INDIAN INSTITUTE OF TECHNOLOGY KANPUR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROJECT SPECIFICATION DOCUMENT
2018-2019

EtherFeeds

SUBMITTED BY

Kuldeep Kushwaha	160349
Abhishek Valsan	17111002
Appu B	18111008
Lt Cdr Karan Basson	18111030
Lt Cdr Rahul Raj	18111053

INSTRUCTOR

Prof. Sandeep Sukhla

Abstract

Employee happiness is the best investment for any business in the long run. A crucial part of this is a feedback mechanism wherein they can honestly open up and such regular communication creates the energizing, healthy work culture that has been found to be the cornerstone of employee engagement.

In most of the scenarios, the feedback mechanism is not only cumbersome but also prone to bias, forging, censorship for illicit intent by corrupt managers or top officials. Such cases employees feel insecure voicing their honest opinions on how things are moving.

By harnessing the power of blockchain, a DApp can solve this issue by creating a transparent, immutable feedback communication channel is made available thus solving the aforementioned problems in traditional systems.

Introuction

Feedback and effective organizational communication is the key factor which is linked with overall employee satisfaction, employee retention, healthier business performance, etc. Problems associated with traditional systems are prone to bias, forging and censorship for illicit intent which can lead to attrition of high potential employees.

A blockchain based solution as proposed here can be an effective way which can ensure, transparency, immutability at the same time ensure privacy and security. High-performers always understand that there is constant room for improving work efficiency and they thrive when given the opportunity to be heard and valued without fear and reluctance. Therefore, we feel that this a problem that needs to be addressed.

We propose to develop a decentralized application of ethereum network using solidity smart contract with a web interface using Django framework.

System Architecture

Diagrammatic representation

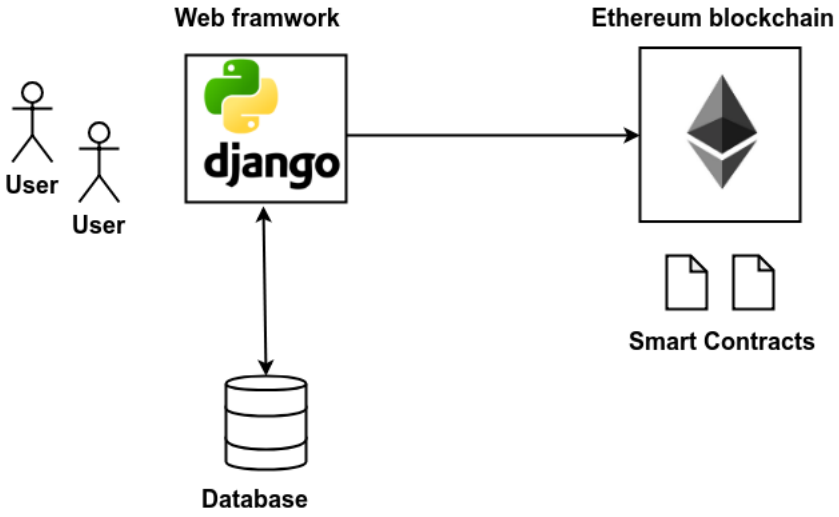

Figure 1: System Architecture

Users
Various user groups are Admin, Managers and Employees.
Blockchain
We will be using ethereum network with solidity smart contracts.
Database
For demo purpose, we will be using database consisting of employees and managers.
Nodes
Test nodes will be created on local machines and will be stored on a shared database.

Feature List

Following are the broad features of the application: -

Serial No.	**Features**	**What they are for**
1	Create/Remove User	Adding/Removing Employee or Managers by admin
2	Propose Questions	For preparing a questionnaire for getting feedback
3	Upvote	To submit positive feedback to the question.
4	Downvote	To submit negetive feedback to the question.
5	ViewResult	To view the final outcome of the questions by anyone.
6	SetTimer	To set expiry for submission of feedback to the questions
7	Distribute Bounty	To incentivize those who are participating in feedback
8	Create/Edit Dept	To create/ remove departments in the organization by admin
9	Add Employee to Departmen	For tagging each employee with a department. Will be done by Manager
10	Authorized Feedback	For authorizing department wise feedback. Only authorized personnel of that particular department will be able to participate in the feedback.

Features to be implemented are in priority order as above. Last three features are planned to be implemented post finalising and testing the rest of the features.

Testing

Level 1 test
Test account addresses in JS virtual machine will be stored in local database as different users categorized as employees and managers and their departments. Will interact with smart contract through web interface using these users to test all the first priority features.
Level 2 test
We will deploy our smart contracts on Ropsten network and check the same features for actual metamask ropsten accounts.

Testing

- Week 1. For the development of basic web interface, DB, smart contract and its integration.
- Week 2. For adding first priority features to the web interface and testing
- Week 3. For adding advanced features and testing.