VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI - 590 018

A PROJECT REPORT

on

"CAMPUS CRYPTO - AN ELECTRONIC CASH SYSTEM IN CAMPUS"

Submitted by

Jasmi Y K	4SF18CS052
K Samarth N Kamath	4SF18CS055
Niraj R Shetty	4SF18CS095

In partial fulfillment of the requirements for the award of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE & ENGINEERING

Under the Guidance of

Dr. Priya R Kamath

Associate Professor, Department of CSE

at

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575 007

2021 - 22

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575 007

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project entitled **"Campus Crypto - An Electronic Cash System in Campus"** has been carried out by **Jasmi Y K (4SF18CS052), K Samarth N Kamath (4SF18CS055) and Niraj R Shetty (4SF18CS095)**, the bonafide students of Sahyadri College of Engineering & Management in partial fulfillment for the award of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belagavi during the year 2021 - 22. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Signature of the Guide	**Signature of the HOD**	**Signature of the Principal**
Dr. Priya R Kamath	Dr. Pushpalatha K	Dr. Rajesha S

External Viva:

Examiner's Name	Signature with Date
1.	.....
2.	.....

SAHYADRI

College of Engineering & Management

Adyar, Mangaluru - 575 007

Department of Computer Science & Engineering

DECLARATION

We hereby declare that the entire work embodied in this Project Report titled **"Campus Crypto - An Electronic Cash System in Campus"** has been carried out by us at Sahyadri College of Engineering and Management, Mangaluru under the supervision of **Dr.Priya R Kamath,** for the award of **Bachelor of Engineering** in **Computer Science & Engineering**. This report has not been submitted to this or any other University for the award of any other degree.

Jasmi Y K (4SF18CS052)

K Samarth N Kamath (4SF18CS055)

Niraj R Shetty (4SF18CS095)

Dept. of CSE, SCEM, Mangaluru

Abstract

A crypto currency is a digitally encrypted data string that represents a unit of money that is protected by cryptography and is thus practically difficult to forge or double spend. Blockchain technology, a distributed ledger enforced by a varied network of computers, is the foundation of several cryptocurrency networks, which are decentralized. To solve the issues with conventional currencies is the basic goal of cryptocurrencies. The planned Campus Crypto project - Tokens that are prone to loss can be replaced on campus with an electronic cash system, a peer-to-peer cryptocurrency. This web-based programme makes use of the blockchain idea; the issuer authority is delegated to collegiate bodies, and the administrator creates the money. Only the college's grounds are worth anything with this cyber money. The project's goal is to set up a fully working E-wallet on campus that ensures the highest level of authenticity and dependability. Additionally, to provide software that enables simple transactions and quick maintenance.

Acknowledgement

It is with great satisfaction and euphoria that we are submitting the Project Report on **"Campus Crypto - An Electronic Cash System in Campus"**. We have completed it as a part of the curriculum of Visvesvaraya Technological University, Belagavi for the award of Bachelor of Engineering in Computer Science & Engineering.

We are profoundly indebted to our guide, **Dr. Priya R Kamath**, Designation, Department of Computer Science & Engineering for innumerable acts of timely advice, encouragement and We sincerely express our gratitude.

We also thank **Mr. Duddela Sai Prashanth**, **Mr. Suhas A Bhyratae** and **Mr. Shailesh Shetty S**, Project Coordinators, Department of Computer Science & Engineering for their constant encouragement and support extended throughout.

We express our sincere gratitude to **Dr. Pushpalatha K**, Head & Professor, Department of Computer Science & Engineering for his invaluable support and guidance.

We sincerely thank **Dr. Rajesha S**, Principal, Sahyadri College of Engineering & Management, **Dr. S. Manjappa**, Director Strategic and Planning, Sahyadri College of Engineering & Management, and **Dr. D. L. Prabhakara**, Director, Sahyadri Educational Institutions,who have always been a great source of inspiration.

Finally, yet importantly, We express our heartfelt thanks to our family & friends for their wishes and encouragement throughout the work.

Jasmi Y K (4SF18CS052)

K Samarth N Kamath (4SF18CS055)

Niraj R Shetty (4SF18CS095)

Table of Contents

Abstract **i**

Acknowledgement **ii**

Table of Contents **iv**

List of Figures **vi**

List of Tables **vii**

1 Introduction **1**

1.1 Purpose 3

1.2 Scope 3

1.3 Overview 3

2 Literature Survey **4**

3 Problem Definition **9**

4 Software Requirements Specification **10**

4.1 Introduction 10

4.2 Purpose 10

4.3 User Characteristics 10

4.3.1 Students 10

4.3.2 College Body 11

4.4 Interfaces 11

4.4.1 Hardware Interfaces 11

4.4.2 Software Interfaces 11

4.5 Functional Requirements 12

4.5.1 Constraints 12

4.6 Non-Functional Requirements 13

5 System Design **15**

5.1 Architecture Design 15

5.2 Decomposition Description 16

5.3 Data Flow Design 18

5.4 Sequence Diagram 19

5.5 Use Case Diagram 20

5.6 Class Diagram 21

5.7 Activity Diagram 22

6 Implementation **24**

7 System Testing **31**

8 Results and Disscussion **33**

9 Conclusion and Future work **39**

List of Figures

5.1 The System Architecture Diagram 15
5.2 Flowchart of the proposed system 17
5.3 Data Flow Design 18
5.4 Sequence Diagram for Proposed System 19
5.5 Use Case Diagram for the proposed system 20
5.6 Class Diagram for Campus Crypto System 21
5.7 Activity Diagram for Campus Crypto System 22

6.1 Folder Structure of the Proposed System 24
6.2 Structure of User Class 25
6.3 Routing method for User Registration 25
6.4 Routing method for User Login 26
6.5 Adding Genesis Block 26
6.6 Generation of Keys 27
6.7 Calculating Hash 27
6.8 Routing method for New Transaction 28
6.9 Method for adding a transaction 28
6.10 Routing method for Mining a block 29
6.11 Method to Mine the Pending Transactions 29

8.1 The Login page 33
8.2 The Home Page 33
8.3 The Profile Page 34
8.4 The Transaction Page 34
8.5 The Successful Transaction 35
8.6 The Invalid User Transaction 35
8.7 The Mine Block Page 36
8.8 After Mining 36
8.9 Blockchain Overview 37
8.10 Block Details Page 37

8.11 The Receiver Page 38

List of Tables

4.1 Software Interface 12
4.2 Functional Requirements 13

7.1 Work Flow 32
7.2 Test cases from User Perspective 32

Chapter 1

Introduction

A blockchain is a digital format for storing transactions that are distributed in a network of connected computers. Data in a blockchain is stored in a block. Whenever a new transaction occurs, it is inserted into a ledger with its hash which is an immutable cryptographic signature. A hash value with its previous hash value constitutes a block that gets added to the chain. This hash value provides security to the transaction that occurred. Blockchain is known as distributed database since the data is managed by nodes in the blockchain. Blockchain provides a mechanism to store the data in a decentralized manner. Since the blockchain is transparent in nature it gains trust among the users. Once the data is entered into the blockchain it is highly impossible to edit the data. Thus it brings a notion of fraud resistance among its users. If a malicious node tries to manipulate the data in the blockchain all the data in the blockchain has to be changed to get a uniform hash value among its all nodes. This showcases the importance of a decentralized system and thus fails the attempt to alter the data.

The nodes in the blockchain agree to be transparent about the transactions that occur in the chain. All the transactions of the blockchain are verifiable and are reflected in real-time. Additionally, a chain is unanimous in nature. Due to all these reasons, data in a blockchain is secure, anonymous, and programmable.

A public blockchain is a kind of blockchain network where nodes do not require permission to join the chain and can participate in the tasks in the chain. This system is completely decentralized. This prohibits the blockchain from being controlled by a single node or a set of nodes. Since there is no controlling authority all nodes equally share the responsibility to secure the blockchain. This makes blockchain secure as it is very difficult

to modify the data once accepted by all the nodes. In contrast to the public blockchain, a private blockchain makes use of an authorized blockchain to handle the request to add particular data to the blockchain. This restricts the permission and participation of other nodes in the blockchain. It adds security by prohibiting third parties from entering the blockchain.

In a blockchain where nodes are competing to add their hash value to the next block, it is indeed very useful to adapt a mechanism called consensus mechanism to achieve data consistency. It also acts like a fault-tolerant mechanism. Every blockchain uses a consensus mechanism that fits their need. There are multiple consensus mechanisms available. Few of them are proof of stake, proof of work, proof of authority, practical byzantine fault-tolerant, and so on. The selection of the consensus mechanism depends on the use case of blockchain.

By using blockchain, the user would gain more control over their data and information. Also, it would reduce the costs associated with using the platforms. It is feasible to have each user protect their data in a decentralized system. Instead of relying on trust, an electronic payment system that allows any two willing parties to interact with each other directly without the use of a reliable third party is required. Seller protection from fraud would come from transactions that are computationally hard to undo, while buyer protection would come from regular escrow methods that are simple to implement. In this research, we offer a method to generate computational verification of the transactional order using a distributed peer-to-peer timestamp server to solve the double-spending problem. As long as the number of trustworthy nodes outnumbers any group of cooperative attacker nodes, the system is safe.

The project deals with creating a peer-to-peer electronic payment system that can be utilized inside the campus. This is a blockchain-based online platform where the administrator creates the currency and college organizations are given the right to issue new tokens. This virtual currency is only usable inside the college's boundaries. The safest method to carry money is with cryptocurrencies since it can be used anywhere and at any time and avoids theft. The transactions are protected by blockchain technology because they cannot be changed.

1.1 Purpose

The purpose of this project is to build peer-to-peer Crypto currency system that can be used in Campus in the place of tokens which are likely to be lost. This proposed project is a web-based application which uses concept of blockchain, where the crypto coins can be transacted between the peers and also between the college bodies and the students. This will also avail the students to explore the world of cryptocurrency via Blockchain technology.

1.2 Scope

1. **Loyalty Program:** The proposed system can be used to distribute and manage the crypto coins transacted by the peers in the network. A complicated system that stores multiple users' accounts can be implemented using the proposed method.

2. **Currency Exchange:** The system can swap a variety of reward coins as well as currencies. The business that allows peers to make transactions via crypto coins can implement this system.

1.3 Overview

The physical tokens given from the college bodies is likely to be lost and also lead to token duplication and accumulation of the same. Using online currency such as crypto coins which doesn't involve any third party fits well in the given situation. This will create a peer to peer network where the peers can transact coins effortlessly.

Chapter 2

Literature Survey

The literature survey helps in understanding the existing research done on blockchain technology. The information gathered here will help identify the gaps in the current work.

Satushi Nakamoto *et al.* [1] suggested that Online payments might be transmitted directly from one party to another without going through a banking institution with a peer-to-peer electronic currency system. Digital signatures contribute to the solution in certain ways, but the primary advantages are lost if a reliable third party is still needed to avoid duplicate spending. This suggests utilising a peer-to-peer network to resolve the double-spending issue. The network timestamps transactions by hashing them into a continuing chain of hash-based proof-of-work, creating a record that cannot be modified without repeating the proof-of-work. Using a peer-to-peer distributed timestamp server to create computational evidence of the chronological sequence of transactions, we suggest a solution to the double-spending problem in this study. As long as there are honest nodes, the system is safe.

Mohammad Azman *et al.*[2] proposed that any licensed Local Broker can be used to facilitate a transaction in the Blockchain and register it in the distributed ledger. Hot-Cold Hybrid Decentralized Exchange presents a method to locally store cryptocurrency wallet data in personal devices and process transactions between two personal devices without needing any common database or centralised server system. Additionally, it has been suggested that the system may be implemented as a smart card and could be made to be only slightly thicker than the commonplace cards used today. It is built on the combination of a secure two-way authentication system that permits reliable handshaking techniques for lightweight nodes acting as local facilitators and E-wallets.

There are several blockchains available and each has different use cases. The interoperability between blockchain is the issue solved by Ajay Kumar *et al.*[3]. The suggested method offers visuals to aid in spotting some trends in the application of technology provided by the bitcoin blockchain. In this article, utilising WebGL technology, author visualises the locations from where peers in the bitcoin blockchain network were conducting their transactions in order to monitor the bitcoin transaction continent-by-continent. Author examined regional Bitcoin blockchain usage trends by tracking the development of respective clusters over time.

Monika di Angelo *et al.*[4] proposed that many users use a software wallet to handle their cryptocurrencies or cryptographic tokens, which makes it easier to connect with a blockchain in general or with on-chain applications (smart contracts) in particular. Some blockchain wallets implement fundamental functionality as smart contracts on-chain, however many blockchain wallets run their main programme code off-chain. This is done with the goal of increasing security and trust through transparent and auditable execution. The usefulness of blockchain technology that uses cryptographically secure smart contracts for wallets is investigated in this paper. To this purpose, author outline techniques for locating wallet contracts by examining source code, bytecode, and execution traces taken from transaction data.

Umair Khan *et al.*[5] proposed that as an enterprise type, the shared economy concept is recommended to become one of the corporations. A number of mutual economy models have emerged in response to the requirement for sharing distinct revenue, particularly with the expanded development of digital smart gadgets and the internet. Utilizing shareable goods and digital material is also a goal. There are a number of risks that might occur during transactions while using digital material in the sharing economy, including the possibility of content theft, modification, and hacking. The security and privacy of Blockchain are presented in-depth in this study. When paired with smart contracts, blockchain technology promises transparent, unbreakable, and secure platforms that can support innovative solutions.

C. E. Veni Madhavana *et al.*[6] introduced three concepts in the paper: (i)metaphysical iterative process for blocks of financial instrument transaction chains, including a proposed form of denominational digital cash (ii)a compositional layout for blocks of content transaction chains and (iii)a new, malleable cryptographic hash function to support all chaining operations. The transactional and financial information are linked together in the overall

design. With the aid of proof-of-work/proof-of-stake computational concepts, the digital currency and blockchains of our system may be converted into decentralised blockchain models analogous to cryptocurrencies. The article recently presented a novel key stream generator that is based on certain sets of integer arithmetic progressions. The plan offers greater adaptability and diversity with regard to parametric options. Author uses the integer sequences obtained from this collection to create the hash chains.

Saeed AlzhaRani *et al.*[7] introduced a idea to provide a money that is not supported, created, or connected to any government. Blockchain technology serves as the financial platform for cryptocurrencies. The rate of acceptance of cryptocurrencies has surged, and the industry has expanded significantly. By examining the current level of cryptocurrency adoption, adoption-influencing variables, offering an in-depth analysis of these elements, and outlining some potential dangers, the appear aims to close the gap in the current level literature. Even if it is challenging to determine the precise number of cryptocurrency users, a reliable approximation may be obtained by looking at the number of bitcoin exchange websites.

Xiao Fan Liu *et al.*[8] proposed that since practically all cryptocurrency users' behaviours are reliably documented in transactions on public blockchains, the cryptocurrency economy offers a thorough digital record of human economic behaviour. Blockchain addresses, which serve as user IDs in the transaction logs, are anonymous. This study analyses Ethereum token transactions, defines the behaviour of important economic agents based on transaction patterns, and investigates how to identify economic agents using interpretable machine learning models. Six categories of the most active economic agents-namely, centralised cryptocurrency exchanges, decentralised exchanges, cryptocurrency wallets, token issuers, airdrop services, and gaming services are specifically taken into consideration. The findings demonstrated that bitcoin exchanges and online wallets exhibit distinctive behavioural tendencies, making them distinct from other agents

Saurabh Suratkar *et al.*[9] presented that a bitcoin wallet stores Blockchain private and public keys but not the real cash values. The paper focuses on multi-currency wallets review exploring on features like supported currencies, anonymity, cost, platform support, key management, wallet recovery methods and fiat currencies supported. Customers may use wallets to interface with blockchains to send and receive virtual currency tokens and adjust their balance. The three subcategories of multicurrency wallets are software, hardware, and paper. There are desktop, mobile, and online software wallets. One must have a thorough

understanding of wallets given the growing use of blockchain in several businesses. Many hash functions have been developed to improve the security of data blocks.

There are several hash functions invented to increase the security of the data blocks. Nagendra Singh Yadav *et al.*[10] introduced that Since we have entered the era of digital financing, where we can see various security flaws particular to payment gateways, where hackers steal money from credit or debit cards by redirecting the OTP to themselves, E-wallets have evolved into a conventional means of banking. Everything begins with a minimal sum, but the impact grows as the amount per transaction rises with each attempt. The effectiveness of the banking system in handling transaction-related frauds by assuring authenticity through the adoption of a system powered by blockchain is discussed in this research study.

The transaction settlement between entities takes a long time with the intervention of a third-party entity.XiaoJian HE *et al.*[11] proposes that the keys are the primary means of accessing digital assets in modern cryptocurrencies. Secure and stable key management is crucial. In order to store the keys safely and reliably in a decentralised network, the article presents a unique cryptocurrency wallet management technique. As a data distribution technique, DMCD can ensure the security and stability of essential storage and recovery due to its high data dispersion and better balance between the rate of storage space usage and contribution. All of the technologies suggested can guarantee that our system operates well in a decentralised state. The proposed system is effective, reliable, and safe in the decentralised network, which is proved by the tests and assessments.

Weiqi Dai *et al.*[12] proposed that the security of an encrypted account is becoming more and more crucial as the overall amount of digital money rises. The software-based wallet is handy, but the safety cannot be guaranteed. The hardware-based wallet is secure, but it is cumbersome since users must carry a second physical device. All of these wallets require blockchain synchronisation, however the majority of mobile devices today lack the storage capacity for all blocks. In this work, author build an SPV-protecting Secure Blockchain Lightweight Wallet based on Trustzone. It is safer than a software wallet and more portable than a hardware wallet. Regardless of whether Rich OS is malevolent or not, isolation can prevent attackers from stealing the private key and the wallet address.

Salah Albeshr *et al.*[13] proposed that in the case of blockchain technology, digital data is kept in a distributed public database. This technology became well-known mostly as

a result of the launch of the original cryptocurrency, Bitcoin. This technology is used by bitcoin to provide secure record-keeping. This article will first provide a broad overview of blockchain and demonstrate its operation. Blockchain has the potential to drastically transform financial services because to its high levels of security, transaction transparency, decentralisation, and efficiency. The relationship between blockchain technology, financial technologies, and sustainability will also be covered. Finally, it will be examined how blockchain technology will change the financial sector and how difficult it will be to embrace and execute.

Abdul Ghaffar Khan *et al.*[14] proposed a implemention on a online wallet that prioritise privacy and security, and digital internet transactions involving currency or coins also require a certain amount of protection, both before and during streaming of the transaction. Using an Android application that uses QR codes and secure private key management, author designed and built a cryptocurrency wallet for the Android operating system in this research study. The hot wallet is used to transmit bitcoin to the network, while the cold wallet is used to generate and retain private key addresses for safe transaction confirmation. It was made secure by employing cross-QR code scanning of the hot and cold wallets for identity verification, verification, and authorization.

Sun Liyan *et al.*[15] proposed that trading on the blockchain can only be done by signing transactions using the private key that is kept in the wallet. If the wallet's private key is lost or exposed, it will result in irreversible damage. Currently, the popular wallet management strategies either store the wallet in a single location or depends on a certified center's involvement, which cannot withstand single point failure. This work presents a blockchain wallet protection system against link failure based on incremental elliptic curve digital signature without certified centre in order to safeguard the security of blockchain accounts. Without the help of a certified centre, players in this system work together to produce public and private keys and share private keys. Users who exceed the required number can confirm transactions repeatedly, thereby thwarting single point attacks and ensuring wallet security.

Chapter 3

Problem Definition

To implement a highly scalable and secure peer-to-peer crypto currency online wallet that can be used in campus in the place of physical tokens which are more likely to be lost and also to be duplicated. This proposed project is a web-based application which uses concept of blockchain, where the peers in the network can make transaction effortlessly.

Chapter 4

Software Requirements Specification

4.1 Introduction

Software Requirement Specification totally defines how the projected software behaves without unfolding how the software will perform it. The elementary objective of the requirement stage is to yield the software requirement specification that designates the peripheral performance of the projected software. Software requirement can be well-defined as a condition of a capability required by a user to solve a problem or attain an objective.

4.2 Purpose

The purpose of this project is to create permissioned digital tokens that can be used inside campus for easy transfer of money among operating bodies and/or peers using University Seat Number (USN). The proposed system will help students to understand and implement Blockchain based cryptocurrency that is stored in a decentralized ledger. It authenticates user using a centralized database controlled by the organization.

4.3 User Characteristics

The Access to the users are provided in two levels.

4.3.1 Students

The Students are the users of the system. The students can make a transaction to any of the peers in the network. The students can check their balance in the profile. They even

have access to the decentralized ledger. The Students can make transaction to the college bodies as well.

4.3.2 College Body

The College bodies include canteen, stationary etc. The College Body has a special privilege to register the students. They can even transact coins to any of the peers in the network. The balance can be seen in their profile details. Even the College bodies have access to the decentralized ledger.

4.4 Interfaces

The system will be implemented in Python, Flask with the necessary libraries and modules. Flask will be used as an interfacing platform to illustrate the transaction between the students and the college body.

4.4.1 Hardware Interfaces

Since neither the mobile web application nor the web application has any designated hardware, it does not have any direct hardware interfaces. The hardware connection is between the nodes in the blockchain is wireless which is managed by the network itself. Each node will have dedicated hardware for processing

- Processor : Any processor above 500 MHz
- RAM : 1GB
- Hard Disk : 5GB
- Input Device : Standard keyboard and Mouse
- Output Device : Monitor

.

4.4.2 Software Interfaces

The software interfaces of the proposed system are as shown in the Table 4.1.

Table 4.1: Software Interface

Software Interface	
Operating System	Windows 7 and above, Linux
Programming Language	Python, Flask Framework, py-crypto module
IDE	VS Code

4.5 Functional Requirements

4.5.1 Constraints

The internet connection is a constraint for the application. Since the application fetches data from over the internet, whenever new nodes join into the network or any transaction is done, there must be an internet connection for the application to function. Setting can be changed by the organization according to the requirement but they will not be able to change the transaction details which are stored in the node in the network.

With the number of transactions increasing day by day, the blockchain becomes heavy. All transactions have to be stored for validating the transaction. Because of the initial block size restriction and the time interval utilised to generate a new block, it is not possible to execute millions of transactions in real time. As a result of the extremely limited block capacity, many tiny transactions may experience delays since miners favour those with a large transaction fee. Scalability is a challenging issue because a large block size will slow down the speed of propagation and result in blockchain branches. Whenever a new node wants to join the network the node contains transaction details that need to be validated by the other nodes present in the network based on the transaction history. It consumes lots of time and because of which transaction becomes slow. Here for validating the data lots of resources are consumed, so maintenance cost is high. According to the framework, it is assumed it will always be used on the dedicated platform stated above that has enough performance.

The functional requirements of the proposed system are as shown in the Table 4.2

Table 4.2: Functional Requirements

FR-#	**Description**
FR-1	The system shall provide a user interface to give the input.
FR-2	The system shall provide organization and students to create an account.
FR-3	The system shall provide the organization to have a unique token.
FR-4	The system shall provide organization to issue coin to the students.
FR-5	The system shall provide the organization to create and manage, delete the account.
FR-6	The system shall provide a wallet for each account.
FR-7	The system will provide an option to the student to buy a product by using real money in addition to campus crypto.
FR-8	The system doesn't allow students to exchange the cryptocurrency of one organization with another.
FR-9	The system shall provide an option to the student to link his wallet with the existing environment of the organization.

4.6 Non-Functional Requirements

There are several attributes of software that can serve as requirements. Required attributes must be specified so that their achievement can be objectively verified. The following items provide a partial list of examples.

- **Security:**

  Blockchain is designed with some built-in functionalities like cryptography, hashing and digital signatures. The Cryptography technique ensures data integrity and message authentication. Hashing can be used for data privacy. Also, centralized authentication is added to help authenticate users by organization.

- **Availability:**

  There may be some situations like whenever a system fails or a system is being accessed by the unauthorized entity or any subsystem fails then the data should be consistent without any effect on it. This system will ensure data consistency by replicating the data in the different parts of the system.

- **Portability:**

  Since the product is developed in python and flask framework, it can be portable with any operating system. The usage of the system will make the requirement of the user simpler to achieve 20% of component host-dependent, 5% of the code host-dependent, the system can be operated in any operating system.

- **Maintainability:**

  Well-organized data helps in searching users' information at a faster rate. The transaction can be easily traced and the fake transaction can be detected. The system makes transactions tamper-proof.

- **Reliability:**

  The consensus mechanism used in the system will take some amount of time which is more than transaction time in a centralized system, so to enhance the transaction speed reliability of the system must be cared for. This might be fixed by using various consensus techniques, such as Proof of Work. Memory management issues could be solved via sharding.

Chapter 5

System Design

5.1 Architecture Design

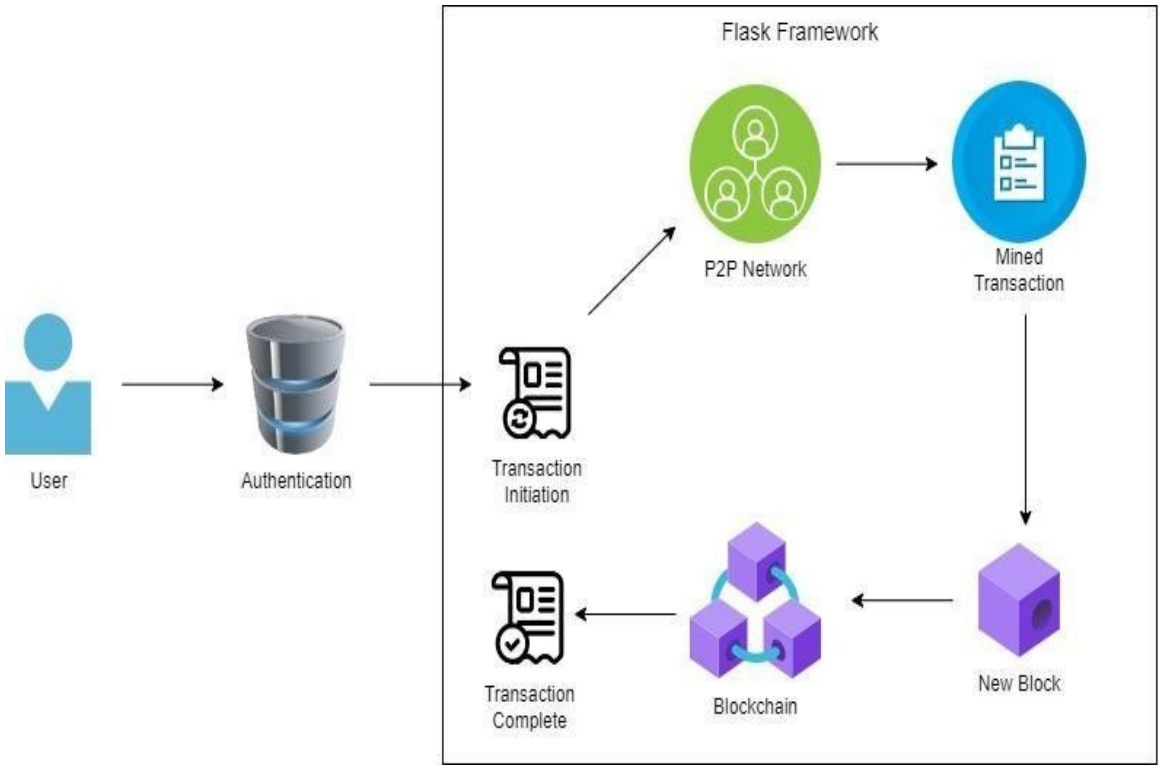

Figure 5.1: The System Architecture Diagram

The architecture diagram of the system is shown in Figure 5.1. Initially the peer logs in to the application and is checked for authorization. If the user is authorized, he/she will be redirected to the Home page where there are various options such as view profile, transact, mine block, blockchain viewer. For example, suppose User A wants to send 10 SCEM coins to User B, User A will initially log in to the application, clicks on the Make a Transaction option, where he/she will fill the username of User B and number of coins to be sent to User B and then hits the submit button. But the transaction is not completed

yet. As in blockchain, each transaction is verified by all the peers in that network. For this User A must click on Mine Block option and now the transaction is completed. Also the blockchain view page is also updated where a new block of this transaction will be added. Since it is decentralized, all the peers in that network will be able to view all the transaction details. The blockchain concepts were used to develop Smart Start. The proposed system employed the Flask-Framework. Using Flask framework, the peer to peer transaction is implemented that ensures the blockchain concepts.

The proposed project is about sending or receiving cryptocurrency. The college body will act as an Admin, who will be given with 10000 coins by default in the beginning. And the student acts as the remaining peer. Consider a scenario in the college canteen where, a student wants to buy a mini meal worth Rs.35 and he pays Rs.40 hard cash to the cashier. Now the cashier instead of providing the student with a token of Rs.5, can easily transact 5 coins to that student using his/her username. The students in order to get them involved in this, have to register their account in the college for availing their credentials. Once the student receives the credentials, he/she can transact the coins to any of the peers. The transaction will be done based on the username of the peers. So if a peer say User A wants to send coins to peer User B, then User A will use User B's username to send coins. So this means that the receiver's public key is automatically fetched through his/her username.

5.2 Decomposition Description

The system will be having multiple components. The system can be divided into meaningful sub-systems such as components. The components of the system will be student and the college body. These components will be combined together to form a network of peer-to-peer exchange system.

Figure 5.2 depicts the flowchart for the proposed system. Initially the user registers to the application with the user credentials. Login to the student's account occurs only if the student account is authenticated and confirmed by the college body. If the login is successful, coins/tokens can be transacted to the receiver and each time the transaction is done, it is validated through mining of block. After the transaction is completed, the balance will be updated in both the accounts along with the timestamp.

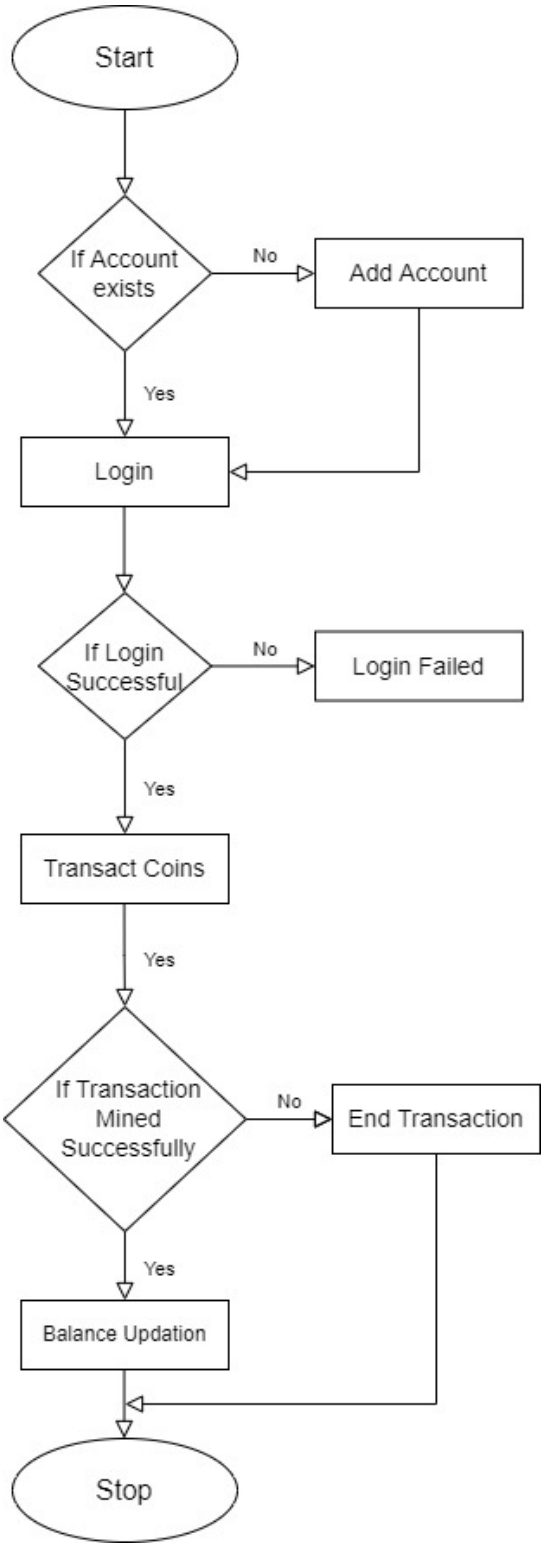

Figure 5.2: Flowchart of the proposed system

5.3 Data Flow Design

A data flow diagram (DFD) shows how data "flows" through an information system graphically. DFDs may also be used for data processing visualisation. On a DFD, an internal process transfers data items from an external data source or internal data store to an external data sink or internal data store. A DFD does not include details regarding the scheduling or sequencing of processes, or whether they will run sequentially or concurrently. As a result, it differs significantly from a flowchart, which depicts the control flow through an algorithm and lets users choose which actions will be taken, when and how, but not what kinds of data will be input into and output from the system, where they will go, or how they will be kept.

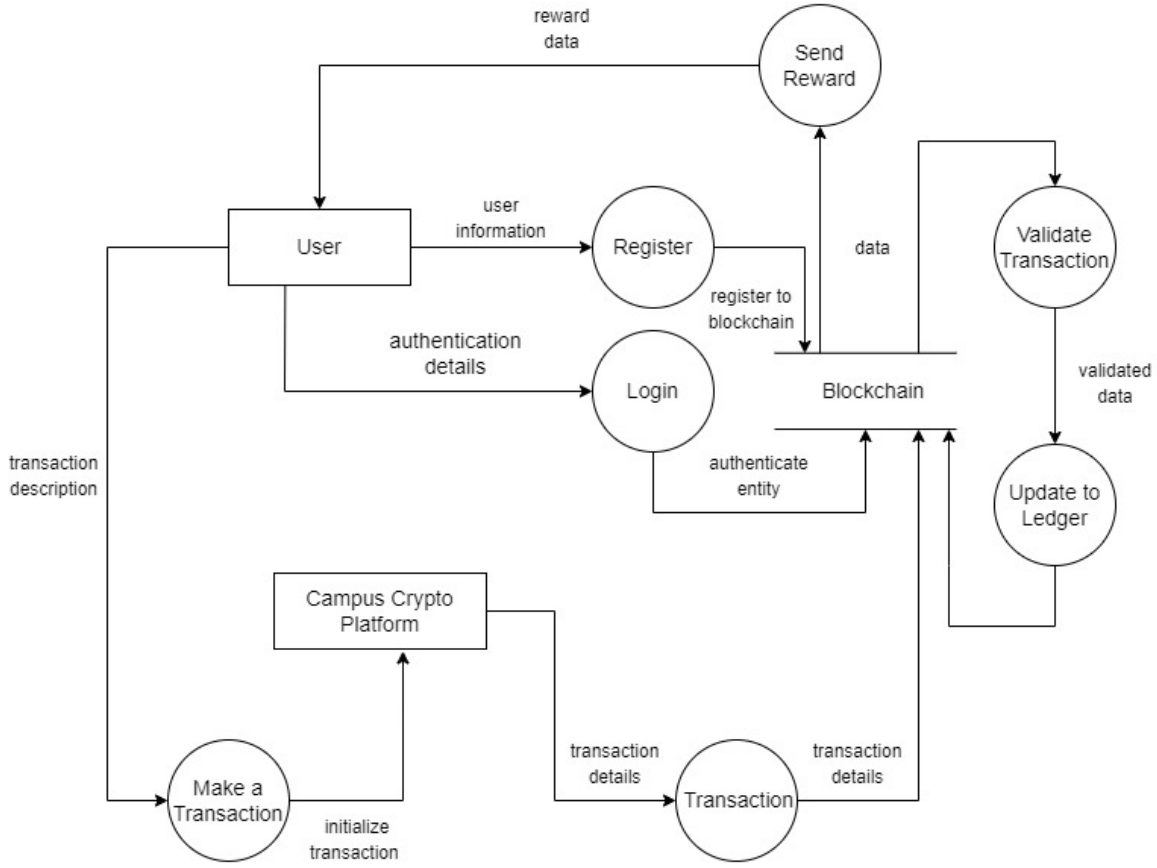

Figure 5.3: Data Flow Design

The Data Flow diagram of the proposed system is shown in Figure 5.3. Initially, the user will register to the system by using the User Credentials. If the user is already registered then he/she can login to the system. The user after signing in can visit the Campus Crypto site. If the user wishes to transact the coins, it is possible to transact coins from the user to the college body or vice-versa with high reliabilty and authentication. After the

transaction is performed, validation is performed and the transaction details is updated one by one to the ledger which maintains high security.

5.4 Sequence Diagram

A type of interaction diagram used in the Unified Modeling Language (UML), a sequence diagram demonstrates how and when processes interact with one another. It is a part of the message sequence chart construct. A sequence diagram displays several processes or things that exist at the same time as parallel vertical lines (sometimes referred to as "lifelines") and horizontal arrows. The messages exchanged between them, in the order in which they occur.

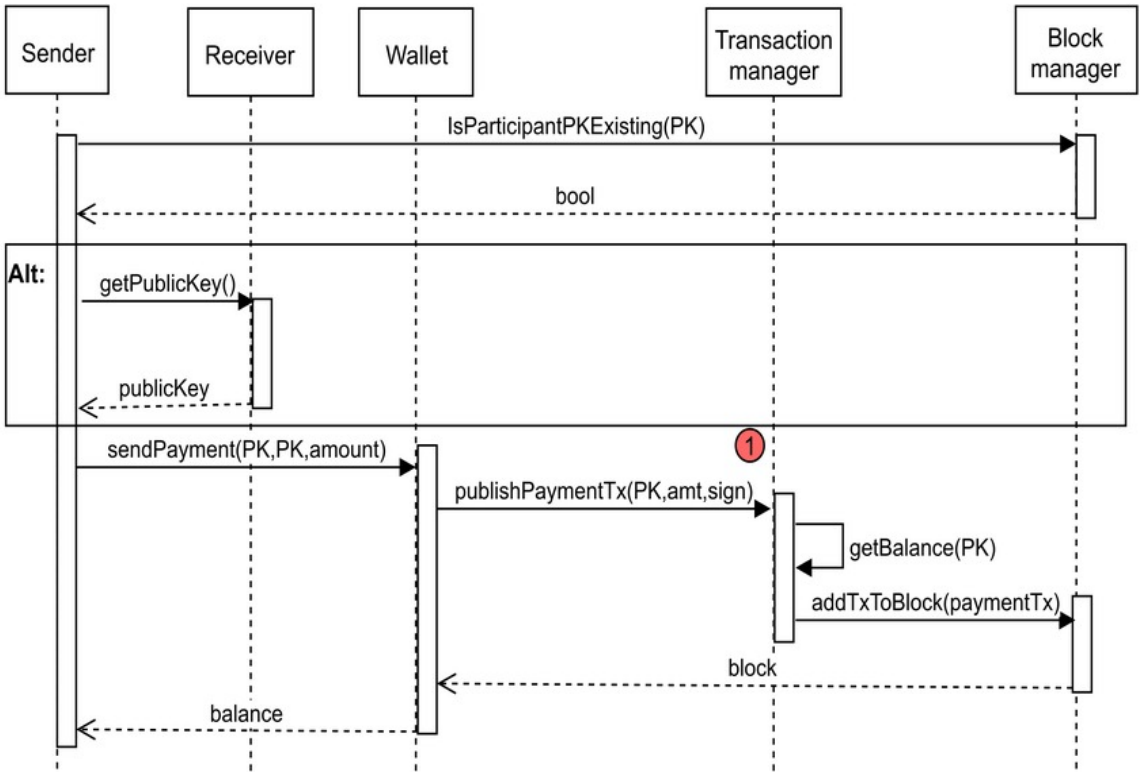

Figure 5.4: Sequence Diagram for Proposed System

Figure 5.4 shows the sequence diagram for the proposed system. Method IsParticipantPKExisting(PK) is used to check if the sender's primary key is present in the block manager or not initially. It returns true or false value based on key is present or not. Method getPublicKey() is used by the sender to request the public key from the receiver. Hence, publicKey is returned back to sender by the receiver. Method sendPayment(PK,PK,amount) is used by the sender encrypts the number of coins, by using the primary key of the sender. Method publishPaymentTx(PK,amt,sign) is used to sign the

transaction; where the transaction manager updates the balance to the receiver and also updates the balance of the sender. Finally, Method addTxToBlock(paymenttx) is used to add new block to the Blockchain overview. Block is returned to the wallet. Balance is updated on both sender and receiver's wallet.

5.5 Use Case Diagram

Use Case diagram shows the interaction between user and the system. Use Case diagram captures goals of the user and responsibility of system to its users. Use Case diagram shows various functions that can be performed by various actors.

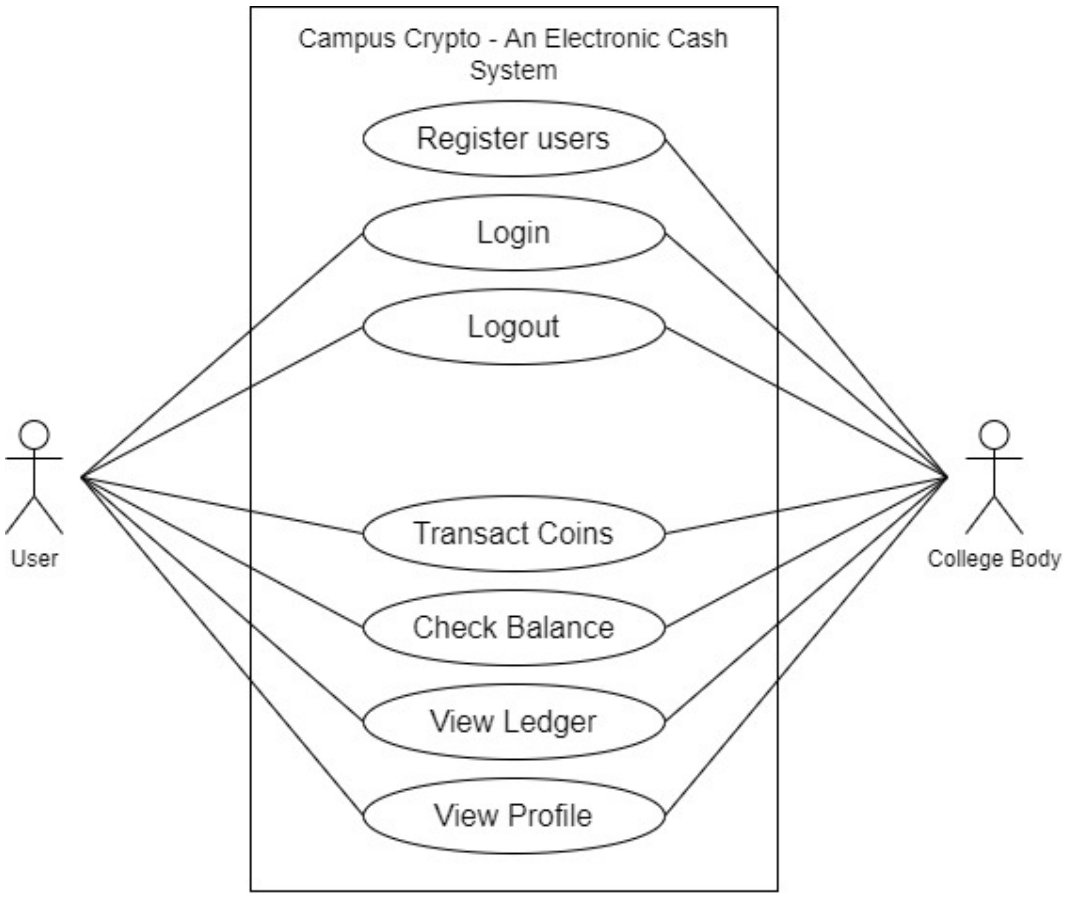

Figure 5.5: Use Case Diagram for the proposed system

Figure 5.5 depicts the user privileges. User can register in order to login to the application. A user can make a transaction to any other peer in the network. Each user will be having his/her own profile and thus can view it. The proposed system involves interaction between student and college body. The balance can be checked in the user's account. College body has the privilege to add users into the peer-to-peer network. Transaction Ledger will be updated with each transaction which leads to high reliability and security.

5.6 Class Diagram

Class diagrams offer a visual language for representing classes and their relationships by outlining potential objects. Class diagrams are useful both for abstract modelling and for design for designing actual programs. They are clear, simple to understand and practical.

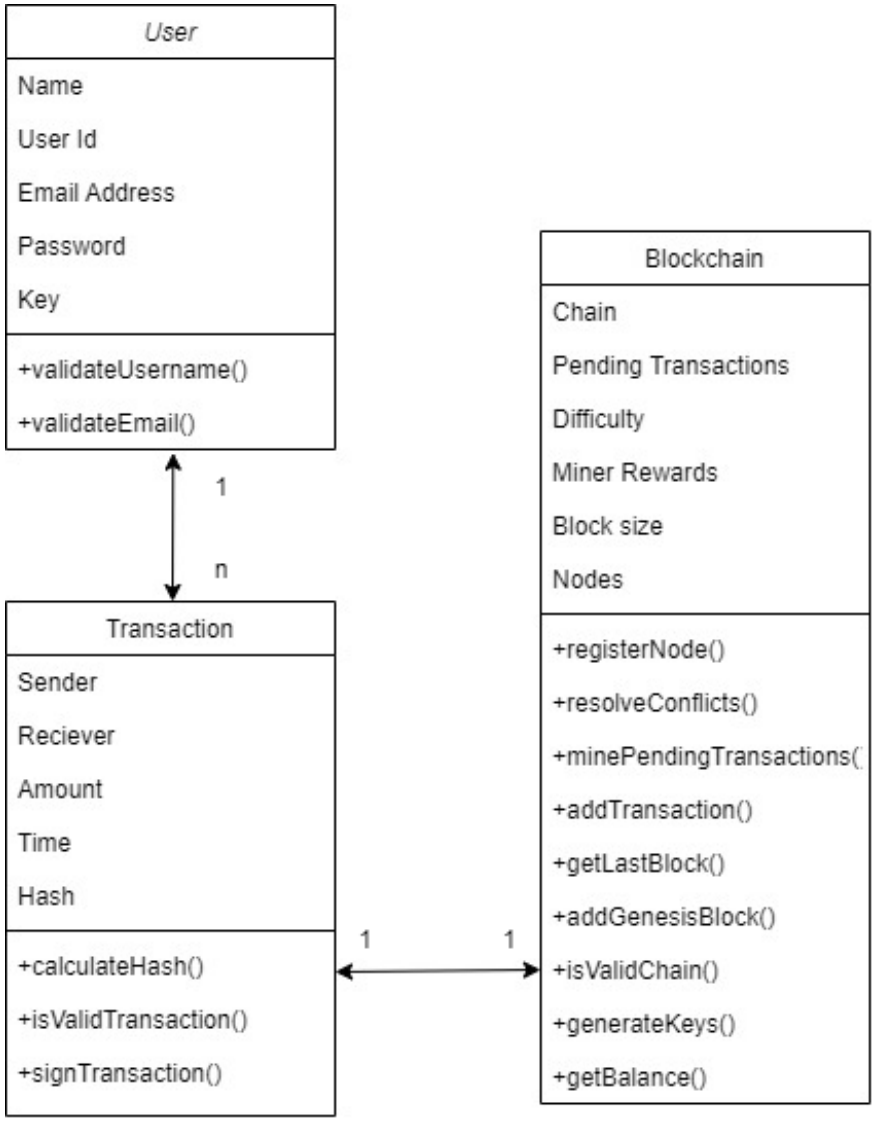

Figure 5.6: Class Diagram for Campus Crypto System

Figure 5.6 shows the class diagram of the proposed system. The system consists of three main classes: User, Transaction and Blockchain. The User class consists of all the basic user details such as Name, User id, Email Address, Password and Key. It has two methods to validate Username and Email. The Transaction class has the details related to each transaction such as the sender details, receiver details, the number of coins to be sent, the hash and the timestamp. It has methods to calculate hash, validate the transaction and also to sign a transaction. The Blockchain class has details related to chain, pending

transactions, Miner rewards and Nodes. It also has methods to register the node, resolve the conflicts, mining the pending transactions, validating the chain, generating the keys, adding the transactions, getting the balance key and adding the genesis block.

5.7 Activity Diagram

An activity diagram depicts the processes that make up a complicated process, such as an algorithm or work flow. Similar to a sequence diagram, an activity diagram depicts the flow of control but places more of an emphasis on processes than on objects. When building algorithms and work processes, activity diagrams are most helpful in the first phases.

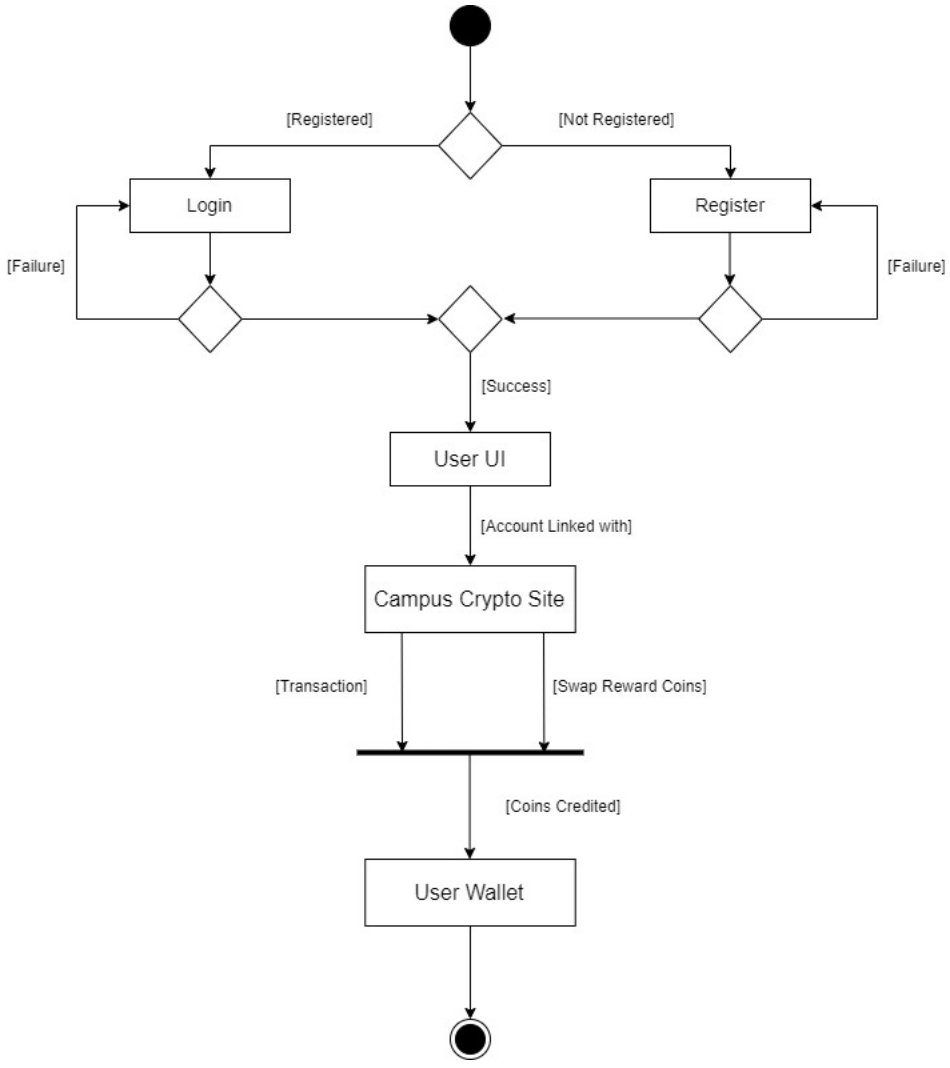

Figure 5.7: Activity Diagram for Campus Crypto System

Figure 5.7 shows the activity diagram for the Campus Crypto System. At the starting point of the system, the user will be having the option to login or register. Based on his state of registration the options are given. After successful login, it directly takes to the home screen where the blockchain overview are shown. Once the user logs into the system, he/she can transact coins with the peers in the network. And also the user node on successful mining of other blocks receives rewards. Finally the wallet is updated every time a transaction is made.

Chapter 6

Implementation

This section helps in understanding the implementation of the "Campus Crytpo - An Electronic Cash System in Campus". This gives us an overall idea of the different modules present in the system.

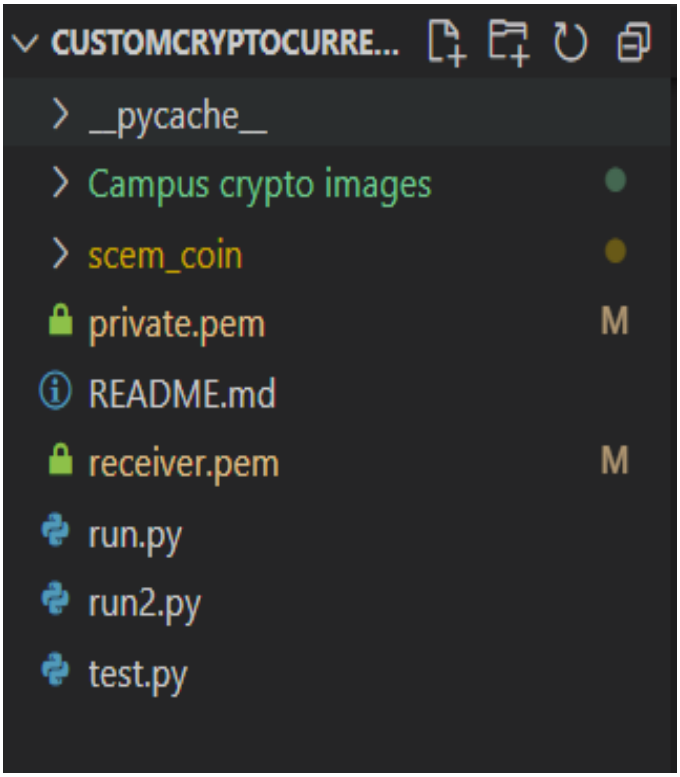

Figure 6.1: Folder Structure of the Proposed System

Figure 6.1 depicts the file structure of the project. The file run.py contains the run-time configuration of the application. The file private.pem and receiver.pem contains the certificate containers to store the private and public key respectively.

```
from scem_coin import db, loginManager
from datetime import time, datetime
from flask_login import UserMixin


@loginManager.user_loader
def loadUser(user_id):
    return User.query.get(int(user_id));

class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key = True)
    name = db.Column(db.String(35), unique = False, nullable=False);
    username = db.Column(db.String(15), unique = True, nullable=False);
    email = db.Column(db.String(100), unique = True, nullable=False);
    password = db.Column(db.String(120), unique = False, nullable=False);
    key = db.Column(db.String(100000), unique = True, nullable=False);

    def __repr__(self):
        return f"User('{self.name}', '{self.username}', '{self.email}')";
```

Figure 6.2: Structure of User Class

Figure 6.2 shows the user class definition which contains id, name, username, email, password and key as the data members.

```
@app.route("/register", methods=['GET', 'POST'])
def register():
    form = RegistrationForm()
    if form.validate_on_submit():
        #password hashing
        hashed_password = bcrypt.generate_password_hash(form.password.data).decode('utf-8');
        keyGen = blockchainObj.generateKeys();
        user = User(name=form.name.data, username=form.username.data, email=form.email.data, password=hashed_password, key = keyGen);
        db.session.add(user);
        db.session.commit();
        login_user(user);
        nextPage = request.args.get('next');
        flash(f'Account created for @{form.username.data}! You are now logged in as well.', 'success')
        return redirect(nextPage) if nextPage else redirect(url_for('home'));
    return render_template('register.html', form=form)
```

Figure 6.3: Routing method for User Registration

Figure 6.3 shows the routing method for user registration, where the user entered password is encrypted and hashed, the key is also generated. New user object is instantiated and finally the user is registered successfully.

```
@app.route("/login", methods=['GET', 'POST'])
def login():
    form = LoginForm()
    if form.validate_on_submit():
        user = User.query.filter_by(email=form.email.data).first();
        if user and bcrypt.check_password_hash(user.password, form.password.data):
            login_user(user, remember=form.remember.data);
            nextPage = request.args.get('next');
            flash(f'Welcome! You are now logged in', 'success');
            return redirect(nextPage) if nextPage else redirect(url_for('home'));
        else:
            flash('Login Unsuccessful. Please check email and password', 'danger');
            #return redirect(url_for('login'))
    return render_template('login.html', form=form);
```

Figure 6.4: Routing method for User Login

Figure 6.4 shows routing method for User Login, where the user is checked for authorization. If the authorization is successful, then the user will be indirected to the homepage else the login is unsuccessful.

```
def addGenesisBlock(self):
    tArr = [];
    tArr.append(Transaction("me", "you", 10));
    genesis = Block(tArr, datetime.now().strftime("%m/%d/%Y, %H:%M:%S"), 0);

    genesis.prev = "None";
    return genesis;
```

Figure 6.5: Adding Genesis Block

Figure 6.5 shows the method for adding the genesis block, every time the application is started. Basically, the genesis block is the initial block in the blockchain network where it just holds the hash of the next block. But, the hash of the previous block is null.

```
def generateKeys(self):
    key = RSA.generate(2048)
    private_key = key.export_key()
    file_out = open("private.pem", "wb")
    file_out.write(private_key)

    public_key = key.publickey().export_key()
    file_out = open("receiver.pem", "wb")
    file_out.write(public_key)

    print(public_key.decode('ASCII'));
    return key.publickey().export_key().decode('ASCII');
```

Figure 6.6: Generation of Keys

Figure 6.6 shows the generation of keys for that particular user. Key generation uses RSA algorithm.

```
def calculateHash(self):

    hashTransactions = "";

    for transaction in self.transactions:
        hashTransactions += transaction.hash;
    hashString = str(self.time) + hashTransactions + self.gym + self.prev + str(self.nonse);
    hashEncoded = json.dumps(hashString, sort_keys=True).encode();
    return hashlib.sha256(hashEncoded).hexdigest();
```

Figure 6.7: Calculating Hash

Figure 6.7 shows the method for calculating the hash for a node. Since a block in a blockchain network contains the hash of the current block and the previous block, this method is used to calculate the hash for particular node.

```
@app.route("/transaction", methods=['GET', 'POST'])
def transaction():
    form = TransactionForm();
    formNL = TransactionFormNotLoggedIn();
    if form.validate_on_submit():
        print("hi");

        user = User.query.filter_by(username=form.reciever.data).first();
        if not user:
            flash(f'Invalid Reciever Username', 'danger');
        else:
            feedback = blockchainObj.addTransaction(form.sender.data, form.reciever.data, form.amount.data, form.key.data, form.key.data)
            if feedback:
                flash(f'Transaction Made!', 'success');
            else:
                flash(f'Error!', 'danger');
            return render_template('transaction.html', title = "Transaction", blockchain = blockchainObj, form=form, formNL= formNL);
    return render_template('transaction.html', title = "Transaction", blockchain = blockchainObj, form=form, formNL= formNL);
```

Figure 6.8: Routing method for New Transaction

Figure 6.8 shows the routing method for new transaction. Initially it validates the entered details by the user. Further on it checks if the entered receiver's user id is valid or not. If not, it throws a pop-up message stating that it is a invalid receiver's username else the add transaction method will be called by passing the sender's username, receiver's username, the entered number of coins, sender's key and receiver's key. If this method returns true, then the transaction is susccessfully completed else the transaction is unsuccessful.

```
def addTransaction(self, sender, reciever, amt, keyString, senderKey):
    keyByte = keyString.encode("ASCII");
    senderKeyByte = senderKey.encode("ASCII");

    #print(type(keyByte), keyByte);

    key = RSA.import_key(keyByte);
    senderKey = RSA.import_key(senderKeyByte);

    if not sender or not reciever or not amt:
        print("transaction error 1");
        return False;

    transaction = Transaction(sender, reciever, amt);

    transaction.signTransaction(key, senderKey);

    if not transaction.isValidTransaction():
        print("transaction error 2");
        return False;
    self.pendingTransactions.append(transaction);
    return len(self.chain) + 1;
```

Figure 6.9: Method for adding a transaction

Figure 6.9 shows the method for adding transaction. If the sender or the receiver or the entered number of coins is not valid, then a error is thrown else a new transaction object is instantiated and that particular transaction is signed using the private key of the sender. If the transaction is not valid, Boolean false is returned. Further this transaction is appended to the pending transaction list.

```
@app.route('/mine', methods=['GET'])
def mine():
    print("madeit");
    miner = request.args.get('miner', None);
    lastBlock = blockchainObj.getLastBlock();

    if len(blockchainObj.pendingTransactions) <= 1:
        flash(f'Not enough pending transactions to mine! (Must be > 1)', 'danger');
    else:
        feedback = blockchainObj.minePendingTransactions(miner);
        if feedback:
            flash(f'Block Mined! Your mining reward has now been added to the pending transactions!', 'success');
        else:
            flash(f'Error!', 'danger');
    return render_template('minerPage.html', title = "Mine", blockchain = blockchainObj);
```

Figure 6.10: Routing method for Mining a block

Figure 6.10 shows routing method for mining a block. If the pending transaction is less than or equal to one, it throws a message saying that not enough pending transactions to mine. Hence, whichever transactions are pending, it will be mined by passing the miner. If the mine pending transactions method becomes true, then mining is successful else mining becomes unsuccessful.

```
def minePendingTransactions(self, miner):

    lenPT = len(self.pendingTransactions);
    if(lenPT <= 1):
        print("Not enough transactions to mine! (Must be > 1)")
        return False;
    else:
        for i in range(0, lenPT, self.blockSize):

            end = i + self.blockSize;
            if i >= lenPT:
                end = lenPT;

            transactionSlice = self.pendingTransactions[i:end];

            newBlock = Block(transactionSlice, datetime.now().strftime("%m/%d/%Y, %H:%M:%S"), len(self.chain));
            #print(type(self.getLastBlock()));

            hashVal = self.getLastBlock().hash;
            newBlock.prev = hashVal;
            newBlock.mineBlock(self.difficulty);
            self.chain.append(newBlock);
        print("Mining Transactions Success!");

        payMiner = Transaction("Miner Rewards", miner, self.minerRewards);
        self.pendingTransactions = [payMiner];
    return True;
```

Figure 6.11: Method to Mine the Pending Transactions

Figure 6.11 shows method to mine the pending transactions. If the pending transaction is less than or equal to one, it throws a message saying that not enough pending transactions to mine. Else, each transaction in the pending transaction list will be mined individually. Further for each transaction, a new block is created with the current time stamp. The Hash of this block will be generated. The value for the hash of the previous block, will be fetched from the previous and assigned to it. Finally, the mining is succeeded.

Chapter 7

System Testing

Testing is a procedure of executing the program with unequivocal intention of discovering mistakes, assuming any, which makes the program, fall flat. This stage is an essential piece of the product improvement.

It plays out an exceptionally basic part for quality affirmation and for guaranteeing unwavering quality of programming. It is the way toward finding the mistakes and missing operation and furthermore an entire confirmation to decide if the targets are met the client prerequisites are fulfilled.

The objective of testing is to reveal prerequisites, outline or coding blunders in the projects. Therefore, unique levels of testing are utilized in programming frameworks. The testing results are utilized amid upkeep .

This area manages the points of interest in the various classes of the test which should be directed to approve capacities, imperatives and execution. This can be accomplished fundamentally by using the methods for testing, which assumes a crucial part in the improvement of a product.

The structure of the program is not being considered in useful testing. Test cases are exclusively chosen on the premise of the prerequisites or particulars of a program or module of program but the internals of the module or the program are not considered for determination of experiments.

The program to be tried is executed with an arrangement of experiments and the yield of the program for the experiments is assessed to decide whether the program is executing not surprisingly. The accomplishment of testing in uncovering mistakes in projects depends basically on the experiments. There are two fundamental ways to deal with testing Black Box or functional Testing and White Box or structural testing.

Table 7.1: Work Flow

Sl No	Work	Duration(in Weeks)
1	Environment Setup and Installations	1
2	Frontend development & Database Setup	3
3	Information Collection on Blockchain concepts	1
4	Library Installation	3
5	Development of Business Logic	6
6	Integrating Frontend and Backend	1
7	Validation & Testing	2

Table 7.2: Test cases from User Perspective

TC#	Description	Expected Result	Actual Result	Status
TC-1	User registering through already taken User name or Mobile number	Throw an error message stating that the user name/ Mobile number is already taken	Throws an error message	Pass
TC-2	User trying login through invalid username or password	Throws an error stating that the entered user credentials is incorrect	Throws an error message	Pass
TC-3	User entering the invalid user name for the receiver's address while making a transaction	Throws an error stating that the entered receiver's username is invalid	Throws an error message	Pass
TC-4	User trying to mine blocks that has no pending transactions	Throws an error stating that there are no enough transaction to mine	Throws an error message	Pass
TC-5	User trying to register with appropriate details	Successfully registered and the user is added	Successfully Registered	Pass
TC-6	User trying to Login with user credentials	User authorized and login successful	Successful Login	Pass
TC-7	User trying to transact coins to other peer	Transaction is successfully completed	Successful Transaction	Pass

Chapter 8

Results and Disscussion

This chapter contains all the outcomes of the project.

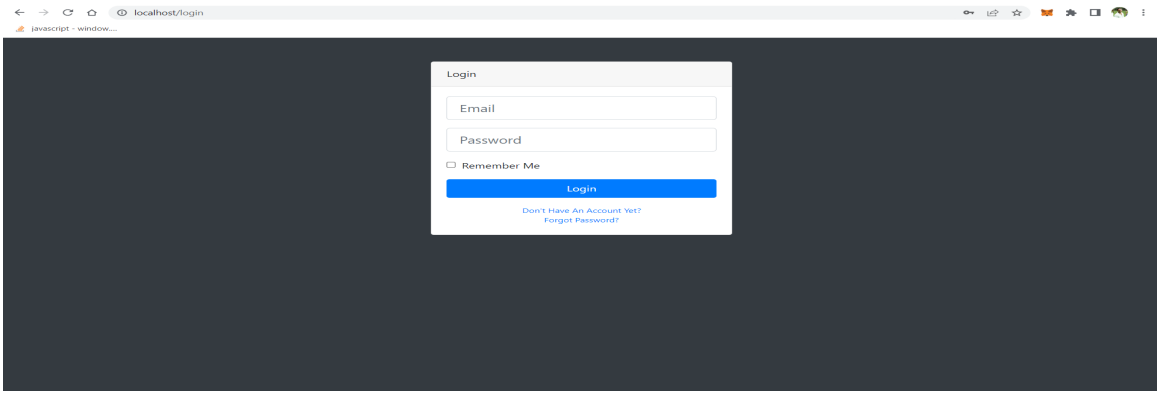

Figure 8.1: The Login page

Figure 8.1 shows the login page where the user enters the user credentials and gets authorized into the website.

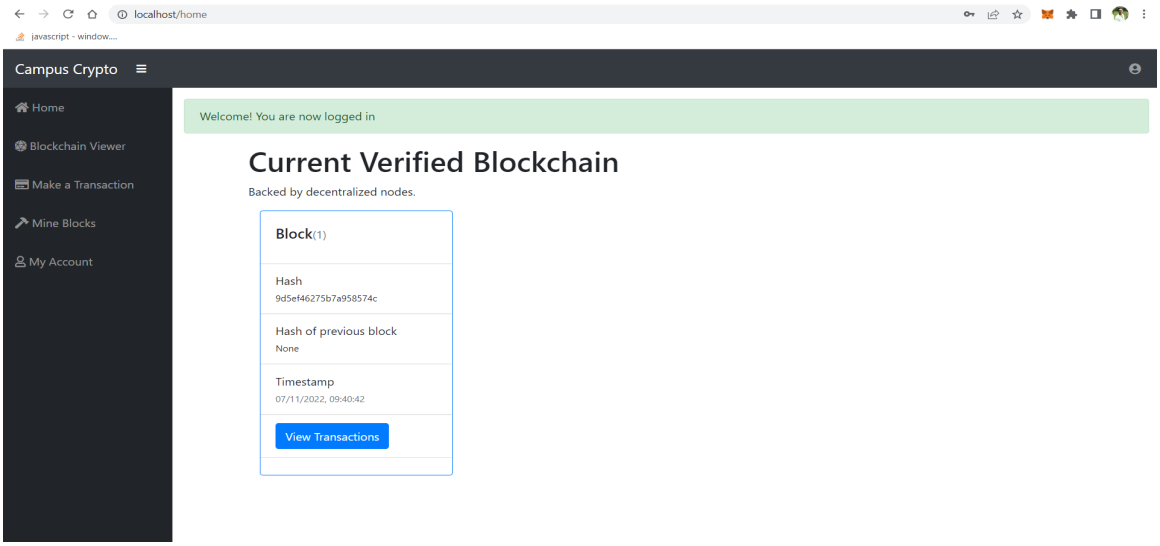

Figure 8.2: The Home Page

Figure 8.2 shows the home page where it shows all the transaction blocks which is

created each time a new transaction is initiated. Initially, there will be only one block known as the Genesis Block.

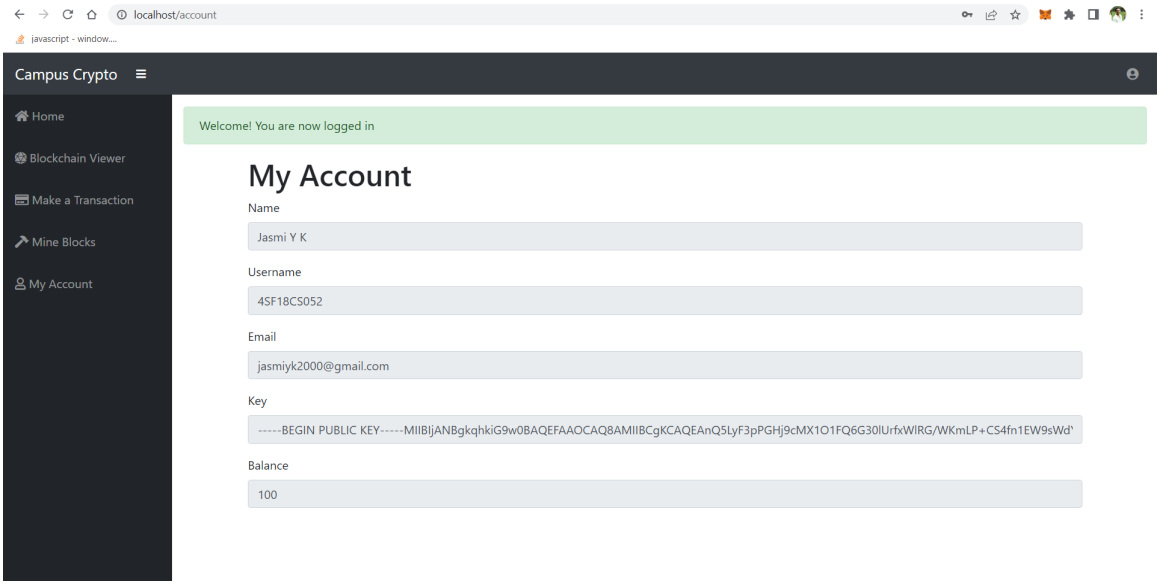

Figure 8.3: The Profile Page

Figure 8.3 shows the profile page where the account details of the user are shown. It consists of the name, username, email, key and the balance amount respectively.

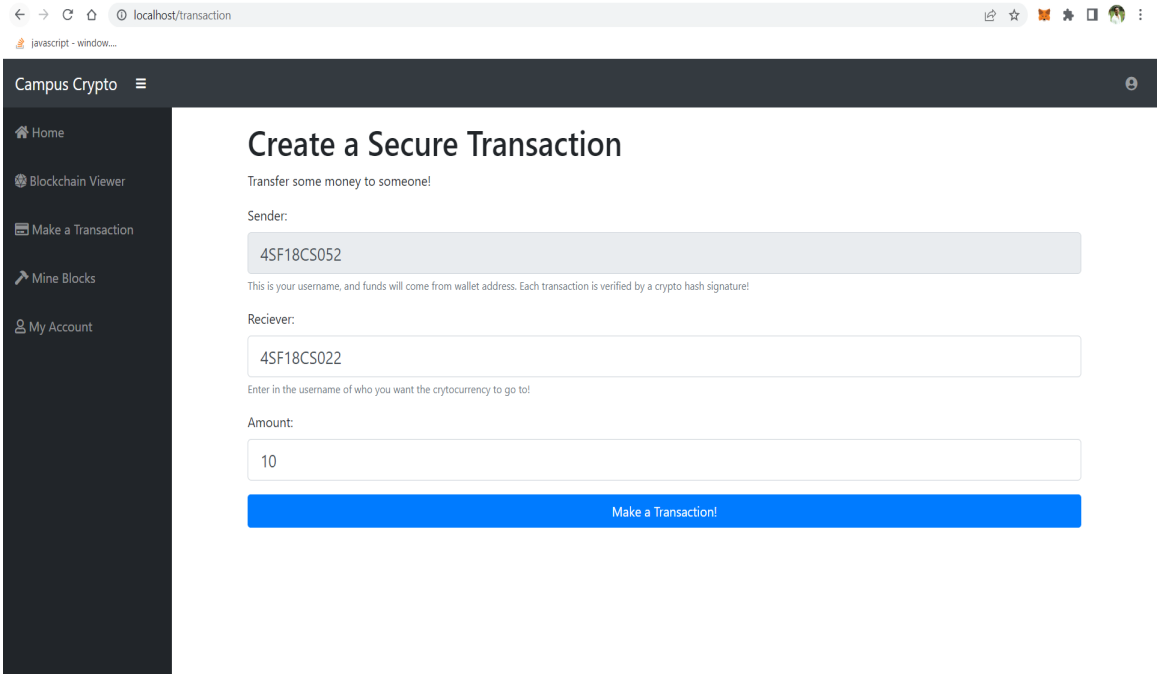

Figure 8.4: The Transaction Page

Figure 8.4 shows the transaction page where the user can transact coins to any authorized user, where the user can enter the receiver's user id and the number of coins he/she wants to send.

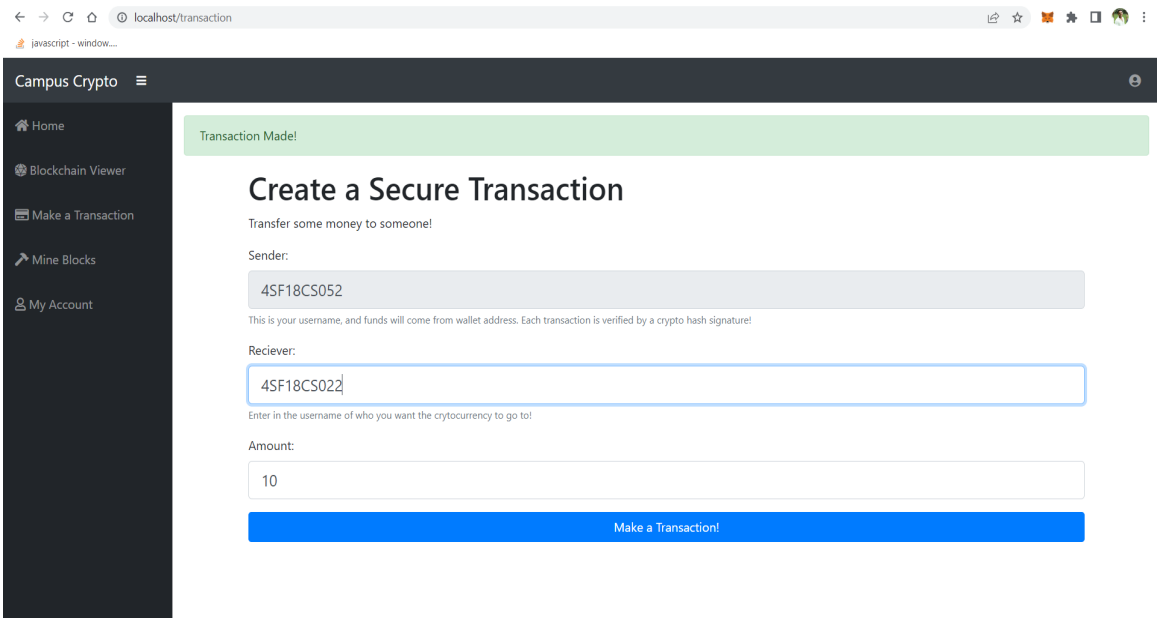

Figure 8.5: The Successful Transaction

Figure 8.5 shows the successful transaction pop-up message indicating that the transaction initiation was successful.

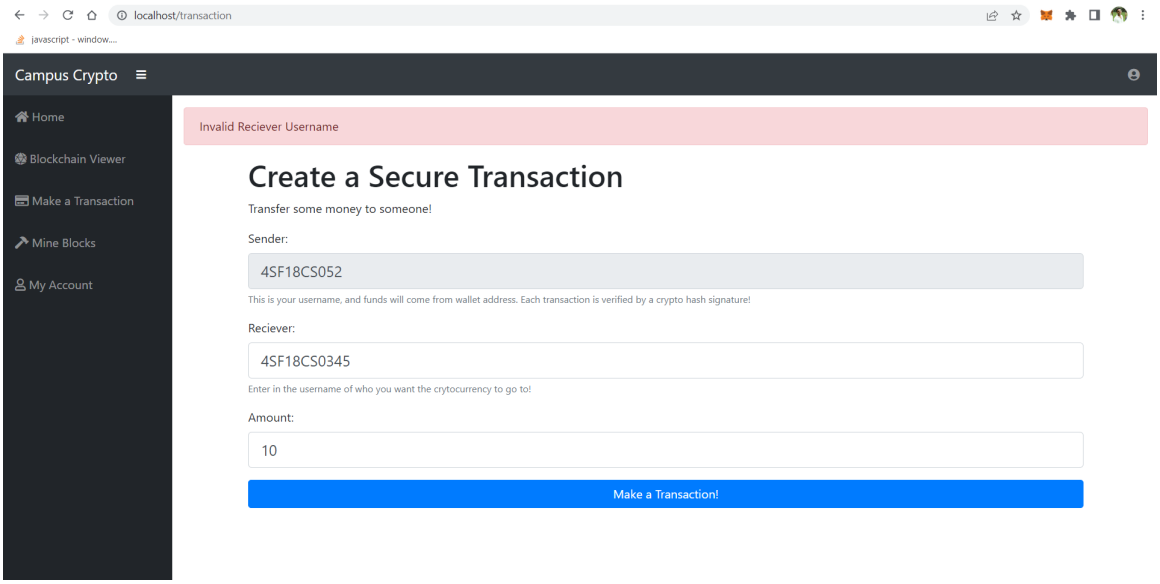

Figure 8.6: The Invalid User Transaction

Figure 8.6 shows the invalid user pop-up message indicating that the entered receiver's user id is invalid.

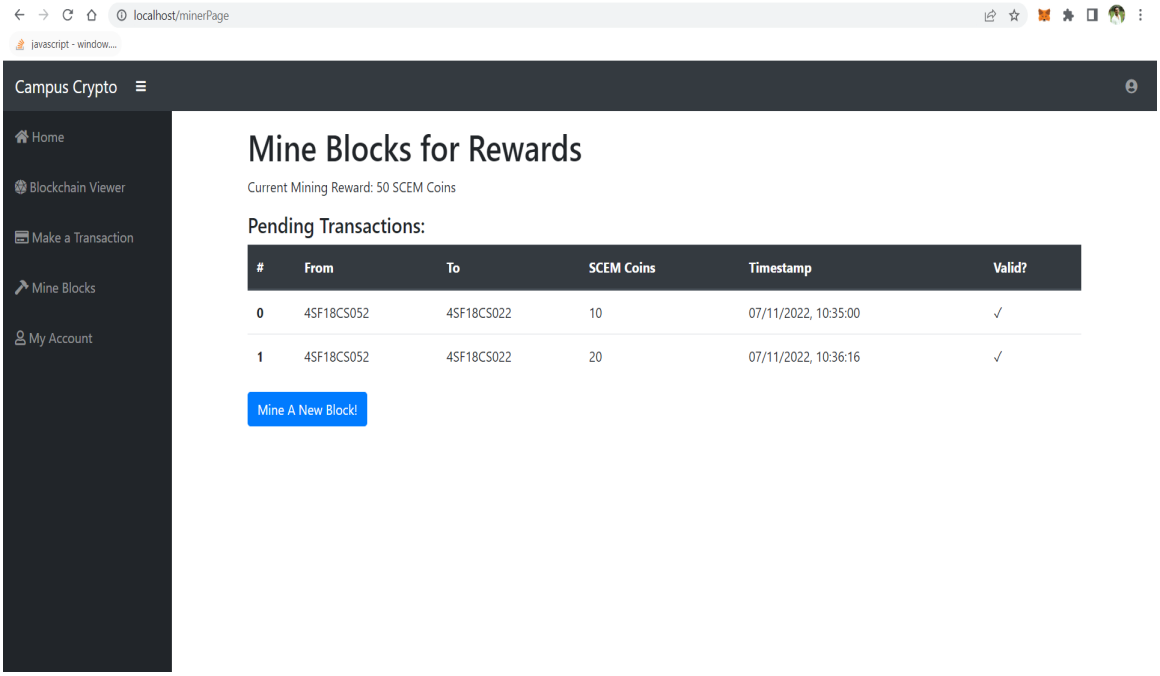

Figure 8.7: The Mine Block Page

Figure 8.7 shows all the recent transactions initiated. These transactions are said to be fully completed only when they are mined by the peers in the network. So, the user has to click on the mine block button to mine all the pending transactions.

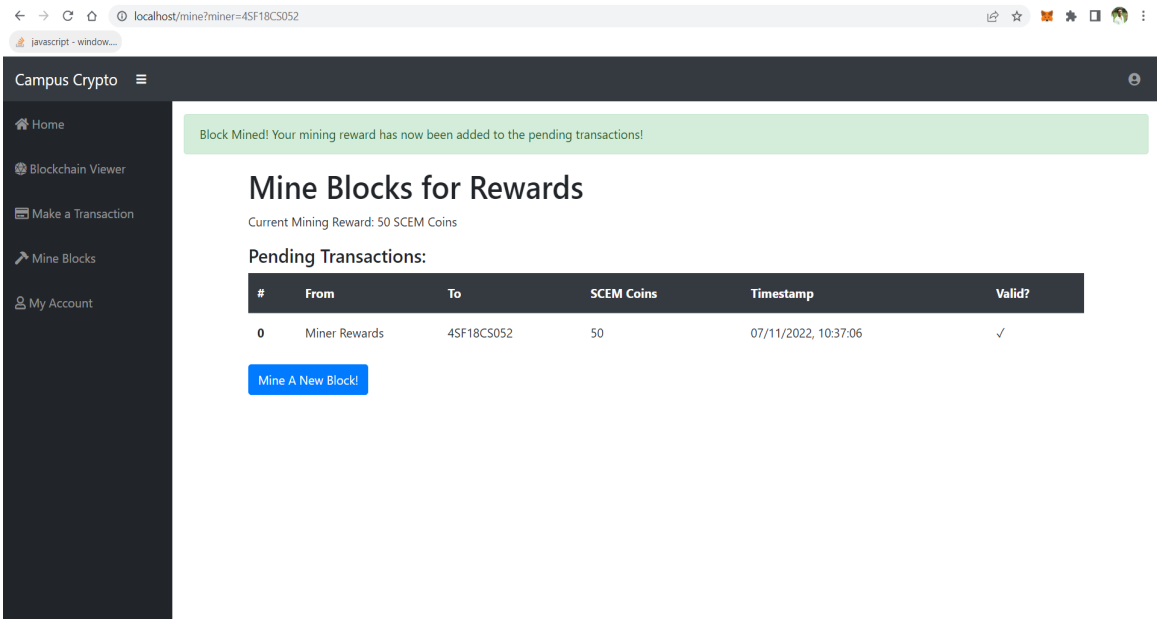

Figure 8.8: After Mining

Figure 8.8 shows the page after mining the pending transactions and the current pending transaction will be the reward that will be credited to the user when a new transaction is initiated in the network.

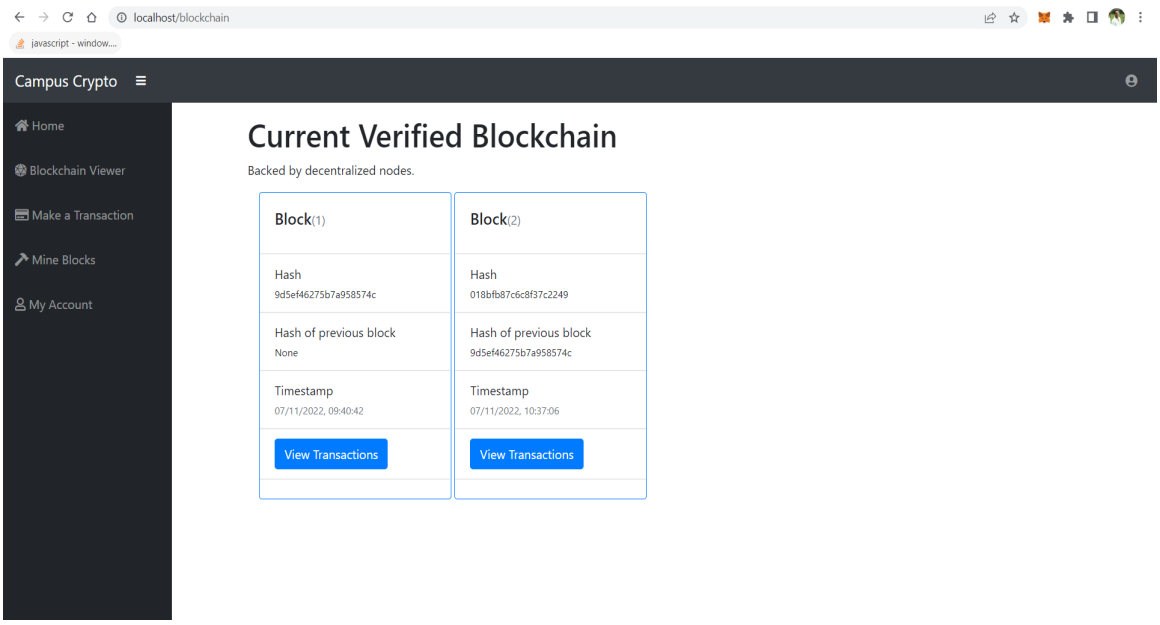

Figure 8.9: Blockchain Overview

Figure 8.9 shows the new block that is added to the existing blockchain. The new block refers to the previous transactions.

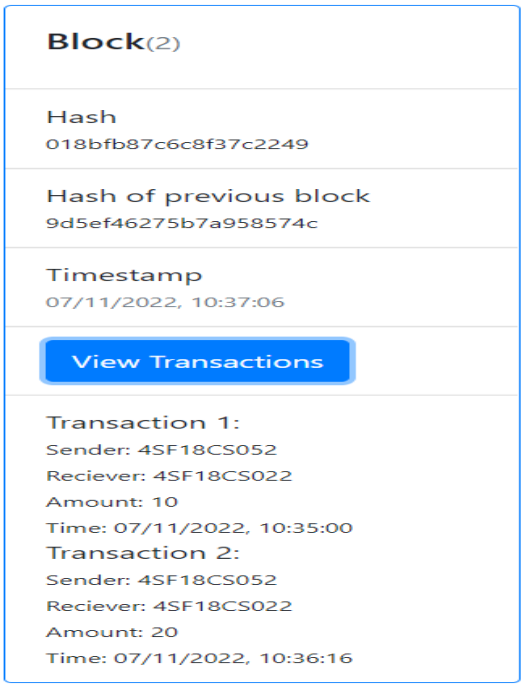

Figure 8.10: Block Details Page

Figure 8.10 shows the new block details. This block consists of the hash of the previous block and hash of its own block. It also contains the transaction details.

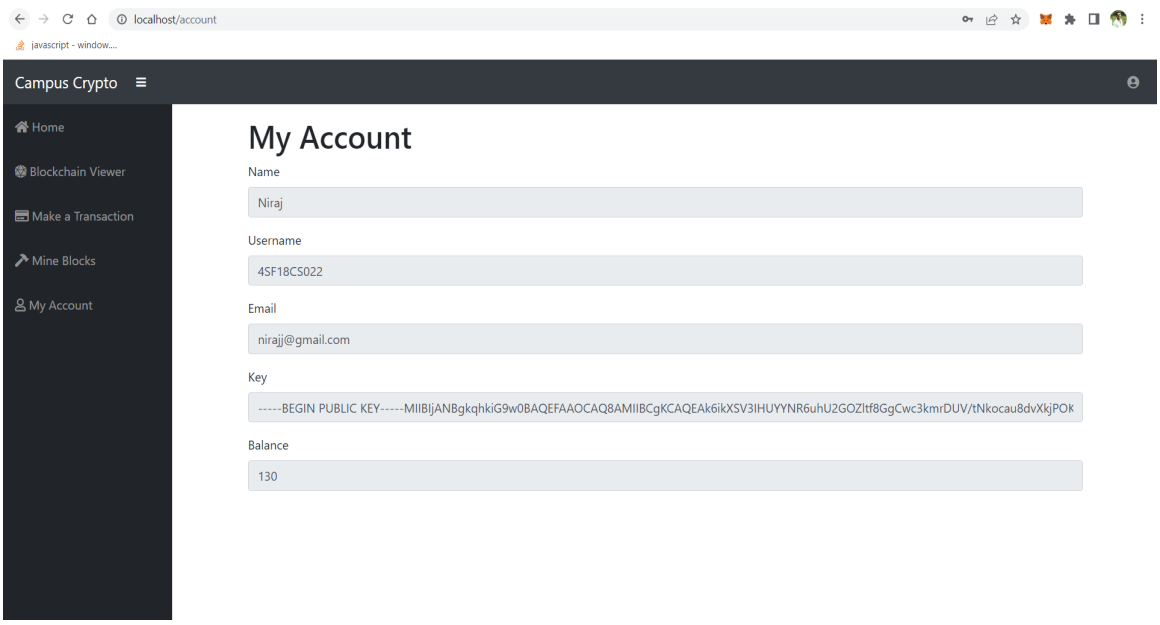

Figure 8.11: The Receiver Page

Figure 8.11 shows the receiver's account details where the balance is updated due to the previous transaction.

Chapter 9

Conclusion and Future work

The world is moving towards digital inclusion and the digital payments are increasing at a very high rate. Almost all the nations are switching to digital mode of payments. But all the payment methods are mostly centralized and hence susceptible to be hacked. A decentralized crypto currency based system inside an organization can act as an starting point for decentralized crypto currency. This will help the students to understand the working of crypto currency and gain experience at a smaller simulated scale. The system also intends to replace the current physical tokens that are used inside an organization. These physical tokens are easily replicable and are also comparatively costlier to produce. It is mostly likely that the users can forget to carry the tokens or might get lost in the long run. To avoid these we use the customized and organization specific approach for creating the decentralized crypto currency. The system also aims at replacing the sodexo like centralized token system and also centralized rewards based systems. Which will eliminate the threat of data manipulation as a single authority would not be responsible for managing the system. This eliminates the risk of maintaining a centralized database and due to decentralization, consensus and authentication can be done through the decentralized ledger.

Due to constant improvements in technology there is always scope for enhancements in proposed system. Especially when the project is based on a new and emerging technology like blockchain and thus provides good scope for the future. The web based application provided can be transformed into a Progressive Web Application(PWA) or browser plug in that will help users with easy access Currently only organization specific transactions are allowed, which can be later upgraded to any transactions around the world using conversion from tokens to local currency. This will provide flexibility for users to use the tokens in daily life.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[2] M. Azman and K. Sharma, "Hch dex: A secure cryptocurrency e-wallet & exchange system with two-way authentication," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 305–310, IEEE, 2020.

[3] A. K. Shrestha and J. Vassileva, "Bitcoin blockchain transactions visualization," in *2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB)*, pp. 1–6, IEEE, 2018.

[4] M. Di Angelo and G. Salzer, "Characteristics of wallet contracts on ethereum," in *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, pp. 232–239, IEEE, 2020.

[5] U. Khan, Z. Y. An, and A. Imran, "A blockchain ethereum technology-enabled digital content: Development of trading and sharing economy data," *IEEE access*, vol. 8, pp. 217045–217056, 2020.

[6] C. V. Madhavan, C. Srikanth, and H. K. Swamy, "Vsk chains: Integrated content and currency transaction blockchains," in *2017 23RD Annual International Conference in Advanced Computing and Communications (ADCOM)*, pp. 18–26, IEEE, 2017.

[7] S. Alzahrani and T. U. Daim, "Analysis of the cryptocurrency adoption decision: Literature review," in *2019 Portland International Conference on Management of Engineering and Technology (PICMET)*, pp. 1–11, IEEE, 2019.

[8] X. F. Liu, H.-H. Ren, S.-H. Liu, and X.-J. Jiang, "Characterizing key agents in the cryptocurrency economy through blockchain transaction analysis," *EPJ Data Science*, vol. 10, no. 1, p. 21, 2021.

[9] S. Suratkar, M. Shirole, and S. Bhirud, "Cryptocurrency wallet: A review," in *2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, pp. 1–7, IEEE, 2020.

[10] A. Khan, M. Ansari, Y. M. Ishaque, M. Khan, A. Ahmedali, *et al.*, "Hardware cryptocurrency wallet," 2022.

[11] X. He, J. Lin, K. Li, and X. Chen, "A novel cryptocurrency wallet management scheme based on decentralized multi-constrained derangement," *IEEE Access*, vol. 7, pp. 185250–185263, 2019.

[12] W. Dai, J. Deng, Q. Wang, C. Cui, D. Zou, and H. Jin, "Sblwt: A secure blockchain lightweight wallet based on trustzone," *IEEE access*, vol. 6, pp. 40638–40648, 2018.

[13] S. Albeshr and H. Nobanee, "Blockchain applications in banking industry: A mini-review," *Available at SSRN 3539152*, 2020.

[14] A. G. Khan, A. H. Zahid, M. Hussain, and U. Riaz, "Security of cryptocurrency using hardware wallet and qr code," in *2019 International Conference on Innovative Computing (ICIC)*, pp. 1–10, IEEE, 2019.

[15] Z. Jian, Q. Ran, and S. Liyan, "Securing blockchain wallets efficiently based on threshold ecdsa scheme without trusted center," in *2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)*, pp. 47–51, IEEE, 2021.

Appendix

Jasmi Y K-4SF18CS052

ORIGINALITY REPORT

12%	5%	6%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Visvesvaraya Technological University Student Paper	3%
2	ieeexplore.ieee.org Internet Source	1%
3	Mohamed Azman, Kunal Sharma. "HCH DEX: A Secure Cryptocurrency e-Wallet & Exchange System with Two-way Authentication*", 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020 Publication	1%
4	memoryleak.weebly.com Internet Source	1%
5	www.coursehero.com Internet Source	1%
6	Zhou Jian, Qu Ran, Sun Liyan. "Securing Blockchain Wallets Efficiently Based on Threshold ECDSA Scheme Without Trusted Center", 2021 Asia-Pacific Conference on	<1%

	Communications Technology and Computer Science (ACCTCS), 2021 Publication	
7	www.studymode.com Internet Source	<1 %
8	Submitted to University of Sunderland Student Paper	<1 %
9	Submitted to University of Ulster Student Paper	<1 %
10	"Blockchain and Trustworthy Systems", Springer Science and Business Media LLC, 2020 Publication	<1 %
11	Hanqing Wu, Jiannong Cao, Yanni Yang, Cheung Leong Tung, Shan Jiang, Bin Tang, Yang Liu, Xiaoqing Wang, Yuming Deng. "Data Management in Supply Chain Using Blockchain: Challenges and a Case Study", 2019 28th International Conference on Computer Communication and Networks (ICCCN), 2019 Publication	<1 %
12	Monika di Angelo, Gernot Salzer. "Characteristics of Wallet Contracts on Ethereum", 2020 2nd Conference on Blockchain Research & Applications for	<1 %

	Innovative Networks and Services (BRAINS), 2020 Publication	
13	ebin.pub Internet Source	<1 %
14	www.researchgate.net Internet Source	<1 %
15	Submitted to Jacobs University, Bremen Student Paper	<1 %
16	Guojia Li, Lin You. "A Consortium Blockchain Wallet Scheme Based on Dual-Threshold Key Sharing", Symmetry, 2021 Publication	<1 %
17	Zibin Zheng, Shaoan Xie, Hong Ning Dai, Xiangping Chen, Huaimin Wang. "Blockchain challenges and opportunities: a survey", International Journal of Web and Grid Services, 2018 Publication	<1 %
18	Submitted to INTI University College Student Paper	<1 %
19	Ajay Kumar Shrestha, Julita Vassileva. "Bitcoin Blockchain Transactions Visualization", 2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB), 2018 Publication	<1 %

20	agile-giss.copernicus.org Internet Source	<1 %
21	Submitted to Bournemouth University Student Paper	<1 %
22	epjds.epj.org Internet Source	<1 %
23	Arthur Henrique de Andrade Melani. "Diagnose de falhas em sistemas baseada em redes bayesianas e SysML.", Universidade de Sao Paulo, Agencia USP de Gestao da Informacao Academica (AGUIA), 2020 Publication	<1 %
24	Submitted to University of Westminster Student Paper	<1 %
25	Xiaojian He, Jinfu Lin, Kangzi Li, Ximeng Chen. "A Novel Cryptocurrency Wallet Management Scheme Based on Decentralized Multi-Constrained Derangement", IEEE Access, 2019 Publication	<1 %
26	downloads.hindawi.com Internet Source	<1 %
27	3dvar.com Internet Source	<1 %
28	cps-vo.org Internet Source	<1 %

29	deepai.org Internet Source	<1 %
30	Ch. V. N. U. Bharathi Murthy, M. Lawanya Shri, Seifedine Kadry, Sangsoon Lim. "Blockchain Based Cloud Computing: Architecture and Research Challenges", IEEE Access, 2020 Publication	<1 %
31	Saurabh Suratkar, Mahesh Shirole, Sunil Bhirud. "Cryptocurrency Wallet: A Review", 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP), 2020 Publication	<1 %
32	C. E. Veni Madhavan, Ch. Srikanth, H.V. Kumar Swamy. "VSK Chains: Integrated Content and Currency Transaction Blockchains", 2017 23RD Annual International Conference in Advanced Computing and Communications (ADCOM), 2017 Publication	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		