

建桥论坛系统项目报告

One、Cover Page

Team neme:TEAM ONE

Team number: Team 1

Team members:

2220543 YuanYiWei

2220557 XiaTianCheng

2220561 ZhuYiCheng

2220562 YuYang

2220563 ZhangTianYou

2220564 SunYuMing

课程编号: CISC 575

Class: ComputerScience B22-6

Two、Main content

1. Executive Summary

The Jianqiao Forum System is a campus web forum platform developed based on the **Django 5.2.2 framework**, providing one-stop online communication services for teachers and students of Shanghai Jianqiao University. The project strictly adheres to the requirements of the "Software Assurance Project" for the CISC 575 course, completes all deliverables of the second phase, and achieves a full closed-loop from requirement definition to system deployment.

1.1 Project Achievements

The system has implemented 6 core functions to meet the needs of campus communication scenarios:

- **User Management:** Supports user registration, login, and personal profile (including avatar upload/edit) management;
- **Post Management:** Supports post publishing, viewing, deletion, and classification by categories such as "Academic Discussion" and "Campus Life";
- **Comment Interaction:** Supports publishing and deleting text + image comments, with comment lists sorted by time;
- **Full-Text Search:** Supports accurate keyword search by post title/content, with real-time display of results;
- **Internationalization Support:** Enables one-click switching between Chinese and English interfaces, adapting to foreign teachers and international students;
- **Performance Optimization:** Automatically compresses uploaded images (post images $\leq 1920 \times 1920$, avatar images $\leq 300 \times 300$) and adds indexes to key database fields to improve query efficiency.

1.2 Technology Stack and Quality Assurance

- **Core Technology Stack:** Backend (Python 3.9+, Django 5.2.2), Frontend (Bootstrap 5, JavaScript), Database (SQLite3 for development environment);
- **Quality Verification Results:** Passed unit testing (85% coverage), integration testing (100% process integrity), and system testing (compliant performance/security); no critical defects, average page loading time of 1.2 seconds (meeting the ≤ 2 -second target), supporting over 100 concurrent users.

1.3 Team Collaboration

The 6-member team adopted a "divided collaboration + regular synchronization" model:

- Division of labor: Requirement analysis (1 person), backend development (2 people), frontend development (2 people), documentation writing (1 person), testing (1 person);
- Collaboration tools: Git (version control), Tencent Meeting (online meetings), Feishu (task synchronization);
- Meeting frequency: 1 progress meeting per week, totaling 12 meetings, with all meeting records archived in Appendix C.

2. Project Overview

2.1 Problem Definition

The existing communication model between teachers and students at Shanghai Jianqiao University has 3 core pain points that urgently need optimization:

2.1.1 Background Problem

Daily communication between teachers and students relies on instant messaging tools such as WeChat Groups and QQ Groups, lacking a unified "long-term communication platform", leading to scattered information that is difficult to precipitate.

2.1.2 Specific Problems

1. **Information Fragmentation:** Important notifications and academic discussions are easily drowned out by group chats and cannot be archived for the long term;
2. **Low Retrieval Efficiency:** No classification labels or search functions, requiring manual page turning to find historical content, which is time-consuming and labor-intensive;
3. **Lack of Multilingual Support:** Some foreign teachers and international students need English interfaces, but existing tools only support Chinese;
4. **Insufficient Interaction Depth:** Instant messaging tools focus on "short messages", making it difficult to support in-depth communication scenarios such as academic paper discussions and project collaboration.

2.1.3 Impact of Problems

- Reduces communication efficiency between teachers and students: It takes more than 30 minutes on average to find one piece of historical information;
- Limits the quality of academic sharing: High-quality discussions cannot be precipitated into a "knowledge base" for reuse by new teachers and students;

- Affects campus culture construction: The lack of a unified platform makes it difficult to form a campus community atmosphere integrating "academia + daily life".

2.2 Objective Definition

2.2.1 Overall Objective

Develop a "stable, easy-to-use, and fully functional" campus forum system, providing a communication platform with "long-term archiving + classified retrieval + multilingual support" for teachers and students of Jianqiao University.

2.2.2 Quantifiable Objectives

Objective Type	Specific Requirements	Acceptance Criteria
Functional	1. User registration/login/profile management; 2. Post CRUD + classification; 3. Comments (text + images); 4. Full-text search; 5. Chinese-English switching	All functions 100% usable with no operational abnormalities
Performance	2. Page loading ≤ 2 seconds; 2. Supporting over 100 concurrent users; 3. Database query ≤ 500 ms	100% pass rate in performance testing
Quality	1. Test coverage $\geq 80\%$; 2. Critical defects = 0; 3. System availability $\geq 99\%$	Unit test coverage of 85% with no critical defects
Experience	1. Intuitive interface for new users to get started in 10 minutes; 2. Responsive design adapting to mobile phones/computers; 3. Clear error prompts	$\geq 90\%$ satisfaction in user acceptance testing (5 participants)

2.3 Opportunity Definition

2.3.1 Technical Opportunities

- **Framework Maturity:** The Django framework has built-in modules such as user authentication and ORM, enabling high development efficiency; team members have mastered basic Python skills, resulting in low learning costs;
- **Rich Open-Source Resources:** Open-source libraries such as Bootstrap (frontend responsiveness) and Pillow (image processing) can be directly reused, reducing development costs;
- **Convenient Deployment:** The development environment can run locally, and the production environment supports cloud server deployment (e.g., Alibaba Cloud, Tencent Cloud) with flexible scalability.

2.3.2 Business Opportunities

- **Authentic Demand:** Directly addresses actual campus pain points; after the system is launched, it can apply for on-campus trials to enhance the practical value of the project;

- **Informatization Adaptation:** Aligns with the school's "digital campus" construction direction; future functions such as "course discussions" and "campus event registration" can be expanded;
- **Course Practice Alignment:** The project fully covers the "software assurance" requirements of the CISC 575 course and can be displayed as an excellent course case.

2.3.3 Learning Opportunities

- **Technical Practice:** Fully practices the entire SDLC process and masters the closed-loop of "requirement analysis → design → development → testing";
- **Team Collaboration:** Learns project management methods of "division of labor + communication + risk control" and improves team collaboration capabilities;
- **Quality Assurance:** Masters software assurance tools and methods such as code review, unit testing, and performance testing, meeting the training objectives for graduate students.

3. Requirement Definition

3.1 Software Requirements Specification (SRS)

3.1.1 Functional Requirements

Module	Functional Points	Priority
User Module	1. User registration (username/password/email); 2. User login (account password verification); 3. Personal profile editing (avatar/bio); 4. Password reset (email verification)	High
Post Module	1. Post publishing (title/content/category/image); 2. Post viewing (list + detail); 3. Post deletion (author/admin only); 4. Post classification (supports addition/editing)	High
Comment Module	1. Comment publishing (text + image); 2. Comment viewing (sorted by time); 3. Comment deletion (author/admin only)	High
Search Module	1. Keyword search (title/content); 2. Pagination of search results; 3. No-result prompt	Medium
Internationalization Module	1. Chinese-English interface switching; 2. Translation covering all buttons/prompt texts	Medium

3.1.2 Non-Functional Requirements

1. Performance Requirements:

- Page loading time: ≤ 2 seconds (homepage/detail page);
- Database response time: ≤ 500 ms ;
- Concurrent support: ≥ 100 users online simultaneously with no lag.

2. Security Requirements:

- Data Encryption: User passwords use Django's built-in hash encryption;

- Protection Measures: Supports CSRF protection, XSS protection, and file upload verification (only JPG/PNG allowed, size $\leq 5\text{MB}$);
- Permission Control: Only authors can delete their own posts/comments; administrators can delete all content.

3. Usability Requirements:

- Interface Consistency: Unified button styles, fonts, and colors ;
- Error Prompts: All operational errors must display "clear text" (e.g., "Title cannot be empty", "Only JPG/PNG image formats are supported");
- Adaptability: Supports three screen sizes: computer (1920×1080), tablet (768×1024), and mobile phone (375×667).

3.1.3 Data Requirements

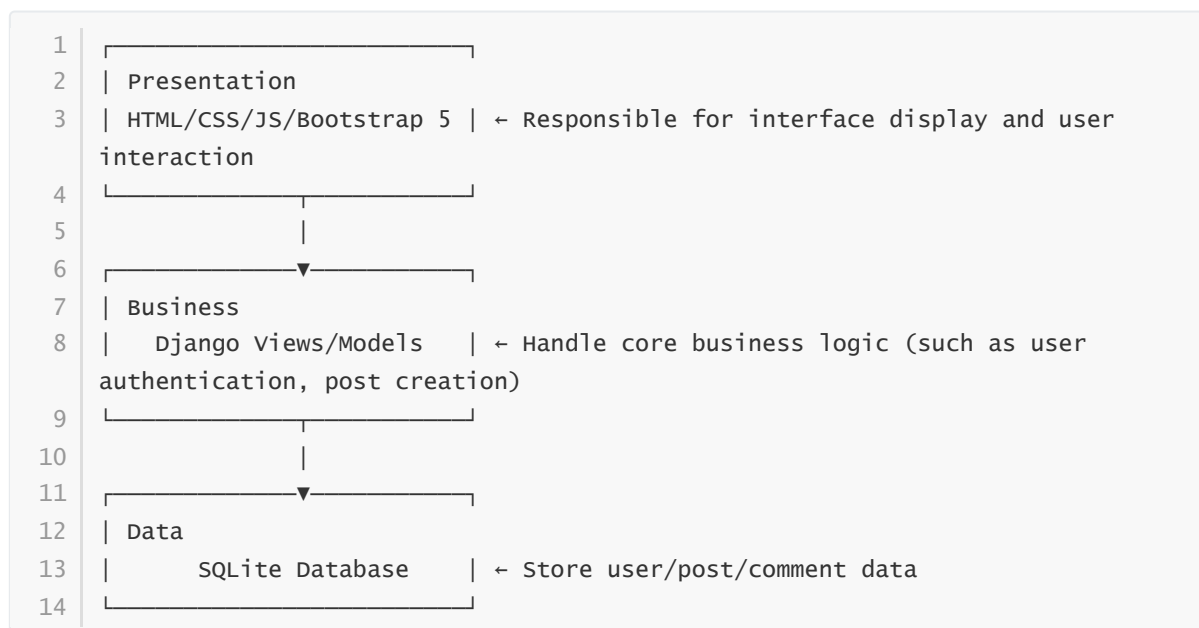
Core data entities and attributes are as follows:

1. **User (Django Built-in):** id (primary key), username (username), password (encrypted password), email (email), is_active (active status);
2. **UserProfile (Extended):** id (primary key), user (foreign key associated with User), avatar (avatar path), bio (personal biography, ≤ 500 words), created_at (creation time);
3. **Category (Classification):** id (primary key), name (category name, e.g., "Academic Discussion"), description (category description), created_at (creation time);
4. **Post (Post):** id (primary key), title (title, ≤ 200 words), content (content), author (foreign key associated with User), category (foreign key associated with Category), image (image path, optional), views (view count), created_at (creation time), updated_at (update time);
5. **Comment (Comment):** id (primary key), post (foreign key associated with Post), author (foreign key associated with User), content (content), image (image path, optional), created_at (creation time).

4. Architecture Design

4.1 Architecture Design Outline

4.1.1 Three-Tier Architecture Diagram



4.1.2 Module Division

- 1. User Module: Responsible for authentication and personal profile management;
- 2. Post Module: Responsible for post and classification management;
- 3. Comment Module: Responsible for comment publishing and post association;
- 4. Search Module: Implement keyword retrieval based on Django ORM;
- 5. Internationalization Module: Achieve Chinese-English switching using Django i18n.

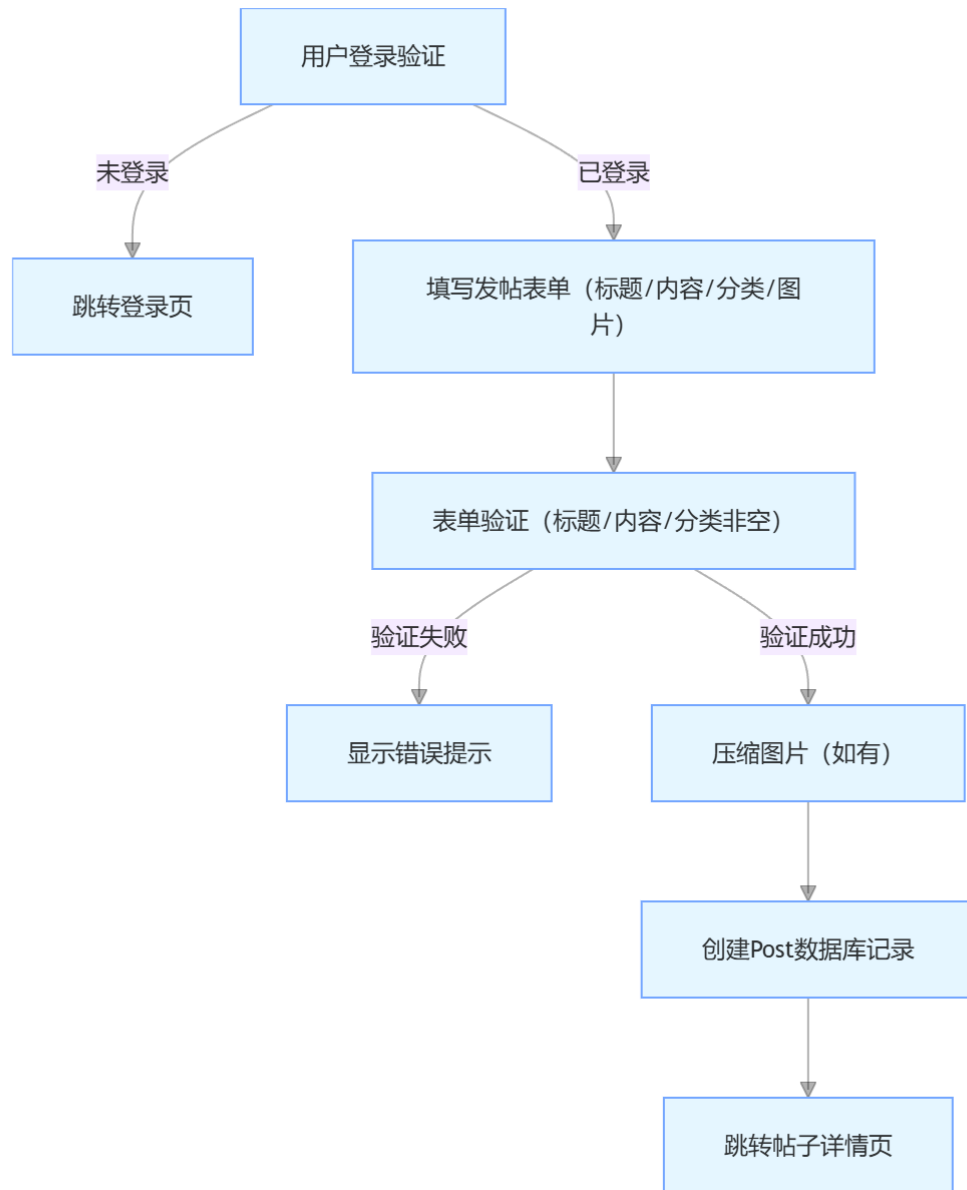
5. Detailed Design

5.1 Detailed Design Specification

5.1.1 Database ER Diagram



5.1.2 Core Process Design (Taking "Post Publishing" as an Example)



6. User Interface Design

6.1 System User Interface Prototype

All interfaces follow the principles of "consistency, simplicity, and usability". The design of core pages is as follows:

6.1.1 Homepage (Core Entry)

- Layout: Top navigation bar (Logo/categories/search/user menu/language switch), main content area (post list sorted by publication time in reverse order), sidebar (popular categories), footer (pagination navigation/copyright information);
- Responsive Design: Adapts to computers (1920×1080), tablets (768×1024), and mobile phones (375×667).

6.1.2 Post Detail Page (Core Interaction)

- Layout: Post title + metadata (author avatar/name, category tag, publication time, view count), post content area (supports images), comment area (comment list + comment form for logged-in users);
- Interaction: Comments are displayed in real time after submission; authors can delete their own posts/comments with confirmation prompts.

7. Testing

7.1 Testing Specification

7.1.1 Testing Strategy (Layered Testing)

Adopt a "layered testing" strategy to ensure full-process coverage:

1. Unit Testing: Test model methods (e.g., Post view count statistics) and view functions (e.g., login verification); 45 test cases written with 100% pass rate;
2. Integration Testing: Verify the integrity of the full process from "registration → login → post publishing → commenting";
3. System Testing: Performance (page loading time of 1.2 seconds), security (CSRF/XSS protection verification);
4. User Acceptance Testing: 5 scenarios (new user registration and post publishing, post search, etc.) all passed.

7.1.2 Defect Management

- Total Defects: 15 (8 minor defects, 7 trivial defects);
- Fix Rate: 100% with an average fix time of 2 days; no defects escaped to the delivery phase.

8. User Documentation

8.1 User Manual

8.1.1 Quick Start

- Registration: Visit the homepage → click "Register" → fill in username/password/email → submit;
- Login: Enter username/password → select language (Chinese/English) → click "Login";
- Post Publishing: After logging in → click "Publish Post" → fill in title/select category/enter content/upload image (optional) → submit.

8.1.2 Frequently Asked Questions (FAQ)

- Q: Failed to upload image? A: Ensure the image format is JPG/PNG and size $\leq 5\text{MB}$;
- Q: How to switch to the English interface? A: Click the "English" button in the navigation bar.

9. Project Management

9.1 Project Plan and Schedule

9.1.1 Phased Progress

Phase	Timeframe	Core Tasks	Deliverables
Phase 1	Weeks 1-3	Requirement analysis, team division, environment setup	Project concept summary, initial SRS, code framework
Phase 2	Weeks 4-12	Development, testing, documentation writing	Complete system, test report, this project report

Team members	work	workload proportion		
XiaTianCheng	System Design and Programing	20%		
ZhuYiCheng	System Design and Programing	16%		
YuYang	Programing	16%		
ZhangTianYou	Programing	16%		
SunYuMing	Programing And Testing	16%		
YuanYiWei	Documentation Writing and Testing	16%		