

DECENTRALIZED CROWDFUNDING PLATFORM

A Project Report

Submitted in Partial Fulfillment of the Requirement for the
Degree of

BACHELOR OF TECHNOLOGY

In

(INFORMATION TECHNOLOGY)

To



**Dr. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY
LUCKNOW**

Submitted by

SNEHIL GUPTA

University Roll No. 1801013074

SAUMYA SHUKLA

University Roll No. 1801013058

Under the Supervision of

Mr. PRASHANT SONI

Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNITED COLLEGE OF ENGINEERING AND RESEARCH,
PRAYAGRAJ**

JUNE 2022

CANDIDATE'S DECLARATION

We, **SNEHIL GUPTA** (1801013074) and **SAUMYA SHUKLA** (1801013058), students of B.Tech of Information Technology hereby declared that we own the full responsibility for the information, results etc. provided in this project titled “**DECENTRALIZED CROWDFUNDING PLATFORM**” submitted to Dr. A.P.J Abdul Kalam Technical University, Lucknow for award of B.Tech (Information Technology) degree. We have taken care in all respect to honor the intellectual property right and have acknowledged the contributions of others for using them in this academic purpose. We further declared that in case of any violation of intellectual property right or copyright, we as the candidate would be fully responsible for the same. Our supervisor and institute should not be held for full or partial violation of copy right if found at any stage of our degree.

Date:

Place: Prayagraj

Snehil Gupta

(1801013074)

Saumya Shukla

(1801013058)

CERTIFICATE

This is to certify that the project work entitled “**DECENTRALIZED CROWDFUNDING PLATFORM**”, submitted by SNEHIL GUPTA (1801013074), SAUMYA SHUKLA (1801013058), to the Dr. A.P.J. Abdul Kalam Technical University, Lucknow, for the partial fulfilment of the requirement for the award of **Bachelor of Technology (Information Technology)** degree, is a record of student’s own study carried out under my supervision and guidance.

This project has not been submitted to any other university or institution for the award of any other degree.

SUPERVISOR

(Mr. Prashant Soni)

Assistant Professor

Department of CSE

UCER, Prayagraj

ACKNOWLEDGEMENT

We, SNEHIL GUPTA (1801013074) and SAUMYA SHUKLA (1801013058) are grateful to the management of United College of Engineering & Research for providing us an opportunity to undertake our major project in its prestigious college. We are grateful and thankful to **Mr. PRASHANT SONI** and all the other senior staff of the college, as they have helped us in every way possible. They put us under good supervision which helped in learning a lot of new things about the project and its various applications. They also provided us with all the necessary information needed. We also take the opportunity to offer our sincere thanks and deep sense of gratitude to **Mr. PRASHANT SONI** for attending us throughout the course of this project. We must make special mention of our **H.O.D. Dr. Vijay Kumar Dwivedi**, for providing us a platform to complete our project successfully. We would thank all the lab maintenance staff for providing assistance in various H/W & S/W problem encountered during course of our project. We are also very thankful to respected principal sir who gave us an opportunity to present this project.

ABSTRACT

The Crowdfunding platform in block chain makes different possibilities for the startups by raising the funds to create their own business and it is a peer-to-peer fundraising model. Once it works, it will offer your business an enormous boost. Firms like Kickstarter, Crowd funder were among the earliest to create it well-liked. One drawback is that even with crowdfunding, the model remains very inefficient. In the case of Kickstarter, most of the campaigns that got raised were mostly fraud.

This project is a trusted and decentralized crowdfunding platform which is implemented using block chain as backend and react as front end.

Table of Content

Contents

Candidate's Declaretion	2
Certificate	3
Acknowledgement	4
Abstract	5
Chapter 1: Introduction	9
1.1 Project Description	9
1.2 Methodology.	18
1.2.1 Traditional Crowdfunding	11
1.2.2 Blockchain Based Crowdfunding.	12
1.3 Blockchain Technology.	13
1.4 Innovation of Idea.	14
1.4.1 Kickstarter's Idea	21
1.4.2 Demerits of such idea	22
1.4.3 Existing System	23
1.5 System Description.	15
1.6 Advantages	16
1.7 Objective	19
1.8 Scope	20
1.9 Tools and Technologies Used	24
1.9.1 Software Tools	25
1.9.2 Hardware Specifications	26
CHAPTER 2: Design	25
2.1 Software Development Life Cycle Model	28

2.2 Module Description	29
2.3 Data Flow Diagram	31
2.4 Use Case Diagram	33
2.5 Class Diagram	35
2.6 Sequence Diagram	36
2.7 Structure of the Campaign	37
Chapter 3: Interface Design	38
3.1 Index	38
3.2 Create a new campaign	39
3.3 View details of the new campaign	40
3.4 Create a spending request	41
3.5 View details of the spending request	42
Chapter 4: Testing	43
Chapter 5: Outcome	45
Chapter 5: Conclusions	47
Chapter 6: References	49
Chapter 7: Certificates	54
Chapter 8: Plagiarism Check	55

CHAPTER 1: INTRODUCTION

1.1 PROJECT DESCRIPTION

In today's world, block chain-based systems are in demand across various industries, because of its secure, trusted, and decentralized network as well as for being more efficient than the traditional methods.

However, the traditional ways these days are facing a lot of issues and challenges because of the complex and less secure network. Block chain network integration overcomes the problems faced by traditional methods across industries. The block chain integrated network provides benefits such as increased security, increased transparency, increased efficiency and decreased chances of fraud. Although the blockchain-based systems provide various benefits, due to lack of knowledge about this technology, the implementation rate is low. In this work, we have highlighted the distinction between the traditional crowdfunding platform as well as blockchain network-based crowdfunding platform and the benefits of implementing blockchain network in other sectors.

This work highlights the issues and challenges faced by the industries, as mentioned earlier, by using the traditional methods as well as the solutions to the problems provided by the blockchain network-based systems to those industries. This work helps the people to understand the benefits of blockchain network-based systems in their respective industries as well as execute it to improve the transparency, efficiency, and security of the system altogether.

1.2 METHODOLOGY

1.2.1 Traditional Crowdfunding Concept

Most ancient business funding takes one in all 3 forms: self-funding, Bank funding, or working capital. The problem is that for many folks, self-funding is implausibly restricted. Bank funding needs having an existing business with sensible revenues and income. And venture fund capital nearly invariably needs a product or service that has mass Attractiveness. This makes ancient funding terribly restricted and Laborious to induce for newer businesses. It will inhibit growth even for products and services with immense potential.

Crowdfunding permits businesses with very nice product and repair ideas to lift funds from regular folks in tiny investment amounts.

Once it works, it will very offer your business an enormous boost.

Firms like Kick-starter, Indiegogo, and Crowd funder were among the earliest to create it well-liked.

One drawback is that even with crowd funding, the model remains very inefficient. In step with Kickstarter, seventy eight of campaigns that raise 2 hundredth of their goal ultimately become absolutely funded, whereas Martinmas of comes end having ne'er received any funding the least bit.

This brings United States of America to however blockchain is dynamical the crowdfunding landscape.

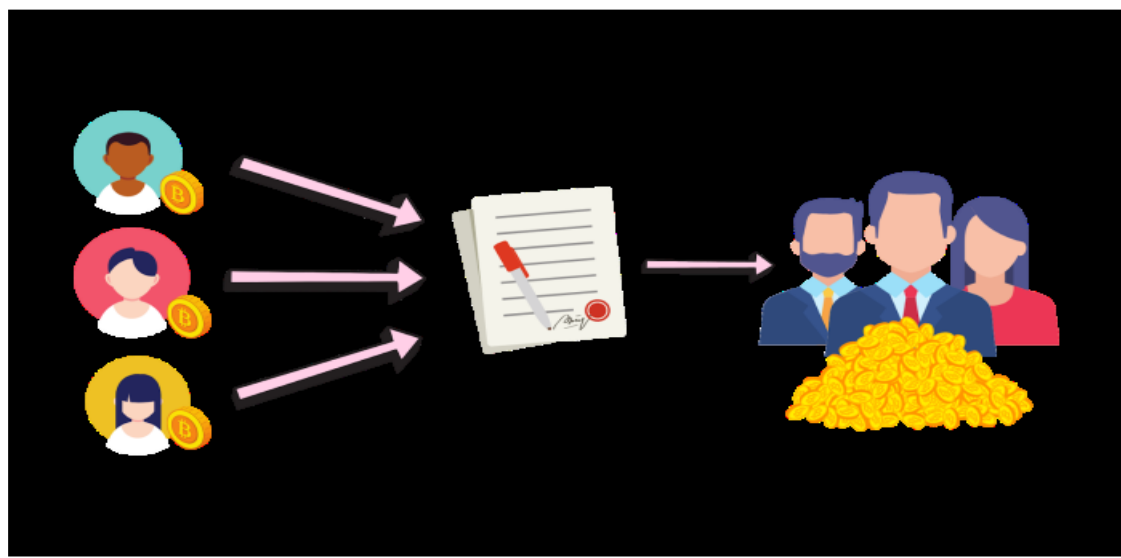
1.2.2 Blockchain Based Crowdfunding

Blockchain, typically named as Distributed Ledger Technology (DLT), makes the history of any digital plus unalterable and clear through the utilization of decentralization and science hashing.

A simple analogy for understanding blockchain technology could be a Google Doc. Once we produce a document and share it with a gaggle of individuals, the document is distributed rather than derived or transferred. This creates a decentralised distribution chain that provides everybody access to the document at identical times. nobody is fastened

out awaiting changes from another party, whereas all modifications to the doc are being recorded in a period of time, creating changes fully clear. Of course, blockchain is additional difficult than a Google Doc, however the analogy is apt as a result of it illustrates 3 important ideas of the technology.

Blockchain is Associate in Nursing particularly promising and revolutionary technology as a result of it helps cut back risk, stamps out fraud and brings transparency during a climbable means for myriad uses.



SMART CONTRACT

1.2.3 Blockchain Technology

Satoshi Nakamoto invented blockchain in 2008. Nakamoto's view is that blockchain provides peer to peer topology, i.e. There is no third party involved. The payments are sent from the sender to the receiver without going through the third party. The central authority prevents dishonest users to prevent fraudulence activities. Blockchain is a distributed ledger in which all transaction of the funders is recorded. The transactions which occur are all stored permanently in the blockchain network. The main reason for the blockchain to be trusted is that the information that is stored in the blocks cannot be changed. Block are record book which contain the details of transaction data. A block consists of four details- hash of previous block, transaction data, value of nonce, hash. As each block contains the hash value of previous node, if one tries to change the data of one node it will affect all further node of it, thus other on the network can easily find that fault occurred on that blockchain or some data has been changed by someone. The categories of blockchain are permissioned and permissionless. In permissioned blockchain the users who publish the block must be an authorized user. In permissionless blockchain the user doesn't need any permission from the authority to publish a block. Anyone can publish in it. The features of blockchain are- SHA-256 encryption, public & private key, distributed public ledger, proof of work, and mining. Blockchain is based on the transactions which occurs. The transactions are sent to each node. Each nodes records data for each new transaction. The current transactions are checked against the previous transactions. It is called as Proof of work. At each nodes proof of work is established. Once proof of work is generated, the nodes along with the proof of work is sent to all the other nodes. The other nodes validate and approves the transaction. A new block is formed in the blockchain when all the other blocks accept the transaction. The characteristics of blockchain are immutable, decentralized, consensus driven and transparent. The

transactions which are stored in the blockchain cannot be changed and increases the trust on the transactions that are recorded. Hence it is immutable. The information that are stored in blockchain can be accessed from anywhere. Hence it is decentralized. Blockchain contains certain rules to independently verify and it displays the efforts. Hence it is consensus driven. All the parties in the network can see and verify the transactions. Hence it is transparent.

1.3 INNOVATION OF IDEA

1.3.1 Kickstarter's idea

- Kickstarter is an American crowdfunding application, where creator can put up his idea and in return receive fund.
- At first, the project creator will put up his idea on this application along with its requirements.
- Upon liking the idea, the investors can fund the project in terms of cost and maintenance required for the project.
- The project creator will choose a deadline and minimum funding goal.
- If the goal is not met by the deadline, no fund are collected.

1.3.2 Disadvantages of such idea

In case of Kickstarter, the data is stored in a normal database which is mutable in nature.

So, some of the issues for this kind of service is that:-

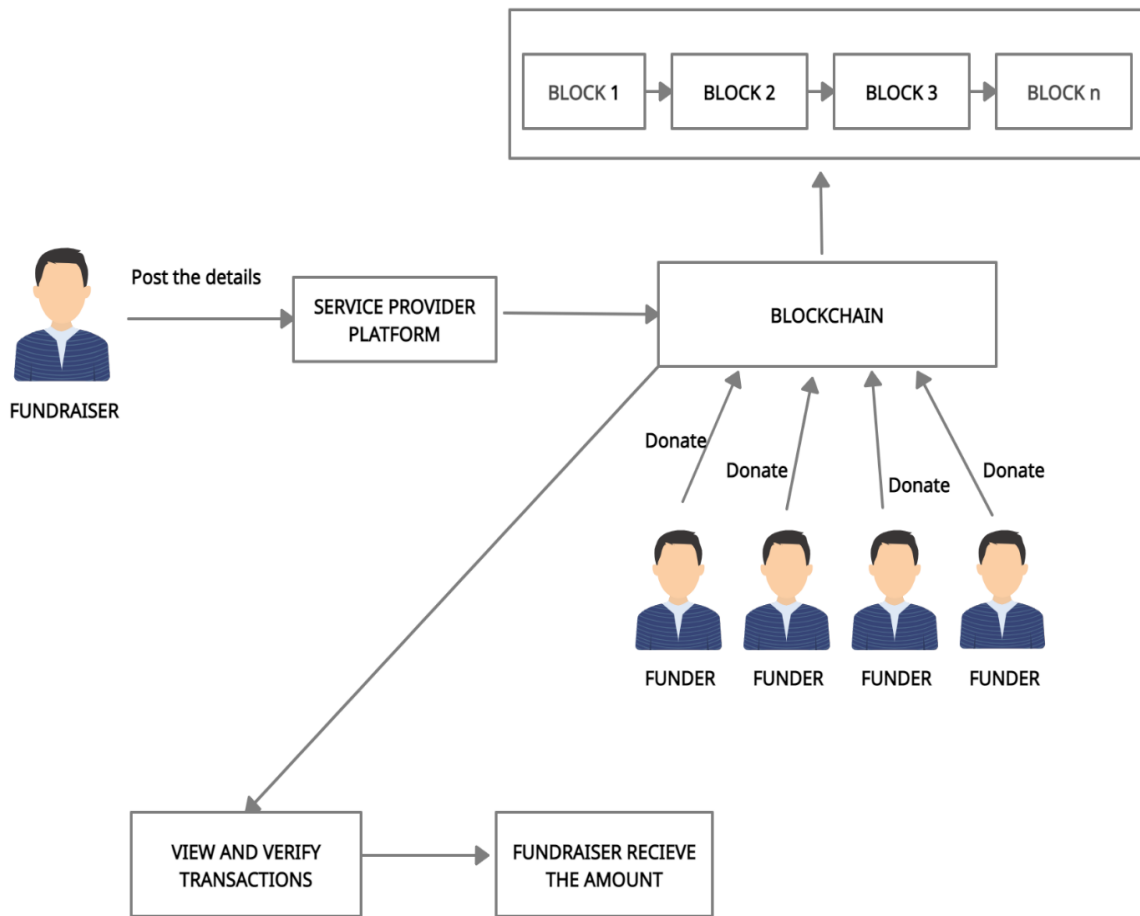
- People who post the project will deliver on their project or not.
- The creator can even do some fraudulent activity such as running away with the fund.
- Either the completed project will meet the funder's expectations or not.

1.3.3 Existing System

In Present Crowdfunding system, the fundraiser and the funder will have to log in to the fundraising platform. And if they are a new user then they have to register in the fundraising platform. Here the fundraising platform is monitored by the fundraising agency(third-party). The fundraiser posts the need to raise funds in the fundraising platform. The funders who wish to donate can donate in this platform. Once the fundraiser received the entire amount that he has specified the post will be removed by the fundraising agency. But the funders will have no control over the money they have donated. The funder will have no idea if the money that he has donated has reached the fundraiser or not. The third-party i.e., the fundraising agency will have complete control over the money and hence they can perform malicious activities. The third-party involvement may also cause additional cost. As this platform is centralized by someone there is a chance to use the money the funders have donated in an illegal way.

1.4 SYSTEM DESCRIPTION

This is a website-based system in which the fundraiser and funder have to register in the service provider platform. The fundraiser will raise their funds by describing the need and providing some proof if available. Then the funder can view all the information posted by various fundraiser. If the funder wishes to donate for any particular cause they can proceed with it. Then the money that is donated by the funder will be added to the account of the fundraiser. The sum of the amount that has been donated by all the funders will also be displayed, with the help of this we can identify whether the specified amount has been reached or not. If the specified amount has been reached then the particular post will be automatically removed from the platform. Once they donated the money to the fundraiser the information of the transaction is stored in blockchain and that information can be viewed by both the fundraiser and funder. The funder can track their money until the transacted amount reaches the valid recipient.



System Description

The modules involved in this project are

1. Registration-Fundraiser
2. Registration-Funder
3. Service provider platform
4. Transaction platform
5. Block chain creating and mining

The fundraiser will register in the service provider platform. For storing the meta details about the fundraiser we can use the MySQL data base. Also, every time scanned the immutable entry added in the table. It consists of 2 entities, the registration platform and the fundraiser. Similar to the fundraising registration process, donors certainly need to register with a fundraising platform service provider. It consists of 2 entities, the registration platform and the fundraiser. For storing the meta details about the funder we can use the MySQL data base. Also, every time scanned the immutable entry added in the table. In the service provider platform, the fundraiser will raise their fund by posting their requirement and details about the requirement. These data are stored in MySQL database. In the transaction platform each transaction details are recorded in the blockchain for e.g.: If the funder donates the amount to the fundraiser this transaction is stored in the blockchain. All the transaction received can be viewed by the fundraiser. The funder can track the transaction and check whether the amount is received by the valid fundraiser. In blockchain creation and mining the blocks are created. The data is scanned on each new block created.

1.5 OBJECTIVE

- To implement a decentralized application based on crowdfunding.
- To use a front-end framework in order to interact with the application.
- To allow the fund raisers to fund a project by sending ether to the e-wallet of the project creator.
- To test the application before deploying it into the main ethereum blockchain network.
- For future scope, we'll add a functionality for investors to report the person if he attempts to fraud.
- Our main aim for this project is to help the funders to avoid any kind of fraudulent risks that were mentioned before.
- For this, we will use Ethereum blockchain in place of a normal database to make sure the application is immutable and tamper-free.
- The fund will only be get transacted permanently only if the project is delivered to the investors properly.

1.6 ADVANTAGES

Decentralization:

Startups are not going to rely on any platform or combination of platforms to enable creators to raise funds. Startups no longer be beholden to the rules, regulations, and which of the most popular crowdfunding platforms on the internet. This makes crowdfunding less expensive for creators and investors.

Smart Contracts:

There are several ways in which blockchain-enabled smart contracts could provide greater accountability in crowdfunding. Primarily, these contracts would provide built-in milestones that would prevent funds from being released without provenance as to a project or campaign's legitimacy.

1.7 SCOPE

Use in Digital Advertising:

Digital advertisers and brands are facing several challenges in the form of bot traffic, payment inefficiencies, lack of data transparency, etc. With some industry players indulging in malpractices to drive profits, it has become ever so important for others to take corrective measures. Decentralized verification can address issues in payments and fake traffic.

Implementation in Finance:

Security, and transparency are two of the underlying features of Blockchain, With blockchain implementation, sectors like trade finance witnessed reduced processing time, eliminated paperwork, and became cost-efficient while maintaining security and trust.

1.8 TOOLS AND TECHNOLOGY USED

1.7.1 Software tools

UI/UX design:

1) Balsamiq wireframes:

Balsamiq Wireframes is **a user interface design tool for creating wireframes** (sometimes called mockups or low-fidelity prototypes). You can use it to generate digital sketches of your idea or concept for an application or website, to facilitate discussion and understanding before any code is written.

2) Adobe XD:

Adobe XD is a powerful and easy-to-use vector-based experience design platform that **gives teams the tools they need to craft the world's best experiences collaboratively**.

Available on Mac and Windows systems, XD meets teams where they're working with cross-platform compatibility.

Front-end framework :

1) React:

React is **a declarative, efficient, and flexible JavaScript library for building user interfaces**. It's 'V' in MVC. ReactJS is an open-source, component-based front-end library responsible only for the view layer of the application. It is maintained by Facebook.

One of the main benefits of using React JS is **its potential to reuse components**. It saves time for developers as they don't have to write various codes for the same features. Furthermore,

if any changes are made in any particular part, it will not affect other parts of the application.

Features of React:

- JSX (JavaScript Syntax Extension)
- Virtual DOM.
- One-way data binding.
- Performance.
- Extensions.
- Conditional statements.
- Components.
- Simplicity.

2) React Router:

ReactJS Router is **mainly used for developing Single Page Web Applications**. React Router is used to define multiple routes in the application. When a user types a specific URL into the browser, and if this URL path matches any 'route' inside the router file, the user will be redirected to that particular route.

3) Next.js:

Next.js is a JavaScript framework created by [Zeit](#). It lets you build server-side rendering and static web applications using React. It's a great tool to build your next website. It has many great features and advantages, which can make Nextjs your first option for building your next web application.

Back-end :

1) Ethereum Blockchain:

Ethereum is a **decentralized blockchain platform that establishes a peer-to-peer network that securely executes and verifies application code, called smart contracts**. Smart contracts allow participants to transact with each other without a trusted central authority.

2) Solidity (for creating smart contracts):

Solidity is **an object-oriented programming language created specifically by the Ethereum Network team for constructing and designing smart contracts on Blockchain platforms**. It's used to create smart contracts that implement business logic and generate a chain of transaction records in the blockchain system.

Testing :

Mocha test framework:

Mocha is a **feature-rich JavaScript test framework running on Node. js and in the browser, making asynchronous testing simple and fun**. Mocha tests run serially, allowing for flexible and accurate reporting, while mapping uncaught exceptions to the correct test cases.

Deployment:

Ganache (Local Ethereum Test Network):

Ganache is a **personal blockchain for rapid Ethereum and Corda distributed application development**. You can use

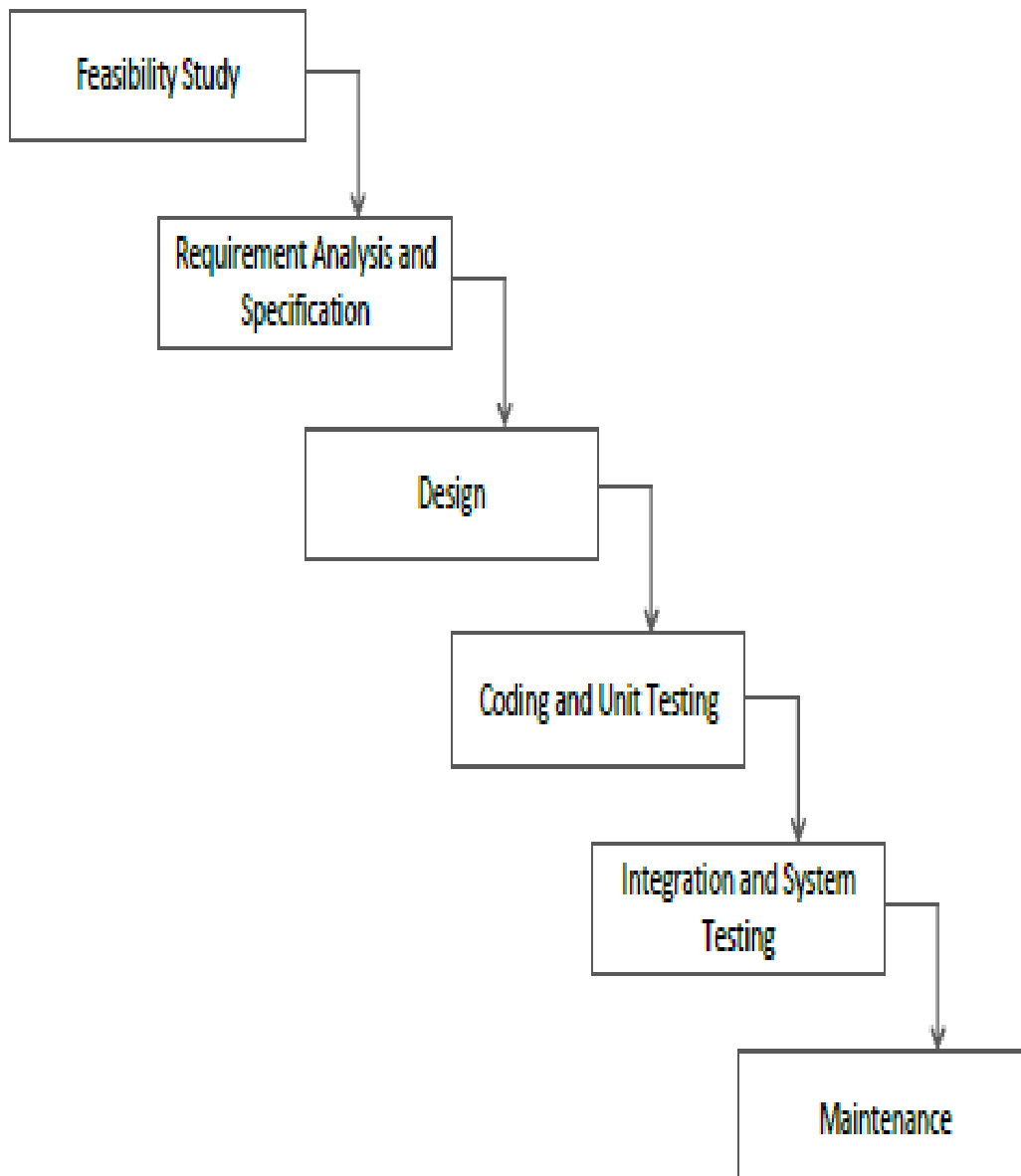
Ganache across the entire development cycle; enabling you to develop, deploy, and test your dApps in a safe and deterministic environment. Ganache comes in two flavors: a UI and CLI.

1.7.2 Hardware Specifications

1. Windows, Linux or Mac OS X
2. NodeJS v8.9.4 or later
3. Truffle also requires that you have running Ethereum client which supports the standard
4. JSON RPC API
5. Web-Browser: Google Chrome, Mozilla Firefox, Opera, Safari, Internet Explorer

CHAPTER 2: DESIGN

2.1 SOFTWARE DEVELOPMENT LIFE CYCLE MODEL

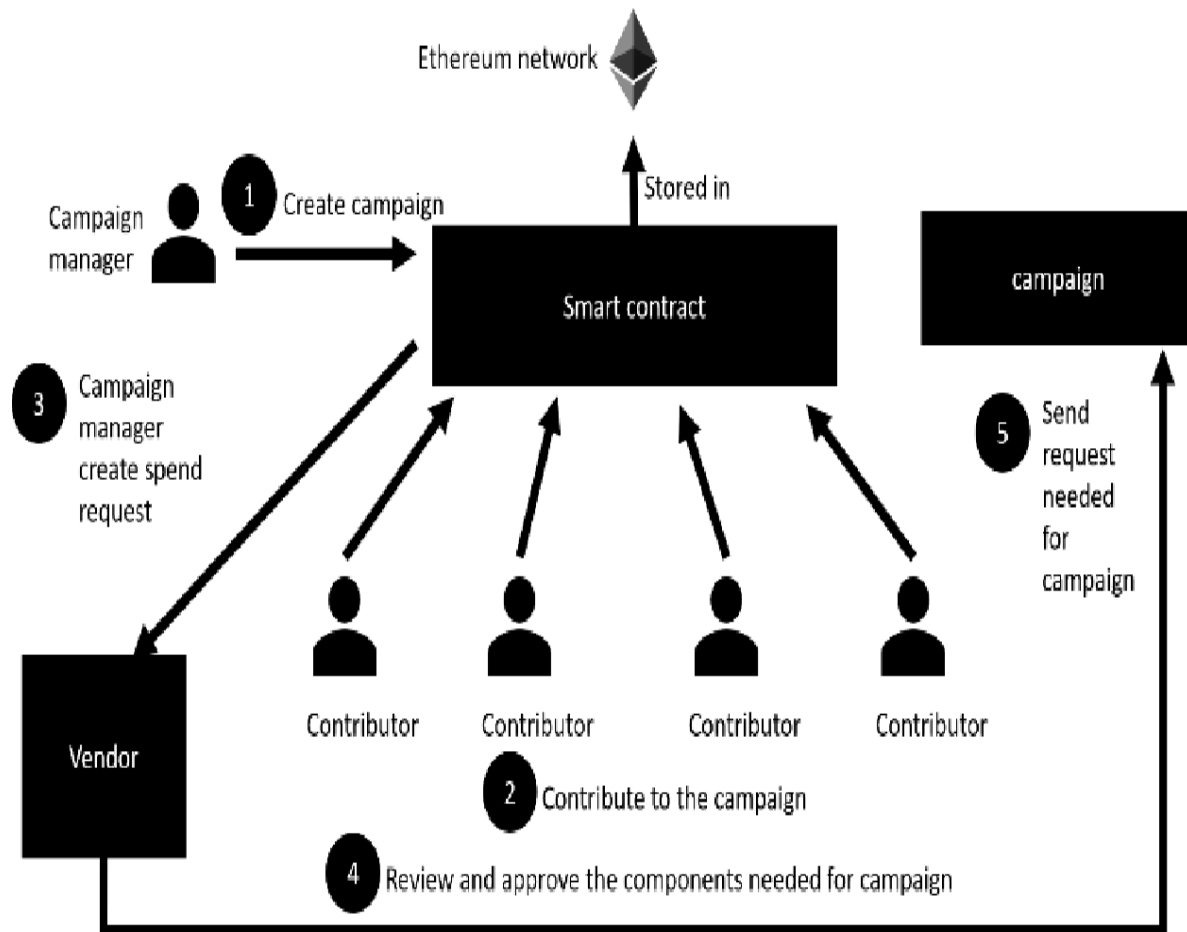


We are using classical waterfall model. As, classical waterfall model is the basic **software development life cycle** model. It is very simple but idealistic. It divides the life cycle into a set of phases. This model considers that one phase can be started after completion of the previous phase. That is the output of one phase will be the input to the next phase. Thus the development process can be considered as a sequential flow in the waterfall. Here the phases do not overlap with each other.

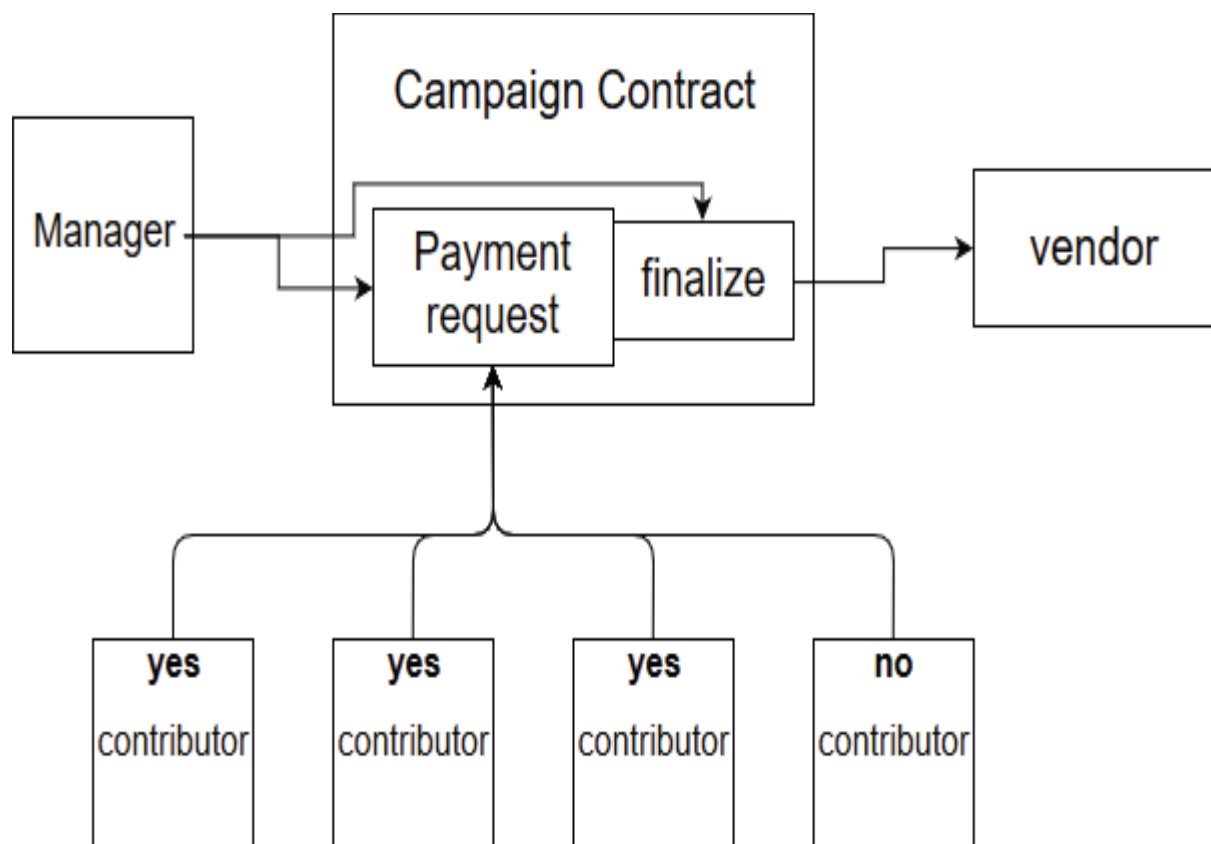
Classical waterfall model is an idealistic model for software development. It is very simple, so it can be considered as the basis for other software development life cycle models. Below are some of the major advantages of this SDLC model:

- This model is very simple and is easy to understand.
- Phases in this model are processed one at a time.
- Each stage in the model is clearly defined.
- This model has very clear and well understood milestones.
- Process, actions and results are very well documented.
- Reinforces good habits: define-before- design, design-before-code.
- This model works well for smaller projects and projects where requirements are well understood.

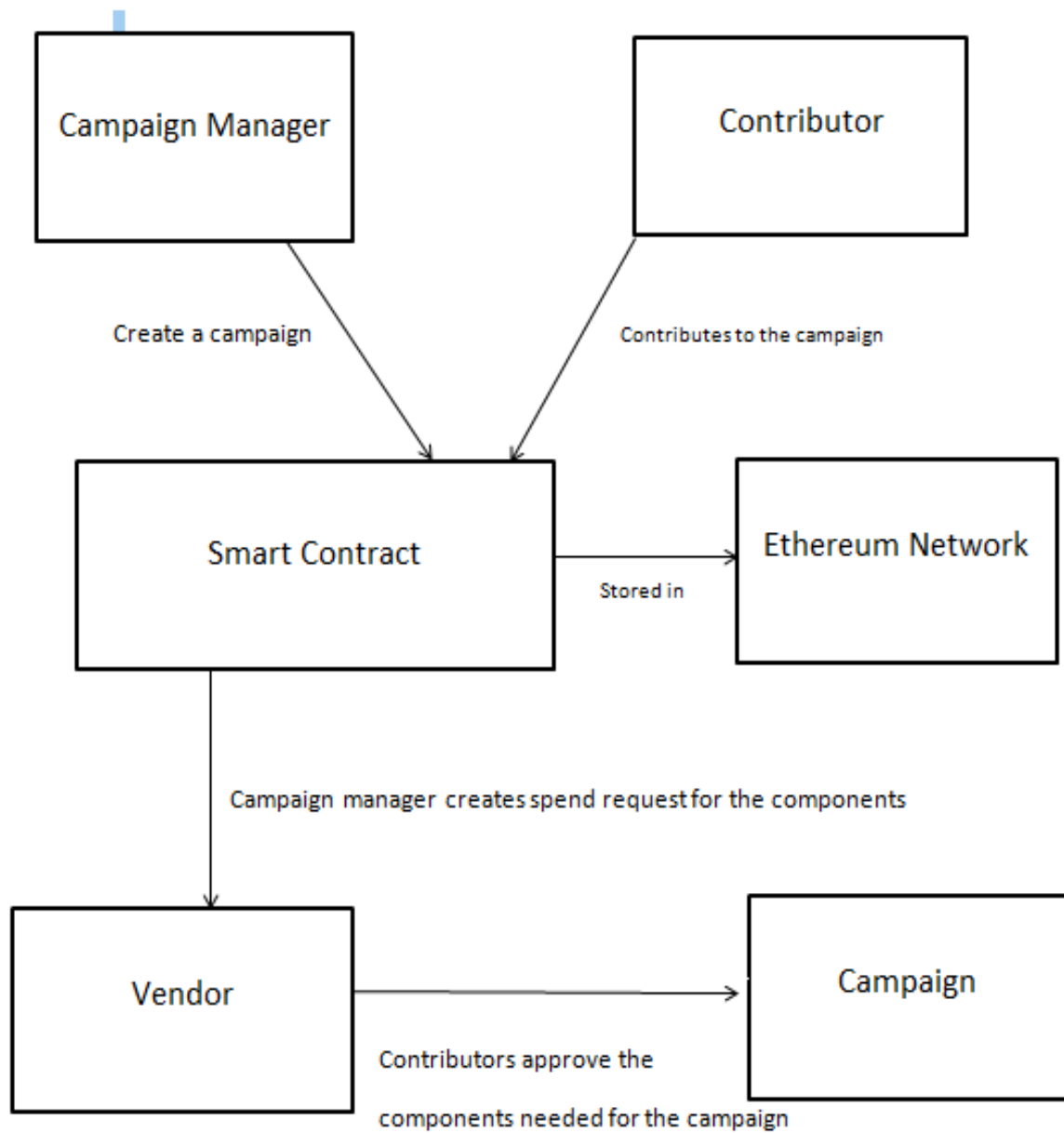
2.2 MODULE DESCRIPTION



2.3 FUNCTIONALITY OF CAMPAIGN CONTRACT



2.4 DATA FLOW DIAGRAM



2.5 USE CASE DIAGRAM

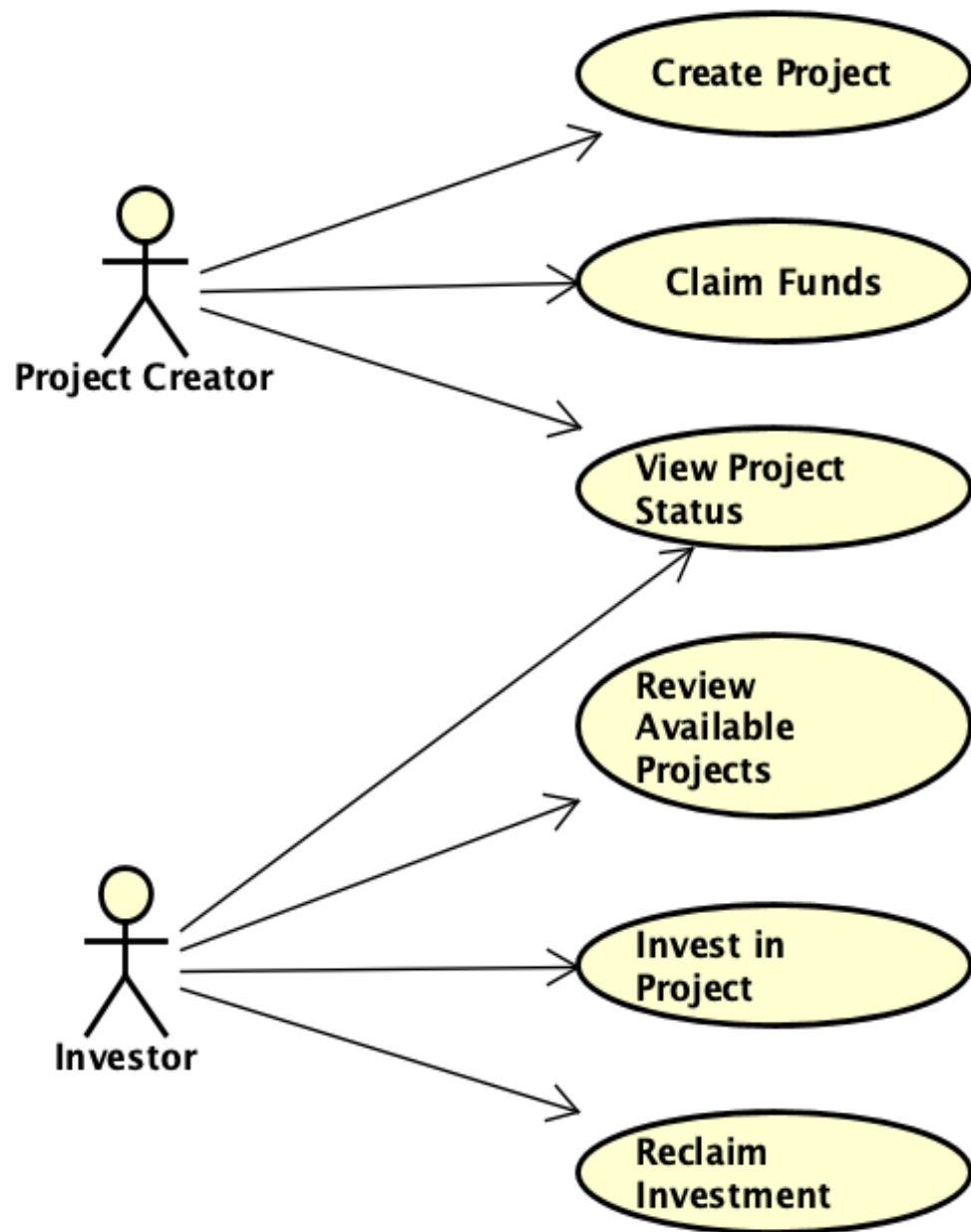
A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

Purpose of Use Case Diagrams

The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

Following are the purposes of a use case diagram given below:

1. It gathers the system's needs.
2. It depicts the external view of the system.
3. It recognizes the internal as well as external factors that influence the system.
4. It represents the interaction between the actors.



2.5 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

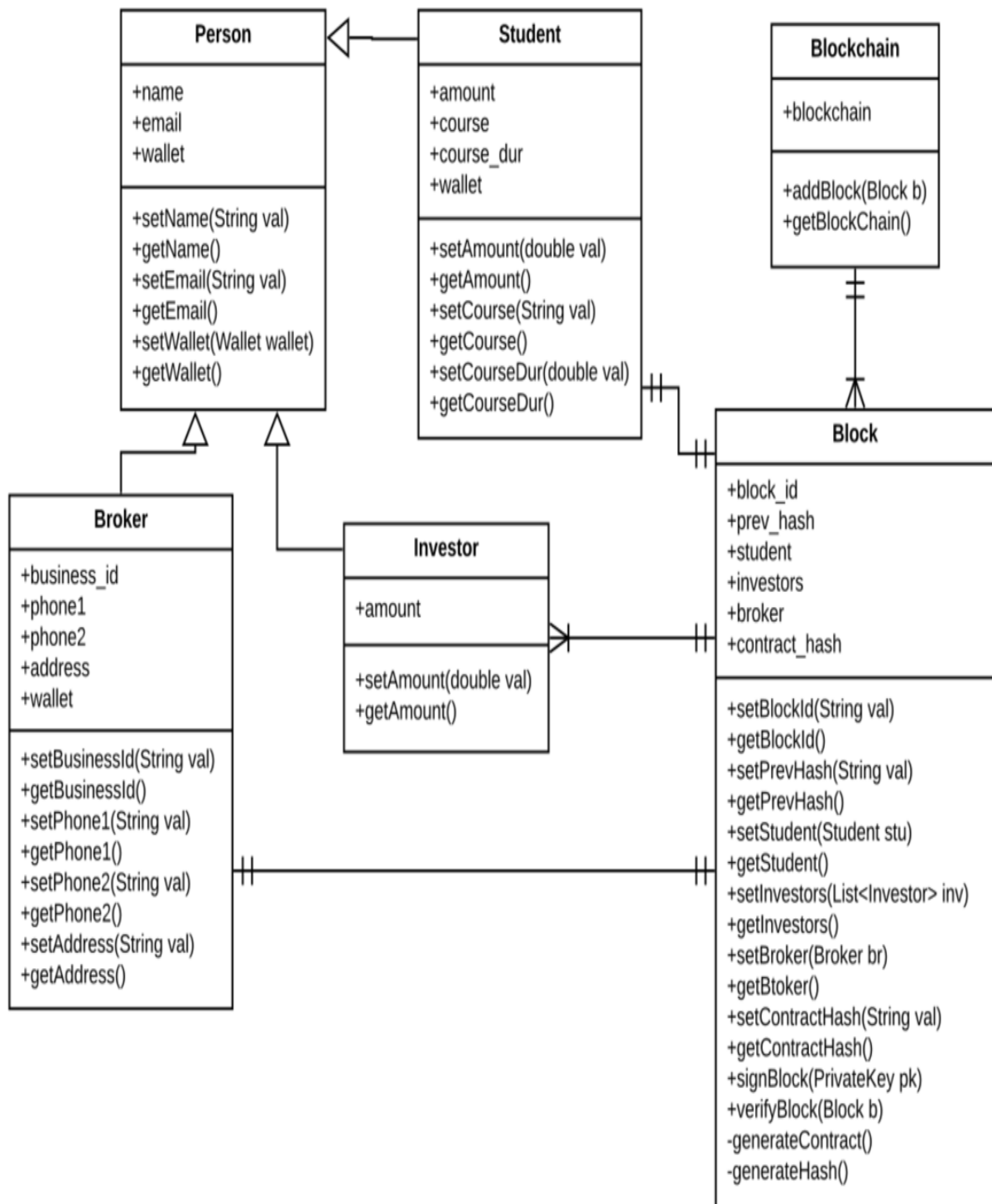
Purpose of Class Diagrams

The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as –

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.



2.5 SEQUENCE DIAGRAM

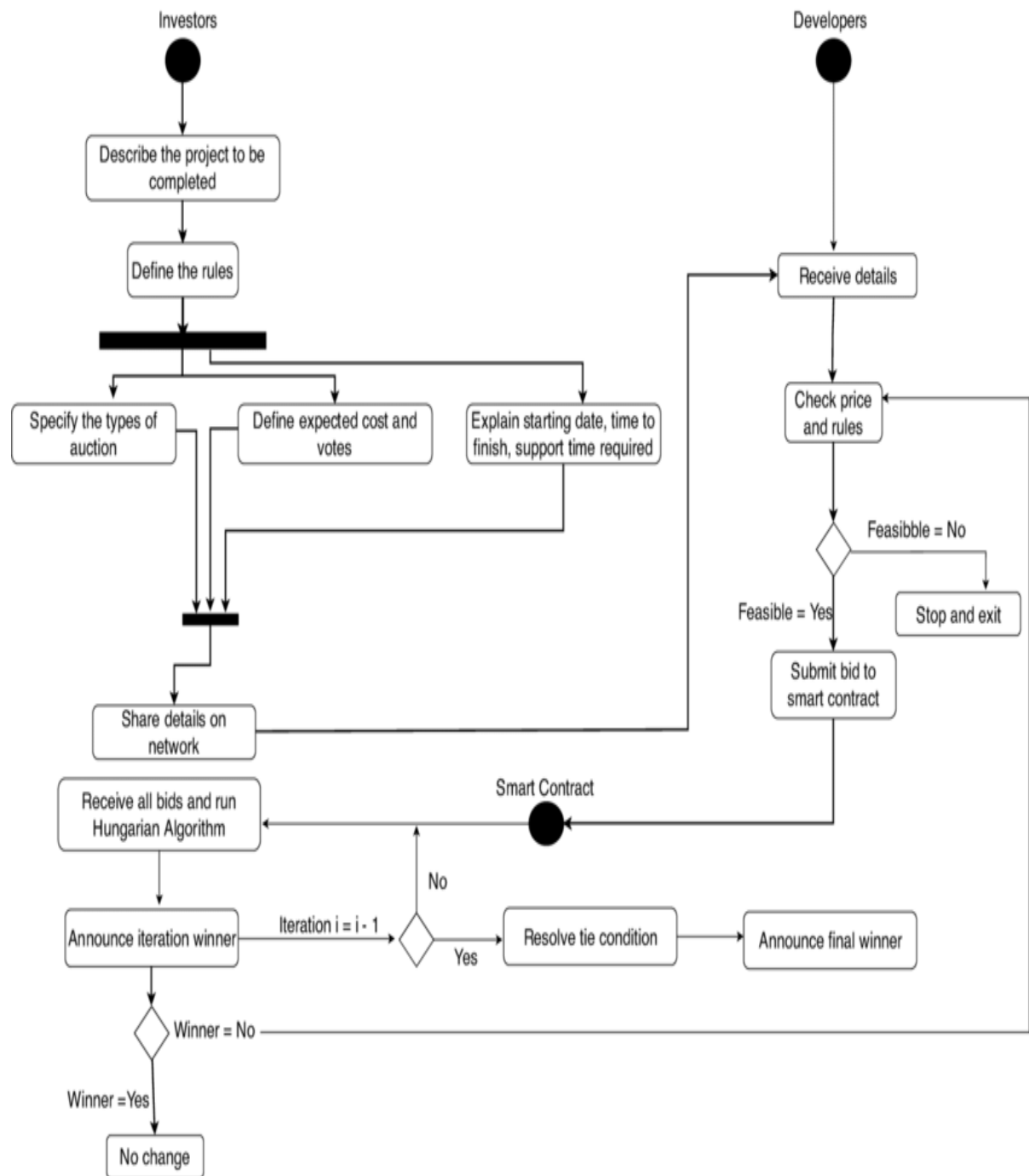
UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

Sequence Diagrams captures:

- the interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)
- high-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams)

Purpose of Sequence Diagram

- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration that realizes a use case
- Model the interaction between objects within a collaboration that realizes an operation
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)



STRUCTURE OF THE CAMPAIGN

Campaign Contract		
Variables		
manager	address	address of the person who is managing this campaign
minimumContribution	uint	Minimum donation required to be considered a contributor or 'approver'
approvers	address[]	List of addresses for every person who has donated money
requests	Request[]	List of requests that the manager has created.
Functions		
Campaign	Constructor function that sets the minimumContribution and the owner	
contribute	Called when someone wants to donate money to the campaign and become an 'approver'	
createRequest	Called by the manager to create a new 'spending request'	
approveRequest	Called by each contributor to approve a spending request	
finalizeRequest	After a request has gotten enough approvals, the manager can call this to get money sent to the vendor	

Campaign Contract

Variables

manager	address	address of the person who is managing this campaign
minimumContribution	uint	Minimum donation required to be considered a contributor or 'approver'
approvers	mapping	List of addresses for every person who has donated money
requests	Request[]	List of requests that the manager has created.

Functions

Campaign	Constructor function that sets the minimumContribution and the owner
contribute	Called when someone wants to donate money to the campaign and become an 'approver'
createRequest	Called by the manager to create a new 'spending request'
approveRequest	Called by each contributor to approve a spending request
finalizeRequest	After a request has gotten enough approvals, the manager can call this to get money sent to the vendor

Request Struct

description	string	Purpose of request
amount	uint	Ether to transfer
recipient	address	Who gets the money
complete	bool	Whether the request is done
approvals	mapping	Track who has voted
approvalCount	uint	Track number of approvals

Chapter 3: Interface Design

3.1 Index Page

CrowdCoin		Campaigns	+
-----------	--	-----------	---

Open Campaigns

0x4fCd33Da70A2F83B88E26192Dcf4B0aB7F7a51EE
[View Campaign](#)

Create Campaign

3.2 Create a new campaign page

CrowdCoin

Campaigns

+

Create a Campaign

Minimum Contribution

wei

Create!

3.2 View details of a campaign

CrowdCoin

Campaigns

+

CampaignShow

0x5d9e15e7C1E1d576CDF72F70b9FdaEea116b8813
Address of Manager
The manager created this campaign and can create requests to withdraw money.

1
Minimum Contribution (wei)
You must contribute at least this much wei to become an approver.

0
Number of Requests
A request tries to withdraw money from the contract. Requests must be approved by approvers.

0
Number of Approvers
Number of people who have already donated to this campaign.

0
Campaign Balance (ether)
The balance is how much money this campaign has left to spend.

View Requests

Amount to Contribute

ether

Contribute!

3.4 Create a new spending request

CrowdCoin		Campaigns	+
-----------	--	-----------	---

[Back](#)

Create a Request

Description

Need Microchips, LED

Value in Ether

0.002

Recipient

[Create!](#)

3.5 View details of the request created

CrowdCoin

Campaigns

+

Request List

Add Requests

ID	Description	Amount	Recipient	Approval Count	Approve	Finalize
----	-------------	--------	-----------	----------------	---------	----------

Found 0 requests.

CHAPTER 4: TESTING

```
const assert = require('assert');
const ganache = require('ganache-cli');
const Web3 = require('web3');
const web3 = new Web3(ganache.provider({gasLimit: 10000000}));

const compiledFactory = require('../ethereum/build/CampaignFactory.json');
const compiledCampaign = require('../ethereum/build/Campaign.json');

let accounts;
let factory;
let campaignAddress;
let campaign;

beforeEach(async () => {
  // get a list of all the accounts
  accounts = await web3.eth.getAccounts();

  // use one of them to deploy the contract
  factory = await new web3.eth.Contract(compiledFactory.abi)
    .deploy({data: compiledFactory.evm.bytecode.object})
    .send({from: accounts[0], gas: '10000000'});

  await factory.methods.createCampaign('11').send({
    from: accounts[0],
    gas: '1000000'
  });

  [campaignAddress] = await factory.methods.getDeployedCampaigns().call();

  campaign = await new web3.eth.Contract(compiledCampaign.abi,
    campaignAddress);
});

describe('Campaigns', () => {
  it('deploys a factory and a campaign', done => {
    assert.ok(factory.options.address);
    assert.ok(campaign.options.address);
    done();
  });

  it('marks caller as the campaign manager', async () => {
    const manager = await campaign.methods.manager().call();
    assert.equal(accounts[0], manager);
  });
});
```

```

    it('allows people to contribute money and marks them as approvers', async ()
=> {
    await campaign.methods.contribute().send({
      value: '12',
      from: accounts[1]
    });
    const isContributor = await
campaign.methods.approvers(accounts[1]).call();
    assert(isContributor);
  });

  it('requires a minimum contribution', async () => {
    try {
      await campaign.methods.contribute().send({
        value: '5',
        from: accounts[1]
      });
      assert(false);
    } catch (error) {
      assert(error);
    }
  });

  it('allows a manager to make a payment request', async () => {
    await campaign.methods.createRequest('Buy batteries', '11',
accounts[1]).send({
      from: accounts[0],
      gas: '1000000'
    });
    const request = await campaign.methods.requests(0).call();
    assert.equal('Buy batteries', request.description);
  });

  it('processes requests', async () => {
    await campaign.methods.contribute().send({
      from: accounts[0],
      value: '100'
    });

    await campaign.methods.createRequest('A', '50', accounts[1]).send({
      from: accounts[0],
      gas: '1000000'
    });

    await campaign.methods.approveRequest(0).send({
      from: accounts[0],
      gas: '1000000'
    });
  });

```

```
    await campaign.methods.finalizeRequest(0).send({
      from: accounts[0],
      gas: '1000000'
    });

    let balance = await web3.eth.getBalance(accounts[1]);
    balance = web3.utils.fromWei(balance, 'ether');
    balance = parseFloat(balance);
    assert(balance > 98);
  });
});
```

CHAPTER 5: CONCLUSION

Finally, the disadvantage faced by the traditional crowdfunding has been overthrown with the help of blockchain technology in crowdfunding. An important factor for the people involved in raising these funds is trust. With the use of the blockchain concept in the crowdfunding, ensures trust of this crowdfunding platform and also reduces the cost of the third party. In future our work can be further enhanced by adding timestamp to the platform i.e., if the funder donates some amount to fundraiser and if the amount doesn't reach the fundraiser within the timestamp specified then the amount is returned back to the funder. With the help of blockchain technology in the crowdfunding platform the people trust the crowdfunding platform and be a part in this crowdfunding network.

Crowdfunding is a method of raising funds from a large number of funders to start a new business or for charitable purposes using the internet. Crowdfunding enables us to easily access a vast network of people through crowdfunding website that brings funder and fundraiser together. In this if the funder wishes to donate for any cause he can donate the money. In present day crowdfunding platform, the money donated by the funder is temporarily stored in fundraising platform so the funder will have no control over the money that they have donated. The third-party involvement incurs additional cost. To overcome this issue many decentralized application have been created for peer-to-peer communication between the funder and fundraiser. The advantages of blockchain have created a way to make peer to peer communication more secure, cost efficient and transparent. Hence by using blockchain in crowdfunding we can change the traditional way of crowdfunding.

This is a website-based system in which the fundraiser and funder have to register in the service provider platform. The fundraiser will raise their funds by describing the need and providing some proof if available. Then the funder can view all the information posted by various fundraiser. If the funder wishes to donate for any particular cause they can proceed with it. Then the money that is donated by the funder will be added to the account of the fundraiser. The sum of the amount that has been donated by all the funders will also be displayed, with the help of this we can identify whether the specified amount has been reached or not. If the specified amount has been reached then the particular post will be automatically removed from the platform. Once they donated the money to the fundraiser the information of the transaction is stored in blockchain and that information can be viewed by both the fundraiser and funder. The funder can track their money until the transacted amount reaches the valid recipient.

REFERENCES & BIBLIOGRAPHY

- [1] Geeks for Geeks (<https://www.geeksforgeeks.org/>)
- [2] Medium ([A week with the blockchain & React | by Dave Hudson | Medium/](#))
- [3] YouTube (<https://www.youtube.com/>)
- [4] Kickstarter ([Kickstarter](#))

CERTIFICATES



Aug 16, 2020

Saumya Shukla

has successfully completed the online, non-credit Specialization

UI / UX Design

In this Specialization, learners summarized and demonstrated all stages of the UI/UX development process, and reviewed current best practices in UX design to create effective and compelling screen-based experiences for websites or apps. Learners completed a series of short exercises, both visual and non-visual, to apply acquired knowledge, from organizing and structuring screen-based content, defining goals and strategy, to creating wireframes and visual mockups. Finally, learners produced a mid-level digital prototype with simulated functionality, as well as a comprehensive plan for a complex website.

The online specialization named in this certificate may draw on material from courses taught on-campus, but the included courses are not equivalent to on-campus courses. Participation in this online specialization does not constitute enrollment at this university. This certificate does not confer a University grade, course credit or degree, and it does not verify the identity of the learner.

Michael Worthington
Faculty, Program in
Graphic Design
School of Art

Roman Jaster
Visiting Faculty,
Program in Graphic
Design
School of Art

Verify this certificate at:
<https://coursera.org/verify/specialization/6U7R6LXTMT5>



COURSE CERTIFICATE

Sep 6, 2020

Saumya Shukla

has successfully completed

Front-End Web Development with React

an online non-credit course authorized by The Hong Kong University of Science and Technology
and offered through Coursera



Jogesh K. Muppala
Associate Professor, Department of Computer Science and Engineering
The Hong Kong University of Science and Technology

Verify at coursera.org/verify/RSTD27AW47J4

Coursera has confirmed the identity of this individual and their
participation in the course.



Certificate no: UC-83319c20-0578-4b26-a810-fda927e28897
Certificate url: ude.my/UC-83319c20-0578-4b26-a810-fda927e28897
Reference Number: 0004

CERTIFICATE OF COMPLETION

Modern React with Redux

Instructors **Stephen Grider**

Saumya Shukla

Date **Sept. 9, 2020**

Length **52.5 total hours**





May 26, 2020

Snehil Gupta

has successfully completed

Blockchain Basics

an online non-credit course authorized by University at Buffalo, The State University of New York and offered through Coursera

Bina Ramamurthy
Teaching Professor
Computer Science and Engineering

**COURSE
CERTIFICATE**



Verify at:

<https://coursera.org/verify/FU4PR4HHNZY3>

Coursera has confirmed the identity of this individual and their participation in the course.



May 27, 2020

Snehil Gupta

has successfully completed

Smart Contracts

an online non-credit course authorized by University at Buffalo, The State University of New York and offered through Coursera

Bina Ramamurthy
Teaching Professor
Computer Science and Engineering

**COURSE
CERTIFICATE**



Verify at:

<https://coursera.org/verify/5S5EG97S2S64>

Coursera has confirmed the identity of this individual and their participation in the course.

PLAGIARISM CHECK

Plagiarism Detector v. 1991 - Originality Report 26-05-2022 15:16:03

Analyzed document: FinalYearReport_1801013074and1801013058.pdf Licensed to: Originality report generated by unregistered Demo version!

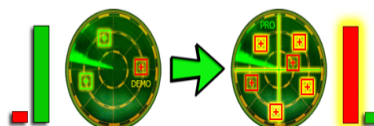
Comparison Preset: Word-to-Word Detected language: En

Check type: Internet Check

[tee_and_enc_string] [tee_and_enc_value]

Warning: Demo Version - reports are incomplete!

Detect more Plagiarism with Licensed Plagiarism Detector.



Order your **Lifetime License** packed with features:

1. Complete resources processing - with more results!
2. Side-by-side compare with detailed analysis!
3. Faster processing speed, deeper detection!
4. Advanced statistics, Originality Reports management!
5. Many other cool functions and options!

Get your 5% discount:



Detailed document body analysis:

Relation chart:

