A Study on Blockchain and Cryptography

By

Ms. Sanika Pratap Raut

Mr. Sarthak Shantaram Jagtap

Mr. Siddhesh Avinash Sankpal

Under the Guidance of

Mrs. Poonam Thakare
Department of Computer Engineering

Vidya Vikas Education Trust's Universal College of Engineering

Kaman, Vasai - 401208

Abstract

This project explores the development of a decentralized chat application leveraging blockchain technology. Unlike traditional chat apps that rely on central servers, our application utilizes a blockchain network to securely manage messages and user interactions.

By distributing data across a network of nodes, the system enhances security, privacy, and resistance to censorship. Users maintain control over their personal data, as the blockchain ensures that all messages are encrypted and immutable. This approach not only reduces the risk of data breaches but also fosters a more transparent and equitable communication environment. Our solution aims to address the limitations of conventional chat systems and offer a robust alternative for secure digital communication.

With the rapid development of Internet technology in recent years, Electronic Commerce has been an inevitable product of the economy, the science and the technology. This project is an online bookstore. This application is developed using Android, PHP and MySQL. The main objective of the project is to create an online book store that allows users to search and purchase a book online based on title, author and subject. The selected books are displayed in a list format and the user can purchase their books online through credit card payment. Using this app the user can purchase a book online instead of going out toa book store and wasting time. The customer can purchase books online. Through a web browser the customers can search for a book by its title or author, later can purchase using credit card transaction. They

should give the details of their name, contact number and email. The user can also give feedback to a book by giving ratings on a score of five. The Administrator will have additional functionalities when compared to the common user. He can add, delete and update the book details, book categories, member information and also assign a new user role.

Keywords: Android, PHP, Online Book Store, Internet, Decentralized, Html, CSS, JavaScript.

Chapter 1

Introduction

The BlockChain is the invention that allows digitally generated information to be allocated without being copied. BlockChain Technology is the heart of the new interneti.e. digital currency, BitCoin and any other online transaction. Tech experts found a big potential in this technology."BlockChain is an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions but virtually everything of value.. In plain layout, the data is not owned by any single computer but by a chain of computers so that the blocks of data are secured and bound to each other using a chain, that technology is known as BlockChain technology. There is no transaction cost due to BlockChain, in Layman language BlockChain is a process to pass information or data from A to B in a safe and automated manner. Cryptocurrency works on the principle of BlockChain Technology, that is why, BlockChain is the most trending item of current era, due to it's secure nature cryptocurrency is widely accepted.

Project Idea

This project explores the development of a decentralized chat application leveraging blockchain technology. Unlike traditional chat apps that rely on central servers, our application utilizes a blockchain network to securely manage messages and user interactions.

By distributing data across a network of nodes, the system enhances security, privacy, and resistance to censorship. Users maintain control over their personal data, as the blockchain ensures that all messages are encrypted and immutable. This approach not only reduces the risk of data breaches but also fosters a more transparent and equitable communication environment. Our solution aims to address the limitations of conventional chat systems and offer a robust alternative for secure digital communication.

As we all know, traditional chat applications are centralized i.e., all the data is stored on a centralized server. Therefore, major problem of this structure is, if the central server fails then whole network collapses. For example, WhatsApp server stored all the data on a central server, if in case that server is destroyed then there can be a loss of user data, or they can even leak the user information stored on the server. To overcome this, our project makes the use of decentralized Application approach (dApps). In our application all the user data is stored on a block which is connected to other blocks forming a chain. As the name suggests, a decentralized application does not have a centralized server. It is basically a peer-to-peer network. Also, the data that is stored in block is almost impossible to view as a very secure encryption and hashing functions (256 bits) are used. Also are a hacker tries to make changes to the information in block then, he/she will have to make changes to all the copies of that block on whole blockchain network and that can be quite impossible. Though block is on all nodes, they cannot access the information in it, only the person for whom the information if can access it.

Chapter 2

Review of Literature

A literature survey was conducted to find various papers published in international journals, such as IEEE, related to tracing missing people using facial recognition to obtain the best algorithm for the same.

2.1 Existing System

A decentralized chat application leveraging blockchain technology can enhance security, privacy, and user control by eliminating centralized servers. One notable example is **Whisper**, a communication protocol built on the Ethereum blockchain that enables private messaging through end-to-end encryption. Whisper allows users to send messages directly between peers, with the blockchain serving as a means for identity verification and message storage. Another example is **Status**, an open-source messaging app that combines a secure messaging platform with a Web3 wallet, enabling users to chat, browse decentralized applications, and transact using cryptocurrencies. Both systems utilize blockchain's immutability and transparency, ensuring that messages cannot be tampered with and providing users with full ownership of their data. Additionally, platforms like **Matrix** have integrated blockchain solutions for decentralized identity management, further enhancing privacy and interoperability in the decentralized chat space. Overall, these existing systems demonstrate the potential of blockchain technology to create

secure, decentralized communication channels while empowering users with greater control over their interactions.

2.2 Literature Survey

We examined various research papers in the domain of AI-Interviews or online interviews for our project to delve deeper into the details of the various studies conducted in the field of AI interviews. Table 2.1 shows a survey of the research papers conducted for the project.

Paper No.	Paper Title	Year	Advantages	Dis-Advantages
1	An Overview of Blockchain Technology (Zibin Zheng, Shaoan Xien)	2017	Comprehensive overview of blockchain's architecture and consensus mechanisms	Scalability and latency issues still unresolved in large-scale blockchain systems
2	Research for Practice: Cryptocurrencies, Blockchains, and Smart Contracts (Narayanan & Miller)	2017	Discusses the integration of smart contracts with blockchain technology, offering real-world use cases	Legal and regulatory challenges need to be addressed for widespread adoption
3	Blockchain Revolution (Don and Alex Tapscott)	2016	Explores blockchain's potential to transform various sectors, offers insights into its applications	Lacks in-depth discussion on technical limitations and scalability challenges
4	Bitcoin: A Scalable Blockchain Protocol (I. Eyal et al.)	2016	Proposes solutions to improve blockchain scalability while maintaining security	Mining power consolidation still an issue; storage limitations persist

5	The Mini-Blockchain Scheme (J. Bruce)	2014	Proposes a more efficient blockchain model that reduces storage requirements	Early-stage concept with few practical implementations; potential security trade-offs
6	Zerocoin: Anonymous Distributed E-cash from Bitcoin (Miers et al.)	2013	Enhances privacy by introducing anonymity in Bitcoin transactions	Increased computational overhead and complexity in implementation
7	Bitcoin: A Peer-to-Peer Electronic Cash System (Satoshi Nakamoto)	2008	First implementation of blockchain technology with Bitcoin, decentralized currency concept	High energy consumption due to mining, scalability issues, vulnerable to illicit use

Table 2.1 – Literature Survey table

Decentralized application makes use of peer-to-peer networks, this ensures that no network failure can occur due to central node failure. Blockchain serves as an immutable ledger which allows messaging to take place in a decentralized manner. A decentralized application for communication and resource sharing is need in today's world, where keeping data on a centralized server can be risky and costly experience. With the help of various consensus, we can implement different ways to share resources and communicate. Together with Blockchain and Decentralized Applications, we can create a secure and reliable messaging application that overcomes the drawbacks of traditional messaging applications Is a paper in which author has introduced all the uses and possible ways the blockchain can be used along with decentralization. Also, the author emphasizes on, what the future blockchain applications will be. There is also a detailed report on advantages it provides, different areas in which blockchain can improve computing and how it is better and the current traditional systems. Most popular online chat sites used by millions of users are WhatsApp and Snapchat. As per recent news of August Here are the few takeaways from the article: 1. The vulnerability concerns WhatsApp's encryption process, which is meant to protect every 2. message, picture, call, video or other content sent in chats. 2. However, when decrypted, the Check Point team realized that the protocols being used by WhatsApp could be converted and accessed, allowing them to see exactly what rules were being used, and also to change them to their liking. 3. This may allow hackers to modify a group chat text by putting words or using the quote feature in a chat group discus Sign to modify the sender's identity. 4. Hackers could also send a personal message to another group participant posing as a public message for all so that when the person in question answers, he is visible to all.

The similarities of the system can be related to how Snapchat works on the frontend. By default, upon the chat window is closed everyone message in the conversation is deleted from the user's device. This makes the idea rather confusing to naive users. This idea makes more sense in a scenario where the privacy of the

conversation is really important. The message is deleted from the conversation the companies claim that the information is deleted off the companies' servers unless the conversation is reported for violating the policies of the application. In that case the conversation is reviewed before it is deleted from the company servers. In either case the integrity of privacy is often affected in some way of the other. These messaging systems never function on a peer-toper basis. There is no relation to each computer of your friends (from your device). You attach your computer to your server instead. You can then use a custom TCP protocol to send your communications to the server, or perhaps to HTTP. The procedure of visiting a website is firstly your browser requests the server upon which the web service is running on. Even in case in which these servers are far remotely across the world.

2.3 Problem Statement and Objective

Problem statement:

The systems we currently use have a centralized approach to resource sharing and communication. Here, all the data • rules to follow:1. Merkle tree root hash: the hash value of all the transactions in the block. 2.Timestamp: current time as seconds in universal time since January 1, 1970. 3.nBits: target threshold of a valid block hash. 4. Nonce: a 4-byte field, which usually starts with 0 and increases for every hash calculation. 5.Parent block hash: a 256-bit hash values that points to the previous block.

- **Protecting Privacy:** Encrypting messages and spreading data across a network.
- **Preventing Censorship:** Removing central control that could limit free speech.
- **Giving Users Control:** Allowing users to own and manage their own data.
- **Improving Reliability:** Avoiding single points of failure by distributing data.

Objective:

The main Stakeholders of this application are users, I will list the most important quality goals for them:

• Usability: Ease of use by non-technical people.

• Responsiveness: The application must send messages in real time.

• Privacy: The application must keep the data private (this is what solid is based on, each one owns their data). • Security: The application must be secure; users must know who they are talking to and who has access to the chat.

• To provide more secure environment for chatting and

• resource sharing.

• To reduce the possibility of immutability.

2.4 Project Scope

The project aims to develop a decentralized chat application leveraging blockchain technology to enhance security, privacy, and user control by eliminating centralized servers. Key objectives include implementing end-to-end encryption for secure messaging, utilizing blockchain for identity verification, and enabling users to manage their own data. The application will support features such as secure file sharing, group chats, and decentralized storage through solutions like IPFS, ensuring a seamless user experience across web and mobile platforms. The technology stack will consist of blockchain platforms like Ethereum, messaging protocols such as Whisper or Matrix, and modern frontend frameworks like React or React Native. The development will progress through phases including research and design, smart contract development, frontend integration, testing, and launch. Potential challenges include scalability, usability for non-technical users, and compliance with regulatory standards.

Chapter 3

Proposed System

This chapter includes a brief description of the proposed system and explores the different modules involved along with the various models through which this system is understood and represented.

3.1 Analysis/Framework/ Algorithm

The development of a decentralized chat application using blockchain technology involves a multi-layered architecture that ensures security, privacy, and user control. At the core, smart contracts deployed on a blockchain platform like Ethereum facilitate identity verification and manage user interactions, allowing for secure and transparent operations. The application employs a decentralized identity management system utilizing decentralized identifiers (DIDs), enabling users to authenticate themselves without relying on central authorities. For real-time messaging, protocols like Whisper or Matrix are integrated to enable peer-to-peer communication with end-to-end encryption. The encryption is accomplished through asymmetric cryptography, where messages are encrypted with the recipient's public key, ensuring that only the intended recipient can decrypt them. Hashing algorithms, such as SHA-256, are used to create unique identifiers for messages, ensuring data integrity and preventing tampering. Additionally, a consensus mechanism—like Proof of Authority or Proof of Stake validates transactions on the blockchain, ensuring that only legitimate users can send messages or create groups. The overall message flow involves user registration via key generation, secure message sending through encryption and direct peer-to-peer routing, and group chat management via smart contracts that control permissions. By addressing performance

considerations such as latency and scalability, this framework aims to deliver a robust decentralized chat application that empowers users with secure and private communication.

3.2 System Requirements

This section will provide the user the required specification of the hardware and software components on which

the proposed system is to be implemented.

3.2.1 Hardware Requirements

This subsection will provide the minimum requirements that must be fulfilled by the hardware components. The hardware requirements are as follows: -

- A desktop with
  - ➢Processor- i7
  - ➢ SSD - 256GB
  - ➢ Memory-16 GB RAM

3.2.2 Software Requirements

This subsection will provide the versions of software applications that must be installed. The software requirements are as follows: -

➢ Windows 10,11
➢ Android Studio
➢ Visual Studio
➢ Xampp

3.3 Design Details

The design of a decentralized chat application using blockchain technology focuses on creating a secure, user-centric communication platform. At the foundation is a blockchain network, such as Ethereum, which hosts smart contracts for user identity management and message integrity. Each user generates a cryptographic key pair for authentication, with the public key registered on the blockchain, ensuring decentralized identity verification. The messaging protocol, potentially based on Whisper or Matrix, facilitates real-time communication, enabling direct peer-to-peer message exchanges. Messages are encrypted using the recipient's public key, ensuring that only the intended recipient can decrypt them with their private key. For data integrity, each message is hashed using a secure algorithm like SHA-256 before being stored on the blockchain, preventing tampering. Additionally, group chats are managed through smart contracts that define permissions and membership, ensuring that only authorized users can participate. The

application employs decentralized storage solutions like IPFS for sharing files and media, providing resilience against data loss. The user interface is designed for simplicity and ease of use, ensuring a seamless experience across web and mobile platforms. Overall, this architecture emphasizes security, privacy, and user empowerment, creating a robust decentralized communication tool.

3.3.1 System Architecture

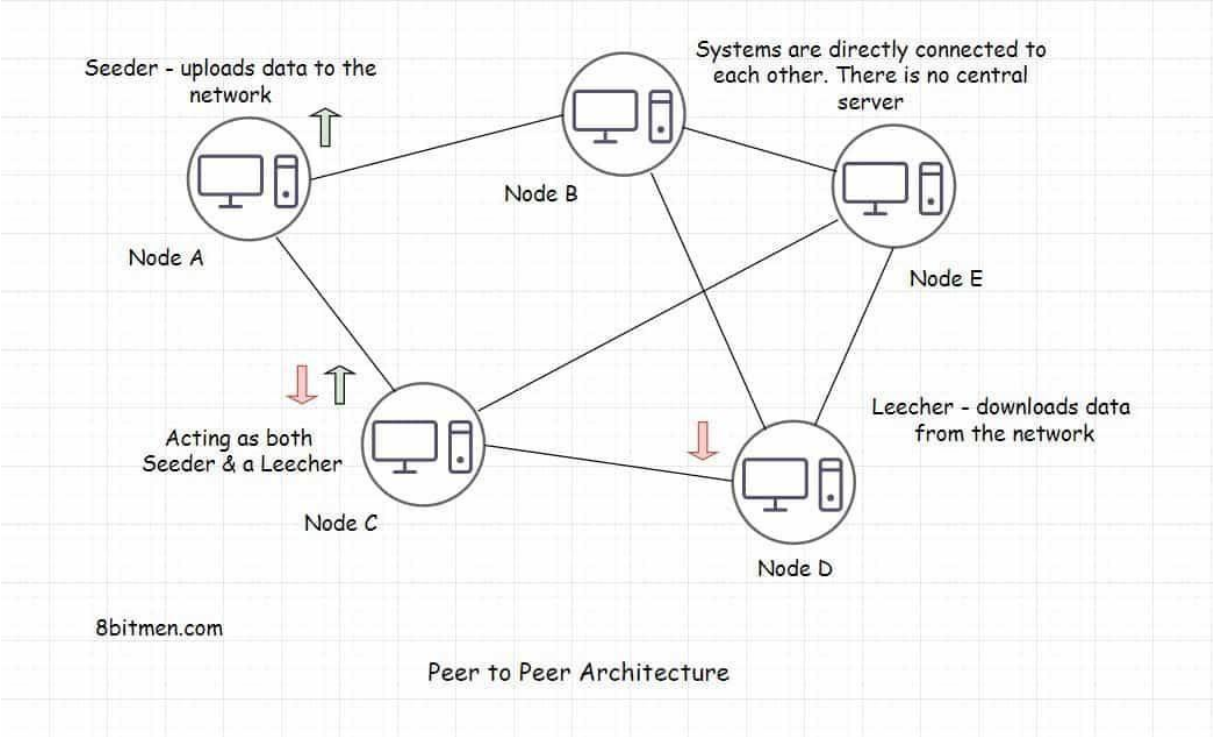

Figure. 3.1 – System Architecture

The system architecture of a decentralized chat application using blockchain technology is designed to ensure secure, efficient, and user-controlled communication. At the top level, the architecture comprises three main layers: the client layer, the blockchain layer, and the storage layer. The **client layer** consists of web and mobile applications that provide intuitive user interfaces for sending and receiving messages, managing contacts, and creating groups. The **blockchain layer** is built on a platform like Ethereum, where smart contracts handle user registration, identity verification, and message integrity. Each user generates a unique cryptographic key pair, with the public key stored on the blockchain, facilitating decentralized identity management. For messaging, a peer-to-peer protocol, such as Whisper or Matrix, enables real-time communication by allowing direct message transmission between users without relying on centralized servers. The **storage layer** utilizes decentralized solutions like IPFS to store shared files and data securely, ensuring that users retain control over their information. Together, these components create a cohesive architecture that prioritizes security, privacy, and scalability, enabling a robust decentralized chat experience.

3.3.2 Details of Modules

For a decentralized chat application using blockchain, with a focus on security and privacy, the modules are:

- User Management Module
- Messaging Module
- Blockchain Module

1. User Registering

Manages user-related operations such as user registration, login, profile management, and authentication.

Functions:

- User Registration: Allows users to sign up with a unique ID.
- User Authentication: Secure login using public/private keys.
- Profile Management: Update and manage user details.
- Key Generation: Each user is assigned a public and private key for encryption.

2. Applying for job

Handles the core functionality of sending and receiving messages. Already applied for that job.

Functions:

• Message Encryption: Encrypts messages using the sender’s private key and the receiver’s public key.

• Message Decryption: Decrypts messages using the receiver's private key.

• P2P Communication: Direct peer-to-peer messaging without a central server.

• Message History: Stores the message history on the blockchain or in decentralized storage.

3. Blockchain Module

Manages the integration of blockchain for decentralized communication.

Functions:

- **Block Creation**: Creates new blocks for storing message data.
- **Transaction Handling**: Handles the transactions of sending/receiving encrypted messages.
- **Consensus Mechanism**: Validates the transactions using consensus algorithms (e.g., PoW, PoS).
- **Data Immutability**: Ensures messages once sent cannot be altered.

3.4 Data Model and Description

Data Model describes the relationship and association among data which includes Entity Relationship Model.

3.4.1 Entity Relationship Model

The Entity-Relationship (ER) diagram of a decentralized chat application represents the key components and relationships required for a secure, user-centric communication platform. The diagram outlines the structure of the system, defining how different entities interact within the application.

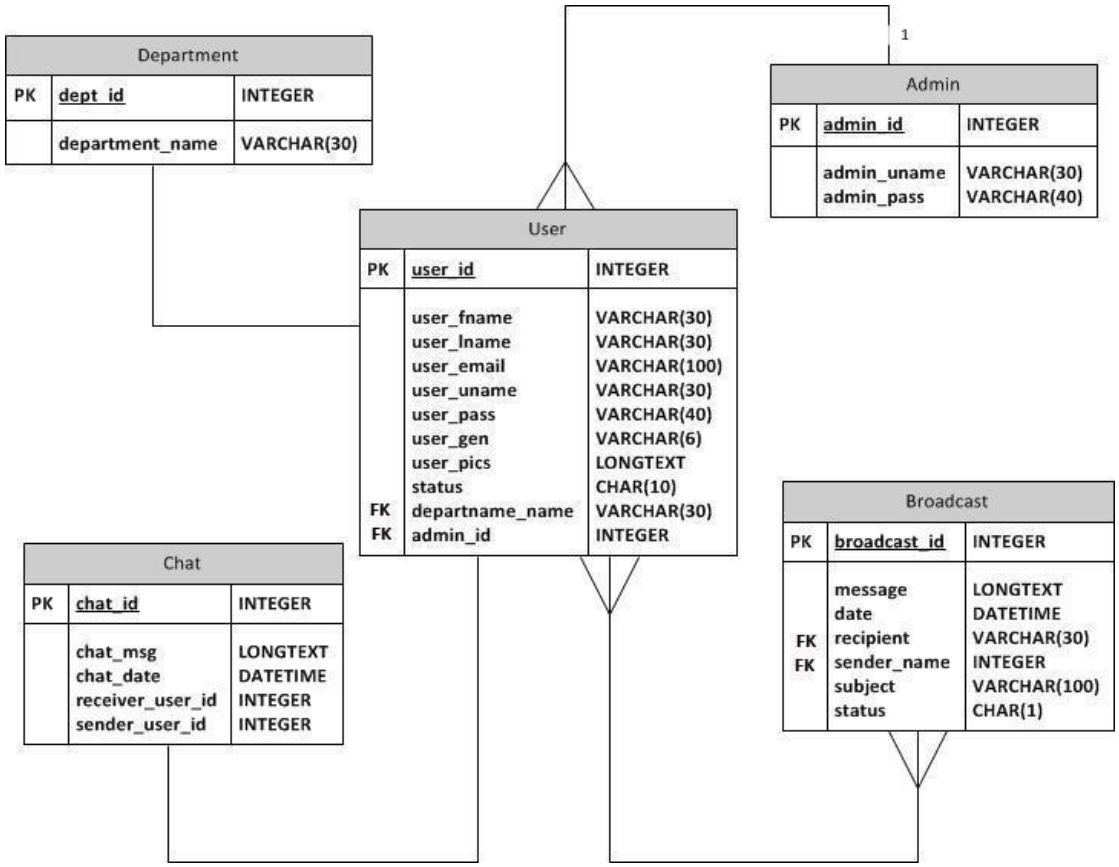

Figure 3.4 - Entity Relationship Diagram

3.5 Fundamental Model

Fundamental model of the project gives an overall idea about the project. How the entities are related to each other, what are the attributes of the entities, how the data flows between the entities are shown by the fundamental model

3.5.1 Data Flow Model

Data Flow Diagram (DFD) shows graphical representation of the flow of data through an information system, modeling its process aspects. It includes data inputs and outputs, data stores, and the various sub processes the data moves through. DFDs are built using standardized symbols and notation to describe various entities and their relationships.

DFD LEVEL 0

The Level 0 Data Flow Diagram (DFD) illustrates the core components of an online chat application. At the center of the diagram is the "Online Chat Application," representing the main system. This central process is connected to six primary functions: Chat History Management, Chat Profile Management, System User Management, Login Management, Chat User Management, and Chat Management. These functions interact with the central system to handle different aspects of the application's operation, such as managing user data, storing chat histories, handling user login processes, and organizing chat-related activities. This diagram provides a high-level overview of the data flow within the system, showing how each component contributes to the overall functionality of the chat application.

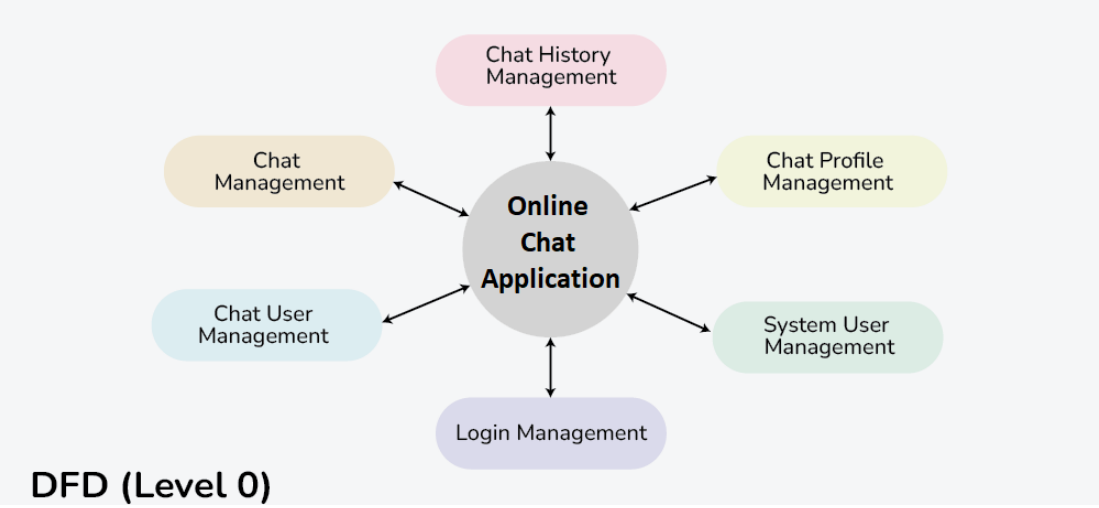

Figure 3.5 – DFD Level 0

DFD Level 1

The diagram represents a data flow for a system involving user authentication and encryption processes. Initially, the user interacts with the system by either creating a new account or signing in using existing login details. Upon signing in, the login details are processed, and if valid, the user receives confirmation. If the user creates a new account, their details are stored, and they receive feedback for successful registration. Once logged in, the user accesses the encryption system, where they can choose between different encryption algorithms, such as AES (Advanced Encryption Standard) or 3DES (Triple Data Encryption Standard). The chosen algorithm is then applied to text or file data for encryption or decryption. The diagram illustrates

how the system manages user authentication, encryption, and decryption tasks, ensuring secure data handling through these processes.

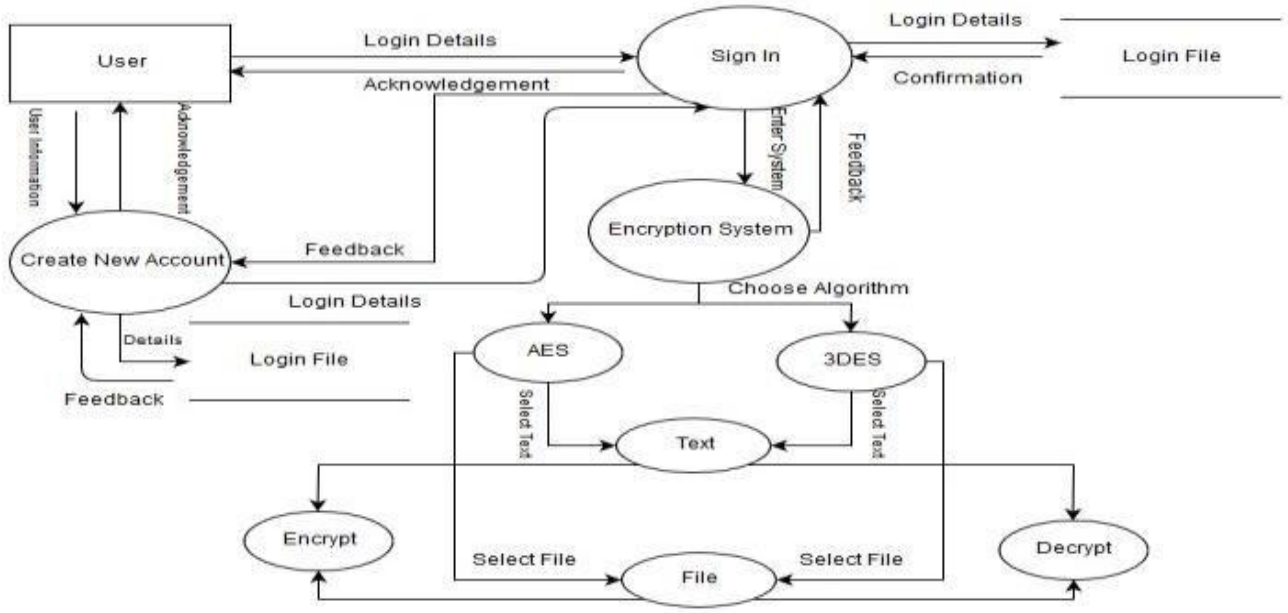

Figure 3.6 – DFD Level 1

3.6 UML (Unified Modelling Language) Diagram

language in the field of software engineering that is intended to provide a standard way to visualize the design of a system. We have prepared and designed the UML diagrams of – Use Case, Activity, Component, Deployment and Sequence Diagrams.

3.6.1 Use Case Diagram

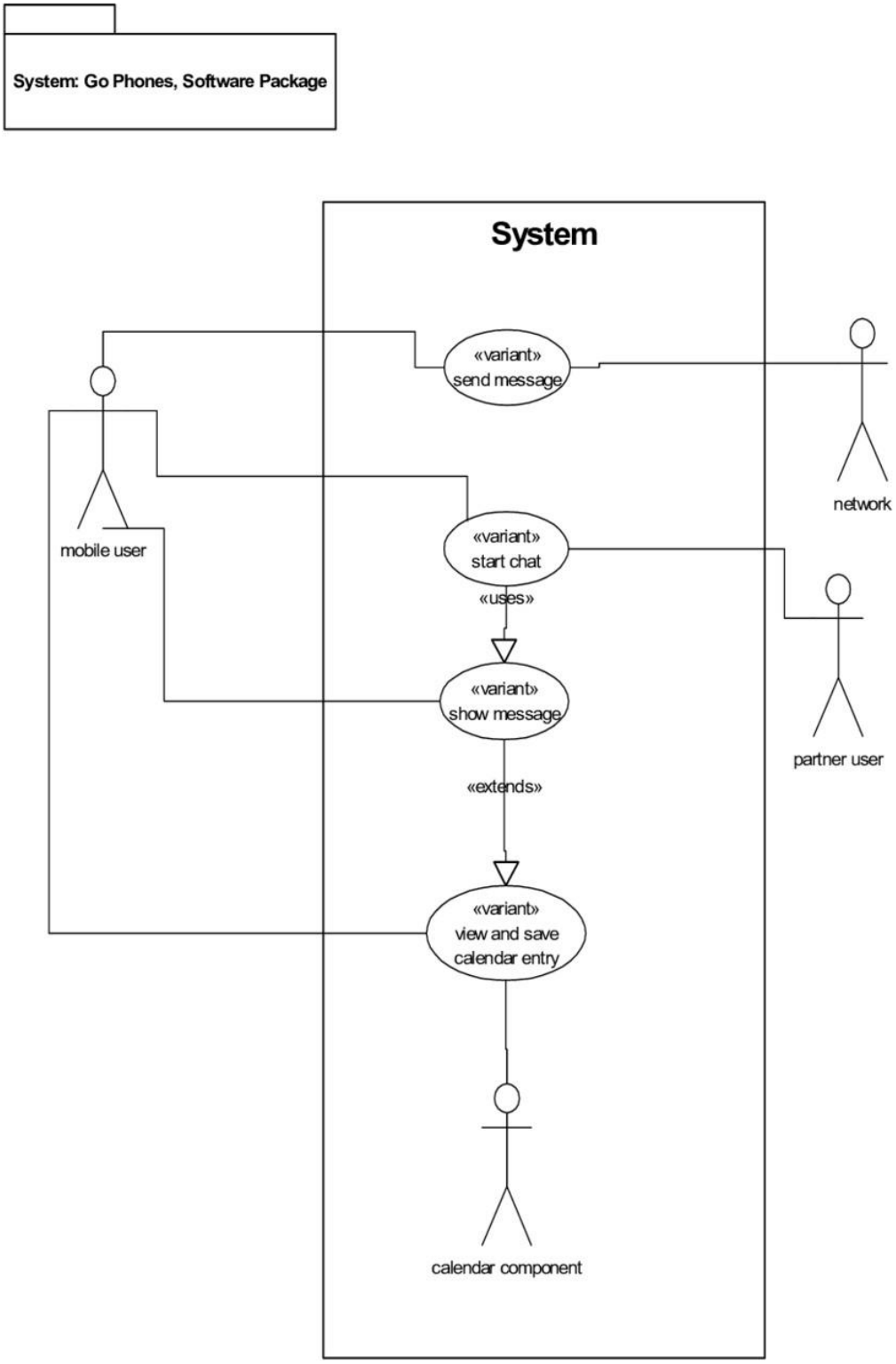

Figure 3.7 – Use Case Diagram

3.6.2 Activity Diagram

In figure 3.8, the Activity Diagram for HireVision.AI outlines the flow of activities during the The figure 3.10 represents the flow of communication in an NDN-based chat system involving a Receiver, NDN Hub, and Sender. The Receiver logs in, synchronizes with the network, and expresses interest in chat data. The NDN Hub processes the interest, either forwarding it to the Sender or sending a NACK if no matching data is found. The Sender responds with synchronization or chat data, which the NDN Hub forwards to the Receiver, completing the data exchange loop.

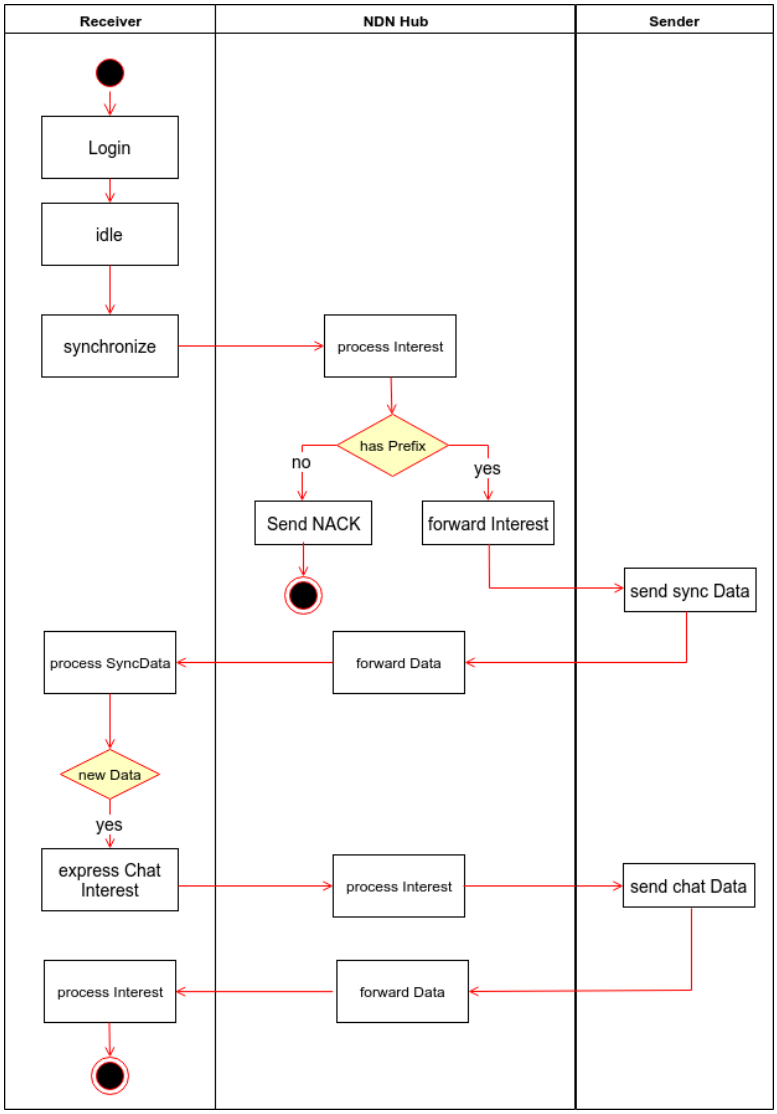

Figure 3.8 – Activity Diagram

3.6.3 Sequence Diagram

This figure 3.11 illustrates the sequence of operations in sending a message between two subscribers, A and B. The A-subscriber begins by entering the first digit of the recipient's information, which is processed by the A Handler, and the input signal is acknowledged. After receiving the rest of the digits, the A Handler

notifies the B Handler about the B-subscriber. If B is available, the system connects both subscribers through the network. A notification tone is set for the A-subscriber, while a message signal is activated for the B-subscriber, indicating the incoming message.

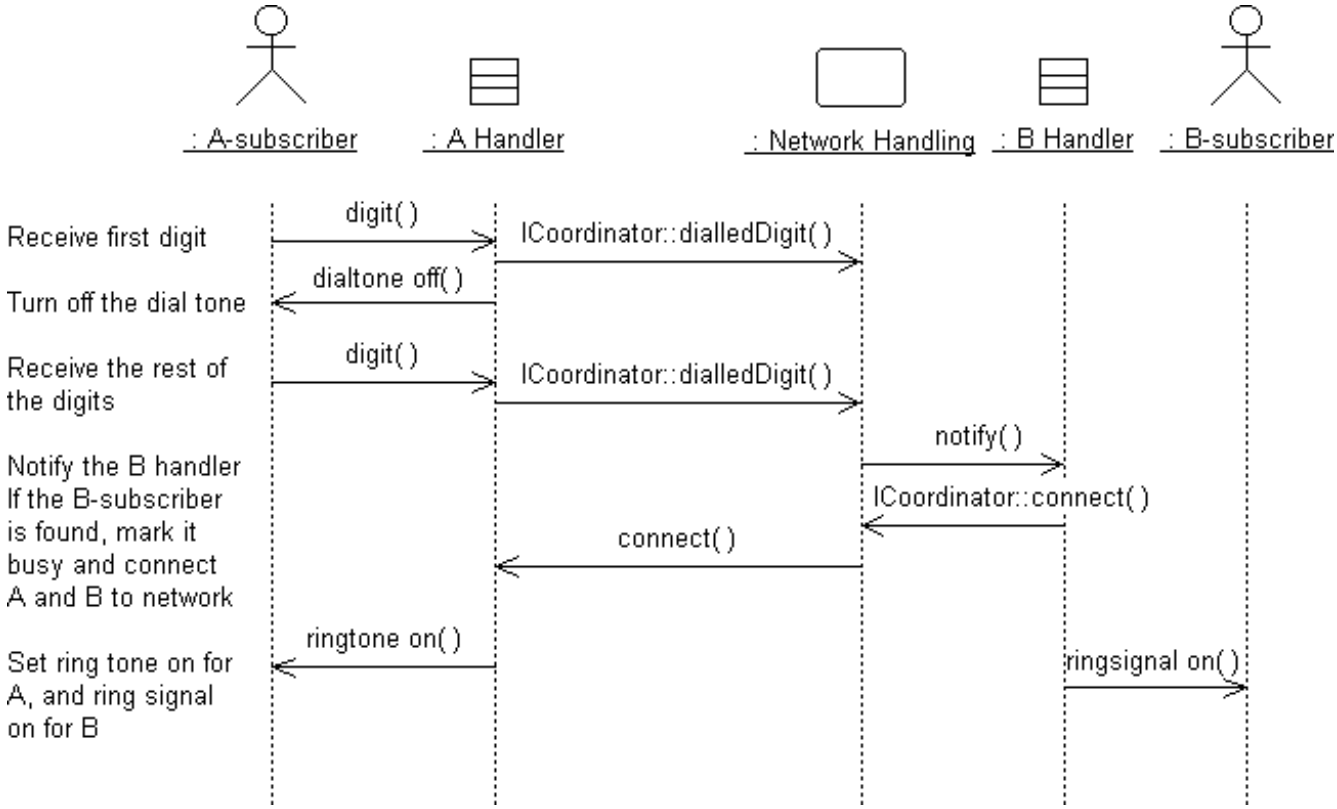

Figure 3.9 – Sequence Diagram

3.6.4 Component Diagram

The Figure 3.10 illustrates the structure of a chat application system, showing how different components are interconnected. At the center is the Chat Application System, which interacts with multiple features such as chat, chat profiles, chat history, group chat, and notifications. Each of these features accesses data through a dedicated data access layer, allowing them to communicate with the database.

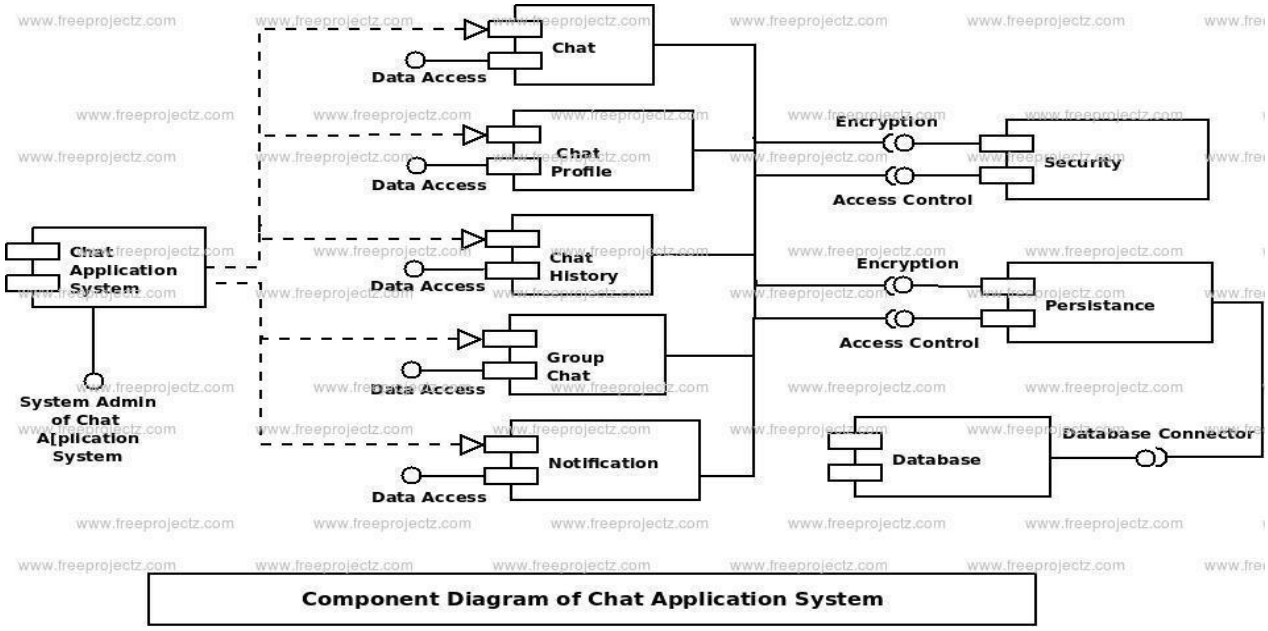

Figure 3.10- Component Diagram

3.6.5 Deployment Diagram

The diagram represents a system involving several key components. It features an Ethereum Node that provides the environment for running an Ethereum Virtual Machine (EVM), which is responsible for executing smart contracts. Alongside this, an IPFS Node is shown, which is linked to a Document Store component, likely handling decentralized document storage using the Interplanetary File System (IPFS).

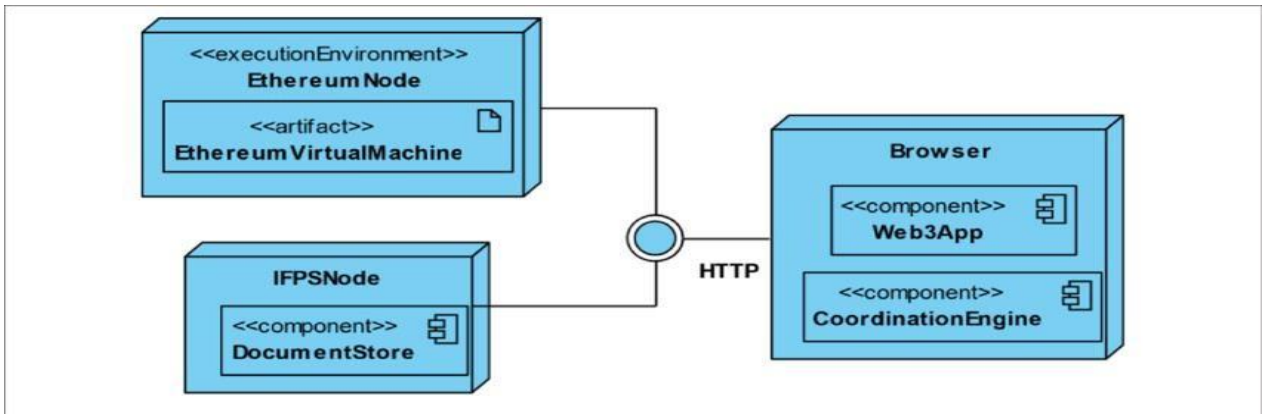

Figure 3.11 – Deployment Diagram

3.7 Methodology

For a decentralized chatting application, the methodology involves leveraging distributed network architectures to ensure secure, private, and reliable communication without relying on a central server.

- **Peer-to-Peer (P2P) Architecture:** The backbone of the decentralized chatting application will be a P2P network. Each user's device will function as both a client and a server, allowing direct communication between peers without a central authority. This reduces the risk of single points of failure and eliminates the need for centralized data management.

- **End-to-End Encryption:** To ensure that conversations remain private, the application will use end-to-end encryption techniques. Each message will be encrypted using the sender's private key and can only be decrypted by the recipient using their public key. This approach guarantees that even if the data is intercepted, it cannot be read by anyone other than the intended recipient.

- **Blockchain or Distributed Ledger Integration:** A blockchain or distributed ledger will be employed to manage user identities, message verification, and transaction logs. This technology will provide a secure and immutable record of all activities within the network, ensuring transparency and accountability while preventing unauthorized access.

- **Consensus Mechanisms:** To validate transactions and secure the integrity of messages across the network, consensus algorithms such as Proof of Work (PoW), Proof of Stake (PoS), or Byzantine Fault Tolerance (BFT) may be utilized. These mechanisms will help in maintaining the consistency and reliability of the distributed database that manages chat history and user interactions.

- **Decentralized Identity Management:** The application will utilize decentralized identifiers (DIDs) to authenticate users without relying on a centralized authority. DIDs provide users with control over their own digital identities, enabling secure authentication while protecting their privacy.

- **Data Storage Solutions:** Chat history and other user data will be stored using decentralized storage solutions like Inter Planetary File System (IPFS) or similar platforms. By distributing data across multiple nodes in the network, these systems prevent data loss and ensure data remains accessible even if some nodes go offline.

- **Onion Routing for Privacy:** To enhance user privacy and anonymity, the application will incorporate techniques like onion routing (similar to the Tor network). This method will ensure that the user's messages are routed through multiple layers of encryption, making it difficult for anyone to trace the origin or destination of the messages.

- **Scalability and Latency Optimization:** A decentralized chatting application needs to handle a large number of concurrent users efficiently. The methodology will focus on optimizing latency and

improving scalability by using lightweight protocols like WebRTC for real-time communication, which supports low-latency data transmission between peers.

- **Fault Tolerance and Redundancy:** The system will be designed to tolerate faults and maintain service availability, even in the event of node failures or network issues. Techniques like data replication and distributed consensus will be employed to ensure that messages are delivered reliably and that the application remains operational.

- **Interoperability:** To promote seamless communication across different decentralized networks, the application will support interoperability standards. This will allow it to interact with other decentralized messaging systems or platforms, enabling broader connectivity and data exchange.

- **Open-Source Protocols:** The application will be built using open-source protocols to ensure transparency, security, and continuous improvement. Community contributions and peer reviews will be encouraged to identify and fix vulnerabilities and enhance the overall functionality of the application.

- **User Experience (UX) Design:** Even with the focus on security and decentralization, the application's user interface should be intuitive and user-friendly. The goal is to make the transition from centralized chat apps to decentralized ones as seamless as possible for end-users, minimizing the learning curve.

Chapter 4

Result and Discussion

This chapter includes the snapshots of the actual outputs that were seen by the user and this chapter also contains the results of the proposed system.

4.1 Proposed System Result

The proposed decentralized chatting application leverages blockchain technology to provide secure and transparent communication for users. The system will have two primary types of users: Chat Users and Administrators.

Chat Users can create accounts, send messages, and participate in group chats without the fear of data breaches or unauthorized access to their information. Each message sent will be stored as a transaction on the blockchain, ensuring that the conversation history is immutable and verifiable. Users will also have the option to set privacy levels for their chats, allowing for confidential conversations.

Administrators will oversee the platform, managing user accounts and monitoring interactions to ensure compliance with community guidelines. They can also implement features such as reporting and moderation of inappropriate content.

The application will utilize smart contracts to facilitate various functionalities, such as automatic message encryption and user authentication, providing an added layer of security. The user interface will include a home screen that displays ongoing chats, recent messages, and user settings, as depicted in the accompanying diagram.

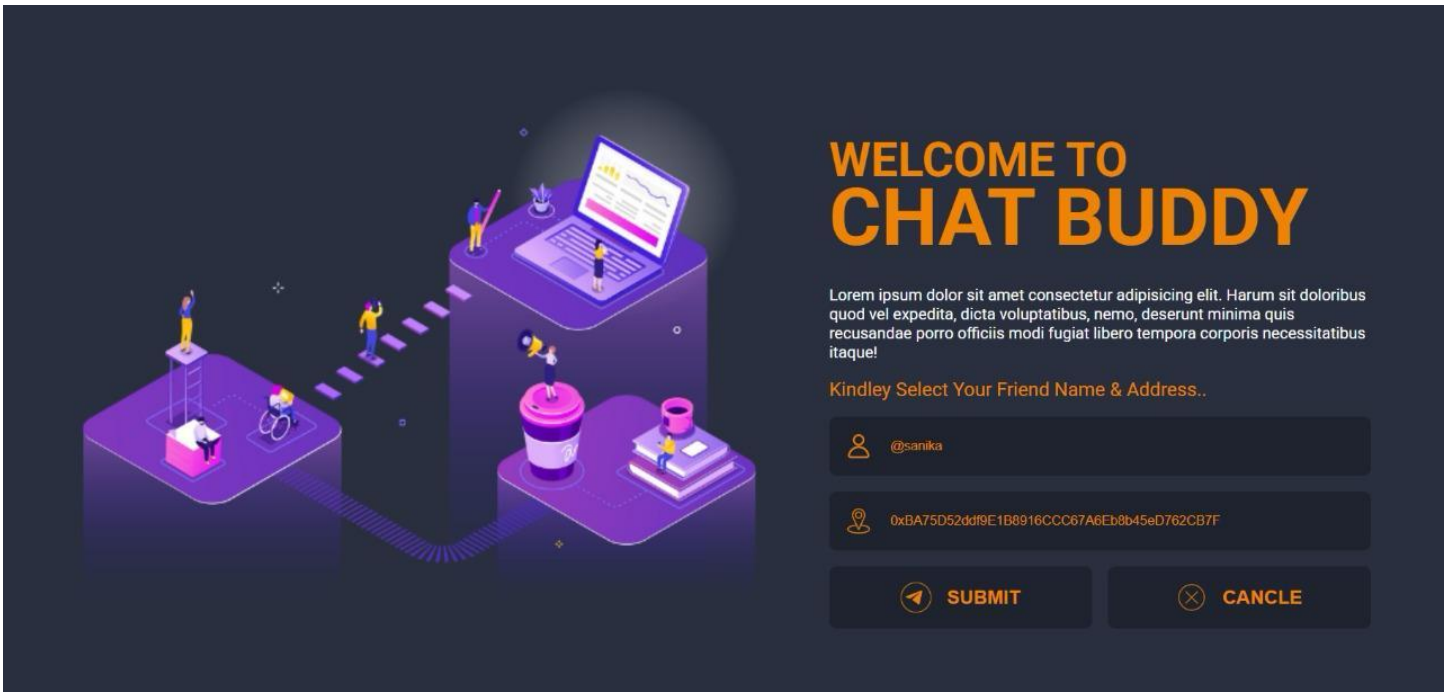

Figure 4.1 – Home Page

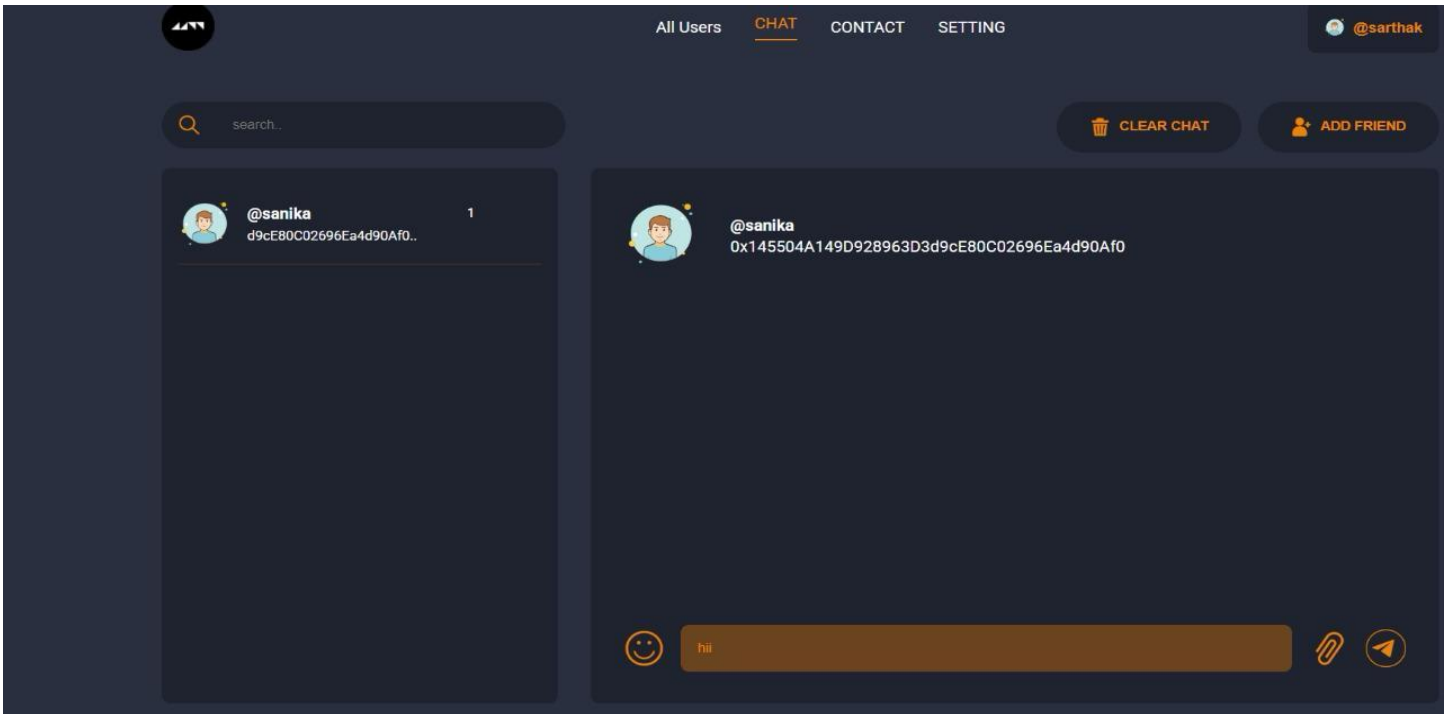

Figure 4.2 – Chat Page

4.2 Proposed system versus existing system

The proposed decentralized chatting application enhances user interaction by allowing not only messaging but also the transfer of photos and files. This addresses a common limitation of existing systems, which typically restrict interactions to text messages only. By leveraging blockchain technology, the proposed system ensures security, privacy, and data ownership, creating a more robust communication platform that empowers users while providing a versatile and feature-rich experience.

Table 4.1 – Comparison between existing and proposed system.

Feature	**Existing System**	**Proposed System**
Architecture	Centralized servers manage all user data and messages.	Decentralized architecture using blockchain for data storage and management.
Data Security	Vulnerable to data breaches and unauthorized access; user data is stored centrally.	Enhanced security with immutable records on the blockchain, reducing the risk of unauthorized access.
User Privacy	Users' messages and data can be accessed by the service provider.	Users maintain control over their messages, with encrypted communication and privacy settings.

User Authentication	Standard username/password authentication, susceptible to hacking.	Utilizes public and private keys for secure user authentication, reducing identity theft risk.
Message Integrity	Messages can be modified or deleted by service providers, leading to potential misinformation.	Messages are stored as transactions on the blockchain, ensuring integrity and verifiability.
Ownership of Data	Users do not own their data; it is stored and controlled by the service provider.	Users retain ownership of their data and conversations, as they are stored on the blockchain.

CONCLUSION

The development of a decentralized chatting application powered by blockchain technology offers a transformative approach to digital communication. By removing central control, this application empowers users to communicate securely and privately, significantly reducing the risk of data breaches and unauthorized access.

Utilizing blockchain's key features—such as transparency and immutability—ensures that every message is securely stored and tamper-proof. The implementation of smart contracts can further automate processes, enhancing security and user authentication without sacrificing efficiency.

Moreover, the focus on end-to-end encryption guarantees that messages are only accessible to intended recipients, fostering a safe environment for sensitive conversations. By enabling user-controlled data ownership, this application addresses growing concerns regarding privacy and data exploitation, allowing individuals to retain authority over their information.

As digital communication evolves, the demand for secure and reliable platforms is paramount. A decentralized chatting app meets this demand by offering a resilient, user-centric solution that prioritizes privacy and security. This innovation not only revolutionizes how individuals interact online but also sets a precedent for future applications in various sectors, from social networking to professional communication.

Appendix

- **PHP (Hypertext Preprocessor)**: A widely-used open-source scripting language designed for web development but also used as a general-purpose programming language. PHP is especially suited for server-side scripting, allowing developers to create dynamic web pages.

- **SQL (Structured Query Language)**: A standard programming language used for managing and manipulating relational databases. SQL allows users to create, read, update, and delete (CRUD) data stored in database systems.

- **XML (eXtensible Markup Language)**: A markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. XML is used to store and transport data, providing a way to describe the data structure.

- **E-R (Entity Relationship)**: A data modelling technique used to describe the data structure and relationships between different entities in a database. E-R diagrams visually represent the entities, attributes, and relationships, aiding in database design.

- **DFD (Data Flow Diagram)**: A graphical representation of the flow of data within a system. DFDs illustrate how data moves between processes, data stores, and external entities, helping to understand system requirements and design.

- **API (Application Programming Interface)**: A set of rules and protocols for building and interacting with software applications. APIs allow different software systems to communicate with each other, enabling integration and functionality across various platforms.

References

- https://www.academia.edu/122586801/A_STUDY_ON_BLOCKCHAIN_AND_CRYPTOGRAPHY
- https://www.researchgate.net/publication/327711685_Decentralized_Applications_The_Blockchain-Empowered_Software_System
- https://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf
- https://www.semanticscholar.org/paper/Decentralized-Applications%3A-The-Software-System-Cai-_Wang/acea0aada3b7c58b3711507f0935cfb4606eab72
- https://repository.iiitd.edu.in/xmlui/bitstream/handle/123456789/656/Agam%20Singh%20Bajaj_2014006%2C%20Lakshit%20Tyagi_2014058%2C
- %20Paarth%20Arora_2014150.pdf?sequence=1&isAllowed=y
- https://link.springer.com/chapter/10.1007/978-3-031-59100-6_5
- https://ieeexplore.ieee.org/document/7956725
- https://dl.acm.org/doi/10.1145/3132747

- https://www.amazon.com/Blockchain-Revolution-Technology-Changing-Business/dp/1101980133
- https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/eyal
- https://bitcointalk.org/index.php?topic=88208.0
- https://ieeexplore.ieee.org/document/6547123
- https://bitcoin.org/bitcoin.pdf