TUNIS BUSINESS SCHOOL
UNIVERSITY OF TUNIS

UNIVERSITY OF TUNIS
TUNIS BUSINESS SCHOOL

UNIVERSITÉ DE TUNIS
جامعة تونس

Graduation Final Project Report

As a partial fulfillment of the degree of

BACHELOR DEGREE IN BUSINESS ADMINISTRATION

MAJOR : INFORMATION TECHNOLOGY

MINOR : BUSINESS ANALYTICS

Elaborated by

Mechergui Rami

DECENTRALIZED FUNDRAISING PLATFORM

Academic Advisor: DR. BEN MESSOUAD MONTASSAR
Professional Advisor: MR. BEN MILED MONDHER

Hosted by: INFINITY MANAGEMENT

INFINITY MANAGEMENT

Tunis, September 2023

Dedicated,

To my parents,

To my sisters,

and everyone I love.

Abstract

A Decentralized Fundraising Platform (DFP) serves as a dynamic tool that enables startups and investors to engage in tokenization strategies, fostering increased financial returns. One of the primary advantages of utilizing this platform stems from its capacity to provide users with accessible and comprehensible data. This functionality empowers users to construct valuable insights, stay attuned to shifting trends, and enact optimal decisions concerning token pricing and inventory management.

The purpose of this internship is to explore a decentralized application based on Ethereum blockchain. The primary goal is to transform the current startup fundraising process by linking startups with a diverse pool of investors and allowing investors to uncover appealing opportunities in an efficient, secure, and transparent manner.

Key Words: Decentralized Fundraising, Blockchain Technology , Ethereum ,Tokenization , Startups, Investors, Capital Access, Investment Opportunities

ACKNOWLEDGEMENTS

This report is the result of a rewarding internship at Infinity Management during my senior year. That pleasure stems from my interactions with my managers, colleagues, and coworkers.

I'd like to express my heartfelt gratitude to my supervisor, Mr. Mondher Ben Miled, System Architect, for providing kind guidance, support, and counsel throughout my internship. He always pays close attention to my questions and helps me solve any problems that arise.

Dr. Montassar Ben Massaoud, my academic advisor, deserves special recognition for his help, advice, and encouragement. Thank you for pushing me out of my comfort zone.

For their unflinching support, I will be eternally thankful to my family and my friends.

Contents

List of Figures

List of Tables

List of keywords

API	=	**A**pplication **P**rogramming **I**nterface
DPoS	=	**D**elegated **P**roof **O**f **S**take
DeFi	=	**De**centralized **Fi**nance
EVM	=	**E**thereum **V**irtual **M**achine
Geth	=	**G**o **E**thereum
KYC	=	**K**now **Y**our **C**ustomer
PoA	=	**P**roof **O**f **A**uthority
PoS	=	**P**roof **O**f **S**take
PoW	=	**P**roof **O**f **W**ork
dApp	=	**D**ecentralized **A**pplication

General Introduction

In the dynamic and ever-changing sector of information technology, the introduction of blockchain technology has sparked enormous curiosity and disruptive possibilities.Decentralized blockchain platforms like Ethereum have paved the way for innovative applications and solutions across a wide range of sectors.Among these, the application of blockchain in startup finance and investing is particularly appealing.

The purpose of this project is to explore a decentralized application (dApp) based on the Ethereum blockchain. The primary goal is to transform the current startup fundraising process by linking startups with a diverse pool of investors and allowing investors to uncover appealing opportunities in an efficient, secure, and transparent manner.

The main idea is to design, develop, and test the efficacy of a custom-built dApp designed specifically for startups and investors. The dApp will employ Ethereum's smart contract capabilities to ensure the security and confidence of both parties. Through this project, we seek to gain a better understanding of the possible benefits and drawbacks of employing blockchain technology in the context of startup fundraising, as well as its impact on traditional fundraising model.

Chapter 1

General Presentation

Plan

1.1 Introduction

This chapter attempts to give a comprehensive overview of our project, including a wide range of topics relevant to its scope. It will look into the significance of different situations.

1.1.1 Academic Context

This project is the product of an end-of-studies internship completed as part of the requirements for a Bachelor Degree in Business Administration. From February 8th to August 8th, 2023, we had the honor of collaborating with the prestigious Infinity Management team.

1.1.2 Host Organization

Infinity Management provides consultancy and IT engineering services. It assists clients in conceptualizing and implementing the digital era, understanding their objectives, and providing solutions tailored to their strategic development through branches in Tunisia, France, and Spain. Its mission is to assist clients in building business-oriented solutions through the use of our experienced and highly qualified analysts and architects. Its brand is founded on three pillars: consultancy, human resources, and innovation.

Figure 1.1: Infinity Management Logo

1.1.2.1 Activities and areas of expertise

Our hosting company serves a wide range of industries, offering a variety of solutions in fields such as industrial, automotive, banking, finance, insurance, telecommunications, and more.

1.2 Study of The Existing

This section provides an in-depth study of today's startup financing model. Due to liquidity difficulties, many businesses struggle to find potential investors willing to fund their business projects. In contrast, a large proportion of investors actively seek out new and appealing investment opportunities to diversify their portfolios. While traditional fundraising strategies are well-established, they can create roadblocks for entrepreneurs, restricting their growth potential and access to a diverse pool of investors. Furthermore, such restrictions can have a significant impact on any country's economy by suppressing potential new initiatives, impacting job creation and overall economic advancement.

1.3 Description and Criticism of The Existing

We delve deeper into the weaknesses of the traditional fundraising approach, emphasizing the time-consuming and resource-intensive processes that entrepreneurs must endure. Traditional fundraising methods can include large administrative burdens, complex legal frameworks, and high transaction costs, making it difficult for startups, particularly those in their early stages, to attract investors efficiently. Furthermore, the lack of access to a global network of investors limits the opportunities available to enterprises. Potentially disruptive projects may struggle to get the necessary financing to realize their ideas.

1.4 Proposed Solution

We suggest a revolutionary fundraising model based Ethereum blockchain. Our dApp, which harnesses the power of blockchain and smart contracts, provides businesses with a transparent, safe, and efficient platform for launching fundraising campaigns. the platform also enables investors from around the world to locate and invest in companies that meet their investment criteria.

With our Ethereum-based dApp, startups get access to a wider pool of potential investors, allowing them to solicit money more effectively and with less administrative overhead. This increased accessibility and inclusivity supports a strong entrepreneurial ecosystem, promoting the growth of businesses.

1.5 Market Trends in Fundraising Platforms

The market of fundraising platforms is characterized by a diverse array of actors, each contributing to its dynamic landscape. These actors include traditional nonprofit organizations that have adapted to online fundraising like Crypto for Charity, crowdfunding platforms like innmind and CroFundsMe. Additionally, blockchain-based fundraising platforms like Gitcoin and innovative decentralized finance (DeFi) projects have introduced new paradigms to the fundraising ecosystem. With this multitude of players, the fundraising platform market remains vibrant, offering various options for individuals and organizations to engage with donors and supporters.

Figure 1.2: Leaders in the market of decentralized fundraising

1.6 Evaluating Competitive Fundraising Platforms

In this context,it's essential to highlight that many competitor solutions are designed to cater to a global audience, spanning multiple countries worldwide. However, these existing platforms often come with a substantial bureaucratic burden and meticulous document verification processes. In contrast, our solution aims to streamline and simplify the entry into the fundraising arena. We prioritize removing barriers and complexities, making it easier for individuals and organizations to participate in fundraising initiatives without the hindrance of extensive administrative hurdles and document checks. Our approach is geared towards facilitating a smoother, more accessible path into the world of fundraising.

1.7 Conclusion

In conclusion, this project discusses the critical liquidity issues that startups face as well as investors' hunt for investment opportunities. To address these issues, we provide a modern fundraising approach based on the Ethereum blockchain. Our dApp aims to improve startup fundraising by leveraging blockchain technology, giving firms greater access to capital while offering investors diverse and transparent investment opportunities.

Chapter 2

Study and Project Planning

Plan

2.1 Introduction

The study and planning step is intended to determine the various system functionalities. Indeed, in this chapter, we first introduce the actors in our system. The investigation of functional and non-functional demands follows. These requirements will be expressed in the form of use case diagrams, which will outline the various scenarios that the various actors can carry out. Finally, we'll look at the global class diagram.

2.2 Identification of actors

Any interactive system must ensure and facilitate user participation. An actor plays the role of an external entity that operates the system via its numerous interfaces. We define the system actors who are relevant to this project, which are grouped as follows:

Main Actors :

- Token Issuer
- Investor
- Administrator

Secondary Actors :

- System

2.3 Identification of needs

In this section, we will expose the functional and non-functional needs of our application.

2.3.1 Functional needs

1.Token Issuer functional needs :

1. **User Registration :** Token Issuer should be able to register on the platform by providing necessary details such as name, contact information.

2. **Token Creation:** Token Issuer should have the ability to create and issue their own tokens on the platform. They should be able to define parameters such as token name, symbol, total supply, and any additional metadata or attributes associated with the tokens.

3. **Token Sale Configuration :** The Token Issuer should be able to configure the token sale event They should be able to set the fundraising amount they aim to achieve and define the number of tokens that will be available for sale during the fundraising event.

4. **Concept Presentation :** The Token Issuer should be able to present their startup concept on the platform.

2.Investor Functional Needs:

1. **User Registration**: Investors should have the ability to register on the platform by providing necessary details such as name, contact information.

2. **Token Discovery** : Investors should be able to explore and discover various startups and their associated tokens on the platform.

3. **Concept Review** : Investors should be able to view detailed information about each startup and their concept on the platform.

4. **Investment Process** : Investors should have the capability to participate in the token sale event.They should be able to purchase tokens from the startups they are interested in.

3.Administrator Functional Needs:

1. **User Registration and Management** Administrators should also be responsible for managing user accounts, including approving new registrations, suspending or banning accounts when necessary, and handling account deletion requests.

2. **Token Issuer Verification** Administrators should have the capability to verify the identity and authenticity of token issuers before their concepts and token sales are listed on the platform.

3. **User Management** Ability for the administrator to manage user accounts, including registration approval, suspension, and deletion.

2.3.2 Non-Functional needs :

The non-functional needs explain all of the constraints to which the system is subjected in order for it to be realized and function properly.

1. **Security :** The DApp should implement robust security measures to protect user data, transactions, and sensitive information. This includes encryption of data, secure user authentication, and protection against common security threats such as unauthorized access, and data breaches.

2. **Scalability :** The DApp should be designed to handle a growing number of token issuers, startups, and investors without compromising performance or user experience.

3. **Reliability :** The DApp should be reliable and available for users at all times. It should minimize downtime, implement backup and recovery mechanisms, and handle system failures gracefully.

4. **Usability :** The user interface (UI) and user experience (UX) of the DApp should be intuitive and user-friendly.

5. **Interoperability :** The DApp should be designed to integrate and interact seamlessly with external systems or blockchain platforms.

6. **Compliance :** The DApp should adhere to relevant legal and regulatory requirements, such as data privacy regulations, financial regulations, and industry-specific regulations. It should implement necessary compliance measures, including KYC and AML procedures, to meet regulatory standards.

7. **Performance :** The DApp should be optimized for high performance, with fast response times and minimal latency. It should be capable of handling concurrent user requests, large transaction volumes, and complex computations efficiently

8. **Maintainability :** The DApp should be designed and developed using modular and well-documented code to facilitate ease of maintenance and future enhancements.

9. **Data Privacy :** The DApp should protect user privacy and ensure that personal information is collected, stored, and processed in accordance with data privacy regulations.

10. **Auditability :** The DApp should provide transparent tracking and auditing of token transactions, fundraising amounts, and user interactions.

These non-functional requirements are essential to ensure the overall quality, performance, security, and user satisfaction of the DApp.

2.4 Global use case diagram

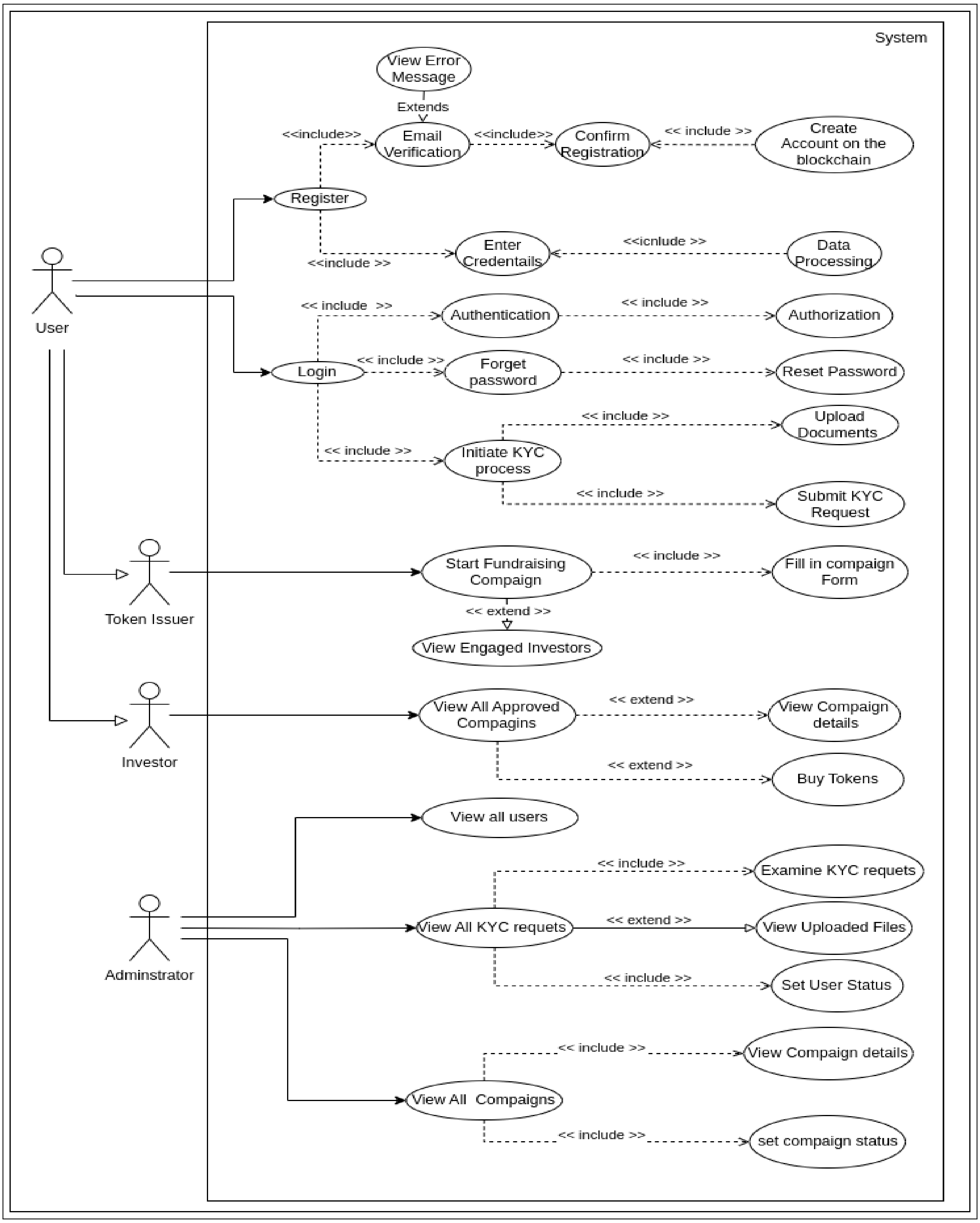


Figure 2.1: Global use case diagram

2.5 Software Architecture

The organization or settings within a system are referred to as software architecture. It includes the software components, their visible properties, and the links between them.

2.5.1 Physical Architecture

A physical architecture model is the arrangement of tangible components, such as system parts and physical interfaces, that are intended to provide a solution for a product, service, or company. Its goal is to meet the requirements established by logical architecture elements and system prerequisites.

The figure 2.2 below illustrates the physical architecture of our application

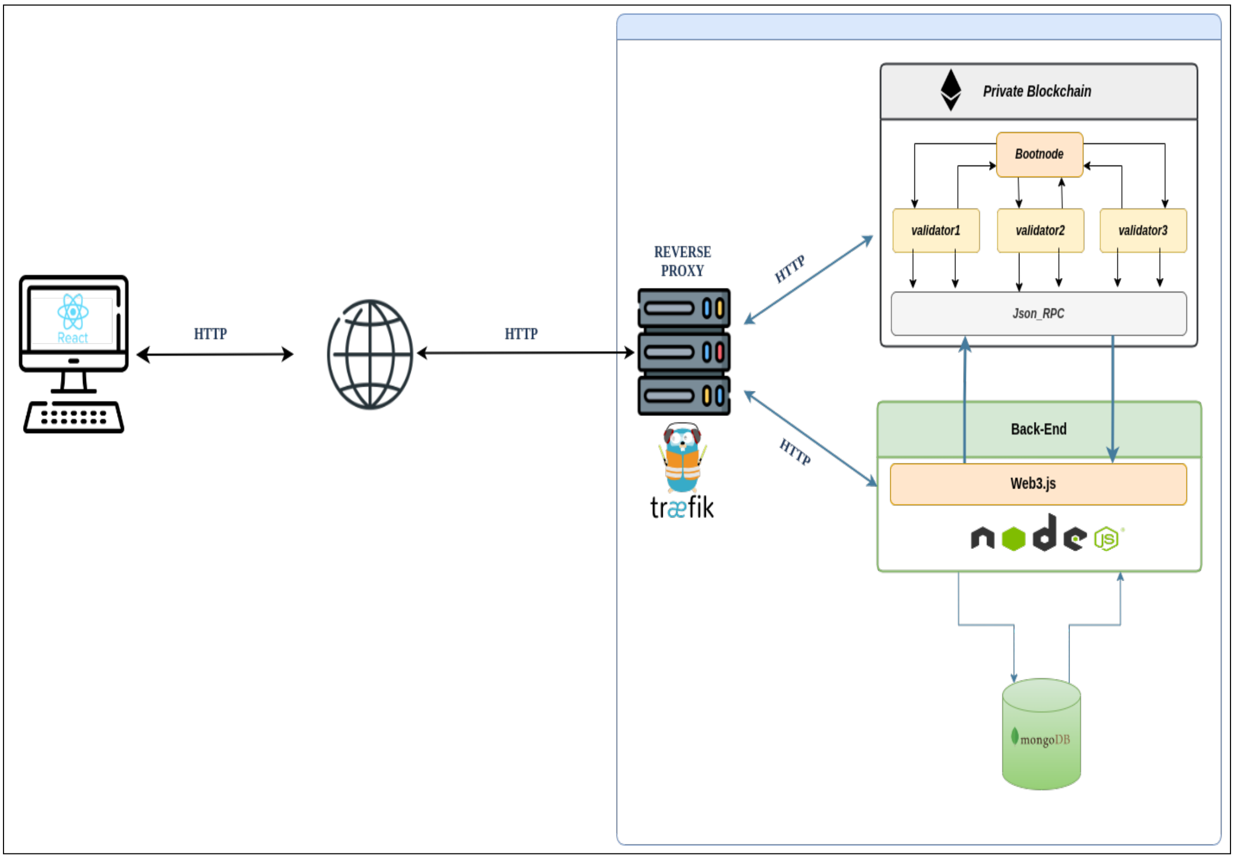


Figure 2.2: Physical Architecture

2.5.2 Logical Architecture

The logical architecture is recognized as a type of structural planning used to provide a comprehensive portrayal of the system without identifying the technology or setting. It includes all logically linked facts and provides a complete description free of technical terminology.

The figure 2.3 below illustrates the logical architecture of our application

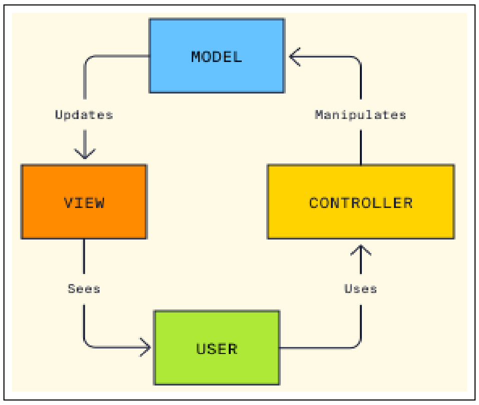


Figure 2.3: Application Logical Architecture [1]

MVC (Model-View-Controller) is a popular software design pattern that is used to realize user interfaces, data management, and control logic. It highlights the importance of separating the operational logic of software from its presentation. This 'segregation of responsibilities' allows more efficient division of activities and improves maintenance. [1]

2.6 Development environment

2.6.1 Software environment

In this section, we will enumerate the various environmental attributes (software, technologies) that have been employed in the creation of this project.

- **Docker:** Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.[2]. [2]

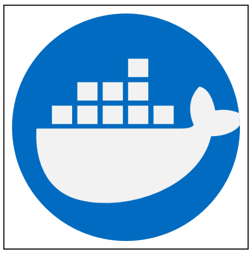


Figure 2.4: Docker Logo

- **Visual Studio Code :** is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for

[1] https://www.codecademy.com/article/mvc

[2] https://docs.docker.com/get-started/overview/

JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions[3]. [3]

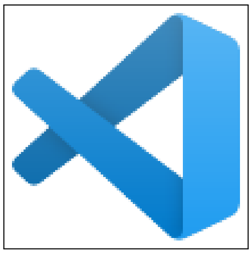


Figure 2.5: Visual Studio Logo

2.6.2 Used technologies

n this section, we will outline the array of software solutions that played a pivotal role in crafting our decentralized web application.

- **Go-ethereum (geth) :** is a Go implementation of Ethereum - a gateway into the decentralized web. Geth has been a core part of Ethereum since the very beginning. Geth was one of the original Ethereum implementations making it the most battle-hardened and tested client. [4]. [4]

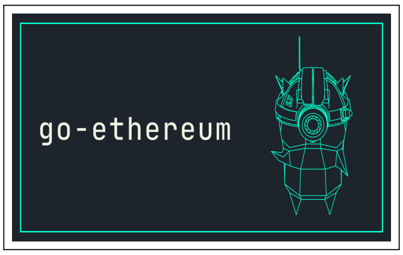


Figure 2.6: Go-Ethereum Logo

- **Solidity :** Solidity is an object-oriented, high-level language for implementing smart contracts. Smart contracts are programs that govern the behavior of accounts within the Ethereum state. [5]. [5]

Figure 2.7: Solidity Logo

[3] https://code.visualstudio.com/docs
[4] https://geth.ethereum.org/
[5] https://docs.soliditylang.org/en/v0.8.21/

- **React js :** is a free and open-source front-end JavaScript library for building user interfaces based on components. It is maintained by Meta and a community of individual developers and companies.[6]. [6]

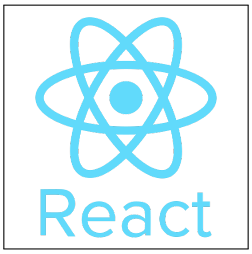


Figure 2.8: ReactJS Logo

- **Node js :** is a back-end JavaScript runtime environment, runs on the V8 JavaScript Engine, and executes JavaScript code outside a web browser[7]. [7]

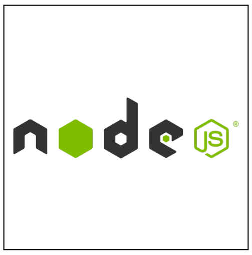


Figure 2.9: Nodejs Logo

- **Mongo-db :** is a source-available cross-platform document-oriented database , classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.[8]. [8]

Figure 2.10: MongoDB Logo

- **Traefik :** is a modern, cloud-native HTTP reverse proxy and load balancer. It provides strong support for microservices and integrates with Docker, Kubernetes, Rancher, and Consul.[9]. [9]

[6] https://en.wikipedia.org/wiki/React_(software)
[7] https://en.wikipedia.org/wiki/Node.js
[8] https://en.wikipedia.org/wiki/MongoDB
[9] https://www.kubecost.com/kubernetes-devops-tools/traefik-vs-nginx/

Figure 2.11: Traefik Logo

- **Mailjet :** lets you build email templates and email campaigns, automate, send, and monitor your email marketing campaigns. Ensure your emails hit the inbox with top-notch deliverability[10]. [10]

Figure 2.12: Mailjet Logo

2.6.3 Version Control Tool :

In this part, we will list the different environment characteristics(Software,Technologies) that has been used to develop this project.

- **GitHub :** is a platform and cloud-based service for software development and version control using Git, allowing developers to store and manage their code.[11]. [11]

Figure 2.13: Github Logo

2.6.4 Testing Tool:

In this part, we will list the different environment characteristics(Software,Technologies) that has been used to develop this project.

[10] https://www.mailgun.com/email-marketing/mailjet/
[11] https://en.wikipedia.org/wiki/GitHub

- **Truffle :** is a development environment, testing framework, and asset pipeline for Ethereum, aiming to make life as an Ethereum developer easier[12]. [12]

Figure 2.14: Truffle Logo

2.6.5 Documentation tool :

- **Gitbook :** is a collaborative documentation tool that allows anyone to document anything—such as products and APIs—and share knowledge through a user-friendly online platform.[13]. [13]

Figure 2.15: Gitbook Logo

2.7 Conclusion

In summary, this project plan provides an in-depth look at the software development process. Its emphasis on selecting an ideal development environment for successful coding and collaboration.

[12] https://github.com/trufflesuite/truffle

[13] https://arctype.com/blog/gitbook-technical-documentation-guide/

Chapter 3

Setting up Private Ethereum Blockchain

Plan

3.1 Introduction

This chapter delves into the process of establishing a Private Ethereum Blockchain. While the public Ethereum network is open and accessible to everybody, there are times when organizations and developers prefer a private, restricted blockchain environment to satisfy specific requirements. This chapter will walk you through the necessary processes and considerations for creating private Ethereum Blockchain.

3.2 Understanding Web 2.0 and Web 3.0

In this section, I will discuss the the main differences between Web 2.0 and Web 3.0

3.2.1 WEB2.0

Web2 denotes the web's second phase, known as the'read/write' web. This progression highlights optimized interactivity and interoperability, as evidenced by the characteristics and restrictions listed below:

Tableau 3.1: Attributes and Drawbacks of Web 2.0

Attributes of Web 2.0	**Drawbacks of Web 2.0**
• Informations formed by individuals • Clarity in data and interconnections • Considering the internet as a platform, not merely a network • Utilization of Software as a Service (SaaS) via API implementation technology • Retaining outdated information	• Reliance on internet connectivity for networking • Keyword-driven • Proliferation of hackers and fraudulent activities • Constrained cognitive capabilities

3.2.2 WEB3.0

Web3.0 represents the upcoming phase in the evolution of the World Wide Web. It distinguishes itself from Web 2.0 by placing a greater emphasis on decentralization, enhanced data compatibility, heightened intelligence within web systems, and the incorporation of cryptocurrency and Non-Fungible Tokens (NFTs).

Tableau 3.2: Features of Web 3.0

Features of Web 3.0	Limitations of Web 3.0
• **Decentralized**: In contrast to Web 2.0, Web 3.0 enables ownership to be distributed among its users and creators. • **Permissionless**: Everybody has equal access to Web 3.0 • **Native payments**: Web 3.0 employs cryptocurrencies for online transactions and financial transactions. • **Trustless**: It utilizes incentives and economic mechanisms for its operation.	• Blockchain and cryptocurrency adoption curve • Regulatory and legal uncertainties • Decentralized applications misuse potential • Dependency on internet for decentralized services • Maturity of used technologies

3.3 Decentralized Application

A decentralized application is one that is created on a decentralized network and includes a smart contract as well as a front-end user interface. It is a peer-to-peer blockchain network-based distributed open-source software application. The usage of blockchain in dApps enables them to handle data. [1]

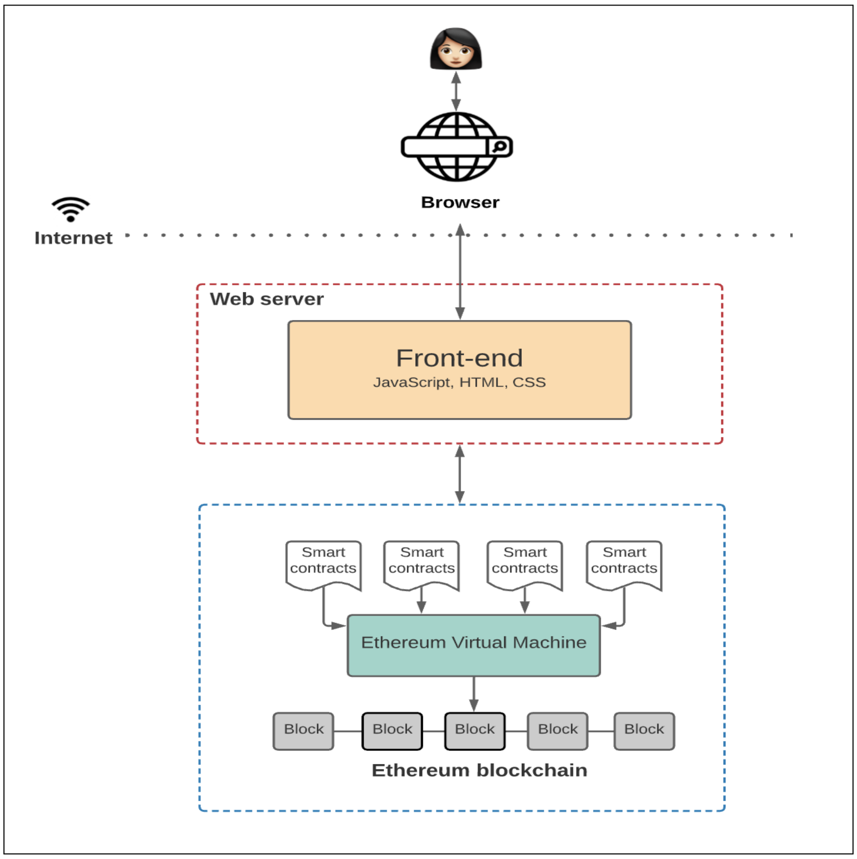


Figure 3.1: Decentralized Application Architecture [14]

[1] https://www.preethikasireddy.com/post/the-architecture-of-a-web-3-0-application

3.4 Basics of Blockchain Technology

In this section, we will delve into the fundamentals of Blockchain Technology. We will explore the essential concepts and features that underpin this technology and trace its evolution from its origins with Bitcoin to its broader applications, particularly in the case of Ethereum.

3.4.1 Introduction to Blockchain

A blockchain is a distributed, decentralized ledger that records transactions over a nodes. These transactions are arranged into blocks and subsequently included into the chain, resulting in a clear and unalterable history of all network operations. [15]. [2]

3.4.2 Key Attributes of Blockchain

1. **Decentralization:** Decentralization is a core feature of blockchain technology. Unlike traditional centralized systems, where a single entity controls data and transactions, blockchains are supported by a network of nodes (computers) that collaborate to authenticate and log transactions.

2. **Transactions and Blocks:** Transactions are the foundation of a blockchain. When a user initiates a transaction, it is broadcast to the network for validation and verification by miners. When a transaction is authorized, it is added to a block together with other transactions. Each block carries a reference to the one before it.

3. **Consensus mechanism:** Consensus mechanisms, like as PoW, ensure that all network members agree on the truthfulness of transactions. A majority of nodes must agree for a new block to be added to the chain.

4. **Immutability:** After a block is added to the blockchain, it becomes extremely difficult to edit or delete the data it contains. This immutability is achieved using cryptographic hashing, in which each block includes a unique code (hash) defined by its content and the hash of the previous block.

3.4.3 How Blockchain works ?

In this section , will go into the inner workings of blockchain, beginning with the fundamentals of blockchain and the crucial genesis block, and progressing through how users interact with the blockchain and the mechanics of transactions within this distributed ledger.

[2] https://en.wikipedia.org/wiki/Blockchain

Fundamentally, within any blockchain network, nodes play a central role.Nodes are essentially computers that are actively involved in the operations of the blockchain network. They keep a comprehensive copy of the blockchain ledger and are in charge of the network's security and reliability. There are two types of nodes in a blockchain network:

1. **Full Nodes:** Full nodes are the backbone of the blockchain network. They maintain a complete copy of the blockchain ledger and validate all transactions and blocks added to the chain.

2. **Light nodes:** Light nodes, on the other hand, do not store the entire blockchain. They rely on full nodes to provide them with the necessary information when required. Light nodes are more lightweight in terms of storage and computational requirements, making them suitable for devices with limited resources like smartphones.

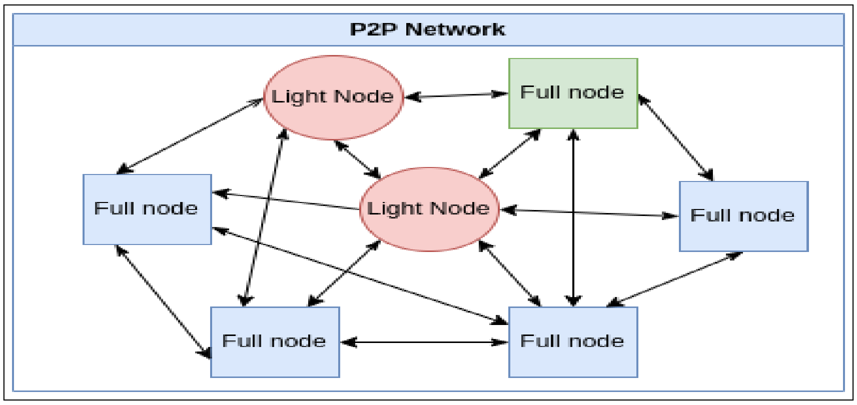


Figure 3.2: Blockchain network[14]

3.4.3.1 Blockchain Foundation

Each blockchain begins with a genesis block, which marks a watershed moment in its history. This first block, which is empty of transaction details, serves as the foundation upon which all subsequent blocks are built. It establishes the first set of rules for the blockchain, including the cryptographic hash that ensures the chain's integrity.

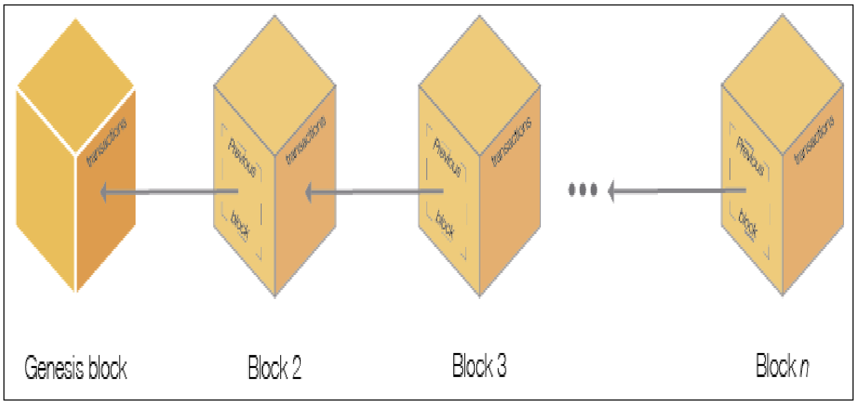


Figure 3.3: Blockchain Genesis Block

3.4.3.2 User Interaction with the Blockchain

Users interact with the blockchain primarily through digital wallets. These wallets contain cryptographic keys that allow users to send and receive digital assets or interact with smart contracts (self-executing contracts with the terms of the agreement directly written into code). Here's how user interaction with the blockchain typically occurs:

1. **Creating a Wallet:** Users start by creating a digital wallet. A wallet generates a pair of cryptographic keys: a public key (similar to an account number) and a private key (similar to a password). The private key must be kept secure and secret.

2. **Making Transactions:** When a user wants to send cryptocurrency or interact with a smart contract, they create a transaction. This transaction includes the recipient's public key and the amount to be sent. The user then signs the transaction with their private key, proving ownership and authenticity.

3. **Broadcasting the Transaction:** The signed transaction is broadcast to the blockchain network. Nodes in the network validate the transaction to ensure it follows the blockchain's rules and has the necessary funds.

4. **Confirmation and Inclusion:** Valid transactions are grouped together into a block by miners (participants who perform the computational work to secure the network). Once the block is mined, it is added to the blockchain, and the transaction is considered confirmed.

3.4.4 Bitcoin: The Pioneer of Blockchain

Bitcoin, created by an anonymous person or group using the pseudonym Satoshi Nakamoto, is the pioneering application of blockchain technology. It was introduced in a whitepaper in 2008 and launched as an open-source software in 2009. Bitcoin's primary use case is as a digital currency, allowing for peer-to-peer transactions without the need for intermediaries like banks. It relies on a consensus mechanism called Proof of Work (PoW) to validate and record transactions on the blockchain.

3.5 Ethereum: Advancing Blockchain Technology

Ethereum acts as a technology for crafting applications and entities, overseeing assets, executing transactions, and enabling communication, all liberated from centralized authority. It forms the foundation of a fresh, decentralized internet.

The Ethereum Foundation's goal is to encourage and back the creation of decentralized applications on the Ethereum platform. Geth stands as a fundamental element of the Ethereum network, and it comes in two variants for download: Geth without tools and Geth with tools.

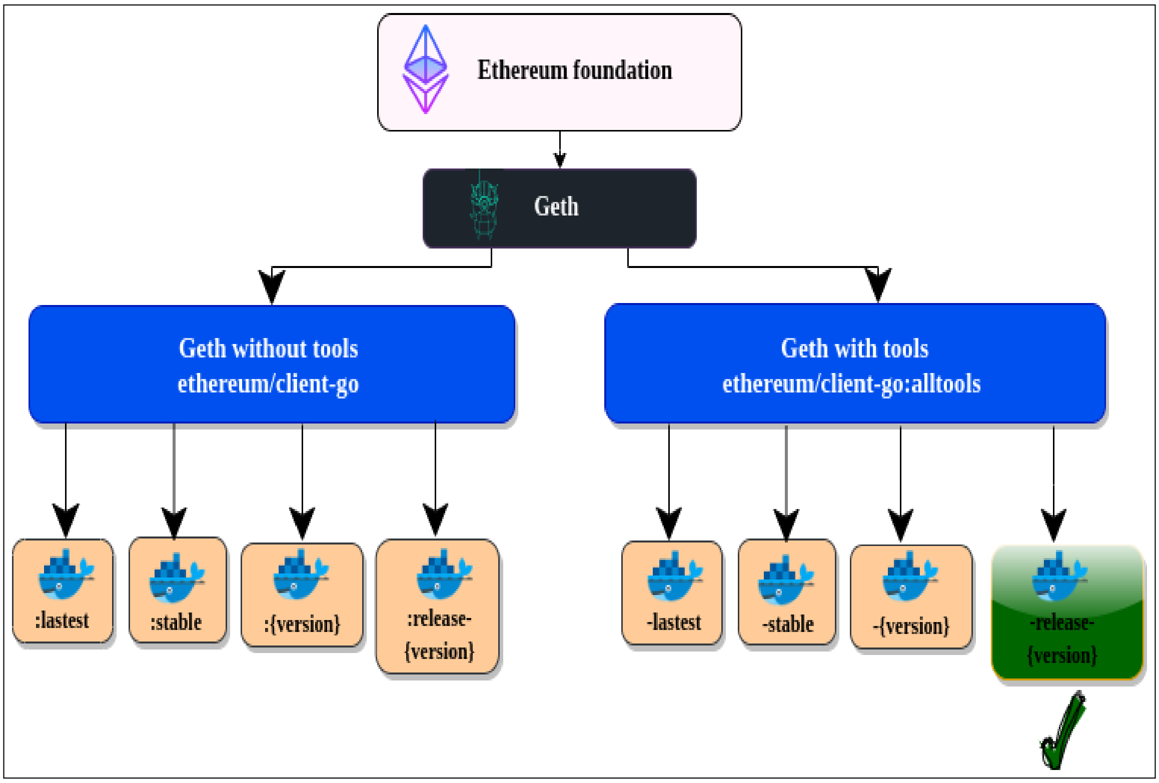


Figure 3.4: Different Geth versions released by Ethereum foundation

3.5.1 Characteristics of Ethereum

1. **Decentralization**: Like Bitcoin, Ethereum operates on a decentralized network of nodes, making it resistant to censorship and control by any single entity.

2. **Smart Contracts** : Ethereum's ability to execute smart contracts programmatically opens up a wide range of applications, from decentralized finance (DeFi) to supply chain management.

3. **EVM (Ethereum Virtual Machine)** : Ethereum has its virtual machine, allowing developers to build DApps using various programming languages.

4. **Token Standards** : Ethereum introduced ERC-20 and ERC-721 token standards, enabling the creation of fungible and non-fungible tokens (NFTs).

3.5.2 Ethereum Smart Contracts

Smart contracts on Ethereum are self-executing agreements with predefined rules and conditions. They automatically enforce contract terms, eliminating the need for intermediaries. Developers can create complex applications by combining smart contracts, opening up innovative possibilities.

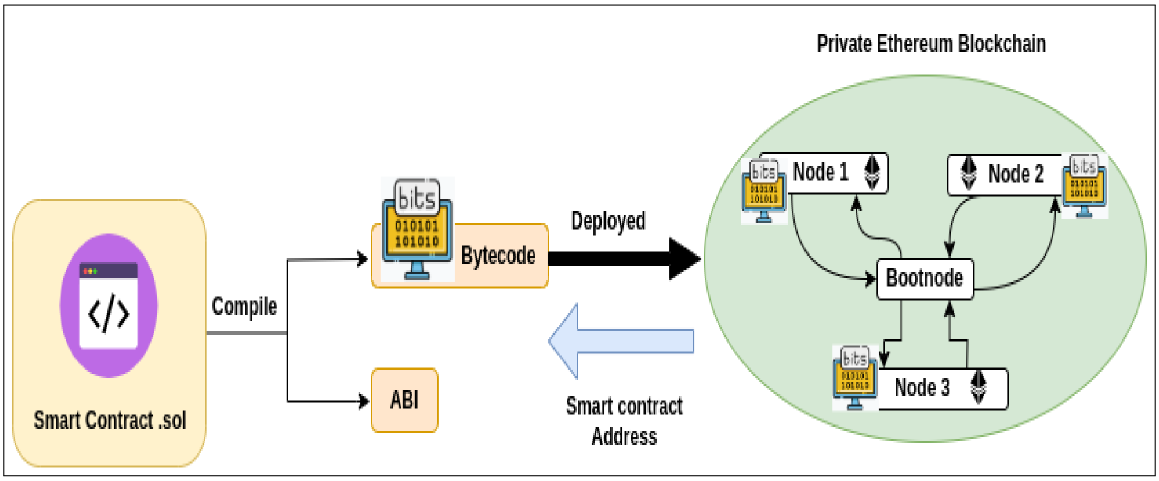


Figure 3.5: Smart contract deployment

Smart contract deployment to the blockchain involves compiling the contract into bytecode and generating an ABI. Subsequently, copies of the bytecode are deployed to each node on the blockchain network. This ensures that the contract's code is securely distributed across the network, allowing it to be executed and interacted with by participants and other contracts.

Once the smart contract is successfully deployed, the blockchain will return the address of the smart contract. This address serves as a unique identifier for the contract on the blockchain and will be used later to interact with its functionalities, enabling participants to send transactions to the contract, invoke its methods, and access the data it stores.

3.5.3 Comparison between Bitcoin and Ethereum

While both Bitcoin and Ethereum utilize blockchain technology, they serve different purposes. Bitcoin primarily functions as a digital currency and store of value, while Ethereum is a versatile blockchain platform for creating DApps and executing smart contracts. Ethereum's programmability and broader use cases distinguish it from Bitcoin.

	Ethereum Blockchain	**Bitcoin Blockchain**
Founder	Vitalik Buterin	Satoshi Nakamoto
Launched On	2015	2009
Total Supply	Infinite	21 Million BTC
Cryptocurrency	ETH	BTC
Consensus Algorithm	Transition from PoW to PoS	PoW
Smart Contracts	Yes	No

Tableau 3.3: Comparison between Ethereum and Bitcoin

3.6 Choosing the right consensus

A consensus algorithm is a tool for coordinating machines in a distributed situation. It must ensure that, even if some agents fail, all agents in the system can agree on a **single source of truth.**

	Description	**Scalable**	**Efficiency**	**security**	**decentralized**	**Fast**
Proof of Work (PoW)	- Requires miners to solve complex mathematical problems to add blocks to the chain.	- No	- No	- High	- Yes	- No
Proof of Stake (PoS)	- Validators are chosen based on the amount of cryptocurrency they stake .	- Yes	- Yes	- High	- Yes	- Yes
Proof of Authority (PoA)	-Validators are known entities that are authorized to add blocks to the chain.	- Yes	- Yes	- High	- No	- No
Delegated Proof of Stake (DPoS)	- Token holders elect a smaller group of delegates to validate transactions on their behalf.	- Yes	- Yes	- Meduim	- No	- No

Tableau 3.4: comparative table highlighting differences between different types of consensus mechanism

Proof of Work (PoW) stands as the superior consensus mechanism, excelling in security and decentralization. Its energy-intensive computations and extensive validator network offer unparalleled protection against attacks. PoW's decentralized nature prevents single-point control, reducing the risk of failure or censorship. It remains the preferred choice for blockchain systems seeking immutability, trustlessness, and robust security.

3.7 Choosing the right provider

In the upcoming section, we'll explore different blockchain providers and evaluate the pros and cons of each

A provider is a software component or library that helps a decentralized application (DApp) communicate with a blockchain. These providers often provide a set of application programming interfaces (APIs) that allow developers to easily connect with blockchain.

	Infura	**Moralis**	**Web3.js**
Cost	- Free to use up to certain limits, paid plans available	- Free to use up to certain limits, paid plans available	- Free and open source
Transparency	- Publicly auditable nodes	- Open source, publicly auditable	- Publicly auditable nodes
Scalability	- Can handle high volumes of requests	- Highly scalable infrastructure	- Limited by Ethereum's scalability
Speed	- Fast response times	- Fast response times	- Can be slow due to reliance on Ethereum network
Security	- High reliability and security	- High reliability and security	- High reliability and security

Tableau 3.5: comparative table highlighting differences between different providers

While there are many blockchain providers available, each with its own set of strengths and weaknesses, web3.js stands out as a suitable library for interacting with Ethereum blockchain networks

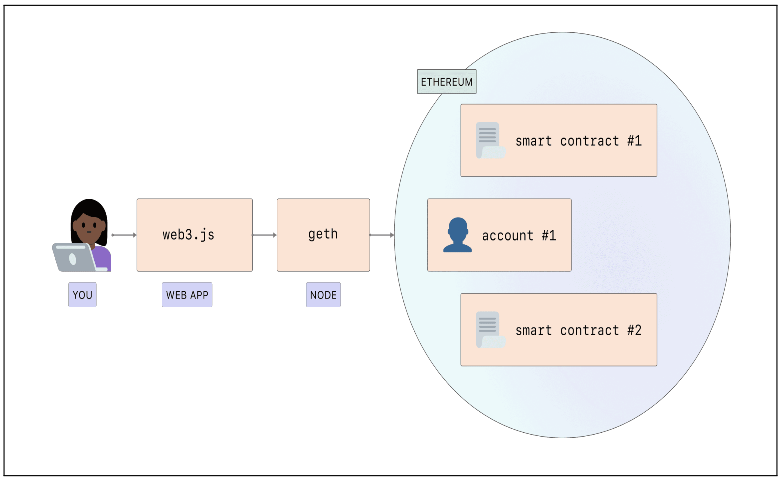
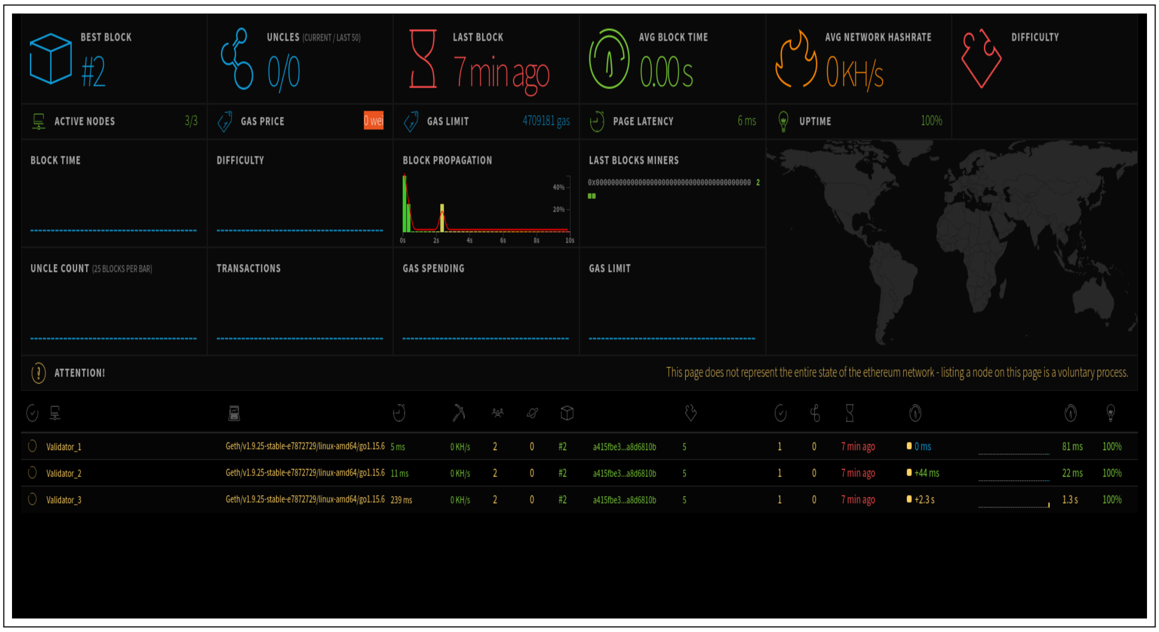


Figure 3.6: Web3js

3.8 Monitoring Private Blockchain

This section is dedicated to setting up Ethstats in go-ethereum and configuring it to monitor our private blockchain

Ethstats is a platform that provides current and historical data about single network nodes as well as the network as a whole. Individual node information includes information such as the most recently received block, block processing time, how quickly data propagates, the number of associated peers, delays, and more. The larger network statistics include information like as total node count, average block processing time, node geographical locations, transaction counts, and other comparable data points.



BEST BLOCK #2

UNCLES (CURRENT / LAST 50) 0/0

LAST BLOCK 7 min ago

AVG BLOCK TIME 0.00 s

AVG NETWORK HASHRATE 0 KH/s

DIFFICULTY

ACTIVE NODES 3/3

GAS PRICE 0 wei

GAS LIMIT 4709181 gas

PAGE LATENCY 6 ms

UPTIME 100%

0x0000000000000000000000000000000000000000 2

ATTENTION! This page does not represent the entire state of the ethereum network - listing a node on this page is a voluntary process.

Validator_1	Geth/v1.9.25-stable-e7872729/linux-amd64/go1.15.6	5 ms	0 KH/s	2	0	#2	a415fbe3...a8d6810b	5	1	0	7 min ago	0 ms		81 ms	100%
Validator_2	Geth/v1.9.25-stable-e7872729/linux-amd64/go1.15.6	11 ms	0 KH/s	2	0	#2	a415fbe3...a8d6810b	5	1	0	7 min ago	+44 ms		22 ms	100%
Validator_3	Geth/v1.9.25-stable-e7872729/linux-amd64/go1.15.6	239 ms	0 KH/s	2	0	#2	a415fbe3...a8d6810b	5	1	0	7 min ago	+2.3 s		1.3 s	100%

Figure 3.7: Screenshot of Ethstat Dashboard

3.9 Conclusion

In summary, this chapter has provided a comprehensive guide on establishing a private Ethereum blockchain. It has emphasized key concepts, including Web3, decentralized applications, consensus mechanisms, selecting the optimal provider, and monitoring blockchain activity.

Chapter 4

Implementation

Plan

4.1 Introduction

This chapter outlines the comprehensive implementation plan for the project, offering a detailed roadmap for the execution of each release, including its corresponding sprints and featured components. The successful realization of any project hinges on a well-structured implementation strategy, and this plan has been meticulously crafted to ensure that each release achieves its objectives effectively and efficiently.

4.2 Release 01

In this release, our central focus is on implementing authentication and authorization functionalities that cater to both investors and startups, enabling them to submit KYC (Know Your Customer) requests on the platform.

Our first release will consist of two sprints:

Authentication : This sprint covers the implementation of Authentication

KYC Process: This sprint focuses on the tasks and activities related to KYC process which enables the user to deposit KYC request.

4.2.1 Sprint One: Authentication

During this sprint, the team will focus on developing and implementing user authentication features for the first release of the software.

4.2.1.1 Sprint One Backlog

In this sprint, our primary goal is to enhance the platform's authentication and user management features. We've identified key user stories and tasks to achieve this objective. We'll begin by creating a user-friendly Register Page, streamlining the user registration process. Additionally, we will configure MailJet to send verification emails when users register with their email addresses, promoting communication and user engagement.Furthermore, to enhance security and user convenience, we'll also configure MailJet to send verification links for password recovery.Additionally, a Logout feature will be implemented to provide users with control over their sessions, ensuring the secure termination of user sessions. Lastly, recognizing the significance of blockchain technology, we'll integrate it by creating a blockchain account for each new user.

Tableau 4.1: Authentication Sprint Backlog table

ID	User Story	Task	Estimation
1.1	As a user to user I want to register on the platform	Create Register Page	40h
1.2	As a user to user I want to be notified once a user is registered with my email	Configure MailJet to send a verification email	20h
1.3	As a user I want to login to the platform	Create Login Page Create authentication module to allow users to login	20h
1.4	As a user I want to recover my password	Configure MailJet to send a verification Link to the registered user	40h
1.5	As a user I want to logout	Delete JWT token from local storage and redux store	20h
1.6	As a user I want to get a a wallet on the blockchain	An account on the blockchain will be created to the each new user	20h

At the end of this sprint , the user experience within the platform has significantly improved, particularly in terms of authentication and user management. Users now benefit from a more streamlined registration process with the introduction of a user-friendly Register Page.

4.2.1.2 User registration

User registration within a platform is a fundamental process that allows individuals to create their own accounts and gain access to the platform's services or content. During registration, users typically provide essential information such as their full name, email address, and a password. This information is used to uniquely identify and authenticate users. User registration serves several important purposes, including personalizing the user experience, enabling secure access to the platform, and facilitating communication between the platform and the user. It's an essential step that lays the foundation for user interaction and engagement within the platform's ecosystem.

4.2.1.2.1 User Registration Use Case

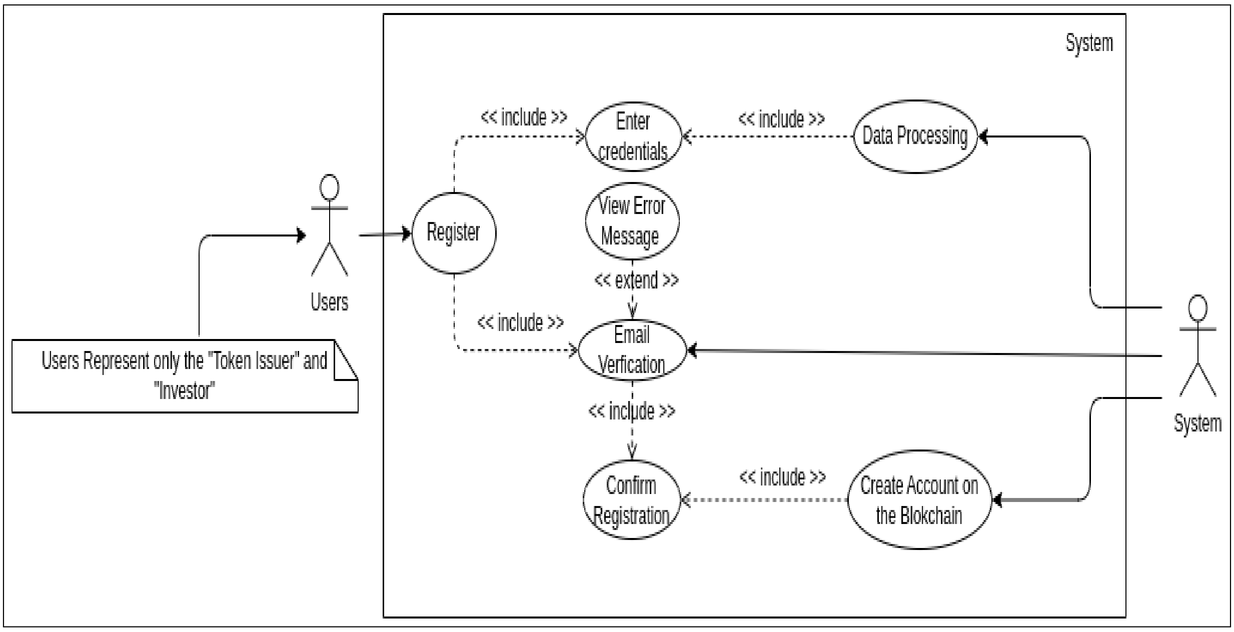


Figure 4.1: User registration use case

User registration is a pivotal component of digital platforms, allowing individuals to create their own accounts. During this process, users provide essential information, such as their name, email address, and a chosen password. This information is used to establish their unique identity within the platform, granting them access to its features and services.

Tableau 4.2: Textual Description Of The Registration Use Case

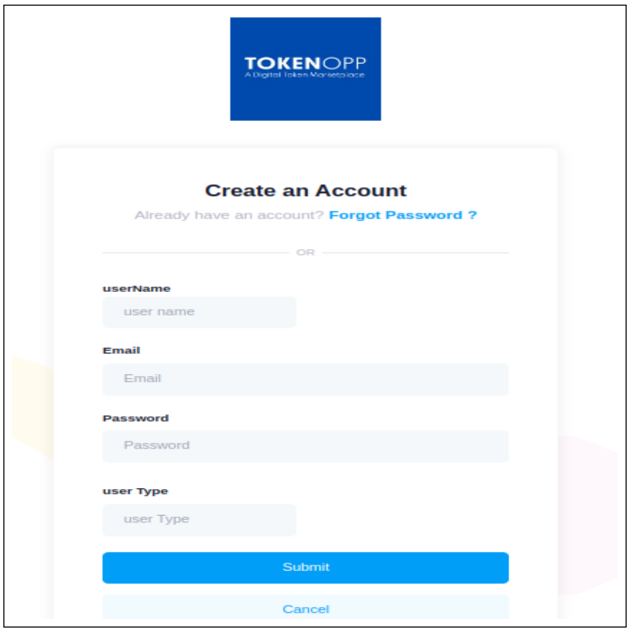
Actor	Token Issuer , Investor , Admin
Goal	Allows user to undergo the registration process
Precondition	Nil
Post-condition	User register on the platform
Nominal Scenario 1. The user fills his own credentials 2. the user press register button 3. the system send a verification email the user 4. the user clicks on the verification link then he will be logged in to the platform	
Alternative 5. If the user doesn't click on the verification link he will not be allowed to register on the platform	

The user registration involves three key actors: Token Issuer, Investor, and Admin, and aims to enable users to complete the registration process on the platform. There are no specific preconditions, and the post-condition is that the user successfully registers on the platform. In the nominal scenario, the user provides their credentials, clicks the register button, and the system sends a verification email to the user. Upon clicking the verification link, the user gains access to the platform. However, in the alternative scenario, if the user does not click on the verification link within the specified time frame, they will not be allowed to complete the registration process, ensuring security and validation of user accounts.

4.2.1.2.2 Registration Steps

The following steps represent the sequence of actions and requirements individuals must follow when signing up or registering on the platform. These steps are crucial for user onboarding and account creation.

1. **Fill registration form :**



TOKENOPP
A Digital Token Marketplace

Create an Account

Already have an account? Forgot Password ?

OR

userName
user name

Email
Email

Password
Password

user Type
user Type

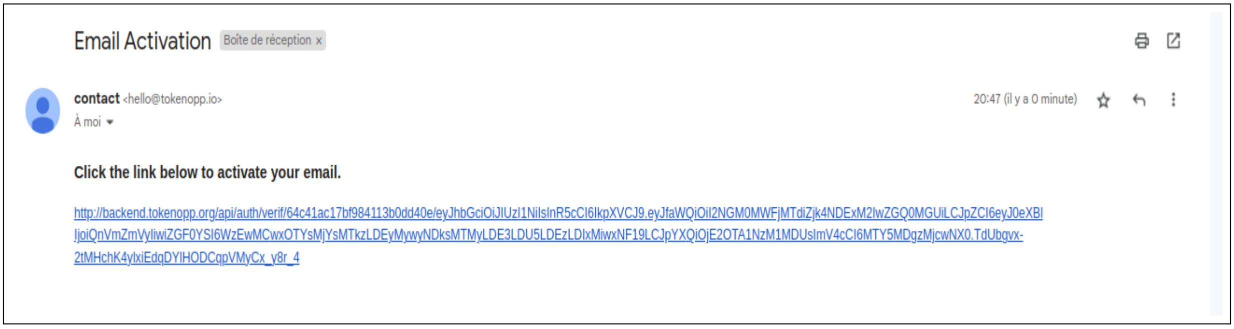
Submit

Cancel

Figure 4.2: User registration Interface

In first step of user registration , users are prompted to enter specific information, including their full name, a valid email address, and a secure password as shown in figure 4.2. The email address serves as a unique identifier and facilitates important account-related notifications. The password, on the other hand, is a critical element of account security, and we encourage users to create a strong, confidential password to safeguard their information. By providing these details accurately, users can take their first step towards accessing our platform has to offer.

2. **Receiving Activation Email:**



Email Activation

contact <hello@tokenopp.io>
À moi

20:47 (il y a 0 minute)

Click the link below to activate your email.

http://backend.tokenopp.org/api/auth/verif/64c41ac17bf984113b0dd40e/eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NGM0MWFjMTdiZjk4NDExM2IwZGQ0MGUiLCJpZCI6eyJ0eXBlIjoiQnVmZmVyIiwiZGF0YSI6WzEwMCwxOTYsMjYsMTkzLDEyMywyNDksMTMyLDE3LDU5LDEzLDIxMiwxNF19LCJpYXQiOjE2OTA1NzM1MDUsImV4cCI6MTY5MDgzMjcwNX0.TdUbgvx-2tMHchK4ylxiEdqDYIHODCqpVMyCx_y8r_4

Figure 4.3: Email verification

In the second step , users will receive a clickable link that will lead them to the login interface. the email verification is a pivotal moment in the user registration process. By clicking on the provided link within the email, the user is seamlessly directed to the login interface, where they can access their account and begin utilizing the platform's services. The email verification (figure 4.3) emphasizes the user-centric approach to account activation, ensuring a secure and straightforward onboarding experience.

4.2.1.3 User Login

User login is a core function in digital platforms and systems that enables registered users to access their accounts securely. It typically involves users providing their previously created credentials, which commonly include a username or email address and a password. After entering this information, the system verifies the authenticity of the user's credentials and, if they are accurate, grants access to the user's account.

4.2.1.3.1 User Login Use Case

The User Login use case (figure 4.4) is a fundamental interaction in software systems, allowing registered users to access their accounts securely. Users provide their authentication credentials, such as a username or email address and a password, to gain entry. The system then verifies the provided information against its stored records. If the credentials match, the user is granted access to their account, enabling them to use the system's features, access their data, and perform actions specific to their user role. User Login is a critical component for ensuring account security, privacy, and personalized user experiences within digital platforms and applications.

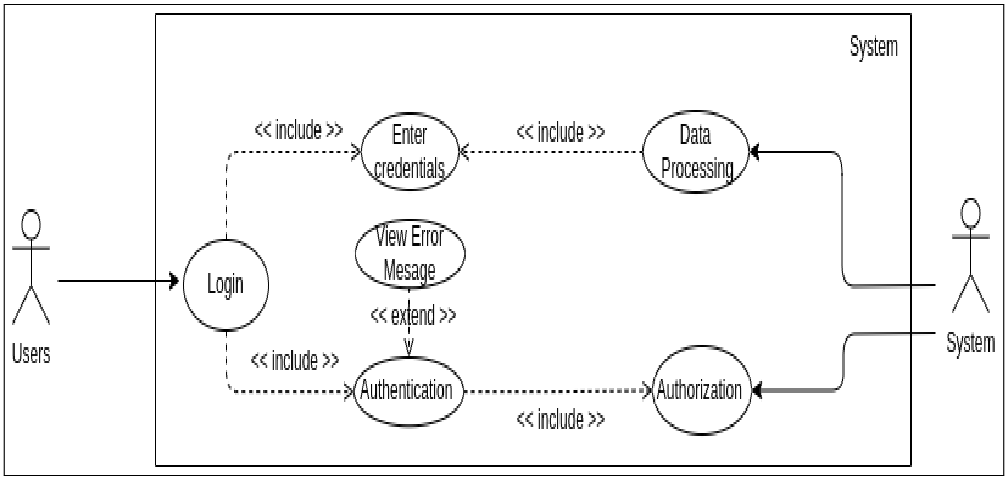


Figure 4.4: User Login use case

Tableau 4.3: Textual Description Of Login Use Case

Actor	Token Issuer , Investor , Admin
Goal	Allows user to login to the paltform
Precondition	Registered user
Post-condition	User Login to the platform
Nominal Scenario 1. The user fills his own credentials 2. the user press Login button 3. the system checks the correctness of his credentials 4. if the credentials are correct the uer will be logged in	
Alternative 5. If the user's credentials are not correct he will not be allowed to get access to the platform	

The login process includes three actors : Token Issuers, Investors, and Admins, can access the platform. The preconditions require that the user must already be a registered user, and the post-condition is the successful login to the platform. In the nominal scenario, the user inputs their credentials, clicks the "Login" button, and the system validates the accuracy of the provided information. If the credentials are verified as correct, the user gains access to the platform. However, in the alternative scenario, if the user's credentials are not accurate, access to the platform is denied, ensuring security and safeguarding the integrity of user accounts.

4.2.1.3.2 User Login Process

User Login Interface (figure 4.5) illustrates a fundamental step in the user login process, emphasizing the importance of entering the correct email and password. In this step, users are prompted to carefully input their accurate email address and password into the designated fields. Following this, they are required to press the "Login" button. This straightforward and secure authentication procedure ensures that only authorized individuals can gain access to the platform, contributing to the overall security and integrity of user accounts and data.

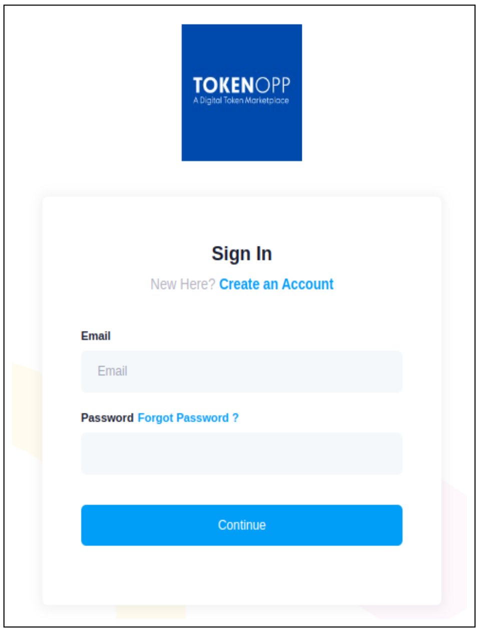

TOKENOPP
A Digital Token Marketplace
Sign In
New Here? Create an Account
Email
Email
Password Forgot Password ?
Continue

Figure 4.5: User Login Interface

4.2.1.4 Password reset

The Password Reset functionality is a crucial feature in digital systems and online platforms that allows users to regain access to their accounts when they have forgotten or need to change their passwords. Typically, it involves a user-initiated process where the user provides some form of identity verification, such as an email address, to confirm their account ownership. Once their identity is verified, the system generates a temporary verification code and sends it to the user through email. With this code , the user can then reset their password to something new, ensuring the security of their account. This functionality is essential for user account security and usability, offering a straightforward way for users to regain access to their accounts while protecting them from unauthorized access.

4.2.1.4.1 Password Reset Use Case

This use case (figure 4.6) is designed to facilitate password resets for registered users, including Token Issuers, Investors, and Admins. The primary goal is to allow users to reset their passwords securely.

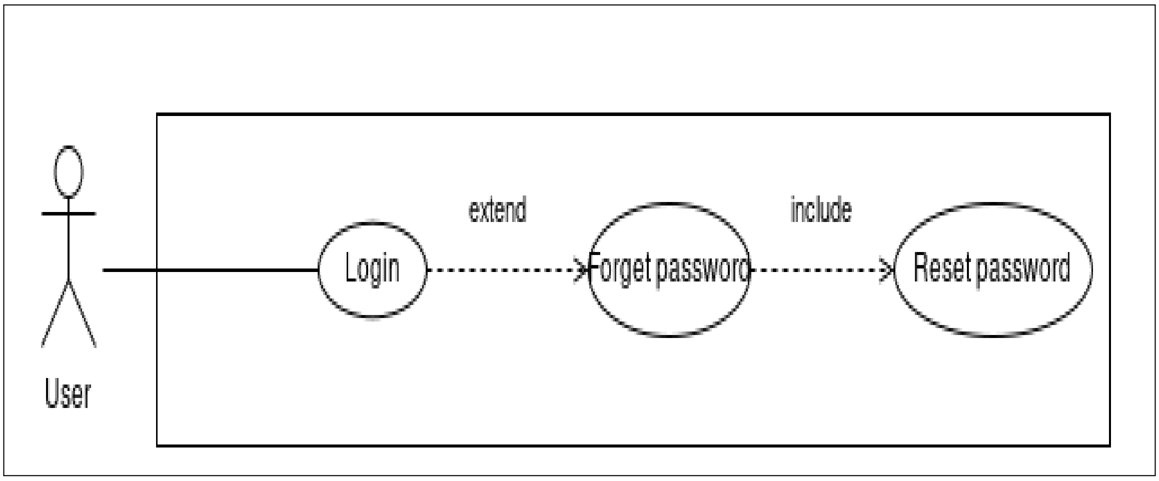


Figure 4.6: Password reset use case

Tableau 4.4: Textual description of password reset use case

Actor	Token Issuer , Investor , Admin
Goal	Allows user to reset his password
Precondition	Registered user
Post-condition	User's password reset successfully
Nominal Scenario 1. The user put the email related to his account 2. the system send a verification code the user 3. the system send a verification email the user 4. the user should enter the sent code in order to reset his password	
Alternative 5. If the user doesn't enter the correct code he will not be allowed to reset the password	

In the nominal scenario, the process for resetting a user's password unfolds as follows: the user begins by providing their account-associated email, prompting the system to send a verification code and a verification email to the registered email address. To complete the password reset, the user must enter the verification code received in the email, which serves as a secure confirmation of their identity.

Conversely, in the alternative scenario, if the user enters an incorrect verification code, they are denied access to the password reset feature. This safeguard ensures that only authorized users can initiate password resets, bolstering the overall security of user accounts and sensitive information.

4.2.1.4.2 Password Reset Sequence Diagram

A Password Reset Sequence Diagram (figure 4.7) illustrates the step-by-step interactions between actors and components involved in the password reset process, where a verification code is used for added security. This diagram typically involves actors such as the user, the system, and possibly email services. It begins with the user initiating a password reset request and then proceeds to depict the system's actions, such as generating a verification code and sending a verification email to the user. The user's steps include receiving the email, extracting the code, and entering it into the system to confirm their identity and reset their password. This diagram provides a clear and concise representation of the secure password reset procedure, emphasizing the importance of the verification code in enhancing account security.

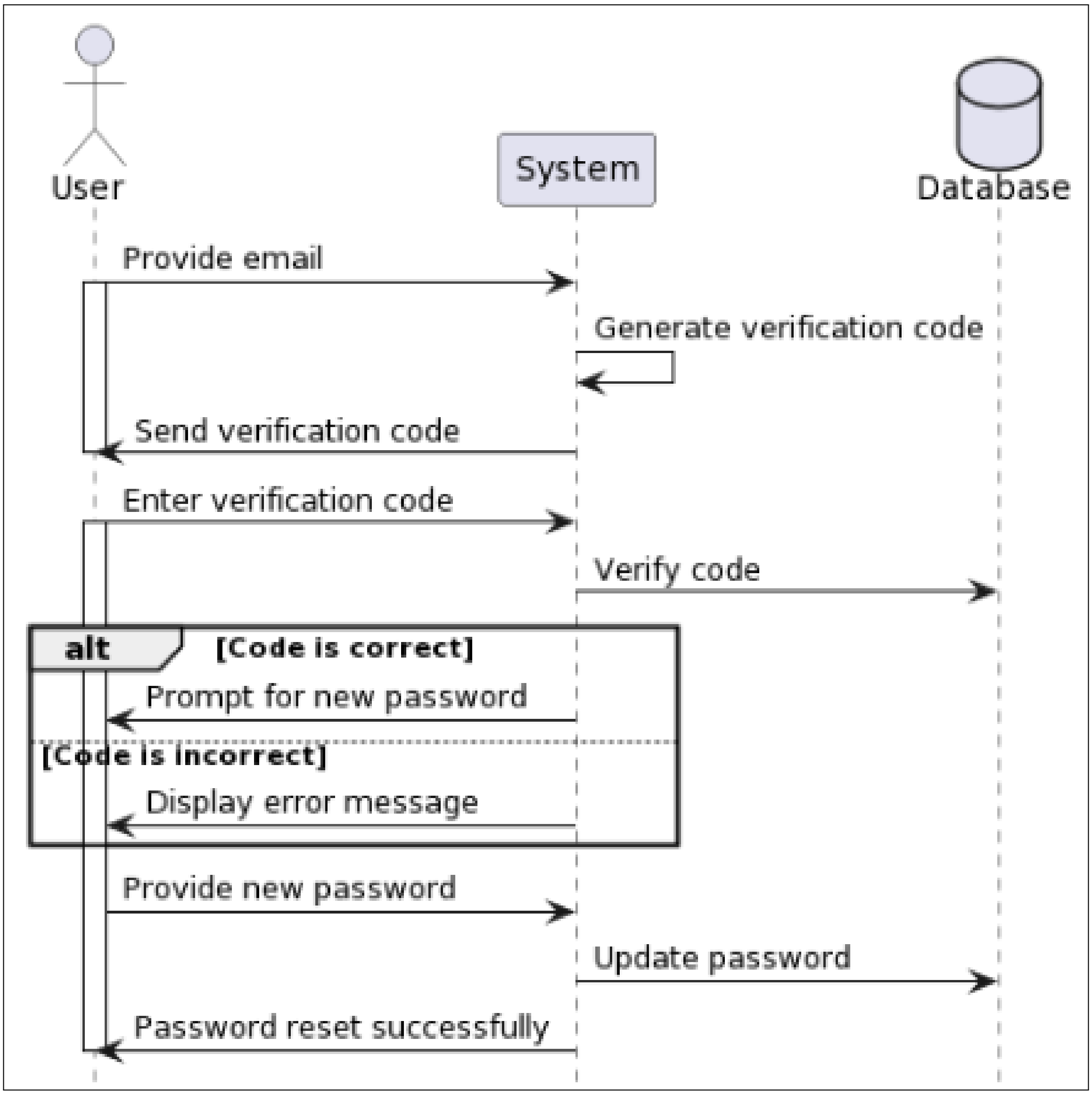


Figure 4.7: Password reset sequence diagram

4.2.1.4.3 Password Reset Steps

The following steps represent the sequence of actions and requirements individuals must follow in order to reset their password and regain access to their accounts.

1. **Initiating a password reset request:** The first step towards password resetting is to fill in the specified form as shown in the initial interface of the password reset process (Figure 4.8). Users are prompted to enter their valid email address into the designated field. Upon entering a valid email, the system will send a verification code to the provided email address. This crucial step in the password reset procedure ensures the security of the account recovery process, as it confirms the user's identity and authorizes the issuance of the verification code. It highlights the user-friendly design and security measures in place to facilitate a smooth and secure password reset experience.

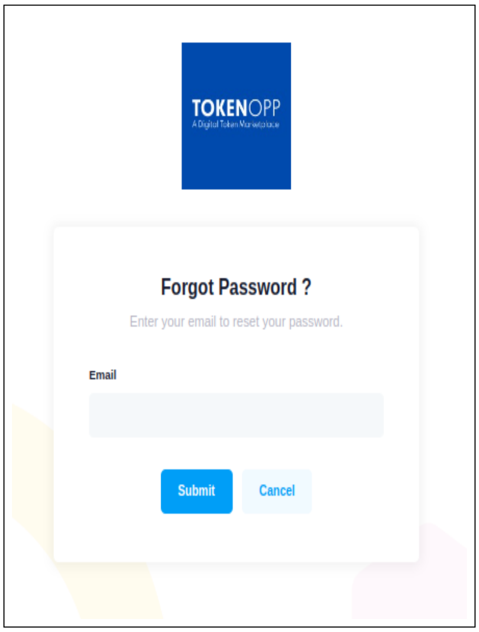
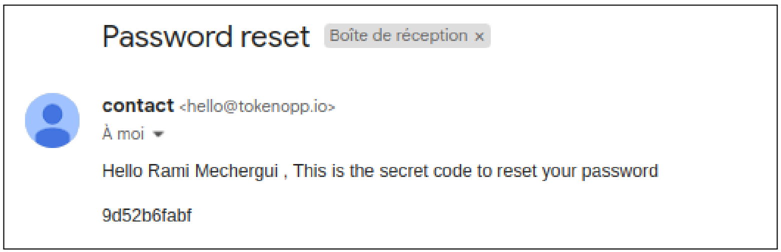

TOKENOPP
A Digital Token Marketplace
Forgot Password ?
Enter your email to reset your password.
Email
Submit
Cancel

Figure 4.8: Password Reset First Interface

2. **Receiving verification code:**

   In this step, users typically receive a verification code via email initiating a password reset request (figure 4.9). This code serves as a secure means of confirming the user's identity and authorizing the password reset. The screenshot underscores the importance of this verification code in the security of the account recovery process, ensuring that only authorized users can successfully reset their passwords and regain access to their accounts.


Password reset

contact <hello@tokenopp.io>
À moi

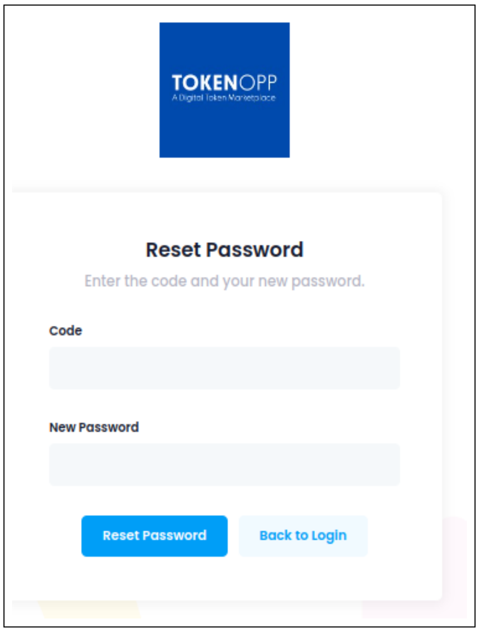
Hello Rami Mechergui , This is the secret code to reset your password

9d52b6fabf

Figure 4.9: Screenshot of verification code sent to reset password

3. **Entering correct code and password reset:**

   In this step , users are required to follow the third step of the password reset process. After receiving the verification code in their email, users must enter this code into the designated field on this interface (figure 4.10) , confirming their identity securely. Following this, users are prompted to create a new password, enhancing the security of their account. After inputting the new password, users press the "Reset Password" button to finalize the process. This step ensures that the user has full control over resetting their password, contributing to account security and providing a streamlined experience for account recovery.



Reset Password

Enter the code and your new password.

Code

New Password

Figure 4.10: Password Reset Second Interface

4.2.2 Sprint Two: KYC Process

Through this sprint, both Token Issuers and Investors will have a seamless experience submitting their KYC requests, uploading required documents, and receiving prompt notifications about the status of their requests.

4.2.2.1 Sprint Two Backlog

This sprint focuses on optimizing the KYC workflow to provide users with a streamlined and efficient journey while maintaining security and compliance.

Tableau 4.5: Investor KYC Request Sprint Backlog Table

ID	User Story	Task	Estimation
1	As an investor, I want to submit a KYC request	Create KYC Request Submission Page	40h
2	As an investor, I want to upload required documents for KYC	Implement file upload functionality for documents	30h
3	As an investor, I want my uploaded documents to be securely stored	Set up storage for uploaded documents	20h
4	As an investor, I want to be notified about the status of my KYC request	Implement notification system for request status (accepted, refused, pending)	30h
5	As an investor, I want to see the status of my KYC request on the blockchain	Integrate blockchain data with request status	20h

The Investor KYC Request Sprint Backlog (Tableau 4.5) outlines the key tasks and user stories to be tackled in this sprint, with a primary focus on enhancing the Know Your Customer (KYC) process for investors. Firstly, we'll create an intuitive KYC Request Submission Page to simplify the submission process. Secondly, we will implement file upload functionality, allowing investors to securely upload the necessary documents for KYC verification. Ensuring the security and confidentiality of these documents, we'll set up a robust storage system.

Tableau 4.6: Token Issuer KYC Request Sprint Backlog Table

ID	User Story	Task	Estimation
1	As a token issuer, I want to submit a KYC request	Create KYC Request Submission Page	40h
2	As a token issuer, I want to upload required documents for KYC	Implement file upload functionality for documents	30h
3	As a token issuer, I want my uploaded documents to be securely stored	Set up storage for uploaded documents	20h
4	As a token issuer, I want to be notified about the status of my KYC request	Implement notification system for request status (accepted, refused, pending)	30h

5	As a token issuer, I want to see the status of my KYC request on the blockchain	Integrate blockchain data with request status	20h

The Token Issuer KYC Request Sprint Backlog (Tableau 4.6) outlines the key tasks and objectives for this sprint, with a primary focus on enhancing the Know Your Customer (KYC) process specifically tailored for token issuers. The first task involves the creation of a dedicated KYC Request Submission Page, designed to simplify the submission process and improve user-friendliness. Additionally, the team will implement file upload functionality in the second task, enabling token issuers to securely upload the necessary documents for KYC verification. Ensuring the security and confidentiality of these documents, the third task aims to establish a robust storage system.

4.2.2.2 Investor KYC Submission :

Investor KYC Submission is a crucial process where investors provide personal information and documents for identity verification. It's essential for preventing fraud, ensuring compliance with regulations, and establishing trust in the financial industry.

4.2.2.2.1 Investor KYC Submission Use Case

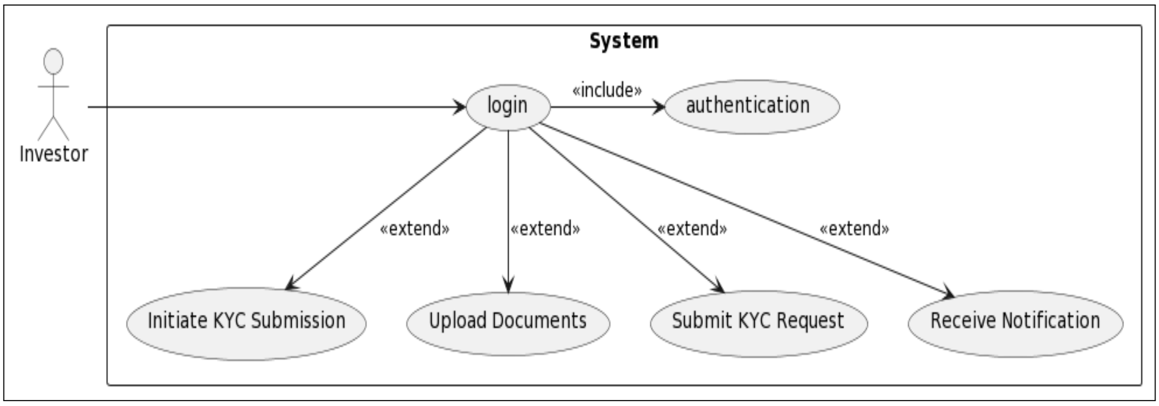


Figure 4.11: Investor KYC Submission use case

The Investor KYC Submission use case (figure 4.11) involves a step-by-step process for investors. First, the investor must log in and undergo authentication to access their account securely. Once authenticated, they are granted permission to initiate the KYC submission. During this phase, the investor fills out a KYC form with their personal information and uploads the necessary documents. Once the form is complete and all required documents are uploaded, the investor can submit their KYC request. This use case ensures that the investor's identity is verified, and their information is securely collected and stored, adhering to regulatory requirements and bolstering trust within the financial platform.

Tableau 4.7: Textual description of KYC Submission Use Case

<table>
<tr><th>Actor</th><th>Investor</th></tr>
<tr><td><b>Goal</b></td><td>Allows the investor to submit a KYC (Know Your Customer) request</td></tr>
<tr><td><b>Precondition</b></td><td>The investor is registered with the platform</td></tr>
<tr><td><b>Post-condition</b></td><td>KYC request is successfully submitted along with required documents</td></tr>
<tr><td colspan="2"><b>Nominal Scenario</b>
1. The investor logs into his account on the platform.
2. The investor selects the option to initiate KYC submission.
3. The system presents a form to the investor, requesting personal and contact information, as well as any necessary credentials.
4. The investor provides the required information and uploads documents.
5. Confirmation message displayed indicating successful submission.</td></tr>
<tr><td colspan="2"><b>Alternative</b>
4a. If files were not uploaded, the system presents an error message.
4b. In case of incomplete information, an error message will displayed.</td></tr>
</table>

In the nominal scenario, the Investor KYC Submission unfolds as follows: First, the investor securely logs into their platform account, establishing their identity. They then opt to initiate the KYC submission process, at which point the system presents a comprehensive form. This form solicits essential personal and contact details, along with any necessary documentation. The investor dutifully provides the requested information and uploads the required documents, ensuring that all components are complete. Following successful submission, a confirmation message affirms the completion of the KYC process.

However, in the alternative scenario, deviations may occur: If the investor neglects to upload the required files, the system promptly triggers an error message, prompting them to address this omission. Likewise, if any of the information fields within the form remain incomplete, an error message surfaces to guide the investor in rectifying the issue. These alternative paths ensure the completeness and accuracy of the KYC submission, vital for regulatory compliance and maintaining the security and trustworthiness of the financial platform.

4.2.2.2.2 Investor KYC Submission Sequence Diagram

This investor KYC Submission sequence diagram (figure 4.12) provides a representation of the Investor KYC Submission process. It illustrates the step-by-step interactions and flow of events between various components and actors involved in the KYC submission. Typically, it begins with the investor logging into their account on the platform, followed by their selection to initiate the KYC submission. The system then presents a form requesting personal and contact information, as well as any necessary credentials from the investor. The investor provides the required data and uploads the necessary documents. Finally, upon successful completion, a confirmation message is displayed. This sequence diagram offers a clear overview of the Investor KYC Submission process, emphasizing the interaction between the investor and the system in a concise and structured manner.

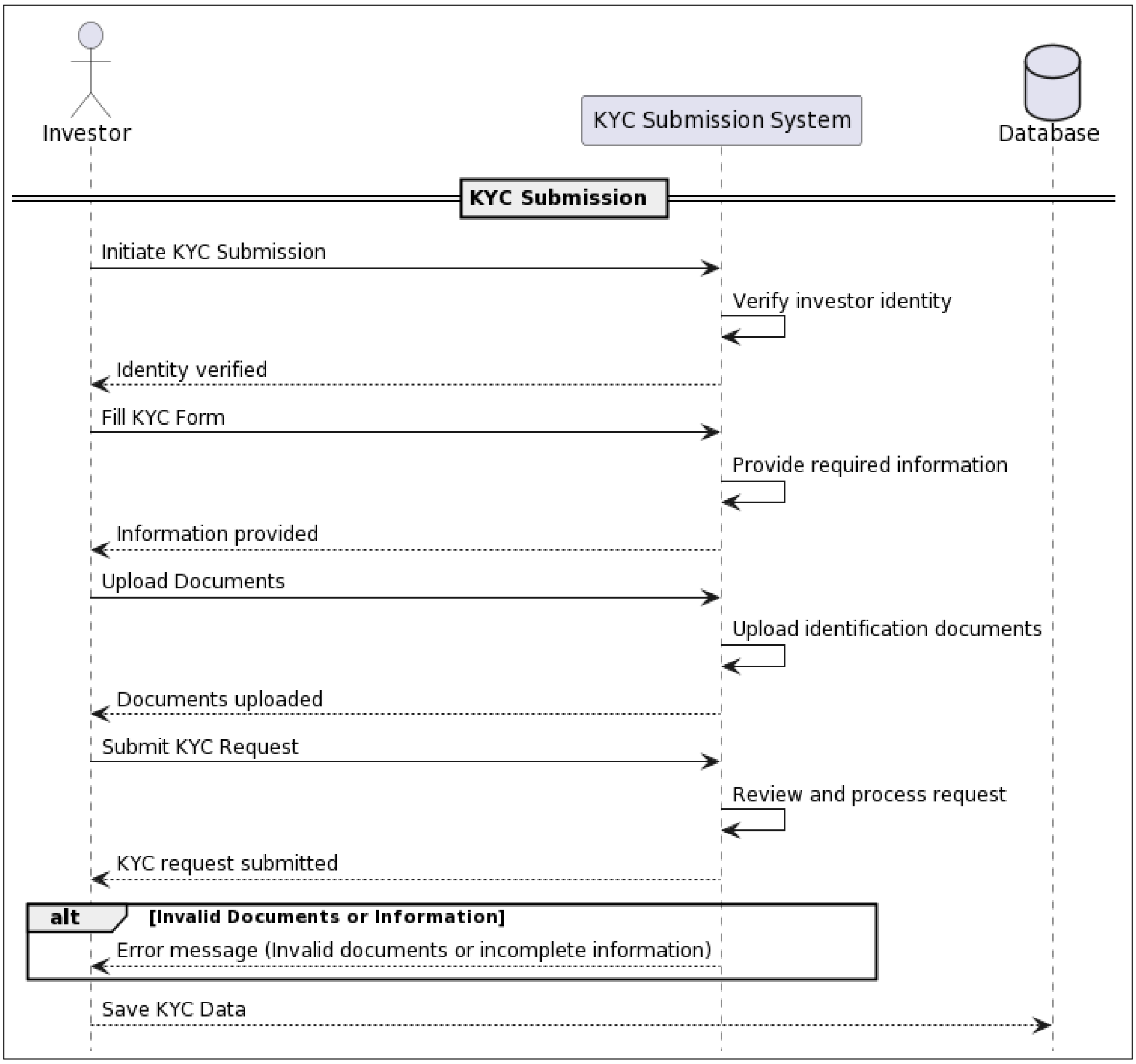


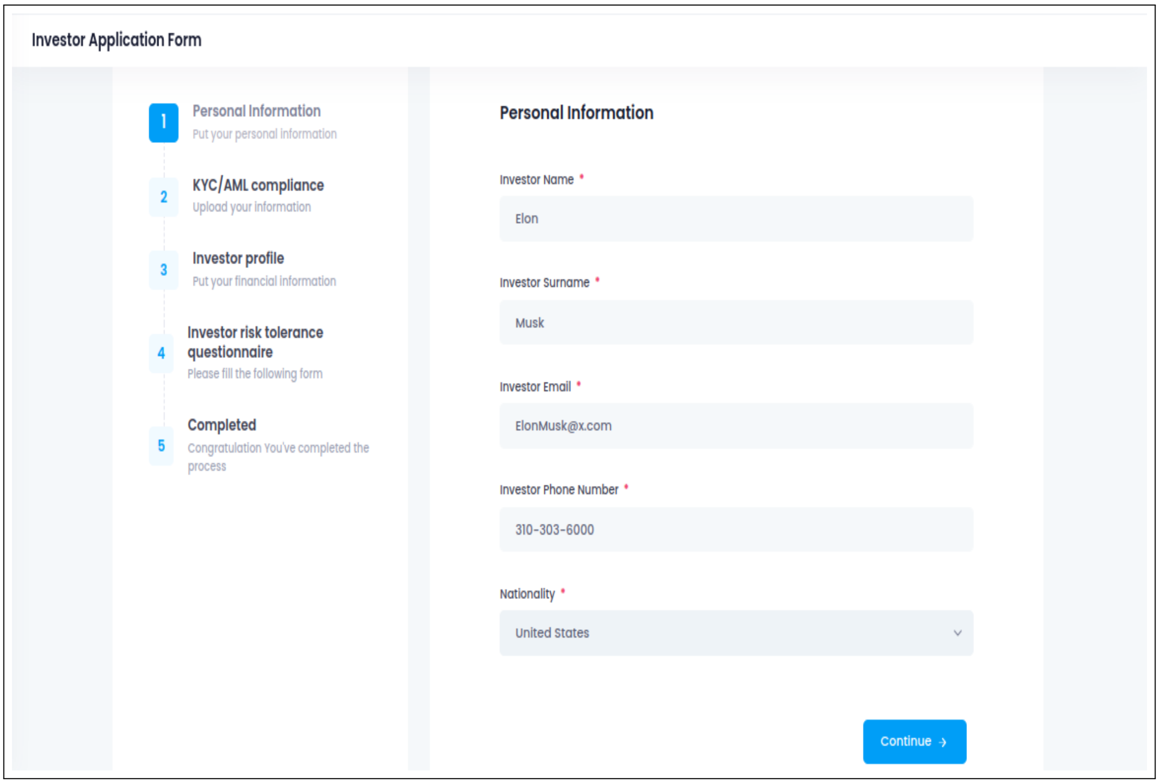
Figure 4.12: Investor KYC Submission sequence diagram

4.2.2.2.3 Submission Steps

The subsequent steps delineate the sequential actions that an investor needs to undertake in order to submit their Know Your Customer (KYC) request. These steps outline the procedural path an investor must follow to initiate and complete the KYC process.

1. **Fill Personal Information :**

   In the first step of KYC submission, the investor is required to provide several pieces of personal information, including their first name, last name (surname), email address, phone number, and nationality (figure 4.13) . This step is typically the starting point for verifying the investor's identity and compliance with regulatory requirements.



Investor Application Form

1 Personal Information
Put your personal information

2 KYC/AML compliance
Upload your information

3 Investor profile
Put your financial information

4 Investor risk tolerance questionnaire
Please fill the following form

5 Completed
Congratulation You've completed the process

Personal Information

Investor Name *
Elon

Investor Surname *
Musk

Investor Email *
ElonMusk@x.com

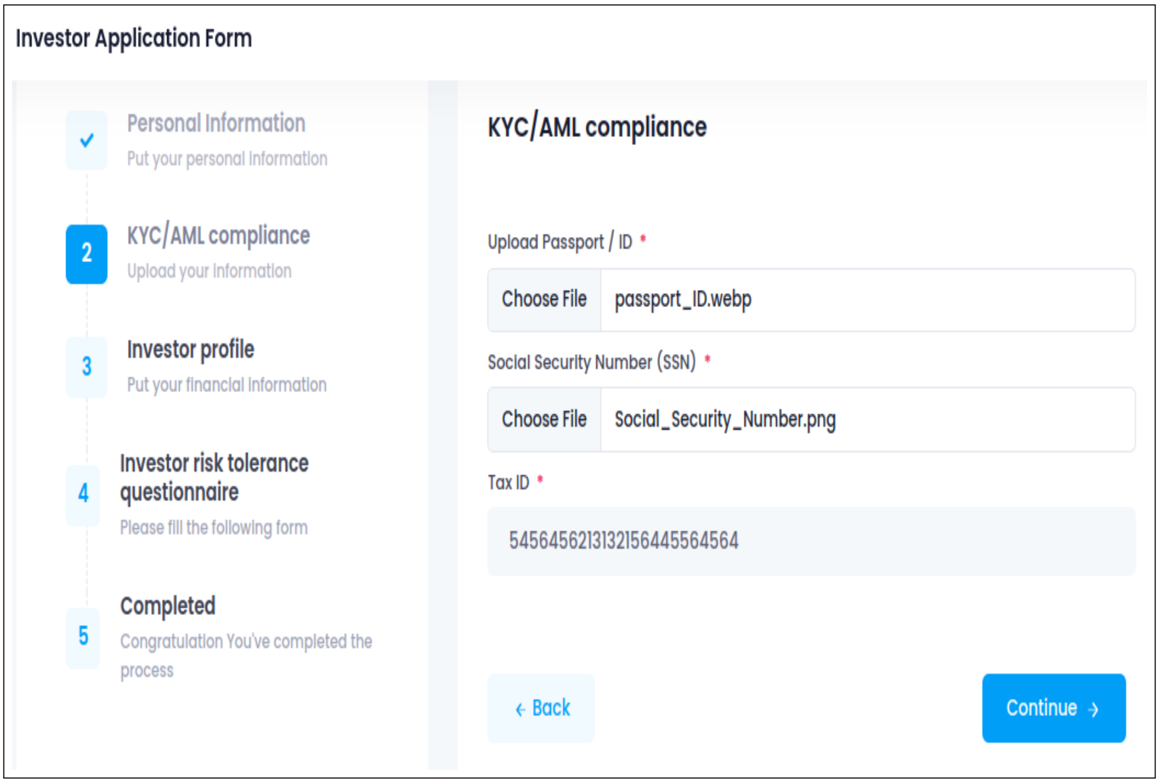
Investor Phone Number *
310-303-6000

Nationality *
United States

Figure 4.13: Fill Personal Information Interface

2. **Fill KYC/AML compliance :**

   In the second step of KYC submission, the investor is prompted to upload essential identification documents, such as a passport or ID card (figure 4.14). Additionally, the investor is asked to provide their Social Security Number and tax identification number (Tax ID). This step is a crucial part of the KYC process, as it helps verify the investor's identity and ensures compliance with regulatory requirements.


Investor Application Form

KYC/AML compliance

Upload Passport / ID *
passport_ID.webp

Social Security Number (SSN) *
Social_Security_Number.png

Tax ID *
5456456213132156445564564

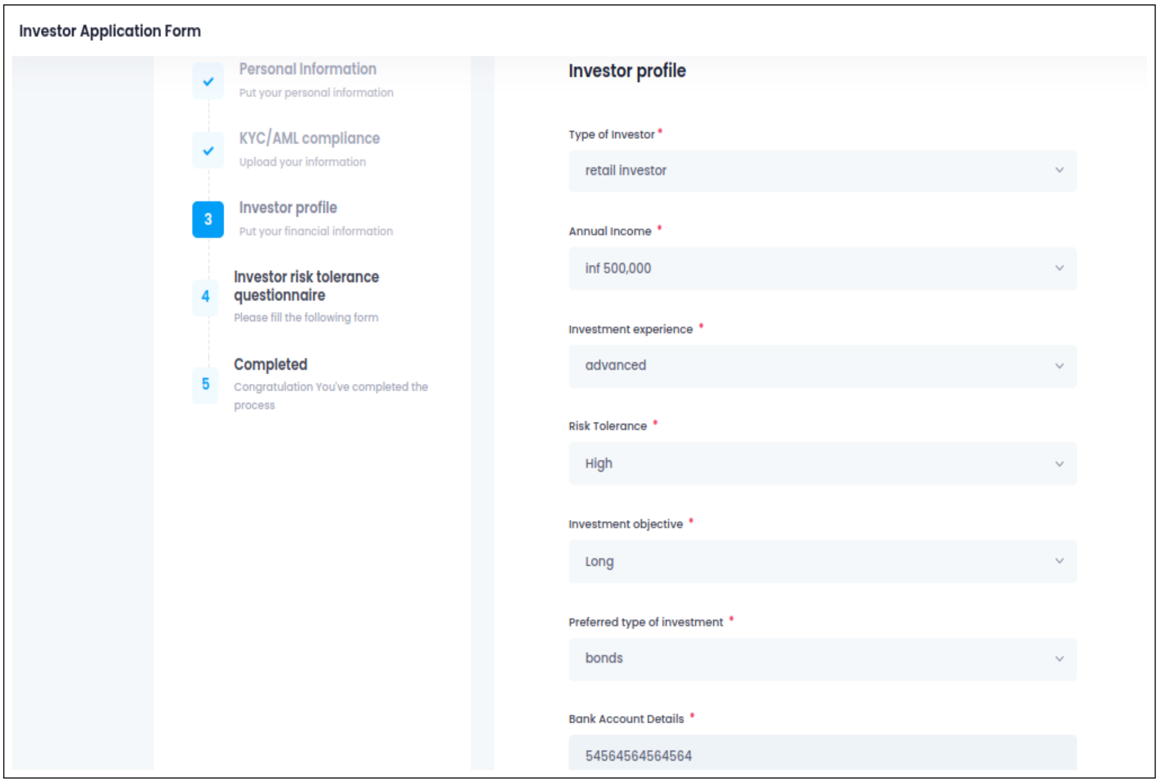
✓ Personal Information
Put your personal information
2 KYC/AML compliance
Upload your information
3 Investor profile
Put your financial information
4 Investor risk tolerance questionnaire
Please fill the following form
5 Completed
Congratulation You've completed the process
Choose File
Choose File
Back
Continue

Figure 4.14: Fill KYC/AML compliance Interface

3. **Fill investor profile :**

In the third step of KYC submission, the investor is required to complete their personal profile by providing essential information (figure 4.15). This includes specifying their investor type, such as "beginner," "intermediate," or "advanced," which helps tailor investment recommendations based on their experience level. The investor is also prompted to disclose their annual income, a critical factor for determining suitable investment opportunities.

Additionally, the interface collects data regarding the investor's investment experience, aiding in the customization of investment suggestions to align with their background. The investor is asked about their risk tolerance level, allowing the platform to recommend investment products that match their risk preferences, such as "low risk," "medium risk," or "high risk." Moreover, the investor specifies their preferred type or category of investment, whether it's stocks, bonds, real estate, or other asset classes.



Investor Application Form

✓ Personal Information
Put your personal information

✓ KYC/AML compliance
Upload your information

3 Investor profile
Put your financial information

4 Investor risk tolerance questionnaire
Please fill the following form

5 Completed
Congratulation You've completed the process

Investor profile

Type of Investor *
retail investor

Annual Income *
inf 500,000

Investment experience *
advanced

Risk Tolerance *
High

Investment objective *
Long

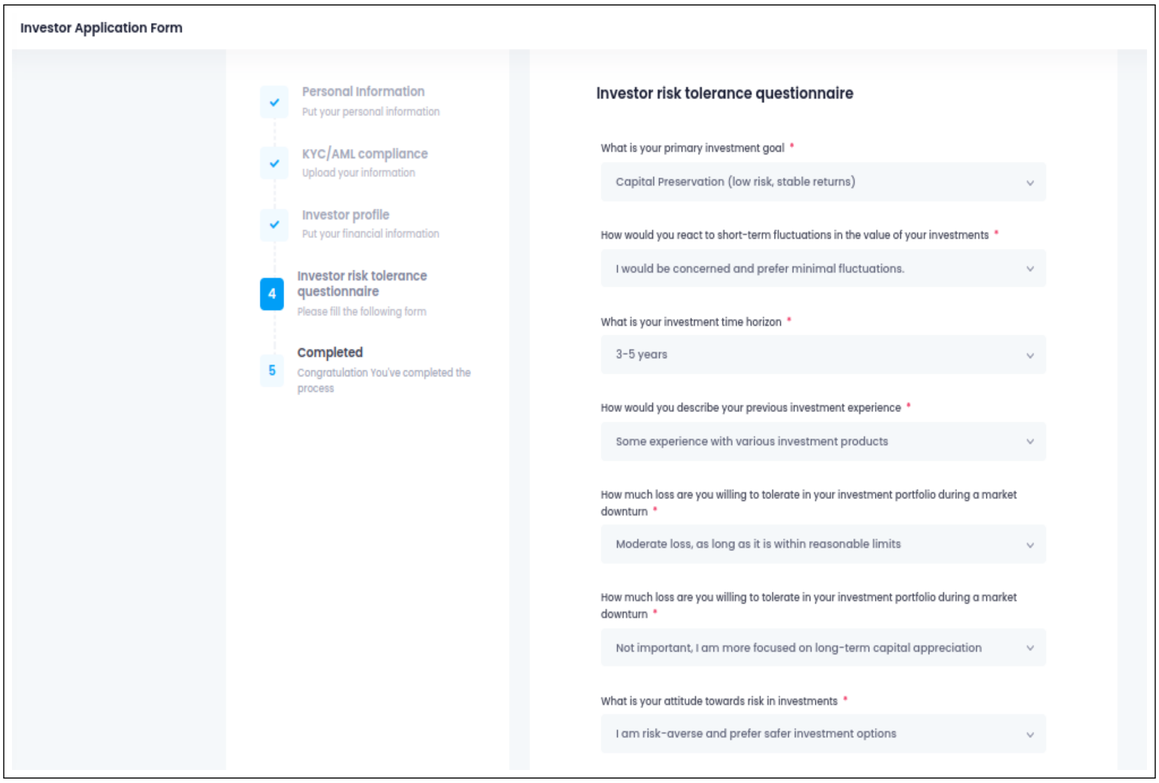
Preferred type of investment *
bonds

Bank Account Details *
54564564564564

Figure 4.15: Fill Investor Profile Interface

4. **Fill tolerance questionnaire :**

   In the fourth step of KYC submission, the investor is asked to complete a comprehensive risk tolerance questionnaire that probes various fundamental aspects of their investment preferences and their attitude toward risk (figure 4.16). The questionnaire encompasses key dimensions, including the investor's primary investment goal, their reaction to short-term fluctuations in financial markets, their intended investment time horizon, their past investment experience, their typical response during market downturns or economic instability, and their overall attitude toward risk in investment.



Investor Application Form

✓ Personal Information
Put your personal information

✓ KYC/AML compliance
Upload your information

✓ Investor profile
Put your financial information

4 Investor risk tolerance questionnaire
Please fill the following form

5 Completed
Congratulation You've completed the process

Investor risk tolerance questionnaire

What is your primary investment goal *
Capital Preservation (low risk, stable returns)

How would you react to short-term fluctuations in the value of your investments *
I would be concerned and prefer minimal fluctuations.

What is your investment time horizon *
3-5 years

How would you describe your previous investment experience *
Some experience with various investment products

How much loss are you willing to tolerate in your investment portfolio during a market downturn *
Moderate loss, as long as it is within reasonable limits

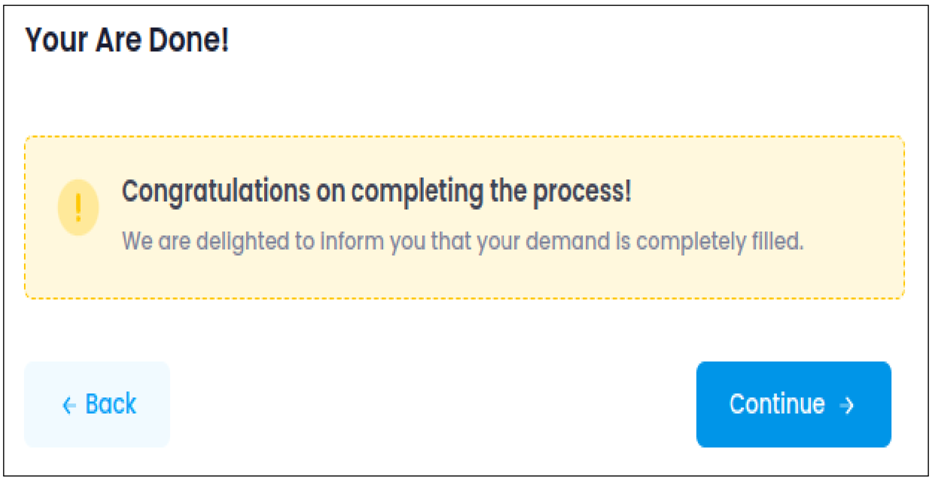
How much loss are you willing to tolerate in your investment portfolio during a market downturn *
Not important, I am more focused on long-term capital appreciation

What is your attitude towards risk in investments *
I am risk-averse and prefer safer investment options

Figure 4.16: Fill tolerance questionnaire interface

5. **Submitting Request :**

In the fifth step of KYC submission, investor has reached the concluding phase of their submission, and the screenshot illustrates that they are required to take the final action of clicking on the "Submit" button to formally submit their KYC (figure 4.17). This step marks the culmination of the KYC process, where all the necessary information. By clicking the "Submit" button, the investor initiates the official processing of their request, which includes the evaluation of their suitability for various financial products and services.



Your Are Done!

Congratulations on completing the process!

We are delighted to inform you that your demand is completely filled.

Figure 4.17: Submitting Request Interface

4.2.2.3 Token Issuer KYC Submission :

Token Issuer KYC Submission is the process where users issuing tokens, provide the required documentation to verify their identity and legitimacy. This procedure holds significant importance for various reasons. Firstly, it ensures compliance with regulatory requirements, such as Know Your Customer (KYC) and Anti-Money Laundering (AML) regulations, which are essential to avoid legal complications and penalties.

4.2.2.3.1 Token Issuer KYC Submission Use Case

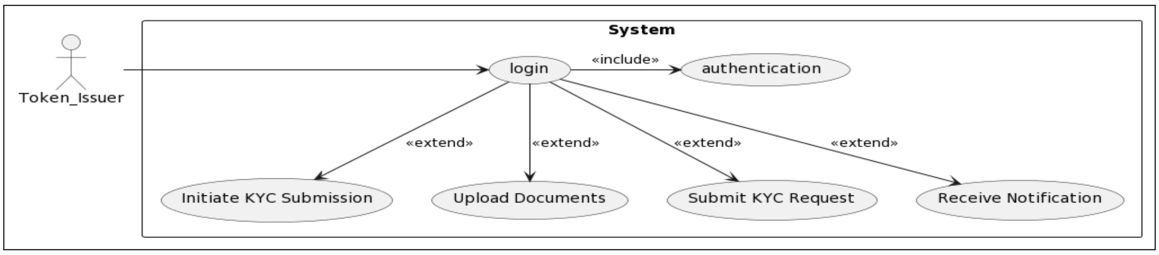


Figure 4.18: Token Issuer KYC submission use case

This use case (figure 4.18) is vital for ensuring that token issuers comply with legal and regulatory standards, verify their legitimacy, and build trust with investors and partners.

Tableau 4.8: Textual description of Token Issuer KYC Submission Use Case

Actor	Token Issuer
Goal	Allows the token issuer to submit a KYC (Know Your Customer) request.
Precondition	The token issuer is registered on the platform
Post-condition	KYC request is successfully submitted along with required documentation
Nominal Scenario 1. The token issuer logs into his account on the platform. 2. The token issuer initiates KYC submission. The system presents a form to the investor, requesting personal and contact information 3. The token issuer provides the required information and uploads documents. 4. Confirmation message displayed indicating successful submission	
Alternative 4a. If files were not uploaded or incomplete information , the system presents an error message.	

The textual description of the Token Issuer KYC Submission Use Case (Tableau 4.8) provides a comprehensive description of KYC request submission use case. In this scenario, the key actor is the

Token Issuer, and the primary objective is to enable the submission of a Know Your Customer (KYC) request. It is assumed that the token issuer is already registered on the platform as a precondition.

The nominal scenario unfolds as follows: Firstly, the token issuer accesses their account on the platform by logging in. Subsequently, they initiate the KYC submission process, prompting the system to present a form. This form requests essential personal and contact information from the token issuer.Finally, upon successful completion of this process, a confirmation message is displayed, indicating that the KYC request has been successfully submitted.

However, the use case also anticipates alternative paths. If the token issuer neglects to upload the necessary files, the system intervenes by presenting an error message, guiding them to rectify the omission.

4.2.2.3.2 Token Issuer KYC Submission Sequence Diagram

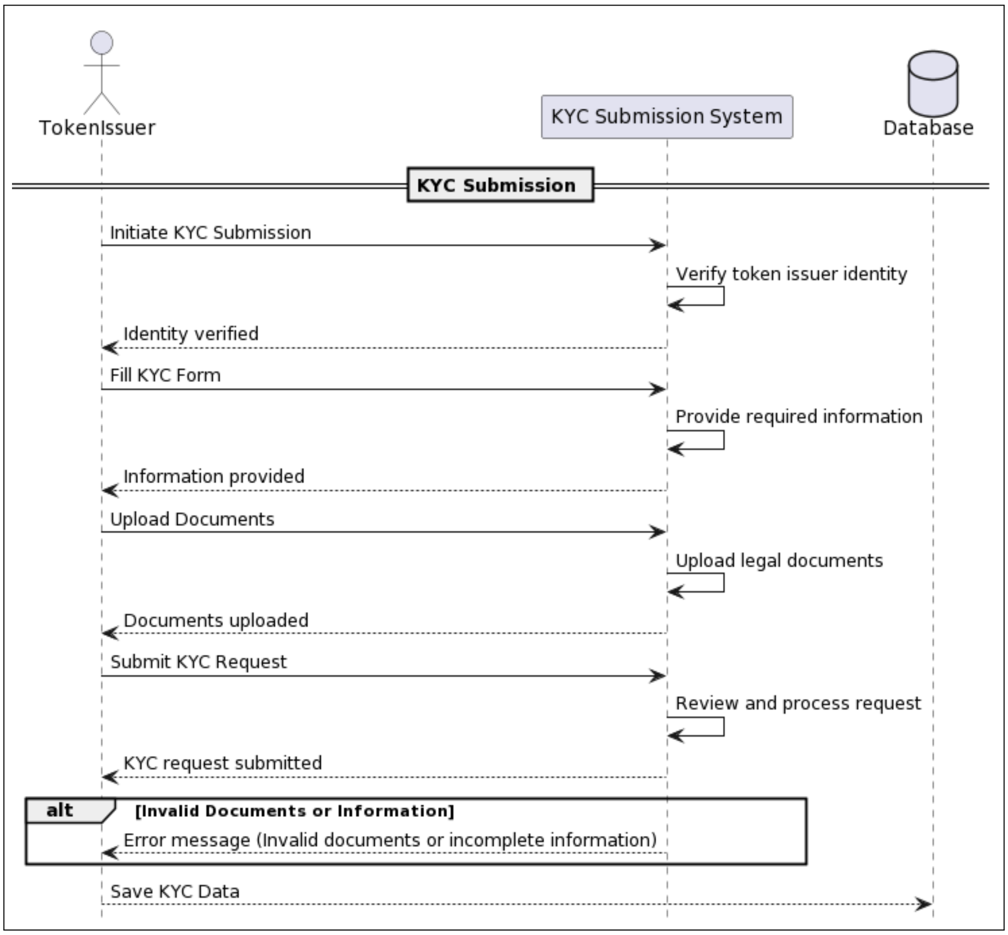


Figure 4.19: Token Issuer KYC Submission sequence diagram

Token Issuer KYC Submission sequence diagram (figure 4.19) illustrates the sequential steps involved in the Token Issuer's Know Your Customer (KYC) submission process. The user is asked to fill in essential personal and contact information. Additionally, the form allows the token issuer to upload the necessary documents required for identity verification.

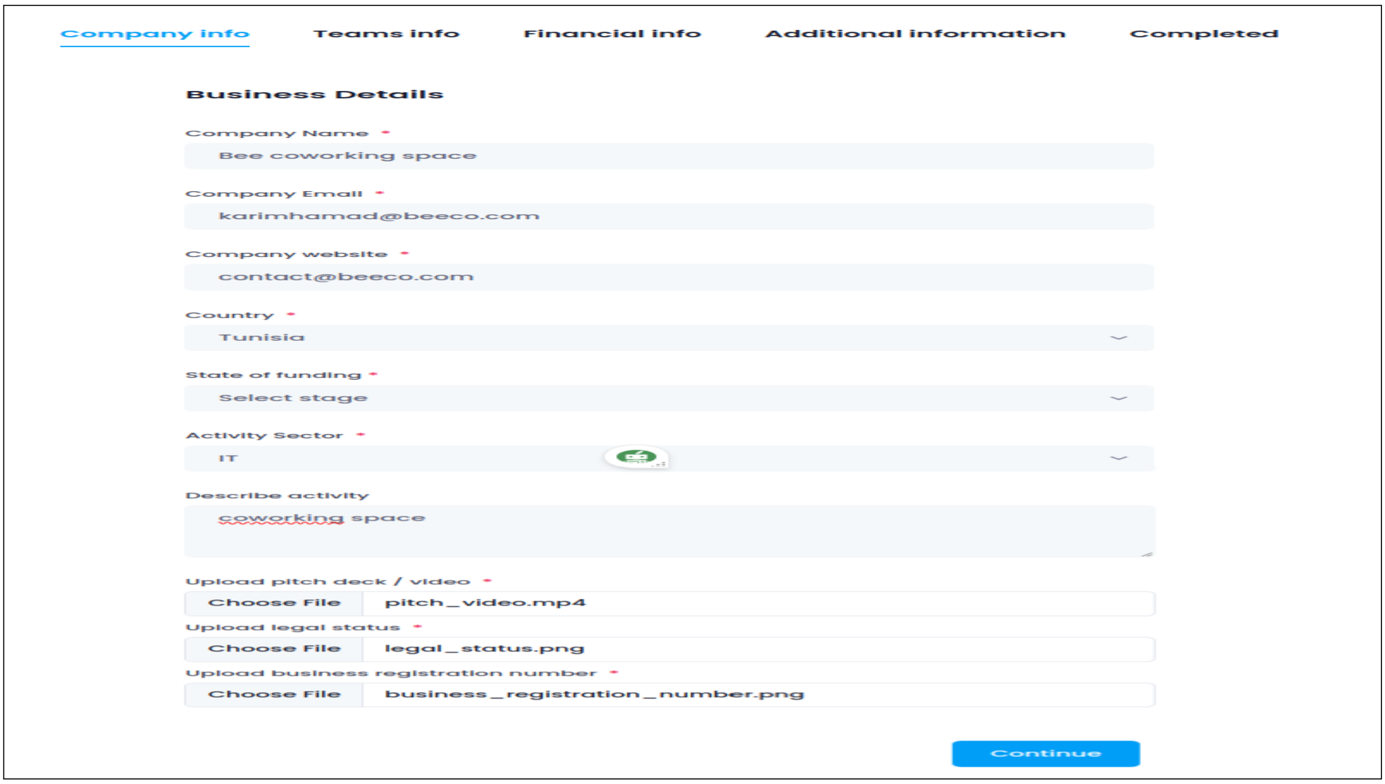
Importantly, the sequence diagram accounts for error handling. If the documents provided by the token issuer or the information submitted are found to be invalid or incomplete during the verification process, the system generates an error message.

4.2.2.3.3 Submission Steps :

The subsequent steps delineate the sequential actions that token issuer needs to undertake in order to submit his own KYC request.

1. **Fill Company Information :**

   In the first step of KYC submission, the token issuer is asked to provide specific details including the company name, company email address, company website, country of operation, state of funding, activity sector, comprehensive description of the company's activities and upload of a pitch or video presentation, the legal status of the company, and the provision of the business registration number (figure 4.20).



Business Details

Company Name *
Bee coworking space

Company Email *
karimhamad@beeco.com

Company website *
contact@beeco.com

Country *
Tunisia

State of funding *
Select stage

Activity Sector *
IT

Describe activity
coworking space

Upload pitch deck / video *
Choose File pitch_video.mp4

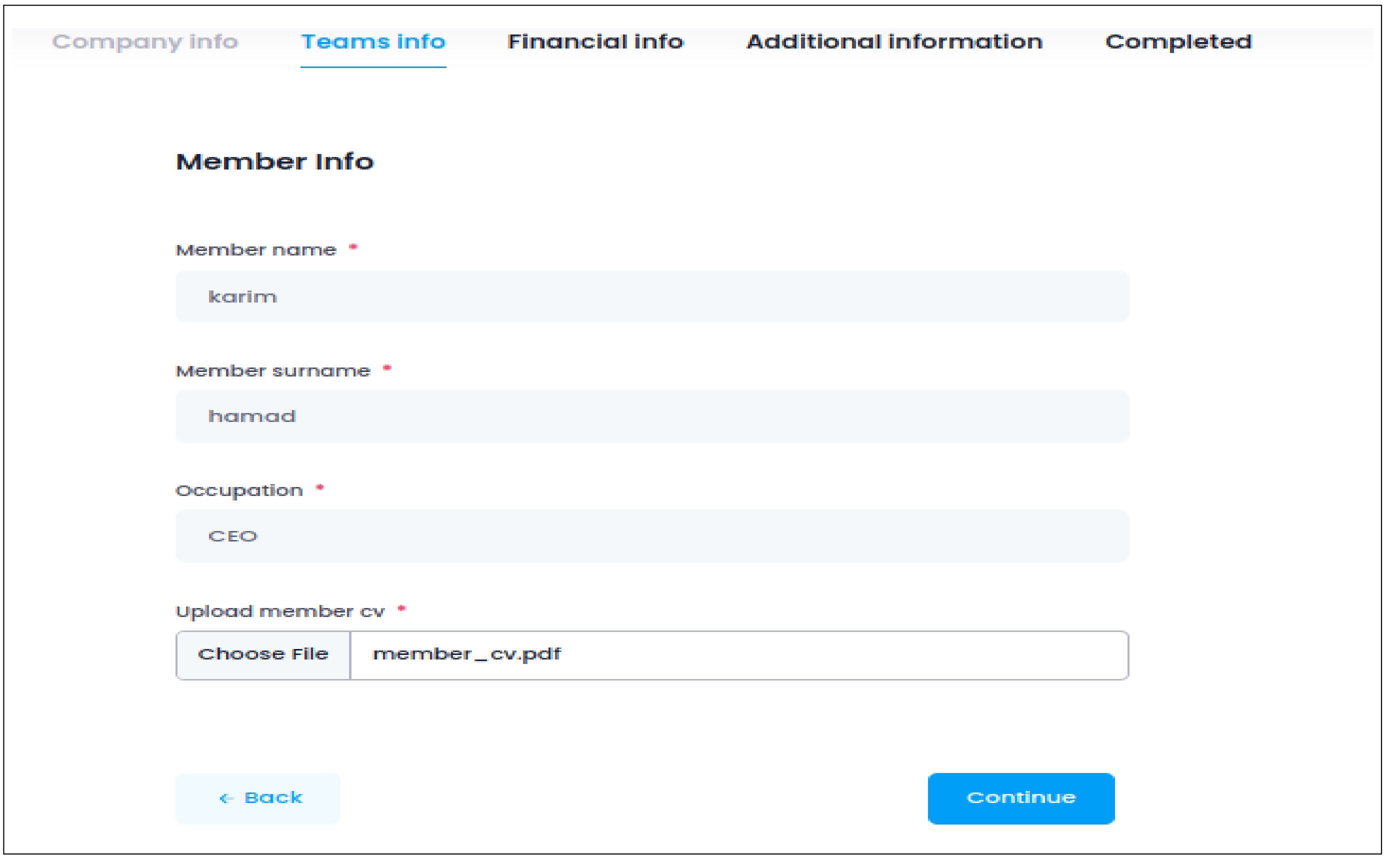
Upload legal status *
Choose File legal_status.png

Upload business registration number *
Choose File business_registration_number.png

Figure 4.20: Fill Company Information Interface

2. **Fill Team Information :**

   In the second step of KYC submission, the token issuer is required to provide detailed information about their team members (figure 4.21). This information includes the names and surnames of each team member, along with their respective positions within the organization. Additionally, the token issuer must upload the curriculum vitae (CV) of each team member.



Member Info

Member name *

karim

Member surname *

hamad

Occupation *

CEO

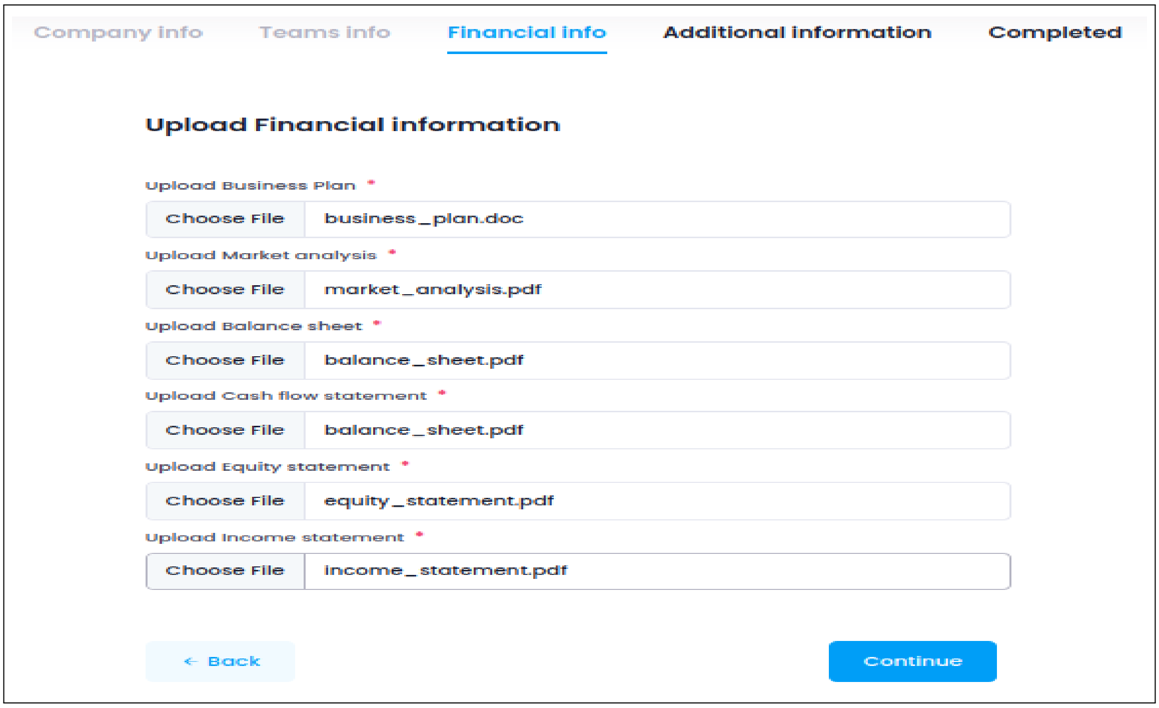
Upload member cv *

Choose File member_cv.pdf

Figure 4.21: Fill Team Information Interface

3. **Fill Financial Information :** In the third step of KYC submission, the token issuer is required to provide comprehensive financial information. This includes the submission of essential documents such as the business plan, market analysis, balance sheet, cash flow statement, equity statement, and income statement (figure 4.22).

   These financial documents play a pivotal role in the KYC process, as they offer valuable insights into the financial health and viability of the token issuer's project. They allow potential investors and regulatory bodies to assess the feasibility and sustainability of the venture. By uploading these documents, the token issuer demonstrates transparency and a commitment to providing a complete and accurate picture of their project's financial standing.



Upload Financial information

Upload Business Plan *
Choose File business_plan.doc

Upload Market analysis *
Choose File market_analysis.pdf

Upload Balance sheet *
Choose File balance_sheet.pdf

Upload Cash flow statement *
Choose File balance_sheet.pdf

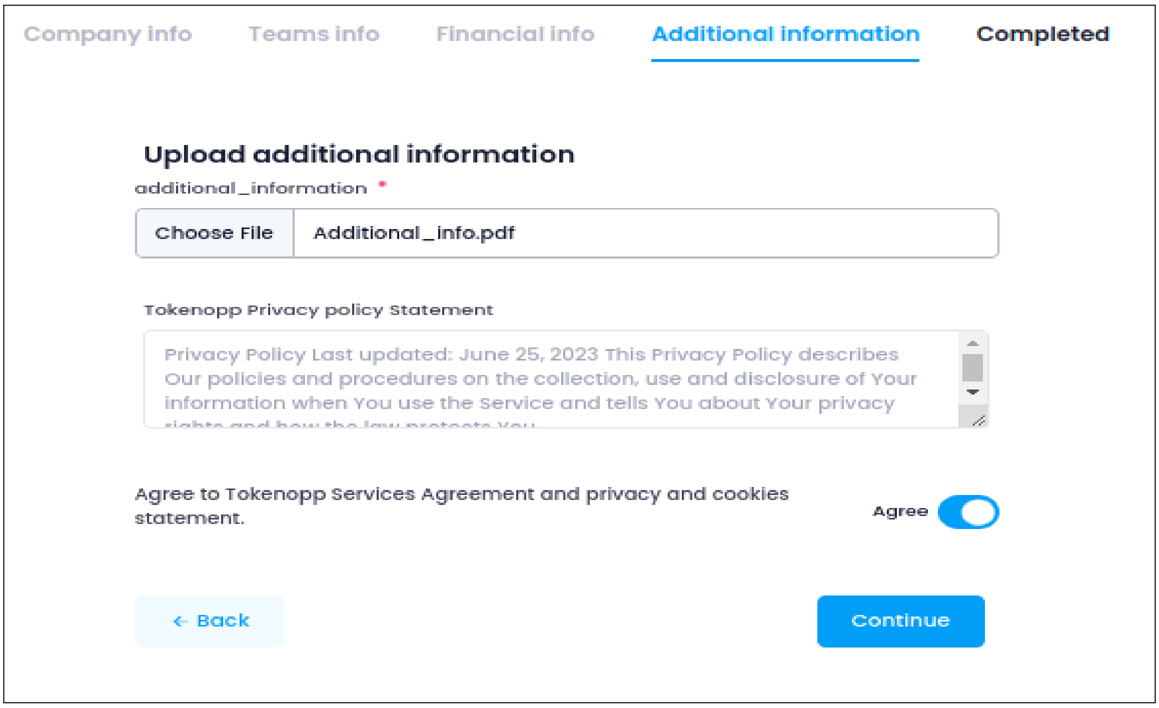
Upload Equity statement *
Choose File equity_statement.pdf

Upload Income statement *
Choose File income_statement.pdf

Figure 4.22: Fill Financial Information Interface

4. **Fill Additional Information :**

In the fourth step of KYC submission, the token issuer is required to provides additional information beyond the basic documentation (figure 4.23). In this step, the token issuer is prompted to fill out a form with the required supplementary details.



Upload additional information

additional_information *
Choose File Additional_info.pdf

Tokenopp Privacy policy Statement

Privacy Policy Last updated: June 25, 2023 This Privacy Policy describes Our policies and procedures on the collection, use and disclosure of Your information when You use the Service and tells You about Your privacy rights and how the law protects You

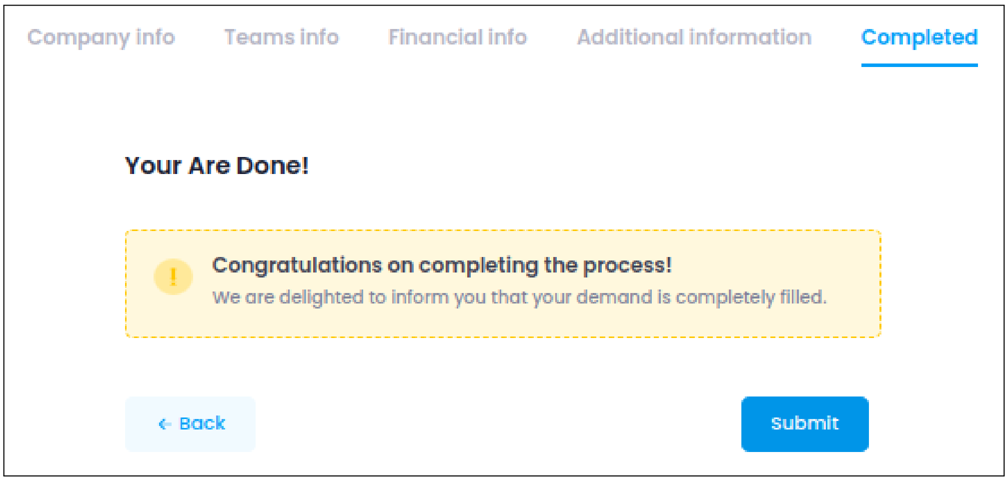
Agree to Tokenopp Services Agreement and privacy and cookies statement.

Agree ●

Figure 4.23: Fill Additional Information Interface

5. **Request submission :**

In the fourth step of KYC submission, the token issuer is required , the token issuer is required to take the last action by clicking the "Submit" button(figure 4.24). This action formally submits their KYC request for review and verification.



Your Are Done!

Congratulations on completing the process!
We are delighted to inform you that your demand is completely filled.

Figure 4.24: Request submission interface

4.3 Release 02

In this release, our primary focus is on enhancing user management functionalities by providing the admin with more control over the user request process. The main objective is to streamline the user verification process and improve overall user experience. This release is divided into two sprints, and this introduction will provide an overview of Sprint One.

Our second release will consist of one sprint:

User Management : This sprint focuses on the tasks and activities related to user management handled by administrator.

4.3.1 Sprint One: User Management

We will go deep into User Management, a vital component of our project. This sprint focuses on improving administrators' skills through a variety of functions while maintaining a uniform experience for both administrators and users.

4.3.1.1 Sprint One Backlog

In this sprint ,the main focus is on introducing new features to enhance administrative capabilities. These features include an admin dashboard that allows administrators to effortlessly view all deposited requests. Additionally, a "view details" feature is being added, enabling administrators to access comprehensive information about each KYC request with ease.

Tableau 4.9: User Management Sprint Backlog table

ID	User Story	Task	Estimation
2.1	As admin, I want to view All deposited KYC requests	Create admin dashboard	40h
2.2	As admin, I want to view all details about each request	Create view details modal	20h
2.3	As admin, I want to download uploaded files and view them	Configure admin dashboard to allow him to download files	20h
3.4	As admin, I want to set the user status (Process, Accepted, , Rejected)	Allow admin to set user status	40h

Furthermore, administrators will gain the ability to download uploaded files associated with KYC requests and view them directly through the admin dashboard. Lastly, administrators will have the capability to set user statuses, marking them as "In Process," "Accepted," or "Rejected".

4.3.1.1.1 Sprint Use Case:

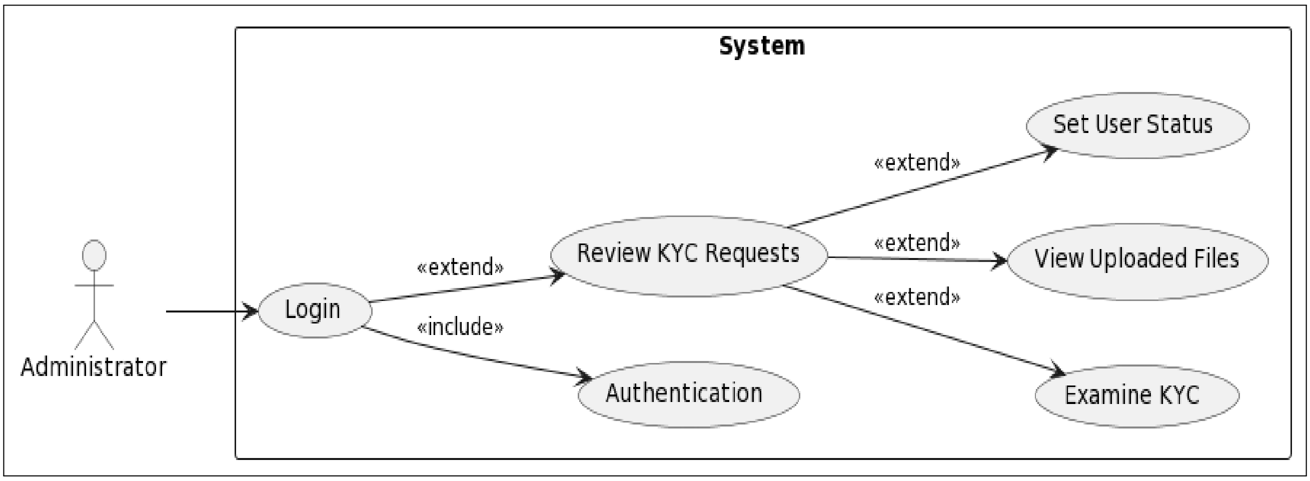


Figure 4.25: Sprint Two use case

This use case (figure 4.25) is designed to enable the administrator to review KYC requests from investors and startups, view uploaded files, and manage user status and set decision concerning their request ("In Process," "Accepted," or "Rejected"). This streamlined process enhances regulatory compliance and user management within the system.

4.3.1.2 KYC Request Overview

The KYC Request Overview for administrators provides a streamlined and efficient way for them to review and manage KYC (Know Your Customer) requests submitted by both startups and investors.

In the Nominal Scenario, the user scenario unfolds as follows: First, the administrator initiates the process by logging into the admin dashboard. Upon successful login, the dashboard promptly displays a consolidated list encompassing all the deposited KYC (Know Your Customer) requests, allowing for a comprehensive overview.

However, there exists an alternative scenario as well: If no KYC requests are currently available in the system, the dashboard responds by displaying a message to inform the administrator of the absence of pending requests, ensuring clarity and transparency in system communication.

Tableau 4.10: Textual Description Of KYC Request Overview Use Case

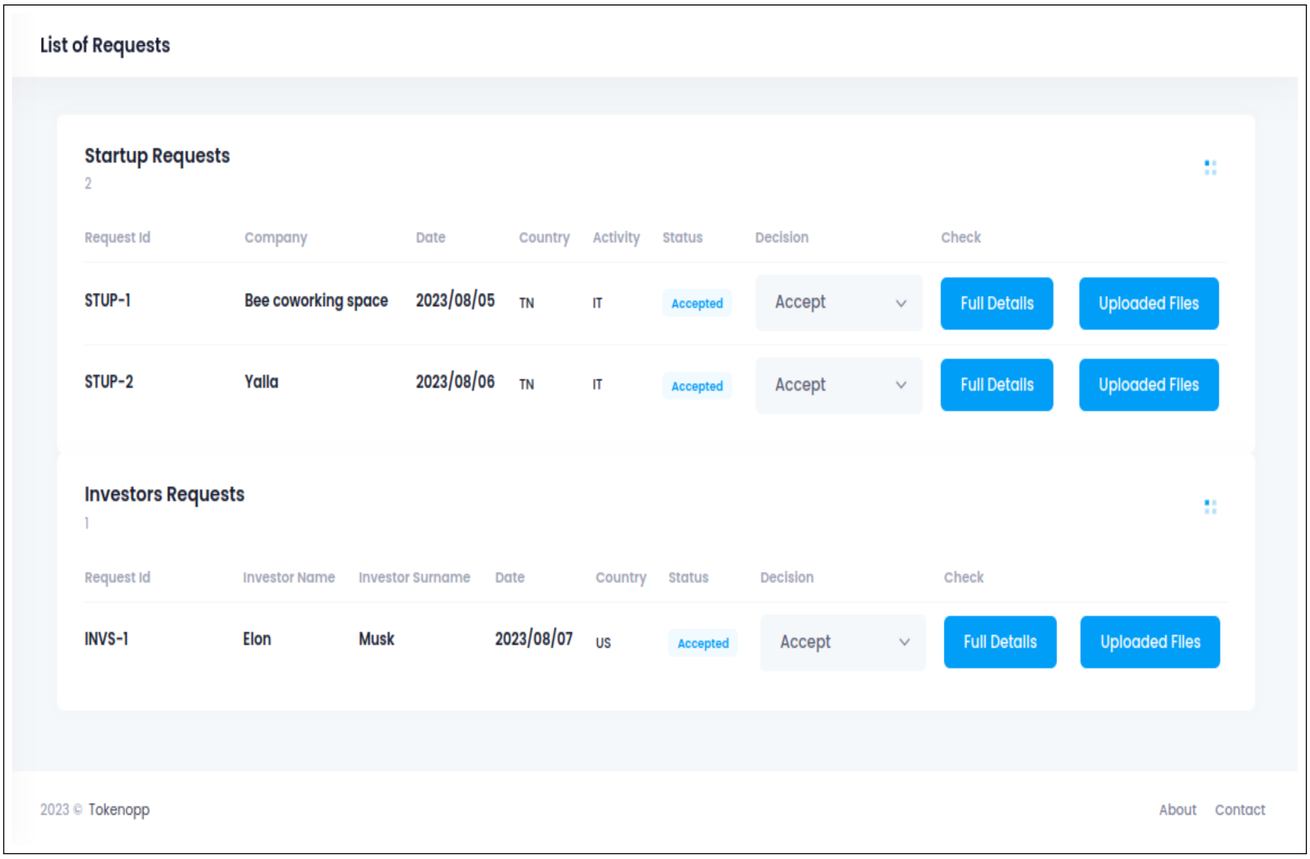
Actor	Administrator
Goal	Provide administrators with a comprehensive view of KYC requests
Precondition	Administrator is logged into the admin dashboard
Post-condition	Administrators gain insights into the overall status of KYC requests

Nominal Scenario

1. The administrator logs into the admin dashboard.
2. The dashboard presents a consolidated list of all deposited KYC requests.
3. The administrator can apply filters, sorting, and search options to refine the view.
4. Each request entry provides essential details.
5. Administrators can make informed decisions based on the overview.

Alternative

5. If no KYC requests are available, the dashboard displays a message.



List of Requests

Startup Requests

2

Request Id	Company	Date	Country	Activity	Status	Decision	Check	
STUP-1	Bee coworking space	2023/08/05	TN	IT	Accepted	Accept	Full Details	Uploaded Files
STUP-2	Yalla	2023/08/06	TN	IT	Accepted	Accept	Full Details	Uploaded Files

Investors Requests

1

Request Id	Investor Name	Investor Surname	Date	Country	Status	Decision	Check	
INVS-1	Elon	Musk	2023/08/07	US	Accepted	Accept	Full Details	Uploaded Files

2023 © Tokenopp

Figure 4.26: Admin Dashboard Interface

The Admin dashboard interface (figure 4.26) shows the Key functionalities of administrator which includes the ability to view uploaded documents associated with each request, facilitating the examination and verification process. Additionally, administrators can download these documents when necessary for further evaluation or record-keeping purposes.

4.3.1.3 KYC Request Examination

The KYC Request Examination feature allows administrators to comprehensively examine all requests submitted by startups and investors. Administrators can access detailed information about these requests, which includes data and documentation provided by users. This functionality empowers administrators to conduct thorough evaluations.

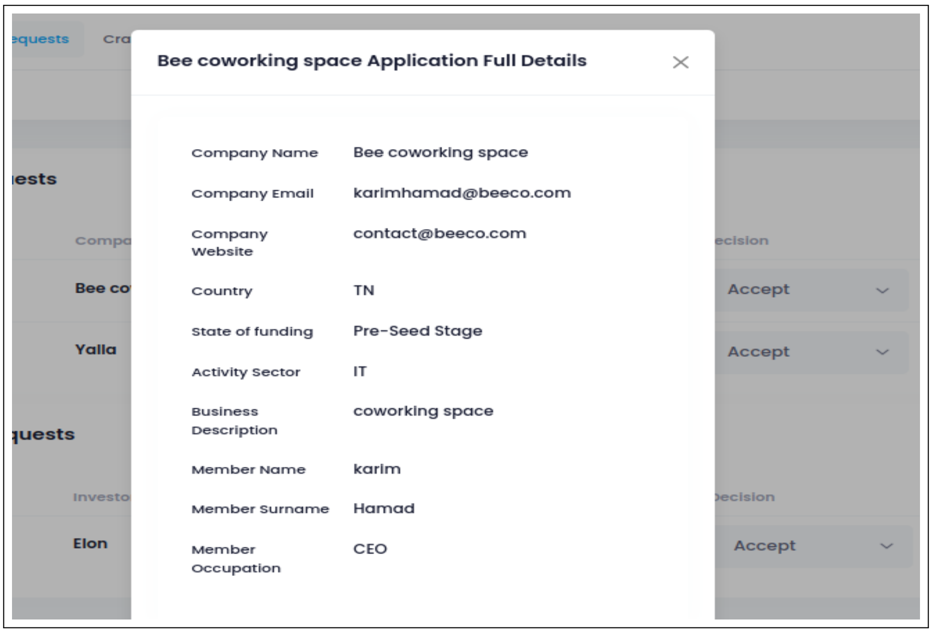
Tableau 4.11: Textual Description Of KYC Request Examination Use Case

Actor	Administrator
Goal	Enable administrators to gain a deeper understanding of individual KYC requests
Precondition	Administrator is logged into the admin dashboard
Post-condition	Administrators can obtain comprehensive insights into specific KYC requests
Nominal Scenario 1. The administrator logs into the admin dashboard. 2. The dashboard displays a list of KYC requests. 3. The administrator clicks on a specific request entry. 4. A "view details" modal window opens, presenting detailed information. 5. Administrators can evaluate the request thoroughly without leaving the dashboard.	
Alternative 5. If no detailed information is available, the modal displays a message.	

The textual description of KYC Request Examination Use Case (tableau 4.11) outlines the feature of examination added to administrator to gain a deeper understanding of individual KYC requests submitted by users, such as startups and investors. To achieve this, administrators must first log into the admin dashboard, ensuring proper authentication. Once logged in, the dashboard displays a list of KYC requests. Administrators can then click on a specific request entry to access a "view details" modal window, which presents comprehensive information about the selected request. Within this modal, administrators can thoroughly assess the request without leaving the dashboard. In an alternative scenario where no detailed information is available, the modal displays a message to inform administrators. This use case enhances efficiency and transparency in the KYC request examination process, contributing to regulatory compliance and informed decision-making.

4.3.1.3.1 View Full Details Of Startup Request

Request full detail interface concerning startup request (figure 4.27) provides a visual representation of a comprehensive view of a startup's KYC (Know Your Customer) request details within the system. This figure showcases a detailed breakdown of information submitted by the startup for verification and compliance purposes. It serves as a critical tool for administrators to conduct in-depth assessments, review essential data, and ensure regulatory compliance while making informed decisions regarding the startup's KYC request.



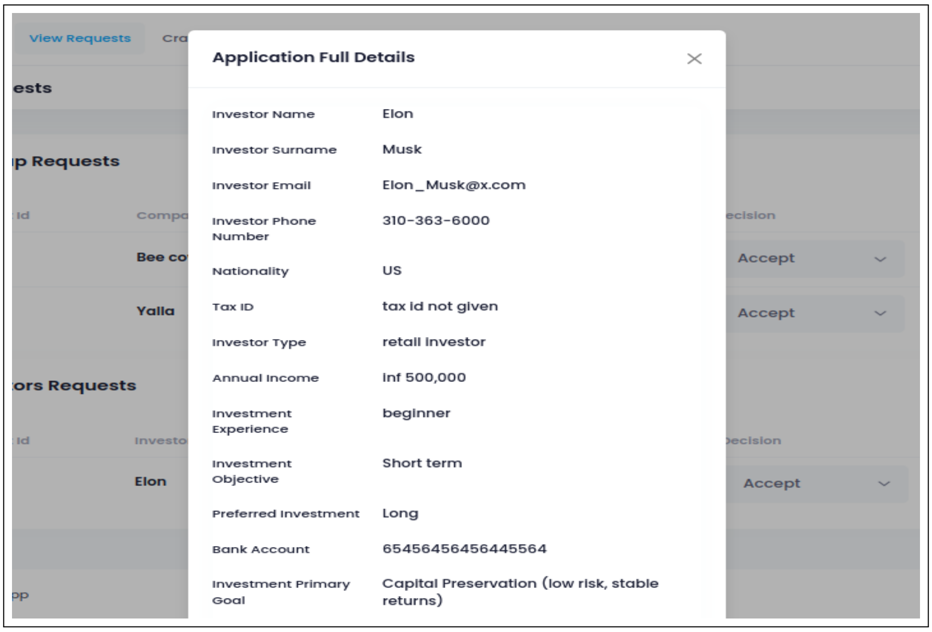
Bee coworking space Application Full Details

Company Name	Bee coworking space
Company Email	karimhamad@beeco.com
Company Website	contact@beeco.com
Country	TN
State of funding	Pre-Seed Stage
Activity Sector	IT
Business Description	coworking space
Member Name	karim
Member Surname	Hamad
Member Occupation	CEO

Figure 4.27: Startup KYC-Request full details Interface

4.3.1.3.2 View Full Details Of Investor Request

Request full detail interface concerning investor request (figure 4.28) presents a visual representation of the complete details of an investor's KYC (Know Your Customer) request within the system. This figure offers a comprehensive overview of the information and documents submitted by the investor for verification and compliance purposes. It serves as a valuable tool for administrators to conduct a thorough evaluation, review crucial data, and ensure regulatory compliance while making informed decisions regarding the investor's KYC request.



Application Full Details

Investor Name	Elon
Investor Surname	Musk
Investor Email	Elon_Musk@x.com
Investor Phone Number	310-363-6000
Nationality	US
Tax ID	tax id not given
Investor Type	retail investor
Annual Income	Inf 500,000
Investment Experience	beginner
Investment Objective	Short term
Preferred Investment	Long
Bank Account	65456456456445564
Investment Primary Goal	Capital Preservation (low risk, stable returns)

Figure 4.28: Investor KYC-Request full details interface

4.3.1.4 Document Access and Review

Document Access and Review is a crucial function for administrators, enabling them to verify and ensure compliance with user-submitted documents in KYC requests. This feature helps mitigate risks, supports informed decision-making, maintains transparency, and creates an audit trail, ensuring the platform's integrity and adherence to regulatory standards.

Tableau 4.12: Textual Description Of Document Access and Review Use Case

Actor	Administrator
Goal	Facilitate administrators' access to and review of uploaded documents
Precondition	Administrator is logged into the admin dashboard
Post-condition	Administrators can access and review uploaded documents

Nominal Scenario

1. The administrator logs into the admin dashboard.
2. The dashboard displays a list of KYC requests.
3. The administrator selects a request entry.
4. Within the request details, the administrator can click on uploaded documents to view them.
5. Administrators can choose to download the documents for offline review.

Alternative

5. If no documents are uploaded, the dashboard displays a message.

The textual description of document access and review use case (tableau 4.12) , designed to facilitate administrators' access to and review of uploaded documents. In this scenario, administrators are required to be logged into the admin dashboard, ensuring proper authentication. The key objective is to empower administrators to access and thoroughly review documents associated with requests submitted by users.

In an alternative scenario where no documents have been uploaded for a specific request, the dashboard thoughtfully displays a message, ensuring that administrators are promptly informed of the absence of documents, thereby promoting transparency and clarity in system communication.

4.3.1.4.1 View Uploaded Files By The Startup

Startup Uploaded Files Review Interface (figure 4.29) provides a visual representation of the interface where administrators can review files uploaded by startups within the system.administrators have access to a detailed overview of the files submitted by startups as part of their requests. This functionality is essential for administrators as it allows them to verify the legitimacy of documents, assess compliance, and ensure the accuracy of information provided by startups.

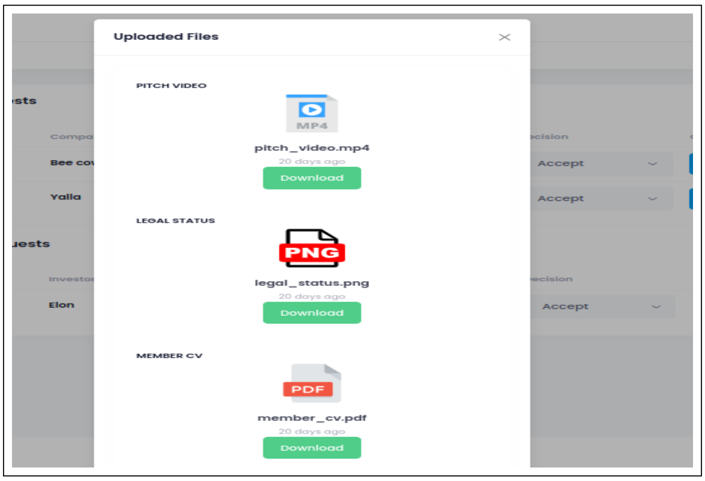
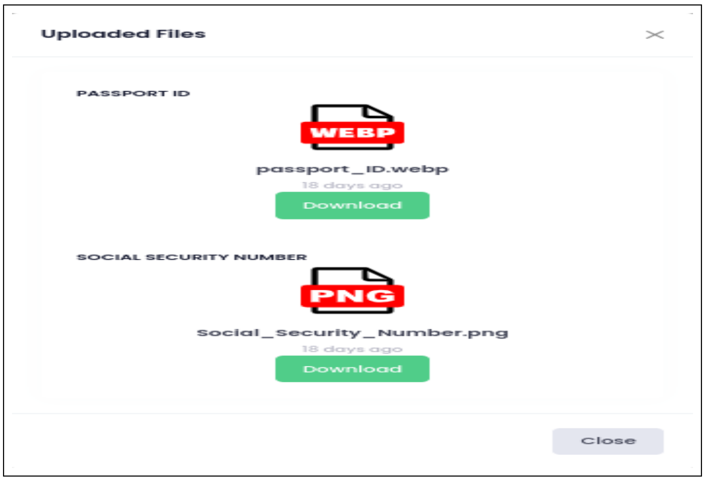


Figure 4.29: Startup Uploaded Files Review Interface

4.3.1.4.2 View Uploaded Files By The investor

Investor Uploaded Files Review Interface (figure 4.30) provides a visual representation of the interface where administrators can review files uploaded by investors within the system. In this screenshot, administrators have access to a detailed overview of the files submitted by investors as part of their KYC (Know Your Customer) requests. This functionality is essential for administrators as it allows them to verify the legitimacy of documents, assess compliance, and ensure the accuracy of information provided by investors.



Uploaded Files

PASSPORT ID

WEBP

passport_ID.webp

18 days ago

Download

SOCIAL SECURITY NUMBER

PNG

social_security_Number.png

18 days ago

Download

Close

Figure 4.30: Investor Uploaded Files Review Interface

4.3.1.5 User Status Management

User Status Management by administrators is a critical system function, especially in the context of KYC compliance. It allows administrators to categorize and update user statuses, typically as "In Process," "Accepted," or "Rejected." This feature ensures regulatory compliance, transparency in communicating user account statuses, and supports informed decision-making regarding user accounts. It streamlines user onboarding processes, promoting efficiency within the system.

Tableau 4.13: Textual Description Of User Status Management Use Case

Actor	Administrator
Goal	Empower administrators to manage user statuses and oversee interactions
Precondition	Administrator is logged into the admin dashboard
Post-condition	Administrators can effectively set and modify user statuses
Nominal Scenario 1. The administrator logs into the admin dashboard. 2. The dashboard provides options to set user statuses. 3. The administrator selects a user and sets their status. 4. A mechanism is in place to periodically reset user statuses.	
Alternative 5. If a user status is set or modified incorrectly, administrators can correct it.	

Textual description of user status management use case (tableau 4.13) , is a pivotal function designed to empower administrators within the system. The primary objective is to facilitate administrators in managing user status and overseeing interactions effectively. For this process to take place, administrators must first be logged into the admin dashboard, ensuring proper authentication.

In the nominal scenario, the process unfolds as follows: After logging in, the dashboard offers options for administrators to set and modify user statuses. Administrators can select specific users and adjust their statuses based on various criteria. Additionally, a mechanism is in place to periodically reset user statuses, ensuring that they remain accurate and up-to-date.

In an alternative scenario, if a user status is set or modified incorrectly, administrators have the capability to rectify it. This functionality ensures that user statuses are consistently accurate, fostering transparency and efficiency in user management within the system. Ultimately, this use case is instrumental in maintaining compliance, user transparency, and streamlined user interactions.

4.3.1.5.1 User Managment Interface

In the context of user management , the user management interface (figure 4.31) depicts that administrators possess the authority to manage user statuses, providing options such as "In Process," "Accept," or "Reject." Within this interface, administrators can perform these status adjustments while simultaneously reviewing user documents for verification and compliance purposes. This functionality empowers administrators to efficiently oversee the KYC (Know Your Customer) process, ensuring regulatory adherence and making well-informed decisions regarding user account statuses based on the evaluation of submitted documents.

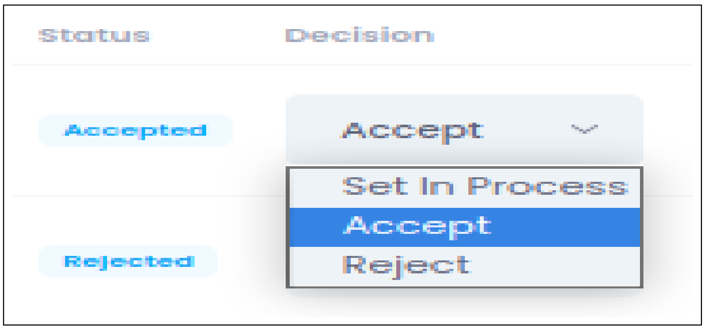


Figure 4.31: User Management Interface

4.4 Release 03

In this release , we're introducing a significant feature: the ability for token issuers to launch fundraising campaigns. After passing the KYC process, token issuers can define campaign details, such as campaign name, fundraising amount, share types, and more. Campaigns await administrator approval before investors can participate. This release enhances fundraising opportunities on our platform.

Our third release will consist of three sprints:

Token Issuer Campaign Setup: This sprint focus on enabling token issuers to create and submit campaign details.

Administrator Approval Workflow: This sprint concentrate on the workflow for administrators to review and approve/reject campaigns.

Investor Interaction and Finalization: This sprint focuses on enabling investors to view and participate in approved campaigns.

4.4.1 Sprint One: Token Issuer Campaign Setup

This sprint focuses on empowering token issuers to create campaigns efficiently. They can define campaign names, fundraising goals, share types, percentages, and the number of investors they wish to involve.

4.4.1.1 Sprint One Backlog

In this sprint ,the main focus is on enhancing the campaign creation process for token issuers. Firstly, token issuers will be able to create campaigns by specifying campaign names, fundraising goals, share types, and their associated percentages through a dedicated interface within the user dashboard. This feature will facilitate campaign management and is estimated to require 20 hours of development effort.

Furthermore, token issuers will have the capability to specify the number of investors they want to involve in their campaigns. This involves the development of backend logic to store this information for each campaign. Additionally, campaigns will go through an administrator approval process before fundraising can begin, necessitating backend logic to manage the pending status until approval is granted. This task is estimated to require 30 hours.

Tableau 4.14: Token Issuer Campaign Setup Backlog

ID	User Story	Task	Estimation
1.1	As a token issuer, I want to create a campaign with a campaign name, fundraising goal, share types, and percentages	Implement the campaign creation interface in the user dashboard	20h
1.2	As a token issuer, I want to specify the number of investors I want to involve in my campaign	Develop the backend logic to store the number of investors for each campaign	10h
1.3	As a token issuer, I want to go through administrator approval before start fundraising compaign	Set backend logic to handle pending status till administrator approval	30h
1.4	As a token issuer, I want to receive notification of compaign approval or rejection	Implement user notifications of compaign approval or rejection	10h
1.5	As a token issuer, I want validation checks to prevent the input of invalid data for campaign details	Design and implement validation checks for campaign details (e.g., goal amount, share percentages)	10h

At the end of this sprint, the objective is to improve the Token Issuer Campaign Setup process as shown in Tableau 4.14 by providing token issuers with robust tools for campaign creation, investor involvement, administrator approval, notification management.

4.4.1.1.1 Starting Fundraising Compaign Use Case

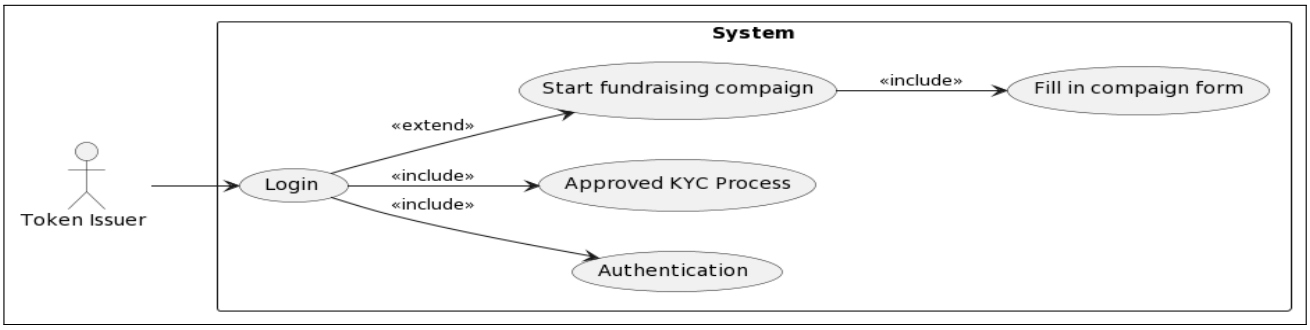


Figure 4.32: Token Issuer Campaign use case

The Token Issuer Campaign Use Case (figure 4.32) involves token issuers filling out a campaign form to create fundraising campaigns. Token issuers, once logged into their accounts, access a campaign creation interface where they provide essential campaign details, such as the campaign name, fundraising goal, share types, percentages, token name, and the desired number of investors. This process aims to streamline the creation of campaigns within the system, making it accessible and efficient for token issuers to initiate fundraising efforts and engage with potential investors.

Tableau 4.15: Textual Description of Token Campaign Creation Use Case

Actor	Token Issuer
Goal	Enable token issuers to create fundraising campaigns
Precondition	Token issuers are logged into their accounts
Post-condition	Token issuers successfully create campaigns
Nominal Scenario 1. Token issuers log into their accounts. 2. They are presented with a campaign creation interface. 3. Token issuers input campaign details, including name, fundraising goal, share types, percentages, token name and the number of investors. 4. Validation checks ensure that data entered is accurate. 5. Campaigns are created and stored in the system.	
Alternative 5. If validation checks fail, token issuers receive error messages.	

In the context of compaign creation (Tableau 4.15), primarily involving Token Issuers. Its aim is to enable token issuers to efficiently create fundraising campaigns. To initiate this process, token issuers must be logged into their accounts. Once logged in, they are presented with a campaign creation interface where they input campaign details, including name, fundraising goal, share types, percentages, token name, and the desired number of investors. Validation checks ensure data accuracy. Successfully validated data results in the creation and storage of campaigns within the system. In case of validation errors, token issuers receive error messages, ensuring the integrity of entered data.

4.4.1.1.2 Starting Fundraising Compaign Sequence Diagram

The Token Issuer Campaign Sequence (figure 4.33) begins with the issuer logging into their account, followed by authentication. After logging in, the issuer initiates a campaign creation process, leading to the deployment of a smart contract on the blockchain. If successful, the blockchain returns a unique contract address, and a confirmation notification is sent to the issuer. However, if any errors occur during deployment, an error message is displayed to the user. This sequence ensures a streamlined and secure process for token issuers to create and manage their campaigns on the blockchain.

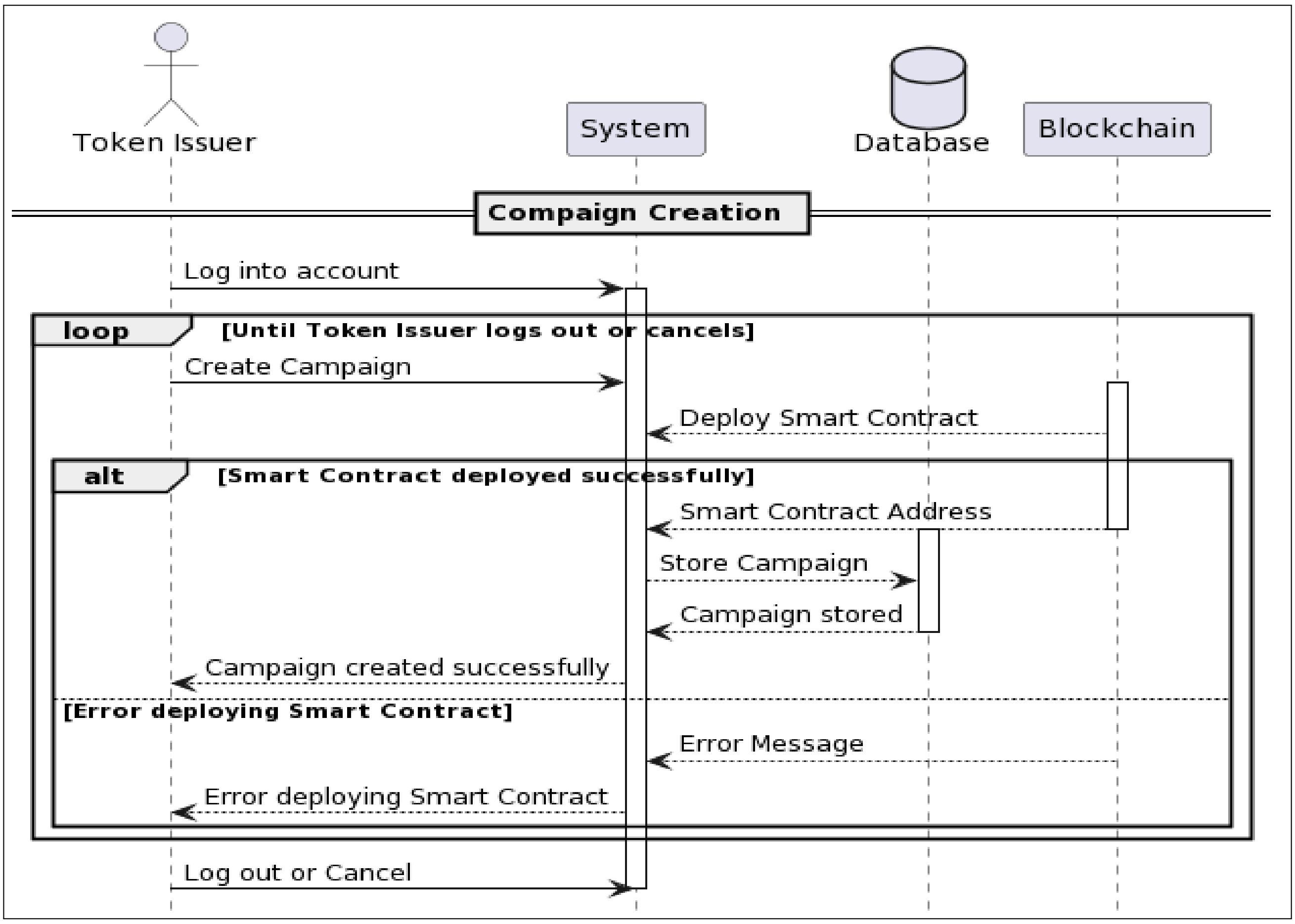


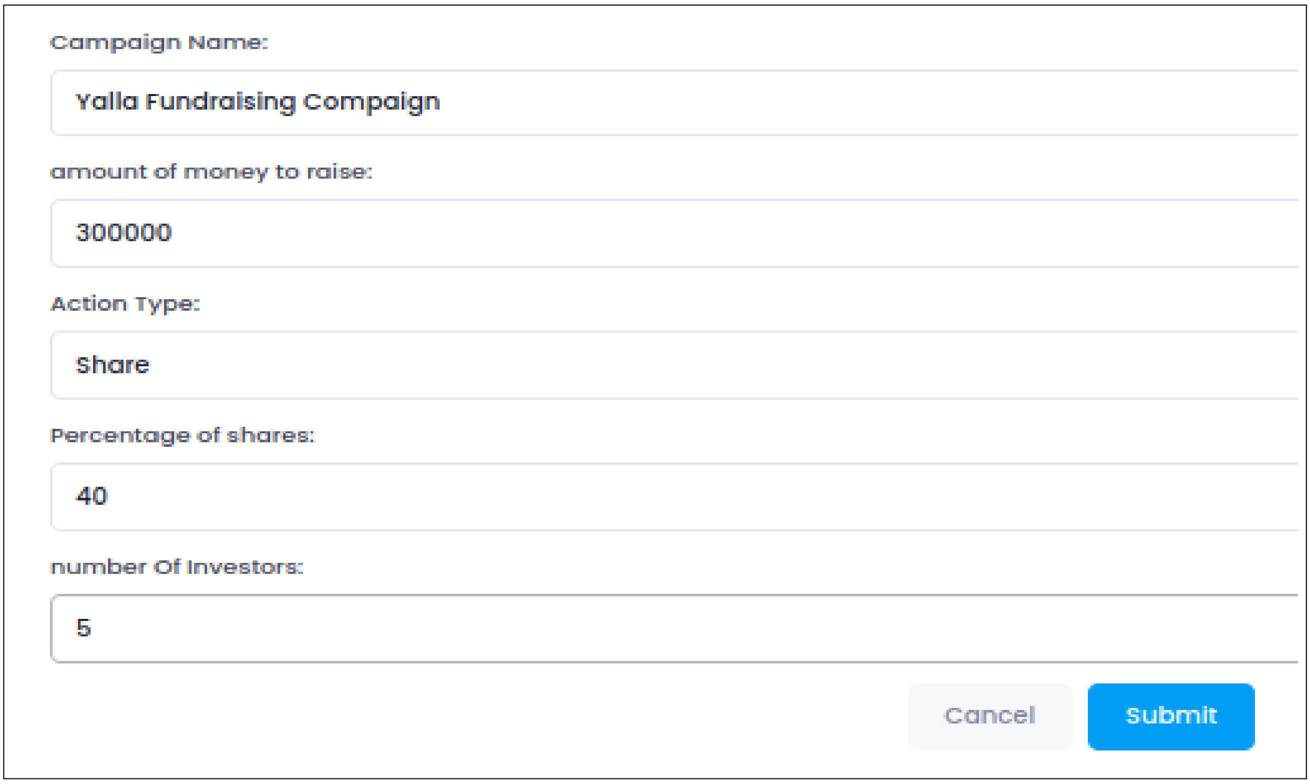
Figure 4.33: Token Issuer Campaign sequence diagram

4.4.1.1.3 Token Issuance Steps

The following steps represent the sequence of actions and requirements individuals must follow in order to issue tokens.

1. **Filling compaign details:**

   The initial step in the token issuance process involves the user filling out a campaign form (figure 4.34). In this form, the user provides essential details such as the campaign name, the desired fundraising amount, the type of action to be taken, the percentage of shares offered, and the number of investors the campaign is intended for. This step is crucial as it defines the parameters and goals of the fundraising campaign, setting the stage for the subsequent steps in the token issuance process.



Campaign Name:

Yalla Fundraising Compaign

amount of money to raise:

300000

Action Type:

Share

Percentage of shares:

40

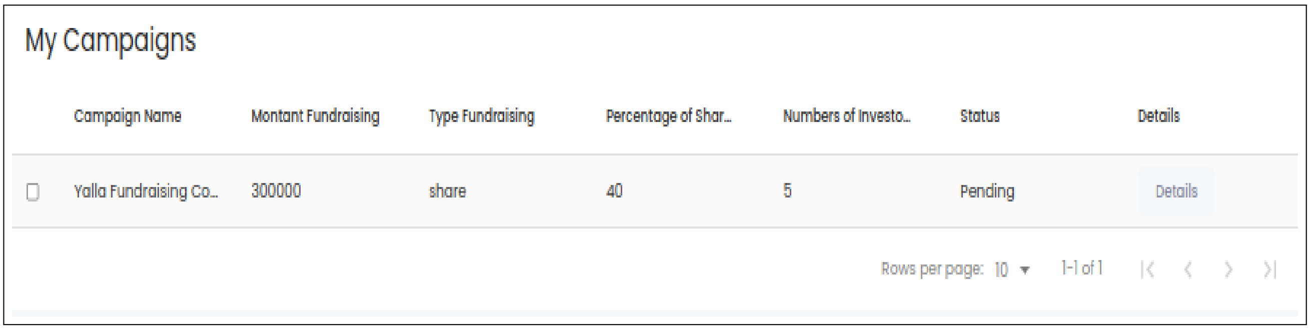
number Of Investors:

5

Figure 4.34: Interface of filled compaign form

2. **View his own running compaign:**

   The second step in the token issuance process allows the user, after successfully completing the token issuance, to access and view their own running campaign with all its details (figure 4.35). This step provides the user with transparency and visibility into the campaign's progress, including information about contributors, funds raised, and other relevant campaign details. It enables the user to monitor and manage their campaign effectively, ensuring that they have a clear overview of its performance throughout its duration.



My Campaigns

	Campaign Name	Montant Fundraising	Type Fundraising	Percentage of Shar...	Numbers of Investo...	Status	Details
□	Yalla Fundraising Co...	300000	share	40	5	Pending	Details

Figure 4.35: Interface of pending compaign

4.4.2 Sprint Two: Administrator Approval Workflow

In this sprint, the emphasis is on streamlining the workflow for administrators. They gain the capability to review and approve/reject campaigns. Additionally, we work on notifications and tracking for clear communication and auditing.

4.4.2.1 Sprint Two Backlog

In this sprint , the team is dedicated to enhancing the administrative aspects of the system. They're working on equipping administrators with a more efficient campaign approval process for token issuers. This sprint involves creating an admin dashboard, which will be the central hub for administrators to conveniently view all pending campaign requests as shown in Tableau 4.16 . Additionally, administrators will gain the ability to thoroughly review campaign details, including options to approve or reject campaigns based on predefined criteria. To keep token issuers informed, a notification system will be integrated, promptly updating them on the status of their campaigns. Furthermore, a campaign review log and tracking mechanism will be implemented to maintain a comprehensive record of all approval and rejection actions for auditing purposes, ensuring transparency and accountability throughout the process.

Tableau 4.16: Administrator Campaign Review Sprint Backlog

ID	User Story	Task	Estimation
3.1	As an admin, I want to view all pending campaign requests from token issuers	Develop admin dashboard to display pending campaign requests	30h
3.2	As an admin, I want to review campaign details and approve or reject them	Implement campaign review functionality, including approval and rejection options	40h
3.3	As an admin, I want to notify token issuers about the status of their campaigns	Integrate notification system to inform token issuers about campaign approval/rejection	20h
3.4	As an admin, I want to track and log the approval/rejection actions for auditing	Implement campaign review log and tracking mechanism	20h

4.4.2.1.1 Administrator Campaign Review Use Case

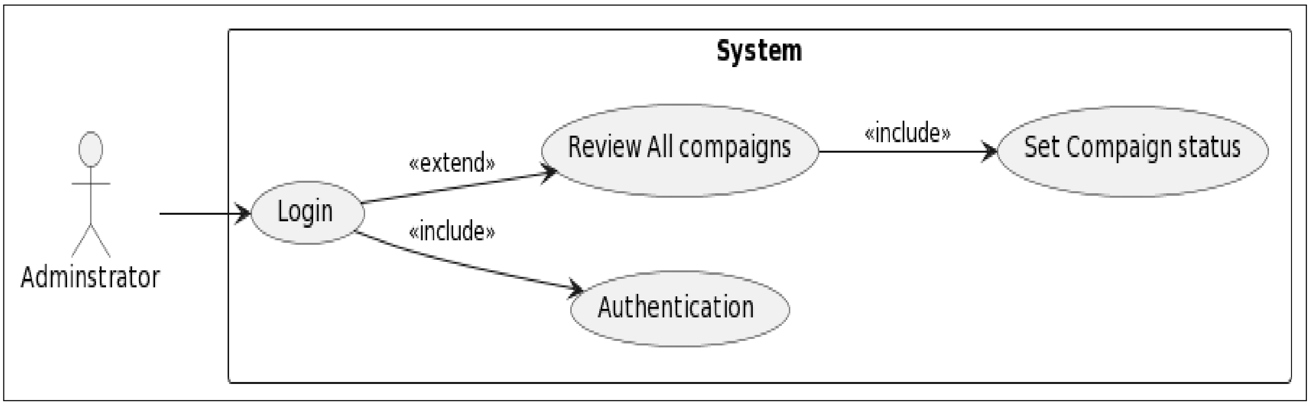


Figure 4.36: Administrator Approval use case

The Administrator Approval use case (figure 4.36) simplifies the user's role to a straightforward process. As an administrator, the primary tasks involve logging into the system and reviewing all campaigns that require approval. This includes examining campaign details and, based on established criteria, setting the campaign status to either "approved" or "rejected." This use case streamlines the administrative workflow, making it easy for administrators to make informed decisions and efficiently manage the approval status of campaigns submitted by token issuers.

Tableau 4.17: Textual Description Of Administrator Approval Workflow Use Case

Actor	Administrator
Goal	Streamline the campaign approval workflow for administrators
Precondition	Administrator is logged into the admin dashboard
Post-condition	Administrators efficiently review and manage campaign requests
Nominal Scenario 1. The administrator logs into the admin dashboard. 2. The dashboard presents a consolidated list of all pending campaign requests. 3. Administrators can apply filters, sorting, and search options to refine the view. 4. For each campaign request, administrators can review campaign details and choose to approve or reject it. 5. Administrators can make informed decisions based on the overview.	
Alternative 5. If no pending campaign requests are available, the dashboard displays a message.	

The Administrator's goal is to streamline campaign approval workflow. They log into the admin dashboard, where they view a list of pending campaign requests. Administrators can refine their view with filters and search options. For each request, they review campaign details and decide to approve or reject it based on predefined criteria. This process ensures efficient management of campaign requests. In an alternative scenario where no pending requests exist, the dashboard displays a notification to inform the administrator.

4.4.2.1.2 Administrator Approval Sequence Diagram

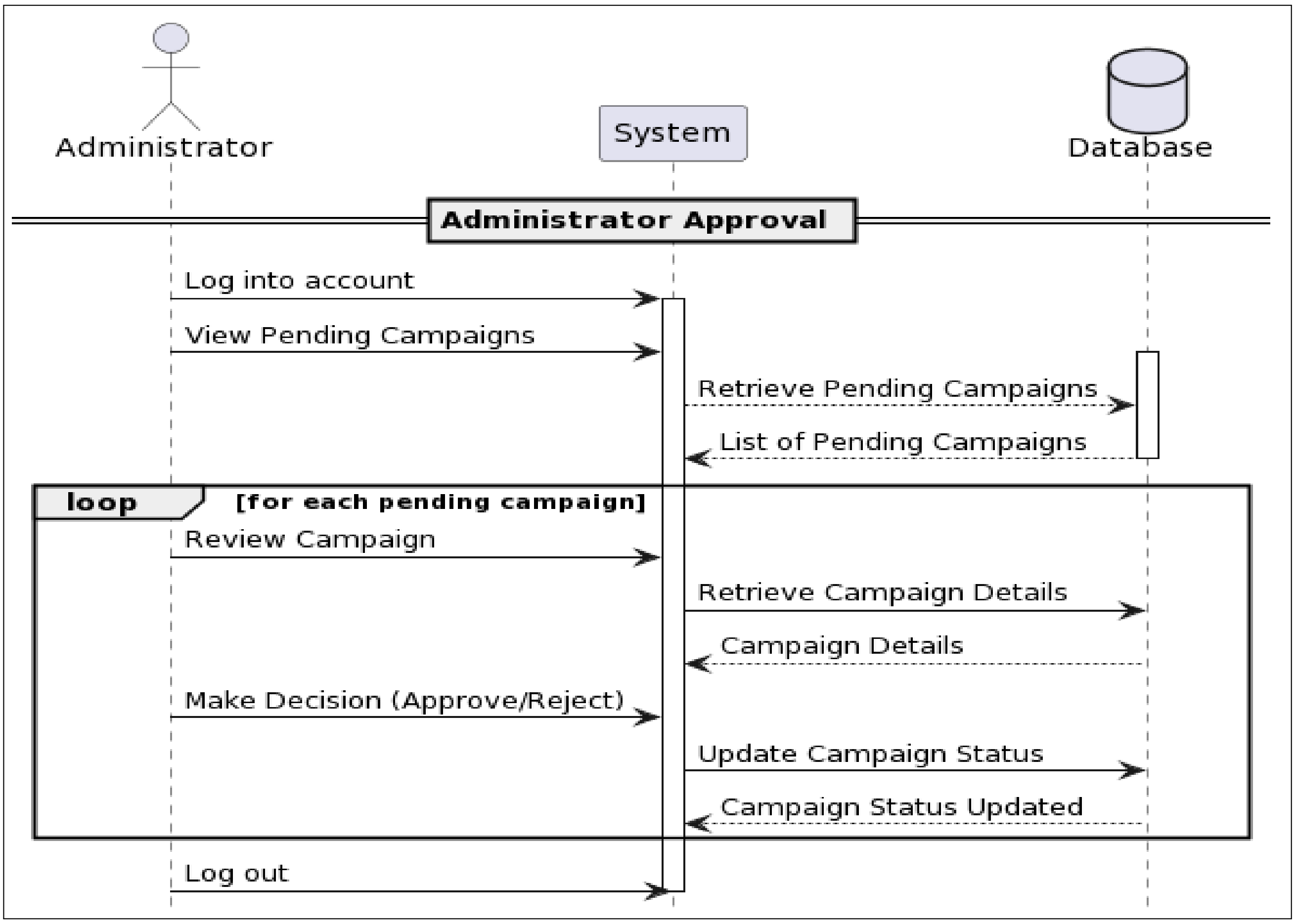


Figure 4.37: Adminstrator Approval sequence diagram

The administrator approval sequence diagram (figure 4.37) depicts the step-by-step process through which an administrator manages campaign approval. It begins with the administrator logging into the admin dashboard. The dashboard provides a consolidated list of pending campaign requests, which the administrator can refine using filters and search options. For each request, the administrator reviews campaign details and selects either approval or rejection based on predefined criteria. This sequence ensures an organized and efficient workflow for administrators to manage campaign requests effectively. In an alternative scenario where no pending requests exist, the dashboard displays a notification to inform the administrator of this status.

4.4.3 Sprint Three: Investor Interaction and Finalization

Sprint three is dedicated to enhancing the investor experience. Investors can now view and engage with approved campaigns. They gain access to campaign details and the ability to purchase tokens, supported by confirmation notifications.

4.4.3.1 Sprint Three Backlog

In this sprint , the primary objective is to enhance the investor experience within the system. This sprint includes tasks such as creating an investor dashboard to display approved campaigns, developing a campaign details view with token purchase options, implementing the ability for investors to purchase tokens within campaigns, and integrating a confirmation notification system for successful token purchases (Tableau 4.18). These improvements aim to provide investors with a seamless and informative platform for exploring, investing in campaigns, and receiving prompt confirmations for their token purchases.

Tableau 4.18: Investor Interaction and Finalization Sprint Backlog

ID	User Story	Task	Estimation
4.1	As an investor, I want to view all approved campaigns	Implement investor dashboard to display approved campaigns	30h
4.2	As an investor, I want to see campaign details and token purchase options	Develop the campaign details view with options for token purchase	40h
4.3	As an investor, I want to purchase tokens within campaigns	Implement the token purchase functionality	30h
4.4	As an investor, I want to receive confirmation of token purchase	Integrate a confirmation notification system for successful token purchases	20h

The investor campaign engagement use case is designed to enable investors to actively participate in approved campaigns. Investors begin by logging into their accounts, and upon doing so, they are greeted with a dashboard showcasing all available approved campaigns. Within this dashboard, investors have the flexibility to explore campaign details and access token purchase options. The system facilitates token purchases directly within the campaigns, making it convenient for investors to invest. Additionally, to ensure a smooth experience, investors receive confirmation notifications upon

Tableau 4.19: Textual Description Of Investor Campaign Engagement Use Case

Actor	Investor
Goal	Enable investors to view and participate in approved campaigns
Precondition	Investors are logged into their accounts and approved campaigns are available
Post-condition	Investors successfully engage with campaigns and purchase tokens
Nominal Scenario 1. Investors log into their accounts. 2. They are presented with a dashboard displaying approved campaigns. 3. Investors can view campaign details and token purchase options. 4. Token purchase functionality is available within campaigns. 5. Investors receive confirmation notifications for successful token purchases.	
Alternative 5. If no approved campaigns are available, investors receive a message on the dashboard.	

successfully purchasing tokens. In the alternative scenario where no approved campaigns are available, the dashboard communicates this status to investors through a notification message, keeping them informed. This use case streamlines the investor's journey, allowing them to engage effectively with campaigns and purchase tokens with ease.

4.4.3.1.1 Investor Engagement Use Case

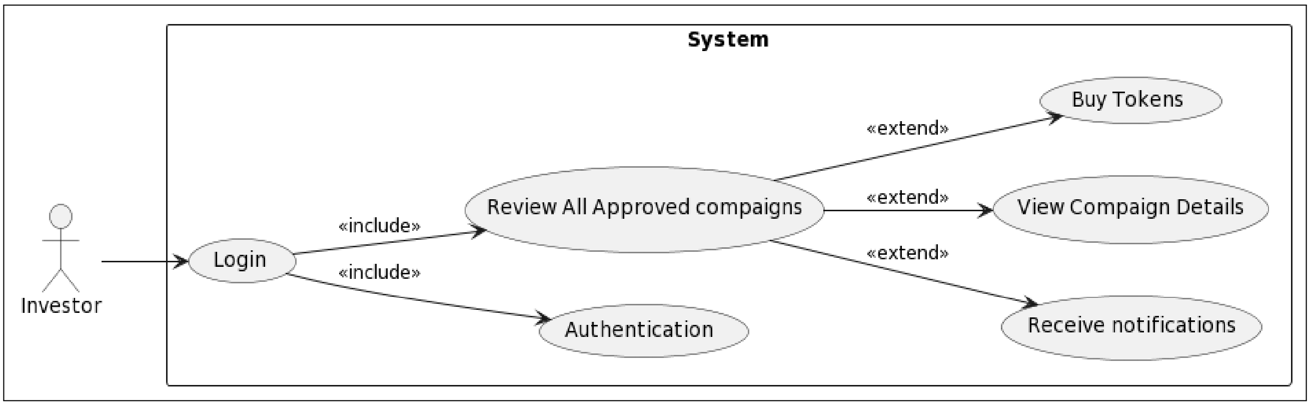


Figure 4.38: Investor Engagement use case

The investor engagement use case as shown in figure 4.38 allows users to engage as investors by reviewing all approved campaigns, exploring the details of each campaign, and purchasing tokens within these campaigns. This use case empowers investors to make informed decisions about their investments, ensuring a seamless and informative experience as they browse, evaluate, and participate in campaigns.

4.4.3.1.2 Investor Engagement Sequence Diagram

In the investor engagement sequence diagram (figure 4.39), investors begin by logging into their accounts and viewing all approved campaigns provided by the administrator. If an investor decides to buy tokens within a specific campaign, they initiate a token purchase request. The system then facilitates the transfer of token ownership on the blockchain, moving tokens from the issuer's address to the buyer's address securely. Following this successful transaction, a notification is automatically sent to the token issuer, informing them of the new investor's engagement in their fundraising campaign. This process ensures a seamless and transparent experience for both investors and issuers.

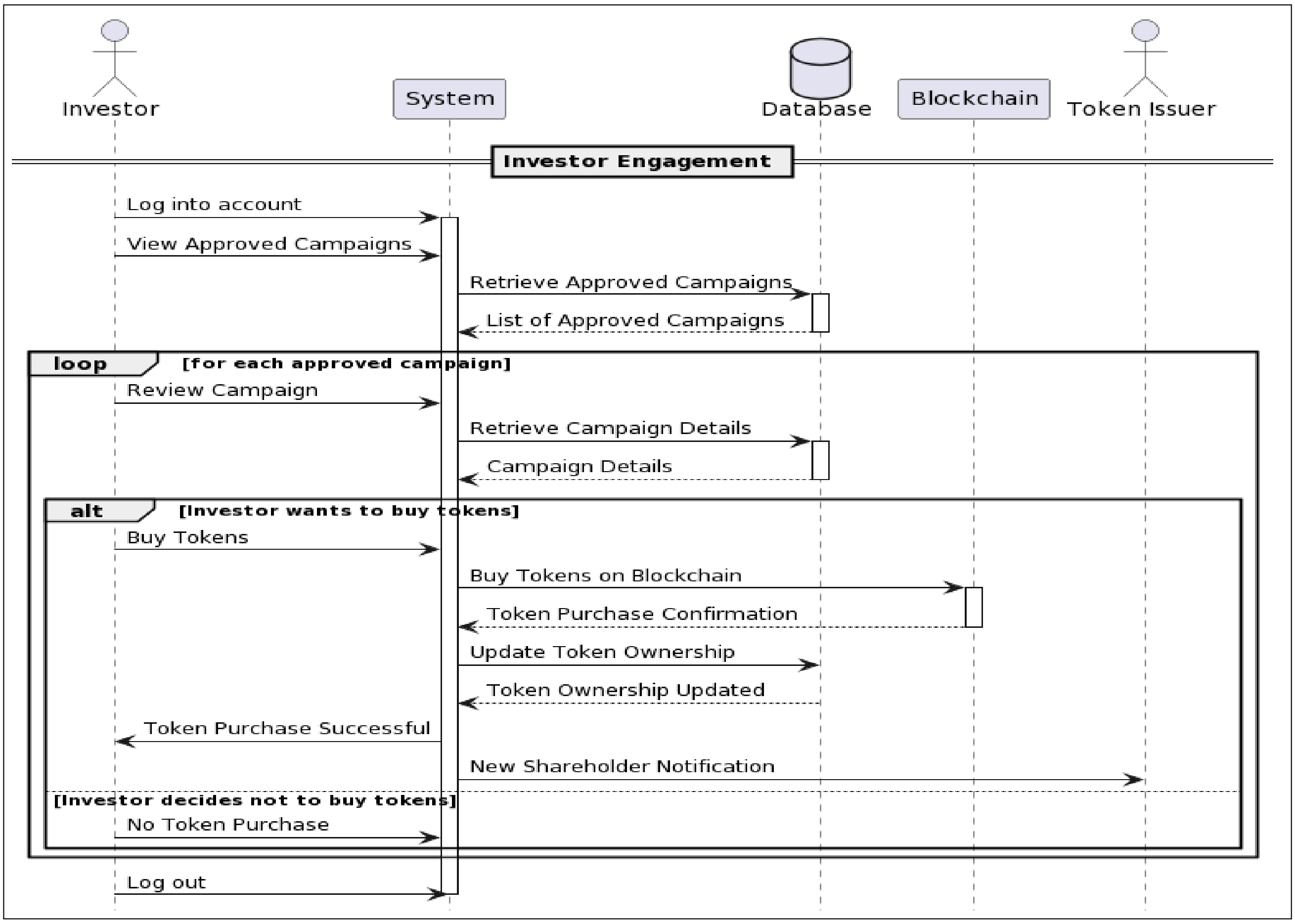
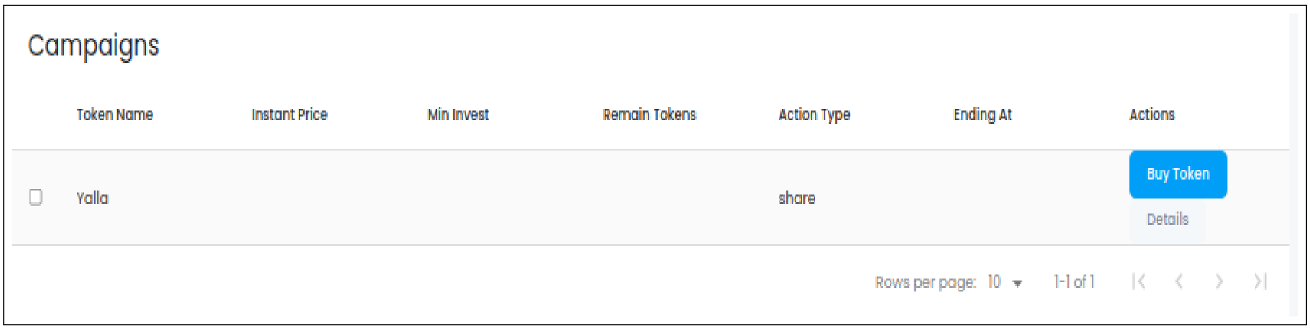

Investor
System
Database
Blockchain
Token Issuer
Investor Engagement
Log into account
View Approved Campaigns
Retrieve Approved Campaigns
List of Approved Campaigns
loop
[for each approved campaign]
Review Campaign
Retrieve Campaign Details
Campaign Details
alt
[Investor wants to buy tokens]
Buy Tokens
Buy Tokens on Blockchain
Token Purchase Confirmation
Update Token Ownership
Token Ownership Updated
Token Purchase Successful
New Shareholder Notification
[Investor decides not to buy tokens]
No Token Purchase
Log out

Figure 4.39: Investor engagement sequence diagram

4.4.3.1.3 Token Purchase Steps

The following steps represent the sequence of actions that the investor must follow in order to buy tokens.

1. **Initiate token purchase request :** The initial step in the token purchase process involves the user selecting the "Buy Token" option on an approved campaign (figure 4.40). This action serves as the trigger to initiate a token purchase request. By clicking this option, users express their intent to acquire tokens within a specific campaign, commencing the purchasing procedure and allowing them to further define their investment.



Campaigns

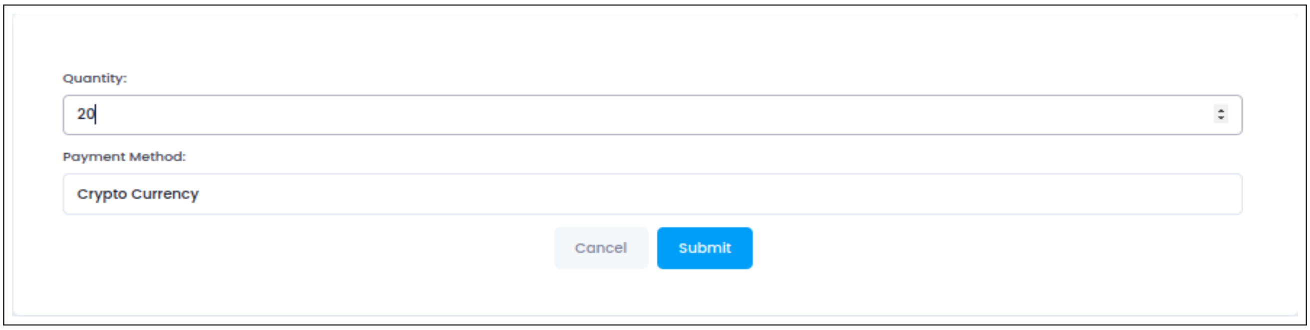
	Token Name	Instant Price	Min Invest	Remain Tokens	Action Type	Ending At	Actions
□	Yalla				share		Buy Token Details

Rows per page: 10 ▾ 1-1 of 1

Figure 4.40: Initiate token purchase request interface

2. **Token quantity selection and submission:**

   In the second step of the token purchase process, users are required to specify the quantity of tokens they wish to purchase and then proceed by pressing the "Submit" button (figure 4.41). This step is essential for users to define the quantity of tokens they want to acquire within the approved campaign, facilitating a seamless and user-driven token purchase experience.



Quantity: 20

Payment Method: Crypto Currency

Figure 4.41: Token quantity selection and submission interface

4.5 Conclusion

In conclusion, this chapter has provided a comprehensive overview of the implementation plan for our project. We have meticulously detailed the roadmap for each release, complete with sprints and the distinctive features that will be incorporated. By doing so, we have laid the foundation for a structured and well-organized approach to project execution.

General conclusion and perspectives

This report is the product of my internship at Infinity Management. The core focus of my internship was the development of a decentralized fundraising application built on Ethereum blockchain. This application revolutionizes the way startups can secure funding by enabling them to issue tokens while also providing a seamless investment platform for potential investors.

We were able to enhance our practical skills and apply the theoretical knowledge acquired during our training at Tunis Business School, providing us with valuable exposure to the professional world.

One of the primary objectives of this project was to empower startups with a powerful fundraising tool that fosters innovation and inclusivity. By allowing startups to issue tokens, we aimed to democratize investment opportunities and connect them with a diverse pool of investors from around the world.

This internship at Infinity Management allowed me to deepen my understanding of blockchain technology, smart contracts, and decentralized applications. Moreover, it provided me with valuable insights into the regulatory landscape surrounding blockchain-based fundraising initiatives.

In conclusion, my final graduation internship was an enriching experience that not only solidified my academic knowledge but also exposed me to the transformative power of blockchain technology in the world of business and finance. I'm excited about the possibilities this project opens up for startups and investors alike, and I'm eager to explore further opportunities in this dynamic field."

Bibliography

[1] MVC, [Access on 23-April-2023]. [Online]. Available: https://www.codecademy.com/article/mvc.

[2] Docker, [Access on 30-March-2023]. [Online]. Available: https://docs.docker.com/get-started/overview/.

[3] VScode, [Access on 02-March-2023]. [Online]. Available: https://code.visualstudio.com/docs.

[4] Geth, [Access on 02-March-2023]. [Online]. Available: https://geth.ethereum.org/.

[5] Solidity, [Access on 06-March-2023]. [Online]. Available: https://docs.soliditylang.org/en/v0.8.21/.

[6] React, [Access on 11-April-2023]. [Online]. Available: https://en.wikipedia.org/wiki/React_(software).

[7] Nodejs, [Access on 11-April-2023]. [Online]. Available: https://en.wikipedia.org/wiki/Node.js.

[8] MongoDB, [Access on 11-April-2023]. [Online]. Available: https://en.wikipedia.org/wiki/MongoDB.

[9] Traefik, [Access on 11-April-2023]. [Online]. Available: https://www.kubecost.com/kubernetes-devops-tools/traefik-vs-nginx/.

[10] MailJet, [Access on 11-April-2023]. [Online]. Available: https://www.mailgun.com/email-marketing/mailjet/.

[11] Github, [Access on 11-April-2023]. [Online]. Available: https://en.wikipedia.org/wiki/GitHub.

[12] Truffle, [Access on 11-April-2023]. [Online]. Available: https://github.com/trufflesuite/truffle.

[13] Gitbook, [Access on 11-April-2023]. [Online]. Available: https://arctype.com/blog/gitbook-technical-documentation-guide/.

[14] Dapp, [Access on 23-April-2023]. [Online]. Available: https://www.preethikasireddy.com/post/the-architecture-of-a-web-3-0-application.

[15] Blockchain, [Access on 23-April-2023]. [Online]. Available: https://en.wikipedia.org/wiki/Blockchain.