DAPP for IITM Project Staffs And Interns

IITM Blockchain Vertical - Application Development

CS23E002 Delina Soubam
AE20B035 Mohammed Anshad K

May 29, 2024

1 Introduction

In the ever-evolving landscape of digital innovations, the emergence of blockchain technology has sparked a revolution, offering a decentralized framework that transcends the limitations of traditional centralized systems. At the forefront of this revolution are Decentralized Applications (known as DApps). These innovative software solutions integrate blockchain's secure ledger and smart contracts, revolutionizing trust, security, and autonomy in digital transactions. At the heart of this innovation are smart contracts, self-executing contracts with predefined rules encoded on the blockchain. These smart contracts enable automated and trust-neutral execution of agreements, facilitating seamless interactions between users while ensuring transparency and security. By leveraging distributed networks, DApps eliminate the need for intermediaries, enabling direct peer-to-peer interactions and granting users unprecedented control over their data and assets.

In this project, we have developed a Decentralized Application (DApp) tailored for managing project staff under ICSR (The Centre for Industrial Consultancy and Sponsored Research) at IIT Madras. This application is implemented using the Hedera testnet and aims to serve the needs of external project staff and interns affiliated with the institute. By harnessing the power of smart contracts and the decentralized nature of blockchain technology, our DApp offers a transparent, secure, and efficient platform for handling various administrative tasks related to project management within the ICSR ecosystem.

2 Problem Statement

The ICSR recognizes the critical importance of efficient project management for successfully executing research and development initiatives. However, the existing management procedures suffer from various inefficiencies, resulting in delays, opacity in approval status tracking, challenges in salary management, etc. These issues not only hinder the morale and productivity of project staff but also impact project progress.

The following includes some of the main efficiencies of the current existing model of governance:-

1. **Delays in Procedure**: The current project management system often experiences delays in various procedural aspects. These delays lead to prolonged decision-making cycles, causing setbacks in project timelines and impeding timely project execution.

2. **Approval Status Opacity**: Transparency in tracking the approval status of project proposals is lacking within the institute's project management framework. Project staff often face challenges in monitoring the progress of their proposals, resulting in uncertainty and frustration regarding the status of their projects.

3. **Salary Management Challenges**: Managing salary payments for project staff is a complex process prone to inefficiencies and errors. Manual salary allocation and payment processes are time-consuming, error-prone, and often subject to delays, leading to dissatisfaction among project staff and administrative overhead for the institute.

3 Related Work

1. **University of Cagliari's dApp for Managing Workers Contacts**:

   An application of the BOSE and ABCDE development methodology to build a dApp system for managing real-world contracts for temporary workers so that, by design, agreements, commitments, and rules are respected for the specific domain and employment sector and so that employers and employees are safeguarded by design.

   Ethereum Solidity Smart Contracts are designed to manage all the steps and to keep track of all Commitments and Agreements of Employers and Employees, and job history.

2. **MIT's OpenCourseWare**:

   Recognized for providing free online access to course materials from more than 2,500 courses, has embraced blockchain-powered storage solutions. This approach is focused on cataloging and preserving educational content, ensuring its long-term availability and integrity.

3. **University of Utah's National Science Data Fabric (NSDF) and National Data Platform (NDP)**:

   The initiative was to improve global and equitable access to large scientific datasets, including multiple petabytes of NASA climate data.

   By leveraging blockchain to enable interactive processing and exploration of massive datasets without extensive local resources, the university is overcoming traditional barriers to data access

4 Scope

Through this project, our goal is to develop a decentralized application (dApp) leveraging blockchain technology to target inefficiencies inherent in centralized models. Our primary objectives revolve around enhancing transparency and addressing critical development initiatives. We are primarily focused on managing issues related to under-processed approvals and delayed salary payments. Additionally, we aim to ensure that project information is readily accessible to all users of the blockchain network.

While our project's main focus is on these specific inefficiencies, we acknowledge that there may be other areas for improvement, which we will consider for future enhancements, albeit beyond the current scope.

5 Contribution/Novelty

Beyond its decentralized nature and blockchain foundation, this DApp fills a critical need within our institute by revolutionizing staff management practices. Moreover, its user-friendly interface and interoperability ensure accessibility and scalability, catering to the institute's evolving needs and future growth. Though not a part of the current project, as a future scope one can integrate advanced technologies like AI analytics alongside smart contracts, offering the DApp streamlines operations, enhances communication, and fosters a collaborative work environment.

Ultimately, this DApp represents a transformative solution that empowers staff, promotes efficiency, and drives organizational excellence.

6 Overview of Implemented Module

We designed the application with mainly two actor typologies:- the human actors and the system components.

- **The Human actors**:
  1. Project Staffs:- The applicant for project. Maybe categorized into different positions based on academic levels. Enters the details for the project proposal. If the proposal is approved, they work for a contract period within which they are allotted a monthly salary. The contract period may be extended based on the project's requirements.
  2. Faculties:- They review the project proposal and provide acknowledgment. They have access to the data of those staff that work under them. Each staff member works under at least two faculties.
  3. ICSR:- It connects an external entity with the institute. It would act as the admin who deploys the contract and makes payments to staff if the end date is reached.
- **The System Components**:
  1. **The Web Platform**: The simple web interface that allows users to easily access key functionalities such as project submission, approval tracking, salary management, and communication tools.
  2. **The blockchain infrastructure**: It records the smart contracts and transactions giving room to make direct payments while granting security and privacy.

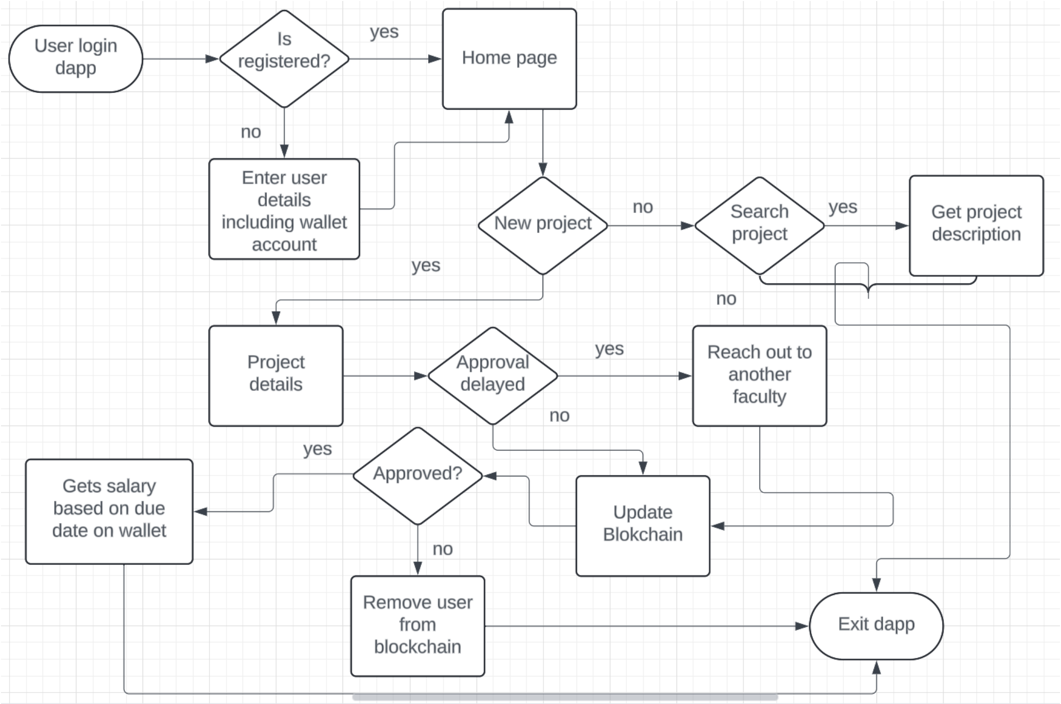


Figure 1: Activity diagram for a project staff

6.1 Model Architecture:

In light of the current challenges, by leveraging the concept of blockchain, our dApp aims to offer the features such as:

1. User Registration Module:
   - Functionality: It allows users to register by providing personal information and associating a wallet address.
   - Input: User's name, email address, contact information, wallet address, category user belong (i.e. research associates, assistants, post-docs, etc)
   - Output: Confirmation of successful registration.
2. Security Identification Module:
   - Functionality: It implements secure authentication mechanisms to verify user identities.
   - Input: User hashpack wallet, authentication tokens.
   - Output: Access granted or denied based on authentication.
3. Project Description and Assignment Module:

- Functionality: It allows project staff to submit project proposal requests and assign project details.
- Input: Project name, members, contract period, resource allocation, department coordinators - the faculties.
- Output: Confirmation of project submission, assignment of project status.

4. Time-Based Escalation Module:
   - Functionality: It detects delays in project approvals and escalates overdue projects.
   - Input: Project approval timestamps, predefined deadlines.
   - Output: Automatic escalation of overdue projects to another designated faculty.
5. Salary Allocation Module:
   - Functionality: It automates salary allocation based on staff categories and predefined schedules.
   - Input: Staff categories, salary amounts, predefined payment schedules.
   - Output: Automatic release of salary payments according to schedule.
6. Accessible Data Module:
   - Functionality: Stores project details on the blockchain for access by anyone on the blockchain.
   - Input: Project descriptions, outcomes, participant information.
   - Output: Publicly accessible project data for querying.

7 Implementation Details

The application is designed using solidity smart contract as both back-end and data storage, and React JS to build the front-end.

7.1 Smart Contract:

Details such as IITM faculty, project staff, and projects are stored in contract states using structured arrays with address mapping to reduce loops.

1. IITM Faculty:
   - `registerFaculty(name, department, email)`: For registering, the faculty has to authenticate with IITM LDAP first. Then, the faculty's wallet address, name, department, and email are stored in the contract.
   - `loginFaculty()`: Verifies the caller's address is stored in the contracts faculties state.

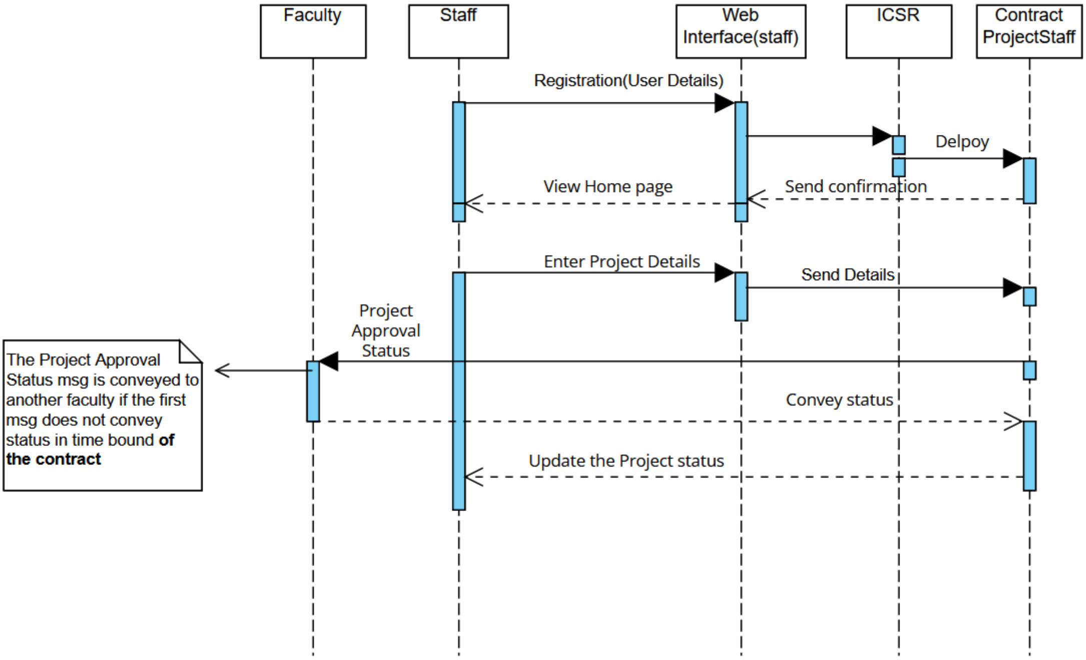

Faculty
Staff
Web Interface(staff)
ICSR
Contract ProjectStaff
Registration(User Details)
Delpoy
View Home page
Send confirmation
Enter Project Details
Send Details
Project Approval Status
The Project Approval Status msg is conveyed to another faculty if the first msg does not convey status in time bound of the contract
Convey status
Update the Project status

Figure 2: Simple Sequence diagram for Project Proposal

- `approveProject(projectId, approval)`: The faculty sends the project ID that is intended to be approved or rejected. After verifying that the caller is part of the project faculty, the project status is updated accordingly. The faculty can approve new projects as well as extend the ongoing projects.

2. Project Staff:

- `registerProjectStaff(name, email)`: For registering, the project staff must first authenticate with IITM LDAP. Then, the faculty's wallet address, name, and email are stored in the contract.
- `loginFaculty()`: Verifies the caller's address is stored in the contracts project staff state.
- `addProject(project_details)`: After validating the caller address with project staff and verifying arguments, the project structure is added to the project's state. Also, the project ID is included in the project staff and faculty's state. The new project is pending status, which the faculty has to approve later.
- `getExtendedPeriod(projectId)`: This function changes the end date of the argument project and changes the status to extend. The faculty has to approve this change.

3. ICSR as admin:

    - `constructor()`: The ICSR deploys the contract. The constructor will set its address as admin and initialize the data structures for the states.
    - `makePayments()`: The admin could automate this process on a daily basis. It will check the end dates of the projects and make salary transfers to project staff in the specified amount from the ICSR account. After the transfer, the project status is set to complete.

7.2 Front End

The front end is developed using react JS along with Hashpack wallet. Hashpack wallet is associated with the ESDSA Hedera account. The front-end functionality is built on the known deployed smart contract.

- The home page contains buttons to connect the wallet and login for each specified user.
- For executing the contract with any specified function, the contract ID should be valid, and the user should be associated with a Hashpack wallet account. Each transaction is signed by the user with a Hashpack wallet.
- For registering faculty or project staff and adding a new project, the forms are given on the respective user page.
- The functions in the contract executed with transaction specified with gas signed by the user wallet. The return values are obtained from transaction receipts.
- The returned values are used in the react states to display or modify the page accordingly.

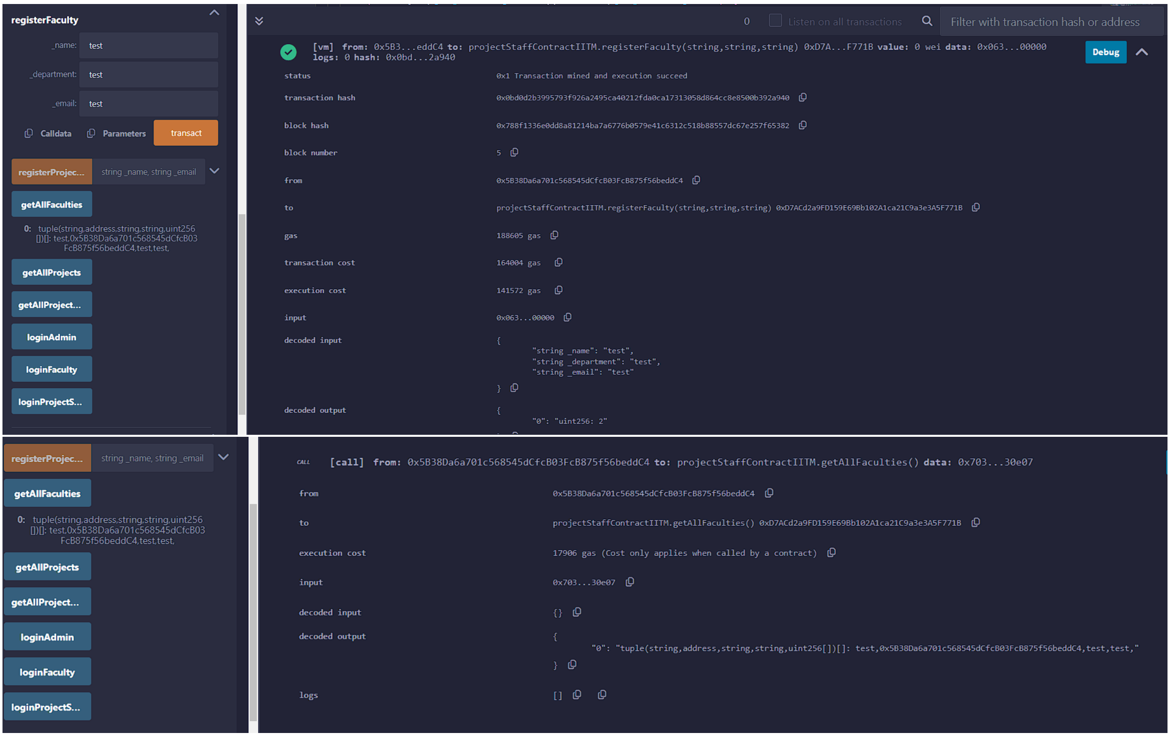
8 Experimental Results

8.1 Transaction cost of functions

The gas used to execute a function depends on the bytes of memory used, such as reading, writing, or updating, and the amount of processing, like loops.

- To avoid loops to iterate through arrays, using mapping is found to be quite efficient. The address-to-index mapping is used throughout the contract. Also, each user element has the project IDs associated with that user.
- For preventing memory wastage, identifying the max range values used in the state variables is important. The solidity allows for the specification of the byte length of an element.
- Returning large arrays in a function costs more gas. Returning only relevant data to the user with a loop is more efficient than returning the large array. This includes only elements that are associated with the caller and essential properties of each element to the caller.

Since a user has to execute multiple functions, compensating for the processing cost of some functions back to the verified user from the admin is a viable solution. This will also discourage unnecessary executions from the user.



```
_name: test
_department: test
_email: test

registerProjec...  string _name, string _email

getAllFaculties
0: tuple(string,address,string,string,uint256[])[]: test,0x5B38Da6a701c568545dCfcB03FcB875f56beddC4,test,test,
```

```
[vm]  from: 0x5B3...eddC4 to: projectStaffContractIITM.registerFaculty(string,string,string) 0xD7A...F771B value: 0 wei data: 0x063...00000
logs: 0 hash: 0x0bd...2a940
status              0x1 Transaction mined and execution succeed
transaction hash    0x0bd0d2b3995793f926a2495ca40212fda0ca17313058d864cc8e8500b392a940
block hash          0x788f1336e0dd8a81214ba7a6776b0579e41c6312c518b88557dc67e257f65382
block number        5
from                0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                  projectStaffContractIITM.registerFaculty(string,string,string) 0xD7ACd2a9FD159E69Bb102A1ca21C9a3e3A5F771B
gas                 188605 gas
transaction cost    164004 gas
execution cost      141572 gas
input               0x063...00000
decoded input       {
                        "string _name": "test",
                        "string _department": "test",
                        "string _email": "test"
                    }
decoded output      {
                        "0": "uint256: 2"
```

```
CALL  [call]  from: 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4 to: projectStaffContractIITM.getAllFaculties() data: 0x703...30e07
from                0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                  projectStaffContractIITM.getAllFaculties() 0xD7ACd2a9FD159E69Bb102A1ca21C9a3e3A5F771B
execution cost      17906 gas (Cost only applies when called by a contract)
input               0x703...30e07
decoded input       {}
decoded output      {
                        "0": "tuple(string,address,string,string,uint256[])[]: test,0x5B38Da6a701c568545dCfcB03FcB875f56beddC4,test,test,"
                    }
logs                []
```

Figure 3: Simple implementation of ProjectStaffContractIITM contract in Remix IDE

8.2 Development Process

The application's development includes VS code IDE. Various functionalities are introduced using NPM packages. Used GitHub for source control.

- Programming Languages and frameworks: Solidity, JavaScript (ES6), React.js, hashgraph SDK, hashconnect
- Used Remix Web IDE for developing the smart contract. Which also able to test and debug the solidity contract.
- The challenging part of the development was executing smart contract functions using the Hashpack wallet. Executing transactions with Hashpack wallet is throwing processing errors.
- We could not find a solution for scheduling a salary transfer to the project staff exactly on the date. This issue can be overcome with admin executing `makePayments()`

function daily or weekly, which will transfer the amount for the projects with the end date that has passed.

- Currently, any faculty associated with a project can approve it. An automated email service to notify the faculty about the pending projects will decrease the delay in approval. The development of this feature is currently not in the scope of the dApp.
- When registering, the user has to enter an LDAP username and password to verify. Which is a Web 2 login. So, Web 2 to Web 3 login should be implemented in ICSR, which associates the Hedera account the Institute account.

9 Conclusion

In conclusion, our DApp project offers a solution within the institute to streamline administrative processes, enhance communication, and improve overall efficiency. By leveraging blockchain technology and smart contracts, we aim to revolutionize the way staff members are managed, ensuring transparency, accountability, and security throughout the entire process.

Through the implementation of this DApp, we envision a future where one could include administrative tasks such as staff onboarding, scheduling, leave management, and performance evaluation in a seamlessly integrated decentralized ecosystem. We believe that our DApp will serve as a model for institutes seeking to embrace blockchain technology for administrative innovation and excellence.