

Decentralized Social Network using Blockchain (Dinke-Din)



Submitted in partial fulfillment of the requirements for the degree
Of
Bachelor of Science in Computer Science and Engineering

JAMSHEDUL ALAM (C171029)

MOHAMMED RAIHAN (C163069)

JAMSHEDUL ALAM (C171029)
MOHAMMED RAIHAN (C163069)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF
B. SC. IN COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (CSE)
INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG (IIUC)
CHITTAGONG, BANGLADESH

DECLARATION

We hereby declare that Dinke-din is our own unique creation and we try to add knowledge to existing blockchain based previous knowledge .

27 September , 2021

JAMSHEDUL ALAM
C171029

MOHAMMED RAIHAN
C163069

DECLARATION OF SUPERVISOR

Supervisor's word

Date

SAZID ZAMAN KHAN
Lecturer, Dept. of CSE, IIUC

DECLARATION OF PROJECT REPORT AND COPYRIGHT**PROJECT REPORT TITLE:**

Decentralized Social Network using Blockchain (Link-din)

AUTHORS:

SL NO.	AUTHOR'S NAME	STUDENT ID	SIGNATURE
1	JAMSHEDUL ALAM	C171029	
2	MOHAMMED RAIHAN	C163069	
NAME OF SUPERVISOR: SIGNATURE OF SUPERVISOR:			

We declare

1. Our thesis to be published as online open access (full text) at IIUC database or archive.

2. Dept. of CSE, IIUC reserves the right as follows:

i. The thesis is the property of Dept. of CSE, IIUC

ii. The Library of IIUC has the right to make copies for the purpose of research only.

iii. The Library has the right to make copies of the thesis for academic exchange.

ACKNOWLEDGEMENT

Thanks to Almighty Allah for giving us Barakah to reach the end of Dinke-din project accomplishment .Without Allah's Barakah it was impossible for us to reach the end of our Dinke-din project .

We feel proud about our Supervisor Sazid Zaman Khan . Who helps to us to become the best version of ourselves .He is very helpful .However , we are grateful to all of the IIUC Teacher , Authority , staffs .

JAMSHEDUL ALAM

MOHAMMED RAIHAN

ABSTRACT

We all heard about blockchain .But we have a curiosity to calm our mind what is actually blockchain does ? what it provide to the humanity ? let's comes to the motto of the blockchian ,”To stroe the digital data so that not to get modified”[5]. it actually works as a open source database .

In our Dinke-din we are going to explore the blockchian world so that in social network uses can be safe and it's owner can keep the data in safe .We all are familiar with sharing post to social media .We all feel insecure about data that we are sharing to the online social network (OSN) system.What will happen to our life if any of the data are hacked or modified or shared to any corrupt person .Then our daily life's becomes a hard one to live .Dinke-din is the palce where we explore the Blockchain and it's most eye charming technology IPFS(Inter Planetary File system). In the Dinke-din , we used the blockchian hash ,the most buzzing word we always listen to secure the data , is like fingerprint for digital data . In our Dinke-din we uses different hashing algorithm.

TABLE OF CONTENTS

	Page
DECLARATION	iv
SUPERVISOR’S DECLARATION	v
DECLARATION OF THESIS / PROJECT REPORT AND COPYRIGHT	vi
ACKNOWLEDGEMENT	viii
ABSTRACT	ix
TABLE OF CONTENTS	x
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	Error!
Bookmark not defined.	
CHAPTER I INTRODUCTION	
1.1 Background of the Study	1
1.2 Problem Statement	1
1.3 Motivation	2
1.4 Objective of the Project	2
CHAPTER II LITERATURE REVIEW	
2.1 Introduction	4
2.2 Existing Work Reveiw	4
2.3 Limitations of the Existing Works	5
2.4 Problems That Are Focused From Study	5
2.5 Summery	5
CHAPTER III REQUIREMENT SPECIFICATION & ANALYSIS	
3.1 Why Requirement Specification Is Needed ?	6
3.2 Requirement Defination	7
3.3 Requirement Validation	8
3.4 Sytem Requirements	9
3.5 Feasibility Study	10
3.6 Gantt Chart	12

CHAPTER IV METHODOLOGY

4.1	Process Model	13
4.2	Recent Trends In Process Model	13
4.3	Why We Choose Agile Model ?	13
4.4	Agile Model Implementations	14
4.5	Limitations of Agile Model	17

CHAPTER V SYSTEM ANALYSIS

5.1	System Analysis	18
5.2	Data Flow of SDMS App	18
5.3	Data View	19
5.4	Security	19
5.5	Version Support	20
5.6	System Design	20
5.7	A General Model of Software Design Process	21
5.8	Flow Chart Diagram	22
5.9	Use Case Diagram	25
5.10	Entity Relationship Diagram	28
5.11	Data Flow Diagram	35

CHAPTER VI SYSTEM IMPLEMENTATION & TESTING

6.1	System Implementations in Details	39
6.2	UI Design Implementation	42
6.3	Testing	58

CHAPTER VII CONCLUSION & FUTURE PLAN

7.1	Conclusion	62
7.2	Contribution of the Project	62
7.3	Future Plan	62

CHAPTER I

INTRODUCTION

1.1 BACKGROUND OF THE STUDY

Online social networks have exploded in popularity, allowing users to exchange images, videos, and other content with their peers. Facebook, for example, has more than two billion members.

At the same time, consumer's privacy in online social networks(OSN) has been discovered to be easily compromised . Although members of Online Social Networks (OSN) can alter privacy settings to limit who has access and who does not, We don't have any effective technical solutions to provide access or prevent them from disclosing user data to third parties in this case. The New York Times reported in March 2018 that a business fraudulently got information from more than 50 million Facebook users. The primary cause of these incidents is that present OSN's with centralized structures can completely comprehend all user data..

The solution is a decentralized online social network based on BlockChain (OSN).

1.2 PROBLEM STATEMENT

When we post content to an OSN, we worry that our account will be hacked or blocked, and we'll lose all control over it. The most pressing concern on our minds is how to keep our data safe. Instead of giving the main copy of data to OSN, what if we had a system where we stored data and just gave OSN system an authorized hash value?.

1.3 MOTIVATION

From professional to social worker , every one are uses social network . We all are there for long time and our most of the data are in social network not safe . Nowadays hacker and evil personality in the social network have been increased at a alarming rate . What if we in such social network where we will have a shared cloud system to store the users data and full system data .What if we are not dependant on the centralization of the Social network storage rather we are thinking about decentralize system .if we can make a dapps where users data will not be stored in the direct storage of owner of social network.

1.4 OBJECTIVE OF DINKE-DIN

1. The purpose of this Social Network is to safeguard an OSN user's personal data from being accessed and being lost in the OSN platform.

The following are the desired goals:

- 1.1. In order to Protect user Account.
- 1.2. To keep personal data access as secure as possible.
- 1.3. To store all resources in a separate cloud storage, the OSN will communicate with that cloud storage to retrieve the desired data based on the user's rights and public key.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

Literature review is the first step to our Dinke-din . We have to carry out the Literature survey of others project to find out what others have been done on the same theme of the Dinke-din and identify the gaps and what to do next .

It helps us :

- 1 . To see how our research work is original and provides something new .
- 2 .To make sure that our work isn't redundant or merely reproducing knowledge that's already been done by others .

2.1 EXISTING WORK REVIEW

Work Review By IPFS :

Traffic Police Assistant System[1] is a blockchain based system used India . It offers several features of protecting the Traffic data and other handy data. But It fails to reduce the cost of implementing and designing smart contract . which is an important part of an blockchain based system.

Medical image sharing system [2] blockchain based image sharing system . It has features like zero trust principles, role based authentication etc. It also offers data encryption features. Since each transaction requires peer-to-peer verification , it becomes time consuming especially in a public blockchain with many nodes .

Work Review By Authentication :

IdM system [3] is based on Single Sign-On services . SP validates the identity and credentials with an Idp , preventing the user needs to enter it's identifier and password to access the services. But it has some authentication layer left risky.

2.2 LIMITATION OF THE EXISTING WORKS

1. Most of the applications are highly expensive .
2. They have limitations in combining authentication and blockchain in same system.
3. Some of the existing Application are not with full socail Network system. For that they only provide solution to either IPFS blockchain or authentication security.

2.3 PROBLEMS THAT ARE FOCUSED FROM STUDY

1. This system needs a cloud system where any user's post can be stored in database system . From where a OSN provider securely pull the request of the data for OSN system.
2. In the Application real-time data showing and posting to cloud with same time posting it to ipfs blockchain is must need . which will make the OSN more user-friendly.
3. A well-secured authentication and data security is needed fo better data safeguarding .

2.4 SUMMARY

After mining the research paper about Dinke-din ,we got some interesting information between Dinke-din and other previous work:

- The basic security layer's is a Must have thing.
- The main limitations of previous blockchain based system that must have to overcome.
- After the brief study , we finally figure out our OSN system's features.

CHAPTER III

REQUIREMENT SPECIFICATION & ANALYSIS

Requirement Analysis is very critical to the success or failure of a software project. It enables our team to figure out full process to reach the end of the project .As it is very important to create a good set of requirements to ensure that project have a good chance of success .

We have to think about some point in this stage:

- 1) This requirement will be too expensive?
- 2) How likely that it'll take too long to achieve ?
- 3) How likely that it may not be doable ?
- 4) How likely that it won't be very reliable?

3.2.1 User functionality

- OSN User
- Administrator

3.3 SYSTEM REQUIREMENTS

- Our Dinke-din are developed in web Application.
- As we create back-end API it can be also implement in any native application

3.3.1 Web Application

- 1) Solidity (Programming language for Decentralize app development)
- 2) Laravel (Framework for back-end API building)
- 3) React js (JavaScript framework for front-end development)

3.3.2 Database for the System

- 1) PhpMyAdmin(Sql)

3.4 FEASIBILITY STUDY

Feasibility study keeps us from expending time and racecourses on a project which we will be unable to successfully completed .it helps us to think carefully before deciding to start the project . it is similar to the most used proverb for decision making the - look before leap .In this stages we have to find answers of few questions such as :

1. Can it be done?
2. Will it be acceptable?
3. What are the alternatives?
4. Is it affordable ?

Economic Feasibility:

Our Dinke-din is in affordable .

- A user can afford the cost to participate in the smart contract .

- We used here ether-um .

Technical Feasibility:

The technical viability of our project relates to whether the project is likely to succeed or is too difficult to attempt. This may depend on how easy the materials are to work with or if there are weaknesses in the design or engineering of our project, is the technical aspects.

Operational Feasibility:

This is about how a selected framework will be survive in the challenges and How it will beneficiary to our project.

Schedule Feasibility:

We try our best to ensure that our deadline of our specific task wasn't missed . We try to use Force efficiency to meet the deadline of the project work .

Gantt Chart of Thesis/Project

	March	April	June	July	August	September	October	November
Registration								
Project selection								
Planing								
Feasibility Study								
System Design								
Implementation								
Testing								
Submission								

CHAPTER IV

METHODOLOGY

A software development methodology is similar to a cooking recipe. A software development methodology, like a recipe, tells Us how to build a software. The ultimate goal of software methodology is to create high-quality, maintainable software in a reasonable time frame and at a reasonable cost. A successful software development methodology describes how all of our tools, techniques, and practices work together to create a winning Project.

4.1 PROCESS MODEL

Software process have to include main functionalities of the software and the constraints (Specification). Software needs to be designed and programmed. A software must meet the specifications and to cope up in the software competition we need to evaluate our software.

4.2 DIFFERENT SOFTWARE PROCESS MODELS

1. Waterfall Model.
2. V-model
3. Iterative and Incremental Model
4. Spiral Model
5. Iterative Development Model
6. Agile Model
7. Big Bang Model
8. DevOps Model

4.3 WHY WE CHOOSE SPIRAL MODEL (SDM) ?

The spiral model combines iterative project development from evolutionary implementation of a prototype model with the systematic and controlled features of the sequential waterfall model. The spiral model is better suited for large-scale projects that require consistent improvement and refinement with each iteration around the spiral. The output of one iteration's specific activities is a small proof-of-concept (POC) prototype, which is part of the larger software and is used to gather user feedback. The same activities are repeated in subsequent spirals, with refinement of the POC prototype, to produce a working model of the software called build with a version ID/number. Each version of the build is distributed to users in order to gather feedback for future enhancements in the next version until the final system is developed.

The difficulty in calculating the Gas price, the cost of running a smart contract on a public blockchain such as the Ethereum platform, is particularly high for large-scale projects with complex smart contract coding.

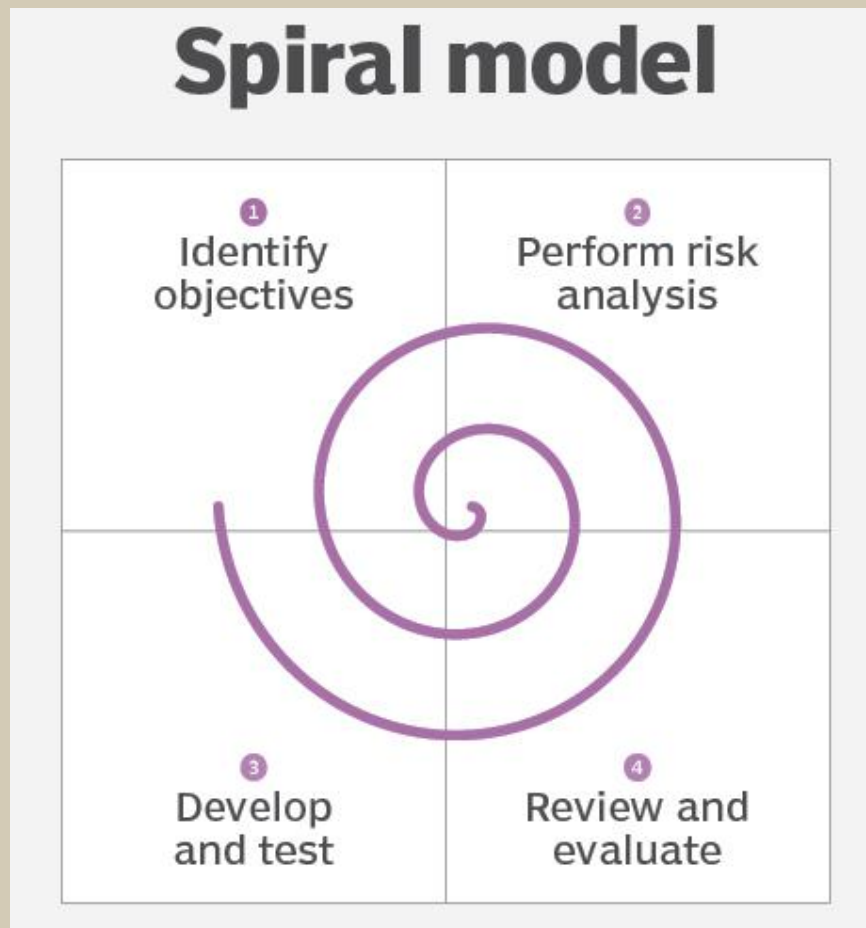


Figure4.2 Agile Model [9]

4.4 AGILE MODEL IMPLEMENTATION

As we used agile approach in our project. We completed the phases properly one after another to avoid conflict in development. The phases are described below in detail:

4.4.1 Requirements

It is the primary period of the development where every one of the prerequisites accumulated and documented. For our project we have examined previous related systems and the users. After that we have finalized our system features.

4.4.2 Design

As we are finished the testing and requirement analysis of our project .Now it's time for designing's the front-end part . In that stage we use React js , the JavaScript framework , to develop a dynamic front-end part. To make our project eye catching we used Material UI , Bootstrap, React js Styled Components to style different parts of page components .

4.4.3 Implementation

As we achieving the product requirements and design , It's time for starting our implementation or development stage in web development terminology we can say we are now going to Back-end system development.

In this phase we strictly follow the spiral model . It helps us to be on track of project requirement and takes us to next steps. We hard coded a function and then test it and optimize it and then we this function for our final figured function .

We implement user backend using solidity and web3.js and admin back-end using Laravel with phpMyAdmin .

4.4.4 Testing

As testing is very important for Sdlc model . We first test our wallet connection by Ethereum Tester and Ganache .In blockchain as everything is node , we node test to keep everything smooth. For Admin Backend we Test our Api .

We observe every error and failed response in our project . Then we debug it and also pay attention to corner stage for which the error are causing.

We also checked our project using White Box testing and Truffle Migration testing in white box approach .We will presenting system testing in chapter 6 ‘System Implementation and Testing’.

4.4.5 Deployment

As we are now in the Deployment stage , we previously debug and log to console bar to detect any hiding error. And Then Now our project is ready for Deployment.Once it is Deployed to production ,Now It is ready for everyone to use in website .

4.4.6 Review & Maintenance

We regularly keep an eye on Maintenance so that we can provide our visitors with the best possible user experience.keeping Updated with valuable information, fixing broken links, removing duplicate pages to boost our website traffic.



Figure4.2 Spiral Model for Software Development [10]

4.5 LIMITATIONS OF SPIRAL MODEL

There are some limitations of using Spiral model in project [11] -

- 1) Difficulty in time management. As the number of phases is unknown at the start of the project, so time estimation is very difficult.
- 2) A large number of intermediate stages necessitate an abundance of documentation.
- 3) Spiral development is best suited for large projects and necessitates risk assessment expertise.

CHAPTER V

SYSTEM ANALYSIS

A system is a special design that is physical and functional need. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives [13].

5.1 SYSTEM ANALYSIS

System analysis is the technique of studying a system or its components in order to identify its objectives. It improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

5.2 DATA FLOW OF DINKE-DIN APP

Data flow is which is the way of sending data to server and receive them to android application. The process is figured out below:

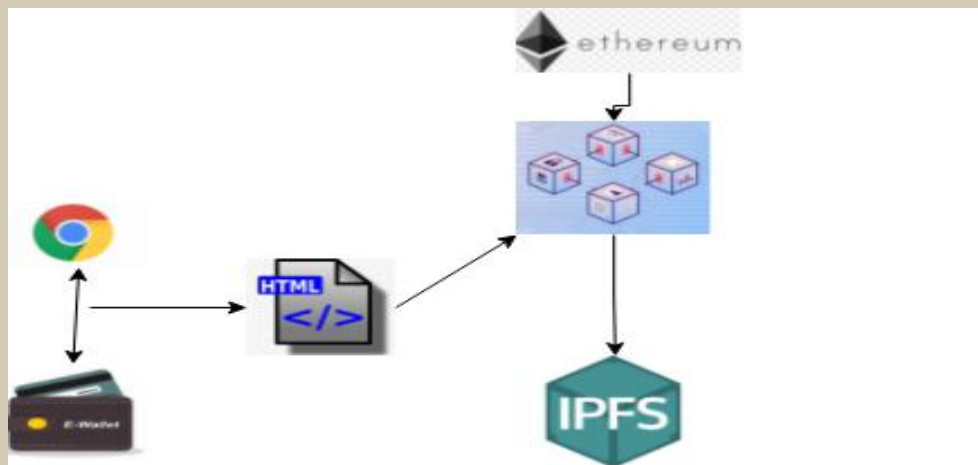


Figure 5.1: System Data Flow of Decentralized Social Network

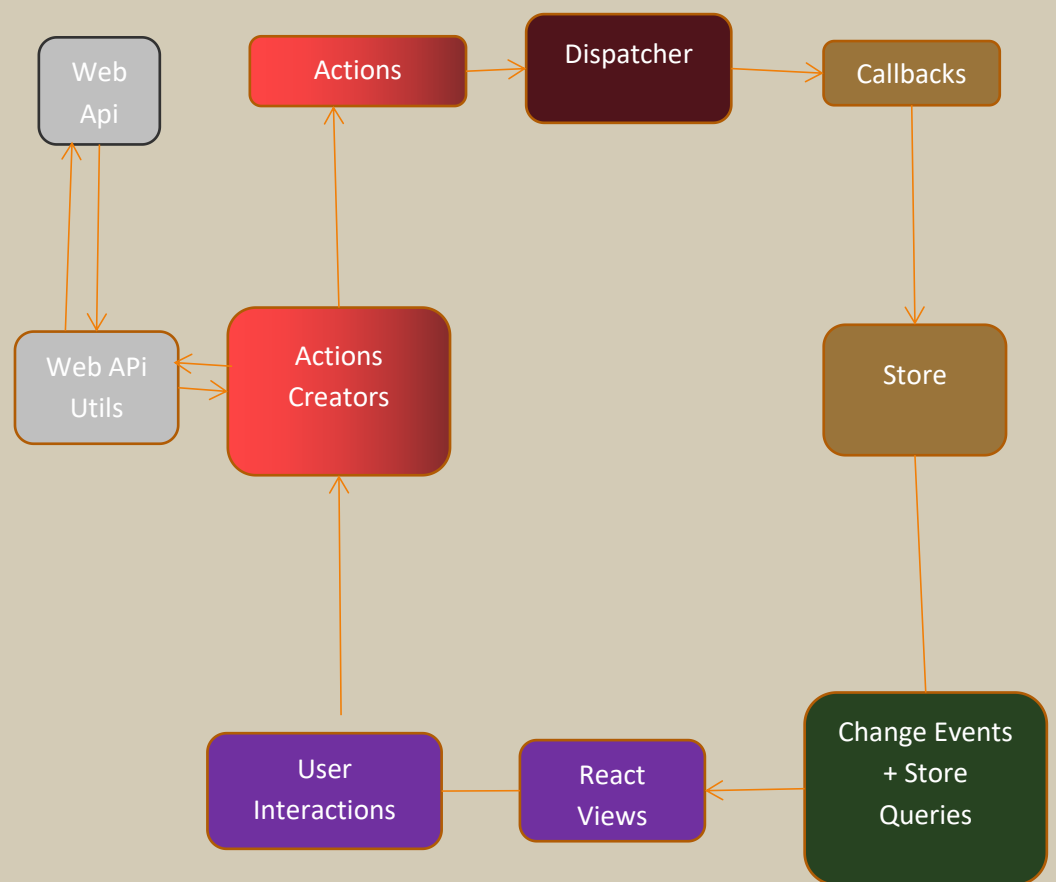


Figure 5.2: System Data Flow of Decentralized Social Network

5.3 DATA VIEW

In Our Dapps we show post of user using grid view. In a post there is two part one is text and other is image .As we request for user data , then the post of every user will be appeared in Grid view. In our Dapps we followed a excenent view of post so that grid view will be responsibe.

5.4 SECURITY

In our Dapps we try to maintain best security practices. When ever a user try to post he/she must have to pay some gas fee and he must have to be connected with his metamask Ethereum wallet. So if any user will not connect his metamask account he will not get the facility to post and even unable to see the post of others. We store the User Post data to IPFS(InterPlanetary File System) to secure the user Shared data .And IPFS will not give the main copy of the data rather it will give it a url hash value which is generated by IPFS system. To secure the user data to be shared by OSN system admin we used JSON Web Token (JWT) . Whenever a OSN admin pull all user data he/she must have to show a badge given the jwt authenticaiton system . If

he shared the data and jwt token to others ,the surprising part of jwt token is that in every different login a different token will be given to admin .So there is no benifit of sharing every data to others, other's will not get the data.

5.5 VERSION SUPPORT

As we building this website with the help of framework , we try to keep up to date of our every frameworks version up to date. Laravel 8, and solidity 0.5.0 version and web3.js version are supported by most of the web browser.

5.6 SYSTEM DESIGN

Objective: To change of the system detail into and operational system plan.

Input: project goals, users requirements and specifications.

Process: Making alternatives to make economic benefits and building a qualityful system for major uses.

Output: system design, schedule, software and hardware requirements.

Fig 5.1: Elements of system [13]

5.8 FLOW CHART DIAGRAM

5.8.1 Flow Chart Diagram for Posts List

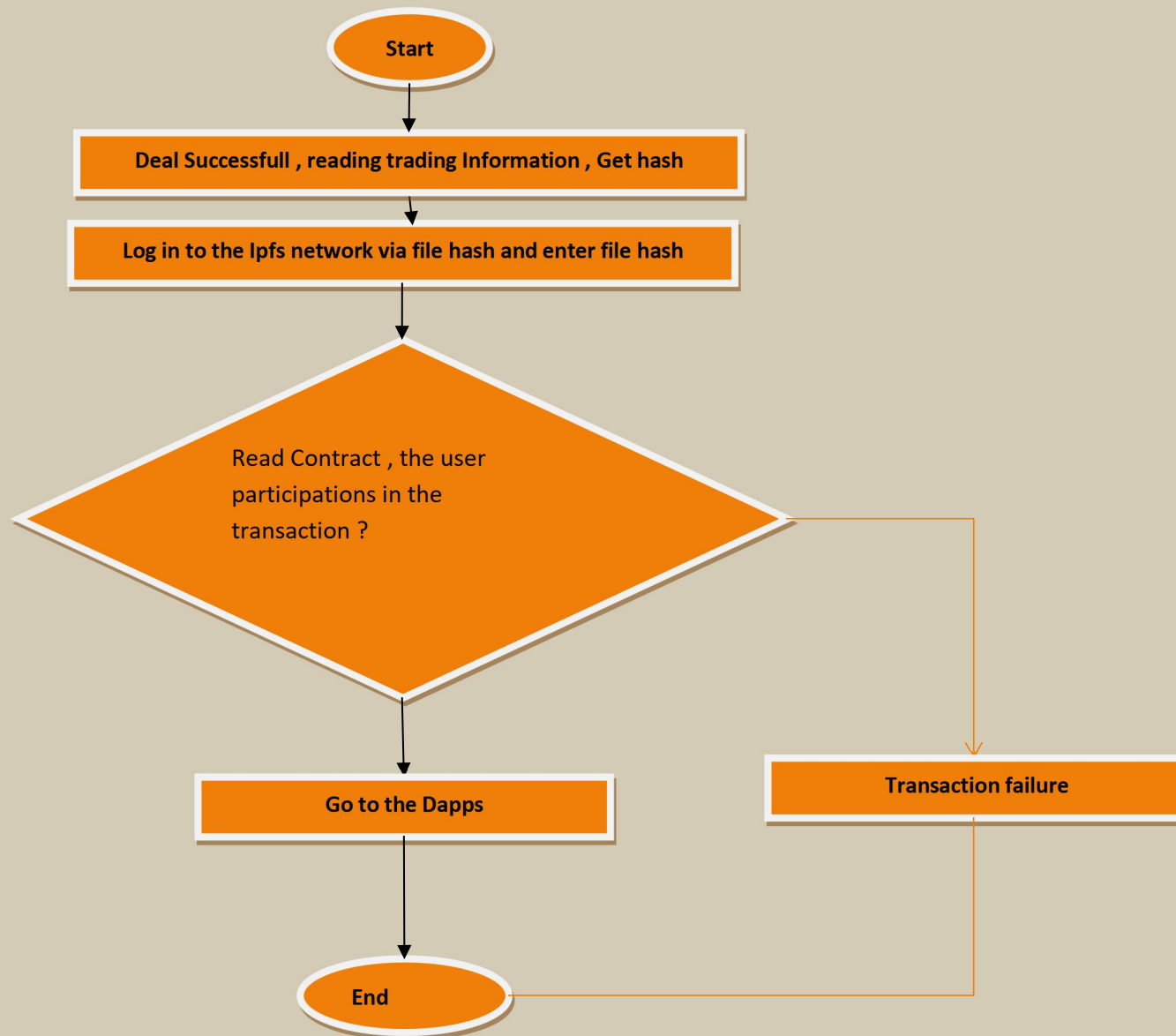


Figure 5.3: User Activity diagram of the system.

5.8.2 Flow Chart Diagram for User Posting panel

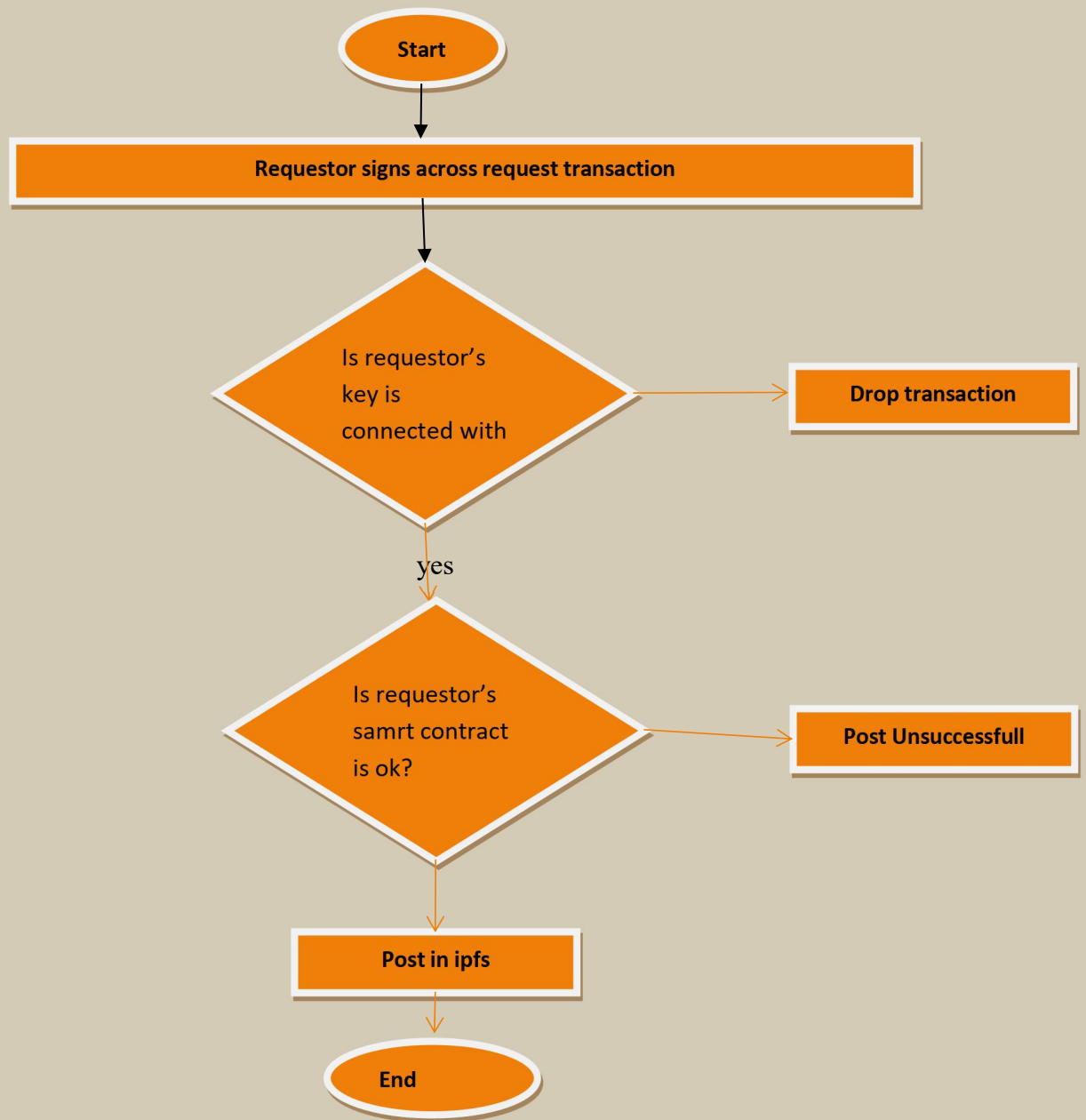


Figure 5.4: User Activity diagram of the system.

5.8.3 Flow Chart Diagram for Admin Panel

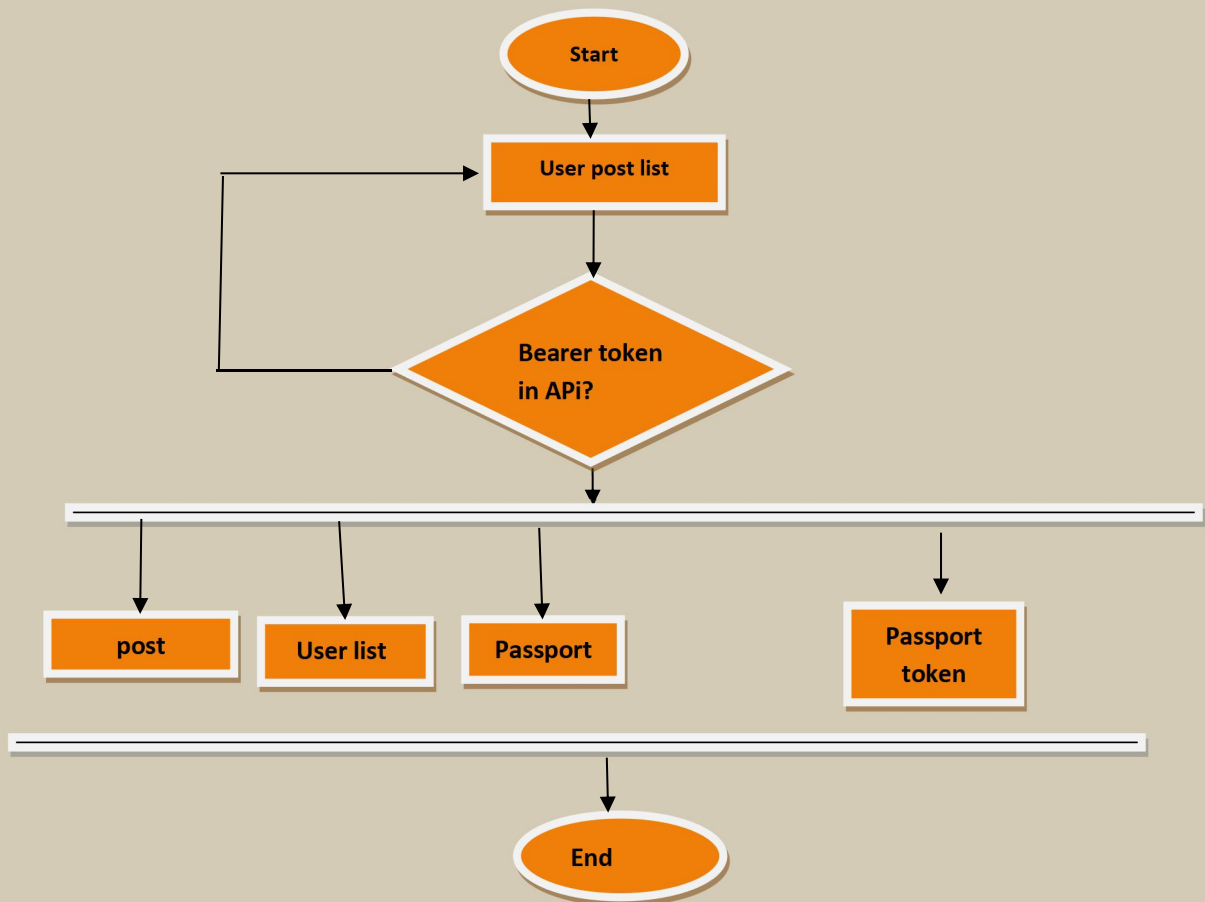


Figure 5.5: Admin Activity diagram of the system.

5.9 USE CASE DIAGRAM

5.9.1 Use Case Diagram for Post Insert Panel

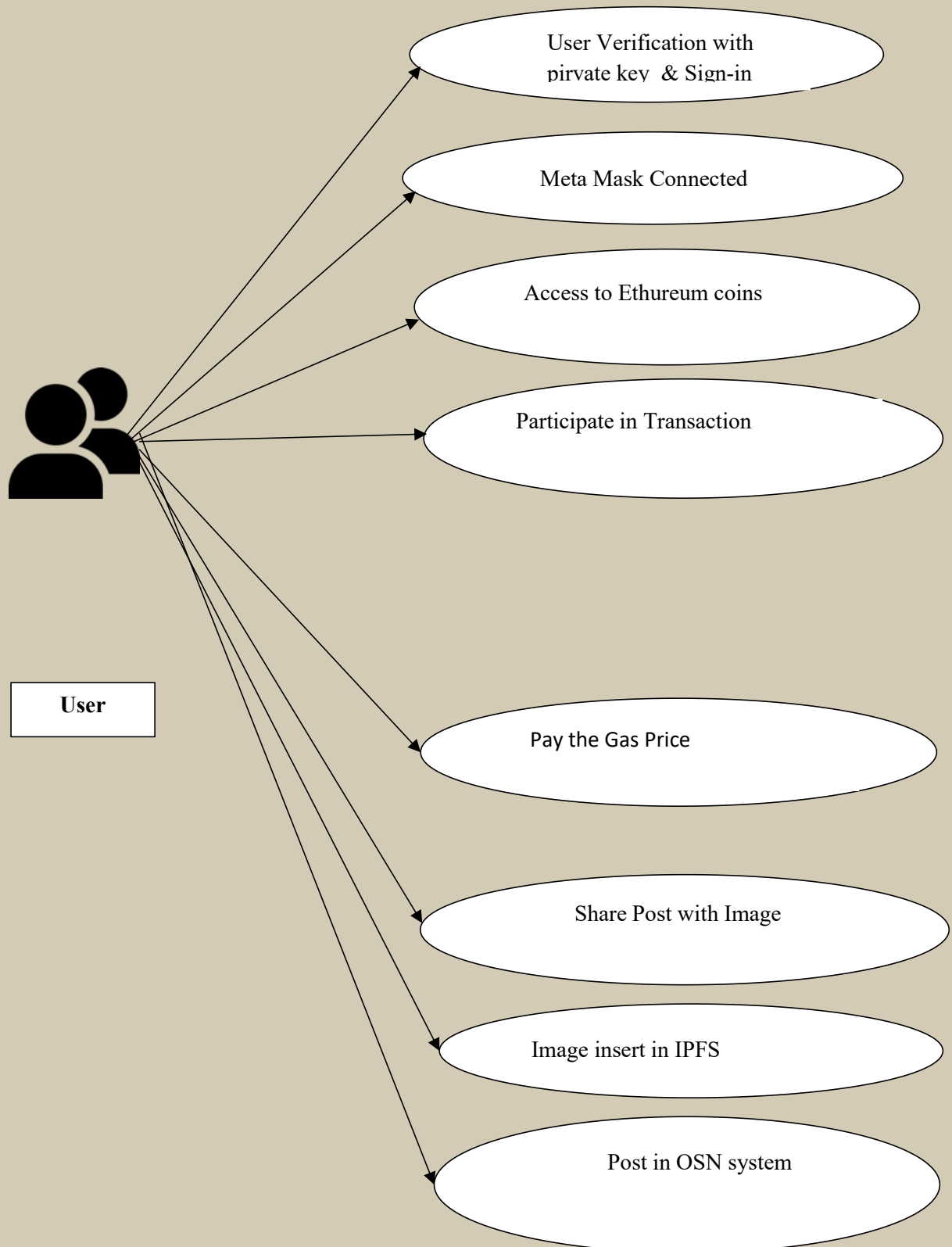


Figure 5.6: Use Case diagram for User

5.9.3 Use Case Diagram for Dinke-din Admin

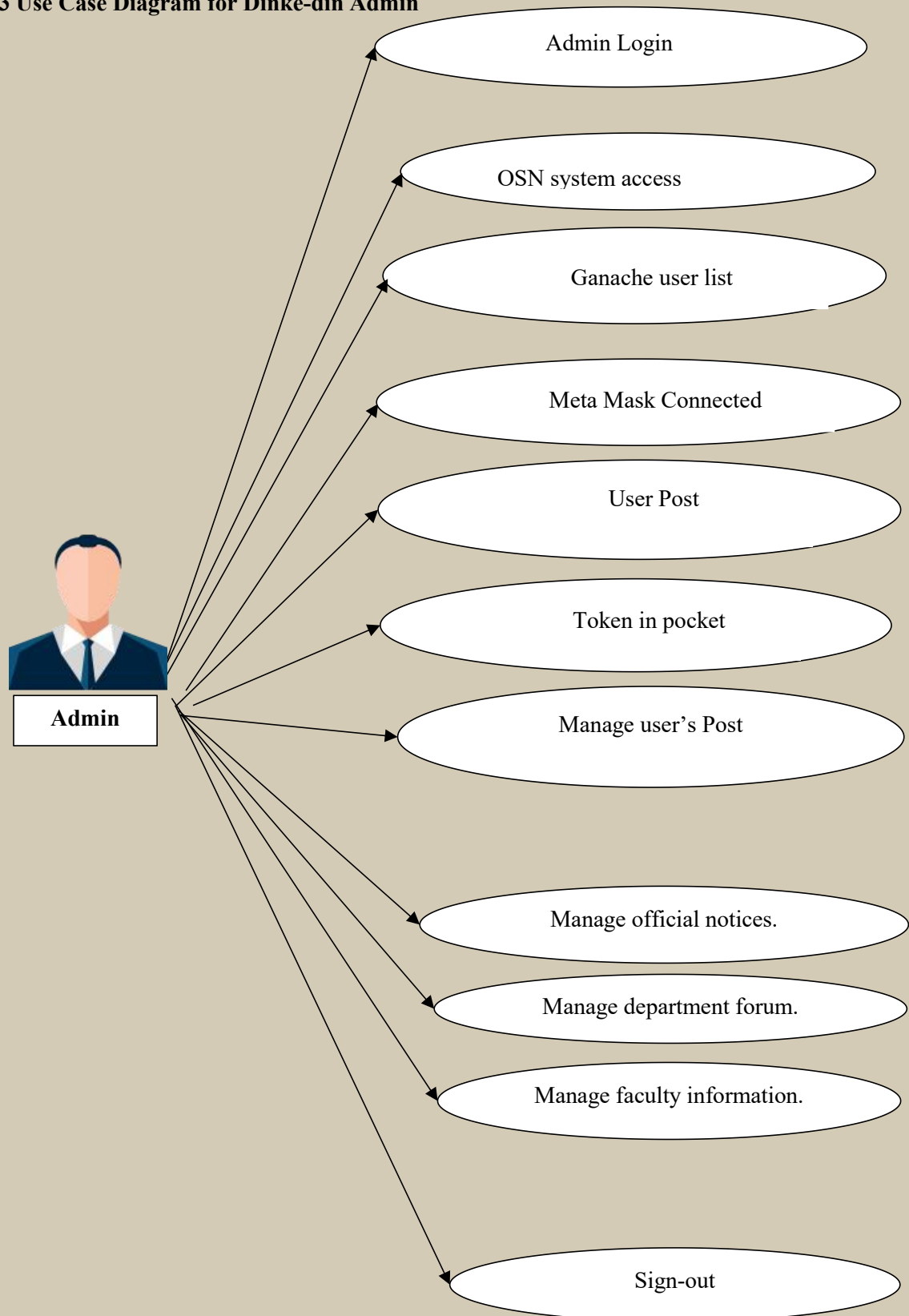


Figure 5.8: Use Case diagram for Admin

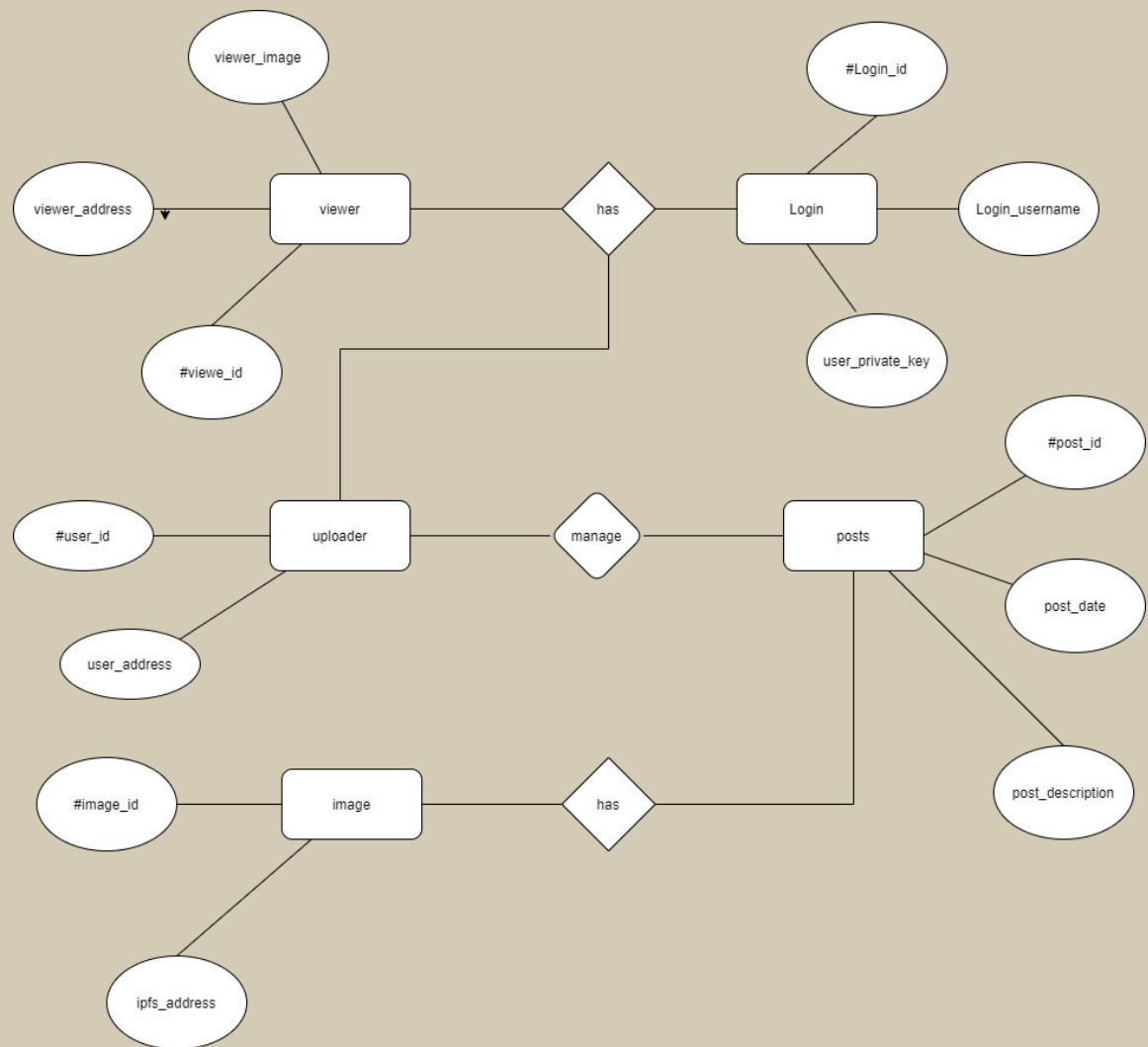


Figure 5.9: Entity Relationship Diagram

5.10.1 Logical Schema

- admin (id, name , email, password , Bearer Token)
- User (id, name , email, email_verified_at , password , remember_token , created_at , updated_at)
- Post(id, user_id , title , image , created_at , updated_at)



























#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	 Change  Drop  More
2	name	varchar(255)	utf8mb4_unicode_ci		No	None			 Change  Drop  More
3	email 	varchar(255)	utf8mb4_unicode_ci		No	None			 Change  Drop  More
4	email_verified_at	timestamp			Yes	NULL			 Change  Drop  More
5	password	varchar(255)	utf8mb4_unicode_ci		No	None			 Change  Drop  More
6	remember_token	varchar(100)	utf8mb4_unicode_ci		Yes	NULL			 Change  Drop  More
7	created_at	timestamp			Yes	NULL			 Change  Drop  More
8	updated_at	timestamp			Yes	NULL			 Change  Drop  More

Fig 5.11 : Table (Admin)




















#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id 	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	 Change  Drop  More
2	user_id	varchar(255)	utf8mb4_unicode_ci		No	None			 Change  Drop  More
3	title	varchar(255)	utf8mb4_unicode_ci		No	None			 Change  Drop  More
4	image	varchar(255)	utf8mb4_unicode_ci		No	None			 Change  Drop  More
5	created_at	timestamp			Yes	NULL			 Change  Drop  More
6	updated_at	timestamp			Yes	NULL			 Change  Drop  More

Fig 5.12 : Table (Posts)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	tokenable_type	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	tokenable_id	bigint(20)		UNSIGNED	No	None			Change Drop More
4	name	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
5	token	varchar(64)	utf8mb4_unicode_ci		No	None			Change Drop More
6	abilities	text	utf8mb4_unicode_ci		Yes	NULL			Change Drop More
7	last_used_at	timestamp			Yes	NULL			Change Drop More
8	created_at	timestamp			Yes	NULL			Change Drop More
9	updated_at	timestamp			Yes	NULL			Change Drop More

Fig 5.13 : Table (Admin access_tokens)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	varchar(100)	utf8mb4_unicode_ci		No	None			Change Drop More
2	access_token_id	varchar(100)	utf8mb4_unicode_ci		No	None			Change Drop More
3	revoked	tinyint(1)			No	None			Change Drop More
4	expires_at	datetime			Yes	NULL			Change Drop More

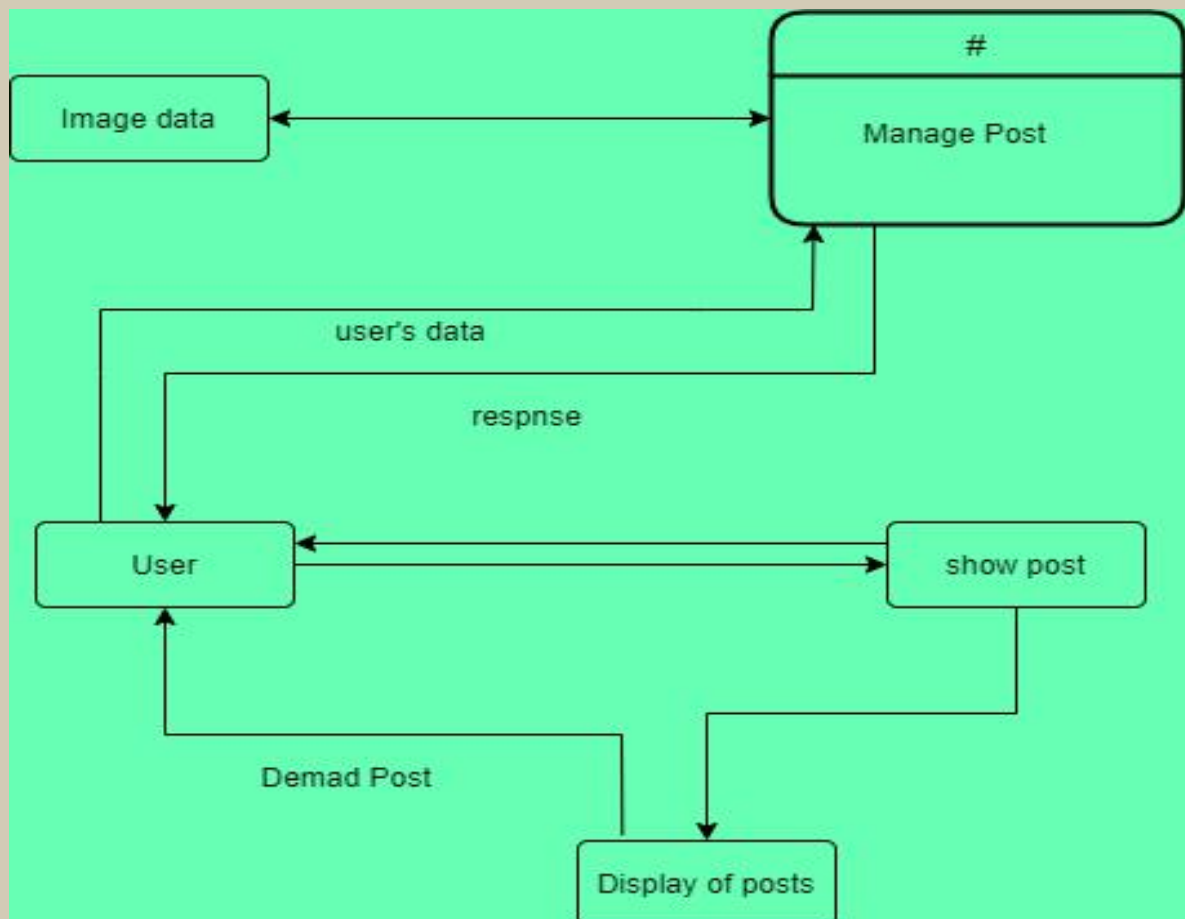
Fig 5.15 : Table (Admin Token refresh)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	email	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
2	token	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	created_at	timestamp			Yes	NULL			Change Drop More

Fig 5.14 : Table (Admin Password reset)

5.11 DATA FLOW DIAGRAM

5.11.1 Level 0 DFD



.11.2 Level 1 DFD for Admin

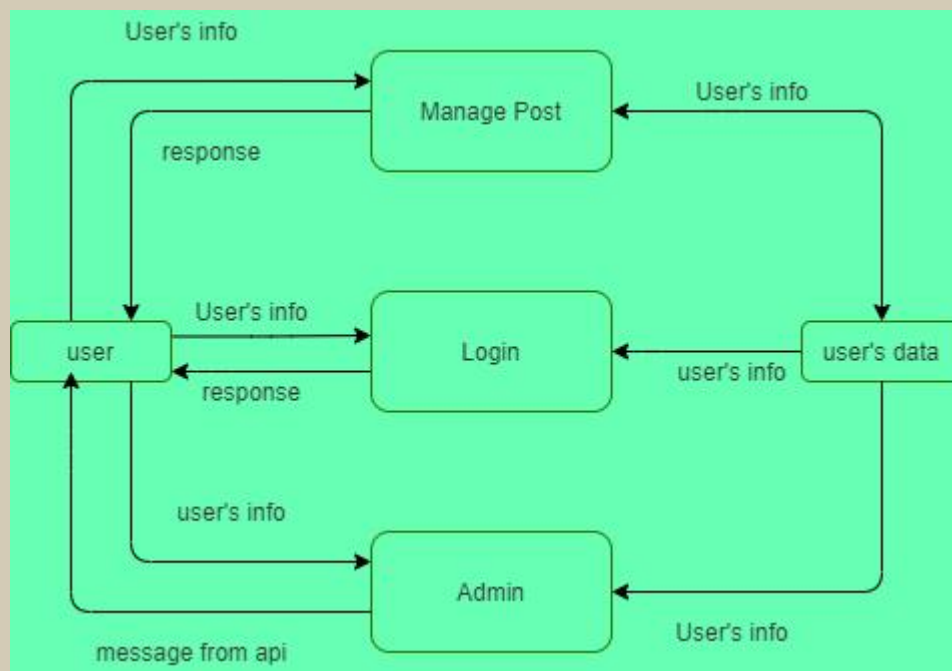


Figure 5.29: Level 1 Data Flow Diagram for Admin .

5.11.3 Level 1 DFD for Teacher

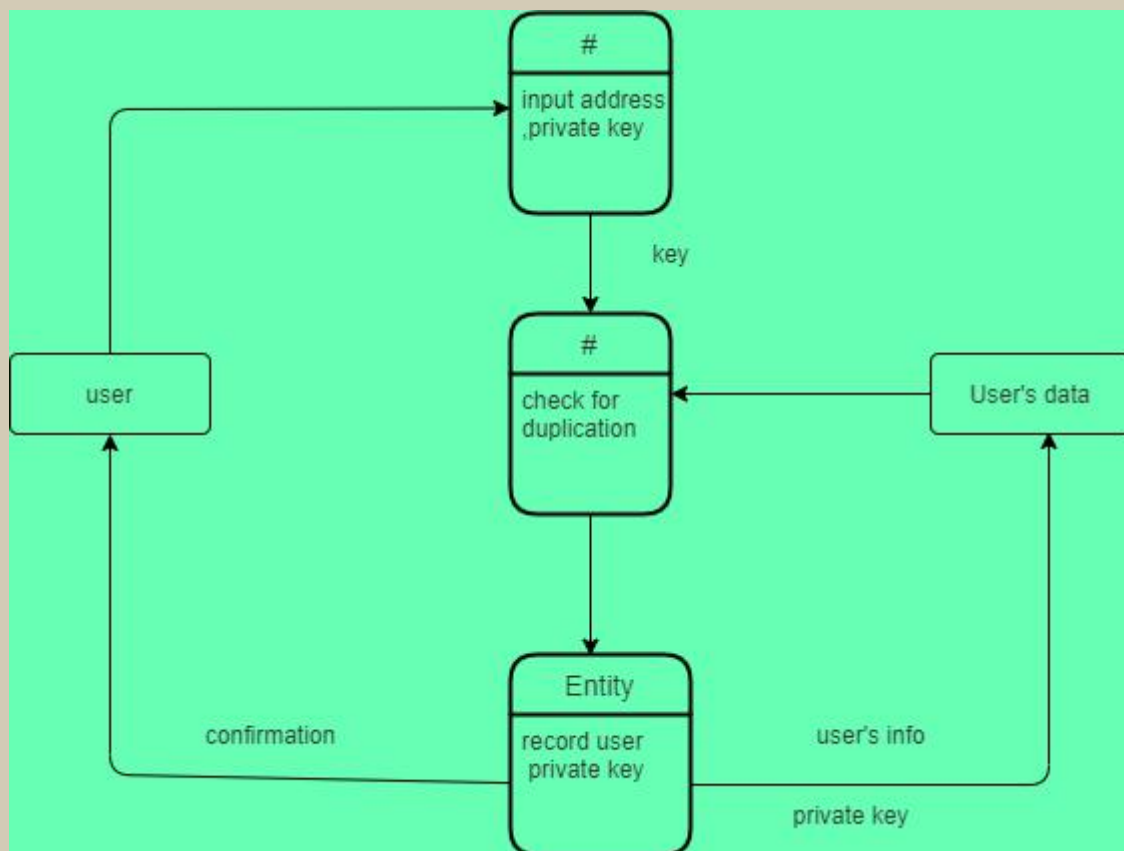
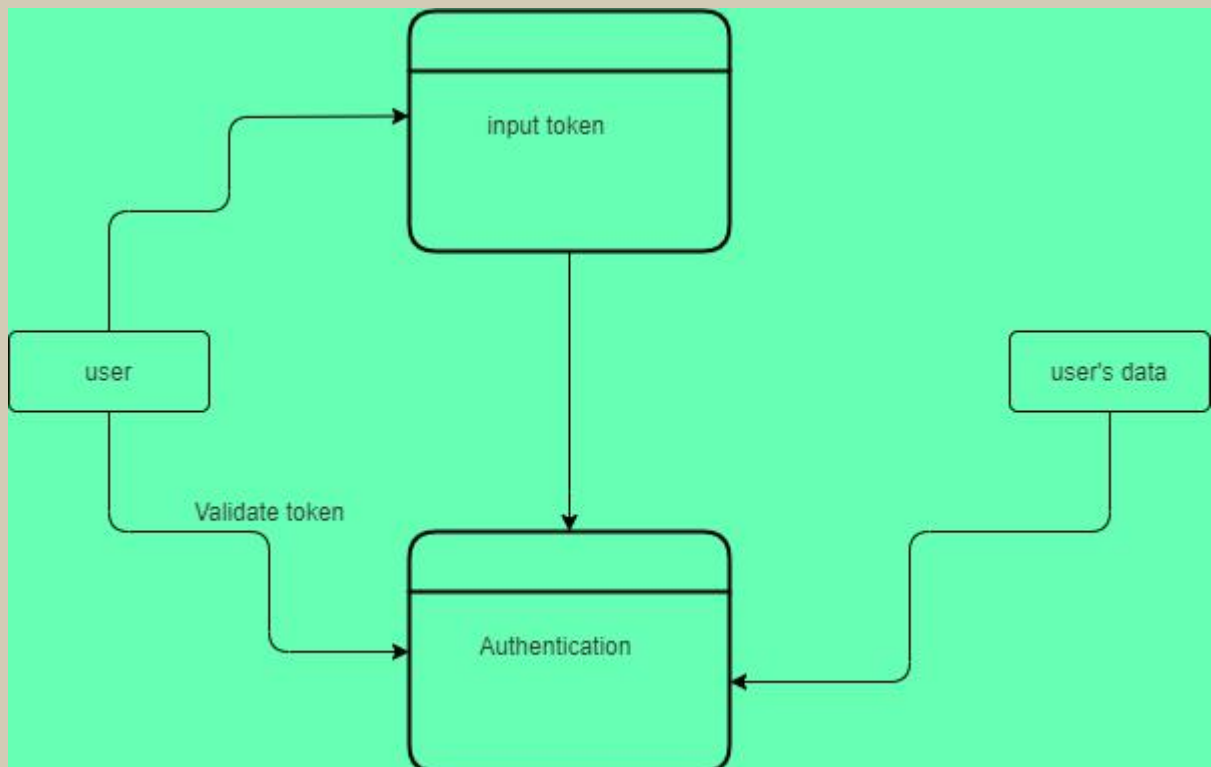


Figure 5.30: Level 1 Data Flow Diagram for User



CHAPTER VI

SYSTEM IMPLEMENTATION & TESTING

In ensuring security in Social network , People always want a secure and decentralize platform . As day by day blockchian becoming a very trusted technology to adopt in securing the social network . By using Blockchain technology we make our Dinke-din a place for trust , hope for social network user .

6.1 SYSTEM IMPLEMENTATION IN DETAILS

In OSN system there are 2 panels.

- User Panel
- OSN Owner Panel

6.1.1 User Panel

To use the User panel user must need to first open the blockchain server provided by Ganache .Without opening the Ganache the user database provided by blockchain terminal will not be accessed . When Ganache will be open it will create a connection between Ganache and Meta mask in browser .Then user signin in metamask by putting Private key mentioned in Account information in Ganache . When a user import his Account to meta mask by putting by private key , then account imported done . Now user has access to user panel . In our PWA user can able to do:

- **Share Post**
 - All Posts – Every user can post in OSN system and they have to pay some gas fee to continue the post to be inserted in the system . whenever a user post in system his data will be uploaded in IPFS system .
 - My Posts – The previous given post will be shown here. Can delete or modify any post.
- **Tip Amount** – Can as in normal OSN system user's post are promoted by likes and comment .In our blockchain system we provide user to promote other's post by giving coin to other , we can call it Tip amount .

6.1.2 Dinke-din Owner Panel

This panel provides the unique facility to the Admin to control all the aspects of the whole system .This panel provide a user-secured data flow within the system. Viewing and managing all old data .it consists of the sub panel activity :

- **Post Control**

- All Post --

Admin can get all posts control with passing some security pathway to confirm that admin are not a malusious user to stole the data.

Admin can get all data of user and post data in this panel . He can delete the data too. .

6.2 UI DESIGN IMPLEMENTATION

Here we will describe the UI Implementation into two main parts. These are:

1. User Panel
2. OSN Admin Panel

6.2.1 User Panel

User is the main attention in our Dinke-din system. User can register and sign in by setting up Meta mask account . Once the Meta mask is setup our system will count the meta mask account as a user . If anything wrong or not registered he will not able to start the Dinke-din system .

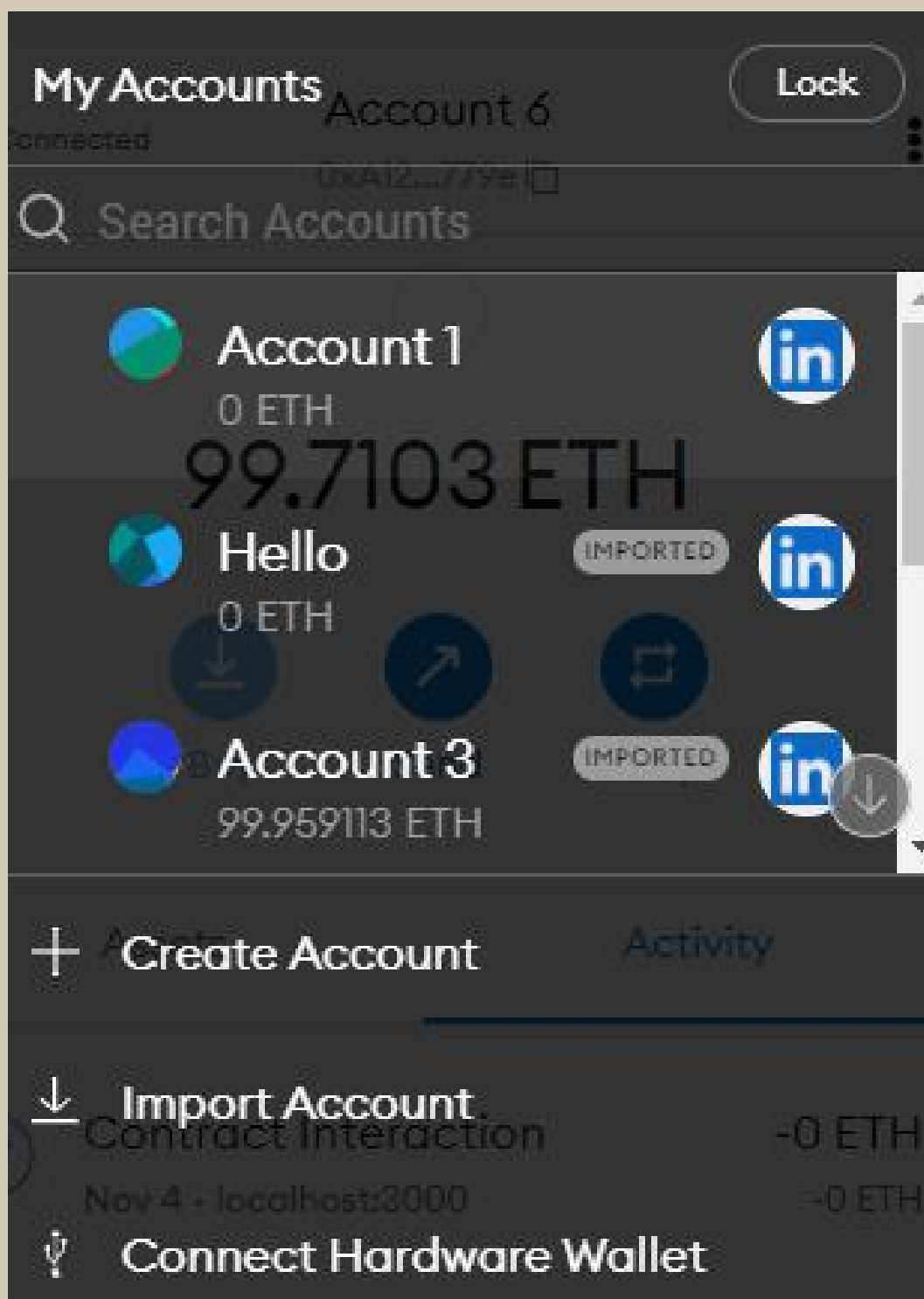






Fig 6.1:User Account in Meta Mask




 MyNetwork 



Import Account

Imported accounts will not be associated with your originally created MetaMask account Secret Recovery Phrase. Learn more about imported accounts [here](#)

Select Type

Private Key 

Paste your private key string here:

Cancel

Import


Fig 6.1:User Account import panel

ACCOUNTS	BLOCKS	TRANSACTIONS	CONTRACTS	EVENTS	LOGS
<div> <div>CURRENT BLOCK: 110</div> <div>GAS PRICE: 20000000000</div> <div>GAS LIMIT: 6721975</div> <div>HARDFORK: MUIRGLACIER</div> <div>NETWORK ID: 5777</div> <div>RPC SERVER: HTTP://127.0.0.1:7545</div> <div>MINING STATUS: AUTOMINING</div> <div>WORKSPACE BLOCKER</div> <div>SWITCH</div> <div>⚙️</div> </div>					
ADDRESS	BALANCE	TX COUNT	INDEX		
0xcb9e2aC57f0AE2F997d5082870d6e5434Ed858e4	100.00 ETH	2	3		
0xEeF7DE542560b113a262Bc796b64E0d3a6FA7b8d	100.00 ETH	0	4		
0xD55Cd0d92e14ee9DEA4bb49a6e65453b84cF6B67	99.96 ETH	18	5		
0x0acD39f8cb9266E9666D070aFA9baB9Fe430Fd4B	100.00 ETH	0	6		
0x8f077628a86491D82fb230276857C6a3757b9842	100.00 ETH	0	7		
0x1Dc4BF561627011d41f519aa252bE0a9ba85f80c	99.96 ETH	20	8		
0xE896124c3F0e7807FC780145c2a83fc48349Cdc7	100.00 ETH	0	9		


Fig 6.1:User Account provided by Blockchain



Fig 6.1: User Account Address in BlockChain



MyNetwork ▼



Import Account

Imported accounts will not be associated with your originally created MetaMask account Secret Recovery Phrase. Learn more about imported accounts [here](#)

Select Type

Private Key ▼

Paste your private key string here:

.....

Cancel

Import

Fig 6.1: User Account Imported

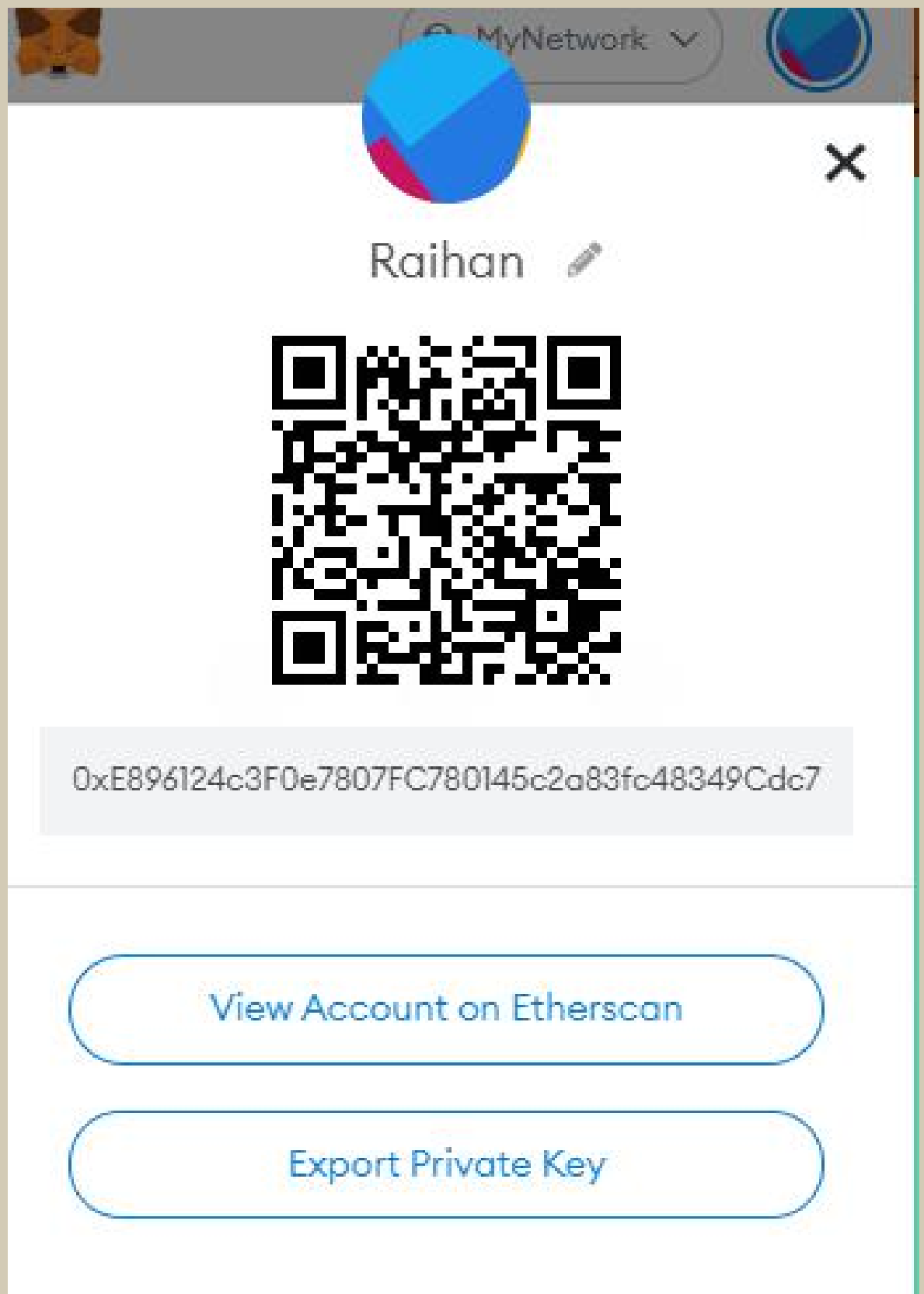


Fig 6.1: Account Details of User

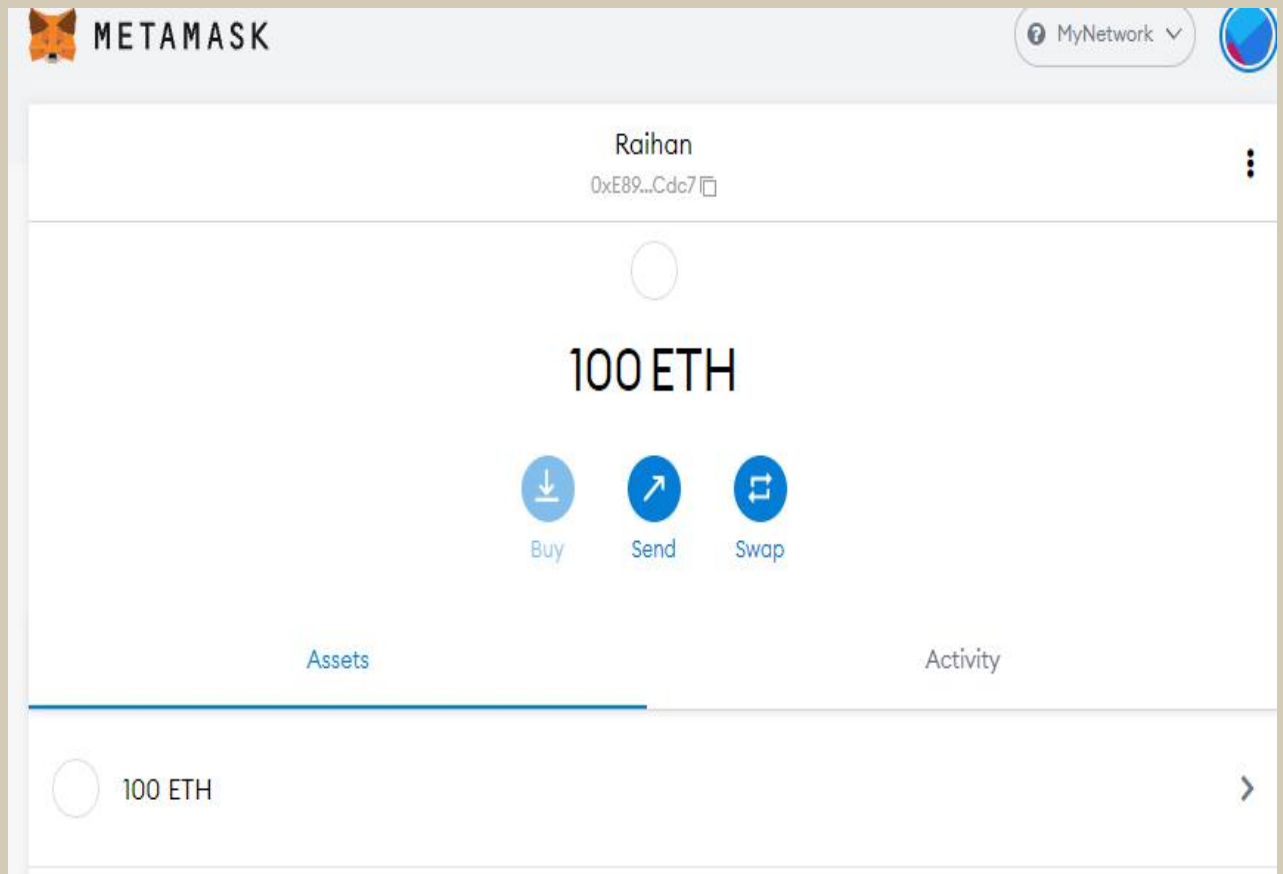





Fig 6.1:User Smart Wallet

Send


✕

Details

 From: 0xE89...Cdc7

>

 To: 0xD55...6B67

Transaction

Nonce	0
Amount	-0 ETH
Gas Limit (Units)	21000
Gas Used (Units)	21000
Gas price	20
Total	0.00042 ETH

Activity Log




-  Transaction created with a value of 0 at 15:50 on 11/4/2021.
-  Transaction submitted with estimated gas fee of 420000 GWEI at 15:50 on 11/4/2021.
-  Transaction confirmed at 15:50 on 11/4/2021.

Fig 6.1:User's Transaction with ethereum



Fig 6.1:Dinke-din's landing page

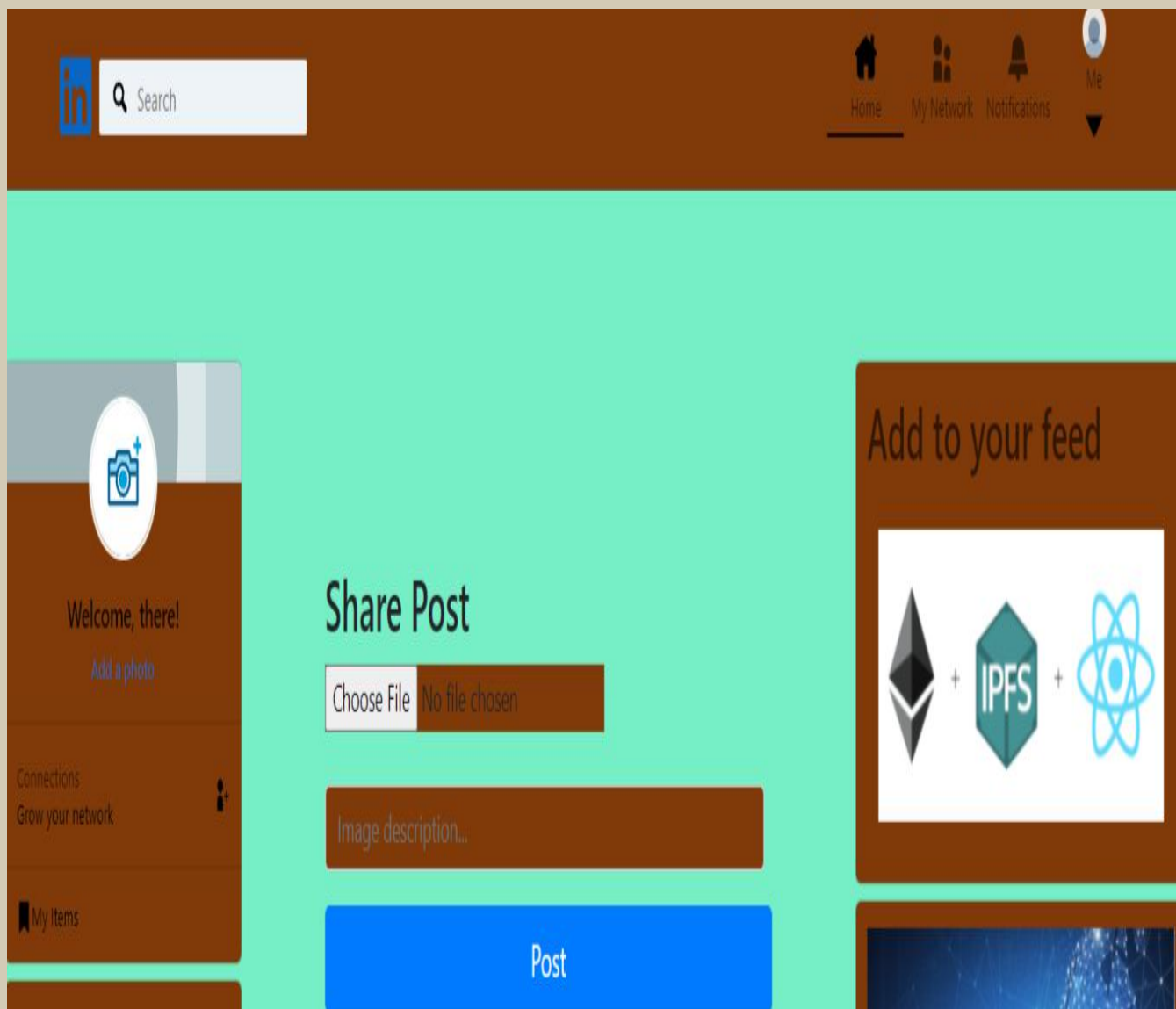


Fig 6.1:User post share options

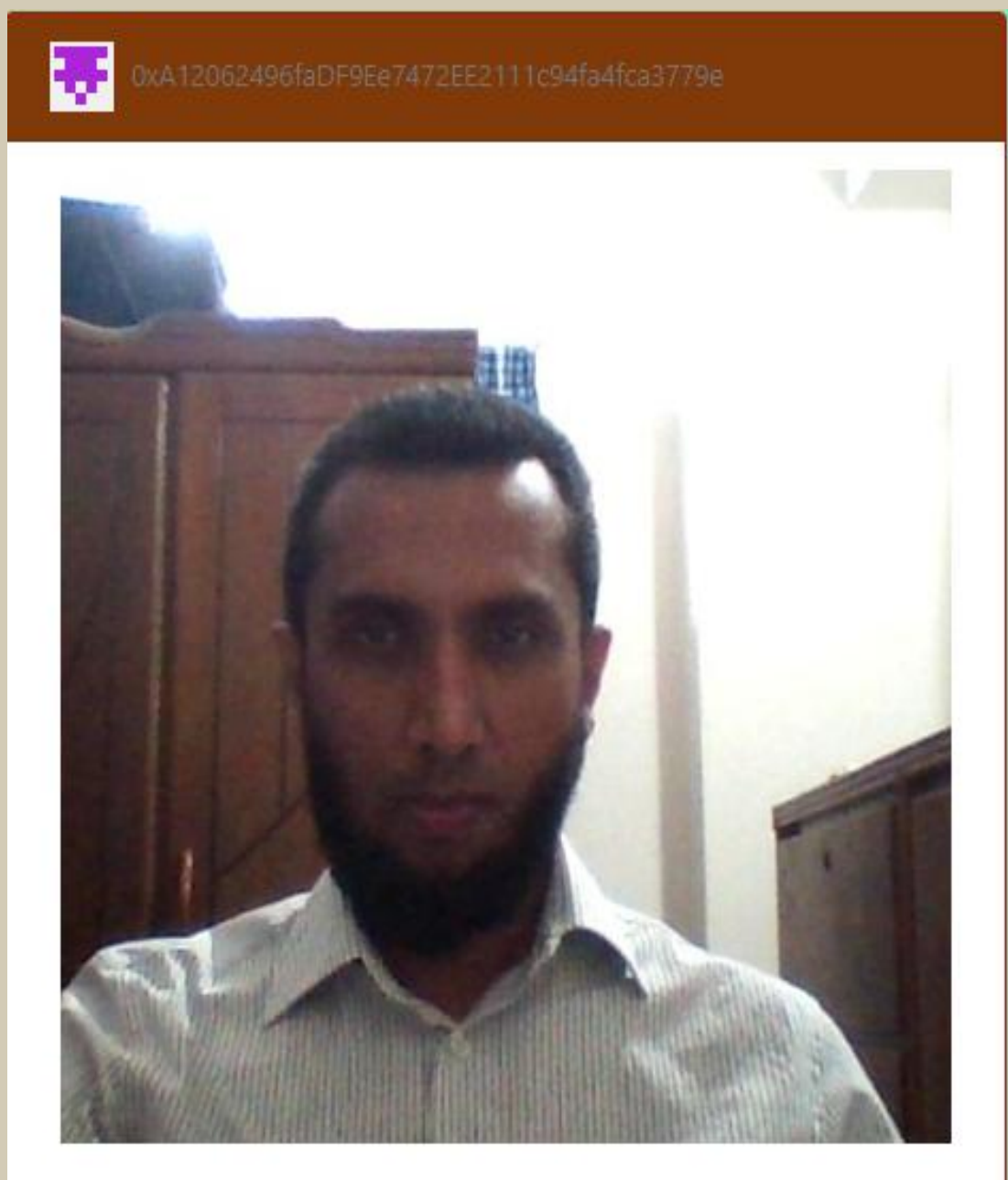


Fig 6.1: Shared post in Dinke-dn



Fig 6.1: Shared post in Dinke-dn

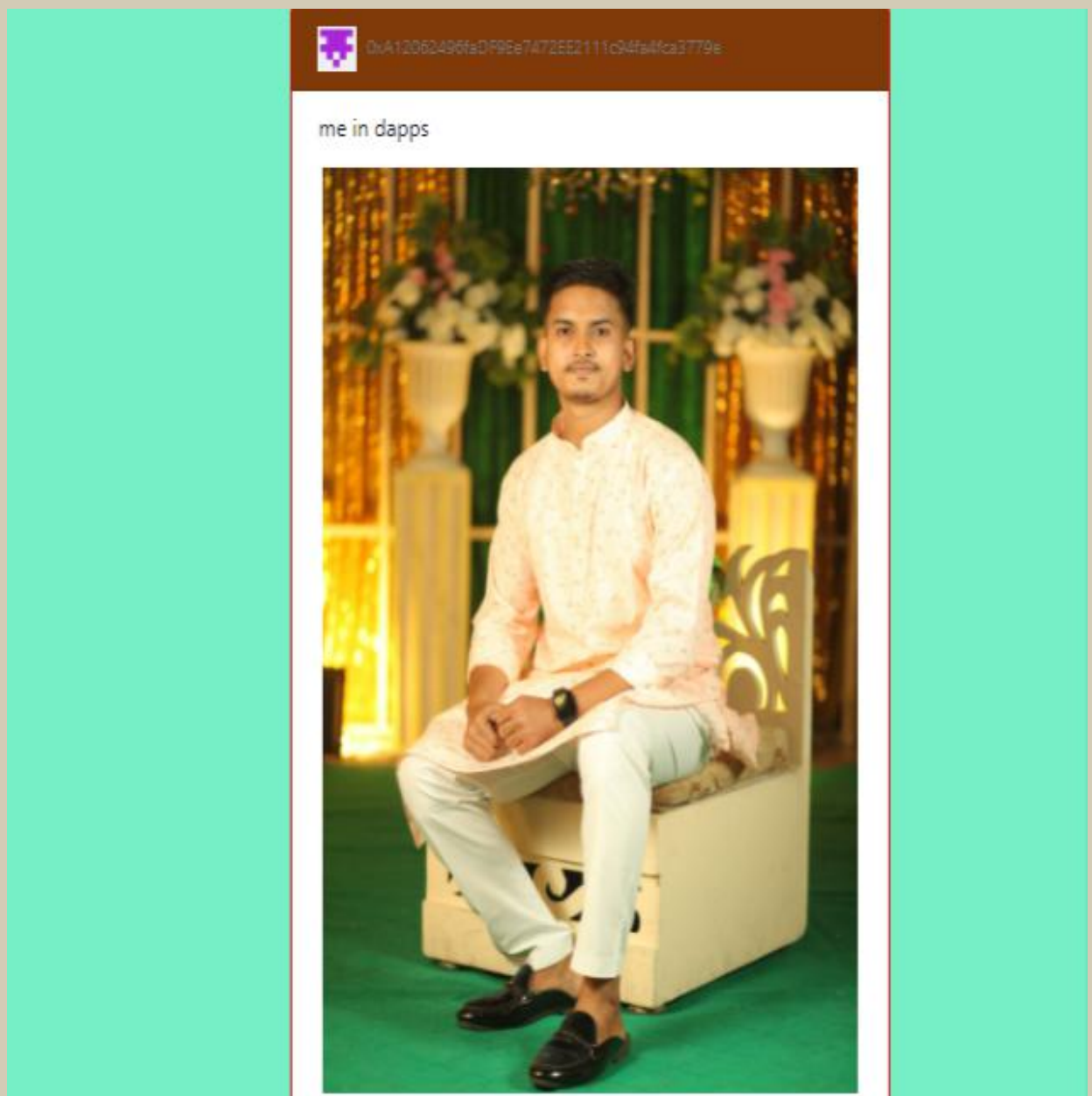


Fig 6.1: Shared post in Dinke-dn

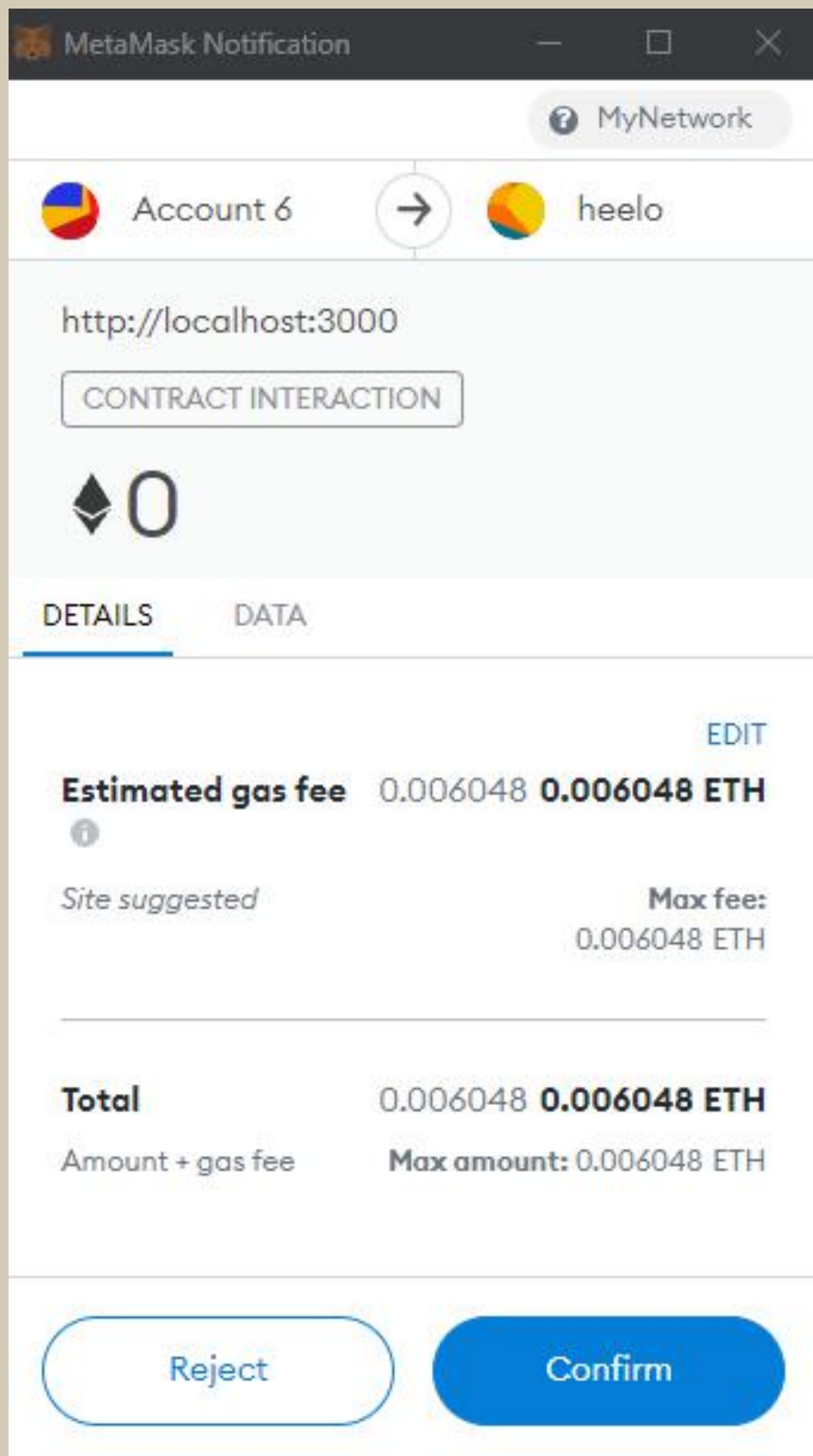


Fig 6.1: Confirmation of sharing post to Dinke-din with smart contract

Admin panel is in our Dinke-din system is like the role of Mark Zuckerberg in Facebook system . To get the data of all user Owner of Dinke-din system must have to have the Bearer Token to get the all data in the own system .

Fig 6.1:Admin Data pull from database using bearer Token

Fig 6.1:Admin Data pull from database using bearer Token

user	title	image	created at
0xA12062496faDF9Ee7472EE2111c94fa4fca3779e	I am Jamshedul Alam. I work as a Frontend Developer in this work.	QmYhTeDFSJYDwuWczBqMepjQBBWBuY8FhqReKo5745gAU4	19 minutes ago
0xA12062496faDF9Ee7472EE2111c94fa4fca3779e	Our sir , our inspiration	QmVs8jFUyUdoGDCQhq8UwdaRUEBfaAxy1kpqPHyyySb4LW	15 minutes ago
0xA12062496faDF9Ee7472EE2111c94fa4fca3779e	We are preparing for presentation.	QmaANq6REAV7LEqUzjH66yk4MizPKUWWAdGoqjLoYWPrvg	13 minutes ago
0xA12062496faDF9Ee7472EE2111c94fa4fca3779e	Our sir , our inspiration	QmVs8jFUyUdoGDCQhq8UwdaRUEBfaAxy1kpqPHyyySb4LW	5 minutes ago
0xA12062496faDF9Ee7472EE2111c94fa4fca3779e	I am Jamshedul Alam. I work as a Frontend Developer in this work.	QmYhTeDFSJYDwuWczBqMepjQBBWBuY8FhqReKo5745gAU4	3 minutes ago

Fig 6.1:Admin's All post show in Dinke-din

6.3 TESTING

Software testing is a process where in the software is evaluated to find the difference between the given input and expected output. Software testing is done to check if the software is working as expected or it is deviating from the original expectation defined. Accordingly to the SWEBOK: "Software testing consists of the dynamic verification that a program provides expected behaviors on a finite set of test cases, suitably selected from the usually infinite execution domain. "[10] .

6.3.1 Objectives of Testing

There have some different goals and objectives in software testing. The main objectives are as follows [16] -

1. To find failures and defects. .
2. Increase the likelihood that the test application will meet all of the requirements described
3. Increase the likelihood that the application intended for testing will work correctly under all circumstances
4. Helps to ensure that product is safe and secure for end-user/customer.

6.3.2 Testing Details

The testing methodologies are provided below:

- a) Unit testing with white testing
- b) Black box testing

1.12.1Unit Testing:

Test Case	Output	Expected Result
posts?id=`id`-10	As we send Get Method to the api back-end , it give us error .	As there is no such routes in our system it should return a routes error.

- We have debugged all inputs which are mandatory as user inputs. We have tested another important field, file input. If file upload manager accepts all file type without specific one, then it can cause security issue to our database and server. We got error exception in debugger while checked with invalid or unknown file types.

```
Submitting file to ipfs...
Ipfs result undefined
✖ ▶ Error: Input not supported. Expected Buffer|ReadableStream|PullStream|File|Array<Object> got undefined.
Check the documentation for more info https://github.com/ipfs/interface-js-ipfs-core/blob/master/SPEC/FILES.md#add
    at validateAddInput (0.chunk.js:78895)
    at Function.<anonymous> (0.chunk.js:73618)
    at 0.chunk.js:140821
    at Object.add (0.chunk.js:73642)
    at Main.uploadImage (main.chunk.js:1961)
    at onSubmit (main.chunk.js:2199)
    at HTMLUnknownElement.callCallback (0.chunk.js:150255)
    at Object.invokeGuardedCallbackDev (0.chunk.js:150304)
    at invokeGuardedCallback (0.chunk.js:150364)
    at invokeGuardedCallbackAndCatchFirstError (0.chunk.js:150379)
    at executeDispatch (0.chunk.js:154614)
    at processDispatchQueueItemsInOrder (0.chunk.js:154646)
    at processDispatchQueue (0.chunk.js:154659)
    at dispatchEventsForPlugins (0.chunk.js:154670)
    at 0.chunk.js:154881
    at batchedEventUpdates$1 (0.chunk.js:168566)
    at batchedEventUpdates (0.chunk.js:150053)
    at dispatchEventForPluginEventSystem (0.chunk.js:154880)
    at attemptToDispatchEvent (0.chunk.js:152363)
    at dispatchEvent (0.chunk.js:152281)
    at unstable_runWithPriority (0.chunk.js:189631)
    at runWithPriority$1 (0.chunk.js:157661)
    at discreteUpdates$1 (0.chunk.js:168583)
    at discreteUpdates (0.chunk.js:150065)
    at dispatchDiscreteEvent (0.chunk.js:152247)
```

```
Error: Request failed with status code 500
    at createError (0.chunk.js:19327)
    at settle (0.chunk.js:19561)
    at XMLHttpRequest.onloadend (0.chunk.js:18730)
```

Fig 6.35: connection request execution testing output

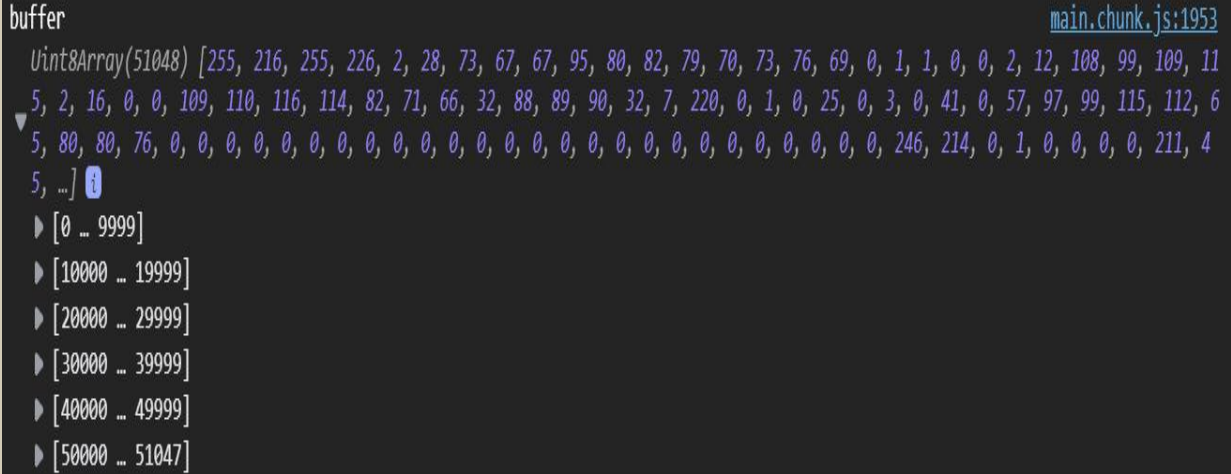
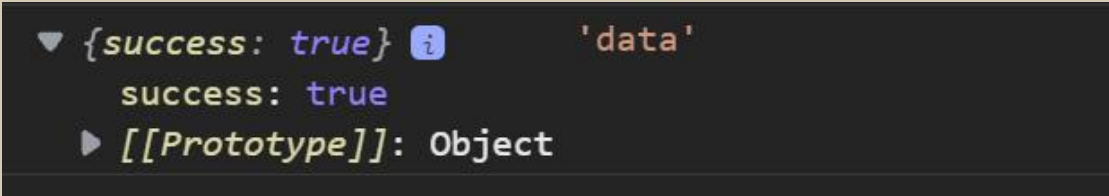
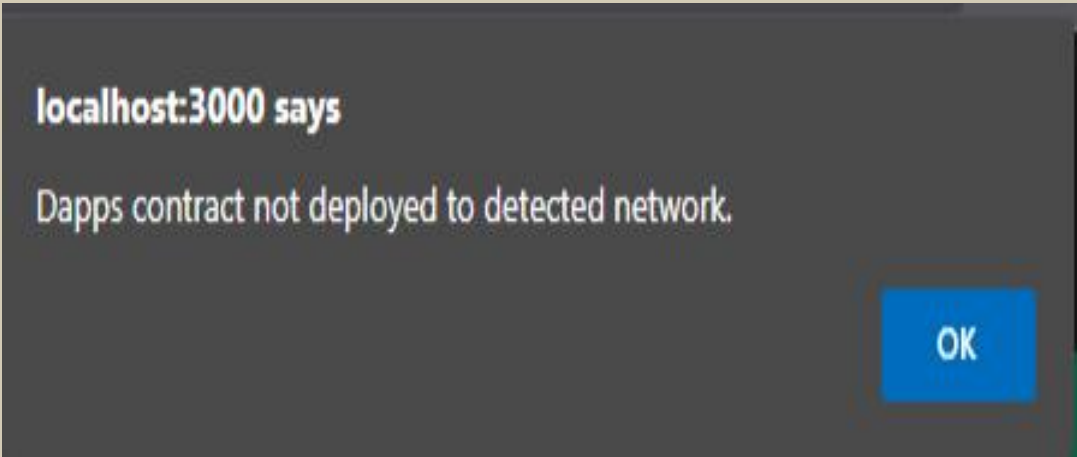


Fig 6.34: File type checking output



- if the website will not deploy to blockchain , then the error will showed in there to notify the user .



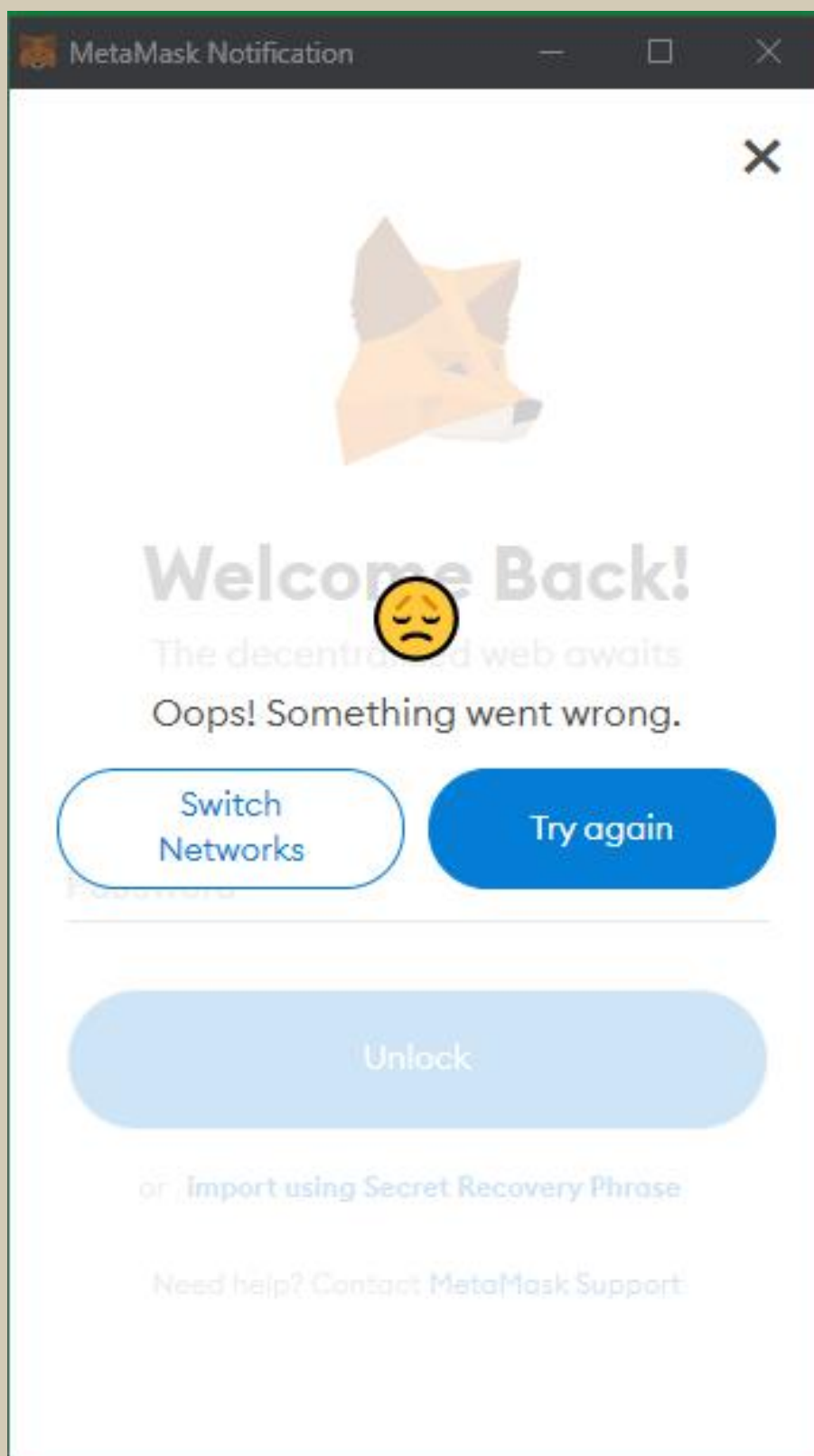


Fig 6.36: Output of manual connection check of application

Testing Black Box

Test Case	Output	Expected Result
posts?id=10	Http Get will be sent and it is returning about post	It should be returning with data.
User = admin Pass = admin	Http post method was sent and then it is returning a success messages .	Should return in ok from back-end .

/posts	Request for all post in return .	Should return all data about posts.
Header: user.token	It is allowing the user to take control of the data and admin panel .	Should recongize as a familiar user.

CHAPTER VII

CONCLUSION & FUTURE PLAN

7.1 CONCLUSION

Our Dinke-din was a product to ensure our security in online social network as a sample we choose Linke-din. Our Dinke-din is a solution to the social network data losing and data corruption .In Dinke-din , we try to solve the most devating talk we always have with each other about what we sharing in the Dinke-din is it safe? Will Dinke-din Owner share the all user data and user info to the thrird party ?

To solve this problem in our Dinke-din we uses IPFS blockchain system .

7.2 CONTRIBUTION OF THE PROJECT

The proje ,BlockChain helps to solve the security problem about user's shared data and user data. But we make a separate cloud system where we stored the ipfs returned hash value in database .In this cloud system whenever a user post in our Dinke-din , this post account address , image ipfs hash address , description of the post and date of the post will be stored in the database . The most intersting thing about that is that it is a place for user and owner of Dinke-din .Here if any party want the post data or user data he must have to bearer token in his hand , without that he will be detected as a evil user.

7.3 FUTURE PLAN

- We have plan to make the Dinke-din platform more secured for the user.
- We are planning to add new feature to the Dinke-din so that Dinke-din becomes the best version of it.

REFERENCES

- [1] What is blockchain ? what is the future of it [8 Reasons Why Blockchain Technology Is the Future | upGrad blog](#)
- [2] What is testing [What is software testing? - Quora](#)

- [3] Adriano Witkovski and Altair Santin “An IdM and Key-Based Authentication Method for Providing Single Sign-On in IoT ”.
- [4] . Swapnil Ninawe and Pallapa Venkataram “Authentication schemes for social network users: a review ”.

The End