Revolutionizing Microfinance: A Blockchain-Driven Decentralized Finance (DeFi) Model for Collateral-Free Loans

by

Md. Ishmam Tasin
23141062
Md. Rabib Hossain
23141064
Nahin Chowdhury
19141004
S.M. Azwad-Ul-Alam
22341064

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
School of Data and Sciences

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md. Ishmam Tasin

23141062

Md. Rabib Hossain

23141064

Nahin Chowdhury

19141004

S.M. Azwad-Ul-Alam

22341064

Approval

The thesis/project titled "Revolutionizing Microfinance: A Blockchain-Driven Decentralized Finance (DeFi) Model for Collateral-Free Loans" submitted by

1. Md. Ishmam Tasin (23141062)
2. Md. Rabib Hossain (23141064)
3. Nahin Chowdhury (19141004)
4. S.M. Azwad-Ul-Alam (22341064)

Of Spring, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science.

Examining Committee:

Supervisor:
(Member)

Md. Sadek Ferdous

Dr. Md Sadek Ferdous

Associate Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Arif Shakil

Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Microfinance, providing essential banking services to low-income and unbanked individuals, faces numerous challenges such as high-interest rates, cumbersome intermediary processes, and lack of transparency. Decentralized Finance (Defi) has recently been identified as a transformative technology with the potential to ameliorate these concerns and advance microfinance. In this paper, we present a layered Decentralized Application (DApp) designed using Defi to alleviate the security issues intrinsic to conventional microfinance. Our model enables uncollateralized lending to the unbanked population, thus tackling one of the significant barriers in traditional microfinance. In addition to discussing its design and architecture, we demonstrate the DApp on the Ethereum (ETH) network that supports open-source DApps. We then propose a detailed roadmap to reduce intermediaries, lower costs, improve loan accessibility, and engender a more transparent lending environment using blockchain technology. We also examine sustainable methods for uncollateralized loans in the microfinance space, focusing on regions such as Bangladesh. This research underscores the transformative power of Defi and blockchain in reshaping the microfinance landscape.

Keywords: Microfinance; Decentralized Finance; Blockchain; Smart Contracts; Decentralized Autonomous Organization; Decentralised Application; Ethereum; Crypto Wallet; Uncollateralized Loan; Protocol;

Acknowledgement

First and foremost, we would like to give appreciation to the Great Allah for enabling us to conclude our thesis paper on "Revolutionizing Microfinance: A Blockchain-Driven Decentralized Finance (DeFi) Model for Collateral-Free Loans" without any significant setbacks.
Second, we would like to thank our adviser, Dr. Md. Sadek Ferdous, for his kind assistance and guidance. We value the opportunity to collaborate with him highly. The time and energy we invested in this study have given us priceless insights and experiences.
And finally, to our parents whose prayers, love, and encouragement over the years made it possible for us to get where we are now, and for that, we are eternally grateful.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

DAO Decentralised Autonomous Organization

DApp Decentralized Application

DeFi Decentralized Finance

DLT Distributed Ledger Technology

DoS Denial-of-Service

ERC-20 Ethereum Request for Comments 20

ETH Ethereum

EVM Ethereum Virtual Machine

IDE Integrated Development Environment

IMC Information Management Centre

KYC Know Your Customer

MCP Microcredit Programs

MFI Microfinance Institution

MSP Membership Service Provider

NGO Non-Governmental Organization

P2P Peer-to-Peer

PoET Proof of Elapsed Time

PoW Proof of Work

RFM Rural Financial Market

SSR Server-Side Rendering

TVL Total Value Locked

UTXO Unspent Transaction Output

Chapter 1

Introduction

The multifaceted potential of blockchain technology extends far beyond its initial application in the realm of cryptocurrencies, becoming increasingly influential in various industry sectors. Particularly in finance, blockchain and smart contract applications are garnering considerable attention, with a growing number of solution providers turning their focus towards the development of blockchain-powered payment platforms. Such platforms leverage the inherent advantages of blockchain technology: affordability, rapid transaction speeds, transparency, and robust security measures.

Over the past two decades, financial inclusion has been at the forefront of innovative developments, with the primary aim being to ensure widespread access to financial resources. The advent of modern technology has facilitated the evolution of ingenious strategies to promote inclusivity by providing cost-effective financial services. Microfinance, a cornerstone of financial inclusion, extends small-scale loans to underprivileged individuals and nascent enterprises.

Despite the advancements in microfinance services over time, several barriers remain, preventing the intended beneficiaries from fully leveraging these crucial services. High interest rates, extensive intermediary intervention, uneven evaluation methodology, protracted Know Your Customer (KYC) procedures, and a general lack of confidence in financial service providers and microfinance institutions are some of these barriers.

Blockchain technology, with its vast potential, holds promise in addressing these challenges and revolutionizing the microfinance sector in its entirety. Its potential for impact extends to creating a more streamlined, secure, and transparent financial ecosystem, thereby eliminating many of the roadblocks that currently inhibit financial inclusion and microfinance initiatives.

1.1 Problem Statement

The traditional approach to microfinancing poses a myriad of challenges that impede its effectiveness and efficiency. A substantial component of these issues arises from the involvement of numerous intermediaries such as societies or agents. Here are a few of the difficulties:

- **Higher Interest Rates for Loans:** Since microloans typically involve lower loan amounts compared to conventional commercial loans, the cost of customer

acquisition and transaction execution is proportionally high. This necessitates higher interest rates on the loans, essentially to compensate for the increased operational costs, thereby placing a heavier burden on the borrowers.

- **Presence of Multiple Intermediaries:** The presence of several intermediaries in the microfinancing process adds layers of complexity, time, and cost to the transaction. Each intermediary often adds their own fees, further increasing the cost of loans for borrowers and diminishing the effectiveness of microfinance as a tool for financial inclusion.

- **Collateralized Traditional Banking:** Traditional banking systems typically require collateral, an asset pledged as security for repayment of a loan. This requirement poses a significant barrier for many potential borrowers in the microfinance sector who may not possess eligible collateral, thereby excluding them from access to financial resources.

- **Higher Infrastructure and Loan Processing Costs:** The costs associated with maintaining physical banking infrastructure and manually processing loans are significant. These overhead costs inevitably translate to higher costs for the end customer, further compounding the issue of high-interest rates.

- **Lack of Transparency and Data Immutability:** Traditional microfinancing systems often suffer from a lack of transparency with borrowers sometimes unsure of the full terms and costs of their loans. Furthermore, the absence of data immutability makes the system vulnerable to fraud and manipulation.

- **Absence of Centralized Credit Scoring:** Without a centralized credit scoring system, assessing the creditworthiness of potential borrowers becomes a challenge. This absence of a uniform evaluation process can lead to inconsistency in loan approvals and terms, and may also result in deserving individuals being denied access to credit.

- **Higher Customer Acquisition Costs:** The costs of acquiring new customers in the traditional microfinancing sector are often high due to the need for physical meetings, paperwork, and manual verification processes. These higher costs add to the financial burden placed on borrowers, making microfinance less accessible to those who need it the most.

Through our research, we aim to address these problems by leveraging blockchain technology, aiming to reduce intermediaries, lower costs, improve loan accessibility, and create a more transparent and trustful lending environment.

1.2 Research Objective

- **Understanding Traditional Microfinance Models, Blockchain Technology, and Associated DeFi Models:** This involves a comprehensive exploration of existing microfinance structures alongside a thorough understanding of blockchain technology and its implications within decentralized finance (DeFi) models.

- **Comparative Analysis of DeFi Protocols:** To understand the strengths and limitations of existing DeFi protocols, we will conduct a comparative analysis. This will help identify features that may enhance our proposed model.
- **Proposal of a New Stratified DeFi Framework:** Our research aims to propose a novel stratified DeFi protocol. This protocol will be centered around building a DApp, with the aim to change the way microfinance operations are conducted.
- **Designing the Architecture of DApp:** This involves a rigorous process of threat modeling and requirement analysis to ensure the proposed architecture is secure, efficient, and meets the specific needs of the microfinance sector.
- **Development of the DApp Architecture:** Using suitable technology stacks, we aim to develop the proposed DApp's architecture, ensuring its practical applicability and robustness.
- **Building an Uncollateralized Lending Solution:** One of the main objectives of this research is to develop a solution that enables uncollateralized lending in the microfinance space. This would make microfinancing more accessible, especially for individuals without eligible collateral.
- **Development of Appropriate Protocol for the System Workflow:** We aim to develop an efficient and secure protocol to govern the workflow of the system, ensuring seamless operation and user interactions within the platform.
- **Development and Evaluation of the Platform:** Our final objective is to develop the platform based on the proposed architecture and protocol. We will evaluate its performance to ensure its efficiency, security, and usability and make necessary adjustments and improvements. This will involve testing, debugging, and refining the platform before its final release.

1.3 Report Structure

I. In chapter-2, we have added background and talked about all the relevant technologies and current microfinance models.

II. In chapter-3, we have explored the literature review where we critically examine the current research or systems to understand their strengths and weaknesses which could help us to identify gaps in research.

III. In chapter-4, we have proposed thread modeling, conducted requirement analysis, and given an overview of our designed architecture for developing the platform.

IV. In chapter-5, we have designed the protocol along with the sequence diagram for different features of the platform. After that, we implement the platform using the appropriate technology stack and following the designed protocol.

V. In chapter-6, we have discussed the objectives achieved in this report, conducted some cost analysis, and evaluated future goals.

VI. In chapter-7, we conclude by summarizing the report.

Chapter 2

Background Study

In this chapter, we briefly summarize blockchain and microfinance systems relevant to our thesis.

2.1 Blockchain

Blockchain technology ensures the trustworthiness, security, transparency, and traceability of data shared across a business network. It is a collection of records called blocks that store data in a public, chronological sequence among a cluster of Peer-to-Peer (P2P) network participants or nodes [21]. Bitcoin [3] was the first decentralized digital money to use this Distributed Ledger Technology (DLT) [17]. Data on the blockchain is encrypted using cryptography to ensure that the user's privacy is protected and that the data cannot be manipulated. Each block in the network directs to its previous block using a cryptographic hash, which directs to its previous block, and so on. This forms a chain, which is called blockchain [18]. Information on a blockchain network is not governed by a centralized authority, in contrast to contemporary financial organizations. The network's nodes, who also have the democratic power to authorize each transaction that may take place on a blockchain network, retain the data. Therefore, a public blockchain is the most widely used kind of blockchain network. As long as you have network connectivity, you may access the data on the blockchain. You may access the same version of the ledger as the other members of the blockchain network if you sign up for it. The other participants will be immediately informed and will be able to fix the issue as soon as it is practical, even if one node or piece of data on one participant's computer gets corrupted. [14]. There are currently at least four different types of blockchain networks:

- Public blockchains
- Private blockchains
- Consortium blockchains
- Hybrid blockchains

2.1.1 Public Blockchain

In the case of a public blockchain, it is a distributed ledger where everyone is welcome to enter. The general public can partake in the governance of blockchain and all transactions are open to the public [19]. It is also known as a permissionless blockchain. Ethereum [43], Bitcoin [41], Solana [61], and Litecoin [53] are some of the most famous public blockchain networks out there. We will briefly talk about Ethereum as we will be building our model on this network.

2.1.1.1 Ethereum

Ethereum is a worldwide, decentralized, smart contract-based software platform built on the blockchain [7]. Ether, abbreviated ETH, is the most well-known cryptocurrency created on the platform. Ethereum is scalable, programmable, secure, and decentralized. Smart contracts [26],the foundation of decentralized apps (DApps) [13],are natively supported in Ethereum. Smart contracts are digitally signed immutable contracts that are made by computer programs. They run independently on the backend of open-source blockchain software applications known as DApps. More on these two topics will be elaborated later on.

2.1.2 Private Blockchain

Blockchain technology may still function under certain constraints, despite the fact that the original blockchain was meant to run democratically without the interference or influence of banks or central authorities. Private blockchains, also known as managed blockchains, are permission-driven blockchains that are administered by a single organization [12]. The central authority in a private blockchain decides who can be a node. Furthermore, the central authority does not always grant each node equal rights to execute functions. Because public access to private blockchains is restricted, they are only partially decentralized. This means that a bank or organization may create its own blockchain and choose which transactions to include in it. It will remain a secure system, but, like traditional banking, it will be predicated on the trust of the decision-maker [27].
Next, we discuss a few private blockchain platforms.

2.1.2.1 Hyperledger Fabric

It is an enterprise-focused, permissioned blockchain technology that is open source [49]. Its versatility and simplicity of integration with membership services and other blockchain technologies make it a game-changer [20]. Like other blockchain technologies, it employs smart contracts, a ledger, and participant control for transactions [50]. A trustworthy Membership Service Provider (MSP) is used instead of a permissionless system that requires "proof of work" to enroll, validate transactions and defend the network. Its plug-and-play component ecosystem and modular design are strong. It improves anonymity and speed on private blockchains. Due to its open smart contract architecture, it works with several data models, including accounts and unspent transaction outputs, or UTXOs [50].

2.1.2.2 Hyperledger Sawtooth

Developed by Intel, it creates, deploys, and enforces distributed ledgers. The platform provides consensus techniques based on network size. Proof of Elapsed Time (PoET) is one example of a salable method used here [20]. Its flexible and modular architecture isolates the core system from the application domain, allowing programmers to define application business rules without understanding the underlying system's design while maintaining system security [51]. Sawtooth, in contrast to other blockchains, is designed specifically for managing business supply chains. All transactions are bundled into a block before being signed and sent to a validator by the client. Sawtooth increases productivity by processing REST API-based transactions in parallel rather than serially [15].

2.1.2.3 Quorum

Owned by ConsenSys, Quorum blockchain [60] is built on top of Ethereum, which permits the creation of a permissioned blockchain network that guarantees transaction privacy [29].

2.2 Smart Contract

A new kind of distributed ledger has developed from the Bitcoin ledger that makes it possible to install and run contracts written in computer code on top of the ledger itself. This computer program is called smart contract [17]. The term 'smart contract' describes the concept of digitally signed and automatically carried out contracts [13]. They facilitate the development of decentralized apps (DApps), which are computer programs that run independently of the rest of the system [13]. Thanks to their inclusion in the ledger, smart contracts achieve the properties of immutability and irreversibility. These are highly desirable qualities with several practical applications during execution [17]. Ethereum developers may construct a variety of smart contracts, which are essentially executable programs inscribed into blocks, thanks to Solidity [62], a Turing-complete programming language [13].

2.2.1 Decentralized Applications (DApps)

Decentralized Applications (DApps) are computer applications that operate on the distributed blockchain or peer-to-peer (P2P) network of computers [13]. Decentralized applications employ smart contracts to operate independently on a blockchain system. A DApp that has been successfully deployed would, in an ideal world, not need any further supervision from developers or maintenance. This ideology originated the idea of Decentralised Autonomous Organizations (DAOs) [5]. These DAOs are the paradigm that should be used for blockchain applications and services because of their capacity to operate autonomously. Their activities are managed according to set protocols that are carried out via smart contracts based on blockchain technology. By documenting all transactions in the blocks, a DAO is able to automatically and openly redistribute the costs and profits of running the organization among all of its members. Bitcoin, the most well-known implementation of blockchain technology, is really a DAO [13].

2.2.1.1 DApp Characteristics and Architecture

According to [10], there are four key characteristics that set DApps apart:

- **Open Source:**
  Codes of DApps should be open source so that they may be verified by third parties who are not affiliated with the blockchain.
- **Internal Currency:**
  DApps should be allowed to utilize tokenized assets that are already established inside the ecosystem of DApps. Within a decentralized application, the quantification of credits and transactions involving content providers, distributors, and consumers is made possible via the use of tokens.
- **Decentralized Consensus:**
  A decentralized consensus is necessary for transparency in order to ensure that all nodes are on the same page. This is because it guarantees that all nodes have the same information.
- **No Central Point of Failure:**
  There should be no single point of failure since all of the application's components are stored and executed on the blockchain.

DApps vary from Web 2.0 [4] apps in architecture. Without servers or databases, DApps utilize the distributed nature of blockchain maintained by computers worldwide. Based on [30], top-level architecture of an Ethereum DApp is illustrated in Figure 2.1. The architecture of DApp generally consists of 3 main components namely, Crypto Wallet to hold assets [23], the Front End of a DApp which is hosted on a web server, and the Ethereum blockchain.

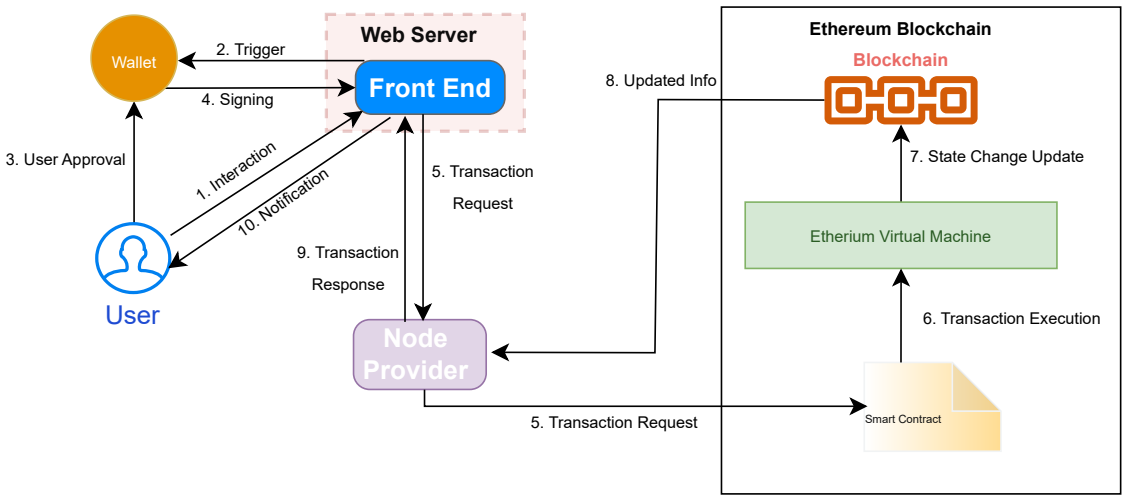

Figure 2.1: Ethereum DApp Architecture

The components interact in the following way:

1. **User Interaction:** A user enters a web browser and interacts with the DApp through its front-end interface. This could either involve -

I. Submitting a form to execute a function on a smart contract.

II. Reading data from the Ethereum blockchain.

2. **Wallet Invocation/ Trigger:** If the user invokes a transaction (i.e., writing data to Ethereum), the front-end triggers the Crypto Wallet. Metamask [54] crypto wallet is generally used for Ethereum. It is a web browser-based wallet that opens a transaction approval request [33]. The user then reviews the transaction in detail.

3. **Transaction Approval:** If the user is satisfied with the transaction details, he approves it on Metamask. If he is not satisfied, he cancels the transaction and begins a new one.

4. **Transaction Signing:** After approval of the transaction, Metamask uses the private key of the user to digitally sign the transaction. This online signature proves that the transaction was actually authorized by the user.

5. **Node Provider Transaction Request:** The read request or signed transaction is sent to a node in the Ethereum network. Node providers allow the user to use blockchain services without managing your own node by connecting you to them remotely [34]. If it is a signed transaction request, the Ethereum node then sends it across the Ethereum network to be included in a blockchain. If it is a read request, the node directly returns the requested data from the blockchain.

6. **Smart Contract Execution:** The smart contract executes specific functions with the inputs given in the signed transaction. This might also involve altering the state of the contract from the initial conditions. The transactions are gathered up by miners [22] on the network. They affirm its validity and then execute the instructions encoded in it. This execution involves running the specified function in a smart contract within the Ethereum Virtual Machine (EVM) [44].

7. **State Change Updated to Blockchain:** Once the smart contract function is executed and the state changes, the block containing this new transaction is confirmed and added to the blockchain.

8. **Node Updated Info:** The updated blockchain ledger is sent to each node. The front-end fetches the date from the node provider.

9. **Transaction Response:** Transaction response is shown in the user interface of the DApp. The front-end queries the new state of the blockchain to update the user interface. Filed transactions are also shown here.

10. **User Notification:** In the end, the user is notified of the result of the transaction or the data he requested to be displayed.

2.3 Traditional Banking

Traditional banking is the financial system that offers a variety of services, such as accepting deposits, providing loans, and providing financial advisory services. They serve as financial intermediaries between depositors and borrowers, playing a critical role in the economy.

2.3.1 Collateralized Loans

In traditional banking, collateralized loans are common. They are a type of loan backed by an asset that serves as collateral. If the borrower defaults on the loan, the lender has the right to seize the collateral to recoup its losses. Examples of collateral include real estate, vehicles, and other valuable assets.

2.3.2 Uncollateralized Loans

Uncollateralized loans, on the other hand, do not require borrowers to pledge an asset as collateral. They are more risky for the lender because if the borrower defaults, the lender has no assets to seize. These loans are often based on the borrower's creditworthiness than their assets.

2.3.2.1 Semi collateralized Loans

Semi-collateralized loans represent a hybrid between collateralized and uncollateralized loans. In this case, a borrower is required to pledge some collateral, but the value of the collateral is typically less than the loan amount. This scenario is common when the borrower does not have sufficient assets to fully secure the loan but is deemed creditworthy enough to mitigate the lender's risk partially.

2.4 Crypto Loans

Crypto loans are a type of digital lending that occurs within the DeFi space. Borrowers can secure loans using their cryptocurrency holdings as collateral, and lenders can earn interest on their crypto assets by lending them out.

2.4.1 Collateralized Crypto Loans

Collateralized crypto loans are those where borrowers provide cryptocurrency as collateral to secure their loans. If the borrower defaults, the lender can seize the crypto assets to offset their losses. This type of loan offers a way for crypto owners to access funds without having to sell their holdings.

2.4.2 Uncollateralized Crypto Loans

Uncollateralized crypto loans are less common but represent an exciting frontier in the DeFi space. These are loans given out based on the borrower's reputation or

other non-asset-based factors, making them accessible to a wider range of borrowers. However, due to the higher risk involved, these loans often require innovative solutions to ensure the lender's security.

2.5 Microfinance

Microfinance, or microcredit, refers to financial services, such as loans, savings, and insurance, given to poor and economically disadvantaged clients who lack access to traditional banking services [9]. In a developing nation like Bangladesh, member-based Microfinance Institutions (MFIs), organizations that give out these microloans, are a rapidly growing subset of the Rural Financial Market (RFM). A variety of official financial institutions, including nationalized commercial and specialized banks, specialized government agencies, and Non-Governmental Organizations (NGOs), dispense these Microcredit Programs (MCP) for the poor and destitute [55]. Some of the top MFIs in Bangladesh are Grameen Bank [47], BRAC [42], and ASA [40].

2.5.1 Overview of Microfinace

2.5.1.1 Role

Since the majority of the people in Bangladesh are poor, microfinance has a significant crucial role in the country. Its reach enhances financial inclusion and poverty reduction for millions of low-income people and small business owners. This is mostly due to governments and NGOs expanding financial services, as 30 million impoverished individuals directly benefit from microcredit programs [55]. The journey of microcredit began when Nobel laureate Dr. Muhammad Yunus, in the 1970s, lent an amount of his own money to 42 rural residents in Jobra village chosen by his students [9], thus transpiring the idea of a non-collateralized lending system.

2.5.1.2 Objectives

The microfinance system in Bangladesh serves multiple objectives. Predominantly, it offers small loans to entrepreneurs without access to standard banking services, enabling them to establish or expand their businesses. This catalyzes economic growth at the grassroots level. Moreover, MFIs in Bangladesh provide savings programs[16], insurance products [32], and promote financial literacy and stability among marginalized communities [8].

2.5.1.3 Microfinance Model

MFIs in Bangladesh primarily operate through group-based lending models [11]. Individual borrowers have access to loans as part of a group, and each member of the group ensures the repayment of the entire group's debt. This mitigates the risk for the lending institutions, as a mechanism of trust is there. According to [9], generally MFIs have the following delivery models and products:

- **Targeting Impoverished Households:**
  Following the Grameen Classic System, Bangladeshi MFIs prioritize low-income

households for giving out loans. For example, Grameen Bank targets families with less than half an acre of land. They are also considered if their total assets are equal to or less than a medium-quality acre of land.

- **Group Organization:**
  Each gender forms "like-minded" groups with a few members. A certain number of these groups are called to host weekly village bank meetings. Everyone is able to speak here. In these meetings new loan ideas are proposed, loan paybacks are made and obligatory savings are collected.

- **Local Offices:**
  MFIs establish local offices to support weekly in-person meetings with rural communities and field officials. These congregations promote friendship and economic efficiency.

- **Group Autonomy and Responsibility:**
  Groups are the model's basic organizational units. They decide on loan applications and purposes for loans. They also share saving deposits and debt repayment responsibilities as a whole.

- **Savings Deposits:**
  Members deposit their weekly savings to protect their money and receive interest. Savings enable the accumulation of capital for lending.

- **Loan Products:**
  The most common microcredit product is short-term loans with interest. And they are payable weekly. After repaying lesser debts, members can borrow more later on. In addition to savings and loan products, some MFIs provide "quasi or partial insurance" and remittance services for clients.

- **Flexibility and Variation:**
  MFIs have grown more adaptive, removing the shared liability of group members while preserving the group-based approach. Some MFIs offer optional savings programs. Branch managers are given the freedom to determine loan amounts and payback durations in many MFIs based on judgment and intuition.

- **Non-Financial Services:**
  Some MFIs provide entrepreneurial training, consulting, and market links to low-income families. These programs are often referred to as "Microfinance-plus.".

So from [9] we get workflow for MFIs, which is expected in Figure 2.2. It is described below:

1. MFIs target clients from impoverished households.
2. The client joins a group.
3. The groups attend organization meetings per week.
4. The group members receive loans at the meeting.

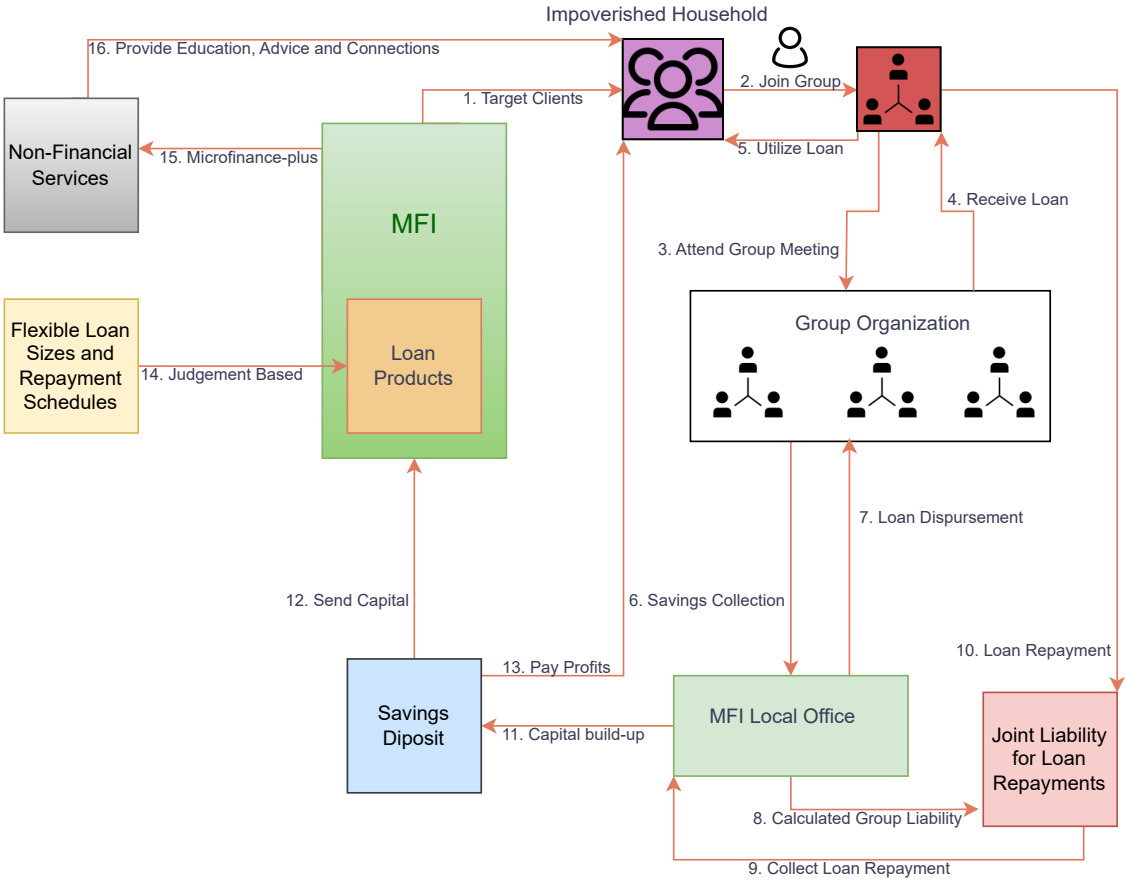

Figure 2.2: Microfinance Model

5. Each of them utilizes that loan.
6. From the MFI local office, savings are collected from the group organization meeting.
7. Loans are also disbursed from the MFI local office.
8. The local office also calculates the joint liability for each group.
9. They also collect the loans repaid by each group
10. Loans are repaid collectively by each group as they are counted as joint liability
11. The local MFI office sends the collected capital into the savings account.
12. Savings are used to provide loans, so the capital is sent to the MFI.
13. Some of the profit is provided to impoverished households.
14. Flexible loan products are nowadays provided by MFIs based on judgments given by MFI officials.
15. Microfinance-plus services are provided nowadays, which are non-financial services.

16. These services provide education, connections, and advice to impoverished households.

2.5.1.4 Challenges and Privacy Issues

Despite its meaningful contributions, the microfinance sector in Bangladesh faces several challenges. These include things like high operations costs, over-indebtedness, high-interest rates, a lack of regulation and transparency, staff indifference to responsibility, and a high dropout rate among the borrowers [28]. Moreover, with the increasing adoption of digital technology in financial transactions, new risks concerning data security and privacy have emerged. These include multiple borrowing, weak verification processes, staff corruption, and poor loan accounting [28]. Ensuring these issues are addressed is critical for the sustainable development of the sector.

2.6 DeFi- Decentralized Finance

Decentralized finance, or DeFi, is a revolutionary model that leverages blockchain technology to recreate and improve upon traditional financial systems. DeFi applications operate on open, permissionless networks and aim to eliminate intermediaries. They offer a variety of financial services, including lending and borrowing, insurance, asset trading, and more, in a more transparent, secure and accessible way.

Chapter 3

Literature Review

In this chapter, we will have a detailed understanding of how each of the microfinance and Defi segments fundamentally work. Here we will go in-depth about all the current research or systems and critically analyze them. We will identify their pros and cons so that we can have a solid understanding of the research gaps.

3.1 Collateralized Lending with Blockchain

3.1.1 MakerDAO

MakerDAO [63] is a pioneer in the field of collateralized lending via blockchain. It allows users to borrow its stablecoin, DAI, by locking up their Ethereum assets as collateral. In the event of a default, the smart contract autonomously liquidates the Ethereum collateral to cover the loss. MakerDAO introduced a slew of game-changing improvements to the banking industry that together shook things up significantly. To begin with, it removed the requirement for a centralized authority in the lending process by enabling any user to borrow a stablecoin called DAI by putting up ETH as collateral. The MakerDAO team also developed USD-pegged ERC-20 stablecoin DAI using just Maker algorithms and incentives to ensure the dollar's value. On the Ethereum mainnet, the protocol became live on December 18th, 2017, and it has already amassed a Total Value Locked (TVL) of $17.9 billion. Beyond its initial application in financial transactions, blockchain technology has advanced significantly thanks to this landmark.
Currently, Maker Protocol users may deposit a wide variety of tokens in addition to ETH in order to mint DAI. The protocol relies on a pair of tokens for its functionality. The first token is called DAI, and it's an Ethereum-based stablecoin with a "soft peg" to the US dollar and collateral assets as backing. The second token, MKR, is used by MakerDAO members as part of a system of governance known as "Maker Governance." MKR holders function as a board of directors, and the public has full access to the organization's operations, voting procedures, and source code. Members with the most voting power also have the most financial stake in the organization since their voting power is proportional to the amount of MKR they have put in a voting contract.

3.1.2 Aave

Aave [36] is another notable DeFi platform that facilitates collateralized lending. Aave's users can lend and borrow a variety of cryptocurrencies, with loan terms enforced by smart contracts. Aave has added innovative features such as "flash loans", which are uncollateralized but must be paid back within the same transaction.

3.2 Uncollateralized Lending with Blockchain

3.2.1 TrueFi

TrueFi [64] is an example of a platform offering uncollateralized lending in the blockchain sphere. TrueFi uses an on-chain credit model to issue loans without requiring collateral. It uses a credit prediction market where TRU token holders vote on loan approvals and rate risk, allowing the platform to lend without collateral based on the assessed creditworthiness of the borrower. As per its app main page, TrueFi boasts of having offered uncollateralized loans on-chain totaling more than $1 billion with zero defaults. Even by itself, this accomplishment deserves studying. While anyone can, in a permissionless fashion, act as a lender in TrueFi, loans are only made to vetted and white-listed institutional borrowers. Large institutional debtors with public reputations would pay significant intangible costs in the event of failure; hence that decision was taken.

3.3 Microfinance Digital Platform Without Blockchain

3.3.1 Kiva

Kiva [52] is a popular microfinance digital platform that does not use blockchain. Kiva connects lenders with borrowers from low-income backgrounds or regions with limited access to traditional banking systems. However, Kiva relies on a network of field partners, typically microfinance institutions, to administer the loans.

3.4 Microfinance Digital Platform With Blockchain

3.4.1 Moeda Seeds

Moeda Seeds [58] is a cooperative banking-as-a-service platform that provides a digital token called MDA pegged to the fiat currency and allows for peer-to-peer payments, remittances, microfinancing and more. It leverages blockchain technology to create a transparent and traceable system to boost financial inclusion, especially for underbanked or unbanked individuals. Moeda Seeds aims to empower local businesses by providing microloans for entrepreneurial projects with positive social impacts.

3.5 Comparative Analysis of Different Platforms

Here's a comparative analysis of different platforms in tabular format:

Platform	Uncollateralized	With Blockchain	Support Microfinance
MakerDAO	No	Yes	No
Aave	No (Except Flash Loans)	Yes	No
TrueFi	Yes	Yes	No
Kiva	Varies (Based on Field Partners)	No	Yes
Moeda Seeds	No	Yes	Yes

Table 3.1: Comparative analysis of different lending platforms

3.6 Identifying the Need For Uncollateralized Microfinance Lending Protocol

Despite the advent of blockchain and its application in various sectors, there is still a significant gap in its utilization within the microfinance industry, especially in developing countries like Bangladesh. This necessitates the development of a new protocol that caters specifically to the unique needs and challenges faced by microfinance institutions and their clients in these countries. The research problems we previously identified highlight several fundamental issues within the current microfinance system that can be addressed through a novel blockchain-based protocol:

- **High Interest Rates:** The high cost of operations and credit risk in microfinance often lead to high-interest rates, making loans less accessible to the underprivileged. A blockchain-based system can significantly reduce operational costs through automation and smart contracts, thus potentially lowering interest rates.
- **Numerous Intermediary:** Multiple intermediaries involved in the loan process not only increase the cost but also the time taken for loan approval and disbursal. Blockchain technology can enable peer-to-peer lending, reducing the need for intermediaries.
- **Lack of Transparency and Data Immutability:** Traditional microfinance systems lack transparency, leading to trust issues. Blockchain can provide a transparent, immutable ledger of transactions that is auditable by all parties, enhancing trust and credibility.
- **Inefficient Credit Scoring:** The absence of a central credit scoring mechanism can lead to unequal and unfair lending practices. Blockchain technology can enable a transparent and universal credit scoring system, making the lending process more equitable.
- **Operational Inefficiency:** Traditional microfinance suffers from slow processes and high operational costs due to paperwork and manual processes. Blockchain can automate many of these processes using smart contracts, improving efficiency and reducing costs.
- **Collateralized Lending:** Conventional microfinance often requires collateral, which many underprivileged individuals lack. The new blockchain protocol can allow for uncollateralized lending based on trust and social reputation metrics, making loans more accessible

In light of these issues, the development of a blockchain-based protocol for microfinance tailored to the unique socio-economic context of developing countries like Bangladesh is not only justified but also crucial for promoting financial inclusion and economic development.

Chapter 4

Proposal

This chapter details the strategies we have employed to tackle the issues highlighted in the problem statements. It also discusses our analysis of various existing models. Our goal is to create a suitable architectural model and protocol that can effectively solve these problems and fulfill our research objectives.
Our aim is to create a system that enables borrowers to access loans through the platform without the need for collateral while allowing lenders to provide funds without relying on a central institution. This objective aligns with our vision of decentralizing the system. To accomplish this, we propose a novel Defi DApp.
By leveraging blockchain technology, we can eliminate the need for intermediaries, thereby reducing transaction costs. The decentralized nature of blockchain further enhances the efficiency and transparency of the system.
To better implement the proposed model we first need to do threat modeling and requirement analysis.

4.1 Threat Modelling

Establishing such a platform would need careful consideration of a number of factors, not the least of which is security and risk assessment. We have chosen to simulate risks using STRIDE [6], a well-established threat model. There are many different types of security threats that this model accounts for. According to [24], there are some security considerations, and keeping that in mind, we build the platform:

- **T1-Spoofing Identity:** An assailant impersonates a user of the system for malicious purposes, using spoofed email addresses or phone numbers, for instance.
- **T2-Tampering With Data:** An attacker may alter the data of an entity (or other entities) stored in our system in a manner that prevents the platform from functioning normally. For instance, destroying, manipulating, or modifying information through unapproved channels.
- **T3-Repudiation:** A repudiation attack occurs when an application or system fails to implement controls to properly monitor and log users' actions, allowing malicious manipulation or forgery of the identification of new actions.
- **T4-Information Disclosure:** Sensitive system data is accidentally disclosed to an adversary.

- **T5-Denial of Service (DoS):** The purpose of a Denial-of-Service (DoS) attack is to render a machine or network inaccessible to its intended consumers.
- **T6-Elevation of Privilege:** By using additional attack vectors, such as malicious software, to undermine the system's authorization capabilities, an attacker might increase his privileges without the awareness of a responsible user (for example, an administrator). We take into account the following risks in addition to STRIDE threats:
- **T7-Replay Attack:** Any prior communication may be captured and reused by an attacker to dynamically add a deleted entity. A replay attack takes place when a hacker listens in on a secure network connection, intercepts it, and then purposefully delays or resends it to trick the recipient into doing what the hacker wants. Replaying earlier transactions may be useful in this situation.

4.2 Requirement Analysis

We give a variety of functional requirements in this part to make sure the system works as intended and security needs to counteract the dangers that have been discovered.

4.2.1 Functional Requirements (FR)

Our system is designed to foster decentralization, allowing borrowers to obtain loans and lenders to offer them without the involvement of a centralized institution. It would formalize and record the terms of the agreement between borrowers and financial institutions using smart contracts.

- **F1:** Developing a system where the borrowers can borrow money using the platform, lenders can lend money, and stakers can stake their money without any central institution, thus achieving decentralization of the system.
- **F2:** Developing a system where we integrate the existing NID verification system into our platform so that we can carry out the KYC for each customer. Financial institutions have access to customer information that has been securely kept. This streamlines KYC and minimizes manual involvement.
- **F3:** Develop a system where the stakers can stake money using the platform through appropriate terms and conditions.
- **F4:** In order to offer credit history for verification by the financial institutions, create a ledger record utilizing many characteristics, such as property records, financial transactions involving the sale of products, and other papers. The authenticity of this information is not an issue for financial institutions since these records are unchangeable and cannot be altered. We are able to assess any customer's creditworthiness in this way.
- **F5:** Use the smart contract to create and save the contract with all the required terms and conditions between the borrower and the lending company.

4.2.2 Security Requirements (SR)

The security of our proposed system is of the utmost importance; therefore, we conducted extensive planning to ensure the necessary security protocols. Important criteria included:

- **S1:** Only authenticated and authorized users must be able to access the relevant services, according to the system. This solves T2.
- **S2:** Any data involving user information and transaction history must be transmitted securely over the network to ensure the confidentiality, authenticity, and integrity of the request and response. This diminishes T2 and T4.
- **S3:** Every request and answer needs to be immutably recorded. Combining S2 and S3 may reduce T3 and T6.
- **S4:** Ensuring high levels of network security and continuous network traffic monitoring can prevent DoS attacks, thereby mitigating T5.
- **S5:** In order to stop an adversary from leveraging a prior request and answer, the system must take the proper security measures. This will lower the threatT7.

4.3 Architecture Design

In this section, we present the architecture of our proposed framework, as depicted in Figure: 4.1. The architecture comprises various entities that interact and share data to facilitate the desired actions. At the highest level, there are five key entities within the architecture: Users (including lenders and borrowers), identity providers for the identity verification server, decentralized applications (DApps), and stakeholders. Next, we discuss the functionalities of each of these components.
Let's delve into the functionalities of each component:

- **Protocol Manager:** Protocol managers are granted access to the system by logging in using their assigned credentials. Upon successful login, they are provided with the requisite authority to perform various actions, including the approval of loans and the management of group vaults established by group lenders. Group vaults are the lending pool filled with monetary assets. It is created by lenders for borrowers to take loans from. The idea of group vaults will be discussed later on in detail. Additionally, protocol managers have the capability to initiate borrowing groups for borrowers who do not have any stakers available to stake their loans. This enables protocol managers to efficiently oversee and facilitate the borrowing process within the microfinance system.
- **Borrower:** The platform accommodates two types of users, with borrowers being one of them. Borrowers can access loans without the requirement of collateral, enabling them to obtain funds for their financial needs.
- **Lender:** Lenders are individuals who invest their money to earn interest over a specified period. The funds contributed by lenders are pooled together and subsequently disbursed to borrowers following the predefined protocol.

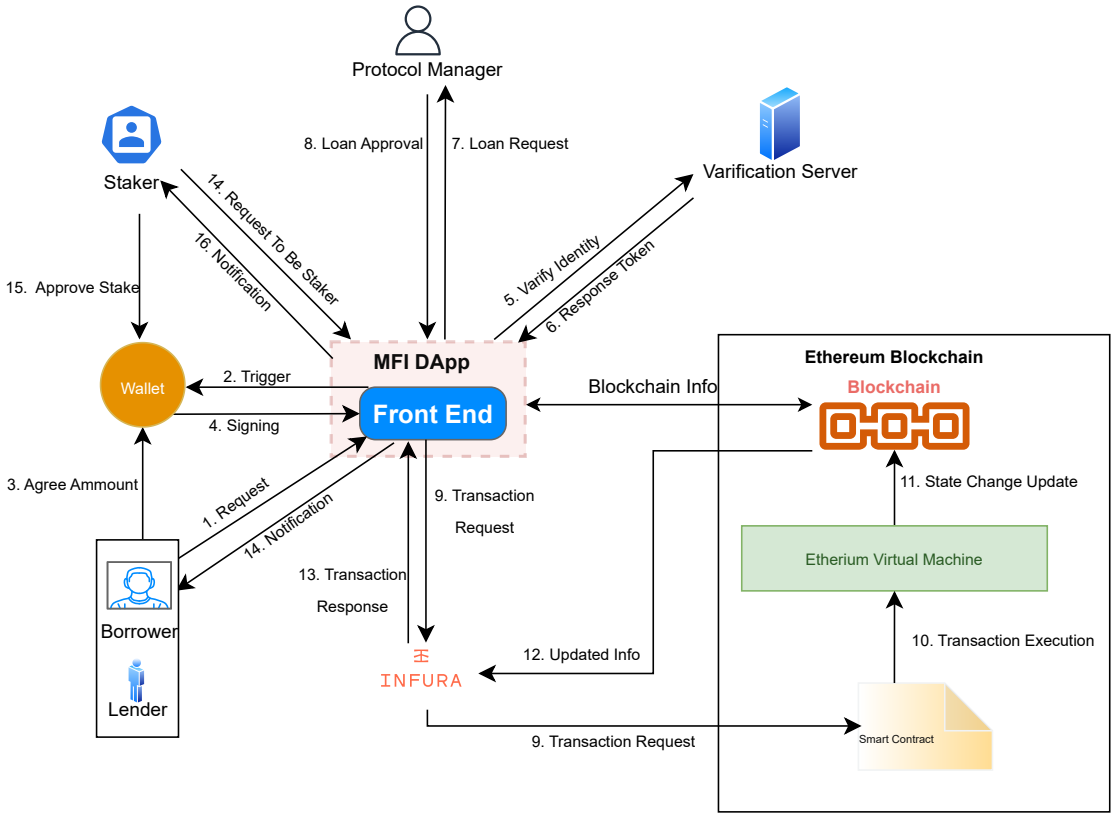

Figure 4.1: Architecture

- **The Staker:** Stakers possess the unique ability to invite individuals into the system. To become a staker, a user must hold a certain amount of tokens (Stablecoin). The benefits of being a staker include the opportunity to earn profits. When they refer someone to the system, they receive profits based on the interest earned from borrowers. However, in case of default, stakers incur a loss proportionate to the defaulted amount. Additionally, defaulting users adversely affect the stakers' credit scores, reducing their chances of availing loans in the future. This on-chain data can also influence future financial decisions for the particular user.

- **On-chain data:** On-chain data refers to the information stored on blockchain, accessible to smart contracts for making specific decisions. A ledger record is created with various attributes, including financial transactions, previous loan amounts, default rates, and repayment rates in money per day. The immutability of this data ensures its accuracy and eliminates concerns for financial institutions.

- **Identity Providers:** Existing NID verification systems will be integrated into our platform to facilitate Know Your Customer (KYC) processes for each customer. Blockchain technology helps establish a secure digital identity for customers, stored in a protected manner and accessible to financial institutions. This streamlines the KYC process, reducing the need for manual intervention.

- **DApps(The Platform):** A DApp acts as a web server, serving as mid-

dleware between web applications and the blockchain. The platform's main objective is to replicate the functions of traditional organizations through an intricate system of smart contracts, eliminating the need for corporate executives and hierarchical structures. Policy decisions within the DApp are determined through a weighted voting system, where members with larger token holdings possess greater voting power. This concept is based on the notion that individuals who have invested more funds into a platform are more likely to participate honestly for the organization's benefit.

Chapter 5

Protocol Flow and Implementation

In this section, we will gain a comprehensive understanding of the fundamental operations of the system and the protocols involved in performing specific tasks within the system. As mentioned earlier, there are various entities within the system, each responsible for different actions.
To illustrate the workflow, let's consider a scenario: A lender joins the platform and deposits tokens with the intention of earning interest. The deposited funds are then combined with the funds from other lenders, creating a pool. This pool serves as the central resource for providing loans to borrowers, following the designated protocol. Subsequently, we will explore different components of the protocol by employing relevant scenarios. This approach will help us comprehend the individual elements of the protocol and how they contribute to the overall workflow.

5.1 Onboarding and Authentication

When a user comes into the platform they need to go through 2 stages of verification. These data are then stored on-chain along with other relevant information. If any borrower wants to avail the services they have to be registered using their NID. The 2 processes are mentioned below:

- **Sign Up Using Private key**
- **NID verification**

Here in Figure 5.1 we show the protocol flow of signup and verification. The figure shows different interactions between the user, DApp, NID server, Biometrics, and On-chain data. The user signs up and verifies his information through NID and gets back notification based on the authenticity of the information given by them.

5.2 Becoming a Staker

Users have the opportunity to participate as stakers within the system by actively choosing this option. However, it is important to note that in order to become a staker, users must undergo the NID verification process. Additionally, to stake a loan, users are required to allocate funds to their staker balance, which should be distinguished from their regular user balance.

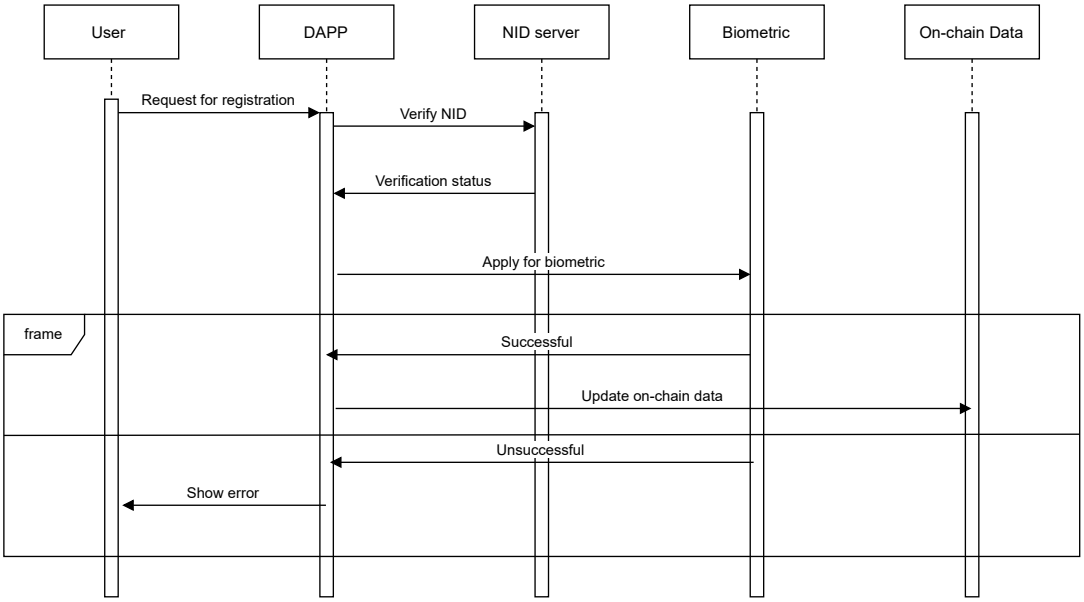

Figure 5.1: Signup and verification

Once users have sufficient funds in their staker balance, they can commence the process of staking for loans within the system.

5.3 Protocol Manager login

Protocol managers can access the system by logging in using their designated credentials. Upon successful login, they will be granted the necessary privileges to approve loans and manage group vaults created by group lenders.

5.4 Adding User Balance

In order to create a vault, users are required to add funds to their balance. These funds should be acquired by purchasing tokens that are pegged to BDT (Bangladeshi Taka) in exchange for Ethereum. It is important to note that users with insufficient balance will not be able to create new vaults or join existing group vaults. Sufficient funds in the appropriate tokens are a prerequisite for participating in these activities.

5.5 Adding Staking Balance

The process of adding staking balance is similar to adding user balance, with the distinction that it is specifically tracked as staking balance and cannot be utilized for staking loans. Once the staker has accumulated sufficient staking balance in tokens that are pegged to BDT, they can proceed to stake loans. It is important to note that the same staking balance can be used to stake multiple loans, as long as those loans do not default. The flexibility of utilizing the staking balance for multiple loans is contingent upon the loans being successfully repaid within the designated terms.

5.6 Creating Vaults

Within the system, users have the option to create and participate in two distinct types of vaults: individual vaults and group vaults. Users have the option to create individual vaults, subject to approval, where they can set interest rates and offer loans. Additionally, protocol managers initiate group vaults, determine the interest rates, and approve the participation of lenders in those vaults. The functionalities and procedures associated with each vault type are outlined as follows:

5.6.1 Individual Vault

In Figure 5.2, users who possess sufficient balance in their wallets can create individual vaults. Once these vaults are established, they must undergo approval by the protocol manager. The creator of the vault has the authority to determine the interest rate applicable to the vault. Following approval, borrowers can request and obtain loans from these individual vaults.

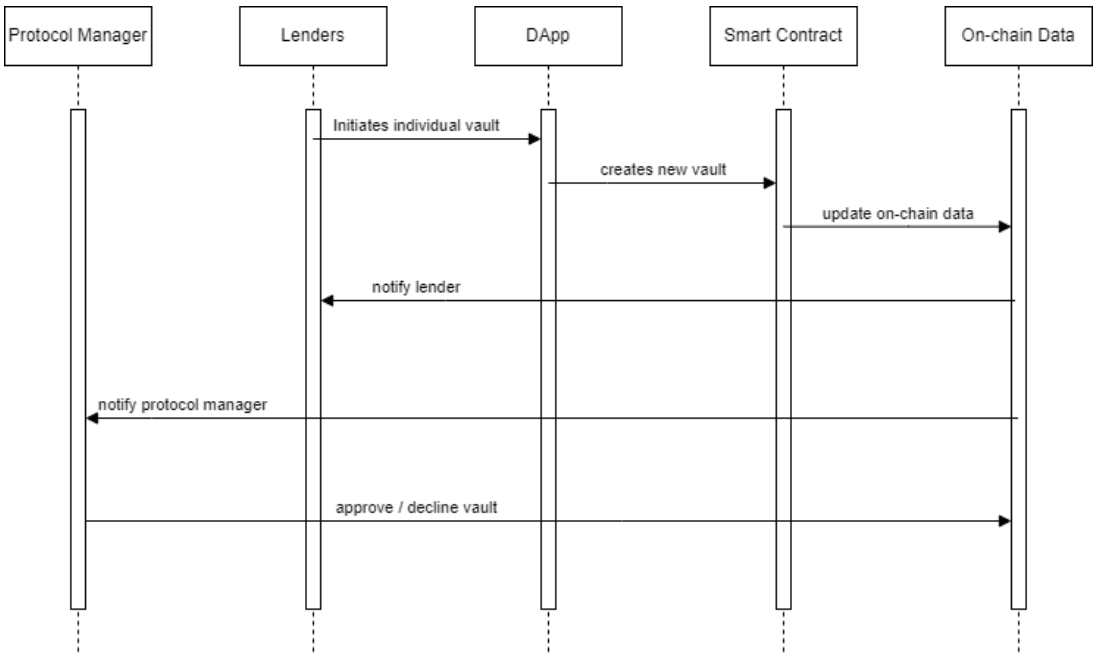

Figure 5.2: Individual Vault Creation.

5.6.2 Group Vault

In Figure 5.3, group vaults operate differently from individual vaults, as they are initiated by the protocol managers themselves. The protocol managers are responsible for setting the interest rates for these group vaults. Subsequently, lenders can participate in these vaults by contributing funds to the specific vault created by the protocol manager. The protocol managers retain the authority to approve these group vaults based on their assessment

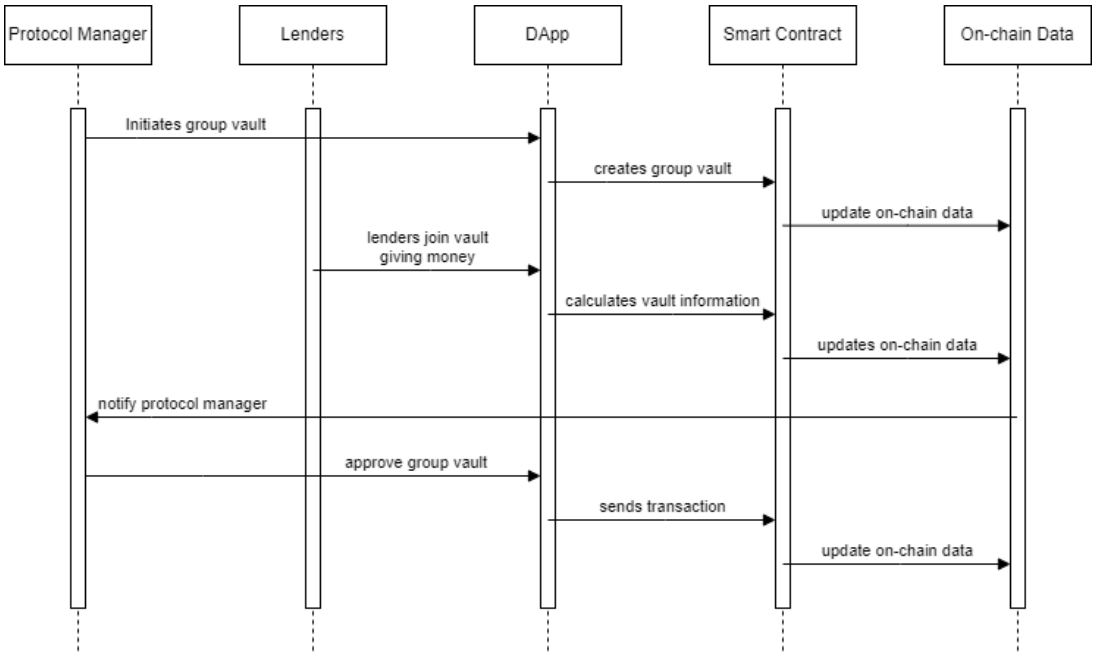

Figure 5.3: Initiating and Joining Group Vault.

5.7 Borrow Money

The borrowing process within the system offers two distinct avenues for borrowers to obtain funds, namely individual borrowing and group borrowing. Borrowers can pursue individual borrowing by requesting loans from individual or group vaults, with stakers playing a crucial role. Alternatively, group borrowing allows borrowers to join borrowing circles initiated by protocol managers, where collective responsibility and NID verification help maintain trust and discourage fraudulent activities. The functionalities and procedures for each approach are detailed below:

5.7.1 Individual Borrowing

In Figure 5.4, individual borrowers have the option to request loans from both individual vaults and group vaults. When making a loan request, a staker is required to stake the loan. The staker stakes the loan with the expectation of earning rewards from the system. Once the loan is staked, the protocol manager reviews and approves the loan, and the borrower receives the loan amount as a balance. The borrower can then choose to convert this balance into ether through a cashout process.

5.7.2 Group Borrowing

We depict group borrowing in Figure 5.5. Protocol managers initiate an empty borrowing circle, which serves as a unique feature within the system. Borrowers can join these circles to avail themselves of loans. The borrowing circle operates as a semi-autonomous system. If any member of the circle defaults on their loan, the remaining members are obligated to either make the repayment on behalf of the defaulter or face a decrease in their credit score. Borrowers who do not have an associated staker often choose this route to access loans. Notably, each account

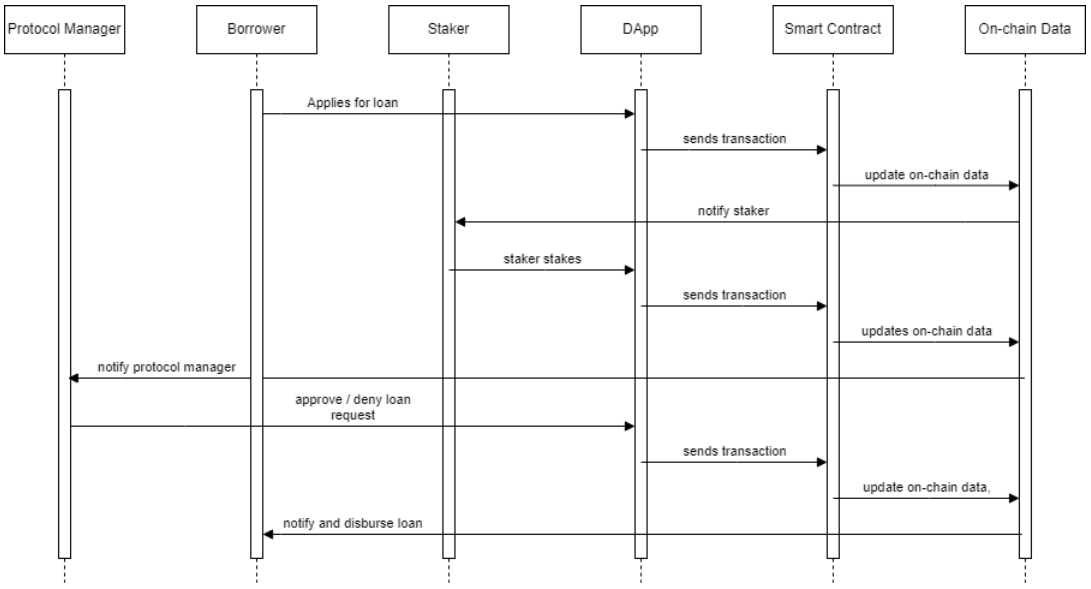

Figure 5.4: Individual Borrowing.

within the system is linked to a NID, thereby preventing fraudulent activities and maintaining transparency regarding user identity.

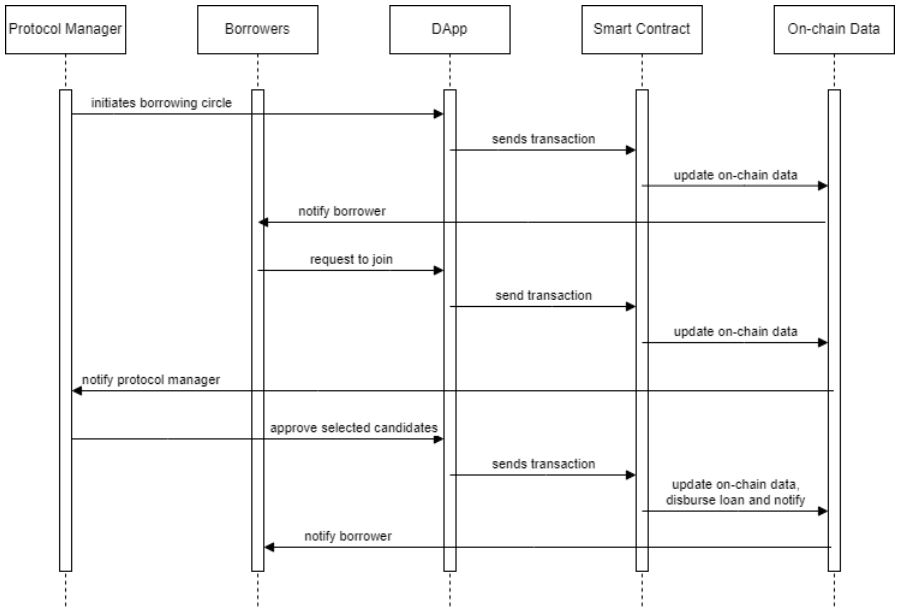

Figure 5.5: Initiating Group Borrowing

5.8 Repayment Of Loan

Loan repayment within the system necessitates the borrower transferring an amount equivalent to the loan they received, along with the accrued interest, back to the contract. Successful repayment of the loan leads to positive outcomes for both the borrower and the staker.

When a borrower successfully repays a loan, their credit score increases. This credit score serves as an indicator of the borrower's trustworthiness and responsible repay-

ment behavior within the system. A higher credit score enhances the borrower's credibility and improves their prospects for future loan availment and participation in the system.

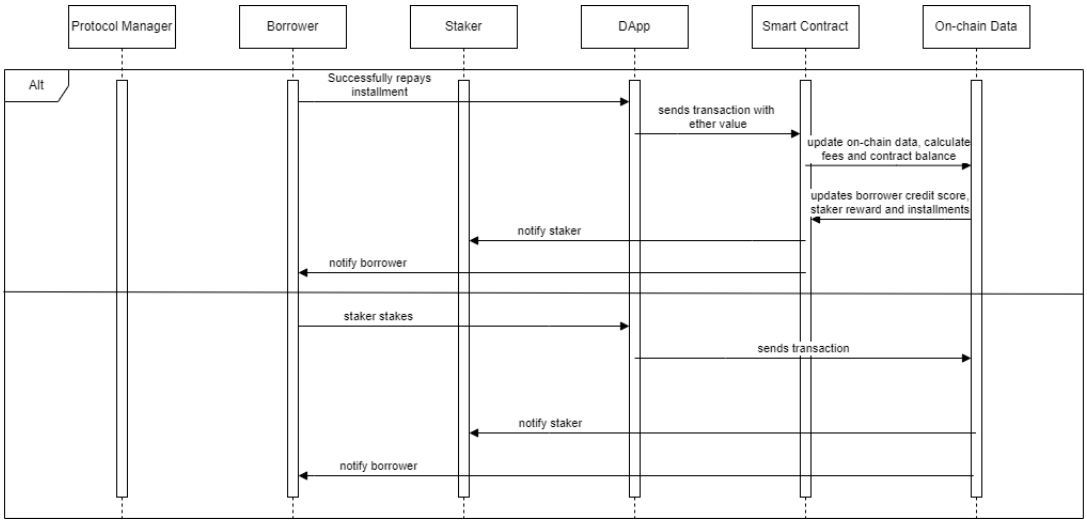

Figure 5.6: Loan repayment.

Additionally, according to the Figure: 5.6, the staker, who staked the loan with the expectation of earning rewards, gains reputation points and receives the corresponding rewards when the loan is repaid. These reputation points recognize the staker's contribution to the system and reflect their reliability in assessing and supporting loan requests. The rewards serve as an incentive for stakers to actively engage in the loan staking process, contributing to the overall functionality and sustainability of the system.
In summary, repayment of loans requires borrowers to transfer the loan amount along with interest to the contract. Successful repayment results in increased credit scores for borrowers and rewards, in terms of reputation points and incentives, for the stakers who supported the loan.

5.9 Updating Credit Score

Credit scores play a pivotal role in determining the loan eligibility and interest rates for individuals within the system. These scores are derived from a set of data points that provide insights into the borrower's financial history and repayment behavior. The following data are considered in the calculation of credit scores:

- **Total amount of loans taken till date:** This parameter assesses the cumulative value of loans that the individual has availed from the system over time. It provides an indication of the borrower's borrowing activities and the level of financial exposure.

- **Total amount of loan repaid:** The total amount of loans that the borrower has successfully repaid is taken into account. This factor reflects the borrower's track record of meeting their repayment obligations and demonstrates their reliability in honoring their financial commitments

- **Average repay amount per day:** This metric calculates the average daily repayment amount made by the borrower. It helps assess the borrower's consistency and regularity in repaying their loans, providing insights into their financial discipline.

- **Number of defaults to date:** This parameter considers the number of instances in which the borrower has defaulted on their loan repayments. Defaults indicate a failure to fulfill repayment obligations, raising concerns about the borrower's ability to manage debt and meet future payment commitments.

By analyzing these data points, the system generates credit scores for individual borrowers. These scores serve as a quantitative representation of the borrower's creditworthiness and assist in determining the loan amounts they can access and the corresponding interest rates. A higher credit score indicates a more favorable borrowing profile, while a lower score may result in more limited borrowing options or higher borrowing costs.
In summary, credit scores within the system are derived from a range of factors, including the total loan amount taken, the total amount repaid, the average daily repayment amount, and the number of defaults. These scores provide valuable insights into the borrower's borrowing history, repayment behavior, and overall creditworthiness, guiding the loan eligibility and interest rate determination processes.

5.10 Implementation

The thesis project encompasses several platforms, frameworks, and tools that are utilized for its development. These components provide a robust and efficient environment for implementing the proposed system. The key platforms, languages, and tools employed in the thesis are outlined as follows:

5.10.1 Development Platform and Language

I. **Smart Contract:** A self-executing contract having the conditions of the agreement encoded straight into code is known as a smart contract. It runs on a blockchain platform, like Ethereum, and enforces the agreed-upon norms and conditions automatically without the need for intermediaries.

   - **Solidity:** The computer language Solidity is used to create smart contracts on the Ethereum network. It offers the syntax and capabilities required to specify the contracts' behavior and logic.
   - **Remix online IDE:** Remix is an online Integrated Development Environment (IDE) specifically designed for developing and testing smart contracts. It offers a user-friendly interface and a range of debugging and testing features.

II. **Truffle Framework:** The Truffle framework is employed for the development, testing, and deployment of smart contracts. It provides a suite of development tools, including a development environment, testing framework, and build pipeline, to streamline the smart contract development process.

III. **Ganache:** Ganache serves as a local blockchain environment for testing and development purposes. It allows developers to simulate a private Ethereum network, enabling rapid testing and debugging of smart contracts without the need for deploying them on the live network.

IV. **Metamask:** Metamask is a browser wallet extension that enables users to interact with the Ethereum blockchain through their web browsers. It provides a secure and convenient way to manage accounts, sign transactions, and interact with decentralized applications (DApps).

V. **Next.js:** Next.js is a popular React framework used for building server-side rendered (SSR) and static websites. It offers a range of features, such as efficient routing, server-side rendering, and static site generation, that enhance the performance and user experience of the web application.

VI. **Express.js:** Express.js is a backend framework built with Node.js that facilitates the development of web applications and APIs. It provides a minimal and flexible approach to building server-side functionality and handling HTTP requests and responses.

VII. **MongoDB:** MongoDB is employed as a NoSQL database for storing and managing data in the thesis project. It offers scalability, flexibility, and a document-oriented data model, making it suitable for handling the system's data requirements.

VIII. **VS Code:** Visual Studio Code (VS Code) serves as the integrated development environment (IDE) for writing, editing, and managing the project's source code. It provides a range of features and extensions that enhance productivity and facilitate collaboration.

IX. **Git and GitHub:** Git is a distributed version control system utilized for tracking changes, managing source code, and facilitating collaboration among developers. GitHub, a web-based hosting service for Git repositories, is used as a platform for version control, code sharing, and collaboration on the thesis project.

By leveraging these platforms, frameworks, and tools, the thesis project aims to create a robust and efficient development environment for implementing the proposed system. These components provide the necessary infrastructure and functionality to develop smart contracts, interact with the Ethereum blockchain, build web applications, and manage the project's source code effectively.

5.10.2 Platform Features based on The Protocol

In this section we will discuss and show the features in the platform that are built based on the protocol.

- **User Sign up:** In the Figure 5.7 we can see users can sign up using their metamask wallet address and private key and its Algorithm: 1 is also given.

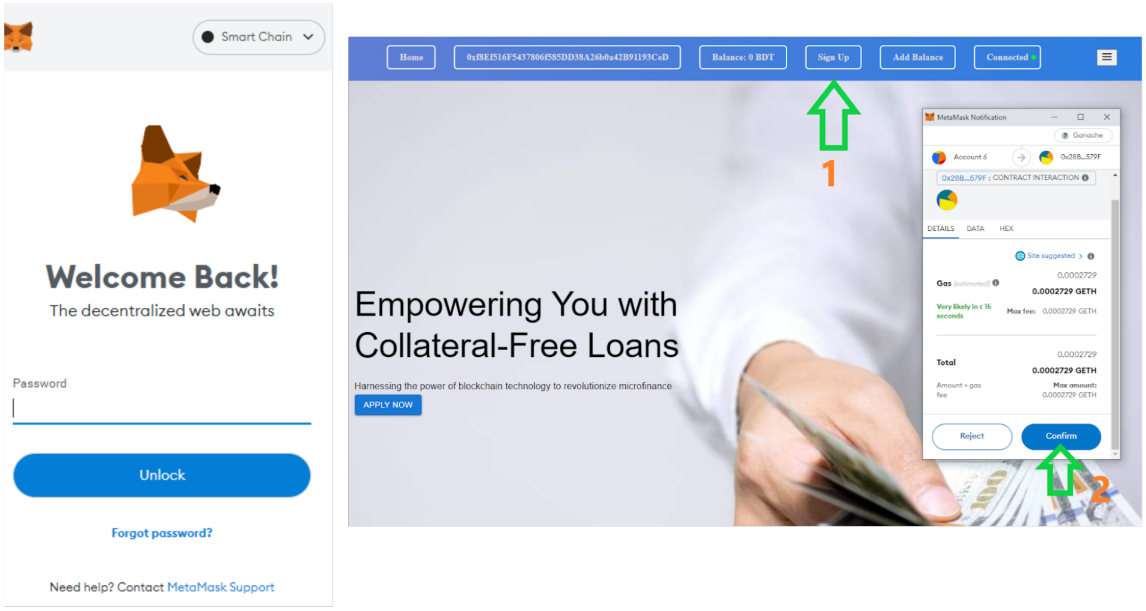

Figure 5.7: User signup using Metamask

- **Verification:** In Figure 5.8 we can see users can verify their account using their NID information. The backend server verifies the NID and sends back a verification token to the Smart Contract. Smart contract stores the verification information corresponding to that particular hexadecimal address. The whole process of verification and signup is represented in Algorithm 1. We have based the algorithm on the protocol flow represented in Figure 5.1.

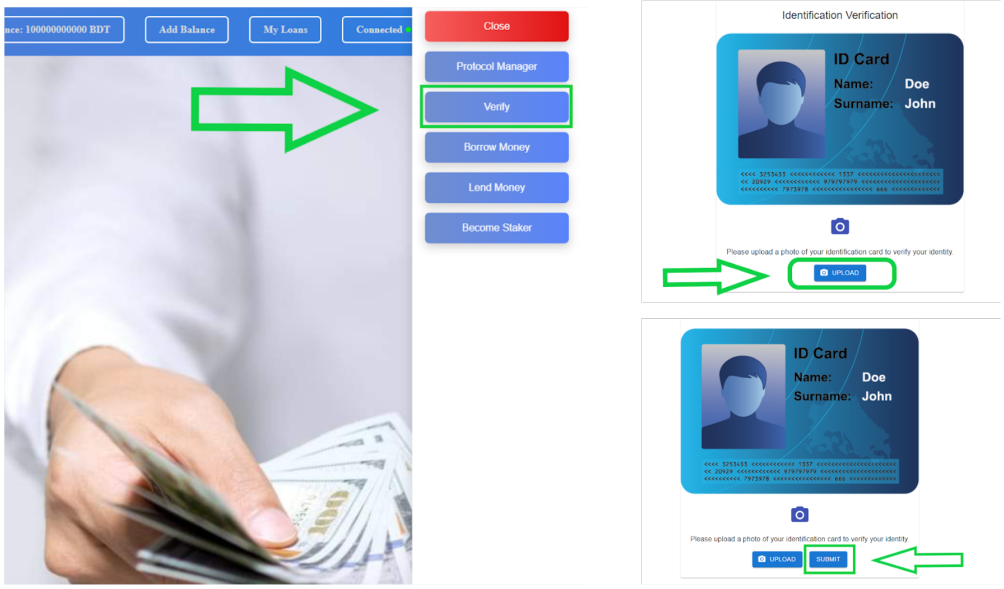

Figure 5.8: User Verification:

Algorithm 1: Signup Algorithm

```
   Input: req → request to register from user
 1 Function
 2 |  initiate(request)
 3 end
 4 data := request.data;
 5 type := request.type;
 6 if request.type == login then
 7 |  response := loginFunction(data);
 8 else if request.type == registration then
 9 |  response := regFunc(data);
10 end
11 send response back to user;
12 Function
13 |  regFunc(userInfo, biometricData, NIDdata)
14 end
15 if userInfo == null or userInfo.email == null or userInfo.password ==
     null then
16 |  return "Error: Invalid user info";
17 user := getUser(userInfo.email);
18 if user != null then
19 |  return "Error: User is already registered";
20 NIDverificationStatus := verifyNID(NIDdata);
21 if NIDverificationStatus != "Verified" then
22 |  return "Error: NID verification failed";
23 biometricVerificationStatus := verifyBiometric(biometricData);
24 if biometricVerificationStatus != "Verified" then
25 |  return "Error: Biometric verification failed";
26 userID := generateUniqueID();
27 user := createUser(userID, userInfo.email, userInfo.password, NIDdata,
     biometricData, NIDverificationStatus, biometricVerificationStatus);
28 if user == null then
29 |  return "Error: Unable to register user";
30 storeUserData(user);
31 return "Registration successful. Your user ID is " + userID;
```

- **Become a Staker** Anyone can become a Staker by registering as Staker and NID verification. Once they are verified they can add staking balance for staking any loan. Algorithm 2 is based on this idea of becoming a staker.

- **Protocol Manager login:** Protocol manager addresses are defined by the contract and they can approve vaults and loans, initiate group vaults after logging in.

- **Adding User Balance:** Users can add balance by adding the BDT amount and that amount is automatically converted into ether for a successful transaction. In the figure 5.9 we can see a user can add balance. Algorithm 3 represents this. It allows the users If the user logs in as a general "user," the protocol increases the balance of the user's address by the requested amount.

Algorithm 2: Become Staker

Require: *is_nid_verified*: a boolean indicating whether the staker's NID is verified

function BECOMESTAKER(*_is_nid_verified*)
    stakerAddress ← msg.sender
    total_amount_staked ← 0
    reputation_score ← 100
    is_nid_verified ← _is_nid_verified
    balance ← 0
    reward ← 0
    referred_borrower ← create new empty address array with size 10
    Create a new Staker with the above values
    Assign the new Staker to *address_staker* mapping using *stakerAddress* as the key
end function

On the other hand, if the user type is "staker," the protocol fetches the particular staker object which is connected to the user's address. Then the staker's balance increases by the funding amount added to the protocol.

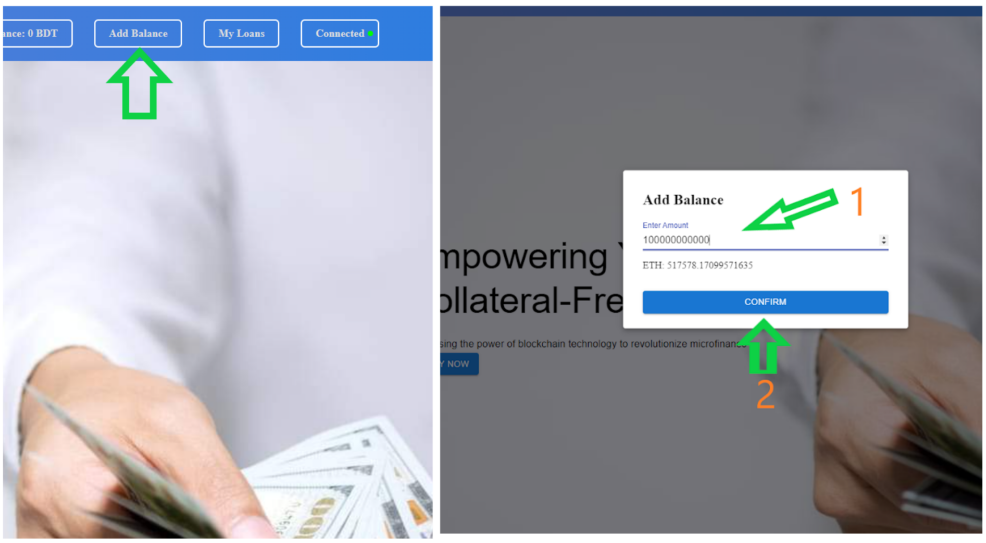

Figure 5.9: Adding balance

- **Create Vault:** As mentioned earlier, there are two types of vaults. Let's have a look at how each of them works

    I **Creating Individual Vault:** Lenders can create their individual vaults and set the interest rate on their own. After it is created, they remain in a pending state until they are approved by a protocol manager. Algorithm 4 gives an overview of this idea.

    II **Creating Group Vault:** In a group vault, the protocol manager initiates the vault and interest rate of the vaults. After that, the lenders

Algorithm 3: Add Balance

function ADD_BALANCE($user\_type$)
    $amount \leftarrow$ msg.value
    if $user\_type =$ "user"
    $address\_user$[msg.sender].$balance$ += $amount$
    if $user\_type =$ "staker"
    $staker \leftarrow address\_staker$[msg.sender]
    $staker.balance$ += $amount$
end function

Algorithm 4: Creating Individual Vault

function CREATE_INDIVIDUAL_VAULT($total\_amount, interest\_rate$)
    $stakerAddress \leftarrow$ msg.sender
    $userBalance \leftarrow$ address_user[$stakerAddress$].$balance$
    $isNIDVerified \leftarrow$ address_user[$stakerAddress$].$is\_nid\_verified$
Require: $userBalance \geq total\_amount$
    if $userBalance < total\_amount$ **then**
    return "Insufficient balance"
Require: $isNIDVerified$
    if $isNIDVerified =$ false **then**
    return "NID is not verified"
    $updatedUserBalance \leftarrow userBalance - total\_amount$
    $vaultID \leftarrow$ last_vault_id + 1
    Increment last_vault_id by 1
    $vault \leftarrow$ vaultId_vault[$vaultID$]
    $vault.vault\_owner \leftarrow stakerAddress$
    $vault.vault\_id \leftarrow vaultID$
    $vault.total\_supply \leftarrow total\_amount$
    $vault.remaining\_supply \leftarrow total\_amount$
    $vault.interest\_rate \leftarrow interest\_rate$
    $vault.status \leftarrow$ Status.pending
    Update address_user[$stakerAddress$].$balance$ with $updatedUserBalance$
end function

can join those vaults by giving their investment amount. Algorithm 5 represents the idea of protocol manager initiating a group vault.

Algorithm 5: Initiate Group Vault

function INITIATE_GROUP_VAULT(*interest_rate*)
    protocolManager ← msg.sender
    groupVaultID ← last_group_vault_id + 1
    Increment last_group_vault_id by 1
    groupVault ← groupVaultId_vault[*groupVaultID*]
    groupVault.protocol_manager ← *protocolManager*
    groupVault.interest_rate ← *interest_rate*
    groupVault.status ← Status.pending
    groupVault.vault_id ← *groupVaultID*
end function

- **Approving Vaults:**

  I. **Individual Vaults:** Once the vaults are created by an individual lender, it is set up for approval by the protocol manager. The protocol manager oversees the overall vault condition and approves the vault.

  II. **Group Vaults:** In the case of a group vault, the protocol manager oversees the overall vault condition and approves the vault after significant funds are raised. Algorithm 6 shows a lender joining a group vault.

Algorithm 6: Join Group Vault

function JOIN_GROUP_VAULT(*_contribution*, *_vault_id*)
    stakerAddress ← msg.sender
    userBalance ← address_user[*stakerAddress*].*balance*
    isNIDVerified ← address_user[*stakerAddress*].*is_nid_verified*
Require: *userBalance* ≥ *_contribution* **if** *userBalance* < *_contribution* **then**
    return "Insufficient balance"

Require: *isNIDVerified* **if** *isNIDVerified* = false **then**
    return "NID is not verified"

    groupVault ← groupVaultId_vault[*_vault_id*]
    updatedTotalSupply ← *groupVault.total_supply* + *_contribution*
    groupVault.total_supply ← *updatedTotalSupply*
    updatedUserBalance ← *userBalance* − *_contribution*
    address_user[*stakerAddress*].*balance* ← *updatedUserBalance*
    contribution ← Contribution({vault_id : *_vault_id*, member : *stakerAddress*, contribution : *_contribution*})
    groupVaultId_contribution[*_vault_id*].push(*contribution*)
end function

- **Borrowing from Vaults:**

I. **Individual Borrow:** Individual borrowers have the option to request loans from the vaults available within the system. Upon initiating a loan request, the relevant stakeholders, particularly the staker, are notified about the loan details and conditions. If a staker decides to stake the loan, the request is then forwarded to the protocol manager for approval. Once the protocol manager grants approval, the borrower is provided with the loan amount as a balance. This balance represents a token that is pegged to the Bangladeshi Taka (BDT). Subsequently, the borrower has the flexibility to convert this balance into ether, a cryptocurrency, and utilize it according to their requirements and preferences. Algorithm 7 refers to this idea of individual borrowing.

Algorithm 7: Individual Borrow

1: **function** INDIVIDUAL_BORROW(*_vault_id*, *_vault_type*, *_amount*, *_each_installment_amount*, *_no_of_installments*, *_each_term*) *_vault_id* (uint256) - the ID of the vault
2: *_vault_type* (string) - the type of vault
3: *_amount* (uint256) - the amount of the loan
4: *_each_installment_amount* (uint256) - the amount of each installment
5: *_no_of_installments* (uint256) - the total number of installments
6: *_each_term* (uint256) - the duration of each term

7: **Require:** The loan amount does not exceed the limit of 100000 Taka.
Increment last_loan_id by 1 and assign it to loan_id.
Create a new Loan struct named loan with the following properties:
- borrower: set as msg.sender
- amount: set as *_amount*
- status: set as Status.pending
- vault_id: set as *_vault_id*
- vault_type: set as *_vault_type*
- no_of_installments: set as *_no_of_installments*
- each_installment_amount: set as *_each_installment_amount*
- each_term: set as *_each_term*
- staker: set as address(0) or a suitable default value
- start_date: set as 0 or a suitable default value
- borrowing_group_id: set as 0 or a suitable default value
- no_of_installments_done: set as 0 or a suitable default value
- next_term_due_date: set as 0 or a suitable default value

Assign the loan struct to loanId_loan.
8: **end function**

II. **Group Borrow:** Borrowers have the opportunity to borrow funds by participating in groups formed by the Protocol Manager. Upon the approval of the group borrowing, each individual within the group receives their allocated portion of the funds as determined within the borrowing group structure.

- **Staking Loans:** Stakers engage in the process of staking loans that are awaiting approval from the protocol manager. After staking a loan, it is subsequently reviewed and approved by the protocol manager. Figure 5.10 shows staker staking loans.

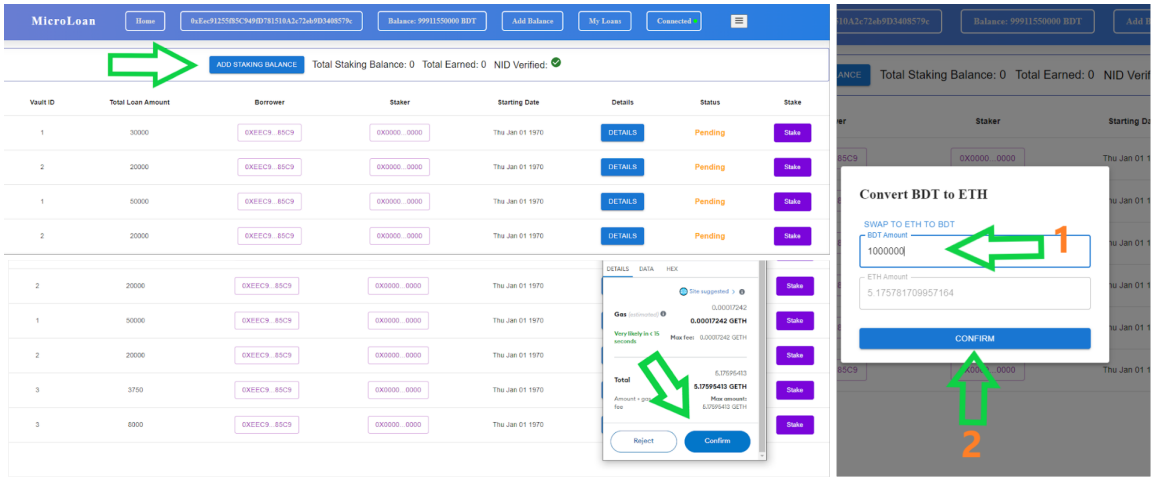

Figure 5.10: Staking Loans

- **Approving Loans:** The protocol manager approves the loans based on the aforementioned conditions.

- **Loan Repayment:** As the installment time of the repayment arrives, the borrowers are notified of their state of installment and are prompted to repay their loans on time. In Figure 5.11, we can see a borrower repaying his loan. Algorithm 8 refers to this process of repayment of loans. Here, loan ID and vault type are required input parameters. At first, the installment amount is detected and the initialization of variables is done. Then the vault type is decided. For private vaults (type 0), calculations are made on capital amounts, how much money is left in the vault for borrow, how much interest was made for staking, and how much money the staker had to pay as fees. The staker is rewarded borrower's credit score is updated. Similarly, calculations are done on group vaults(type 1) after the vault data is retrieved from different storage.

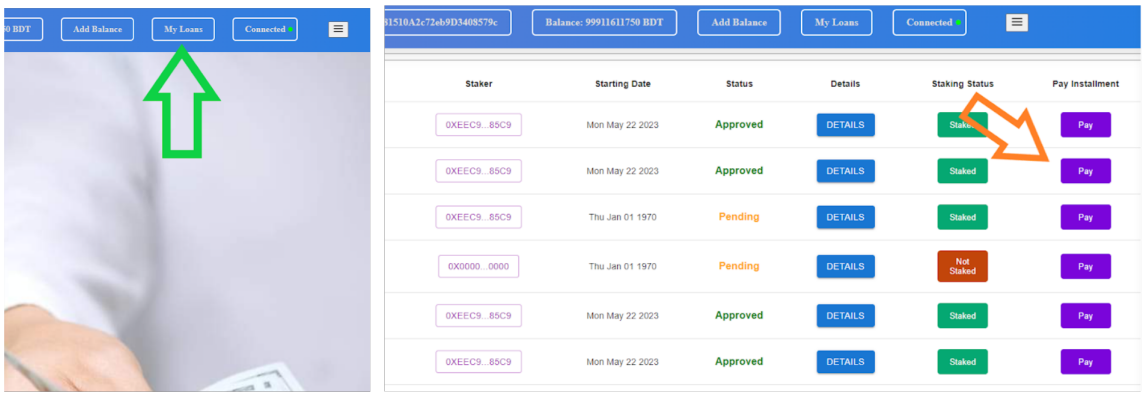

Figure 5.11: Loan Repayment

Algorithm 8: Loan Repayment

function LOAN_REPAYMENT($loan\_id, vault\_type$)
  $installment\_amount \leftarrow loanId\_loan[loan\_id].each\_installment\_amount$
  $interest\_earned, staker\_fee \leftarrow 0$
  $vault\_type \leftarrow detect\_vault()$
  If $vault\_type = 0$
  $vault\_id \leftarrow loanId\_loan[loan\_id].vault\_id$
  $interest\_rate \leftarrow vaultId\_vault[vault\_id].interest\_rate$
Require: $msg.value \geq installment\_amount$
  $no\_of\_installments\_done \leftarrow no\_of\_installments\_done + 1$
  $capital\_amount \leftarrow breakdown\_repayment()$
  $vault.remaining\_supply \leftarrow vault.remaining\_supply + capital\_amount$
  $interest\_earned \leftarrow installment\_amount - capital\_amount$
  $staker\_fee \leftarrow calculate\_fees()$
  $vault.interest\_earned \leftarrow vault.interest\_earned - staker\_fee$
  $update\_credit\_score()$
  $address\_staker[loanId\_loan[loan\_id].staker].reward \leftarrow staker\_fee$
  Else if $vault\_type = 1$
  $vault\_id \leftarrow loanId\_loan[loan\_id].vault\_id$
  $interest\_rate \leftarrow groupVaultId\_vault[vault\_id].interest\_rate$
Require: $msg.value \geq installment\_amount$
  $no\_of\_installments\_done \leftarrow no\_of\_installments\_done + 1$
  $capital\_amount \leftarrow breakdown\_repayment()$
  $vault.remaining\_supply \leftarrow vault.remaining\_supply + capital\_amount$
  $interest\_earned \leftarrow installment\_amount - capital\_amount$
  $staker\_fee \leftarrow calculate\_fees()$
  $vault.interest\_earned \leftarrow vault.interest\_earned - staker\_fee$
  $update\_credit\_score()$
  $address\_staker[loanId\_loan[loan\_id].staker].reward \leftarrow staker\_fee$

end function

- **Credit Score Update:** Upon the successful repayment of the loan install-

ment along with the accrued interest, the borrower experiences an increase in their credit score. This credit score enhancement serves as a measure of their creditworthiness and has implications for their future financial privileges. Additionally, the staker who participated in staking the loan is rewarded and earns reputation points, which hold significance in terms of their standing and eligibility for potential future benefits within the system. Algorithm 9 represents the idea of updating credit score.

Algorithm 9: Update Credit Score

function UPDATECREDITSCORE(user_address)
    user_storage ← address_user[*user_address*]
    total_loans_taken ← 0
    total_loan_repaid ← 0
    timely_repayments ← 0
    total_repayment_days ← 0
    defaults ← 0
    time_elapsed ← 0
    avg_repayment_per_day ← 0
    for $i \leftarrow 0$ *to length*(*user_storage.loan_id*) **do**
    loan_storage ← loanId_loan[*user_storage.loan_id*[*i*]]
    total_loan_repaid ← total_loan_repaid + loan_storage.installment_amount × loan_storage.num_installments_done
    total_repayment_days ← total_repayment_days + $\frac{\text{loan\_storage.installment\_term}}{86400}$ × loan_storage.num_installments_done
    updated_credit_score ← CalculateUpdatedCreditScore(total_loans_taken, total_loan_repaid, timely_repayments, total_repayment_days, defaults, time_elapsed, avg_repayment_per_day)
    return updated_credit_score
end function

Chapter 6

Discussion and Evaluation

6.1 Analyzing Requirements

Here, we investigate if our MFI DApp meets the set requirements of Section Requirement Analysis of Chapter 2.

6.1.1 Functional Requirements

- Our DApp enables lenders, borrowers, and stakers to carry out balance queries and transfer amounts from their wallets to the platform, thus fulfilling F1.
- The NID is used to verify the identity of the user via a 3rd party server that sends back a token. That is added to the blockchain, which satisfies F2.
- We gave a terms and conditions page for stakers. Thus solving F3.
- Each user's financial information is stored in the blockchain, creating a secure and immutable ledger. We collect data manually at first, then upload them to the blockchain, and after that, we will create a central database. Thus solving F4.
- We implement the DApp to create and record data between the borrower and our MFI. Thus solving F5.

6.1.2 Security Requirements

- The DApp protocol requires every user to be enlisted in the platform and authenticated before they submit any transactions. This ensures S1.
- A secure and detached session is kept for each user who login to create separation. Thus, one user can't access another user's information, thereby satisfying S2.
- We used HTTS to protect data from the user, DApp, and protocol flow. The user's private key is used to digitally signs all requests except the request for registration. It is approved if the digital signature is only validated. These are kept immutable as the log-in session, and requests are updated and stored in the blockchain. These acts meet S3.

- Ethereum is distributed blockchain platform which has so many nodes. Thus it natively is protected against any DoS attack. As the Dapp is built on top of this platform, it satisfies S4.
- One-time values (nonces) have been used all over the place in our protocol. This protects us from any replay attack, thus satisfying S5.

6.2 Cost Analysis

6.2.1 Analyzing the Cost of Different Use-cases

We analyze the incurred cost for each activity on our DApp. As we have built the protocol on the Ethereum blockchain, we calculate the cost of the transaction fee for each activity, e.g. registration, loan application, approval, disbursement, and so on.
According to [7], Ethereum transactions are generally determined by Gas prices - the unit measuring the amount of computational effort needed to perform specific procedures on the Ethereum blockchain network. Here Smart contract deployment and operations need gas. Miners [57] get ether-based gas for their processing power and smart contract computations. EVM [44] opcodes determine smart contract gas requirements.
This is how Ethereum transaction fees are calculated:

$$\text{Gas Price} \times \text{Gas Amount} = \text{Gas Fee}$$

The gas amount is measured in 'units'. By default, the Ethereum gas limit is set at 21,000 units in Metamask wallet. This limit can be modified in the respective Ethereum wallets. The quantity of Ethereum to be paid to ship a transaction to the Ethereum blockchain is the 'Gas Price'. It's unit is 'gwei'. One 'gwei' equals one-billionth of one Ethereum or 0.000000001 Ethereum [45]. If we assume 1 Ethereum = 2,000 USD then the 1 gwei = 0.000002 USD. Ethereum's gas price fee varies based on total network traffic. The price is decreased when there is little network activity [65]. We used this formula of calculating transaction fees to determine how much each operation will cost in the DApp considering the conversation rate of 1500$ per ETH. These operations are:

- Signup
- Verify
- Add balance
- Create individual vaults
- Join group vaults
- Individual borrowing
- Create group vault
- Approve individual vault

Table 6.1: Ethreum Transaction cost for different use-cases

Transaction	Gas Price (gwei)	Gas Amount	Gas Fee (ETH)	Gas Fee (USD)	Gas Fee (BDT)	Conversion Rate
Signup	20	17860	0.0003572	0.5358	57.24	$1500/ETH
Verify	20	15860	0.0003172	0.4758	50.98	$1500/ETH
Add balance	20	26800	0.000536	0.804	86.86	$1500/ETH
Create individual vault	20	84134.8	0.0016827	2.524	270.78	$1500/ETH
Join Group Vault	20	79420	0.0015884	2.3826	255.18	$1500/ETH
Individual Borrowing	20	107887.6	0.0021578	3.2367	346.74	$1500/ETH
Create group vault	20	63507.6	0.0012701	1.9052	204.00	$1500/ETH
Approve individual vault	20	40900	0.000818	1.227	131.33	$1500/ETH
Approve group vault	20	41284	0.0008257	1.2386	132.28	$1500/ETH
Approve loan	20	65760	0.0013152	1.9728	211.26	$1500/ETH
Become staker	20	17860	0.0003572	0.5358	57.24	$1500/ETH
Add staking balance	20	17800	0.000356	0.534	57.05	$1500/ETH
Stake	20	51240	0.0010248	1.5372	164.29	$1500/ETH

Table 6.2: Polygon Transaction cost for different use-cases

Transaction	Gas Price (gwei)	Gas Amount	Gas Fee (MATIC)	Gas Fee (USD)	Gas Fee (BDT)	Conversion Rate
Signup	20	17,860	0.3572	0.3287	35.19	$0.92/MATIC
Verify	20	15,860	0.3172	0.2918	31.23	$0.92/MATIC
Add balance	20	26,800	0.536	0.4931	52.77	$0.92/MATIC
Create individual vault	20	84,134.8	1.6827	1.5474	165.74	$0.92/MATIC
Join Group Vault	20	79,420	1.5884	1.4612	156.43	$0.92/MATIC
Individual Borrowing	20	107,887.6	2.1578	1.9863	212.62	$0.92/MATIC
Create group vault	20	63,507.6	1.2701	1.1694	124.96	$0.92/MATIC
Approve individual vault	20	40,900	0.818	0.7526	80.72	$0.92/MATIC
Approve group vault	20	41,284	0.8257	0.7601	81.34	$0.92/MATIC
Approve loan	20	65,760	1.3152	1.2093	129.71	$0.92/MATIC
Become staker	20	17,860	0.3572	0.3287	35.19	$0.92/MATIC
Add staking balance	20	17,800	0.356	0.3275	35.06	$0.92/MATIC
Stake	20	51,240	1.0248	0.9435	101.07	$0.92/MATIC

- Approve group vault
- Approve loan
- Become staker
- Add staking balance
- Stake

Table 6.1 refers to all the use-cases of DApp where a transaction is incurred. 20 units of gwei are used as the supposed standard for the platform to measure gas price. The gas amount is set by the Metamask wallet. Then the calculation is made for each transaction, with prices in ETH, USD, and BDT using the formula mentioned in this section for transaction fees. As we plan to move to Polygon [59] as proposed in the section 6.9, we also would like to represent the transaction costs that might occur for a shift into the layer 2 platform in the Table 6.2.
Here Table 6.3 represents the cost calculation to deploy the smart contract of the DApp. Gas price is taken as 15 gwei and as such total cost of contract deployment in the Ethereum blockchain comes out as 6,612.6 BDT. An argument can be made

Table 6.3: Cost of deploying the smart contract

Description	Calculation	Amount
Gas Used for Transaction		2,746,558 gas
Gas Price		15 gwei/gas
Cost in Gwei	2,746,558 gas $\times$15 gwei/gas	41,198,370 gwei
Cost in ETH	41,198,370 gwei $\div 1e9$ gwei/ETH	0.04119837 ETH
Exchange Rate (ETH to USD)		1,500 USD per ETH
Cost in USD	0.04119837 ETH $\times 1,500$ USD/ETH	61.80 USD
Exchange Rate (USD to BDT)		107 BDT per USD
Cost in BDT	61.80 USD $\times$107 BDT/USD	6,612.6 BDT

for the volatile gas prices, but the contract deployment is anticipated to be done when the network is not busy.

6.2.2 Additional Business Costs:

We expect that our DApp will save the borrower significant time getting loan approvals while reducing the overhead costs of running an MFI. Because of its decentralized nature, people will be able to lend their money in the DApp.
Conservative estimates of the tangible value of the DApp include the following:

- **Employee Cost:** We are looking to reduce the cost of running an MFI significantly. As of April 2023, Grameen Bank alone has 21,016 employees [48]. Generally, it's seen that employees earn from 20,000 - 40,000 BDT per month from MFI jobs in Bangladesh [37]. If we consider the salary per employee as 20,000 BDT or 186.56 USD per month at the current exchange rate [66], which is the lowest amount seen in the circulars, and multiply it by the current total employees of Grameen Bank and 12 (for the months of a year), we get:

  $$186.56 \times 21,016 \times 12 = 47,033,808$$

  That's what Grameen Bank pays its employees per month in USD at the very least at the current exchange rate [66]. And that's without the overhead cost of managing offices and documents. According to the 2022 annual report of the Bangladesh Microcredit Regulatory Authority, there are about 791 MFIs with 207,000 total employees besides Grameen Bank [56]. Paying them annually costs about 463 Million USD considering an average salary of 20,000 BDT per employee.
  For the project, as the basic structure and implementations are already in place, a minimal amount of employees would be required for the upkeep. Thus produces a cost structure of extreme efficiency. The charge of hiring blockchain developers is typically between 81 - 100 USD per hour on average [39]. So, the cost of hiring a developer would be tantamount to 192,000 USD per year considering they work for 8 hours for 5 days a week. Ideally, a development team with a minimum of 5 members is essential to reduce risk [25] which would cost about 960,000 USD. According to [46], average branch managers net 207,000 BDT per month in Bangladesh. Hiring 40 branch managers per zone following the Grameen Bank model [48] would cost only 926,800 USD

per year in current exchange rates [66]. So in total, the protocol would need in total for the monetary sum:

$$960,000 + 926,800 = 1,888,800$$

That should be the cost to hire people for one year which is a 99.59% decrease in employee cost, down from 463 Million USD to 1.8888 Million USD per month. This is obviously a hypothetical number but gives us an understanding of how much cost in employee fees the deployed DApp can save the MFI industry in Bangladesh.

- **Office Overhead:** MFIs require a lot of office space to run and host employees. For example, only for Grameen Bank, there are 40 zones with offices in each place, along with 40 audit offices for the zones. Additionally, there are area offices in 240 places, along with 2568 office branches [48]. No physical offices are required for our system, as verification and client onboarding is done completely online, saving money and space.

6.3 Credit Score and Social Collateral

Microfinance employs a collective lending strategy, where loans are distributed to a group of individuals. This strategy banks on the concept of "social collateral," which allows beneficiaries to access financial services without needing physical collateral [1]. This concept ensures that recipients, even those with limited or no financial collateral, can repay their loans. Social collateral is established by forming a group of recipients, who then feel a societal obligation to repay the loan. Group members share the responsibility for repaying the loan. In a microfinance program, every member must repay their current loan to be eligible for future loans. Group members are also expected to supervise each other and exert social pressure, ensuring that each member repays their loan [2].
In our protocol, we have the concept of creating borrowing circles for group borrowing. Based on the performance of the individuals in the group borrowing circle, everyone earns a credit score. For example, in a borrowing circle if any individual borrower defaults on any loan, then two things can happen.

6.3.1 Mutual Installment Compensation

In this scenario, the other members of the group will pay the installments of the defaulted loan and that amount will be repaid by the actual borrower at any time in the future. This way all the members sustain their credit score and create the trust in that group even stronger.

6.3.2 Eliminating the Borrower

In this scenario, the borrower who defaults on any installment will be removed from the group and lose a significant credit score which will adversely affect their future availability of financial services since these credit scores are accessible by any financial service provider. This negative effect on their creditworthiness will likely impede future access to financial services, given that credit scores are accessible

to any financial service provider. This way they remain obligated to repay their defaulted installments in the future.
As we can see, it's not easy to escape the obligations the users have agreed to follow. Even though the platform does not have any collateral for giving loans to its borrower, it uses the concept of social collateral where the borrowers are obligated to repay their loan to avoid being marked as 'less creditworthy person'. This whole process is carried out using the power of smart contracts where each and every data can be tracked easily thus ensuring transparency.

6.4 Easing Data Management System:

Traditional MFIs require a lot of manual input for upkeep. Grameen Bank has 240 Information Management Centres (IMCs) with 836 computers serving 2568 branches, each serving 3 branches only [38]. This cluster of computers serves to reduce paperwork at the grassroots level. But with the use of the Ethereum blockchain, this kind of cluster would not be required to serve the clients. All the information would be synced into the Ethereum blockchain and it could be pulled up at any moment.

6.5 Saving Time for Employees and Clients:

In Bangladesh, it takes 15-30 days to disburse the microcredit loans because of the bureaucratic structure [31]. Considering the lower end of the spectrum, even if each transaction took only 15 days to process, the current 10.3 million borrowing members of Grameen Bank [48] had to wait a combined 424230 years approximately to take out microcredit loans, which is very inefficient. Using blockchain technology solves this. Transactions in Ethereum are processed in a median of 57 seconds and that 90% of them are down within 8 minutes [35]. Even if each microcrdit transaction take 8 minutes, it would have taken clients approximately 157 years combined to take loans in the Dapp, which is approximately a 99.96% reduction in transaction processing time.

6.6 Analysing Research Objectives

In this research, we set out to address the challenges and limitations of traditional microfinance models by leveraging the potential of blockchain technology. The primary objective of this study was to detect gaps in the current microcredit system and develop a protocol for an uncollateralized blockchain-based microfinance system that would provide borrowers with access to loans without the need for traditional collateral requirements while ensuring the security, transparency, and efficiency of the lending process. We used Bangladesh's MFI industry as an example to show how the implemented model can help the microcredit industry.
Throughout the research process, we have successfully achieved the following objectives:

6.6.1 Designed and Developed the Protocol

We have architectured and designed a protocol that forms the backbone of the proposed uncollateralized microfinance system. This protocol enables borrowers to request loans, stakers to stake loans, and protocol managers to approve loans, all in a decentralized manner using smart contracts on the Ethereum blockchain.

6.6.2 Increased Accessibility and Efficiency

The developed protocol enhances accessibility to microfinance services by leveraging blockchain's decentralized nature, eliminating the need for intermediaries, and reducing transaction fees. This opens up opportunities for individuals who were previously excluded from traditional financial systems to participate in the microfinance ecosystem.

6.6.3 Improved Transparency and Trust

The protocol's utilization of blockchain technology ensures transparency and immutability of loan records, making the entire lending process transparent and auditable. This enhances trust among all stakeholders, including borrowers, stakers, and protocol managers, leading to a more reliable and accountable microfinance ecosystem.

6.6.4 Cost Reduction and Scalability

By leveraging the efficiency of blockchain technology, our protocol reduces administrative overhead, eliminates intermediaries, and minimizes transaction fees. This cost reduction enables the microfinance system to reach more borrowers while maintaining its scalability and sustainability.

6.7 Advantages

Our DApp platform has got have certain advantages, as given below:

- The DApp is the first fleshed-out system to integrate the idea of microfinance and blockchain with an easily implementable algorithm and protocol flow. This enables marginal people to borrow money in a secure manner without collateral.
- The DApp will serve as the guideline for any MFI to implement a decentralized platform on the blockchain. For example, Grameen Bank can use such a DApp to mitigate the cost of hiring people to run their organization.
- We have introduced the concept of social collateral that enables borrowers to borrow money without any collateral. This is done by developing the algorithm to determine credit scores.
- As the DApp is powered by Ethereum, it has got all the advantages of the platform. Immutable data transactions, decentralized nodes, transparent dataflow,

smart contracts, and automized execution of those contracts, etc., are some of those advantages. These make the platform incredibly secure. It also makes sure that only authorized and signed-in users can participate in the blockchain.

6.8 Disadvantages

Our current DApp has some limitations:

- The current Dapp does not facilitate transactions from banks as users require some reset amount of Ethereum in their wallet. However, this part can be added easily. We can use the APIs of banks to transfer money, or their individual blockchain can be added. We can also modify the DApp and algorithms to add them later
- We require certain primary users to be on the platform before we can use the financial data to determine creditworthiness.

6.9 Future Works

We would like to examine the following for our future work:

- We want to investigate how our DApp can be integrated with the layer 2 Ethereum service Polygon [59] so that users can process transactions with lower gas fees. This would also allow us to bypass the network congestion in Ethereum.
- We want a system where we combine the existing verification system into our platform with biometrics so that we can carry out the for each customer for more secure entry. The client's information will be held in a protected server and can be accessed by MFIs. This will help as no manual intervention as well.
- We would like to develop a system where multiple banks would be connected to the frontend server so people can easily transfer money into their wallets for staking and lending. Also, borrowers can directly transfer money to their bank account.
- **W**e would like to explore the idea of moving to a private blockchain platform like Hyperledger Fabric, as it has channels. It is a new concept that allows the use of many blockchains within the same network while protecting their privacy, which is essential to financial system secrecy [18].

Chapter 7

Conclusion

In this thesis report, we have presented an MFI DApp that will allow people to borrow, stake and lend money. Firstly, we identified a set of requirements based on a threat model for the MFI DApp.The architecture of the DApp has been gone along with protocol flow diagrams and algorithms running the platform. We have developed prototype hoisted on the Ethereum blockchain and showed its practical applicability. We gave the app a front end and worked on the user interface of the protocol. Furthermore, we have examined its privacy and security issues. We also discussed the advantages, and disadvantages of the designed protocol which satisfies the formulated requirements. This mitigates the identified threats. Using the DApp, one can easily borrow money without collateral. However, its true possibility can be improved furthermore. It can be integrated with any existing MFI. So we have to lay out the foundation for wide-scale adoption. Thus, it can be regarded as research with far-reaching adoption potential.

Bibliography

[1] J. Conning, U. of Maryland at College Park. Institutional Reform, and the Informal Sector Center, *Group Lending, Moral Hazard, and the Creation of Social Collateral* (Working paper (University of Maryland at College Park. Institutional Reform and the Informal Sector Center)). Center for Institutional Reform and the Informal Sector, University of Maryland at College Park, 1996. [Online]. Available: https://books.google.com.bd/books?id=jS0wHAAACAAJ.

[2] G. Ritzer, *The blackwell encyclopedia of sociology*. Blackwell Publishing, 2007.

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Cryptography Mailing list at https://metzdowd.com*, Mar. 2009.

[4] G. Blank and B. Reisdorf, "The participatory web," *Information*, vol. 15, May 2012. DOI: 10.1080/1369118X.2012.665935.

[5] V. Buterin, *Daos, dacs, das and more: An incomplete terminology guide*, May 2014. [Online]. Available: https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide.

[6] A. Shostack, "Chapter 3 stride," in *Threat modeling: Designing for security*. Wiley, 2014, pp. 61–85.

[7] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[8] F. Nawaz, "Microfinance, financial literacy, and household power configuration in rural bangladesh: An empirical study on some credit borrowers," *VOLUNTAS: International Journal of Voluntary and Nonprofit Organizations*, vol. 26, pp. 1100–1121, 2015.

[9] H. Scheyvens, "The role of microfinance and microfinance institutions in climate change adaptation: Learning from experiences in bangladesh," Institute for Global Environmental Strategies, Tech. Rep., 2015, pp. 19–28. [Online]. Available: http://www.jstor.org/stable/resrep00725.10.

[10] S. Raval, "What is a decentralized application?" In *Decentralized applications: Harnessing bitcoin's blockchain technology*. O'Reilly Media Inc., 2016, pp. 1–8.

[11] M. Hasan and M. Malek, "Microfinance market in bangladesh," in May 2017, pp. 271–306, ISBN: 978-981-3147-94-2. DOI: 10.1142/9789813147959_0008.

[12] M. Vukolić, "Rethinking permissioned blockchains," Apr. 2017, pp. 3–7. DOI: 10.1145/3055518.3055526.

[13] W. Cai, Z. Wang, J. Ernst, Z. Hong, and C. Feng, "Decentralized applications: The blockchain-empowered software system," *IEEE Access*, vol. 6, pp. 53 019–53 033, Oct. 2018. DOI: 10.1109/ACCESS.2018.2870644.

[14] Y. Xinyi, Z. Yi, and Y. He, "Technical characteristics and model of blockchain," in *2018 10th International Conference on Communication Software and Networks (ICCSN)*, 2018, pp. 562–566. DOI: 10.1109/ICCSN.2018.8488289.

[15] B. Ampel and M. Patton, "Performance modeling of hyperledger sawtooth blockchain," Jul. 2019, pp. 59–61. DOI: 10.1109/ISI.2019.8823238.

[16] D. M. Choudhury, "Mfi microsavings: Inclusive or transformative? - a participatory survey of household savings in northeast rural bangladesh," vol. 1, pp. 79–102, Dec. 2019.

[17] M. S. Ferdous, F. Chowdhury, and M. Alassafi, "In search of self-sovereign identity leveraging blockchain technology," *IEEE Access*, vol. 7, pp. 1–1, Jul. 2019. DOI: 10.1109/ACCESS.2019.2931173.

[18] M. S. I. Bhuiyan, A. Razzak, M. S. Ferdous, M. J. M. Chowdhury, M. A. Hoque, and S. Tarkoma, "Bonik: A blockchain empowered chatbot for financial transactions," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 1079–1088. DOI: 10.1109/TrustCom50675.2020.00143.

[19] F. Eigelshoven, A. Ullrich, and B. Bender, *Public blockchain–a systematic literature review on the sustainability of consensus algorithms*, Jun. 2020.

[20] D. Li, W. E. Wong, and J. Guo, "A survey on blockchain for enterprise using hyperledger fabric and composer," in *2019 6th International Conference on Dependable Systems and Their Applications (DSA)*, 2020, pp. 71–80. DOI: 10.1109/DSA.2019.00017.

[21] S. Sarker, A. Saha, and M. S. Ferdous, "A survey on blockchain cloud integration," Dec. 2020. DOI: 10.1109/ICCIT51783.2020.9392748.

[22] D. R. Silva, "Characterizing relationships between primary miners in ethereum by analyzing on-chain transactions," in *2020 2nd Conference on Blockchain Research Applications for Innovative Networks and Services (BRAINS)*, 2020, pp. 240–247. DOI: 10.1109/BRAINS49436.2020.9223303.

[23] N. S. Yadav, V. Goar, and M. Kuri, "Crypto wallet: A perfect combination with blockchain and security solution for banking," *International Journal of Psychosocial Rehabilitation*, vol. 24, pp. 6056–6066, Feb. 2020. DOI: 10.37200/IJPR/V24I2/PR2021078.

[24] I. Alom, R. M. Eshita, A. Ibna Harun, *et al.*, "Dynamic management of identity federations using blockchain," in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2021, pp. 1–9. DOI: 10.1109/ICBC51069.2021.9461128.

[25] A. Marques, "How many software engineers does it take to launch a startup?" *Pluo*, Sep. 2021. [Online]. Available: https://pluo.jobs/blog/how-many-software-engineers-does-it-take-to-launch-a-startup/.

[26] W. Zou, D. Lo, P. S. Kochhar, *et al.*, "Smart contract development: Challenges and opportunities," *IEEE Transactions on Software Engineering*, vol. 47, no. 10, pp. 2084–2106, 2021. DOI: 10.1109/TSE.2019.2942301.

[27] Foley.com. "Types of blockchain: Public, private, or something in between." (2022), [Online]. Available: https://www.foley.com/en/insights/publications/2021/08/types-of-blockchain-public-private-between%3E.

[28] M. M. Hoque, "Microfinance challenges and the potential benefits of blockchain technology and mobile money," Ph.D. dissertation, Queensland University of Technology, 2022. DOI: 10.5204/thesis.eprints.228731. [Online]. Available: https://eprints.qut.edu.au/228731/.

[29] M. Mazzoni, A. Corradi, and V. Di Nicola, "Performance evaluation of permissioned blockchains for financial applications: The consensys quorum case study," *Blockchain: Research and Applications*, vol. 3, no. 1, p. 100 026, 2022, ISSN: 2096-7209. DOI: https://doi.org/10.1016/j.bcra.2021.100026. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S209672092100021X.

[30] H. Papacharissiou, "How to build a dapp in three steps," *Chainlink Blog*, Jul. 2022. [Online]. Available: https://blog.chain.link/how-to-build-a-dapp/.

[31] "Alternative Credit Scoring: A Tool for Broadening Digital Financial Inclusion," *LightCastle Partners*, Mar. 2023. [Online]. Available: https://www.lightcastlebd.com/insights/2023/03/alternative-credit-scoring-financial-inclusion-bangladesh.

[32] "Evolution of micro insurance in bangladesh: Financial cushion for the bottom of the pyramid population," *LightCastle Partners*, Jan. 2023. [Online]. Available: https://www.lightcastlebd.com/insights/2021/05/evolution-of-micro-insurance-in-bangladesh-financial-cushion-for-the-bottom-of-the-pyramid-population/.

[33] W.-M. Lee, "Using the metamask crypto-wallet," in *Beginning Ethereum Smart Contracts Programming: With Examples in Python, Solidity, and JavaScript*. Berkeley, CA: Apress, 2023, pp. 111–144, ISBN: 978-1-4842-9271-6. DOI: 10.1007/978-1-4842-9271-6_5. [Online]. Available: https://doi.org/10.1007/978-1-4842-9271-6_5.

[34] "Node as a service," *ethereum.org*, 2023. [Online]. Available: https://ethereum.org/en/developers/docs/nodes-and-clients/nodes-as-a-service/.

[35] M. Pacheco, G. Oliva, G. K. Rajbahadur, and A. Hassan, "Is my transaction done yet? an empirical study of transaction processing times in the ethereum blockchain platform," *ACM Trans. Softw. Eng. Methodol.*, vol. 32, no. 3, Apr. 2023, ISSN: 1049-331X. DOI: 10.1145/3549542. [Online]. Available: https://doi.org/10.1145/3549542.

[36] "Aave/aave-protocol: Aave protocol version 1.0 - decentralized lending pools." [Online]. Available: https://github.com/aave/aave-protocol.

[37] "All NGO Job Circular 2023 (All in One)," Accessed: May. 20, 2023. [Online]. Available: https://www.notice24x7.com/en/ngo-job-circular/.

[38] "Annual Report 2021 of Grameen Bank," Grameen Bank. [Online]. Available: https://grameenbank.org/public/assets/archive/annual_report/Annual_Report_2021.pdf.

[39] Arc, *Blockchain developer hourly rate (2023)*, Accessed: May. 20, 2022. [Online]. Available: https://arc.dev/freelance-developer-rates/blockchain.

[40] "Asa." [Online]. Available: https://asa.org.bd/.

[41] "Bitcoin." [Online]. Available: https://www.bitcoin.org/.

[42] "Brac." [Online]. Available: https://www.brac.net/.

[43] "Ethereum." [Online]. Available: https://www.ethereum.org/.

[44] "Ethereum virtual machine (evm)," Accessed: May. 20, 2022. [Online]. Available: https://ethereum.org/en/developers/docs/evm/.

[45] "Gas and fees," Accessed: May. 20, 2022. [Online]. Available: https://ethereum.org/en/developers/docs/gas/.

[46] Glassdoor, *Branch manager salaries(2023)*, Accessed: May. 20, 2022. [Online]. Available: https://www.glassdoor.com/Salaries/bangladesh-branch-manager-salary-SRCH_IL.0,10_IN27_KO11,25.htm.

[47] "Grameen Bank." [Online]. Available: https://grameenbank.org/.

[48] "Grameen Bank Now," Grameen Bank. Accessed: May. 10, 2023. [Online]. Available: https://grameenbank.org/about/gb-now.

[49] "Hyperledger." [Online]. Available: https://www.hyperledger.org/.

[50] "Hyperledger Fabric." [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html.

[51] "Hyperledger SawtoothDoc." [Online]. Available: https://sawtooth.hyperledger.org/docs/1.2/.

[52] "Kiva." [Online]. Available: https://www.kiva.org/.

[53] "Litecoin." [Online]. Available: https://litecoin.org/.

[54] "Metamask." [Online]. Available: https://metamask.io/.

[55] "Micro finance institutions (mfis)," *Bangladesh Bank*, [Online]. Available: https://www.bb.org.bd/en/index.php/financialactivity/mfi.

[56] "Microfinance in Bangladesh (Annual Statistics), 2022," Microcredit Regulatory Authority (MRA). [Online]. Available: https://mra.portal.gov.bd/site/page/2f639723-7e4d-4cac-aa14-539c10af3284/-.

[57] "Mining," Accessed: May. 20, 2022. [Online]. Available: https://ethereum.org/en/developers/docs/consensus-mechanisms/pow/mining/.

[58] "Moeda finance." [Online]. Available: https://moeda.finance/.

[59] "Polygon." [Online]. Available: https://polygon.technology/.

[60] "Quorum." [Online]. Available: https://www.goquorum.com/.

[61] "Solana." [Online]. Available: https://www.solana.org/.

[62] "Solidity documentation." [Online]. Available: https://docs.soliditylang.org/en/v0.8.20/.

[63] "The maker protocol white paper." [Online]. Available: https://makerdao.com/en/whitepaper/.

[64] "Truefi uncollateralized loans: High yield lending amp; borrowing without collatera." [Online]. Available: https://truefi.io/.

[65] "UNDERSTANDING CRYPTO COSTS: TRANSACTION AND GAS FEES," *Worldcoin*, Accessed: May. 20, 2022. [Online]. Available: https://worldcoin.org/articles/crypto-gas-fees.

[66] "Xe Currency Converter," *XE*, Accessed: May. 20, 2022. [Online]. Available: https://www.xe.com/currencyconverter/convert/?Amount=1&From=BDT&To=USD.