TRUST-CENTRIC PRIVACY-PRESERVING BLOCKCHAIN BASED DIGITAL CERTIFICATE LOCKER

Abstract

Millions of students complete their education each year and go on to do higher studies or a corporate job. In this case student credentials are verified through a lengthy document verification process. This results in significant overhead as documents are transferred between institutions for verification. It is a costly, lengthy, and time-consuming procedure as university authorities invest millions of dollars in maintaining the entire process each year. The employer also takes plenty of time to verify the authenticity of the applicant’s and applicant's certificate. People frequently lie about their degrees and qualifications by counterfeiting certificates. A fake certificate generated by skilful scammers is always tough to identify and address as the original one. Therefore, there is a crucial need to upgrade the certification and verification process. Blockchain has recently emerged as a potential alternative to manual student verification process. This project introduced a Blockchain-based decentralized Student Verification platform that offers an easy way to issue, check, and verify educational certificates. The student's identity and document are both verified by matching the hashes already present in the Blockchain. Also, in the proposed method the documents are linked to the student to add another layer of verification. The implementation of this proposed platform can be used to issue, receive and verify the student and their certificates. This system will help students as well as institutions to maintain security and transparency at the same time.

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1. OVERVIEW

Education certificate verifications are a valuable tool in pre-employment background checks, since they confirm whether or not a candidate has earned the diploma or degree claimed, hence highlighting a candidate's qualifications and possibly revealing information about your candidate's honesty and integrity. An Education Verification search confirms the education, degree, training, or certification claims of a candidate are true and identifies potential discrepancies before you hire. Sometimes referred to as an Education Background Check or an Education Check, this service is used to confirm educational experience at high schools, universities, colleges. To prevent tampering or reproduction by copier machines, most of the genuine educational institutions will have some physical authentication features such as micro-text lines, UV invisible ink, watermark, security hologram, anti-scanning ink, etc. Most probably, fake degree certificate sellers may not put a fake watermark on their fake degrees to give them a real look. The security hologram, Anti-Scanning Ink, and void features provide an additional feature of anti-scanning and prevent these from making a colour replica. If scanned or photocopied, the matter/design would be far different than the original colour. In case of a void feature, the word COPY appears when an attempt is made to copy a degree. This feature will not be seen in the original document. However, if photocopied, the feature appears on duplicate copy.

1.1 Background of the Project

In this modern age, computers have verified the cause of their existence. The advent of computers in our society caused a lot of criticism on the danger it poses on the society. Critics of computer and new techniques express their fear on how computers will displace and replace all human skills thus resulting to mass unemployment. The presence of computer on virtually every field of today’s fast life has proved the critic wrong as the invention of computers and new technologies continues to create additional jobs for those who identify themselves with computers and new technologies. This makes computes partner to human beings in any fields of human endeavour. Over the past decades, students' identification and verification has been a major problem in large institutions as documents, certificate and studentship can be forged at a great rate and easy way, using the computer negatively. Forgers fail to know that in this fast-moving world, the computer has equally served as an aid to decision making, verification and authentication.

This is because of computers efficiency in terms of speed, accuracy, reliability, cost and security among others. In recent years, awareness has existed and created in people as it concerned the use of computer in verifying, authenticating and security activities either through web-based (internet) or window-based. Decision support system (DSS) and others system like management information system (MIS) and information system (IS) are used in organization.

Decisions are largely based on experience and principle. The value of every degree is the reputation of the institution and the students produced, hence, the greatest obstacle to any academic institution which is dishonesty and forgery has to be balanced with verification and authentication systems and processes.

1.2. Problem Statement

An important step in job recruiting is to go through résumés and job applications and check if they contain incorrect or fraudulent information. In fact, hiring the wrong person can be an expensive torment and may be extremely disruptive. Providentially, verification of the genuineness of a certificate or mark list issued is possible over using a proposed certificate verification system. Thus, most major companies use background checks as part of the standard practice for screening potential employees. Such measure "should be used within every organization that has serious consequences associated with security breaches," and more penetratingly for employees in highly sensitive or trusted positions. As opposed to a background check, which only looks into a person's past, honesty/integrity tests have been utilized to assist in the hiring process since the middle of the last century. Such assessments are considered hypothetically useful supplements to the standard background check as they help identify and select better workers and accordingly "improve the quality of an organization's overall performance". Prevention, detection and deterrence of fraud qualifications are crucial in making well informed hiring decisions. Academic institutions must use anti-counterfeit technologies with blockchain that make illegal copying or reproduction difficult and provide a means to verify legitimate credentials. Furthermore, the issuance of these credentials must be tightly controlled by each educational institution so that electronic verification can be easily conducted on credentials presented for employment.

1.3. Blockchain Technology

Blockchain is defined as a ledger of decentralized data that is securely shared. Blockchain technology enables a collective group of select participants to share data. With blockchain cloud services, transactional data from multiple sources can be easily collected, integrated, and shared. Data is broken up into shared blocks that are chained together with unique identifiers in the form of cryptographic hashes. Blockchain provides data integrity with a single source of truth, eliminating data duplication and increasing security. In a blockchain system, fraud and data tampering are prevented because data can't be altered without the permission of a quorum of the parties. A blockchain ledger can be shared, but not altered. If someone tries to alter data, all participants will be alerted and will know who make the attempt.

Decentralized trust:

The key reason that organizations use blockchain technology, instead of other data stores, is to provide a guarantee of data integrity without relying on a central authority. This is called decentralized trust through reliable data.

Blockchain blocks:

The name blockchain comes from the fact that the data is stored in blocks, and each block is connected to the previous block, making up a chainlike structure. With blockchain technology, you can only add (append) new blocks to a blockchain. You can't modify or delete any block after it gets added to the blockchain.

Consensus algorithms:

Algorithms that enforce the rules within a blockchain system. Once the participating parties set up rules for the blockchain, the consensus algorithm ensures that those rules are followed.

Blockchain nodes:

Blockchain blocks of data are stored on nodes—the storage units that keep the data in sync or up to date. Any node can quickly determine if any block has changed since it was added. When a new, full node joins the blockchain network, it downloads a copy of all the blocks currently on the chain. After the new node synchronizes with the other nodes and has the latest blockchain version, it can receive any new blocks, just like other nodes.

There are two main types of blockchain nodes:

- Full nodes store a complete copy of the blockchain.
- Lightweight nodes only store the most recent blocks, and can request older blocks when users need them.

.CHAPTER 2
PROJECT DESCRIPTION

2.1. AIM AND OBJECTIVES

The importance of authentic document and certificate cannot be overemphasized as the reputations of institution are affected in every counterfeit or forged document or certificate.

- To design and implement student verification system using Blockchain Technology. The system will
- Prove the authenticity of document and certificate belonging to particular college or university thereby expunging forged and counterfeit certificate and document from circulation.
- Show the valid and legally registered student
- Prove its supremacy over the existing manual system of verification and identification by using dummy data.
- To highlight the importance of verifying and authenticating document and certificates.

2.2. SCOPE OF THE PROJECT

The scope of the project "Certificate Locker" includes the development of a secure platform for educational institutions to store and manage digital certificates of their students. The project aims to address the challenges associated with traditional paper-based certificates, which are often prone to loss, damage, and forgery. The use of Public Key Infrastructure (PKI) for encryption and decryption of certificates and Blockchain technology for verification of the integrity of certificates provides an added layer of security and reliability to the platform. The project includes the development of three user roles: Web Admin, Certificate Holder (students), and Certificate Verifier. The Web Admin will have the authority to approve registration requests from Certificate Holders and Certificate Verifiers, and to integrate the system into the Blockchain network. Certificate Holders will be able to register, upload their certificates, and receive a public-private key pair for all their certificates. They will also be able to receive certificate verification requests from Certificate Verifiers and respond with a verification link with certificates and private key, or verify the integrity of certificates using the hash key provided by the Blockchain. Certificate Verifiers will be able to register, request certificate verification from Certificate Holders, and use the private key provided by the Certificate Holders to verify the authenticity and integrity of the certificates.

CHAPTER 3

SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

Employers evaluate applicants' education background to screen out applicants who have forged and or exaggerated their qualifications and to protect the company against scam, such as degrees purchased from and verified by illegal institutions. Verifications are done through direct contact with the school or through the aid of a third party such as trusted background screening companies. Job seekers provides information to background screening agencies and pay the stipulated amount accrued to such service. Some pre-verification service provides a secured online cite where job seekers upload their education credentials to enable employers have access to view such verified information at any point in time.

3.1.1. Disadvantages

- Centralized-completely dependent on certificate issuing authority
- Manual-verification is usually done through emails, phone calls, or web forms
- Time-consuming could take weeks or months
- Easy to breach and tamper need for a decentralized trust system that is verifiable and tamper-proof is automatic, real-time, and is fraud-proof

3.2. PROPOSED SYSTEM

Companies today are looking for the best and are recruiting only the 'cream of the crop'. Since certificate storage and its security is a matter of concern to the university, students, and employers, the proposed system provides a platform to store and verify the student credentials using blockchain technology. By adopting to Blockchain, colleges can now ensure their students are not only qualified and skilled, but also verified and are job-ready. It is imperative for any organization to hire genuine candidates with valid education and authentic degrees. Certificate Verification system closer analysis of the original copy of certificates provided by a student. Through blockchain to check the genuineness of the certificates and student.

each and everything are digitalized in which the certificate of SSLC, HSC, and academic certificate are digitalized in the educational institution and provided to the students. Then the certificates are store in blockchain. By using blockchain technology we can provide a more secure and efficient digital certificate validation. The proposed concept is based on decentralisation and an open source strategy. There is no single entity that controls the system or the blockchain.

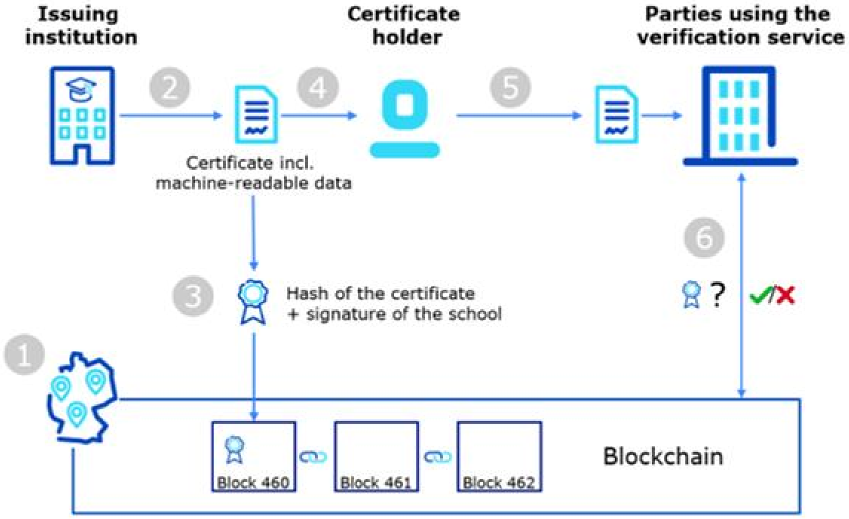
The architecture consists of three essential parts:

- **Blockchain as a database**

The database (blockchain) does not store any personal data; it only stores public keys, hashes10 and references to public institutions such as schools and universities. Because it operates multiple nodes, it is fail-safe and can withstand attacks. As envisaged by the authors of this white paper, the infrastructure would be operated by a consortium of municipal and public data centres. This would make it easier to secure and thus increase confidence in the system. The blockchain is private and access-protected. Certificate files are checked by certificate holders or users via the web service provided or interfaces that interact with one or more blockchain nodes. The system benefits from the known advantages of a blockchain (forgery protection, immutability, etc.) and at the same time avoids the disadvantages of completely public blockchain infrastructures (e.g. increased power consumption due to methods for building trust). The envisaged infrastructure does not offer crypto money that can be used speculatively. The costs for operating the blockchain are comparable to those of other distributed IT systems.

- **Web client for creating certificates**

Certificates are generated via a web service that can either run in the browser in the issuing institution or be integrated into an existing system via an interface. If required, the web service can be integrated into a client program. The web service can only be accessed by authorised institutions whose identity and authorisation to issue certificates has been confirmed by overarching authorities. The digital certificates created in this manner are transferred to the certificate holder securely. The current concept does not provide for the storage of the digital certificate file (either centrally or at the issuing institution). If, in the future, legislators impose requirements for the digital archiving of certificates, these can be easily implemented within the present concept.


Issuing institution
Certificate holder
Parties using the verification service
2
4
5
Certificate incl. machine-readable data
6
3
Hash of the certificate + signature of the school
?
1
Block 460
Block 461
Block 462
Blockchain

3.1. Fig 1: Web Client Create Certificate Basic View

- **Web service for checking certificates**

A publicly accessible web service shall be provided to verify the authenticity and integrity of certificate files. This web service can be used by anyone who has a certificate file, that is, by the certificate holder themselves and all third persons and institutions to whom the certificate holder submits their digital certificate file. By presenting the file, the hash value is calculated in the browser and checked for its presence in the blockchain. The certificate itself does not leave the browser. While there may be various causes for the result of the checking process to be negative (hash not written in the blockchain, file manipulation, etc.), a positive result confirms without doubt the document's authenticity and integrity as well as the issuing institution.

3.2.1. ADVANATGES

- Automate and ensure compliance for the verification process
- Reduce stress on staff for compliance issues
- Verification review and approval, comment codes and professional judgements
- simplifies and manages workflows

- Faster and efficient certificate verifications to your students.
- Paperless transactions.
- Transparent and can be accessed from anywhere in the world.
- High degree of data privacy.
- Considerable saving on account of courier time, use of paper, defined search parameters etc.
- Highly secured with best technologies and authentications.
- Reduced cost for students and recurring revenues for institutions.

CHAPTER 4

SYSTEM SPECIFICATION

4.1. Hardware Requirements:

- **Processors:** Intel® Core™ i5 processor 4300M at 2.60 GHz, 8 GB of DRAM
- **Disk space:** 320 GB
- **Operating systems:** Windows® 10, macOS*, and Linux*

4.2. Software Requirements:

Language	: Python 3.7.4(64-bit) or (32-bit)
Web Design	: HTML, CSS, Bootstrap
Development	: Flask 1.1.1
Database	: MySQL 5.
Local Server	: Wampserver 2i
Blockchain	: JSON

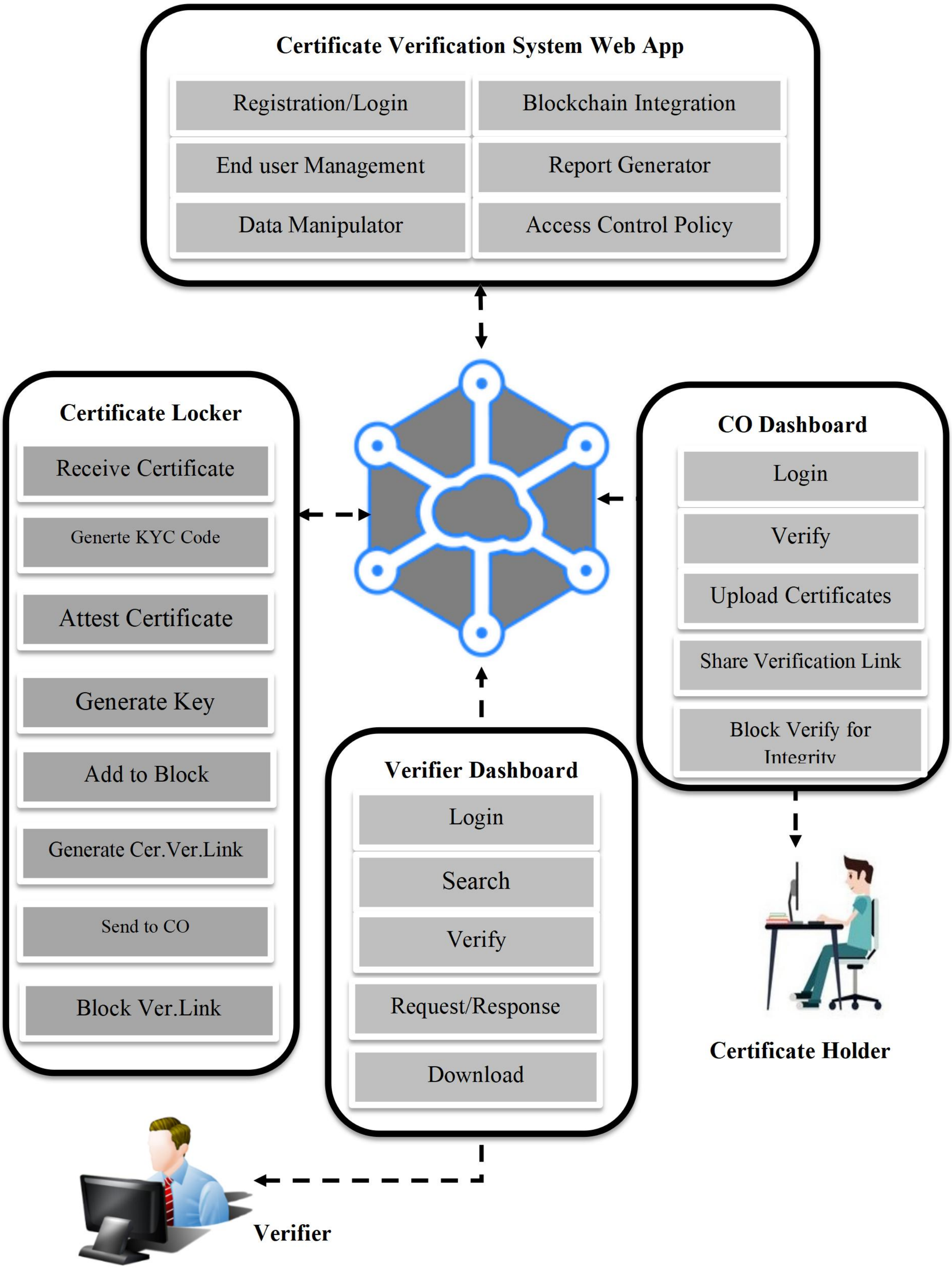
CHAPTER 5
SYSTEM DESIGN

5.1. SYSTEM ARCHITECTURE

A system architecture or systems architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behavior) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

Various organizations define systems architecture in different ways, including:

- An allocated arrangement of physical elements which provides the design solution for a consumer product or life-cycle process intended to satisfy the requirements of the functional architecture and the requirements baseline.
- Architecture comprises the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e., essential elements and their relationships) and associated characteristics and behavior.
- If documented, it may include information such as a detailed inventory of current hardware, software and networking capabilities; a description of long-range plans and priorities for future purchases, and a plan for upgrading and/or replacing dated equipment and software
- The composite of the design architectures for products and their life-cycle processes.

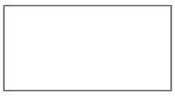
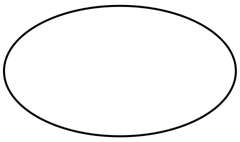
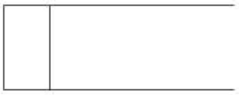


5.1. Fig 2: Architecture Diagram

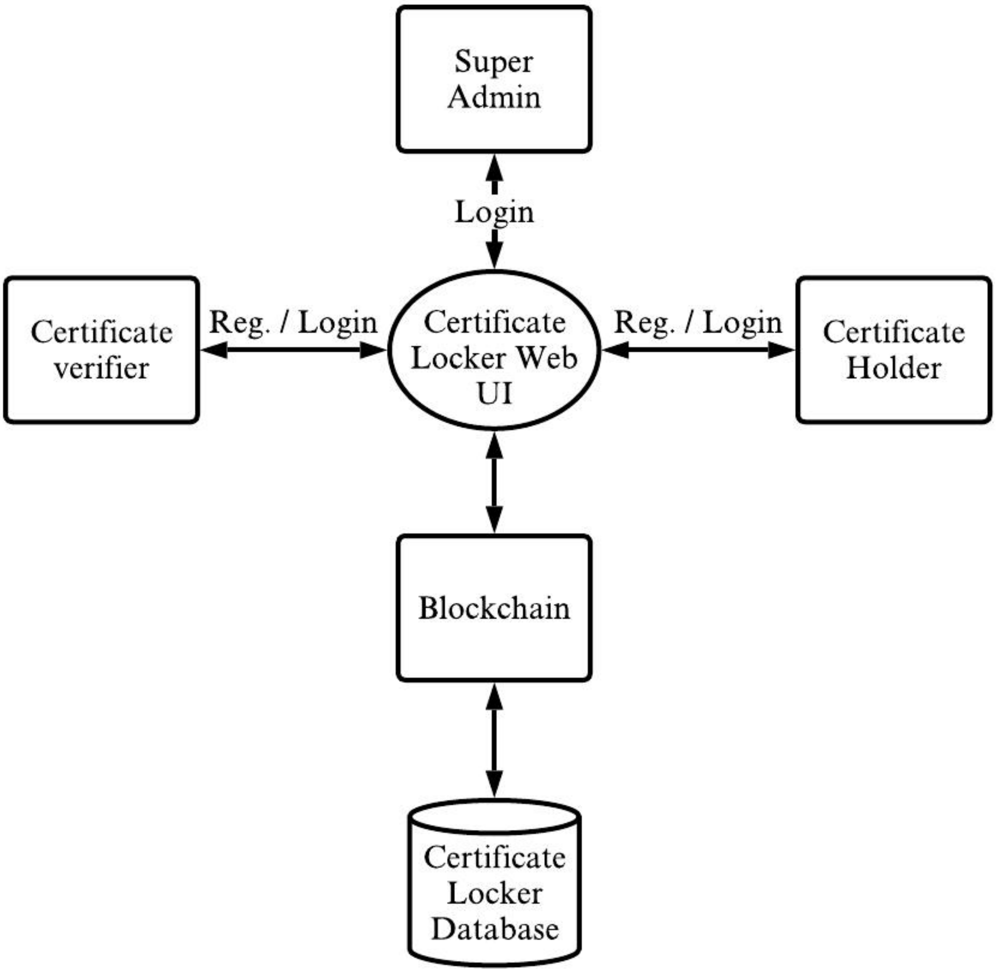
5.2. DATA FLOW DIAGRAM

A two-dimensional diagram that explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must (1) identify external inputs and outputs, (2) determine how the inputs and outputs relate to each other, and (3) explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

5.1. Data flow Symbols:

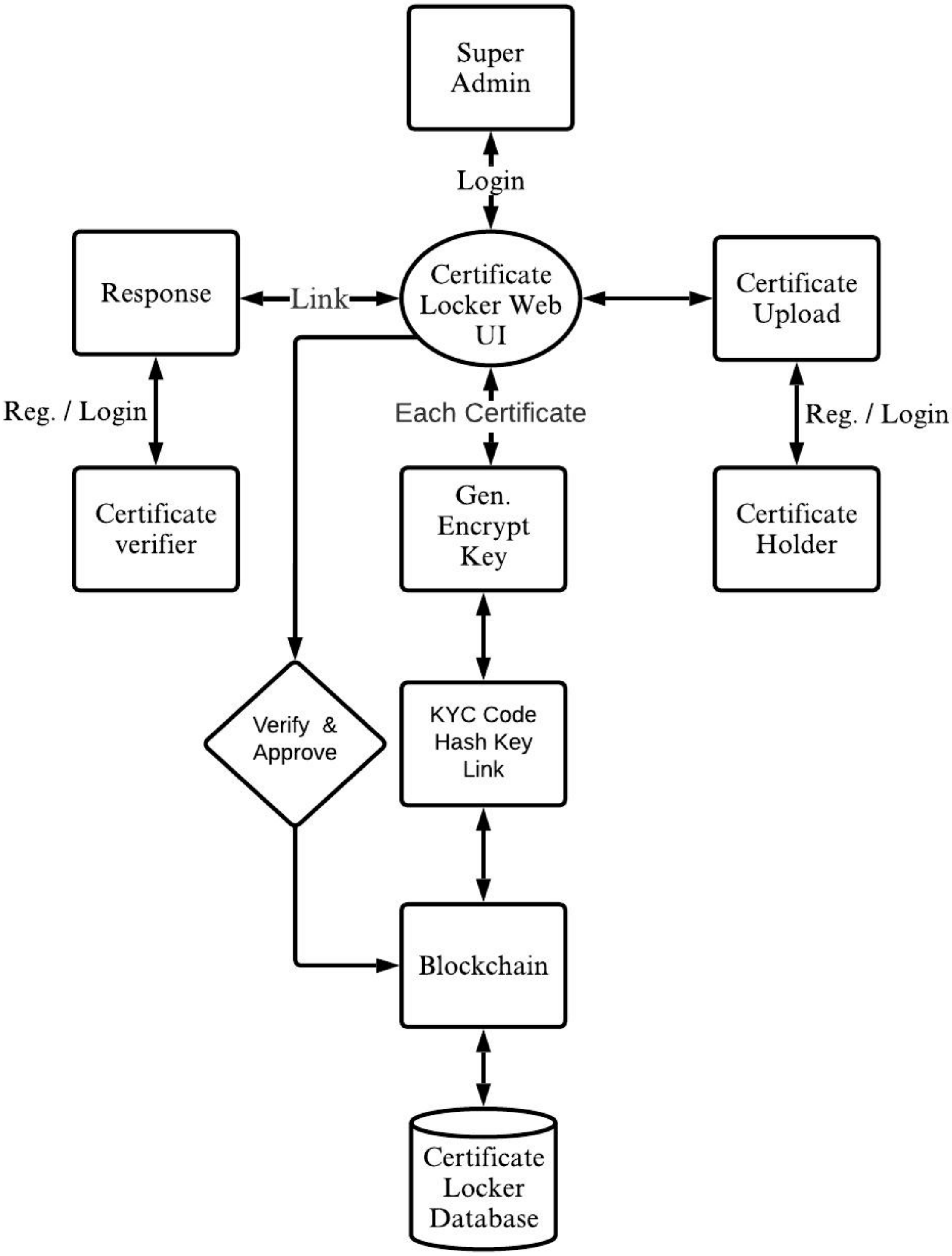
Symbol	**Description**
	An **entity**. A source of data or a destination for data.
	A **process** or task that is performed by the system.
	A **data store**, a place where data is held between processes.

5.2.1. LEVEL 0



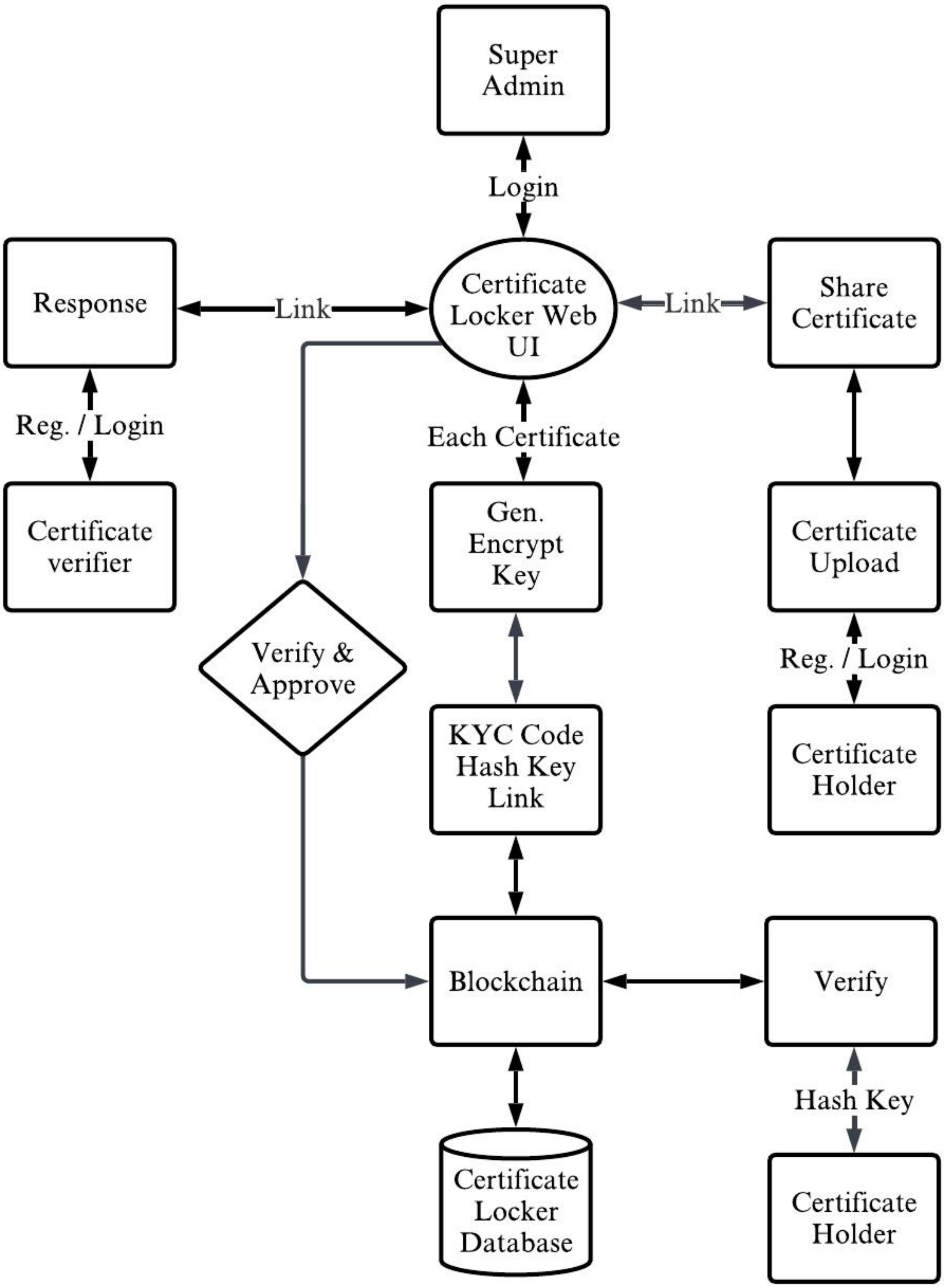
5.2. Fig 3: Data Flow Level Zero Diagram

5.2.2. LEVEL 1



5.3. Fig 4: Data Flow Level One Diagram

5.2.3. LEVEL 3



5.4. Fig 5: Data Flow Level Two Diagram

5.3. DATABASE DESIGN

5.3.1. Table Name: Admin

Table Name: Admin					
S.no	**Field**	**Data type**	**Field size**	**Constraint**	**Description**
1	User Name	Varchar	20	Null	Admin name
2	Password	Varchar	20	Null	Admin Password

5.3.2. Table Name: Certificate Holder Register

Table name: Certificate Holder Register					
S.no	**Field**	**Data type**	**Field size**	**Constraint**	**Description**
1	Id	Int	11	Null	Id
2	Name	Varchar	20	Null	Register Name
3	Mobile number	Bigint	20	Null	Mobile number
4	Email id	Varchar	40	Null	Email id
5	Address	Varchar	50	Null	Address
6	Certificate holder id	Varchar	20	Primary key	Certificate holder id
7	Password	Varchar	20	Null	password
8	Register date	Timestamp	Timestamp	Null	Register date

5.3.3. Table Name: Register

Table name: Verifier Register					
S.no	**Field**	**Data type**	**Field size**	**Constraint**	**Description**
1	Id	Int	11	Null	Id
2	Name	Varchar	20	Null	Register name
3	Mobile	bigint	20	Null	Mobile
4	Email	Varchar	40	Null	Email
5	Address	Varchar	40	Null	Address
6	Verifier id	Varchar	20	Primary key	Verifier id
7	Password	Varchar	20	Null	Password

5.3.4. Table Name: Register

Table name: Certificate Repository					
S.no	**Field**	**Data type**	**Field size**	**Constraint**	**Description**
1	Certificate id	Int	11	Primary key	Certificate id
2	Certificate holder id	Varchar	20	Foreign key	Certificate holder id
3	Certificate type	Varchar	30	Null	Certificate type
4	Certificate file link	Varchar	50	Null	Certificate file link
5	Certificate detail	Varchar	100	Null	Certificate detail
6	Certificate date	Varchar	20	Null	Certificate date
7	Certificate number	Varchar	20	Null	Certificate number

5.3.5. Table Name: Register

Table name: Certificate Request					
S.no	**Field**	**Data type**	**Field size**	**Constraint**	**Description**
1	Id	Int	11	Null	Id
2	Verifier id	Varchar	20	Foreign key	Certificate verifier id
3	Certificate id	Int	11	Null	Certificate id
4	Certificate holder id	Varchar	20	Null	Certificate holder id
5	Request date	Varchar	20	Null	Request date

5.3.6. Table Name: Register

Table name: Certificate Response					
S.no	**Field**	**Data type**	**Field size**	**Constraint**	**Description**
1	Id	Int	11	Null	Id
2	Certificate holder id	Varchar	20	Foreign key	Certificate holder id
3	Certificate id	Int	11	Null	Certificate id
4	Description key	Varchar	20	Null	Description key
5	Time date	Timestamp	Timestamp	Null	Time date

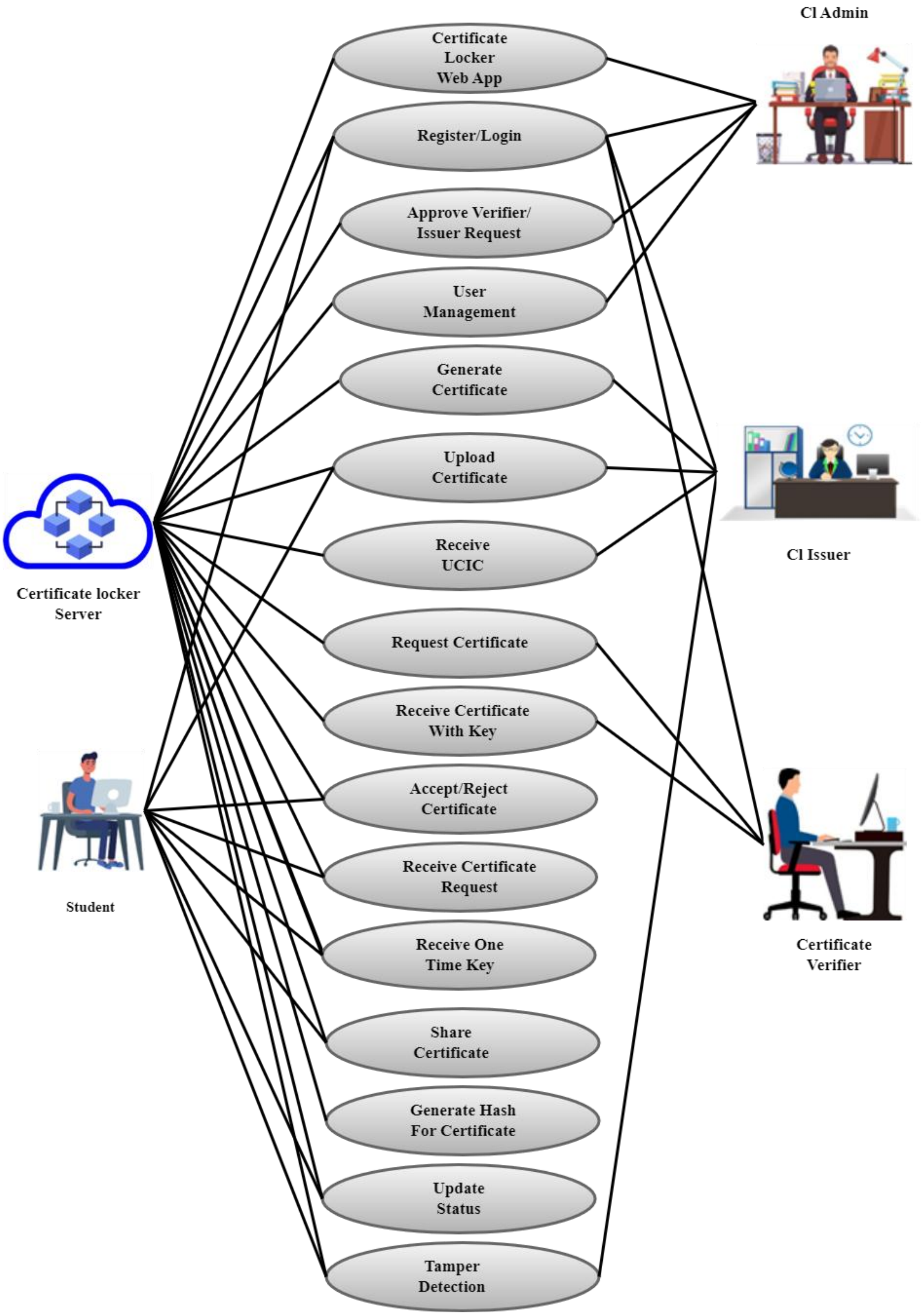
5.3.7. Table Name: Register

Table name: Certificate Transfer details					
S.no	**Field**	**Data type**	**Field size**	**Constraint**	**Description**
1	Id	Int	11	Null	Id
2	Certificate holder id	Varchar	20	Foreign key	Certificate holder id

3	Certificate id	Int	11	Null	Certificate id
4	Verifier id	Varchar	20	Null	Transfer Verifier id
5	Transfer date	Varchar	20	Null	Transfer date
6	Revoke status	Int	11	Null	Revoke status

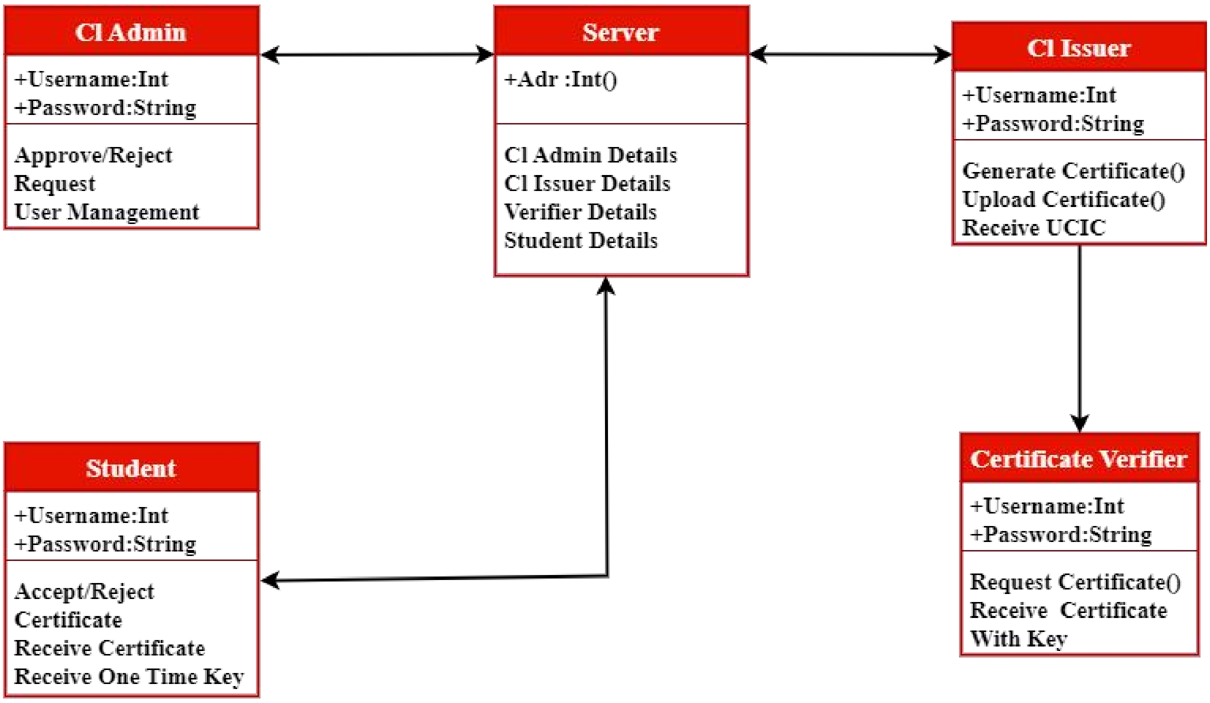
5.4. UML DIAGRAM

5.4.1. USE CASE



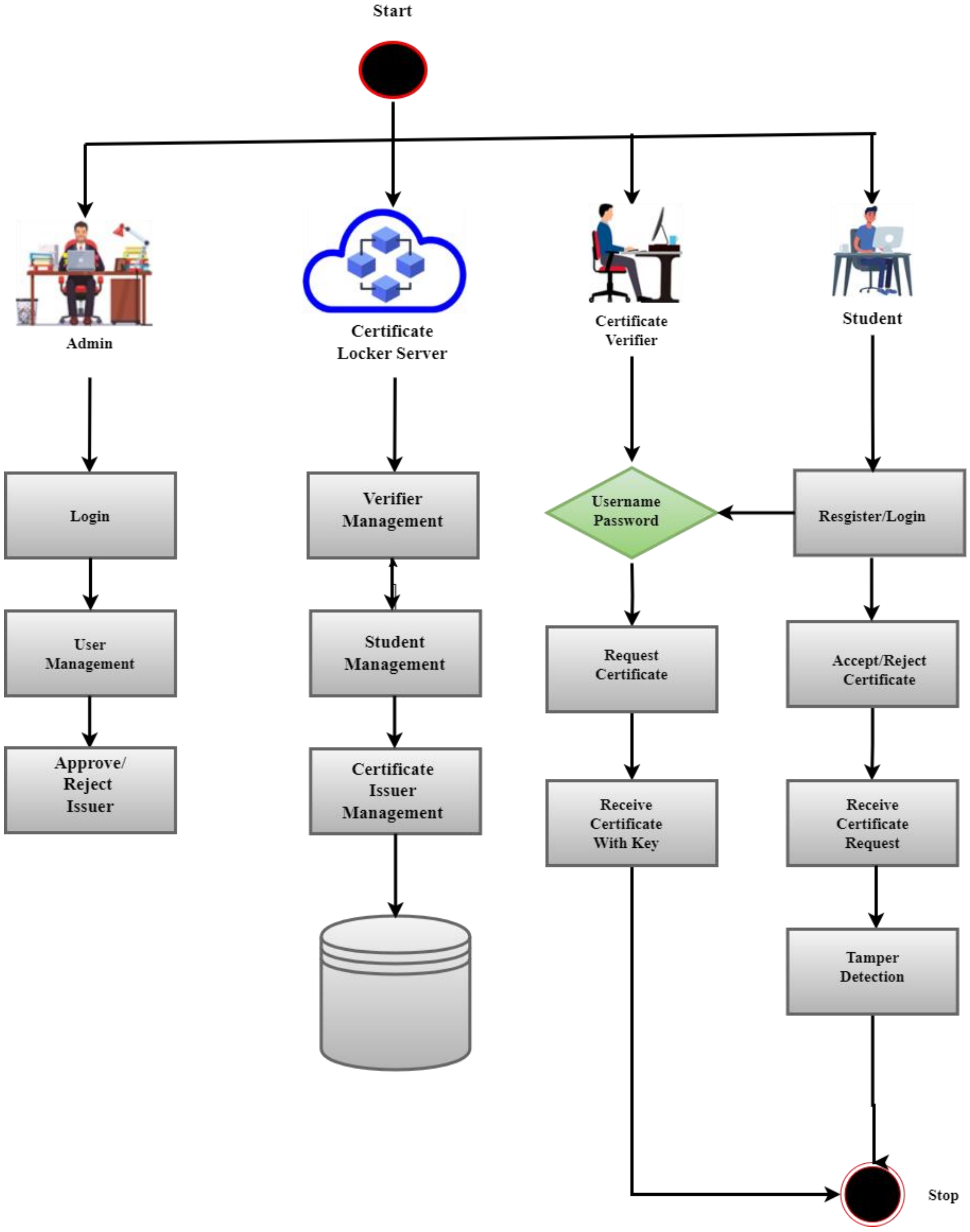
5.5. Fig 6: UML - Use Case Diagram

5.4.2. CLASS DIAGRAM



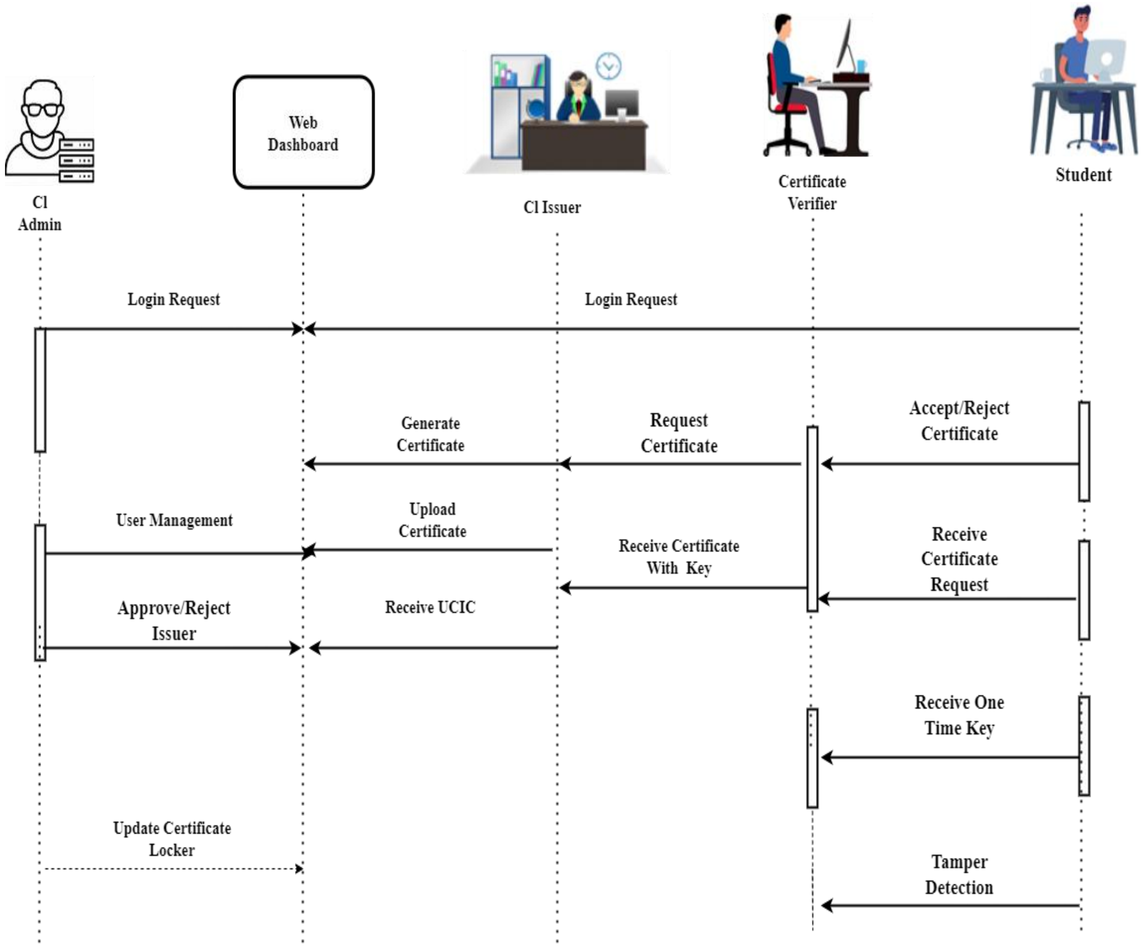
5.6. Fig 7: Class Diagram

5.4.3. ACTIVITY DIAGRAM



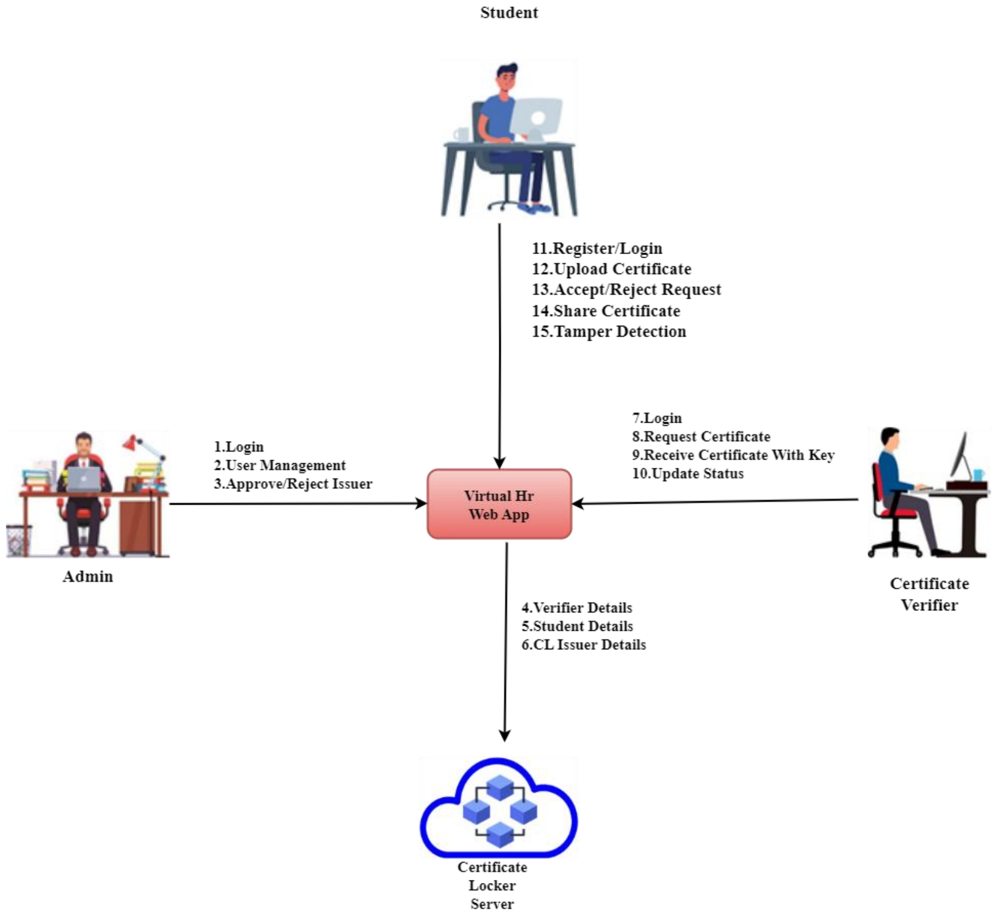
5.7. Fig 8: Activity Diagram

5.4.4. SEQUENCE DIAGRAM



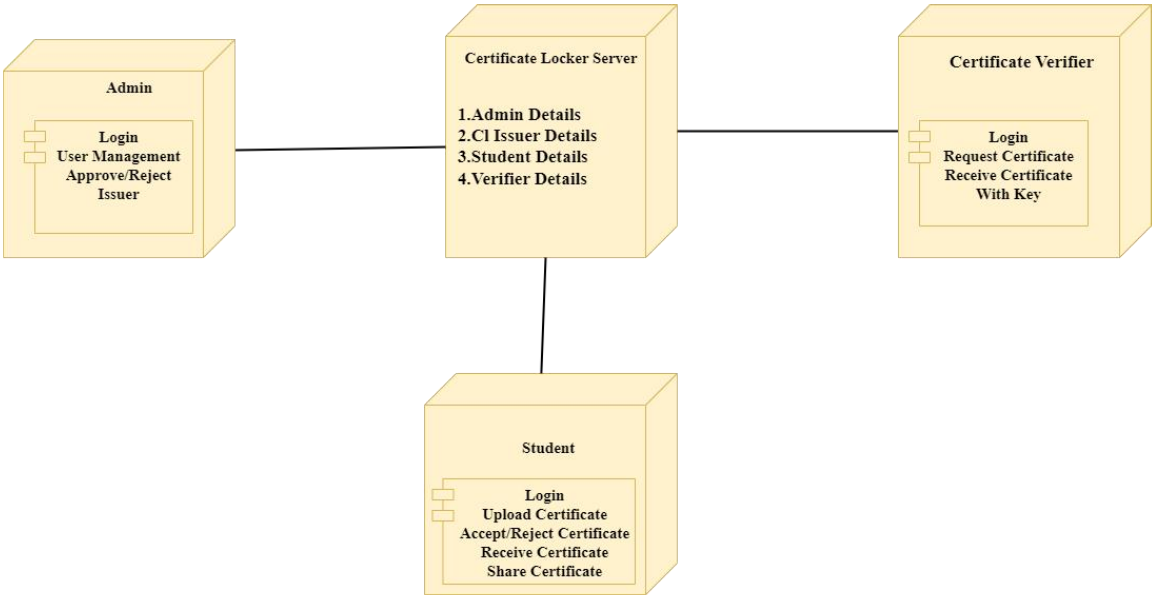
5.8. Fig 9: Sequence Diagram

5.4.5. COLLABORATION DIAGRAM



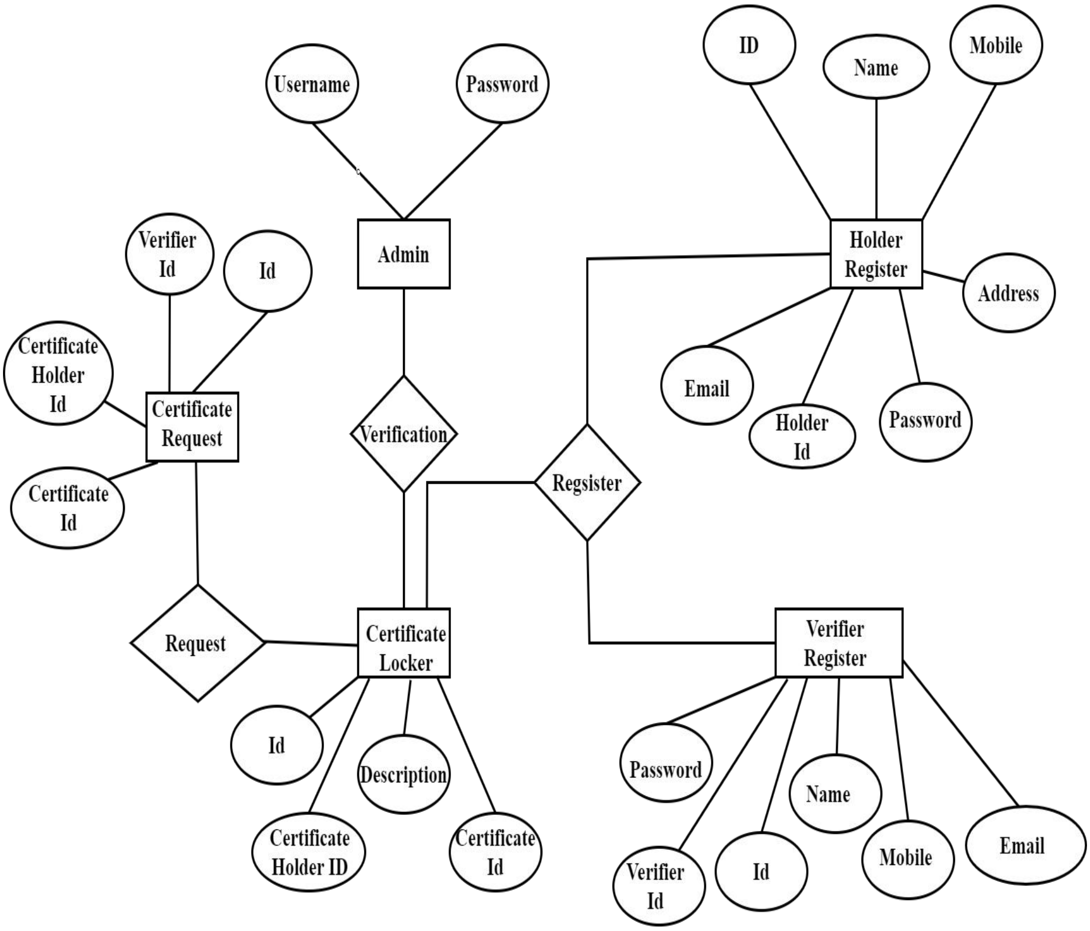
5.9. Fig 10: Collaboration Diagram

5.4.6. DEPLOYMENT DIAGRAM



5.10. Fig 11: Deployement Diagram

5.5. ER DIAGRAM



5.11. Fig 12: ER Diagram

CHAPTER 6

SOFTWARE DESCRIPTION

4.1. PYTHON 3.7.4

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain.

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc. The biggest strength of Python is huge collection of standard libraries which can be used for the following:

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)

- Test frameworks
- Multimedia
- Scientific computing
- Text processing and many more.

Pandas

pandas are a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. pandas are a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.

Pandas is mainly used for data analysis and associated manipulation of tabular data in Data frames. Pandas allows importing data from various file formats such as comma-separated values, JSON, Parquet, SQL database tables or queries, and Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features. The development of pandas introduced into Python many comparable features of working with Data frames that were established in the R programming language. The panda's library is built upon another library NumPy, which is oriented to efficiently working with arrays instead of the features of working on Data frames.

NumPy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

Scikit Learn

scikit-learn is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license.

Scikit-learn (formerly scikits. learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

4.2. MYSQL

MySQL tutorial provides basic and advanced concepts of MySQL. Our MySQL tutorial is designed for beginners and professionals. MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle

Company. MySQL database that provides for how to manage database and to manipulate data with the help of various SQL queries. These queries are: insert records, update records, delete records, select records, create tables, drop tables, etc. There are also given MySQL interview questions to help you better understand the MySQL database.

MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications. It is developed, marketed, and supported by MySQL AB, a Swedish company, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is My Ess Que Ell. However, you can pronounce it in your way. Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

4.3. WAMPSERVER

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your database.

WAMPServer is a reliable web development software program that lets you create web apps with MYSQL database and PHP Apache2. With an intuitive interface, the application features numerous functionalities and makes it the preferred choice of developers from around the world. The software is free to use and doesn’t require a payment or subscription.

4.4. BOOTSTRAP 4

Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites.

It solves many problems which we had once, one of which is the cross-browser compatibility issue. Nowadays, the websites are perfect for all the browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones). All thanks to Bootstrap developers -Mark Otto and Jacob Thornton of Twitter, though it was later declared to be an open-source project.

Easy to use: Anybody with just basic knowledge of HTML and CSS can start using Bootstrap

Responsive features: Bootstrap's responsive CSS adjusts to phones, tablets, and desktops

Mobile-first approach: In Bootstrap, mobile-first styles are part of the core framework

Browser compatibility: Bootstrap 4 is compatible with all modern browsers (Chrome, Firefox, Internet Explorer 10+, Edge, Safari, and Opera)

4.5. FLASK

Flask is a web framework. This means flask provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have built-in abstraction layer for database handling, nor does it have formed a validation support. Instead, Flask supports the extensions to add such functionality to the application. Although Flask is rather young compared to most Python frameworks, it holds a great promise and has already gained popularity among Python web developers. Let’s take a closer look into Flask, so-called “micro” framework for Python.

Flask is part of the categories of the micro-framework. Micro-framework are normally framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins.

4.6. JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON consists of "name: object" pairs and punctuation in the form of brackets, parentheses, semi-colons and colons. Each object is defined with an operator like "text:" or "image:" and then grouped with a value for that operator. The simple structure and absence of mathematical notation or algorithms, JSON is easy to understand and quickly mastered, even by users with limited formal programming experience, which has spurred adoption of the format as a quick, approachable way to create interactive pages.

CHAPTER 7
SYSTEM IMPLEMENTATION

7.1. System Description

In this project, a blockchain certificate verification system was developed based on relevant technology. The system's application was programmed on the Private Blockchain platform and is run by the Python. In the system, three groups of users are involved, Admin, CV and CO or CH have access to the system, and can browse the system database. When CO fulfilled certain requirements, the authorities grant a certificate through the system. After the CO have received their certificate, they are able to inquire about any certificate they have gained.

There are several parameters that are to be checked and validated while conducting education verification.

Confirming Name and registration number- This is basically a form of identity check. By tallying the name and roll number of the candidate with official institute records, one can confirm whether the individual has truly attended the university that is being claimed.

Verifying course details- This education verification parameter is of critical importance as it forms a direct link between the education the applicant has received and the job role that is offered. This involves making a note of all the subjects that were taken by the candidate and verifying its relevance to the job.

Year of passing- Many candidates falsify their dates of graduation as young candidates are increasingly sought after by companies. Education verification of graduation records can be easily accessed by contacting the institution mentioned by the candidate.

Status of graduation- This basically ensures the quality of the candidate. Candidates with lower grades or drop years are usually not preferred by businesses. The nature of the job may require candidates with high skill and knowledge. Candidates may lie about their grades or hide their drop years. Such discrepancies are to be identified during education verification to ensure high quality hiring. Verifying the authenticity of the educational institute can help safeguard the company against candidates from such dubious institutions.

Process flow

1. Trustworthy consortium based on and administered by municipal and public data centres operates a distributed infrastructure consisting of a blockchain and web services

2. Approved, authorised schools generate digital certificates. These contain the print layout

of the certificate and machine-readable certificate data

3. Authorised schools write a hash value associated with the certificate together with the identity of the school in the blockchain

4. Digital certificates are securely distributed to certificate holders

5. School student uses digital certificates to apply for a job on a web portal or by e-mail

6. Parties using the verification service can ensure the authenticity of the certificate via a web service for checking the certificates. This service calculates the hash of the certificate file and compares it with the data in the blockchain. The identity of the issuing institution is displayed.

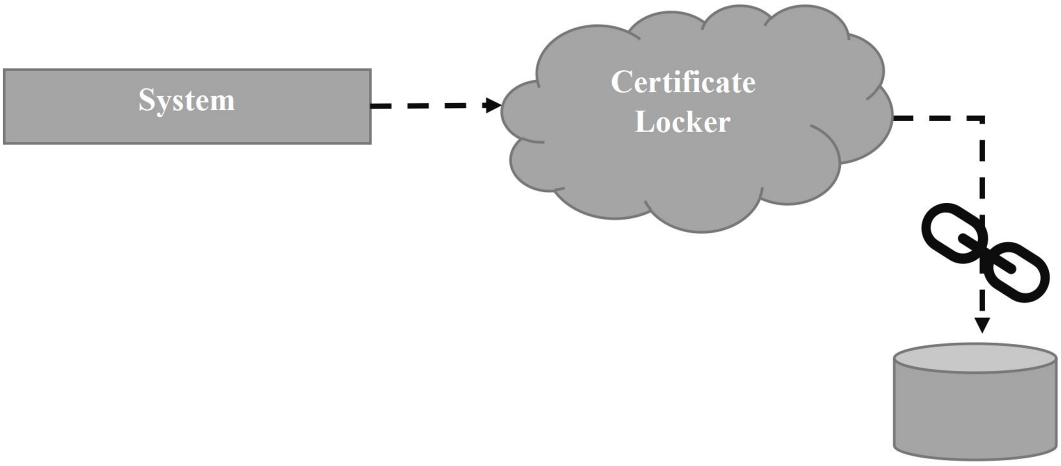
7.2. System Flow

7.2.1. Certificate Locker Webapp

This project designed considering the need of colleges in today's time for end-to-end digitisation. It brings all the stakeholders on a single cloud-based web app platform to provide strong platform-based connectivity for truly digital operations using blockchain Technology. Decentralized system with integrated management approach makes is highly relevant for all kinds of colleges and educational organisations. Cloud facility allows the system to store bulk data and save time with additional benefits like reliability, backup, high speed, and mobility.

7.2.2. Blockchain Integration

This module design and develop a system for dynamic and secure e certificate generation system using smart contracts in a blockchain environment. Design and integrate own blockchain in an open-source environment with a custom mining strategy as well as a smart contract. Finally, validate and explore system performance using a consensus algorithm for proof of validation.



7.1 Fig.13 Block Chain Intrgration Diagram

Blockchain technology assures that the integrity of data once written in the chain. The promises like tampering resistance, non-repudiation and its traceability make blockchain a good candidate to provide some of the notarization capabilities.

Blockchain Registration: the SII referring to the certificate issuer where the Certificate owner records her DVN on the blockchain so that it is available to the notary's office network that has access to the blockchain.

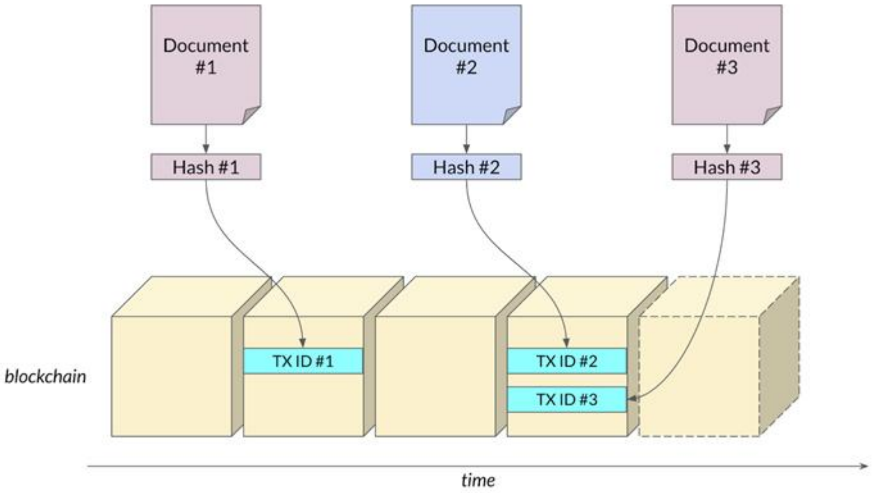
7.2.2.1. Proof of Existence

Proof of Existence is defined as "an online service that verifies the existence of computer files as of a specific time via time-stamped transactions." And, in its essence, notarizing is about certifying the existence of a document and blockchain can carry out this function with little effort. When a document is saved on the blockchain, no one can modify it without leaving a time stamp. This would be the most common use case of blockchain in notarization, based on two steps: hashing the document and storing it on the blockchain.

- Hashing the document: In direct replacement of uploading the certified document on the blockchain, it is more practical to upload the hashed copy of the certified document. It is also costly to have numerous (large) documents stored on the blockchain. Data privacy also would need to be taken into consideration.
- Storage on the blockchain: The hash value is then recorded on the blockchain.

7.2.2.2. Recording into Blockchain

Once the hash result is computed, it will be recorded to blockchain. Because each blockchain is built with its own purpose, the implementation may be of a bit difference. In proof-of-existence type of service, only the hash value is stored in the blockchain. The data space requirement is quite low and therefore almost all blockchain can handle this. Several notary services implemented on Bitcoin network. The method is quite straightforward: spend a small bitcoin payment good for miner processing the record, and place the hash value using OP_RETURN feature. The transaction then returns a transaction ID. And this is the transaction ID associated with the hash just written on the blockchain. Here is the model.



7.2 Fig.14 Recording into Block Chain Diagram

7.2.2.3. Proof of Ownership

When we must transfer the ownership of a document, the system records the user identification and uses it for verification. Through this process, a signature is created and blockchain provides it storage during contract deployment. Furthermore, smart contracts can be used to provide contract capability, which is useful in adding certain flexibility in the design of the overall application.

7.2.3 Authorized Users

The system has four types of users. They are:

- **Central Certificate Authority**

CCA is a company or institution, public or private, in which the issuance, analysis, authentication, registration, and filling of notes and documents takes place, giving public faith to the presented documents.

Login Service: allows CCA to log in to the system so that they can perform the services requested by the CO.

Order Service: maintains all customer orders and their services requested on a given date.

A certificate authority is an entity similar to a notary public. It issues digital certificates, signs certificates to verify their validity and tracks which certificates have been revoked or have expired. CA must enter a unique Certificate Authentication Number (CAN) when notarizing a document on the system. CCA enter their username and password, and then add their signature to the electronic document.

- **Central Certificate Verifier**

A Notary's signature must be applied shortly after a CCV verifies the Notarize document. CCV who verifies the certificates and gets the list of certificates with respect to the Aadhar card number of the CO.

There are three possible statuses:

Pending - added document/version but not all participants are signed it.

Approved - all participants signed the document / new version.

Rejected - one of the participants rejected the version.

- **State Identity Verifier**

The identity owner can store those credentials in their personal identity wallet and use them later to prove statements about his or her identity to a third party (the verifier).

- **State Identity Issuer**

SSI is the organization that the notary needs to consult to continue the activities related to the respective service. Each service provided by the notary may depend on authenticated and validated information from a specific EI. In addition, those entities may also consult and record information on the blockchain in accordance with their respective services.

The State identity issuer, a trusted party such as local government, can issue personal credentials for a certificate owner (the user). By issuing a credential, the SII attests to the validity of the personal data in that credential (e.g., last name and date of birth).

- **Certificate Owner**

Certificate Registration Service: records all certificates that have been requested by CCA through SII (e.g., birth, marriage, death).

To notarize documents, a user must have the **Notarize Documents** permission. The user profile must also include all required notary information. To create a digital signature, CO need a signing certificate, which proves state identity issuer. When user send a digitally-signed macro or document, user also send their certificate and public key.

- **System Administrators**

Admin has full control access rights to the system. Admin is one of the employees assigned to hold the system. Admin access rights have the following authority: processing client data, processing parameter data consisting of (service type data, form data, phase data, condition data, and notary data), processing request data, updating progress, viewing and printing reports, setting the service type phase, setting the service type form.

5.2.4. Public key infrastructures (PKIs)

PKIs can come in the form of key management servers or centralized directories. They store key pairs, digital signatures, digital certificates and hash values.

The creation of the key pair – used the analogy of keys unlocking a safe. The private key unlocks the safe while the public key locks the safe. To decrypt Alice's document, Bob creates a private-public key pair by running a key generation algorithm from the PKI.

The creation of digital certificates – certificates verify the digital signature by displaying the link between Alice and her public key. In those systems that issue certificates, the signature is known as a "qualified digital signature". They produce the validity period, the signature algorithm, a serial number and the name of the certification authority. These validity periods can be of a long duration and they also rely on the sustained readability and integrity of the **signatures.**

Private key protection – in key pairings, the encrypted private key is mathematically linked to the public key, which is unencrypted. Despite this link, it is computationally infeasible to deduce the value of the private key from the value of the public key.

Certificate revocation in the event of a compromised private key – once a user's certificate has been revoked, the PKI must preserve the certificate on a database accessible to all users in the network so that it cannot be re-used. A public file encryption function has a single point of failure. Once breached, the attacker can pass encryption functions that are bogus.

Private key backup and recovery – if the user loses his private key, any files encrypted with that key will be lost. The PKI needs a backup and recovery mechanism for lost private keys.

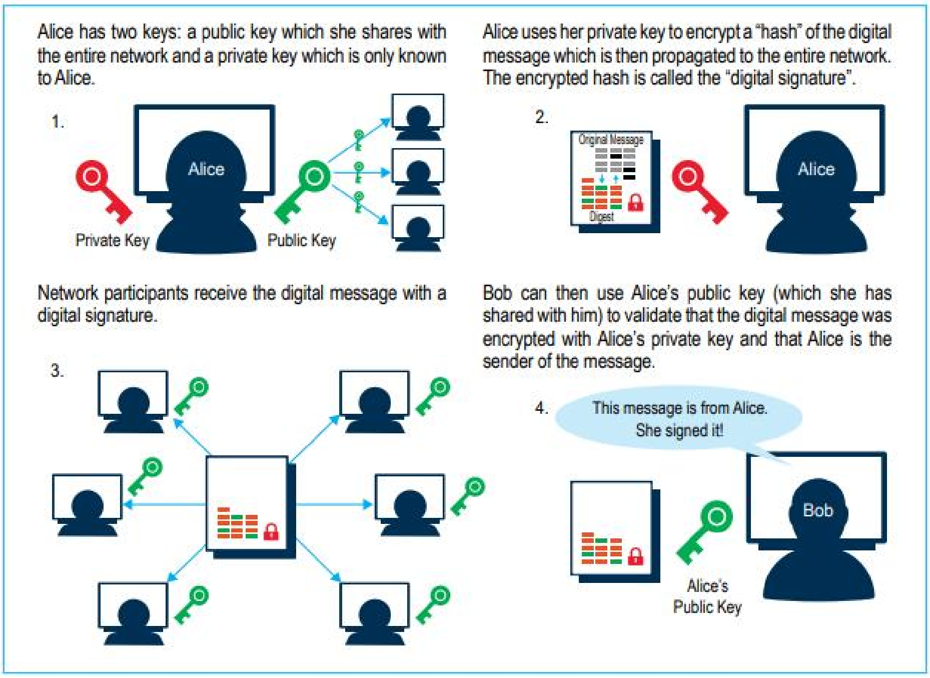
Key and certificate update – this is a mechanism for the renewal of expiring digital certificates. The PKI achieves this by carrying out the renewal automatically or notifying the user to carry out an operation that updates the certificate himself. Blanchette stated that the idea behind fixed expiry dates is to mitigate against incremental damage to the network's integrity due to corrupted public keys.

Key history management – following a key update that generates new key pairs, history management makes it easier for the user to determine which private key to use for decrypting files.

Certificate access – Lightweight Directory Access Protocol (LDAP) directory for convenient access to certificates. PKIs therefore require the preservation of at least three components: key pairs, active digital certificates and revoked digital certificates.

7.5.2.4.1. PKI E-signature service

CO and required parties can electronically sign documents while generating publicly verifiable proof of the timestamp and integrity of signatures.



Alice has two keys: a public key which she shares with the entire network and a private key which is only known to Alice.

Alice uses her private key to encrypt a "hash" of the digital message which is then propagated to the entire network. The encrypted hash is called the "digital signature".

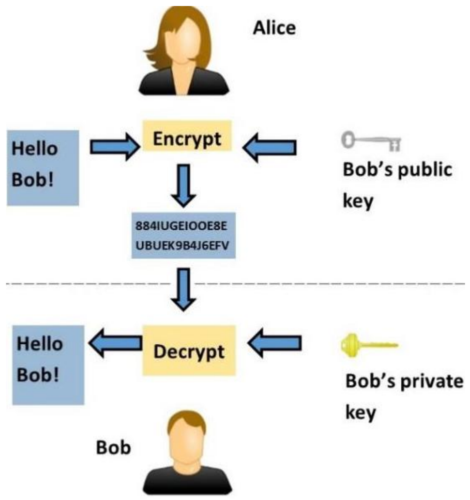
Network participants receive the digital message with a digital signature.

Bob can then use Alice's public key (which she has shared with him) to validate that the digital message was encrypted with Alice's private key and that Alice is the sender of the message.

7.3 Fig.16 Signature Service Diagram

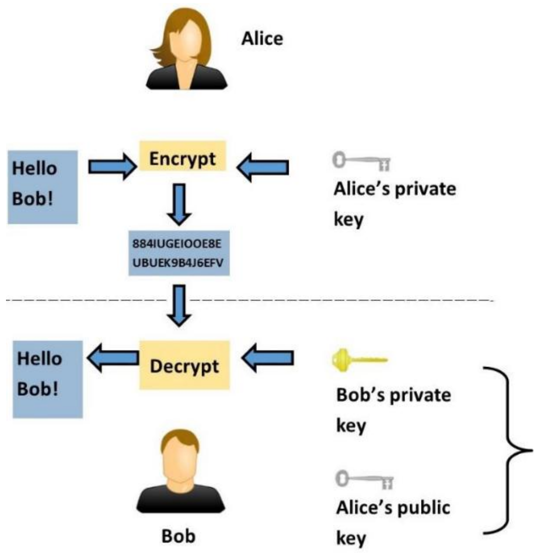
To create a digital signature, counterparties sign the document directly. This structure differs from that of the blockchain, where the counterparties sign a hash that represents the document. This paper will refer to asymmetric or "public key" cryptography, which involves an interaction between public and private keys. The public key is stored on a server accessible to other users on the network, while the private key remains a secret.

Public key cryptography operates under a dual procedure of which signatures form a part. Assuming there are two parties, each party possesses a key pair: the public and private keys. Figure 1 shows the following process: Alice and Bob are fellow archivists about to manage the transmission of a document. Alice is about to send a document to Bob across the network. Before she does so, Alice encrypts the document with Bob's public key. Alice sends it across, and Bob decrypts the file with his private key.



7.4. Fig.17 Public key encryption of a document

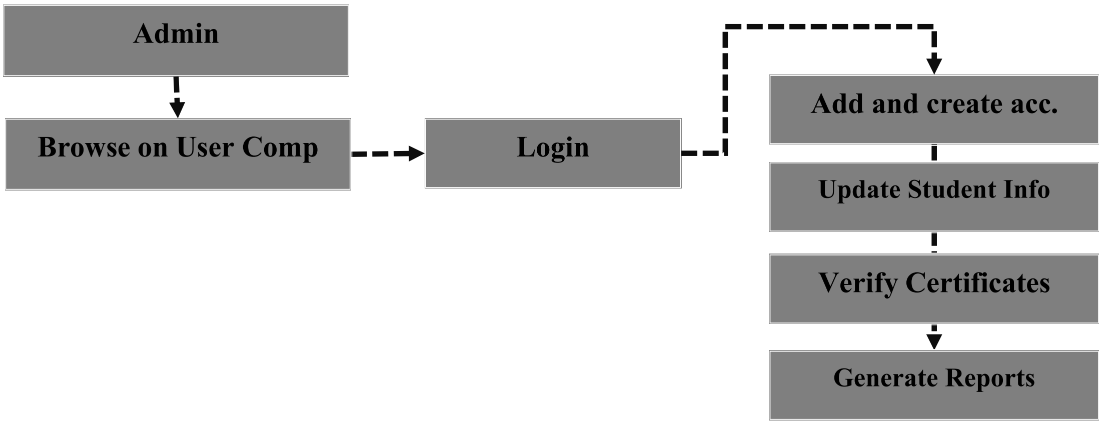
For the signature (Figure 2), the roles are reversed: Alice encrypts that same document with her private key and then sends it to Bob. Bob decrypts it with both his private key and Alice's public key. If he decrypts it successfully, Bob can then verify that Alice was the sender. The document, standing as a new file, should state that it has been verified. The resulting digital signature is intended to be available for anyone to verify the identity of the party that signed the document (in this case, Alice). The signature will be available not only to Bob, but to subsequent third parties as well. From the point-of-view of the archivist, the signature is genuine in that it is what it claims to be, and it is authentic in that the elements that are required for that authenticity are present.



7.5. Fig.17 Public key encryption for the signature of a document

Certificate Verifier

Parties using the verification service are institutions where a certificate holder presents his or her certificate, such as universities, central application institutions or companies. This module is handled by top management to create role wise user logins to company verifiers accessing Certificate Verification System.

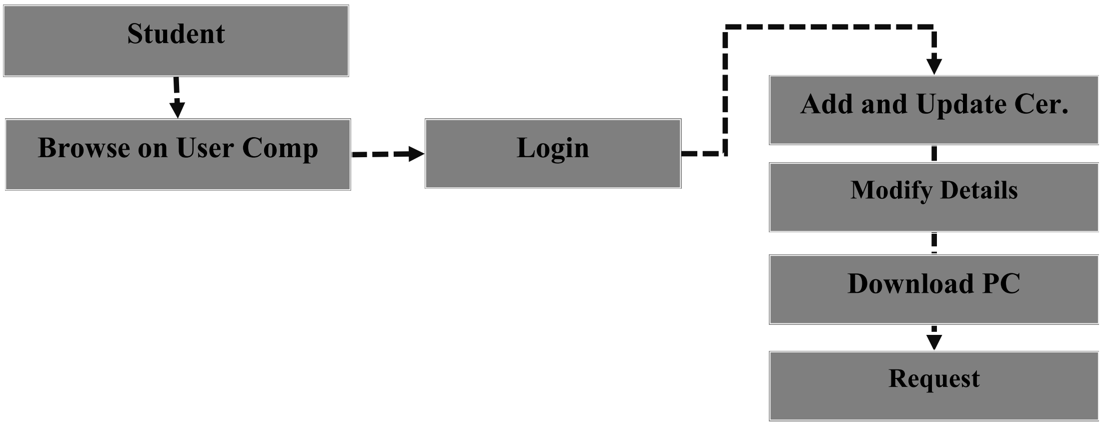


7.6. Fig 18: Certificate Verifier Diagram

7.2.5.2. Certificate Holder

Certificate holders such as school students, university students or other persons issued with proof of an educational qualification. A certificate as envisaged by this concept can be a school certificate or university degree or another type of certificate.

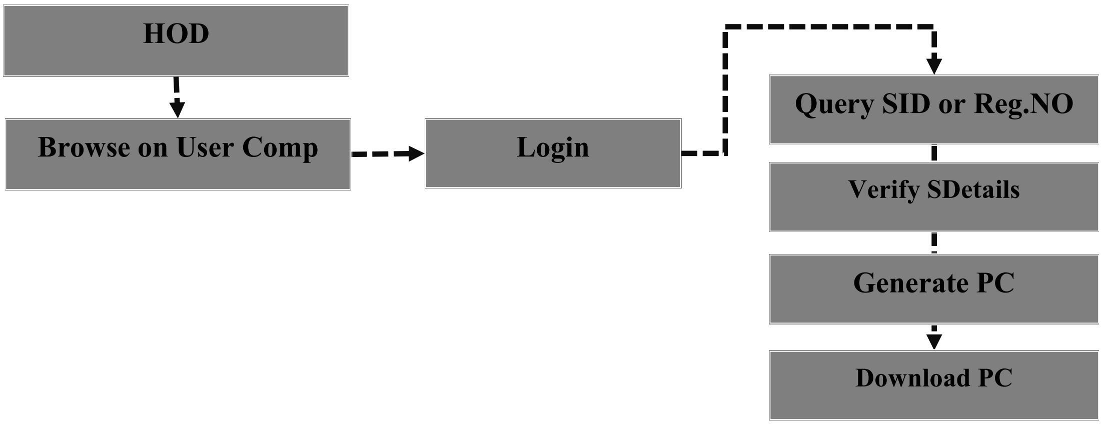
Here CO or CH can login, view profile, update profile, upload their educational certificates, and download their certificates too.



7.7. Fig 19: Certificate Holder Diagram

Super Admin

Admin can generate notifications for CO or CH and VA; send SMS, emails, reminders time to time. Here Admin can add/update/delete student and student education certificates /employee/courses, view course list/student list or many different modules.



7.8. Fig 20: Super Admin Diagram

Validations of Certificates

This phase is considered the practical and actual result of the system, as it represents the benefit of applying the proposed model. Actual works of this phase begin when an institution or a company wanted to verify the validity of the certificates submitted to it to obtain the advertised position. As the proposed system provides a webpage or application that can be accessed from anyone, any time, and anywhere in the world without the need to participate or pay for the system. The gate could be a portal or an application based on the Internet, which enables anyone to verify the correctness of the students' certification instantly. Even when the absence of the institution that granting this certificate for any reason, such as stopping its work or being exposed to any natural or intentional disaster such as fires or floods. When the graduate student provides his gained higher certificate information to obtain a specific position within a specific company (get hire). Therefore, the company or organization needs to verify the validity of the submitted certificate from the concerned person. In this case, the company employer can verify the validity of the provided certificate in a real, reliable, and safe way through a portal belong to the system.

The work of this phase begins when enters only the important information of the certificate which is the name, average grade, date, issue number, agency, or the certificate is uploaded electronically (if the certificate is submitted to the company electronically) through the portal by an employer in the company. The received information of the certificate submit to the portal is entered at the same hash function used when creating and stored the original certificate hash on the blockchain network. Then a query is created to perform a search for a match of the generated hash within universities blockchain network stored hashes. If a match is found means that the certificate information is correct and the validity message will be shown. If there is no match for any reason such as a mistake in the entry process or a process that changes any information from the certificate information (forging the certificate). In this case, there will be no match of the stored hashes on the blockchain network and the system will be shown a message of failure or error through the portal.

Access Control Policy

Access control enables you to remotely manage processes, such as adding or revoking user access, without requiring any hardware.

7.3. SOURCECODE

Certificate Owner Creation

```
@app.route('/login_cca',methods=['POST','GET'])
def login_cca():
cnt=0
act=""
msg=""
if request.method == 'POST':
username1 = request.form['uname']
password1 = request.form['pass']
mycursor = mydb.cursor()
mycursor.execute("SELECT count(*) FROM nt_cca where uname=%s && pass=%s
&& utype='CCA'",(username1,password1))
myresult = mycursor.fetchone()[0]
if myresult>0:
session['username'] = username1
#result=" Your Logged in sucessfully**"
return redirect(url_for('home_cca'))
else:
msg="Your logged in fail!!!"
return render_template('login_cca.html',msg=msg,act=act)
@app.route('/register',methods=['POST','GET'])
def register():
result=""
act=request.args.get('sid')
mycursor = mydb.cursor()
if request.method=='POST':
name=request.form['name']
mobile=request.form['mobile']
email=request.form['email']
address=request.form['address']
uname=request.form['uname']
pass1=request.form['pass']
```

```
now = datetime.datetime.now()
rdate=now.strftime("%d-%m-%Y")
mycursor = mydb.cursor()
mycursor.execute("SELECT count(*) FROM nt_register where
uname=%s",(uname, ))
cnt = mycursor.fetchone()[0]
if cnt==0:
mycursor.execute("SELECT max(id)+1 FROM nt_register")
maxid = mycursor.fetchone()[0]
if maxid is None:
maxid=1
result = hashlib.md5(uname.encode())
key=result.hexdigest()
pbkey=key[0:8]
prkey=key[8:16]
sql = "INSERT INTO nt_register(id, name, mobile, email, address,  uname,
pass,private_key,public_key) VALUES (%s, %s, %s, %s, %s, %s, %s,%s,%s)"
val = (maxid, name, mobile, email, address, uname, pass1,pbkey,prkey)
act="success"
mycursor.execute(sql, val)
mydb.commit()
print(mycursor.rowcount, "record inserted.")
act="success"
return redirect(url_for('register',act=act))
else:
act="wrong"
result="Already Exist!"
```

Upload & Encrypt Certificate

```
detail=request.form['detail']
mycursor.execute("SELECT max(id)+1 FROM nt_certificate")
maxid = mycursor.fetchone()[0]
if maxid is None:
maxid=1
cno="CN"+mm+yy+str(maxid)
```

```
if 'file' not in request.files:
flash('No file part')
return redirect(request.url)
file = request.files['file']
file_type = file.content_type
# if user does not select file, browser also
# submit an empty part without filename
if file.filename == '':
flash('No selected file')
return redirect(request.url)
if file:
fname = "F"+str(maxid)+file.filename
filename = secure_filename(fname)
file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
result = hashlib.md5(cno.encode())
key=result.hexdigest()
ckey=key[0:8]
##encryption
password_provided = ckey # This is input in the form of a string
password = password_provided.encode() # Convert to type bytes
salt = b'salt_' # CHANGE THIS - recommend using a key from os.urandom(16), must
be of type bytes
kdf = PBKDF2HMAC(
algorithm=hashes.SHA256(),
length=32,
salt=salt,
iterations=100000,
backend=default_backend()
key = base64.urlsafe_b64encode(kdf.derive(password))
input_file = 'static/upload/'+fname
output_file = 'static/upload/E'+fname
with open(input_file, 'rb') as f:
```

```
data = f.read()

fernet = Fernet(key)
encrypted = fernet.encrypt(data)
with open(output_file, 'wb') as f:
f.write(encrypted)
message="Certificate Owner:"+uname+", KYC Code:"+cno+", Key:"+ckey
act="yes"
##store
sql = "INSERT INTO nt_certificate(id,uname,ctype,filename,detail,rdate,canno,ckey)
VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"
val = (maxid,uname,'',filename,detail,rdate,cno,ckey)
mycursor.execute(sql,val)
mydb.commit()
##BC##
sdata="ID:"+str(maxid)+",User:"+name+", KYC Code:"+cno+", Key:"+ckey+",
RegDate:"+rdate
result = hashlib.md5(sdata.encode())
key=result.hexdigest()
mycursor1 = mydb.cursor()
mycursor1.execute("SELECT max(id)+1 FROM nt_blockchain")
maxid1 = mycursor1.fetchone()[0]
if maxid1 is None:
maxid1=1
pkey="00000000000000000000000000000000"
else:
mid=maxid1-1
mycursor1.execute('SELECT * FROM nt_blockchain where id=%s',(mid, ))
pp = mycursor1.fetchone()
pkey=pp[3]
sql2 = "INSERT INTO nt_blockchain(id,block_id,pre_hash,hash_value,sdata)
VALUES (%s, %s, %s, %s, %s)"
val2 = (maxid1,maxid,pkey,key,sdata)
mycursor1.execute(sql2, val2)
```

```
mydb.commit()
```

Request & Decrypt Certificate -

```
now = datetime.datetime.now()
rdate=now.strftime("%d-%m-%Y")
k=data[12]
efile="E"+data[3]
dfile=data[3]
if request.method=='POST':
key=request.form['key']
if k==key:
act="yes"
##BC##
sdata="ID:"+cid+",User:"+name+", KYC Code:"+cno+", RegDate:"+rdate
result = hashlib.md5(sdata.encode())
key=result.hexdigest()
mycursor1 = mydb.cursor()
mycursor1.execute("SELECT max(id)+1 FROM nt_blockchain")
maxid1 = mycursor1.fetchone()[0]
if maxid1 is None:
maxid1=1
pkey="00000000000000000000000000000000"
else:
mid=maxid1-1
mycursor1.execute('SELECT * FROM nt_blockchain where id=%s',(mid, ))
pp = mycursor1.fetchone()
pkey=pp[3]
sql2 = "INSERT INTO nt_blockchain(id,block_id,pre_hash,hash_value,sdata)
VALUES (%s, %s, %s, %s, %s)"
val2 = (maxid1,cid,pkey,key,sdata)
mycursor1.execute(sql2, val2)
mydb.commit()
#Decrypt
password_provided = k # This is input in the form of a string
password = password_provided.encode() # Convert to type bytes
```

```
salt = b'salt_' # CHANGE THIS - recommend using a key from os.urandom(16), must be of type bytes
kdf = PBKDF2HMAC(
algorithm=hashes.SHA256(),
length=32,
salt=salt,
iterations=100000,
backend=default_backend()
key = base64.urlsafe_b64encode(kdf.derive(password))
input_file = 'static/upload/'+efile
output_file = 'static/decrypted/'+dfile
with open(input_file, 'rb') as f:
data = f.read()
fernet = Fernet(key)
encrypted = fernet.decrypt(data)
with open(output_file, 'wb') as f:
f.write(encrypted)
```

CHAPTER 8
SYSTEM TESTING

6.1. SOFTWARE TESTING

Software testing is an essential step in ensuring the quality and reliability of any software system. In the case of the "Certificate Locker" system, some of the software testing activities that can be performed are:

1. **Unit Testing:** This involves testing individual units or components of the system to ensure that they are working as expected. For example, testing the encryption and decryption functions using PKI or the hash key verification functions using blockchain.
2. **Integration Testing:** This involves testing how different modules or components of the system work together. For example, testing the integration of the web admin module with the certificate holder and verifier modules.
3. **System Testing:** This involves testing the system as a whole to ensure that it meets the requirements and functions as expected. For example, testing the registration, login, and certificate upload functions for certificate holders, or the certificate verification function for verifiers.
4. **Performance Testing:** This involves testing the system's performance under different loads to ensure that it can handle the expected number of users and transactions. For example, testing the system's response time when multiple certificate holders are uploading certificates simultaneously.
5. **Security Testing:** This involves testing the system's security measures to ensure that they are effective in protecting sensitive data and preventing unauthorized access. For example, testing the encryption and decryption functions to ensure that certificates cannot be accessed by unauthorized users.

6.2. Testing Methodologies

Black box testing and white box testing are two commonly used methodologies for software testing.

- **Black Box Testing**

Black box testing is a testing approach that focuses on the behavior and functionality of the software without considering its internal structure or code. The tester performs tests by providing input values to the system and observing the corresponding outputs, with the objective of verifying that the system works as expected and meets the specified requirements. In the case of Certificate Locker, black box testing would involve testing the user interface, such as verifying that users are able to register and log in successfully, upload and access their certificates, and request and receive certificate verifications.

- **White Box Testing**

White box testing, on the other hand, is a testing approach that examines the internal workings and structure of the software, including its code, data structures, and algorithms. The tester performs tests by analysing the code and its execution to identify potential errors, with the objective of verifying that the system works correctly and efficiently. In the case of Certificate Locker, white box testing would involve analysing the system's encryption and decryption methods, as well as its use of PKI and blockchain technology, to verify that they are secure and functioning properly. In practice, both black box and white box testing are typically used in combination to thoroughly test a system. Black box testing is useful for identifying issues related to functionality and usability, while white box testing is useful for identifying issues related to security and performance.

6.3. Test Cases and Expected Results

Test Case ID: TC001

Input:

- Certificate Holder registers for an account with valid information.

Expected Result:

- Web Admin approves the registration request.
- Certificate Holder receives a notification that their account has been approved.
- Certificate Holder is able to log in to their account.

Test Case ID: TC002

Input:

- Certificate Holder uploads a valid certificate.

Expected Result:

- Certificate Holder receives a notification that the certificate has been successfully uploaded.
- Certificate Holder is able to view the uploaded certificate in their account.

Test Case ID: TC003

Input:

- Certificate Holder requests verification for a certificate.

Expected Result:

- Certificate Verifier receives a notification of the verification request.
- Certificate Holder receives a verification link with the certificate and private key.
- Certificate Verifier is able to use the private key to verify the authenticity and integrity of the certificate.

Test Case ID: TC004

Input:

- Certificate Holder attempts to upload an invalid certificate file format.

Expected Result:

- Certificate Holder receives an error message stating that the file format is invalid and the upload is not processed.
- Certificate Holder is not able to view the invalid certificate in their account.

Test Case ID: TC005

Input:

- Certificate Verifier attempts to verify an invalid certificate with an incorrect private key.

Expected Result:

- Certificate Verifier receives an error message stating that the private key is incorrect and the verification is not processed.
- Certificate Verifier is not able to verify the invalid certificate.

Test Case ID: TC006

Input:

- Certificate Holder attempts to log in with an incorrect password.

Expected Result:

- Certificate Holder receives an error message stating that the password is incorrect and is not able to log in.
- Certificate Holder is not able to access their account.

Test Case ID: TC007

Input:

- Certificate Verifier attempts to log in with invalid credentials.

Expected Result:

- Certificate Verifier receives an error message stating that the credentials are invalid and is not able to log in.
- Certificate Verifier is not able to access the verification system.

6.4. Test Report

Introduction

This testing report outlines the results of the testing conducted on the Certificate Locker system, which is designed to provide secure storage and accessing of educational digital certificates for students. The system uses PKI for encrypting and decrypting the certificates and Blockchain for verifying the integrity of the certificates. The testing was performed to ensure that the system functions as intended and meets the requirements specified by the stakeholders.

Test Objective

The main objective of the testing was to verify that the Certificate Locker system functions correctly and meets the requirements of the stakeholders. Specifically, the testing aimed to ensure that the system:

- Allows Certificate Holder to register, upload, and access their certificates securely.
- Allows Certificate Holder to verify the integrity of their certificates using Hash Key provided by Blockchain.
- Allows Certificate Verifier to verify the authenticity and integrity of the certificates using Private Key provided by Certificate Holder.
- Allows Web Admin to approve registration requests from Certificate Holders and Certificate Verifiers and integrate the system into Blockchain.

Test Scope

The testing was conducted on the Certificate Locker system, including its features for Certificate Holder, Certificate Verifier, and Web Admin. The testing covered the following areas:

- User registration and login process.
- Certificate uploading and accessing process.
- Certificate verification process.
- Integration of the system into Blockchain.

Test Environment

The testing was conducted in a controlled environment using the following tools and technologies:

- Operating System: Windows 10
- Browser: Google Chrome
- Tools: PyTest

Test Result

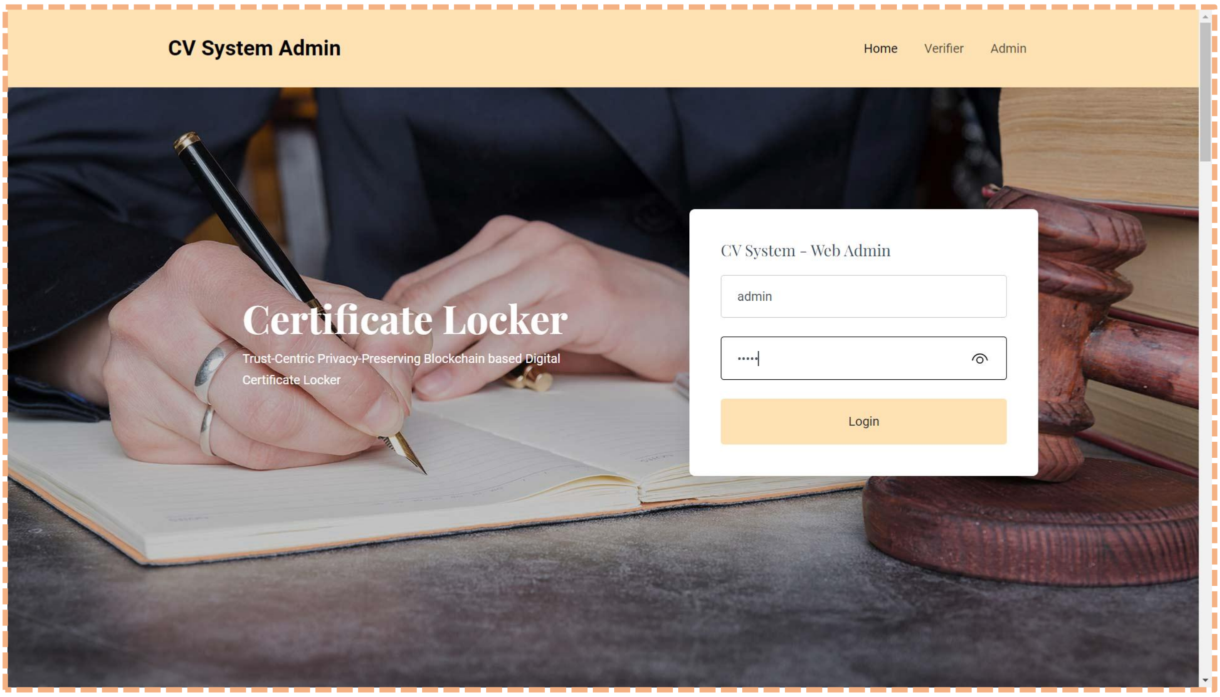
Overall, the Certificate Locker system functioned as intended and met the requirements of the stakeholders. All the features were tested, and the results are as follows:

- User registration and login process: The registration and login process for all user types (Certificate Holder, Certificate Verifier, and Web Admin) worked without any issues.
- Certificate uploading and accessing process: Certificate Holder was able to upload and access their certificates securely, and the system used PKI to encrypt and decrypt the certificates.
- Certificate verification process: Certificate Holder was able to verify the integrity of their certificates using the Hash Key provided by Blockchain. Certificate Verifier was able to verify the authenticity and integrity of the certificates using the Private Key provided by the Certificate Holder.
- Integration of the system into Blockchain: The system was successfully integrated into Blockchain, and Web Admin was able to approve registration requests from Certificate Holders and Certificate Verifiers.

Test Conclusion

The Certificate Locker system has been tested and found to be functioning correctly and meeting the requirements of the stakeholders. No major issues were found during the testing process. However, minor issues related to user interface and error handling were identified and will be addressed in future updates. Overall, the system is ready for deployment and use.

CHAPTER 9
SCREENSHOTS



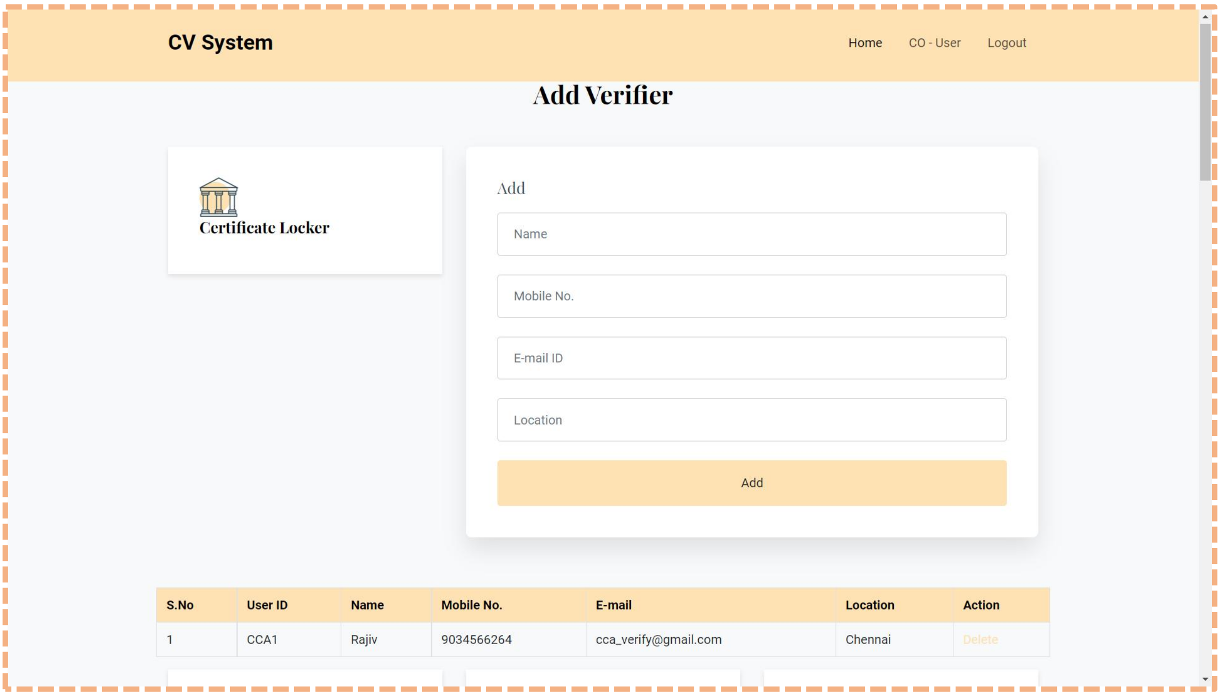
Certificate Locker

Trust-Centric Privacy-Preserving Blockchain based Digital Certificate Locker

CV System - Web Admin

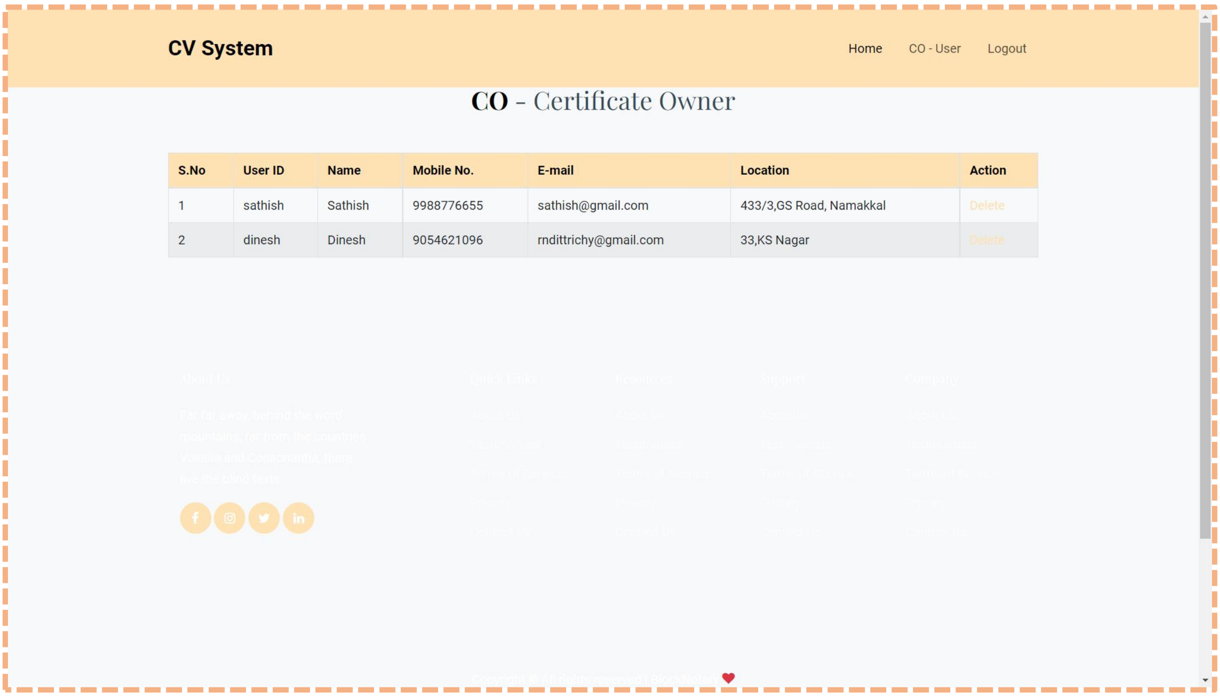
admin

•••••



Add Verifier

S.No	User ID	Name	Mobile No.	E-mail	Location	Action
1	CCA1	Rajiv	9034566264	cca_verify@gmail.com	Chennai	Delete



CV System

CO - Certificate Owner

S.No	User ID	Name	Mobile No.	E-mail	Location	Action
1	sathish	Sathish	9988776655	sathish@gmail.com	433/3,GS Road, Namakkal	Delete
2	dinesh	Dinesh	9054621096	rndittrichy@gmail.com	33,KS Nagar	Delete

About Us

Far far away, behind the word mountains, far from the countries Vokalia and Consonantia, there live the blind texts.

Quick Links

About Us
Testimonials
Terms of Service
Privacy
Contact Us

Resources

About Us
Testimonials
Terms of Service
Privacy
Contact Us

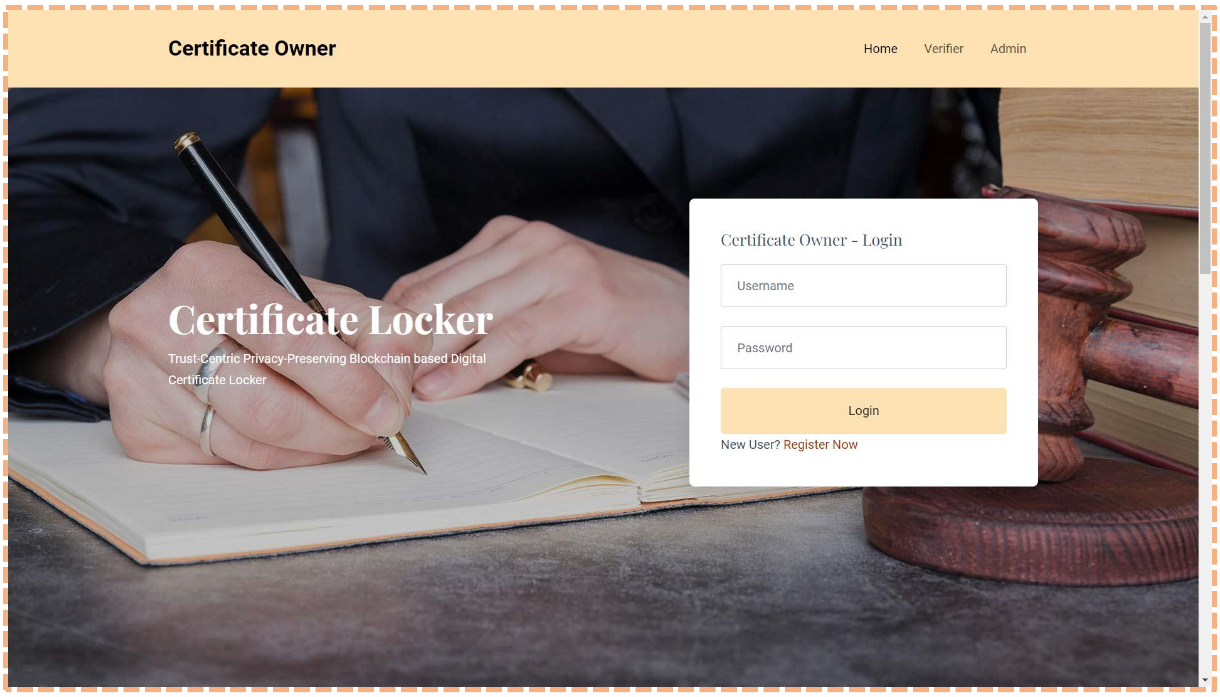
Support

About Us
Testimonials
Terms of Service
Privacy
Contact Us

Company

About Us
Testimonials
Terms of Service
Privacy
Contact Us

Copyright © All rights reserved | BlockNotary ❤



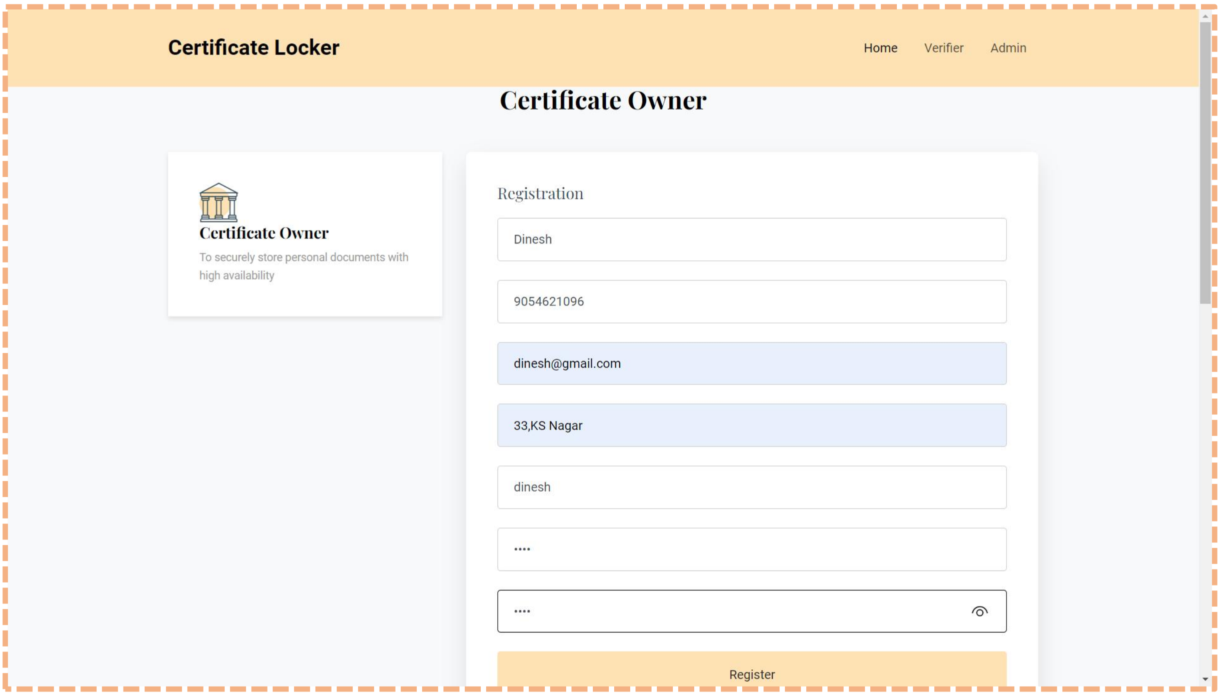
Certificate Owner

Certificate Locker

Trust-Centric Privacy-Preserving Blockchain based Digital Certificate Locker

Certificate Owner - Login

New User? Register Now



Certificate Owner

To securely store personal documents with high availability

Registration

Dinesh

9054621096

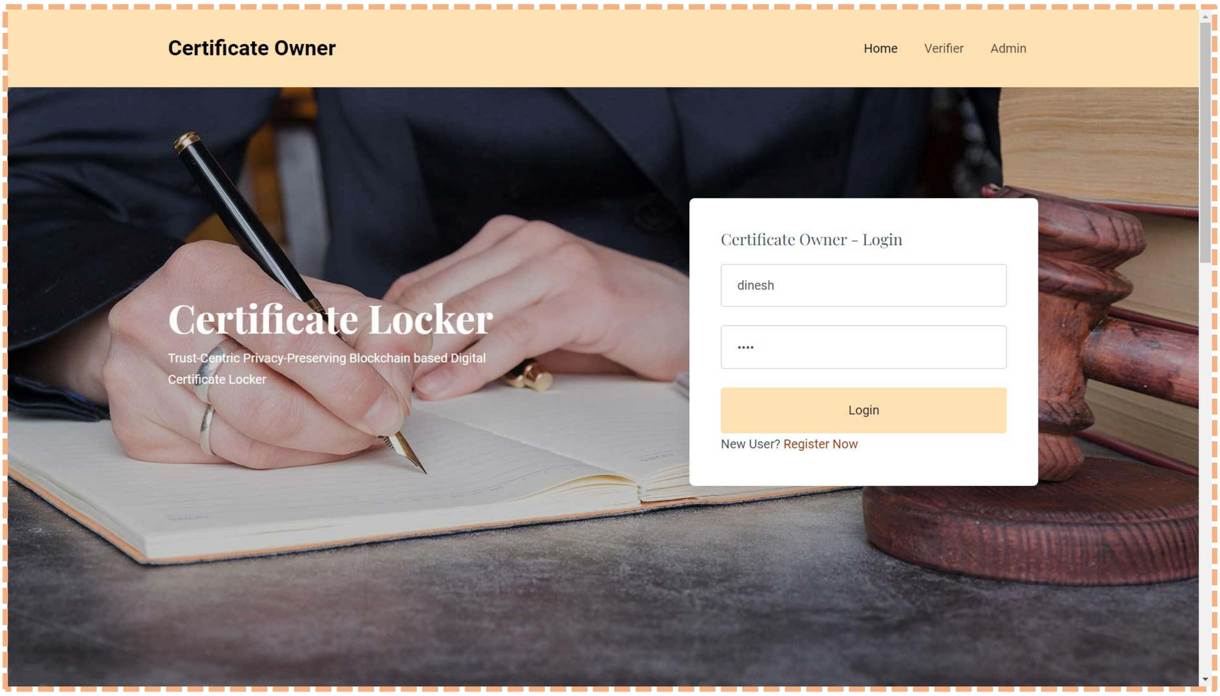
dinesh@gmail.com

33,KS Nagar

dinesh

••••

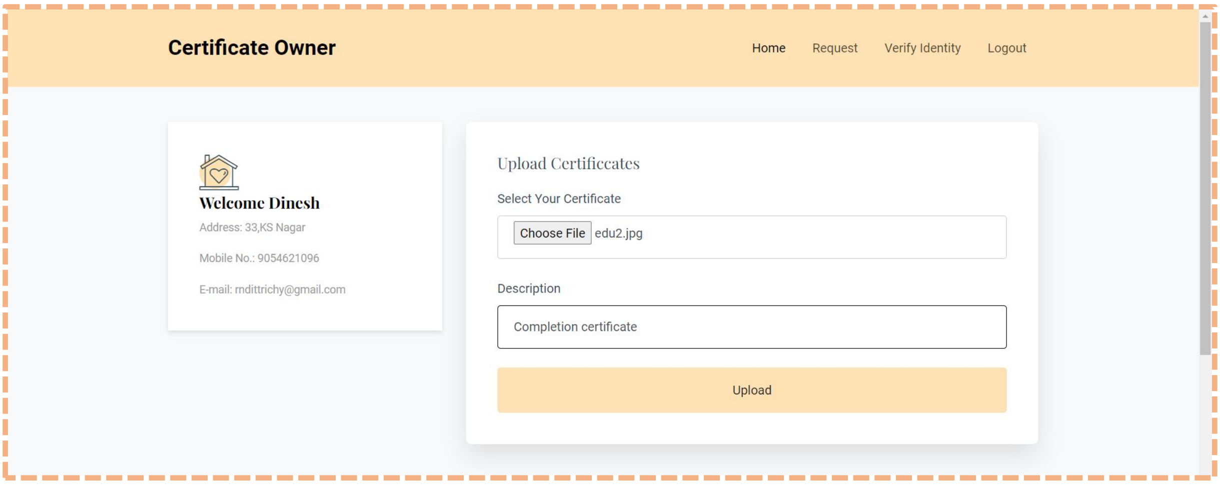
••••


Certificate Owner
Home
Verifier
Admin
Certificate Locker
Trust-Centric Privacy-Preserving Blockchain based Digital Certificate Locker

Certificate Owner - Login

dinesh

••••



Welcome Dinesh

Address: 33,KS Nagar

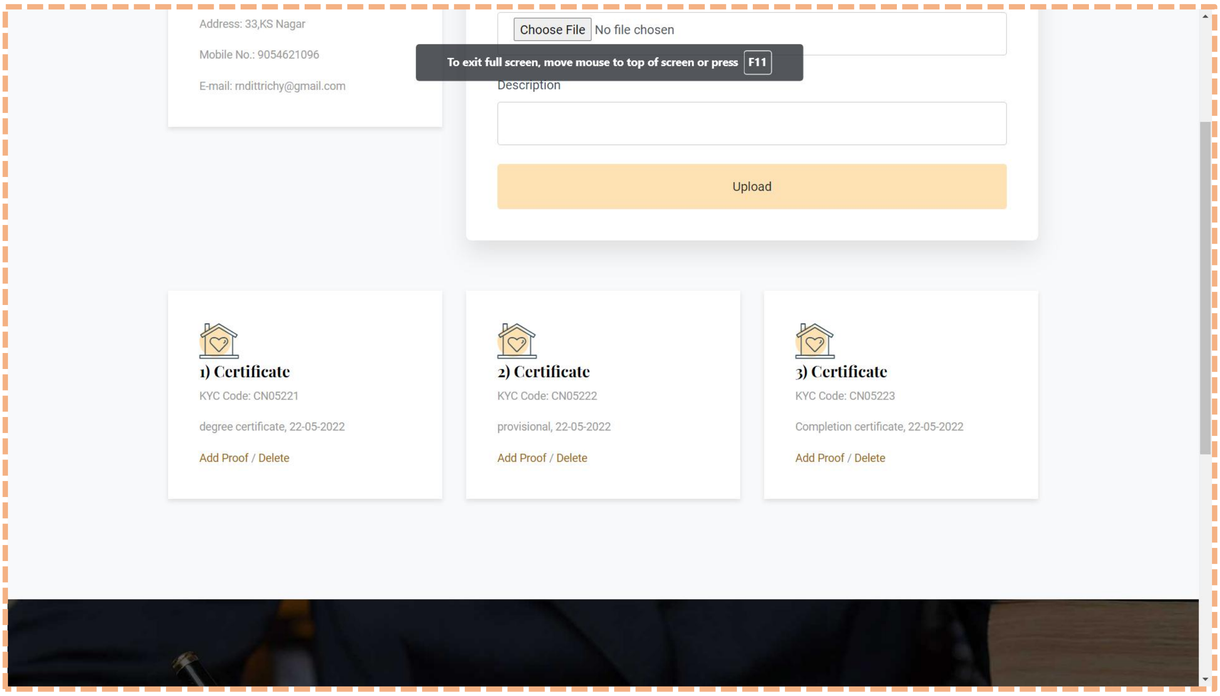
Mobile No.: 9054621096

E-mail: mdittrichy@gmail.com

Upload Certificcates

Select Your Certificate: edu2.jpg

Description: Completion certificate



Address: 33,KS Nagar

Mobile No.: 9054621096

E-mail: mdittrichy@gmail.com

Choose File No file chosen

To exit full screen, move mouse to top of screen or press F11

Description

1) Certificate

KYC Code: CN05221

degree certificate, 22-05-2022

Add Proof / Delete

2) Certificate

KYC Code: CN05222

provisional, 22-05-2022

Add Proof / Delete

3) Certificate

KYC Code: CN05223

Completion certificate, 22-05-2022

Add Proof / Delete

CHAPTER 10
CONCLUSION

The proposed system is a consortium blockchain among universities, their affiliated colleges, autonomous colleges, and the companies. Typically, universities first add the students' certificates and subsequently the companies or any other verifier can verify the credentials by using student's registration number or name. The data stored in a blockchain will be protected as no one can tamper it or add new transactions to it with a back date. The generated unique ID for each transaction is later used to verify the certificates. This system can be used by all the universities and colleges, in order to provide extra security to the certificates and the students' data. The problem of fake certificates can be eradicated and there will be no question of its validation.

CHAPTER 11
FUTURE ENHANCEMENT

In the future, this can be extended to provide integrity to any type of documents not only to the education sector but also to government sectors where a digital document time stamp is required. Not only to store the student marks information but also to store their employment and experience data, and can also be tracked by using this proposed system.

REFERENCES

1. Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang, "An Overview of Block chain Technology: Architecture Consensus and Future Trends", *2017 IEEE International Congress on Big Data (BigData Congress)*, pp. 557-564, 2017.
2. Dylan Yaga, Peter Mell, Nik Roby and Karen Scarfone, "Block chain Technology Overview", 2019 National Institute of Standards and Technology Cryptography and Security.
3. A Alammary, S Alhazmi, M Almasri and S. Gillani, "Block chain-Based Applications in Education: A Systematic Review", Applied Sciences, vol. 9, no. 12, pp. 2400, 2019, [online] Available: https://doi.org/10.3390/app9122400.
4. Q. Zheng, Y. Li, P. Chen and X. Dong, "An Innovative IPFS-Based Storage Model for Blockchain", 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 704-708, 2018.
5. M. Turkanović, M. Hölbl, K. Košič, M. Heričko and A. Kamišalić, "EduCTX: A Blockchain-Based Higher Education Credit Platform", IEEE Access, vol. 6, pp. 5112-5127, 2018
6. H. Li and D. Han, "EduRSS: A Block chain-Based Educational Records Secure Storage and Sharing Scheme", IEEE Access, vol. 7, pp. 179273-179289, 2019.
7. Emanuel Estrela Bessa and Joberto Martins, "A Block chain-based Educational Record Repository", 2019 CoRR abs 1904.00315.
8. A. Alkouz, A. Hai Yasien, A. Alarabeyyat, K. Samara and M. Al-Saleh, "EPPR: Using Block chain For Sharing Educational Records" in 2019 Sixth HCT Information Technology Trends (ITT), Ras Al Khaimah, United Arab Emirates, pp. 234-239, 2019.
9. T. Kanan, A. T. Obaidat and M. Al-Lahham, "Smart Cert Block Chain Imperative for Educational Certificates", 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), pp. 629-633, 2019.

10. Yaoqing Liu, Guchuan Sun and Stephanie. Schuckers, "Enabling Secure and Privacy-Preserving Identity Management via Smart Contract", pp. 1-8, 2019.

12.1. Book References

1. Python Crash Course: A Hands-On, Project-Based Introduction to Programming by Eric Matthes
2. Flask Web Development: Developing Web Applications with Python by Miguel Grinberg
3. Learning MySQL: Get a Handle on Your Data by Seyed M.M. Tahaghoghi and Hugh E. Williams
4. Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained, 2nd Edition by Imran Bashir
5. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython by Wes McKinney
6. JSON at Work: Practical Data Integration for the Web by Tom Marrs and O'Reilly Media Inc.

12.2. Web References

Python:

- Official Python documentation: https://docs.python.org/
- Python for Data Science Handbook: https://jakevdp.github.io/PythonDataScienceHandbook/

Flask:

- Flask documentation: https://flask.palletsprojects.com/en/2.1.x/
- Flask Mega-Tutorial: https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world

MySQL:

- MySQL documentation: https://dev.mysql.com/doc/
- MySQL Tutorial: https://www.mysqltutorial.org/

Blockchain:

- Blockchain Basics: https://www.ibm.com/blockchain/what-is-blockchain
- Blockchain Python tutorial:

https://developer.ibm.com/tutorials/blockchain-develop-python-smart-contract-fabric-network/

JSON:

- JSON documentation: https://www.json.org/json-en.html
- Working with JSON in Python: https://realpython.com/python-json/