Team Member Weekly Updates

张茁含睿:

Multi-module system architecture sketch design and optimization.Responsible for sorting out core modules such as front-end, back-end, model, task scheduling, database, deployment platform. Based on discussion feedback, the architecture diagram is optimized into a hierarchical and logical diagram to facilitate team members to understand.

陶子芾:

Over the last two weeks, I have been primarily focused on backend model integration and testing. Here's a detailed breakdown of my work:

1. Model Deployment & Setup
    - Successfully deployed the open-source detection model (from GitHub) to our local development environment.
    - Configured the necessary dependencies and resolved compatibility issues to ensure smooth execution.
    - Verified that the model runs as expected with sample input data.
2. Debugging & Performance Testing
    - Conducted extensive debugging to identify and fix issues related to model inference, input/output formatting, and latency.
    - Performed integration testing to ensure the model works seamlessly with our backend framework.
    - Optimized model performance by adjusting parameters and improving data preprocessing steps.

黄俊哲

Over the last two weeks, I've led the requirements analysis phase to solidify our system's functional and behavioral specifications. Here's a detailed breakdown of my contributions:

1. Use Case Diagram Development

- Conducted stakeholder interviews to identify core system interactions
- Designed and finalized comprehensive use case diagrams covering:
  - Primary user workflows (e.g., news detection, reporting)
  - Exception handling scenarios
- Validated diagrams through team reviews to ensure alignment with business goals

2. Activity Diagram Specification
   - Mapped end-to-end processes for critical workflows (e.g., "User submits news for analysis")
   - Defined decision points, parallel actions, and system responses
   - Collaborated with frontend/backend teams to identify edge cases needing special handling
3. Requirements Documentation
   - Translated diagrams into written functional/behavioral requirements
   - Prioritized features based on technical feasibility and user impact
   - Resolved 5+ requirement ambiguities through cross-team discussions

胡家豪:

Over the past two weeks, I have focused on building core backend infrastructure and made significant progress in the following areas:

1. MySQL Database Design and Initialization

Requirement Analysis & Data Modeling
After thoroughly analyzing business processes and functional module requirements, I completed data modeling for the database. Using ER diagrams, I systematically analyzed and abstracted core entities in the system, defining primary/foreign key relationships and constraints between them.

Database Schema Design
Using MySQL as the relational database management system, I designed table structures for each business module, carefully selecting field types, default values, NOT NULL constraints, unique indexes, and composite primary keys to ensure data integrity,

scalability, and query efficiency. Additionally, I added necessary comments to key fields to improve readability and maintainability.

Initialization Script Development & Execution
Developed database initialization scripts covering:

- Database and table creation
- Field annotations
- Initial data insertion
  The scripts support idempotent execution, enabling quick environment setup during development, testing, or deployment. After completion, I validated the database structure and data consistency through local and development environment testing.

2. Backend System Development Based on Flask Framework

Project Structure Planning
Built the foundational backend project structure using Python Flask, adopting a modular design approach with the following key directories:

- models (data models)
- routes (API endpoints)
- services (business logic)
- config (configuration management)
- utils (utility libraries)
  This ensures clear code organization, separation of concerns, and scalability for future maintenance.

Database Connection & Validation
Implemented database connection configurations and health check mechanisms, followed by connectivity testing with MySQL to ensure the backend service can successfully establish and operate database connections upon startup.

Initial API Template Setup
Developed foundational API templates, including:

- /ping (health check endpoint)
- Database status verification API
- Mock user login API

These implementations validate the functionality of the backend framework and routing logic.