Digital Signature Verification System

CSC 675-775-01 Summer 2025

Matthew Martinez
Student ID: 923707132
GitHub: matthewmartinez-java

Milestone	**Date Submitted**
M1	06/16/2025

Table of Contents

Project Description

Increasing dependence on digital communication and electronic documents, in a remote work setting, has made verification of document authenticity more necessary than ever. In sectors such as healthcare and legal services, unauthorized tampering or impersonations can result in serious consequences like data breaches and fraud. The motivation behind this project is to develop a safe Digital Signature Verification System that assists organizations in securing their workflow of electronic documents by providing secure, auditable digital signature verification.

At a high level, this database system will serve as the backbone of a digital signature verification service. It will store and manage user information, digital certificates, signed documents, audit logs, and signature validation records. The system will allow users to upload documents with attached digital signatures and verify their authenticity in real time. Factors like mismatches, expirations, or evidence of document tampering will result in a validation failure, and the system will automatically flag the issue and log it for further examination.

This database system will contain various unique features on the basis of a cybersecurity mindset and approach. Most importantly, it will:

- Support Public Key Infrastructure (PKI) integration for digital certificates/keys
- Offer role-based access control (RBAC) to protect sensitive data and operations
- Include timestamping and event logging for auditing and analysis

These features build upon existing systems by offering finer-grained access controls, transparency of the verification process, and detailed logging, which is common practice in a cybersecurity environment.

Two existing products that would benefit from using this database system are:

1. *DocuSign* - DocuSign currently has electronic signatures, but it would benefit from a customizable, internally-hosted database approach that allows for greater control over how verification events are stored and accessed. For example, firms with demanding data residency would be able to put this system in place to meet compliance without depending on a third party SaaS platform.
2. *Adobe Acrobat Sign* - Acrobat Sign can expand its current PKI functionality with stronger transparent audit trails and RBAC security by using this database system as a backend module. This is especially beneficial in industries like healthcare where secure logging is required for regulatory compliance.

Functional Database Requirements

1. **User**
    1.1. A user shall create at most one account
    1.2. A user shall have a unique user ID
    1.3. A user shall be able to register using an email and password
    1.4. A user shall have a role assigned via role entity
    1.5. A user shall be linked to multiple sessions
    1.6. A user without an account shall not be allowed to verify documents
    1.7. A user shall verify their email address before gaining access to the system
    1.8. A user shall have the ability to reset passwords via a secure token-based system
    1.9. A user shall be locked out after 5 failed login attempts within 15 minutes
    1.10. A user shall be alerted via notification if account activity is detected on a new device

2. **Registered User**
    2.1. A registered user is also a user
    2.2. A registered user shall have exactly one account
    2.3. A registered user shall be able to upload and sign multiple documents
    2.4. A registered user shall view verification history
    2.5. A registered user shall be able to revoke their own signatures
    2.6. A registered user shall be assigned to one organization

3. **Admin**
    3.1. An admin is also a registered user
    3.2. An admin shall manage all access control entities
    3.3. An admin shall revoke certificates and monitor audit logs
    3.4. An admin shall promote or demote other users to admin level
    3.5. An admin shall trigger emergency shutdown or lockdown of the system
    3.6. An admin shall be able to create and archive organization profiles

4. **Role**
    4.1. A role shall be linked to many users
    4.2. A role shall have a unique role ID
    4.3. A role shall define access permissions
    4.4. A role shall be customizable by the admin
    4.5. A role shall include a descriptive title

5. **Account**
    5.1. An account shall be associated with one user only
    5.2. An account shall have a unique account ID

5.3. An account shall record the creation timestamp
5.4. An account shall include a status field
5.5. An account shall be updated only by the user or an admin

6. Digital Certificate

6.1. A certificate shall be assigned to one user
6.2. A certificate shall have an expiration and issue date
6.3. A certificate shall be issued by one certificate authority
6.4. A certificate shall include the public key
6.5. A certificate shall be revocable by an admin
6.6. A certificate shall be linked to multiple signatures
6.7. A certificate shall be validated against Certificate Revocation Lists
6.8. A digital certificate shall be exportable in PEM format by the owning user
6.9. A certificate shall include a fingerprint (SHA-256) for integrity verification

7. Certificate Authority

7.1. A certificate authority shall issue many certificates
7.2. A certificate authority shall have a unique ID and public key
7.3. A certificate authority shall have a verified status
7.4. A certificate authority shall be associated with an issuing organization
7.5. A certificate authority shall store revocation timestamps for invalid certificates

8. Signature

8.1. A signature shall be linked to one document
8.2. A signature shall be generated using a specific certificate
8.3. A signature shall store the hash of signed content
8.4. A signature shall include a timestamp and status
8.5. A signature shall be associated with a user
8.6. A signature may be a part of a recursive signature chain

9. Document

9.1. A document shall have a unique ID
9.2. A document shall be uploaded by one user
9.3. A document shall have one or more signatures
9.4. A document may be shared with multiple users
9.5. A document shall link to verification events
9.6. A document shall contain metadata

10. Verification Event

10.1. A verification event shall be triggered by a user
10.2. A verification event shall include status

10.3. A verification event shall generate an audit log

10.4. A verification event shall record document ID and timestamp

10.5. A verification event may trigger a notification

11. Audit Log

11.1. An audit log shall record every verification event

11.2. An audit log shall include timestamp, user ID, IP address, and result

11.3. An audit log shall be immutable once written

11.4. An audit log shall support compliance inspection and export

11.5. An audit log may be linked to a signature revocation event

11.6. An audit log shall support encrypted hashing to detect tampering

11.7. An audit log shall be linked to the specific action or API call that triggered it

11.8. An audit log shall differentiate between successful and failed actions

12. Signature Revocation

12.1. A revocation shall be linked to one signature

12.2. A revocation shall store the reason for revocation

12.3. A revocation shall be time stamped and recorded in audit logs

12.4. A revocation shall be performed by users with revocation privileges

12.5. A revocation shall invalidate the associated certificate in the database

13. Access Control Entry

13.1. An access entry shall define permissions

13.2. An access entry shall be assigned to one user and one document

13.3. An access entry shall be editable only by an admin

13.4. An access entry shall define an access start and end period

14. Public Key

14.1. A public key shall be part of one certificate

14.2. A public key shall be uniquely stored using encrypted fields

14.3. A public key shall be used during the verification process

14.4. A public key shall support validation using hash comparison

15. Private Key

15.1. A private key shall be stored using industry-standard encryption

15.2. A private key shall never be directly queried

15.3. A private key shall be rotated on a schedule

15.4. A private key shall be associated with only one user

15.5. A private key shall be inaccessible through any API endpoints

15.6. A private key shall require MFA authentication for re-issue after deletion

16. Notification

16.1. A notification shall be generated for failed verification events
16.2. A notification shall be linked to one user and one document
16.3. A notification shall include content, timestamp, and urgency level
16.4. A notification shall support read/unread status tracking

17. Organization

17.1. An organization shall have a unique ID and name
17.2. An organization shall contain multiple users
17.3. An organization shall be associated with documents and certificate authorities
17.4. An organization shall have a designated admin contact

18. Session

18.1. A session shall be started at every login event
18.2. A session shall track login time, logout time, and IP address
18.3. A session shall be associated with one user
18.4. A session shall be automatically expired after timeout or logout

19. Hash Record

19.1. A hash record shall store a cryptographic hash of each document
19.2. A hash record shall be linked to a document ID and verification result
19.3. A hash record shall be used to detect tampering
19.4. A hash record shall be compared during signature verification
19.5. A hash record shall be calculated using SHA-256 or stronger algorithms
19.6. A hash record shall be immutable after the document is signed
19.7. A hash record mismatch during verification shall trigger an audit log alert

20. Device

20.1. A device shall be registered to a user for signature activity
20.2. A device shall store device fingerprinting metadata
20.3. A device shall be logged during session creation
20.4. A device shall store trust level status
20.5. A device shall require re-verification if flagged or inactive for 30+ days
20.6. A device shall be eligible for two-factor authentication binding
20.7. A device shall be deletable only by the user or an admin

21. IP Address Log

21.1. An IP address log shall track every user interaction
21.2. An IP log shall link to a session and verification event
21.3. An IP log shall be used for geo-location analysis
21.4. An IP log shall support compliance with access policies

Non-Functional Database Requirements

1. **Performance**
   1.1. The database system shall support concurrent transactions for multiple users without race conditions
   1.2. The system shall return query results for digital signature validation in under 2 seconds 95% of the time
   1.3. Indexes shall be created on frequently queried fields such as ‘user_id’, ‘document_id’, and ‘certificate_id’ to optimize search performance
   1.4. The system shall process and log at least 100 verification requests per minute during peak hours

2. **Security**
   2.1. Only hashed and salted passwords shall be stored using secure algorithms like bcrypt
   2.2. All connections to the database shall use TLS 1.2 or higher for encryption in transit
   2.3. The database shall implement role-based access control (RBAC) at the query and table level
   2.4. The system shall log all failed authentication attempts with timestamps and originating IP

3. **Scalability**
   3.1. The database system shall support horizontal scaling of read replicas to manage growing query loads
   3.2. The system shall be designed to handle at least a 10x increase in users and stored documents without performance degradation
   3.3. Database tables such as ‘audit_log’, ‘session’, and ‘verification_event’ shall support partitioning by time or user ID for scalability

4. **Capability**
   4.1. The system shall support document verification against both self-signed and certificate authority-issued certificates
   4.2. The system shall be able to integrate with third-party APIs for certificate validation
   4.3. The database shall support time-based queries to enable efficient log and session analysis

5. **Environmental**
   5.1. The database system shall run on a Linux-based cloud environment (Ubuntu on AWS EC2)

5.2. The system shall consume less than 500 MB of RAM under normal load conditions

5.3. The system shall be deployable in a containerized environment using Docker

5.4. The system shall operate reliably in environments with intermittent internet by caching critical data locally

6. Coding Standard

6.1. SQL queries and database schema scripts shall follow the ANSI SQL standard

6.2. All code shall follow naming conventions for clarity

6.3. All queries shall be parameterized to prevent SQL injection

6.4. Database triggers and stored procedures shall be version controlled and commented clearly

7. Storage

7.1. The database system shall support persistent storage for all core entities

7.2. Each table shall have a disk quota policy and alert administrators when 80% of capacity is reached

7.3. The system shall support automatic archival of records older than 5 years

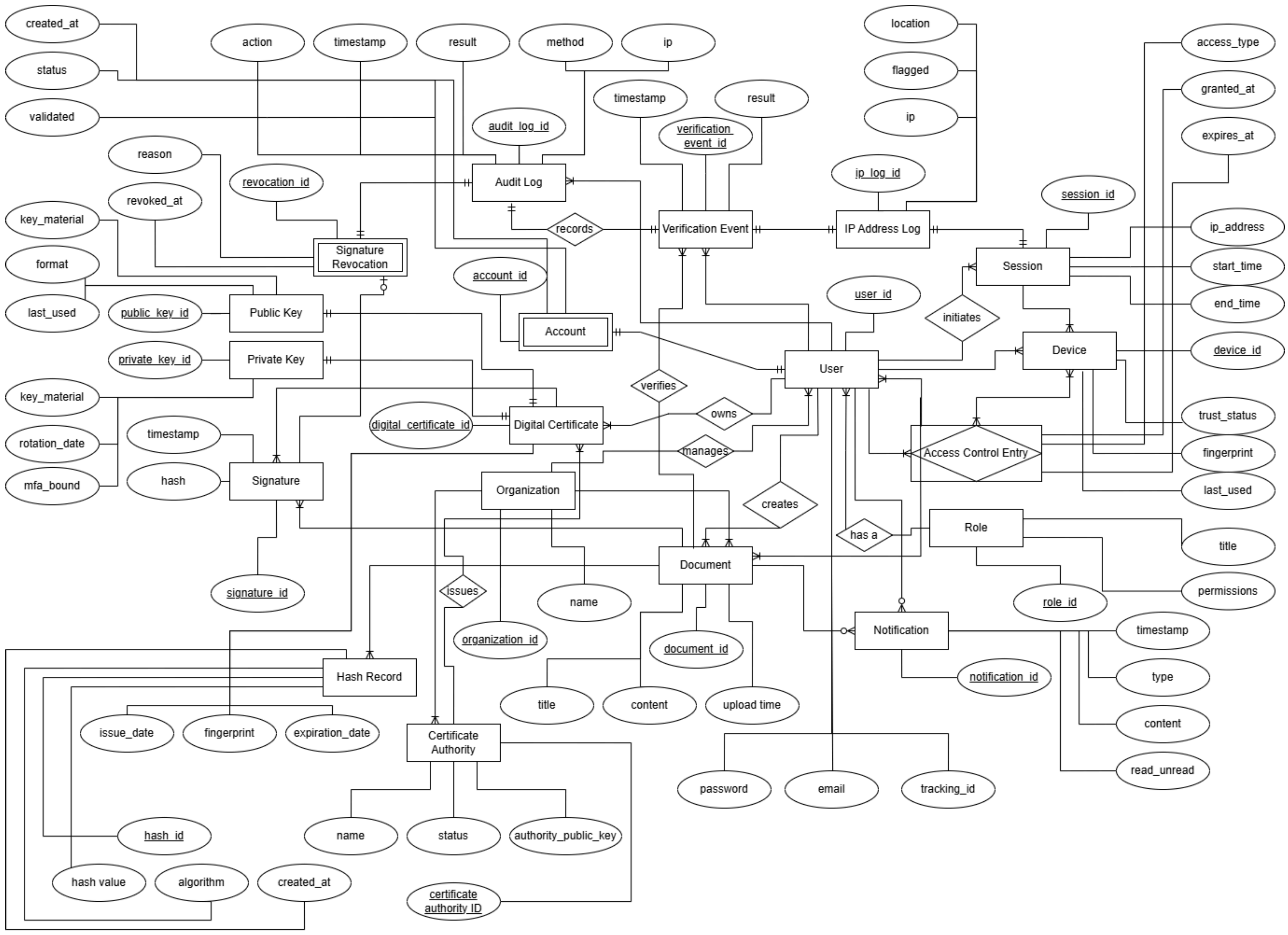
7.4. Uploaded documents shall be stored in a dedicated, encrypted file system or object storage layer

8. Privacy

8.1. Personally identifiable information (PII) such as email and IP address shall be stored encrypted at rest

8.2. The system shall restrict access to sensitive fields based on user roles

8.3. Privacy actions such as document sharing or verification attempts shall require user consent where applicable


created_at
status
validated
action
timestamp
result
method
ip
audit_log_id
Audit Log
reason
revocation_id
revoked_at
Signature Revocation
key_material
format
last_used
public_key_id
Public Key
private_key_id
Private Key
key_material
rotation_date
mfa_bound
timestamp
hash
Signature
signature_id
records
timestamp
result
verification event_id
Verification Event
location
flagged
ip
ip_log_id
IP Address Log
session_id
Session
access_type
granted_at
expires_at
ip_address
start_time
end_time
account_id
Account
user_id
initiates
User
Device
device_id
verifies
owns
manages
digital_certificate_id
Digital Certificate
trust_status
fingerprint
last_used
Access Control Entry
Organization
creates
Role
has a
title
permissions
role_id
Document
issues
name
organization_id
document_id
Notification
notification_id
timestamp
type
content
read_unread
title
content
upload time
Hash Record
issue_date
fingerprint
expiration_date
Certificate Authority
name
status
authority_public_key
certificate authority ID
hash_id
hash value
algorithm
created_at
password
email
tracking_id

Entity Set Description

1. **User (Strong)**
    a. Key Attributes
        i. user_id: numeric, primary key
    b. Non-Key Attributes
        i. email: alphanumeric, unique key
        ii. password: alphanumeric, hashed
        iii. tracking_id: numeric, unique key
        iv. role_id: foreign key (references Role)
        v. organization_id: numeric, foreign key (references Organization)

2. **Account (Weak)**
    a. Key Attributes
        i. account_id: numeric, primary key
        ii. user_id: numeric, foreign key
    b. Non-Key Attributes
        i. created_at: datetime
        ii. status: alphanumeric (active, suspended, etc.)
        iii. validated: boolean

3. **Role (Strong)**
    a. Key Attributes
        i. role_id: numeric, primary key
    b. Non-Key Attributes
        i. title: alphanumeric
        ii. permissions: alphanumeric

4. **Document (Strong)**
    a. Key Attributes
        i. document_id: numeric, primary key
    b. Non-Key Attributes
        i. title: alphanumeric
        ii. content: text (encrypted)
        iii. upload_time: datetime
        iv. organization_id: numeric, foreign key

5. **Signature (Strong)**
    a. Key Attributes
        i. signature_id: numeric, primary key
    b. Non-Key Attributes

      i. hash: alphanumeric
      ii. timestamp: datetime
      iii. digital_certificate_id: numeric, foreign key

6. **Signature Revocation(Weak)**
   a. Key Attributes
      i. revocation_id: numeric, primary key
   b. Non-Key Attributes
      i. reason: text
      ii. revoked_at: datetime
      iii. signature_id: numeric, foreign key (references signature)

7. **Digital Certificate (Strong)**
   a. Key Attributes
      i. digital_certificate_id: numeric, primary key
   b. Non-Key Attributes
      i. issue_date: datetime
      ii. expiration_date: datetime
      iii. fingerprint: alphanumeric
      iv. user_id: numeric, foreign key

8. **Access Control Entry (Associative)**
   a. Key Attributes
      i. user_id: numeric, primary key and foreign key
      ii. document_id: numeric, primary key and foreign key
   b. Non-Key Attributes
      i. access_type: alphanumeric (read, write, revoke)
      ii. granted_at: datetime
      iii. expires_at: datetime

9. **Session (Strong)**
   a. Key Attributes
      i. session_id: numeric, primary key
   b. Non-Key Attributes
      i. user_id: numeric, foreign key (references User)
      ii. start_time: datetime
      iii. end_time: datetime
      iv. ip_address: alphanumeric

10. **Device (Strong)**
   a. Key Attributes

        i. device_id: numeric, primary key

    b. Non-Key Attributes
        i. user_id: numeric, foreign key (references User)
        ii. trust_status: alphanumeric (trusted, untrusted, flagged)
        iii. fingerprint: alphanumeric
        iv. last_used: datetime

11. Notification (Strong)

a. Key Attributes
    i. notification_id: numeric, primary key

b. Non-Key Attributes
    i. user_id: numeric, foreign key (references User)
    ii. document_id: numeric, foreign key (references Document)
    iii. timestamp: datetime
    iv. type: alphanumeric (error, alert, success)
    v. content: text
    vi. read_unread: boolean

12. Audit Log (Strong)

a. Key Attributes
    i. audit_log_id: numeric, primary key

b. Non-Key Attributes
    i. user_id: numeric, foreign key (references User)
    ii. verification_event_id: numeric, foreign key (references VerificationEvent)
    iii. action: alphanumeric
    iv. timestamp: datetime
    v. result: alphanumeric (success, failure)
    vi. method: alphanumeric (manual, auto)
    vii. ip: alphanumeric

13. Verification Event (Strong)

a. Key Attributes
    i. verification_event_id: numeric, primary key

b. Non-Key Attributes
    i. user_id: numeric, foreign key (references User)
    ii. document_id: numeric, foreign key (references Document)
    iii. timestamp: datetime
    iv. result: alphanumeric
    v. audit_log_id: numeric, foreign key (references Audit Log)

14. IP Address Log (Strong)

a. Key Attributes
    i. ip_log_id: numeric, primary key
b. Non-Key Attributes
    i. session_id: numeric, foreign key (references Session)
    ii. verification_event_id: numeric, foreign key (references VerificationEvent)
    iii. ip: alphanumeric
    iv. location: alphanumeric
    v. flagged: boolean

15. Public Key (Strong)

a. Key Attributes
    i. public_key_id: numeric, primary key
b. Non-Key Attributes
    i. digital_certificate_id: numeric, foreign key (references Digital Certificate)
    ii. key_material: alphanumeric
    iii. format: alphanumeric
    iv. last_used: datetime

16. Private Key (Strong)

a. Key Attributes
    i. private_key_id: numeric, primary key
b. Non-Key Attributes
    i. digital_certificate_id: numeric, foreign key (references Digital Certificate)
    ii. key_material: alphanumeric (encrypted)
    iii. rotation_date: datetime
    iv. mfa_bound: boolean

17. Organization (Strong)

a. Key Attributes
    i. organization_id: numeric, primary key
b. Non-Key Attributes
    i. name: alphanumeric

18. Hash Record (Strong)

a. Key Attributes
    i. hash_id: numeric, primary key
b. Non-Key Attributes
    i. document_id: foreign key (references Document)
    ii. hash_value: alphanumeric
    iii. algorithm: alphanumeric (SHA-256, SHA-3, etc.)
    iv. created_at: datetime

19. Certificate Authority (Strong)

a. Key Attributes
   i. certificate_authority_id: numeric, primary key
b. Non-Key Attributes
   i. name: alphanumeric
   ii. authority_public_key: alphanumeric
   iii. organization_id: numeric, foreign key (references Organization)
   iv. status: alphanumeric (active, revoked)