



# DecentraLend: A Blockchain-based Monetization for Decentralized Lending System

Mostofa Kamal Rasel<sup>1</sup>, Mahamudul Hasan<sup>1</sup>, Mohammad Rifat Ahmmad Rashid<sup>1</sup>, Md Hasanul Ferdous<sup>1</sup>, Mohammad Manzurul Islam<sup>1</sup>, Md Sawkat Ali<sup>1</sup>, Maheen Islam<sup>1</sup> and Md Ariful Islam<sup>1</sup>

<sup>1</sup>Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

Received 9 Feb. 2024, Revised 27 Mar. 2024, Accepted 30 Mar. 2024, Published 1 May. 2024

**Abstract:** Borrowing or lending goods or tools is commonly based on the centralized distribution in which the transaction of items generally occurs from an owner to a borrower. This centralized system may reduce the reachability of an item to potential borrowers, hinder the opportunity for the borrower to choose better alternatives, and increase the cost of distribution for an owner. In this paper, we proposed a novel decentralized virtual lending system based on blockchain to overcome the above limitations of a centralized lending system that allows any owner to monetize their items in possession. Our proposed system enables any possessor of an item to become a distributor by creating a chain of possessors from the owner through every borrower. On the other hand, the system incorporated a recommendation mechanism to suggest the best alternatives for an item based on a cost model that considers the borrowing fee, security deposit, gas, and distribution and return costs. To reduce the cost of gas and overall running time, we adopt a hybrid implementation of blockchain and RDBMS for storing the data and transactional information. Experimental results show that our hybrid approach can minimize gas usage significantly and reduce the running time by two orders of magnitude compared to a blockchain-only approach.

**Keywords:** Decentralized app, blockchain app, decentralized monetization, decentralized recommendation.

## 1. INTRODUCTION

Lending or rental systems for physical goods or tools, such as vehicles [1], industrial equipment [2], and artworks [3] have existed for a long time. Multi-location-based rental systems [4] also exist to ease distribution and reduce distribution costs. However, these systems are centralized and operated by organizations, where an item is transferred from an owner to a borrower or returned from a borrower to the owner. As a result, a borrower finds a lower chance of choosing the best item among the alternatives from other lenders. Besides, the lender loses opportunities to reach their items to the vast potential borrowers. On the other hand, decentralized lending systems or content-sharing concepts have long been utilized among social network users or electronic communications for trading and monetizing digital content [5] as modern communication technologies have been introduced [6], [7]. The aforementioned decentralized lending system usually disperses a digital item from any owner instead of a central hub or warehouse. However, a lending system for sharing physical goods and tools cannot adopt the concept of decentralized digital content sharing or lending. Decentralizing the physical item distribution not only considers distribution but also focuses on relocating and securing the physical item.

The decentralized physical item-sharing concept is not

new and was implemented as a book-sharing social system called Bookcrossing [8]. In this system, a reader shares a book with another reader after finishing reading the book by leaving that book in public places, such as benches in a park. Any surrounding reader who liked or had a common interest in the shared book could take the book to read. After reading the book, the reader could share it the same way with others. Thus, every reader becomes a sharer. We note that this kind of book sharing helped free circulation of knowledge through books. However, the Bookcrossing system works in the physical world and entirely depends on people's good faith. The availability of book information is also limited unless a reader physically visits a book's location. Moreover, tracing the previous readers and the current holder of a book is almost impossible in Bookcrossing.

Nowadays, innovative and emerging blockchain technology [9], [10] offers to store and track digital content safely and securely. Blockchain ensures secured transactions by encrypting all transactional records or blocks and establishes tracing of any transaction by creating links from one block to another [11]. Since blockchain technology can perform secured financial transactions, provide transparency on transactional information, and trace any transaction, the technology can be exploited for developing decentralized lending services.



Blockchain is being successfully adopted for developing secured and traceable systems for various purposes, such as, for artworks [3], supply chain management [12], and land registry [13], [14], [15]. However, the above-mentioned systems are centralized and cannot be adopted to decentralize the lending or rental services. Based on our careful observation of the concept of Bookcrossing and the properties of blockchain technology, we propose a novel decentralized lending system (DecentraLend) that utilizes the storing, tracing, and securing contents mechanism of blockchain technology that implements the book-sharing concept of Bookcrossing in the digital world to share physical items. In addition to a decentralized physical item-sharing system, our proposed system implements a mechanism to monetize an item borrowed by a system user in the form of a borrowing fee and security money. We note that the proposed decentralized lending system allows any user, such as individuals, groups, or an organization, to become a lender. For example, our proposed blockchain-based decentralized lending system can be used by an individual user, a traditional library, or a bookseller to lend books easily. The users of such decentralized book lending systems benefit financially without investing in the delivery system or fearing losing investment in books. Moreover, typical readers can also find the system beneficial as they may not need to visit libraries physically. Instead, they can collect it from the nearest user who previously borrowed the book. We summarize our contribution in this paper as follows -

- A blockchain-based decentralized online lending system enables any user to become financially beneficial by lending their goods or tools while avoiding loss of investment by tracing borrowers, upholding security money, and lending fees.
- A decentralized system enables an individual to pose as a lending company by lending their items.
- A recommender mechanism for the borrower that recommends items that are beneficial for a borrower, however, prevents an item from being listed in the recommended list if it is not beneficial for the lender (i.e. distance between the borrower and the owner may significantly increase the return cost of an item).
- An extensive experience to show the effectiveness of the proposed system.

The remaining paper is organized as follows. We discuss the background and some preliminaries in Section 3 before discussing the overall system. Section 4 elaborately describes the system and includes an architectural overview, methodology, and the considerations that are taken into account for implementing the proposed system. We discuss our findings from the experiments in Section 5. We describe the objectives and methodologies of the related works and discuss the relationship with our proposed system in Section

2. Finally, we conclude the proposed system with future directions in Section 6.

## 2. RELATED WORKS

Blockchain technology has opened several avenues for innovative services and applications to decentralize lending systems. In such systems, blockchain's transparent and secure ledger mechanism enables peer-to-peer lending without the requirement of traditional financial intermediaries. Smart contracts [16], one of the most crucial components of blockchain technology, facilitate the automatic execution of lending agreements, reducing the risk of defaults, and simplifying the borrowing process. Decentralized lending platforms empower individuals to lend and borrow funds directly from one another, thus increasing financial inclusivity. Additionally, blockchain's immutable record-keeping ensures the integrity of transaction history, which is vital for maintaining trust in such lending systems. Blockchain has paved the way for a more inclusive, efficient, and secure decentralized lending landscape.

Several recent works have been proposed to decentralize lending systems and decentralized finance. Decentralized financial, often termed DeFi, refers to financial systems that run on blockchain technologies without the help of any traditional intermediaries, such as banks and other financial organizations. DeFi systems mostly operate on public blockchains, thereby expanding financial services globally. Schär [17] examined the potential security vulnerability and usability within the DeFi ecosystem and proposed a multi-layered framework for analyzing its architecture, covering various building blocks. The analysis suggests that while DeFi remains a niche market with potential risks, it offers notable advantages in efficiency, transparency, accessibility, and composability, potentially upholding a transparent robust financial infrastructure. Ozili [18] highlighted the advantages of DeFi, including enhanced inclusivity, innovation, cost efficiency, resistance to censorship, and transaction immutability. In addition, the author underscored several risks associated with DeFi, such as smart contract execution, legal liabilities, data security, and illicit activities, while noting a scarcity of empirical research and comprehensive analysis of challenges. The efficiency of a recommendation mechanism on a decentralized system depends on multiple aspects, such as the gas requirements for the transaction, transportation cost of an item, security deposit, and borrowing fee. The researches [19], [20], [21] discussed optimizing the cost in the environments affected by the multi-dimensional aspects.

Permission-less blockchains create an uncensorable information ecosystem, enabling private interactions. Smart contracts within decentralized finance (DeFi) facilitate automated, identity-agnostic financial services. Saengchote [22] discussed Compound, a leading DeFi lending protocol, by exploring user behavior, loan durations, and systemic risk factors, that revealed the challenges in applying traditional risk management to the DeFi environment. Joseph and



Karunan [23] proposed a novel decentralized transaction settlement system for the banking sector built on the Ethereum platform. This system supports different financial services using distributed ledger technology. Kaplan et al. [24] discussed a blockchain-based decentralized lending protocol to analyze returns between S&P 500 and DeFi assets. However, the authors did not propose any model for lending goods to users.

In the domain of loan management, blockchain-powered lending management systems offer enhanced capabilities in inventory record maintenance, material identification, and item storage compared to the conventional loan management system. Verma [25] outlined an architectural framework for a library management system that utilizes blockchain technology. This system is designed to house user-specific book identifications and associated records within its database. The proposed system implemented Peer-to-Peer (P2P) instances to optimize network building and use the InterPlanetary File System (IPFS) to manage library books. Moreover, the author recommended a blockchain-based transaction management system similar to the International Federation of Library Associations and Institutions (IFLA). Also, the researcher intended to upgrade the library administration system and introduce blockchain-based user verification and record-keeping in all relevant institutes. Similar to the approach adopted in [25], we have proposed mechanisms to create any physical item lending and borrowing chain. Furthermore, our analysis also incorporates monetization, providing security money and lending fees to any item owner to provide financial security.

The mobile bookcrossing approach is widely known in this modern era of technology, where people tend to share books through mobile networks. One of the hindrances behind the adoption of bookcrossing is that the process is not transparent, which causes the loss of books. Another problem is that the current mobile bookcrossing system is challenging to manage while including the new book curriculum. To address these issues, textcolorblueZeng et al. [26] introduced a blockchain-based book-sharing system that can trace books efficiently on a campus. This system helps to make the book borrowing process transparent by showcasing the book borrowing history to the users. The proposed system, termed BookChain, circulates smart contracts to automate the book borrowing process with minimal human intercessions. To implement the blockchain, the authors created a web server, tokenizers, smart contracts, an authority management system, and consensus activities for managing the library and storing data in blocks. The authors also simulated the results among specific schools to evaluate the BookChain mechanism and efficiency. They have included blockchain-based transactions showcasing the average gas price in deploying contracts, adding books to the chain, and borrowing and returning books.

The traditional procedures of book borrowing systems are pretty lengthy and time-consuming. As a solution to these issues, Cabello et al. [27] introduced LibChain, a mechanism that makes the processes of book borrowing from libraries more efficient and reliable. The authors argue that the implementation of blockchain in LibChain will encourage the users to borrow books from each other, reserve the books without the involvement of the library authority, and provide loans directly from a library to users and from users to users. The authors intend to implement this project in the future through four work packages: (1) introducing requirements and overall architecture, (2) basic system implementation, (3) determining graphical user interfaces (GUI), and (4) improvising edge cases and reserving presentation. Adopting the idea of providing loans presented in this work [27], our proposed DecentraLend has incorporated security fees for a borrowed item to benefit item owners.

Liu [28] introduced a blockchain-based smart library management system that aims at reducing the tempering of the information stored and processed in the system. As blockchain is a consensus algorithm, the author argued that it would enrich the system's security and increase efficiency. In this paper, the author described the system architecture that includes the SDK (Software Development Kit) cluster as a middle-term and a group of blockchain nodes. The overall system is designed with six procedures: (1) donation application, (2) book renewal application, (3) book management, (4) borrower information management, (5) borrowing application management, and (6) notice management. According to the analysis presented in the work, the author determined that the application would help improve the overall library management system. In the future, the author has a plan to implement the proposed framework in a library and devise appropriate methods according to customers' requirements. In our paper, we have kept provisions of the borrowed items by a single user. If any of the items is transferred to any other user, the status of the item's availability will also change in our system.

Implementing blockchain-based solutions can also ease the process of user management and authorization process by librarians. Blockchain can help preserve digital information, track books borrowed by users, initiate inter-library loans and vouchers, verifications of credentials, and so on [29]. The demand for diverse knowledge and information is rapidly growing with the changing world's growth. However, traditional library management systems do not have the advantage of incorporating digital management systems into libraries. To this end, Cao and Yang [30] introduced a new model (Peer Book Search and Library, or PBSL) for a digital library management system that leverages P2P and blockchain. This PBSL is divided into three portions. Firstly, the materials and books would be saved in a digital system to enable access from anywhere in



the world. Secondly, DHT (Distributed Hash Table) would enable the search, distribution, and transmission of various digital resources. Finally, blockchain technology would be used for digital services, such as copyright protection, trade, propagation link maintenance, and calculating reading charges. The data security of any library management system is a matter of concern as it involves a person's sensitive information. To ensure strong privacy, and data security, and reduce the work tools in a library, Tawornittayakun and Leelasantitham [31] proposed a new library management system that has been developed with blockchain. In this work, the authors tested and compared the performance of the bibliography of the MARC 21 and AACR2R for both the Proof of Work (PoW) state and IPFS in blockchain.

Another application area of the overall lending system is the bike-sharing system. The traditional bike-sharing system lacks a proper mechanism for ensuring privacy, integrity, and confidentiality of user information. Guo et al. [32] proposed a framework based on the operation mode of the C2C (Consumer-to-consumer) business model, which would help ensure a transparent payment procedure and store the information through smart contracts. In C2C operation, the transaction procedures are divided into five methods, where the damage control of bike sharing and payment procedures are acknowledged together through one blockchain network. The authors planned to implement this framework in the future by simulating an actual work field. However, in addition to the aforementioned proposed bike-sharing system, our DecentraLend system has incorporated the security deposit for the lenders and provided a recommendation system for the borrowers to avail the best alternatives.

### 3. PRELIMINARIES

Our proposed blockchain-based decentralized lending system (DecentraLend) aims to provide complete decentralization of lending processes for monetization by increasing the re-usability of personal or organizational goods and tools. In this section, we discuss the notions and notations used in this paper for describing the proposed system. We also describe here the background factors considered for the proposed system.

Our proposed decentralized system considers both an organization and an individual as a user. The lending items can be any type of physical tools, goods, machinery, or equipment, and hereafter we call them items. The proposed system has two main actors or stakeholders: *lenders* and *borrowers*. We define the lenders and borrowers in our proposed DecentraLend as follows.

*Definition 1 (Lender):* Let  $u_i$  be a user of the DecentraLend. Let  $u_i$  own an item  $p_k$  and let  $u_i$  add information about  $p_k$  in the DecentraLend. If  $u_i$  makes  $p_k$  available to be borrowed by other users of the DecentraLend, then we define  $u_i$  as the lender of  $p_k$ .

*Definition 2 (Borrower):* Let  $u_j$  be a user of the De-

centraLend and let  $u_j$  request an item  $p_k$  through the DecentraLend. If  $u_j$  completes the payment for borrowing  $p_k$  successfully for  $p_k$ , then we define  $u_j$  as the borrower of  $p_k$ .

As a decentralized item lending system, DecentraLend considers several costs related to each transaction placed in the blockchain and to transfer of an item from one user to another. Moreover, there is a cost for enabling the monetization as well. We define the *borrow cost* to illustrate the overall cost for borrowing an item as follows:

*Definition 3 (Borrow Cost):* Let  $p_k$  be an item marked available. Let  $u_l$ ,  $u_c$ , and  $u_f$  be the users denoted for the lender, current borrower, and future borrower of  $p_k$ , respectively. Let  $S_{u_c, u_f}$  be the cost for sending  $p_k$  from  $u_c$  to  $u_f$  and let  $R_{u_l, u_f}$  be the cost for returning  $p_k$  from  $u_f$  to  $u_l$ . Let  $F_k$  be the borrowing fee for  $p_k$  and let  $T_g$  be the transaction cost in the blockchain. Then, the borrow cost  $C_k$  for  $p_k$  is given by the following equation -

$$C_k = S_{u_c, u_f} + F_k + R_{u_l, u_f} + T_g \quad (1)$$

Book sending cost  $S_{u_c, u_f}$  usually depends on the distance between  $u_c$  and  $u_f$ . The type of transportation, required time for transportation, and other situations may also affect the sending cost. Similarly, returning cost  $R_{u_l, u_f}$  also depends on the distance  $u_l$  and  $u_f$  as well as on the aforementioned criteria. However, we note that DecentraLend considers the direct distance between users as the monetary cost of sending an item for the sake of simplicity.

The security deposit for borrowing an item is always returned to a borrower. Therefore, we exclude the security deposit for evaluating the borrowing cost. We note that the cost  $R_{u_l, u_f}$  is paid by the lender  $u_l$ . Therefore, the real cost a borrower pays for borrowing an item comes from  $S_{u_c, u_f}$ ,  $F_k$ , and  $T_g$ . The cost  $R_{u_l, u_f}$  usually increases as the distance between the lender and the future borrower increases. Taking into consideration the cost  $R_{u_l, u_f}$  in evaluating the borrowing cost helps to indirectly discourage a future borrower who is far away from the lender. However, this consideration is beneficial for the lender since it avoids paying high costs for returning. On the other hand, we note that  $F_k$  must be higher than  $R_{u_l, u_f}$  to make the lending profitable. For the sake of simplicity, we overlook the reality that the lending still be profitable when  $(\sum_{k=1}^n F_k) > R_{u_l, u_f}$ , where  $n$  represents the number of times an item borrowed and  $n$  is sufficiently large enough.

### 4. DECENTRALIZED LENDING SYSTEM

Our proposed DecentraLend system focuses on decentralizing the lending of personal and organizational goods and tools. Moreover, the proposed system concentrates on monetizing from the lending and secures the transaction. DecentraLend also addresses the tracking of an item from the owner to any borrower through the history of lend-





ing. We describe the fundamental functional procedures to achieve the aforementioned objectives of the proposed DecentraLend system below.

- *Determining the identification* of a user to the DecentraLend (i.e. KYC). The address of a user can also be collected through this functionality.
- *Add an item* by a user as a lender. A user provides information about an item to DecentraLend to be borrowed by other users and monetized.
- *Borrow an item* by a user as a borrower. A user searches for an item in the DecentraLend by the name of that item and borrows it by selecting a source from a DecentraLend recommended list and paying the lending fee and security deposit. The source of an item can either be a lender or a borrower.
- *List of recommended sources* for a desired item is generated according to the Equation 1 by the DecentraLend. The best lenders in terms of cost of borrowing appear first in the recommended list.
- A lender or a borrower can *mark available an item* in possession so that other users in the DecentraLend system can borrow it.
- A lender or a borrower can *mark unavailable an item* in possession so that the possessed item is ignored by the DecentraLend while generating the list of recommended sources.
- A borrower can initiate a *return a request for an item* in possession to the owner of that item.
- A lender can *approve returning an item* which was requested from its borrower.

According to the aforementioned functional procedures, DecentraLend creates a chain of possessors<sup>1</sup> for each item. A chain starts from the action of adding an item by a lender in the blockchain. After that whenever an item is moved from one possessor to another, a financial transaction takes place through MetaMask, and the length of the chain increases by adding a new block. An item is moved from one possessor to another in case of the action of borrowing an item or approval of returning an item. On the other hand, the actions for marking available or unavailable an item and return requests only change the availability status of an item for borrowing. The information, i.e., status, for an item at the last record in the chain is only updated for these aforementioned actions. Given that our proposed DecentraLend system requires storing information about the goods and tools, to provide tracking and tracing of the borrowers of an item, and ensure a secured financial transaction, we endorse the best-known technologies for

implementing the above functionalities for achieving an efficient and cost-effective solution.

#### A. System Architecture

Based on the gathered knowledge and careful observation, we adopt blockchain to implement the functional procedures of the proposed DecentraLend since blockchain can store and trace information about any transaction. We build our proposed DecentraLend system upon the Ethereum platform [33], which is a popular decentralized open-source blockchain platform with an integrated programming language. Ethereum enables to deployment of smart contracts and decentralized apps with user-defined arbitrary rules for ownership, transaction formats, and state transition functions.

The smart contract is responsible for storing transactional data, managing ownership of products, and handling payment-related logic. It includes functions for creating transactions, recording ownership changes, calculating fees for the transaction, and more. The contract's data and events are stored on the Ethereum blockchain, ensuring transparency and immutability. However, storing all data and the functionalities led by the smart contract may cost a high amount of gas and execution time as the transactional cost. For example, the processing time of a single transaction in the blockchain network may be 80-2000 times slower than the MySQL database. On the other hand, the amount of data a blockchain network can afford in a single transaction is about 1/10 of MySQL [34]. Sanka et al. [35] suggest different scalable solutions to address the aforementioned limitations of blockchain, such as considering smart contracts to store the minimum amount of transactional data. We also observe that most decentralized applications generally use a centralized relational database to store data.

Based on the study of existing research and systems, we deploy a relational database (MySQL) in our proposed DecentraLend along with the blockchain to store and manage data. In our proposed system, we use a smart contract on the Ethereum blockchain to store and track only transactional data. On the other hand, we consider the MySQL database to store the off-chain data on users, items, and transaction history. Off-chaining strategies are required to mitigate the performance constraints and enormous charges associated with on-chain storage and computation [36]. An overview of the DecentraLend system is illustrated in Figure 1.

DecentraLend consists of multiple components working together to store user data, items, and transactional history in a MySQL database, to store and trace transactional data on the Ethereum blockchain, to manage transaction fees and payments for borrowing fee and security deposits through MetaMask, and a frontend to interact with the system through a python implemented client using web3.py. We discuss each system component and its role in DecentraLend as follows:

*Smart contract* stores transactional data, manages own-

<sup>1</sup>A possessor of an item can either be the lender(owner) or a borrower of that item

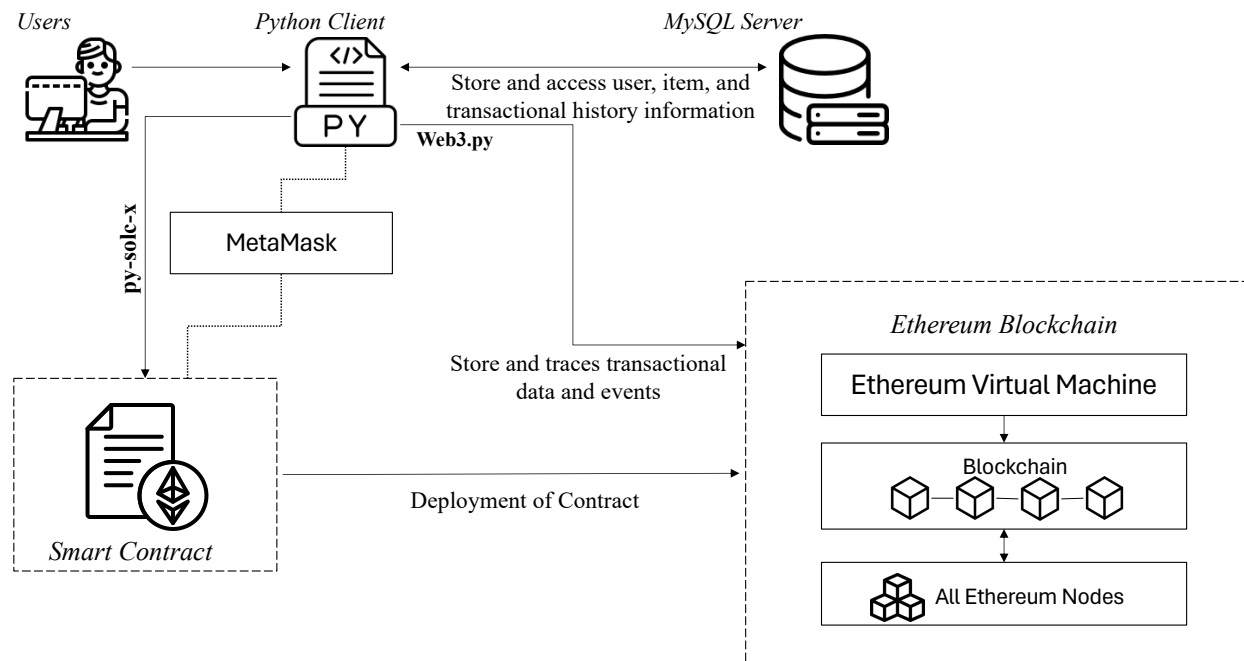


Figure 1. Overall architecture of the proposed DecentraLend.

ership and lending of items, and handles payments. It consists of functions for creating transactions to borrow or approve the return of an item, recording possession changes, and calculating fees for transactions.

*Ethereum blockchain* is a blockchain network platform where the smart contract of DecentraLend is deployed. Ethereum blockchain holds the transactional data and events of the smart contract that occur in DecentraLend to ensure transparency and immutability. On the other hand, the self-executing code of smart contracts for adding, borrowing, or returning an item runs on the Ethereum Virtual Machine (EVM). It also emits events to notify the frontend of important state changes, such as adding, borrowing, or approving the return of an item.

*MySQL database* stores the user's data and the items information that a user of DecentraLend adds to the system. It also keeps records of transaction history, the current status of an item, and other relevant information. The frontend python client of DecentraLend interacts with the MySQL database through *mysql-connector* to perform different query operations, such as insert, update, delete, and select. MySQL database server also returns the recommended list of possessors of a searched item. It also estimates the cost of collecting an item from different possessors.

*Frontend python client* acts as the user interface for interacting with the DecentraLend system. Users can perform various actions, such as add items, view their own or borrowed items, search for an item, select an item from the recommended sources for a searched item, borrow an

item or request to return an item, check transaction history, and mark or unmark an item's availability. The Frontend Python client interacts with the Ethereum blockchain using `web3.py`.

*MetaMask* is integrated to manage the user's Ethereum wallets and to interact with DecentraLend. Through the `web3.py`, DecentraLend allows its users to pay payments using their Ethereum accounts through MetaMask.

Our proposed architecture for DecentraLend ensures a seamless flow of data and operations between the frontend Python client, Ethereum blockchain, and MySQL database. Users can view and manage items, initiate transactions, and make payments using their Ethereum accounts, while all relevant data is securely stored and traced on the blockchain and in the database.

### B. Implementation

In our proposed DecentraLend system, a numeric ID is used as an identification of an item. On the other hand, we use a unique address, such as *Metamask wallet address* for the identification of a user. DecentraLend uses both identifications for identifying the owner, borrower, or possessor of an item. We use a simple data structure, such as `struct Item(itemID: uint, possessorAddress: address, prevBlockIndex: uint)`, to store a piece of item information in the blockchain. However, we store the identifications of both owner and possessor of an item in the MySQL database along with other required and optional information. We note that the owner and possessor of an item with different values indicate that the item is borrowed by the possessor.

TABLE I. Core functionalities of the DecentraLend System

Procedure Name	Type	Procedure Purpose
addItem()	Public	A user adds a new item
borrowItem()	Public, Payable	A user borrows an item
recommendations()	Public	A user searches for a desired item
markAvailable()	Public	Make available an item for lending
returnRequest()	Public	A borrower requests an owner to take back a borrowed item
returnApproval()	Public, Payable	An owner approves a request from a borrower to take back a lent item
myItems()	Public	Show the list of items owned by a user
borrowedItems()	Public	Show the list of items borrowed by a user

The core functionalities used to access (i.e. insert, update, and search) the data in a smart contract and MySQL database are listed in Table I. In the procedure of borrowItem(), a financial transaction occurs to pay the borrowing fee and security deposit. A financial transaction also occurs for paying back the security deposit to the last borrower in the procedure of returnApproval(). We exploit the smart contract to manage the transactions among users that take place for adding, lending, or returning an item in our proposed DecentraLend. The transactional data are stored and traced using the blockchain. On the other hand, information about an item and the transaction are also stored in the MySQL database for tracing the status of that item. Based on our strategies, DecentraLend accesses the blockchain for only the procedures addItem(), borrowItem(), and returnApproval().

The addItem() procedure is invoked to add a new item. This procedure has been illustrated in Algorithm 1. The algorithm takes information about an item as a parameter at line 1. Detail information for the item is stored in the MySQL database in line 2. MySQL provides a unique ID for the item. Then, a brief information including the id of the item is stored in a block of the blockchain from lines 4 to 7. Line 7 adds a new block in the blockchain and creates a new chain for the item. Line 7 also returns the block number in the blockchain. The item information in the database is then updated at line 8 with the block number as a block index and the user address as a possessor of that item. We note that adding an item in the system doesn't cause any financial transaction, however, a cost such as the gas fee is charged for adding a block in the blockchain.

The borrowItem() procedure is invoked by a user to borrow an available item. To borrow an item, a user searches in the DecentraLend with the name of the desired item. DecentraLend then generates a list of recommended sources for the desired item based on the data available in the MySQL database. DecentraLend uses the cost model of Equation 1 to estimate the borrowing cost for each source and sorts them according to the lowest cost at the

---

**Algorithm 1** Pseudo code for adding a new item

---

```

1: Procedure addItem(userAddr, itemName, fee, deposit)
2:  $itemID \leftarrow insert(userAddr, itemName, fee, deposit)$ 
3:  $block \leftarrow Item()$ 
4:  $block.itemID \leftarrow itemID$ 
5:  $block.possessorAddress \leftarrow userAddr$ 
6:  $block.prevBlockIndex \leftarrow 0$ 
7:  $blockIndex \leftarrow createNewChain(block)$ 
8:  $update(itemID, blockIndex, userAddr)$ 
9: end Procedure

```

---



---

**Algorithm 2** Pseudo code for generating recommendations

---

```

1: Procedure recommendations(itemName, userAddr)
2:  $items \leftarrow getItemsFromDB(itemName, userAddr)$ 
3:  $itemsWithCost \leftarrow estimateCost(items, userAddr)$ 
4:  $recommendedItems \leftarrow sortByCost(itemsWithCost)$ 
5: RETURN  $recommendedItems$ 
6: end Procedure

```

---

beginning. A user borrows a desired item by selecting the source from this recommended list. We illustrate the process recommendations() for generating the recommendation in Algorithm 2. Line 2 of Algorithm 2 finds all sources for the desired item from the database. The cost to collect from each source is estimated at line 3 based on the cost of the security deposit, borrowing fee, and the distance between the possessor and the current user. Then, the sources are sorted in ascending order by their cost to generate the recommendation list at line 4. Once the user gets the recommendation list for an item, DecentraLend allows the user to select any source from the list for borrowing. The process of borrowing a selected item is illustrated in Algorithm 3. Here, a new block because of borrowing an item is created and appended to the chain from lines 2 to 6 of Algorithm 3. The payment for the borrowing fee and the security deposit occur from lines 7 to 13. If the owner and the possessor of the selected item are different users, then the current user transfers the borrowing fee to the owner of

**Algorithm 3** Pseudo code for borrowing an item

---

```

1: Procedure borrowItem(item, userAddr)
2:  $block \leftarrow Item()$ 
3:  $block.itemID \leftarrow item.id$ 
4:  $block.possessorAddress \leftarrow userAddr$ 
5:  $block.prevBlockIndex \leftarrow item.blockIndex$ 
6:  $blockIndex \leftarrow appendToChain(block)$ 
7: if item.owner = item.possessor then
8:    $amount \leftarrow item.fee + item.deposit$ 
9:    $payment(userAddr, item.owner, amount)$ 
10: else
11:    $payment(userAddr, item.owner, item.fee)$ 
12:    $payment(userAddr, item.possessor, item.deposit)$ 
13: end if
14:  $update(item.id, blockIndex, userAddr)$ 
15: end Procedure

```

---

the selected item. On the other hand, the security deposit is transferred to the possessor of the selected item. Finally, the information for the last block index and the possessor of the selected item are modified in the database at line 15. In the case of the procedure, returnApproval() that allows an owner to return an item also appends a new block to the chain, however, the owner transfers only the security deposit to the possessor of the item.

In cases of markAvailable() and returnRequest(), DecentraLend only updates the current status<sup>2</sup> of the item in the database which neither cause access to the blockchain nor require any cost for the gas. On the other hand, myItems() and borrowedItems() can prepare a list of all the corresponding items by reading the data from the database only since all required information is available in the database.

DecentraLend creates a chain for each item by introducing a new block whenever the item is borrowed, added, or returned. A block  $b_i$  in blockchain  $ch_k$  contains information related to a transaction that is related to an item  $p_k$ , a user  $u_i$ , and the index of previous block  $b_{i-1}$  in  $ch_k$ . A user  $u_i$  initiates a chain  $ch_k$  to add a new item  $p_k$  and introduces the first block  $b_i$  of that chain  $ch_k$ . The value for  $b_{i-1}$  is set to zero (0) for the first block of a chain and the possession of  $p_k$  belongs to  $u_i$  at the moment of initiation of a chain. After that whenever  $p_k$  is lent to a user  $u_j$ , the possession information of  $p_k$  is changed to  $u_j$ . DecentraLend appends another block  $b_{i+1}$  to  $ch_k$  to store this lending information. To traverse all blocks in a chain for an item  $p_x$ , we first retrieve the block index  $b_i$  for  $p_x$  from the database that was stored for the last transaction for  $p_x$ . DecentraLend retrieves the block from the blockchain using block index  $b_i$  and recursively uses the value of the previous block index of each retrieved block to all blocks in the chain until the value of the previous block index is found zero (0).

<sup>2</sup>There are three types of status of an item, i) Available to borrow, ii) Not available to borrow, and iii) Item to be returned and unavailable to borrow

**5. EXPERIMENTAL STUDIES**

The experimental environment comprised a 64-bit 2.7 GHz Dual-Core Intel Core i5 CPU, 8 GB of DDR3 RAM, and a 120-GB SSD. We compiled and deployed our proposed DecentraLend system to the Ganache [37] provided personal Ethereum blockchain test network.

In this experimental study, we evaluate the efficiency of the proposed DecentraLend system by comparing the cost of the core functional procedures in Table I with the following two different implementation strategies:

- *BlockchainDB* is the proposed strategy for DecentraLend that exploits blockchain for tracking the borrowers of an item and for secure financial transactions and that uses MySQL database to store users, items, and transaction history information.
- *BlockchainOnly* is another strategy for DecentraLend that does not use MySQL database and considers only blockchain to store and trace all information of users, items, and transactions history and to make secure financial transactions.

We conducted the experiments to calculate the amount of gas required for executing each core functional procedure. We also measured the running time required for executing each core functional procedure.

**A. Required Gas for Execution**

In this experiment, we observed the cost in terms of the required gas for executing the functional procedures that make changes in the blockchain. We plot Figure 2 to represent the evaluated results for different core functional procedures. The y-axis of Figure 2 presents the amount of gas required for executing each procedure on a logarithmic scale.

We note that DecentraLend does not need any gas for executing markAvailable() and returnRequest() in the case of BlockchainDB since here DecentraLend has to access only the database during the execution of these two procedures. However, DecentraLend requires significantly higher gas in other procedures in the case of BlockchainOnly. We observed that DecentraLend has to store more information in the case of BlockchainOnly compared to BlockchainDB. Therefore, these procedures require a higher amount of gas for execution in the environment of BlockchainOnly.

**B. Execution Time**

In this experiment, we observed the cost in terms of running time for executing different functional procedures. We plot Figure 3 to represent the evaluated running time of the selected functional procedures. The y-axis of Figure 3 presents the running time required for executing each procedure on a logarithmic scale. Here, we note that we measured the average running time of a procedure by considering at least 100 executions of that procedure.



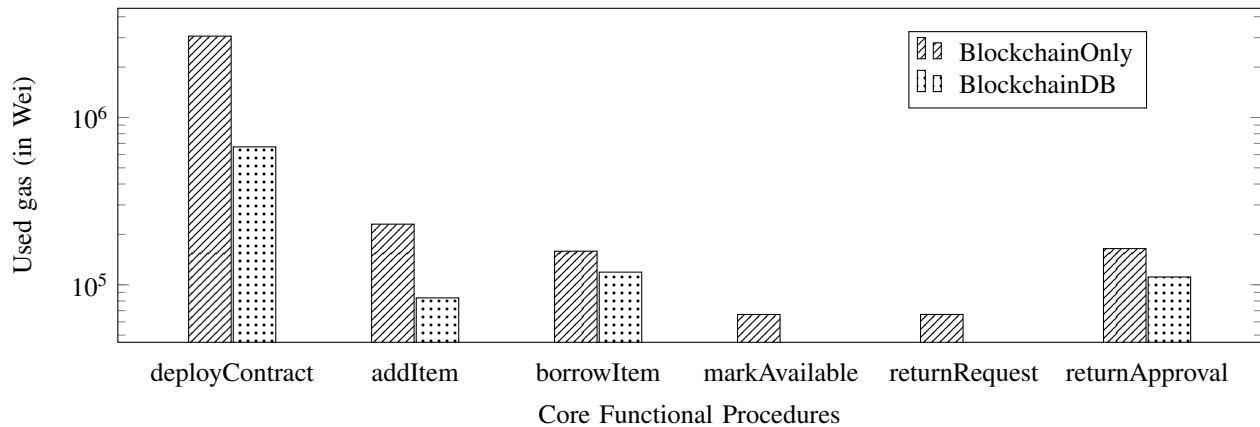


Figure 2. Gas used by different versions of DecentraLend for executing the core functional procedures. The figure shows that DecentraLend uses more gas for executing the procedures in the case of BlockchainOnly.

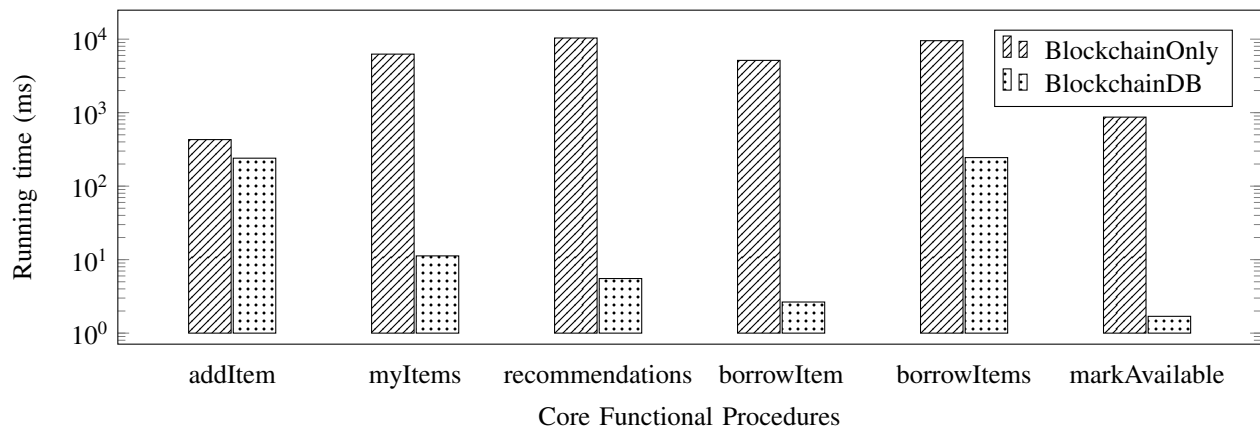


Figure 3. Running time for executing different procedures. DecentraLend takes a significantly higher running time to execute any functional procedure in the environment of BlockchainOnly.

Figure 3 shows that DecentraLend is over two orders of magnitude faster in the case of BlockchainDB compared to BlockchainOnly strategy for the procedures, such as myItems(), borrowedItems(), recommendations(), and markAvailable(). In the case of BlockchainDB, DecentraLend only reads information from the database, however, DecentraLend only accesses the blockchain in the case of BlockchainOnly for these aforementioned procedures. We observed that DecentraLend has to access nodes in the blockchain network in case BlockchainOnly, the cost of accessing the networks significantly affects the running time. On the other hand, both addItem() and borrowItem() procedures add a new block to the blockchain in both strategies. However, the size of information in a block for BlockchainOnly is larger than BlockchainDB, which also affects the running time of these procedures. Therefore, BlockchainDB outperforms BlockchainOnly in the aforementioned two procedures also.

### C. Effect of Chain Length

In this experiment, we observed the effect of the length of a chain on the running time for executing borrowItem(). We measured the average running times to execute the borrowItem() at each case of chain lengths 0, 1, 2, 3, and 4 respectively. We plot Figure 4 to represent the evaluated running times to borrow items in logarithmic scale at different chain-length.

We note that borrowItem() adds a new block to the chain. On the other hand, the validation time overhead required to validate a longer chain increases [38] for an increasing number of blocks in the chain. Therefore, we observed in Figure 4 that DecentraLend incrementally takes a longer time to execute borrowItem() as the chain-length increases.

## 6. CONCLUSIONS AND FUTURE WORK

In decentralized lending systems, the proposed DecentraLend system underscores a fundamental shift from traditional lending services. Our proposed system auto-

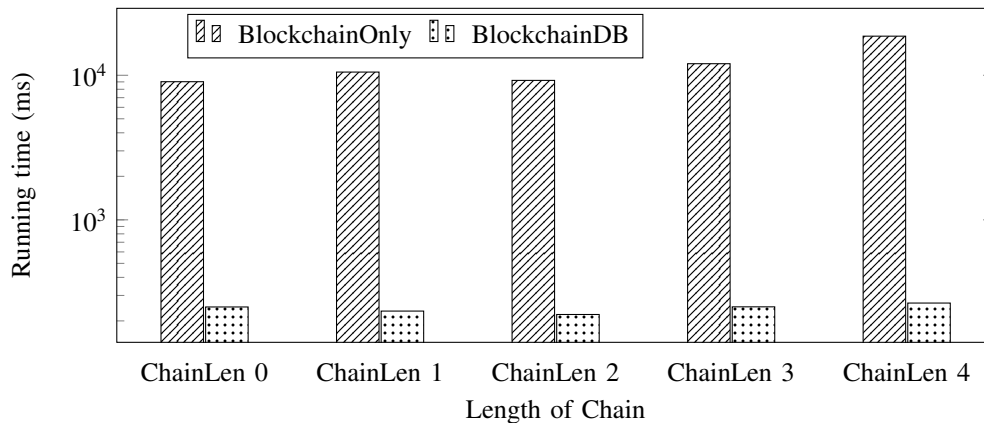


Figure 4. Running time for executing borrowItem() at different lengths of chains. DecentraLend requires gradually longer running time as the length of the chain increases.

mates relaying an item from one borrower to another. It introduces transparency and automation for the lenders and the borrowers by providing the best available alternatives through a recommendation model. Experimental results showed that our proposed strategy of implementing the DecentraLend system significantly reduces gas usage and the overall running time for various operations, such as generating recommendations. DecentraLend currently considers the normalized values of the distance from the processor of an item to a borrower and the distance from the borrower to the owner of that item as the delivery cost. However, we observed that DecentraLend can obtain the precise delivery cost of an item from an integrated delivery service. Moreover, some borrowers may prefer a quick delivery of an item, which was considered in the cost model of DecentraLend. Therefore, we aimed to upgrade the recommendation system of DecentraLend by integrating a delivery service and incorporating the delivery time in the cost model.

## REFERENCES

- [1] L. Rothkrantz, J. Boehle, and M. van Wezel, "A rental system of electrical cars in amsterdam," *Transportation Letters*, vol. 5, no. 1, pp. 38–48, 2013.
- [2] P. Coelho, C. Silva, L. Ferreira, and B. Franca, "Operational improvement of an industrial equipment rental system using discrete event simulation," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 478–483, 2018.
- [3] D.-C. Huang, L.-C. Liu, Y.-Y. Deng, and C.-L. Chen, "An artwork rental system based on blockchain technology," *Symmetry*, vol. 15, no. 2, p. 341, 2023.
- [4] G. van der Heide, "Inventory control for multi-location rental systems," 2016.
- [5] G. Heo, D. Yang, I. Doh, and K. Chae, "Efficient and secure blockchain system for digital content trading," *IEEE Access*, vol. 9, pp. 77 438–77 450, 2021.
- [6] J. Shi, X. Wang, L. Jianmeng, Z. Mingwei, and H. Min, "Content-centric community-aware mobile social network routing scheme," in *2018 14th International conference on mobile ad-hoc and sensor networks (MSN)*. IEEE, 2018, pp. 55–60.
- [7] 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. 217 045–217 056, 2020.
- [8] M. Corciolani and D. Dalli, "Gift-giving, sharing and commodity exchange at bookcrossing. com: New insights from a qualitative analysis," *Management Decision*, vol. 52, no. 4, pp. 755–776, 2014.
- [9] P. Zhu, J. Hu, X. Li, and Q. Zhu, "Using blockchain technology to enhance the traceability of original achievements," *IEEE Transactions on Engineering Management*, vol. 70, no. 5, pp. 1693–1707, 2021.
- [10] R. K. Ray, F. R. Chowdhury, and M. R. Hasan, "Blockchain applications in retail cybersecurity: Enhancing supply chain integrity, secure transactions, and data protection," *Journal of Business and Management Studies*, vol. 6, no. 1, pp. 206–214, 2024.
- [11] F. Krause and E. Rother, "Interlibrary lending in the german democratic republic: problems of centralization in a decentralized system," *Interlending Review*, vol. 10, no. 1, pp. 8–11, 1982.
- [12] U. Agarwal, V. Rishiwal, S. Tanwar, R. Chaudhary, G. Sharma, P. N. Bokoro, and R. Sharma, "Blockchain technology for secure supply chain management: A comprehensive review," *IEEE Access*, vol. 10, pp. 85 493–85 517, 2022.
- [13] S. Krishnapriya and G. Sarath, "Securing land registration using blockchain," *Procedia Computer Science*, vol. 171, pp. 1708–1715, 2020.
- [14] M. Kaczorowska et al., "Blockchain-based land registration: Possibilities and challenges," *Masaryk University Journal of Law and Technology*, vol. 13, no. 2, pp. 339–360, 2019.
- [15] M. Themistocleous et al., "Blockchain technology and land registry," *Cyprus Review*, vol. 30, no. 2, pp. 195–202, 2018.
- [16] H. Taherdoost, "Smart contracts in blockchain technology: A critical review," *Information*, vol. 14, no. 2, p. 117, 2023.

- [17] F. Schär, "Decentralized finance: On blockchain-and smart contract-based financial markets," *FRB of St. Louis Review*, 2021.
- [18] P. K. Ozili, "Decentralized finance research and developments around the world," *Journal of Banking and Financial Technology*, vol. 6, no. 2, pp. 117–133, 2022.
- [19] V. Anand, D. Agrawal, P. Tirkey, and S. Pandey, "An energy efficient approach to extend network life time of wireless sensor networks," *Procedia Computer Science*, vol. 92, pp. 425–430, 2016.
- [20] Y. Purwanto, B. Rahardjo *et al.*, "Acquaintance management algorithm based on the multi-class risk-cost analysis for collaborative intrusion detection network," *Journal of Engineering & Technological Sciences*, vol. 53, no. 6, 2021.
- [21] F. A. Aziz, A. Alsaeed, S. Sulaiman, M. Ariffin, and M. F. Al-Hakim, "Mixed reality improves education and training in assembly processes," *Journal of Engineering and Technological Sciences*, vol. 52, no. 4, pp. 598–607, 2020.
- [22] K. Saengchote, "Decentralized lending and its users: Insights from compound," *Journal of International Financial Markets, Institutions and Money*, vol. 87, p. 101807, 2023.
- [23] S. Joseph and S. Karunan, "A blockchain based decentralized transaction settlement system in banking sector," in *2021 Fourth International Conference on Microelectronics, Signals & Systems (ICMSS)*. IEEE, 2021, pp. 1–6.
- [24] B. Kaplan, V. F. Benli, and E. A. Alp, "Blockchain based decentralized lending protocols: A return analysis between s & p 500 and defi assets," *Journal of Emerging Economies & Policy*, vol. 8, no. 1, 2023.
- [25] M. Verma, "Amalgamation of blockchain technology and knowledge management system to fetch an enhanced system in library," *International Journal of Innovative Research in Technology*, vol. 7, no. 11, pp. 474–477, 2021.
- [26] J. Zeng, X. Dai, J. Xiao, W. Yang, W. Hao, and H. Jin, "Bookchain: Library-free book sharing based on blockchain technology," in *2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*. IEEE, 2019, pp. 224–229.
- [27] J. Cabello, G. Janßen, and A. Mühle, "Distributed library management system based on the blockchain technology," *Atos IT challenge*, 2017.
- [28] X. Liu, "A smart book management system based on blockchain platform," in *2019 International Conference on Communications, Information System and Computer Engineering (CISCE)*. IEEE, 2019, pp. 120–123.
- [29] N. Hasan, "Blockchain technology and its application in libraries," *Library Herald*, vol. 58, no. 4, pp. 118–125, 2020.
- [30] L. Cao and H. Yang, "Building virtual digital library based on p2p and blockchain," in *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 2. IEEE, 2019, pp. 341–345.
- [31] J. Tawormittayakun and A. Leelasantham, "A comparative study of library information storage system through blockchain technology: A case study of undergraduate researches," in *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, 2020, pp. 5–8.
- [32] H. Guo, J. Zhou, J. Wang, and X. Wang, "A bike sharing system based on blockchain platform," in *MATEC Web of Conferences*, vol. 232. EDP Sciences, 2018, p. 01027.
- [33] D. Vujičić, D. Jagodić, and S. Ranić, "Blockchain technology, bitcoin, and ethereum: A brief overview," in *2018 17th international symposium infoteh-jahorina (infoteh)*. IEEE, 2018, pp. 1–6.
- [34] S. Chen, J. Zhang, R. Shi, J. Yan, and Q. Ke, "A comparative testing on performance of blockchain and relational database: Foundation for applying smart technology into current business systems," in *Distributed, Ambient and Pervasive Interactions: Understanding Humans: 6th International Conference, DAPI 2018, Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15–20, 2018, Proceedings, Part I 6*. Springer, 2018, pp. 21–34.
- [35] A. I. Sanka and R. C. Cheung, "A systematic review of blockchain scalability: Issues, solutions, analysis and future research," *Journal of Network and Computer Applications*, vol. 195, p. 103232, 2021.
- [36] J. Eberhardt and S. Tai, "On or off the blockchain? insights on off-chaining computation and data," in *Service-Oriented and Cloud Computing: 6th IFIP WG 2.14 European Conference, ESOC 2017, Oslo, Norway, September 27-29, 2017, Proceedings 6*. Springer, 2017, pp. 3–15.
- [37] "Ganache - Truffle Suite — trufflesuite.com," <https://trufflesuite.com/ganache/>, [Accessed 27-09-2023].
- [38] D. Deuber, B. Magri, and S. A. K. Thyagarajan, "Redactable blockchain in the permissionless setting," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 124–138.



**Mostofa Kamal Rasel** received the Ph.D. degree from Kyung Hee University, South Korea. He also worked as a Post-Doctoral Researcher at the same university. He is an Assistant Professor at the Department of Computer Science and Engineering, East West University, Bangladesh. His research interests include Big Data Compression and Management, SQL Script Optimization, and App Decentralization.



**Mahamudul Hasan** received his M.Sc. degree and B.Sc. degree in Computer Science and engineering from the University of Dhaka, Bangladesh. He is a Senior Lecturer in the Department of Computer Science and Engineering, at East West University, Bangladesh. His research interests include Recommender Systems, Blockchain, Deep Learning, Machine Learning, Artificial Intelligence, and Financial Computing.



**Mohammad Rifat Ahmmad Rashid** received the Ph.D. degree from Polytechnic University of Turin. He worked as a researcher at the LINKS Foundation. He is an Assistant Professor at the Department of Computer Science and Engineering, East West University, Bangladesh. His research interests include Machine Learning, Deep Learning, Anomaly Detection, Time Series Analysis, and more.



**Md Sawkat Ali** received the Ph.D. from Central Queensland University, Rockhampton, Australia. He worked as a Post-Doctoral Research Fellow at Central Queensland University and Deakin University, Australia. He is an Associate Professor at the Department of Computer Science and Engineering, East West University, Bangladesh. His research interests include IoT, machine learning, power electronics, linear and nonlinear control theory and its applications, microgrids, and renewable energy.



**Md Hasanul Ferdous** is currently working as an Assistant Professor at the Department of Computer Science and Engineering, East West University, Bangladesh. He obtained his Ph.D. degree in Computer Science from the Faculty of Information Technology at Monash University, Australia. His current research interests are focused on Cloud Computing, Cybersecurity, the Internet of Things, and Software Engineering.



**Maheen Islam** received the B.Sc. and M.Sc. degrees from the University of Dhaka, Bangladesh, in 1998 and 1999, respectively, and the Ph.D. degree in Wireless Mesh Networking from the Department of Computer Science and Engineering, University of Dhaka, in 2017. She is currently working as an Associate Professor in the Department of Computer Science and Engineering, East West University, Bangladesh. Her current research interests include Wireless Mesh Networks, Wireless Sensor Networks, Cognitive Radio Networks, and Software Defined Networks. She is a member of the Green Networking Research Group (GNR), IEEE. BWIT.



**Mohammad Manzurul Islam** received the Ph.D. degree from Federation University, Australia. He is an Assistant professor at East West University, Bangladesh. He served over 16+ years in both academia and IT industries. His research interests include machine learning, data analysis, IoT, and cybersecurity.



**Md Ariful Islam** received his B.Sc. degree in Computer Science & Engineering from the East West University, Bangladesh. His research interest includes the area of Recommending Systems, Machine Learning, and App Decentralization.