

# Investor Protector



Anuhya Gankidi , Jayanth R.Sheri , Anujot Singh, Xialu Zou  
Team 10

# Problem facing

FundRaising sites do not have effective measures to limit startup managers. They are unable to monitor the flow of money, resulting in many fund raising programs becoming scams.

We want to create a platform where the probability of investors losing money is drastically decreased, the uncertainty of the project is reduced, and the chances of fraud are mitigated.

# Project Objectives

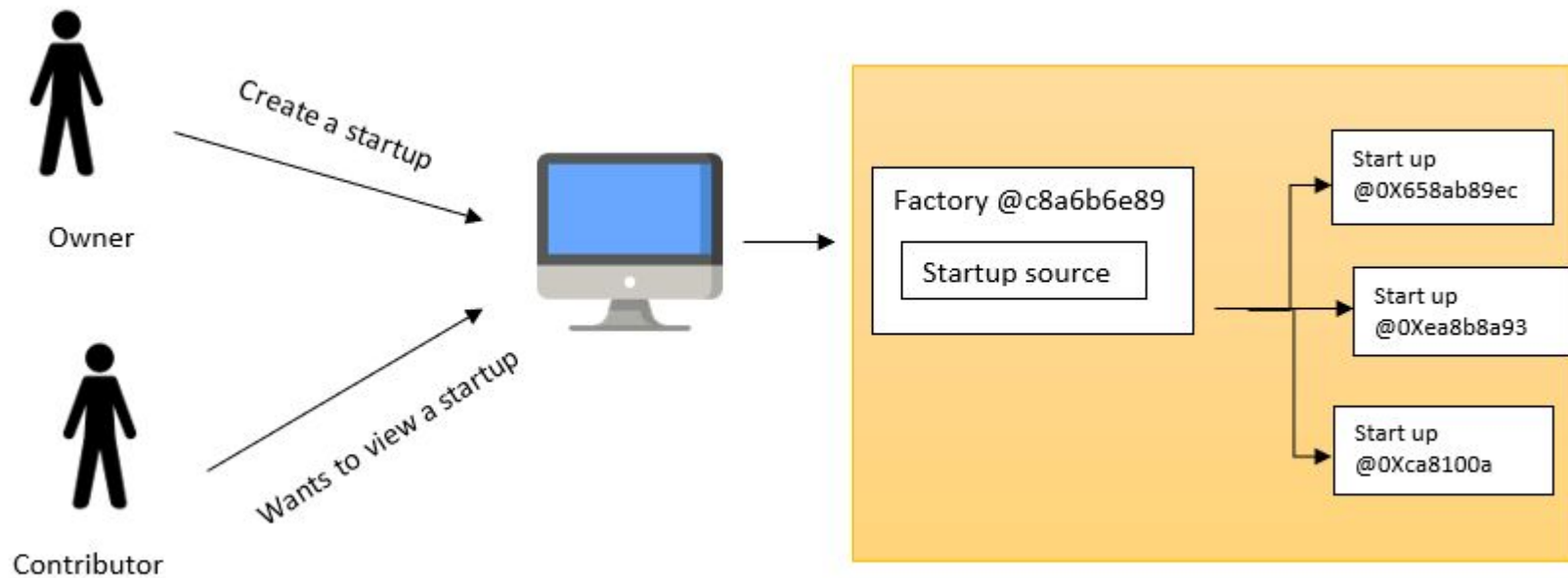
- Start-up manager be able to create a fundraising event.
- A secure portal will be in place for investors to contribute.
- Start-up manager will be able to create a request for funds.
- Investors will be able to approve request for funds.
- We are able to verify the account where the transactions are leading to this helps confirms where the money is actually going.
- know how much progress has been made on the project.

# Techniques and Tools used

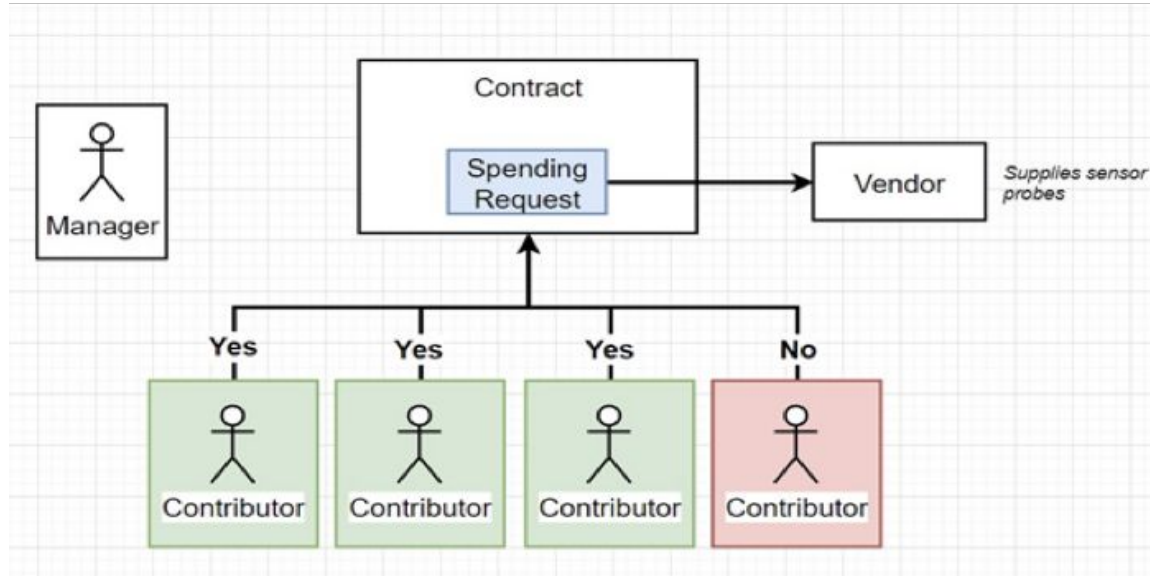


- Ethereum
- Smart Contracts written in Solidity
- Node.Js
- React
- Mocha and Remix
- Rinkeby as a test network.
- Metamask as a ethereum wallet account manager.
- Web3.js

# System Architecture



# Introducing Smart Contract



# Fundraising Contract

Variables		
manager	address	Address of the person who is managing the start up.
minimumContribution	uint	Minimum donation required to be considered a contributor or “approver”
approvers	mapping	List of address for every person who has donated money
requests	Requests[]	List of requests that the manager has created

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

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

Open Startups

 Create Startup

0xf7eeeAa62B1e3b33F87f2521C4F7BeDEF345b76C

[View Startup >](#)

0xb93eBAa2323F7EeBDEe58E8eFdc2a4D0564afB6

[View Startup >](#)



## Create a Startup

Minimum contribution

wei

Create!

Amount to Contribute

ether

Contribute!

## Startup Info

**0x3afcFFd873918bdE8Eb12ffE  
E9c5b602F814aA0D**

Address of Manager

The manager created this Startup and can create requests to withdraw money

**10000**

Minimum Contribution (wei)

You must contribute at least this much wei to become an approver

**1**

Number of Requests

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

**2**

Number of Contributors

Number of people who have already donated to this Startup

**0.00109**

Startup Balance (ether)

The balance is how much money this Startup has left to spend.

View Requests

## Requests

[Add Request](#)

ID	Description	Amount	Recipient	Approval Count	Approve	Finalize
0	Material x	0.00001	0x0E65Da8ED13367eBF38386204e45811cDC3d0A3b	2/2		
1	Material Y	0.000055	0x3afcFFd873918bdE8Eb12ffEE9c5b602F814aA0D	0/2	<a href="#">Approve</a>	<a href="#">Finalize</a>

Found 2 requests.



Rinkeby Test Network



## Networks

○ Not connected

Account

0x3af...aA0D1

Show/hide test networks

Dismiss



Ethereum Mainnet



Ropsten Test Network



Kovan Test Network



Rinkeby Test Network



Goerli Test Network

Assets

Activity



Localhost 8545



Create Request

Nov 30 · localhost

Add Network



Finalize Request

Nov 30 · localhost:3000

-0 ETH

-0 ETH

**DEMO**