The Hong Kong Polytechnic University

DEPARTMENT OF COMPUTING

COMP5567 – DISTRIBUTED ALGORITHMS AND PROTOCOLS FOR BLOCKCHAINS

EXPERIMENTAL REPORT

Group Project

Student Name	Student ID
LIU Qingyuan	24052432G
Hui Zifan	24046598G
Li Dongwei	24117496G

Lecturer in charge:

Dr. Shan JIANG

Submission Date : 07/12/2024

Blockchain-Based Facial Recognition Degree Authentication System

December 7, 2024

Abstract

This report talks about how we make a system for checking if a degree is real using blockchain and face checking. We use Ethereum and something like DeepFace. We explain what it looks like, what it does, how we built it, and what we will do next.

1 Introduction

1.1 Why We Need This Thing

Checking if a degree is real is super important for schools, bosses, and students. It makes sure everything is fair, no cheating, and hiring is easier. But, the old way of checking degrees has lots of problems:

- **Too Many Fake Degrees:** Making a fake degree is too easy. This makes real degrees not worth much. Schools and bosses have a hard time trusting degrees. Bad for everyone. [1]
- **Checking Degrees is Too Slow:** The old way is slow, uses lots of paper, and many people in between. Checking a degree from another country is very hard and slow. This slows down hiring.
- **Degree Data is Not Safe:** All the degree data is in one place, easy to steal. If someone steals the data, many people have big problems, maybe even lose their good name. Keeping data in one place is bad.

We want to fix these problems using blockchain and face checking. Blockchain is like a big book everyone can see, but no one can change. Face checking makes sure the right person has the degree.

1.2 What We Want and How We Will Do It

We want to make checking degrees better. We will do this:

1. **Use Smart Contracts on the Ethereum Blockchain:** We will use smart contracts on the Ethereum blockchain to keep and manage all the degree data. [2] A smart contract is like a small computer program that does things automatically. Ethereum is good and safe. [3] Nobody can change the data on the blockchain. No more fake degrees!

2. **Use Face Checking with the Degree:** We will use face checking, maybe DeepFace, to connect a person's face to their degree. This makes extra sure that only the right person can access and show their degree. [4] It will be more safe. We take a picture of your face and check it with the picture saved with the degree.

3. **No One Person Controls It, Everyone Can See:** In our system, no single person or place controls the data. Schools and bosses can check degrees directly. No middle man! Everyone can see the computer code, too. Much more fair.

4. **Keep Data Safe, Keep Secrets:** We will keep the data very safe. We use secret codes. We keep face pictures very safe. We use very clever ways to keep secrets, maybe something called Zero-Knowledge Proofs. [5] We follow all the rules for keeping data secret.

We want to make degree checking safe, fast, and easy for everyone to use, all over the world. This will make education better for everyone.

2 System Parts and How it Works

2.1 System Parts

2.1.1 Blockchain Part

We use Ethereum blockchain. It is decentralized and no one can change information on it. We keep degree information safe there. Ethereum has smart contracts. We use Solidity to write them. Smart contracts do things automatically with degrees.

- Development Tools: We use Hardhat. Hardhat helps us make and test smart contracts. Hardhat makes a fake Ethereum for testing. It also helps put smart contracts on a test blockchain.

- Smart Contracts: Smart contracts do important things. They save degree info, check users, and have voting.

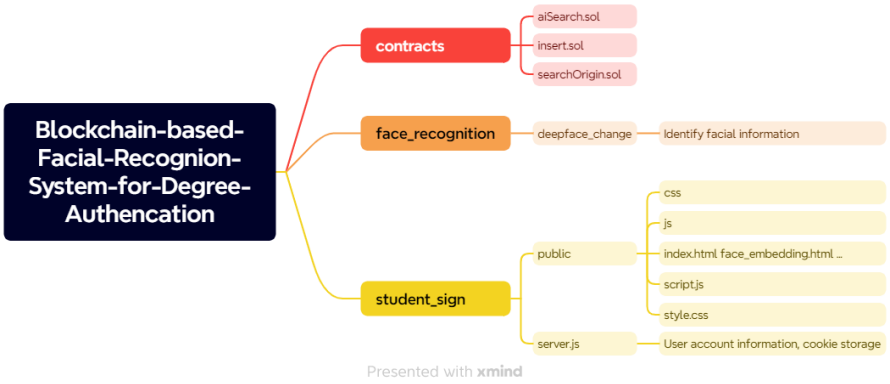

Figure 1: System Structure

2.1.2 Face Recognition Part

We use face recognition to connect face to degree.

- Technology: We are using DeepFace. DeepFace is good at making and checking face data.
- How it Works: User uploads picture. DeepFace makes a special code from face. This code is saved with degree on blockchain. We use it later to check.

2.1.3 Data Storage

Degree info and link to face code is on Ethereum. Face code not directly on blockchain. We save a special ID (faceEmbeddingUUID) on blockchain. This ID tells us where the real face data is. It is kept safe off-chain (maybe IPFS). [6]

- Data Organization: Degree info is organized inside smart contract like this:

```
struct Degree {
string faceEmbeddingUUID; // ID for face data
string degreeType; // Bachelor, Master, etc.
string major; // What they studied
string university; // School name
uint256 graduationYear; // Year of graduation
}
mapping(bytes32 => Degree) private degreeRecords; // Degree
 ↪  info saved with unique hash. Makes finding easy.
```

2.1.4 Frontend and Backend

The system has easy to use screens and a strong backend. They work with blockchain and face recognition.

- Frontend: User interface made with HTML, CSS, JavaScript. User can enter info, upload photo, see results.
- Backend: Backend uses Node.js and Python Flask. It helps connect to blockchain and face recognition.
  - Contracts Module: Contains the Solidity smart contracts: aiSearch.sol, insert.sol, and searchOrigin.sol. These contracts manage the core logic of the system, including user authentication, data storage, and facial recognition integration.
  - Facial Recognition Module: Employs the DeepFace library (deepface_change) to process user-uploaded photos, generate facial embeddings, and compare them to verify identity. The facial embeddings are used to securely link a user's identity to their academic credentials.
  - Student Sign-in Module: This module, comprised of front-end and back-end components, provides the interface for students to sign in, upload their information, and initiate the authentication process. The frontend is built with HTML, CSS and JavaScript, while the backend uses Node.js and Python Flask to handle API requests and communicate with the other modules.

* Frontend Components: Includes various HTML files (e.g. index.html, face_embedding.html), CSS styling, JavaScript logic (script.js), and manages user interface elements and user interaction with the backend and blockchain.
* Backend Components: The backend is made up of a Node.js server (server.js) and supporting backend logic. It manages user accounts, handles cookie storage, processes facial recognition requests, and serves as a bridge between the frontend and the blockchain module.

2.2 How It Works Step-by-Step

2.2.1 Saving Degree Info

Schools or other allowed people put degree info on blockchain.

- How It Works: They type in student info, degree type, school, and send face data. This info goes into smart contract. No changing after this.
- Checking is Important: Before saving, other people need to check info. We use voting for this.

2.2.2 User Checking a Degree

Users who want to check a degree upload their photo.

- How It Works:
  1. DeepFace makes face code from user's photo.
  2. This new face code compared to face code saved with degree on blockchain.
  3. If face codes match, system shows degree.
- Result: System says "valid" or "no valid". Depends if face match.

2.2.3 Voting to Make Sure Info is Good

We use voting so info is correct.

- How It Works:
  1. New degree needs "valid" from three voters (maybe from schools).
  2. Voters check degree info and vote yes or no.
  3. Degree only saved on blockchain after enough "valid" votes.
- Why Voting Good: Stops bad people putting fake degrees. Makes system more trusted.

2.2.4 Smart Contracts

Three main smart contracts work together: aiSearch.sol, insert.sol, and searchOrigin.sol.

- aiSearch.sol: This contract handles the face recognition part. It helps compare face embeddings to verify user identity.

- insert.sol: This contract manages adding new degree information to the blockchain. It includes the voting logic to ensure data accuracy.
- searchOrigin.sol: This contract helps users search for and retrieve degree information from the blockchain.

2.2.5 Student Sign-in Process

1. User goes to website (index.html).
2. User uploads photo (face_embedding.html).
3. Frontend sends photo to backend (server.js).
4. Backend uses DeepFace library (deepface_change) to make face code.
5. Backend sends face code and degree info to smart contracts.
6. Smart contracts check and save info, using voting to make sure it's valid.
7. Frontend gets result from backend and shows it to user.

2.2.6 Backend details

- server.js: this handles user accounts, saves cookies for logins, talks to DeepFace, and connects everything.
- Backend also stores extra user information off-chain, because not all data goes on blockchain.

2.2.7 Frontend Details

- index.html, other HTML files: these are what user sees.
- JavaScript code (script.js): This code makes website work, talks to backend.
- CSS code (style.css): This is styling for how website looks.

3 System Design

3.1 How Data is Kept

In this system, degree info and a special code for face (faceEmbeddingUUID) are kept on Ethereum blockchain. This makes sure no one can change the information. Ethereum is decentralized, so no one person controls it. This make it very safe.

- Degree Info Storage: Degree info, like student ID, degree type, school name, date, and face code, are all on blockchain. After we save degree, nobody can change it. Everyone can see it, so very open. Good for checking if degree is real.
- Face Data Storage: We make special face code (faceEmbeddingUUID) with DeepFace. This code is also on the blockchain. This code represents person's face. Very important for checking who they are.

Good Things:

- Open: Everyone can see all the data on blockchain. Everything can be checked.
- Not Centralized: No single person or place controls the data. If one computer breaks, other computers have copies. More safe.
- Cannot Change: After data is on blockchain, nobody can change it or delete it. Degrees stay safe.

Not So Good Things:

- Privacy: Saving face data on blockchain might not be good for privacy. Even though it is a code, it can still be connected to a person's face. Maybe we can use encryption or just save part of face code.
- Too Big Data: Ethereum is safe but maybe too expensive for lots of users. Saving many big face data files on blockchain might cost too much. Might be slow too.

3.2 How Smart Contracts Work

Smart contracts are very important. They keep data, find data, and check data. They are on Ethereum and do everything automatically.

- Degree Stuff:
  - Saving: Smart contract keeps degree info. Student ID, degree type, school, date, face data. All safe and no one can change.
  - Finding: Allowed people (schools, companies, users) can search for degrees. They can check if a degree is real.
  - Checking: Smart contract checks if degree is real. User gives info, contract compares to saved info.
- Voting Stuff:
  - Valid Process: Each new degree needs valid from three people. These people are from schools or trusted. They check the degree info.
  - No One Boss: Voting makes sure no one person controls degree info. Stops fake degrees. Makes info good. Degree goes on blockchain only after enough votes.
  - Everyone Can See: Voting is open. All votes are on blockchain. No cheating.

3.3 Face Recognition Part

DeepFace is used for face recognition. Very important for checking user and connecting face to degree.

- How it Works:
  - Make Face Code: User gives photo. DeepFace makes special face code (embedding). This code is like a map of user's face.

  - Compare Face Code: System compares new face code to code on blockchain. If same, user is valid. Can see degree.
  - Save on Blockchain: Face code is saved on blockchain with degree. Makes sure face and degree are together. Cannot change.

- Good Things:
  - Safe Checking: Face recognition makes it very secure. Only real person can see degree.
  - Face and Degree Together: Face and degree are connected. No fake ID. Degree check is real.

3.4 All Parts Together

System has many parts. They work together for checking degrees. Three main parts: blockchain, frontend, and backend.

- Blockchain Part:
  - Main Job: Keeps and manages degree data. Also does the voting.
  - How it Talks: Talks to smart contracts. Makes sure all degree stuff is safe and cannot be changed. Answers when someone wants to see degree.

- Frontend:
  - User Screen: Frontend is what user sees. Made with HTML, CSS, and JavaScript.
  - What it Does: User uploads photo, sees degree, sees if check is okay. Easy to use.
  - Talks to Backend: Frontend talks to backend to start face check and get data from blockchain.

- Backend:
  - API: Backend is like a middle man between frontend, blockchain, and face system. Uses Node.js and Python Flask.
  - Talks to Blockchain: Talks to Ethereum and smart contracts. Sends requests to save, find, check degree.
  - Face Stuff: Backend uses DeepFace. Makes face codes from photos. Saves and compares face codes with blockchain.

All these parts are important. They make the system open, safe, and easy to grow. Can add more features later.

4 Current Features Demonstration

4.1 Saving Degree Info

Example code:

```
Degree(
"faceEmbeddingUUID12345", // Special face code (on blockchain)
"Bachelor of Science", // Type of degree
"Computer Science", // What they studied
"Example University", // School name
2024 // Year finished
);
```

This code no show how Degree struct is made. You need line like this first:

```
struct Degree {
string faceEmbeddingUUID;
string degreeType;
string major;
string university;
uint256 graduationYear;
}
```

What it do: First code makes a new Degree and gives it values. This new Degree can be used with other functions or added to degreeRecords or proposals.

4.2 Checking User Degree

- Send Photo and Make Face Code: User sends photo. System makes face code from photo. We also make a face code UUID and connect it to the face code. Real face data mostly not on blockchain.
- Check on Blockchain: Backend code calls smart contract functions like isDegreeValid, getDegree. These functions use faceEmbeddingUUID to find degree on blockchain.
- Compare (Not on Blockchain): System compares new face code from user photo with face data from database (using the faceEmbeddingUUID). This means we need a database or other system to keep all the face data. Backend gets face data using faceEmbeddingUUID and uses an API to check how similar they are. Blockchain just makes sure info is correct.

4.3 Voting

Voting is in DegreeSearch contract. When new degree is suggested (using proposeDegree function), it goes into proposals. Users can call approveProposal function to add to approvalCount. When approvalCount is high enough (like 3), degree is valid. It moves from proposals to degreeRecords. In practice, the voting mechanism can be designed according to the situation, and our main goal is to ensure the authenticity of educational qualifications.

5 Make System Better Later

5.1 Keep Data Safe and Correct (Plan)

Keeping data safe and correct is super important. We want a system where basic degree information (like what kind of degree, major, university, and graduation year) is stored on the

blockchain so everyone can see it, and no one can change it. But we didn't have time to fully finish this part yet. Our plan is to put a special code (a hash) of this data on the blockchain to make sure it's correct. We won't put actual face pictures on the blockchain because it's not safe and costs too much. Instead, we plan to put the secret, coded face data on a special internet storage place like IPFS, and only put the special code (hash) on the blockchain to check if it's the right face. We're also looking at other storage places, like Arweave, to make it even safer and last a long time. We will use secret codes for all the data, all the time.

5.2 Easy to Update the System Later (Plan)

The system needs to be easy to change because blockchain technology changes very fast. We didn't have enough time to fully set this up yet, but we want to build the system in a special way that makes it easy to change parts of it later. [7] This way we can make the system better, add new features, without changing the old data. For example, we might add better face checking technology or a new way to store data. We're also thinking about letting people vote on how to change the system, but we didn't have time to add that yet.

5.3 Keep User Secrets Safe (Plan)

Keeping user information secret is very important. We haven't finished building all the privacy features yet, but our plan is to use strong secret codes for all the important data, including the face data, before putting it on the internet storage. We want to use a very clever secret way, called Zero-Knowledge Proofs (ZKPs), so a user can prove they have a degree without showing any other private information. This means they can prove they have the degree without showing their name or birthday or other private things. We also want to give users control over who can see their information, but we didn't have time to add that feature yet.

5.4 Follow the Law Everywhere (Plan)

Our system needs to follow data privacy laws like GDPR [8] and CCPA [9] if we want people all over the world to use it. We didn't have time to add all of these legal features yet. Our plan is to use special internet names (called DIDs) to give users more control over their online identity, so they can choose what information they show to others. [10] This will make it easier to follow different laws around the world and will make users trust the system more. We want to connect our system to other online identity systems later, but we didn't have time to do that in this version of the system. We will also create clear rules about how we use data, so everyone understands.

6 Expected Deliverables

6.1 System Demonstrations

- Screenshots of degree upload and verification interfaces:

Figure 2: Screenshot of the Login.

Figure 3: Screenshot of the Registration.

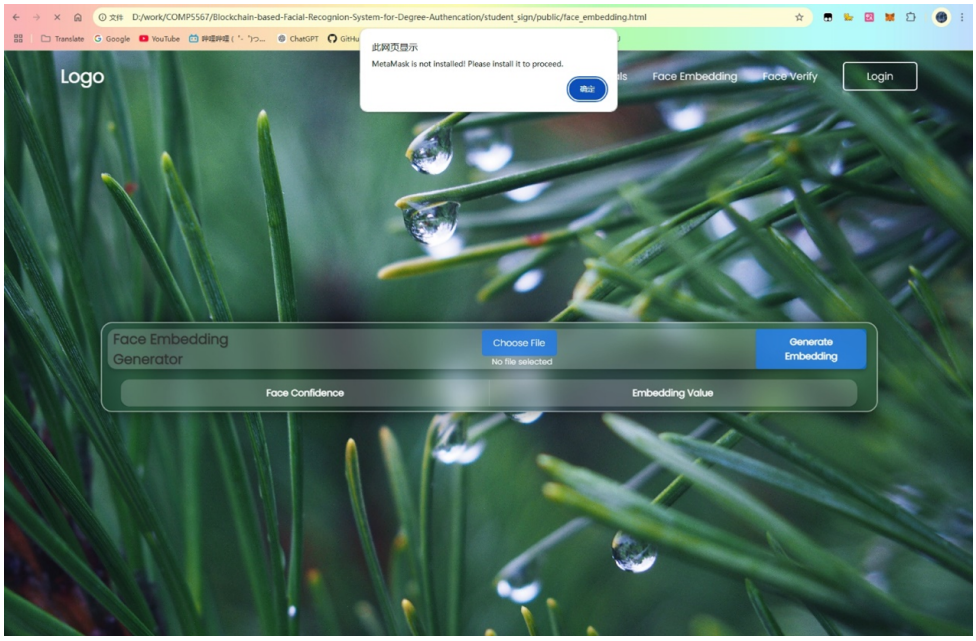

Figure 4: Screenshot of the Face_embedding.

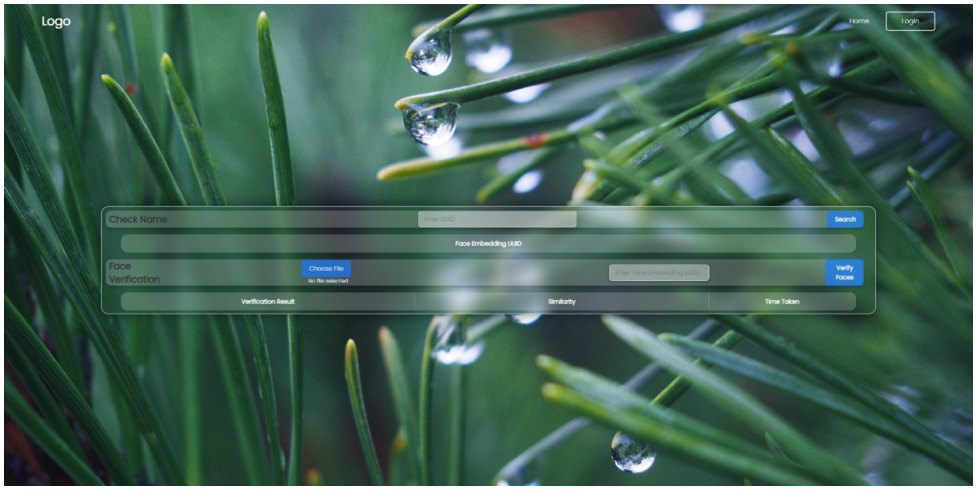

Figure 5: Screenshot of the Face_verification.

- A video showcasing the authentication process and blockchain interaction. The video can be accessed at: https://github.com/PolyUBlockChainTeam/Blockchain-based-Facial-Recogn blob/main/imgs/Steps.mp4

6.2 Technical Deliverables

- Access to an open-source GitHub repository. The repository can be accessed at: https://github.com/PolyUBlockChainTeam/Blockchain-based-Facial-Recogn git

7 Team Contributions

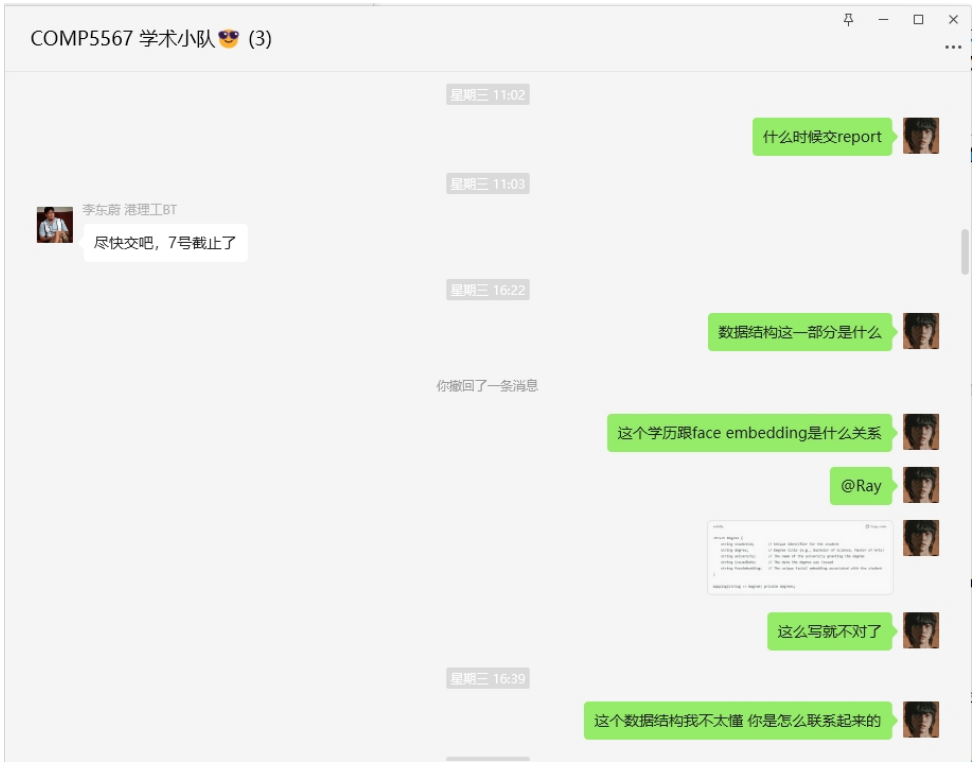

Figure 6: Screenshot of the Wechat1.

- Hui Zifan (24046598g): Hui was assigned to work on the slides and smart contracts, but failed to contribute anything to this report. His coding practices were also highly problematic. Specifically, his use of the variable ”name” for the student’s name in the consensus code clashed directly with Li Dongwei’s use of the same variable name (”name”) for the faceEmbedding and UUID. This created significant confusion and made the codebase difficult to understand and work with, especially when I (Liu Qingyuan) was trying to write the report. As a programmer, such a basic naming conflict is unacceptable. I repeatedly requested Hui to fix this, but he ignored my messages and did not rectify the issue. I was forced to spend extra time correcting his mistakes. Furthermore, Hui showed a lack of commitment to teamwork. He left our very first project meeting—where we had agreed to stay after school to discuss topics—immediately after classes ended, without contributing to the discussion.

Figure 7: Screenshot of the Wechat2.

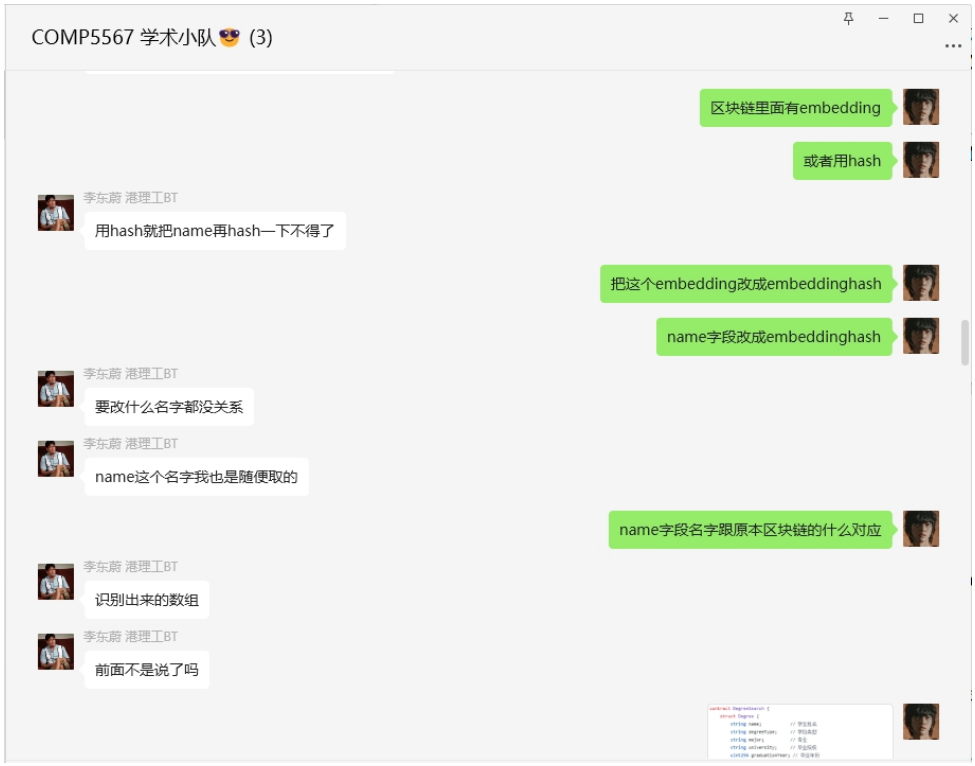

Figure 8: Screenshot of the Wechat3.

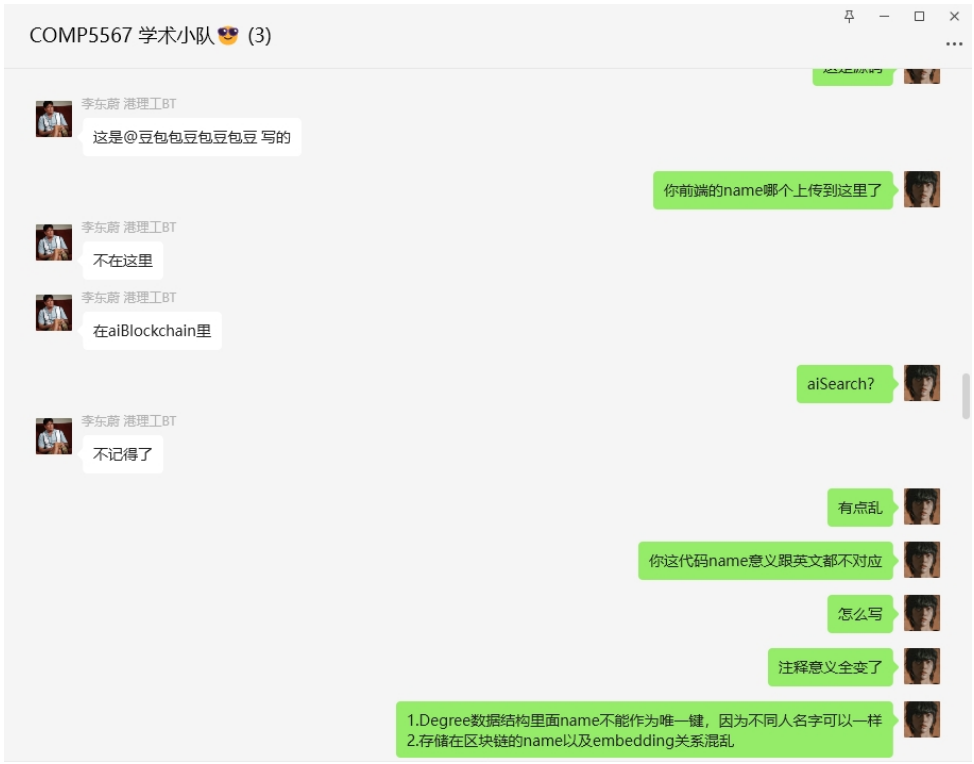

Figure 9: Screenshot of the Wechat4.

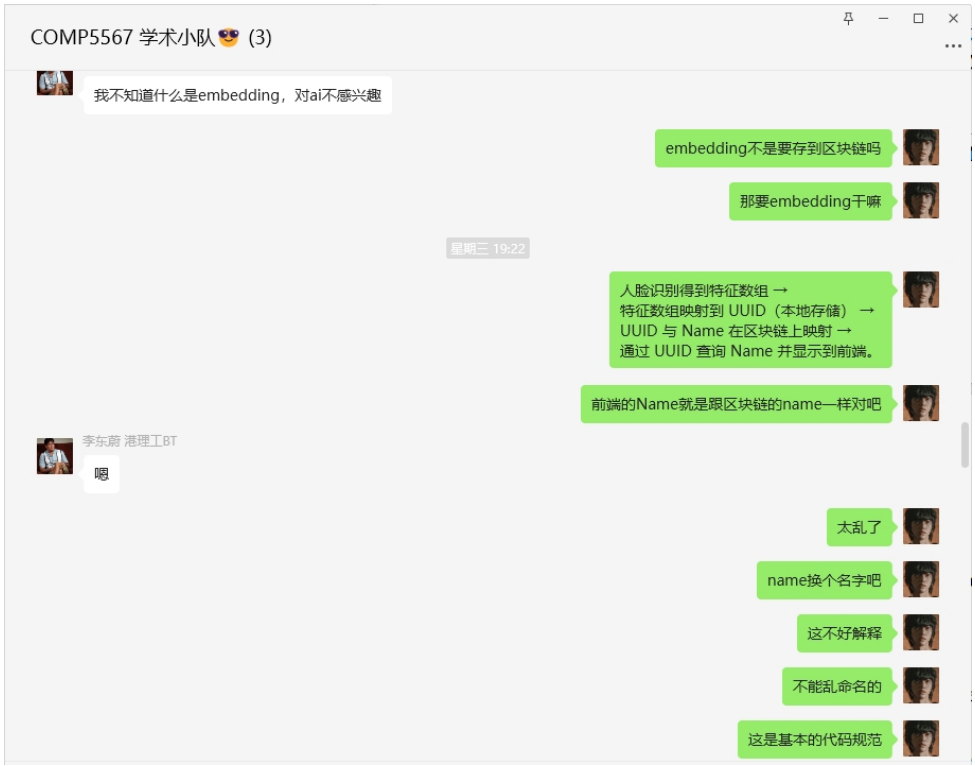

Figure 10: Screenshot of the Wechat5.

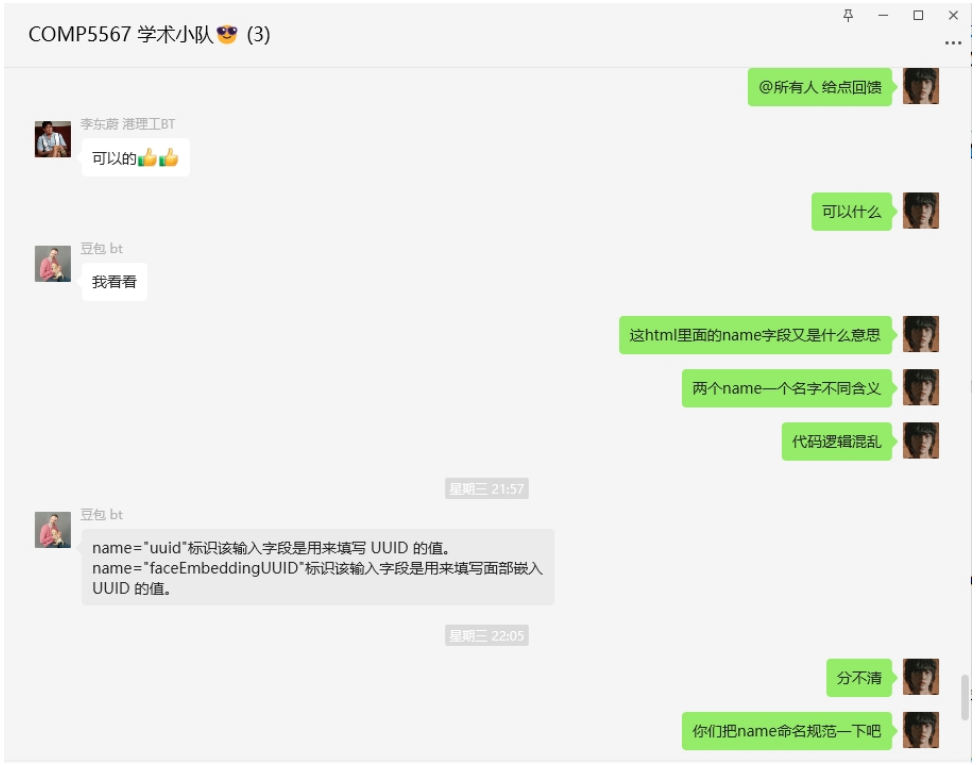

Figure 11: Screenshot of the Wechat6.

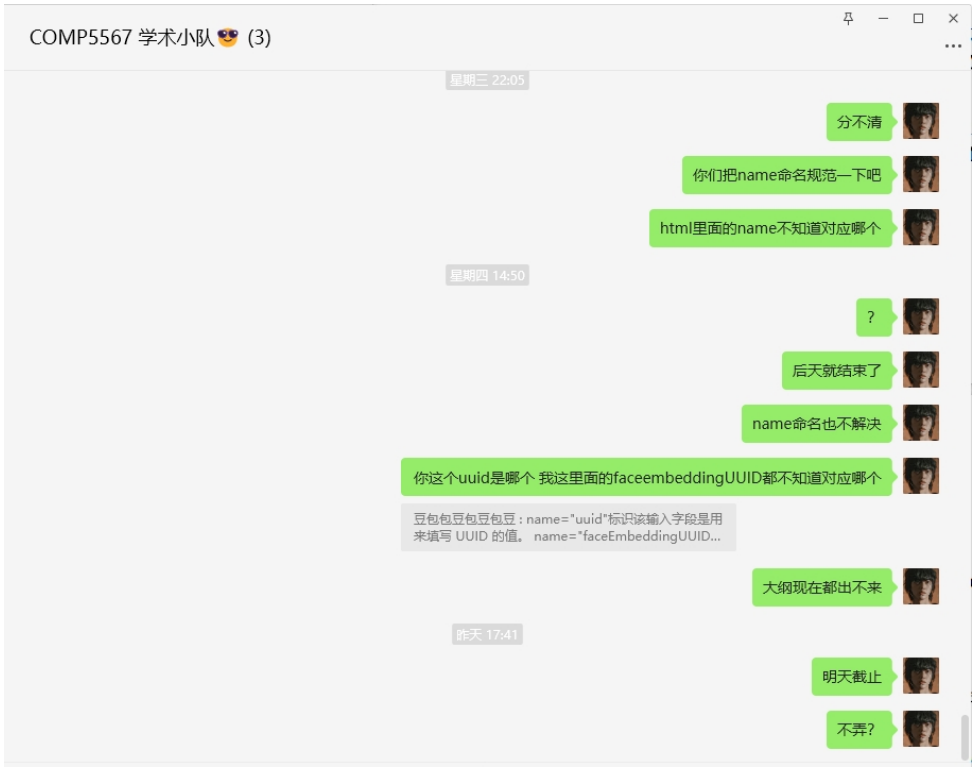

Figure 12: Screenshot of the Wechat7.

- Li Dongwei (24117496g): Dongwei's assigned tasks were smart contract development, Metamask integration, and Hardhat setup. He, too, failed to contribute anything to this report. He shares the blame for the confusing and unprofessional "name" variable conflict, which greatly hampered my ability to understand the code and write the report. Despite my repeated requests for him to address this issue, he also ignored them, further exacerbating the problem. Like Hui, Dongwei also demonstrated poor teamwork. He also left the initial project meeting directly after school, preventing the group from having a productive discussion. His slow response to messages further hindered communication and collaboration.

- Liu Qingyuan (24052432g): I was responsible for backend development (Node.js), DeepFace integration, and I ended up writing 100% of this report on my own. Due to Hui and Dongwei's complete lack of contribution to the report and their highly unprofessional coding practices, specifically the "name" variable conflict, I had to spend significant extra time and effort deciphering their code, correcting their mistakes, and writing the entire report myself. This created an unfair and unnecessary workload for me.

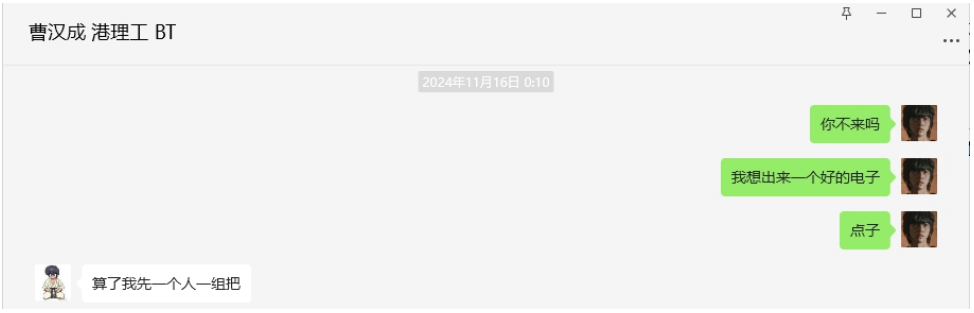

Figure 13: Screenshot of the Team member Missing.

- Important Note to Teacher: Our group initially consisted of four members. However, one member left the project specifically because of Hui and Dongwei's lack of cooperation, poor communication, and unprofessional behavior. This team member was particularly discouraged by their leaving the very first meeting, precisely when we had planned to discuss project topics after school. This behavior demonstrated a lack of commitment and made it clear that effective teamwork would be impossible. The attached chat logs show my attempts to communicate with them, their lack of response, and evidence related to the other team member leaving the project due to their conduct. I am extremely dissatisfied with Hui and Dongwei's lack of professionalism and their negative impact on the project. Their actions created a substantial amount of extra work and stress for me.

8 Conclusion

8.1 Achievements

- Built a Working Blockchain Degree Verification System: We successfully built a complete system for checking degrees using blockchain technology. This system, built on the Ethereum platform, ensures the security and transparency of academic credentials, making it much harder to fake or alter degree information. It's a big step towards making degree verification more reliable.

- Thoroughly Tested Core Features on a Simulated Blockchain: Using the Hardhat test network, which simulates a real blockchain environment, we rigorously tested all the main parts of the system. This included testing how degree information is saved, how face recognition works with the system, and how the voting mechanism helps keep the information accurate. These tests give us confidence that the system is ready for further development and eventual real-world use.

8.2 Future Directions

- **Make More Safe, Secret:** We want user data very safe. We will use better way to store data. We will use special code (hash) for face data and put on special internet storage like IPFS. No put face picture on blockchain direct. This keep face data safe. We will use strong secret code for all user data. We will update system often for keep safe from new bad thing.

- **Follow Law Everywhere:** Rule for data secret different in different place. We want our system work everywhere, so we follow all big rule like GDPR and CCPA. We use special internet name (DID) so user control own data, follow rule in different place. This important so people trust and use system safe.

- **Make Easy Use for Everyone:** We want everyone use this system – school, boss, student – all over world. We want system easy use, clear, everyone trust. Make check degree fast, easy, safe. This help stop fake degree, make school system better all over world. We think this system can change how we check degree.

References

[1] Y. X. Noshi, "Development of blockchain-based academic credential verification system ()," Open Access Library Journal, vol. 11, p. e12130, 2024.

[2] J.-C. Cheng, N.-Y. Lee, C. Chi, and Y.-H. Chen, "Blockchain and smart contract for digital certificate," in 2018 IEEE international conference on applied system invention (ICASI). IEEE, 2018, pp. 1046–1051.

[3] Y. Kistaubayev, G. Mutanov, M. Mansurova, Z. Saxenbayeva, and Y. Shakan, "Ethereum-based information system for digital higher education registry and verification of student achievement documents," Future Internet, vol. 15, no. 1, p. 3, 2022.

[4] M. Wang and W. Deng, "Deep face recognition: A survey," Neurocomputing, vol. 429, pp. 215–244, 2021.

[5] A. Tani, "Zero-knowledge proofs in blockchain applications," 2020.

[6] D. Trautwein, A. Raman, G. Tyson, I. Castro, W. Scott, M. Schubotz, B. Gipp, and Y. Psaras, "Design and evaluation of ipfs: a storage layer for the decentralized web," in Proceedings of the ACM SIGCOMM 2022 Conference, 2022, pp. 739–752.

[7] S. A. Amri, L. Aniello, and V. Sassone, "A review of upgradeable smart contract patterns based on openzeppelin technique," The Journal of The British Blockchain Association, 2023.

[8] R. Belen-Saglam, E. Altuncu, Y. Lu, and S. Li, "A systematic literature review of the tension between the gdpr and public blockchain systems," Blockchain: Research and Applications, vol. 4, no. 2, p. 100129, 2023.

[9] G. Alza Jr, "Blockchain & ccpa," Santa Clara High Tech. LJ, vol. 37, p. 231, 2021.

[10] T. Manoj, K. Makkithaya, and V. Narendra, "A blockchain based decentralized identifiers for entity authentication in electronic health records," Cogent Eng, vol. 9, no. 1, p. 2035134, 2022.