Biweekly Report

Name: Jingxiao Han

Period: April 21 - May 11, 2025

Project: Smart Maintenance Platform for Aero Engine Industrial Equipment

Week 1: Non-Core Function Development Initiation (April 21-27, 2025)

During this week, I began implementing the non-core functions of our predictive maintenance platform. My specific contributions included:

1. **Monitoring Center Development** (April 21-27)
    - Implemented real-time device health status monitoring functionality
    - Created energy consumption monitoring components
    - Developed parameter visualization mechanisms
    - Established data refresh mechanisms and caching strategy
2. **Alert System Development** (April 25-27)
    - Created alert viewing interface with filtering capabilities
    - Implemented alert confirmation workflow
    - Developed alert categorization and prioritization logic

Time spent: 20 hours

Week 2: Integration Testing and Function Completion (May 5-11, 2025)

After the holiday break, this week was dedicated to completing non-core function development and conducting comprehensive integration testing:

1. **Non-Core Function Completion** (May 5-7)
    - Completed Report Generation Module with template system
    - Finalized alert management functionality with notification services
    - Implemented data export features with multiple format support
    - Added final unit tests and documentation
2. **Core System Integration Testing** (May 7-11)
    - Executed comprehensive integration test suite covering all system interfaces
    - Verified API compatibility and data flow between components
    - Tested authentication and authorization across system boundaries

Time spent: 36 hours

Completed WBS Items

- 4.3.2 Non-Core Function Development with Unit Test (WBS Item, 28 hours) - 100% Complete
- 5.1 Core System Integration Testing (WBS Item, 28 hours) - 100% Complete

Challenges & Solutions

The main challenge I faced was implementing efficient real-time data monitoring while maintaining acceptable system performance. This required balancing update frequency with server load. I addressed this by:

1. Implementing a smart polling mechanism that adjusts frequency based on data volatility
2. Creating a layered caching strategy to reduce unnecessary data transfers

Next Steps

1. Support system testing activities
2. Assist with user acceptance test preparation
3. Develop deployment verification tests

Total Hours Worked

Total hours for this reporting period: 56 hours