CERTIFICATE VALIDATION USING BLOCKCHAIN

A Major Project Report

Submitted to

Jawaharlal Nehru Technological University, Hyderabad

In partial fulfillment of the requirements for the

award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

BUDDALA RAMYA	**(18VE1A0566)**
THALLADA HARIKA	**(18VE1A05B4)**
BEERAM SUPRIYA	**(18VE1A0564)**
NASIHA TABASSUM	**(19VE5A0510)**

Under the Guidance

of

Mr A. SRINIVASA REDDY

ASSISTANT PROFESSOR

SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(Affiliated to JNTUH, Approved by A.I.C.T.E and Accredited by NAAC, New Delhi)

Bandlaguda, Beside InduAranya, Nagole,

Hyderabad-500068, Ranga Reddy Dist.

2018-2022

SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project Report on **"Certificate Validation Using Blockchain"** submitted by **Buddala Ramya, Thallada Harika, Beeram Supriya, Nasiha Tabassum** bearing Hall ticket Nos. **18VE1A0566, 18VE1A05B4, 18VE1A0564, 19VE5A0510** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **COMPUTER SCIENCE AND ENGINEERING** from Jawaharlal Nehru Technological University, Kukatpally, Hyderabad for the academic year 2021-2022 is a record of bonafide work carried out by them under our guidance and Supervision.

Project Coordinator	**Head of the Department**
Dr. S. VENKATA ACHUTA RAO	**Dr. SHAIK ABDUL NABI**
Professor	**Professor**
Internal Guide	**External Examiner**
Mr. A. SRINIVASA REDDY	
Assistant Professor	

SREYAS INSTITUTE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, **Buddala Ramya, Thallada Harika, Beeram Supriya, Nasiha Tabassum** bearing **Roll Nos 18VE1A0566, 18VE1A05B4, 18VE1A0564, 19VE5A0510** hereby declare that the Major Project titled **CERTIFICATE VALIDATION USING BLOCKCHAIN** done by us under the guidance of **Mr. A. SRINIVASA REDDY, Assistant Professor** which is submitted in the partial fulfillment of the requirement for the award of the B. Tech degree in **Computer Science and Engineering** at **Sreyas Institute of Engineering and Technology** for Jawaharlal Nehru Technological University, Hyderabad is our original work.

Buddala Ramya	**18VE1A0566**
Thallada Harika	**18VE1A05B4**
Beeram Supriya	**18VE1A0564**
Nasiha Tabassum	**19VE5A0510**

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without mention of the people who made it possible through their guidance and encouragement crowns all the efforts with success.

We take this opportunity to acknowledge with thanks and deep sense of gratitude to **Mr. A. Srinivasa Reddy, Assistant Professor, Department of Computer Science and Engineering** for his constant encouragement and valuable guidance during the Project work.

A Special vote of Thanks to **Dr. Shaik Abdul Nabi, Head of the Department** and **Dr. S. Venkata Achuta Rao, Project Co-Ordinator** who has been a source of Continuous motivation and support. They had taken time and effort to guide and correct us all through the span of this work.

We owe very much to the **Department Faculty, Principal** and the **Management** who made our term at Sreyas a stepping stone for our career. We treasure every moment we had spent in the college.

Last but not the least, our heartiest gratitude to our parents and friends for their continuous encouragement and blessings. Without their support this work would not have been possible.

Buddala Ramya	**18VE1A0566**
Thallada Harika	**18VE1A05B4**
Beeram Supriya	**18VE1A0564**
Nasiha Tabassum	**19VE5A0510**

ABSTRACT

In this project to secure academic certificate and for accurate management and to avoid forge certificate we are converting all certificates into digital signatures and this digital signatures will be stored in Blockchain server as this Blockchain server support tamper proof data storage and nobody can hack or alter its data and if by an chance if its data alter then verification get failed at next block storage and user may get intimation about data alter.

In Blockchain technology same transaction data stored at multiple server with hash code verification and if data alter at one server then it will detected from other server as for same data hash code will get different. For example in Blockchain technology data will be stored at multiple servers and if malicious users alter data at one server then its hash code will get changed in one server and other servers left unchanged and this changed hash code will be detected at verification time and future malicious user changes can be prevented.

In Blockchain each data will be stored by verifying old hash codes and if old hash codes remain unchanged then data will be consider as original and unchanged and then new transaction data will be appended to Blockchain as new block. For each new data storage all blocks hash code will be verified.

KEYWORDS: Block chain, digital certificate, hashing, hyper ledger, Segmentation Algorithm, SHA-512.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

S.NO	ABBREVIATION	STANDS FOR
1	SRS	Software Requirement Specification
2	QA	Quality Assurance
3	UML	Unified Modelling Language
4	PER	Primary End-User Representative
5	PDR	Preliminary Design Review
6	ODBC	Open Database Connectivity
7	JDBC	Java Database Connectivity
8	EVM	Ethereum Virtual Machine
9	OMT	Object Modelling Technique
10	SDLC	Software Development Life Cycle
11	OOSE	Object Oriented Software Engineering
12	TPS	Transactions Per Second
13	IAM	Identity and Access Management

LIST OF FIGURES

LIST OF SCREENSHOTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The project consists in designing and implementing the system which covered the above solutions. The project also involves a comprehensive evaluation of the system security, and the assessment outcomes provide compelling evidence to prove that implementation is practical, reliable, secured, which might give some hints of important architectural considerations about the security attributes of other blockchain-based systems.

In this section, we discuss the implementation from the point of view of system architecture, database architecture. The system architecture and database architecture show how the system is designed from the engineering point of view.

The issuing applications are responsible for the main business logic which include the certificates applying, examining, signing and issuing. The issuing applications are designed to merge the hash of the certificate in a Merkle tree and send the Merkle root to Blockchain amidst signing by the majority of community members. Also, the issuing applications involved the revocation of certificate. The issuing applications are responsible for the main business logic which includes the applying for, examining, signing and issuing of the certificates. The issuing applications are designed to merge the hash of the certificate with a Merkle tree and send the Merkle root to the Blockchain. Also, the issuing applications deal with the revocations of certificates.

The verification application focuses on checking the authenticity and integrity of the certificates that have been issued. It includes two main components: a web-based page and an Android-based application. They use the same mechanism, and fetch the transaction message through the blockchain API and compare the transaction message with the verification data from the receipt. The mechanism can be briefly described in the following way: check the authentication code is valid; check the hash with the local certificate; confirm the hash is in the Merkle tree; ensure the Merkle root is in the blockchain; verify the certificate has not been revoked; validate the expired date of the certificate. Also, it has to be mentioned that for the convenience of sharing the certificates, the Android-based application allows for verification of the documents by scanning the QR code directly. The blockchain acts as the infrastructure of trust and a distributed database for saving the authentication data. Typically, the authentication data consist of the Merkle root generated using hashed data from thousands of certificates. JSON-based certificates and provides high availability and scalability.

Advances in information technology, the wide availability of the Internet, and common usage of mobile devices have changed the lifestyle of human beings. Virtual currency, digital coins originally designed for use online, has begun to be extensively adopted in real life. Because of the convenience of the Internet, various virtual currencies are thriving, including the most popular Bitcoin, Ether, and Ripple the value of which has surged recently. People are beginning to pay attention to blockchain, the backbone technology of these revolutionary currencies. Blockchain features a decentralized and incorruptible database that has high potential for a diverse range of uses.

Blockchain is a distributed database that is widely used for recording distinct transactions. Once a consensus is reached among different nodes, the transaction is added to a block that already holds records of several transactions. Each block contains the hash value of its last counterpart for connection. All the blocks are connected and together they form a blockchain Data are distributed among various nodes (the distributed data storage) and are thus decentralized. Consequently, the nodes maintain the database together. Under blockchain, a block becomes validated only once it has been verified by multiple values.

1.1 MOTIVATION

Counterfeit academic certificates have been a longstanding issue in the academic community. Not until the Massachusetts Institute of Technology Media Lab released their project of Block-certs, a technique which is mainly implemented by conflating the hash value of local files to the blockchain but remains numerous issues, did an effective technological approach protecting authentic credential certification and reputation appear.

Based on Block-certs, a series of cryptographic solutions are proposed to resolve the issues above, including, utilizing a multi-signature scheme to ameliorate the authentication of certificates; exerting a safe revocation mechanism to improve the reliability of certificates revocation; establishing a secure federated identification to confirm the identity of the issuing institution.

1.2 OBJECTIVE

The Objective of blockchain is to allow digital information to be recorded and distributed, but not edited. In this way, a blockchain is the foundation for immutable ledgers, or records of transactions that cannot be altered, deleted, or destroyed.

1.3 SCOPE

The Scope of this project is to take all the details of the user including the certificate and save those details in a block. A hash value is then generated by the system and a block number is given. A digital code is given to that particular certificate used to verify the certificate when uploaded. Then immediately when a user uploads the certificate another hash is generated and then compared with the hash value present with us thus we can know whether the certificate is original or not. Our proposed system will reduce the disadvantages of existing system and its help to reduce the Block-certs, a series of cryptographic solutions which are proposed to resolve the issues, including, utilizing a multi-signature scheme to ameliorate the authentication of certificates; exerting a safe revocation mechanism to improve the reliability of certificates revocation; establishing a secure federated identification to confirm the identity of the issuing institution.

1.4 OUTLINE

The verification system is very useful for the user to verify the certificate and inform whether it is an original certificate or a fake one. In the digital world, SSLC, HSC, academic background, everything is digitized without exception in educational institutions and provided to students. It is difficult for students to maintain their academic achievement. For organizations and institutions, verifying and validating certificates is tedious and cumbersome. Our project helps to verify certificates using blockchain system and provides security. We improvised training and test features for different set of certificates based on features and finally the design is compared with gradient boosting algorithm. An encryption feature with data hash key is randomly generated with blockchain feature.

CHAPTER 2
LITERATURE SURVEY

The project focuses on building an immutable certificate generation as well as a validation system. For this, we have referred few previously published papers and works of the various individual in this field. Our Literature Survey mainly focused on Blockchain Technology, an advanced Storage System, and Digital Certificate Validations.

Our first paper was titled, An Overview of Blockchain Technology [1] which provided in-depth knowledge regarding Blockchain. It introduced various terms regarding this technology and the most important concept called a smart contract. In the Blockchain, the hash of the data is stored in its preceding block, and it forms a long chain of nodes. If data is changed, its hash will change, and it won't match with the hash value stored in the previous block and hence letting us know about the tampering of data.

The second paper was titled, Blockchain and Smart Contract for Digital Certificate [2]. Their design consisted of 3 actors. First, there were institutions, second being the students, and last the service provider. The drawback of their method was that they were using 'one hash as a key', which makes it publicly accessible once they have the hash.

The Final Paper was a Blockchain-Based Identity Verification Model. Similar to the second and third paper their system consisted of an Issuing Authority who will generate the document, a hashing algorithm works over it, and its value is stored. Since other systems had public hash keys, they increased security by using asymmetric encryption.

Jiin-Chiou Cheng, Narn-Yih Lee, Chien Chi, and YiHua Chen(2018) ‘Blockchain and Smart Contract for Digital Certificate’ Developed a decentralized application and designed a certificate systembased on Ethereum blockchain. This technology was selected because it is incorruptible, encrypted, and trackable and permits data synchronization. By integrating the features of blockchain, the system improves the efficiency operations at each stage. The system saves on paper, cuts management costs, prevents document forgery, and provides accurate and reliable information on digital certificates.

Murat Yasin Kubilay, Mehmet SabırKiraz and Hacı Ali Mantar(2018) ‘CertLedger: A New PKI Model with Certificate Transparency Based on Blockchain’ The current trust model, CAs have the absolute responsibility to issue correct certificates for the designated subject.

However, CAs can still be compromised and fake but valid certificates can be issued due to inadequate security practices or non-compliance with the Certificate Policy(CP) and Certificate Practice Statement(CPS).

Marco Valdi, Franco Chiaraluce, Emanuele Frontoni, Luca Spalazzi(2017) ‘Certificate validation through public ledger and blockchain’ The project is proposed for the solution of addressing the issues of reliability and security of certificate revocation information. Its main advantages are in removing any single POF and being relatively simple to implement by leveraging existing open source platforms.

Rujia Li, Yifan Wu(2016) ‘Block chain based academic certificate authentication system’ The project consists in designing and implementing by conflating the hash value of local files to the blockchain but remains numerous issues, did an effective technological approach protecting authentic credential certification and reputation appear.

Ze Wang, Jiwu Jing, Daren Zha, JingqiangLin(2016) ‘Blockchain based certificate transparency and revocation transparency’ In this they maintain a database to record the certificates and revocation status information in the global certificate blockchain which is inherently append only, to achieve certificate transparency and limited grained revocation transparency.

2.1 EXISTING SYSTEM

The certificates are stored in centralized manner and verified manually, so it takes too much time to verify. There is no safety to the certificate that are given to any private sectors (banks). But the data may be changed, deleted or modified. Certificates are easily hacked and make duplicate of that certificate. Students bring their certificates on interview places. There is no security for certificates.

Disadvantages

- Validation is delayed.
- False sense of security.

2.2 PROPOSED SYSTEM

In this study, a blockchain certificate system was developed based on relevant technology. The system's application was programmed on the platform and is run by the EVM. In the system, three groups of users are involved, Schools or certification units grant certificates, have access to the system, and can browse the system database. When students fulfilled certain requirements, the authorities grant a certificate through the system. After the students have received their certificate, they are able to inquire about any certificate they have gained. The service

Advantages

- Unsupervised
- Faster and secure.
- Making use of blockchain and its hash key generation validates certificate with accuracy.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATIONS

3.1 INTRODUCTION

A software requirement specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide. Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven project, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirements specifications can help prevent software project failure.

Purpose

An SRS forms the basis of an organization's entire project. It sets out the framework that all the development teams will follow. It provides critical information to all the teams, including development, operations, quality assurance (QA) and maintenance, ensuring the teams are in agreement.

Counterfeit academic certificates have been a longstanding issue in the academic community. Not until the Massachusetts Institute of Technology Media Lab released their project of Block-certs, a technique which is mainly implemented by conflating the hash value of local files to the blockchain but remains numerous issues, did an effective technological approach protecting authentic credential certification and reputation appear.

Based on Block-certs, a series of cryptographic solutions are proposed to resolve the issues above, including, utilizing a multi-signature scheme to ameliorate the authentication of certificates; exerting a safe revocation mechanism to improve the reliability of certificates revocation; establishing a secure federated identification to confirm the identity of the issuing institution.

Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.

Product Scope

The core function of this project is to enable the model with the capability learn throughout its life. Rather than learning statically and train the model on the static dataset our model will be able to extend its capability throughout its lifetime. This model can learn as long as user has new data and wants to train with new data.

3.2 SDLC METHODOLOGIES

SDLC MODEL

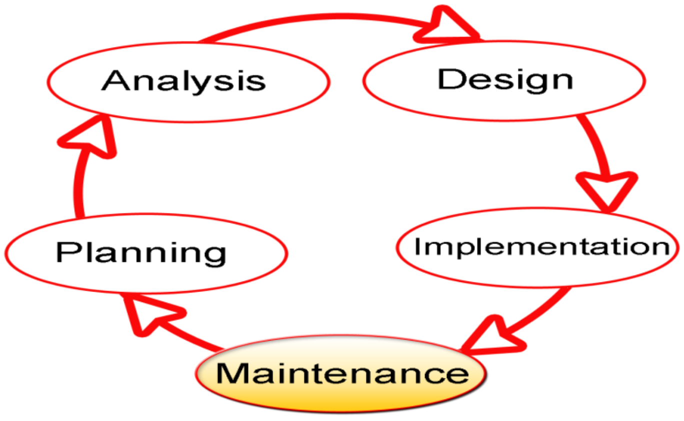


Fig.3.2.1 SDLC Model

The Software Development Life Cycle(SDLC) for small to medium database application development efforts.

This project uses iterative development lifecycle, where components of the application are developed through a series of tight iteration. The first iteration focus on very basic functionality, with subsequent iterations adding new functionality to the previous work and or correcting errors identified for the components in production.

- **Planning :** The planning phase of the SDLC is also when the project plan is developed that identifies, prioritizes, and assigns the tasks and resources required to build the structure for a project. With that said, this step culminates in a detailed project plan.

- **Analysis :** The analysis phase also gathers business requirements and identifies any potential risks. This step in SDLC also includes a feasibility study, which defines all fortes and weak points of the project to assess the overall project viability.

  The analysis phase is where multiple collected and processed items are examined, correlated, and given the necessary context the make them useful. This is where intelligence goes from just being loosely related pieces of data to a finished product that is useful for decision-making.
- **Design :** The Design Phase is an essential phase of the Software Development Life Cycle. The list of requirements that you develop in the definition phase is used to make design choices. In the design phase, one or more designs are created to achieve the project result. Depending on the project subject, the design phase products include dioramas, flow-charts, sketches, site trees, HTML screen designs, photo impressions, prototypes, and UML schemas.

  The project supervisors use these designs to choose the definitive design that you can produce in the project. The development phase follows it. Once you have selected the design in the definition phase, you cannot make changes in the project's later stage.
- **Implementation :** Implementation phase in SDLC is the process of configuring the software for certain conditions of use, as well as training customers to work with the product. This stage begins after the system has been tested and accepted by the company. At the time, a program is installed to support the intended business functions.

The six stages of the SDLC are designed to build on one another, taking outputs from the previous stage, adding additional effort, and producing results that leverage the previous effort and are directly traceable to the previous stages. During each stage, additional information is gathered or developed, combined with the inputs, and used to produce the stage deliverables. It is important to note that the additional information is restricted in scope, new ideas that would take the project in directions not anticipated by the initial set of high-level requirements or features that are out-of-scope are preserved for later consideration.

Too many software development efforts go awry when development team and customer personnel get caught up in the possibilities of automation. Instead of focusing on high priority features, the team can become mired in a sea of nice to have features that are not essential to solve the problem, but in themselves are highly attractive. This is the root cause of large percentage of failed and or abandoned development efforts and is the primary reason the development team utilizes the iterative model.

Roles and Responsibilities of PDR AND PER

The iterative lifecycle specifies two critical roles that act together to clearly communicate project issues and concepts between the end-user community and the development team.

Primary End-user Representative (PER)

The PER is a person who acts as the primary point of contact and principal approver for the end-user community. The PER is also responsible for ensuring that appropriate subject matter experts conduct end-user reviews in a timely manner.

PER-PDR Relationship

The PER and PDR are the brain trust for the development effort. The PER has the skills and domain knowledge necessary to understand the issues associated with the business processes to the supported by the application and has a close working relationship with the other members of the end-user community. The PDR has the same advantages regarding the application development process and the other members of the development team together; they act as the concentration points for knowledge about the application to be developed.

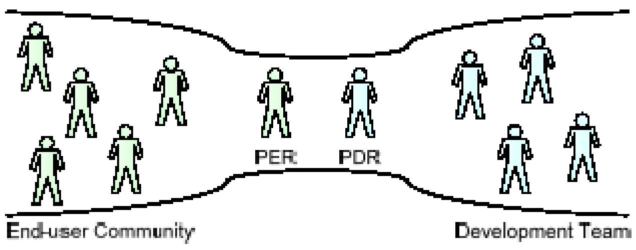


Fig.3.2.2 PER PDR

The objective of this approach is to create the close relationship that is characteristic of a software project with one developer and one end-user in essence, this approach the "pair programming" concept from Agile methodologies and extends it to the end-user community. While it is difficult to create close relationships between the diverse members of an end-user community and a software development team, it is much simpler to create a close relationship between the lead representatives for each group.

When multiple end-users are placed into relationship with multiple members of a development team, communication between the two groups degrades as the number of participants grows. In this model, members of end-user community may communicate with members of the development team as needed, but it is the responsibility of all participants to keep the PER and PDR apprised of the communications for example, this allows the PER and PDR to resolve conflicts that arise when two different end-users communicate different requirements for the same application feature to different members of the development team.

3.3 FUNCTIONAL REQUIREMENTS

a. Software Requirements

- Operating System - Windows 10
- IDE - Juypter Notebook
- Technologies - Python 3.6

b. Hardware Requirements

- **Processer -** Intel-i5
- **Ram -** 4GB
- **Storage -** 256GB

3.4 NON-FUNCTIONAL REQUIREMENTS

Performance Requirements

The performance requirements refer to static numerical requirements placed on the interaction between the users and the software.

Response Time

Average response time shall be less than 2 sec.

Recovery Time

In case of system failure, redundant system shall resume operations within 30 secs. Average repair time shall be less than 1 hr.

Start-Up/Shutdown Time

The system shall be operational within 1 minute of starting up.

Capacity

The system accommodates 4000 Concurrent Users.

Utilization of Resources

The system shall store in the database no more than one million transactions. If the database grows over this limit, old transaction shall be backed up and deleted from the operational database.

Security Requirements

The system will be made secure by assigning all users with separate registered ids i.e. each user will be responsible for his/her assigned id.

Software Quality Attributes

Reliability

The system shall be reliable i.e. in case the server crashes, a backup server will be there to work which will be maintained continuously.

CHAPTER 4
SYSTEM DESIGN

4.1 IMPORTANCE OF DESIGN

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. It is the process of defining software methods, functions, objects and overall structure and interaction of your code so that the resulting functionality will satisfy your users requirements. It allows you to do the best abstraction, to understand the requirements better and meet them better. This prevents redundancy and increases reusability. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed, design takes us towards how to satisfy the needs. The design of a system is perhaps the most critical factor affection the quality of the software; it has a major impact on the later phase, particularly testing, maintenance. The output of this phase is the design document. This document is similar to a blueprint for the solution and is used later during implementation, testing and maintenance. The design activity is often divided into two separate phases System Design and Detailed Design.

System Design also called top-level design sign aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. During, Detailed Design, the internal logic of each of the modules specification in system design is decided. During this phase, the details of the data is usually specified in a high-level design description language, which is independent of the target language in which the software will eventually be implemented.

In system design the focus is on identifying the modules, whereas during detailed design the focus is on designing the logic for each of the modules. During the system design activities, Developers bridge the gap between the requirements specification, produced during requirements elicitation and analysis, and the system that is delivered to the user.

System design is important for defining the product and its architecture. It is necessary for the interfaces, design, data, and modules to satisfy the system requirements. Thus, a good system design strategy is key for enabling optimal product development. The purpose of the System Design process is to provide sufficient detailed data and information about the system.

4.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

- Actors
- Business processes
- (logical) Components
- Activities
- Programming Language Statements
- Database Schemes
- Reusable software components

UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies· UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

The goal is for UML to become a common language for creating models of object oriented computer software. In the future, some form of method or process may also be added to; or associated with, UML.

Goals

The Primary goals in the design of the UML are as follows:

- Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
- Provide extendibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development process.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of OO tools market.
- Support higher level development concepts such as collaborations, frameworks, patterns and components.

Types Of UML Diagrams

The current UML standards call for 13 different types of diagrams: class, activity, object, use case, sequence, package, state, component, communication, composite structure, interaction overview, timing and deployment.

These diagrams are organized into two distinct groups: structural diagrams and behavioral or interaction diagrams.

Structural UML Diagrams

- Class diagram
- Package diagram
- Object diagram
- Component diagram
- Composite structure diagram
- Deployment diagram

Behavioral UML Diagrams

- Activity diagram
- Sequence diagram
- Use case diagram
- State diagram
- Communication diagram
- Interaction overview diagram
- Timing diagram

4.2.1 USE-CASE DIAGRAM

A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

The purpose of a use case diagram in UML is to demonstrate the different ways that a user might interact with a system. In UML, use-case diagrams model the behavior of a system and help to capture the requirements of the system. Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

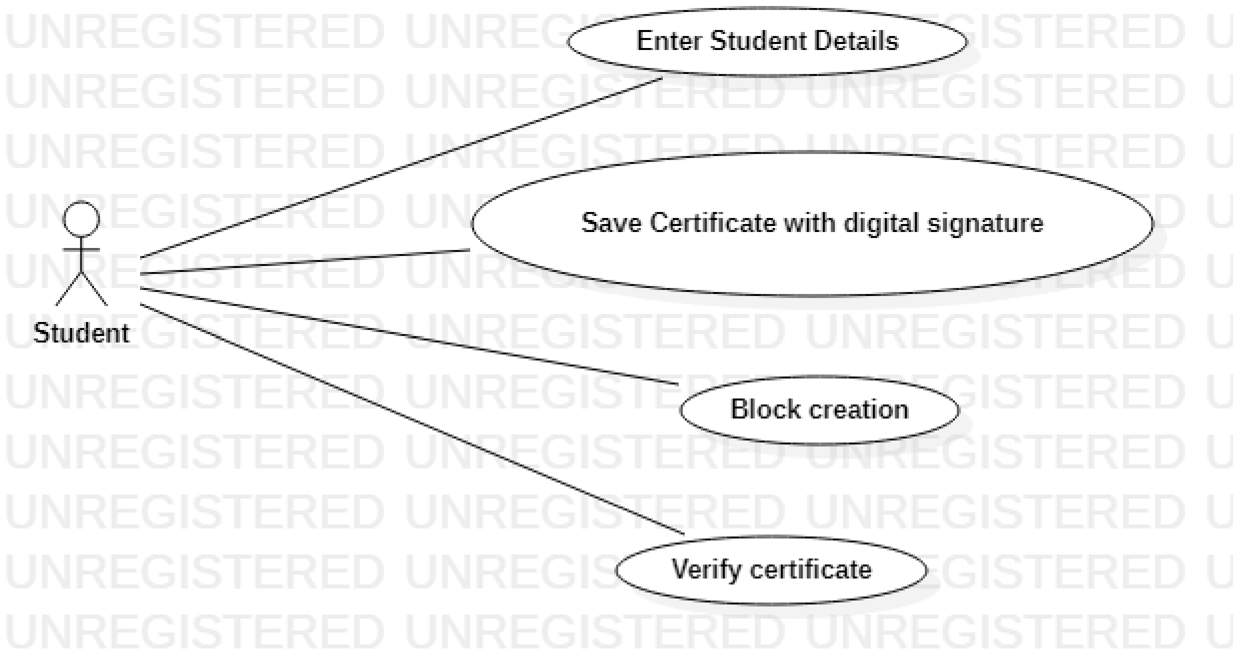


Fig.4.2.1 Use Case Diagram

4.2.2 SEQUENCE DIAGRAM

Sequence Diagrams represent the objects participating the interaction horizontally and time vertically. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

Sequence diagrams are a popular dynamic modeling solution in UML because they specifically focus on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

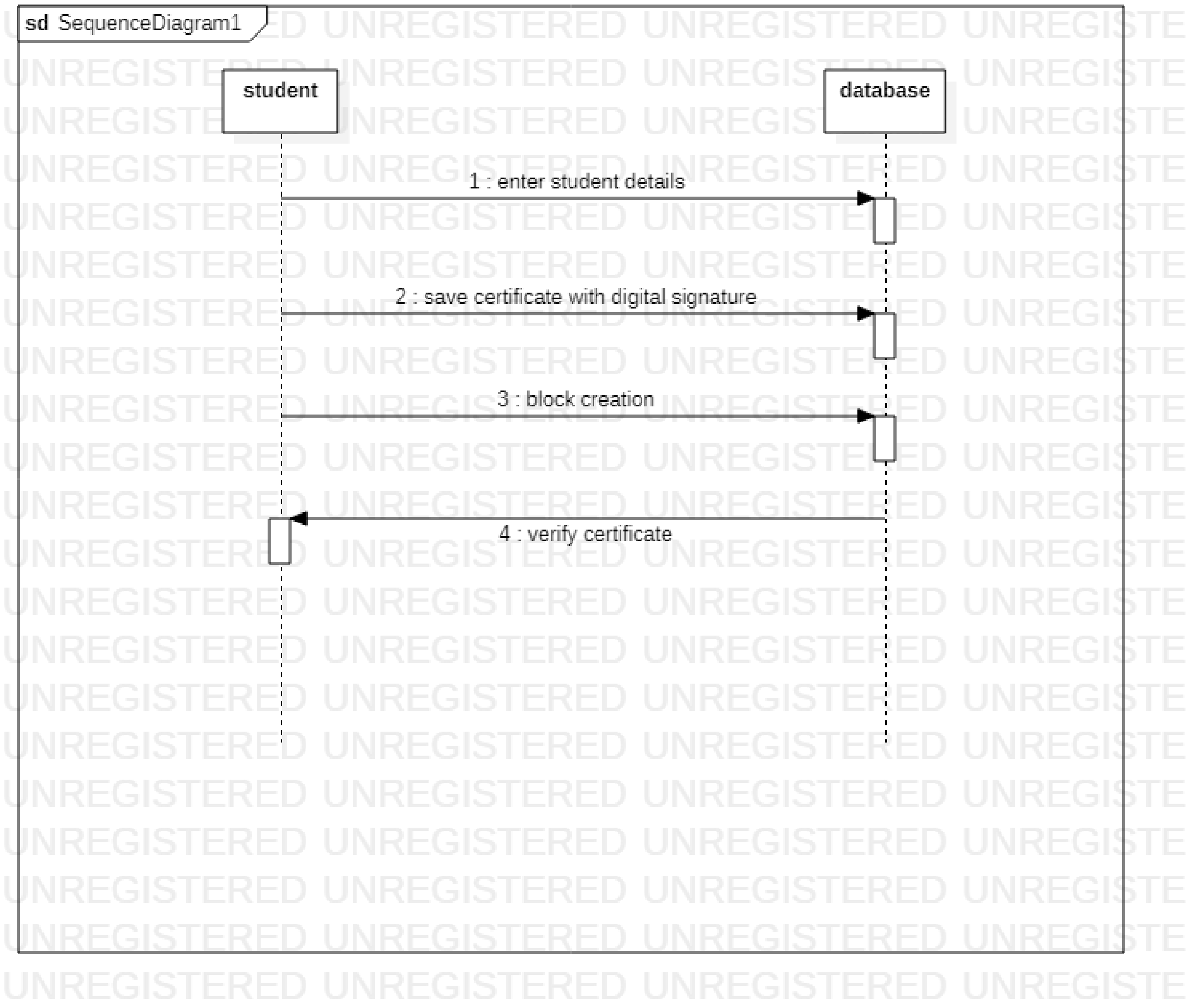


Fig.4.2.2 Sequence Diagram

4.2.3 ACTIVITY DIAGRAM

An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

It captures the dynamic behavior of the system. An activity diagram shows business and software processes as a progression of actions. These actions can be carried out by the people, software components or computers. Activity diagrams are used to describe business processes and use the cases as well as to document the implementation of system processes.

Activity diagram is used to show message flow from one activity to another. An activity diagram shows business and software processes as a progression of actions.

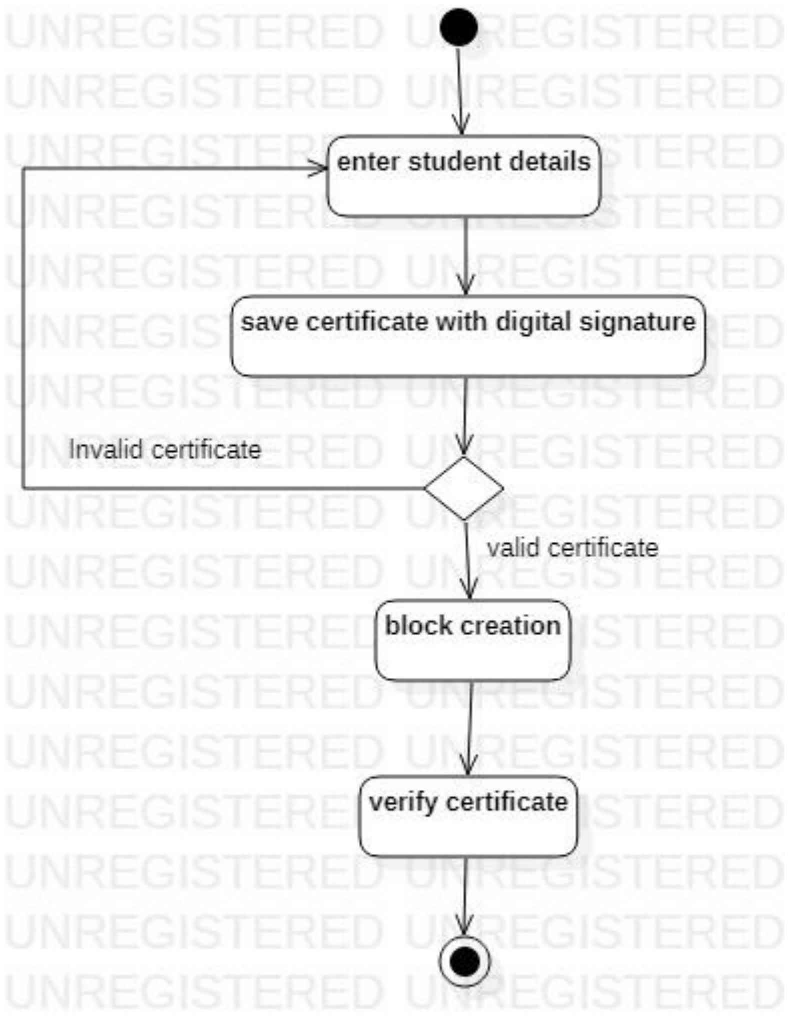


Fig.4.2.3 Activity Diagram

4.2.4 CLASS DIAGRAM

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.

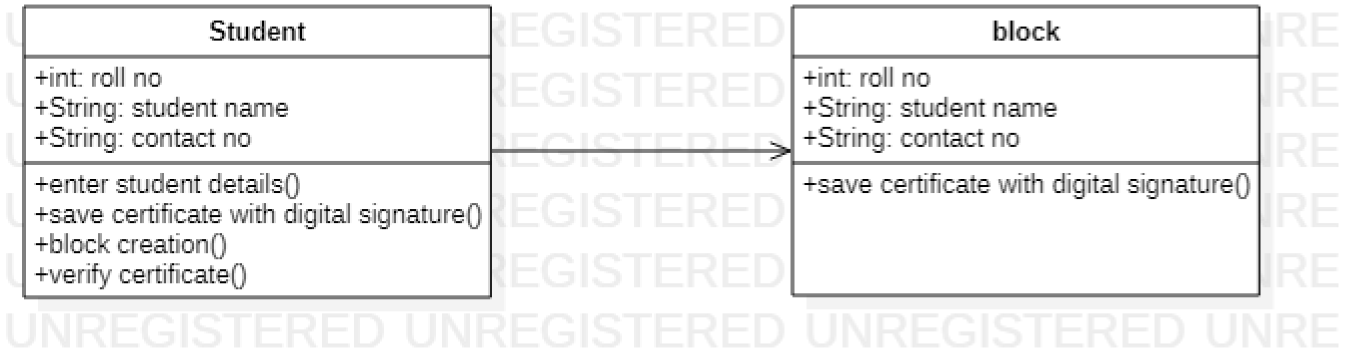


Fig.4.2.4 Class Diagram

4.2.5 SYSTEM ARCHITECTURE

A system architecture diagram would be used to show the relationship between different components. Usually they are created for systems which include hardware and software and these are represented in the diagram to show the interaction between them. However, it can also be created for web applications.

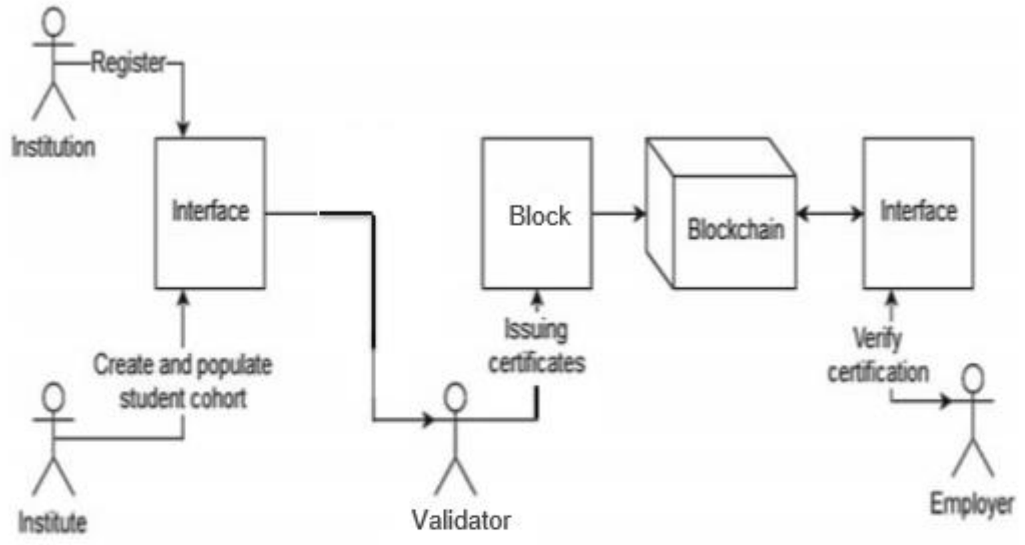


Fig.4.2.5 Architecture Diagram

4.3 SYSTEM STUDY

4.3.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 5
IMPLEMENTATION

5.1 MODULE DESCRIPTION

Implementation includes all those activities that take place to covert from old system to new system. The old system consists of manual operations, which is operated in a very difficult manner from the proposed system. A proper implementation is essential to provide a reliable system to meet the requirements of the organization.

- **Save Certificate with Digital Signature**

  Using this module admin user can upload student details and student academic certificate and then application convert certificate into digital signature and then signature and other student details will be saved in Blockchain database.

- **Verify Certificate**

  In this module verifier or companies or admin will take certificate from student and then upload to application and then application will convert certificate into digital signature and this digital signature will get checked/verified at Blockchain database and if matched found then Blockchain will retrieve all student details and display to verifier and if match not found then this certificate will be consider as fake or forge.

tkinter :- It is a standard GUI library for python
block :- Information is stored in it
blockchain :- High performance and security aware and higher degree of privacy
hashlib :- It is an interface for hashing messages easily
os :- It provides functions for interacting with the operating system
json :- Used to work with JSON data
random :- It is used to generate random numbers
base64 :- Used to encode and decode the data

PYTHON

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library

The biggest strength of Python is huge collection of standard library 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

Characteristics of PYTHON

- Easy Language. Python is an easy language.
- Readable. The Python language is designed to make developers life easy.
- Interpreted Language. Python is an interpreted language.
- Dynamically-Typed Language.
- Object-Oriented.
- Popular and Large Community Support.
- Open-Source.
- Large Standard Library.

Advantages of using Python

- **Easy to learn**

Python is considered to be one of the easiest programming languages to learn for beginners. It is a high-level programming language, meaning that it has a clear syntax that reads a lot like English.

Choosing a language that's quick to learn means you will spend less time troubleshooting your code and fixing bugs. Hence, you will have more time for actually learning how to code and create useful stuff.

- **Availability of support**

When you learn to code, you are going to run into difficulties – which you will overcome!

Because no matter how good you are at learning new things, sometimes your code just doesn't do what it's supposed to.

- **Large global community**

Speaking of how easy it is to find help for your Python projects, you can most often rely on the massive global community for help and support.

Since so many developers use Python, you can find solutions to a wide variety of problems quickly and easily.

Moreover, having a large pool of developers working with the language, you can easily network with other like-minded students or professional developers worldwide.

Hence, if you are entirely new to coding, make sure to see if there are Python study groups in your area!

- **In-demand skill in the job market**

If you want to learn to code, you're probably doing it at least partly because of the lucrative career prospects in the long run.

And while money shouldn't be your #1 motivator, it is a sweet little carrot that helps you keep going when things feel difficult.

- **Extensive libraries**

When you start building your first coding projects, you'll most probably use at least one Python library to get things done faster.

Libraries are essentially collections of Python code you can use for specific purposes.

And since most of your programs in a particular field will repeat at least some of your code, libraries help you avoid repeating yourself.

The Python standard library allows you to choose from a wide range of different modules according to what you need for your project. Each module lets you add a feature or functionality to your Python program without writing additional code.

For instance, let’s say two developers want to build a web application. They will use similar Python code snippets to create specific features for their dynamic web pages.

- **Powerful frameworks**

To speed up your workflow, you can choose from several open-source Python frameworks, too.

Say you want to build a web application. You can easily save time and simplify your development process by using powerful Python web frameworks such as Django or Flask, for example. They contain ready-to-use Python code for standard web app functionalities, such as creating and managing user accounts.

Similarly, you can speed up your desktop GUI app development with Python frameworks and toolkits such as PyQT, PyJs, or PyGUI, for instance.

- **Versatility**

Depending on what types of projects you want to work on in the future, you need to choose the right programming language to work with.

Because the thing is: any language is simply a means to an end. It allows you to solve specific types of problems with code.

- **Free and open-source**

Python is 100% free to use. You don’t need a particular subscription or a custom-built platform to start building projects with it.

- **Interpreted Language**

Python is an interpreted language which means that Python directly executes the code line by line. In case of any error, it stops further execution and reports back the error which has occurred.

How Python Works

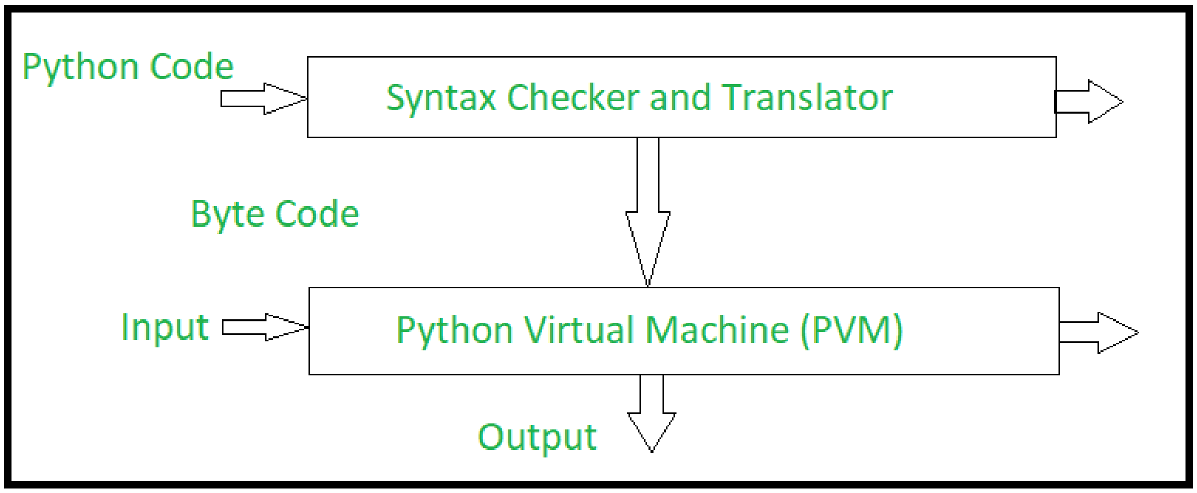


Fig.5.1 Working of PYTHON

Advantages Of Python Over Other Languages

- **Less Coding**

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

- **Affordable**

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

- **Python is for Everyone**

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

Disadvantages Of Python

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

- **Speed Limitation**

  We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

- **Weak in Mobile Computing and Browsers**

  While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

- **Design Restrictions**

  As you know, Python is dynamically-typed. This means that you don’t need to declare the type of variable while writing the code. It uses duck-typing. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

- **Underdeveloped Database Access Layers**

  Compared to more widely used technologies like JDBC (Java DataBase Connectivity) and ODBC (Open DataBase Connectivity)**,** Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

- **Simple**

  No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

History Of Python

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners[1], Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it. "Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

Block Chain

Blockchain is a system of recording information in a way that makes it difficult or impossible to change, hack, or cheat the system. A blockchain is essentially a digital ledger of transactions that is duplicated and distributed across the entire network of computer systems on the blockchain.

A blockchain is a digital ledger of transactions maintained by a network of computers in a way that makes it difficult to hack or alter. The technology offers a secure way for individuals to deal directly with each other, without an intermediary like a government, bank or other third party.

Categories Of Block Chain

At the most fundamental level, Block Chain can be categorized into four main types:

- Public Blockchain
- Private Blockchain
- Consortium Blockchain
- Hybrid Blockchain

Public Blockchain

A public blockchain is a non-restrictive, permission-less distributed ledger system. Anyone who has access to the internet can sign in on a blockchain platform to become an authorized node and be a part of the blockchain network. A node or user which is a part of the public blockchain is authorized to access current and past records, verify transactions or do proof-of-work for an incoming block, and do mining. The most basic use of public blockchains is for mining and exchanging cryptocurrencies. Thus, the most common public blockchains are Bitcoin and Litecoin blockchains. Public blockchains are mostly secure if the users strictly follow security rules and methods. However, it is only risky when the participants don't follow the security protocols sincerely.

Example: Bitcoin, Ethereum, Litecoin

Private Blockchain

A private blockchain is a restrictive or permission blockchain operative only in a closed network. Private blockchains are usually used within an organization or enterprises where only selected members are participants of a blockchain network. The level of security, authorizations, permissions, accessibility is in the hands of the controlling organization. Thus, private blockchains are similar in use as a public blockchain but have a small and restrictive network. Private blockchain networks are deployed for voting, supply chain management, digital identity, asset ownership, etc.

Examples of private blockchains are; Multichain and Hyperledger projects (Fabric, Sawtooth), Corda, etc.

Consortium Blockchain

A consortium blockchain is a semi-decentralized type where more than one organization manages a blockchain network. This is contrary to what we saw in a private blockchain, which is managed by only a single organization. More than one organization can act as a node in this type of blockchain and exchange information or do mining. Consortium blockchains are typically used by banks, government organizations, etc.

Examples of consortium blockchain are; Energy Web Foundation, R3, etc.

Hybrid Blockchain

A hybrid blockchain is a combination of the private and public blockchain. It uses the features of both types of blockchains that is one can have a private permission-based system as well as a public permission-less system. With such a hybrid network, users can control who gets access to which data stored in the blockchain. Only a selected section of data or records from the blockchain can be allowed to go public keeping the rest as confidential in the private network. The hybrid system of blockchain is flexible so that users can easily join a private blockchain with multiple public blockchains. A transaction in a private network of a hybrid blockchain is usually verified within that network. But users can also release it in the public blockchain to get verified. The public blockchains increase the hashing and involve more nodes for verification. This enhances the security and transparency of the blockchain network.

Example of a hybrid blockchain is Dragonchain.

Private Blockchain

Private blockchains are a restricted network of authorized nodes. No one outside the private network can access information exchanged between two nodes. As impressive as private blockchains are, they have their own pros and cons.

Advantages Of Private Blockchain

- **Speed:** Private blockchains' transactions occur at greater speed as compared to public blockchains. That means the transactions per second (TPS) rate is higher in the case of private blockchains. This is because there is a limited number of nodes in a private network as opposed to a public network. This fastens the consensus or verification process of a transaction by all the nodes in a network. Also, the rate of adding new transactions in a block is fast. Private blockchains can facilitate the transactions at a rate of up to thousands or hundred thousand TPS at a time.

- **Scalability:** Private blockchains are pretty scalable. That is, you can choose the size of your private blockchain as per your needs. For instance, if there is an organization that needs a blockchain of only 20 nodes, they can easily deploy one. Then after expansion, if they need to add more nodes, they can easily do so. This makes private blockchains very scalable as it gives an organization the flexibility to increase or decrease the size of their network without much effort.

Disadvantages Of Private Blockchain

- **Needs Trust-building:** As far as a public blockchain is concerned, it is like an open book or as we call it, an open ledger. This ensures the security and legitimacy of every user. Whereas, in a private network, there are limited participants in a restricted network. Especially within an organization, where colleagues know each other. They need to build trust to transmit confidential information within a network.
- **Lower Security:** As a private blockchain network has lesser number of nodes or participants, it runs a higher risk of a security breach. If anyone of the nodes gains access to the central management system, it can gain access to all the nodes in the network. This makes it easier for a node to hack the entire private blockchain and misuse the information.
- **Centralization:** Private blockchains are restricted that is they need a central Identity and Access Management (IAM) system for functioning properly. This system has all the monitoring and administrative rights. It gives permissions to add a new node in the network or decide the level of access they get for the information stored in the blockchain. This whole system contradicts the idea of decentralization which is one of the pillars of blockchain technology.

Public Blockchain

After discussing the pros and cons of a private blockchain, let us turn our heads to the other side, that is, public blockchain. As opposed to a private blockchain, the public blockchain is an unrestricted open ledger system. It can have as many numbers of nodes as there can be from all over the world. The data recorded on a blockchain in a public network is equally accessible by any node.

Advantages Of Public Blockchain

- **Trustable:** Unlike in private blockchain, two nodes or participants do not need to worry about the authenticity of the other. In other words, they don’t need to personally know or trust the other nodes as the process of proof-of-work makes sure there can be no fraud in transactions. So, one can trust public blockchains blindly without feeling the needing to trust individual nodes.
- **Secure:** There can be as many participants or nodes in a public network which makes it a secure network. The larger the network, greater the distribution of records and harder it is for hackers to hack the entire network. In addition to this, every node will do verification of transactions and proof-of-work which makes every transaction and block legitimate. Due to these practices and thoughtful cryptogenic encrypting methods, a public blockchain is much safer than the private one.
- **Open and Transparent:** Public blockchain is open and the data is transparent to all the participant nodes. A copy of the blockchain records or digital ledger is available at every authorized node. This makes the entire blockchain system completely open and transparent. No one shows a fake transaction or hides an existing one as every node has an updated copy of the database at any given point of time.

Disadvantages Of Public Blockchain

- **Lower TPS:** The rate of transactions per second in a public blockchain is very low. This is because it is a huge network with a lot of nodes and for every node to verify a transaction and do proof-of-work is time-consuming. This is why public blockchains like Bitcoin can process only 7 transactions per second or Ethereum network has a rate of 15 TPS. On the other hand, a private network such as Visa has a rate of 24,000 TPS indicating a huge difference in speed of transaction processing and execution.
- **Scalability Issues:** Like we just saw in the point above, that public blockchain have a slow rate of processing and completing transactions. This causes issues in scalability as well. Because the more we try to increase the size of the network, the slower it will get. However, solutions like Bitcoin’s Lightning Network helps in overcoming this problem. It maintains a rate of the transaction as we increase the size of the network.
- **High Energy Consumption:** The process of proof-of-work is highly energy consuming as it needs specialized systems (hardware components) to run a special algorithm.

Private Or Public Blockchain, Which One Is Better

Well, before passing a final verdict, we have thoroughly studied two main types of blockchains i.e. private and public blockchains. Both of them have certain distinctions from one another. However, the main differences lie in terms of security, scalability, and transparency. On one hand, where a private network might not seem very trustworthy, you can completely rely on a public network for its intact consensus (proof-of-work) system.

So, in a nutshell, every instance or case of a successful blockchain use that we have seen till date is of a public blockchain. Public blockchain guarantees security as hacking the entire network is almost impossible. In addition to this, it offers data transparency as every node has equal access to the record stored in the blockchain. One of the very successful examples of a public blockchain is the Bitcoin system.

Applications Of Blockchain

Blockchain Applications in Business

- **Supply Chain Management**

Blockchain's immutable ledger makes it well suited to tasks such as real-time tracking of goods as they move and change hands throughout the supply chain. Using a blockchain opens up several options for companies transporting these goods. Entries on a blockchain can be used to queue up events with a supply chain allocating goods newly arrived at a port to different shipping containers, for example. Blockchain provides a new and dynamic means of organizing tracking data and putting it to use.

- **Healthcare**

Health data that's suitable for blockchain includes general information like age, gender, and potentially basic medical history data like immunization history or vital signs.

As specialized connected medical devices become more common and increasingly linked to a person's health record, blockchain can connect those devices with that record. Devices will be able to store the data generated on a healthcare blockchain and append it to personal medical records. A key issue currently facing connected medical devices is the soiling of the data they generate but blockchain could be the link that bridges those silos.

- **Real Estate**

The average homeowner sells his or her home every five to seven years, and the average person will move nearly 12 times during their lifetime. With such frequent movement, blockchain could certainly be of use in the real estate market. It would expedite home sales by quickly verifying finances, reduce fraud thanks to its encryption, and offer transparency throughout the entire selling and purchasing process.

- **Media**

Media companies have already started to adopt blockchain technology to eliminate fraud, reduce costs, and even protect Intellectual Property (IP) rights of content like music records. According to MarketWatch, the global market for blockchain in media and entertainment is estimated to reach $1.54 billion by 2024.

One platform that has taken the spotlight in leveraging blockchain for media, is Eluvio Inc Formally launched in 2019, Eluvio Content Fabric uses blockchain technology to enable content producers to manage and distribute premium video to consumers and business partners without content delivery networks.

And recently, the platform has been tapped by media giant, MGM Studios for "global streaming to web, mobile, and TV everywhere audiences of 'certain properties."

- **Energy**

Blockchain technology could be used to execute energy supply transactions, but also to further provide the basis for metering, billing, and clearing processes, according to PWC. Other potential applications include documenting ownership, asset management, origin guarantees, emission allowances, and renewable energy certificates.

- **Voting**

Regardless of your field, voting likely comes into play on some level. Boards, shareholders, and employees are just three classes that are asked to hold votes from time to time, some more frequently than others.

This is especially the case with shareholder votes means by which democratic tallies are conducted aren't always conducive to achieving an effective or complete ballot.

Blockchain Applications In Government

- **Record Management**

  National, state, and local governments are responsible for maintaining individuals' records such as birth and death dates, marital status, or property transfers. Yet managing this data can be difficult, and to this day some of these records only exist in paper form. And sometimes, citizens have to physically go to their local government offices to make changes, which is time consuming, unnecessary, and frustrating. Blockchain technology could simplify this recordkeeping and make the data far more secure.

- **Identity Management**

  Proponents of blockchain tech for identity management claim that with enough information on the blockchain, people would only need to provide the bare minimum (date of birth, for example) to prove their identities.

- **Taxes**

  Blockchain tech could make the cumbersome process of filing taxes, which is prone to human error, much more efficient with enough information stored on the blockchain.

- **Non-Profit Agencies**

  Blockchain could solve the anti-trust problems charities are increasingly facing through greater transparency; the technology has the ability to show donors that NPOs are in fact using their money as intended. Furthermore, blockchain tech could help those NPOs tribute those funds more efficiently, manage their resources better, and enhance their tracking capabilities.

- **Compliance/Regulatory Oversight**

  The majority of regulatory oversight stems from recordkeeping, but the consequences of not maintaining records is inarguably much worse. Thus, compliance is non-negotiable for companies. Blockchain can make record updates available to regulators and businesses in real time, in turn reducing time lags and allowing red flags and inconsistencies to be spotted sooner.

Blockchain Applications in Other Industries

- **Financial Management and Accounting**

  If the blockchain is truly as secure as it has shown itself to be in the last several years, then such impenetrable security would be tantalizing for customers concerned with financial fraud.

- **Record Management**

  As stated earlier, the encryption that is central to blockchain makes it quite useful for record management because it prevents duplicates, fraudulent entries, and the like.

- **Cybersecurity**

  The biggest advantage for blockchain in cybersecurity is that it removes the risk of a single point of failure. Blockchain tech also provides end-to-end encryption and privacy.

- **Big Data**

  The immutable nature of blockchain, and the fact that every computer on the network is continually verifying the information stored on it, makes blockchain an excellent tool for storing big data.

- **Data Storage**

  The same principles for big data apply to data storage, as well.

- **IoT**

  Blockchain is poised to transform practices in a number of IoT sectors, including:

  The supply chain: Tracking the location of goods as they are shipped, and ensuring that they stay within specified conditions.

  Asset tracking: Monitoring assets and machinery to record activity and output as an alternative to cloud solutions.

Despite these key areas where blockchain can be leveraged, the technology in the IoT is still dependent on startups. In fact – only 17% of respondents to Business Insider Intelligence's survey of IoT providers think that blockchain will become a universal standard in the IoT.

5.2 EXECUTABLE CODE

Main.py

```
from tkinter import messagebox

from tkinter import *

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

from tkinter.filedialog import askopenfilename

from Block import *

from Blockchain import *

from hashlib import sha256

import os



main = Tk()

main.title("Blockchain Based Certificate Validation")

main.geometry("1300x1200")

global filename



blockchain = Blockchain()

if os.path.exists('blockchain_contract.txt'):

    with open('blockchain_contract.txt', 'rb') as fileinput:
        blockchain = pickle.load(fileinput)
    fileinput.close()

def saveCertificate():
    global filename
    text.delete('1.0', END)
    filename = askopenfilename(initialdir = "certificate_templates")
    with open(filename,"rb") as f:
        bytes = f.read()
    f.close()
    roll_no = tf1.get()
    name = tf2.get()
    contact = tf3.get()
    if len(roll_no) > 0 and len(name) > 0 and len(contact) > 0:
        digital_signature = sha256(bytes).hexdigest();
        data = roll_no+"#"+name+"#"+contact+"#"+digital_signature
        blockchain.add_new_transaction(data)
        hash = blockchain.mine()
        b = blockchain.chain[len(blockchain.chain)-1]
        text.insert(END,"Blockchain Previous Hash : "+str(b.previous_hash)+"\nBlock No :
"+str(b.index)+"\nCurrent Hash : "+str(b.hash)+"\n")
```

```
        text.insert(END,"Certificate Digital Signature : "+str(digital_signature)+"\n\n")
        blockchain.save_object(blockchain,'blockchain_contract.txt')
    else:
        text.insert(END,"Please enter Roll No")


def verifyCertificate():
    text.delete('1.0', END)
    filename = askopenfilename(initialdir = "certificate_templates")
    with open(filename,"rb") as f:
        bytes = f.read()
    f.close()
    digital_signature = sha256(bytes).hexdigest();
    flag = True
    for i in range(len(blockchain.chain)):
        if i > 0:
            b = blockchain.chain[i]
            data = b.transactions[0]
            arr = data.split("#")
            if arr[3] == digital_signature:
                text.insert(END,"Uploaded Certificate Validation Successfull\n")
                text.insert(END,"Details extracted from Blockchain after Validation\n\n")
                text.insert(END,"Roll No : "+arr[0]+"\n")

                text.insert(END,"Student Name : "+arr[1]+"\n")

                text.insert(END,"Contact No   : "+arr[2]+"\n")

                text.insert(END,"Digital Sign : "+arr[3]+"\n")

                flag = False

                break

    if flag:

        text.insert(END,"Verification failed or certificate modified")



font = ('times', 15, 'bold')

title = Label(main, text='Blockchain Based Certificate Validation')

title.config(bg='bisque', fg='purple1')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)



font1 = ('times', 13, 'bold')



l1 = Label(main, text='Roll No :')

l1.config(font=font1)

l1.place(x=50,y=100)
```

```
tf1 = Entry(main,width=20)

tf1.config(font=font1)

tf1.place(x=180,y=100)



l2 = Label(main, text='Student Name :')

l2.config(font=font1)

l2.place(x=50,y=150)



tf2 = Entry(main,width=20)

tf2.config(font=font1)

tf2.place(x=180,y=150)



l3 = Label(main, text='Contact No :')

l3.config(font=font1)

l3.place(x=50,y=200)

tf3 = Entry(main,width=20)

tf3.config(font=font1)

tf3.place(x=180,y=200)



saveButton = Button(main, text="Save Certificate with Digital Signature",
command=saveCertificate)
```

```
saveButton.place(x=50,y=250)

saveButton.config(font=font1)



verifyButton = Button(main, text="Verify Certificate", command=verifyCertificate)

verifyButton.place(x=420,y=250)

verifyButton.config(font=font1)

font1 = ('times', 13, 'bold')

text=Text(main,height=15,width=120)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10,y=300)

text.config(font=font1)



main.config(bg='cornflower blue')

main.mainloop()
```

Blockchain.py

```
from hashlib import sha256

import json

import time

import pickle

from datetime import datetime

import random

import base64

from Block import *



class Blockchain:

    # difficulty of our PoW algorithm

    difficulty = 2 #using difficulty 2 computation



    def __init__(self):

        self.unconfirmed_transactions = []

        self.chain = []

        self.create_genesis_block()

        self.peer = []

        self.translist = []

    def create_genesis_block(self): #create genesis block
```

```
        genesis_block = Block(0, [], time.time(), "0")

        genesis_block.hash = genesis_block.compute_hash()

        self.chain.append(genesis_block)


    @property

    def last_block(self):

        return self.chain[-1]


    def add_block(self, block, proof): #adding data to block by computing new and previous
hashes

        previous_hash = self.last_block.hash


        if previous_hash != block.previous_hash:

            return False


        if not self.is_valid_proof(block, proof):

            return False


        block.hash = proof

        #print("main "+str(block.hash))

        self.chain.append(block)
        return True


    def is_valid_proof(self, block, block_hash): #proof of work
        return (block_hash.startswith('0' * Blockchain.difficulty) and block_hash ==
block.compute_hash())


    def proof_of_work(self, block): #proof of work
        block.nonce = 0


        computed_hash = block.compute_hash()
        while not computed_hash.startswith('0' * Blockchain.difficulty):
            block.nonce += 1
            computed_hash = block.compute_hash()


        return computed_hash


    def add_new_transaction(self, transaction):
        self.unconfirmed_transactions.append(transaction)


    def addPeer(self, peer_details):
        self.peer.append(peer_details)
```

```
def addTransaction(self,trans_details): #add transaction
    self.translist.append(trans_details)


def mine(self):#mine transaction
    if not self.unconfirmed_transactions:
        return False


    last_block = self.last_block


    new_block = Block(index=last_block.index + 1,
                      transactions=self.unconfirmed_transactions,
                      timestamp=time.time(),
                      previous_hash=last_block.hash)


    proof = self.proof_of_work(new_block)
    self.add_block(new_block, proof)
    self.unconfirmed_transactions = []
    return new_block.index
def save_object(self,obj, filename):
    with open(filename, 'wb') as output:
        pickle.dump(obj, output, pickle.HIGHEST_PROTOCOL)
```

Block.py

```
from hashlib import sha256

import json

import time

class Block:

    def __init__(self, index, transactions, timestamp, previous_hash):

        self.index = index

        self.transactions = transactions

        self.timestamp = timestamp

        self.previous_hash = previous_hash

        self.nonce = 0


    def compute_hash(self):

        block_string = json.dumps(self.__dict__, sort_keys=True)

        return sha256(block_string.encode()).hexdigest()
```

CHAPTER 6

TESTING

6.1 IMPORTANCE

Testing is a process, which reveals error in the program. It is the major quality measure employee during software development during software development. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of software as a system element and attendant costs associated with a software failure are motivating factors for we planned, through testing. Testing is the process of executing a program with the intent of finding an error.

6.2 TYPES OF TESTING

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at different phases of software development are:

Unit testing

Unit testing is done on individual modules as a computer and become executable. It is confined only to the designer's requirements. Each module can be tested using the following strategies.

Black box testing

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This Testing has been uses to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structure or external database access
- Performance errors
- Initialization and termination errors

In this testing the output is checked for correctness. The logical flow of the data is not checked.

White box testing

In test cases are generated on the logic of each module by drawing flow graphs of that module and logical decision are tested on all the cases. It has been uses to generates the test cases in the following cases:

- Guarantee that all independent paths have been executed.
- Execute all logical decisions on their true and false slides.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structure to ensure their validity.

Integrating Testing

Integration testing ensures that software and subsystems work together a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

System Testing

Involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user system meets all requirements of the client's specifications.

Acceptance testing

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

Testing can be done in two ways:

- Bottom up approach
- Top down approach

Bottom up Approach

Testing can be performed starting form smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded within the larger system. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

Begins construction and testing with atomic modules. As modules are integrated from the bottom up, processing requirement for modules subordinate to a given level is always available and need for stubs is eliminated. The following steps implements this strategy.

- Low-level modules are combined in to clusters that perform a specific software sub function.
- A driver is written to coordinate test case input and output.
- Cluster is tested.
- Drivers are removed and moving upward in program structure combines clusters.

Integration moves upward, the need for separate test driver's lesions.
If the top levels of program structures are integrated top down, the number of drivers can be reduced substantially and integration of clusters is greatly simplified.

Top down approach

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interacting occurred. No attempt is made to verify the correctness of the lower level module.

Modules are integrated by moving downwards through the control hierarchy beginning with main program. The subordinate modules are incorporated into structure in either a breadth first manner or depth first manner. This process is done in five steps:

- Main control module is used as a test driver and steps are substituted or all modules directly to main program.
- Depending on the integration approach selected subordinate is replaced at a time with actual modules.
- Tests are conducted.
- On completion of each set of tests another stub is replaced with the real module
- Regression testing may be conducted to ensure trha5t new errors have not been introduced.

This process continuous from step 2 until entire program structure is reached. In top down integration strategy decision making occurs at upper levels in the hierarchy and is encountered first. If major control problems do exists early recognitions is essential. If depth first integration is selected a complete function of the software may be implemented and demonstrated.

Some problems occur when processing at low levels in hierarchy is required to adequately test upper level steps to replace low-level modules at the beginning of the top down testing. So no data flows upward in the program structure.

Validation

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error message are displayed.

6.3 TEST CASES

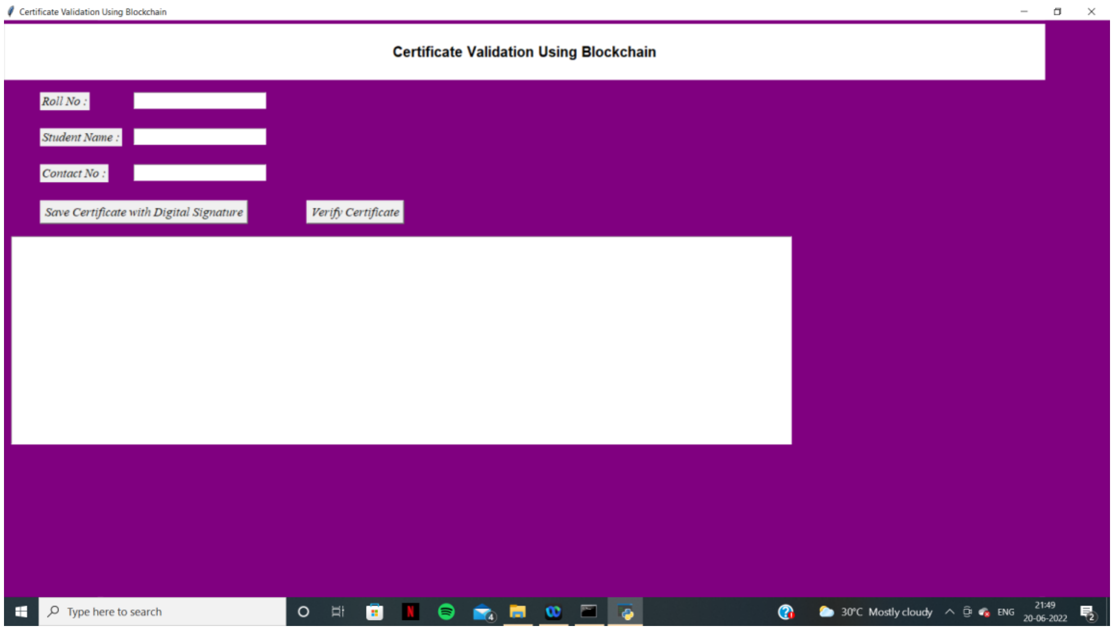
S.No	**Test Case**	**Result**
1.	Uploading	Success
2.	Validating	Success
3.	Verifying	Success
4.	Verifying Certificate when it is not present in the blockchain	Failed

Table.6.3 Test Cases with their respective results

CHAPTER 7

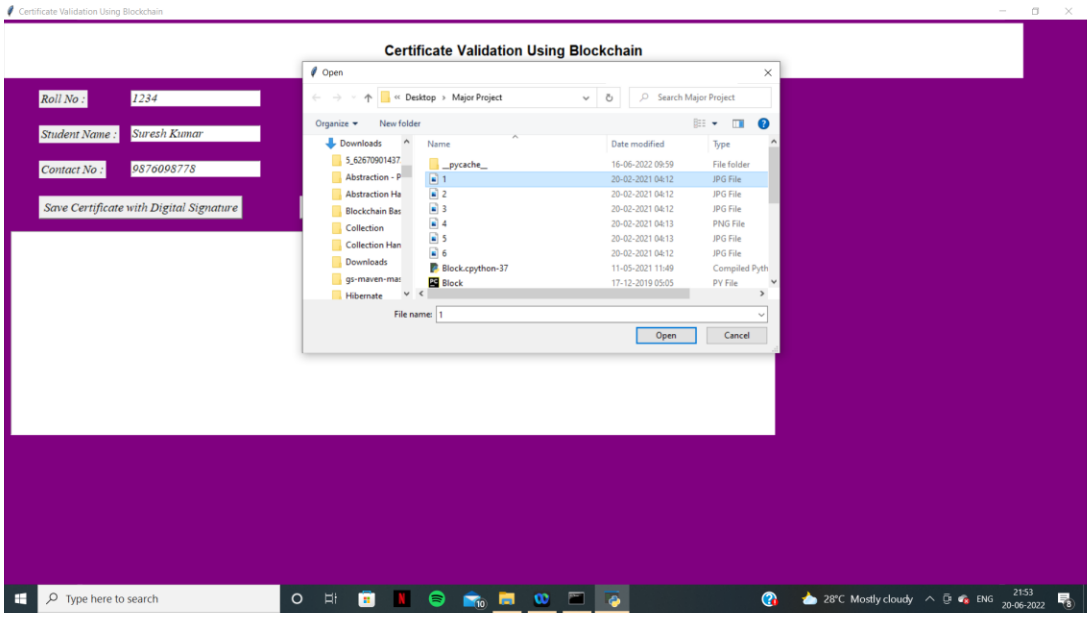
RESULTS

To run code double click on ‘run.bat’ file to get below screen


Certificate Validation Using Blockchain
Certificate Validation Using Blockchain
Roll No :
Student Name :
Contact No :
Save Certificate with Digital Signature
Verify Certificate
Type here to search
30°C Mostly cloudy
ENG
21:49
20-06-2022

7.1.1: Screen after running the program.

In above screen enter student details and then click on ‘Save Certificate with Digital Signature’ button to convert certificate into digital signature and then saved in Blockchain.



Roll No : 1234

Student Name : Suresh Kumar

Contact No : 9876098778

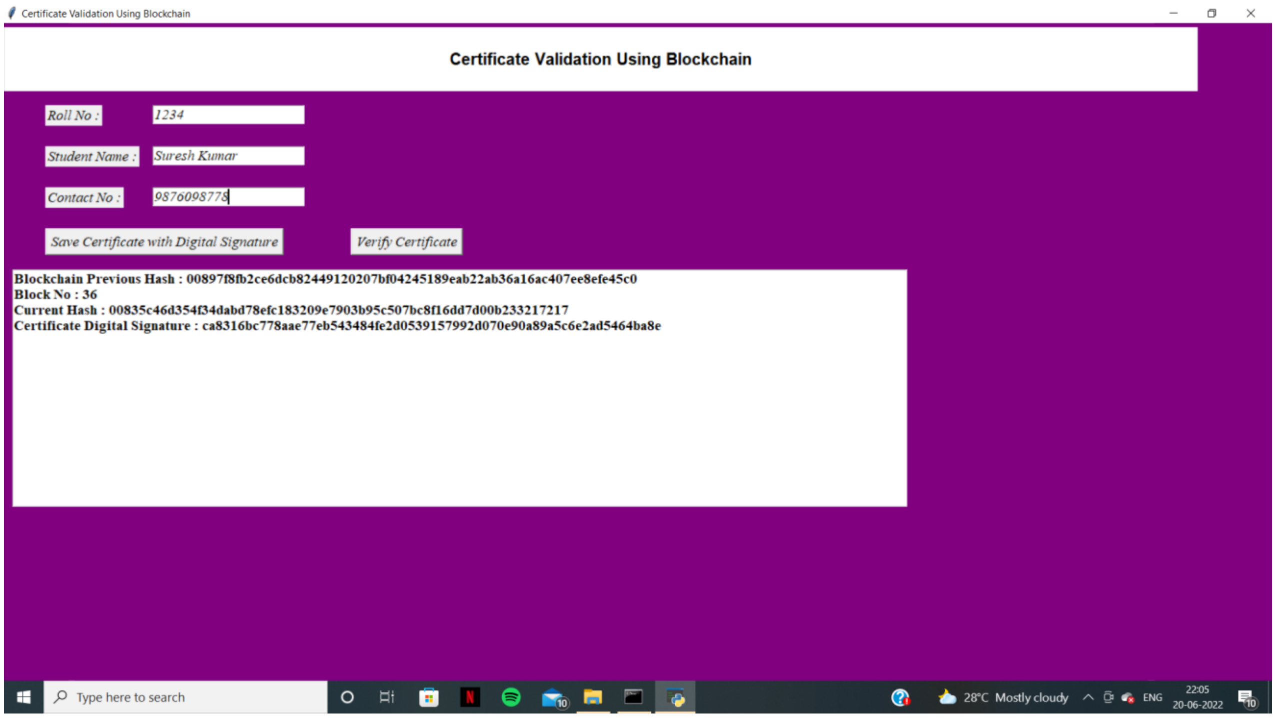
Name	Date modified	Type
__pycache__	16-06-2022 09:59	File folder
1	20-02-2021 04:12	JPG File
2	20-02-2021 04:12	JPG File
3	20-02-2021 04:12	JPG File
4	20-02-2021 04:13	PNG File
5	20-02-2021 04:13	JPG File
6	20-02-2021 04:12	JPG File
Block.cpython-37	11-05-2021 11:49	Compiled Pyth
Block	17-12-2019 05:05	PY File

Downloads
5_62670901437
Abstraction - P
Abstraction Ha
Blockchain Bas
Collection
Collection Han
Downloads
gs-maven-mas
Hibernate

File name: 1

7.1.2: Screen to enter the student details in order to save the certificate.

In above screen entered some student details and then click on ‘Save Certificate with Digital Signature’ button and then selecting and uploading ‘1.jpg’ file and then click on ‘Open’ button to get below screen.


Certificate Validation Using Blockchain
Certificate Validation Using Blockchain
Save Certificate with Digital Signature
Verify Certificate
Type here to search
28°C Mostly cloudy
ENG
22:05
20-06-2022

Roll No : 1234

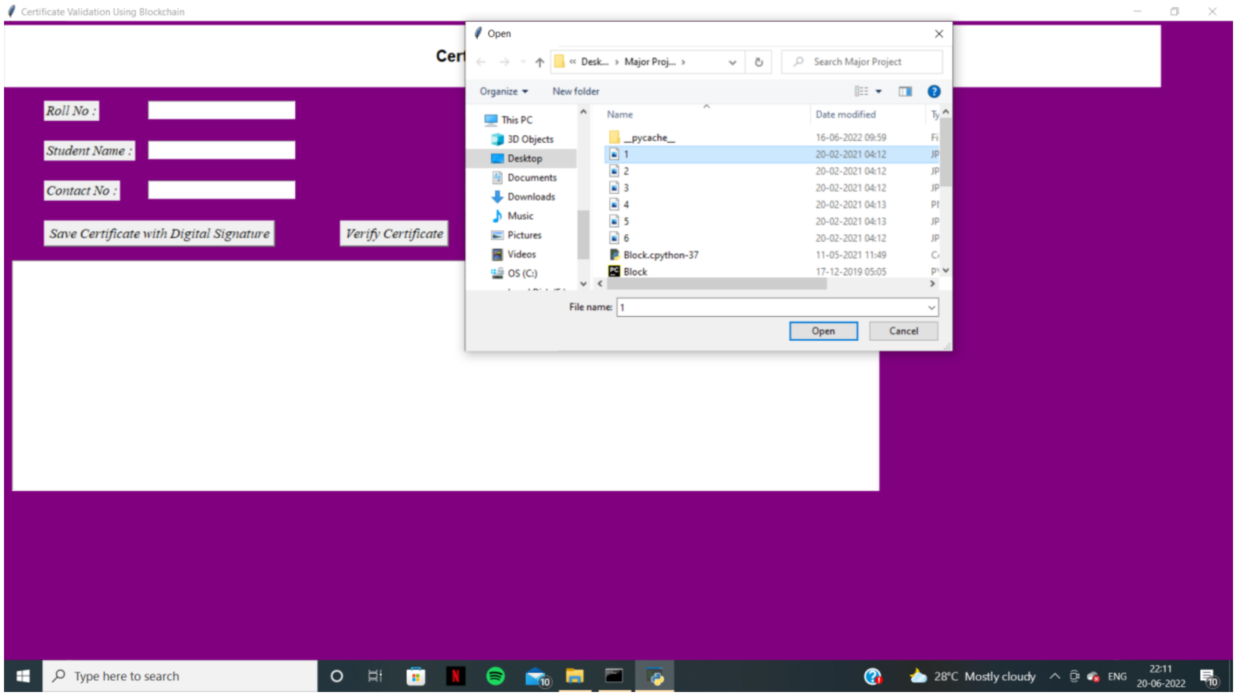
Student Name : Suresh Kumar

Contact No : 9876098778

```
Blockchain Previous Hash : 00897f8fb2ce6dcb82449120207bf04245189eab22ab36a16ac407ee8efe45c0
Block No : 36
Current Hash : 00835c46d354f34dabd78efc183209e7903b95c507bc8f16dd7d00b233217217
Certificate Digital Signature : ca8316bc778aae77eb543484fe2d0539157992d070e90a89a5c6e2ad5464ba8e
```

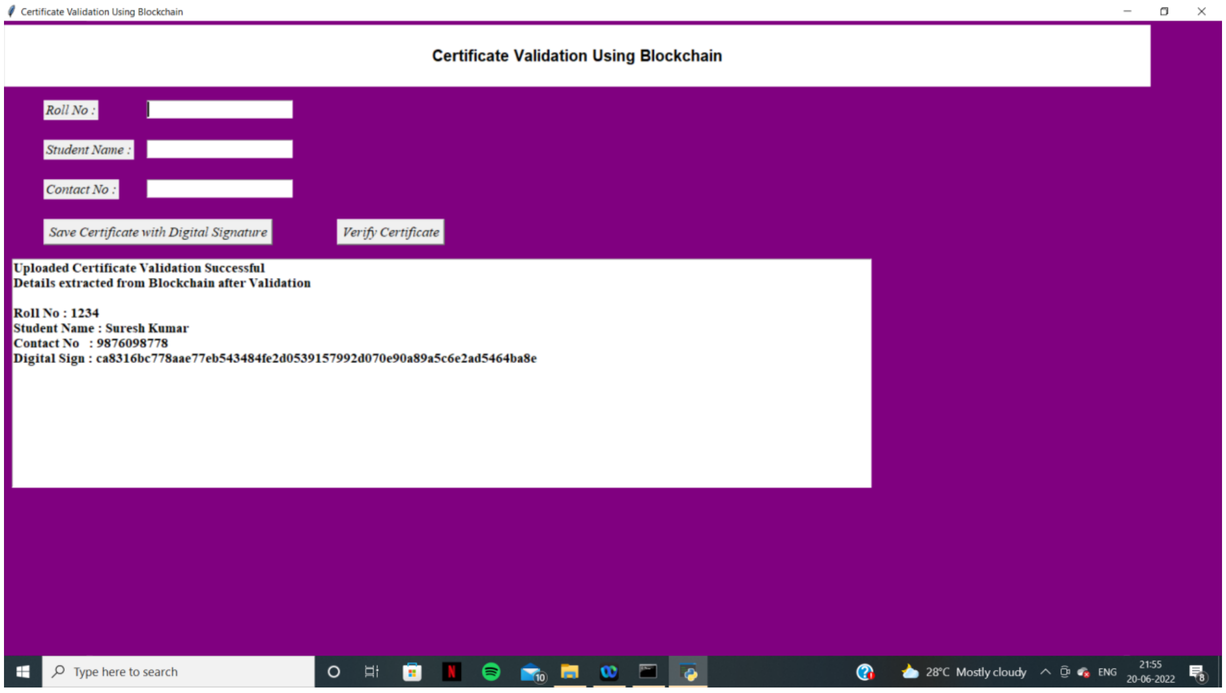
7.1.3: Screen when a block is created.

In above screen we can see Blockchain generated previous hash with block no 36 and its current hash and then keep on generating new blocks with each certificate upload and while running you can see that previous hash of new record will get matched with current hash of old record and this matched hash code proof that Blockchain verify old and new hash code before storing new block to confirm data is not altered. So above details stored at Blockchain and now verifier can click on ‘Verify Certificate’ button and upload same or other images to get below result.



7.1.4: Screen for selecting and uploading.

In above screen selecting and uploading ‘1.jpg’ file and then click on ‘Open’ button to get below result.

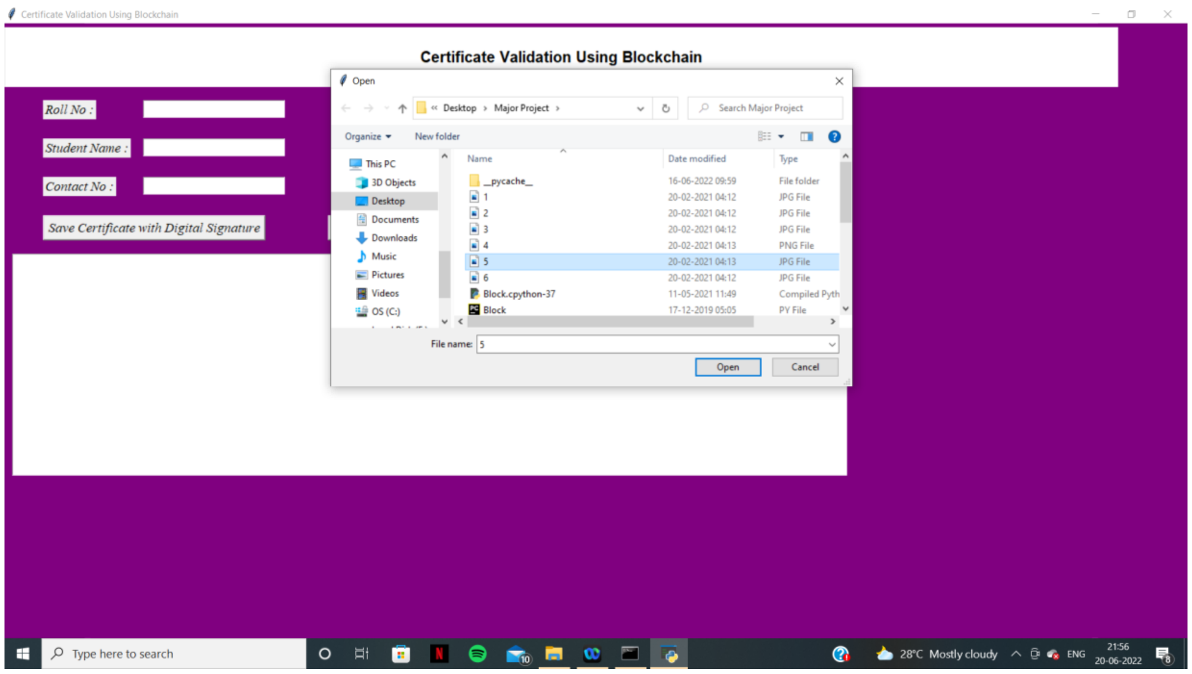

Certificate Validation Using Blockchain
Roll No :
Student Name :
Contact No :
Save Certificate with Digital Signature
Verify Certificate

```
Uploaded Certificate Validation Successful
Details extracted from Blockchain after Validation

Roll No : 1234
Student Name : Suresh Kumar
Contact No   : 9876098778
Digital Sign : ca8316bc778aae77eb543484fe2d0539157992d070e90a89a5c6e2ad5464ba8e
```

7.1.5: Screen after successful execution.

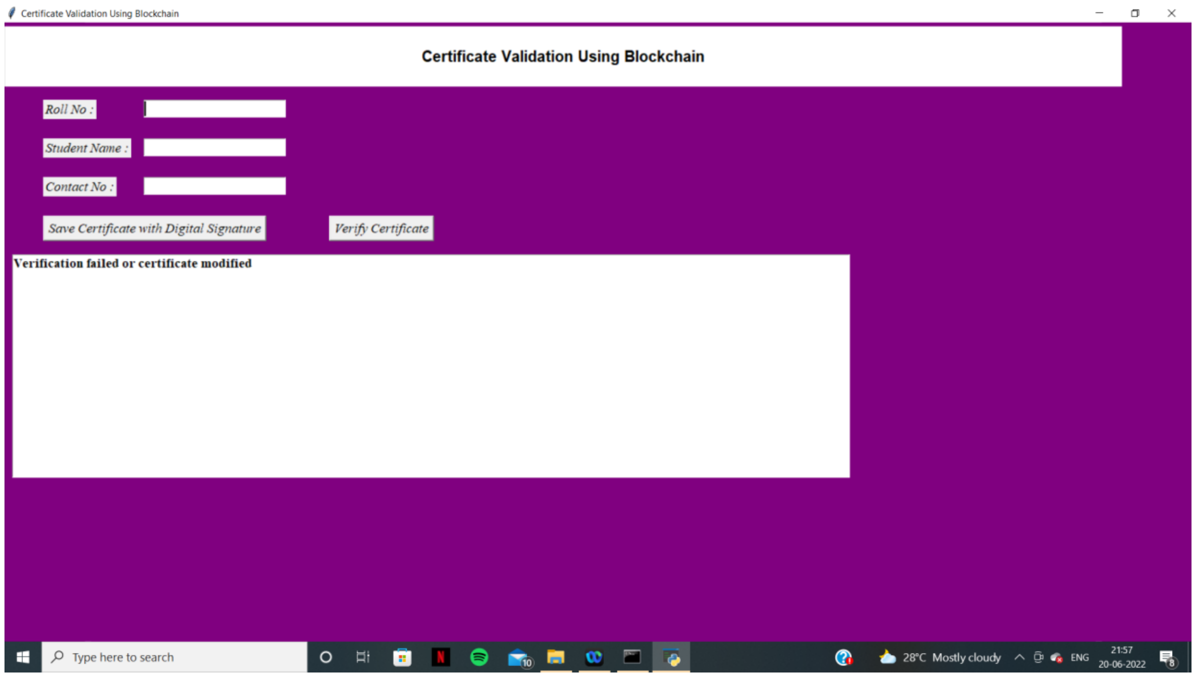
In above screen we uploaded same and correct image so application matched digital signature and then retrieve details from Blockchain and now try with some other image.



Name	Date modified	Type
__pycache__	16-06-2022 09:59	File folder
1	20-02-2021 04:12	JPG File
2	20-02-2021 04:12	JPG File
3	20-02-2021 04:12	JPG File
4	20-02-2021 04:13	PNG File
5	20-02-2021 04:13	JPG File
6	20-02-2021 04:12	JPG File
Block.cpython-37	11-05-2021 11:49	Compiled Pyth
Block	17-12-2019 05:05	PY File

7.1.6: Screen for selecting and uploading.

In above screen selecting and uploading '5.jpg' file and then click on 'Open' button to get below result.


Certificate Validation Using Blockchain
Roll No :
Student Name :
Contact No :
Save Certificate with Digital Signature
Verify Certificate

```
Verification failed or certificate modified
```

7.1.7: Screen for verification failed.

In above screen verification got failed as uploaded certificate not matched with stored certificates in Blockchain.

Similarly you can upload any other certificate and convert them to digital signature

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

Conclusion

In this project, blockchain based certificate validation has been introduced to validate the certificates which became a common challenge for many organizations such as colleges, companies, etc...

The proposed blockchain based certificate validation system takes the details of the user including the certificate and when a certificate is uploaded to our system to get verified, our system validates that certificate based on the details uploaded earlier.

A hash code is generated when a certificate is uploaded which is used to verify the certificate.

In June 2016, the MIT media lab released their blockchain-based credential system which is more secure, more reliable and harder to forge, in contrast to existing technologies that based on the third party arbitration. However, there are some serious authentication defects and vulnerable revocation mechanism which limits the prevalence and application of the project. In our project, to solve these problems and make its concept more practical, we proposed and designed a set of innovative cryptographic protocols which includes multi-signature, BTC address-state-based revocation mechanism and trusted federated identity

Among these protocols, the multi-signature scheme most notably increases the difficulty of forging owing to the fact that each issuing progress is obliged to be signed by the majority of the academic committee members. Besides, it enhances the safety of the private keys storing for the reasons that the private keys are possessed by separated devices and people. Besides, BTC address-based revocation mechanism improved the stability of the certificate revocation because BTC address is accessible and stable at any time. Moreover, this approach reduced the failure probability of revocation, because the cancellation process adheres the same the multi-signature algorithm, alike, involving several people. Trusted federated identity innovatively proved the authenticity of the certificate through the trusted path and federated identity. What’s more, the protocol of our project can be used in other related realms such as digital right protecting and contract proof. Case in point, our protocol enables the two companies to attach their contract onto the block chain with multi signature, which is different from the traditional third party-based work mode and dispel the worries of forging credentials.

Moreover, we implemented a blockchain-based certificate system, which embraced all the above protocols, by utilizing Java and JavaScript. This system has remedied the defect in Blockcerts to a certain extent, which makes the theory of blockchain-based certificate more practicable. Eventually, we conducted a series of security assessment from the perspective of operational safety, data security, network security and protocol security. The assessment outcomes provide compelling evidence that system is secured enough to meet the enterprise application standards.

Lastly, there are some limitations remained to be discussed, albeit, these considerations fall outside the scope of this paper: Our project is based on the Bitcoin blockchain, the maintenance of which relies on thousands of participants in the cryptocurrency ecosystem. Admittedly, it is imprudent to assume that the Bitcoin would work well continuously in the future because myriad types of stakeholders influence blockchain ecosystem or business model. In the years to come, we will adopt multiple blockchain sources such as Hyperledger and Ethereum to eliminate the factors of instability

Future scope

- The future scope of this project is to adapt multiple blockchain sources such as Hyperledger and Ethereum to eliminate the factors of instability.
- Ethereum blockchain environment is more reliable, transparent and deals with transactions.
- If there is a change to be made in a blockchain then a separate block is created our future scope is to change should be made in the block itself so that memory is not wasted.
- Our system accepts the certificate which is already verified and creates another block for the certificate which is uploaded again for this. Our future scope is to validate the certificate just for once and create only one block such that the number of blocks will not be increased.

CHAPTER 9

REFERENCES

[01] Tom M. Mitchell, "machine learning", Machinery Industry Press, Beijing, 2003.

[02] For Python and other algorithms: We collected data from websites like “geeks for geeks” , “w3schools” .

[03] Tengyu Yu, Blockchain operation principle analysis: 5 key technologies, iThome,

[04] Yong Shi, “Secure storage service of electronic ballot system based on block chain algorithm”, Department of Computer Science, Tsing Hua University, Taiwan, R.O.C., 2017.

[05] ZhenzhiQiu, “Digital certificate for a painting based on blockchain technology”, Department of Information and Finance Management, National Taipei University of Technology, Taiwan, R.O.C., 2017.

[06] JingyuanGao, The rise of virtual currencies! Bitcoin takes the lead, and the other 4 kinds can't be missed. Digital Age, https://www.bnext.com.tw/article/47456/bitcoinether-li tecoin-ripple-differences-between cryptocurrencies

[07] Smart contracts whitepaper, https://github.com/

[08] Gong Chen, Development and Application of Smart Contracts,

[09] Weiwei He, Exempted from cumbersome auditing and issuance procedures, several national junior diplomas will debut next year.iThome

[10] Xiuping Lin, “Semi-centralized Blockchain Smart Contracts: Centralized Verification and Smart Computing under Chains in the Ethereum Blockchain”, Department of Information Engineering, National Taiwan University, Taiwan, R.O.C., 2017.

[11] Weiwen Yang, Global blockchain development status and trends,

[12] Benyuan He, “An Empirical Study of Online Shopping Using Blockchain Technology”, Department of Distribution Management, Takming University of Science and Technology, Taiwan, R.O.C., 2017.

[13] Chris Dannen, Introducing Ethereum and Solidity,

[14] Jan Xie, Serpent GitHub,