

EmoGenius Crew

An Innovative Android Application

TEAM MUMBERS: WangYifan, LuYuhao, WangYanbo, ZhengBinyue
Zhangxindan, ZhengHaoyuan, Wuguohua, LiGuandong, ShiTianjun, FanJinzhen, WangZhaoyou, XueYiyang

Software Engineering

CONTENTS

- 1 Member division of labor
- 2 Project progress
- 3 Testing
- 4 Current problems
- 5 Next phase of our plan

01

Member division of labor

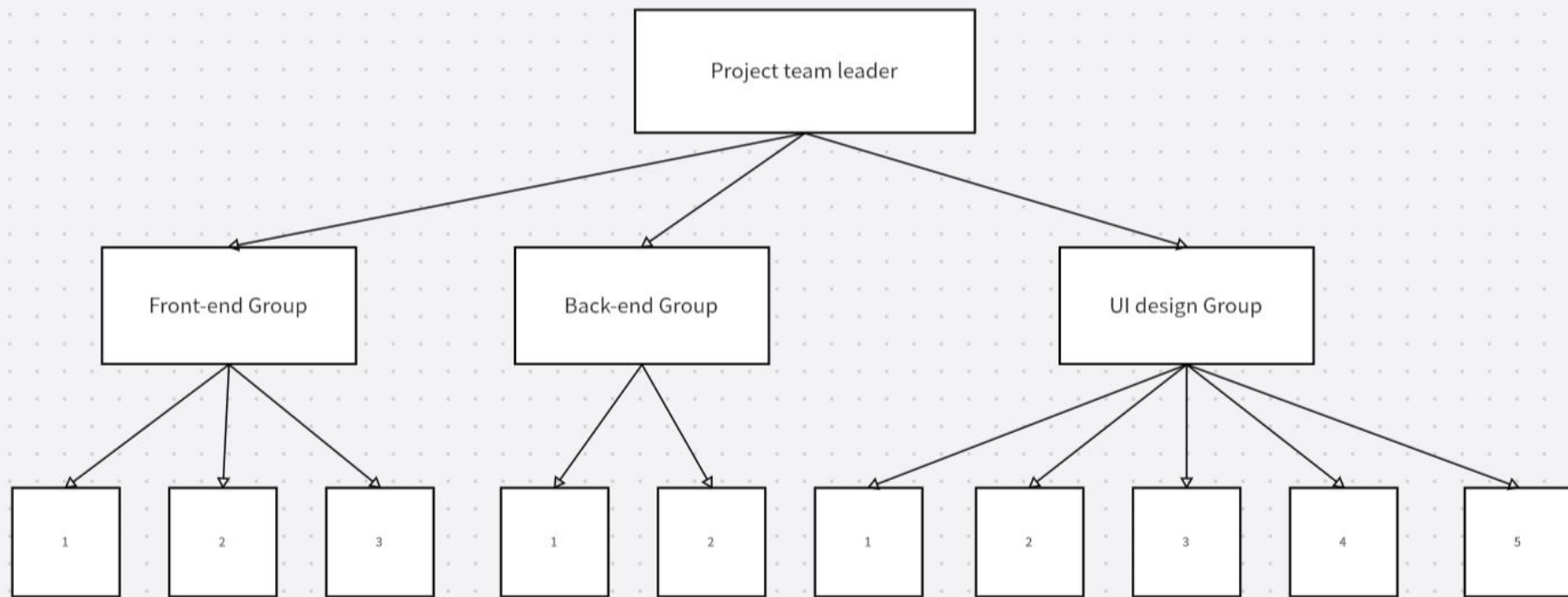
PART ONE

1

Member division of labor

<i>Student Name</i>	<i>Job Duties</i>
Wang Yifan	Front-end development, Front and Back end communication docking, Blog essay, Project Management, Software Testing
Wu Guohua	Back-end development, Front and Back end communication docking, Software Testing
Shi TianJun	Front-end development, Front and Back end communication docking, Software Testing
Zhang Xindan	UI Graphics Deployment & Design, Blog essay , Front-end development
Zheng Haoyuan	UI Graphics Design, Blog essay, Front-end development
Lu Yuhao	Back-end development, Front and Back end communication docking, Blog essay, Software Testing
Zheng Binyue	UI Graphics Design, Blog essay
Li Guandong	Front-end development, Blog essay
Wang Yanbo	UI Graphics Design
Wang zhaoyou	UI Graphics Design
Fan Jinzheng	Front-end development
Yiyang Xue	UI Graphics Design

1 Member division of labor



Teamwork





02

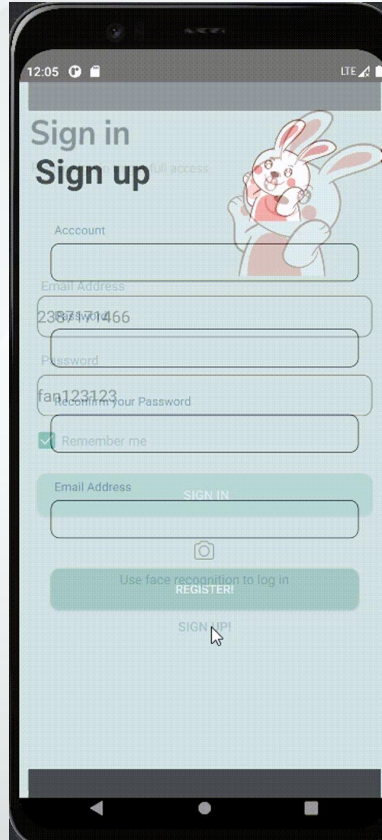
Project progress

PART TWO

Progress review

Function	Front-end completion	Back-end completion
Login/register module	Finished	Finished
Personal information module	Finished	Finished
Emotional diary module	Finished	Finished
Emotion forum module	Finished	Finished
Chatbot module	Finished	Finished
Emotion calendar module	Finished	Finished
Emotion Music Player	Unfinished	Finished
Voice Changer	Unfinished	Finished

I Login/Registration Module



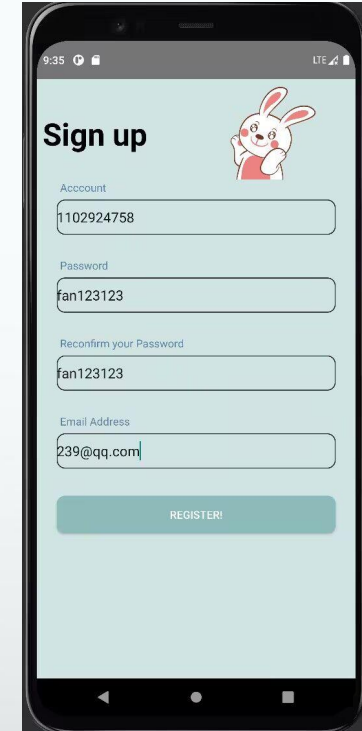
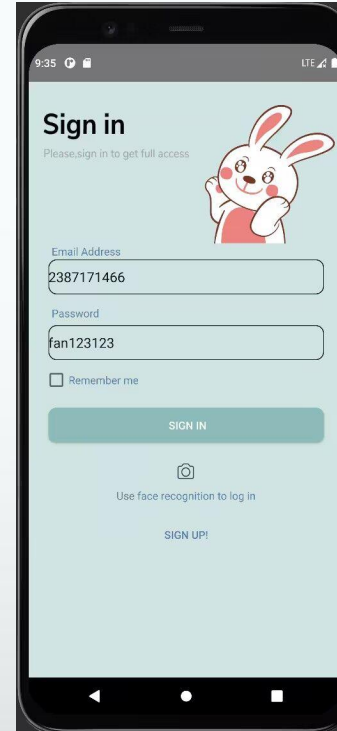
Users can input their username, password, confirmed password, and email.

If the confirmed password does not match the password, or if the account has already been registered, the registration will fail.

This module offers two methods of login and registration: **facial recognition** and **account-password**, which can better meet user needs and enhance the user experience.

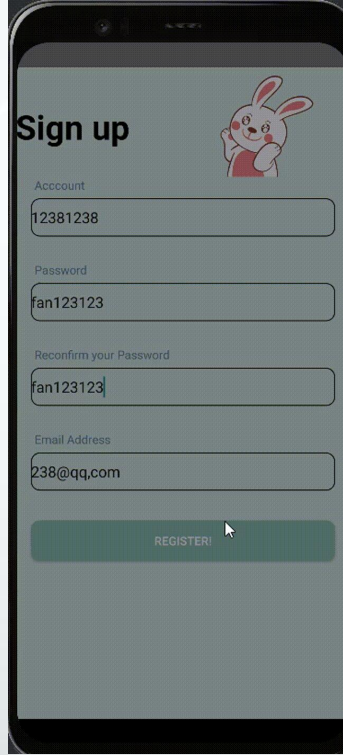
I Login module

- 1 **Account Password login**
Users can **register an account and set a password** to log in to the system. At the same time, the module also provides the function of confirming the password and remembering the password, ensuring that the password entered by the user is correct, and convenient to fill in the account information automatically when the next login.



Demonstration of completed functionality

I Login module



2

Face recognition login

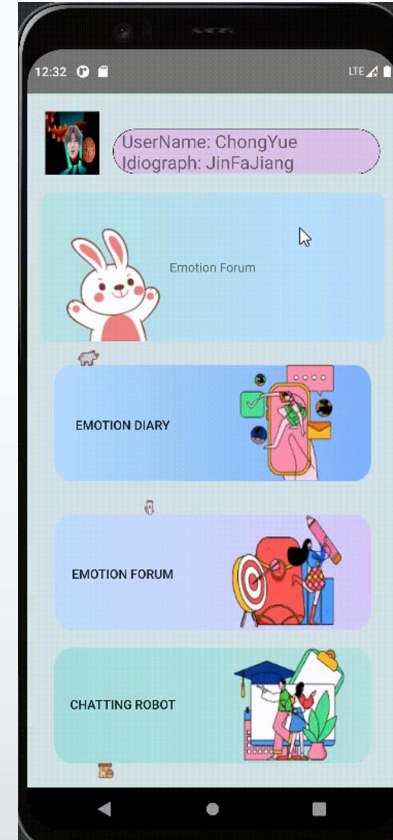
Users can log in through face recognition. The system can save the user's **face information**, and identify and log in directly at the next login, providing the user with another login method, and eliminating the step of entering the account and password.

II Personal information module

1

Function description

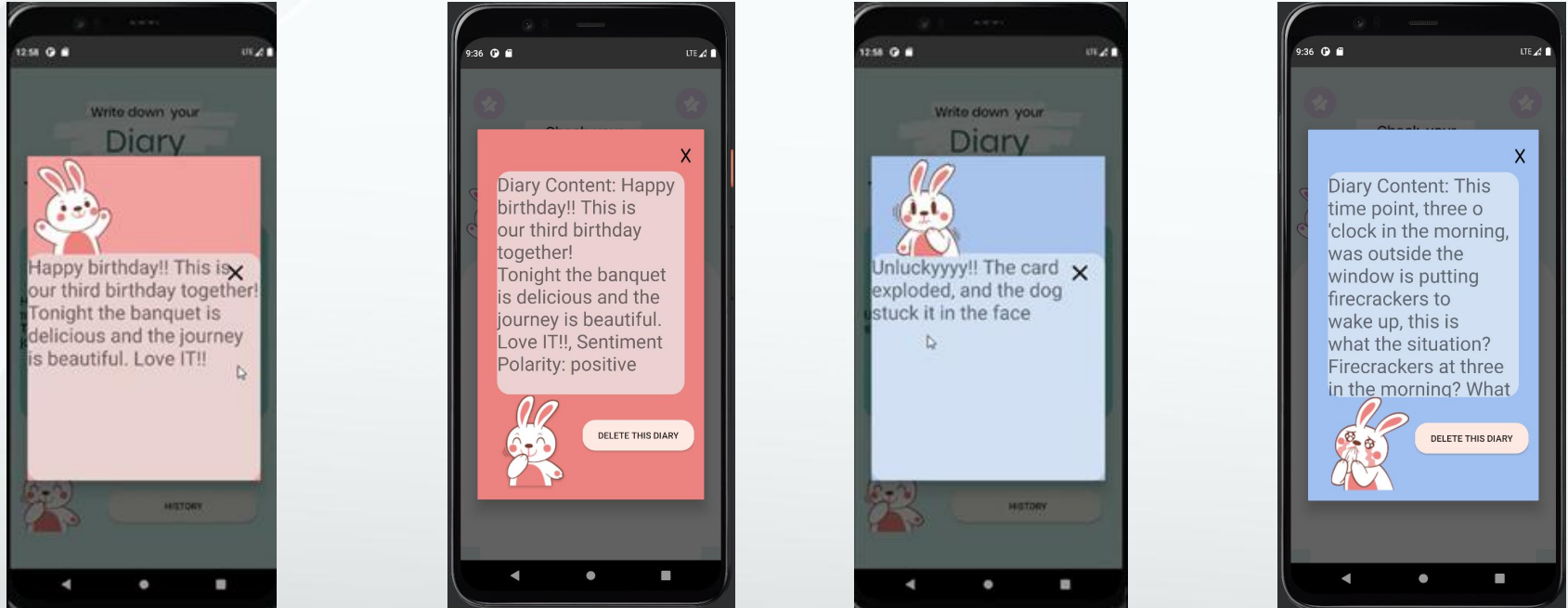
Users can freely modify their **profile picture, personal signature, name, age, and other personal data** on their personal homepage to better display their individuality and uniqueness. At the same time, changes to the profile picture and personal signature are synchronized on the app's home page, which allows the software to better display the user's personal style and interests.



Demonstration of completed functionality

III Emotion diary module

This module aids users in documenting their emotional experiences, offering feedback and suggestions. It includes features for **writing, saving, viewing, and deleting diaries**. The interface adapts to emotional polarities, and a **rich text editor** allows for expressive entries with formatting options like italic, bold, and image insertion.



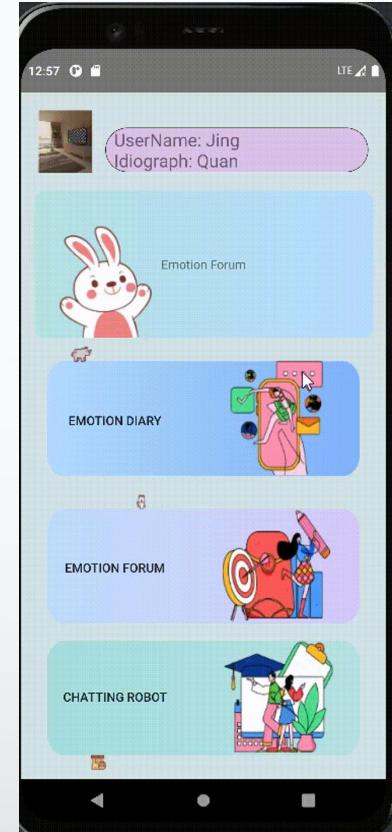
Demonstration of completed functionality

III Emotion diary module

1

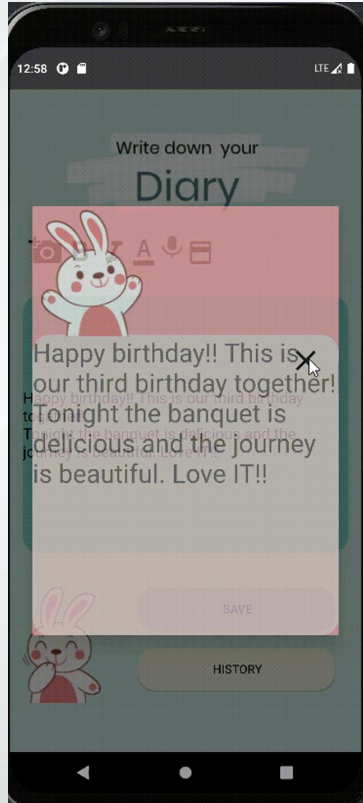
Writing a diary and displaying emotional advice feedback:

In the Emotion Diary module, users can write a diary and store their mood skillfully. Our project uses a **deep learning model** to analyze the emotional polarity of diary contents, which not only helps users record their emotional history, but also provides warm emotional feedback and intimate suggestions. In order to make emotional expression more colorful, we have developed rich text editing functions on the basis of text input, and users can set font styles and embed vivid photos in the diary to promote the depth management and vivid expression of emotions.



Demonstration of completed functionality

III Emotion diary module



2

Displaying history records:

The emotional diary helps the user record the emotional history will be saved in the history, here the user can **browse the diary content and the emotional state** at the time, the user can also delete the emotional record.

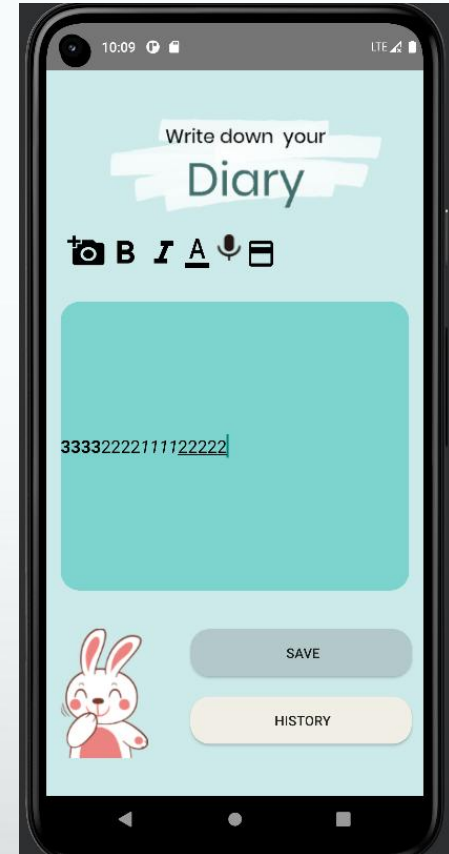
Demonstration of completed functionality

III Emotion diary module

3

Rich Text Editor:

Rich text editors can provide users with personalized modification of **text or images** (such as boldface, bold, add images, etc.).



Demonstration of completed functionality

IV Emotion forum module



The emotion forum is mainly divided into **positive emotion zone** and **negative emotion zone**, and the emotion in different zones is different, and the post content of the corresponding emotion polarity is displayed.

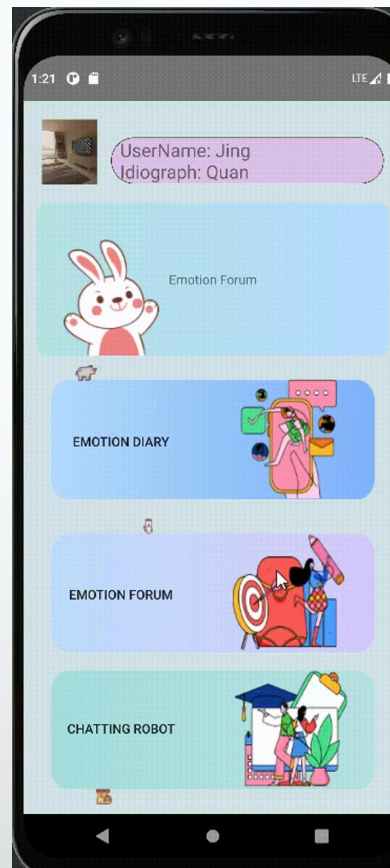
Demonstration of completed functionality

IV Emotion forum module

1

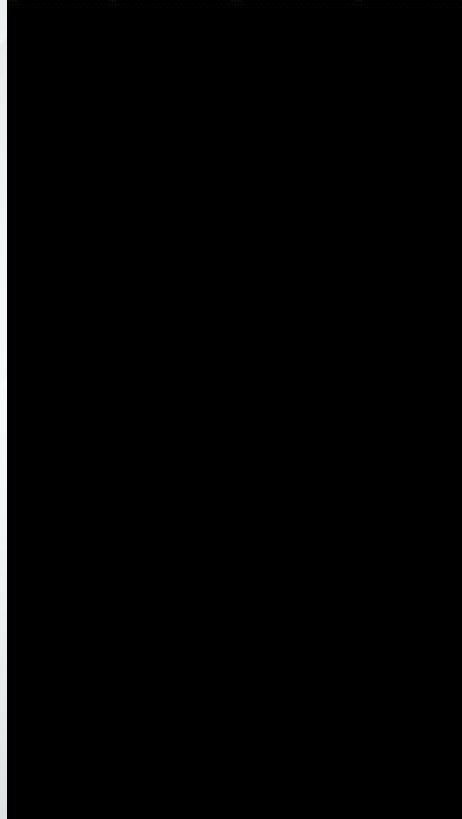
Function description

The emotional forum is mainly divided into positive and negative sections, displaying posts corresponding to emotional polarities. Users can post in the respective areas and engage in interactions such as liking and commenting within the forum, promoting user support, sharing, and emotional resonance, and creating a positive atmosphere, offering companionship and belonging to hearts in distress.



Demonstration of completed functionality

V Chatbot module



The chatbot module, utilizing **OpenAI's APIs**, is tailored for emotional communication through **parameter adjustments**. Users can personalize their experience by modifying the bot's thoughtfulness and selecting models, leading to enhanced user satisfaction and engagement through emotional connection and interaction.

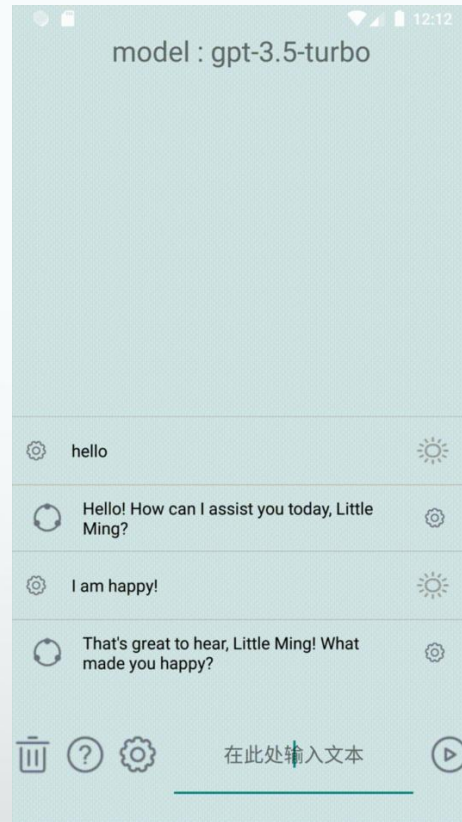
Demonstration of completed functionality

V Chatbot module

1

Function description

The chatbot module uses OpenAI's APIs to design a chatbot suitable for emotional exchange by **adjusting relevant parameters**. Users can also choose to adjust the robot's responsiveness and select different models to meet their conversational needs.



Demonstration of completed functionality

VI Emotion calendar module



1

Function description

The Emotional Calendar features display and recording functions. Upon entering the interface, the calendar for the current month is presented, with the current date highlighted in orange. Based on the user's emotions for the day, they can choose between happiness or sadness, and the color of the date will correspondingly change. Users can input daily diary entries and save them.

VI Emotion calendar module

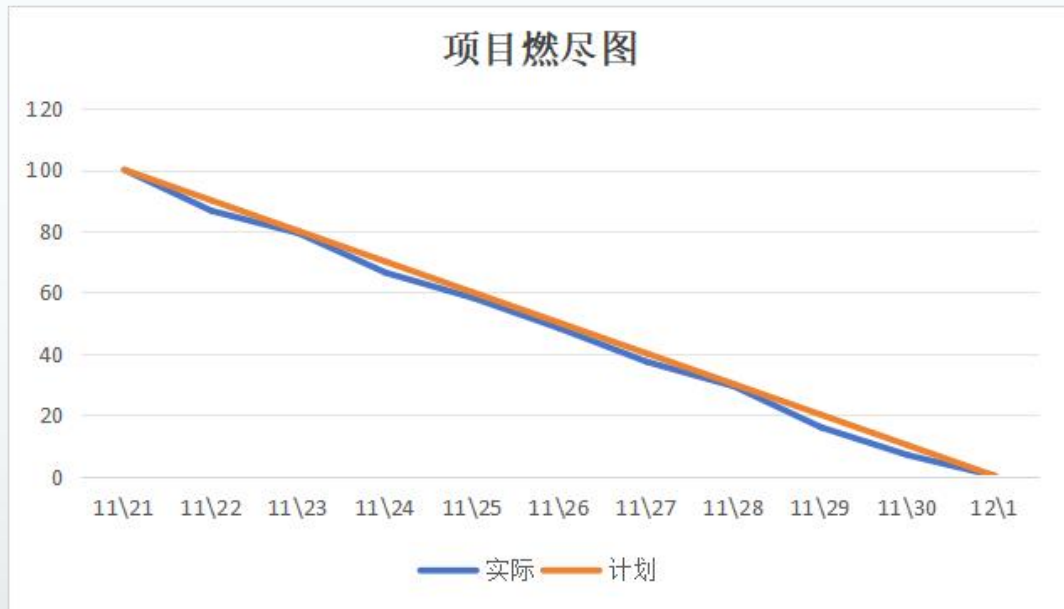
2

Function description

Users have ability to view daily or past emotional diaries, and the system provides search, add, and delete functions, facilitating easy management and review of emotional records. This design allows users to effortlessly document and track their emotional fluctuations, aiding them in gaining a better understanding of their emotional states.

Project burn-down chart

Due to the reasonable division of labor and cooperation ability, our group often made progress faster than expected and achieved good results. Finally, new features were added.





03

Testing

PART TWO

Testing contents

Page Layout and Design: Testing whether the page layout and design meet user expectations and if the UI displays correctly on different models of smartphones.

Functional Testing: Verifying the proper functioning of various software program features and confirming whether the execution results align with expectations.

User Experience Testing: Evaluating the user experience when interacting with the software program, focusing on aspects such as usability and navigation.

Stress Testing: Assessing the performance and stability of the software program under high concurrent requests.

Smoke Testing: Confirming the core functionality of the software program to determine if further detailed testing can proceed.

Regression Testing: Ensuring that defects, once fixed, do not introduce new issues, thereby safeguarding product quality.

Testing results

In summary, we conducted comprehensive testing on various modules of the project. Throughout the testing process, we continuously refined test cases and procedures, enhancing both testing efficiency and quality. Test execution adhered to pre-designed test cases to ensure standardized testing with repeatable results. Recording test outcomes, identified issues, and defects, and analyzing their root causes were integral to the process, facilitating prompt team fixes and improvements. The test results indicate that our software has several issues across multiple aspects, necessitating additional time for modification and enhancement, which constitutes our team's focus in the upcoming phase.

Testing results

25 Testing Totally

Front-end Testing

1. User Interface

1.1. Layout and Design

1.2. Color and Font

1.3. Image and Video

1.4. Animation and Transition

1.5. Responsive Design

1.6. Accessibility

2. Functionality

2.1. Form Validation

2.2. Navigation

2.3. Search Function

2.4. User Registration and Login

2.5. Data Entry and Retrieval

2.6. Error Handling

3. Performance

3.1. Load Time

3.2. Response Time

3.3. Throughput

3.4. Scalability

4. Security

4.1. Data Encryption

4.2. Session Management

4.3. Input Validation

4.4. Authentication

4.5. Authorization

5. Compatibility

5.1. Browser Support

5.2. OS Support

5.3. Mobile Device Support

5.4. Print Support

6. Localization

6.1. Language Support

6.2. Date and Time Format

6.3. Currency Format

6.4. Character Encoding

7. Usability

7.1. User Flow

7.2. Help and Documentation

7.3. Feedback Mechanism

7.4. User Satisfaction

8. Integration

8.1. API Integration

8.2. Database Integration

8.3. Third-party Service Integration

8.4. Payment Gateway Integration

8.5. Social Media Integration

9. Deployment

9.1. Environment Setup

9.2. Configuration Management

9.3. Backup and Restore

9.4. Monitoring and Logging

9.5. Disaster Recovery

10. Maintenance

10.1. Bug Tracking

10.2. Feature Request Management

10.3. Version Control

10.4. Release Management

10.5. Documentation Updates

11. Testing Tools

11.1. Selenium

11.2. Cypress

11.3. Jest

11.4. Mocha

11.5. Puppeteer

11.6. Protractor

11.7. Nightwatch

11.8. WebdriverIO

11.9. TestCafe

11.10. Playwright

12. Test Data

12.1. Test Data Generation

12.2. Test Data Management

12.3. Test Data Security

12.4. Test Data Backup

12.5. Test Data Archiving

13. Test Environment

13.1. Test Environment Setup

13.2. Test Environment Configuration

13.3. Test Environment Monitoring

13.4. Test Environment Maintenance

13.5. Test Environment Security

14. Test Results

14.1. Test Results Collection

14.2. Test Results Analysis

14.3. Test Results Reporting

14.4. Test Results Archiving

14.5. Test Results Security

15. Test Automation

15.1. Test Automation Framework

15.2. Test Automation Configuration

15.3. Test Automation Execution

15.4. Test Automation Maintenance

15.5. Test Automation Security

16. Test Case Management

16.1. Test Case Creation

16.2. Test Case Management

16.3. Test Case Execution

16.4. Test Case Reporting

16.5. Test Case Archiving

17. Test Plan

17.1. Test Plan Creation

17.2. Test Plan Management

17.3. Test Plan Execution

17.4. Test Plan Reporting

17.5. Test Plan Archiving

18. Test Strategy

18.1. Test Strategy Creation

18.2. Test Strategy Management

18.3. Test Strategy Execution

18.4. Test Strategy Reporting

18.5. Test Strategy Archiving

19. Test Process

19.1. Test Process Creation

19.2. Test Process Management

19.3. Test Process Execution

19.4. Test Process Reporting

19.5. Test Process Archiving

20. Test Framework

20.1. Test Framework Creation

20.2. Test Framework Management

20.3. Test Framework Execution

20.4. Test Framework Reporting

20.5. Test Framework Archiving

21. Test Suite

21.1. Test Suite Creation

21.2. Test Suite Management

21.3. Test Suite Execution

21.4. Test Suite Reporting

21.5. Test Suite Archiving

22. Test Run

22.1. Test Run Creation

22.2. Test Run Management

22.3. Test Run Execution

22.4. Test Run Reporting

22.5. Test Run Archiving

23. Test Report

23.1. Test Report Creation

23.2. Test Report Management

23.3. Test Report Execution

23.4. Test Report Reporting

23.5. Test Report Archiving

24. Test Summary

24.1. Test Summary Creation

24.2. Test Summary Management

24.3. Test Summary Execution

24.4. Test Summary Reporting

24.5. Test Summary Archiving

25. Test Conclusion

25.1. Test Conclusion Creation

25.2. Test Conclusion Management

25.3. Test Conclusion Execution

25.4. Test Conclusion Reporting

25.5. Test Conclusion Archiving

Back-end Testing

1. Database Testing

1.1. Data Integrity

1.2. Data Consistency

1.3. Data Security

1.4. Data Backup and Restore

1.5. Data Archiving

2. API Testing

2.1. API Endpoints

2.2. API Parameters

2.3. API Responses

2.4. API Errors

2.5. API Security

3. Server Testing

3.1. Server Configuration

3.2. Server Performance

3.3. Server Security

3.4. Server Maintenance

3.5. Server Archiving

4. Network Testing

4.1. Network Configuration

4.2. Network Performance

4.3. Network Security

4.4. Network Maintenance

4.5. Network Archiving

5. Security Testing

5.1. Security Configuration

5.2. Security Performance

5.3. Security Security

5.4. Security Maintenance

5.5. Security Archiving

6. Performance Testing

6.1. Performance Configuration

6.2. Performance Performance

6.3. Performance Security

6.4. Performance Maintenance

6.5. Performance Archiving

7. Usability Testing

7.1. Usability Configuration

7.2. Usability Performance

7.3. Usability Security

7.4. Usability Maintenance

7.5. Usability Archiving

8. Integration Testing

8.1. Integration Configuration

8.2. Integration Performance

8.3. Integration Security

8.4. Integration Maintenance

8.5. Integration Archiving

9. Deployment Testing

9.1. Deployment Configuration

9.2. Deployment Performance

9.3. Deployment Security

9.4. Deployment Maintenance

9.5. Deployment Archiving

10. Maintenance Testing

10.1. Maintenance Configuration

10.2. Maintenance Performance

10.3. Maintenance Security

10.4. Maintenance Maintenance

10.5. Maintenance Archiving

11. Testing Tools

11.1. Selenium

11.2. Cypress

11.3. Jest

11.4. Mocha

11.5. Puppeteer

11.6. Protractor

11.7. Nightwatch

11.8. WebdriverIO

11.9. TestCafe

11.10. Playwright

12. Test Data

12.1. Test Data Generation

12.2. Test Data Management

12.3. Test Data Security

12.4. Test Data Backup

12.5. Test Data Archiving

13. Test Environment

13.1. Test Environment Setup

13.2. Test Environment Configuration

13.3. Test Environment Monitoring

13.4. Test Environment Maintenance

13.5. Test Environment Security

14. Test Results

14.1. Test Results Collection

14.2. Test Results Analysis

14.3. Test Results Reporting

14.4. Test Results Archiving

14.5. Test Results Security

15. Test Automation

15.1. Test Automation Framework

15.2. Test Automation Configuration

15.3. Test Automation Execution

15.4. Test Automation Maintenance

15.5. Test Automation Security

16. Test Case Management

16.1. Test Case Creation

16.2. Test Case Management

16.3. Test Case Execution

16.4. Test Case Reporting

16.5. Test Case Archiving

17. Test Plan

17.1. Test Plan Creation

17.2. Test Plan Management

17.3. Test Plan Execution

17.4. Test Plan Reporting

17.5. Test Plan Archiving

18. Test Strategy

18.1. Test Strategy Creation

18.2. Test Strategy Management

18.3. Test Strategy Execution

18.4. Test Strategy Reporting

18.5. Test Strategy Archiving

19. Test Process

19.1. Test Process Creation

19.2. Test Process Management

19.3. Test Process Execution

19.4. Test Process Reporting

19.5. Test Process Archiving

20. Test Framework

20.1. Test Framework Creation

20.2. Test Framework Management

20.3. Test Framework Execution

20.4. Test Framework Reporting

20.5. Test Framework Archiving

21. Test Suite

21.1. Test Suite Creation

21.2. Test Suite Management

21.3. Test Suite Execution

21.4. Test Suite Reporting

21.5. Test Suite Archiving

22. Test Run

22.1. Test Run Creation

22.2. Test Run Management

22.3. Test Run Execution

22.4. Test Run Reporting

22.5. Test Run Archiving

23. Test Report

23.1. Test Report Creation

23.2. Test Report Management

23.3. Test Report Execution

23.4. Test Report Reporting

23.5. Test Report Archiving

24. Test Summary

24.1. Test Summary Creation

24.2. Test Summary Management

24.3. Test Summary Execution

24.4. Test Summary Reporting

24.5. Test Summary Archiving

25. Test Conclusion

25.1. Test Conclusion Creation

25.2. Test Conclusion Management

25.3. Test Conclusion Execution

25.4. Test Conclusion Reporting

25.5. Test Conclusion Archiving



04

Current problems

PART FOUR

Problem 1

The voice changing and music player functionalities cannot be deployed in the application.

Problem 2

Layout issues and image display anomalies during real-device debugging

Problem 3

Lack of UI displays for historical records in the emotional diary module and posts in the emotional forum.



05

Next phase of our plan

PART FIVE

Follow-up planning

Serial number	Function
1	The timbre modification and music player functionalities have been implemented and need deployment in the application.
2	Integrate third-party login features, including WeChat, QQ, and Weibo login.
3	Enhance emotional forum capabilities by adding post forwarding and sharing functionalities.
4	Improve the social performance of the application by developing friend features, enabling friend chat, and diary sharing.
5	Resolve layout issues and image display anomalies during real-device debugging.
6	Based on recent emotional records, provide a more diverse range of emotional recommendations.
7	Optimize the dataset for deep learning models to achieve more nuanced emotional classifications, such as happiness, sadness, and anger.
8	Beautify all interface UIs, learning and implementing UI displays for historical records in the emotional diary module and posts in the emotional forum.
9	Subsequently, collectively brainstorm to optimize the layout of other interface UIs.



Thank You

Software Engineering