

# **CERTIFICATE VALIDATION WITH SHA**

Submitted in partial fulfillment of the requirements for the award of  
the Degree of

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted by**

**R. G V D Nagesh**  
**20ME5A0514**

**N. Ganga Mahalakshmi**  
**19ME1A05D3**

**V. Swetha Yamini**  
**19ME1A05I2**

**M. Srinivas**  
**19ME1A05E1**



**Under the Guidance of**

**Mr. B. Prasad Babu**

**Assistant Professor**

**Department of Computer Science and Engineering**

**RAMACHANDRA COLLEGE OF ENGINEERING**

(Approved by AICTE, Affiliated to JNTUK, Kakinada)

Accredited by NBA, NAAC A+

NH-16 Bypass, Vatluru (V), Eluru -534007, W.G. Dist., A.P

**2019 – 2023**

# RAMACHANDRA COLLEGE OF ENGINEERING

(Approved by AICTE, Affiliated to JNTUK, Kakinada)

Accredited by NBA, NAAC A+

NH-16 Bypass, Vatluru (V), Eluru -534007, W.G. Dist., A.P

## DEPARTMENT OF COMPUTER SCIENCE& ENGINEERING



### CERTIFICATE

This is to certify that **R. Ganesh Venkata Durga Nagesh (20ME5A0514), N. ganga Mahalakshmi (19ME1A05D3), V. Swetha Yamini (19ME1A05I2), M. Srinivas(19ME1A05E1)** students of Bachelor of Technology in Computer Science & Engineering have successfully completed their project work entitled “**Certificate Validation With SHA**” at Ramachandra College of Engineering, Eluru during the Academic Year 2022-2023. This document is submitted in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science & Engineering and the same is not submitted elsewhere.

**Mr. B. Prasad Babu**  
Assistant Professor  
CSE Project Guide

**Dr. P M Prasuna**  
Professor & HOD,

**External Examiner**

## DECLARATION

We are, **R. Ganesh(20ME5A0514)**, **N. Ganga Mahalakshmi(19ME1A05D3)**, **V. Swetha Yamini (19ME1A05I2)**, **M. Srinivas(19ME1A05E1)** hereby declares the project report titled “Certificate Validation with Sha” under the supervision of **Mr. B. Prasad Babu**, Assistant Professor Department of Computer Science and Engineering is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering.

This is a record of work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other University or Institute for the award of any other degree or diploma.

## ACKNOWLEDGEMENT

We wish to take this opportunity to express our deep gratitude to all the people who have extended their cooperation in various ways during our project work. It is our pleasure and responsibility to acknowledge the help of all those individuals.

We sincerely thank our guide **Mr. B. Prasad Babu**, Assistant Professor in the Department of CSE for helping us in successful completion of our project under his supervision.

We are very grateful to **Dr. P M Prasuna**, Head of the Department, Department of Computer Science & Engineering for her assistance and encouragement in all respects in carrying throughout our project work.

We express our deepest gratitude to **Dr. V Srinivasa Rao**, Principal, Ramachandra College of Engineering, and Eluru for his valuable suggestions during preparation of draft in our document.

We sincerely thank all the faculty members and staff of the Department of CSE for their valuable advices, suggestions and constant encouragement which played a vital role in carrying out this project work.

We express our deepest gratitude to **The Management** of Ramachandra College of Engineering, Eluru for their support and encouragement in completing our project work and providing us necessary facilities.

Finally, we thank one and all who directly or indirectly helped us to complete our project work successfully.

**R. G V D Nagesh**  
**20ME5A0514**

**N. Ganga Mahalakshmi**  
**19ME1A05D3**

**V. Swetha Yamini**  
**19ME1A05I2**

**M. Srinivas**  
**19ME1A05E1**

<b>TITLE OF CONTENTS</b>		
<b>Chapter</b>	<b>NAME OF THE TITLE</b>	<b>Page No</b>
	<b>Abstract</b>	01
1	<b>Introduction</b>	02-04
2	<b>System Analysis</b>	
	2.1 Existing system	05
	2.2 proposed system	06
3	<b>Requirement Analysis</b>	07
	3.1 Preliminary investigation	08-09
4	<b>System study</b>	
	4.1 Feasibility Study	10
	• Operational Feasibility	10
	• Economy Feasibility	11
	• Technical feasibility	11
5	<b>System Requirements</b>	
	5.1 Hardware Requirements	12
	5.2 Software Requirements	12
6	<b>Modules</b>	13
7	<b>Design and diagrams</b>	
	7.1 Scenario based diagram	14
	7.2 Use case diagram	15
	7.3 Class diagram	16
	7.4 Sequence diagram	17-18
	7.5 Collaboration diagram	19
	7.6 Activity diagram	20-21
	7.7 State chart diagram	22
	7.8 Component diagram	23
	7.9 Deployment diagram	24
8	<b>Technologies</b>	25-32
9	<b>Coding</b>	33-38
10	<b>Screenshots</b>	39-45
11	<b>System testing</b>	
	11.1 Unit Testing	46
	a. Integration Testing	46
	b. Acceptance Testing	47-49
12	<b>Conclusion</b>	50
13	<b>References</b>	51

## **ABSTRACT**

Lakhs of people getting Degree's year after year, due to the lack of effective anti-forgery mechanism, events that cause the graduation certificate to be forged often get noticed. In order to solve the problem of counterfeiting certificates, the digital certificate system based on SHA and Digital signature technology. All the illegal activities filled against a person and all the activities are updated in the Personal ID. Using the modification process we would monitor the degree certificate. We deploy Unique based monitoring using this system. The main aim of this project is to secure academic certificate and for accurate management and to avoid forged certificate. To achieve all these features, we are converting all certificates into digital signatures and this digital signature will be stored in local server the digital signature is retrieved from the file at the time we need to verify. The same data is stored in different blocks to perform security feature. If by any chance if its data got altered then verification gets failed at next block storage and user may get intimation about data altered. Here in this project, we need provide two views for students and institutes but the views are separate we will provide first they need to login and they can validate the certificates the software technology we are used in this project to develop the frontend we use the python Tkinter library. Middleware python we used to develop and the details are stored in the local server file. The user first registers with mail id and he will direct to the login entering the password he can perform the internal operations. The original certificate has particular value when user needs validation his certificates, he needs to upload his certificate the digital signature will check whether certificate is original or not.

# CHAPTER 1

# 1. INTRODUCTION

Now a days the world is developing fast the internet use age is high as we compare to olden days. Our project is certificate validation with SHA in this the user can validate the certificate is original or not. He checks with the help of Digital signature with is produced to the certificate. Now days the companies are verify the certificate through online it may chance to get fraud the certificate. To avoid the certificate fraud, we need to develop this certificate validation with SHA. Here in this first the has value and the digital certificate was generated to the certificate with user details. The hash value and the digital signature was stored in the text file of local server.

At the time of validation of certificate, the system will take the hash value and the digital signature is taken from the text file which is already there. And for uploaded file the hash code is generate then it compares the both digital signatures.

The system says that whether the certificate is original or not. In this way it will check to develop this we used some algorithms they are SHA256(secure hashing algorithm) and Digital signature with help of this we perform the operation.

Certificate validation is a crucial part of digital security, and it is essential to ensure that the certificate presented is genuine and has not been tampered with. In this project, we will discuss certificate validation using SHA and digital signatures. We will explore the concepts of SHA and digital signatures and how they are used to ensure the authenticity of certificates.

SHA (Secure Hash Algorithm):

SHA is a cryptographic hash function that generates a fixed-length output from an input of variable length. The output, known as a hash value, is unique to the input, and even a slight change in the input results in a completely different hash value. The SHA algorithm was developed by the National Security Agency (NSA) and is widely used in digital security.

SHA-1 was the first version of the SHA algorithm, but it has been deprecated due to security concerns.

SHA-2 is the current version of the SHA algorithm and is considered more secure than SHA-1. SHA-2 has several variants, including SHA-256, SHA-384, and SHA-512.

SHA-0: A retronym applied to the original version of the 160-bit hash function published in 1993 under the name "SHA". It was withdrawn shortly after publication due to an undisclosed "significant flaw" and replaced by the slightly revised version SHA-1.

SHA-1: A 160-bit hash function which resembles the earlier MD5 algorithm. This was designed by the National Security Agency (NSA) to be part of the Digital Signature Algorithm.



Cryptographic weaknesses were discovered in SHA-1, and the standard was no longer approved for most cryptographic uses after 2010.

SHA-2: A family of two similar hash functions, with different block sizes, known as SHA-256 and SHA-512. They differ in the word size; SHA-256 uses 32-bit words where SHA-512 uses 64-bit words. There are also truncated versions of each standard, known as SHA-224, SHA-384, SHA-512/224 and SHA-512/256. These were also designed by the NSA.

SHA-3: A hash function formerly called Keccak, chosen in 2012 after a public competition among non-NSA designers. It supports the same hash lengths as SHA-2, and its internal structure differs significantly from the rest of the SHA family.

#### Digital Signature:

A digital signature is a mathematical technique used to verify the authenticity of digital messages or documents. It uses a public key infrastructure (PKI) to generate a unique digital signature, which is appended to the document. The digital signature can then be verified using the public key of the sender.

To create a digital signature, the sender first generates a hash of the document using a cryptographic hash function such as SHA-2. The sender then encrypts the hash using their private key, creating the digital signature. The recipient can then use the sender's public key to decrypt the signature and verify the document's authenticity.

The newest specification is: FIPS 186-4 from July 2013. DSA is patented but NIST has made this patent available worldwide royalty-free. A draft version of the specification FIPS 186-5 indicates DSA will no longer be approved for digital signature generation, but may be used to verify signatures generated prior to the implementation date of that standard.

#### Certificate Validation:

Certificate validation is the process of verifying the authenticity of a certificate presented by a website or device. The certificate contains information about the website or device, including the name, public key, and expiration date. Certificate validation is essential to ensure that the certificate is genuine and has not been tampered with. When a user visits a website with an SSL/TLS certificate, the browser checks the certificate's validity. The browser first checks whether the certificate is signed by a trusted certificate authority (CA). If the certificate is signed by a trusted CA, the browser then checks whether the certificate has expired or has been revoked. To check whether the certificate is signed by a trusted CA, the browser uses the CA's public key to verify the digital signature on the certificate. The digital signature is created using the CA's private key and the SHA-2 hash function. The browser then checks whether the certificate has expired or has been revoked by checking the certificate's validity period and the certificate

revocation list (CRL).

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 signature will be stored in local server as this tamper proof data storage and 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 SHA technology same transaction data stored at multiple servers with hash code verification and if data alter at one server, then it will be detected from other server as for same data hash code will get different.

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

In this project we have designed following modules

**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 local server in file format.

**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 local server and if matched found then it will retrieve all student details and display to verifier and if match not found then this certificate will be considered as fake or forge.

# CHAPTER 2

## 2. SYSTEM ANALYSIS

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of software development methodologies.

### 2.1 Existing System:

The existing system are two type one is physical checking of the certificate and another is online verification.

Physical checking verification the certificate is check by humans with there hands fir example the certificate is send to the college or any other organization they will check the certificate with hands whether the certificate is original or not.

The certificate is stored in centralized manner and verified manually, so it takes too much time to verify. There is no safety to the certificate that is given to any private sector.

Online verification: In this they validate the certificate with the help of any applications are they will check the certificate.

There are some applications in online they are.

#### **Document verification:**

In this application you can upload the certificate and verification.



#### **certificate validation: -**

This Application is used to verify the correctness of the Certificate produced by Candidate. This application is useful at the time of Interview in Government Department



## **DISADVANTAGES OF EXISTING SYSTEM:**

Time consuming

Error prone

Large volume of data

The user interface is not so good in some applications

Only some certificates can be verified.

## **2.2 PROPOSED SYSTEM:**

In our proposed system we need provide the admin view and user view separately because to improve the security and not get the certificate fraud.

The admin can create the digital signature for the certificate and this digital signature and hash value was stored in the local server file. The user can check or validate the certificate is original or not. For example, here one college or university will give the certificate to the student if they hardcopy they may can create another duplicate certificate or he they can forgery the certificate. If they give digital certificate to the student then there is less chance to get forgery. In this case the university should first create the digital signature to the certificate with the student details. And digital signature of the certificate should store in the file.

In case of any companies ask the certificate verification of the student, in this case the student will submit his digital certificate to that company.

To validate the certificates the college or university gives the digital signature to the company they can easily validate the user certificate through our system.

The companies may easily now weather the certificate are original or not. And the certificates are belonged to that student or not they can easily now.

The digital signature makes the work simple and easy.

## **ADVANTAGES OF PROPOSED SYSTEM**

- The user view is simple.
- The organization only create the digital signature to certificate.
- The user can easily validate the certificate.
- Reduce the time by avoiding manual validating the certificate.
- Eliminates manual intervention as far as possible
- Error free modification facilities

# CHAPTER 3

### 3. REQUIREMENTS ANALYSIS

Here in the requirements analysis, we analyzed what are the requirements need to develop the software and why we need to develop this software we must analysis it first.

There are some steps we need to analysis to identify the requirements.

To perform certificate validation with SHA, you need to identify the requirements and constraints of the system. Here are some key steps to perform requirement analysis for certificate validation with SHA:

**Define the system scope:** Identify the purpose and scope of the system. Determine the stakeholders and their requirements. You need to identify the types of certificates that you want to validate and the purposes for which they will be used.

**Determine the certificate sources:** Determine the sources of the certificates that you will validate. This includes the certificate authorities (CA) that issued the certificates, the end users who received the certificates, and the applications that use the certificates.

**Identify the validation requirements:** Determine the requirements for certificate validation. This includes the types of checks that need to be performed, such as verifying the certificate chain, checking the revocation status of the certificate, and validating the digital signature using SHA.

**Define the system architecture:** Define the architecture of the system for certificate validation with SHA. This includes the components that are required, such as the certificate validation engine, the CA trust store, and the revocation checking mechanism.

**Define the user interface:** Define the user interface for the certificate validation system. This includes the user interface for end-users, administrators, and developers. The user interface should be intuitive and easy to use, and it should provide clear feedback on the validation process.

**Define the system performance requirements:** Define the performance requirements for the certificate validation system. This includes the response time for certificate validation, the maximum number of certificates that can be validated simultaneously, and the system availability.

By following these steps, you can perform a comprehensive requirement analysis for certificate validation with SHA. This will help you to design and implement a robust and secure system for validating digital certificates.

#### 3.1 PRELIMINARY INVESTIGATION

##### Functional Requirements:

The functional requirement refers to the system needs in an exceedingly computer code engineering method.

The key goal of determinant “functional requirements” in an exceedingly product style and implementation is to capture the desired behavior of a software package in terms of practicality and also the technology implementation of the business processes

- The system shall be able to accept a digital certificate in X.509 format for validation.
- The system shall be able to extract the digital signature from the certificate.
- The system shall be able to validate the digital signature using SHA.
- The system shall be able to verify the certificate chain up to a trusted root CA(Certificate Authority).
- The system shall be able to generate a validation report indicating whether the certificate is valid or not.

#### **Non-Functional Requirements:**

All the other requirements which do not form a part of the above specification are categorized as Non-Functional needs. A system perhaps needed to gift the user with a show of the quantity of records during info. If the quantity must be updated in real time, the system architects should make sure that the system is capable of change the displayed record count at intervals associate tolerably short interval of the quantity of records dynamic

- The system shall be implemented in Python 3.x.
- The system shall use SHA-256 or higher as the hash algorithm for digital signature validation.
- The system shall have a modular architecture, with separate modules for certificate validation, certificate chain verification, and revocation checking.
- The system shall have a well-defined and documented API for easy integration with other applications.
- The system shall be able to handle large certificates without consuming excessive memory.
- The system shall be able to handle errors gracefully and provide meaningful error messages to the user.
- The system shall be secure, with proper protection of private keys and use of secure communication channels.

#### **Usability Requirements:**

The system shall have a GUI interface for end-users, with clear and concise instructions for certificate validation. The system shall provide clear feedback to the user on the validation results,



including the reasons for certificate rejection.

### **Performance Requirements:**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The system shall be able to validate a certificate within a reasonable time frame, based on the size and complexity of the certificate.

The system shall be able to handle a large number of concurrent certificate validations without significant performance degradation.

The system shall be scalable, with the ability to add additional resources to improve performance as needed.

### **Support and Maintenance Requirements:**

- The system shall be well-documented, with detailed user manuals and developer documentation.
- The system shall have a support team available for resolving issues and answering user questions.
- The system shall have a regular maintenance schedule for updates and bug fixes.
- The system shall have a backup and disaster recovery plan in place to ensure system availability in the event of a failure.

By considering these requirements, you can design and implement a certificate validation system that meets the needs of your users and stakeholders while ensuring security, scalability, and maintainability.

# CHAPTER 4

## 4. FEASIBILITY STUDY

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility
- Economic Feasibility

### 4.1 Technical Feasibility:

Determine whether the project can be technically implemented within the available resources, such as hardware, software, and personnel. Assess whether the project aligns with the organization's existing IT infrastructure and architecture. Consider the availability of APIs, libraries, and other technical resources required for implementing the project.

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment's have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?

### 4.2 Operation Feasibility:

#### User-friendly

The user or client can easily operate the system he can enter to the system by login and he can perform the operation

#### Reliability

The package will pick-up current transactions on line. Regarding the old transactions, User will enter them in to the system.

#### **Security**

The web server and database server should be protected from hacking, virus etc

#### **Portability**

The application will be developed using standard open source software (Except Oracle) like python Internet Explorer Browser etc. these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

**Availability**

This software will be available always.

**Maintainability**

The system uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end and the 2nd tier is the database, which uses local server , which is the back-end.

**4.3 Economic Feasibility:**

The computerized system takes care of the present existing system's data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports.

It should be built as a web-based application with separate web server and database server. This is required as the activities are spread throughout the organization customer wants a centralized database. Further some of the linked transactions take place in different locations.

**Schedule feasibility:**

Determine whether the project can be completed within the available time frame. Assess the project timeline, including the design, development, testing, and deployment phases. Consider the potential risks and delays that may impact the project timeline.

By conducting a feasibility study for the certificate validation with SHA project, you can assess whether the project is viable and practical within the available resources and budget. This will help you to make informed decisions about whether to proceed with the project and how to allocate resources effectively.

# CHAPTER 5

## **5. SYSTEM REQUIREMENTS**

### **Requirements Specification:**

Requirement Specification provides a high secure storage to the web server efficiently. Software requirements deal with software and hardware resources that need to be installed on a server which provides optimal functioning for the application. These software and hardware requirements need to be installed before the packages are installed. These are the most common set of requirements defined by any operation system. These software and hardware requirements provide a compatible support to the operation system in developing an application.

### **5.1 HARDWARE REQUIREMENTS:**

The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

System : Pentium IV 2.4 GHz.

Hard Disk : 100 GB.

Monitor : 15 VGA Color.

Mouse : Logitech.

RAM : 1 GB.

### **5.2 SOFTWARE REQUIREMENTS:**

The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

Operating system : Windows XP/7/10

Coding Language : Python 3.7

IDLE : PYCHARM

# CHAPTER 6

## 6. MODULES

### **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 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 database and if matched found then it will retrieve all student details and display to verifier and if match not found then this certificate will be consider as fake or forge.

### **Cryptography library:**

This library provides a wide range of cryptographic algorithms and protocols, including SHA-1, SHA-256, and SHA-512, which can be used for generating and validating digital certificates.

Certificate authority (CA) integration: To validate certificates, the system must be able to access and verify certificate information from trusted CAs. Integration with the CA's API or web service may be required to access this information.

### **User interface module:**

A user interface module can be used to provide a user-friendly interface for interacting with the certificate validation system. This can include features such as certificate lookup, certificate revocation checking, and certificate chain validation

### **Database module:**

A database module can be used for storing and retrieving certificate data, including certificate revocation lists (CRLs) and certificate status information.

### **Logging and monitoring module:**

A logging and monitoring module can be used to record and monitor certificate validation activity, including successful and failed validation attempts, and any errors or exceptions that occur during the validation process.

### **tkinter module**

The tkinter module is a standard Python library that provides a GUI (Graphical User Interface) toolkit for creating desktop applications with a graphical user interface. tkinter provides a set of tools and widgets that enable developers to create windows, dialog boxes, buttons, menus, and other user interface elements for their applications



# CHAPTER 7

## 7. DESIGN AND DIGRAMS

### 7.1 Scenario based diagram: -

A scenario-based diagram is a type of UML diagram that shows a sequence of interactions between objects or actors in a system. It typically describes a specific scenario or use case in a system. The diagram is used to model the behavior of the system, and it helps to identify the various components of the system and how they interact with each other.

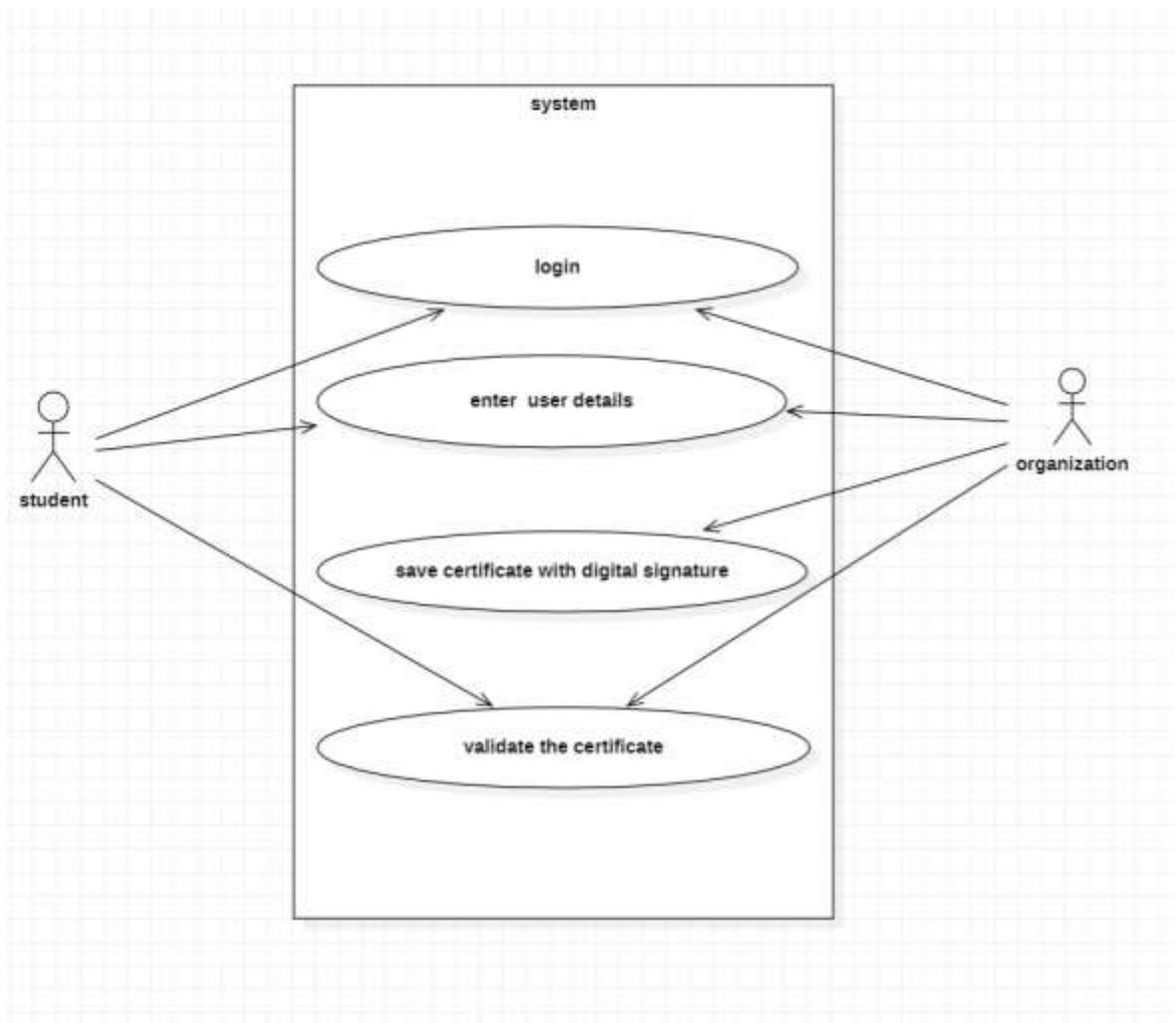


Fig: Scenario based diagram

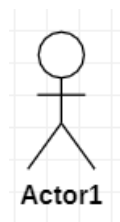
## 7.2 Use case diagram:

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application.

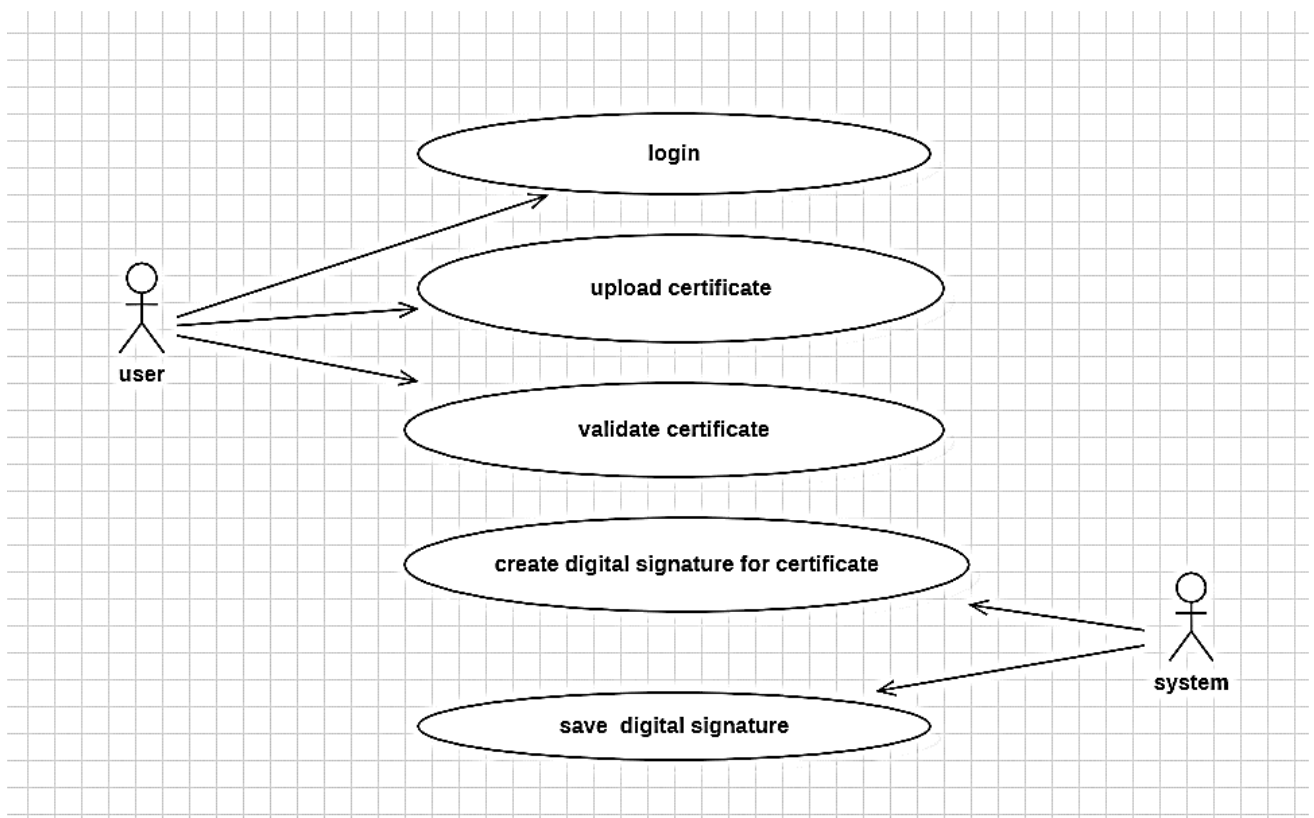
Use Case: A use case is an activity the system carries out, usually in response to a request by user.



**An actor:** represents a role that an outsider takes on when interacting with the business system. For instance, an actor can be a customer, a business partner, a supplier, or another business system. Every

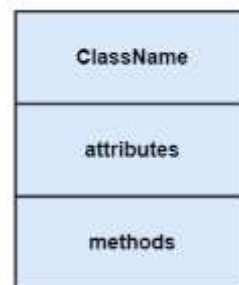


actor has a name.



### 7.3 Class diagram:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



#### organization class diagram:

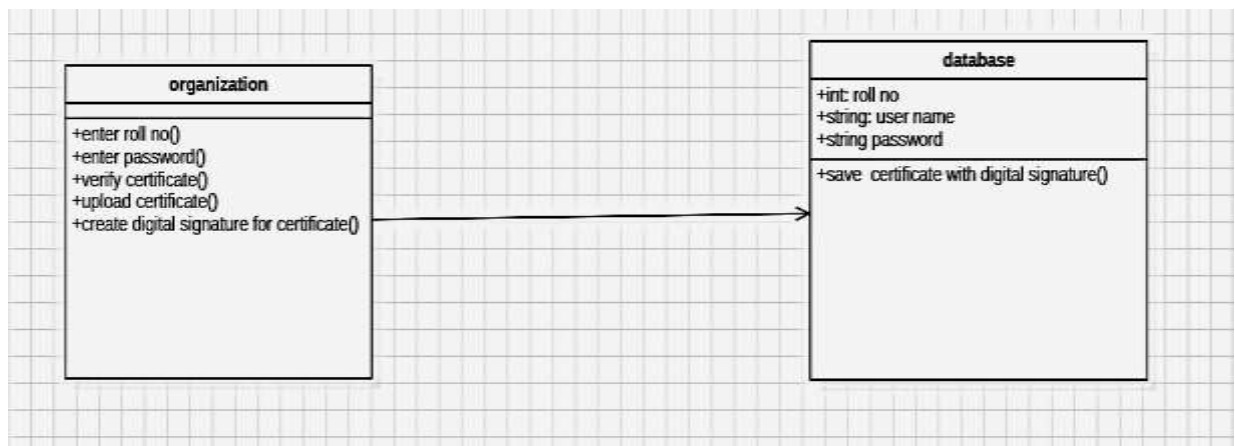


fig: - class diagram for organization

#### Student class diagram

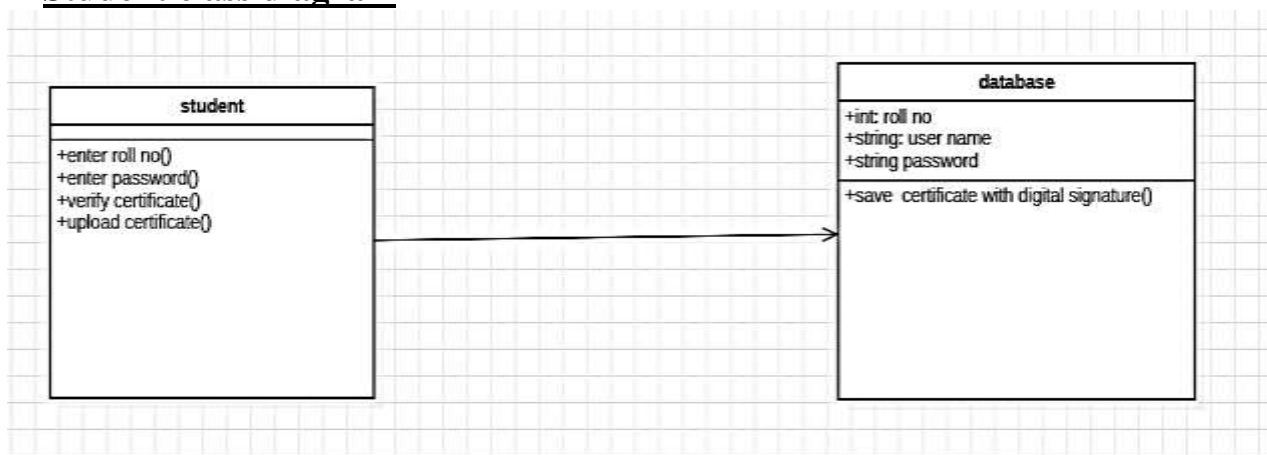


Fig: -class diagram for student

## **7.4Sequence diagram:**

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

### **Lifeline**

An individual participant in the sequence diagram is represented by a lifeline. It is positioned at the top of the diagram.

### **Activation**

It is represented by a thin rectangle on the lifeline. It describes that time period in which an operation is performed by an element, such that the top and the bottom of the rectangle is associated with the initiation and the completion time, each respectively.

### **Messages**

The messages depict the interaction between the objects and are represented by arrows. They are in the sequential order on the lifeline. The core of the sequence diagram is formed by messages and lifelines. Following are types of messages enlisted below:

### **organization sequence diagram:**

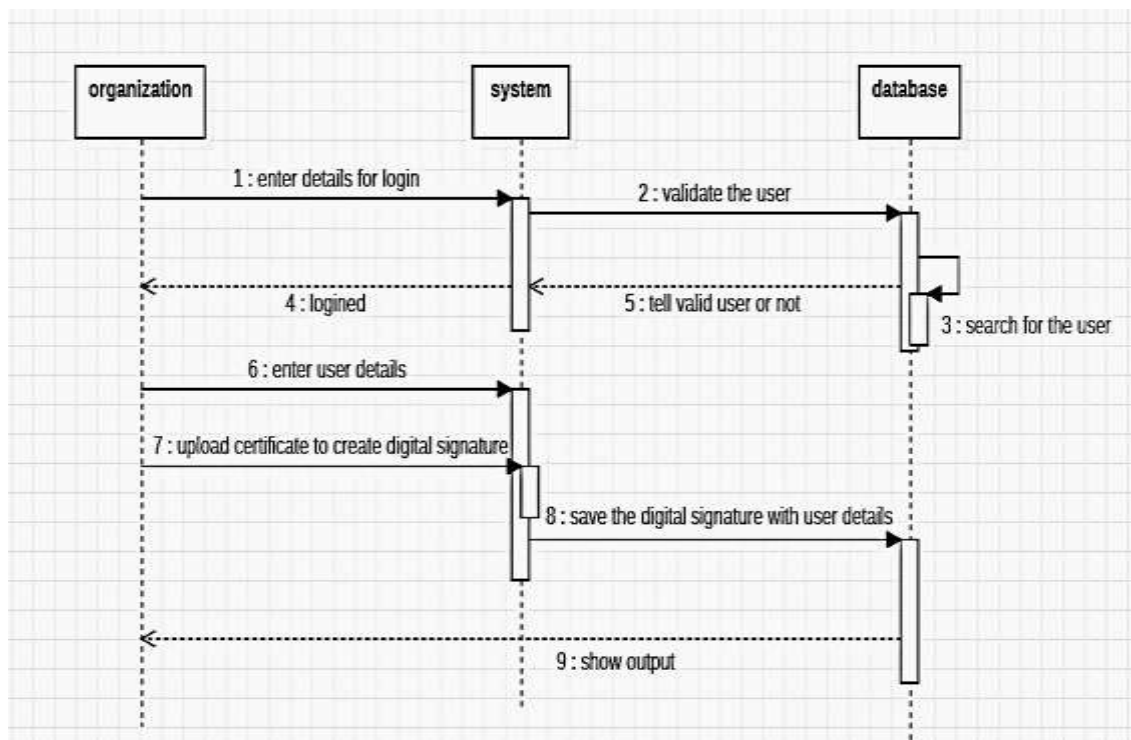
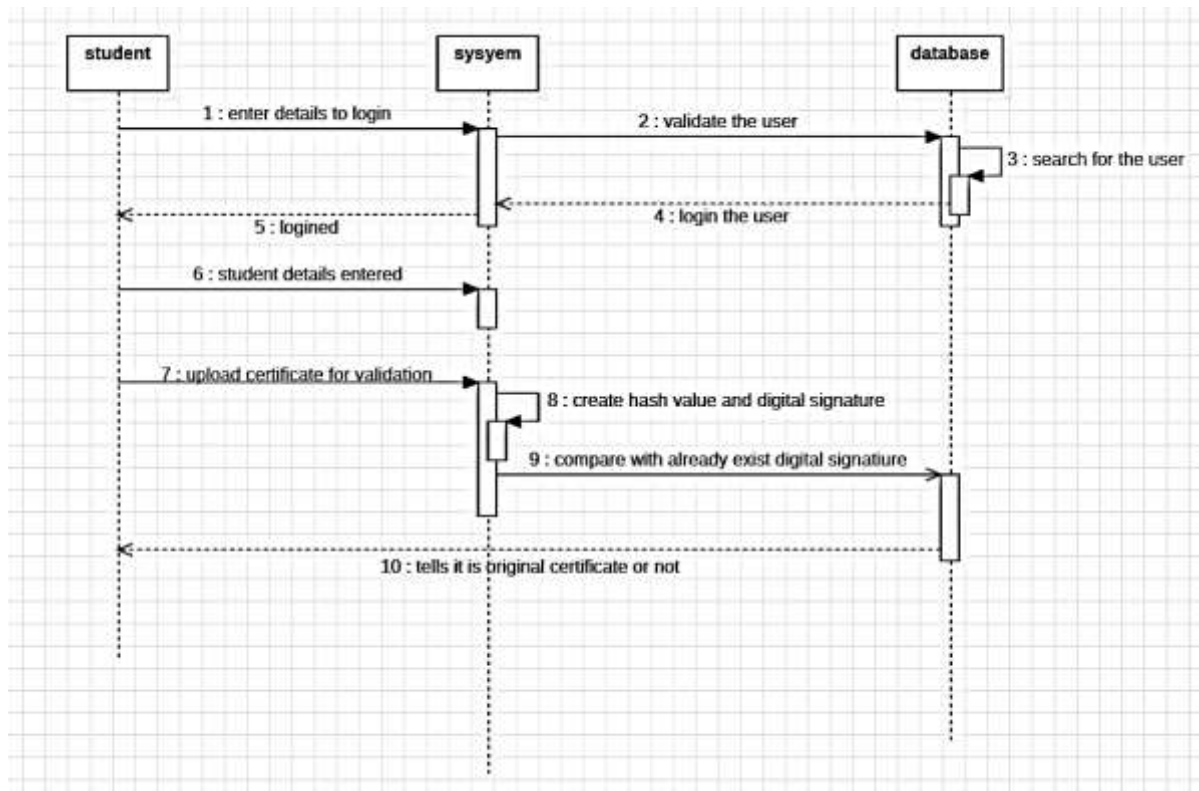


fig:- organization sequence diagram



student sequence diagram

## 7.5 Collaboration diagram

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system

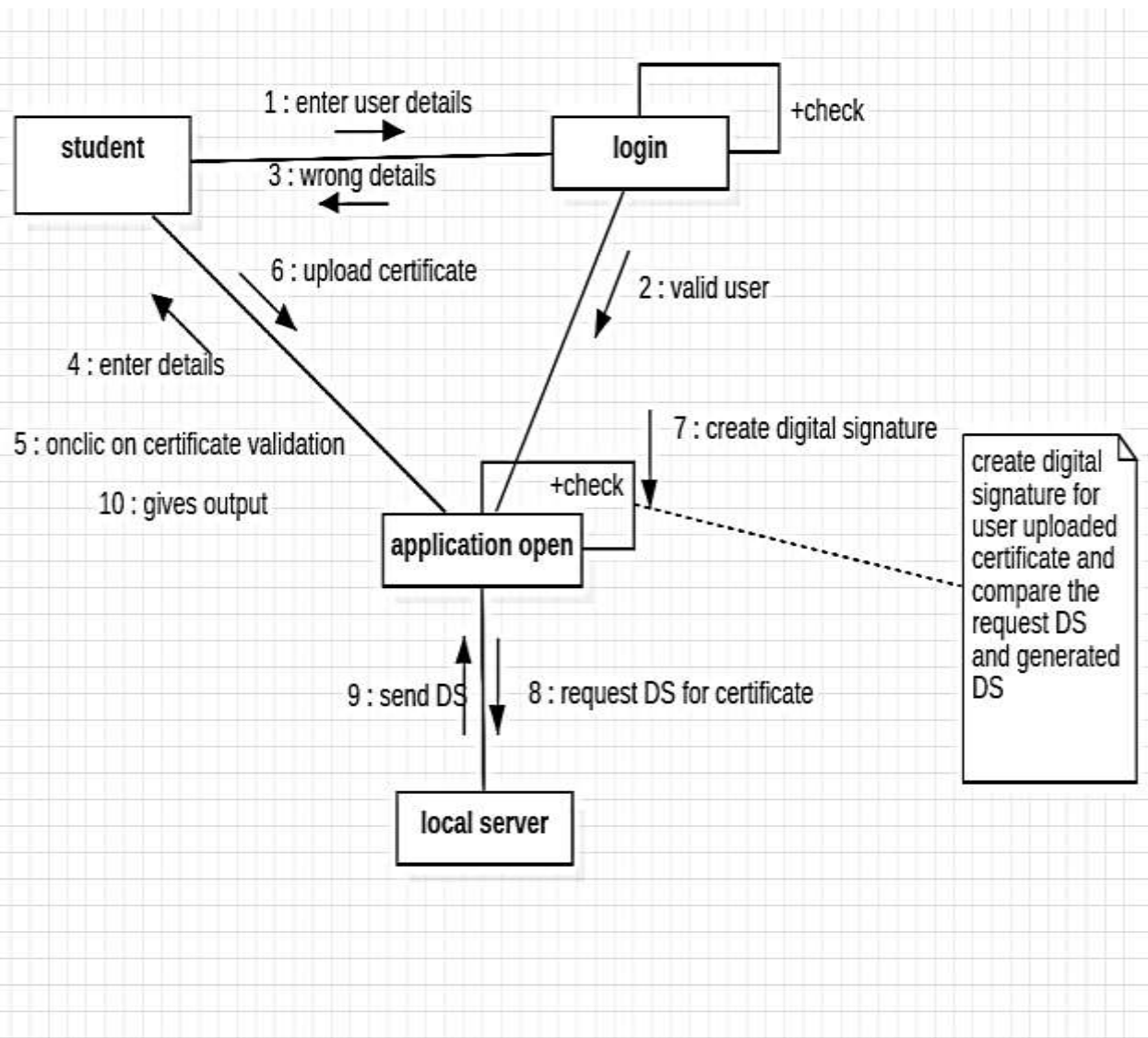


Fig: -collaboration diagram for certificate validation with sha

## 7.6 Activity diagram:

The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds of flows, the activity diagram has come up with a fork, join, etc.

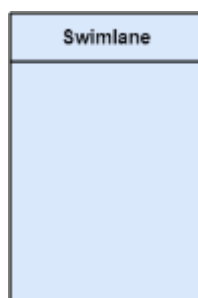
### Activities

The categorization of behavior into one or more actions is termed as an activity. In other words, it can be said that an activity is a network of nodes that are connected by edges. The edges depict the flow of execution. It may contain action nodes, control nodes, or object nodes.



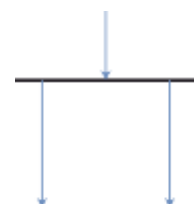
### Activity partition /swim lane

The swim lane is used to cluster all the related activities in one column or one row. It can be either vertical or horizontal. It used to add modularity to the activity diagram. It is not necessary to incorporate swim lane in the activity diagram. But it is used to add more transparency to the activity diagram.



### Forks

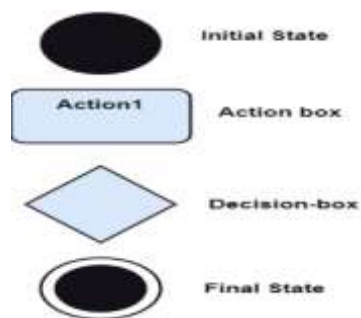
Forks and join nodes generate the concurrent flow inside the activity. A fork node consists of one inward edge and several outward edges. It is the same as that of various decision parameters. Whenever a data is received at an inward edge, it gets copied and split crossways various outward





edges. It split a single inward flow into multiple parallel flows.

Notation of an Activity diagram



## ACTIVITY DIAGRAM

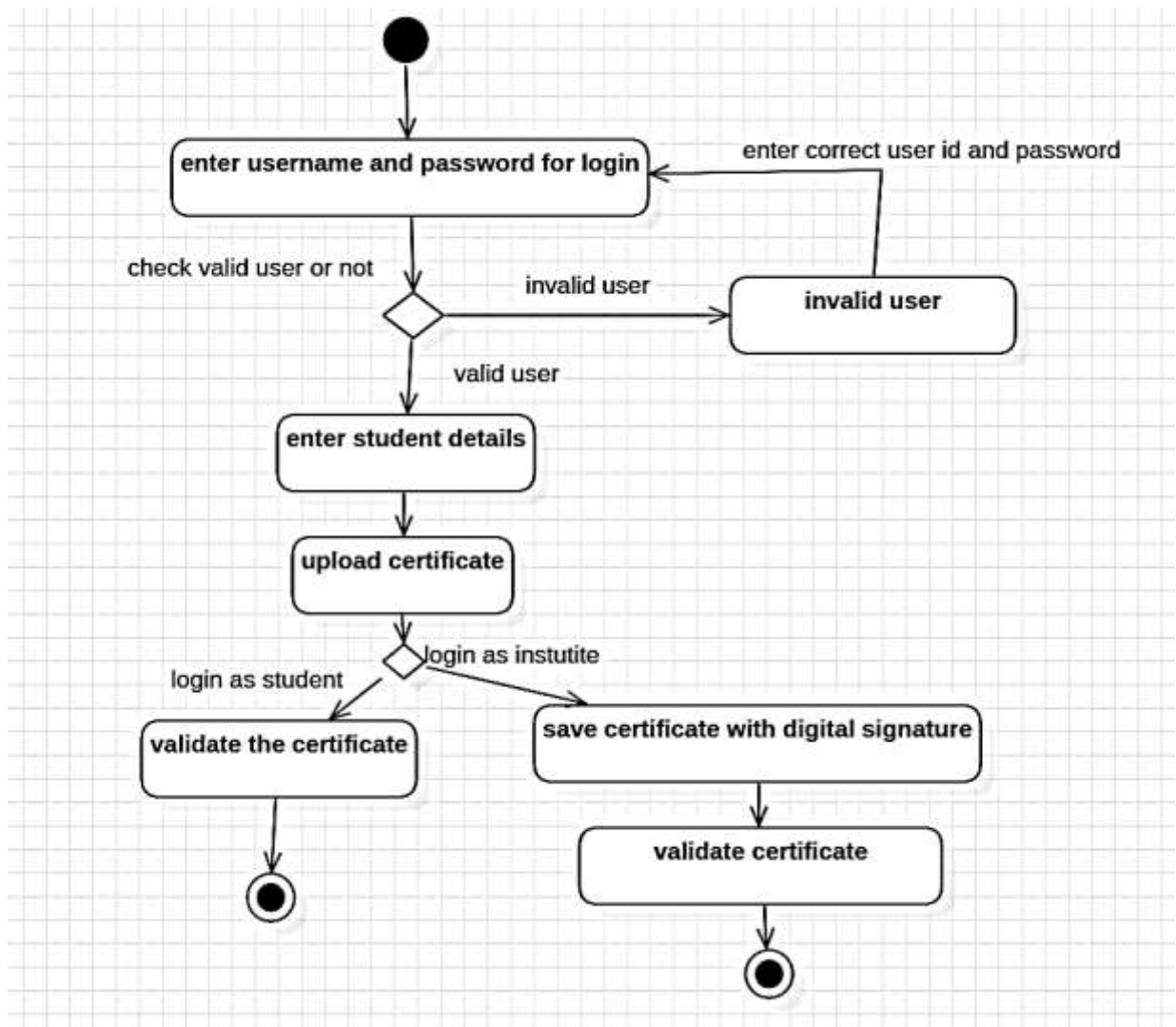


Fig:- activity diagram for certificate validation with sha

### 7.7 State chart diagram:

A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

Activity diagram explained in the next chapter, is a special kind of a State chart diagram. As State chart diagram defines the states, it is used to model the lifetime of an object.

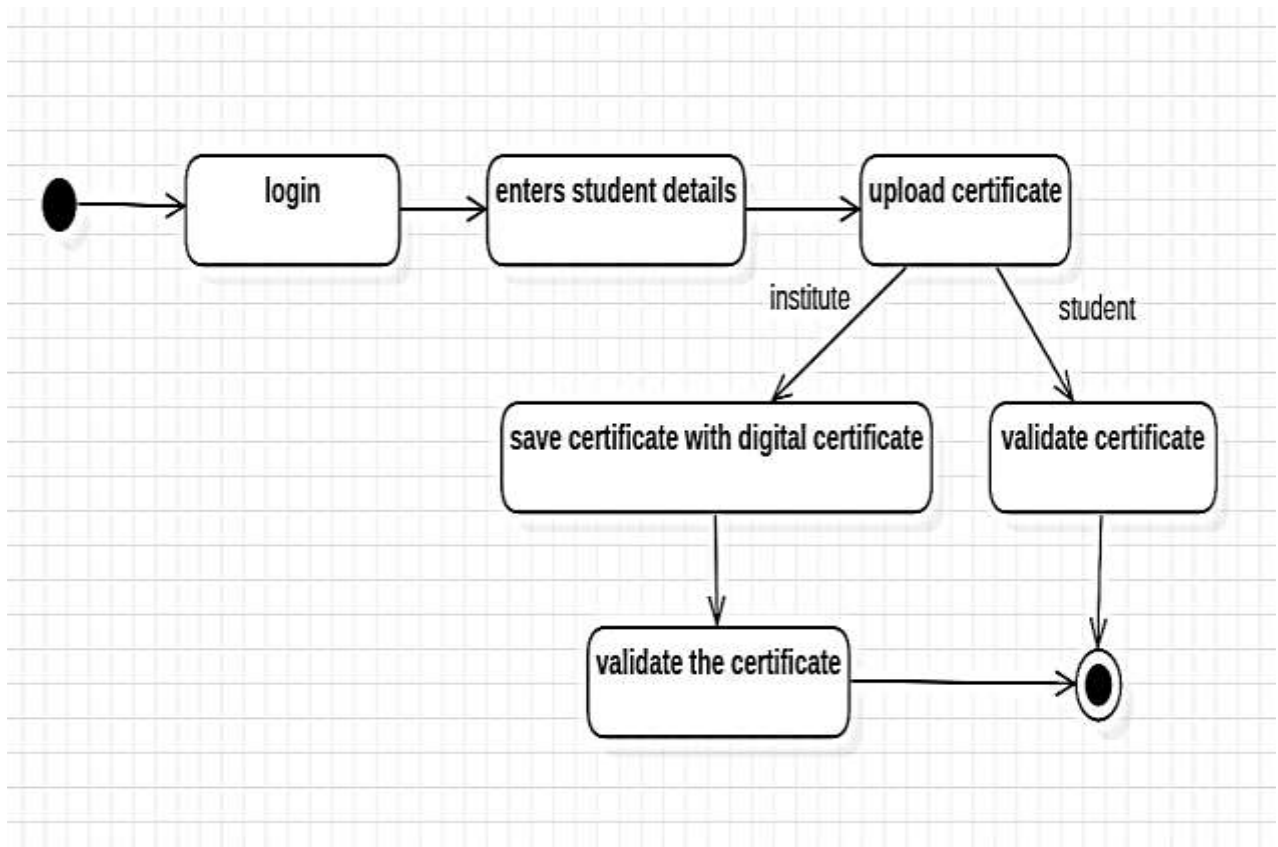
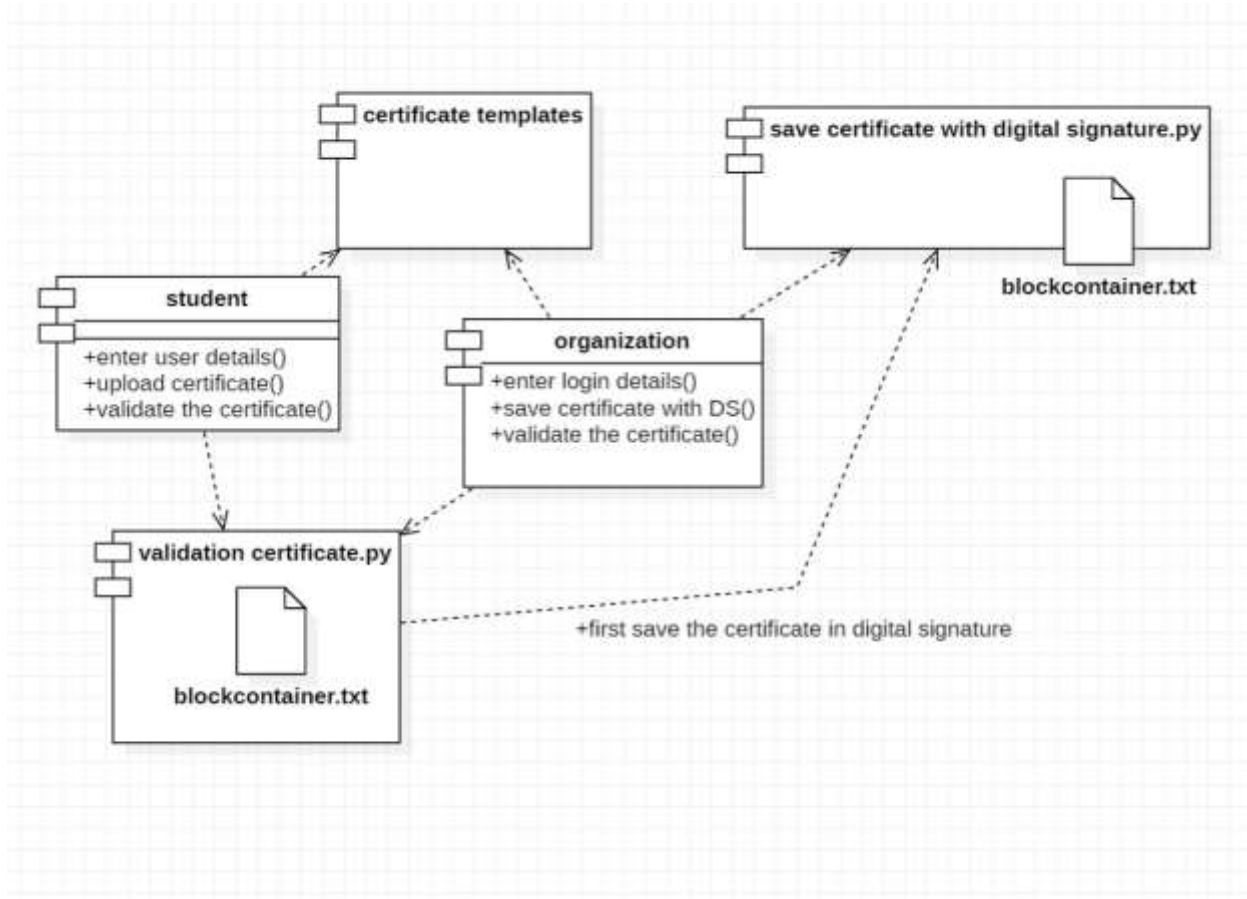


Fig:- state chart diagram for certificate validation with sha

## 7.8 Component diagram

component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node.

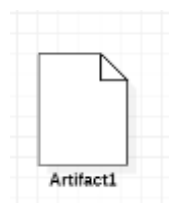


### Notation of a Component Diagram

#### Component

A component represents a modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces.

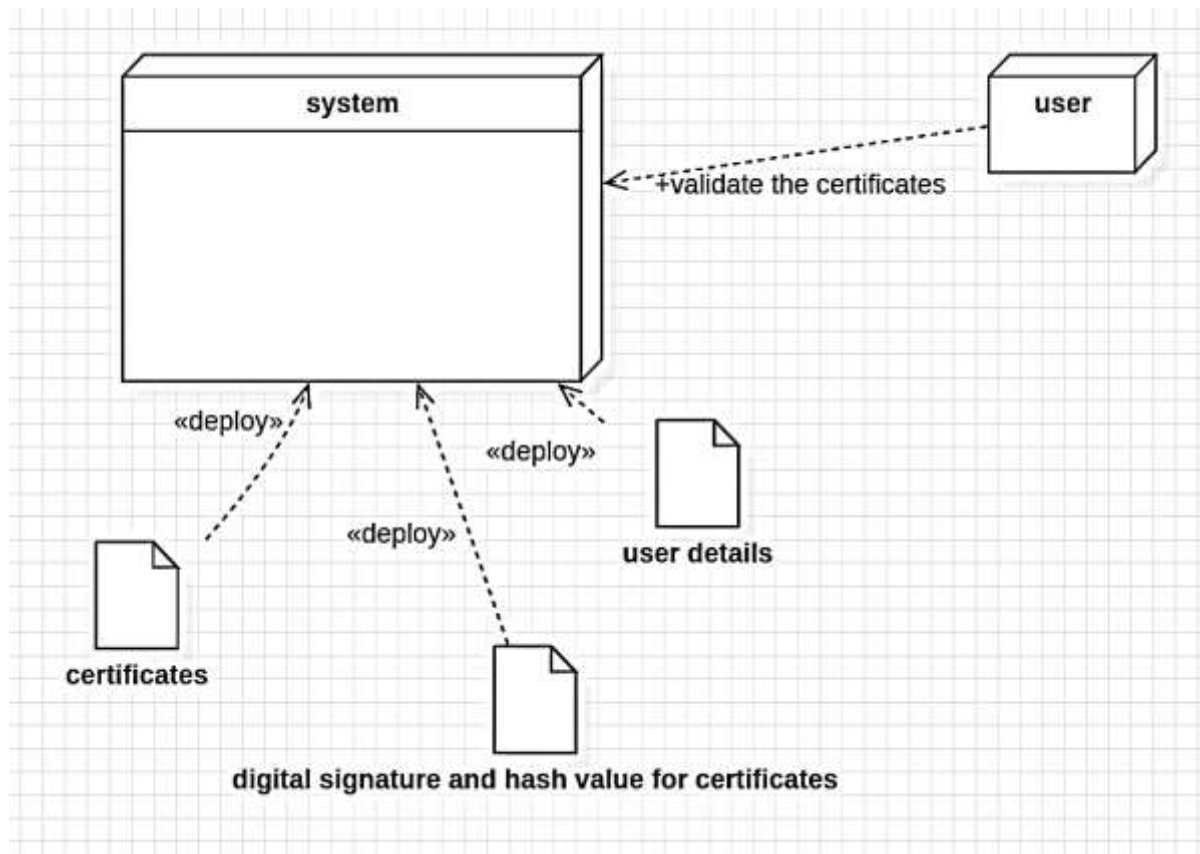
Artifact: An Artifact represents a physical piece of information that is used or produced by a software development process. Examples of Artifacts include models, source files, scripts, and



binary executable files. An Artifact may constitute the implementation of a deployable component.

## 7.9 Deployment diagram:

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships



### Symbol and notation of Deployment diagram

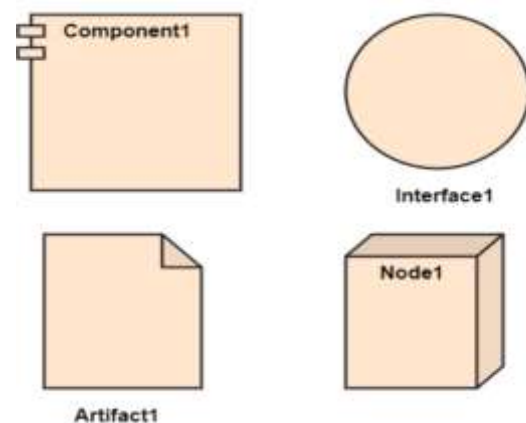
The deployment diagram consists of the following notations

A component

An artifact

An interface

A node



# CHAPTER 8

## 8. TECHNOLOGIES

The technologies used to develop certificate validation with SHA are

- 1.python
- 2.pycharm is the environment development we use.
- 3.In python we used some libraries they are hashlib(SHA-128)
4. For framework we used Tkinter.

### 8.1 Python

Python is an interpreted scripting language also. Guido Van Rossum is known as the founder of Python programming.

python is a general purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

Python is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development.

Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development.

Python supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles.

Python is not intended to work in a particular area, such as web programming. That is why it is known as multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc.

We don't need to use data types to declare variable because it is dynamically typed so we can write `a=10` to assign an integer value in an integer variable.

Python makes the development and debugging fast because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

The biggest strength of Python is huge collection of standard libraries which can be used for the following –

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

**Advantages of Python: -**

Let's see how Python dominates over other languages.

### **1. Extensive Libraries**

Python downloads with an extensive library and it contains code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

### **2. Extensible**

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

### **3. Embeddable**

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

### **4. Improved Productivity**

The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

## **8.2 History of python:**

Python is a widely-used general-purpose, high-level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

In the late 1980s, history was about to be written. It was that time when working on Python started. Soon after that, Guido Van Rossum began doing its application-based work in December of 1989 at Centrum Wiskunde & Informatica (CWI) which is situated in the Netherlands. It was started firstly as a hobby project because he was looking for an interesting project to keep him occupied during Christmas. The programming language in which Python is said to have succeeded is ABC Programming Language, which had interfacing with the Amoeba Operating System and had the feature of exception handling. He had already helped to create ABC earlier in his career and he had seen some issues with ABC but liked most of the features. After that what he did was really very clever. He had taken the syntax of ABC, and some of its good features. It came with a lot of complaints too, so he fixed those issues completely and had created a good scripting language that had removed all the flaws. The inspiration for the name came from BBC's TV Show – 'Monty Python's Flying Circus', as he was a big fan of the TV show and also he wanted a short, unique and slightly mysterious name for his invention and hence he

named it Python! He was the “Benevolent The language was finally released in 1991. When it was released, it used a lot fewer codes to express the concepts, when we compare it with Java, C++ & C. Its design philosophy was quite good too. Its main objective is to provide code readability and advanced developer productivity. When it was released, it had more than enough capability to provide classes with inheritance, several core data types exception handling and functions.

Python laid its foundation in the late 1980s.

The implementation of Python was started in December 1989 by Guido Van Rossum at CWI in Netherland.

In February 1991, Guido Van Rossum published the code (labeled version 0.9.0) to outsources.

In 1994, Python 1.0 was released with new features like lambda, map, filter, and reduce.

Python 2.0 added new features such as list comprehensions, garbage collection systems.

On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify the fundamental flaw of the language.

ABC programming language is said to be the predecessor of Python language, which was capable of Exception Handling and interfacing with the Amoeba Operating System.

### **How to download the python:**

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3. Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices. Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python. The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: <https://www.python.org>





Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.



**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

Step 4: Scroll down the page until you find the Files option.

### Looking for a specific release?

Python releases by version number:

Release version	Release date	Click for more	
<a href="#">Python 3.10.9</a>	Dec. 6, 2022	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.9.16</a>	Dec. 6, 2022	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.8.16</a>	Dec. 6, 2022	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.7.16</a>	Dec. 6, 2022	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.11.0</a>	Oct. 24, 2022	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.9.15</a>	Oct. 11, 2022	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.8.15</a>	Oct. 11, 2022	<a href="#">Download</a>	<a href="#">Release Notes</a>

[View older releases](#)

Step 5: Here you see a different version of python along with the operating system

## Files

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore
<a href="#">Gzipped source tarball</a>	Source release		6dbe644dd1a520d9853cf6648084c346	26071329	<a href="#">SIG</a>	<a href="#">CRT</a> <a href="#">SIG</a>
<a href="#">XZ compressed source tarball</a>	Source release		7bf85df71bbe7f95e5370b983e6ae684	19627028	<a href="#">SIG</a>	<a href="#">CRT</a> <a href="#">SIG</a>
<a href="#">macOS 64-bit universal2 installer</a>	macOS	for macOS 10.9 and later	892634724ab799569b512082c8f48c83	41005648	<a href="#">SIG</a>	<a href="#">CRT</a> <a href="#">SIG</a>
<a href="#">Windows embeddable package (32-bit)</a>	Windows		a681a7f9b242fe35b4d96d79e15e57d6	7663448	<a href="#">SIG</a>	<a href="#">CRT</a> <a href="#">SIG</a>
<a href="#">Windows embeddable package (64-bit)</a>	Windows		f38a9e7e02a992daa62569b758d0a388	8625602	<a href="#">SIG</a>	<a href="#">CRT</a> <a href="#">SIG</a>
<a href="#">Windows help file</a>	Windows		448f8401ade49a7e2156d02512f2f9bf	9391521	<a href="#">SIG</a>	<a href="#">CRT</a> <a href="#">SIG</a>
<a href="#">Windows installer (32-bit)</a>	Windows		a81b81687bc2575c05a30f4b31d6ea00	27859200	<a href="#">SIG</a>	<a href="#">CRT</a> <a href="#">SIG</a>
<a href="#">Windows installer (64-bit)</a>	Windows	Recommended	9735797853cba809b13c8396c91354a0	29010904	<a href="#">SIG</a>	<a href="#">CRT</a> <a href="#">SIG</a>

To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

### SHA (secure hash algorithm) for create hash value for the certificate:

The "Secure Hash Algorithm" is widely known as SHA. The Secure Hash Algorithm is a cryptographic hash function. A cryptographic hash function is an algorithm that randomly takes data as input without a specific reason and produces an output of text in a coded form called "Hash value". The coded text will be stored instead of the password that is used to verify the user, and this enciphered text is used to verify the user instead of the password. The SHA is also a non-reversible function similar to other cryptographic hash functions. SHA can be used to create a text signature by taking input of 20 bytes long maximum. The Secure Hash Function returns a 40-digit hexadecimal hash value as its output. Even the smallest changes in the input can make a big difference in the coded text output. The phenomenon is called the avalanche effect. The avalanche effect helps in securing the user data from attackers as it makes the decrypting of code difficult.

types of SHA

There are several different forms of the Secure Hashing Algorithm. The following forms of SHA are mentioned below:

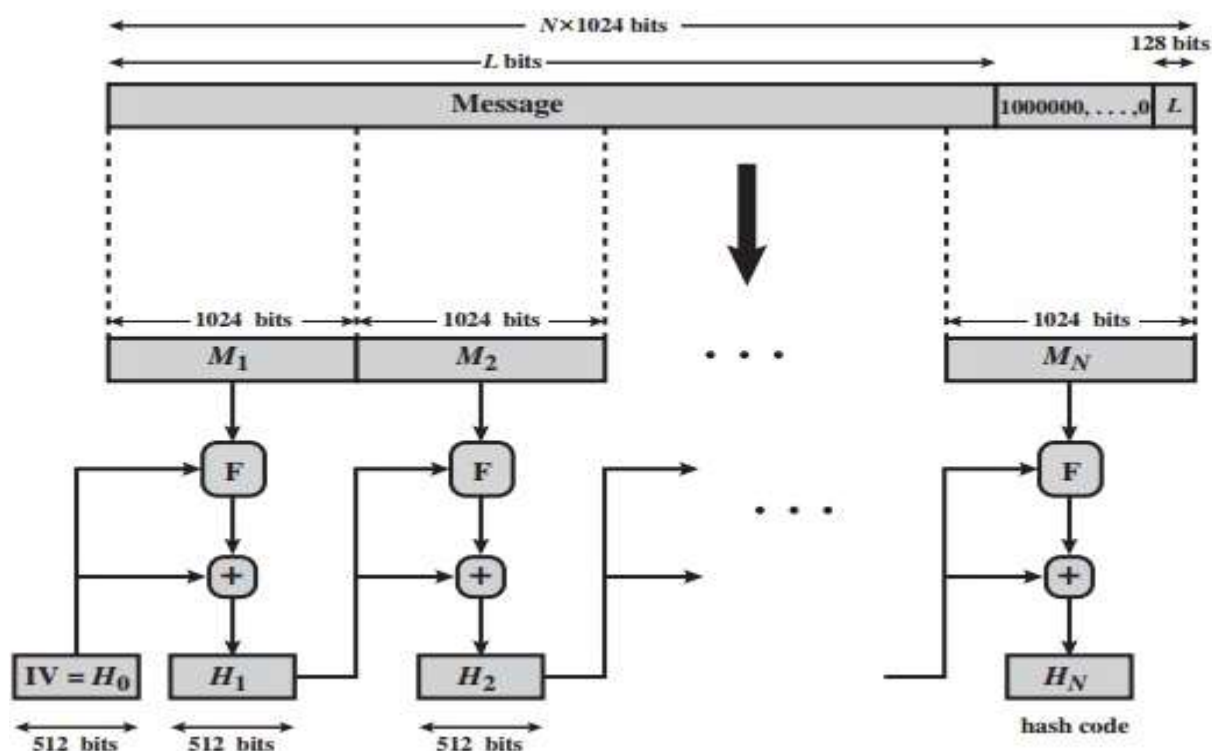
SHA-1

SHA-2

SHA-256

SHA-512

SHA-224



## IN PYTHON TO CREATE THE HASH VALUE WE USE HASHLIB LIBRARY:

hashlib is a Python library module that provides a collection of hash functions, which are algorithms that take input data and produce a fixed-size, deterministic output called a hash value or digest. This library can be used to securely store passwords, verify the integrity of data, and to ensure that data hasn't been tampered with or modified

Here are some of the commonly used hash functions provided by hashlib:

md5(): Computes the MD5 hash of the input data.

sha1(): Computes the SHA-1 hash of the input data.

sha224(): Computes the SHA-224 hash of the input data.

sha256(): Computes the SHA-256 hash of the input data.

sha384(): Computes the SHA-384 hash of the input data.

sha512(): Computes the SHA-512 hash of the input data.

```
python
```

[Copy code](#)

```
import hashlib
```

## TO CREATE THE FRAME WORK WE USE THE TKINTER:

### Python Tkinter Tutorial

Tkinter tutorial provides basic and advanced concepts of Python Tkinter. Our Tkinter tutorial is designed for beginners and professionals.

Python provides the standard library Tkinter for creating the graphical user interface for desktop-based applications.

Developing desktop-based applications with python Tkinter is not a complex task. An empty Tkinter top-level window can be created by using the following steps.

import the Tkinter module.

Create the main application window.

Add the widgets like labels, buttons, frames, etc. to the window.

Call the main event loop so that the actions can take place on the user's computer screen

## Example

```
#!/usr/bin/python3
from tkinter import *
#creating the application main window.
top = Tk()
#Entering the event main loop
top.mainloop()
```

For work environment we used PyCharm or vs studios:

### PyCharm:

PyCharm is an Integrated Development Environment (IDE) used for programming in Python. It is developed by JetBrains and is available as a commercial version and as a free, open-source community edition. PyCharm provides a range of features to help developers write, debug, and test Python code more efficiently. Some of the key features of PyCharm include code analysis, code completion, debugging, version control integration, and testing tools. It also supports various frameworks such as Django, Flask, and Pyramid, making it a popular choice for web



development projects.

### VS STUDIOS:

Visual Studio (VS) is an Integrated Development Environment (IDE) developed by Microsoft. It is used to develop applications for Windows, web, and mobile platforms, as well as cloud-based services. VS supports a range of programming languages, including C++, C#, Python, JavaScript, and others.

Some of the key features of Visual Studio include code editing, debugging, profiling, version control integration, and automated testing. It also offers a range of tools for creating user interfaces, such as WPF, WinForms, and

ASP.NET.



# CHAPTER 9

## 9 CODE

The certificate validation with SHA project is a Python-based application that allows users to create and validate digital certificates using the SHA-256 hash algorithm. The project is designed to ensure the integrity and authenticity of certificates by verifying that they have not been tampered with or altered in any way.

The application consists of a few key functions that work together to create and validate certificates. The `saveCertificate()` function is used to generate a SHA-256 hash value for a given input string, while the `create certificate` combines user-provided information such as the name, roll no, mobile number and course to create a unique certificate using the SHA-256 hash algorithm. Finally, the `verfiycertificate()` function checks if a given certificate is valid by comparing the hash value of the certificate with a newly generated hash value based on the user-provided information. To use the application, users can simply provide their information such as their name, the date of the course completion, and the course name to generate a unique certificate. The resulting hash value is then stored alongside the certificate, which can be shared with others as a proof of completion. To validate a certificate, the user can input the certificate details and hash value and check whether the hash values match. Overall, the certificate validation with SHA project provides a simple yet effective way to ensure the integrity and authenticity of digital certificates, making it a useful tool for organizations or individuals looking to issue and verify certificates.

In this project we have designed following modules

**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 local server.

**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 database and if matched found then it will retrieve all student details and display to verifier and if match not found then this certificate will be considered as fake or forge.

To design certificate validation with sha we have written following code shown in screen shots and in below screen shots read red color comments to understand code

```
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)

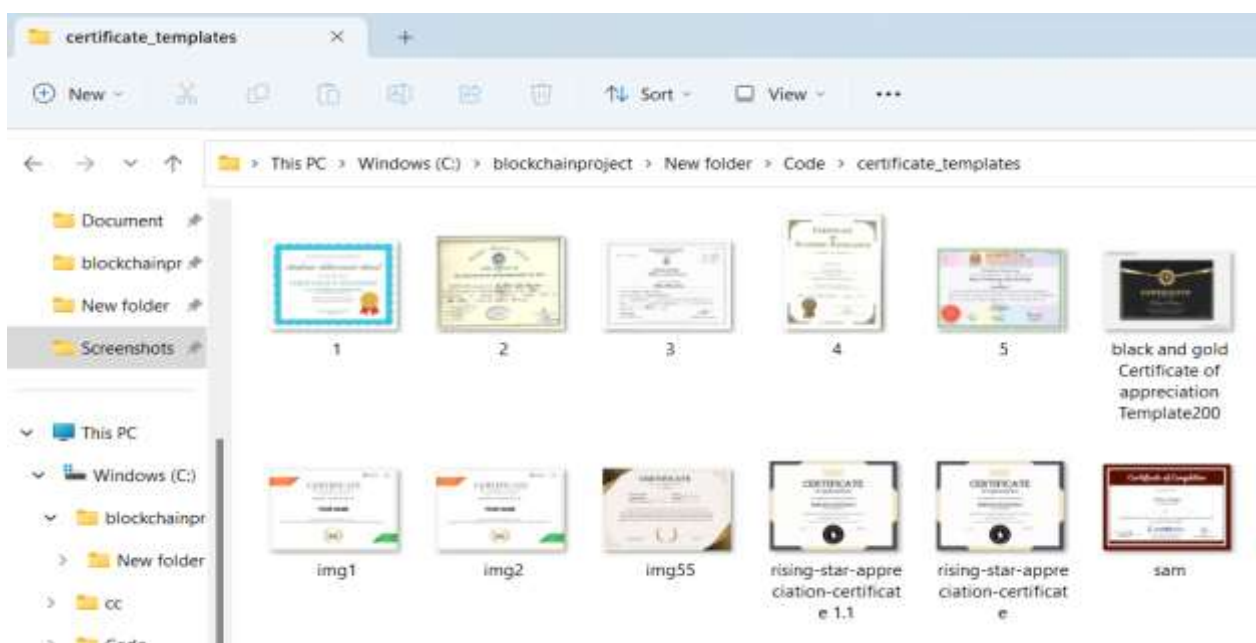
    proof = self.proof_of_work(new_block)
    self.add_block(new_block, proof)

    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)

```

To implement this project, we have taken some certificates and this certificates stored inside 'certificate\_templates' and you can use those or you own certificates to upload to Blockchain and below is the certificate screen shots.



### **To save and validate the certificate code**

```
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("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()
```

### **save certificate with digital signature:**

```
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")

```

#### **verify certificate code**

```

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 database 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")

```

#### **Tkinter Code for User Inter Face:**

```

font = ('times', 15, 'bold')
title = Label(main, text='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)

```

```

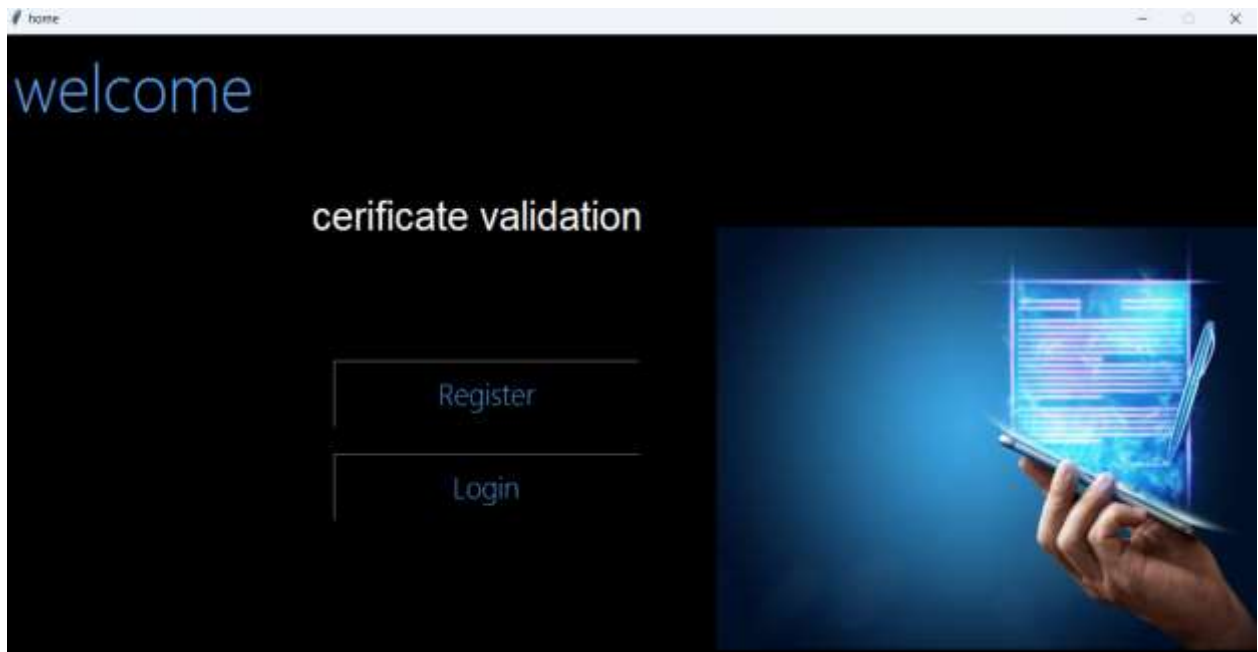
tf2 = Entry(main,width=20)
tf2.config(font=font1)
tf2.place(x=180,y=150)
l3 = Label(main, text='Contact No :')
l3.config(font=font1)
l3.place(x=50,y=200)
tf3 = Entry(main,width=20)
tf3.config(font=font1)
tf3.place(x=180,y=200)
saveButton = Button(main, text="Save Certificate with Digital Signature",
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()

```

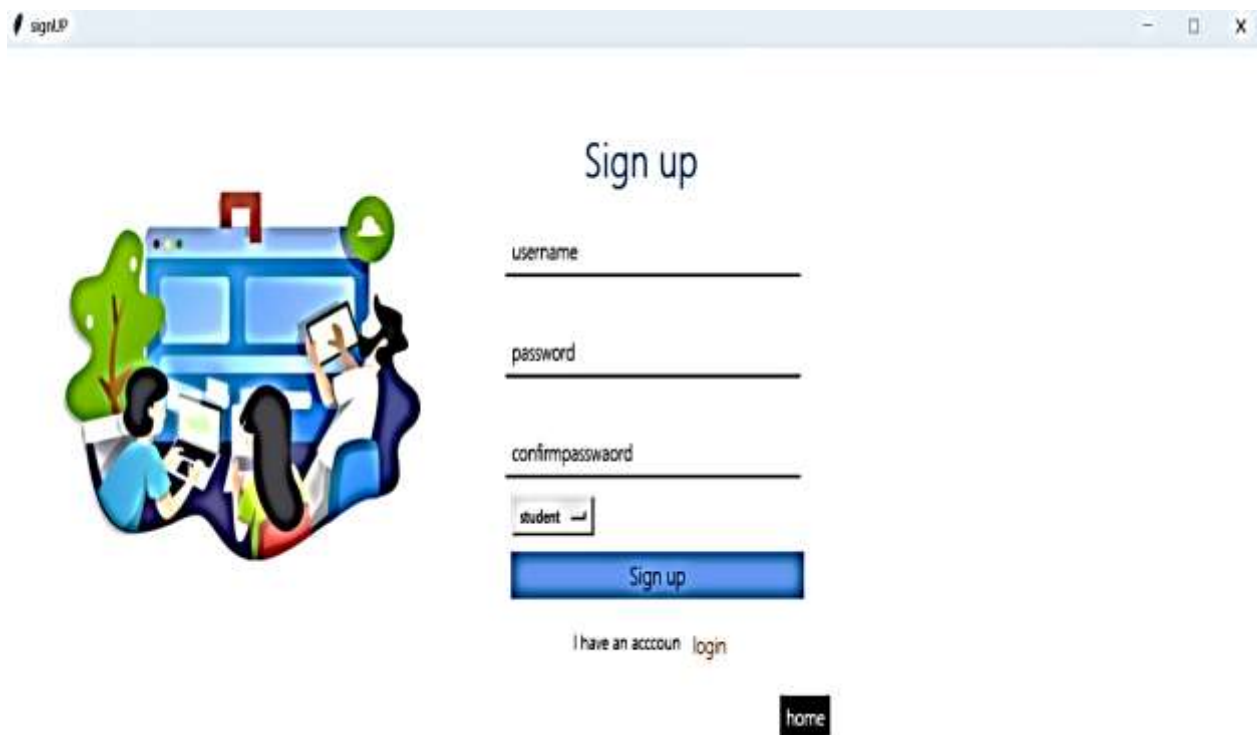
# CHAPTER 10

## 10. SCREENSHOTS

### HOME SCREEN




### OUTPUT:-REGISTRATION PAGE



### LOGIN PAGE SCREEN SHOT

login

## Sign In



username

password

[forgetpassword](#)

[I do not have an account](#) [Signup](#)

### OUTPUT SCREEN FOR INSTITUTE OR ORGANIZATION

Certificate Validation

### Certificate Validation

Roll No :

Student Name :

Contact No :

## Output for students



A screenshot of a web application window titled "Certificate Validation". The window has a light blue header bar with the title. Below the header, there is a large orange banner with the text "Certificate Validation" in purple. The main content area has a blue background. On the left side, there are three input fields with labels: "Roll No :", "Student Name :", and "Contact No :". To the right of these fields is a "Verify Certificate" button. Below the input fields and button is a large white rectangular area, likely for displaying the certificate details.

## Create digital signature

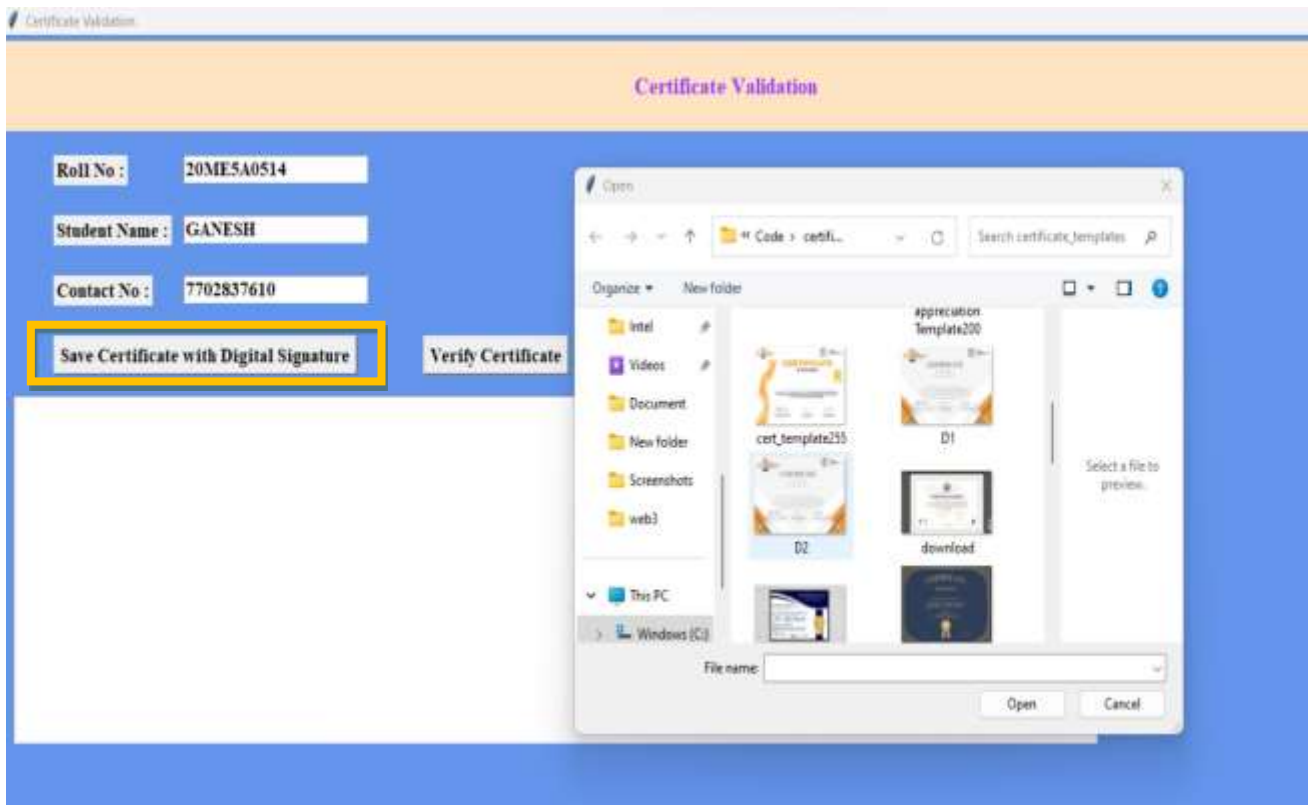
**Step 1:** Enter roll no and student name contact no.



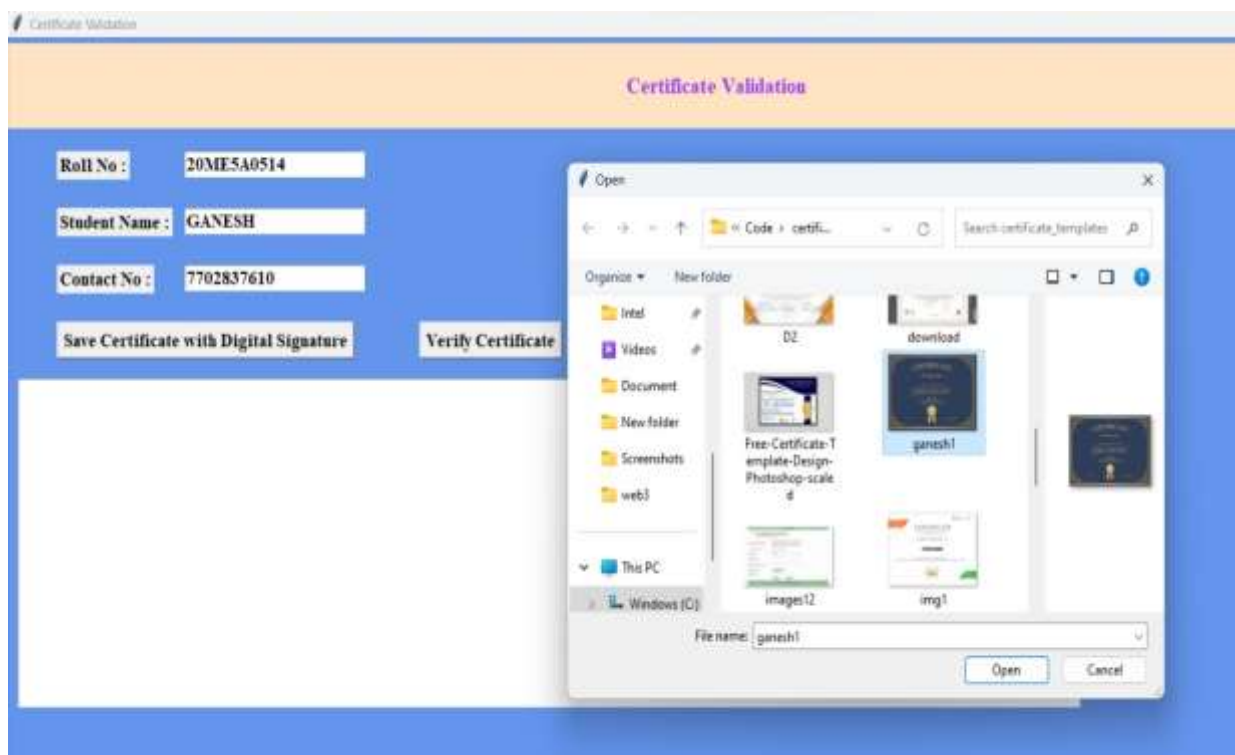
A screenshot of the same "Certificate Validation" web application window. The input fields are now filled with data: "Roll No :" contains "20ME5A0514", "Student Name :" contains "GANESH", and "Contact No :" contains "7702837610". Below the input fields, there are two buttons: "Save Certificate with Digital Signature" and "Verify Certificate". The large white rectangular area at the bottom is still empty.



**Step 2:** Click on save certificate



**Step3:** Select certificate



**Step 4:** Now click on “save certificate with digital signature”

The screenshot shows a web application titled "Certificate Validation". It features a blue header with the title in purple. Below the header, there are three input fields: "Roll No :" with the value "20ME5A0514", "Student Name :" with the value "GANESH", and "Contact No :" with the value "7702837610". Below these fields are two buttons: "Save Certificate with Digital Signature" and "Verify Certificate". At the bottom, there is a white box containing the following text:

Blockchain Previous Hash : 00064a6de37a839cb83e14353fe86c4ffe154202fc4acbf52a4658b92c3ed127  
Block No : 22  
Current Hash : 0026925cc750f55071b73e0c5af8b2cc9e236d599fcc91ff8521270811f060d5  
Certificate Digital Signature : f4460f9ed3ddb3e378cd75283d700a5c6279f2f5bf951ebde05a7d5c778f3b59

The digital signature was saved in this form at local server

The screenshot shows a Notepad++ window with the title "blockchain\_contract". The text inside the window is a JSON string representing a blockchain transaction. The string is as follows:

```
"ubh)"}}"(h
K
]"EP123#kkrr#582582
#ca8316bc778aae77eb543484fe2d0539157992d070e90a89a5c6e2ad5464ba8
e"ahGAU%;1=hxhKSh@
00064a6de37a839cb83e14353fe86c4ffe154202fc4acbf52a4658b92c3ed127
"ubh)"}}"(h
K
]"E]20ME5A0514#GANESH#7702837610
#f4460f9ed3ddb3e378cd75283d700a5c6279f2f5bf951ebde05a7d5c778f3b5
9"ahGAUc
h}hK h@
0026925cc750f55071b73e0c5af8b2cc9e236d599fcc91ff8521270811f060d5
"ubepeer]"E translist]"ub.
```

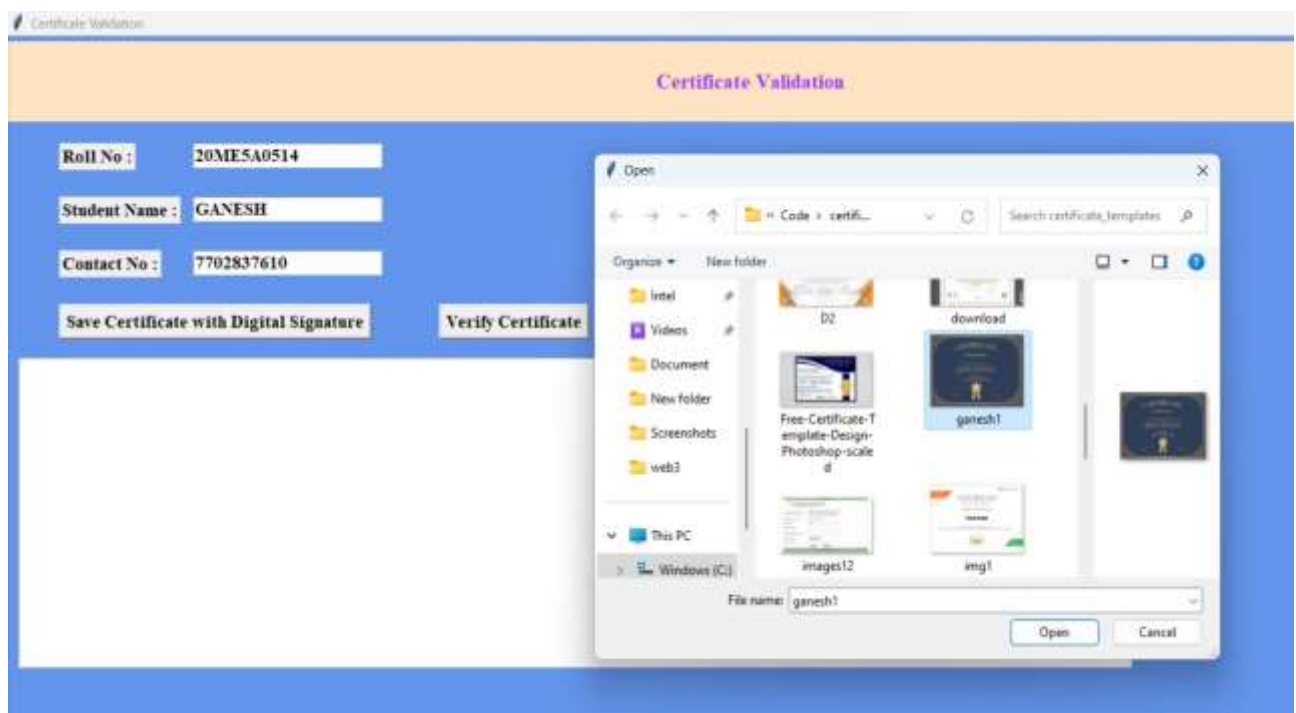
**If you want to validate the certificate these are the steps**

**Step 1:** Fill the roll no, student name, phone number



The screenshot shows a web application titled "Certificate Validation". It features a blue header with the title in purple. Below the header, there are three input fields: "Roll No :" with the value "20ME5A0514", "Student Name :" with the value "GANESH", and "Contact No :" with the value "7702837610". Below these fields are two buttons: "Save Certificate with Digital Signature" and "Verify Certificate". A large white rectangular area is positioned below the buttons.

**Step 2:** Click on verify certificate



Step 3:  
Now click on verify certificate



The screenshot shows a web application titled "Certificate Validation". It features a form with three input fields: "Roll No :" with the value "20ME5A0514", "Student Name :" with the value "GANESH", and "Contact No :" with the value "7702837610". Below these fields are two buttons: "Save Certificate with Digital Signature" and "Verify Certificate". The "Verify Certificate" button is highlighted. Below the buttons, a message states "Uploaded Certificate Validation Successfull" and "Details extracted from database after Validation". This is followed by a list of details: "Roll No : 20ME5A0514", "Student Name : GANESH", "Contact No : 7702837610", and "Digital Sign : f4460f9ed3ddb3e378cd75283d700a5c6279f2f5b0951ebde05a7d5c778f3b59".

**Certificate Validation**

Roll No : 20ME5A0514

Student Name : GANESH

Contact No : 7702837610

Save Certificate with Digital Signature      Verify Certificate

Uploaded Certificate Validation Successfull  
Details extracted from database after Validation

Roll No : 20ME5A0514  
Student Name : GANESH  
Contact No : 7702837610  
Digital Sign : f4460f9ed3ddb3e378cd75283d700a5c6279f2f5b0951ebde05a7d5c778f3b59

This the process and screen shot of the certificate validation with SHA.

# CHAPTER 11

## 11. TESTING

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property function as a unit. The test data should be chosen such that it passed through all possible condition. The following is the description of the testing strategies, which were carried out during the testing period.

### 11.1 Unit Testing:

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus, all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example, the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time

#### a. Integration testing:

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system Integration testing in certificate validation with SHA in a Python project involves testing the interactions between different components or modules that are responsible for certificate validation using SHA. Here are some general steps that can be followed to conduct integration testing in certificate validation with SHA in a Python project.

- Identify the components: Identify the individual components or modules that are responsible for certificate validation with SHA in the Python project. These may include cryptographic libraries, certificate validation libraries, or custom code developed for the project.
- Define test cases: Define test cases that test the interactions between the different components. These test cases should include both positive and negative test cases.
- Develop test scripts: Develop test scripts that automate the execution of the test cases.

These scripts should include code to configure the test environment, execute the test cases, and verify the results.

- Execute the tests: Run the test scripts to execute the test cases. During the execution, the scripts should log any errors or issues encountered during the testing.
- Analyze the test results: Analyze the results of the tests to determine if the components are interacting correctly. If errors or issues are found, debug the code to identify the root cause of the problem.
- Provide feedback: Provide feedback to the development team on the results of the integration testing. This feedback can be used to improve the quality and functionality of the code.
- Document the testing: Document the test cases, test scripts, and test results for the integration testing. This documentation can be used to verify the functionality of the code during future updates or changes.

#### **b. Acceptance Testing:**

When that user find no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

#### **TEST CASES:**

**Test case 1:-To check it create digital signature or not**



Certificate Validation

Certificate Validation

Roll No : 20ME5A0514

Student Name : GANESH

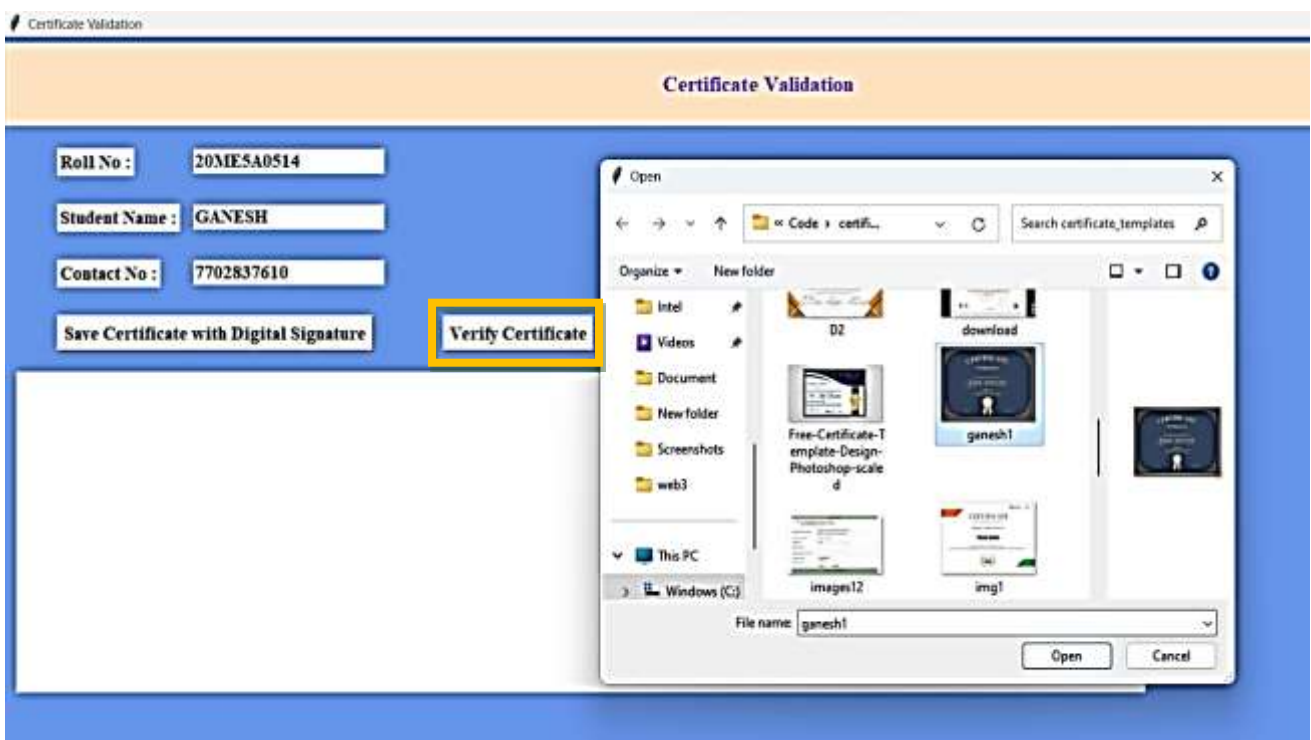
Contact No : 7702837610

Save Certificate with Digital Signature      Verify Certificate

Blockchain Previous Hash : 00064a6de37a839cb83e14353fe86c4ffe154202fc4acbf52a4658b92c3ed127  
Block No : 22  
Current Hash : 0026925cc750f55071b73e0c5af8b2cc9e236d599fcc91ff8521270811f060d5  
Certificate Digital Signature : f4460f9ed3ddb3e378cd75283d700a5c6279f2f5bf951ebde05a7d5c778f3b59

here the output is coming when certificate is converted to digital signature

**test case2: - validate the certificate**





HERE THE OUTPUT IS CORRECT

Here these are some other test cases

Test Case Id	Test Case Name	Test Case Desc.	Test Steps			Test Case Status	Test Priority
			Step	Expected	Actual		
01	Upload the tasks dataset	Verify either file is loaded or not	If dataset is not uploaded	It cannot display the file loaded message	File is loaded which displays task waiting time	High	High
02	login page	Enter correct user detail	Enter details are wrong	Not show the warning message	Shows warning message	low	High
03	Login page to next page correctly	By entering correct details based on details if he is student then it should divert to student page	Enter write details	It will display correct output	Correct output	high	High

# **CHAPTER 12**

## 12. CONCLUSION

certificate validation with SHA is an important part of many software projects that require secure communication between different parties. In a Python project, it is important to test the certificate validation process thoroughly to ensure that it is working as intended and is secure.

Unit testing, module testing, integration testing, and acceptance testing are all important types of testing that can be used to test the certificate validation process in a Python project. These types of testing can help to identify errors or issues early in the development process

Overall, certificate validation with SHA in a Python project requires careful planning, testing, and documentation to ensure that the system is secure and functions correctly. By following best practices for testing and development, software engineers can create robust and secure systems that meet the needs of their users.

Here the certificate validation with SHA is used to verify the certificates. The user can upload his certificate and create the digital signature, hash code to that certificate it was created with the help of hashlib (SHA-256) library, he can verify the certificate with the help of digital signature. The certificate signature and hash code were stored in the database by comparing the original.

# CHAPTER 13

### 13. REFERENCES

- [1] Tengyu Yu, Blockchain operation principal analysis: 5 key technologies, iThome, <https://www.ithome.com.tw/news/105374>
- [2] JingyuanGao, The rise of virtual currencies! Bitcoin takes the lead, and the other 4 kinds can't be missed. Digital Age, <https://www.bnnext.com.tw/article/47456/bitcoinether-li-tecoin-ripple-differences-between-cryptocurrencies>
- [3] Weiwen Yang, Global blockchain development status and trends, 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
- [4] 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.
- [5] Yuan, Yong, and Fei-Yue Wang, "Blockchain: the state of the art and future trends," *Acta Automatica Sinica* 42.4 (2016): 481-494.
- [6] Blockchain Based Storage and Verification Scheme of Credible Degree Certificate 1 Dongwei Liu 2 Xiaojin Guo

These references provide guidance on best practices, standards, and tools for certificate validation with SHA in a Python project. It is important to keep up-to-date with the latest guidelines and best practices for secure software development to ensure the integrity and security of the software system.