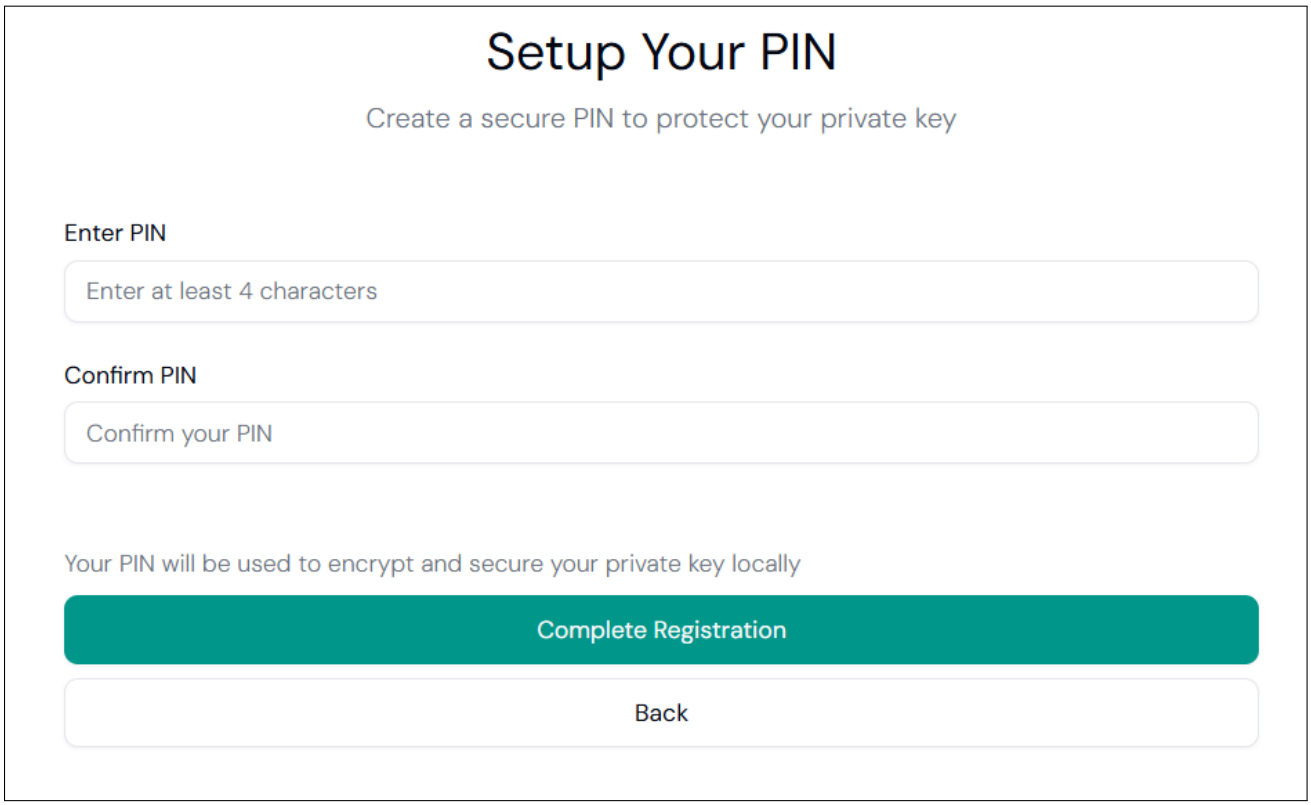

Figure 5.18 - PIN Setup for Convenient Access

5.2.7 Document Management Interface

The document management system provides users with comprehensive tools for uploading, signing, and managing digital documents. The interface is designed to streamline the document workflow while maintaining the highest security standards.

Dashboard and Navigation

Once authenticated, users are presented with a clean and intuitive dashboard that provides access to all document-related functionality. The dashboard displays recent document activity, pending signature requests, and quick access to document creation tools (Figure 5.19).

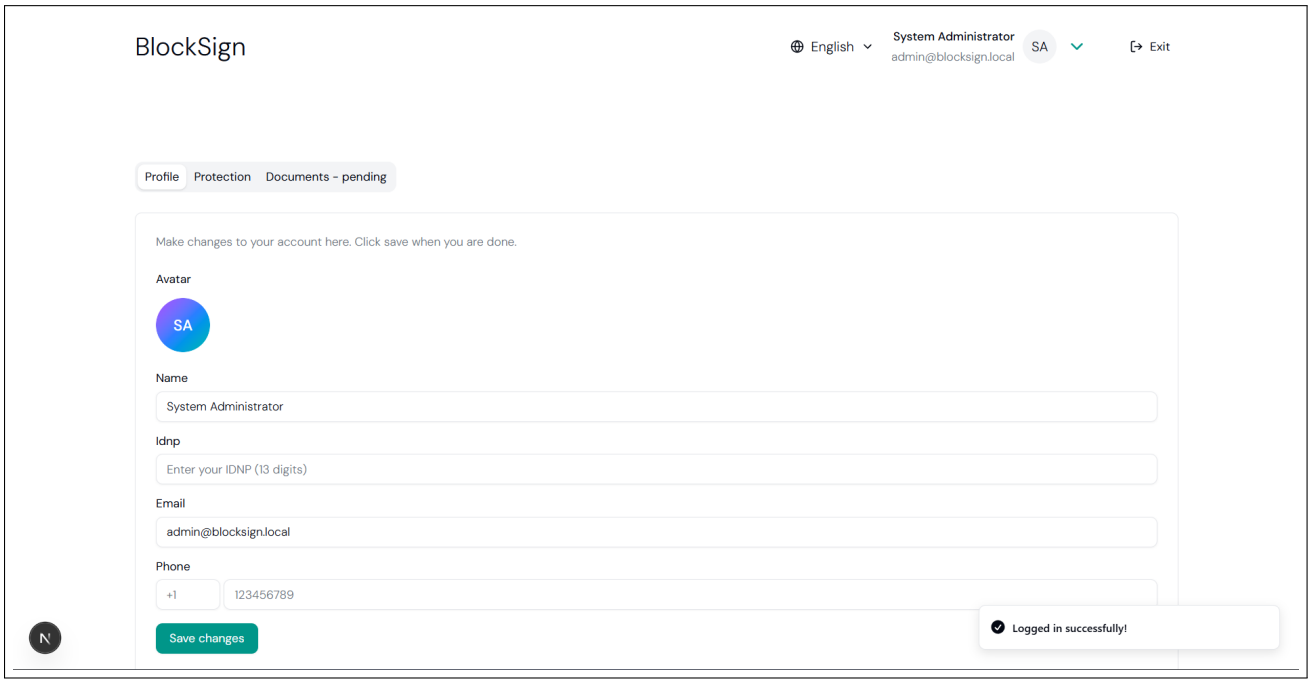

Figure 5.19 - User Dashboard After Login

The navigation system is designed to be intuitive, with clear visual indicators for different user roles and permissions. Authorized users can easily access their document library, create new documents, and manage their account settings.

Document Creation and Upload

The document creation process guides users through uploading PDF files and configuring signature requirements. The interface includes drag-and-drop functionality for easy file upload, along with validation to ensure only supported file formats are accepted.

Users can specify multiple participants for document signing, with each participant receiving notification emails containing the document and instructions for the signing process. The system automatically generates cryptographic hashes for document integrity verification.

Document Signing Workflow

The signing interface provides a secure environment for users to review and sign documents. Before signing, users can verify the document's integrity by comparing cryptographic hashes. The interface clearly displays document metadata, participant information, and signing status.

The signing process requires users to authenticate using their mnemonic phrase or PIN, ensuring that only authorized individuals can create valid signatures. Real-time feedback is provided throughout the signing process, including confirmation of successful signature creation and verification.

Document History and Verification

Users can access a comprehensive history of all documents they have created or signed. The interface provides detailed information about each document, including creation date, participants, signature status, and verification results.

The verification interface allows users to upload documents and verify their authenticity by checking cryptographic signatures and document integrity. This feature is available to both registered users and anonymous visitors, promoting transparency and trust in the digital signature process.

5.2.8 Responsive Design and User Experience

The BlockSign frontend implements a fully responsive design that adapts seamlessly to different screen sizes and devices. The application provides an optimal user experience across desktop computers, tablets, and mobile devices.

Mobile-First Approach

The design follows a mobile-first approach, ensuring that the core functionality is accessible and usable on smaller screens. Tailwind CSS breakpoints are used strategically to provide device-specific layouts and component arrangements.

Key responsive features include:

- Adaptive navigation menus that collapse on mobile devices
- Touch-friendly button sizes and interactive elements
- Optimized form layouts for mobile input
- Responsive image scaling and document previews
- Accessible typography and color contrast ratios

Cross-Browser Compatibility

The application is tested and optimized for modern web browsers, including Chrome, Firefox, Safari, and Edge. The codebase uses progressive enhancement techniques to ensure functionality across different browser versions while providing enhanced features for newer browsers.

5.2.9 Client-Side Security Implementation

The frontend architecture prioritizes security at every level, implementing multiple layers of protection for user data and cryptographic operations. All sensitive operations are performed client-side to minimize the exposure of private information.

Cryptographic Key Management

Private keys and mnemonic phrases are never transmitted to the server. All cryptographic operations, including key generation, signing, and verification, are performed within the user's browser using JavaScript cryptographic libraries.

The system implements secure storage mechanisms for sensitive data:

- Mnemonic phrases are stored temporarily in memory during active sessions
- PIN codes are encrypted using public keys before local storage
- Session tokens are managed securely with automatic expiration
- Private keys are derived from mnemonic phrases on-demand

Secure Communication

All communication between the frontend and backend uses HTTPS encryption. The application implements Content Security Policy (CSP) headers to prevent cross-site scripting attacks and other security vulnerabilities.

API requests are authenticated using JWT tokens with short expiration times, and refresh tokens are stored in secure HTTP-only cookies to prevent JavaScript-based attacks.

5.2.10 Internationalization and Localization

The application supports multiple languages to serve a diverse user base. The internationalization system uses i18next for dynamic language switching and locale-specific formatting.

Multi-Language Support

Currently supported languages include:

- English (default)
- Romanian
- Russian

The language selection interface allows users to switch between supported languages dynamically, with all text content, error messages, and user interface elements updating accordingly.

Localization Features

The localization system handles:

- Date and time formatting according to locale preferences
- Number formatting and currency display

- Text direction support for future RTL language integration
- Culturally appropriate messaging and terminology

5.2.11 Feature Accessibility by User Role

The BlockSign platform implements a role-based access control system that provides different levels of functionality based on user authentication status and administrative privileges.

Public Features (Unauthorized Users)

Unauthorized users have access to essential platform information and basic functionality without requiring registration:

- **Platform Information**: Access to landing page content, feature descriptions, and educational materials about digital signatures
- **User Registration**: Complete registration process including email verification and account request submission
- **Document Verification**: Upload and verify the authenticity of signed documents using cryptographic verification
- **Multi-language Support**: Switch between supported languages for better accessibility

Authenticated User Features

Registered and verified users gain access to comprehensive document management capabilities:

- **Document Creation**: Upload PDF documents and initiate signature workflows with multiple participants
- **Digital Signing**: Sign documents using cryptographic keys with mnemonic phrase or PIN authentication
- **Document History**: View complete history of created and signed documents with detailed metadata
- **Signature Management**: Track signature status for multi-party documents and receive notifications
- **Account Management**: Update profile information, manage security settings, and configure notification preferences
- **Participant Coordination**: Invite other users to sign documents and manage participant workflows
- **Document Verification**: Verify signatures on received documents and check document integrity
- **Secure Storage**: Access encrypted document metadata and signature records

5.2.12 Technology Integration and Performance

The frontend implementation leverages modern web technologies to provide optimal performance and user experience across different devices and network conditions.

Performance Optimization

Key performance features include:

- **Code Splitting**: Dynamic imports and lazy loading for reduced initial bundle size

- **Image Optimization**: Automatic image compression and format selection for faster loading
- **Caching Strategy**: Strategic use of browser caching for static assets and API responses
- **Progressive Loading**: Incremental content loading to improve perceived performance

CONCLUSIONS

The work carried out in this project demonstrates that secure, scalable, and verifiable document management can be achieved through the careful integration of modern cryptographic techniques, structured workflows, and practical user interfaces. By replacing traditional password-based authentication with Ed25519 key pairs and challenge–response mechanisms, the system eliminates a major vulnerability of centralized platforms while improving usability and security.

The design of the registration process, supported by email-based verification and administrative approval, ensures that only legitimate users gain access to the platform. At the same time, the introduction of digital signatures over canonical payloads provides a strong guarantee of document integrity and participant accountability. Each document is uniquely identified by its cryptographic hash, which allows all stakeholders to independently verify its authenticity.

From a functional perspective, the project delivers the core features of a minimum viable product (MVP): user registration, authentication, document creation, participant tagging, and the collection of signatures. Notifications via email strengthen the user experience by ensuring that participants remain informed throughout the signing process. Once all participants have provided their signatures, the system automatically transitions the document to a signed state, establishing a clear and auditable completion of the workflow.

The results confirm the feasibility of using lightweight cryptographic libraries and structured database schemas to implement secure document workflows without excessive infrastructure requirements. While the current MVP uses email for distribution of documents and notifications, the architecture is prepared for future integration with decentralized storage systems or blockchain anchoring, which would provide long-term transparency and persistence.

In conclusion, this project achieves its primary objective of creating a secure system for passwordless user authentication and document signing. It demonstrates the viability of combining cryptographic security with practical user workflows, and it establishes a foundation for further research and development. Future improvements may include integration with external identity providers, support for blockchain-based anchoring, advanced role-based permissions, and the addition of seed-phrase-based account recovery mechanisms on the client side. These directions would enhance resilience, usability, and trust in the system, preparing it for deployment in real-world organizational contexts.