LOYALTY NETWORK ON ETHEREUM BLOCKCHAIN

MD TAUSEEF ALAM
(Roll No.: 17MCA025)

DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF NATURAL SCIENCES
JAMIA MILLIA ISLAMIA
May, 2020

Loyalty Network on Ethereum Blockchain

by

Md Tauseef Alam
(Enrollment No.: 17-0692**)**

Submitted
in partial fulfillment of the requirements of the degree of

Master of Computer Applications (MCA)

to the

Department of Computer Science
Faculty of Natural Sciences
Jamia Millia Islamia
New Delhi – 110025

May, 2020

Declaration

I, MD TAUSEEF ALAM, student of M.C.A hereby declare that the project report entitled "**Loyalty Network on Ethereum Blockchain**" which is submitted by me to the Department of Computer Science, Jamia Millia Islamia, New Delhi, in partial fulfillment of the requirement of the degree of **Master of Computer Applications (MCA)**, is a record of the original bonafide work carried out by me and have not been submitted in part or full to any other university or institute for the award of any degree or diploma.

Md Tauseef Alam
Roll No.: 17MCA025
Enrollment No.: 17-0692

Certificate

On the basis of declaration made by the student **Md Tauseef Alam,** I/we hereby certify that the project report entitled “**Loyalty Network on Ethereum Blockchain”** submitted by **Md Tauseef Alam** to the Department of Computer Science, Jamia Millia Islamia, New Delhi, for the partial fulfillment of the requirements of the degree of **Master of Computer Applications (MCA)**, is carried out by him under my/our guidance and supervision. The report has reached the requisite standards for submission.

Prof. S.M.K. Quadri(HOD)
(Dept. of Comp. Sc. ,JMI)
(Internal Supervisor)

B Vinoth Raj
(Mindfire Solutions)
(External Supervisor)

ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

The Project was undertaken by **Md Tauseef Alam** as his final semester Industrial Project, under the able guidance and supervision of **Dr. S.M.K Quadri(Internal) and B Vinoth Raj(Exterrnal)**. My primary thanks goes to them, who poured over every inch of the project with painstaking attention and helped throughout the working of the project. It's my privilege to acknowledge my deepest sense of gratitude to them for their inspiration which has helped immensely. I am extremely grateful to them for their unstilted support and encouragement in the preparation of this project. I would also thank my team members Asmita Kumari and Pardeep for their guidance throughout the project.

TABLE OF CONTENTS

ABSTRACT

The main goal of this work is the implementation of Loyalty points and reward network on ethereum blockchain platform. A loyalty network with multi-brand, cross-industry support could be implemented with blockchain technology as a backbone. A customer with one digital wallet that can be used across various brands can lead to better customer experience and engagement. Because brands are unique by its nature, each brand may have its own smart contract and point system. The loyalty programs can be coded in smart contracts and customer will access any of them with one single digital wallet. A rewards-exchange mechanism can also be implemented in a smart contract that can facilitate exchanging rewards.

Keywords: **Ethereum, Blockchain, Loyalty, Smart Contract, Solidity**

LIST OF FIGURES

ABOUT ORGANIZATION

Mindfire Solutions started in October of 1999, with the purpose of providing expert software services globally, and has steadily grown to its 1000-seat facilities at 2 engineering centers.

In its heart and soul, Mindfire is a software service provider, with unrelenting focus on small-team offshore software development using Agile methods for distributed teams, all amid a unique Mindfire culture.

We are clear in our vision of building a software engineering powerhouse, and we do not spend time and energy in activities which are not our core competence.

Our vision is "to be a globally respected, professional and innovative software services and technology company".

Our mission statement reinforces our vision: "Imagine. Think. Plan. Act. Deliver. Improve.".

During past 20 years, we have successfully delivered 2000+ engagements with 500+ clients.

A Short History of Time

Soon after we started in 1999, there was the huge hullabaloo of Y2K which fizzled out. We got a few projects and worked diligently, taking each day as it came while keeping an eye on the future.

Then the dot-com bust and 9/11 happened. The IT/software industry in India saw massive slowdown, and a number of companies shut down. We trudged on.

We stabilized and started growing in the mid-2000s again. Having survived the early-2000 downturn, we were determined to not let future events affect us. When the 2009 recession hit, we came through unscathed.

From a position of doing anything we came across (could start-ups be choosers?), we changed course in 2013. We took a strong stand that we would engage only with quality clients, do quality work, and hire quality people.

Thus we stand here today, a focused group of dedicated people, an organization that proudly chose to stick to its knitting!

Some Spectacular Numbers

We have a few clients who have been working with us back from when we started (that's 20 years!), and a good number of clients who have worked with and loved us for more than 10 years!

We have worked with clients across generations. Seriously, some have spanned generations of a family!

A band of 10 people have been with Mindfire since the beginning! 10% of our people have been with us for more than 10 years! And 40% of our people have been at Mindfire for at least 5 years!

Oh, and about 10 couples found and married each other within Mindfire! Talk about varied kinds of success ;)

These are astounding numbers when you consider churn in tech - both clients and people change preferences as soon as something is anything less than ideal!

Source: https://www.mindfiresolutions.com/

CHAPTER 1
INTRODUCTION

1.1 Background and Objectives

A loyalty program is a marketing strategy designed to encourage customers to still patronize or use the services of a business related to the program. Today, such programs cover most sorts of commerce, each having varying features and rewards schemes, including in banking, entertainment, hospitality, retailing and travel.

According to Colloquy Loyalty Census 2017, sadly 57% of the surveyed consumers did not even know their point balance. Around 53% of the consumers weren't even interested in the type of rewards offered. The Loyalty Program industry is already prospering with a mammoth value of over US $360 billion.

All these facts point during a direction that the Loyalty Program today need a revolution to eliminate the stagnation within the industry. Blockchain can bring a breakthrough within the Loyalty Program industry by revolutionizing how loyalty program rewards work. A Ethereum based Blockchain Loyalty Program would address numerous issues that might lure customers faraway from traditional loyalty programs.

Loyalty rewards program industry is big and much of inefficiencies and problems are plaguing the industry. Gartner reports that over half loyalty program memberships are inactive and nearly a 3rd of consumers don't redeem the collected rewards. The fragmented industry, as evident from disparate programs available, is one among the most reasons for the customer inactivity.

So the objective of this project is:

A loyalty network with multi-brand, cross-industry support might be implemented with blockchain technology as a backbone. A customer with one digital wallet which will be used across various brands can cause better customer experience and engagement. Because brands are unique by its nature, each brand may have its own smart contract and point system. The loyalty programs are often coded in smart contracts and customer will access any of them with one single digital wallet. A rewards-exchange mechanism also can be implemented during a smart contract which will facilitate exchanging rewards.

1.2 What is Blockchain?

A blockchain is a decentralized, distributed and public digital ledger that's wont to record transactions across many computers in order that the record can't be altered retroactively without the alteration of all subsequent blocks and therefore the collusion of the network.

Decentralization of information:

Information persisted a blockchain exists as a shared and continually reconciled database. The blockchain database isn't stored in any single location, meaning the records it keeps are truly public and simply verifiable. No centralized version of this information exists for a hacker to corrupt. Hosted by many computers simultaneously, its data is accessible to anyone on the web . Since its data is accessible to anyone on internet it's distributed to several peers(nodes).

Public Shared Digital Ledger:

A ledger is that the principal book or file for recording and totaling economic transactions measured in terms of a unit of measurement of account by account type, with debits and credits in separate columns and a beginning monetary balance and ending monetary balance for every account. When the ledger is out there to everyone and it's in digital form it's referred to as Public Digital Ledger.

Why Blockchain for the project:

i. Connecting a disconnected world.
ii. Reducing costs.
iii. Enabling a friction less system.
iv. Making the process near real time.
v. Providing a secure environment.
vi. Creating unique business opportunities.

To understand the deep concept and idea about blockchain one can refer the books and sites in the references section.

1.3 Ethereum Blockchain Platform

Website:	www.ethereum.org
Written in:	C++, Go, Rust
Initial release:	30 July 2015
Repository:	github.com/ethereum
Original author(s):	Vitalik Buterin, Gavin Wood

The intent of Ethereum is to create an protocol for building decentralized applications, providing a different set of tradeoffs that we believe will be very useful for a large class of decentralized applications, with particular emphasis on situations where rapid development time, security for small and rarely used applications, and the ability of different applications to very efficiently interact, are important. Ethereum does this by building what is essentially the ultimate abstract foundational layer: a blockchain with a built-in Turing-complete programming language, allowing anyone to write smart contracts and decentralized applications where they can create their own arbitrary rules for ownership, transaction formats and state transition functions. A bare-bones version of Namecoin can be written in two lines of code, and other protocols like currencies and reputation systems can be built in under twenty. Smart contracts, cryptographic "boxes" that contain value and only unlock it if certain conditions are met, can also be built on top of the platform, with vastly more power than that offered by Bitcoin scripting because of the added powers of Turing-completeness, value-awareness, blockchain-awareness and states.

The design behind Ethereum is intended to follow the following principles:

1.**Simplicity**: The Ethereum protocol should be as simple as possible, even at the cost of some data storage or time inefficiency. An average programmer should ideally be able to follow and implement the entire specification, so as to fully realize the unprecedented democratizing potential that cryptocurrency brings and further the vision of Ethereum as a protocol that is open to all. Any optimization which adds complexity should not be included unless that optimization provides very substantial benefit.

2.**Universality**: A fundamental part of Ethereum's design philosophy is that Ethereum does not have "features". Instead, Ethereum provides an internal Turing-complete scripting language, which a programmer can use to construct any smart contract or transaction type that can be mathematically defined. Want to invent your own financial derivative? With Ethereum, you can. Want to make your own currency? Set it up as an Ethereum contract. Want to set up a full-scale Daemon or Skynet? You may need to have a few thousand interlocking contracts, and be sure to feed them generously, to do that, but nothing is stopping you with Ethereum at your fingertips.

3.**Modularity**: The parts of the Ethereum protocol should be designed to be as modular and separable as possible. Over the course of development, our goal is to create a program where if one was to make a small protocol modification in one place, the application stack would continue to function without any further modification. Innovations such as Ethash , modified Patricia trees and RLP should be, and are, implemented as separate, feature-complete libraries. This is so that even though they are used in Ethereum, even if Ethereum does not require certain features, such features are still usable in other protocols as well. Ethereum development should be maximally done so as to benefit the entire cryptocurrency ecosystem, not just itself.

4.**Agility**: Details of the Ethereum protocol are not set in stone. Although we will be extremely judicious about making modifications to high-level constructs, for instance with the sharding roadmap, abstracting execution, with only data availability enshrined in consensus. Computational tests later on in the development process may lead us to discover that certain modifications, e.g. to the protocol architecture or to the Ethereum Virtual Machine (EVM), will substantially improve scalability or security. If any such opportunities are found, we will exploit them.

5.**Non-discrimination** and **non-censorship**: The protocol should not attempt to actively restrict or prevent specific categories of usage. All regulatory mechanisms in the protocol should be designed to directly regulate the harm and not attempt to oppose specific undesirable applications.

1.4 Organisation of the project

This project is made because of the problems faced by loyalty program provided by company.

The project is divided into five chapters. The first chapter dealt with the intoduction of what is blockchain. What is loyalty rewards and what is the issue faced by it and how blockchain come into picture to solve the existing issues.

In the second chapter the existing work done in this field is dealt with. The projects like Qiibee, Singapore Airlines' Loyalty Reward Program, American Express Membership Rewards Program, Vexanium(It is an ecosystem), Pei , Coinbase cryptocurrency exchange for goods and services are studied and analysis of these are mentioned.

In third chapter the approach to achieve the goal is discussed like which erc standard to follow, what language or library should be used for the front end part, what language to follow for writing smart contract and the tools description used to give a local environment to test the project.

In the fourth chapter the implementation part is dealt along with screenshot of working project.

And in the final chapter that is fifth chapter conclusion of the project is done.

Hardware Requirements:

1. Processor -Intel Core i5-6200U CPU – 2.5 GHz
2. RAM – 6 GB
3. HDD – 250 GB
4. 64 bit Operating System

Software Requirements:

1. Node8 or greater.

2. VSCode

3. Online Remix IDE

4. @metamask/onboarding version 0.2.1

5. ethers version 4.0.47

6. pdf-viewer-reactjs version 2.0.6

7. react version 16.8.4

8. express version 4.17.1

9. nedb version 1.8.0

10. MythX(for contracts static analysis)

11. Solidity Compiler

CHAPTER 2
RELATED WORKS

2.1 Issues with Loyalty Rewards

Customers loyalty and engagement can be a groundbreaking thing for an organization. Customers loyalty programs are ailing thanks to numerous inefficiencies like low customer retention and redemption rates etc. Customers can redeem points before they expire and keep track of multiple loyalty programs very cumbersome. Low redemption rates are a big hurdle for loyalty marketers.

Customers wish for more flexibility in the Loyalty Program system so that they can opt among several types of rewards offered by the organizations. In traditional Loyalty Program system program providers maintain control regarding how points are valued and dispersed and other elements. Customers wish for flexibility on how and where they use their reward points. Harvard Business Review stated that to set up a cross-border loyalty program might increase the customer acquisition rate by 50% .

1. Low customer retention and redemption rates because of expiry date and various number of points at various places.
2. Independence on how and where to use the reward points is not provided by organizations.

2.2 Qiibee

Descriptions:

With qiibee, business round the world can run their loyalty programs on the blockchain. Easy to use solutions set the new global standard for loyalty on the Blockchain. The permission based qiibee blockchain platform is designed for high speed transactions at low cost.

Features:

1. qiibee allows businesses to run their own branded loyalty programs on the blockchain, creating custom tokens as rewards and helping merchants with setup and rollout.

2. The company Sausalito and Latesso are also examples of new companies what work with the program. and their coins that are also tied to the QBX tokens.

3. Another company Geschenkt is a campaign run by KKiosk and press&books at over 750 POS with 650’000 daily customers. In which also qiibee makes it possible to run the entire campaign including the creation of Gesschenkt coin.

Link:

1.https://qiibee.com/

2.https://static.qiibee.com/qiibee-Lattesso-Roll-Out.pdf

Lattesso is the second largest cold coffee producer in switzerland, serving millions of customer around the world.By combining loyalty network with blockchain technology, Lattesso maintains the complete flexibility over its reward and redemption structure and achieves even higher levels of engagement by the purchasers .

Features:

1. Lattesso is the second largest cold coffee producer in switzerland, serving millions of customers.

2. When customer buy coffee, each cup contains a special code on the lid that customer can use to earn Lattessocoins.

3. This code can be redeemed for Latessocoinss(LTS) on latessocoin.com.

4. Users are required to sign in to receive and store their LTS.

5. If the code is valid, the Lattesscoin will be credited to the user's wallet.

6. Where they can collect, save and view the value developments of their accounts.

7. Customer can use these token to use later when they again buy coffee or they also transfer these coins to their friends.

8. Lettesso created Lattesocoin based on the qiibee platform.

9. Lattesso , in backend uses qiibee wallet address.

Links:

1. **https://www.lattesso.com/**

A popular German Cocktail Bar known as **Sausalitos** has nearly 4 million yearly visitors, 41 restaurants, more than 900 employees with establishment across the country. The German enterprise is now starting its loyalty based program with a cryptocurrency based on the popular blockchain.

Properties:

1. The new program is run on the ERC20 platform.
2. The rewards will be given through the company's infrastructure of the restaurant's partner qiibee.
3. A simple to use plug and play platform - ideal for the loyalties that are paid out via Etherum's network.
4. New customer who use the app simply download and start getting cryptocurrency with each purchase made using a QR code.
5. They can also then exchange the tokens for cash or loyalty tokens taken from other brands before eventually turning them into other cryptocurrencies or altcoins.

Link:

1. https://bitcoinexchangeguide.com/sausalitos-coin/

2. https://www.sausalitos.de/

2.3 Singapore Airlines' Loyalty Reward Program

Singapore Airlines, a leading airline operator, has launched their own updated loyalty program: KrisPay for the users of its KrisFlyer scheme The intent wasn't just to streamline an existing system but to give their frequent fliers more flexibility and ownership of their accumulated points.

Features:

•Earn air miles (Air miles are points that you collect when you buy certain goods or services and which you can use to pay for air travel) when you fly with Singapore Airlines.

•Through KrisPay's digital blockchain wallet, air miles are able to be exchanged for digital currency which, in turn, can be spent at an extensive list of partners, including hotels, petrol stations, and retail establishments.

Links:

•https://www.singaporeair.com/en_UK/sg/ppsclub-krisflyer/

2.4 American Express Membership Rewards Program

The Membership Rewards program offers rewards for customers. Earn points on most everyday purchases, then use points for gift cards, travel rewards, immersive experiences, or maybe to hide recent Card charges. And with no expiration dates and no limit on what percentage you'll earn, Membership Rewards points are easy to earn and versatile to use, so you never need to compromise.

Features:

•Just use your Card for purchases, large or small, and watch the points add up with every purchase.

•There's no limit to how many points you can earn.

•Don't worry about expiration dates because there aren't any.

•Use Membership Rewards points for the loyal customers.

Links:

https://www.americanexpress.com/us/rewards/membership-rewards/about/

2.5 Vexanium(It is an ecosystem)

Key points:

1. The first Indonesia public blockchain, Vexanium are building the next generation blockchain for mass adoption, that is born to support DApps (Decentralized Application) usability and retail

penetration.

2. It does have any transaction fee,and can accomodate 2000 transaction per second.

3. It uses Delegated Proof Of Stakes consensus which means Vexanium will prioritize scalability and network performance by using block producers.

4. vexanium loyalty program : https://www.vexanium.com/loyalty

5. Use case of the VEX Platform is the lucrative “airdrop” market, which will allow blockchain companies to create airdrop campaigns for acquiring new customers and rewarding existing ones, using the VEX token.

Links:

1. https://tokennews.asia/2018/04/23/its-not-just-a-loyalty-program-its-an-ecosystem/
2. https://www.vexanium.com/

2.6 Pei

Link your card. Spend as you would at favorite spots. Receive automatic cashback in Bitcoin or cash.

Key-points:

1. Pei makes earning rewards easy and seamless. Simply link your cards to your Pei account. The next time you shop at a partner merchant using that linked card, Pei will automatically return your cashback rewards.
2. For queries link **https://peitechnology.zendesk.com/hc/en-us/**
3. Keep track of finances with Pei.
4. Pei is constantly offering new cashback with Uber,Lyft,Puma etc.

Links:

1. **https://medium.com/pei-technology-inc**
2. **https://getpei.com/**

2.7 Coinbase cryptocurrency exchange for goods and services

Company/Organization involved: Coinbase and Wegift

Key points:

1. Coinbase customers in the EU and Australia are able to instantly spend their cryptocurrency balances on e-gift cards, making it the first trading platform to offer direct withdrawals into e-gift cards.
2. 120 retailers have partnered with the new platform, among which are such major brands as Nike, Uber, Google Play, Tesco or Costa.
3. The exchange is famous for its selection of digital assets offered for trading with only bitcoin, ethereum, bitcoin cash and litecoin available for its customers, compared to hundreds of digital assets on its competitors.

Links:

1. https://toshitimes.com/coinbase-releases-a-crypto-gift-card-service-for-customers-in-europe-and-australia/

CHAPTER 3

PROPOSED APPROACH

3.1 Blockchain Platform-Ethereum (Metamask , Ganache & Solidity)

Ethereum may be a decentralized blockchain platform for "building unstoppable applications", while Ether is that the cryptocurrency used on this platform.

Metamask:

Metamask may be a cryptocurrency wallet which may be used on the Chrome, Firefox and Brave browsers. It’s also a browser extension. This means that it works sort of a bridge between normal browsers and therefore the Ethereum blockchain.
The Ethereum blockchain is a network where users can build their own apps (which are called dApps) and cryptocurrencies. Ethereum also allows its users to write down transaction guidelines called smart contracts. MetaMask are often wont to store keys for Ethereum cryptocurrencies only.
So, the MetaMask wallet are often used for storing keys for Ether and ERC20 tokens on three different web browsers. It also allows users to browse the Ethereum blockchain from a typical browser. MetaMask requires no login and does not store your private keys in any server, instead they are stored on Chrome and password protected.

***Advantages**:*

- Open source: this suggests that each one the MetaMask code is online and liberal to access. You can build your own version at home! I’m kidding, don’t do that. I mean, you could but don’t. Open-source software can be reviewed and updated by the community, meaning that it can be continuously improved.
- HD settings: Hierarchical deterministic settings help users backup their accounts. They do this by giving the user a list of words called seed phrases.
- Built-in coin purchasing: MetaMask links directly with two exchanges where users can purchase cryptocurrency. Users can select Coinbase to get Ether and ShapeShift to shop for Ether or ERC20 tokens.
- Customer support: MetaMask wants to urge as many of us involved the Ethereum network because it can. It has a video introduction on its homepage, and an in depth support page.
- Simple interface: Once it’s found out , MetaMask is extremely simple to use. All of its features are laid out clearly so sending and receiving currency is easy, even for beginners!

- Local key storage: Some wallet providers store keys on their own servers. This is common on exchanges which provide wallets, like Coinbase. MetaMask keys are stored on the user's own browser, not on any remote servers, so it is secure. This gives the user more control over their public and private keys.
- Community: MetaMask is a crucial a part of the Ethereum community. It's got more than a million active users and has over 40 thousand followers on Twitter

MetaMask wallet disadvantages:

- Browser access: MetaMask doesn't have access to any of your information but the browser it's installed on will. Your browser won't have access to your private codes but it may collect information about when and how you use the app. Mozilla and Google aren't very popular in the crypto community. Many crypto users will feel uncomfortable allowing these companies to gather information about them. This may stop some users from trying the MetaMask wallet.
- Online: Online wallets have advantages and disadvantages. One of the main disadvantages is security. Any information that's stored online is more at risk from hackers than information that's stored offline. MetaMask doesn't provide enough security by itself. Remember; always use quite one quite wallet.

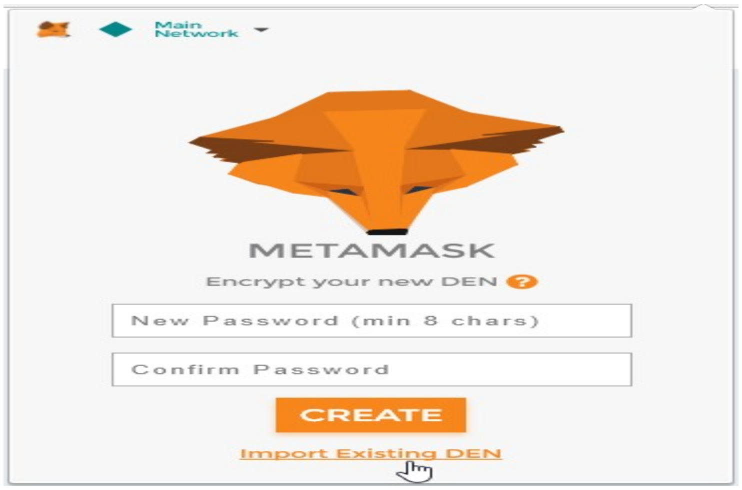
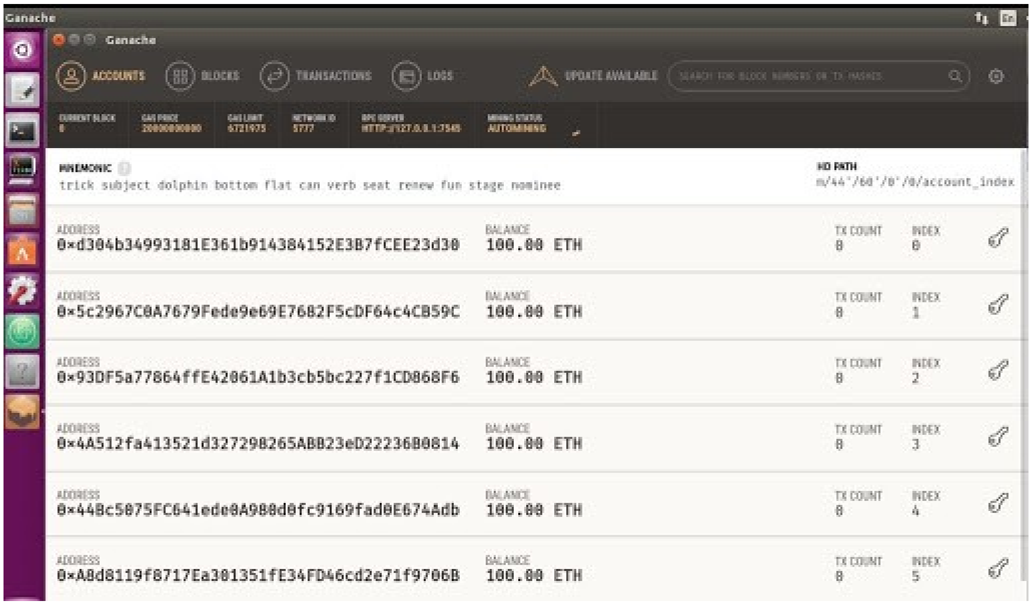


Fig- 3.1.1 Metamask wallet

Ganache:

Ganache is a personal blockchain for rapid Ethereum and Corda distributed application development. You can use Ganache across the entire development cycle; enabling you to develop, deploy, and test your dApps in a safe and deterministic environment.

Ganache UI is desktop application supporting both Ethereum and Corda dapp. In addition, an Ethereum version of ganache is available as a command-line tool: ganache-cli (formerly known as the TestRPC). All versions of Ganache are available for Windows, Mac, and Linux.



CURRENT BLOCK	GAS PRICE	GAS LIMIT	NETWORK ID	RPC SERVER	MINING STATUS
0	20000000000	6721975	5777	HTTP://127.0.0.1:7545	AUTOMINING

MNEMONIC
trick subject dolphin bottom flat can verb seat renew fun stage nominee

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0x d304b34993181E361b914384152E3B7fCEE23d30	100.00 ETH	0	0
0x5c2967C0A7679Fede9e69E7682F5cDF64c4CB59C	100.00 ETH	0	1
0x93DF5a77864ffE42061A1b3cb5bc227f1CD868F6	100.00 ETH	0	2
0x4A512fa413521d327298265ABB23eD22236B0814	100.00 ETH	0	3
0x448c5075FC641ede0A980d0fc9169fad0E674Adb	100.00 ETH	0	4
0xA8d8119f8717Ea301351fE34FD46cd2e71f9706B	100.00 ETH	0	5

Fig- 3.1.2 Ganache Network

Solidity:

Solidity is an object-oriented programing language for writing smart contracts. Solidity is used for implementing smart contracts on various blockchain platforms, such as most Ethereum. It was developed by Christian Reitwiessner, Yoichi Hirai ,Alex Beregszaszi and variety of other former Ethereum core contributors to enable writing smart contracts on blockchain platforms like Ethereum, Quorum.

Solidity may be a statically-typed programing language designed for developing smart contracts that run on the EVM. Solidity is compiled to bytecode that's executable on the EVM. With Solidity, developers can able to develop applications that implement self-enforcing business logic embodied in smart contracts, leaving a non-repudiable and authoritative record of transactions. Writing smart contracts in smart contract specific languages like Solidity is mentioned as easy (ostensibly for those that have already got programming skills).

3.2 ERC Standards

There are a number of ERC standards available but we choose ERC20 Lets explore it:

ERC20

Description:
ERC20 is the standard for transferable fungibles in ethereum.In 'ERC20', ERC stands for Ethereum Request For Comments and 20 stands for a unique ID number to distinguish this standard from others.ERC-20 has emerged as the technical standard used for most of smart contracts on the Ethereum blockchain for token implementation. As of April 16, 2019, more than 181,000 ERC-20-compatible tokens exist on Ethereum main network.ERC20 tokens follow a list of standards so that they can be shared, exchanged for other tokens, or transferred to a crypto-wallet. This token protocol became popular with crowdfunding companies via Initial Coin Offering. In simple words, ERC-20 is a guide of rules and regulations that will help create a blueprint for Ethereum-based smart contracts to create their tokens.
Features:
ERC20 contract, like the standard itself, is quite simple and bare-bones. In fact, if we try deploy an instance of ERC20 as-is it will be quite literally useless… it will have no supply! What use is a token with no supply? The way that provide is made isn't defined within the ERC20 document. Every token is free to experiment with their own mechanisms, ranging from the most decentralized to the most centralized, from the most naive to the most researched, and more. The ERC20 standard consists of three optional rules and 6 mandatory rules. The mandatory rules are as follows:

- 1.totalSupply
- 2.balanceOf
- 3.transfer
- 4.transferFrom
- 5.approve
- 6.allowance

On the other hand, the optional rules are:

- •1.Token Name
- •2.Symbol
- •3.Decimal (up to 18)

and two events:

- •1.Transfer
- •2.Approve

for details check the link: https://eips.ethereum.org/EIPS/eip-20

ADVANTAGES:

1.Makes assets interchangeable;

2.Used on compatible platforms, projects, exchanges;

3.Ensures operation with compatible decentralized applications (DApps);

4.Simplifies transactions with the receipt and sending of coins;

5.Interacts with other currencies, smart contracts;

The developers of the Ether-based token with the specified parameters can be any person who is going to create his/her own project. It can easily expand the functionality and add the right options for the project to the smart contract.

LIMITATIONS:

1.Accidental transfer of tokens to an unaware contract.

2.Inability of handling incoming token transactions(no way to reject or handle any non supported tokens)

3.Token Transfer Uniformity(Different transfer function for transaction with regular/non-contract address and contract)

The popularity of ERC20 is large explained by adoption - that it makes it easier for 3rd parties to integrate it in the outside of Ethereum contexts.

3.3 Front End-React.js

The front end of the project is designed using react js library. So lets explore what is it.

React (also referred to as React.js or ReactJS) may be a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and corporations.

React are often used as a base within the development of single-page or mobile applications. Although React is merely concerned with rendering data to the virtual DOM, then creating React applications eventually requires the utilization of additional libraries for state management and routing. Redux and React Router are respective samples of such libraries. React was created by Jordan Walke, a programmer at Facebook, who released an early prototype of React called "FaxJS". He was influenced by XHP, an HTML component library for PHP. It was first deployed on Facebook's News feed 2011 and afterward Instagram in 2012. It was open-sourced at JSConf US in May 2013.
React Native, which enables native Android, iOS, and UWP development with React, was announced at Facebook's React Conf in February 2015 and open-sourced in March 2015.
On April 18, 2017, Facebook announced React Fiber, a replacement core algorithm of React library for building user interfaces. React Fiber was to become the inspiration of any future improvements and have development of the React library.
On September 26, 2017, React 16.0 was released to the general public .
On February 16, 2019, React 16.8 was released to the general public . The release introduced React Hooks.

Original author(s)	Jordan Walke
Developer(s)	Facebook and community
Initial release	May 29, 2013; 6 years ago
Stable release	16.13.1 / March 19, 2020; 37 days ago
Repository	React Repository
Written in	JavaScript
Platform	Web platform
Type	JavaScript library
License	MIT License
Website	reactjs.org

3.4 Development Methodology-Agile

Throughout the development of the project the agile methodology was followed so lets discuss what is it.

AGILE methodology is a practice that promotes continuous iteration of development and testing throughout the software development lifecycle of the project. Both development and testing activities are concurrent unlike the Waterfall model.

The agile software development emphasizes on four core values.

1. Individual and team interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

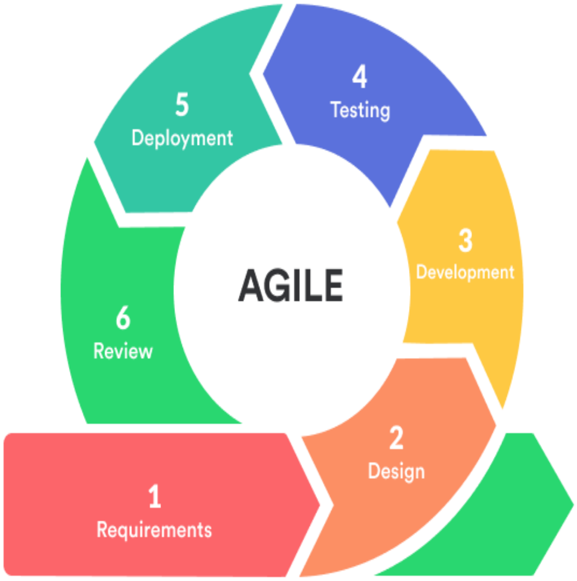


Fig 3.4.1 Agile Methodology

3.5 Apps v/s Dapps

DApps are a bit like apps in their functioning, as they supply services starting from games to financial transactions. However, unlike most regular apps that run on a server and may work without an online connection, Dapps can't be disconnected from the web , as they're going to simply stop functioning. Most Dapps share the subsequent characteristics – they're open source, they store all of their operation records on the blockchain, and that they work on tokens.

The last two years have seen a real explosion of Dapps appearing on the market with the rise in the number of blockchain-based projects developing them for a variety of industries and market sectors. The total number of Dapps on the market has surpassed 2,500 with the amount of daily Dapps users exceeding 96,000. The number of transactions generated by Dapps per day is surpassing 4.4 million 11,500 smart contracts being involved. More importantly, Dapps are a profitable business that's generating cash flows of over $21.5 million per day, proving that the recognition and application of Dapps is increasing.

The number of blockchain-enabled dapps is increasing day by day; most of them are the apps, powered by Ethehreum, EOS, POA, GoChain, and Steam blockchains.

As estimated in 2019, their number has reached 2,432, with about 180 new dapps being released each month. Interestingly enough, about 105 of these new apps are built on Ethereum blockchain, which boasts the largest developer community.

Actual users, though, prefer the EOS network, despite the fact that it doesn't host that many developers. More specifically, EOS outnumbers Ethereum in terms of dapp users by threefold.

How Dapps Are Different From Mobile Apps?

1. Dapps can store value
2. A higher level of synergy
3. Security
4. Speed
5. Resistance to censorship
6. Transparency and control
7. Cost of usage

CHAPTER 4

IMPLEMENTATION

4.1 Work Flow Diagrams:-

1).To brand registration with Loyalty Network-

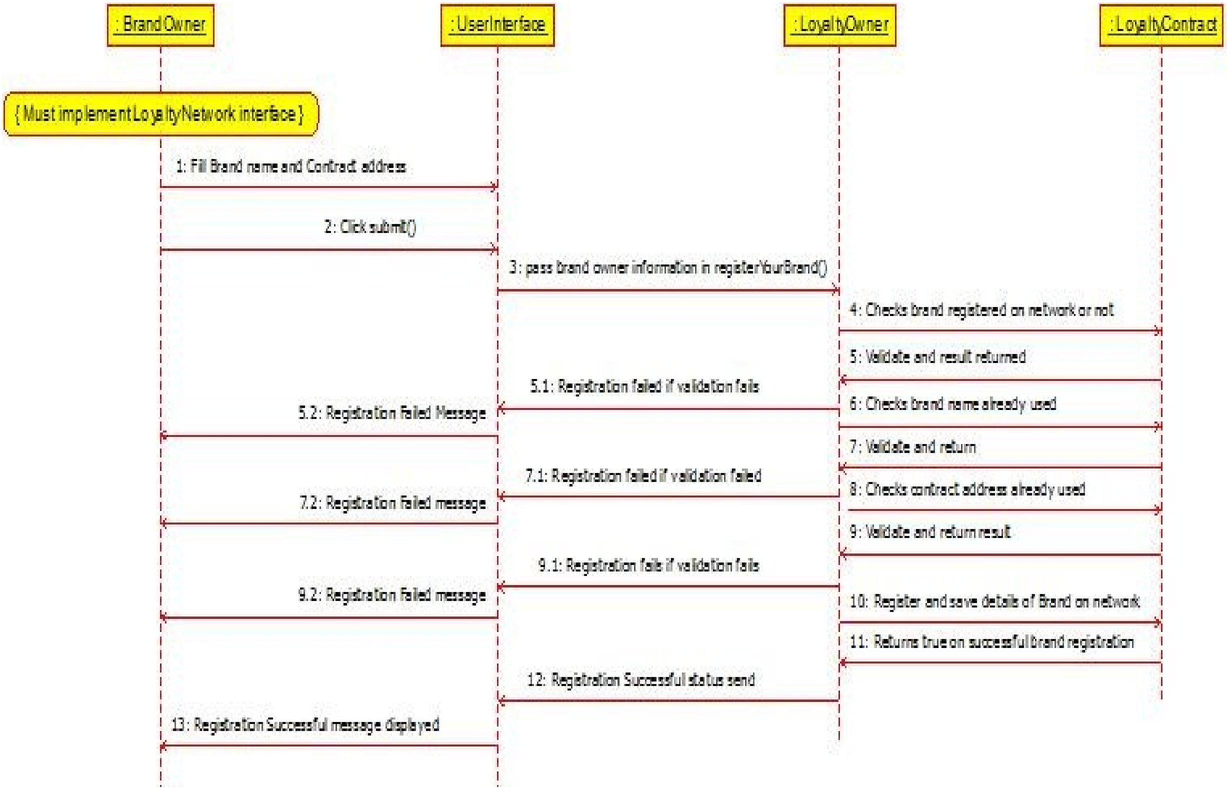

: BrandOwner
: UserInterface
: LoyaltyOwner
: LoyaltyContract
{ Must implement LoyaltyNetwork interface }
1: Fill Brand name and Contract address
2: Click submit()
3: pass brand owner information in registerYourBrand()
4: Checks brand registered on network or not
5: Validate and result returned
5.1: Registration failed if validation fails
5.2: Registration Failed Message
6: Checks brand name already used
7: Validate and return
7.1: Registration failed if validation failed
7.2: Registration Failed message
8: Checks contract address already used
9: Validate and return result
9.1: Registration fails if validation fails
9.2: Registration Failed message
10: Register and save details of Brand on network
11: Returns true on successful brand registration
12: Registration Successful status send
13: Registration Successful message displayed

Fig- 4.1.1 Sequence diagram to register brand

2). To add rewards in user account-

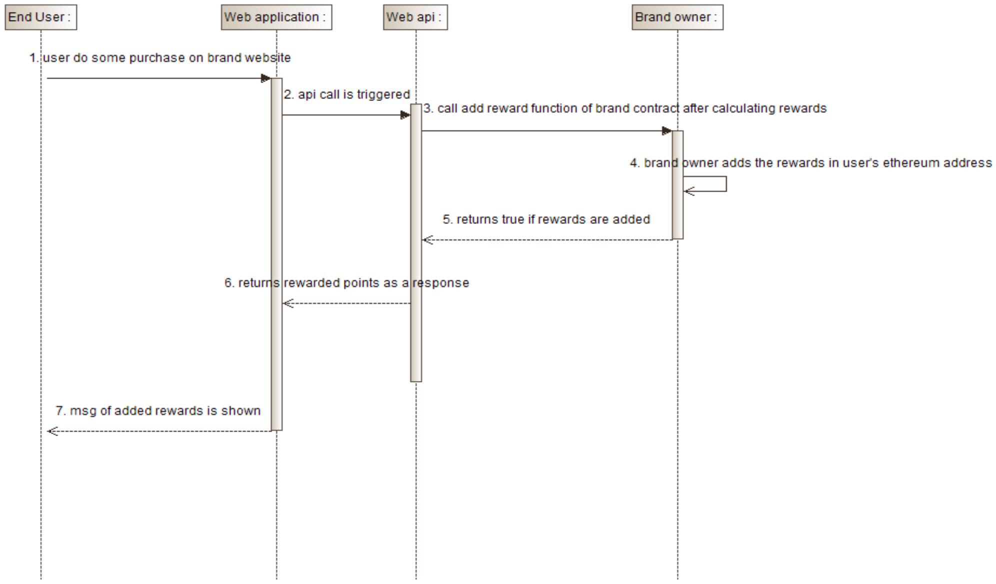

End User :
Web application :
Web api :
Brand owner :
1. user do some purchase on brand website
2. api call is triggered
3. call add reward function of brand contract after calculating rewards
4. brand owner adds the rewards in user's ethereum address
5. returns true if rewards are added
6. returns rewarded points as a response
7. msg of added rewards is shown

Fig- 4.1.2 Sequence diagram to add points

3). To redeem points -

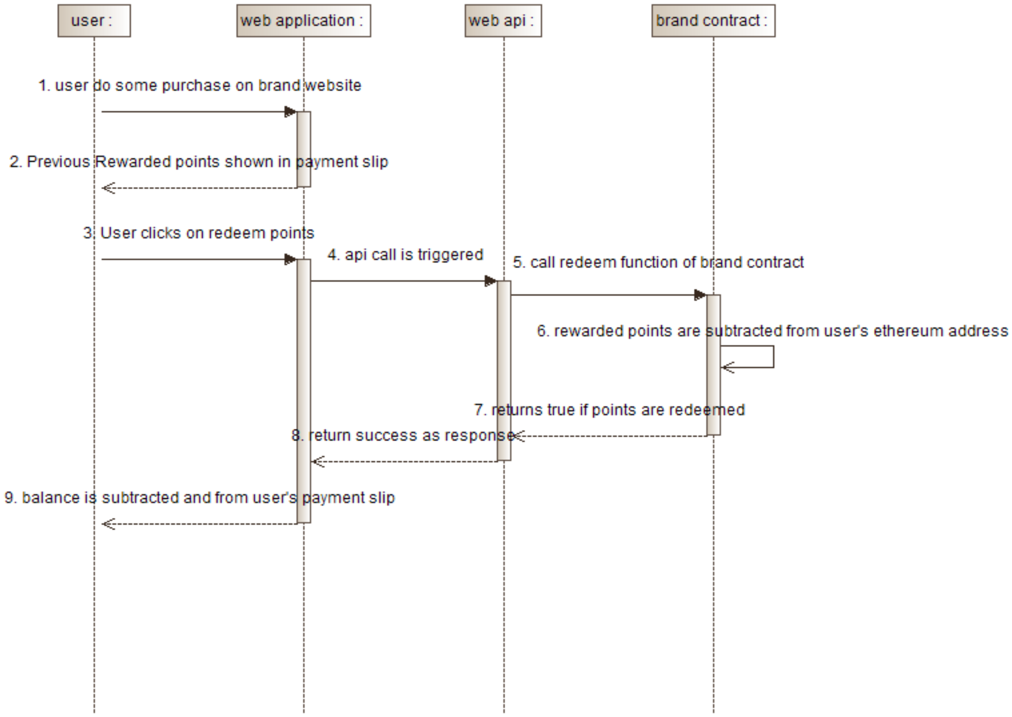

user :
web application :
web api :
brand contract :
1. user do some purchase on brand website
2. Previous Rewarded points shown in payment slip
3. User clicks on redeem points
4. api call is triggered
5. call redeem function of brand contract
6. rewarded points are subtracted from user's ethereum address
7. returns true if points are redeemed
8. return success as response
9. balance is subtracted and from user's payment slip

Fig- 4.1.3 Sequence diagram to redeem points

4)To transfer points-

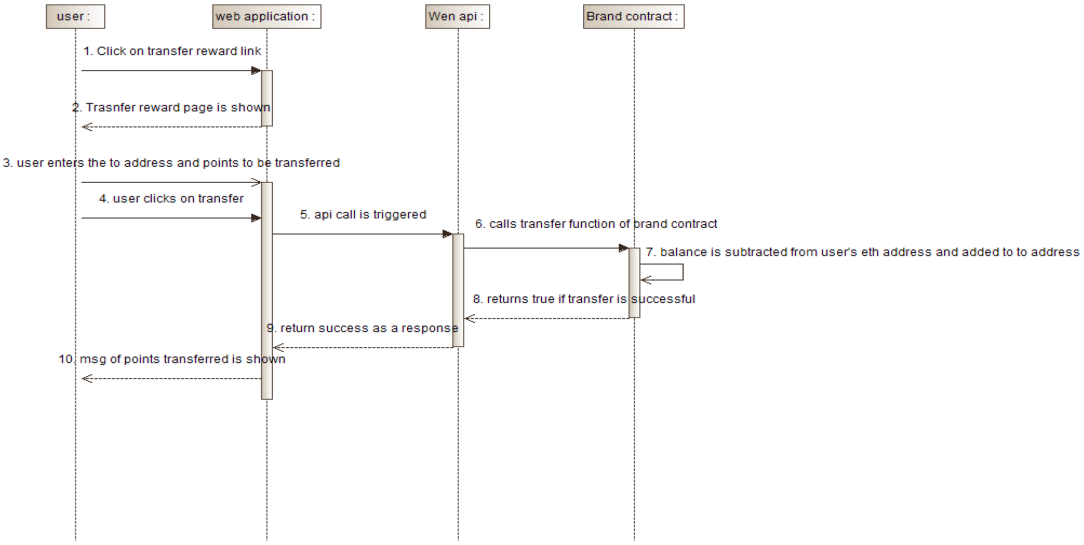

user :
web application :
Wen api :
Brand contract :
1. Click on transfer reward link
2. Trasnfer reward page is shown
3. user enters the to address and points to be transferred
4. user clicks on transfer
5. api call is triggered
6. calls transfer function of brand contract
7. balance is subtracted from user's eth address and added to to address
8. returns true if transfer is successful
9. return success as a response
10. msg of points transferred is shown

Fig- 4.1.4 Sequence diagram to transfer points

5). To exchange points-

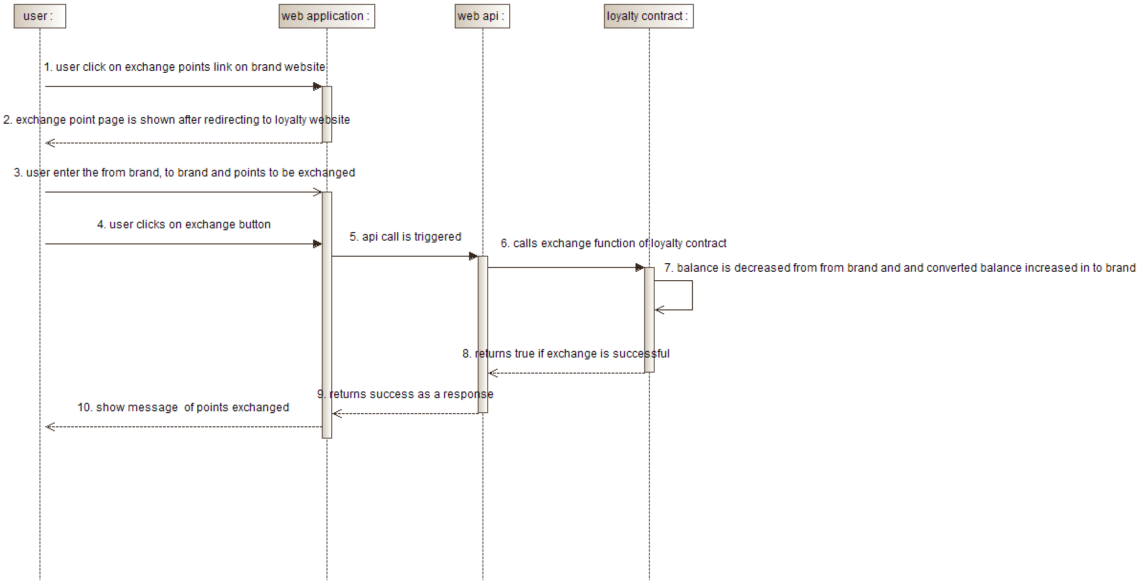
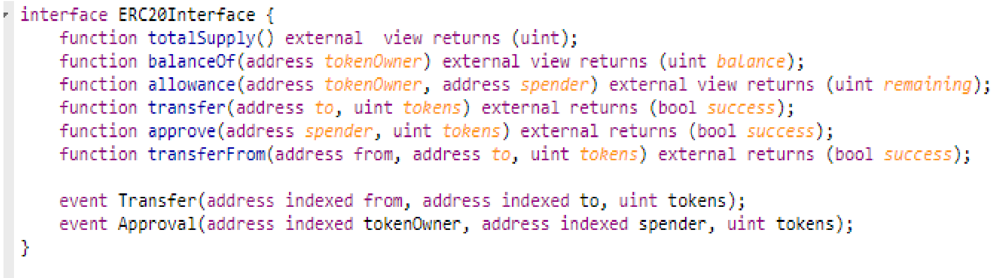

user :
web application :
web api :
loyalty contract :
1. user click on exchange points link on brand website
2. exchange point page is shown after redirecting to loyalty website
3. user enter the from brand, to brand and points to be exchanged
4. user clicks on exchange button
5. api call is triggered
6. calls exchange function of loyalty contract
7. balance is decreased from from brand and and converted balance increased in to brand
8. returns true if exchange is successful
9. returns success as a response
10. show message of points exchanged

Fig 4.1.5- Sequence diagram to exchange points

4.2 Token Contracts

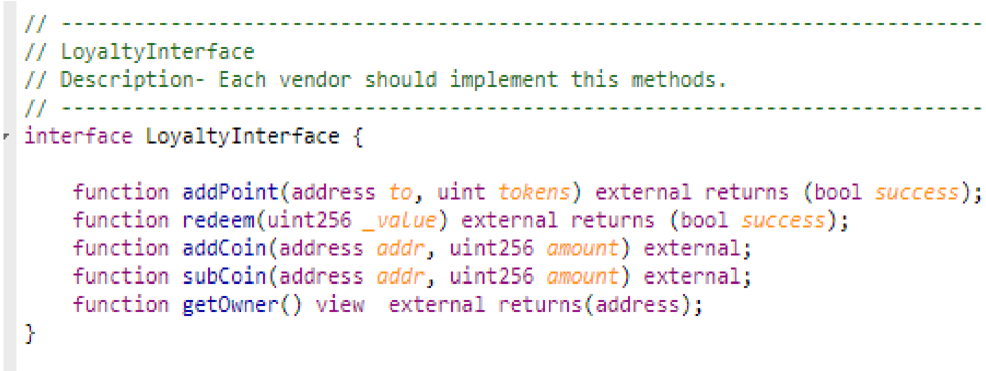
4.2.1 ERC20 Interface: This is the standard of Ethereum. Each contract should implement this interface in this project**.**



```
interface ERC20Interface {
    function totalSupply() external  view returns (uint);
    function balanceOf(address tokenOwner) external view returns (uint balance);
    function allowance(address tokenOwner, address spender) external view returns (uint remaining);
    function transfer(address to, uint tokens) external returns (bool success);
    function approve(address spender, uint tokens) external returns (bool success);
    function transferFrom(address from, address to, uint tokens) external returns (bool success);

    event Transfer(address indexed from, address indexed to, uint tokens);
    event Approval(address indexed tokenOwner, address indexed spender, uint tokens);
}
```

4.2.2 Loyalty Interface: This interface should be implemented by each vendor, before applying for registration with the Loyalty Network.



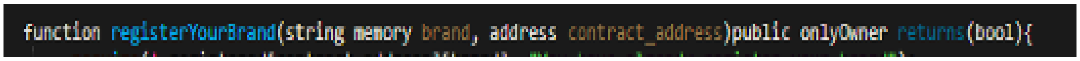
```
// ----------------------------------------------------------------------------
// LoyaltyInterface
// Description- Each vendor should implement this methods.
// ----------------------------------------------------------------------------
interface LoyaltyInterface {

    function addPoint(address to, uint tokens) external returns (bool success);
    function redeem(uint256 _value) external returns (bool success);
    function addCoin(address addr, uint256 amount) external;
    function subCoin(address addr, uint256 amount) external;
    function getOwner() view  external returns(address);
}
```

4.2.3 Loyalty Network Contract- This is main Loyalty network in which all the vendors will be registered. It will handle all the points. This contract has following main features.

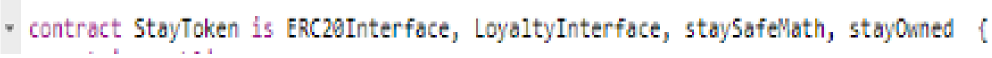
1. Register Brands on Network.
2. Add points to the user accounts
3. Redeem points
4. Exchange points same brand.
5. Exchange points different brands.

4.2.3.1 Registration form: Each vendor should register themselves to the loyalty network , to avail all the features of loyalty network.
The function signature that does this work:



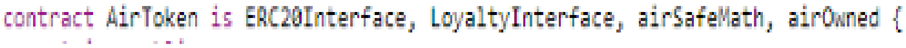
```
function registerYourBrand(string memory brand, address contract_address)public onlyOwner returns(bool){
```

4.2.4 StayToken Contract: This contract is based on Radisson hotel booking. When Radisson user will book the ticket , he/she will get some rewards-points. Reward point value should be prior decided by the vendor.
The Contract signature:



```
contract StayToken is ERC20Interface, LoyaltyInterface, staySafeMath, stayOwned  {
```

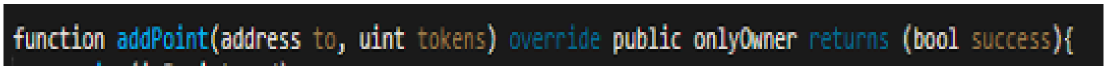
4.2.5 Airline Contract: This contract is based on the Airline booking. When passenger will book the ticket , he/she will get some rewards based on the point value decided by Airline vendor.
The Contract Signature:



```
contract AirToken is ERC20Interface, LoyaltyInterface, airSafeMath, airOwned {
```

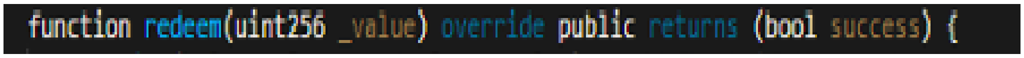
4.3 Features of loyalty network:-

1) Add points - It will add the points in the user accounts.



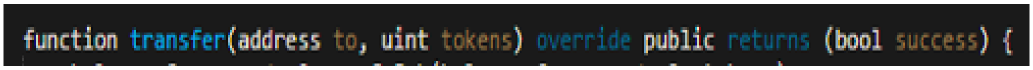
```
function addPoint(address to, uint tokens) override public onlyOwner returns (bool success){
```

2) Redeem Points- The user can use the points for other purchase.



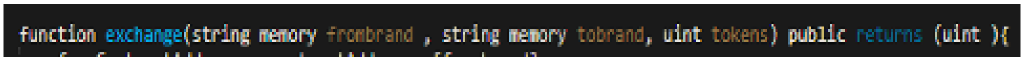
```
function redeem(uint256 _value) override public returns (bool success) {
```

3) Transfer Points- The user can transfer their points to other user.



```
function transfer(address to, uint tokens) override public returns (bool success) {
```

4) Exchange Point- Ther user can exchange point from one token to another.



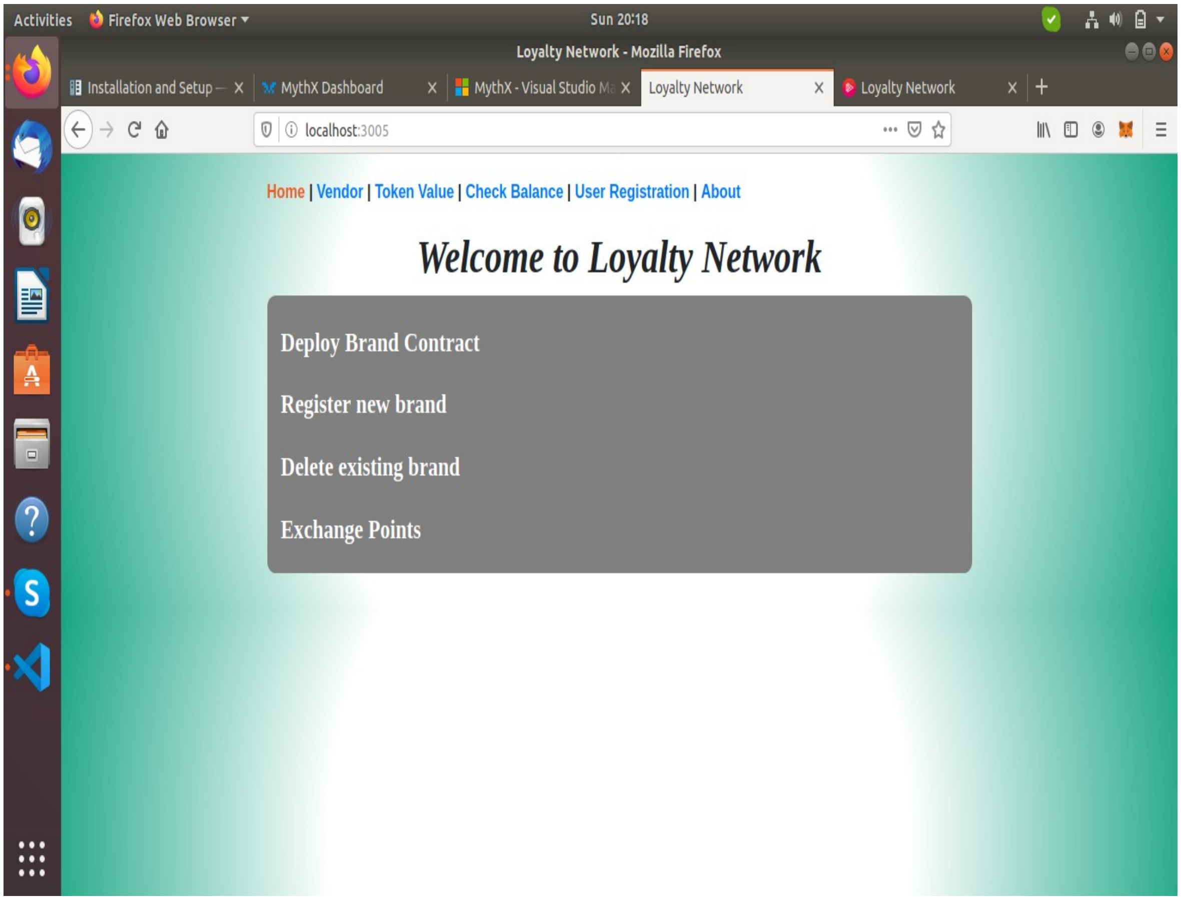
```
function exchange(string memory frombrand , string memory tobrand, uint tokens) public returns (uint ){
```

4.4 React.js - In react front end is designed , through which user can interact with

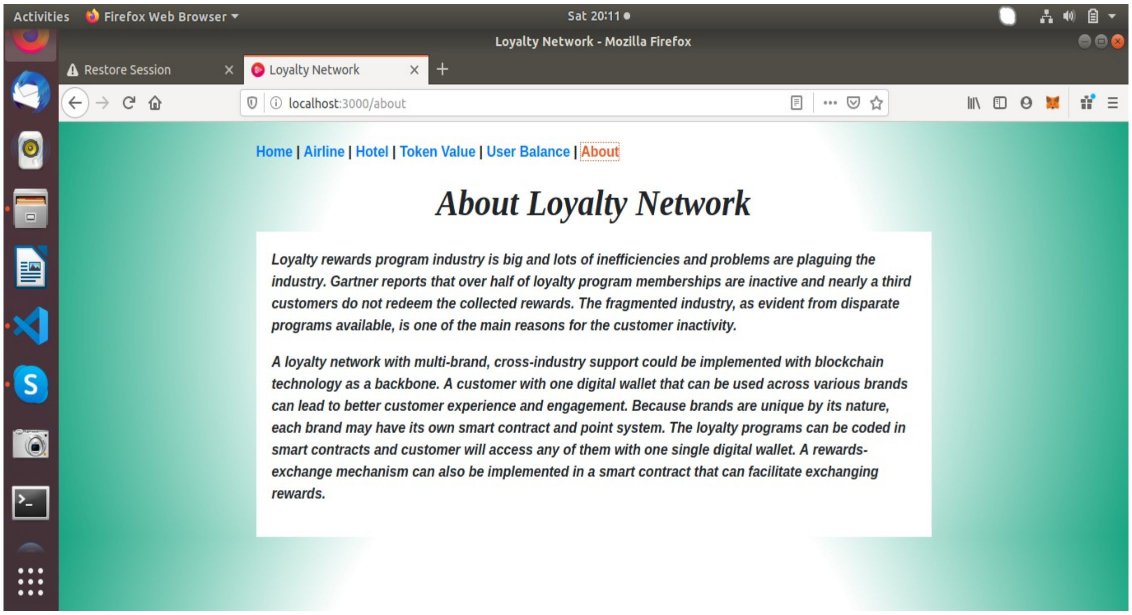
the contract.
The frontend structure contains:
1. AirAsia User Interface.
2. Radisson User Interface.
3. Loyalty Network User Interface.
Node modules used.

4.5 Working And Screenshots

4.5.1 Loyalty Website-
Home Page-



About us Page-



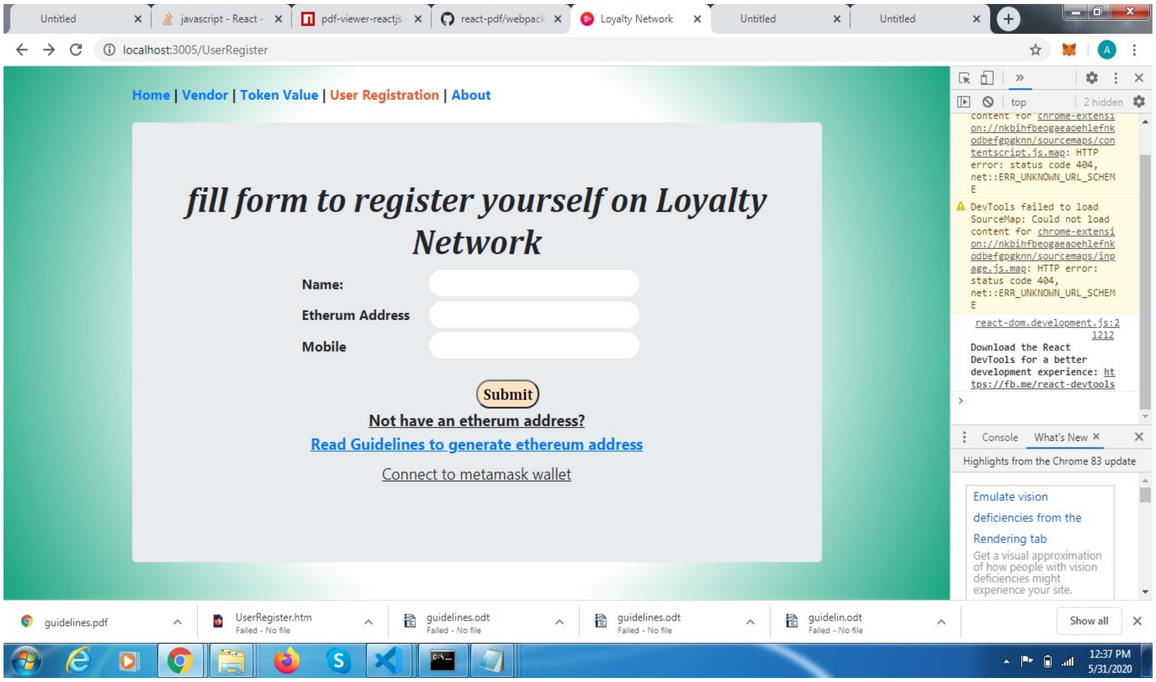
Home | Airline | Hotel | Token Value | User Balance | About

About Loyalty Network

Loyalty rewards program industry is big and lots of inefficiencies and problems are plaguing the industry. Gartner reports that over half of loyalty program memberships are inactive and nearly a third customers do not redeem the collected rewards. The fragmented industry, as evident from disparate programs available, is one of the main reasons for the customer inactivity.

A loyalty network with multi-brand, cross-industry support could be implemented with blockchain technology as a backbone. A customer with one digital wallet that can be used across various brands can lead to better customer experience and engagement. Because brands are unique by its nature, each brand may have its own smart contract and point system. The loyalty programs can be coded in smart contracts and customer will access any of them with one single digital wallet. A rewards-exchange mechanism can also be implemented in a smart contract that can facilitate exchanging rewards.

Guideline to get Ethereum Account:



Home | Vendor | Token Value | User Registration | About

fill form to register yourself on Loyalty Network

Name:	
Etherum Address	
Mobile	

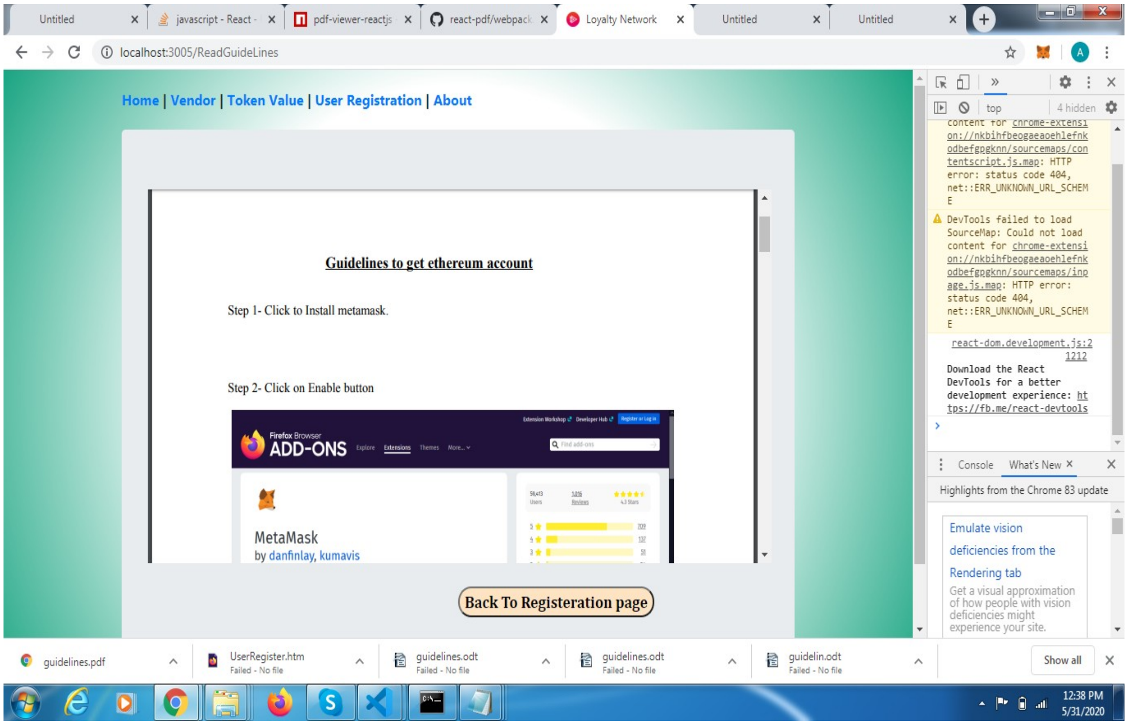
Submit

Not have an etherum address?

Read Guidelines to generate ethereum address

Connect to metamask wallet

```
content for chrome-extensi
on://nkbihfbeogaeaoehlefnk
odbefgpgknn/sourcemaps/con
tentscript.js.map: HTTP
error: status code 404,
net::ERR_UNKNOWN_URL_SCHEM
E
DevTools failed to load
SourceMap: Could not load
content for chrome-extensi
on://nkbihfbeogaeaoehlefnk
odbefgpgknn/sourcemaps/inp
age.js.map: HTTP error:
status code 404,
net::ERR_UNKNOWN_URL_SCHEM
E
                react-dom.development.js:2
                                     1212
Download the React
DevTools for a better
development experience: ht
tps://fb.me/react-devtools
```



localhost:3005/ReadGuideLines
Home | Vendor | Token Value | User Registration | About

Guidelines to get ethereum account

Step 1- Click to Install metamask.

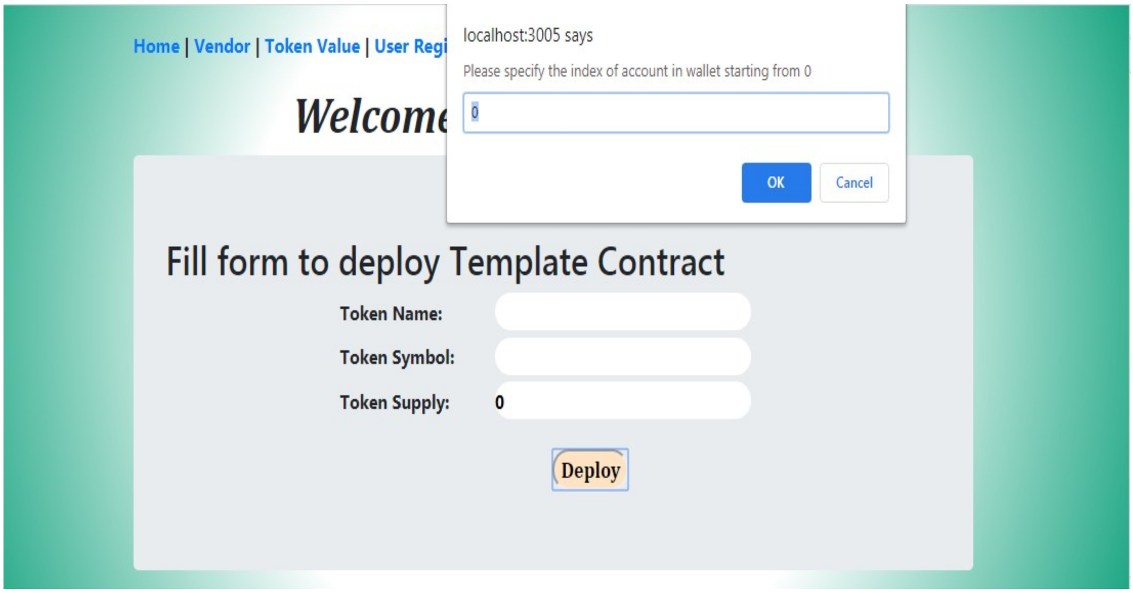
Step 2- Click on Enable button

Firefox Browser ADD-ONS
MetaMask
by danfinlay, kumavis
Back To Registeration page

```
content for chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/sourcemaps/contentscript.js.map: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME
DevTools failed to load SourceMap: Could not load content for chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/sourcemaps/inpage.js.map: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME
react-dom.development.js:21212
Download the React DevTools for a better development experience: https://fb.me/react-devtools
```

Console
What's New
Highlights from the Chrome 83 update
Emulate vision deficiencies from the Rendering tab
Get a visual approximation of how people with vision deficiencies might experience your site.
guidelines.pdf
UserRegister.htm
Failed - No file
guidelines.odt
Failed - No file
guidelines.odt
Failed - No file
guidelin.odt
Failed - No file
Show all
12:38 PM
5/31/2020

Contract Deployment for vendor:-



localhost:3005 says

Please specify the index of account in wallet starting from 0

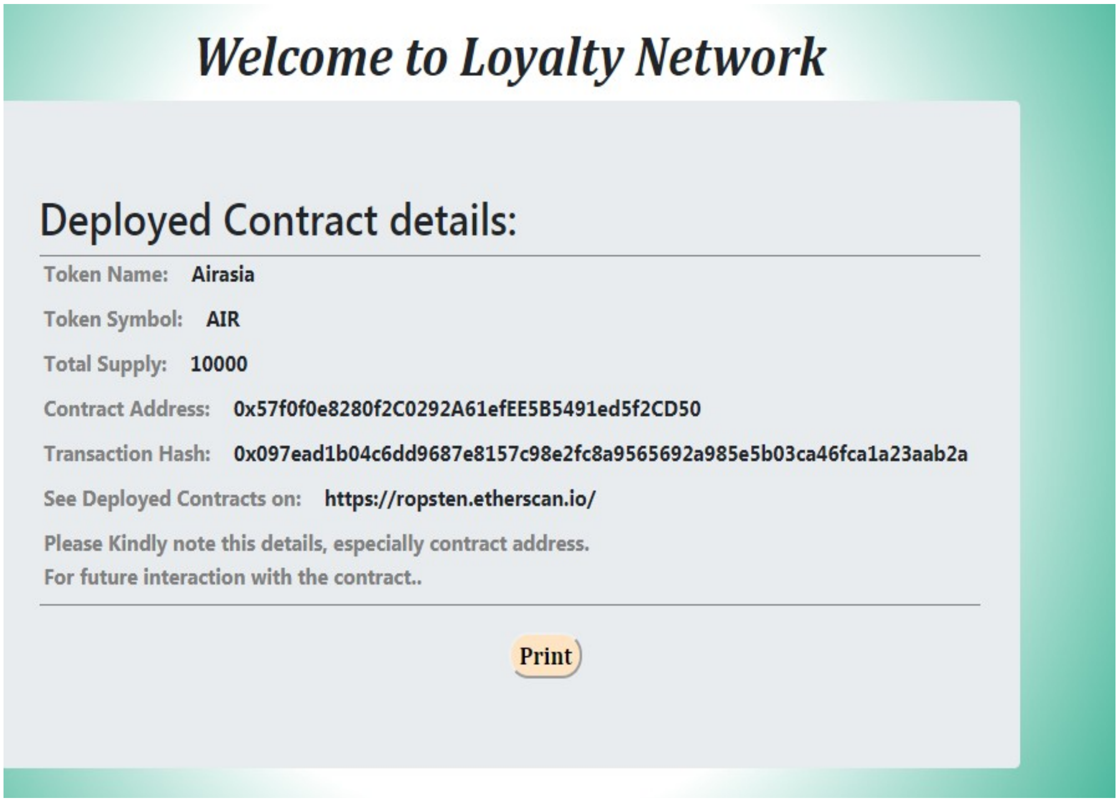
0

Fill form to deploy Template Contract

Token Name:

Token Symbol:

Token Supply: 0



Welcome to Loyalty Network

Deployed Contract details:

Token Name: Airasia

Token Symbol: AIR

Total Supply: 10000

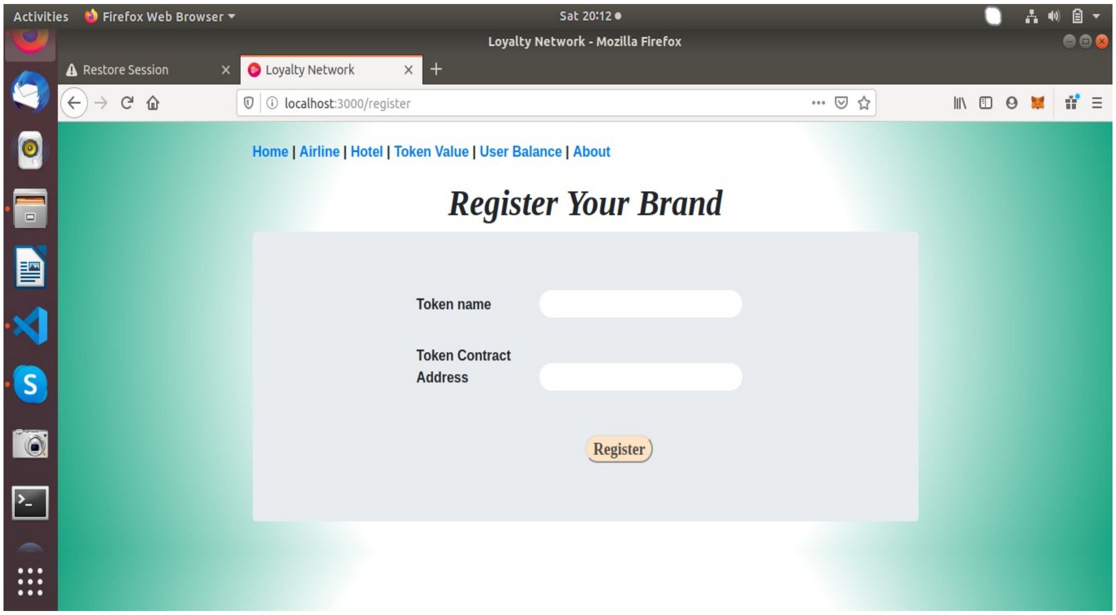
Contract Address: 0x57f0f0e8280f2C0292A61efEE5B5491ed5f2CD50

Transaction Hash: 0x097ead1b04c6dd9687e8157c98e2fc8a9565692a985e5b03ca46fca1a23aab2a

See Deployed Contracts on: https://ropsten.etherscan.io/

Please Kindly note this details, especially contract address.
For future interaction with the contract..

Registration Page -

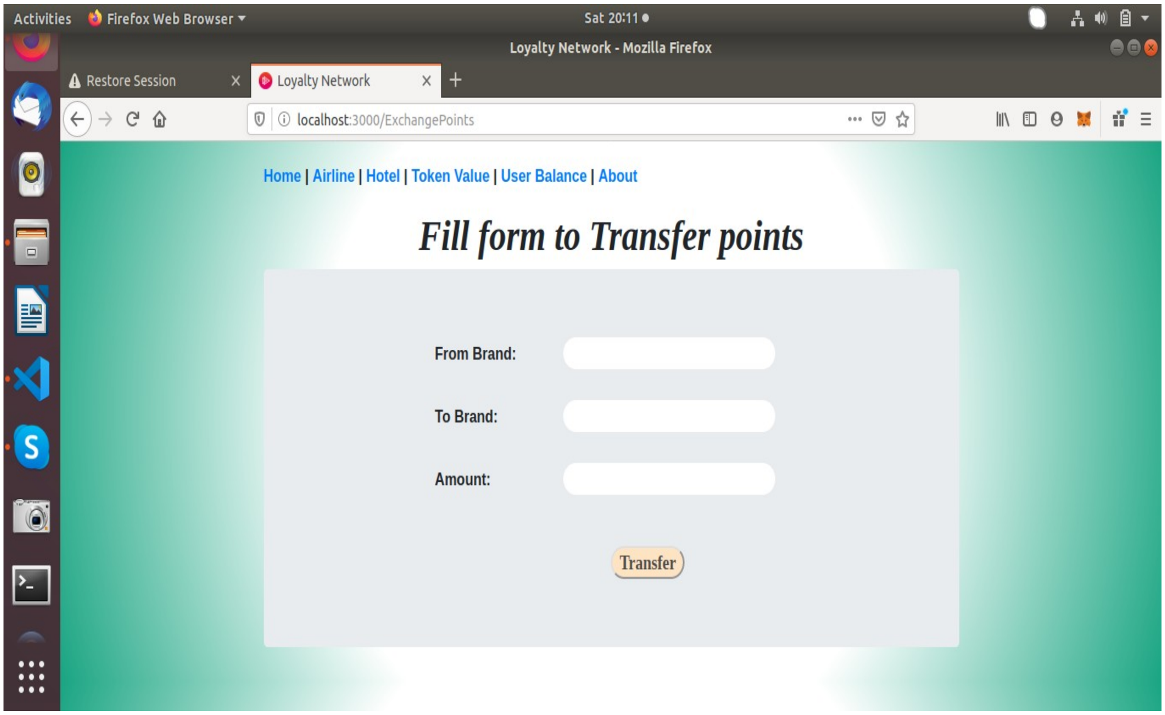


Register Your Brand

Token name

Token Contract Address

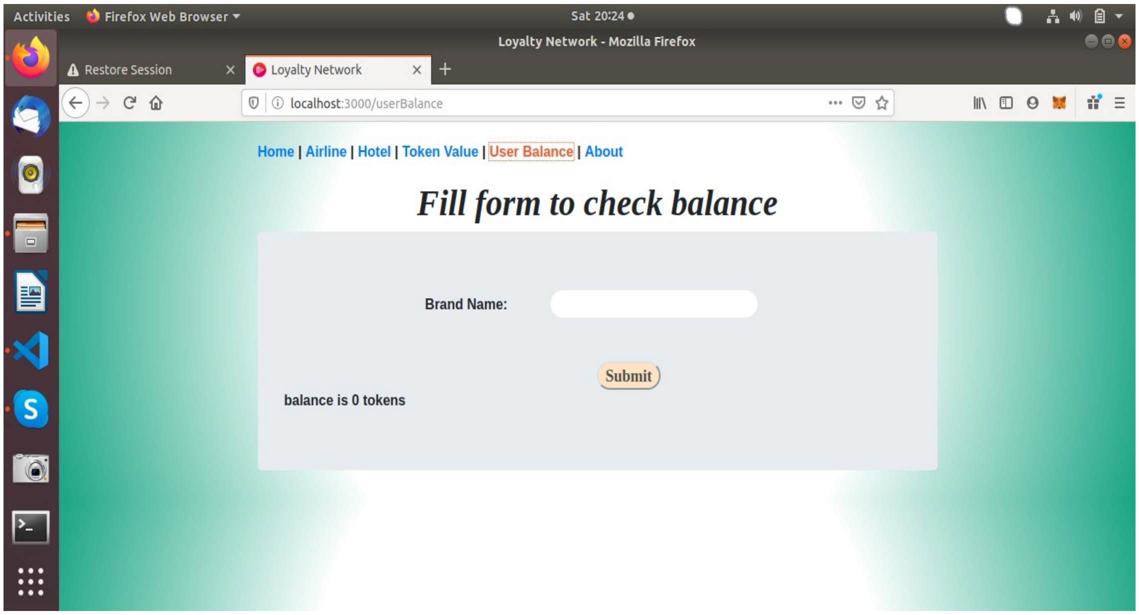
Exchange Token page-


Activities
Firefox Web Browser
Sat 20:11
Loyalty Network - Mozilla Firefox
Restore Session
Loyalty Network
localhost:3000/ExchangePoints
Home | Airline | Hotel | Token Value | User Balance | About

Fill form to Transfer points

From Brand:	
To Brand:	
Amount:	

Balance info page-


Activities
Firefox Web Browser
Sat 20:24
Loyalty Network - Mozilla Firefox
Restore Session
Loyalty Network
localhost:3000/userBalance
Home | Airline | Hotel | Token Value | User Balance | About

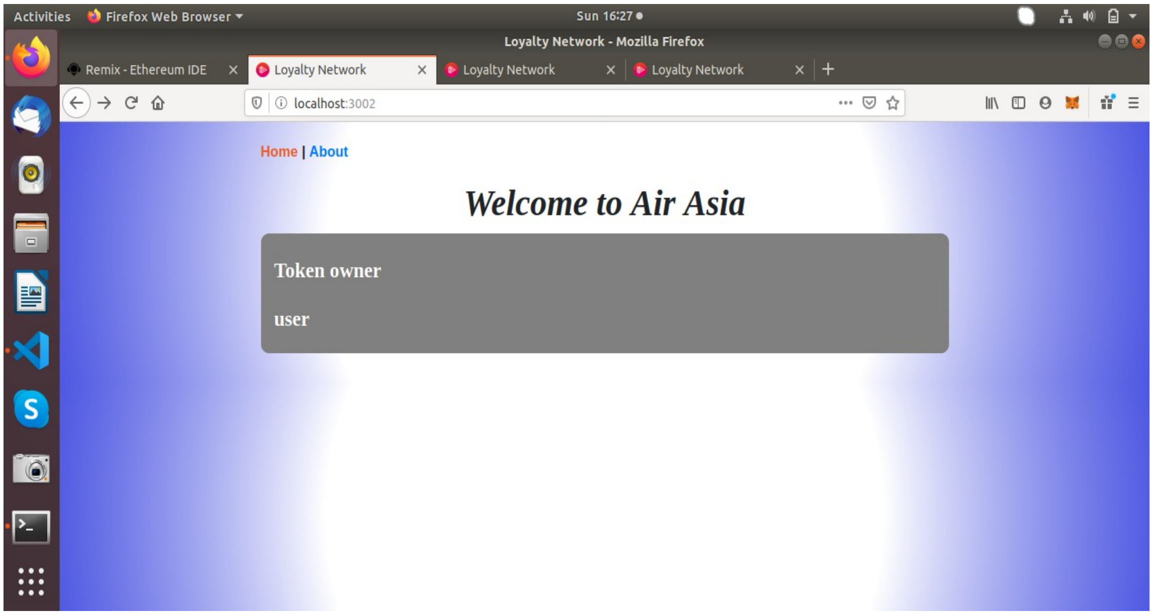
Fill form to check balance

Brand Name:	

balance is 0 tokens

4.5.2 AirAsia Website-

Home Page-



Home | About

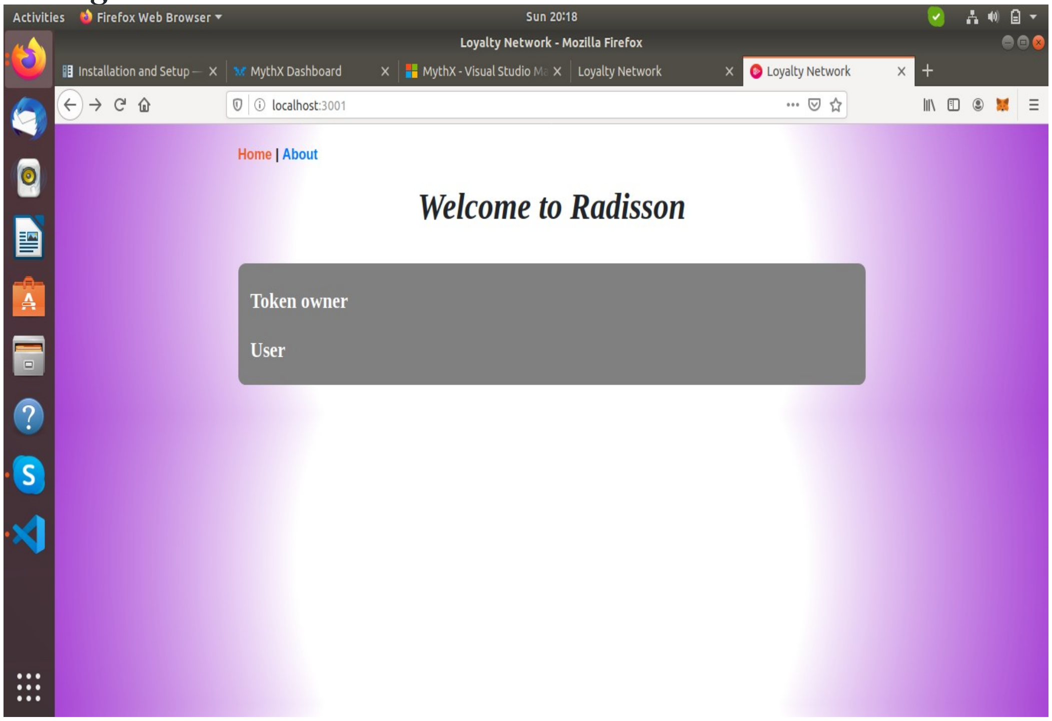
Welcome to Air Asia

Token owner

user

4.5.3 Hotel Website-

Home Page -



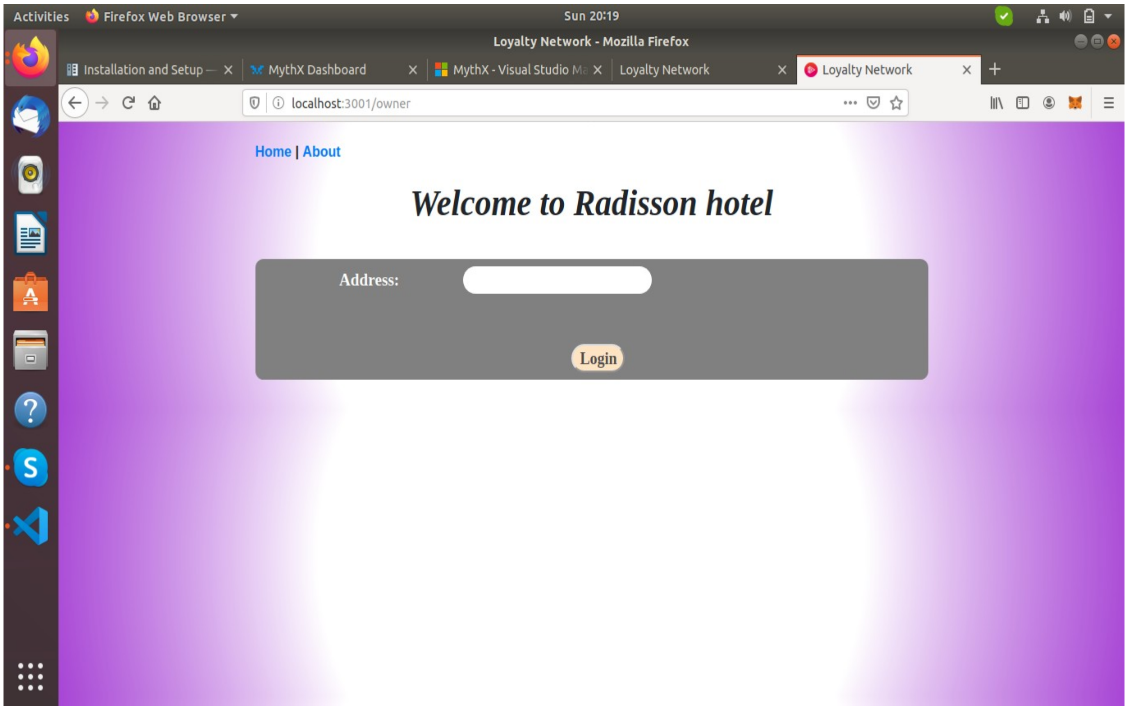
Home | About

Welcome to Radisson

Token owner

User

Hotel Login-



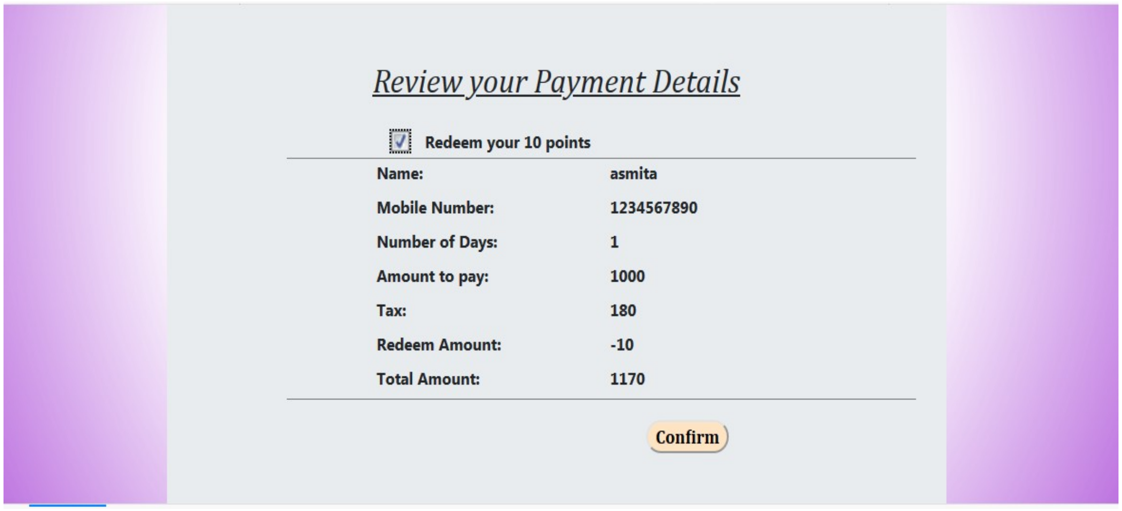
Home | About

Welcome to Radisson hotel

Address:

Login

Payment confirmation page-



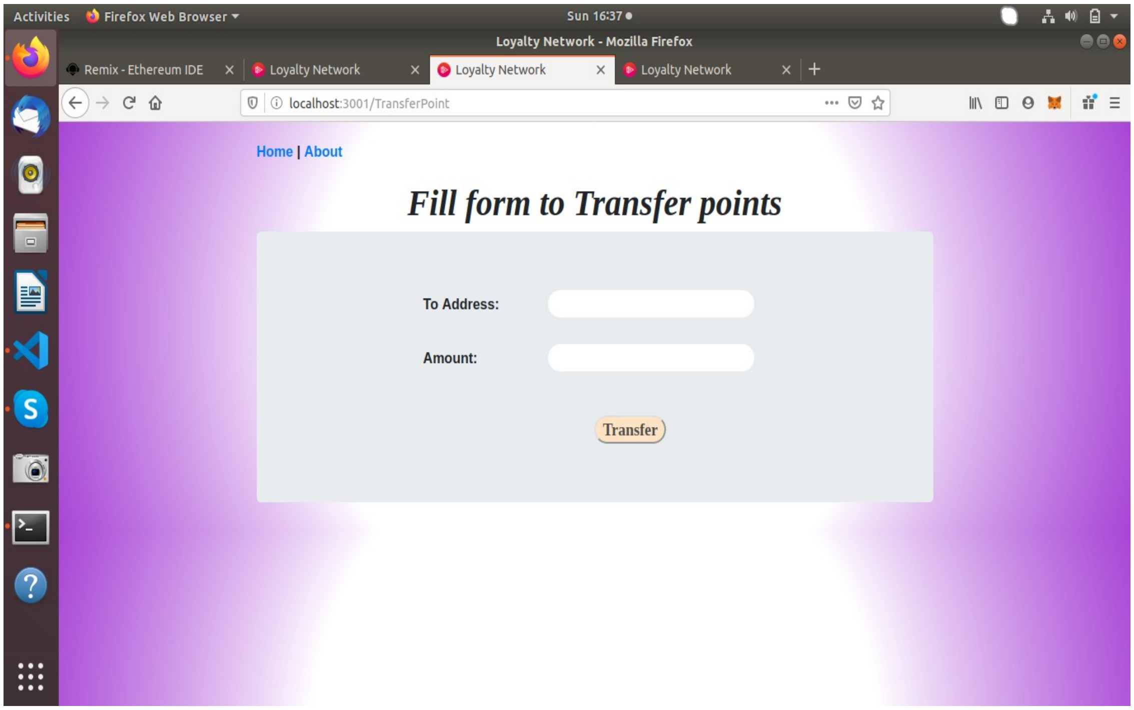
Review your Payment Details

☑ Redeem your 10 points

Name:	asmita
Mobile Number:	1234567890
Number of Days:	1
Amount to pay:	1000
Tax:	180
Redeem Amount:	-10
Total Amount:	1170

Confirm

Transfer Token To Other User-



Fill form to Transfer points

To Address:

Amount:

CHAPTER 5

RESULTS AND CONCLUSION

The Results obtained in test network using Ganache was really impressive. The whole project was done in development mode and then we will move it to production mode. The testing strategy was done by using two approach:

1) **Unit testing** – Final report is satisfactory
2) **Integration testing** - Final report is satisfactory

Unit is a level of software testing where individual units/ components of software are tested. The purpose is to validate that each unit of the software performed as designed. A unit is the smallest testable part of any software. It is usually has one or a few inputs and usually a single output. Unit testing performed on each module or block of code during development. Unit testing is normally done by the programmer who writes the code.

🎬 *Unit Test Plan*

1. Prepare
2. Review
3. Rework
4. Baseline

Integration testing done before, during and after integration of a new module into the main software package. This involves testing of each individual code module. One piece of software can contain several modules which are often created by several different programmers. It is crucial to test each modules effect on the entire program model. After integration testing the project works successfully.

There are various tools used throughout the project for implementing and testing. All the tools are discussed in detail in chapter 3. The backend is an Ethereum blockchain platform on top of which the project is build. The frontend is implemented in React.js. The ether.js library was used for connecting front end with blockchain. For the transaction to happen among various entities we used third party application Metamask. And for the local blockchain environment to perform dynamic analysis and testing of project, Ganache is used. The static analysis of smart contract code written for Ethereum blockchian MythX tool was used.

The whole project was done with the intention to boost the economic advantage for companies and user both. As long as the user get the advantages from any service they do not care about the background functionality of any application. And thus we tried to make it more user friendly. As concluding remark we saw the strength and why we implemented this problem on blockchain.

Chapter 1 dealt with the introduction of the project problem and various important technology. Background, objective of project and various technology along with the organization of project is discussed. In chapter 2, literature review was done and the various ongoing works in this field was tackled and discussed in gist. In chapter 3, It can be seen the proposed approach used in project. Along with this important tools used are also disscused. In chapter 4, the implementation part is worked on. First of all the workflow and sequence diagram is constructed for coding the problem. Then coding part was done along with coding the testing was also done mainly unit testing. Once the project got completed the Integration testing was done. The code written was first statically analysed for any memory leak and then dynamically analysed for checking its various functionality working properly. And last chapter disscuss the result.

While making the project we solved the existing problem statement but there is still some limitation of blockchian because of which we can further improve the project. Presently we constructed this project using two brand vendors and in future it can be increased to n number of vendors. And for that the scalability factor is taken care. Some of loop holes might be present in the project which can be tackled and overcome in next version of this decentralized application.

References:-

1) https://docs.ethers.io/ethers.js/html/
2) https://web3js.readthedocs.io/en/v1.2.8/
3) https://ethereum.org/developers/
4) https://static.qiibee.com/qiibee-White-Paper.pdf
5) https://www.trufflesuite.com/ganache
6) https://docs.metamask.io/guide/
7) https://www.researchgate.net/publication/317663250_Loyalty_programs_in_the_pharmaceutical_retail_The_impact_of_a_network_loyalty_program_on_the_pharmacy_loyalty
8) https://qiibee.com
9) https://static.qiibee.com/qiibee-Lattesso-Roll-Out.pdf
10) https://eips.ethereum.org/EIPS/eip-20
11) https://toshitimes.com/coinbase-releases-a-crypto-gift-card-service-for-customers-in-europe-and-australia/