



Technical Report

**Group 4 of intro of software engineering**

Content

[1. Introduction 3](#_Toc1746)

[2. Project Architecture and Design 3](#_Toc16325)

[2.1 System Architecture 3](#_Toc5107)

[2.2 Class Diagrams 5](#_Toc197)

[3. Sequence Diagrams 7](#_Toc2797)

[4. Testing Strategy 11](#_Toc5258)

[4.1 Types of Tests 11](#_Toc29087)

[4.2 Test Cases 12](#_Toc22030)

[4.3 Test Data and Expected Results 13](#_Toc14698)

[4.4 Test Execution and Results Analysis 14](#_Toc4684)

**[5.Graphical User Interface (GUI) Design](#_Toc8180)** [15](#_Toc8180)

**[5.1 User Interface Components](#_Toc2695)** [15](#_Toc2695)

[5.2 Visual Design 20](#_Toc15067)

[5.3 User Experience (UX) Considerations 22](#_Toc6563)

[6. Implementation Details 23](#_Toc11829)

[6.1 Technologies Used 23](#_Toc7474)

[6.2 Database Schema and Data Models 25](#_Toc19404)

[7. Security Measures 26](#_Toc2105)

[7.1 Data Encryption 26](#_Toc2421)

[7.2 Authentication and Authorization 27](#_Toc29867)

[7.3 Handling Sensitive Information 27](#_Toc28409)

[8. Performance Optimization 28](#_Toc29230)

[8.1 Caching Strategies 28](#_Toc24676)

[8.2 Database Indexing 29](#_Toc942)

[8.3 Code-Level Optimizations 29](#_Toc4030)

[9. Future Work 30](#_Toc2747)

[9.2.1 Scalable Infrastructure: 31](#_Toc31082)

[9.2.5 Monitoring and Analysis: 32](#_Toc5848)

[10. Summary 32](#_Toc10446)

[10.1 Key Findings Summary 32](#_Toc22)

[10.2 Acknowledgments 33](#_Toc12018)

**1. Introduction**

This technical report provides a comprehensive exploration of the technical aspects and design considerations for the development of a Mental Wellness Forum website. The report delves into system architecture, design decisions, and key components critical for project development and maintenance.

The primary objectives of this technical report are to offer a detailed technical guide for the project's technical team, developers, and relevant stakeholders. It aims to ensure a clear understanding of the system's structure and functionality to facilitate effective development and maintenance.

The primary audience for this technical report includes the development team, system architects, project managers, and other stakeholders involved in the project's development and maintenance. The report's scope covers detailed information on system architecture, design decisions, and key components.

**2. Project Architecture and Design**

2.1 System Architecture

2.1.1High-Level Overview

In the context of our Mental Wellness Forum project, we have adopted a layered architecture to ensure scalability and modularity. Here is a high-level overview of the overall system architecture:

1. Client Layer:

- Responsible for the user interface and user interaction.

- Includes functionalities such as browsing the forum and posting messages.

- Utilizes modern front-end frameworks, such as React.

1. Server Layer:

- Handles user requests and manages business logic.

- Includes functionalities like user authentication, post management, and messaging.

- Built using Node.js and the Express framework.

1. Database Layer:

- Manages data storage and retrieval.

- Utilizes a relational database (e.g., MySQL) to store user information, posts, messages, etc.

1. Authentication and Authorization Service:

- An independent service responsible for user authentication and authorization.

- Issues and validates identity tokens using JSON Web Tokens (JWT).

2.1.2 Components and Modules

Offers detailed insights into the functions and relationships of individual components and modules within the system.

1. Client Layer

User Interface Module:

- Responsible for rendering the user interface, including topic lists, post content, etc.

- Communicates with the backend through APIs to fetch and submit data.

User Interaction Module:

- Handles user actions like posting messages, replies, etc.

- Calls backend APIs to update the system state.

2.Server Layer

User Management Module:

- Manages user-related functionalities such as registration, login, and profile modifications.

- Interacts with the database layer for storing and retrieving user information.

Post Management Module:

- Manages forum posts, including creation, editing, and deletion.

- Utilizes the database layer for storing and retrieving post data.

Messaging Module:

- Manages real-time messaging between users.

- Uses technologies like Socket.io for real-time communication.

3.Database Layer

- User Database: Stores user-specific information, such as usernames, hashed passwords, etc.

- Forum Database: Contains data related to forum posts, comments, topics, etc.

4.Authentication and Authorization Service

- Authentication Module: Handles user logins, generates, and validates identity tokens.

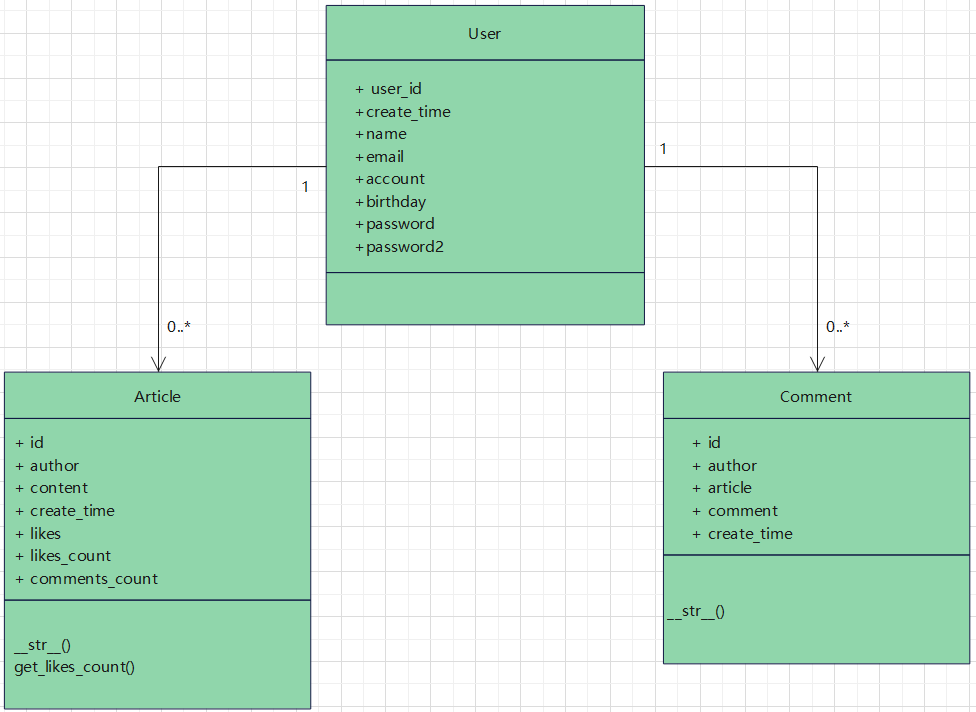
- Authorization Module: Ensures user access rights within the system, controlling access to specific resources.

This architecture and design ensure a clear separation of concerns, making each module and component have a specific function, enhancing the system's maintainability, scalability, and upgradeability.

2.2 Class Diagrams

2.2.1Detailed Class Diagrams

Illustrates detailed structures of key classes in the system, including attributes and methods.

Below is the detailed structure of key classes in the Mental Wellness Forum website, including attributes and methods, along with the relationships and associations. 

Class Diagram Explanation:

1. User Class:

- Attributes:

- user\_id: UUIDField, serves as the primary key.

- create\_time: DateTimeField, records the user's creation time.

- name: CharField, stores the user's name.

- email: EmailField, stores the user's email.

- account: CharField, stores the user's account.

- birthday: DateField, stores the user's birthday.

- password: CharField, stores the user's password.

- password2: CharField, stores a copy of the user's password.

2. Article Class:

Attributes:

- id: AutoField, serves as the primary key.

- author: ForeignKey, linked to the User class, represents the author of the article.

- content: TextField, stores the content of the article.

- create\_time: DateTimeField, records the creation time of the article.

-likes: ManyToManyField, associated with the User class, represents users who like the article.

- likes\_count: IntegerField, records the total number of likes.

- comments\_count: IntegerField, records the total number of comments.

Methods:

- \_\_str\_\_(self): Returns a string representation of the article content.

- get\_likes\_count(self): Returns the number of users who like the article.

3. Comment Class:

- Attributes:

- id: UUIDField, serves as the primary key.

- author: ForeignKey, linked to the User class, represents the author of the comment.

- article: ForeignKey, linked to the Article class, represents the article being commented on.

- comment: TextField, stores the content of the comment.

- create\_time: DateTimeField, records the creation time of the comment.

- Methods:

- \_\_str\_\_(self): Returns a string representation of the comment content.

- 2.2.2 Relationships and Associations

Explains the relationships and associations between classes, emphasizing interactions within the system.

User and Article:

- One-to-many relationship; one user can create multiple articles, but each article can only have one author.

Article and User:

- Many-to-many relationship; multiple users can like one article, and one user can like multiple articles.

Article and Comment:

- One-to-many relationship; one article can have multiple comments, but each comment belongs to a single article.

User and Comment:

- One-to-many relationship; one user can post multiple comments, but each comment can only be posted by one user.

Comment and User:

- One-to-many relationship; one comment can only be posted by one user.

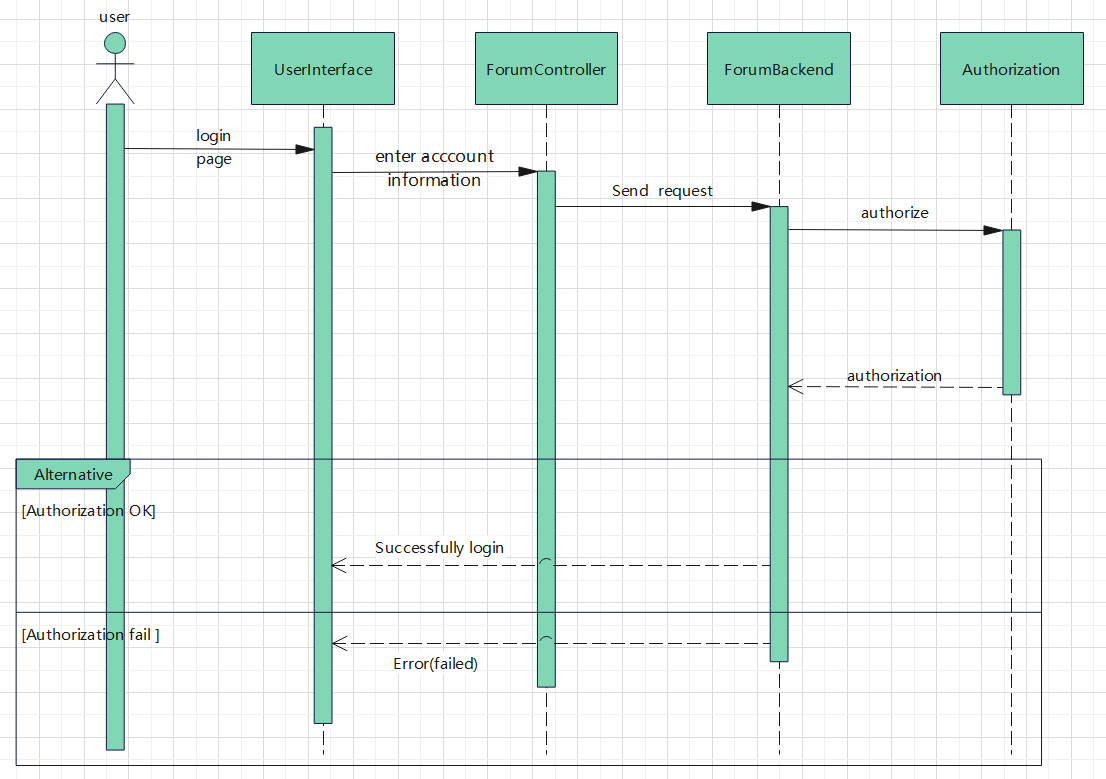
Comment and Article:

- One-to-many relationship; one comment belongs to one article.

**3. Sequence Diagrams**

Timing diagrams are often used to describe the order of interactions between objects in a system. Here are some time charts from the Healing Forum website, including an overview of system interactions and detailed time charts of major processes.

User Login:



Interaction Overview:

1. Trigger Event: The user initiates the login process by opening the login page of the psychological healing website. This could be by accessing the website's homepage, clicking on a login button, or any other method that starts the login flow.

2. Request for Login Page: Upon receiving the user's login request, the system responds by providing the login page. This page typically includes elements such as username and password input fields, a forgot password option, and registration links, offering the necessary tools for user authentication.

3.Input Credentials: The user enters their registered username and password on the login page. These credentials will be used to verify the user's identity.

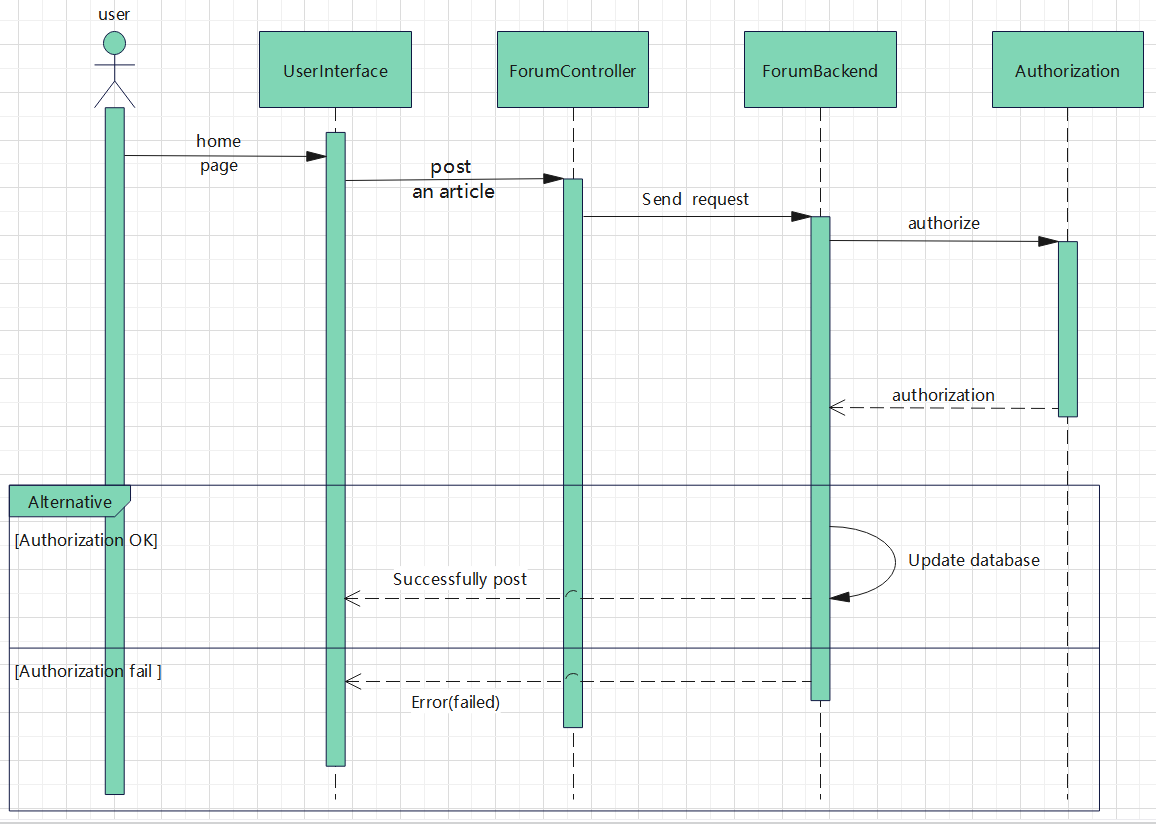
4. Submit Credentials: After completing the input, the user submits the login credentials to the system by clicking the "Login" button or performing a similar action. Upon receiving the credentials, the system begins the process of validating the user's identity.

5. Identity Verification:The system verifies the user's identity by comparing the entered username and password with the corresponding information stored in the database. If the verification is successful, the system generates a temporary identity token for managing the user's session.

6.Token Generation: Upon successful identity verification, the system generates a unique token and returns it to the user's device. This token will be used to authenticate each of the user's subsequent requests during the session.

7. Access Homepage: Once the user successfully logs in and establishes a valid session, the system redirects the user to their personal homepage or another appropriate target page. The user can now access their personal information, perform specific actions, or begin using the services provided by the psychological healing website.

Post Creation:



Interaction Overview:

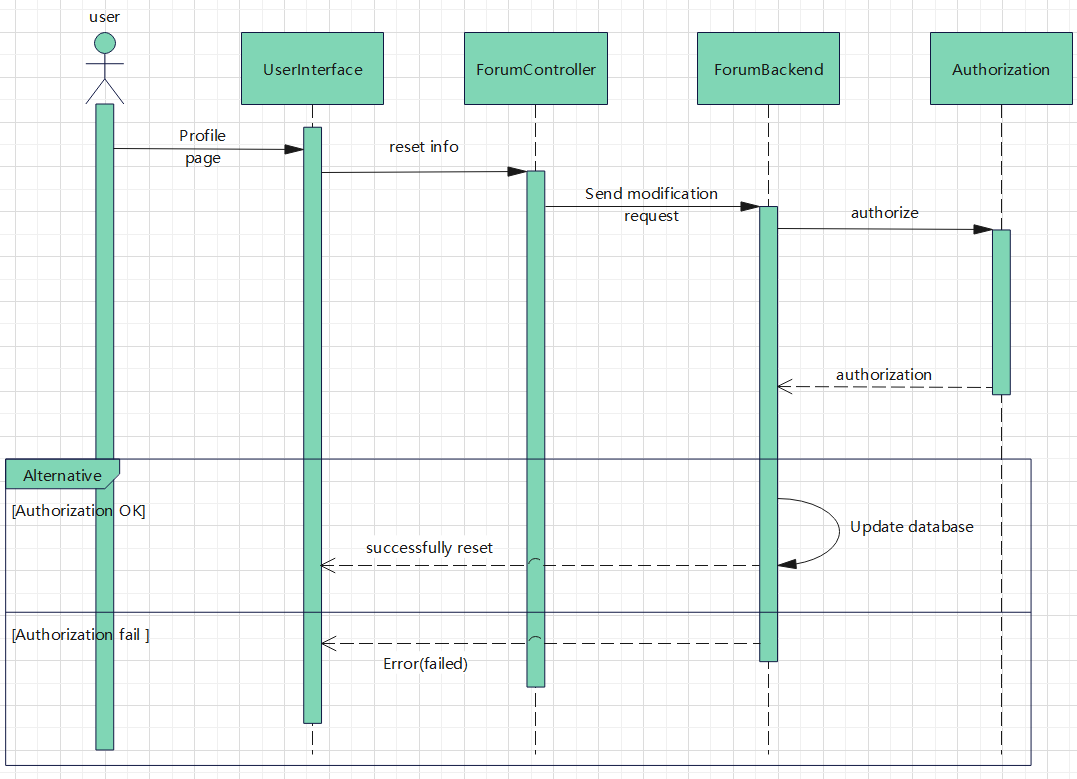
1. User creates a post and inputs content.

2. System verifies user identity and permissions.

3. System stores post information and returns the post identifier.

4.The post was posted successfully or could not be posted due to authentication failure.

User Modifying Personal Profile:



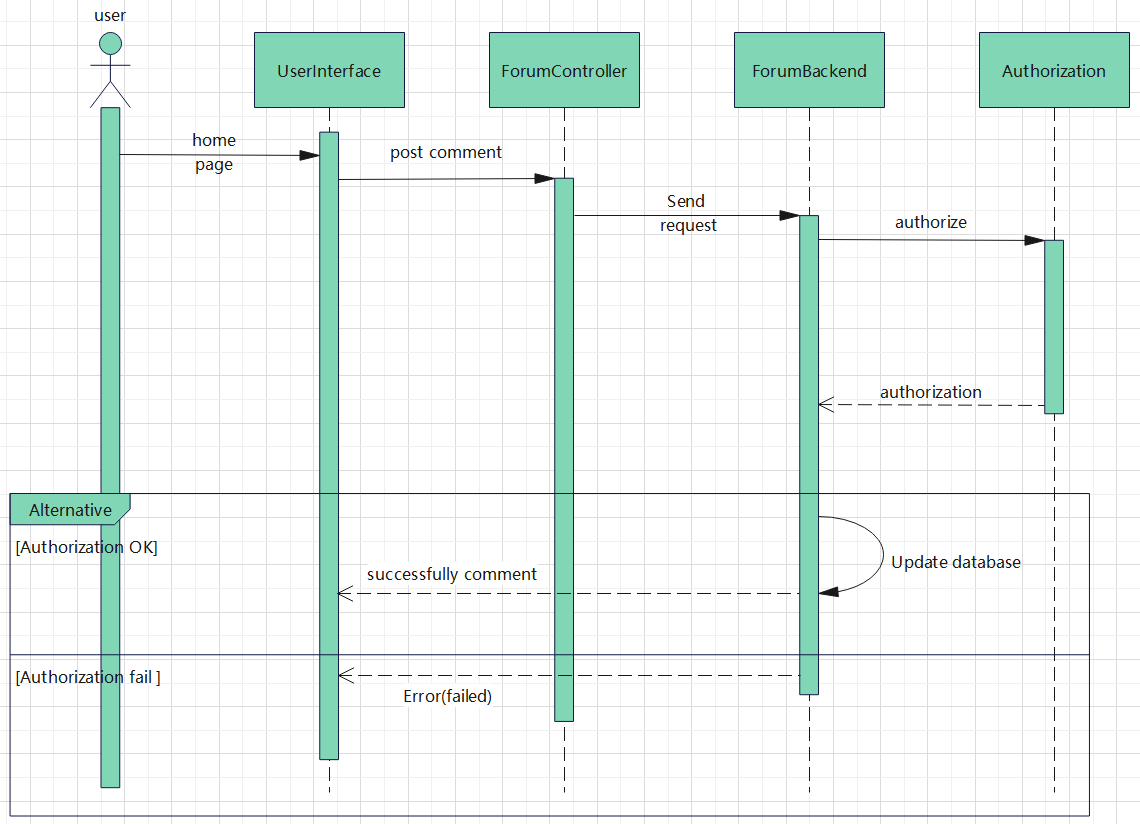
Detailed Sequence Diagram:

1. User enters the profile editing page and updates information.

2. System verifies user identity.

3. System updates user information and stores it in the database.

Post Commenting Process:



Interaction Overview:

1.The user initiates a request to comment on a post through the front-end.

2.The request reaches the back-end, which verifies the user's identity.

3.The back-end creates a new comment, updates the database, and returns a success response.

4.The front-end receives the success response and updates the post interface.

The sequence diagrams provide a clear understanding of major user interactions such as registration, login, and post creation. This aids in identifying potential issues and optimization points.And the sequence diagrams offer profound insights into system interactions, helping identify potential performance bottlenecks and optimization opportunities. Analyzing the sequence diagrams enables better design and adjustment of the system to enhance user experience.

4. Testing Strategy

4.1 Types of Tests

4.1.1 Unit Tests

1. Objective:

- Verify the correctness of individual functions and methods within the system.

1. Approach:

- Use testing frameworks like Jest or JUnit for writing and executing unit tests.

- Ensure each unit test is isolated, independent, and covers various scenarios.

- Mock external dependencies to isolate units and ensure the tests are reliable.

4.1.2 Component Tests

1. Objective:

- Validate the interactions and collaboration between different components within the system.

1. Approach:

- Utilize tools like Enzyme for React components or similar frameworks for other technologies.

- Test component behaviors, state changes, and communication with other components.

- Mock APIs or services to simulate real-world component interactions.

4.1.3 Interface Tests

1. Objective:

- Ensure the correctness and reliability of internal and external interfaces.

1. Approach:

- Perform API testing to validate the requests and responses conform to the API specifications.

- Use tools like Postman or Swagger for API testing.

- For external interfaces, simulate responses from third-party services to ensure seamless integration.

4.1.4 Performance Tests

1. Objective:

- Evaluate the system's responsiveness, stability, and resource usage under different load conditions.

1. Approach:

- Use performance testing tools such as Apache JMeter or Gatling to simulate a variety of load scenarios.

- Measure and analyze response times, throughput, and resource utilization.

- Identify and address potential bottlenecks and optimize system performance.

4.2 Test Cases

4.2.1 Unit Test Cases

1. User Module:

- Test the creation, updating, and deletion of user accounts.

- Verify input validation for user data.

1. Post Module:

- Test the creation, editing, and deletion of posts.

- Validate the content validation logic for posts.

4.2.2 Component Test Cases

1. Front-end,Back-end Interaction:

- Test communication between front-end and back-end components.

- Verify that data is transmitted correctly, and responses meet the expected format.

2. Component Integration:

- Validate the seamless collaboration between different components.

- Test scenarios where one component relies on another for successful operation.

4.2.3 Interface Test Cases

1. Third-Party Service Integration:

- Test the system's ability to handle responses from third-party services.

- Validate error handling when third-party services are unavailable.

1. Database Interface:

- Validate the correctness of data read and write operations.

- Ensure that data integrity is maintained during various database interactions.

4.2.4 Performance Test Cases

1. Normal Load Testing:

- Simulate a normal user load and measure system response times.

- Verify that the system can handle the expected load without degradation.

2. Peak Load Testing:

- Simulate a peak user load to test the system's limits.

- Evaluate system stability and performance under stress conditions.

4.3 Test Data and Expected Results

4.3.1 Unit Test Data

1. User Module:

- Test Data: Various user profiles with valid and invalid data.

- Expected Results: Successful user creation, updates, or deletion with data correctly saved.

2. Post Module:

- Test Data: Posts with different content, comments, and varying lengths.

- Expected Results: Successful creation, editing, or deletion of posts with accurate data storage.

4.3.2 Component Test Data

1. Front-end,Back-end Interaction:

- Test Data: Simulate different types of requests from the front-end.

- Expected Results:Back-end processes requests correctly and returns expected responses.

2. Component Integration:

- Test Data:Combine functionalities of different components.

- Expected Results: Successful collaboration between components without conflicts.

4.3.3 Interface Test Data

1. Third-Party Service Integration:

- Test Data:Simulate responses from third-party services under various conditions.

- Expected Results:System correctly handles responses, including error scenarios.

2. Database Interface:

- Test Data: Test with various data scenarios, including edge cases.

- Expected Results: Data is accurately read from and written to the database.

4.3.4 Performance Test Data

1. Normal Load Testing:

- Test Data: Gradual increase in user load to simulate normal usage.

- Expected Results: System maintains acceptable response times and throughput.

2. Peak Load Testing:

- Test Data: Sudden and significant increase in user load.

- Expected Results:System remains stable, with a reasonable increase in response times.

4.4 Test Execution and Results Analysis

1. Execution:

- Execute automated tests regularly, especially during the development and integration phases.

- For manual tests, create test plans and ensure comprehensive coverage.

2. Analysis:

- Analyze automated test reports for failures and issues.

- Document and analyze issues from manual testing, focusing on root cause analysis.

3. Iterate and Optimize:

- Address issues found during testing promptly.

- Optimize test cases for better coverage and efficiency.

- Update test plans and cases to accommodate new features or changes in the system.

By following this detailed testing strategy, you can systematically verify the functionality, integration, interfaces, and performance of the system, leading to a more robust and reliable application.

**5.Graphical User Interface (GUI) Design**

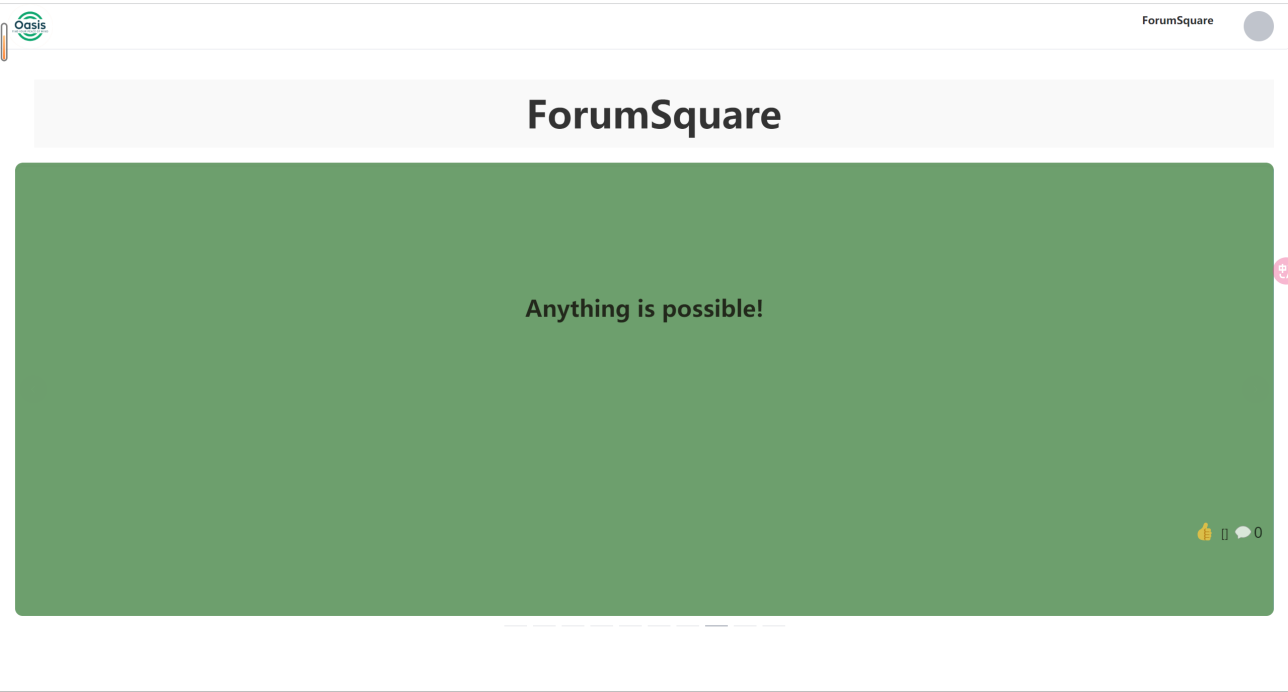
**5.1 User Interface Components**

**5.1.1Screens and Layouts**

**Forum Square**

Forum Square, through the use of vivid green tones and extensive white spaces, creates a fresh and tranquil visual effect. This aligns with the theme of the psychological forum - providing a safe space to support mental health. On this platform, every user's voice is heard, as evident from the showcased user article example, "Anything is possible!" This example is placed centrally on the page, using large, bold fonts to draw visitors' attention.

The top navigation bar is designed to be minimalist and intuitive, offering users an easy way to explore different forum sections and features. Overall, the design of Forum Plaza aims to immediately capture users' attention while providing sufficient information for easy navigation to other parts. This design philosophy reflects our commitment to creating an inclusive, positive, and user-friendly forum environment.

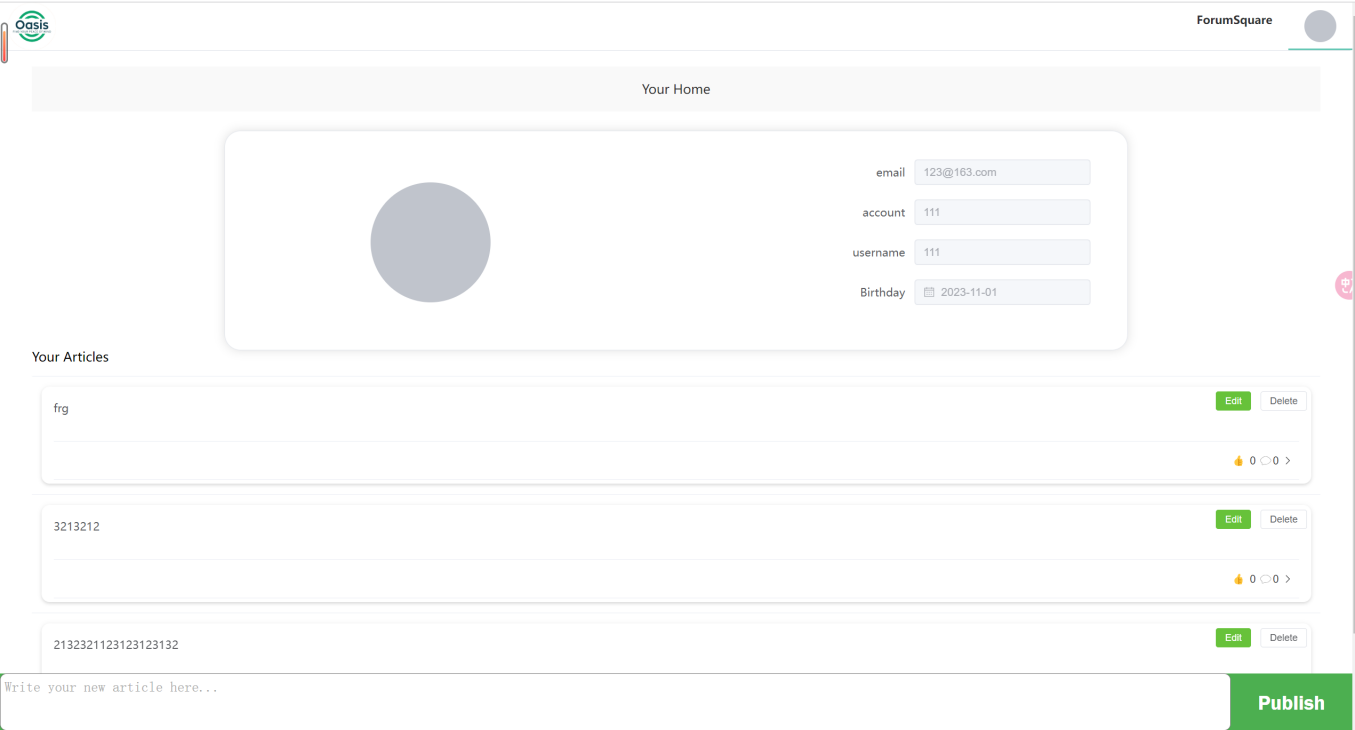


**User Personal Homepage:**

The User Personal Homepage is a user-centric interface designed to offer a personalized overview, including the user's avatar and basic information, and a neat list of articles. This design allows users to access and manage their personal data and content in a single, centralized view.

The user's avatar is prominently placed, usually in the center or top of the page, to immediately draw attention and reinforce identity recognition. Beside the avatar are basic details like email, account, username, and birthday, presented in clear fonts and proper layout for readability. The layout is both intuitive and convenient, allowing users to see all important information without scrolling.

Directly below the user information, we provide a concise list of articles. This list not only displays the titles of all articles published by the user but also allows them to manage their content through prominent "Edit" and "Delete" buttons. The like button is designed as an intuitive icon—a thumbs-up sign—enabling users to show appreciation for an article with a click. This simple interaction not only enhances connections among users but also provides positive feedback for authors. When the like button is clicked, the number beside the icon changes to visually confirm the action and update the like count in real time, showing authors and other users which content is most popular. A comment button next to each article leads users to the detailed page, where they can add their views below the article. To encourage expression, the comment area is designed as an open, friendly space, with a simple text box for entering thoughts. Once a comment is posted, it immediately appears below the article, creating a rich discussion thread with other community members. These buttons are designed to be intuitive and easy to understand, even for new users.



**Login Page:**

The login page features an intuitive dichotomous layout, offering emotional motivation on one side and ensuring functionality on the other. The left side of the page features the slogan "Life is a journey," accompanied by soothing curves and icons representing tranquility—our website's visual symbol, conveying that users' mental health journeys are respected and supported here. This design element aims to communicate the site's core philosophy: each person's life journey is unique and valuable.

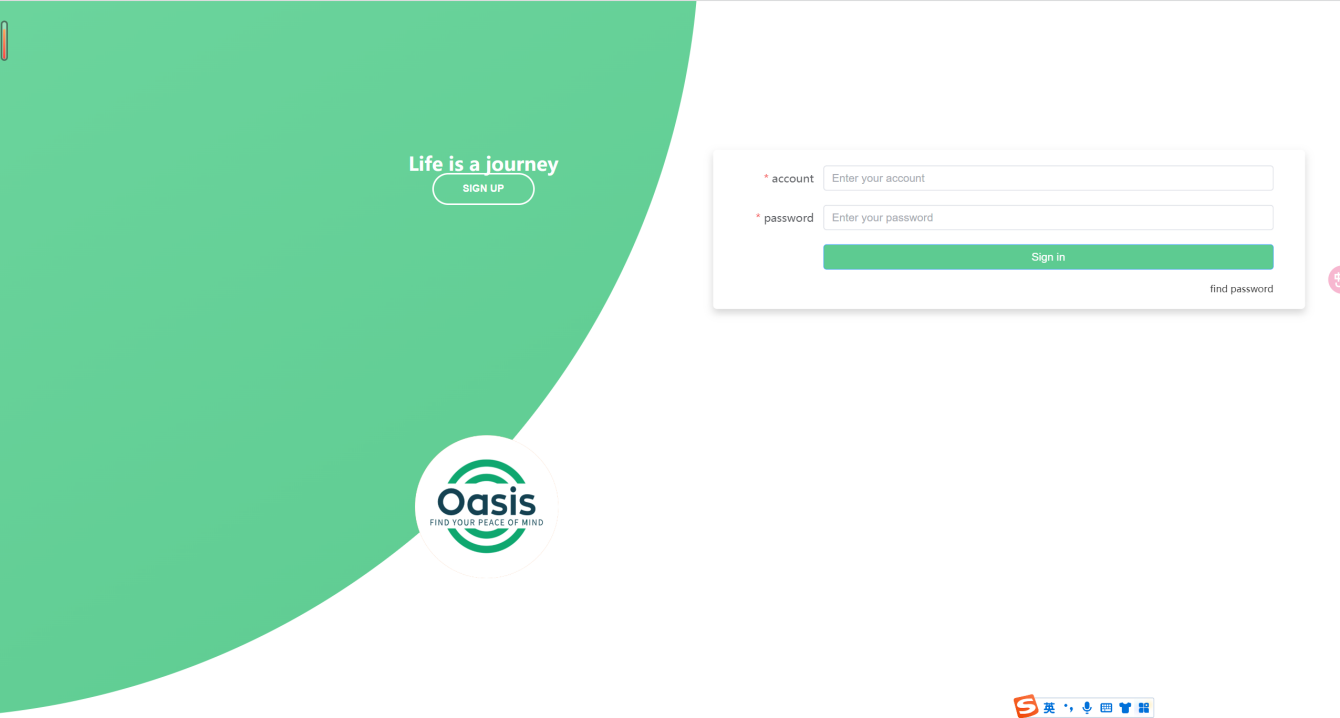
The slogan is not just text, its design integrates the tranquil atmosphere of the psychological forum, using soft colors and curves to alleviate any stress or anxiety users may have. The choice and layout of the slogan's font are carefully crafted to inspire positive emotions in users, setting a positive tone for their subsequent actions.

On the right side of the page, we have a streamlined login form, designed to provide a seamless login experience. The input fields are clearly marked, and the button design is prominent, ensuring that users can quickly find and operate them, even in a hurry. We chose a white background for this side to contrast with the emotional part on the left, guiding users' attention to the login process intuitively.

The overall design of the page considers users' varying skill levels. The login form's fields provide clear prompts, and if users forget their password, they can easily retrieve it through a prominent link. Additionally, for first-time visitors, we provide a clear registration option to ensure new users can easily join our community.

Feedback on button clicks is presented in real time through animations and color changes, like the "Sign up" button smoothly transitioning to the registration interface. These details enhance the page's interactivity and improve the overall user experience.

In designing this page, our goal was to provide a warm yet efficient login process, ensuring users feel welcomed and respected as they enter the doors of the psychological forum. We believe a good start is half the success, and the login page is where users begin their journey on the forum.



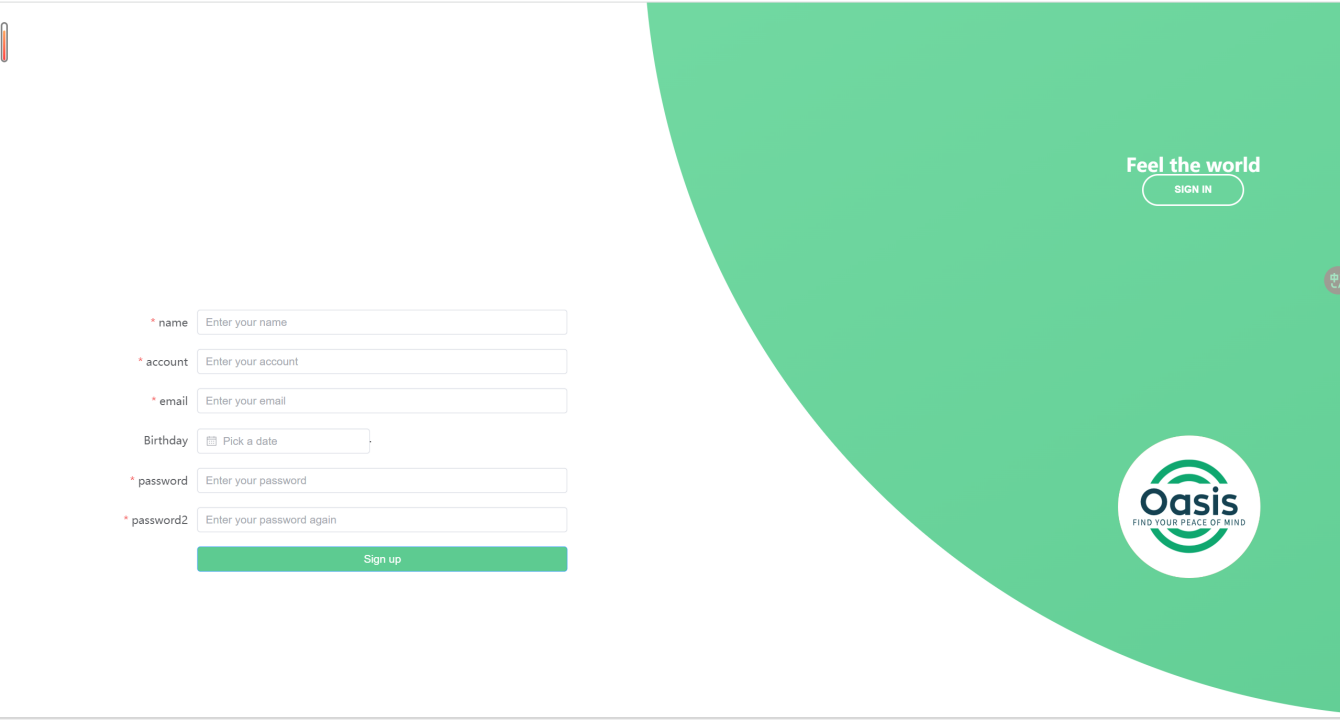
**Registration Interface:**

The registration page maintains visual consistency with the login page, keeping the same split layout to offer new users a clear, easy-to-navigate entry point. The left side of the page continues the gentle color tones and inspiring patterns of the login page, while the right side focuses on the registration form, making the overall experience coherent and harmonious.

The design of the registration form emphasizes user interaction and ease of use. We provide clearly labeled input fields to ensure users do not feel confused while entering information. Labels next to each field explain the required information, and the font size and color choices are optimized for readability. For information like birthdates, we use an intuitive date picker to simplify the date entry process.

At every step of filling out the form, we provide corresponding tips and suggestions, ensuring even website newcomers can complete registration smoothly. By combining engaging visual elements with functional elements, we have created a page that is both aesthetically pleasing and practical. The registration page is not just a place to fill out a form; it also conveys a message of welcoming new users to join this beneficial mental health forum community.

After completing registration, users receive clear feedback confirming their account creation. We also provide immediate email verification to ensure account security and allow users to start their activities on the forum right away.

****

**5.1.2Navigation Flow**

Our website navigation is implemented through a user-centered design principle, ensuring the transition from one page to another is both intuitive and natural. To provide a seamless user experience, our navigation process design focuses on reducing users' cognitive load, enabling them to easily find the content they desire on the site. Here are detailed descriptions of several typical navigation flows within the website:

From the Homepage to the Registration Page: The first guidance new visitors receive on the homepage is the prominent “SIGN UP” button, designed to immediately attract attention and guide users to register. The button, consistent with the overall design language of the website, is highlighted in color and style and is centrally located on the page. Upon clicking, users are instantly redirected to the registration page, with a smooth transition and no reload delay, providing immediate feedback.

From the Login Page to the User Personal Homepage: Registered users find a clear entry on the login page to enter their credentials. Security during login is ensured through the use of advanced encryption and authentication

technologies. Once authenticated, users are automatically redirected to their personal homepage, containing their personal information, published posts, and updates on community interactions. This process is designed to quickly integrate users into the community and start their social interactions.

Navigating between the Personal Homepage and Forum Plaza: The user's personal homepage serves as their “home” in Forum Plaza, from where they can directly access the Forum Plaza. This allows users to easily participate in discussions, view new content, or revisit their favorite posts, regardless of their intent.

Through the integration of these designs and functionalities, our website offers an intuitive and efficient navigation system, ensuring users have a smooth and pleasant experience while seeking information, engaging in social interactions, or managing personal content.

**5.2 Visual Design**

**5.2.1Color Scheme and Branding**

In our website, the color scheme plays a crucial role. It is not just a tool for visual attraction but also a language for conveying emotions and information. Our carefully selected color palette aims to create a comfortable and peaceful environment, fostering open and constructive dialogues about mental health.

Green, as the primary color of our color scheme, is chosen not only because it psychologically evokes calmness and relaxation, but also because it is culturally associated with health and growth. In addition to green, we have carefully selected complementary colors to enhance the main hue. For example, in design areas known as negative space, we use white. This not only reduces visual clutter but also provides a place for visual rest, making the design more refreshing and organized. White is the most commonly used color for this purpose, as it is the least intrusive, offering users a sense of clarity and openness. In web design, ample white space helps highlight more important elements, such as buttons, icons, or key information, while providing a sense of ease for the user's visual experience. Black, often used in web pages to draw attention and emphasize importance, represents trust, stability, and professionalism in color psychology. Black helps users identify which information they need to pay attention to while also conveying the brand's reliability and authority. These colors work together to enhance visual depth and establish distinctions between different parts of the website.



Our brand logo is designed to be simple yet profound, representing inner peace and mental harmony. This logo is not only easy to recognize but also perfectly aligns with our mission of supporting mental health and providing a healing space. The icon is a key component of our brand's visual identity system, playing a critical role in conveying core values and philosophy. In the icon, the green color scheme conveys a sense of calmness and freshness, perfectly matching the brand name "Oasis," symbolizing a haven for mental health and tranquility.

The logo's abstract wave shapes may symbolize calm waters or peaceful mental waves, echoing the brand name "Oasis," and conveying a haven-like image, offering support and shelter for mental health. This design is not only aesthetically pleasing but also rich in symbolism, easily creating emotional connections with users. The slogan "FIND YOUR PEACE OF MIND" directly converses with the brand's mission, encouraging users to find their inner peace. The chosen typography is simple and legible, also conveying the brand's friendly and inclusive voice. Overall, this icon balances aesthetics, functionality, and brand value communication. It is not just a logo but a medium carrying and conveying brand philosophy, building a powerful brand image through a combination of color, shape, and text.

**5.2.2 Typography and Iconography**

Font design is a carefully planned element on our website, as it largely determines the comfort of reading content and the overall perception of the site. In choosing fonts, we are not just selecting a style to display text but choosing a tool to convey brand voice and emotions. Our primary goal is to ensure text clarity and readability regardless of the device used to browse the website. For this, we chose fonts that perform well on both small sizes and large screens. The fonts maintain their shape and readability across different devices and resolutions, avoiding complex decorations in glyph design to ensure clarity when scaled. Furthermore, optimized letter spacing and line height enhance text readability, providing visual comfort for reading.

The style of the font is key to emotional expression. Our chosen fonts not only convey information but also establish emotional connections with users. Whether it's warm, encouraging articles or serious discussions, the selected fonts match the emotional tone of the content, enhancing the communication strength. The font choice reflects our brand personality - reliable, professional, and friendly. The fonts we use provide a cohesive visual language for the website, helping to strengthen brand recognition and establish the brand image in users' minds.

Each icon is designed to be simple and meaningful, allowing users to recognize its function without thought. For example, the like icon often uses an upward thumb pattern, while the comment icon might use a speech bubble shape. The search function is represented by the universally recognized magnifying glass icon. This intuitiveness ensures that users can quickly understand the intent and function of icons, even in the absence of textual explanations.

In the context of globalization, icon design considers cross-cultural accessibility. We ensure that the symbols used are appropriate globally, avoiding icons that might have different meanings in specific cultures, thereby preventing misunderstandings and cultural conflicts. Through this carefully designed icon system, our website offers a visually intuitive and culturally inclusive user interface, greatly enhancing users' navigation efficiency and overall satisfaction.

**5.3 User Experience (UX) Considerations**

**5.3.1Accessibility**

To ensure that all users, including those with visual impairments, can understand and enjoy the website content, we provide alternative text (alt text) for image content to cater to diverse user needs. Our website is designed with an intuitive navigation system, where consistent layout and familiar interface elements enable users to effortlessly find the information and services they need. A clear menu structure and logical page organization allow users to navigate without extra guidance. We provide explicit error messages and operational feedback to assist users in completing tasks correctly. For example, in case of form validation failure, we not only highlight the problematic fields but also provide succinct explanations to help users understand how to resolve the issues. Our interaction design focuses on predicting and responding to user behavior. Every click and selection made by the user receives prompt and clear feedback, whether through dynamic effects or page updates, ensuring users always know their actions have been successfully executed.

**5.3.2Responsiveness**

Our responsive design employs a fluid grid layout, meaning all page elements can scale according to different screen sizes. This layout ensures that the website maintains its content structure and aesthetics on any device, from small-screen phones to large-screen monitors, while keeping clarity and proportion intact. We have implemented lazy loading technology, so content only loads when users scroll to that part of the page. This reduces initial loading times and unnecessary data transfer. Our design adopts a modular approach, allowing individual components to be updated and improved independently of other parts. This approach not only makes maintenance easier but also allows the website to quickly adapt to new design trends. Our responsive design strategy is driven by user feedback. Regular user testing and feedback sessions help us identify areas for improvement, ensuring the design continuously meets the actual needs of users.

**6. Implementation Details**

**6.1 Technologies Used**

**Back-end Technology**

In our project, the core task of back-end technology is to ensure error-free inter

actions between the front-end and back-end, and that the back-end can correctly process requests from the front-end and communicate with the database effectively. Here are the key parts of our back-end technology implementation:

Using Django's Built-In SQLite Database: We have completed the creation of databases, including the storage of user information and article content. The design and implementation of these databases are crucial for supporting website functionalities.

User Registration and Login API Implementation: We successfully implemented back-end APIs for user registration and login, ensuring users can smoothly register and log in on our platform.

Data Processing and Management: For effective data processing and management, we utilized Django Rest Framework for data serialization, enhancing data processing efficiency. We used Django’s ORM for the integrity and efficient management of data, especially in handling comments and article data. Developed APIs for comments and likes functionalities using Django Rest Framework, ensuring back-end strategy consistency.

User Authentication and Verification Process: Implemented necessary authentication and verification processes to ensure the security of user registration and login.

Using django-cors-headers to Resolve Cross-Origin Issues: To handle cross-origin requests between the front-end and back-end, we used django-cors-headers.

Using Postman and Integrated Testing: Debugged back-end API endpoints with Postman tool and fixed some existing issues, ensuring the robustness and reliability of APIs.

**Front-end Technology:**

In our project, the core of frontend development focused on using the Vue3 framework for interface beautification and integrated testing, to enhance user experience and ensure seamless interaction between frontend and backend. Here are the key parts of our frontend technology implementation:

Vue Component Structure Organization: We created a new Vue.js project using Vue CLI and established a type-safe scripting environment to support efficiency and accuracy during development. By configuring Vue Router, we managed different routes within the application, ensuring smooth navigation for users across various sections.

Interface Beautification: We employed CSS for focused beautification of pages, improving layout intuitiveness and navigation usability. We installed and configured Element UI to use ready-made UI components. These components are not only feature-rich but also customized to match the forum's design style. For example, using element-ui plus components, we implemented pagination that is simple and easy to navigate.

Application of Axios: Axios was used to execute HTTP requests to the Django API. We implemented the creation, retrieval, and deletion of forum posts and comments. Using element-ui components, we designed sections to display user details. Through Axios, edited user information was sent to the backend API, achieving real-time updates of user data. Article Editing and Publishing: Users can easily edit and publish articles on their personal homepages. Axios handles the user's editing requests and manages the complete lifecycle of articles through POST, GET, PUT, and DELETE methods.

**6.2 Database Schema and Data Models**

In our project, the design and implementation of the database architecture and data models are one of the core tasks to ensure smooth interaction between front-end and back-end. Here are the main features of our database architecture and data models:

Database Creation and Configuration: We used Django's built-in SQLite database to create and manage the required data structures.

Two Primary Databases: We created two main databases - a User database and an Article database. The User database stores user registration and login information, while the Article database is used to store and manage content posted by users.

Data Models: The design of the data models focused on articles, user comments, and user information. The Article model includes title, content, and associated user interactions (like the number of likes). The Comment model is associated with articles and contains user information and comment content. The User model includes details such as username, password, and email. We successfully implemented back-end APIs for user registration and login. To efficiently process and transmit data, we utilized Django Rest Framework for data serialization. We also implemented necessary authentication and verification processes to ensure the security of user registration and login.

These implementations of database architecture and data models not only ensure that the back-end can correctly process requests from the front-end but also guarantee effective communication with the database. This provides a stable and reliable platform for users, both in terms of user management and content publication.

**6.3 Key Algorithms and Data Structures**

In our project, we focused on implementing key algorithms and data structures to enhance the website's performance and user experience. Here are the main technical details we implemented:

Article Sorting: We developed a specialized algorithm for efficient sorting and filtering of articles. This algorithm can display articles based on various criteria such as publication date, user likes (popularity), and user preferences.

Data Structure Design: In the design of data structures, we ensured that properties of articles are quickly accessible and updatable, supporting efficient sorting operations.

User Authentication: Our user authentication mechanism includes the encryption of passwords to ensure the security of user data. The encryption method used is designed to prevent password leakage or unauthorized access.

Without Page Refresh: To reduce page loading times and enhance user experience, we implemented dynamic content loading technology. This means that when users browse articles or other content, the webpage can update information without a full refresh.

Front-end and Back-end Collaboration: This dynamic loading involves close collaboration between front-end and back-end. The front-end is responsible for presenting the updated content, while the back-end processes and provides the required data in real-time.

Through the implementation of these key algorithms and data structures, our website is not only technically efficient but also superior in terms of user experience. These technical solutions form the foundation of our website's success, ensuring smooth operation and user satisfaction.

**7. Security Measures**

**7.1 Data Encryption**

We have employed robust encryption algorithms to process user passwords. These algorithms transform users' original passwords into a string of ciphertext that cannot be directly deciphered. This transformation is one-way, meaning that even if someone obtains the ciphertext, they cannot restore the original password.

To further enhance security, we added a salt value during the password encryption process. Salt is a randomly generated data combined with the user's original password before encryption. This method effectively prevents rainbow table attacks, ensuring that even if two users have the same original password, their encrypted results will be different.

The encrypted passwords are stored in our database, where we have ensured the security and access control of the database. The way these passwords are stored complies with industry security standards, ensuring that even in the event of a data breach, password information remains secure.

We also implemented a strong password policy, requiring users to use complex password combinations when creating an account, which helps to reduce the risk of being cracked.

**7.2 Authentication and Authorization**

In handling sensitive user information, we adopted UUID (Universally Unique Identifier) to identify users. Each user account has a unique UUID, which is used for internal identification and reference in the system, instead of directly using usernames or email addresses. This method increases the security of the system by reducing the risk of directly exposing sensitive user information.

By using UUIDs, we can manage user identities without exposing their actual account information. This design effectively enhances privacy protection and reduces the risk of data leakage. The use of UUIDs also enhances the anonymity of user interactions. Users’ activities on the forum are not directly linked to their real identity information, increasing their sense of security when using the platform.

We implemented an ID-based access control mechanism, where users access and operate on articles through their unique IDs. This mechanism avoids using article titles or other information that could potentially leak content. This ID-based access method enhances the security of content management. Only the correct ID can access the corresponding article, thereby preventing unauthorized access and content tampering. ID-based content access also optimizes database query efficiency. The database can retrieve information more quickly based on IDs, thereby speeding up content loading and response times. Through these authentication and authorization measures, our system provides dual protection in safeguarding user information and content security. These measures not only strengthen the system's security but also improve the quality of user experience, ensuring users' privacy and security while using the platform.

**7.3 Handling Sensitive Information**

We have implemented sophisticated access control mechanisms to protect sensitive information, such as users' personal profiles and the content of their published articles. These controls ensure that only authenticated and authorized users can access or modify this information. Our system also includes role-based permission allocation, ensuring users can only access sensitive data relevant to their roles. For instance, regular users cannot access other users’ private information, while administrators have broader access rights, enhancing the overall security of the system.

During the transmission of sensitive information, we use encrypted communication protocols to ensure that data is not intercepted or tampered with from the user end to the server end. We establish a secure connection between the client and server, ensuring all transmitted data is encrypted and secure. In some cases, sensitive information is encrypted in segments before transmission, providing an additional layer of security protection. This means that even if data is intercepted during transmission, it cannot be decrypted in its entirety without the corresponding keys.

Through these measures, our system provides multiple safeguards in handling users' sensitive information, from strict access control to secure data transmission. These security practices not only protect users' personal information and content security but also strengthen users' trust in our platform.

**8. Performance Optimization**

**8.1 Caching Strategies**

We have implemented caching on the front-end to store static resources such as CSS files, JavaScript files, and images. These resources are downloaded and stored during the user's first visit, reducing the loading time for subsequent accesses. On the back-end, we have implemented server-side caching for storing dynamically generated content and query results. In handling cache, we differentiated between dynamic and static content. Static content, like user interface elements, is cached for longer periods, while dynamic content, such as the latest articles published by users, is updated as per actual requirements. Cache dependency management was implemented to ensure that related caches are updated or invalidated synchronously when the underlying data is updated. By using caching, we significantly reduced the number of direct queries to the database, thereby reducing database load and response time. Our caching strategy effectively shortens data retrieval time, especially in high-traffic scenarios, significantly enhancing the website's responsiveness and user experience.

Through these comprehensive caching strategies, we have not only improved website performance but also optimized the user's browsing experience. These measures reflect our ongoing commitment to enhancing system efficiency and response speed, ensuring the website remains efficient even under high concurrency.

**8.2 Database Indexing**

We focused on creating indexes for key fields in database tables, especially those frequently used in searches and sorting. For example, fields like username and email in the user table, as well as publication date and author fields in the article table, have been indexed.

These indexes significantly speed up data retrieval. Indexes allow the database engine to quickly navigate to the required data, reducing the amount of data that needs to be traversed during queries. For complex query operations, particularly involving multiple table joins and large datasets, indexes significantly reduce processing time. We regularly assess and maintain our database indexes to ensure they remain effective and match current data usage patterns. Depending on the changes in system usage and user behavior, we adjust and optimize our indexing strategy. This includes adding new indexes or removing ones that are no longer effective. While indexes speed up query times, they also increase the overhead of database write operations. Therefore, we seek a balance between query efficiency and write performance in our index optimization.

These indexes play a key role in enhancing the overall performance of the database. Especially under high load or high concurrency, indexes ensure quick system response. By reducing data retrieval times, indexes indirectly enhance the user's interaction experience. Users experience significantly reduced wait times while using search, filtering, or browsing content. Through these comprehensive database indexing strategies, we ensure efficient data retrieval and optimized database operations. These measures reflect our relentless pursuit of system performance and user experience, ensuring that the system remains efficient and stable, even with large data volumes and complex queries.

**8.3 Code-Level Optimizations**

We adopted modular and object-oriented programming approaches, organizing our code into clear, logical structures. This not only improved the readability of the code but also simplified ongoing maintenance. When writing code, we paid special attention to performance considerations. For example, we chose more efficient data structures and algorithms to reduce CPU and memory usage. We implemented a regular code review process, where team members check each other's code to identify and correct potential issues and shortcomings.

In designing APIs, we focused on making the interfaces as efficient as possible. This included reducing the number of API calls, optimizing data formats and structures, and ensuring fast API response times. We optimized APIs to reduce unnecessary data transfer. For instance, by designing more precise query parameters and using pagination techniques, we only transferred the data that users actually needed.

Through these code-level optimization measures, we ensured the efficiency and maintainability of our code, enhancing the overall system's performance and stability. These measures reflected our commitment to technical quality, ensuring the long-term health and sustainable development of our software product.

9. Future Work

9.1 Identified Improvement Areas and Proposed Enhancements

**Interface Design and Usability Improvement:** Evaluate and improve the overall user interface of the website, ensuring user-friendliness, intuitiveness, and ease of use. Utilize user testing and feedback to optimize navigation, layout, colors, fonts, and other design elements to enhance the user experience.

**Mobile Adaptation:** With an increasing number of users accessing the website through mobile devices, ensuring adaptability is crucial. Ensure a good display on various devices and optimize responsive design.

**Fast Loading and Performance Optimization:** Optimize website loading speed, reduce page load times, and implement caching techniques and other performance optimization strategies to improve user satisfaction and retention.

**Rich Content and Personalized Services:** During the process of our website application, we found that although we have many psychological counseling sentences, the content lacks universality and applicability. Moreover, it cannot achieve the peak push of many self-media software, and there is a lack of online counseling services.

**Increase in Mental Health Content:** Provide more high-quality content on mental health, stress management, emotional regulation, etc., such as blog articles, expert advice, video news, etc.

**Online Counseling Services:** Consider adding online counseling features, allowing users to have real-time or scheduled consultations with mental health experts, providing more professional help and support to users.

**User Engagement and Reward Mechanisms:** Incentivize user participation through methods such as user points, badges, leaderboards, etc., to increase user stickiness and activity.

**Feedback and Improvement Mechanisms:** Establish user feedback channels, collect opinions and suggestions, promptly respond to user needs, and make corresponding improvements to continuously optimize the platform.

**9.2 Considerations for Scalability**

**9.2.1 Scalable Infrastructure:**

**Cloud Architecture:** Utilize cloud service providers such as AWS, Azure, or Google Cloud, leveraging their elastic computing and storage resources to flexibly adjust server resources according to traffic demands.

**Load Balancing:** Implement load balancing technology to effectively distribute traffic to multiple servers, improving website stability and performance.

**9.2.2 Modular and Loosely Coupled System Design:**

**Microservices Architecture:** Adopt a microservices architecture, dividing the system into small, independent services, each responsible for specific functions, making it easy to scale and maintain.

**API Design:** Design clear and easily understandable API interfaces, allowing different modules to operate independently, reducing coupling.

9.2.3 Database Optimization and Cache Strategy:

**Horizontal Scaling of Databases:** Choose databases that can scale horizontally to handle large amounts of user data and high concurrent access.

**Cache Mechanism:** Use caching technologies such as Redis or Memcached to reduce database pressure and improve response speed.

9.2.4 Resilient Design and Future Expansion:

**Fault-Tolerant Design:** Consider fault-tolerance measures such as backups and automatic recovery to ensure continuous website operation.

**Reserved Expansion Space:** Design the system with expansion space, considering the addition of new features and services in the future.

**9.2.5 Monitoring and Analysis:**

**Performance Monitoring:** Implement system performance monitoring, continuously track the website's operational status, and promptly identify and resolve potential issues.

**User Behavior Analysis:** Utilize analytics tools to collect user behavior data, understand user needs and behavior patterns, and provide data support for future decision-making.

**10. Summary**

**10.1 Key Findings Summary**

In today's fast-paced society, young people face unprecedented challenges and stress. Through the analysis of survey data and user feedback, we identified several key points:

**Diversified Sources of Stress:** Academic pressure, job competition, and global events (such as the COVID-19 pandemic) collectively contribute to the increased psychological stress among young people.

**Pervasive Negative Psychological States:** Survey results indicate that negative psychological states such as anxiety, depression, and feelings of loneliness are widespread among the young population, highlighting significant challenges to their mental health.

**Lack of Resources and Support:** Many young people lack adequate support and resources when facing mental health issues, leading to an increase in isolation and a sense of helplessness.

**Importance of Mental Health:** Mental health has become a crucial factor influencing our quality of life, happiness, and achievements, but its significance is often overlooked.

**Demand for Support and Community:** Users strongly express the desire for mental health support and community, hoping for a safe, supportive, and understanding environment for sharing and communication.

In summary, the mental health issues of young people are increasingly prominent, and the lack of support and resources exacerbates this problem.

**10.2 Acknowledgments**

We sincerely thank everyone who supported and assisted in this project. Throughout this journey, we received invaluable support from various sources, allowing this project to proceed smoothly.

Firstly, we want to express our gratitude to our team members. Your dedication and efforts made this project possible. Each person played a crucial role, contributing tirelessly to the success of the project.

We also want to thank our teachers and supervisors for answering our questions. Your expertise and unwavering efforts helped us overcome technical challenges and provided critical support for the smooth progress of the project.

Additionally, we appreciate those who provided opinions and suggestions. Your valuable feedback played a crucial guiding role in improving our project.

Finally, we want to thank everyone who supported our project. Your encouragement and support are the driving force behind our continuous progress.

In this regard, we extend our heartfelt thanks to everyone who supported and helped us.

Sincerely,

Group 4