



USMAN INSTITUTE OF TECHNOLOGY

Affiliated with NED University of Engineering & Technology, Karachi

Department of Computer Science

B.S. Computer Science / Software Engineering FINAL YEAR PROJECT REPORT

Batch-2017

Blockchain Based E-Certification

By

Muhammad Rafay	17B-008-SE
Muhammad Aamir	17B-034-SE
Hassan Ahmed Siddiqui	17B-049-SE
Muhammad Sabih Mohsin	17B-058-SE
Muhammad Ismail	17B-062-SE

Supervised by

Dr. Muhammad Qasim Pasta

Acknowledgments

There have been many individuals whom we are thankful for helping us to accomplish our final year project (FYP), and we would like to express our special thanks to all of them for their support and motivation.

First and foremost, during the project, we would like to thank the Almighty for his blessing. We would not be able to finish this mission without the grace of the Almighty.

Secondly, we would like to thank our director Dr. Zahir Ali Syed and our HoD Parkash Lohana for their efforts and support in overall curriculum.

We would now like to express our special thanks to our supervisor Dr. Muhammad Qasim Pasta who helped us and supported us in our work from the very first day, during this journey he has been an outstanding mentor and counselor because of his untiring efforts in giving directions that helped us to understand and tackle any kind of problem.

We would also like to thank our family who have been incredibly supportive during this project, in particular our parents for their non-stop prayers and true conviction that this could not have been achieved without their inspiration, and for supporting us financially as well as mentally.

We would also like to thank our university, all teachers, FYP team and friends who supported us somehow by offering vital knowledge and constructive criticism that helped us to properly finalize the project.

Abstract

The certification is one of the main sources of recognition of one's work but how do we maintain the authenticity of the certificates is the first thing that pops in mind. Is there any way we can verify the certificate do we always need to contact the organization to verify it what if something happened to the organizations or they decided to switch platforms then what? Our project aims to solve all these problems to maintain the authenticity of the certificate by using distributed architecture of blockchain technology so that no one will tamper the certificate your data is save with us. We provide hassle free verifications and recovery of certificates so even the organization gets shutdown or no matter where you are in world you can still verify and recover your certificates with ease. Some feature will be added in the future and these features include detail audit report, fully customizable certificate designer, etc. also currently we have services that are running on the single http server so we have decided to convert the whole architecture to micro service architecture to make the system more available to the user during the time of maintenance or any service failure.

Table of Contents

Contents

ACKNOWLEDGMENTS	2
ABSTRACT.....	3
TABLE OF CONTENTS	4
LIST OF TABLES	7
LIST OF FIGURES	8
LIST OF SYMBOLS AND UNITS	10
1. INTRODUCTION.....	13
1.1 PROBLEMS.....	13
1.2 SOLUTION.....	13
1.2.1 <i>Distributed Ledger</i>	13
1.2.2 <i>Decentralization</i>	14
1.3 BLOCKCHAIN AND SMART CONTRACT	14
1.3.1 <i>Chaincode</i>	14
1.4 BACKGROUND	14
1.5 SYSTEM DIAGRAM	15
1.6 OVERVIEW	16
2. BACKGROUND AND LITERATURE REVIEW	18
2.1 BLOCKCHAIN.....	18
2.1.1 <i>Different Type of Blockchains</i>	18
2.1.2 <i>Hyperledger Fabric</i>	20
2.2 DEVELOPMENT LANGUAGE.....	21
2.3 DATABASE PLATFORM	22
2.4 BLOCKCHAIN ADVANCEMENT IN CERTIFICATION	22
2.4.1 <i>KMI, OU - UK</i>	22
2.4.2 <i>UNIC</i>	22
2.4.3 <i>MIT Media Lab</i>	23
2.4.4 <i>Blockcert</i>	23
2.4.5 <i>Smartsert</i>	23
2.4.6 <i>RecordsKeeper</i>	23
2.4.7 <i>Accredible</i>	23

3. AIM AND STATEMENT OF PROBLEM.....	25
3.1 VERIFICATION	25
3.2 TAMPERING.....	25
3.3 DATA LOSS DUE TO MISMANAGEMENT OF DOCUMENT	25
3.4 CYBER ATTACK	25
3.5 SERVER DOWN.....	25
3.6 SECURITY LOOPHOLES	26
3.7 DATA REDUNDANCY	26
3.8 DATA BACKUP	26
4. HARDWARE, SOFTWARE ANALYSIS AND REQUIREMENTS.....	27
4.1 COMPARISON OF DIFFERENT BLOCKCHAIN TECHNOLOGIES.....	28
4.2 COMPARISON OF DIFFERENT NoSQL DATABASES.....	29
4.3 COMPARISON OF DIFFERENT WEB FRAMEWORKS	30
4.4 COMPARISON OF AVAILABLE SOFTWARE IN TERMS OF SALIENT FEATURES, FUNCTIONALITY AND SHORTCOMINGS.....	30
4.5 COMPARISON OF AVAILABLE SOFTWARE IN TERMS OF THEIR SECURITY THEMES.....	31
4.5.1 <i>Certifis</i>	33
4.6 DIAGRAMS	33
4.6.1 <i>System Diagram</i>	33
4.6.2 <i>Actor Use Case Diagram</i>	35
4.6.3 <i>Activity Diagrams</i>	37
4.7 REQUIREMENTS	43
4.7.1 <i>Functional Requirements</i>	43
4.7.2 <i>Non-Functional Requirement</i>	44
5. SOFTWARE DESIGN AND MODELING	47
5.1 WHAT IS DISTRIBUTED ARCHITECTURE?.....	47
5.2 BENEFITS OF DISTRIBUTED ARCHITECTURE	47
5.3 WHY ARE WE USING DISTRIBUTED ARCHITECTURE?	47
5.4 WEB INTERFACES.....	48
6. IMPLEMENTATION	60
6.1 CODE.....	60
6.1.1 <i>Smart Contract</i>	60
6.1.2 <i>Publish</i>	61
6.1.3 <i>Verify</i>	64

6.2	COMPONENT DIAGRAM	66
6.3	DEPLOYMENT DIAGRAM	67
6.4	STATE TRANSITION DIAGRAM.....	67
7.	TESTING.....	68
7.1	WHITE BOX TESTING	68
7.1.1	<i>Publish Single Certificate</i>	68
7.1.2	<i>Publish Batch Certificate</i>	71
7.1.3	<i>Verify Certificate</i>	74
7.2	BLACK BOX TESTING	77
7.2.1	<i>Publish Single Certificate Interface</i>	77
7.2.2	<i>Publish Batch Interface</i>	80
7.2.3	<i>Verify Certificate Interface</i>	83
7.2.4	<i>Validation Testing</i>	85
8.	CONCLUSION	92
9.	FUTURE ENHANCEMENT	93
10.	ACHIEVEMENTS.....	94
11.	APPENDICES	95
11.1	ANNEX A: CLASS DIAGRAM	95
11.2	ANNEX B: SEQUENCE DIAGRAM.....	101
11.3	ANNEX C: OBJECT DIAGRAM	112
12.	REFERENCES.....	116

List of Tables

Table 4.1 – Blockchain Technologies’ Comparison.....	28
Table 4.2 – NoSQL Databases’ Comparison.....	29
Table 4.3 - Web Frameworks’ Comparison.....	30
Table 4.4 - Existing Solutions and their Shortcomings	30
Table 4.5 - Certification Solutions Maps with Requirements Theme	31
Table 7.1 - Publish Single Certificate Test Cases.....	70
Table 7.2 - Publish Batch Test Cases.....	73
Table 7.3 - Verify Certificate Test Cases.....	76
Table 7.4 – Publish Single Certificate Functionality Testing	78
Table 7.5 - Publish Batch Functionality Testing	81
Table 7.6 - Verify Certificate Functionality Testing	84
Table 7.7 - Verify Certificate Textbox Validation Testing.....	85
Table 7.8 - Verify Certificate Button Validation Testing	90

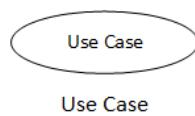
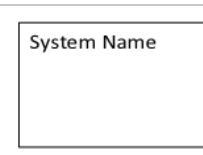
List of Figures

Figure 1.1 - System Diagram	15
Figure 4.1 - System Diagram	33
Figure 4.2 - Actor Use Case Diagram.....	35
Figure 4.3 - Create Single Certificate Activity Diagram	37
Figure 4.4 - Create Batches/ Batch Certificates Activity Diagram.....	38
Figure 4.5 - Certificate Publication Activity Diagram.....	40
Figure 4.6 - Verify/ Recover Certificates Activity Diagram	41
Figure 4.7 - Register Organization Activity Diagram	42
Figure 5.1 - Template Selection Web Interface	48
Figure 5.2 - Single or Batch Selection Web Interface	48
Figure 5.3 - Create Single Certificate Web Interface	49
Figure 5.4 - Create Batch Web Interface	49
Figure 5.5 - Single Certificate List Web Interface.....	50
Figure 5.6 - Batch List Web Interface	50
Figure 5.7 - Create Batch Certificates Web Interface.....	51
Figure 5.8 - Batch Certificates List Web Interface	51
Figure 5.9 - Published Single Certificate Web Interface	52
Figure 5.10 - Count History Web Interface	52
Figure 5.11 - User Management Web Interface.....	53
Figure 5.12 - About Web Interface	53
Figure 5.13 - Client Organization Graphical View Web Interface.....	54
Figure 5.14 - Client Organization List Web Interface	54
Figure 5.15 - Client Organization Count History Web Interface.....	55
Figure 5.16 - Client Organization User Management Web Interface	55
Figure 5.17 - Client Organization Update Profile Web Interface	56
Figure 5.18 - Client Organization About Web Interface	56
Figure 5.19 - Client Organization User Limit Web Interface	57
Figure 5.20 - Client Organization Published Single Certificate View Web Interface	57
Figure 5.21 - Client Organization Published Batch View Web Interface.....	58
Figure 5.22 - User Registration Web Interface	58
Figure 5.23 - Update User Profile Web Interface	59

Figure 5.24 - Certificate Verification Web Interface	59
Figure 6.1 - Component Diagram	66
Figure 6.2 - Deployment Diagram	67
Figure 6.3 - Publish's State Transition Diagram	67
Figure 6.4 - Verification's State Transition Diagram	67
Figure 7.1 - Publish Single Certificate Flow Graph	69
Figure 7.2 - Publish Batch Flow Graph	72
Figure 7.3 - Verify Certificate Flow Graph	75
Figure 7.4 - Publish Single Certificate Interface	77
Figure 7.5 - Publish Batch Interface	80
Figure 7.6 - Verify Certificate Interface	83
Figure 11.1 - Manage Certificate Class Diagram	95
Figure 11.2 - Manage Batch and Batch Certificate Class Diagram	96
Figure 11.3 - Verify Certificate Class Diagram	97
Figure 11.4 - Manage Organization Class Diagram	98
Figure 11.5 - Manage User Class Diagram	99
Figure 11.6 - Complete Class Diagram	100
Figure 11.7 - Manage Certificate Sequence Diagram	101
Figure 11.8 - Manage Batches/ Batch Certificates Sequence Diagram	103
Figure 11.9 - Certificate Publication Sequence Diagram	105
Figure 11.10 - Verify/ Recover Certificates Sequence Diagram	106
Figure 11.11 - Manage Organization Sequence Diagram	108
Figure 11.12 - Manage User Sequence Diagram	110
Figure 11.13 - Manage Certificate Object Diagram	112
Figure 11.14 - Manage Batch and Batch Certificates Object Diagram	113
Figure 11.15 - Manage Organization Object Diagram	113
Figure 11.16 - Manage User Object Diagram	114
Figure 11.17 - Complete Object Diagram	115

List of Symbols and Units

Use Case Diagram



Actor

— — <<exclude>> — —
— — <<include>> — —
Relationships

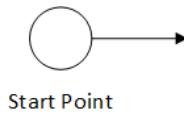
Draw the boundaries of your system using a rectangle containing use cases. Place actors outside of the boundaries of the system.

Draw use cases using ovals. Mark the ovals with nouns and verbs which reflect the functions of the system.

Actors are the users of a system.

Simple line represents relationships between an actor and a use case. For relationships among use cases, use arrows labeled either "include" or "exclude". A "include" relationship means that one use case is used by another in order to perform a task. In a particular use case, a "exclude" relationship implies additional choices.

Activity Diagram

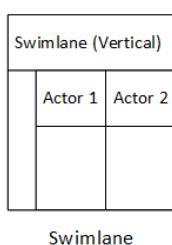


For every activity diagram, a small filled circle preceded by an arrow represents the start point. Make sure that the starting point is positioned in the top left corner of the first column for an activity diagram using swimlanes.

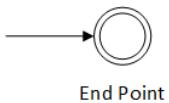
An action state represents the objects' non-interruptible behavior.

The transformations from one action state to another are illustrated by action flows, often called edges and paths.

A diamond reflects a choice of alternative routes. Attach a diamond between the two tasks where an action needs a decision before going on to the next activity. The outgoing alternates should be branded with the term of a state or guard. One of the routes can also be called "else."



Swimlanes group related activities into one column.

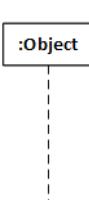


The end point depicts an arrow leading to a filled circle nested within another circle.

Sequence Diagram



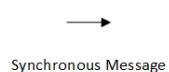
The way an object behaves in context is defined by class role.



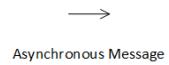
The continuation of an object over time is indicated by lifelines, which are vertical dotted lines.



The time it takes for an object to perform a task is expressed by activation boxes.



Before the connection can proceed, a synchronous message requires an response.



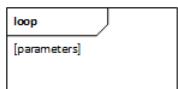
For contact to proceed, asynchronous messages do not require a response.



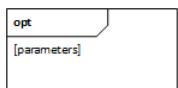
A message received from an object to itself.



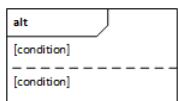
A dotted line and an open arrowhead point back to the initial lifeline in a return letter.



Loop



Optional



Alternate

A rectangle represents a loop in a sequence diagram. Place the condition for leaving the loop in square brackets [] at the bottom left corner.

Only if the supplied condition is correct does the fragment run. With just one trace, it's equivalent to an alt.

Only the condition that is valid will be executed.

1. Introduction

Certification has found its way into almost every industry for a reason: It helps advance the profession. Certification helps employers assess potential new hires, evaluate job performance, evaluate workers, pick contractors, market resources, and encourage employees to develop their skills and abilities. Certification provides recognition of skills, demonstrates devotion to the profession, and assists with career advancement. There has been an explosive growth in professional certification. Professional Certification helps one to: [1]

- Gain practical skills for the job, as it focuses on the knowledge and skills required to fulfill job responsibilities in the real world. [1]
- Realize the maximum benefit from a broad range of areas of expertise. [1]
- Learn all aspects of a particular type of career pursuit while providing these professionals with a standard of assured excellence for organizations. [1]
- Integrate these career pursuits effectively into one's specific work environment. [1]

1.1 Problems

The problem is that different certificates for events, courses, health, etc. are issued to the people from different organizations. But in case of any mishaps (natural disaster, misplace, etc.) person might lose his/her certificate, Sometime the organization shift the platform so it is very difficult process to get that certificate again due to lots of hurdles, verifying those certificates is really big issue, sometime if it is possible so it is very time consuming and costly process and sometime it is not possible due to lack of backup also preparing and distributing those certificates manually is so much time consuming.

1.2 Solution

We are providing a platform of blockchain based e-certification **CERTIFIS** which provides a way to verify the certificates not only by the certificate generating entity also by other users (either the part of any entity in the network or not) and in case any entity loss their data or change their platform then the certificates issued by them still be verifiable.

The Blockchain based e-certification project will meet the following objectives:

- Provide distributed platform for verifiability of certificate.
- No certificate or data loss due to any mishap.
- Tamper free securable environment.

1.2.1 Distributed Ledger

Distributed ledger technology, such as blockchain, are peer-to-peer networks that enable numerous users to manage a mutual ledger's own identical copy. DLTs allow their members to securely verify, execute, and record their own transactions without relying on a middleman, rather than having a central authority to update and communicate records to all participants. [2]

While there are a wide number of DLTs on the market, they all consist of the same building blocks: a public or private distributed ledger, a consensus algorithm (to ensure that all copies of the ledger are identical). [2]

1.2.2 Decentralization

As we know Hyperledger Fabric's design supports blockchain networks that are fully decentralized so we are providing a decentralized architecture with a consortium blockchain network that verifies such certificates by offering a means to validate certificates, not only by the certificate-generating organization, but by other organizations that are part of any entity in the network or not. In case of any server down or data loss due to any accident our data is still verifiable and secure due to decentralized architecture.

1.3 Blockchain and Smart Contract

Hyperledger Fabric smart contracts are written in chaincode and are invoked by an application external to the blockchain where the application wants to communicate with the ledger. In most instances, chaincode only communicates with the ledger's database part, the world state (for instance, querying it) and not the transaction log. [3]

1.3.1 Chaincode

Chaincode is software that identifies an asset or properties, and instructions for changing the asset(s) in the transaction. The chaincode enforces the rules for reading or changing key-value pairs or other information from the state database. Chaincode functions run against the existing state database of the ledger and are triggered by a proposal for a contract. The execution of Chaincode results in a series of key-value writes (write set) which can be sent to the network and added to all peers in the ledger. [4]

1.4 Background

There are approximately 10 companies that worked on digital verification of certificates with blockchain technology but no one is giving the solution that if they themselves dismissed than which entity will provide platform for verification of certificates. So, from here we enter the problem domain, we come with solution that there will be multiple organizations that will host the platform. Each organization will have the blockchain nodes and web servers for verification requests so that if any of the organization dismissed then other organizations hosting the network will provide the verifiability of certificates note that the user's verification request will be handled automatically from any of the available originations' servers.

1.5 System Diagram

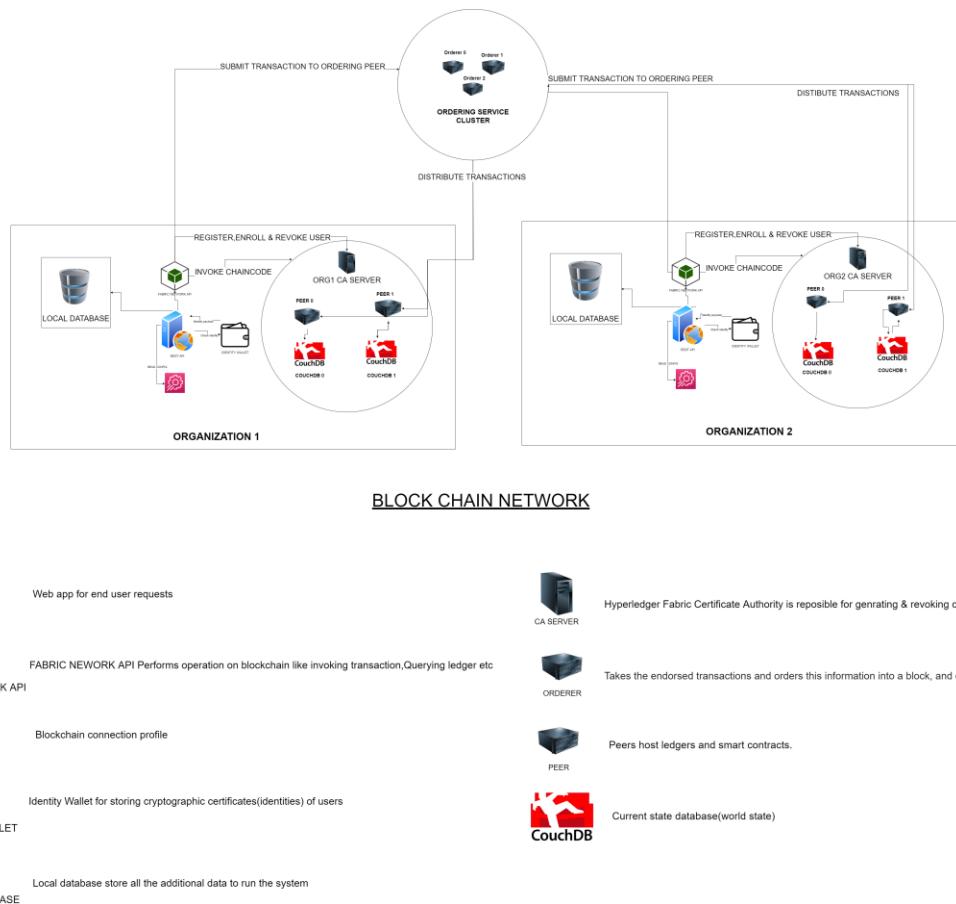


Figure 1.1 - System Diagram

The system includes the following:

- **Websvserver:** Web app which handles the end user requests.
- **Local Database:** Store web generated data like user profiles, certificates data, configs for each tenant
- **Peers:** It hosts the ledger and hold the smart contracts.
- **Ordering Peers:** Take the endorsed transaction and orders this information into a block and deliver the block to all committing peers.
- **World State (CouchDB):** It stores current state of the ledger.
- **Fabric CA Server:** Fabric CA is responsible for generating and revoking cryptographic material for nodes and users.
- **Fabric SDK (Fabric Network API):** A higher-level API for interacting with smart contracts and is the recommended API for client applications to interact with smart contracts deployed to a Hyperledger Fabric blockchain network.
- **Identity Wallet:** It stores the cryptographic identities of users.

1.5.1.1 System Workflow

1.5.1.1.1 Common Operations Flow

The flow is show in figure 1.1 describes when users send request to the webserver and the web server responds to the requests all the data of organization stored on the local DB may be single instance or clustered.

The common operation includes:

- Manage Organizations.
- Manage User.
- Manage Batches.
- Manage Publications.

1.5.1.1.2 User Creation Flow

The flow is show in figure 1.1 describes when users submit the registration request to the webserver now webserver store all the data in its local database and send the request to fabric CA server for cryptographic identity of the user so that the user perform operations on the blockchain.

1.5.1.1.3 Publishing to Blockchain Flow

The flow is show in figure 1.1 describes when the user send the certificate publishing request to the server, then for publishing the data on the blockchain web server check the identity wallet for the user's identity of blockchain to authenticate it and now using the fabric SDK it submits the request to the blockchain the fabric SDK read the connection profile of blockchain and run the discovery service to find the endorser after discovery it will send the endorsing proposal to the endorsing peer after successful response it submits that signed proposal to the ordering service cluster the ordering service orders the transaction and send to the peer organization where the transaction is validated and committed to the ledger.

1.6 Overview

In **chapter 1** we covered the brief introduction of our project. This chapter contains problem and how we are solving this problem. This chapter describe what the project is all about and why it was undertaken also we enlighten our project with some similar applications and discussed the system diagram of our project.

In **chapter 2** we covered background and literature review of our project. This chapter contains all the literature review that is covered for our project. In this chapter we briefly discussed the introduction of all the available blockchain technologies and in detail we discussed about the technology that we are using for our project also which development language and database platform is used, at last we discussed about all the similar software that are available till now.

In **chapter 3** we covered aim and statement of problem of our project. In this chapter we discussed about the strategies of how we are going to tackle the shortcomings of the similar software which are discussed in the previous chapter.

In **chapter 4** we covered software analysis and requirements of our project. This chapter contains the comparison of all the blockchain technologies, database platforms, web frameworks and the similar software also some of the important diagrams of our project and

the description of those diagrams, at last we discussed about the functional and non-functional requirements of our project.

In **chapter 5** we covered Software design and modeling of our project. This chapter contains the brief description of distributed architecture, its benefits and why we used it also this chapter contain all the interfaces of our web application.

In **chapter 6** we covered implementation of our project. This chapter contains all the implementation procedure require for our application along with how to implement it.

In **chapter 7** we covered testing of our project. In This chapter we have created blackbox and whitebox test case and performed testing against those test cases to insure proper working of our application.

2. Background and Literature Review

As we know that e-certificate are digitally verifiable and signed with the cryptographic signature of the issuing entity so the certificate legitimacy is trusted and anyone can verify who issued that certificate, so we are providing a blockchain based e-certification platform which is **CERTIFIS** and its aim is to provide a platform for organizations where users can generate certificates also make them able to verify those certificates. In case any organization removes its data or switches its platform, the certificates provided by them will still be verifiable due to decentralized architecture.

2.1 Blockchain

A special category of ledger is the blockchain. The way it manages information varies from a traditional database; blockchain stores data in blocks that are then chained together. When new data comes in a new block is joined. It is chained to the previous block until the block is loaded with data, which renders the details chained together in chronological order. It is possible to store various kinds of information on a blockchain, but the most common use to date has been as a transaction ledger. Decentralized blockchains are immutable, which means that they are irreversible in the knowledge reached. [5]

2.1.1 Different Type of Blockchains

Some of the different kind of available blockchain are mentioned below:

2.1.1.1 Ethereum

Ethereum is an open-source, decentralized blockchain with smart contract features. Ether is the platform's native crypto-currency. After Bitcoin, it is the second biggest crypto-currency by market capitalization. The most widely used blockchain is Ethereum. [6]

2.1.1.2 Hyperledger Fabric

Hyperledger is an open source blockchain and related tools umbrella project started by the Linux Foundation in December 2015 and has obtained contributions from IBM, Intel and SAP Ariba to promote the collaborative development of distributed blockchain-based ledgers. [7]

2.1.1.3 R3 Corda

Corda is a private blockchain approved network that allows corporations to use smart contracts to trade directly and in strict privacy with each other. [8]

The unique privacy paradigm of Corda Organization helps organizations to interact safely and seamlessly while reducing transaction and record-keeping costs and streamlining company processes in an environment of permission-less blockchain networks in which all data is exchanged with all stakeholders. [8]

2.1.1.4 Ripple

Based on a distributed open-source protocol, Ripple was launched in 2012 and supports tokens representing fiat money, cryptocurrencies, commodities, or other value units such as frequent flying miles or smartphone minutes. Ripple promises to make "secure, instantly and nearly free global financial transactions of any size with no chargebacks." The ledger uses the native cryptocurrency known as XRP. [9]

2.1.1.5 Quorum

Quorum is a blockchain network for companies. It is a fork of the 'geth' public ethereum client with many changes in protocol level to support business needs. The primary aim of the Quorum project is to build an ethereum client enterprise that empowers enterprises to accept blockchain technology and benefit from it. Since Quorum is an open-source initiative, the platform's code base is open to everyone to inspect, fostering trust in the platform. Open-sourcing further enhances acceptance and draws developers from numerous sectors to engage in this platform's growth. [10]

2.1.1.6 Hyperledger Sawtooth

Under the Hyperledger umbrella, Hyperledger Sawtooth is an open-source project that acts as an enterprise-level blockchain framework used for the development and management of distributed ledger applications and networks, primarily for enterprise use. [11]

2.1.1.7 EOS

EOSIO is the leading open-source blockchain platform that enables transparency in transactions at the speed and scale needed to solve real-world challenges. We believe transparent and decentralized systems will keep users and builders in control and help architect integrity into our world. [12]

2.1.1.8 Hyperledger Iroha

Hyperledger Iroha is planned to be quick and easy to integrate into initiatives involving distributed ledger technologies in infrastructure or IoT. A simple, scalable, domain-driven C++ design, emphasis on customer application development and a modern, crash fault tolerant consensus algorithm, called YAC, are features of Hyperledger Iroha. [13]

2.1.1.9 Stellar

Stellar is an open source, decentralized digital currency protocol that allows cross-border transactions between any pair of currencies for fiat money transfers. A 5013 nonprofit, the Stellar Development Foundation, is promoting the Stellar protocol. [14]

2.1.2 Hyperledger Fabric

In 2015, the Hyperledger project was created by the Linux Foundation to support cross-industry blockchain technology. Instead of declaring a common blockchain standard, it supports a collective approach through a group mechanism to the implementation of blockchain technology, including intellectual property protections that enable transparent development and the acceptance over time of key specifications. [3]

One of the blockchain ventures within Hyperledger is Hyperledger Fabric. It has a database, uses smart contracts, and is a mechanism in which users control their transactions, much as other blockchain technology. [3]

If Hyperledger Fabric splits from any other structures of the blockchain, it is private and approved. "The users of a Hyperledger Fabric network join through a trusted Membership Service Provider (MSP) instead of an open permission less scheme that requires anonymous identities to participate in the network (requiring protocols such as "proof of work" to verify transactions and protect the network). [3]

Several pluggable options are also offered by Hyperledger Fabric. In different formats, ledger data can be stored, compromise structures can be switched in and out and various MSPs are supported. [3]

The option to create channels is also provided by Hyperledger Fabric, enabling a group of members to create a different transaction ledger. For networks where some participants might be rivals and do not want any exchange they make, this is a particularly significant choice, a special price they give to some participants and not others, for example, available to every participant. If a channel is created by two participants, only those participants have copies of the ledger for that channel, and no others. [3]

We are using Hyperledger Fabric due to the below mentioned features:

- Permissioned membership.
- Performance, scalability, and levels of trust.
- Data on a need-to-know basis.
- Rich queries over an immutable distributed ledger.
- Modular architecture supporting plug-in components.
- Protection of digital keys and sensitive data.

2.1.2.1 Shared Ledger

Hyperledger Fabric has two elements of a ledger subsystem: the world state and the transaction log. Each member in each Hyperledger Fabric network to which they belong has a copy of the ledger. [3]

At a given point in time, the world state portion defines the status of the ledger. It's the Ledger Database. All transactions which have resulted in the current value of the world state are documented in the transaction log component; it is the world state update history. The ledger, then is a mixture of the record of the world condition and the history of the transaction log. [3]

The ledger has a replaceable data store for the world state. By default, this is a LevelDB key-value store database. The transaction log does not need to be pluggable. It simply records the before and after values of the ledger database being used by the blockchain network. [3]

2.1.2.2 Smart Contract

A smart contract is a computer program or a transaction protocol which is intended to automatically execute, control or document legally relevant events and actions according to the terms of a contract or an agreement. Hyperledger Fabric smart contracts are written in chaincode and are invoked by an application external to the blockchain. In most instances, chaincode only communicates with the ledger's database part, the world state (for instance, querying it and not the transaction log. [3]

In many programming languages, chaincode can be implemented. Go and Node are currently being supported. [3]

2.1.2.3 Privacy

Participants of a Business-to-Business (B2B) network can be highly sensitive about how much data they exchange, based on the needs of a network. Privacy is not going to be a top priority for most networks. [3]

Hyperledger Fabric embraces networks where both anonymity (using channels) and relatively transparent networks are a primary operating prerequisite. [3]

2.1.2.4 Consensus

Transactions must be written to the ledger in the order in which they occur, while within the network they may be between various sets of participants. In order for this to work, it is important to determine the order of transactions and to enforce a mechanism for refusing bad transactions that have been wrongly (or maliciously) entered into the ledger. [3]

This is a field of computer science that has been extensively studied, and there are several approaches to do it, each with multiple trade-offs. [3]

For e.g., a method for file replicas to interact with each other may be given by PBFT (Practical Byzantine Fault Tolerance) to keep each copy accurate, even in the event of corruption. Alternatively, ordering exists in Bitcoin via a mechanism called mining, where competing computers race to solve a cryptographic problem that determines the order that subsequently builds on all systems. [3]

Hyperledger Fabric has been developed to allow network starters to select a process of consensus that best reflects the interactions between participants. There is a variety of needs, as with anonymity, from networks that are heavily organized in their interactions to those that are more peer-to-peer. [3]

2.2 Development Language

- **Language:** Bash, js, html, css, yaml, json.
- **Runtime:** nodejs.
- **Back-End Frameworks:** Express Js.
- **Front-End-Framework:** vuejs.

2.3 Database Platform

We are using NoSQL (MongoDB) for storing the web generated data like user profiles, configs, tenant information etc.

The reason of choosing NoSQL (MongoDB) instead of RDBMS are mentioned below:

Certificate data models can be changed in future as different types of certificates have different requirements so by using MongoDB, we can easily create flexible database schema according to the certificate needs and the second reason to choose MongoDB is horizontal scaling as we are providing services to multiple organization so we can easily scale them according to the needs which means add more servers when load increases, in case of any mishap if server crashes, data loss due to attack or any other reason at any side then any of the replica's node will handle the requests so user will not face any downtimes due to any problem at particular side. Another reason for using NoSQL is performance. NoSQL provides faster read, write time as we are developing multi-tenant application so the performance really matters, also sharding and partitioning (separating very large databases into smaller, faster, more easily managed parts). To achieve this with SQL databases requires additional coding or more complex configuration while NoSQL databases include these features out-of-box. Both SQL and NoSQL offer high availability and replication, but SQL requires more complex configuration while many NoSQL databases automatically include these features.

2.4 Blockchain Advancement in Certification

2.4.1 KMI, OU - UK

2.4.1.1 Salient Features, Functionality

- Badges, certificates and web reputation in the blockchain.

2.4.1.2 Drawback

- Does not support employers as an entity.
- Data is stored on public blockchain.
- The certificate is vulnerable to manipulation.
- No clear method of authenticity of parties.

2.4.2 UNIC

2.4.2.1 Salient Features, Functionality

- Resolve fake certificates.
- Tools available for the authenticity of the certificate.
- Good in integrity, privacy and ownership.

2.4.2.2 Drawback

- Requirements for an employer to verify the certificate is inadequate.
- A student cannot authorize the prospective employer to verify the certificate.
- No clear method of authenticity of parties.

2.4.3 MIT Media Lab

2.4.3.1 Salient Features, Functionality

- Offers more control to students.
- Uses digital keys.

2.4.3.2 Drawback

- Level of trust is low.
- The certificate can be accessed by everyone.
- No clear method of authenticity of parties.

2.4.4 Blockcert

2.4.4.1 Salient Features, Functionality

- Open standard platform.

2.4.4.2 Drawback

- No separate verification services.
- Vulnerable to spoofing attacks.

2.4.5 Smartcert

2.4.5.1 Salient Features, Functionality

- Resolves the problem of fake certificates.
- Student shares the hash with the employer.

2.4.5.2 Drawback

- Vulnerable to attacks.
- Need for basic information security measures.
- No clear method of authenticity of parties.

2.4.6 RecordsKeeper

2.4.6.1 Salient Features, Functionality

- Proof of authenticity in the certificate.
- The entire verification process is based on ownership.

2.4.6.2 Drawback

- Certificate tampering vulnerability.
- Participants can verify after obtaining ownership.

2.4.7 Accredible

2.4.7.1 Salient Features, Functionality

- This platform lets you automatically create, manage, and distribute digital certificates, open badges and blockchain credentials for training, events, course completion, membership, etc.

2.4.7.2 Drawback

- Better filter options for the Credentials.
- Analytics could be better developed.

3. Aim and Statement of Problem

Since it helps advance the profession, certification has made its way into nearly every sector. Certificates are now relevant in today's world, but there are a lot of issues, on the other hand. This requires a solution like **CERTIFIS** to solve those problems. The goal of **CERTIFIS** is to address the challenges encountered by certificate issuing entities such as creation, validation and verification of certificates.

The concerns that **CERTIFIS** are trying to tackle are as follows:

3.1 Verification

Verification is often one of the key problems that occurs in certificate verification. Sometimes there is no proper channel and when there is a way to verify a certificate it is a very time consuming and tiring process and lots of time the verification of certificate is not trustworthy too but **CERTIFIS** solve all of these problems.

3.2 Tampering

As we know data tampering is the act of purposely altering data through unauthorized networks (destroying, controlling, or editing). There are two states where data exists: in transit or at rest. Data could be intercepted and exploited in both scenarios in case of certificate either it is in paper form or whether it is maintained on database the act of tampering can happen in both ways so **CERTIFIS** is going to be one of the platforms which provides tamper free environment.

3.3 Data loss due to Mismanagement of Document

Imagine that we lose any certificate or any valuable information connected to it if our record is not adequately handled or backed up by a certificate-generating body, then it places the protection of our organization at risk by accident if our records are not shielded from the wrong hands, then it breaks regulatory laws and could place us and our company in legal trouble. While weak record protection is an evergreen problem, technological advancements and new management techniques will help us prevent crucial failures and **CERTIFIS** is one of the best options for this task.

3.4 Cyber Attack

It is one of the risky stuffs that many certificate-generating companies are unable to manage and struggle in this area, since we realize that cyber-attacks can cause infrastructure loss and security secret breaches. Valuable, confidential details such as personal details, credential ID, code, applicant addresses, phone number, etc. can be compromised. Cyber-attacks interrupt computer networks or paralyze processes due to security vulnerabilities, rendering data inaccessible, but **CERTIFIS** would be the first certificate agency to aim as best as possible to fix this problem.

3.5 Server down

As we know, several certificate producing organizations are running on centralized server in case of server down as they are highly reliant on network communication, system will fail if the nodes loss connectivity as there is only one central node there is no graceful deterioration of system abrupt failure no one can allow transaction, data backup is also one of the major

problems if the system abrupt failure occur you lose the data straight away but as we know **CERTIFIS** run on decentralized architecture so incase if any organization's server down or data lost due to any mishap so the verifiability of certificate is still possible due to distributed and decentralized architecture. So, in this matter **CERTIFIS** is still on top.

3.6 Security Loopholes

Security is one of the challenges in which many certificate-generating organizations still suffer because virtually every entity has security loopholes, but we know how critical security is, that is why **CERTIFIS** seek to have as much as possible a safe atmosphere due to decentralized architecture.

3.7 Data Redundancy

Data redundancy is one of the problems that arise on a regular basis when many organizations are unable to accommodate vast volumes of data, so they produce data redundancy due to their mismanagement and there is no check and balance and no proper way to plan and handle this whole method but **CERTIFIS** have upper hand their too because as defines earlier we are using NoSQL database so the main thing in NoSQL is that we de-normalize data to get faster read write times in simple words renormalization is increasing the data redundancy by storing related data in a single document. We have clear fully analyze the dataset and tried to denormalized only the data which is required frequently by the client and normalized the data into multiple collections which was not needed frequently by the client.

3.8 Data Backup

Data backup is one of the main issues that is faced by most of the organizations who are using central servers for storing their data. The organizations that are using central servers for storing the information of their certificate then in case any attacker attacks their server and they don't have proper backup of their data then they might lose their data also if the organizations have proper backup but the attacker also harm their backups so in this case organizations can lose their data completely. To resolve this issue **CERTIFIS** is using blockchain for storing the certificate data of organizations. By using blockchain **CERTIFIS** can solve backup problem because in case attacker attacks on any of the node on blockchain then data of the organization will be saved on many other nodes of the blockchain and this will make sure the data availability.

4. Hardware, Software Analysis and Requirements

There is no particular need of hardware, although there exist some similar software and the comparison of those are mentioned below:

4.1 Comparison of Different Blockchain Technologies

Table 4.1 – Blockchain Technologies' Comparison

	Ethereum	Hyperledger Fabric	R3 Corda	Ripple	Quorum	Hyperledger Sawtooth	EOS	Hyperledger Iroha	Stellar
<i>Industry focus</i>	Cross-Industry	Cross-Industry	Financial Services	Financial Services	Cross-Industry	Cross-Industry	Cross-Industry	Cross-Industry	Financial Services
<i>Ledger Type</i>	Permission less	Permissioned	Permissioned	Permissioned	Permissioned	Permissioned	Permissioned	Permissioned	Both Public & Private
<i>Consensus Algorithm</i>	Proof of Work	Pluggable Framework	Pluggable Framework	Probabilistic Voting	Majority Voting	Pluggable Framework	Delegated Proof-of-Stake	Chain-based Byzantine Fault Tolerant	Stellar Consensus Protocol
<i>Smart Contract</i>	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes
<i>Governance</i>	Ethereum Developers	Linux Foundation	R3 Consortium	Ripple Labs	Ethereum Developers and JP Morgan Chase	Linux Foundation	EOSIO Core Arbitration Forum (ECAF)	Linux Foundation	Stellar Development Foundation
<i>Cryptocurrency</i>	Ether	No	No	Ripple (XRP)	Ether	-	-	-	-
<i>Language for Smart Contract</i>	Solidity	JavaScript, Java, Go	Kotlin, Java	C++	Solidity	Go, Rust, Python, JavaScript	C++	No Smart Contracts	JavaScript, Java, Go, Ruby, Python, C#

4.2 Comparison of Different NoSQL Databases

Table 4.2 – NoSQL Databases' Comparison

	MongoDB	Cassandra	CouchDB
Type	Document based	Flexible wide columns	Document
Query Language	JavaScript	CQL	Http API
Multi-Record Acid Transactions	Yes	No	No
Replication Modes	Master/ slave replication	Master/ slave replication	Multi master replication
Partitioning Methods	Sharding	Sharding	Sharding
Server-Side Scripts	JavaScript	No	View functions in JavaScript
Operation	On-premises or in the cloud	On-premises or in the cloud	On-premises or in the cloud
Map Reduce	Yes	Yes	Yes
Protocol	TCP/IP	Thrift	Http/Rest

4.3 Comparison of Different Web Frameworks

Table 4.3 - Web Frameworks' Comparison

	Express.js	ASP.NET	Django	Laravel
<i>Category</i>	Web application framework	Web application framework	Web application framework	Web application framework
<i>Language</i>	JS	C#	Python	PHP
<i>Platform</i>	Cross Platform	Only ASP.NET core is cross platform	Cross Platform	Cross Platform
<i>Processing Model</i>	Asynchronous, Single-thread non-blocking, I/O model	Synchronous, multi-threaded, blocking I/O	Synchronous, multi-threaded, blocking I/O	Synchronous, multi-threaded, blocking I/O
<i>Leading Firms Using these Languages in their Projects</i>	NASA, Walmart, eBay, Uber	AccuWeather, Dell, Stack Overflow	Instagram, Pinterest, Coursera	9GAG, BBC, Tour Radar
<i>Templating Language</i>	EJS, jade, Pug, etc.	Razor	Jinja	Twig, Smarty, Mustache, etc.
<i>Package Manager</i>	Npm	Nuget	PyPI	Packagist
<i>Primarily Classified</i>	Microframeworks (Backend)	Frameworks (Full Stack)	Frameworks (Full Stack)	Frameworks (Full Stack)

4.4 Comparison of Available Software in terms of Salient Features, Functionality and Shortcomings

Table 4.4 - Existing Solutions and their Shortcomings

Institution/ Solution	Salient Features, Functionality	Shortcomings in Feature/ Functionality
KMI, OU - UK	<ul style="list-style-type: none"> Badges, certificates and web reputation in the block chain. 	<ul style="list-style-type: none"> Does not support employers as an entity. Data is stored on public blockchain. The certificate is vulnerable to manipulation. No clear method of authenticity of parties.
UNIC	<ul style="list-style-type: none"> Resolve fake certificates. Tools available for the authenticity of the certificate. Good in integrity, privacy and ownership. 	<ul style="list-style-type: none"> Requirements for an employer to verify the certificate is inadequate. A student cannot authorize the prospective employer to verify the certificate. No clear method of authenticity of parties.
MIT Media Lab	<ul style="list-style-type: none"> Offers more control to students. Uses digital keys. 	<ul style="list-style-type: none"> Level of trust is low. The certificate can be accessed by everyone. No clear method of authenticity of parties.
Blockcert	<ul style="list-style-type: none"> Open standard platform. 	<ul style="list-style-type: none"> No separate verification services. Vulnerable to spoofing attacks.
Smartcert	<ul style="list-style-type: none"> Resolves the problem of fake certificates. Student shares the hash with the employer. 	<ul style="list-style-type: none"> Vulnerable to attacks. Need for basic information security measures. No clear method of authenticity of parties.
RecordsKeeper	<ul style="list-style-type: none"> Proof of authenticity in the certificate. The entire verification process is based on ownership. 	<ul style="list-style-type: none"> Certificate tampering vulnerability. Participants can verify after obtaining ownership.
Accredible	<ul style="list-style-type: none"> This platform lets you automatically create, manage, and distribute digital certificates, open badges and blockchain credentials for training, events, course completion, membership, etc. 	<ul style="list-style-type: none"> Better filter options for the Credentials. Analytics could be better developed.

4.5 Comparison of Available Software in terms of their Security Themes

Table 4.5 - Certification Solutions Maps with Requirements Theme

Essential Security Themes to Fulfill Certificate Verification in Blockchains

Institution/ Solution	Salient Features, Functionality	<i>Authentication</i>	<i>Authorization</i>	<i>Confidentiality</i>	<i>Ownership</i>	<i>Privacy</i>
		No	No	No	No	Yes
<i>KMI, OU - UK</i>	• Badges, certificates and web reputation in the block chain. • Resolve fake certificates.	No	No	No	No	Yes
<i>UNIC</i>	• Tools available for the authenticity of the certificate. • Good in integrity, privacy and ownership.	No	No	No	Yes	Yes
<i>MIT Media Lab</i>	• Offers more control to students. • Uses digital keys.	No	Yes	No	Yes	Yes
<i>Blockcert</i>	• Open standard platform.	No	No	No	No	No
<i>Smartcert</i>	• Resolves the problem of fake certificates. • Student shares the hash with the employer.	No	Yes	No	Shared	No
<i>RecordsKeeper</i>	• Proof of authenticity in the certificate. • The entire verification process is based on ownership.	Yes	No	No	Shared	No
<i>Accredible</i>	• This platform lets you automatically create, manage, and distribute digital certificates, open badges and blockchain credentials for training, events, course completion, membership, etc.	Yes	Yes	Yes	Shared	No

4.5.1 Certifis

4.5.1.1 Salient Features, Functionality

The Blockchain based e-certification project will contain the following features:

- Creation of tamper free & verifiable certificates using blockchain technology.
- Any organization can join a blockchain network by deploying nodes.
- Organizations hosting the blockchain network can provide services to others.
- No certificate loss or any data loss due to any mishap.

4.6 Diagrams

4.6.1 System Diagram

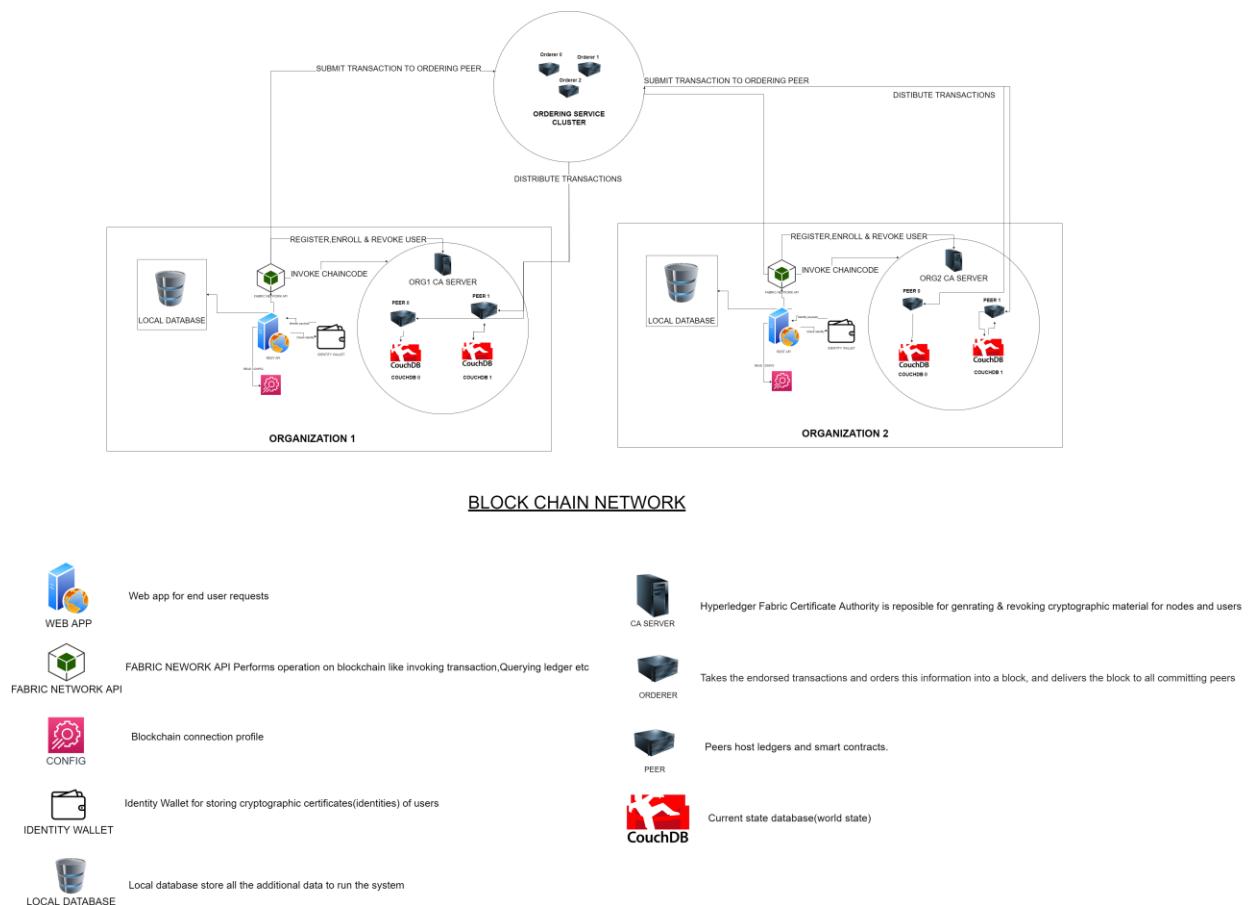


Figure 4.1 - System Diagram

The system includes the following:

- **Websvserver:** Web app which handles the end user requests.
- **Local Database:** Store web generated data like user profiles, certificates data, configs for each tenant
- **Peers:** It hosts the ledger and hold the smart contracts.
- **Ordering Peers:** Take the endorsed transaction and orders this information into a block and deliver the block to all committing peers.
- **World State (CouchDB):** It stores current state of the ledger.
- **Fabric CA Server:** Fabric CA is responsible for generating and revoking cryptographic material for nodes and users.

- **Fabric SDK (Fabric Network API):** A higher-level API for interacting with smart contracts and is the recommended API for client applications to interact with smart contracts deployed to a Hyperledger Fabric blockchain network.
- **Identity Wallet:** It stores the cryptographic identities of users.

4.6.1.1 System Workflow

4.6.1.1.1 Common Operations Flow

The flow is shown in figure 4.1 describes when users send request to the webserver and the web server responds to the requests all the data of organization stored on the local DB may be single instance or clustered.

The common operation includes:

- Manage Organizations.
- Manage User.
- Manage Batches.
- Manage Publications.

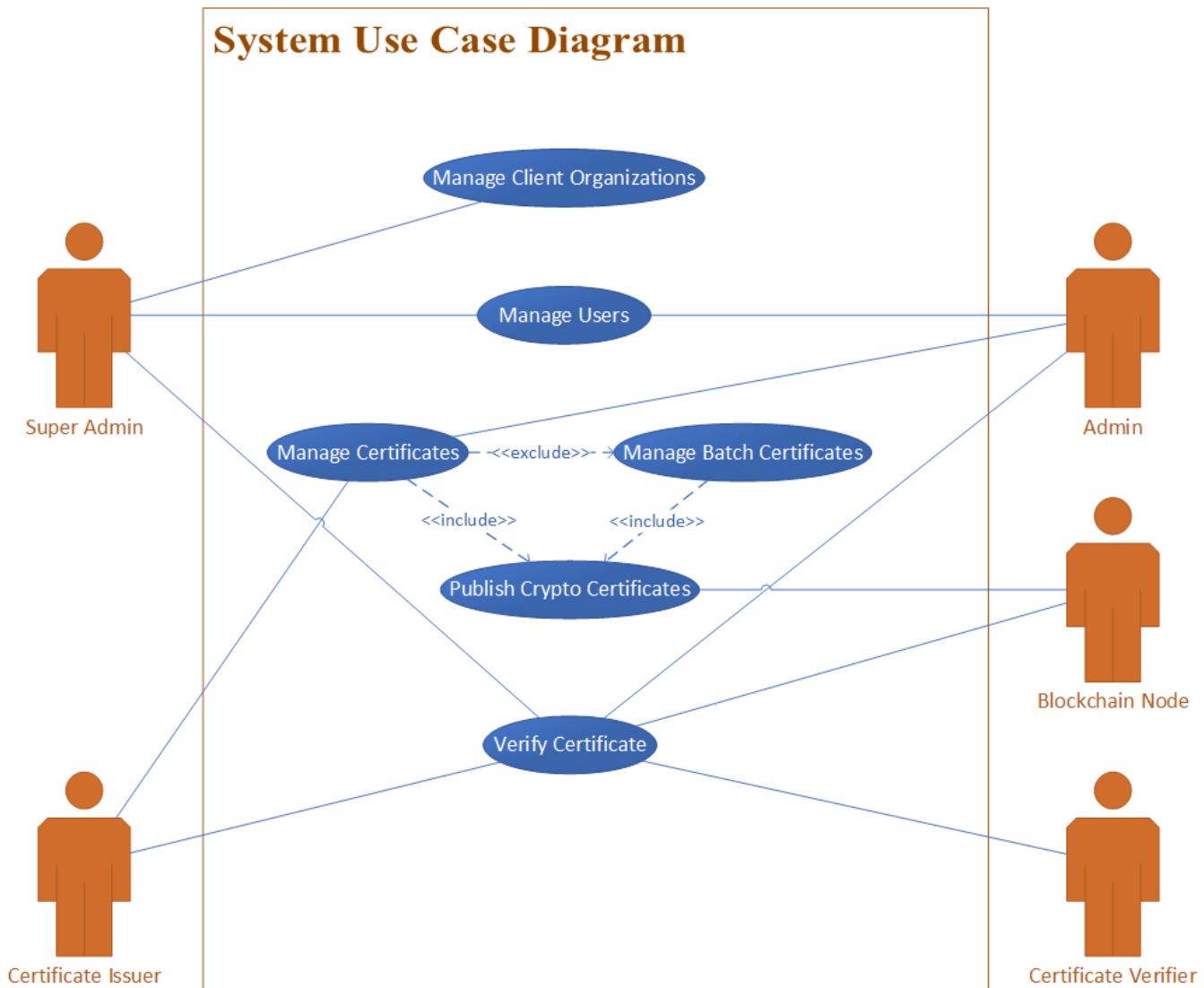
4.6.1.1.2 User Creation Flow

The flow is shown in figure 4.1 describes when users submit the registration request to the webserver now webserver store all the data in its local database and send the request to fabric CA server for cryptographic identity of the user so that the user perform operations on the blockchain.

4.6.1.1.3 Publishing to Blockchain Flow

The flow is shown in figure 4.1 describes when the user send the certificate publishing request to the server, then for publishing the data on the blockchain web server check the identity wallet for the user's identity of blockchain to authenticate it and now using the fabric SDK it submits the request to the blockchain the fabric SDK read the connection profile of blockchain and run the discovery service to find the endorser after discovery it will send the endorsing proposal to the endorsing peer after successful response it submits that signed proposal to the ordering service cluster the ordering service orders the transaction and send to the peer organization where the transaction is validated and committed to the ledger.

4.6.2 Actor Use Case Diagram



Use Case Details

Manage Client Organization: Register Organization, Enable/Disable Client Organization, Update User Limit, Update Organization's Certificate Limit, View Count History, View Single Certificate, View Batches, View Batch Certificates.

Manage User: Register Users, Enable/Disable Users, Reset Password.

Manage Certificates: Create Certificates, Edit Certificates, Delete Certificates, View Certificates, Publish Certificates.

Manage Batch Certificates: Create Batches, Edit Batches, Delete Batches, View Batches, Publish Batches.

Publish Crypto Certificates: Publish Batches, Publish Single Certificates.

Verify Certificates: Verify Certificates, Download Certificates, Recover Certificates.

Figure 4.2 - Actor Use Case Diagram

The figure 4.2 defines all the action that are performed by every actor. There are five actors:

- Super Admin.
- Admin.
- Certificate Issuer.
- Certificate Verifier.
- Blockchain Node.

4.6.2.1 Super Admin

The super admin will be able to perform the following task:

Add Organization, Enable/Disable Client Organization, Assign User Limit, Assign Organization's Certificate Limit, View Count History, View Certificate Balance, View Client Organization's Single Certificate, View Client Organization's Batches, View Client Organization's Batch Certificates, Register Users, Enable/Disable Users, Reset Password, Verify Certificates, Download Certificates, Recover Certificates.

4.6.2.2 Admin

The admin will be able to perform the following task:

Register Users, Enable/Disable Users, Reset Password, Create Certificates, Edit Certificates, Delete Certificates, View Certificate, Create Batches, Edit Batches, Delete Batches, View Batches, Publish Batches, Publish Single Certificates, Verify Certificates, Download Certificates, Recover Certificates.

4.6.2.3 Certificate Issuer

The certificate issuer will be able to perform the following task:

Create Certificates, Edit Certificates, Delete Certificates, View Certificate, Create Batches, Edit Batches, Delete Batches, View Batches, Publish Batches, Publish Single Certificates, Verify Certificates, Download Certificates, Recover Certificates.

4.6.2.4 Certificate Verifier

The certificate verifier will be able to perform the following task:

Verify Certificates, Download Certificates, Recover Certificates.

4.6.2.5 Blockchain Node

The blockchain node is the system actor and this will perform the following task:

Publish Batches, Publish Single Certificates, Verify Certificates, Download Certificates, Recover Certificates.

4.6.3 Activity Diagrams

Activity Diagram: Create Single Certificates
Users: Admin/Certificate Issuer

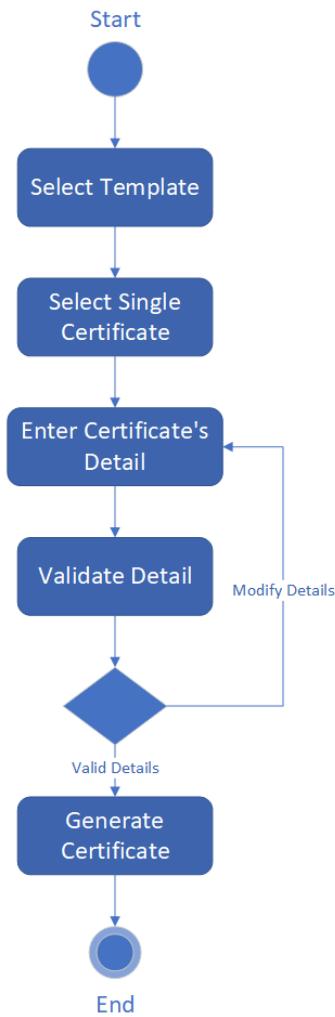


Figure 4.3 - Create Single Certificate Activity Diagram

If the user wants to create a single certificate this will provide him/her the feature of creating single certificates. By choosing a certificate template he/she will be able to create single certificates. The figure 4.3 shows the process of creating single certificates, first user need to go to the create option which is available on the navbar the system will render all the available certificate templates, user will select any of the template then he/she will be able to view or fill as per choice in case if user fills the selected template then system will ask user either they want to create single certificate or they want to creates batch for creating single certificate user needs to selects single certificate then he/she will be allowed to add all the required details of the certificate and when the user clicks on create button then system will validate the formats of all the inputs given by user if the format is correct then the system will generate the certificate else the system will prompt error for correcting the format of the inputted fields.

Activity Diagram: Create Batches/ Batch Certificates
Users: Admin/ Certificate Issuer

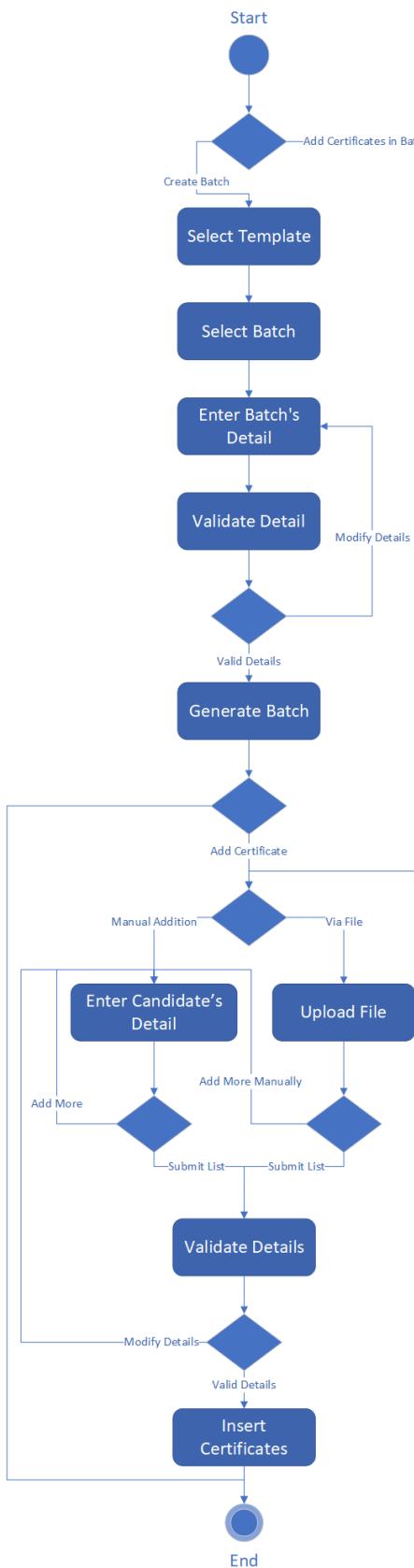


Figure 4.4 - Create Batches/ Batch Certificates Activity Diagram

If the user wants to create batches or wants to add certificates in the batches this will provide him/her the feature of creating batches and add certificates in the batches. In case user wants to create batch so by choosing a certificate template he/she will be able to create batches and if the user wants to add certificates in them then user can also add certificates in them. The figure 4.4 shows the process of creating batches and adding certificates in the batches, first user need to go to the create option which is available on the navbar the system will render all the available certificate templates, user will select any of the template then he/she will view or fill as per choice in case if user fill the selected template then system will ask user either they want to create single certificate or they want to creates batch for creating batch user needs to selects batch then he/she will be allowed to add all the required details of the batch and when the user clicks on create button then system will validate the formats of all the inputs given by user if the format is correct then the system will generate the batch else the system will prompt error for correcting the format of the inputted fields. Once the batch has been created the system will redirect user to add certificates page if the user wants to add certificates in the batch then user will be able to add certificates in it if user don't want to add certificates then this will end the process. If the user doesn't want to create batch and just want to add certificates in the batch this will provide him/her the feature of adding certificates. By choosing the batch he/she will be able to add certificates in that batch. The figure 4.4 also shows the process of adding certificates in the batch, first user need to go to the certificate option which is available on the navbar then system will display two options single certificates and batches for adding certificates in the batches user needs to select batches option so system will display all the unpublished batches to the user then user will select any of the batch and will go to batch detail and when the user clicks on add button then the system will render the page to add certificates, now user will be having multiple options of adding certificate one is manual addition and the second is add certificates by file if the user wants to add manually then he/she can add certificates manually and if user wants to add certificates using file then he/she can upload the excel file then the system will retrieve certificates data from the file also if the user wants to add more certificates after uploading the file then he/she can add them manually once the user clicks on submit button then the system will validate the detail of all the certificate if all certificates' detail is correct then the system will add those certificates to the batch if any of those certificates' detail is invalid so system will prompt an error to the user for correcting the detail.

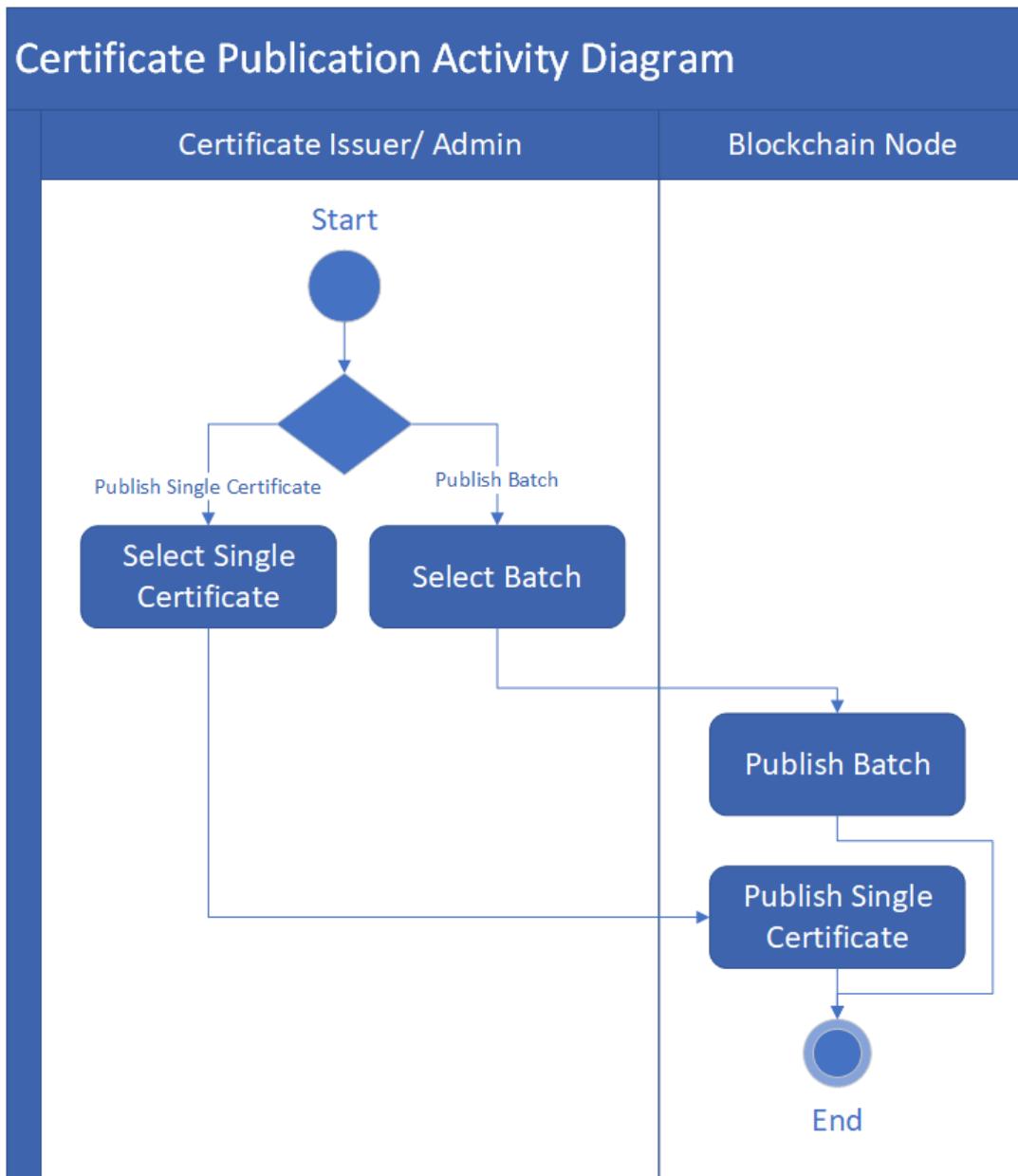


Figure 4.5 - Certificate Publication Activity Diagram

If the user wants to publish single certificates or batches to the blockchain the this will allow users to publish single certificates or batches to the blockchain after publishing, all the certificates will be verifiable. The figure 4.5 shows the process of publishing single certificates or batches to the blockchain, first user need to go to the certificate option which is available on the navbar then system will display two options single certificates and batches, incase if user selects single certificates option so system will display all the unpublished single certificates to the user then user will select any of the single certificate and click on publish button to publish that single certificate to the blockchain and if user selects batches option so system will display all the unpublished batches to the user then user will select any of the batch and click on publish button to publish that batch to the blockchain.

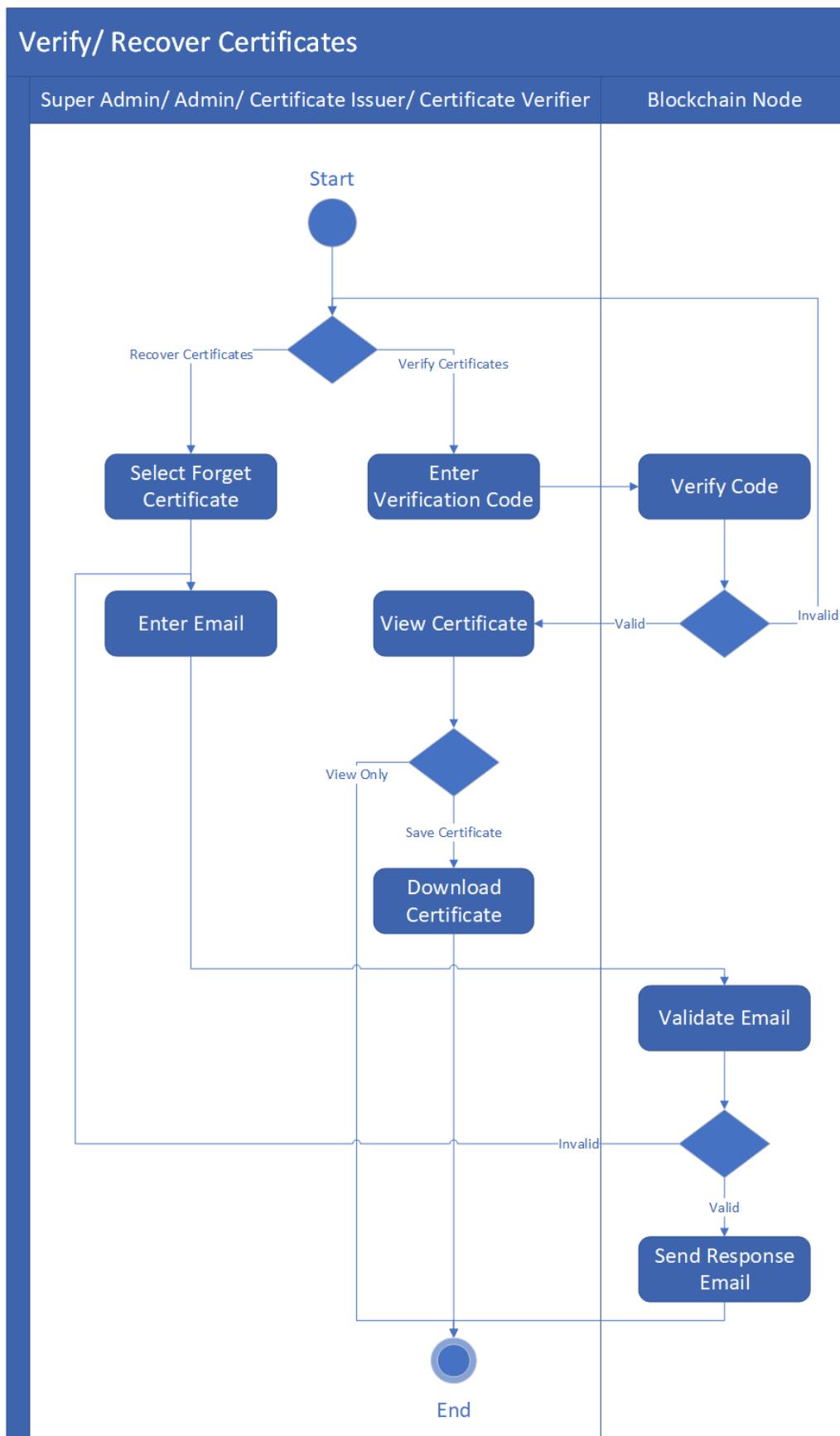


Figure 4.6 - Verify/ Recover Certificates Activity Diagram

If the user wants to verify certificates this will allow users to verify the certificates that have been published to blockchain. The figure 4.6 shows the process of verifying certificates either in batch or single certificates, first user needs to go to the verify option which is available on the navbar then system will render verification page if the user wants to verify the certificate he/she needs to enter the verification code if the code is valid then certificates will be displayed to the user and if user wants to save the certificates then the user can download that certificate and in case if the code is invalid then the system will prompt error of invalid code to the user also if user forgets the certificate code then user can recover those certificates by clicking on forget certificate which will render the forget certificate page to recover certificates now user needs to enter his/her email address, once user submit the request, the system will validate the email if the email is valid then all the certificates against that email will be sent to that email address.

Activity Diagram: Register Organization
Users: Super Admin

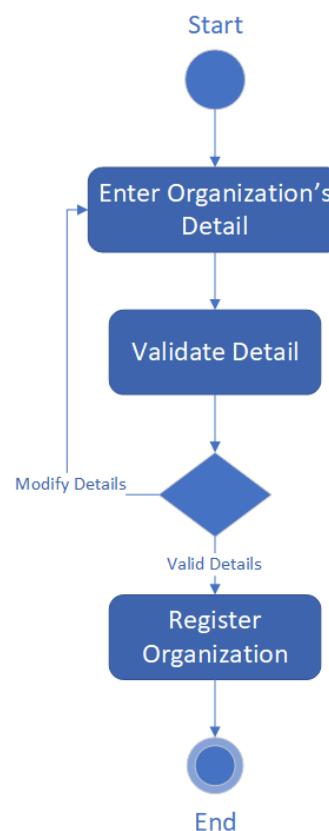


Figure 4.7 - Register Organization Activity Diagram

If the user wants to register new organization. So, figure 4.7 shows the process of registering organization, first user needs to go to the organization option which is available on the navbar then system will render organization page if the user wants to register organization user need to enter the detail of organization when the user submits the detail of the organization the system will validate the detail of the organization if the detail is valid then the system will register the organization if the detail is invalid then the system will prompt user to correct the detail.

4.7 Requirements

The main goal requirements of problem were given by our supervisor and some requirements related to user experience are self-defined also we collaboratively worked with our supervisor to mine many of the core and supportive requirements by prototyping, also held multiple meeting in which we brainstorm many scenarios and analyze the mockups to make the requirements more crystal clear.

4.7.1 Functional Requirements

- As a Super Admin, I want to create batch certificates so that I can issue the same certificates to as many candidates as required.
- As a Super Admin, I want to create single certificates so that I can issue certificates to single candidates.
- As a Super Admin, I want add user functionality so that I can add users in my organization and in client organizations.
- As a Super Admin, I want to enable/disable client organization so that I can disable any client organization due to any reason like policy violation, service unsubscription, etc.
- As a Super Admin, I want to enable/disable users so that I can disable any users of my organization and any users and admin of client Organizations due to any reason like policy violation, left organization, etc.
- As a Super Admin, I want to publish certificates so that I can publish single and batch certificates to the block chain.
- As a Super Admin, I want to register client organization so that our organization provides services to other organizations.
- As a Super Admin, I want certificate count assign feature so that I can set the certificate issuing limit of our client organization.
- As a Super Admin, I want forget password link generation feature so that I can generate the link by which users of my organization and users and admin of client organizations can reset their password.
- As a Super Admin, I want to verify certificates so that I can verify any certificates.
- As a Super Admin, I want forget certificates functionality so that I can recover all the certificates against a particular email.
- As a Super Admin, I want to view single certificates so that I can check all the single certificates issued by my organization
- As a Super Admin, I want to view batch certificates so that I can check all the batches created by my organization and certificates in those batches.
- As a Super Admin, I want to view client organizations' single certificates so that I can check all the single certificates issued by client organizations.
- As a Super Admin, I want to view client organizations' batch certificates so that I can check all the batches created by client organizations and certificates in those batches.
- As a Super Admin, I want to service utility report so that I can check usage of my organization as well as client organization.
- As an Admin, I want to create single certificates so that I can issue certificates to single candidates.
- As an Admin, I want to create batch certificates so that I can issue the same certificates to as many candidates as required.
- As an Admin, I want add user functionality so that I can add users in my organization.
- As an Admin, I want to enable/disable users so that I can disable any users of my organization due to any reason like policy violation, left organization, etc.

- As an Admin, I want to publish certificates so that I can publish single and batch certificates to the block chain.
- As an Admin, I want forget password link generation feature so that I can generate the link by which users of my organization can reset their password.
- As an Admin, I want to verify certificates so that I can verify any certificates.
- As an Admin, I want forget certificates feature so that I can recover all the certificates against a particular email.
- As an Admin, I want to view single certificates so that I can check all the single certificates issued by my organization.
- As an Admin, I want to view batch certificates so that I can check all the batches created by my organization and certificates in those batches.
- As an Admin, I want to service utility report so that I can check usage of my organization.
- As an Issuer, I want to create single certificates so that I can issue certificates to single candidates.
- As an Issuer, I want to create batch certificates so that I can issue the same certificates to as many candidates as required.
- As an Issuer, I want to publish certificates so that I can publish single and batch certificates to the block chain.
- As an Issuer, I want to verify certificates so that I can verify any certificates.
- As an Issuer, I want forget certificates feature so that I can recover all the certificates against a particular email.
- As an Issuer, I want to view single certificates so that I can check all the single certificates issued by organization.
- As an Issuer, I want to view batch certificates so that I can check all the batches created by organization and certificates in those batches.
- As a Verifier, I want to verify certificates so that I can verify any certificates.
- As a Verifier, I want forget certificates feature so that I can recover all the certificates against a particular email.
- As a Verifier, I want to download certificate so that I can use this certificate.

4.7.2 Non-Functional Requirement

4.7.2.1 Reliability

The **overall system** should handle any interruption due to internet failure or any other mishaps.

4.7.2.2 Authorization

The authorization should be imposed on **overall system** operations except the **verification process** which is open to the public.

4.7.2.3 File Integrity

All static content uploaded by user while **creating certificates and batches** should be stored in a manner so that when the user requests his file the system should return the correct files.

4.7.2.4 Audit Trail

Audit trail should be maintained for check and balance for **certificate creation, organization registration, certificate's count assign, publish single certificates and publish batches** so that we can track when and who perform these operations.

4.7.2.5 Continuity of Processing

The log should be maintained while **publishing single certificates and publishing batches** for making the system able to continue publishing where it stops in case of any interruption.

4.7.2.6 Service Level

The response time of **publishing single certificates and publishing batches** should be as minimum as possible and the process which takes long time should be pushed to the background processes and should not make user to wait for the response ones the process finished it should notify the user when he/she is available.

4.7.2.7 Access Control

The level of access should be clearly defined for **overall system**; the user can access only that resources for which they are eligible.

4.7.2.8 Methodology

The **overall system's** requirement, design, construction, documentation, testing, integration and maintenance should be according to the development, design, programming, test, installation and maintenance methodology.

4.7.2.9 Correctness

The features such as **create certificates, create batches, add batch certificates, register organization and register users'** requests should be validated for correctness before proceeding to the further steps.

4.7.2.10 Ease of Use

The features such as **create certificates, create batches, add batch certificates, publish single certificates and publish batches** should be simple and easy to use.

4.7.2.11 Maintainable

The **overall system** should be highly decoupled so that the maintenance will not cause the whole system to go down. The only component under maintenance could be unavailable instead of the whole system.

4.7.2.12 Portable

The **overall system** should not face compatibility issues when deployed to different environments; also, the system component should follow the standard protocol so it can easily integrate or communicate with third party components or services.

4.7.2.13 Coupling

The **verification services** should be highly decoupled so it can easily be integrated in third party applications.

4.7.2.14 Performance

The features such as **publish single certificates and publish batches** should be able to accept faster requests and provide faster responses and capable of providing faster services under heavy load.

4.7.2.15 Ease of Operation

The features such as **create certificates, create batches, add batch certificates, publish single certificates and publish batches** should be developed according to the HCI standards so different type of users can use it easily.

5. Software design and modeling

We are using three types of architecture in our application which includes service, decentralized and distributed architectures. Our backend is based on service-based architecture which means that it is highly decoupled from the front-end application. Our client application interacts with the backend through APIs for retrieving and sending data. The purpose of using services-based architecture is that we will not only be restricted with web applications. So that in future we will be able to extend our application to multiple platforms like iOS, Android, Desktop, etc. also if we want to sell our services to the third party then we will be able to sell those services. Now if we talk about complete system then the overall architecture of our system is distributed because the frontend, backend, database, message broker & blockchain (decentralized distributed system) are deployed on different server even the consumer and producer of message broker can run on different servers. This shows that our application is totally distributed. Now let's discuss some failure cases, in case our frontend application crashed and stopped working due to any reason then our backend will still remain unaffected and still able to serve the clients access the services from different channels, in case any of the backend service fails than other service will not get affected and will continue to run normally, in case of complete backend failure our front-end application still be able to communicate with user about the issues.

5.1 What is Distributed Architecture?

Components are presented on various platforms in a distributed architecture, and multiple components may collaborate over a communication network to accomplish a certain purpose or goal. [15]

- Information processing is spread through many different machines in this architecture, rather than being limited to a single processor. [15]
- The client-server architecture, which is the basis for multi-tier architectures, may be used to illustrate a distributed system; other solutions include broker architectures such as CORBA and Service-Oriented Architecture (SOA). [15]
- .NET, J2EE, CORBA, .NET Web services, AXIS Java Web services, and Globus Grid services are some of the application platforms that support distributed architectures. [15]

5.2 Benefits of Distributed Architecture

- **Resource sharing** – Hardware and software services are shared. [15]
- **Openness** – The ability to use hardware and applications from a variety of vendors. [15]
- **Concurrency** – To improve efficiency, use concurrent processing. [15]
- **Scalability** – By adding additional resources, we were able to increase throughput. [15]
- **Fault tolerance** – Since an error has arisen, the right to continue operating. [15]

5.3 Why are we Using Distributed Architecture?

The aim of using Distributed Architecture is to make our structure as self-contained as possible. Since we use a distributed architecture, our frontend, backend, and database and message broker all function separately. If our frontend application goes down for whatever reason, our backend will be unaffected; but, if one of the backend servers goes down, our frontend application will be unaffected due to other backend server(s).

5.4 Web Interfaces

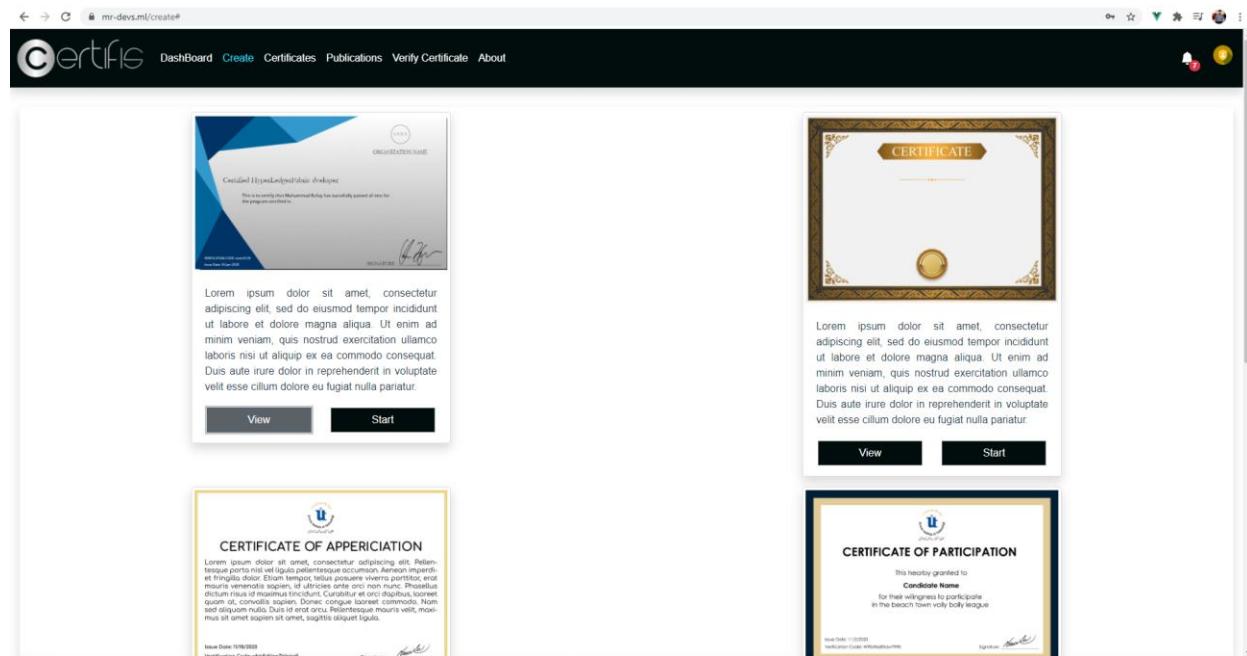


Figure 5.1 - Template Selection Web Interface

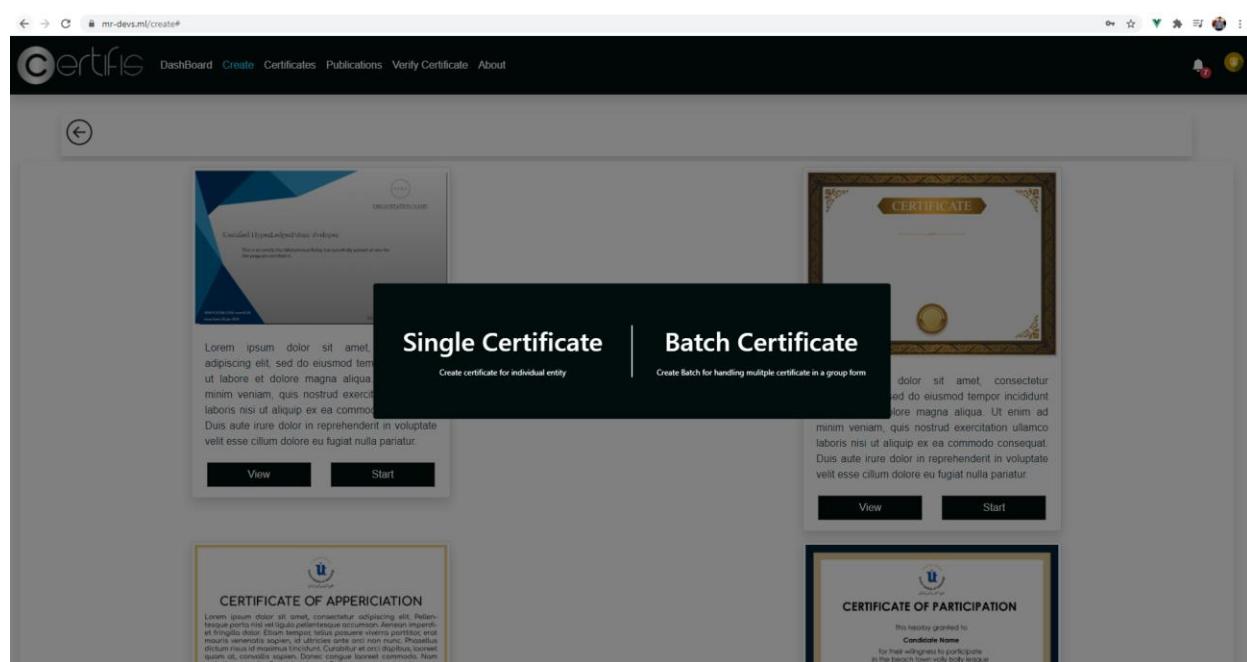


Figure 5.2 - Single or Batch Selection Web Interface

Certificate Title
It is to certify that "Name" "description goes here"

Signature _____

CERTIFICATE INFORMATION

- * Title
- * Candidate Name
- * Candidate Email
- Instructor Name
- Expiry Date mm/dd/yyyy
- * Description

UPLOAD LOGO
UPLOAD SIGNATURE

Figure 5.3 - Create Single Certificate Web Interface

Certificate Title
It is to certify that "Name" "description goes here"

Signature _____

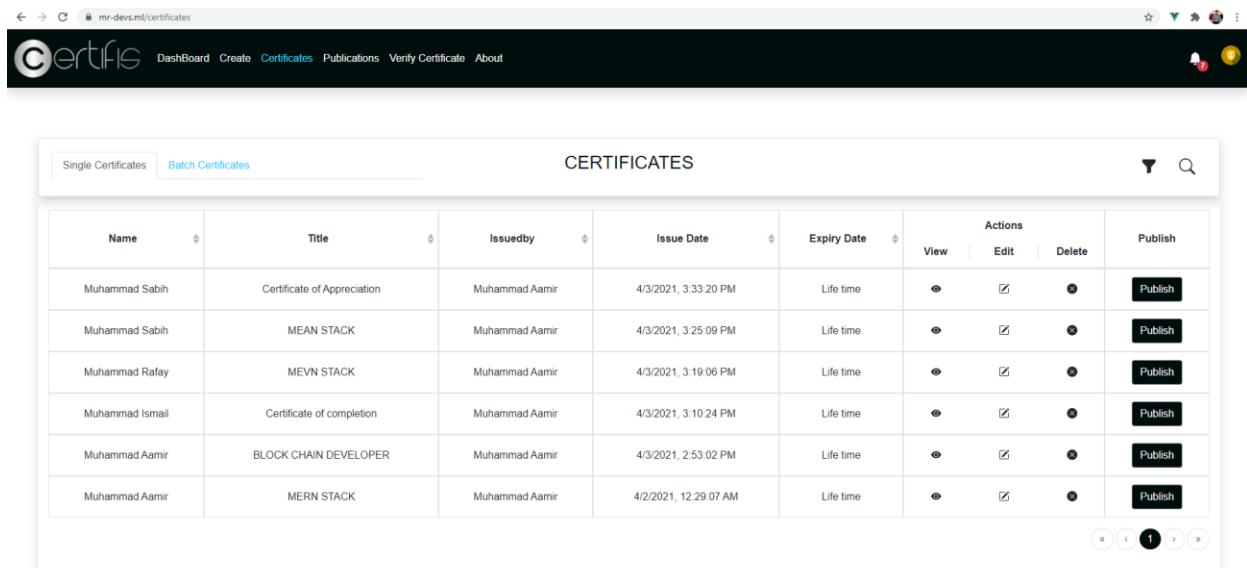
BATCH INFORMATION

- * Title
- * Batch Name
- Instructor Name
- Expiry Date mm/dd/yyyy
- * Description

Note: for life time expiry left empty

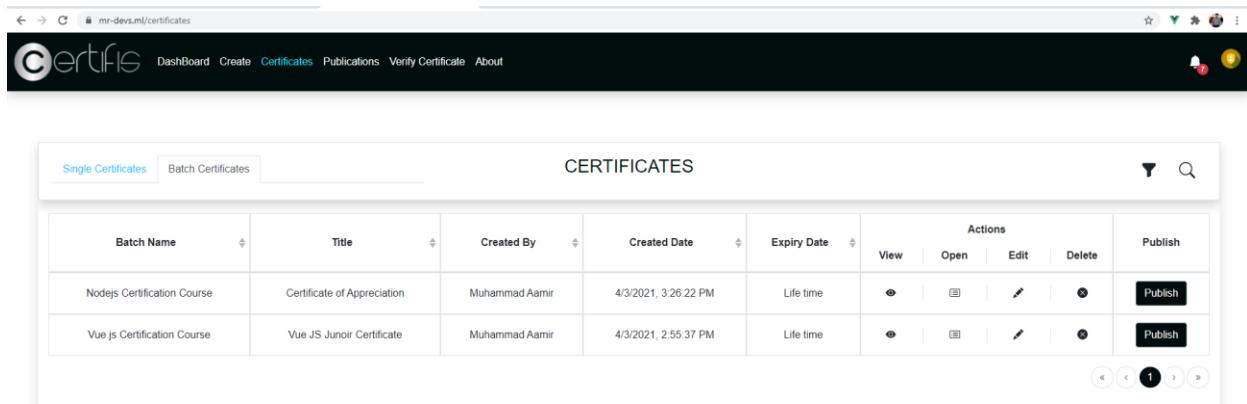
UPLOAD LOGO
UPLOAD SIGNATURE
CREATE

Figure 5.4 - Create Batch Web Interface



The screenshot shows a web application interface titled "CERTIFICATES". At the top, there are two tabs: "Single Certificates" (selected) and "Batch Certificates". Below the tabs is a table with the following columns: Name, Title, Issuedby, Issue Date, Expiry Date, Actions, and Publish. The table contains six rows of data. Each row includes a "View" button, an "Edit" button, a "Delete" button, and a "Publish" button.

Name	Title	Issuedby	Issue Date	Expiry Date	Actions	Publish	
					View	Edit	Delete
Muhammad Sabih	Certificate of Appreciation	Muhammad Aamir	4/3/2021, 3:33:20 PM	Life time			
Muhammad Sabih	MEAN STACK	Muhammad Aamir	4/3/2021, 3:25:09 PM	Life time			
Muhammad Rafay	MEVN STACK	Muhammad Aamir	4/3/2021, 3:19:06 PM	Life time			
Muhammad Ismail	Certificate of completion	Muhammad Aamir	4/3/2021, 3:10:24 PM	Life time			
Muhammad Aamir	BLOCK CHAIN DEVELOPER	Muhammad Aamir	4/3/2021, 2:53:02 PM	Life time			
Muhammad Aamir	MERN STACK	Muhammad Aamir	4/2/2021, 12:29:07 AM	Life time			

Figure 5.5 - Single Certificate List Web Interface


The screenshot shows a web application interface titled "CERTIFICATES". At the top, there are two tabs: "Single Certificates" and "Batch Certificates" (selected). Below the tabs is a table with the following columns: Batch Name, Title, Created By, Created Date, Expiry Date, Actions, and Publish. The table contains two rows of data. Each row includes a "View" button, an "Open" button, an "Edit" button, a "Delete" button, and a "Publish" button.

Batch Name	Title	Created By	Created Date	Expiry Date	Actions	Publish		
					View	Open	Edit	Delete
Nodejs Certification Course	Certificate of Appreciation	Muhammad Aamir	4/3/2021, 3:26:22 PM	Life time				
Vue js Certification Course	Vue JS Junoir Certificate	Muhammad Aamir	4/3/2021, 2:55:37 PM	Life time				

Figure 5.6 - Batch List Web Interface

The screenshot shows a web application interface for creating batch certificates. At the top, there are navigation links: DashBoard, Create, Certificates, Publications, Verify Certificate, and About. On the right side of the header are icons for notifications and user profile.

The main area is titled "BATCH CERTIFICATES". It displays the following details:

- Batch Name:** Nodejs Certification Course
- Certificate Title:** Certificate of Appreciation
- Created Date:** 4/3/2021
- Created By:** Muhammad Aamir

Below these details is a table with columns: Index, Name, Email, and Actions. There is one row with index 1, name (empty), email (empty), and an action button. At the bottom of the table is a "submit" button.

Figure 5.7 - Create Batch Certificates Web Interface

The screenshot shows a web application interface for listing batch certificates. At the top, there are navigation links: DashBoard, Create, Certificates, Publications, Verify Certificate, and About. On the right side of the header are icons for notifications and user profile.

The main area is titled "BATCH CERTIFICATES". It displays the following details:

- Batch Name:** Nodejs Certification Course
- Certificate Title:** Certificate of Appreciation
- Created Date:** 4/3/2021
- Created By:** Muhammad Aamir

Below these details is a table with columns: Issue Date, Candidate Name, Candidate Email, Issued By, and Actions. There are five rows of data, each corresponding to a certificate issued on 4/3/2021 at 9:57:28 PM to MUHAMMAD RAFAY with email muhammadrafay151@gmail.com, issued by Muhammad Aamir. The "Actions" column contains edit and delete icons. At the bottom of the table are navigation arrows and page numbers (1, 2, 3).

Figure 5.8 - Batch Certificates List Web Interface

The screenshot shows a web browser window with the URL mr-devs.ml/publications. The page title is "PUBLICATIONS". At the top, there are two tabs: "Single Certificates" (selected) and "Batch Certificates". Below the tabs is a table with the following columns: Name, Title, Publish By, Publish Date, Expiry Date, and Actions (View and Email). A single row is visible in the table:

Name	Title	Publish By	Publish Date	Expiry Date	Actions
Muhammad Aamir	ReactJS	Muhammad Aamir	4/3/2021	Life time	View Email

Figure 5.9 - Published Single Certificate Web Interface

The screenshot shows a web browser window with the URL mr-devs.ml/organization/config/. The page title is "Fast's Config". On the left, there is a sidebar with icons for "Cert Count", "User Management", and "About". The main content area has a heading "Count History". Below it is a table with the following columns: Date and Count. One row is visible in the table:

Date	Count
4/2/2021, 12:43:14 AM	12

Figure 5.10 - Count History Web Interface

The screenshot shows a web application interface titled "Fast's Config". On the left, there is a sidebar with icons for "Cert Count", "User Management" (selected), and "About". The main content area is titled "Users" and displays a table with one row of data. The table columns are "Register Date", "Name", "Email", and "Roles". The data row shows "3/28/2021", "Muhammad Aamir", "muhammad.aamir.a1@gmail.com", and "Admin Issuer". To the right of the table are three buttons: "Enable/Disable", "Update Profile", and "Reset Password". At the bottom of the table are navigation arrows.

Register Date	Name	Email	Roles	Actions
3/28/2021	Muhammad Aamir	muhammad.aamir.a1@gmail.com	Admin Issuer	<input type="button" value="Enable/Disable"/> <input type="button" value="Update Profile"/> <input type="button" value="Reset Password"/>

Figure 5.11 - User Management Web Interface

The screenshot shows a web application interface titled "Fast's Config". On the left, there is a sidebar with icons for "Cert Count", "User Management" (selected), and "About". The main content area displays a grid of user information. The columns are "Name", "Email", "Phone", "Address", "User Limit", "Ecert Count", "Registration Date", and "Status". The data is as follows:

Name: Fast	Email: fast@certifs.com	Phone: +92 3407838753	Address: FAST University, 3 A K. Brohi Road, H-11/4 H 11/4 H-11, Islamabad, Islamabad Capital Territory	User Limit: 4	Ecert Count: 12	Registration Date: 3/28/2021	Status: Active
------------	-------------------------	-----------------------	---	---------------	-----------------	------------------------------	----------------

Figure 5.12 - About Web Interface

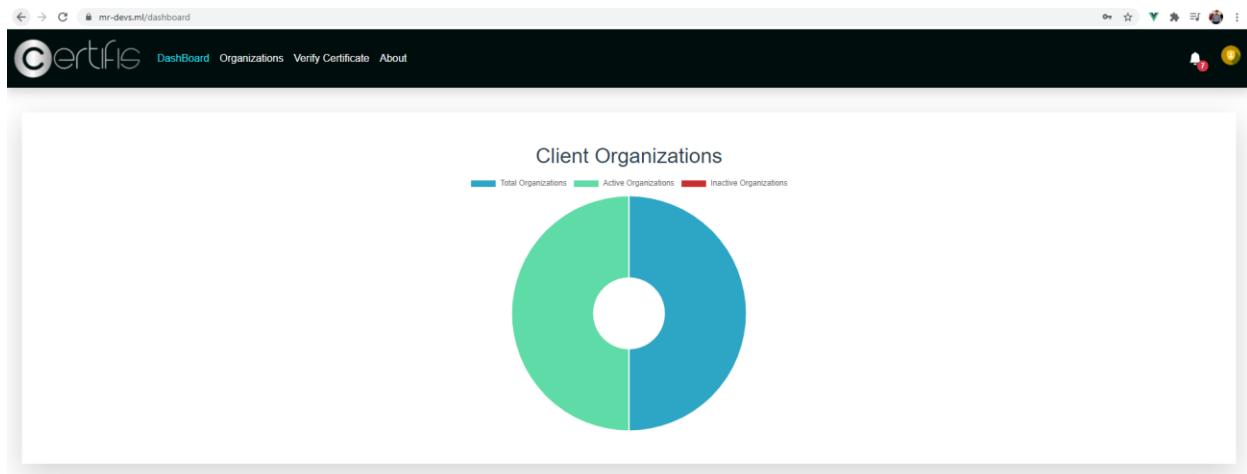


Figure 5.13 - Client Organization Graphical View Web Interface

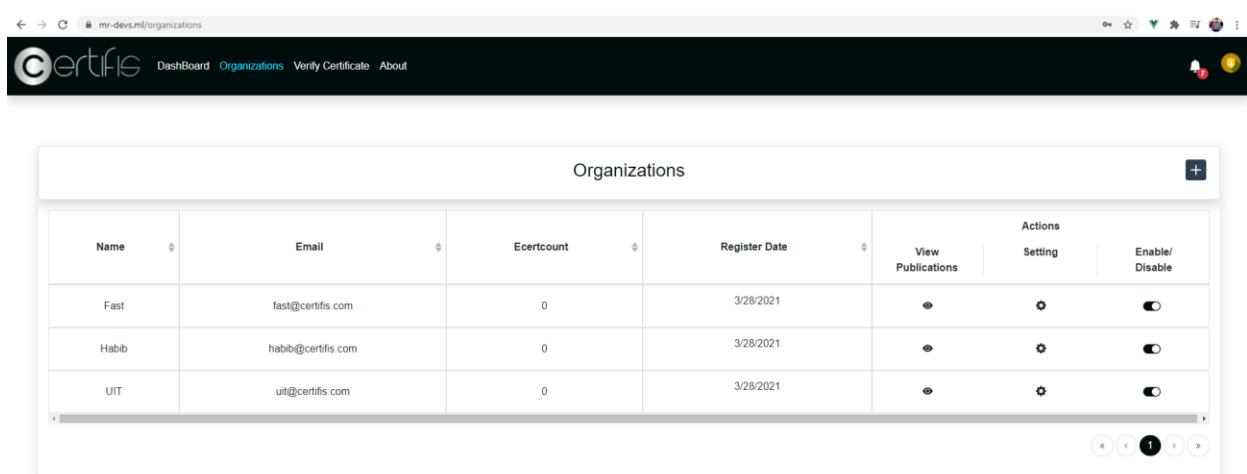


Figure 5.14 - Client Organization List Web Interface

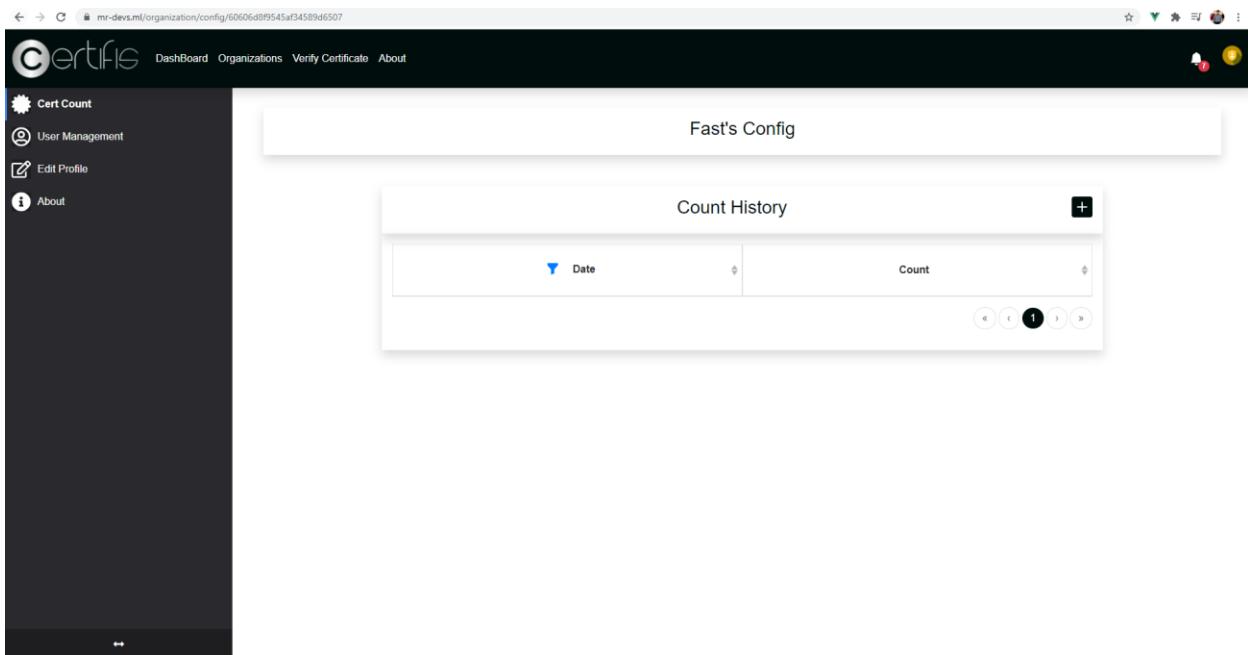


Figure 5.15 - Client Organization Count History Web Interface

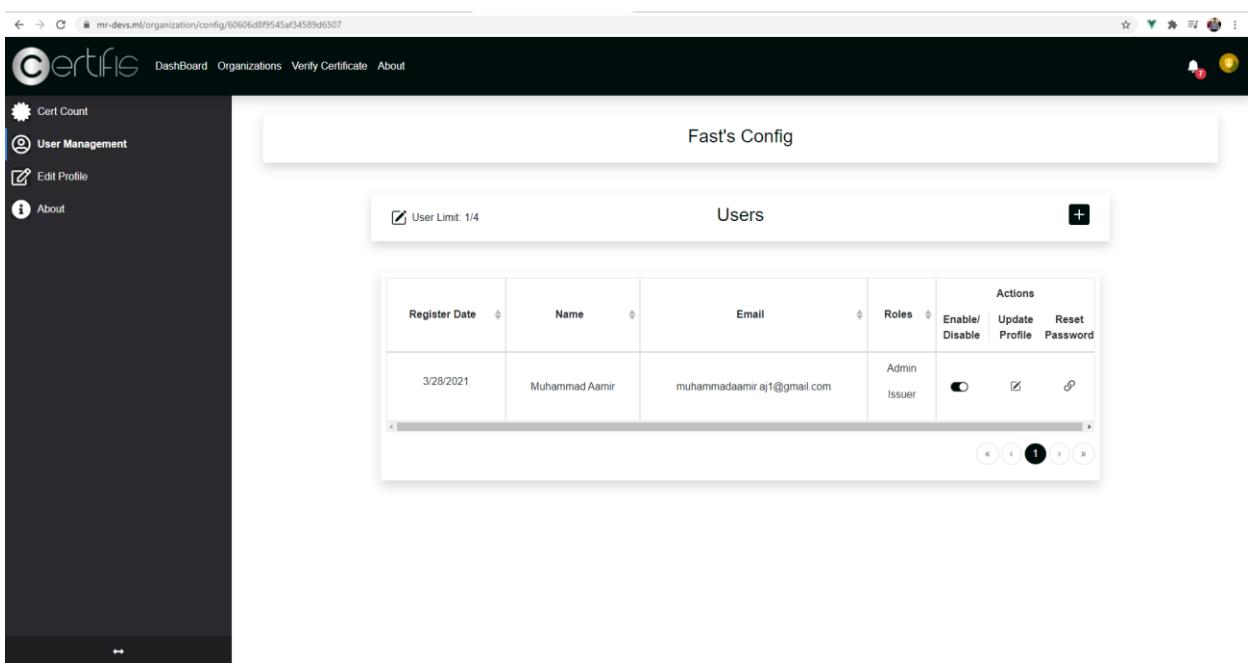


Figure 5.16 - Client Organization User Management Web Interface

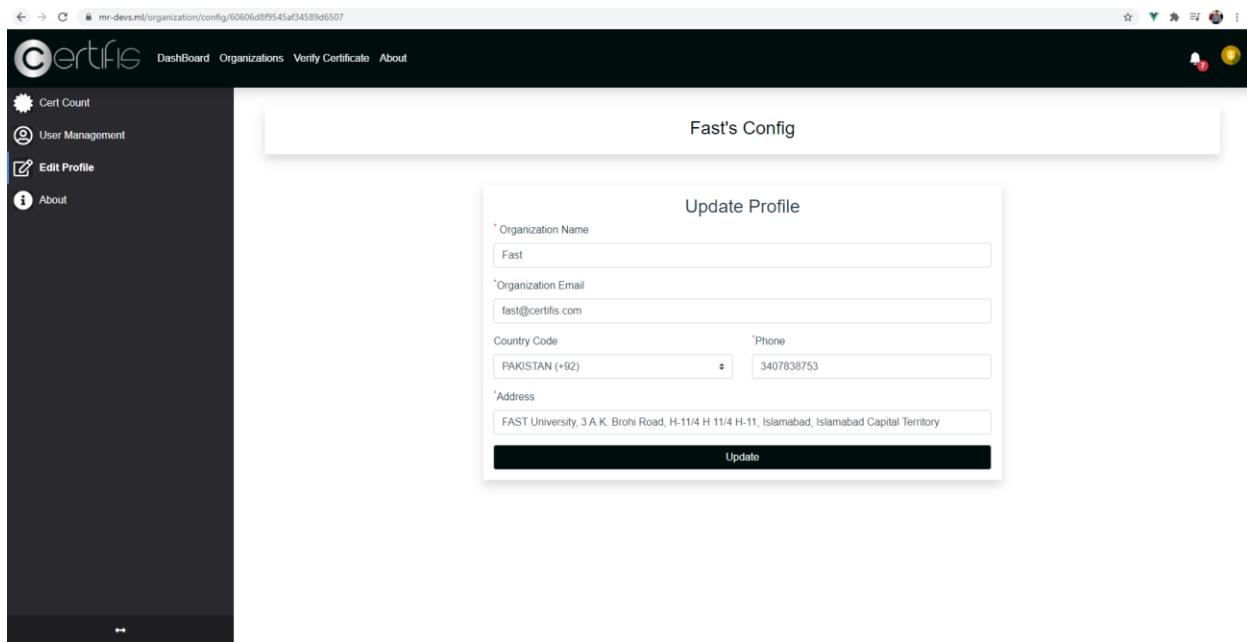


Figure 5.17 - Client Organization Update Profile Web Interface

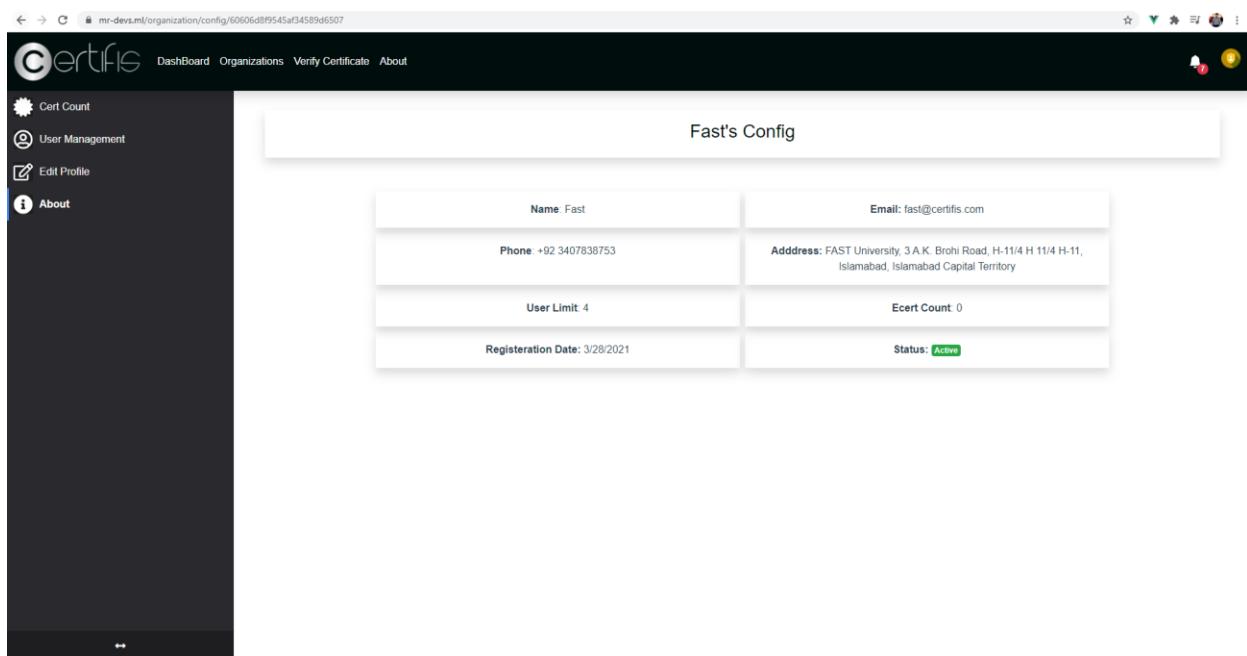


Figure 5.18 - Client Organization About Web Interface

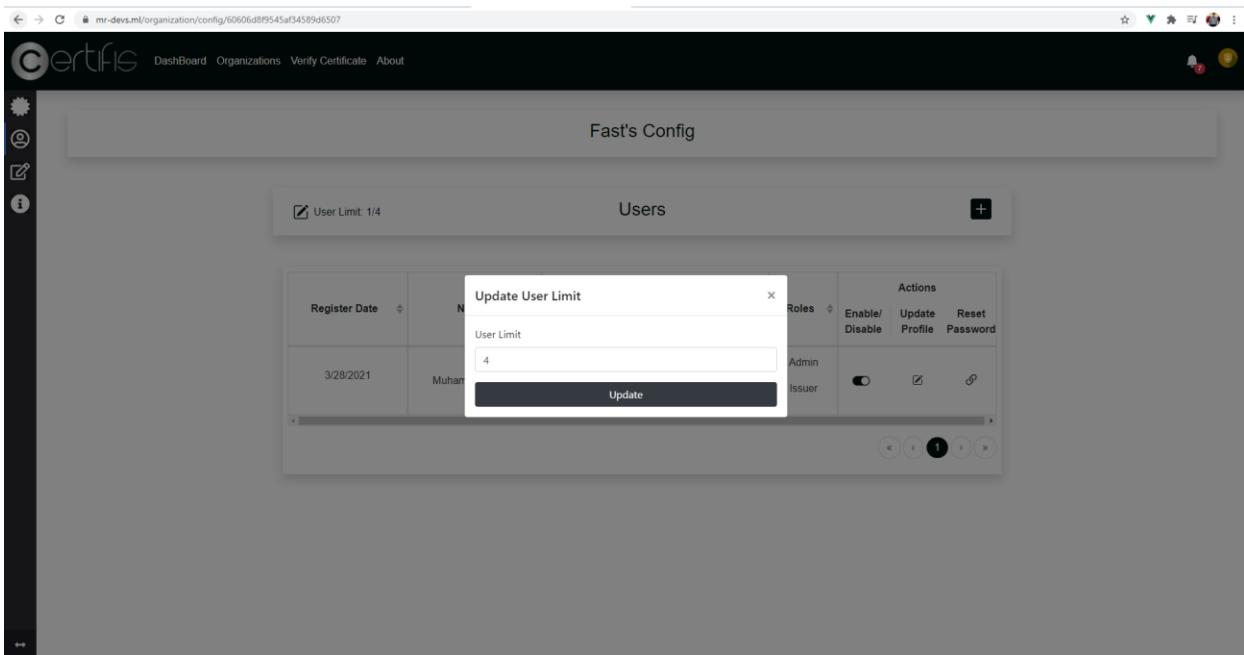


Figure 5.19 - Client Organization User Limit Web Interface

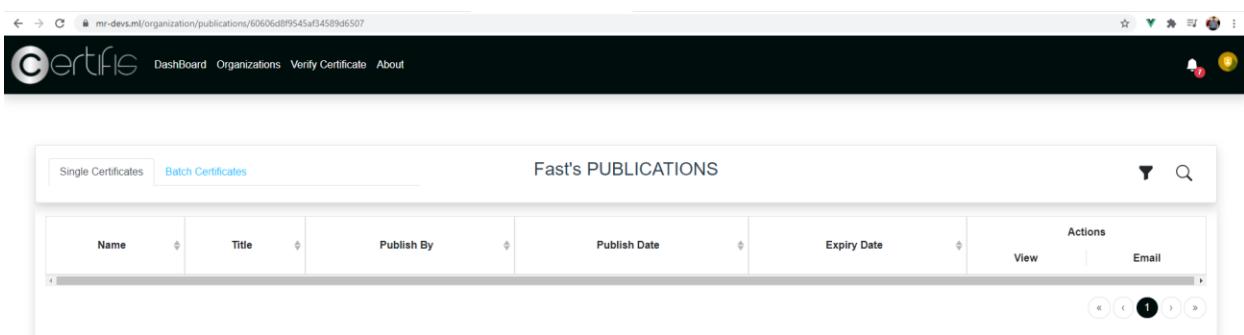


Figure 5.20 - Client Organization Published Single Certificate View Web Interface

The screenshot displays a web-based application interface for managing publications. At the top, there's a header with the 'Certific' logo and navigation links for Dashboard, Organizations, Verify Certificate, and About. Below the header is a search bar and a table titled 'Fast's PUBLICATIONS'. The table has columns for Batch Name, Title, Publish By, Publish Date, Expiry Date, Actions (with 'Open Batch' and 'Email' buttons), and navigation arrows. The table is currently empty.

Figure 5.21 - Client Organization Published Batch View Web Interface

The screenshot shows a user registration process. A modal window titled 'Register Issuer' is open, prompting for 'Name', 'Email address', 'Password', 'Country Code' (set to PAKISTAN (+92)), 'Phone' (3407838753), and 'Address' (street-250 Area 12 karachi). Below the modal, a table lists users with columns for 'Register Date', 'Name', and 'Actions' (Roles: Admin or Issuer, with options to 'Enable/Disable', 'Update Profile', or 'Reset Password'). The table shows one entry for 'Muham'.

Figure 5.22 - User Registration Web Interface

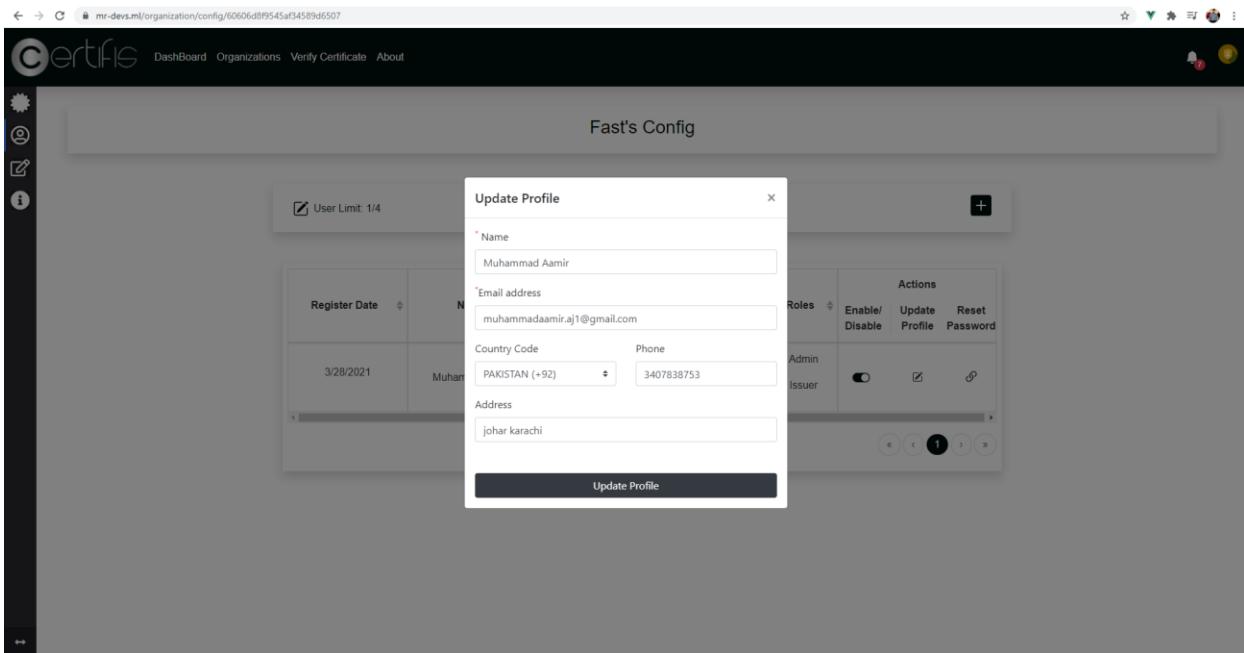


Figure 5.23 - Update User Profile Web Interface

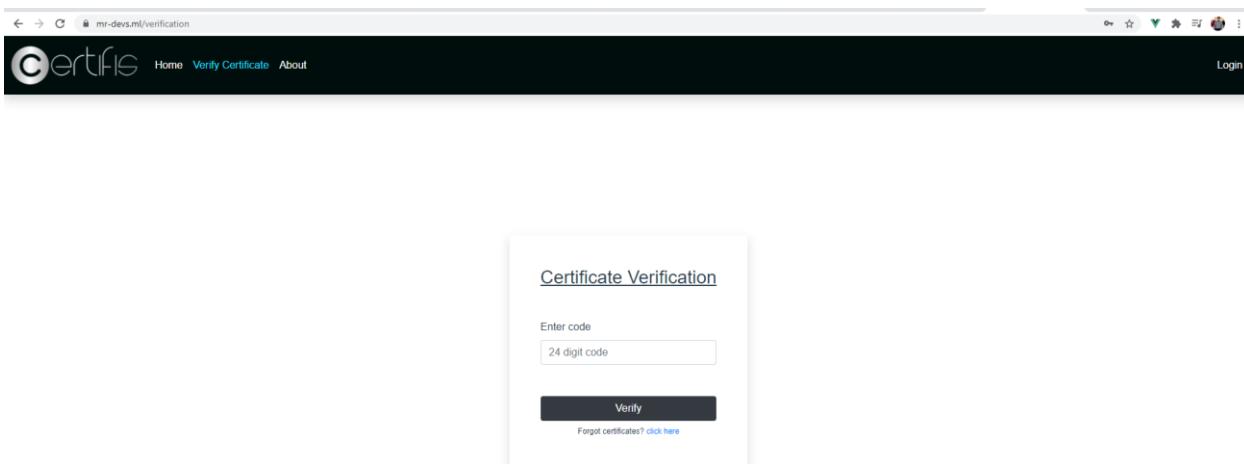


Figure 5.24 - Certificate Verification Web Interface

6. Implementation

This chapter contains code of most critical part of our project such as smart contract, publish and verify. This chapter also contain component, deployment and state transition diagram of our project.

6.1 Code

6.1.1 Smart Contract

```
'use strict';

const { Contract } = require('fabric-contract-api');

class ecert extends Contract {
    //use to initiate the ledger after installing chain code on peers
    async initLedger(ctx) {
        console.info('===== START : Initialize Ledger =====');
        const Certificates = [
            {
                email: 'muhammadrafay151@gmail.com',
                Name: 'Muhammad Rafay',
                Title: 'Js-Certifications',
                description: 'xyz',
                Organizations: 'Uit',
            },
        ];
        for (let i = 0; i < Certificates.length; i++) {
            Certificates[i].docType = 'certificate';
            await ctx.stub.putState('ecert' + i, Buffer.from(JSON.stringify(Certificates[i])));
            console.info('Added <->', Certificates[i]);
        }
        console.info('===== END : Initialize Ledger =====');
    }
    async QueryCertificate(ctx, CertificateNumber) {
        // retrieve the data from the ledger
        const CertificateAsBytes = await ctx.stub.getState(CertificateNumber);
        if (!CertificateAsBytes || CertificateAsBytes.length === 0) {
            // throw exception when certificate not found
            throw new Error(` ${CertificateNumber} does not exist`);
        }
        // return data
        return CertificateAsBytes.toString();
    }
    async PublishCertificate(ctx, data) {
        console.info('===== START : Create Certificate =====');
        //data parsing
        data = JSON.parse(data);
        let key = data._id;
        delete data._id;
        //insert the certificate data in the ledger
    }
}
```

```

        await ctx.stub.putState(key, Buffer.from(JSON.stringify(data)));
        console.info('===== END : Certificate created =====');
    }
}

async PublishBatch(ctx, data) {
    //data parsing
    let batch = JSON.parse(data);
    //iteratively insert the batch certificate data in the ledger
    for (let x = 0; x < batch.length; x++) {
        let key = batch[x]._id;
        delete batch[x]._id;
        await ctx.stub.putState(key, Buffer.from(JSON.stringify(batch[x])));
    }
}
module.exports = ecert;

```

6.1.1.1 Explanation

The smart contract contains the method which will be invoked in order to read and write data to the blockchain ledger.

6.1.2 Publish

```

const express = require('express');
const router = express.Router()
const Auth = require('../Auth/Auth')
const Roles = require('../js/Roles')
const batch_cert = require('../models/batch_certificates')
const batch = require('../models/batch')
const cert = require('../models/certificate')
const MsgBroker = require("../MessageBroker/publisher")
const CountHandler = require("../js/CountHandler");
const config = require("config")
const { StatusCodeException } = require('../Exception/StatusCodeException');
const NotificationHandler = require("../js/NotificationHandler");
const Constants = require("../Constants");
router.post("/single", Auth.authenticateToken, Auth.CheckAuthorization([Roles.Admin, Roles.Issuer]), async (req, res) => {
    try {
        //reduce the count
        await CountHandler.ReduceCount(req.user.org_id, 1);

        //publishing info
        let publish = {
            status: true,
            publisher_name: req.user.name,
            publisher_email: req.user.email,
            publish_date: Date.now(),
            processing: false
        }
        //if blockchain enabled publish certificate to blockchain
        if (req.app.get("BlockChain_Enable")) {

```

```

    //load the certificate and mark it as processing
    let ct = await cert.findOneAndUpdate({ _id: req.body.id, 'issuedby.org_id': req.user.org_id, 'publish.status': false, 'publish.processing': false }, { $set: { 'publish.processing': true } }).lean()
    if (ct) {
        if (config.get("app.debugging") === true) {
            const io = req.app.get("socketio");
            ct.message = "send to message queue";
            io.to("debugging").emit("log", ct);
        }
        //send publish request to message broker
        await MsgBroker.send(true, { user: req.user, certid: req.body.id })
    }
    res.send("Processing started we will notify u soon")
}
else {
    // non blockchain instance
    try {
        //find certificate and chnge the status
        let crt = await cert.findOneAndUpdate({ _id: req.body.id, 'issuedby.org_id': req.user.org_id, 'publish.status': false }, { $set: { publish: publish } })
        if (crt) {
            //send notification
            let message = `${crt.title} certificate with id: ${crt._id} has been published`;
            await NotificationHandler.NewNotification(req.user, message, Constants.Private);
            res.status(200).send("Published successfully")
        } else {
            //certificate not exist rollback the count
            await CountHandler.IncreaseCount(req.user.org_id, 1);
            res.status(404).send("certificate not found")
        }
    } catch (err) {
        //error while updating certificate so rollback the count
        await CountHandler.IncreaseCount(req.user.org_id, 1);
        res.status(500).send(err)
    }
}
} catch (err) {
    //status code exception is used by logic handlers to throw error
    if (err instanceof StatusCodeException) {
        res.status(err.StatusCode).send(err.Message)
    } else {
        //unknown errors
        res.status(500).send(err)
    }
}
})

```

```

router.post("/batch", Auth.authenticateToken, Auth.CheckAuthorization([Roles.Admin, Roles.Issuer]), async (req, res) => {
  try {
    //check if the batch exist against the provided id
    let bt = await batch.findOne({ _id: req.body.id, 'createdby.org_id': req.user.org_id, 'publish.status': false })
    if (bt) {
      let bcert = await batch_cert.find({ batch_id: req.body.id }).countDocuments()
      //batch should contain atleast more than 1 certificate
      if (bcert && bcert > 1) {
        await CountHandler.ReduceCount(req.user.org_id, bcert);

        //if blockchain enabled publish certificate to blockchain
        if (req.app.get("BlockChain_Enable")) {
          //load the certificate and mark it as processing
          let bt = await batch.findOneAndUpdate({ _id: req.body.id, 'createdby.org_id': req.user.org_id, 'publish.status': false, 'publish.processing': false }, { $set: { 'publish.processing': true } }).lean();
          if (bt) {
            if (config.get("app.debugging") === true) {
              const io = req.app.get("socketio");
              bt.message = "send to message queue";
              io.to("debugging").emit("log", bt);
            }
            //send publish request to message broker
            await MsgBroker.send(false, { user: req.user, batchid: req.body.id })
            res.send("Processing started we will notify u soon")
          }
        }
        else
          res.status(404).send()
      } else {
        //publishing info
        let publish = {
          status: true,
          publisher_name: req.user.name,
          publisher_email: req.user.email,
          publish_date: Date.now(),
          processing: false
        }
        try {
          //send notification
          await batch.findOneAndUpdate({ _id: req.body.id, 'createdby.org_id': req.user.org_id, 'publish.status': false }, { $set: { publish: publish } })
          let message = `${bt.batch_name} batch with id: ${bt._id} has been published`;
          await NotificationHandler.NewNotification(req.user, message, Constants.Private);
          res.send("Batch Published...")
        } catch (err) {
          //error while updating certificate so rollback the count
        }
      }
    }
  }
}

```

```

        await CountHandler.IncreaseCount(req.user.org_id, bcert);
        res.status(500).send(err)
    }
}
} else {
    res.status(409).send("batch must have more than 1 certificate to publish")
}
} else {
    return res.status(404).send("batch not found")
}

} catch (err) {
    //status code exception is used by logic handlers to throw error
    if (err instanceof StatusCodeException) {
        res.status(err.StatusCode).send(err.Message)
    } else {
        //unknown errors
        res.status(500).send(err)
    }
}
})

module.exports = router

```

6.1.2.1 Explanation

This API will be invoked in order to publish data to the blockchain, actually this API sends request to the message broker and then the available consumers process the request.

6.1.3 Verify

```

const express = require('express');
const router = express.Router()
const batch = require('../models/batch')
const batch_cert = require('../models/batch_certificates')
const cert = require('../models/certificate');
const mongoose = require("mongoose");
const { GetCertificate } = require("../BlockChain/query")
router.get('/:id', async (req, res) => {
    try {
        if (!mongoose.Types.ObjectId.isValid(req.params.id))
            return res.status(400).send("Invalid Code");
        //if blockchain enabled it will query from database
        if (req.app.get("BlockChain_Enable")) {
            //query the certificate from blockchain using id
            let result = await GetCertificate(req.params.id)
            if (!result)
                return res.status(404).send()
            result = JSON.parse(result)
            result._id = req.params.id
            result.blockchain = true
        }
    }
})

```

```

    }
  else {
    //for non blockchain instance of application
    //first find in single cert collection
    var result = await cert.findOne({ _id: req.params.id, 'publish.status': true }).lean()
    if (!result) {
      //if not found in single cert collection than find in batch cert collections
      var bcert = await batch_cert.findOne({ _id: req.params.id }, { updatedby: 0 })
      if (bcert) {
        var b1 = await batch.findOne({ _id: bcert.batch_id, 'publish.status': true }, {
          updatedby: 0 }).lean()
        if (b1) {
          //mapping fields
          b1.issue_date = bcert.issue_date
          b1.batch_id = b1._id
          b1._id = bcert._id
          b1.name = bcert.name
          b1.email = bcert.email
          b1.blockchain = false
        } else {
          return res.status(404).send()
        }
      }
    }
    else {
      return res.status(404).send()
    }
    result = b1
  }
}
//check the certificate validity
if (result.expiry_date && new Date(result.expiry_date) - Date.now() < 0) {
  result.is_expired = true
  result.message = "The Certificate is verified but the validity is expired"
} else {
  result.is_expired = false
  result.message = "The Certificate is verified"
}
res.json(result)
}
catch (err) {
  //unknown errors
  res.status(500).send()
}
})
module.exports = router

```

6.1.3.1 Explanation

This API sends request directly to the blockchain to verify and retrieve the certificate.

6.2 Component Diagram

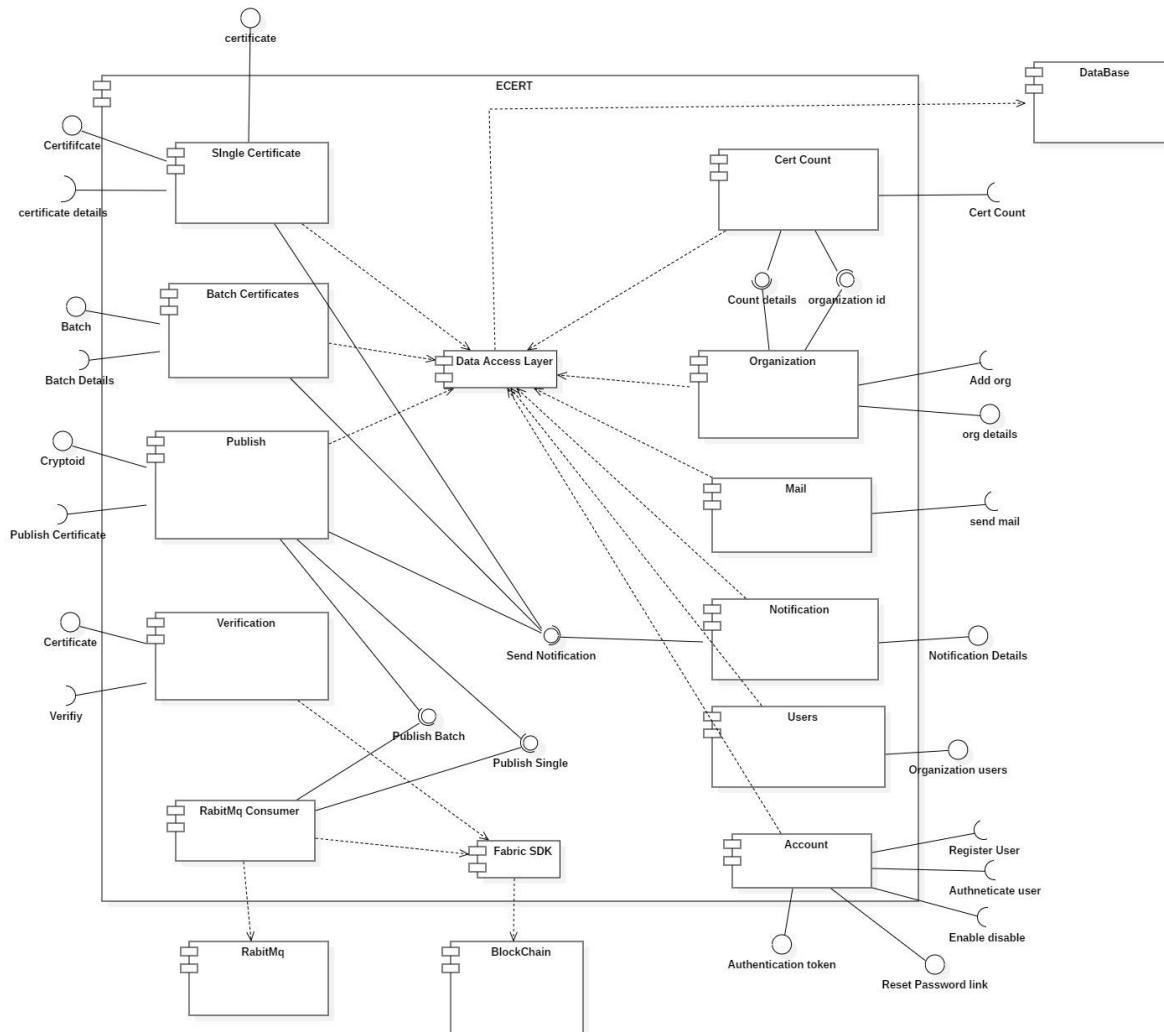


Figure 6.1 - Component Diagram

The figure 6.1 shows all the components of our project. This diagram shows how these components interact with each other.

6.3 Deployment Diagram

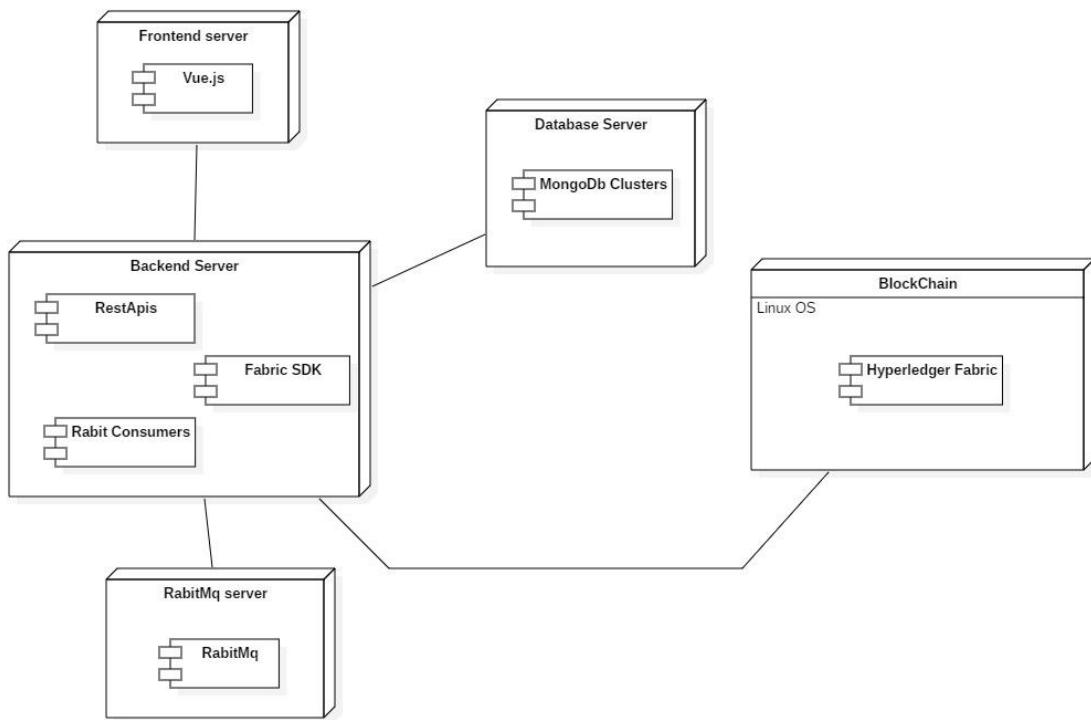


Figure 6.2 - Deployment Diagram

The figure 6.2 shows the process deployment of our project. This diagram shows how one can deploy our project what they need to install in which operating system, to run our project successfully.

6.4 State Transition Diagram

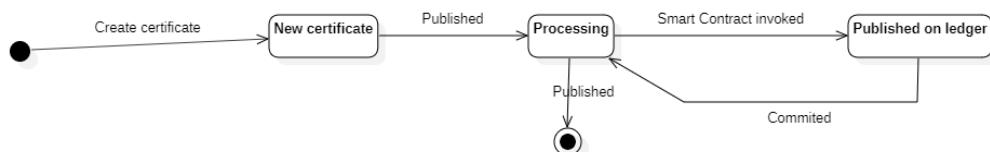


Figure 6.3 - Publish's State Transition Diagram

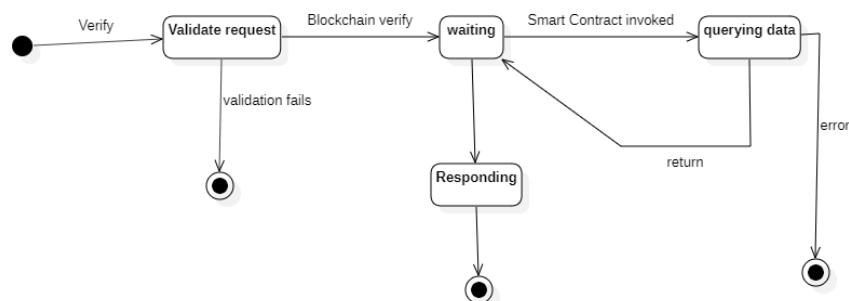


Figure 6.4 - Verification's State Transition Diagram

7. Testing

In this chapter we have cover black box and white box testing of our most logical components. Below we have attached all test cases with their results. The black box testing was carried out manually and for white box testing we used mocha and chaijs package for JavaScript where Mocha is a JavaScript test framework running on Node. Mocha allows asynchronous testing, test coverage reports, and use of any assertion library. White chai is a BDD / TDD assertion library for NodeJS and the browser that can be delightfully paired with any JavaScript testing framework.

7.1 White Box Testing

The testing plan for white box testing is given below:

First, we identified the most logical code from our project then from that we selected the most critical code and performed white box testing. The code we selected is publish single certificates, publish batches and verify certificates. Then we drew cyclometric complexity and selected the appropriate tool to perform testing.

7.1.1 Publish Single Certificate

7.1.1.1 Code

```

1//Entry router.post("/single", Auth.authenticateToken,
Auth.CheckAuthorization([Roles.Admin, Roles.Issuer]), async (req, res) =>
{
2    try {
3        let ct = await cert.findOneAndUpdate({ _id: req.body.id,
'issuedby.org_id': req.user.org_id, 'publish.status': false, 'publish.processing': false }, { $set: { 'publish.processing': true } }).lean()
4        if (ct)
5        {
6            if (config.get("app.debugging") === true)
7            {
8                const io = req.app.get("socketio");
9                ct.message = "send to message queue";
10               io.to("debugging").emit("log", ct);
11            }
12            await MsgBroker.send(true, { user: req.user, certid: req.body.id })
13        }
14        res.send("Processing started we will notify u soon")
15    }
16    catch (err)
17    {
18        console.log(err)
19        res.status(500).send()
20    }
21}
22//exit})

```

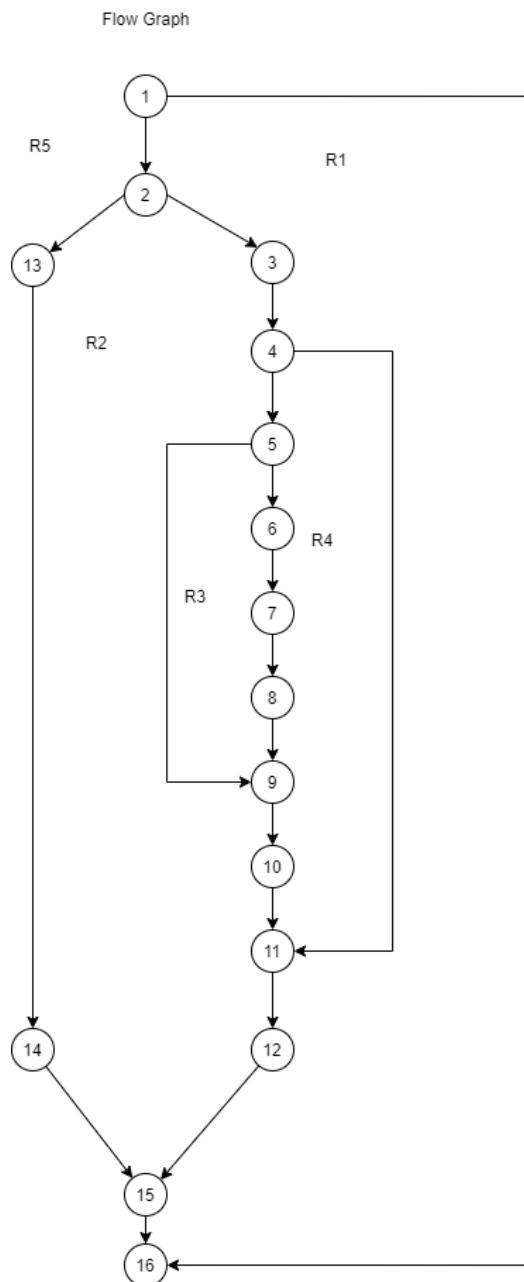


Figure 7.1 - Publish Single Certificate Flow Graph

7.1.1.2 Cyclometric Complexity

$$V(G) = \text{Edges} - \text{Node} + 2 = 18 - 15 + 2 = 5$$

$$V(G) = \text{Predicate node} + 1 = 4 + 1 = 5$$

$$V(G) = \text{Open Region} + \text{Close Region} = 5$$

7.1.1.3 Path

1. 1-16
2. 1-2-13-14-15-16
3. 1-2-3-4-11-12-15-16
4. 1-2-3-4-5-9-10-11-12-15-16
5. 1-2-3-4-5-6-7-8-9-10-11-12-15-16

7.1.1.4 Test Cases

Table 7.1 - Publish Single Certificate Test Cases

Test Case ID	Test Scenario	Test Data	Expected Result	Actual Result	Valid/Invalid
Path-01.	Restrict accesses of user who are not an admin nor an issuer	{"id":"60927dd270c7f13be8c7e39f"}	Return message "Unauthorized".	Message is returned "Unauthorized".	Valid
Path-02.	Check If any unexpected error happens	null	Log the error and return status code 500.	Status of 500 is returned and Log error obtained.	Valid
Path-03.	If certificate is already under publishing process	{"id":"60927dd270c7f13be8c7e39f"}	Return message "Processing already started we will notify you soon".	Message Returned "Processing already started we will notify you soon".	Valid
Path-04.	Check if debugging mode is disable	{"id":"60927dd270c7f13be8c7e39f"}	Don't Generate debugging logs under debugging section.	Debugging mode is Disabled.	Valid
Path-05.	Check if proper valid single certificate is provided with debugging logs enabled	{"id":"60927dd270c7f13be8c7e39f"}	Return message "Processing started we will notify you soon" and generate debugging logs.	Message Returned "Processing started we will notify you soon" and logs are generated.	Valid

7.1.2 Publish Batch Certificate

7.1.2.1 Code

```

1//Entry router.post("/batch", Auth.authenticateToken,
Auth.CheckAuthorization([Roles.Admin, Roles.Issuer]), async (req, res) =>
{
2    try {
3        let bt = await batch.findOneAndUpdate({ _id: req.body.id,
'createdby.org_id': req.user.org_id, 'publish.status': false,
'publish.processing': false }, { $set: { 'publish.processing': true } }).lean();
4        if (bt)
5        {
6            var bcert = await batch_cert.find({ batch_id: req.body.id
}).countDocuments()
7            if (bcert && bcert > 1)
8            {
9                if (config.get("app.debugging") === true)
10                {
11                    const io = req.app.get("socketio");
12                    bt.message = "send to message queue";
13                    io.to("debugging").emit("log", bt);
14                }
15            }
16            else
17            {
18                res.status(409).send("batch must have more than 1 certificate to
publish")
19            }
20        }
21        else
22        {
23            return res.status(404).send("batch not found")
24        }
25    }
26    catch (err)
27    {
28        res.status(500).send(err)
29    }
30}
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
311
312
313
313
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1
```

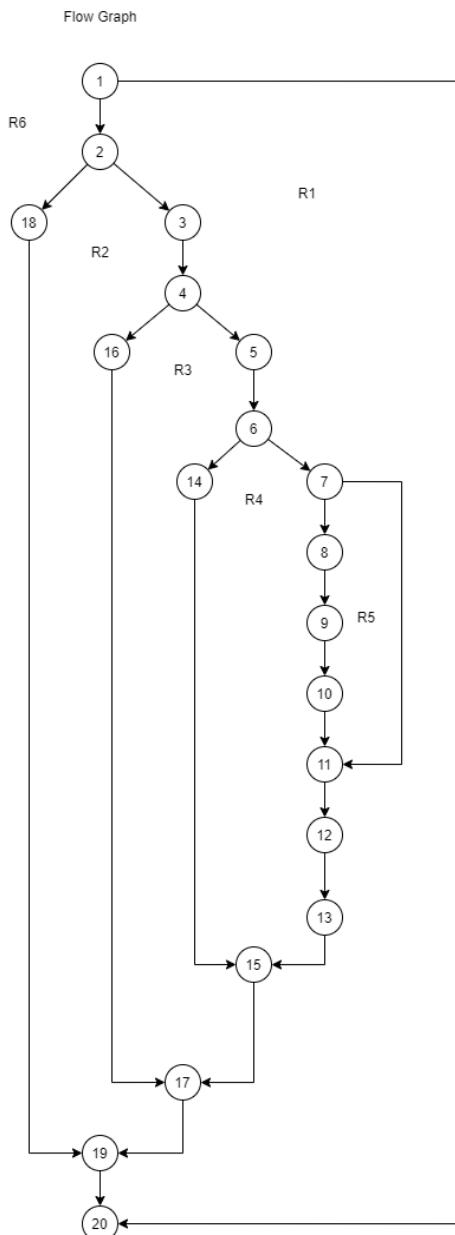


Figure 7.2 - Publish Batch Flow Graph

7.1.2.2 Cyclometric Complexity

$$V(G) = \text{Edges} - \text{Node} + 2 = 24 - 20 + 2 = 6$$

$$V(G) = \text{Predicate node} + 1 = 5 + 1 = 6$$

$$V(G) = \text{Open Region} + \text{Close Region} = 6$$

7.1.2.3 Paths

1. 1-20
 2. 1-2-18-19-20
 3. 1-2-3-4-16-17-19-20
 4. 1-2-3-4-5-6-14-15-17-19-20
 5. 1-2-3-4-5-6-7-11-12-13-15-17-19-20
 6. 1-2-3-4-5-6-7-8-9-10-11-12-13-15-17-19-20

7.1.2.4 Test Cases

Table 7.2 - Publish Batch Test Cases

Test Case ID	Test Scenario	Test Data	Expected Result	Actual Result	Valid/Invalid
Path-01.	Restrict accesses of user who are not an admin nor an issuer	{"id":"6092e2e4939a4600176f77fa"}	Return error message "Unauthorized".	Message is returned "Unauthorized".	Valid
Path-02.	Check If any unexpected error happens	null	Log the error and return status code 500.	Status of 500 is returned and Log error obtained.	Valid
Path-03.	if certificate is already under publishing process	{"id":"6092e2e4939a4600176f77fa"}	Return message "Processing already started we will notify you soon".	Message Returned "Processing already started we will notify you soon"	Valid
Path-04.	Check if a batch is sent for publishing with less than 2 certificates in it.	{"id":"6092e2e4939a4600176f77fa"}	Return status code "409" with error message "batch must have more than 1 certificate to publish".	Batch must have more than 1 Certificate and status of 409 returned.	Valid
Path-05.	Check if debugging mode is disable	{"id":"6092e2e4939a4600176f77fa"}	Don't Generate debugging logs under debugging section.	Debugging mode is Disabled.	Valid
Path-06.	Check if proper valid batch certificate is provided with debugging logs enabled	{"id":"6092e2e4939a4600176f77fa"}	Return message "Processing started we will notify you soon" and generate debugging logs	Message Returned "Processing started we will notify you soon" and logs are generated.	Valid

7.1.3 Verify Certificate

7.1.3.1 Code

```
1//Entry router.get('/:id', async (req, res) =>
{
2    try {
3        if (!mongoose.Types.ObjectId.isValid(req.params.id))
4        {
5            return res.status(400).send("Invalid Code");
5//endif }
6
7        var result = await GetCertificate(req.params.id)
8        result = JSON.parse(result)
9        result.blockchain = true
10
11        if (result.expiry_date && result.expiry_date - Date.now() < 0)
12        {
13            result.is_expired = true
14            result.message = "The Certificate is verified but the validity is
expired"
15
16        }
17        else
18        {
19            result.is_expired = false
20            result.message = "The Certificate is verified"
21//endif}
22
23        res.json(result)
24    }
25    catch (err)
26    {
27        console.log(err)
28        res.status(500).send()
29//endtrycatch    }
30//exit})
```

Flow Graph

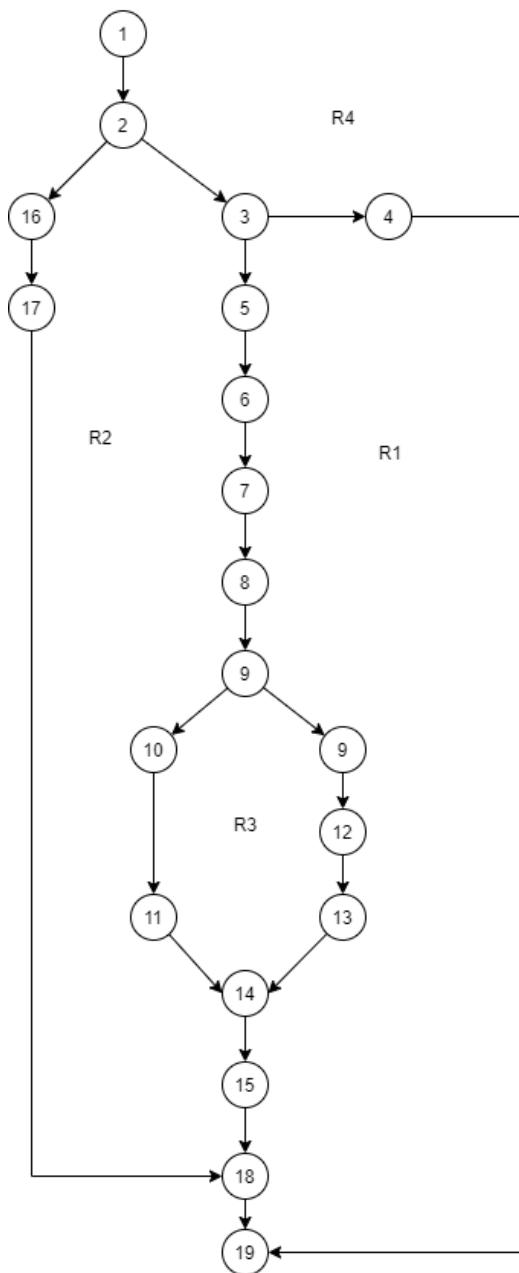


Figure 7.3 - Verify Certificate Flow Graph

7.1.3.2 Cyclometric Complexity

$$V(G) = \text{Edges} - \text{Node} + 2 = 21 - 19 + 2 = 4$$

$$V(G) = \text{Predicate node} + 1 = 3 + 1 = 4$$

$$V(G) = \text{Open Region} + \text{Close Region} = 4$$

7.1.3.3 Path

1. 1-2-16-17-18-19
2. 1-2-3-4-19
3. 1-2-3-5-6-8-9-10-11-14-15-18-19
4. 1-2-3-5-6-8-9-12-13-14-15-18-19

7.1.3.4 Test Cases

Table 7.3 - Verify Certificate Test Cases

Test Case ID	Test Scenario	Test Data	Expected Result	Actual Result	Valid/ Invalid
Path-01.	Check If any unexpected error happens	{"code": "606cac98ec653783a1e62"}	Log the error and return status code 500	Status of 500 is returned and Log error obtained.	Valid
Path-02.	Check If Invalid verification code is provided	{"code": "606cac98eb783a1e62"}	Return status code 400 with error message "invalid code"	The code is invalid and status 400 is obtained.	Valid
Path-03.	Check if expired certificate code is provided	{"code": "606ca3fa11c1641728c127ec"}	Return message "The Certificate is verified but the validity is expired"	The certificate is expired and verified.	Valid
Path-04.	Check if a valid certificate code is provided	{"code": "606cac98eb64c653783a1e62"}	Return message "The Certificate is verified"	The Certificate is verified.	Valid

7.2 Black Box Testing

Black box testing is a form of software testing in which the functionality of a software is tested without regard for its implementation or source code. Black box testing is solely focused on the software's features and requirements. In black box testing, we just rely on two simple operations: a system's input and output. For Black box testing, we used the interfaces of Publish Single Certificate, Publish Batches and Verify Certificate.

7.2.1 Publish Single Certificate Interface

The screenshot shows a web application interface for managing certificates. At the top, there is a navigation bar with links: DashBoard, Create, Certificates (which is highlighted in blue), Publications, Verify Certificate, and About. On the far right of the header, there are notification icons for messages and notifications, with a red badge indicating 7 notifications.

The main area is titled "CERTIFICATES". Below the title, there is a table with the following data:

Name	Title	Issuedby	Issue Date	Expiry Date	Actions	Publish
View	Edit	Delete				
Hassan	React	Muhammad Ismail	5/5/2021, 10:49:45 PM	5/3/2021	<input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>	<button>Publish</button>
Junaid	Graphic Designing	Muhammad Ismail	5/5/2021, 1:06:27 AM	Life time	<input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>	<button>Publish</button>

At the bottom of the table, there are navigation arrows and a page number indicator showing page 1 of 1.

Figure 7.4 - Publish Single Certificate Interface

7.2.1.1 Functionality Testing

Table 7.4 – Publish Single Certificate Functionality Testing

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Valid/ Invalid
TF01.	Does it publish certificate to the blockchain?	1. Go to site: https://mr-devs.ml/ 2. Click on Certificate. 3. Select certificate 4. Click Publish.	Should publish certificates from local database to the blockchain.	Publish certificates to the blockchain.	Valid
TF02.	Does it prompt error while publishing to the blockchain when blockchain is not connected?	1. Go to site: https://mr-devs.ml/ 2. Click on Certificate. 3. Select certificate 4. Click Publish.	Should prompt error.	Prompting error.	Valid
TF03.	Does it decrease certificate count balance after successfully publishing to the blockchain?	1. Go to site: https://mr-devs.ml/ 2. Click on Certificate. 3. Select certificate 4. Click Publish.	Should decrease certificate count balance.	Decreasing certificate count balance.	Valid
TF04.	Does it prompt error while publishing to the blockchain when there is insufficient certificate count balance?	1. Go to site: https://mr-devs.ml/ 2. Click on Certificate. 3. Select certificate 4. Click Publish.	Should not publish certificates to the blockchain.	Not publishing to the blockchain.	Valid

7.2.1.2 Validation Testing

TV01. Button (Publish Single Certificate)

Guidelines

- TV0101. Does the Cursor change?
Yes.
- TV0102. Does it prompt confirmation before publishing to the blockchain?
Yes.
- TV0103. Do the font size similar to the other attributes in the interface?
Yes.
- TV0104. Do the font color similar to the other attributes in the interface?
Yes.
- TV0105. Does the spelling correct?
Yes.
- TV0106. Do the button size similar to the other button?
Yes.
- TV0107. Does the expected result is displayed on click?
Yes.

7.2.2 Publish Batch Interface

The screenshot shows a web-based application interface titled "CERTIFICATES". At the top, there are two tabs: "Single Certificates" (highlighted in blue) and "Batch Certificates". Below the tabs is a search bar with a magnifying glass icon and a filter icon. The main area contains a table with the following columns: "Batch Name", "Title", "Created By", "Created Date", "Expiry Date", "Actions", and "Publish". A single row is visible in the table:

Batch Name	Title	Created By	Created Date	Expiry Date	Actions	Publish
17B	Graphic Designing	Muhammad Ismail	5/6/2021, 10:43:35 AM	Life time	View Open Edit Delete	Publish

At the bottom right of the table, there is a navigation bar with icons for back, forward, and search.

Figure 7.5 - Publish Batch Interface

7.2.2.1 Functionality Testing

Table 7.5 - Publish Batch Functionality Testing

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Valid/ Invalid
TF01.	Does it publish batch to the blockchain?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Certificate. 3. Click on Batch Certificate. 4. Select Batch. 5. Click Publish. 	Should publish Batch from local database to the blockchain.	Publish certificates to the blockchain.	Valid
TF02.	Does it prompt error while publishing to the blockchain when blockchain is not connected?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Certificate. 3. Click on Batch Certificate. 4. Select Batch. 5. Click Publish. 	Should prompt error.	Prompt's error.	Valid
TF03.	Does it decrease certificate count balance after successfully publishing to the blockchain?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Certificate. 3. Click on Batch Certificate. 4. Select Batch. 5. Click Publish. 	Should decrease certificate count balance.	Decreases certificate count balance.	Valid
TF04.	Does it prompt error while publishing to the blockchain when there is insufficient certificate count balance?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Certificate. 3. Click on Batch Certificate. 	Should not publish certificates to the blockchain.	Not publishing to the blockchain.	Valid

	4. Select Batch. 5. Click Publish.			
TF05.	Does it prompt error while publishing to the blockchain when there are less than 2 certificates in the batch?	1. Go to site: https://mr-devs.ml/ 2. Click on Certificate. 3. Click on Batch Certificate. 4. Select Batch. 5. Click Publish.	Should not allow to publish batch.	Not allowing to publish the batch. Valid

7.2.2.2 Validation Testing

TV01. Button (Publish Single Certificate)

Guidelines

- TV0101. Does the Cursor change?
Yes.
- TV0102. Does it prompt confirmation before publishing to the blockchain?
Yes.
- TV0103. Do the font size similar to the other attributes in the interface?
Yes.
- TV0104. Do the font color similar to the other attributes in the interface?
Yes.
- TV0105. Does the spelling correct?
Yes.
- TV0106. Do the button size similar to the other button?
Yes.
- TV0107. Does the expected result is displayed on click?
Yes.

7.2.3 Verify Certificate Interface

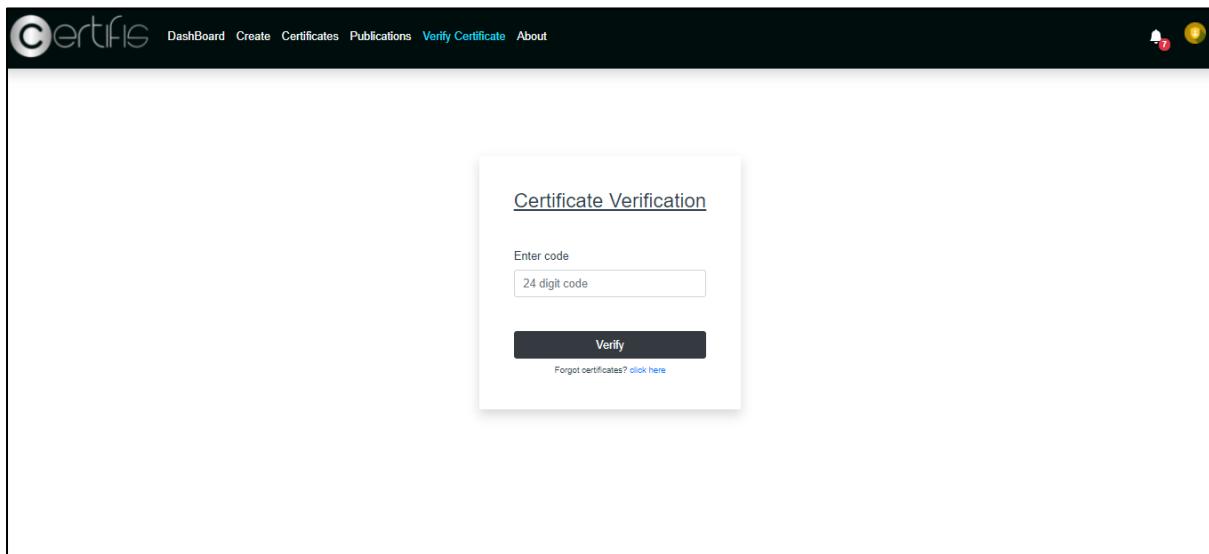


Figure 7.6 - Verify Certificate Interface

7.2.3.1 Functionality Testing

Table 7.6 - Verify Certificate Functionality Testing

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Valid/Invalid
TF01.	Does it verify the certificate on the valid certificate ID?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606cab67eb64c65378 3a1e5c	Certificate should be displayed.	Displays the certificate.	Valid
TF02.	Does it verify the certificate on the invalid certificate ID?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606cab67eb64c65378 3a1e5s	Should prompt Invalid code error.	Prompts an invalid code error.	Valid
TF03.	Does it verify the expired certificate along with expiry status?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606c9f8511c1641728 c127c9	Certificate should be displayed along with expiry status.	Displays the certificate along with expiry status.	Valid

7.2.4 Validation Testing

TV01.Textbox (Code)

Table 7.7 - Verify Certificate Textbox Validation Testing

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Valid/Invalid
TV0101.	Do the textbox editable?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606cab67eb64c65378 3a1e5c	Textbox should be editable.	Textbox is editable.	Valid
TV0102.	Does the textbox allow alphanumeric characters?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606cab67eb64c65378 3a1e5c	Data should be acceptable.	Data is acceptable.	Valid
TV0103.	Does the textbox allow uppercase?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606CAB67EB64C65 3783A1E5C	Data should be acceptable.	Data is acceptable.	Valid

TV0104.	Does the textbox allow lowercase?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606cab67eb64c65378 3a1e5c	Data should be acceptable.	Data is acceptable.	Valid
TV0105.	Does the textbox allow uppercase and lowercase both?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606CAB67EB64C65 3783a1e5c	Data should be acceptable.	Data is acceptable.	Valid
TV0106.	Does the textbox have maximum character limit?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606cab67eb64c65378 3a1e5ct	Size limit should not be more than 24 characters.	Data is not acceptable.	Valid
TV0107.	Does the textbox have minimum character limit?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606cab67eb64c65378 3a1e5	Size limit should not be less than 24 characters.	Data is not acceptable.	Valid

TV0108.	Does the textbox allow entering “Spaces” in the prefix and suffix?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606CAB67EB64C65 3783a1e5c	Data should not be acceptable.	Textbox removes space automatically.	Valid
	Does the textbox allow entering “Spaces” in between the verification code?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606CAB67EB6 4C653783a1e5c	Data should not be acceptable.	Data is not acceptable.	
TV0110.	Does the textbox accept only spaces as input?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code:	Data should not be acceptable.	Data is not acceptable.	Valid
	Does the textbox allow any special characters?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606CAB67EB64C65 3783a1e(c)	Data should not be acceptable.	Prompts Invalid Code error.	

TV0112.	<p>Does the textbox allow any special characters at the start of the verification code?</p> <ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	<p>Code: _06CAB67EB64C65 3783a1e5c</p>	<p>Data should not be acceptable.</p>	<p>Prompts Invalid Code error.</p>	Valid
TV0113.	<p>Does the textbox allow any special characters at the end of the verification code?</p> <ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	<p>Code: 606CAB67EB64C65 3783a1e5_</p>	<p>Data should not be acceptable.</p>	<p>Prompts Invalid Code error.</p>	Valid
TV0114.	<p>Does the textbox allow the user to copy verification code?</p> <ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Select the text. 5. Press Ctrl+C 	<p>Code: 606CAB67EB64C65 3783a1e5c</p>	<p>Should allow coping data.</p>	<p>Allows coping data.</p>	Valid

Guidelines

- TV0115. Do the textbox be selected from the Keyboard "tab " Button?
Yes.
- TV0116. Do the cursor blink when clicking in the textbox?
Yes.
- TV0117. Do the value visible when typed to the textbox?
Yes.
- TV0118. Does the textbox's placeholder visible or not?
Yes.
- TV0119. Does the textbox is null by default?
Yes.
- TV0120. Does the height and alignment of text boxes are the same throughout the site?
Yes.

TV02. Button (Verify)*Table 7.8 - Verify Certificate Button Validation Testing*

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Valid/Invalid
TV0201.	Does it verify the certificate on the valid certificate ID?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606cab67eb64c65378 3a1e5c	Certificate should be displayed.	Displays the certificate.	Valid
TV0202.	Does it verify the certificate on the invalid certificate ID?	<ol style="list-style-type: none"> 1. Go to site: https://mr-devs.ml/ 2. Click on Verify Certificate. 3. Enter 24-digit Code. 4. Click Verify. 	Code: 606cab67eb64c65378 3a1e5s	Should prompt Invalid code error.	Prompts an invalid code error.	Valid

Guidelines

- TV0203. Does the Cursor change?
Yes.
- TV0204. Do the font size similar to the other attributes in the interface?
Yes.
- TV0205. Do the font color similar to the other attributes in the interface?
Yes.
- TV0206. Does the spelling correct?
Yes.
- TV0207. Do the button size similar to the other button?
Yes.

8. Conclusion

The goal of the project "Blockchain Based E-Certification" is to address the problems that "Certificate Issuing Organizations" are facing. The project primarily addresses the issue of certificate verification, allowing anybody to quickly verify certificates at any time.

To protect and maintain those certificates, our team determined that it was necessary to transfer the useful data into electronic data, which, of course, will require a high level of security. As a result, we created a system based on "blockchain" that allows all Certificate Issuing Organizations to be transformed into blockchain technology.

We chose blockchain technology because of its distributed and decentralized nature. When we say "distributed," we mean that each certificate will be held on numerous nodes throughout the world, which is a significantly superior form of data security. We believe that this method is secure since tampering is practically impossible.

Our technology not only ensures the security of certificates, but also allows anybody to verify certificates at any moment. We are able to do so by using Hyperledger Fabric. Hyperledger Fabric is a platform that enables the creation of Smart Contracts.

9. Future Enhancement

There can be many Future enhancement in our project there are lot of possibilities that we can add in our final year project. Starting from the user interface first of all we will focus on the designer of the certificate we will make it completely dynamic so that user can create certificates according to his/ her choice via drag and drop. Also, we will enhance our UI/ UX for better user experience.

In our application there is no feature of user's profile management so in future we will add the feature of user's profile management so that users can manage their profile. Also, we will add FAQs and Guidelines so the user can understand the system easily.

Also, we will provide developer API portal and enhance system architecture like distributed RabbitMQ Consumers, Distributed Socket Servers, Dedicated Email Servers and HLF Automator for production ready HFL network.

In future we will add the feature of payment gateway and reports in our application.

10. Achievements

We have applied in Ignite funding but we have not received any funding and due to Covid'19 we have not applied in any other competitions.

11. Appendices

11.1 Annex A: Class Diagram

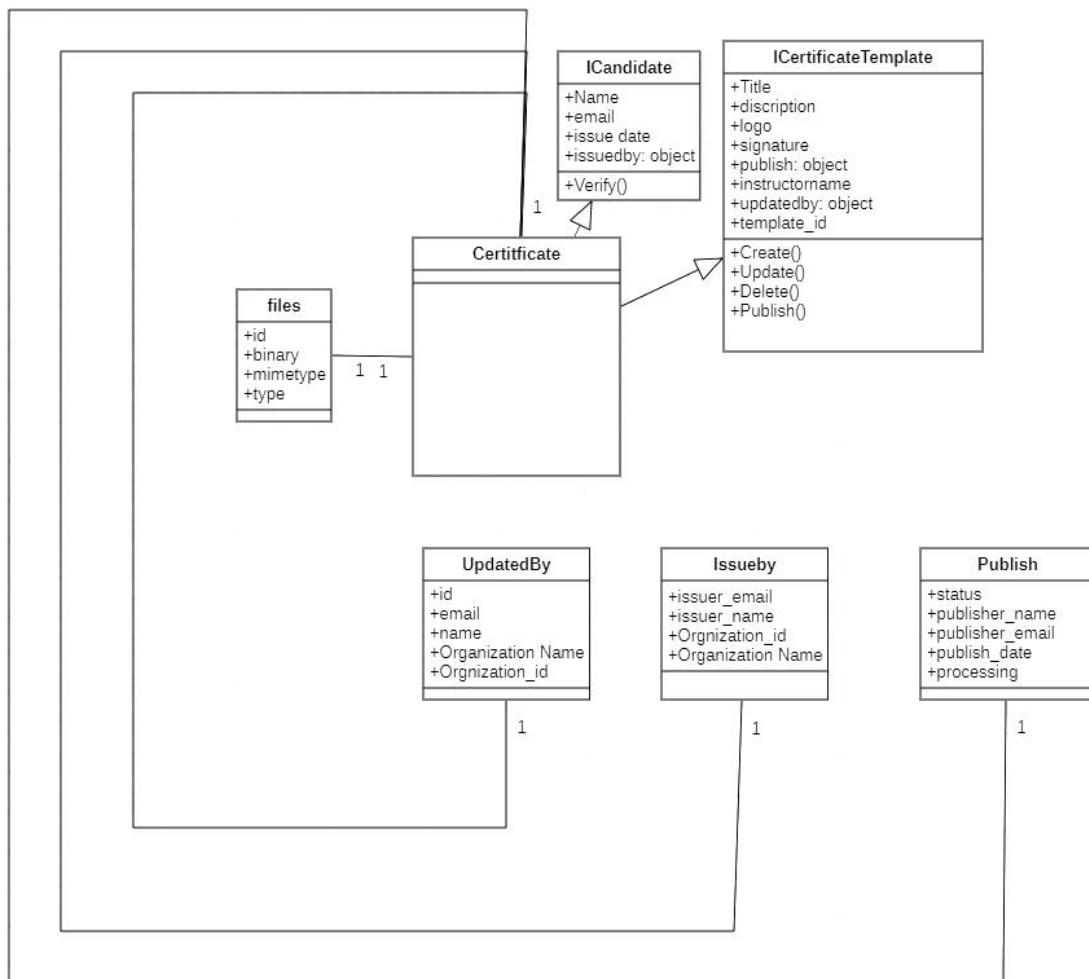


Figure 11.1 - Manage Certificate Class Diagram

The figure 11.1 shows the process of managing certificates also it shows the relationship of certificate class with all other classes.

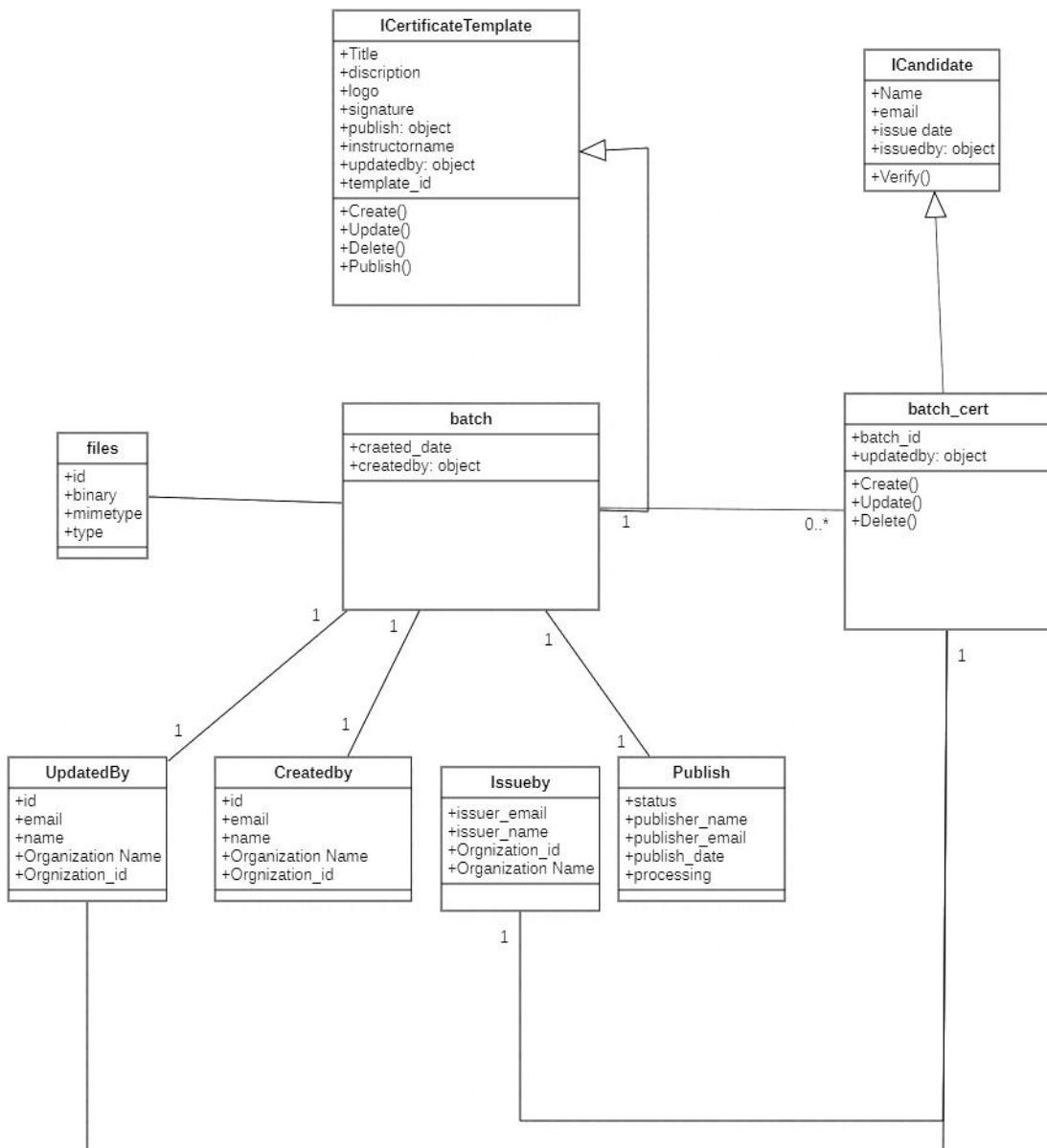


Figure 11.2 - Manage Batch and Batch Certificate Class Diagram

The figure 11.2 shows the process of managing batch and batch certificates also it shows the relationship of batch and batch certificates class with all other classes.

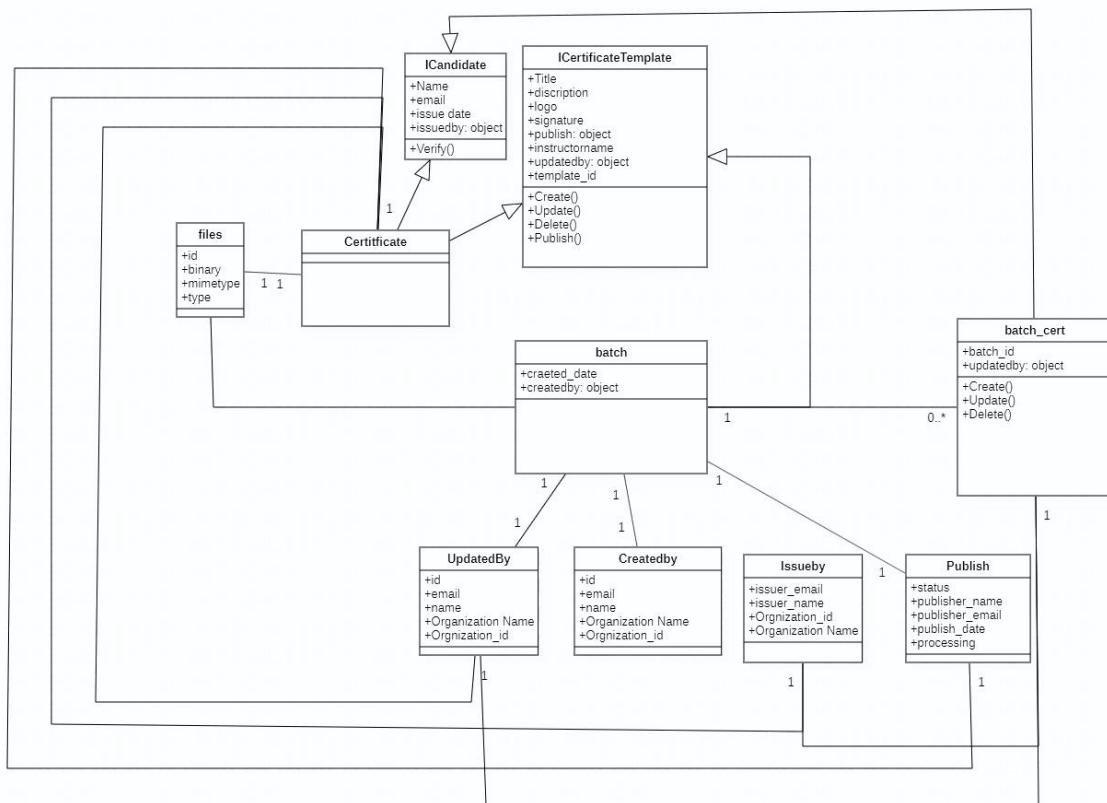


Figure 11.3 - Verify Certificate Class Diagram

The figure 11.3 shows the process of verifying certificates also it shows the relationship of all class with each other.

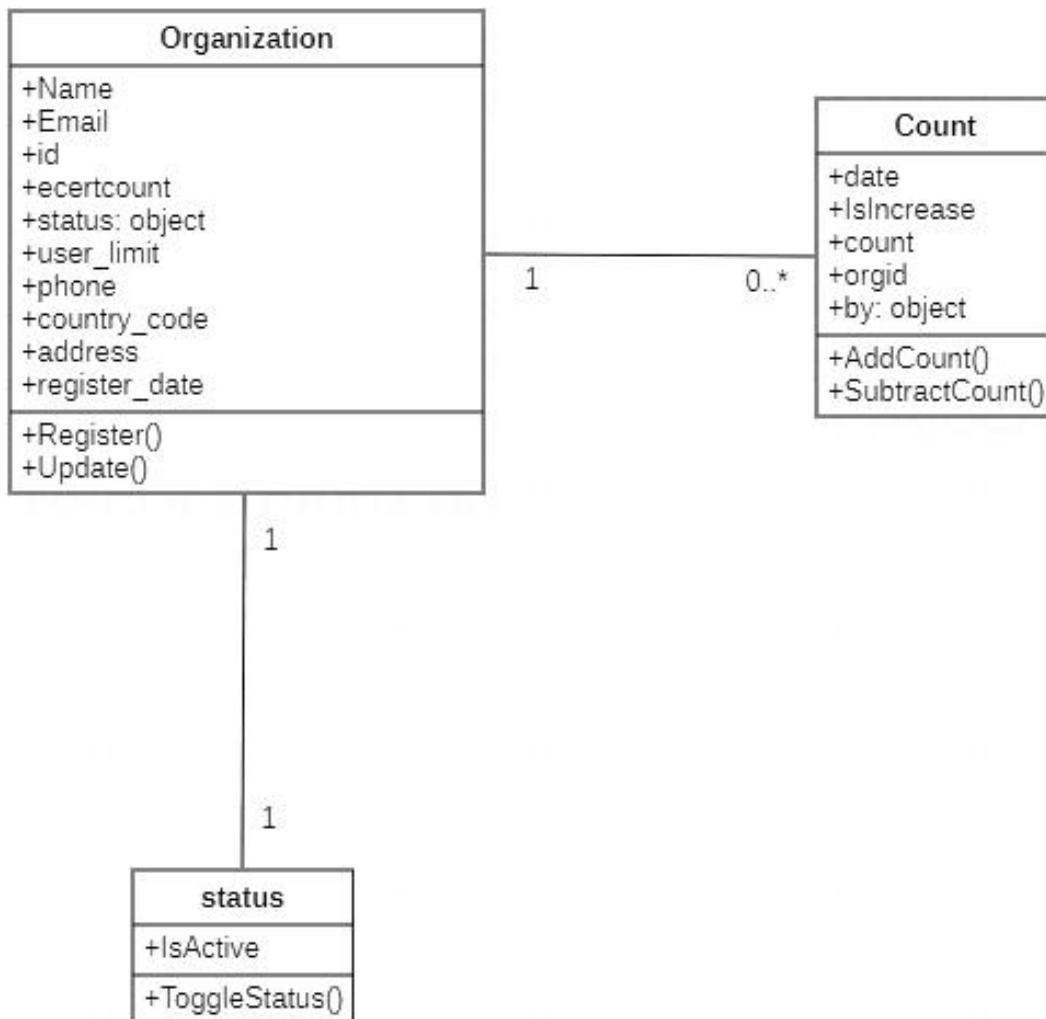


Figure 11.4 - Manage Organization Class Diagram

The figure 11.4 shows the process of managing organization also it shows the relationship of organization class with all other classes.

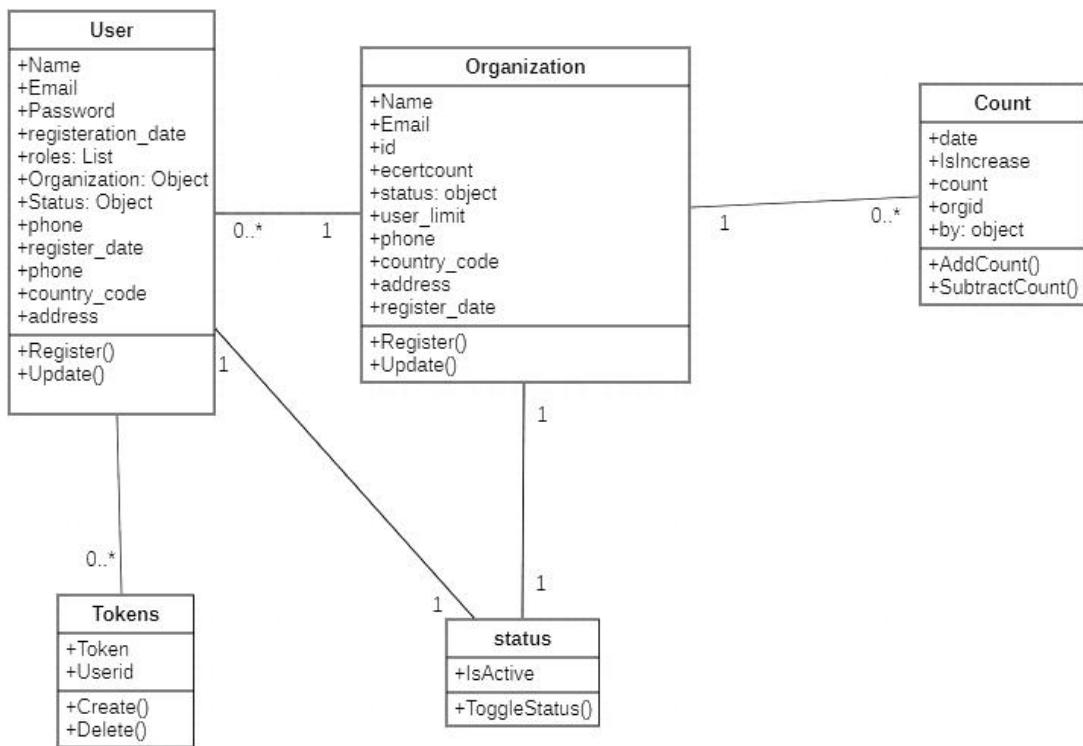


Figure 11.5 - Manage User Class Diagram

The figure 11.5 shows the process of managing user also it shows the relationship of user class with all other classes.

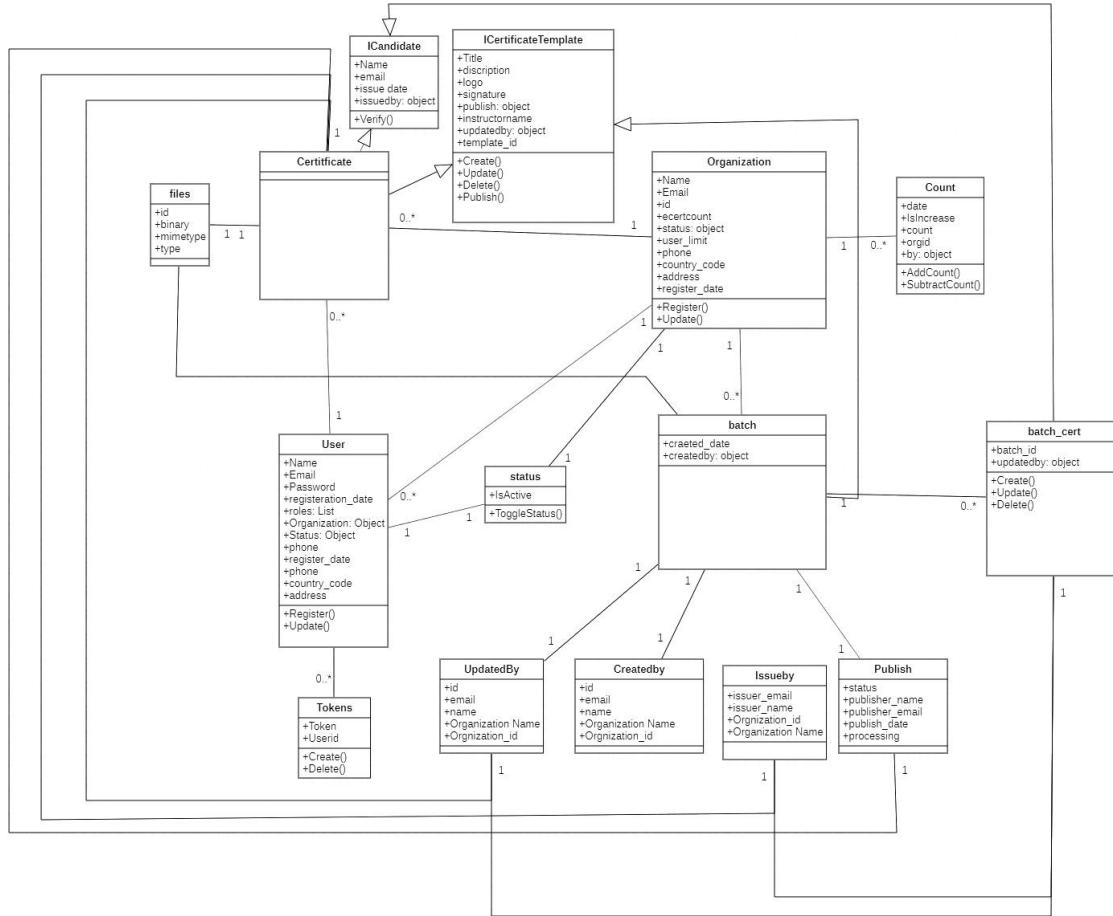


Figure 11.6 - Complete Class Diagram

The figure 11.6 shows the interaction of all classes with each other.

11.2 Annex B: Sequence Diagram

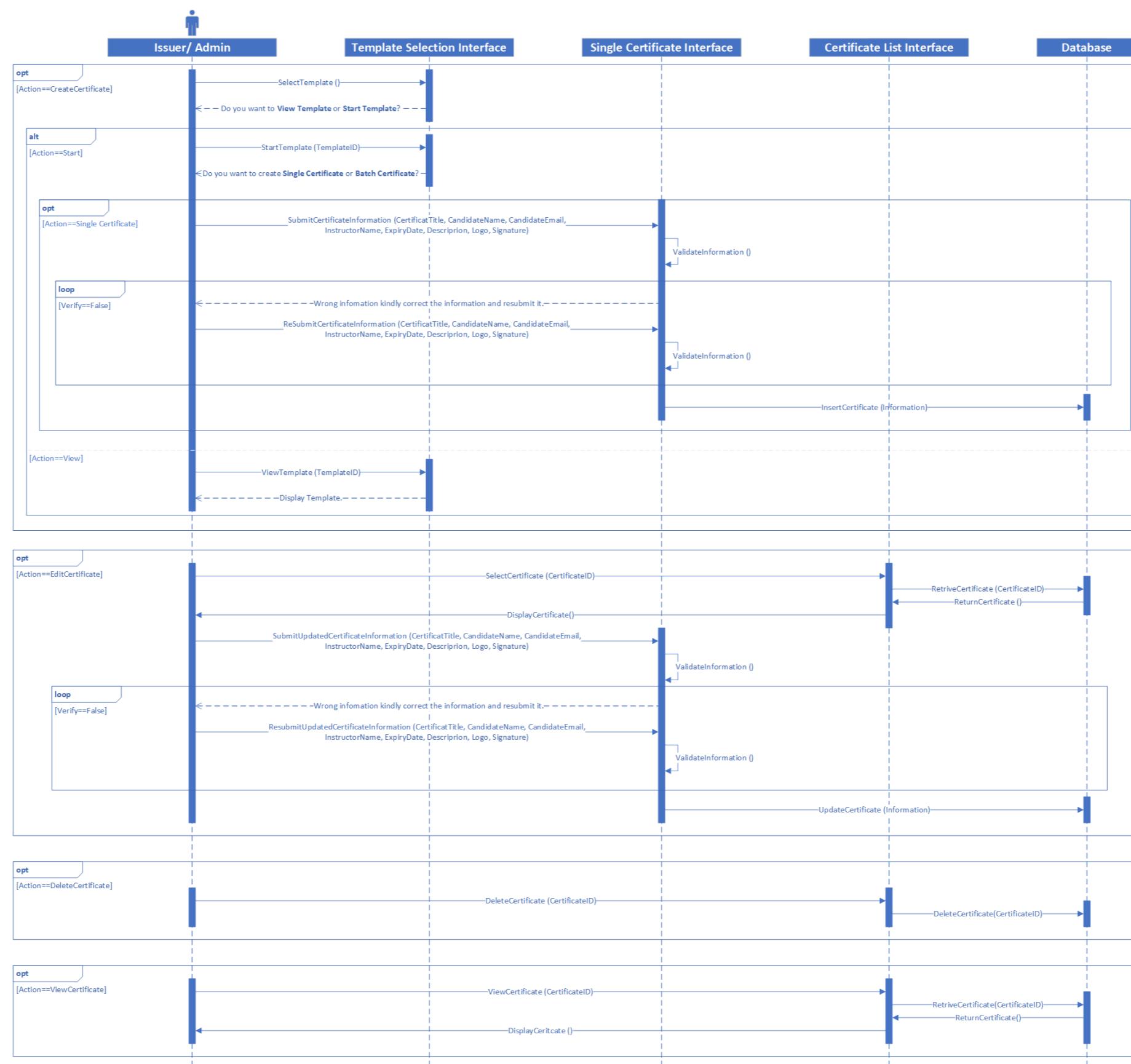


Figure 11.7 - Manage Certificate Sequence Diagram

If the user wants to create a single certificate this will provide him/her the feature of creating single certificates. By choosing a certificate template he/she will be able to create single certificates. The figure 11.7 shows the process of creating single certificates. First user need to go to the create option which is available on the navbar the system will render all the available certificate templates, user will select any of the template then he/she will be able to view or fill as per choice in case if user fills the selected template then system will ask user either they want to create single certificate or they want to creates batch for creating single certificate user needs to selects single certificate then he/she will be allowed to add all the required details of the certificate and when the user clicks on create button then system will validate the formats of all the inputs given by user if the format is correct then the system will generate the certificate else the system will prompt error for correcting the format of the inputted fields. The figure 11.7 also shows the process of edit, delete and view. Now if user want to edit, delete and view generated certificate so he/ she need to select the certificate of his/ her own choice to perform any of the mentioned action. For edit user need to select edit option to update certificate, for delete user need to press the delete button to delete the selected certificate and for view user need to press the view button for viewing the certificate.

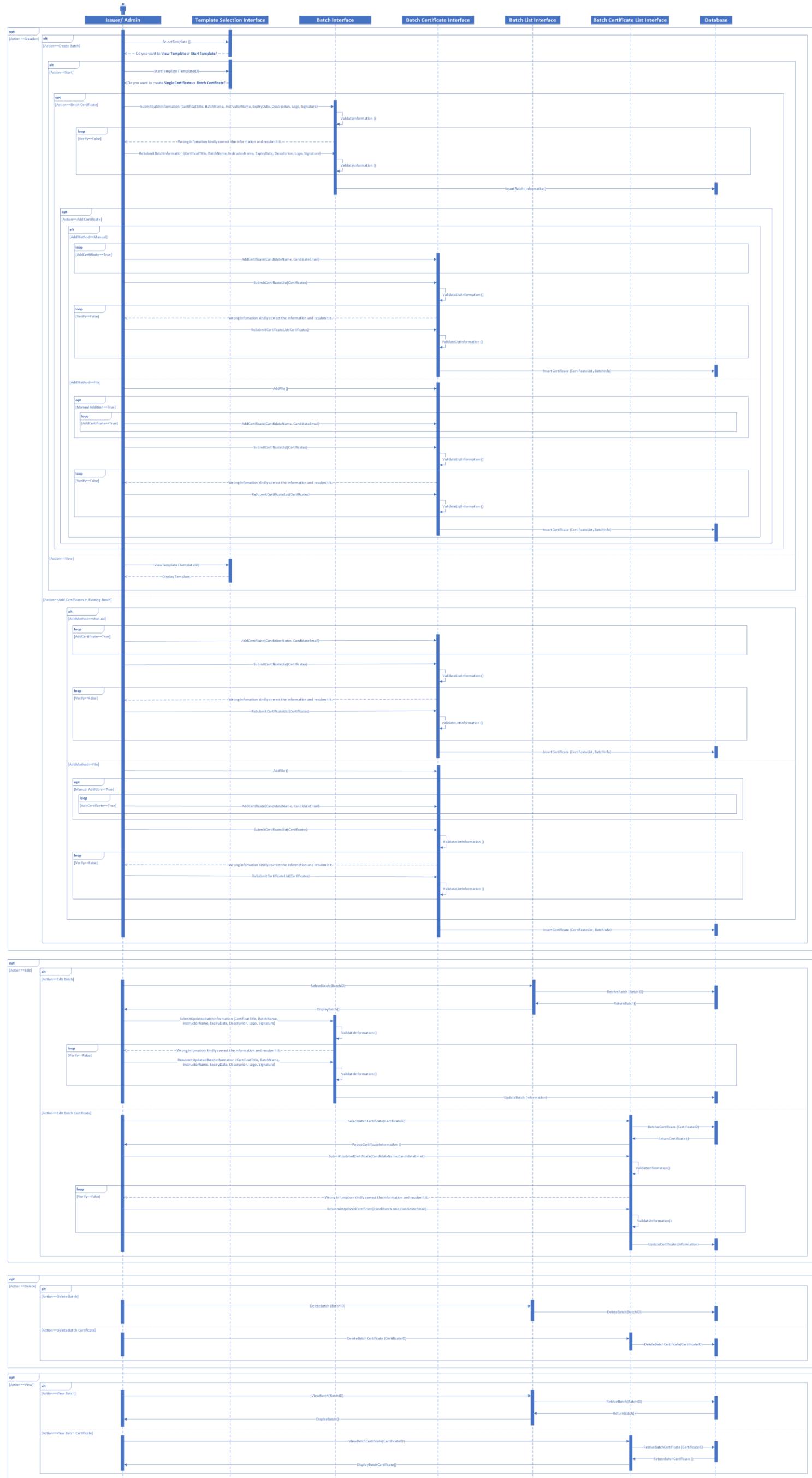


Figure 11.8 – Manage Batches/ Batch Certificates Sequence Diagram

If the user wants to create batches or wants to add certificates in the batches this will provide him/her the feature of creating batches and add certificates in the batches. In case user wants to create batch so by choosing a certificate template he/she will be able to create batches and if the user wants to add certificates in them then user can also add certificates in them. The figure 11.8 shows the process of creating batches and their certificates, first user need to go to the create option which is available on the navbar the system will render all the available certificate templates, user will select any of the template then he/she will view or fill as per choice in case if user fill the selected template then system will ask user either they want to create single certificate or they want to creates batch for creating batch user needs to selects batch then he/she will be allowed to add all the required details of the batch and when the user clicks on create button then system will validate the formats of all the inputs given by user if the format is correct then the system will generate the batch else the system will prompt error for correcting the format of the inputted fields. Once the batch has been created the system will redirect user to add certificates page if the user wants to add certificates in the batch then user will be able to add certificates in it if user don't want to add certificates then this will end the process. If the user doesn't want to create batch and just want to add certificates in the batch this will provide him/her the feature of adding certificates. By choosing the batch he/she will be able to add certificates in that batch. The figure 11.8 also shows the process of adding certificates in the batch, first user need to go to the certificate option which is available on the navbar then system will display two options single certificates and batches for adding certificates in the batches user needs to select batches option so system will display all the unpublished batches to the user then user will select any of the batch and will go to batch detail and when the user clicks on add button then the system will render the page to add certificates, now user will be having multiple options of adding certificate one is manual addition and the second is add certificates by file if the user wants to add manually then he/she can add certificates manually and if user wants to add certificates using file then he/she can upload the excel file then the system will retrieve certificates data from the file also if the user wants to add more certificates after uploading the file then he/she can add them manually once the user clicks on submit button then the system will validate the detail of all the certificate if all certificates' detail is correct then the system will add those certificates to the batch if any of those certificates' detail is invalid so system will prompt an error to the user for correcting the detail. The figure 11.8 also shows the process of edit, delete and view of batches and their certificates. For edit user need to select edit option either for batch or for their certificate, for delete user need to select delete option either for batch or for their certificate and for view user need to select view option either for batch or for their certificate.

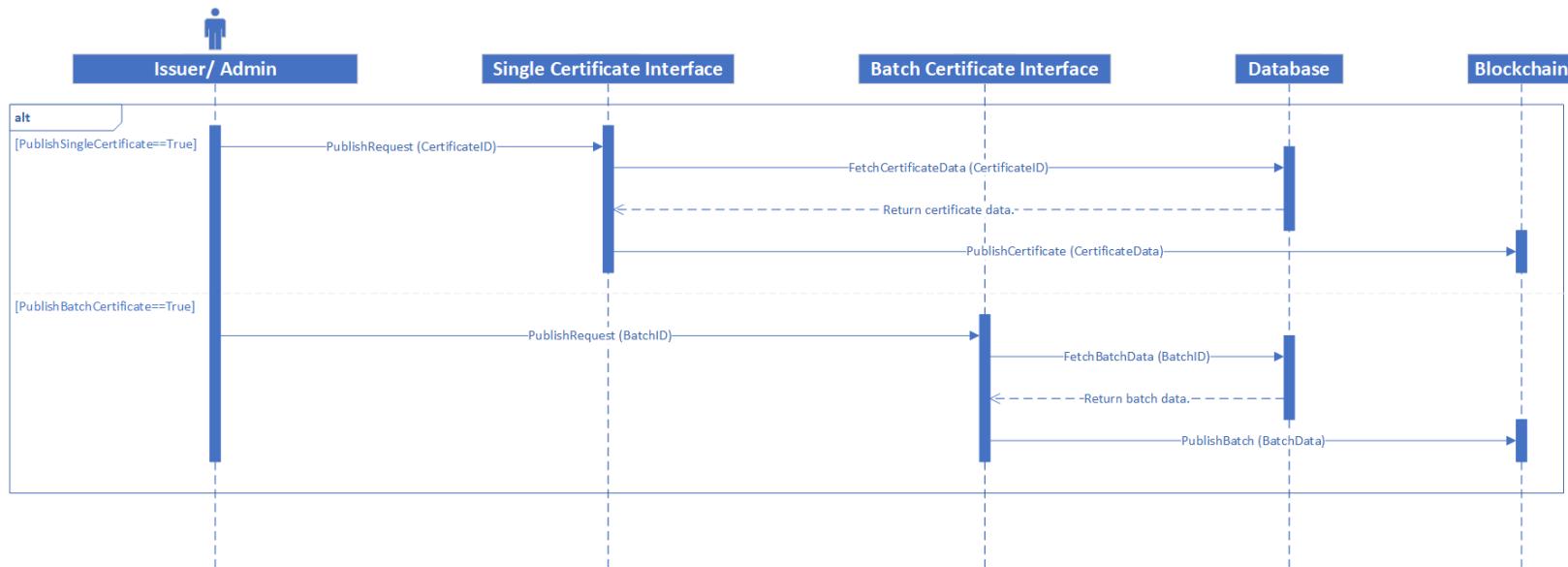


Figure 11.9 - Certificate Publication Sequence Diagram

If the user wants to publish single certificates or batches to the blockchain the this will allow users to publish single certificates or batches to the blockchain after publishing, all the certificates will be verifiable. The figure 11.9 shows the process of publishing single certificates or batches to the blockchain, first user need to go to the certificate option which is available on the navbar then system will display two options single certificates and batches, incase if user selects single certificates option so system will display all the unpublished single certificates to the user then user will select any of the single certificate and click on publish button to publish that single certificate to the blockchain and if user selects batches option so system will display all the unpublished batches to the user then user will select any of the batch and click on publish button to publish that batch to the blockchain.

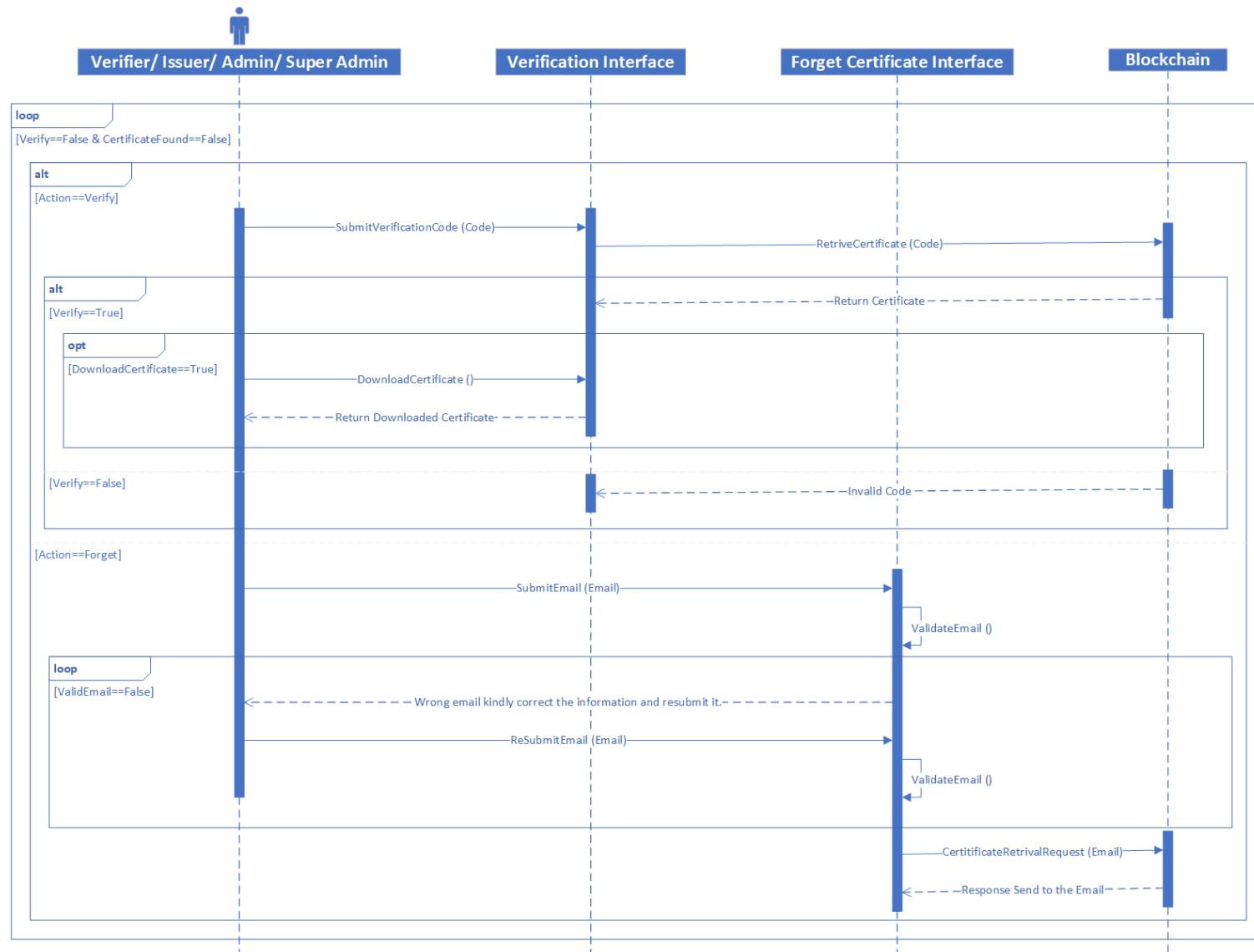


Figure 11.10 - Verify/ Recover Certificates Sequence Diagram

If the user wants to verify certificates this will allow users to verify the certificates that have been published to blockchain. The figure 11.10 shows the process of verifying certificates either in batch or single certificates, first user needs to go to the verify option which is available on the navbar then system will render verification page if the user wants to verify the certificate he/she needs to enter the verification code if the code is valid then certificates will be displayed to the user and if user wants to save the certificates then the user can download that certificate and in case if the code is invalid then the system will prompt error of invalid code to the user also if user forgets the certificate code then user can recover those certificates by clicking on forget certificate which will render the forget certificate page to recover certificates now user needs to enter his/her email address, once user submit the request, the system will validate the email if the email is valid then all the certificates against that email will be sent to that email address.

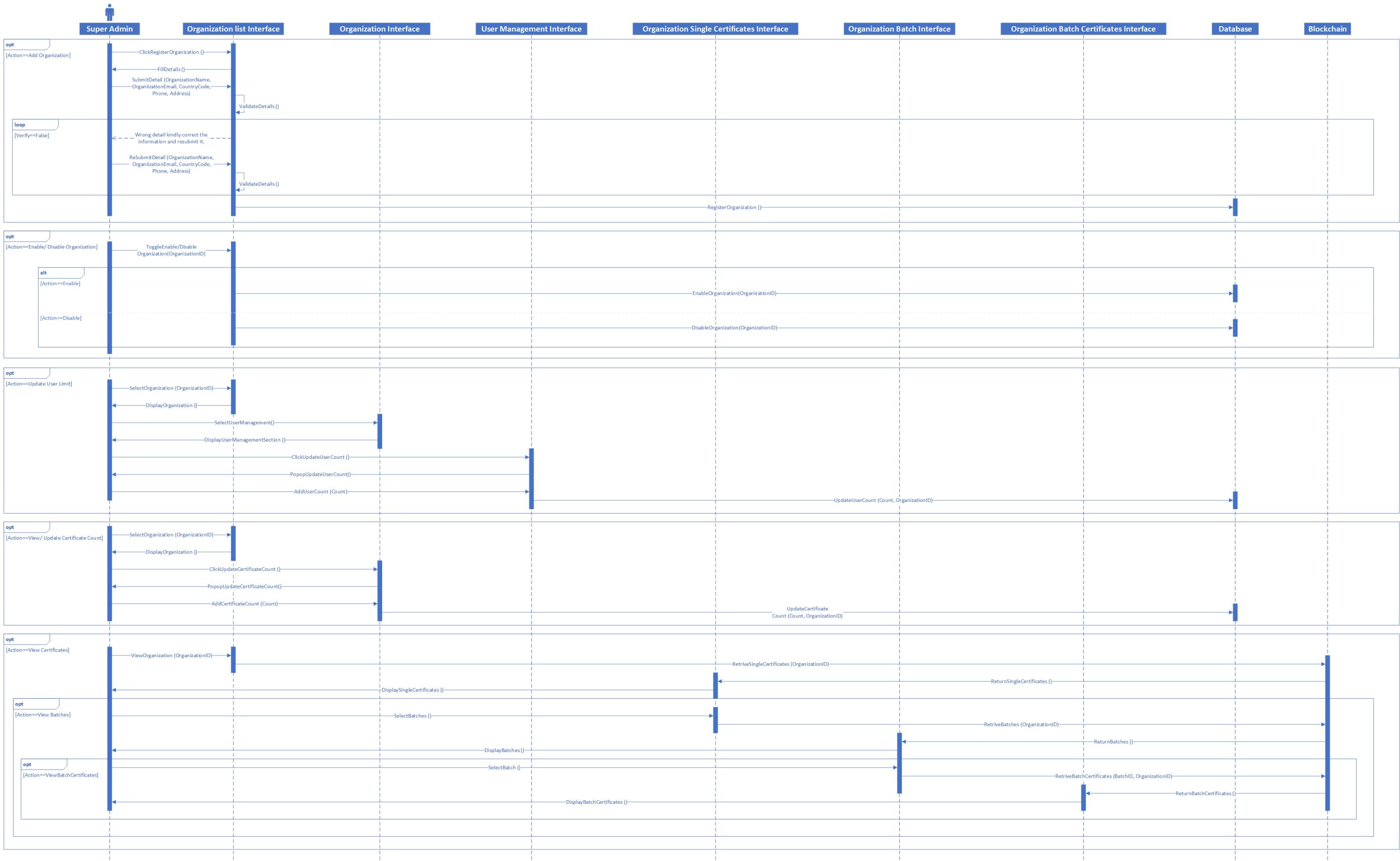


Figure 11.11 - Manage Organization Sequence Diagram

If the user wants to register new organization. So, figure 11.11 shows the process of register organization, first user needs to go to the organization option which is available on the navbar then system will render organization page if the user wants to register organization user need to enter the detail of organization when the user submits the detail of the organization the system will validate the detail of the organization if the detail is valid then the system will register the organization if the detail is invalid then the system will prompt user to correct the detail. The figure 11.11 also shows the process of enable/ disable organization, update user limit, view/ update certificate count, view certificate and view batch. For enable/ disable organization user need to go to the organization list interface and toggle enable/ disable organization switch, for update user limit user need to select organization from the organization list and go to the user management interface then update user limit, for view/ update certificate count user need to select organization from the organization list this will show the old count history and by clicking update count user can update the certificate count, for view certificates and batches user need to view the organization from the organization list this will show all the published certificates of the client organization.

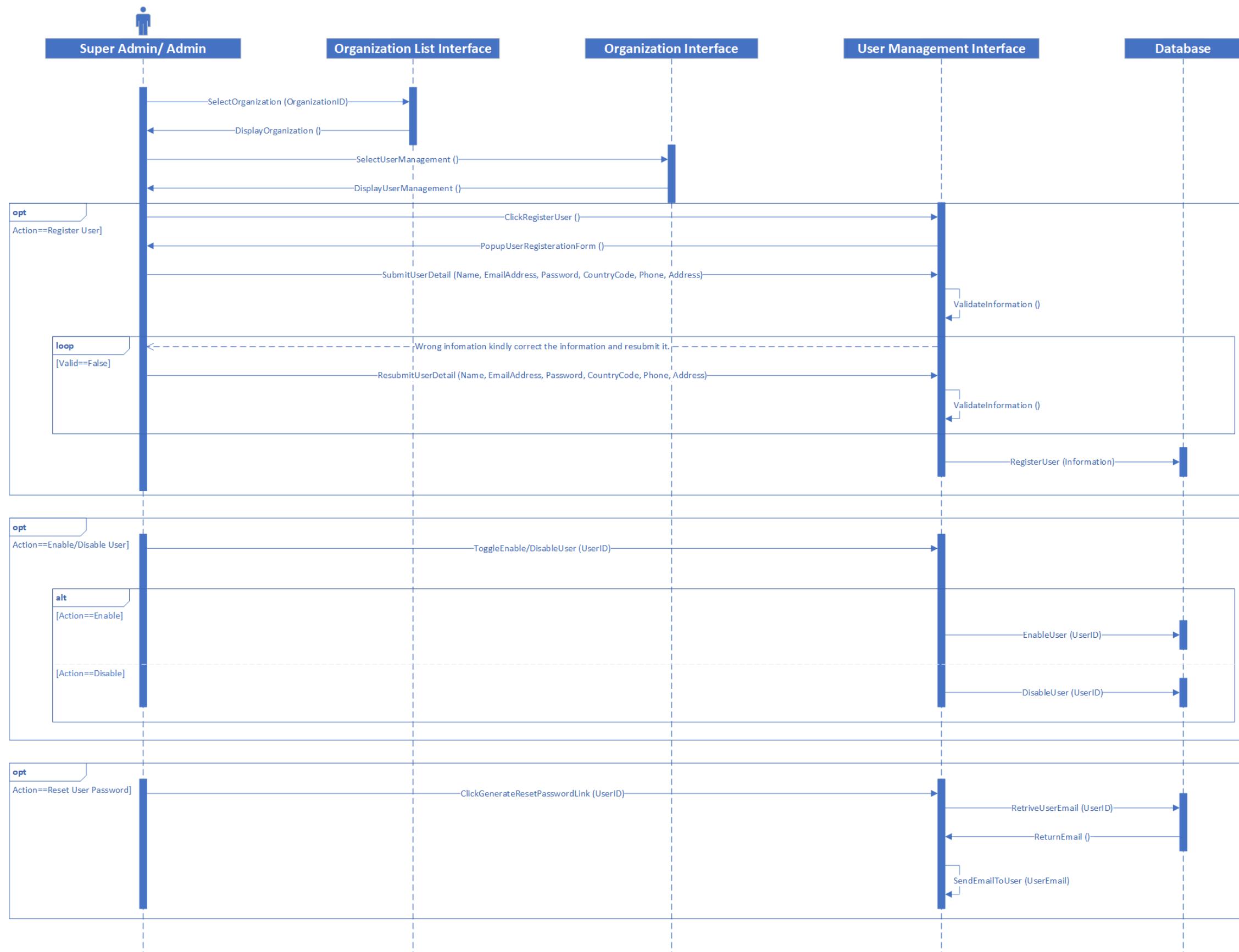


Figure 11.12 - Manage User Sequence Diagram

The Figure 11.12 shows the process of user management that contain the process of registering user, enable/ disable user and reset user password. For registering user, user need to go to the organization in the user management section register user allow to create user and by providing detail user can register user. For enable/ disable user, user need to toggle enable/ disable user switch in the same interface to enable/ disable user and for reset password link generation user need to just click on the reset password button in the same interface.

11.3 Annex C: Object Diagram

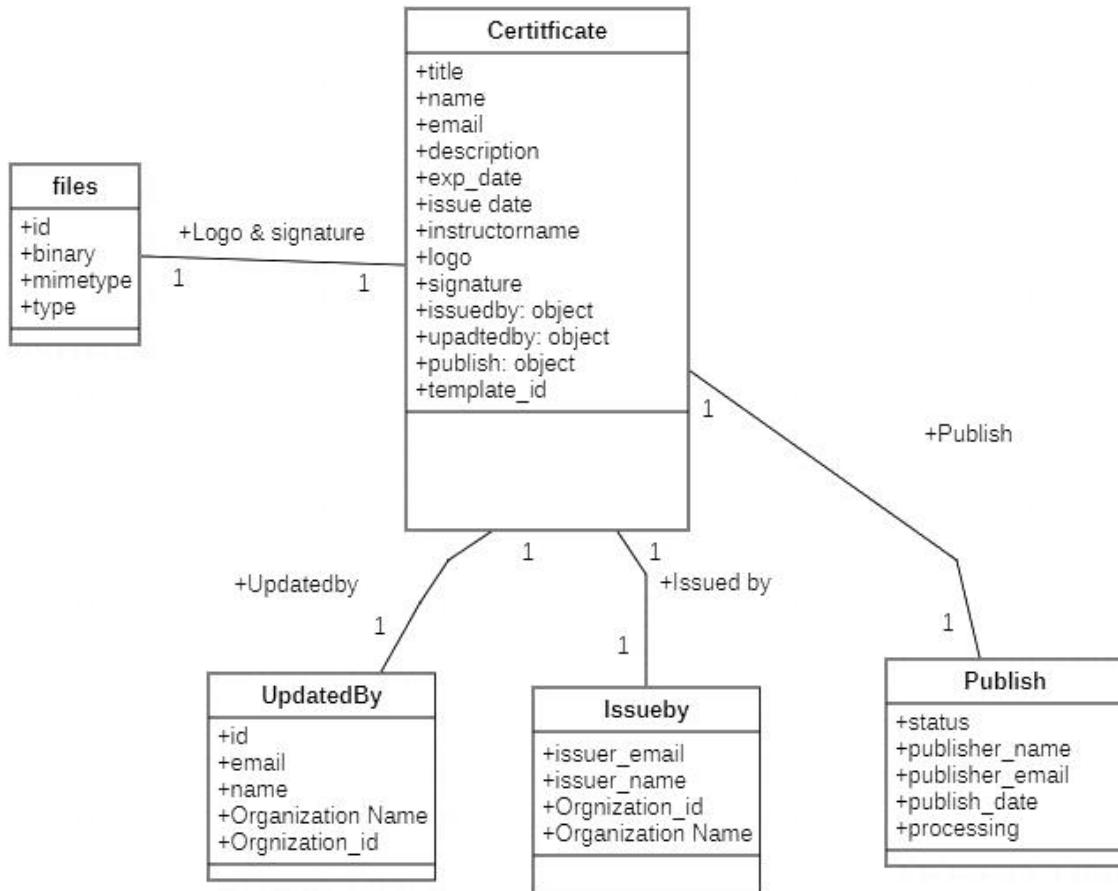


Figure 11.13 - Manage Certificate Object Diagram

The figure 11.13 shows the process of managing certificate also it shows the relationship of certificate object with all other objects.

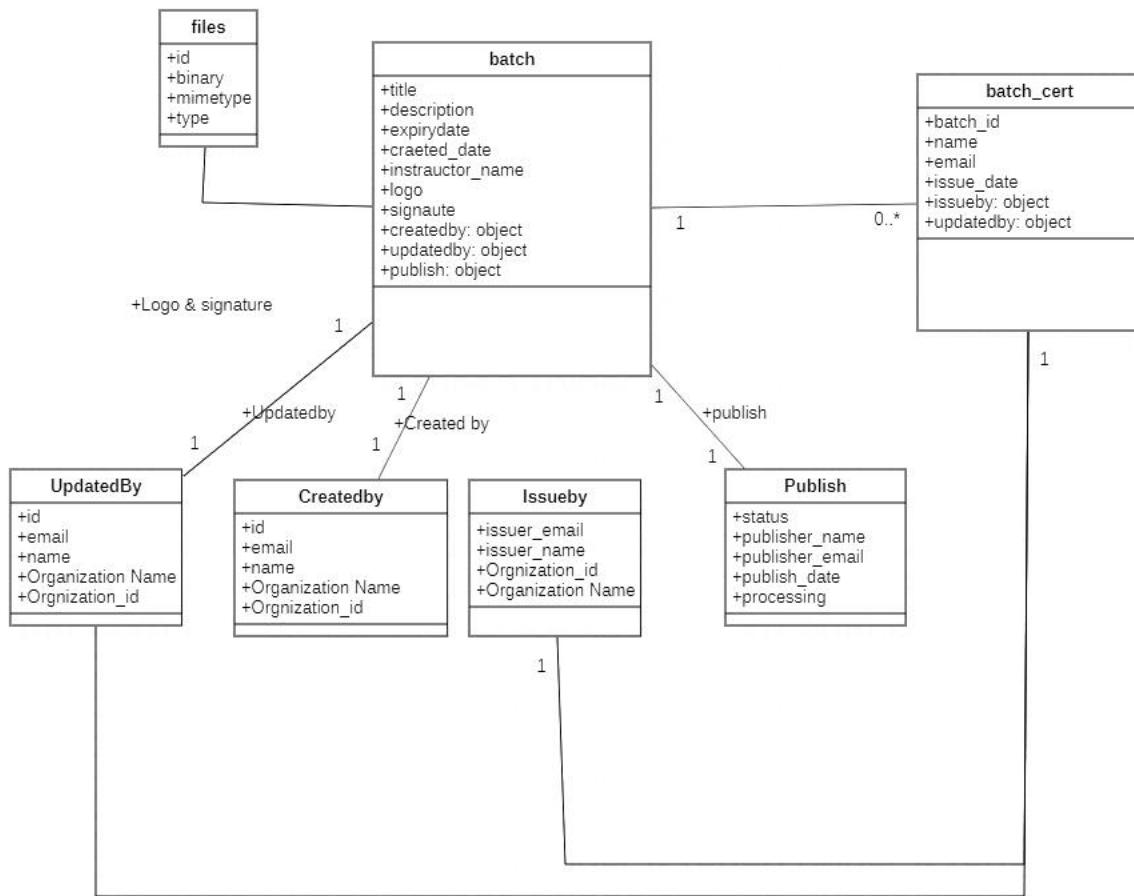


Figure 11.14 - Manage Batch and Batch Certificates Object Diagram

The figure 11.14 shows the process of managing batch and batch certificate also it shows the relationship of batch and batch certificate object with all other objects.

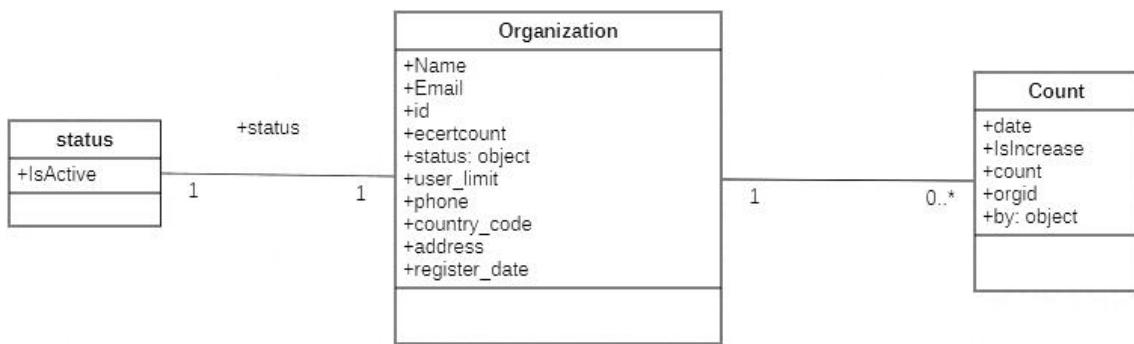


Figure 11.15 - Manage Organization Object Diagram

The figure 11.15 shows the process of managing organization also it shows the relationship of organization object with all other objects.

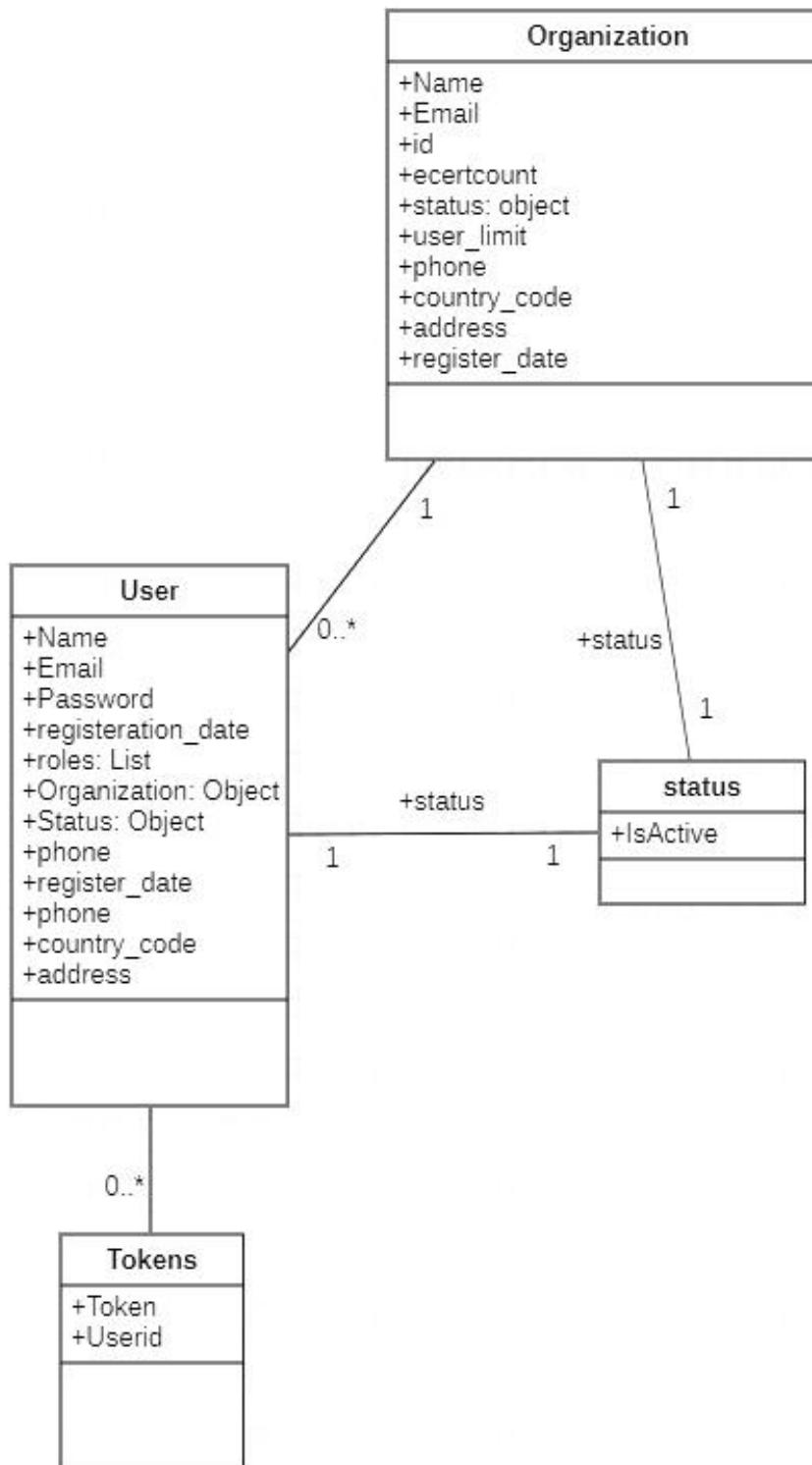


Figure 11.16 - Manage User Object Diagram

The figure 11.16 shows the process of managing user also it shows the relationship of user object with all other objects.

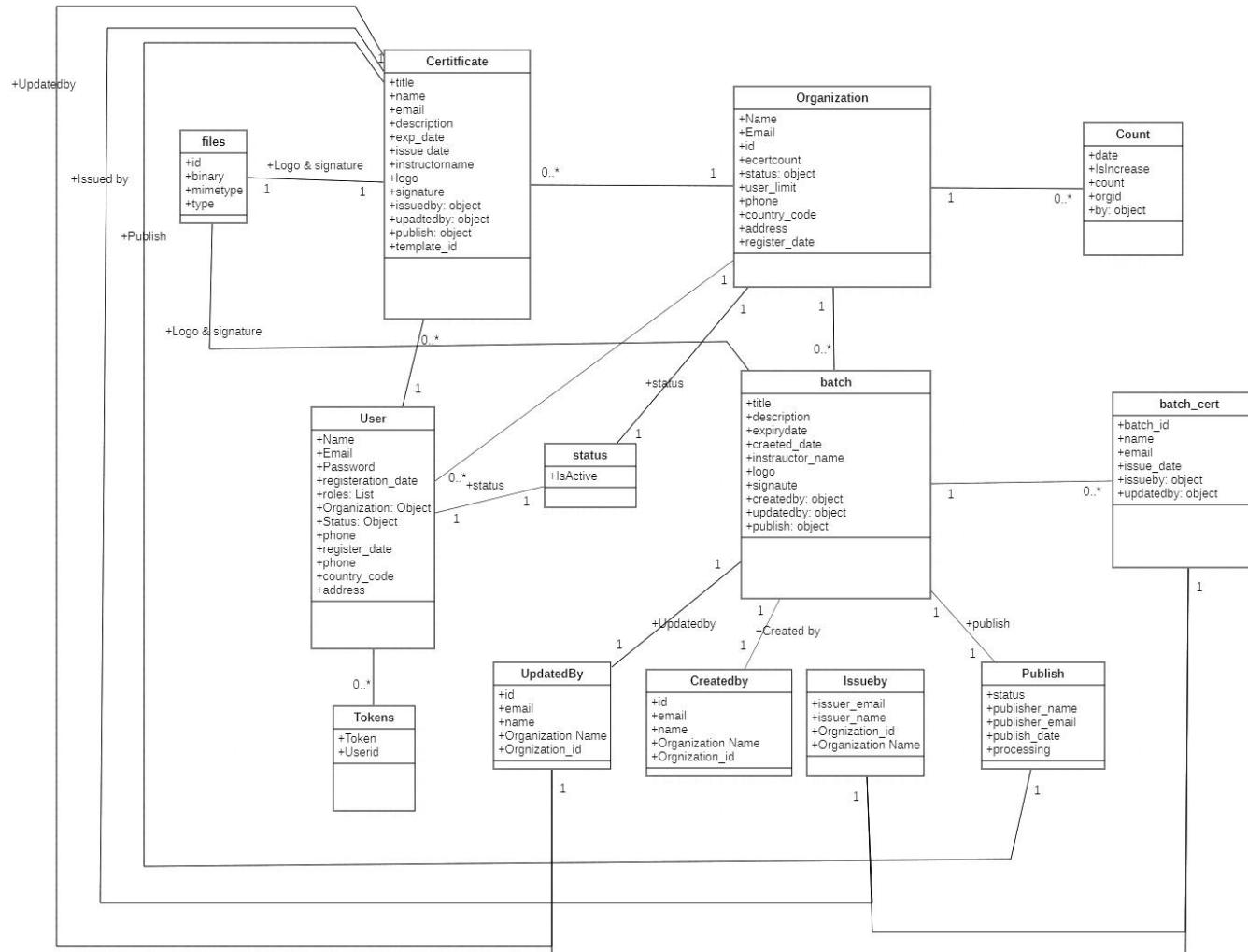


Figure 11.17 - Complete Object Diagram

The figure 11.17 shows the interaction of all objects with each other.

12. References

- [1] Certification Importance, www.aarm.org/certification_importance.html#:~:text=Certification%20has%20found%20its%20way,enhance%20their%20skills%20and%20knowledge.
- [2] “What Are Distributed Ledger Technologies?” Hedera Hashgraph, hedera.com/learning/what-are-distributed-ledger-technologies-dlts?gclid=CjwKCAiAnvj9BRA4EiwAuUMDf1GdDYHz0AhHsnzidNKPWu_unJcQyKhr5vaUHNdlLwFHQDkMVdf4QBoCel0QAvD_BwE.
- [3] “Introduction.” Hyperledger, hyperledger-fabric.readthedocs.io/en/release-2.2/blockchain.html.
- [4] “Hyperledger Fabric Model.” Hyperledger, hyperledger-fabric.readthedocs.io/en/release-2.2/fabric_model.html.
- [5] Conway, Luke. “Blockchain Explained.” Investopedia, Investopedia, 18 Nov. 2020, www.investopedia.com/terms/b/blockchain.asp.
- [6] “Ethereum.” Wikipedia, Wikimedia Foundation, 6 Jan. 2021, en.wikipedia.org/wiki/Ethereum.
- [7] “Hyperledger.” Wikipedia, Wikimedia Foundation, 8 Dec. 2020, en.wikipedia.org/wiki/Hyperledger.
- [8] “Enterprise Blockchain Platform: Corda Enterprise by R3.” R3, www.r3.com/corda-enterprise/.
- [9] “Ripple (Payment Protocol).” Wikipedia, Wikimedia Foundation, 9 Jan. 2021, [en.wikipedia.org/wiki/Ripple_\(payment_protocol\)](https://en.wikipedia.org/wiki/Ripple_(payment_protocol)).
- [10] Author , and Blockgeeks. “What Is Quorum Blockchain? A Platform for The Enterprise.” Blockgeeks, 14 May 2020, blockgeeks.com/guides/quorum-a-blockchain-platform-for-the-enterprise/.
- [11] Frankenfield, Jake. “Hyperledger Sawtooth Definition.” Investopedia, Investopedia, 16 Sept. 2020, www.investopedia.com/terms/h/hyperledger-sawtooth.asp.
- [12] “Blockchain Software Architecture.” EOSIO, 15 Dec. 2020, eos.io/.
- [13] “Hyperledger Iroha.” Hyperledger, 27 Apr. 2020, www.hyperledger.org/use/iroha.
- [14] “Stellar (Payment Network).” Wikipedia, Wikimedia Foundation, 7 Jan. 2021, [en.wikipedia.org/wiki/Stellar_\(payment_network\)](https://en.wikipedia.org/wiki/Stellar_(payment_network)).
- [15] “Distributed Architecture.” *Tutorialspoint*, www.tutorialspoint.com/software_architecture_design/distributed_architecture.htm.