**Introduction**

Background and Challenge

Social networks are an integral part of life for most people. People make friends on Facebook and Twitter, find jobs on LinkedIn. Concerning Twitter, it is the second-largest social network, has more than 1 billion registered users and 326 million active users [1]. It proves that people rely heavily on social networks. They are very willing to share their daily life on those platforms. However, the existence of social networks inevitably forces people to face privacy issues [2]. For professionals, except for the usual information leakage, also have the risk of leaking the content of tweets. Due to hundreds of millions of people using social networking platforms, some people may think that they will not be discovered when unscrupulous talking about sensitive information related to the company, such as salary, revealing the idea of job-hopping, complaining about colleagues, company and employer. However, the recommendation system of social networks would constantly recommend users to workmates or friends of friends [3]. So many ways can lead to leakage of improper comments. Employees who post sensitive information might face pressure from their companies if their inappropriate statements are discovered by companies. Some professionals use non-public social accounts to send posts. But if a post includes any name, location, or company can also be easily recognized by people around them. Therefore, it is practical to post sensitive company information on social platforms without being detected.

In addition, people are increasingly suffering from employment pressure due to the COVID-19 pandemic, leaving many job seekers in a weaker position. From March to April 2020, the unemployment rate in America rose from 4.4% to 14.7% [4]. Not only because of the depressed market, information inequality between job seekers and companies is also a crucial reason. Companies always want to select fewer demanding employees, such as accepting lower pay or working overtime without complaint. Corporations can sift through resumes, compare candidates and choose the best value for money. And job seekers are stuck waiting again and again. Even if he does land a job, he may encounter a poor working environment but doesn't realize it until he starts working. There is also information inequality in terms of salary, which job seekers are most interested in. Newly graduated students have no idea of the salary level of different cities, companies and different positions. So, they may suffer loss in the salary aspect when looking for a job. Besides, people seldom talk about salary. If someone gets a disproportionate amount of money, they may not find out. Some efforts have been made to address this situation. In 2016, LinkedIn launched Salary Insights, which is a system that gathers salary information from members to provide salary insights to job seekers [5]. In 2017, Kenthapadi proposed the LinkedIn compensation product, which helps people calculate their earning potential by collecting a large amount of data [6]. These are rewarding and innovative products. But at present, the coverage of these products is low and is not suitable for many regions and positions. Also, many job seekers are looking for information other than salaries, such as working environment and intensity. Therefore, it is an urgent need for a secure social platform, which allows people to talk freely about their careers without the risk of being discovered.

Solution

SafeChat is a Web-based social networking platform that detects sensitive information in real-time. In SafeChat, authenticated users can post sensitive information without fear of being discovered by their bosses. Users are required to fill out basic information and the company they belong to when signing up for the social networking platform. Unlike other social networks, SafeChat allows people to choose to post anonymously, allowing them to hide their identities while posting. In addition, people can choose to encrypt posts through the Base64 algorithm. If the user encrypts the content, he is also anonymous automatically. All encrypted information cannot be seen by other employees of the same company. When other people read this post, they can click the decryption button next to it, then navigate into the decryption interface. The decryption page contains decrypted contents of tweets, which is forbidden to copy. The page is also full of watermarks with the reading user's real name, which helps prevent people from taking screenshots or photos to spread it. When users publish information, the system will automatically detect sensitive information. If there is sensitive information, NER detection will be performed on it, with the main detection objects being name, organization, location and money. If the post contains sensitive information, but the NER tool does not detect identity information, the system prompts the user to remain anonymous. If sensitive information and entity information are both distinguished, the system prompts the user to encrypt it.

**Related Work**

This section mainly describes the work related to sensitive information detection.

In 2011, Mao proposed three types of tweets that could leak privacy and are worthy of attention [7]. Vacation tweets, drunk tweets and illness tweets. He used naive Bayes and the SVM classifier to classify sensitive information. The experiments indicate that the naive Bayes performed better than SVM. The accuracy of holiday tweets is 76% in naive Bayes. But the range of sensitive information involved is relatively small in this study, with only three aspects. In 2014, Islam divided tweets into 200 topics [8]. The detection probability of privacy information is significantly increased by pre-defining the theme of the content published on social networks and detecting sensitive information according to the related features of the topic. The author also found that naive Bayes had a better performance than in classification. Both above studies are limited to identifying leaks of sensitive information from published tweets. In the context of increasingly serious privacy issues and people's increasing attention to privacy, it reflects the importance of real-time monitoring of sensitive information. In 2017, Cappellari built a privacy decision tool to alert users of potential privacy disclosure risks before sensitive messages leak to social platforms [9]. He used five algorisms, including nearest neighbour, naive Bayes, SVM etc. In his study, the SVM obtained the highest accuracy. Besides, in the same year, Neerbeky developed a real-time privacy detection desktop application based on RNN [10]. However, the author does not provide specific data for model accuracy. In 2018, Canfora used NLP to detect sensitive information in social networks [11]. The method is to judge sensitive information by analyzing sentence structure, word order and context, rather than relying on specific data sets. However, the problem brought by this method is the incompleteness of the heuristic set, which is easy to misjudge complex sentences.

Nowadays, social networks are all based on the Web or mobile end. If it is a privacy detection system for social platforms, embedding the detect system in the Web end would achieve high availability. This thesis will focus on the work scene to study the performance of sensitive information detection in social networks. The data of choice were tweets containing workplace keywords, including work, job, colleague, workmate, boss, salary, wage, overtime, and a host of other privacy terms. As the research direction of this thesis is highly targeted, the selection of keywords is small and accurate. For the same number of tweets, using keywords yields more tweets available.

**Related Tools**

JavaScript

JavaScript is a function-first, lightweight, just-in-time compiled programming language, which on the Website is to control the behaviour of the Web page [12]. JavaScript is one of three languages that Web developers must learn, along with HTML and CSS. Most of all web pages today are developed based on JavaScript.

Vue.js

Vue is a lightweight JavaScript library developed by You in 2014. The features of this framework are data binding and componentized development. For those who have learned the basics of the front-end, the framework is easy to use and has good performance. It is more popular on Github than React and Angular [13]. For the lightweight development goal of this project, Vue is suitable to be used as a front-end framework.

Bootstrap

Bootstrap is a front-end page framework developed by Mark Otto and Jacob Thornton, designers of Twitter. It is based on HTML, CSS and JavaScript and written by the dynamic CSS language Less, which provides an elegant specification for the front-end. Bootstrap also has a framework for Vue, called Bootstrapvue, which help Vue developers use Bootstrap.

Node.js

Node.js is a JavaScript runtime environment based on the Chrome V8 engine. It uses an event-driven, non-blocking I/O model, making it lightweight and efficient [14]. Node.js is powerful, and its package manager, npm, is the largest open-source library ecosystem in the world.

MySQL

MySQL can run all medium and large databases, suitable for Web development. It can handle the database containing tens of millions of orders of magnitude records and run on different systems. It is the most widely used relational database management system [15]. MySQL has the advantages of small size, fast speed and low cost. Besides, it is open-source, allowing most small and medium-sized websites to choose MySQL when choosing their database. MySQL is a very suitable database for this project.

**Requirements**

Requirements gathering

Collect requirements through interviews at the beginning of the project. The purpose of the interview is to understand the behavioural characteristics and preferences of employees using social platforms. And their pain points in using social networks. All the questions should refine according to the interview purpose, and conservations should build around the work and social network. Users interviewed need to be screened in combination with SafeChat features, namely, people who like to use social networks and are already working or about to work. Divide users into core users and potential users. Core users are those who want to be active on workplace social platforms, and potential users are defined as the target user but not currently considered using SafeChat. The core users of this research object are two employees of Internet companies and an HR of an Internet company. Potential users are two graduates. Before the interview, different questions should be chosen for different types of interviewees, and each interviewer should answer about ten questions. During the interview, supplementary questions can be asked by adjusting the space flexibly. After the interview, collate all content for the requirements design of SafeChat and prioritize functions using Moscow. Interview questions are in the appendix.

The following table shows the functions designed according to the requirements, classified by the MoSCoW method [16]. The classification basis of MoSCoW is displayed in Table 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Module** | **Description** | **MoSCoW** | **Implemented** |
| 1 | Basic Function | User can register | M | Y |
| 2 | User can login | M | Y |
| 3 | System should encrypt users' passwords | S | Y |
| 4 | Users can upload profile pictures | S | Y |
| 5 | Users can modify personal information | M | Y |
| 6 | Send Posts | Users can send posts directly | M | Y |
| 7 | Users can insert emoticons in posts | C | Y |
| 8 | Users can send posts anonymously | M | Y |
| 9 | Users can send encrypted posts | M | Y |
| 10 | System must detect sensitive information in posts | M | Y |
| 11 | System must use NER tool detect posts | M | Y |
| 12 | System must suggest sending mode to users | M | Y |
| 13 | Read Posts | Users can view the avatar and name of the publisher | M | Y |
| 14 | Users can read all direct and anonymous posts | M | Y |
| 15 | Users can only view encrypted posts from employees of other companies | M | Y |
| 16 | User can like posts | S | Y |
| 17 | Decrypt Posts | Users can decrypt posts from employees of other companies | M | Y |
| 18 | System could place the watermark of the user's name and email on the decryption interface | C | Y |
| 19 | System must forbid users to copy content on the decryption page | S | Y |
| 20 | Comment Posts | Users can view post comments | M | Y |
| 21 | Users can comment on post | S | Y |
| 22 | Users can insert emojis into comments | C | Y |
| 23 | Users can make comment anonymously | S | Y |
| 24 | Users can encrypt their comments | W | N |
| 25 | Search Posts | User can find posts by entering keywords | S | Y |
| 26 | System should display the content and publisher name for non-anonymous posts | S | Y |
| 27 | System must display only the content of anonymous posts | M | Y |

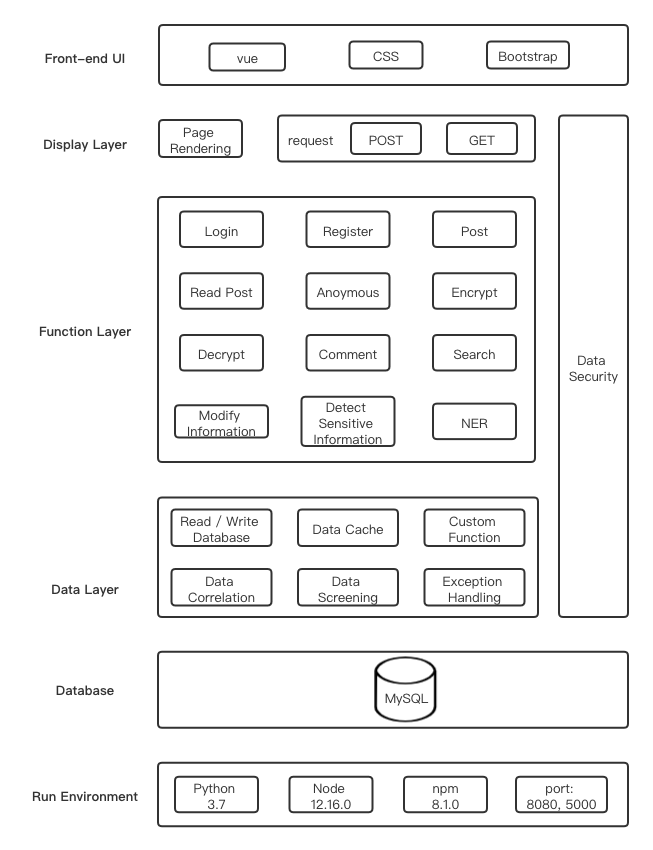
Table 3-1 Requirement List

|  |  |
| --- | --- |
| **Category** | **Criteria** |
| **M**ust | **Must** have requirement |
| **S**hould | **Should** have if at all possible |
| **C**ould | **Could** have but not Critical |
| **W**on't | Would be good to have… (**Won't** have time to do it now, but maybe later) |

Table 3-2 MoSCoW Criteria

**Design**

System Structure

****

The system is a social network based on the Web. The system design divides the project into six parts. Starting from the bottom are the run environment, database, data layer, function layer, display layer and front-end UI.

**Run Environment**

**JavaScript**: Web side interface development has always been based on JavaScript, which is a client-side scripting language [12]. The main features are embedding dynamic text into HTML pages, responding to browser time, validating data before it is submitted to the server, and so on. It is suitable for the front-end development of this project. But JavaScript, as a Web script, has no server-side. Before Node.js, JavaScript development was usually done with the help of PHP, because it was easy to integrate with HTML to help developers build dynamic websites. But since its release, Node.js has been on the radar of developers with remarkable performance and speed. Compared to PHP, Node.js is better suited for real-time applications, dynamic single-page applications, and multiple front-end technologies such as Vue and React [17]. Therefore, this project chooses Node.js as the running environment of JavaScript. There are many versions of Node.js, and the prerequisite for choosing a suitable version is a stable development environment.

**Node.js**: In April 2021, the 10.x version of Node.js will enter the EOL phase, and versions entering this timeline will no longer be maintained. Currently, versions 12.x and 14.x are still being updated and maintained, but 12.x has entered the more stable Maintenance stage. Therefore, we chose the most stable version of JavaScript as the runtime environment, 12.16.0, which solves the problem of packaging speed for many large projects and supports most package installations.

**npm**: A Node.js package management and distribution tool that helps developers quickly install packages and dependencies needed for a project. This project does not have high requirements on the version of NPM. Currently, the latest version is 8.1.2, and I choose to install 8.1.0.

**Python**: An object-oriented interpreted programming language. Python is used in this project to detect sensitive information and use NER tools, so a Python compilation environment is required. Python is mainly available in Python2.x and Python3.x versions. Many of the features of Python2 are incompatible with the current officially supported version. Python3 is already a stable version that is not compatible with Python2.x, and many enterprises are gradually migrating from Python2 to Python3 [18]. Python3.7 was chosen to compile the .py file in the project.

officially

**Port**: Computers use different logical ports to distinguish between different services. Ports cannot be occupied by multiple services. Select 8080 and 5000, which does not conflict with the system, as the front and backports of the project.

**Database**

Database logical structure design

After the demand analysis of the system, the database tables involved in the system are designed, including users, tweets, comments and likes.

The database structure is shown in figure 4-1, and the specific data table designs are displayed in Table 4-11 to Table 4-4.

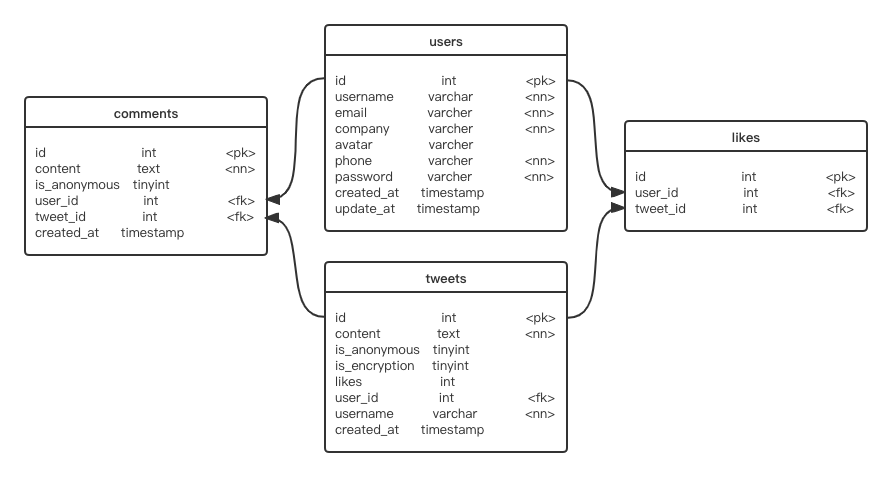


Table 4-1 Stores user information.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Name | Data Type | Length | Description | Requirement |
| id | int | 0 | ID | Primary Key |
| username | varchar | 255 | Full name | Not Null |
| email | varchar | 255 | Email | Not Null |
| company | varchar | 255 | Company | Not Null |
| avatar | varchar | 255 | Avatar |  |
| phone | varchar | 255 | Phone number | Not Null |
| password | varchar | 255 | Password | Not Null |
| created\_at | timestamp | 0 | Create time |  |
| update\_at | timestamp | 0 | Update time |  |

Table 4-1 users

Table 4-2 Stores tweets information.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Name | Data Type | Length | Description | Requirement |
| id | int | 0 | ID | Primary Key |
| content | text | 255 | Content | Not Null |
| is\_anonymous | tinyint | 1 | Whether anonymous |  |
| is\_encryption | tinyint | 1 | Whether encryption |  |
| likes | int | 0 | Number of thumbs up | Not Null |
| user\_id | int | 0 | Post publisher ID | Foreign Key1 |
| username | varchar | 255 | Post publisher name | Foreign Key2 |
| created\_at | timestamp | 0 | Create time |  |

Table 4-2 tweets

In Table 4-2, the primary key of Foreign Key1 is “id” in the users’ table. The primary key of Foreign Key2 is “username” in the users’ table.

Table 4-3 Stores tweets information.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Name | Data Type | Length | Description | Requirement |
| id | int | 0 | Comment ID | Primary Key |
| content | text | 0 | Content | Not Null |
| is\_anonymous | tinyint | 1 | Whether anonymous |  |
| user\_id | int | 0 | “Comment” user’s ID | Foreign Key3 |
| tweet\_id | int | 0 | Post ID | Foreign Key4 |
| created\_at | timestamp | 0 | Create time |  |

Table 4-3 comments

In Table 4-3, the primary key of Foreign Key3 is “id” in the “users” table. The primary key of Foreign Key4 is “id” in the “tweets” table.

Table 4-4 Stores like information.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Name | Data Type | Length | Description | Requirement |
| id | int | 0 | Like ID | Primary Key |
| user\_id | int | 0 | “Like” user’s ID | Foreign Key5 |
| tweet\_id | int | 0 | Post ID | Foreign Key6 |

Table 4-4 likes

In Table 4-4, the primary key of Foreign Key5 is “id” in the “users” table. The primary key of Foreign Key6 is “id” in the “tweets” table.

**Data Layer**

The data layer is responsible for database access and can read database files to access data located in persistent containers. In the data layer, the system receives data from the browser, processes it before passing it to the database.Data processing includes read/write database, data cache, data screening, data correlation, exception handling and custom function.

**Read/Write Database:** Use Node.js to manipulate the database. The "users" table contains an API for adding, updating and searching data. Users can create and modify their accounts and personal information. The system can query user information based on user input to find account to complete login. The "tweets" table contains an API for adding and searching, and users can publish posts or search posts according to keywords. The "comments" table includes the add API, which is called when the user comments. The "likes" table also only sets up the add API, which is called when the user thumbs up tweet.

**Data Cache:** After the user logs in, the system caches the current user's data until he clicks logout to exit.

**Data Correlation:** According to the command of the database, design foreign keys. Associate users' comments and likes with the current posts' ID.

**Custom functions:** Sensitive information detection functions. Including sensitive information detection and NER entity recognition. Functions store as a Python file, called in this project by using python-shell.

**Function Layer**

The functional layer is mainly divided into website construction and sensitive information recognition.

The functions of the website construction include:

1. Basic functions of user login and registration

2. Users send tweets

3. Users read tweets

4. Users review tweets

5. Users search for tweets

Table 3-1 describes the functional design of the website.

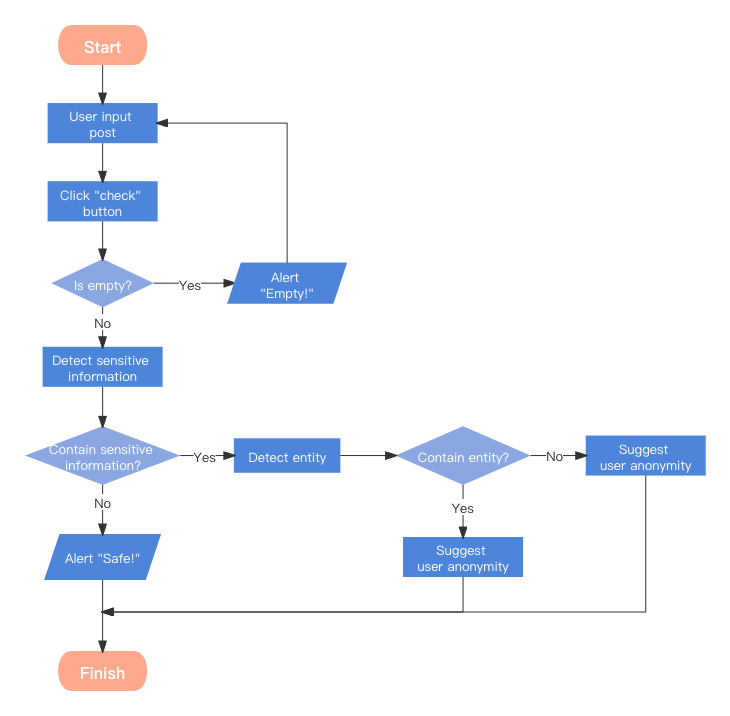
Sensitive information recognition function:

The sensitive information detection system consists of two parts. The first part is sensitive information detection for the content of tweets, and the second part is entity detection using the NER tool. NER tests include name, salary, location and company.

1. If the system detects sensitive information through the sensitive information detection system, but does not detect any entity information through the NER tool, the user will be informed to leak sensitive information, and the user will be advised to post anonymously.

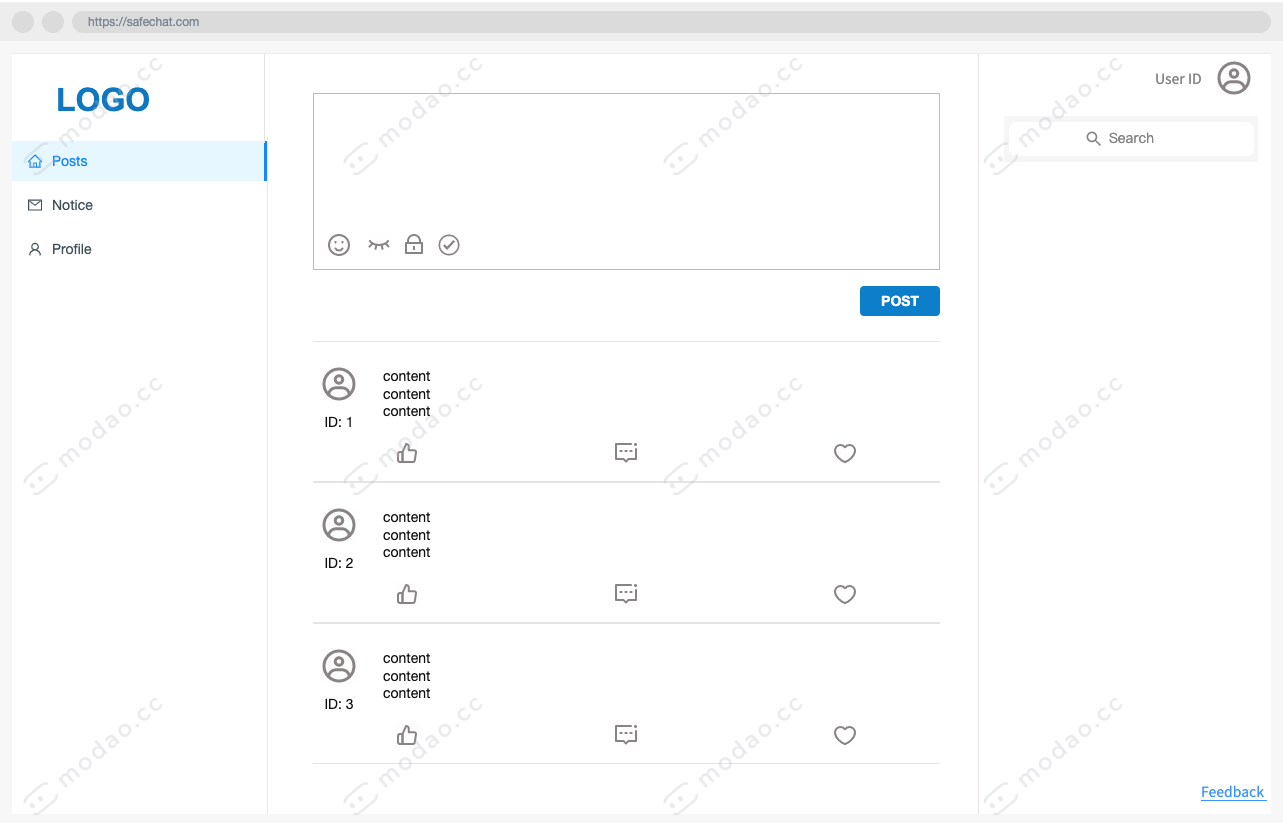
2. If the system detects sensitive information through the sensitive information detection system and detects any entity information through the NER tool, it will notify the user of leaking sensitive information and suggest the user encrypt the Posting.

The logical design is as the following diagram.



**Display Layer and Front-end UI**

The site prototype is displayed below.



The site is a three-column distributed site. The left column is the navigation bar with three modules. The first is the home page, the second is the system notification, and the third is the personal information page. The middle bar is used for Posting and interacting with other posts. On the far right is the search section.

**Implementation**

References

1. Sohail SS, Khan MM, Alam MA. An Analysis of Twitter Users From The Perspective of Their Behavior, Language, Region and Development Indices--A Study of 80 Million Tweets. arXiv preprint arXiv:210510245. 2021.

2. Faisal M, Alsumait A. Social network privacy and trust concerns. Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services; Ho Chi Minh City, Vietnam: Association for Computing Machinery; 2011. p. 416–9.

3. Zheng F, Ma L, editors. A Multi-layered Friend Recommendation System on Twitter. 2021 The 13th International Conference on Computer Modeling and Simulation; 2021.

4. Gezici A, Ozay O. An Intersectional Analysis of COVID-19 Unemployment. Journal of Economics, Race, and Policy. 2020;3(4):270-81.

5. Chen X, Liu Y, Zhang L, Kenthapadi K. How LinkedIn Economic Graph Bonds Information and Product: Applications in LinkedIn Salary. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining; London, United Kingdom: Association for Computing Machinery; 2018. p. 120–9.

6. Kenthapadi K, Ambler S, Zhang L, Agarwal D. Bringing Salary Transparency to the World: Computing Robust Compensation Insights via LinkedIn Salary. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management; Singapore, Singapore: Association for Computing Machinery; 2017. p. 447–55.

7. Mao H, Shuai X, Kapadia A. Loose tweets: an analysis of privacy leaks on twitter. Proceedings of the 10th annual ACM workshop on Privacy in the electronic society; Chicago, Illinois, USA: Association for Computing Machinery; 2011. p. 1–12.

8. Islam AC, Walsh J, Greenstadt R. Privacy Detective: Detecting Private Information and Collective Privacy Behavior in a Large Social Network. Proceedings of the 13th Workshop on Privacy in the Electronic Society; Scottsdale, Arizona, USA: Association for Computing Machinery; 2014. p. 35–46.

9. Cappellari P, Chun SA, Perelman M. A Tool for Automatic Assessment and Awareness of Privacy Disclosure. Proceedings of the 18th Annual International Conference on Digital Government Research; Staten Island, NY, USA: Association for Computing Machinery; 2017. p. 586–7.

10. Neerbeky J, Assentz I, Dolog P, editors. TABOO: Detecting Unstructured Sensitive Information Using Recursive Neural Networks. 2017 IEEE 33rd International Conference on Data Engineering (ICDE); 2017 19-22 April 2017.

11. Canfora G, Sorbo AD, Emanuele E, Forootani S, Visaggio CA. A Nlp-based Solution to Prevent from Privacy Leaks in Social Network Posts. Proceedings of the 13th International Conference on Availability, Reliability and Security; Hamburg, Germany: Association for Computing Machinery; 2018. p. Article 36.

12. Delcev S, Draskovic D, editors. Modern JavaScript frameworks: A survey study. 2018 Zooming Innovation in Consumer Technologies Conference (ZINC); 2018: IEEE.

13. Wohlgethan E. SupportingWeb Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue. js: Hochschule für Angewandte Wissenschaften Hamburg; 2018.

14. Tilkov S, Vinoski S. Node.js: Using JavaScript to Build High-Performance Network Programs. IEEE Internet Computing. 2010;14(6):80-3.

15. Letkowski J. Doing database design with MySQL. Journal of Technology Research. 2015;6:1.

16. Waters K. Prioritization using moscow. Agile Planning. 2009;12:31.

17. Chitra LP, Satapathy R, editors. Performance comparison and evaluation of Node.js and traditional web server (IIS). 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET); 2017 16-18 Feb. 2017.

18. Malloy BA, Power JF. An empirical analysis of the transition from Python 2 to Python 3. Empirical Software Engineering. 2019;24(2):751-78.