CryptoBill - Cryptocurrency Crowdfunding for Billions

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Technology

By

Mr. Raj Shekhar Kumar

(Roll No. 141100053)

Under the guidance of

Mr. Ankur

Assistant Professor,

Department of Computer Science and Engineering,

National Institute of Technology, Delhi

Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY, DELHI

SPRING 2018

Acknowledgement

I would like to express my sincere gratitude to my mentor, philosopher and guide **Mr. Ankur**, Assistant Professor, Department of Computer Science and Engineering, for providing his invaluable guidance, comments and suggestions throughout the course of the project and constantly motivating me to work harder. His dedication and keen interest above all his overwhelming attitude to help his students had been solely and mainly responsible for completing my work. His timely advice, meticulous scrutiny , scholarly advice and scientific approach have helped me accomplish this task.

I would specially thank to my friends who helped me during this process and make this Project possible.

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/ data /fact/ source in our submission. We understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Mr. Ankur
Assistant Professor
Department of Computer Science Engineering
Date: 28/02/2018

Team Member(s)

Mr. Raj Shekhar Kumar (141100053)

Certificate

This is to certify that the project titled "**CryptoBill - Cryptocurrency Crowdfunding for Billions**" submitted by **Raj Shekhar Kumar** to the National Institute of Technology, Delhi, is a record of bonafide work done under my supervision and I consider it worthy of acceptance for the award of the degree of Bachelor of Technology of the Institute.

Signature of Supervisor
Mr. Ankur
Date : ___/___/___

Approval Sheet

This project work entitled by **CryptoBill - Cryptocurrency Crowdfunding for Billions** by **Raj Shekhar Kumar** for the degree of Bachelor of Technology in Computer Science and Engineering.

Examiners

____

____

____

Supervisor(s)

____

____

____

Chairman

____

Date: ___/___/___

Place: __________

Table of Contents

List of Figures

List of Abbreviations

1. API - Application Programming Interface
2. CSS - Cascading Style Sheets
3. DApp - Decentralized Application
4. HTML - Hypertext Markup Language
5. JS - Javascript
6. NPM - Node Package Manager
7. UI - User Interface
8. ETH - Ethereum
9. ASP - Active Server Pages
10. MIT - Massachusetts Institute of Technology
11. NPM - Node Package Manager

Abstract

Crowdfunding is the practice of funding a project or venture by raising money from a large number of people who each contribute a relatively small amount, typically via the Internet. There are multiple crowdfunding platforms like Kickstarter, Patreon, GoFundMe etc. that are using by startups and people for fundraising their cause or company via asking the general public for money.

There is no single major decentralized fundraising platform that is in operation. The reason is due to many people making such platforms, offering Initial Coin Offerings and then scamming people with the money they received. The purpose is to build a peer-to-peer version of the crowdfunding system as a Decentralized Application using the Ethereum network so that the middlemen can be cut from the process of fundraising and the support for the cause can be maximized.

1. Introduction

1.1. Problem Statement

As the whole world moves towards a state of Nationalistic ideologies, more and more people have stopped believing in Financial institutions to take of the money that they worked so hard to earn. The public has gone through the 2008 recession, which was (as some suggest), the birth of the idea of having a peer-to-peer transaction technology such as Blockchain so that the public money is incorruptible and transparent.

Although this technology has gained widespread momentum around the world, some fraudsters have found a way to use it for fraud and then abscond. With the Application that I am building, the public would be able to fund projects or people in need and the transaction that happens on every single project or cause will go through a stringent process of approval so that no fraudsters can maliciously use this software.

1.2. Decentralized Applications

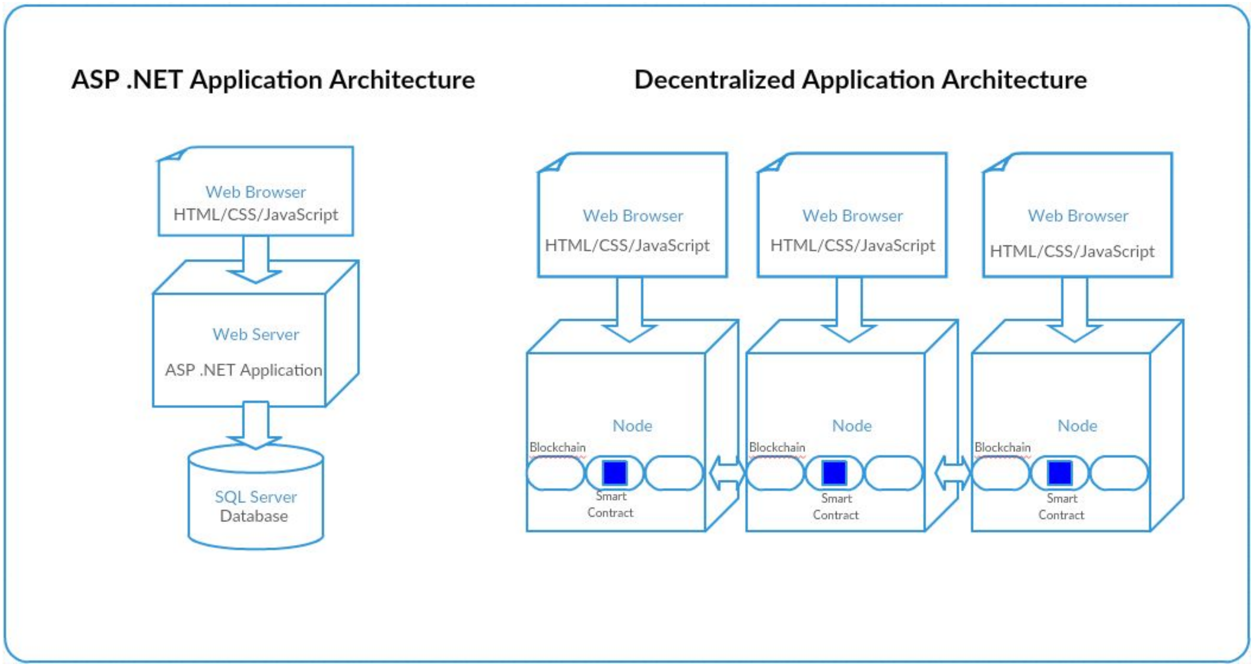


Figure 1.1. Decentralized Application Architecture[11]

DApp is an abbreviated form for *decentralized application*. The majority of Decentralized Applications that are created and are on the Initial Coin Offering stage are built on Ethereum.

The Ethereum, is the Blockchain platform and then there's Ethereum Virtual Machine which is used to run the application. If someone wants to make a Decentralized Application, it will be built on top of Ethereum that means that they will need to use Ethereum's protocols, Smart Contracts, network etc. to build a brand new application.

The biggest problem with today's Decentralized Applications are that the general public do not have enough information about the Blockchain technology and they are investing in the Initial Coin Offerings and losing money. Also, some of the people who are building these Decentralized Applications are making them for the sole purpose of fraud and they are running away with money that is being generated by the Initial Coin Offerings.

1.3. Smart Contracts

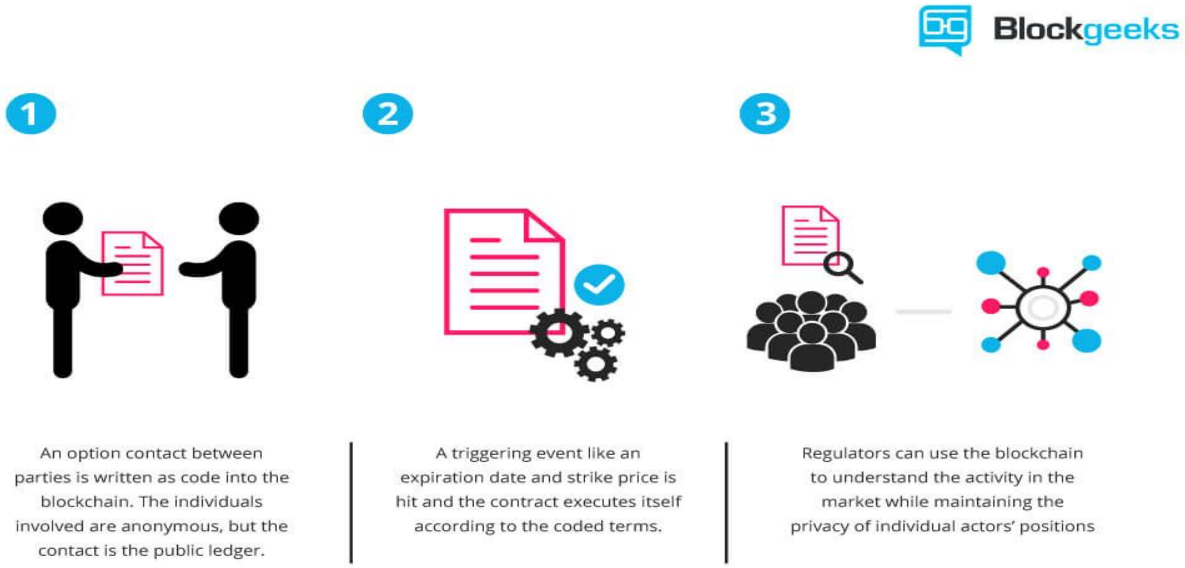

Blockgeeks
1
2
3
An option contact between parties is written as code into the blockchain. The individuals involved are anonymous, but the contact is the public ledger.
A triggering event like an expiration date and strike price is hit and the contract executes itself according to the coded terms.
Regulators can use the blockchain to understand the activity in the market while maintaining the privacy of individual actors' positions

Figure 1.2. Smart Contracts[7]

In Ethereum, the Smart Contract is one of the main features that is responsible for its success. To put it simply, a Smart Contract is an account that is controlled by code. This is not possible on the Blockchain 1.0 but the Ethereum Blockchain, also referred to as Blockchain 2.0 can allow blocks to have code snippets that can take care of the transactions which make the Ethereum Blockchain to be the one of the main programmable Blockchains.

There are three main features of a Smart Contract:

1. Balance: amount of Ether an account holds.
2. Storage: Data storage for this account.
3. Code: Raw machine code for this contract.

These features control a single Block in the Ethereum Blockchain and control transactions. For every single transaction that will take place, there are a lot of features that ensure transparency and incorruptibility:

1. Nonce - How many times a sender sent a transaction.
2. To - Address of the account money is being remitted towards.
3. Value - Amount of Ether sent.
4. Gas Price: For every transaction to take place, it needs some gas for the computational costs on the Blockchain. These are often negligible when compared to what the user would actually have to pay in today’s financial setup.
5. V, S, R: These are the cryptographic pieces of data that are generated form the sender’s private key and they can be used to generate the receiver's account address.
6. Blocktime: Time taken to find the solution.

1.4. Solidity

Solidity is the language that is used to make Smart Contracts to be implemented on the Ethereum Blockchain. Solidity is an object-oriented, high-level programming language that is known as the Ethereum's Smart Contract language.

The best part about Solidity is that it is similar to Javascript which gives it a large developer community from the initial phase. The Solidity builds on the Ethereum Virtual Machine and that makes the Solidity easier to write because it is possible to write high level code that is secure and incorruptible.

There are six main function types in Solidity:

1. Public: A function that can be called by anyone.
2. Private: A function that can only be called by this contract.
3. View: A function that is used to return data and does not modify anything.
4. Constant: A function that is used to return data.
5. Pure: A function that does not read or modify the contract's data.
6. Payable: This is one of the most useful function type of Solidity. When this function is called, it has the potential to send Ether along with it.

In this project, Solidity will be used in the following sequence:

1. Smart Contract definition using Solidity.
2. Using the Solidity compiler to compile the .sol file to a .json file that can be used to make the application.
3. Deployment of the bytecode generated on the Ethereum test network. In the case of this project, we have used the Rinkeby Test Network so that we don't have to spend actual money to make the Application.

Anytime we want to modify the Smart Contract, we have to submit a transaction so that everybody on the Blockchain gets the new Contract definition.

1.5. Metamask

Metamask browser extension turns Google Chrome into an Ethereum browser letting websites retrieve data from the blockchain and letting users securely manage identities and sign transactions. When you first start up metamask, you’re given a seed phrase that can be used to restore all the accounts you ever created with Metamask. You can switch the current account with the Switch Account button and you can add more accounts.

Your account vault is encrypted and it is locally stored within your browser meaning no account information ever touchers their servers. However, with your seed phrase you can easily restore your accounts even to other browsers. At first glance, metamask enables you to send Ether like a normal wallet application. But Metamask’s true strength lies in enabling your browser to visit Ethereum enabled websites.

When you visit a App with Metamask installed, that website has access to the Ethereum blockchain via the standard ‘web3’ javascript API. When it wants to write to the blockchain, it asks web3 to send a transaction prompting Metamask to ask for user confirmation. After you submit a transaction, wait for the block or the change to be reflected on the website.

The main use of Metamask is that it can be used as an educational tool as it has test networks in addition to the connection with the actual Ethereum blockchain. In this project, we used the Rinkeby Test network for the transaction of Ethereum. Metamask connects to these blockchains with no synchronisation time at all because they host the blockchain nodes by default.

1.6. Crowdfunding

Figure 1.3 Crowdfunding Companies[8]

There are multiple crowdfunding platforms like Kickstarter, Patreon, GoFundMe etc. that are using by startups and people for fundraising their cause or company via asking the general public for money.

There is no single major decentralized fundraising platform that is in operation. The reason is due to many people making such platforms, offering Initial Coin Offerings and then scamming people with the money they received. The purpose is to build a peer-to-peer version of the crowdfunding system as a Decentralized Application using the Ethereum network so that the middlemen can be cut from the process of fundraising and the support for the cause can be maximized.

There are different kinds of crowdfunding practices that are in existence today. Some of them are:

- Asking money for causes like medical emergencies or political agendas. These crowdfunding practices are often lead by the family members or political activists.
- When some people are focusing on making a startup, they come to crowdfunding websites which can help them by asking for funds in exchange of a part ownership of the company. This is a very common practice and many famous companies have been initiated by this process.
- Some startups have the strategy of asking the price of the product first and then putting that money into production. This is a highly risky strategy because it has the potential of losing all the money if the production plans never come to life. Some startups have even committed fraud using this strategy. This is one problem that I have specifically tackled in my project. I have made sure that there is not possibility of fraud if a project or a cause is crowdfunded using CryptoBill.

Sometimes, if the total price that a project is asking for is not met, the crowdfunding company, takes all the funds. This is a bad practice in my opinion and I did not follow this strategy as I believe that a project or a cause should be able to get all the funds it has received. There are some companies that follow this practice and give all the funds that are earned to the project or cause.

2. Literature Review

2.1. Blockchain

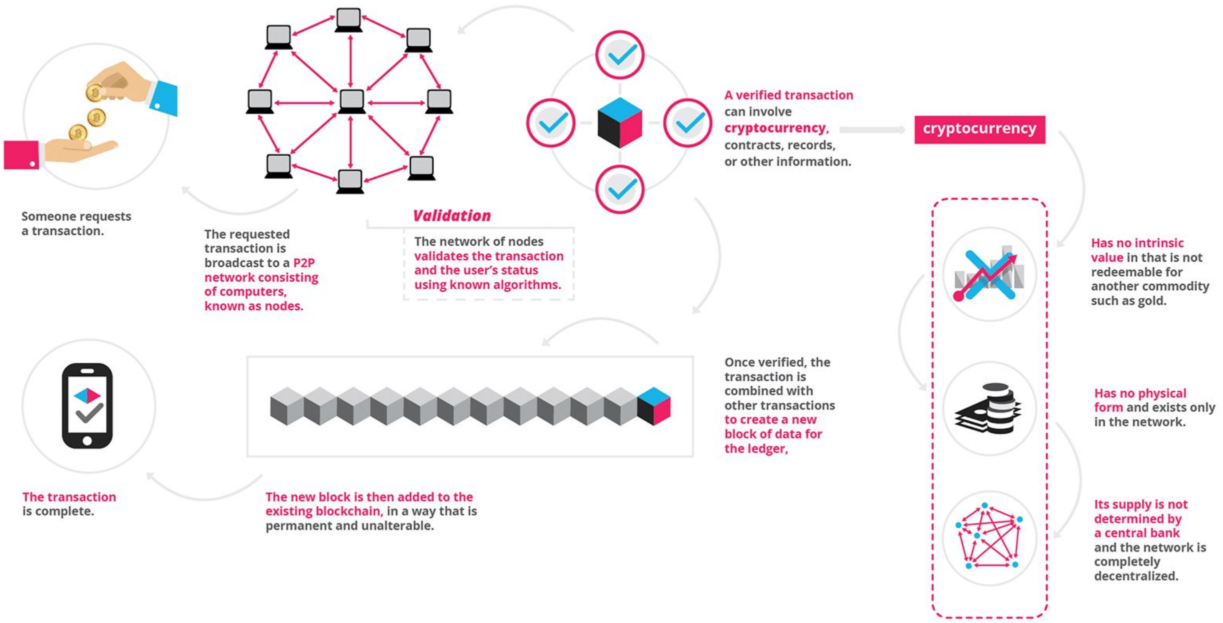

Someone requests a transaction.
The requested transaction is broadcast to a P2P network consisting of computers, known as nodes.
Validation
The network of nodes validates the transaction and the user's status using known algorithms.
A verified transaction can involve cryptocurrency, contracts, records, or other information.
cryptocurrency
Has no intrinsic value in that is not redeemable for another commodity such as gold.
Has no physical form and exists only in the network.
Its supply is not determined by a central bank and the network is completely decentralized.
Once verified, the transaction is combined with other transactions to create a new block of data for the ledger,
The new block is then added to the existing blockchain, in a way that is permanent and unalterable.
The transaction is complete.

Figure 2.1. Blockchain Flow[12]

On October 31st, 2008, when United States got into the recession, Satoshi Nakamoto, a person who is still unknown, published a white paper describing a system to allow peer to peer payments without the need of a financial intermediary. This was Bitcoin. A digital currency built on top of Blockchain which is a database that stores the record of every transaction that ever taken place on the network.

In today’s world, most of the transactions that take place have to go through a financial intermediary like a Bank to help the transaction take place because a Bank acts a trusted centre that both parties can rely upon for security. The Blockchain is a way in which people can directly transfer digital currency through the use of an open ledger which can be seen by everyone which makes it transparent and thus, incorruptible.

2.2. Ethereum

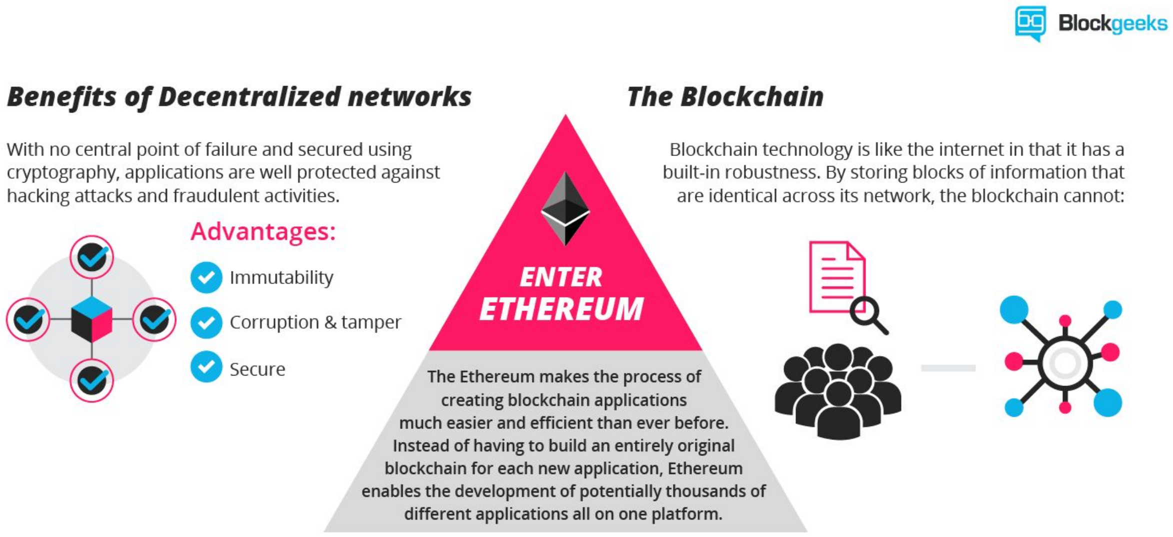

Blockgeeks
Benefits of Decentralized networks
With no central point of failure and secured using cryptography, applications are well protected against hacking attacks and fraudulent activities.
Advantages:
Immutability
Corruption & tamper
Secure
ENTER ETHEREUM
The Ethereum makes the process of creating blockchain applications much easier and efficient than ever before. Instead of having to build an entirely original blockchain for each new application, Ethereum enables the development of potentially thousands of different applications all on one platform.
The Blockchain
Blockchain technology is like the internet in that it has a built-in robustness. By storing blocks of information that are identical across its network, the blockchain cannot:

Figure 2.2. Ethereum[13]

Ethereum is a network of computers that can be used to transfer money and store data. Also, it can be used as a platform to build applications on top of.

Thanks to the power of modern communication we have the ability to create technology that are decentralized. Removing middlemen and letting users interact with each other directly on a Global network. Decentralized applications have been becoming more and more important in the past 10 years and have the benefits of massively reducing cost and barriers to entry removing single points of failure, preventing censorship and ensuring transparency and trust between all the parties involved in the transaction.

In 2003, Bittorent, was the world's first (arguably) decentralised application to be launched. It is a file sharing network that can be used to distribute content quickly and easily even if they do not have the resources to pay for their own website or server.

Five years later, Satoshi Nakamoto came up with the idea of Blockchain, a sort of distributed Database and used it to built Bitcoin, the world's first decentralised currency. Decentralized currencies like Bitcoin allow people to send money instantly anywhere around the world with no regard for national borders with negligible fees.

Bitcoin is increasingly being used for international remittances and micropayment and commerce online. Decentralized application for finance are cloud computing, messaging and governance are soon to come.

Ethereum is a platform that is designed for people to build these kinds of Decentralized applications or apps . The Ether browser will include a built in peer-to-peer network for sending messages and a generalised blockchain with a built in programming language. This will allow people to use the Blockchain for any kind of Decentralized Applications that they want to create.

As with all new platforms for innovations like the internet, it is not always easy to predict what they're going to be used for. Gmail, Facebook, Twitter, Instagram and the modern internet are the early developments of the internet world and the programming language Javascript, the programming language of the World Wide Web from the 1990s. Similarly, by providing a universal programmable blockchain and packaging it out into a client that anyone can use, the Ethereum project hopes to do the same for finance, peer-to-peer commerce, distributed governance and human collaboration as a whole.

3. Project Design, Code and Testing

3.1. Design Diagrams

3.1.1. Class Diagram

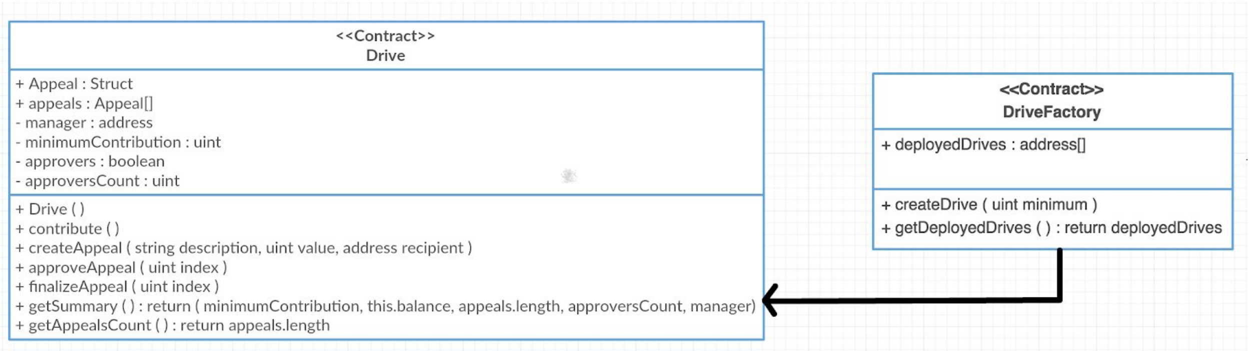

<<Contract>>
Drive
+ Appeal : Struct
+ appeals : Appeal[]
- manager : address
- minimumContribution : uint
- approvers : boolean
- approversCount : uint
+ Drive ( )
+ contribute ( )
+ createAppeal (string description, uint value, address recipient)
+ approveAppeal (uint index)
+ finalizeAppeal (uint index)
+ getSummary ( ) : return (minimumContribution, this.balance, appeals.length, approversCount, manager)
+ getAppealsCount ( ) : return appeals.length
<<Contract>>
DriveFactory
+ deployedDrives : address[]
+ createDrive (uint minimum)
+ getDeployedDrives ( ) : return deployedDrives

Figure 3.1. Class Diagram

The Class diagram connects the two contracts : Drive and DriveFactory together. It shows that the DriveFactory contract is dependent on the Drive contract for it's execution. The figure showcases all the functions and variables used in the two contracts.

3.1.2. Sequence Diagram

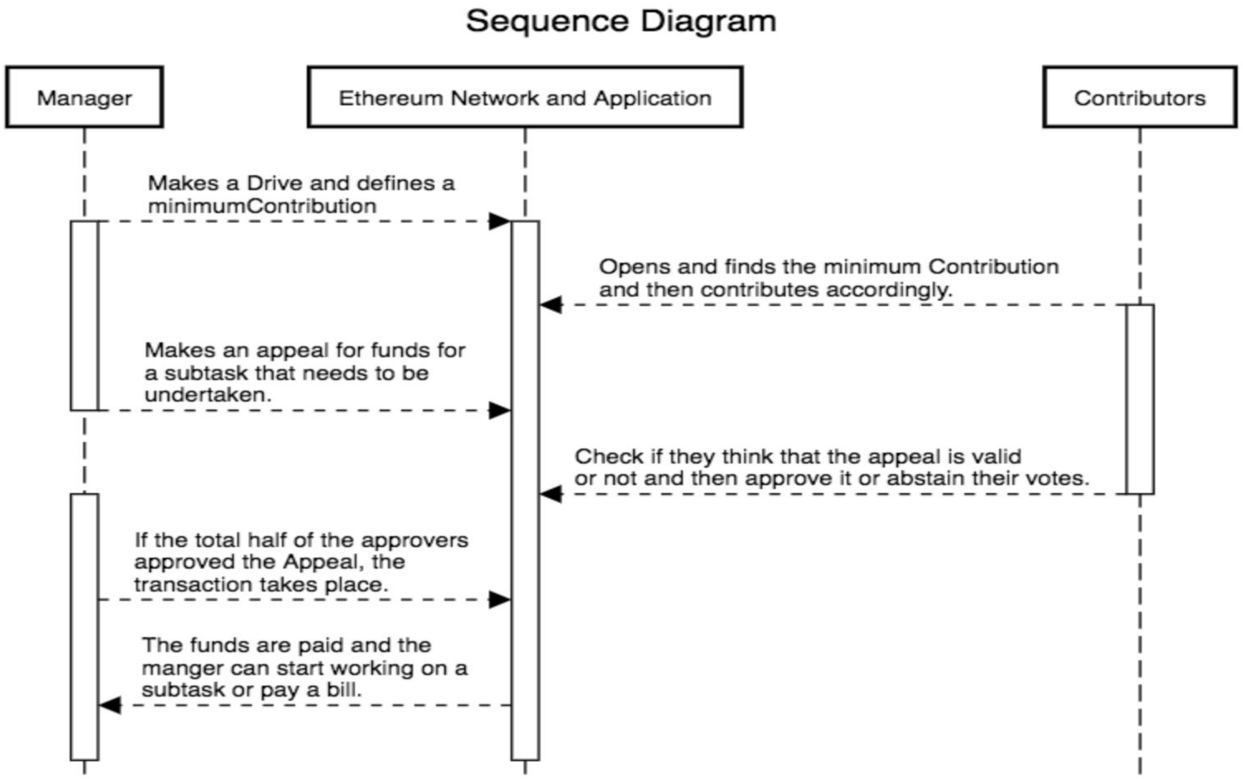

Sequence Diagram
Manager
Ethereum Network and Application
Contributors
Makes a Drive and defines a minimumContribution
Opens and finds the minimum Contribution and then contributes accordingly.
Makes an appeal for funds for a subtask that needs to be undertaken.
Check if they think that the appeal is valid or not and then approve it or abstain their votes.
If the total half of the approvers approved the Appeal, the transaction takes place.
The funds are paid and the manger can start working on a subtask or pay a bill.

Figure 3.2. Sequence Diagram

The Sequence Diagram shows the interactions of objects in a time oriented manner. Here, there are 3 key players:

- Manager - Manager is the person who initially created the Drive for a project or a cause. He has the most control throughout the execution of the project. He undertakes many responsibilities and can make Appeals and Finalize the Appeals that have been approved by the contributors.
- Ethereum Network and Application - The Ethereum network is a living object that is constantly being updated by the numerous amounts of transactions that are taking place every second. The Application plays the part of an intermediary between the contributors and the Manager of the Drive.
- Contributors - Once a contributor pays more than the minimum contribution, they have a lot of power about how the Drive will progress. They can vote on different

appeals and fund accordingly. They will have access to the address that the appeal funds will be remitted towards and from there they can decide whether or not that is a viable and necessary option.

3.1.3. Use Case Diagram

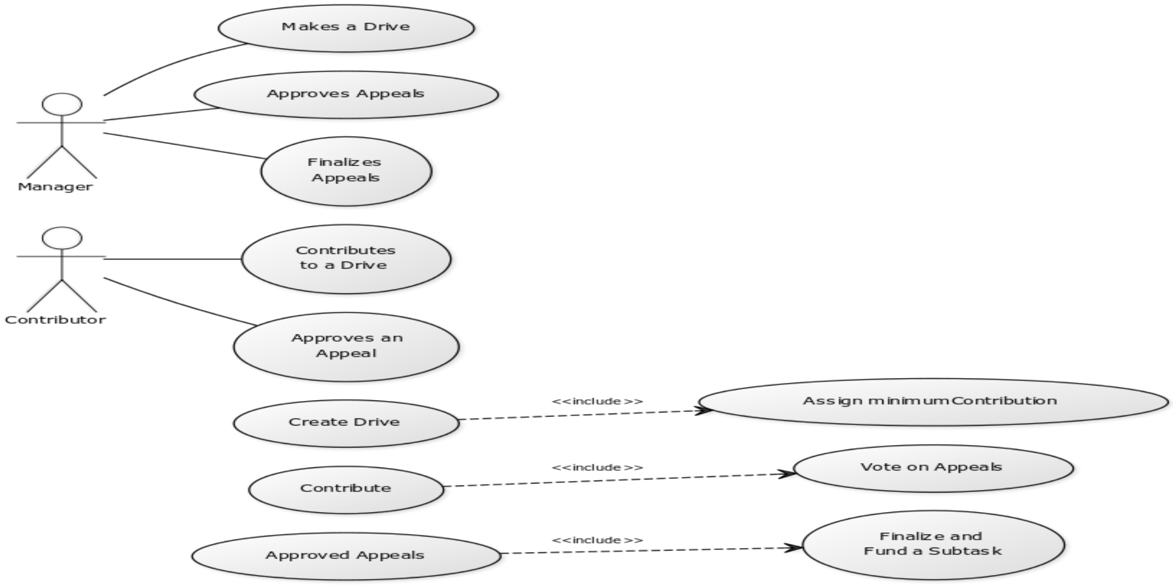


Figure 3.3. Use Case Diagram

The Use Case diagram of the project outlines the two main players of the project:

- Manager
- Contributor

The Use case diagram also shows the tasks that can be undertaken by different personnel in the project.

3.2. Structure of the code

The project has three major components:

1. *Front end Code* - The Front End is coded in Nextjs which is the Server-Side Rendering of React components.
2. *Ethereum code* - The Ethereum code comprises of a Solidity contract file and two files of Campaign Factory and Campaign Javascript Object Notation files which show the compiled Contract.
3. *Testing Code* - The testing code uses Mocha and Truffle Wallet so that they can check if all the tests are passing and that the transactions are happening appropriately.

3.3. Front end Code and Libraries

Front End Code is used to showcase the information in a clean format that has a good User Interface and leads to a good User Experience. I considered using basic HTML, CSS and Javascript for the completion of the project but I decided to go with a Front End framework called next.js because it offers many advantages that HTML, CSS and Javascript won't be able to give.

Next.js is built on top of React and Webpack which uses Node Package Manager for managing the dependencies and libraries. When code is taken and rendered on the next server, it reaches out to the Ethereum network using the APIs and makes some initial calls so that the contract can be used to make transactions.

The following libraries were used in the project:

1. React : React is an open source Javascript library breaks down website contents into components and helps in building extremely fast User Interfaces. It was made by Facebook and it has an MIT license.
2. web3 : web3 is an Ethereum compatible Javascript API that can be used to take to an Ethereum node with the use of Javascript programming language because Javascript is the language of the web. web3 is available in NPM as a node module which I added into the project.
3. Semantic-UI-React : Semantic UI React is available in NPM as a node module that can be used to style the web pages that are based on the React. It is used by the major players of the Tech industry like Microsoft, Netflix, Amazon etc. for their styling needs.

3.4. Solidity Code and Libraries

The Smart Contract is written in the Solidity programming language. It has two main contracts:

- DriveFactory Contract

  The DriveFactory contract creates new Drives and can fetch the deployedDrives. Drives are the projects that people want the funding towards.

  It comprises of 2 public functions:

  - createDrives
    - createDrives, as the name suggests, helps in the creation of new drives.
  - getDeployedDrives
    - getDeployedDrives, as the name suggests, fetches the information about the drives that have already been deployed on the Ethereum network.

- Drive Contract

  The Drive Contract has many functions which facilitate the proper functioning of a particular drive.

  The Drive contract comprises of the Appeal struct definition. An Appeal struct is used to ask for a particular amount of funds needed to complete that definition subtask.

  The Drive contract comprises of 7 public functions:

  - Drive
    - Drive function is a constructor function that builds the contract using a manager and a minimumContribution.
  - Contribute
    - Contribute function is a payable function that can be used for transactions and makes sure that the new contribution is more in value than the minimumContribution.

- createAppeals
    - createAppeals function, as the name suggests, can be used to fund a defined subtask of the project. For an appeal to be created, there needs to be a reason outlined that can be seen by the public.
- approveAppeals
    - approveAppeals function is used to make sure that the task that is going to be undertaken goes through with complete transparency. The verified contributors will vote on every single Appeal to conclude if that Appeal is valid or not. If more than half of the contributors approve the Appeal using the Approve button, and then the funds can be transacted after the Appeal is finalized.
- finalizeAppeals
    - finalizeAppeals function is a function that is added measure to make sure that more than half of the approvers are backing this subtask so that it can be undertaken. After the Appeal is finalized, the ‘Approve’ and ‘Finalize’ buttons corresponding to the Appeal will disappear and it will showcase that the Appeal has been funded.
- getSummary
    - getSummary function is a function that is used to fetch the summary of a contract. It returns the balance of the Drive, the minimumContribution needed to enter into a Drive, the number of Appeals, the number of Approvers that are need to approve a particular Appeal and the address of the manager of the contract.

3.5. Remix Editor

The Remix Editor is an online Integrated Development Environment that can be used to test out the Solidity Contracts and interact with the Ethereum blockchain.

I used the Remix Editor extensively throughout the project to make sure that my contract was running in a secure and proper way. The Remix Editor is an open source software that was made by the Ethereum Organization.

Remix is built out of 4 different modules:

- remix-solidity provides Solidity analysis and decoding functions.
- remix-lib
- remix-core is a utility package, providing high-level abstractions to work with the Ethereum VM.
- remix-debugger contains the debugging webapp.

3.6. Rinkeby Test Network, Rinkeby Faucet and Infura

- Rinkeby Test Network

  For this project, I did not want to spend actual money because the development costs would have skyrocketed as 1 Ether is equal to more than INR 40,000. So, I used the Rinkeby Test Network to use the Ethereum Blockchain for development purposes. It can be used to deploy a contract on the Rinkeby Test Network.

- Rinkeby Faucet

  Although there is a free network to test on, it does not come with Ether initially. The Ether has to be added to an account using the Rinkeby Faucet. Rinkeby Faucet gives out 18 ETH which don't have any monetary value for Testing purposes.

- Infura

  Rather than using a local node for testing and deploying the contract on the Rinkeby Test Network, I made use of Infura, which is a public API which can be signed up for and it gives access to a node on the the Rinkeby Network. The best feature of Infura deployment is that it can just as easily be used to port to the Actual Ethereum Network and make transactions right away.

3.7. Testing with Ganache

For any application that deals with money, security and transparency are two of the most important aspects. This application is no exception to that thought. So, I decided to use the Mocha testing framework where I can describe tests and run them automatically without the need of actually accessing the Ethereum Network because multiple tests could take a lot of time to be processed if it were being tested on the actual network.

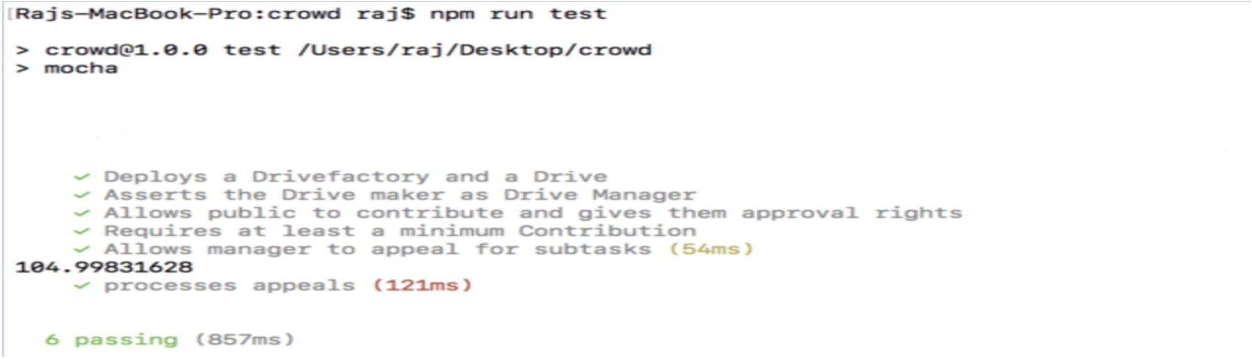
For the purposes of the Testing on the local machine, we'll use 'ganache-cli' API which can give us access to a Ganache Local Test Network to facilitate the testing process. It will be used for testing purposes with conjunction to the 'web3' library.

It undertakes 6 tests for various purposes of the application:

- Deploys DriveFactory and Drive contract.
- Asserts the Drive maker as Drive Manager.
- Allows public to contribute and gives them approval rights.
- Requires at least a minimum Contribution.
- Allows manager to appeal for subtasks.
- Processes Appeals.

All these test cases make sure that the Application is running in a proper way.

The following image shows that all these test cases are passing:

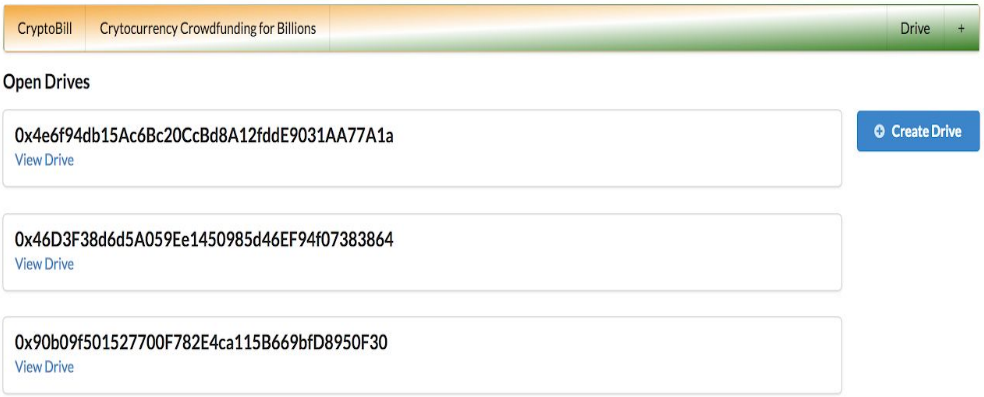


```
[Rajs-MacBook-Pro:crowd raj$ npm run test

> crowd@1.0.0 test /Users/raj/Desktop/crowd
> mocha



      ✓ Deploys a Drivefactory and a Drive
      ✓ Asserts the Drive maker as Drive Manager
      ✓ Allows public to contribute and gives them approval rights
      ✓ Requires at least a minimum Contribution
      ✓ Allows manager to appeal for subtasks (54ms)
104.99831628
      ✓ processes appeals (121ms)


  6 passing (857ms)
```

Figure 3.4. Terminal Testing

4. Project Screens

4.1. Landing Page



Open Drives

0x4e6f94db15Ac6Bc20CcBd8A12fddE9031AA77A1a
View Drive

0x46D3F38d6d5A059Ee1450985d46EF94f07383864
View Drive

0x90b09f501527700F782E4ca115B669bfD8950F30
View Drive

Figure 4.1. Landing Page

Landing page is the first page a user visits when they come on the site. The landing page has the project name and slogan in the header along with the Button to create a Drive.

The Open Drives component showcases the Drives that are live at the moment and their account addresses which can be used to uniquely identify which drive a user might want to fund. The View Drive button opens a different page where all the information about that particular drive is situated.

4.2. Create Drive Page

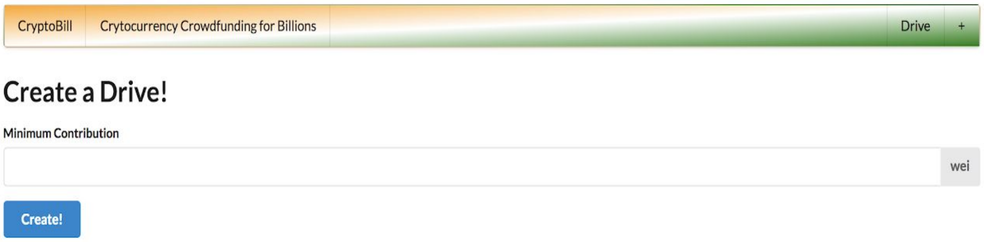
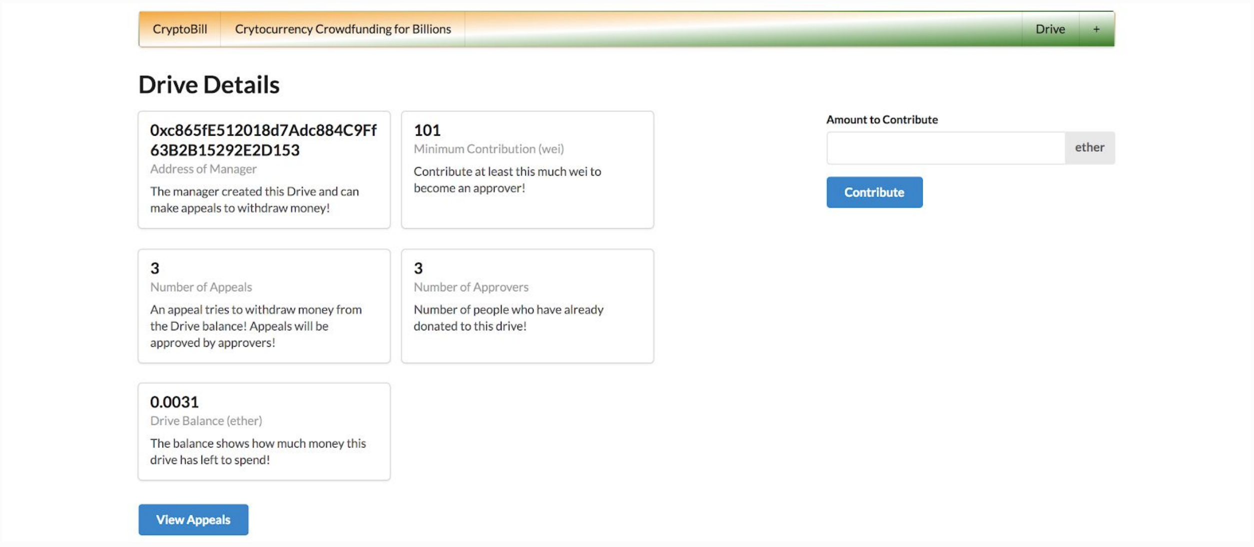


Figure 4.2. Create Drive Page

The Create Drive page is the page that can be used to create a drive for a project or a cause. The person who creates the drive will decide what the minimum Contribution needs to be so that the contributor can be assigned the duty of the approver. This value is decided in ‘wei’ which is 1/100000 value of the Ether.

4.3. Drive Details Page



Drive Details

0xc865fE512018d7Adc884C9Ff63B2B15292E2D153
Address of Manager
The manager created this Drive and can make appeals to withdraw money!

101
Minimum Contribution (wei)
Contribute at least this much wei to become an approver!

3
Number of Appeals
An appeal tries to withdraw money from the Drive balance! Appeals will be approved by approvers!

3
Number of Approvers
Number of people who have already donated to this drive!

0.0031
Drive Balance (ether)
The balance shows how much money this drive has left to spend!

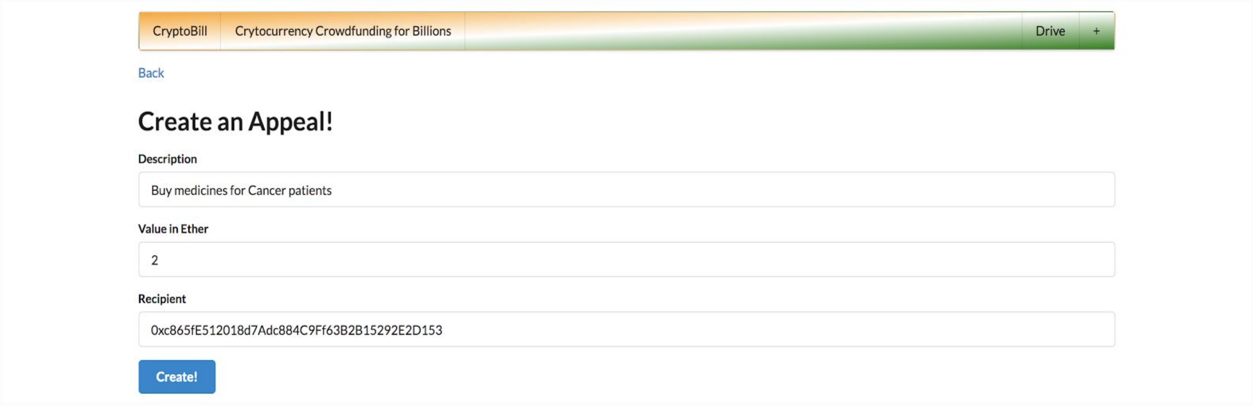
Amount to Contribute: ether

Figure 4.3. Drive Details Page

The Drive Details is the most informative page on the site. It gives all the necessary details about the Drive:

- Address of the manager who created the Drive.
- Number of Appeals made to fund the subtasks.
- Number of Approvers that have made the minimum Contributions to the Drive.
- Drive Balance, the amount of balance left in the Drive funds to fund the rest of the Appeals.
- Minimum Contribution, the minimum amount of Wei someone needs to contribute to become and approver.

4.4. Add Appeal Page



Create an Appeal!

Description

Buy medicines for Cancer patients

Value in Ether

2

Recipient

0xc865fE512018d7Adc884C9Ff63B2B15292E2D153

Figure 4.4. Add Appeal Page

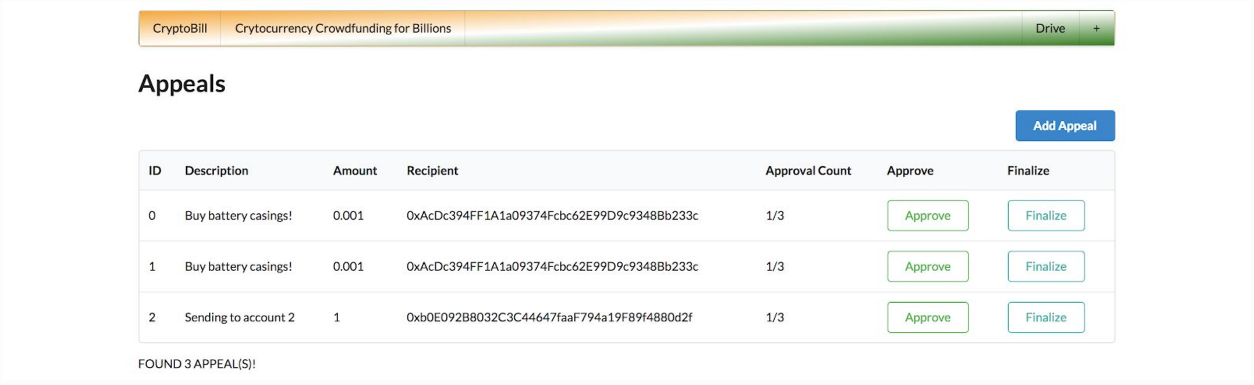
The Add Appeal Page can be used by the manager of the Drive to make an appeal to fund subtasks of the Project or the cause. The Page has three input boxes:

- Description - the reason for the appeal
- Value - the amount needed to fund this particular appeal
- Recipient - the address of the recipient which can be verified so that fraud cannot take place and the funds go towards something helpful for the project or cause.

4.5. Appeal Action Page

There are two views of this page:

- Unapproved Appeal Page



Appeals

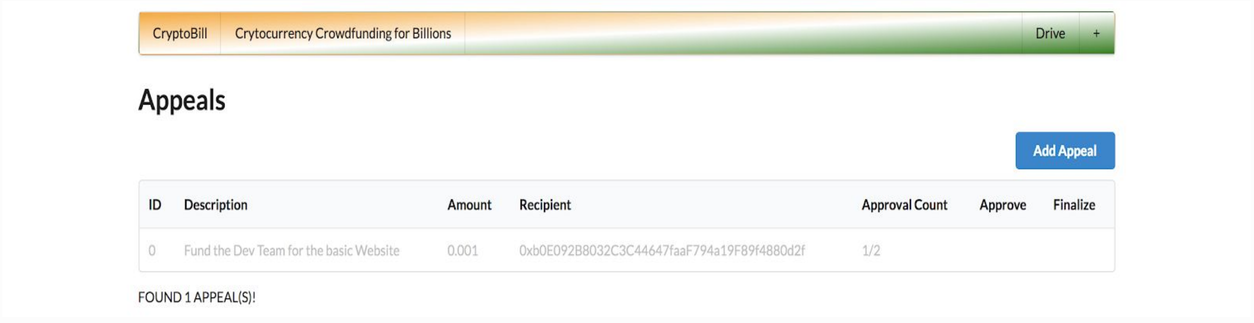
ID	Description	Amount	Recipient	Approval Count	Approve	Finalize
0	Buy battery casings!	0.001	0xAcDc394FF1A1a09374Fcbc62E99D9c9348Bb233c	1/3	Approve	Finalize
1	Buy battery casings!	0.001	0xAcDc394FF1A1a09374Fcbc62E99D9c9348Bb233c	1/3	Approve	Finalize
2	Sending to account 2	1	0xb0E092B8032C3C44647faaF794a19F89f4880d2f	1/3	Approve	Finalize

FOUND 3 APPEAL(S)!

Figure 4.5. Unapproved Appeal Page

The Unapproved Appeals need to go through a stringent process of approval by the approvers so that the people who have funded the project know exactly where the money is going. The Finalize button will work when the value of Approvers count is more than 0.5, which means that more than half of the approvers think that the subtask should be funded towards.

- Approved Appeal Page



Appeals

ID	Description	Amount	Recipient	Approval Count	Approve	Finalize
0	Fund the Dev Team for the basic Website	0.001	0xb0E092B8032C3C44647faaF794a19F89f4880d2f	1/2		

FOUND 1 APPEAL(S)!

Figure 4.6. Approved Appeal Page

The Approved Appeal page shows the appeal that has been Approved and Finalized by the contributors and the manager. In this case, the Developer team has been given the 0.001 Ether for the formation of the basic website for the project.

5. Conclusion

5.1. Result

The Ethereum Application was developed for the use of Crowdfunding using Cryptocurrency which will remove the need of a Financial Institution which would help the organization or cause benefitting much more than a normal Application like GoFundMe, Kickstarter etc. who charge outrageous transaction fees.

5.2. Future Scope

The scope of improvement for this application is nearly limitless. There are several things can be incorporated into this application that will make it better:

- Chat Box : Using the Chat Box, the Managers can directly talk to their contributors and expedite the process of funding for the Drive.
- Partnerships : In the future, this application can be partnered with financial currency alternatives for the purposes of verification of the Drive's credibility and extension of the application itself.
- Stronger Testing : With the addition of other features, there will be a need for extensive testing so that all the transactions and the conversations remain incorruptible.
- Better Design : We can use Google Analytics to find out the behavior patterns of the customers who visit the application and then model the application accordingly so that anyone who visits the site has the best User Experience possible.

6. References

[1] “Bitcoin: A Peer-to-Peer Electronic Cash System”, Satoshi Nakamoto, 2008. [Online] Available: https://bitcoin.org/bitcoin.pdf. [Accessed: 10-Jan-2018].

[2] “DAOs, DACs, DAs and More: An Incomplete Terminology Guide”, Vitalik Buterin, 2013. [Online]. Available: https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide. [Accessed: 15-Jan-2018].

[3] “Ethereum White Paper”, Vitalik Buterin, Dr. Gavin Wood, 2008. [Online]. Available: https://www.ethereum.org/pdfs/EthereumWhitePaper.pdf. [Accessed: 15-Jan-2018].

[4] “What is Ethereum? A Step-by-Step Beginners Guide”, Ameer Rosic, 2017. [Online]. Available: https://blockgeeks.com/guides/ethereum. [Accessed: 5-Feb-2018].

[5]“Solidity”, Numerous contributors from GitHub, 2016. [Online]. Available: https://solidity.readthedocs.io/en/develop/. [Accessed: 5-Feb-2018].

[6] “Introduction to Smart Contracts”, Numerous contributors from GitHub, 2016. [Online]. Available: http://solidity.readthedocs.io/en/develop/introduction-to-smart-contracts.html. [Accessed: 5-Feb-2018].

[7] “Smart Contracts: The Blockchain Technology That Will Replace Lawyers”, Ameer Rosic, 2016. [Online]. Available: http://blockgeeks.com/guides/smart-contracts/. [Accessed: 5-Feb-2018].

[8] “Comparing Crowdfunding Websites: 4 Platforms Rated and Reviewed”, Anonymous site contributor, 2017. [Online]. Available: http://doublethedonation.com/blog/2017/05/comparing-Crowdfunding-websites/. [Accessed: 25-Feb-2018].

[9] “So You Want To Crowdfund Your Startup App?”, Matt Ward, 2015. [Online]. Available: https://www.smashingmagazine.com/2015/08/crowdfunding-a-startup-app/. [Accessed: 25-Feb-2017].

[10] “Ethereum IDE and tools for the web”, Numerous GitHub contributors, 2014. [Online]. Available: https://github.com/ethereum/remix. [Accessed: 22-Apr-2018].

[11] “How to Code a Decentralized Applications or Dapps”, Ameer Rosic, 2018. [Online]. Available: https://blockgeeks.com/guides/decentralized-applications/. [Accessed: 20-Mar-2018].

[12] “What is Bitcoin? A Step-by-step guide for Beginners”, Ameer Rosic, 2017. [Online]. Available: https://blockgeeks.com/guides/what-is-bitcoin/. [Accessed: 20-Mar-2018].

[13] “What is Ethereum? A Step-by-step Beginners guide”, Ameer Rosic, 2017. [Online]. Available: https://blockgeeks.com/guides/ethereum/. [Accessed: 20-Mar-2018].