



南開大學
Nankai University

计算机与网络空间安全学院
软件工程大作业

海洋牧场监测可视化系统

2113285 吴静

2111408 周钰宸

2113997 齐明杰

2113824 杨浩甫

2110803 谢雯菲

学院：计算机与网络空间安全学院

指导教师：刘健

2024 年 10 月 21 日

目录

1 项目概述	3
1.1 开发背景	3
1.2 项目目标	3
1.2.1 功能性目标	3
1.2.2 非功能性目标	3
1.3 开发环境	4
1.4 可行性分析	4
1.5 项目计划	5
1.5.1 时间节点	5
1.5.2 约束条件	5
1.6 项目完成情况	5
2 需求分析与系统设计	6
2.1 需求分析	6
2.1.1 功能需求	6
2.1.2 数据需求	7
2.1.3 运行系统需求	8
2.2 系统设计	8
2.3 详细设计	11
2.3.1 界面用例设计	11
2.3.2 用户流程设计	13
2.3.3 系统架构	15
2.3.4 安全性设计	16
2.4 数据库设计	17
2.5 UI 设计	20
3 系统测试	20
3.1 测试环境	20
3.2 功能测试	21
3.2.1 注册功能	21
3.2.2 登录功能	21
3.2.3 数据分析与可视化模块	24
3.2.4 报警与通知模块	29
3.2.5 智能化模块	32
3.2.6 用户中心	34
3.2.7 用户管理模块	37
3.3 性能测试	37
4 项目管理	38
4.1 参与人员及分工	38
4.2 项目进展记录	39

4.3 项目管理工具	40
5 用户手册	40
5.1 引言	40
5.1.1 编写目的	40
5.1.2 项目背景	40
5.1.3 术语和缩略词	41
5.2 软件概述	41
5.2.1 软件目标	41
5.2.2 软件功能	41
5.2.3 软件性能	41
5.3 软件使用说明	42
5.3.1 安装	42
5.3.2 使用	42
5.3.3 出错与恢复	42
5.3.4 求助查询	42
5.4 运行说明	42
5.4.1 运行步骤	42
5.4.2 操作命令一览表	43
5.5 用户操作示例	43
5.5.1 注册	43
5.5.2 登录	44
5.5.3 查看首页	44
5.5.4 养殖管理	46
5.5.5 大数据分析	47
5.5.6 AI 识别分析	48
5.5.7 数据上传下载	50
5.5.8 发送与接收预警信息	51
5.5.9 用户个人中心	52
5.6 总结	53

1 项目概述

前端 github 地址: <https://github.com/Starlight0798/Software-Engineering-frontend>

后端 github 地址: <https://github.com/Starlight0798/Software-Engineering-backend>

1.1 开发背景

中国在全球渔业养殖中具有重要地位。然而，目前我国水产养殖系统存在一系列问题。国外采用现代水质传感器技术和在线监控系统来全面监控养殖环境，并通过自动化养殖设备实现环境控制，提高效率，对环境起到积极作用。为了应对这些问题，中国在十四五发展计划中计划构建“物联网 + 海洋牧场”，通过物联网、大数据、云计算等现代信息技术，建立全面、实时、智能的养殖监控系统，精确控制养殖环境，提高效率，减少病害的发生。

本系统旨在对中国水产养殖系统存在的问题，借鉴先进技术和经验，构建“AI+ 可视化 + 物联网 + 海洋牧场”的水产养殖监控系统。

1.2 项目目标

本软件通过前后端程序的完成，最终期待实现的目标如下：

1.2.1 功能性目标

基本目标

1. 满足管理需求：满足包括养殖场主、养殖场管理员、数据分析师和普通访客在内的各类用户对于系统使用的需求。
2. 实现可视化展示模块：将处理后的数据以图形化的方式展示出来。展示包括地图、鱼类、天气等数据。此外实现了视频播放功能。
3. 实现报警与通知模块：在数据异常时及时向用户发送报警信息。
4. 实现用户信息模块：用户可以通过手机号或者账号密码注册登录该系统并查看可视化系统，且不同用户权限不同。
5. 实现数据处理与分析模块：对数据进行处理和分析，生成可视化所需的数据。支持上传数据及数据导出功能。

提高目标

1. 完成智能化模块：如鱼类图像识别，运动轨迹追踪，数量、营业额预测等功能。
2. 文件导入导出功能。

1.2.2 非功能性目标

1. 准确性：系统应确保数据的准确性，包括数据库中的数据以及系统计算和分析的结果。
2. 安全性：保护用户及系统数据的隐私。
3. 稳定性：确保系统稳定运行，用户查询及访问的过程不受干扰。

4. 易用性：确保系统具有直观友好的用户界面，操作简单易懂，用户能够轻松地完成各项操作。
5. 易维护性：系统应设计为易于维护和管理，包括清晰的代码结构、模块化设计、良好的文档说明等。

1.3 开发环境

为保证项目可以稳定运行和修改，项目组在统一的开发环境下进行代码编写。

操作系统	
操作系统	Windows
开发语言	
前端开发	HTML+CSS+JavaScript+Vue3
后端开发	Python(3.10+)+MySQL
开发工具	
代码编写	VSCODE
后端数据库	XAMPP+MySQL
数据库连接工具	Navicat
前后端接口	FastAPI
项目代码管理	Github
智能识别	YOLO v10
AI 对话	LLM API
开发框架	
整体框架	Vue3+FastAPI+MySQL

表 1: 开发环境

1.4 可行性分析

1. 经济可行性：本系统的开发和维护需要投入一定的物力成本；但能够大大节省教务人员人工统计数据所带来的开销。考虑到社会需求的支撑，项目经济可行性能得到充分保障。
2. 技术可行性：项目实现主要依赖于稳定安全的网络链接和可靠的数据库存储，对于使用人员没有任何要求。项目开发过程中所使用平台均已高度规范化、安全化，足以支撑本项目搭建完成，项目技术可行性得到充分保障。
3. 社会可行性：本项目开发初衷是为了更好地服务水产养殖管理系统，践行过程中需严格遵守相关法律法规及道德规范约束，保护用户个人隐私。
4. 操作可行性：本项目服务对象为养殖场管理人员及数据分析师等。项目本身操作页面已经进行大量简化操作能够支持对本项目不熟悉的用户轻松掌握各类需求及功能的实现。
5. 战略可行性：从项目整体角度考虑，其受众群体广、维护成本低、节省资源多。具有充分的战略可行性。
6. 计划可行性：本项目总体规模适中、实现功能在可实现范围内；由布置以来，预估完成周期为四个月，期间随着课业进度开展逐步推进项目，项目时间足够，具有计划可行性。
7. 市场可行性：水产养殖为养殖产业中的一大重要产业，管理人员需要集成式的管理系统简化管理过程。本项目即推出于此市场发展之际，拥有充分的市场可行性。
8. 风险可行性：本项目出现的主要风险来自于隐私数据的泄露可能以及错误数据的存储覆盖。因此项目充分考虑各类数据存储错误可能以防止数据错写；同时建立了风险管理机制以防止隐私数据泄露。

1.5 项目计划

1.5.1 时间节点

下面是本项目的实施计划：

任务目标	计划完成时间
后端框架搭建	5.8
后端接口初步实现	5.9
前端页面雏形搭建	5.16
前端主要页面	5.18
前端养殖管理页面	6.8
前端系统管理员	6.22
前端 AI 辅助识别	6.23
前端大数据分析	6.24
后端接口实现	6.25
前后端接口对接	6.26
系统测试	6.27
文档编写	6.28

表 2: 时间节点

1.5.2 约束条件

1. 工期约束：总项目在 6.30 之前全部完成，具体任务应当按照上一小节分期完成阶段性目标。
2. 成本约束：由于本项目为课程设计，开发人员均为学生，因此开发成本应尽量得少。
3. 成员约束：本项目为 5 人团队作业。
4. 开发约束：本项目开发人员应熟练掌握 html 和 JavaScript 语言、vue3 组件、Python 语言；对 fastAPI 框架有大致认识，对开发工具能够熟练掌握。
5. 战略可行性：从项目整体角度考虑，其受众群体广、维护成本低、节省资源多。具有充分的战略可行性。
6. 环境约束：在当前大多数主流浏览器下能正常运行本项目。
7. 性能约束：满足最大连接数，满足一定的网络带宽。

1.6 项目完成情况

作业要求	是否要求	小组完成情况
数据处理与分析模块	√	√
可视化展示模块	√	√
报警与通知模块	√	√
用户信息模块	√	√
完成软件开发文档	√	√
智能化模块		√
用户管理模块		√
个人中心模块		√

表 3: 项目要求与小组完成情况

2 需求分析与系统设计

2.1 需求分析

2.1.1 功能需求

本系统主要用户群体为：开发者，系统管理员养殖户、数据分析师以及普通用户，他们具有不同的用户权限等级。其系统功能可以分为以下几个模块：主要信息界面模块、养殖管理界面模块、智能中心界面模块、数据中心界面模块和系统管理界面模块。

1. 主要信息界面模块：

主要信息界面模块包括多个功能，用于提供系统和养殖场的全面数据展示和管理。首先，历史信息功能显示历史访问量及浏览量等数据，通过时间线展示系统信息，并提供历史养殖场数据。这部分内容帮助用户了解系统和养殖场的历史趋势和变化，从而更好地进行决策和管理。设备实时数据功能显示当前设备信息，使用户能够实时监控各设备的状态和性能，确保设备的正常运行。通过这一功能，用户可以迅速发现和处理设备问题，保障系统的稳定性和养殖场的高效运行。显示鱼群动态功能提供近期鱼群动态以及鱼群品种分布的信息。这一功能展示了鱼群的最新行为和种类分布，便于用户进行管理和决策。用户可以通过此功能了解鱼群的活动情况和变化趋势，从而优化养殖方案。地图显示功能将养殖场数据在地图上进行展示，将各地区的销量数据对应显示在地图位置上。通过这一功能，用户可以直观地查看各地区的养殖场数据，并通过地理位置进行分析和管理。

2. 养殖管理界面模块：

养殖管理界面模块包括多个功能，用于提供监控视频和鱼类数据的展示与交互。首先，视频显示功能能够显示监控视频，支持用户拖动播放。这一功能使用户可以实时监控养殖场的情况，通过拖动播放方便地查看不同时间段的视频内容，及时发现并处理异常情况。其次，鱼类数据显示功能展示当前的鱼类数据，并与后端进行对接交互。用户可以通过这一功能查看养殖场内鱼类的实时数据，如数量、种类和健康状态等。与后端的交互确保了数据的及时更新和准确性，使用户能够掌握最新的养殖情况。此外，养殖场鱼类数据显示功能根据用户选择的条件对鱼类数据进行筛选和展示，同样与后端进行对接交互。用户可以根据需要筛选特定时间段、特定种类或特定区域的鱼类数据，得到更加精确和有针对性的养殖信息。通过这一功能，用户可以更好地分析和管理养殖场的鱼类情况，提高养殖效率和效果。

3. 智能中心界面模块：

智能中心界面模块提供了多种功能，旨在通过 AI 技术提升用户体验和操作效率。首先，AI 辅助决策功能允许用户提问问题，并得到大模型的回答。这一功能可以帮助用户快速获取所需信息，辅助决策过程，提升工作效率。其次，鱼类图像识别功能支持用户上传鱼类图像，并自动识别出鱼的种类。这一功能应用了先进的图像识别技术，方便用户在实际养殖过程中快速识别鱼类种类，从而进行相应的管理和处理。此外，智能中心还提供了导入数据和导出数据的功能。导入数据功能允许用户将数据批量导入系统中，简化数据录入过程，提高数据管理效率。而导出数据功能则支持用户将已有数据导出，方便进行数据备份、共享和进一步分析。最后，视频显示功能能够显示监控视频，并支持拖动播放。这一功能使用户可以实时监控养殖场的情况，通过拖动播放方便地查看不同时间段的视频内容，及时发现并处理异常情况。智能中心界面模块通过这些功能，为用户提供了全面、高效的智能管理工具。

4. 数据中心界面模块：

数据中心界面提供了全面的功能展示和分析工具，用于管理和分析海洋牧场的各类数据。首先，鱼群数量统计功能包括总鱼群数量、今日新增和今日死亡，帮助用户实时了解鱼群的动态变化。环境状况统计

功能则涵盖温度、水质、空气质量和气压等多种环境指标，确保用户能够全面监控养殖环境的健康状况。设备状况统计功能包括设备电量、服务器负载和水位信息，确保设备的正常运行和及时维护。鱼群数量分析及种类分布功能展示了从 1900 年至 2015 年鱼群数量及种类分布的变化趋势，帮助用户了解长期趋势和变化规律。数据交汇处理功能展示了海洋牧场数据处理前后的对比，通过柱状图、折线图和数据视图等多种形式进行数据展示，同时支持数据导出。数据类型占比功能汇总了鱼类数据、鱼类种类、设备数据和气象数据等各类数据的占比情况，提供了全局视图。传感器信息功能展示了不同传感器的信息，包括传感器编号、类型和大小，涵盖水底摄像头、水面摄像头、云台和声呐等多种传感器。结构化数据功能将海洋牧场各个种类的数据分类以结构化的形式绘制出来，帮助用户进行有序管理和分析。最后，设备状态功能展示了各个设备的状态，包括故障、离线或正常，确保用户能够及时了解设备的运行状况，进行必要的维护和管理。通过这些功能，数据中心界面实现了对海洋牧场数据的全面监控和智能分析。

5. 系统管理界面模块：

首先，用户管理模块主要用于管理用户信息，提供用户的增删改查功能。通过这个模块，管理员可以轻松地添加新用户、删除不需要的用户、修改用户信息以及查询用户的详细信息，从而实现对用户的全面管理。其次，角色管理模块主要用于管理用户的角色和权限。管理员可以通过这个模块查看和编辑用户的权限，确保每个用户只能访问和操作其权限范围内的功能。这有助于提高系统的安全性和管理的效率。菜单管理模块主要用于管理系统界面的菜单配置。通过这个模块，管理员可以对界面的 ID 和主题等进行管理，从而实现对系统界面布局和风格的个性化定制。这使得系统界面更加符合用户的需求和审美。最后，报警监管模块用于管理系统的报警信息。管理员可以通过这个模块设置报警阈值，对相关数据进行预警。当系统数据超过设定的阈值时，系统会自动生成报警信息，提醒管理员及时处理。这有助于提高系统的安全性和稳定性，防止潜在的问题和风险。

6. 系统管理界面模块：

在海洋牧场管理系统的报警系统页面中，系统管理人员和开发者可以通过预警监管页面对不同的环境和设备指标进行管理和监控。该页面提供了设置报警阈值的功能，使用户能够根据实际需求调整各项指标的报警条件。用户可以在页面上设置各项指标的报警阈值。指标包括但不限于水温、溶解氧、电导率等关键环境参数。通过调整阈值，用户可以精确控制何时触发报警，以便及时采取措施应对可能出现的问题。页面下方显示了实时更新的各项指标数值，方便用户随时查看当前环境状况。这些数据每 30 秒刷新一次，确保用户可以获取最新的环境信息。实时数据的展示使得阈值的设置和调整更加有据可依。当系统检测到某一指标超出设定阈值时，会自动向所有当前已注册的用户发送报警消息。报警消息包括具体的报警内容和时间，确保所有相关人员都能及时了解并采取必要措施。页面上还显示了报警结果、报警次数以及最新的报警时间。这样，系统管理人员和开发者可以方便地查看历史报警记录和频次，了解系统运行状况和潜在问题。报警系统页面的布局包括阈值设置区域、实时数据区域和报警信息区域。阈值设置区域显示各个指标的名称及当前设置的阈值，并提供输入框和滑动条用于调整阈值，以及保存和重置按钮，确保用户能够保存新的设置或恢复默认值。实时数据区域展示各项指标的实时数值，并每 30 秒更新一次数据。报警信息区域则包含最新的报警消息列表、报警次数统计以及最近一次报警的具体时间，帮助用户了解最新的系统状态。

2.1.2 数据需求

用户数据

1. 用户名：用户设置的用于登录使用的 ID。
2. 密码：用户设置的用于登录使用的密码。

3. 电话号码：用于避免重复注册等。
4. 职位：用于区分用户账号权限。
5. 用户操作记录：记录用户对系统的操作行为，如登录、查看数据、修改设置等。

环境数据

1. 水质参数：如 pH 值、溶解氧、氨氮、亚硝酸盐、硝酸盐等。
2. 水温：养殖水体的温度。
3. 水位：养殖池或水体的水位高度。
4. 水流速度：水体的流动速度。
5. 光照强度：光照对养殖生物的影响。
6. 空气温湿度：养殖场周围环境的温度和湿度。

鱼群数据

1. 鱼群数量：养殖池或水体中鱼类的数量。
2. 鱼体重：鱼类的平均体重。
3. 鱼类健康状况：鱼类的健康状况指标，如生长情况、死亡率等。

2.1.3 运行系统需求

1. 硬件配置要求：建议使用高性能服务器，具备多核处理器和足够的内存，以应对高并发访问和大数据处理需求。确保网络环境的稳定性，建议配置高性能的交换机和路由器，支持负载均衡和高可用性。
2. 软件配置要求：推荐使用稳定且常见的操作系统 Windows。安装和配置适合的数据库管理系统（DBMS），如 MySQL、PostgreSQL、Oracle 等。使用 Vue.js、FastAPI 框架进行系统维护。

2.2 系统设计

总业务说明 总业务说明图如下所示。

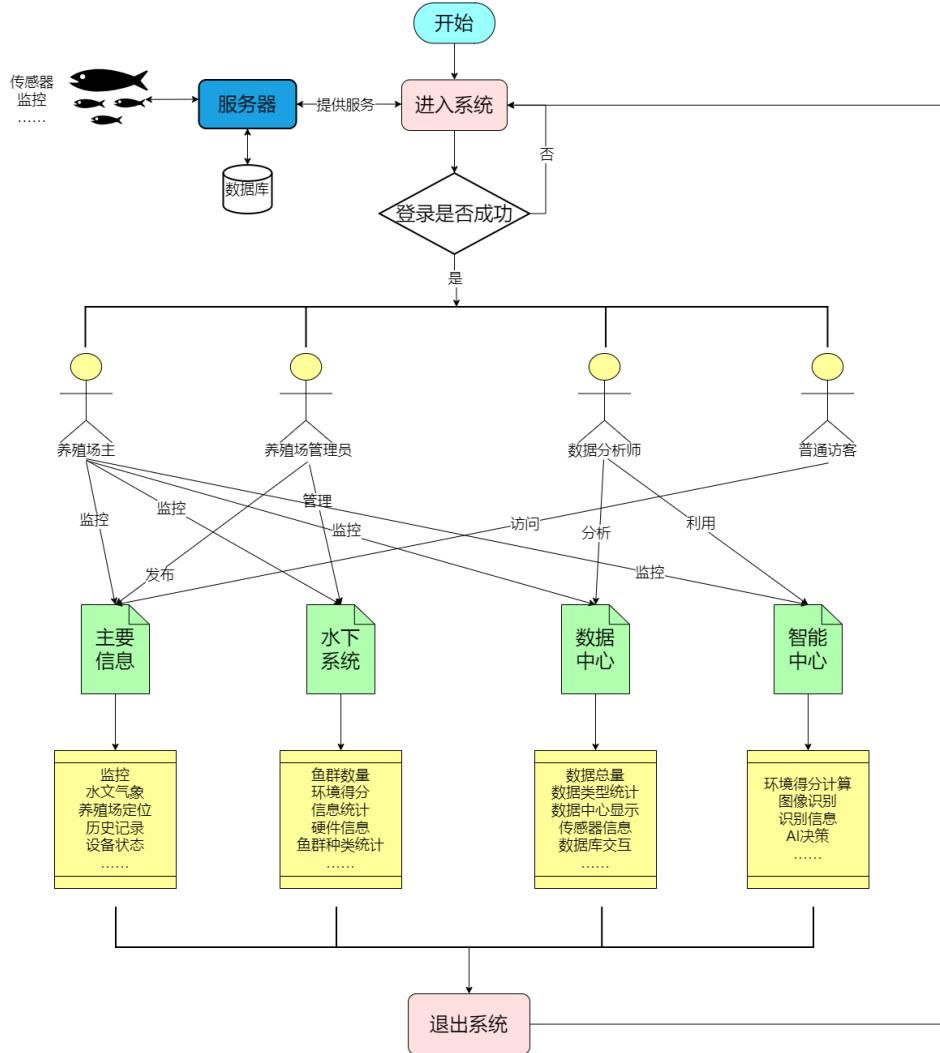


图 2.1: 总业务说明

总业务描述如下：

当水产养殖管理系统启动后，用户需要进行登录操作，如果是新用户需要进行注册。不同的用户账号具有不同的权限，之后系统会根据用户账号的权限为用户展示不同的子系统。

不同用户账号的权限如下：

- 养殖场主：** 养殖场主具有所有模块的监控及管理权限，可以监控整个系统的信息及其他用户的操作，考虑到养殖场主可能不需要过于专业的操作，因此将专业的处理功能分配给其他用户。
- 养殖场管理员：** 养殖场管理员具有管理水下系统以及发布主要信息的权限。养殖场管理员需要对养殖场的监控等设备的开关进行管理，且具有发布养殖场主要信息的责任。
- 数据分析师：** 数据分析师具有使用和管理数据中心以及利用智能中心的权限。考虑到对养殖环境和鱼群的数据的分析可能需要专业人员来负责，数据分析师可以利用掌握的数学分析的知识及 AI 大模型等先进科技对原始数据进行分析和处理，得到管理人员需要的数据。
- 普通访客：** 普通访客具有访问主要信息的权限。普通访客可以通过访问主要信息界面，获取自己需要的相关数据。

子业务设计 子业务围绕不同的用户权限展开，不同权限用户的业务如下图所示。

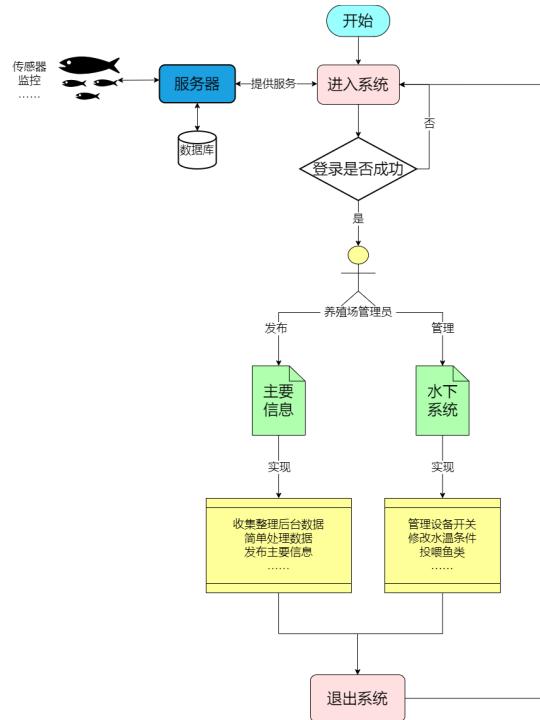


图 2.2: 养殖场管理员业务

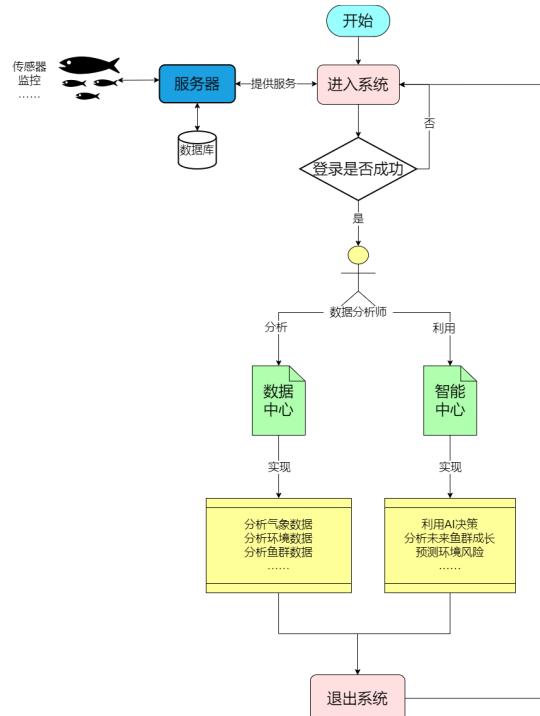


图 2.3: 数据分析师业务

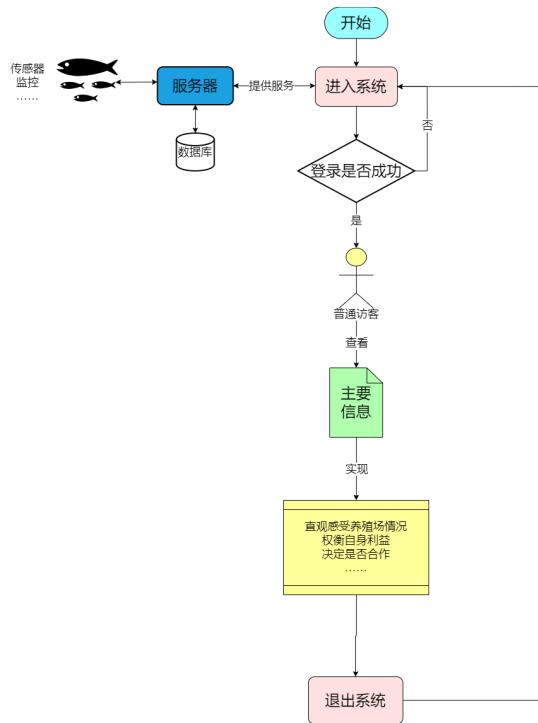


图 2.4: 普通访客业务

用户可访问界面按照上述主要业务流程统一设计，子业务只需对不同权限进行区分即可。

2.3 详细设计

2.3.1 界面用例设计

主要信息界面 主要信息界面包括的用例有：

1. 水文数据展示用例：该用例通过水文数据向用户展示了养殖场的环境特征，包含了具体各个环境或天气数据的展示用例。
2. 海洋牧场位置展示用例：该用例通过地图向用户展示了各地养殖场的情况，拓展了选择特定地点养殖场的用例。
3. 历史水文数据展示用例：该用例向用户展示了数据库中保存的历史水文数据情况，拓展了选择特定日期的水文数据的用例和与当日水文数据比较的用例。
4. 设备状态用例：该用例向用户展示了养殖场设备的情况，包含了设备报警的用例。
5. 鱼群状态用例：该用例向用户展示了养殖场鱼群的情况。

养殖管理界面 养殖管理界面包括的用例有：

1. 监控视频展示用例：该用例通过监视视频向用户展示了养殖场的现状。
2. 信息统计展示用例：该用例向用户展示了通过环境数据及鱼群数据进行的信息统计。
3. 设备管理用例：该用例允许管理员对传感器、摄像头等设备进行开关管理，包含了展示各设备状态的用例。

数据中心界面 数据中心界面包含的用例有：

1. 鱼群数量统计用例：该用例通过统计鱼群数量向用户展示了养殖场鱼群的情况，包括总鱼群数量、今日新增和今日死亡。
2. 环境状况统计用例：该用例通过统计环境状况向用户展示了养殖场的环境特征，包括温度、水质、空气质量等数据的展示。
3. 设备状况统计用例：该用例通过统计设备状况向用户展示了养殖场设备的情况，包括设备电量、服务器负载和水位等数据的展示。
4. 鱼群数量分析及种类分布用例：该用例通过分析鱼群数量及种类分布的变化趋势向用户展示了从 1900 年至 2015 年的数据情况。
5. 数据交汇处理用例：该用例展示了数据交汇处理的情况，具体内容待补充。
6. 数据类型占比用例：该用例通过汇总各类数据类型的占比情况向用户展示了鱼类数据、鱼类种类、设备数据、气象数据等的比例分布情况。
7. 传感器信息用例：该用例向用户展示了不同传感器的信息，包括传感器编号、类型、大小，涵盖水底摄像头、水面摄像头、云台、声呐等不同传感器类型。
8. 结构化数据用例：该用例展示了养殖场的结构化数据情况，具体内容待补充。
9. 设备状态用例：该用例向用户展示了养殖场各设备的运行状态，包括正常、故障或离线等情况的展示。

报警监管页面 报警监管界面包含的用例有：

1. 报警指标用例：该用例展示了不同报警指标的情况，包括盐度（%），溶解氧（mg/L），pH 与水温（°C）的报警情况，可以手动进行选择。
2. 报警阈值用例：该用例展示了各个报警指标的阈值设置情况，用户可以查看和调整不同指标的报警阈值以满足实际需要。
3. 报警方式用例：该用例展示了不同报警方式的设置情况，包括短信、邮件、应用内通知三种不同报警方式的配置和使用情况。
4. 当前报警数量用例：该用例实时展示了当前系统中各类报警的数量，会根据最新报警情况实时更新详细统计。
5. 最新报警时间用例：该用例展示了最近一次报警的时间，帮助用户了解报警的最新动态和处理时效。
6. 当前报警监管时间间隔用例：该用例展示了系统当前的报警监管时间间隔设置，用户可以查看报警监控的频率。
7. 指标报警图表用例：该用例通过图表形式展示了各个指标的数值情况，用户可以直观地查看不同指标的变化趋势，由此来动态地调整报警阈值。
8. 一年内通知方式数量变化趋势用例：该用例通过分析一年内不同通知方式（短信、邮件、应用内通知等）的使用数量变化趋势，帮助用户了解报警通知方式的使用情况和趋势变化。

智能中心界面 智能中心界面包含的用例有：

1. AI 对话用例：该用例允许用户利用对话框与 AI 进行对话，从而得到相关建议。
2. AI 识别图像用例：该用例允许用户上传图像，AI 模型将对该图像进行识别分析并给出结论。
3. AI 识别视频用例：该用例中 AI 将对视频中的鱼类进行识别并给出结论。
4. 上传数据用例：该用例允许用户上传数据至后端数据库。

2.3.2 用户流程设计

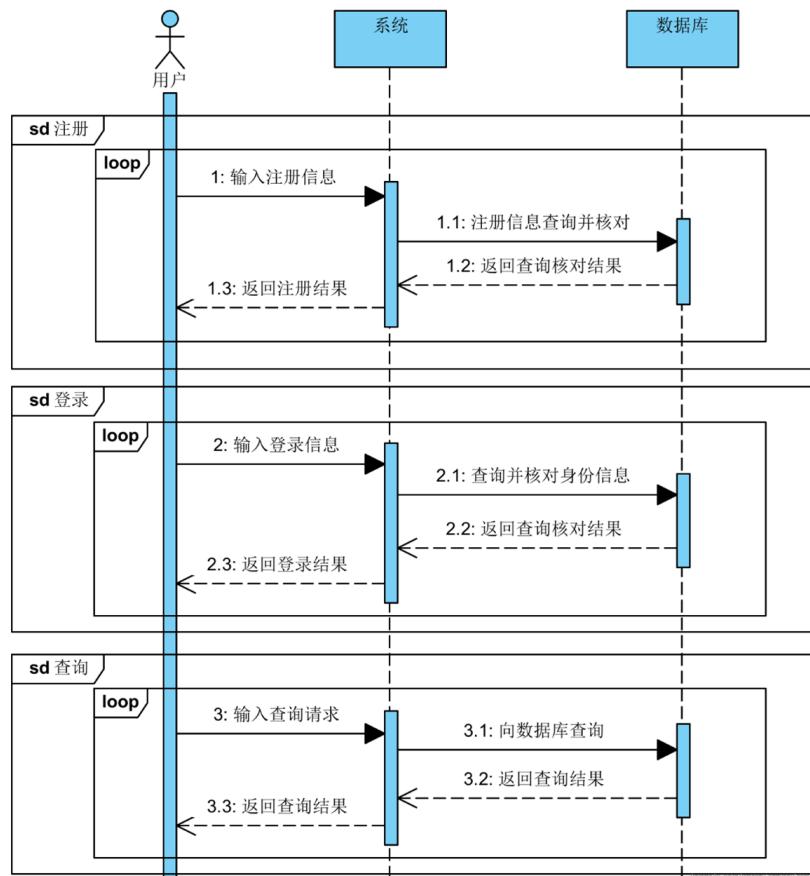


图 2.5: 用户注册使用流程顺序设计

用户注册使用流程顺序设计 用户注册使用流程顺序设计如上图所示。

1. 用户输入注册身份信息，系统向数据库查询消息并进行注册消息核对，向用户返回反馈信息。
2. 用户向系统输入登录账号信息，系统向数据库查询账户消息，并向用户返回反馈信息。
3. 用户向系统发起查询请求，系统查询数据库，并向用户返回查询信息。

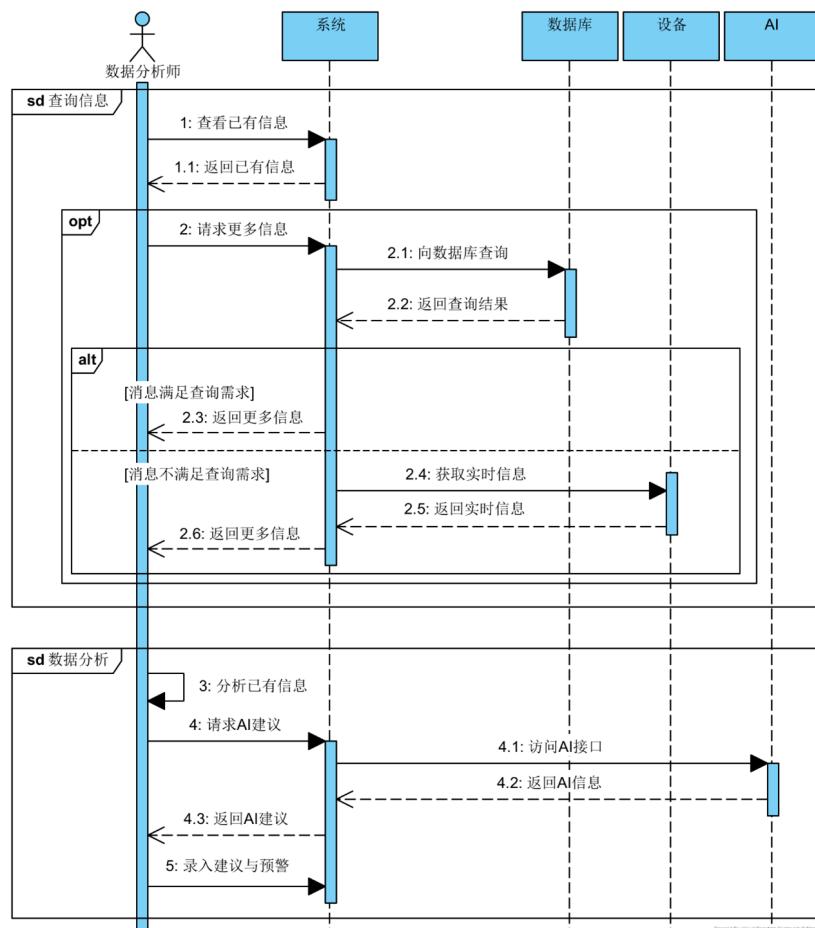


图 2.6: 数据分析师数据分析顺序设计

数据分析师数据分析顺序设计 数据分析师数据分析顺序设计如上图所示。

1. 数据分析师向系统查询查看已有信息，系统返回已有信息。
2. 数据分析师可能会向系统查询更多信息，系统向数据库查询更多信息后，判断信息是否符合数据分析师的查询要求；如果不符合要求，系统会进一步向设备获取当前的实时数据，然后返回给数据分析师。
3. 数据分析师对现有的信息进行分析。
4. 数据分析师向系统请求 AI 建议，系统访问 AI 接口后返回 AI 建议。
5. 数据分析师结合 AI 的建议向系统录入建议和预警。

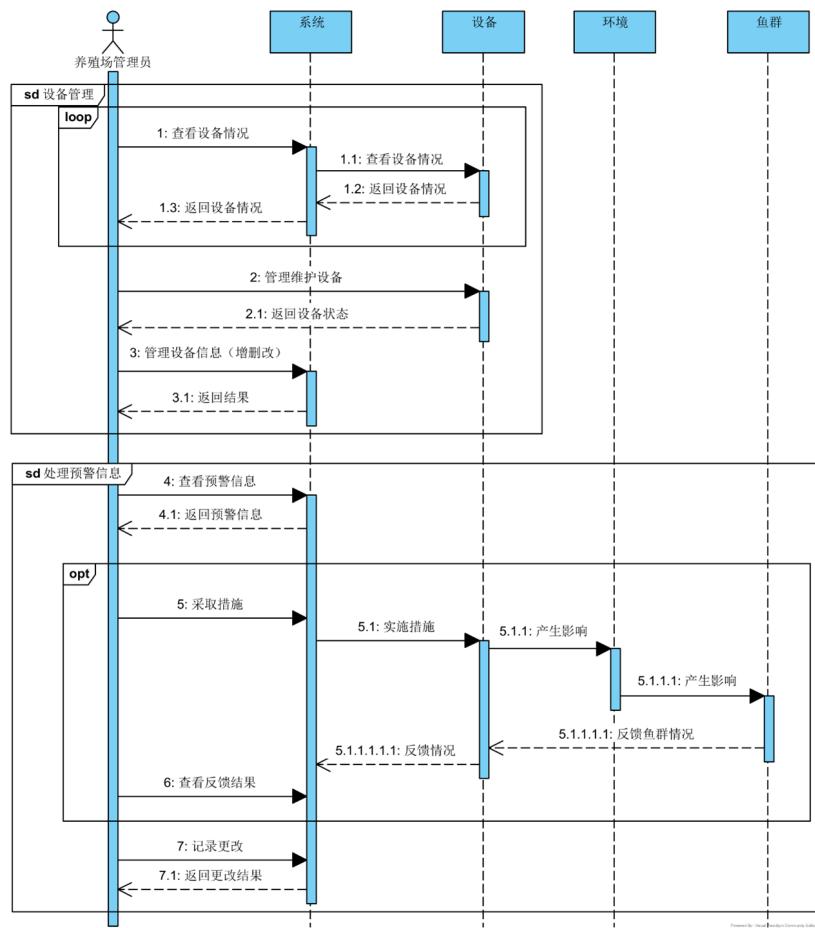


图 2.7: 养殖场管理员管理维护顺序设计

养殖场管理员管理维护顺序设计

养殖场管理员管理维护顺序设计如上图所示。

1. 养殖场管理员循环查看所有设备的情况，系统返回设备情况。
2. 养殖场管理员管理维护设备，返回设备情况。
3. 养殖场管理员在系统中对设备有的更改进行记录，系统返回更改结果。
4. 养殖场管理员在系统中查看预警信息，系统返回预警信息。
5. 养殖场管理员根据预警在系统中采取措施，系统根据采取措施对设备的状况进行更改，设备对环境产生影响，进而对鱼群产生影响；设备反馈鱼群的情况给系统。
6. 管理员在系统中查看反馈的情况，判断是否需要进一步采取措施。
7. 管理员在系统中记录更改，系统反馈更改结果。

2.3.3 系统架构

- 前端界面：使用 HTML、CSS、JavaScript 实现用户界面，通过 vue3 框架提供动态交互和数据展示。
- 后端服务：使用 Python 语言开发，结合 FastAPI 框架实现系统的业务逻辑和数据处理。
- 数据库：使用关系型数据库 MySQL 存储数据。

- API 接口：提供前后端之间的数据交互和功能调用，使用 FastAPI 方式进行通信。

系统总体架构如下图所示。

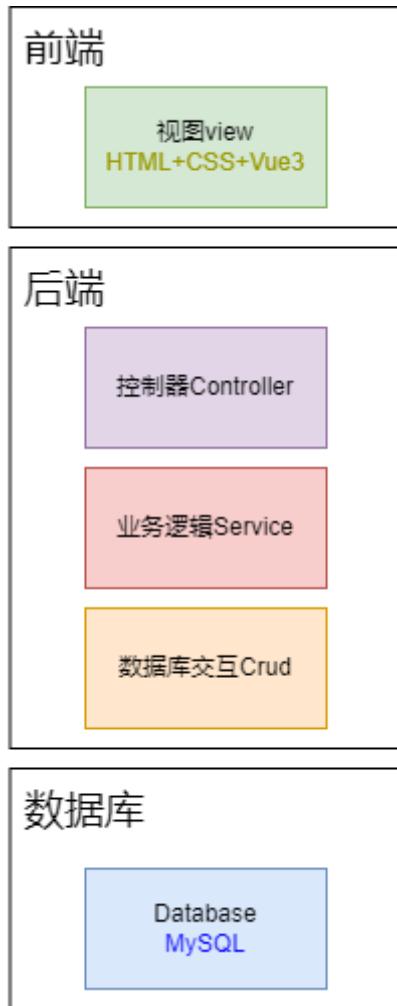


图 2.8: 系统架构图

前端使用 HTML+CSS+Vue3+JavaScript 实现 view 层，后端使用 Python+FastAPI+MySQL 技术栈。

1. View 层：视图层，根据连接到的数据展示页面给用户，同时根据用户权限判断显示内容。
2. Controller 层：响应用户需求，决定用什么页面，需要准备什么数据来显示。Controller 层负责前后端交互，接收前端请求，调用 Service 层，接收 Service 层返回的数据，最后返回具体的数据和页面到客户端。
3. Service 层：Service 层根据前端调用请求，进行数据的计算并响应。如果需要从数据库调取数据，则进一步调用 Crud 层与数据库交互。
4. Crud 层：后端与数据库交互接口层。后端在 Crud 层与数据库进行交互，通过数据库接口传递命令。
5. Database 层：数据库层，执行后端接口传递的命令，在数据库中进行查询操作，并返回数据至后端。

2.3.4 安全性设计

- 用户身份验证：使用用户名和密码进行用户登录认证，并确保密码安全性（如加密存储，密码策略等）。

- 访问控制：根据用户角色和权限，限制用户对敏感数据和功能的访问。

使用 token 机制实现上述特性：将所有传输到服务端的请求拦截下来，判断是否是登录或注册请求，如果是则直接放行，否则解析请求头中携带的 token，解析失败则返回登录失败的响应，解析成功后根据 token 中携带的角色 id 判断其是否有相应权限访问改请求。

token 机制具体流程如下图所示。

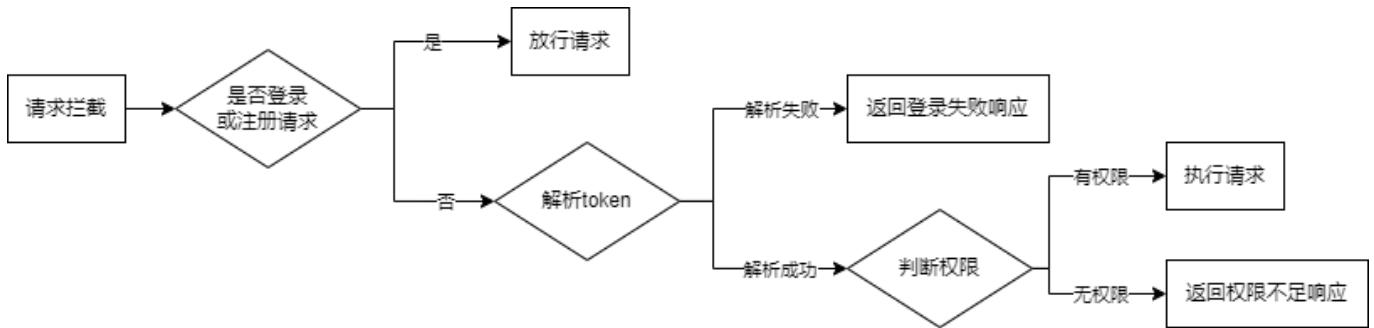


图 2.9: token 机制流程

2.4 数据库设计

养殖场管理系统需要采集的数据种类很多，涵盖了养殖过程中的各个环节和方面。系统所有的数据存在一个数据库中，数据库中各个表的数据字典如下：

用户数据 所有用户数据存储在一张名为 user 的表中。该表结构如下：

数据元素	数据类型	描述	主键
user_id	整数	用户编号	是
username	字符串	用户名	否
password	字符串	密码	否
name	字符串	姓名	否
gender	字符串	性别	否
age	整数	年龄	否
phone_number	字符串	电话号码	否
email	字符串	邮箱地址	否
role	字符串	职位	否

表 4: 用户数据

环境物理数据

所有环境物理数据存储在一张名为 water_physic 的表中。该表结构如下：

数据元素	数据类型	描述	主键
station_id	字符串	站点 id	是
time	字符串	数据记录时间	是

depth	浮点数	水体深度	否
temperature	浮点数	水温	否
transparency	浮点数	透明度	否
solids	浮点数	悬浮质	否
electrical_conductivity	整数	电导率	否

表 5: 环境物理数据

环境化学数据

所有环境化学数据存储在一张名为 `water_chemistry` 的表中。该表结构如下：

数据元素	数据类型	描述	主键
station_id	字符串	站点 id	是
time	字符串	数据记录时间	是
pH	浮点数	pH 值	否
total_N	浮点数	总氮	否
total_P	浮点数	总磷	否
chlorophyll	浮点数	叶绿素	否
chl_without_Mg	浮点数	去除镁的叶绿素	否
kmno4	浮点数	高锰酸盐指数	否
dissolved_o	浮点数	溶解氧	否
BOD5	浮点数	五日生化需氧量	否
NH4N	浮点数	氨氮	否
HNO2	浮点数	亚硝酸盐	否
NO3	浮点数	硝酸盐	否
dissolved_N	浮点数	溶解氮	否
phosphate	浮点数	磷酸盐	否
dissolved_P	浮点数	溶解磷	否
alkalinity	浮点数	碱度	否
potassium_lion	浮点数	钾离子	否
sodium_lion	浮点数	钠离子	否
calcium_lion	浮点数	钙离子	否
magnesium_lion	浮点数	镁离子	否
fluoride_lion	浮点数	氟离子	否
chloride	浮点数	氯离子	否
sulfate	浮点数	硫酸盐	否
silicate	浮点数	硅酸盐	否
alkalinity_as_caco3	浮点数	碳酸盐	否
silicate_as_si	浮点数	硅酸盐	否

表 6: 环境化学数据

鱼类数据

每个地点的鱼类数据各自存储在对应表中，每个地点的表结构如下：

数据元素	数据类型	描述	主键
date	日期	数据记录时间	是
fishname	字符串	鱼类名称	是
lengthrange	字符串	鱼类长度范围	是
weightrange	字符串	鱼类体重范围	是
num	整数	鱼类数量	否

表 7: 鱼类数据

消息数据

在用户中心，为了能够更好地实现对不同类型的消息的获取和查看，这里将具体消息类型也存储在后端，表结构如下：

数据元素	数据类型	描述	主键	外键
msg_id	整数	消息的序号，自增的唯一标志	是	否
user_id	字符串	消息所属用户的序号，外键对应于表 user	否	是
message	字符串	消息具体内容	否	否
status	字符串	消息所处的状态	否	否
date	字符串	消息发送的日期	否	否

表 8: 消息数据

阈值数据

在报警管理模块，为了能够实现对不同指定指标阈值的修改和读取，这里将具体阈值和对应的指标也存储在后端，表结构如下：

数据元素	数据类型	描述	主键
metric	字符串	阈值所属的指标	是
threshold	整数	指标对应的阈值，根据各自的单位不同	否

表 9: 阈值数据

设备数据 所有设备数据存在一张 equipment 表中，表结构如下。

数据元素	数据类型	描述	主键
Equipment Type	字符串	设备类型	否
Equipment ID	字符串	设备编号	是

Equipment Status	字符串	设备状态	否
Operating Time	字符串	设备运行时间	否
Length	浮点数	设备长度	否
Width	浮点数	设备宽度	否
Longitude	浮点数	经度	否
Latitude	浮点数	纬度	否
Depth	浮点数	深度	否
Last Maintenance	日期	上次维护时间	否
Next Maintenance	日期	下次维护时间	否
Warranty Expiry	日期	保修到期时间	否

表 10: 设备数据表

2.5 UI 设计

水产养殖管理系统的 UI 设计，考虑了可视化、可用性、直接映射和有效反馈等原则。

1. 登录界面：提供用户名和密码输入框以及登录按钮。使用清晰的标识和排版，使用户能够快速找到登录入口。
2. 注册界面：提供基本信息输入框，包括姓名、联系方式等，并且指定注册用户的权限，附带验证和提示信息。使用直观的表单布局和必填字段的标识，以及验证和提示信息，确保用户正确填写信息。
3. 主要信息界面：使用直观的数字形式展示访问量等数据；使用波形图及饼图展示最近水文数据及鱼类数据；展示数据时根据数据量调整条状长度，更加直观；并且使用地图展示渔场分布。
4. 养殖管理界面：使用视频播放展示渔场现状；使用波形图及饼状图展示鱼类及水文数据统计信息。
5. 系统管理界面：设计了用户管理、角色管理、菜单管理、预警界面。前三个界面设计了对相关信息进行管理的界面，预警界面可设置阈值对相关信息进行预警。
6. 智能中心界面：设计了 AI 接口，可与 AI 进行对话，并对视频和图像进行识别。可对对话内容进行下载，并且可上传数据至后台数据库。
7. 数据中心界面：设计了动态展示数据的界面，用户可以更加直观地感受养殖场的历史数据和当前数据。

3 系统测试

3.1 测试环境

测试环境与系统开发环境大致相同，具体条件如下所示。

运行操作系统	Windows
环境配置需要	Vue3+Python(3.10+ 及所需库)+MySQL

表 11: 测试环境

3.2 功能测试

3.2.1 注册功能

我们对注册功能进行了详细的测试：

- 输入有效的用户名、密码、权限和其他必填信息。
- 确认密码与密码一致。
- 验证是否成功注册并能够登录系统

测试注册时的边界条件 首先我们对注册的边界条件进行检查：

The figure consists of three side-by-side screenshots of a web application's registration form. All three screenshots show a background image of a sunset over the ocean.

- 图 3.10: 密码不合格** (Screenshot 1): The password field contains two lowercase letters and one digit, failing the requirement for at least one uppercase letter. A red border highlights the field, and a red error message at the top says "注册失败: [object Object]".
- 图 3.11: 邮箱不合格** (Screenshot 2): The email field contains an invalid email address (1960437320@qq.com). A red border highlights the field, and a red error message at the top says "注册失败: Username has already existed".
- 图 3.12: 用户名重复** (Screenshot 3): The username field contains a previously registered username ('analyst123'). A red border highlights the field, and a red error message at the top says "注册失败: Username has already existed".

成功注册 正确输入信息后，注册信息会存储到数据库：

user_id	username	password	name	gender	age	phone_number	email	role
1	erwinzhou123	\$2b\$12\$qhBV(Null)	(Null)	(Null)	(Null)	(Null)	1960437321admin	
2	admin123	\$2b\$12\$jzsCn(Null)	(Null)	(Null)	(Null)	(Null)	1960437321user	
3	analyst123	\$2b\$12\$iXdV4(Null)	(Null)	(Null)	(Null)	(Null)	1960437321analyst	
4	farmer123	\$2b\$12\$2Ifw6(Null)	(Null)	(Null)	(Null)	(Null)	1960437321farmer	

图 3.13: 注册成功

3.2.2 登录功能

测试正常登录

- 输入正常的用户名和密码，验证是否能够成功登录系统。



图 3.14: 登陆界面



图 3.15: 登陆成功

测试登录时的边界条件

- 选择错误的权限进行登录，验证是否无法登录。

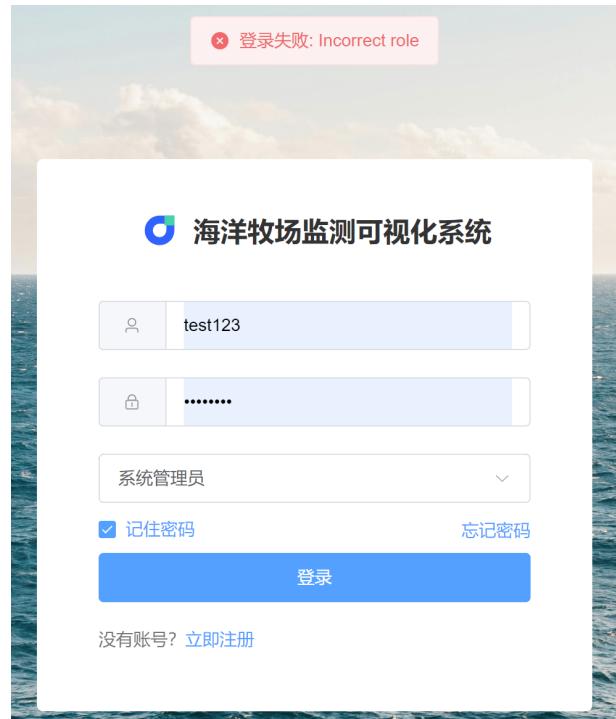


图 3.16: 错误权限登录失败

- 使用空的用户名和密码，或非法输入，验证是否无法登录。

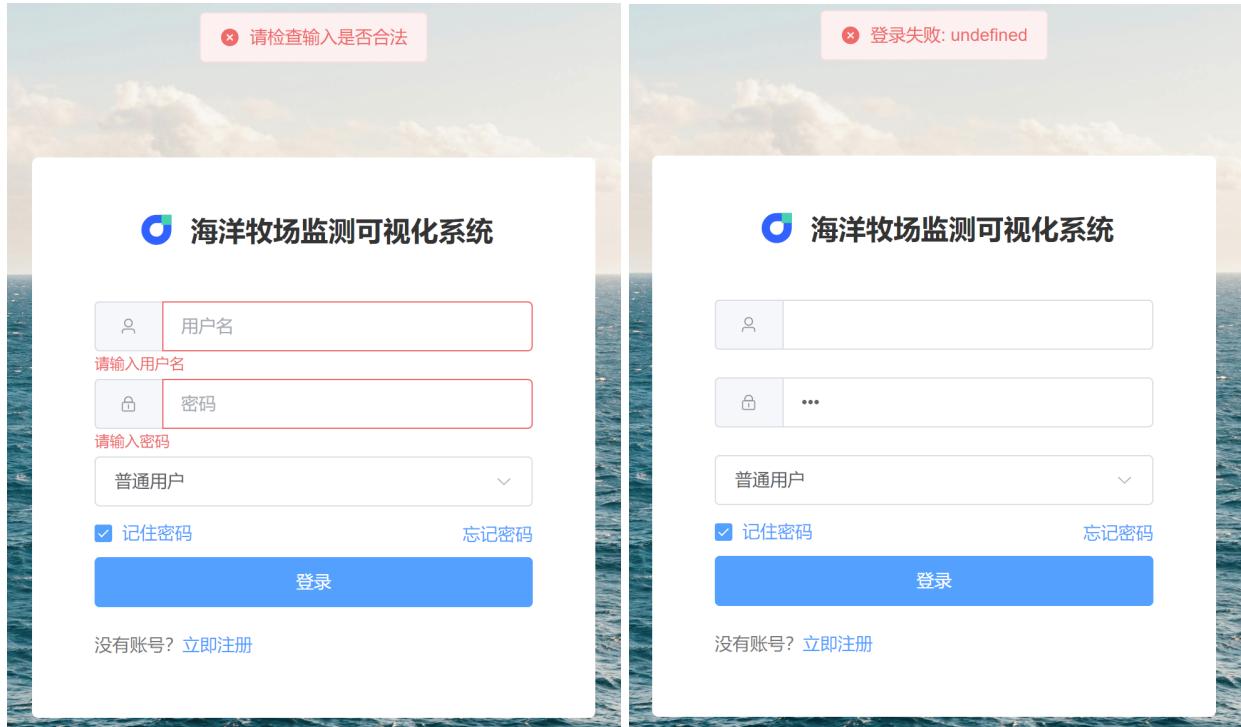


图 3.17: 空用户名-登录失败

图 3.18: 非法输入-登录失败

测试错误的用户名和密码

- 使用错误的用户名和密码，验证是否登录失败。



图 3.19: 错误的用户名和密码-登录失败

3.2.3 数据分析与可视化模块

在大数据分析的过程中，数据可视化是一个至关重要的模块。我们通过可视化模块将复杂的海洋牧场信息转化为直观的图表和图形，使用户能够快速理解数据背后的意义和趋势。在本项目中，数据可视化模块采用了多种工具和方法，为用户提供全面的数据展示和分析。在本小节我们将详细介绍对每个数据的展示方式。

环境数据显示 为了直观展现环境数据，我们利用传感器收集了水文和气象数据进行分析与展示。

首先我们利用柱状图展示对于海洋生物养殖极其重要的环境指标以辅助养殖户做出合理的抉择：

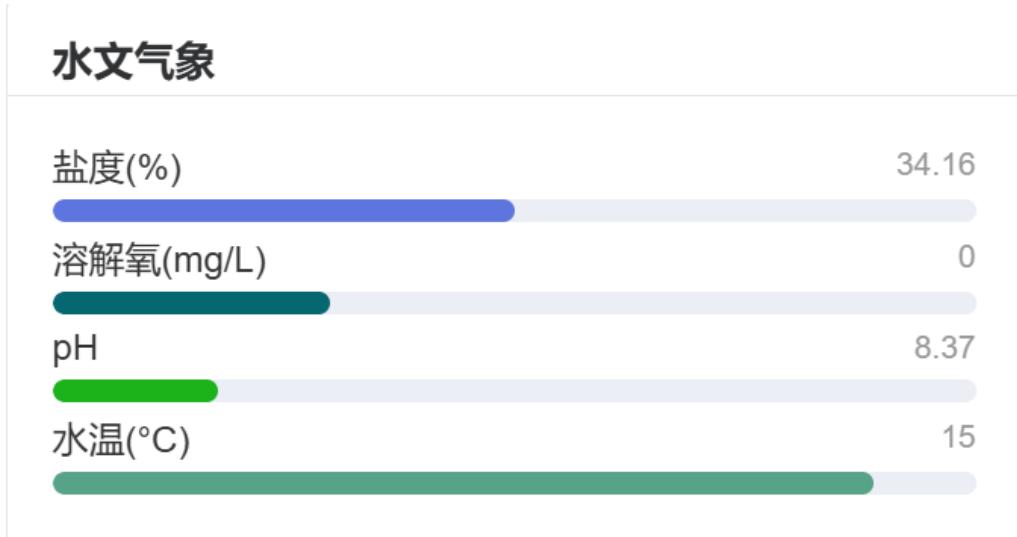


图 3.20: 环境指标

对于非专业人士，我们给出了较为常见的环境指标，可以让游客对海洋牧场的气候有清晰的了解：

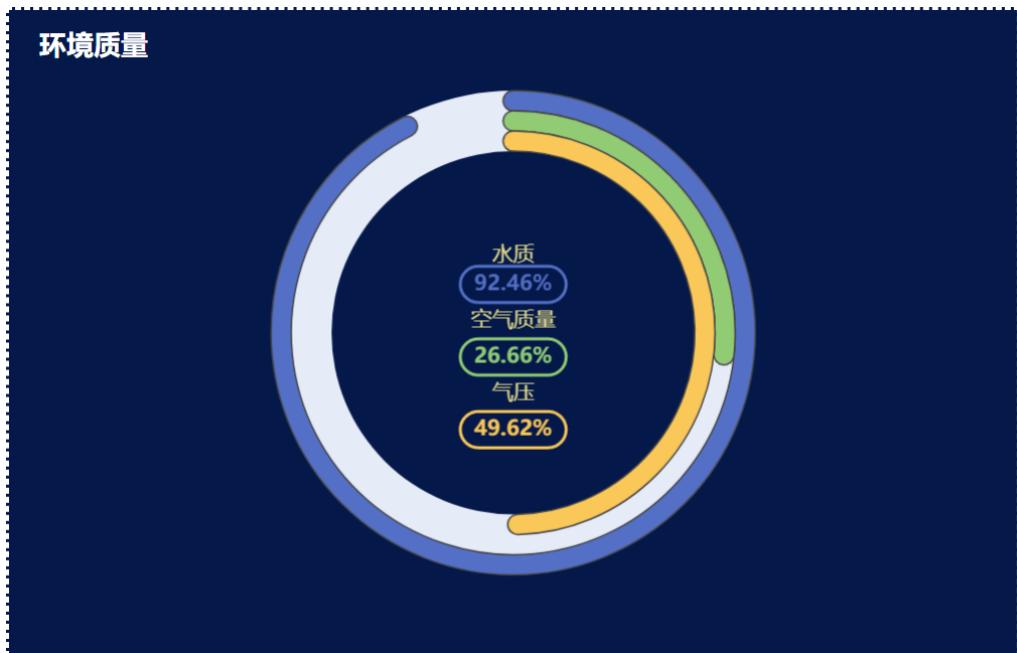


图 3.21: 环境指标 2

除此之外，我们也对近期的气候进行了监控和统计：



图 3.22: 海洋牧场气候

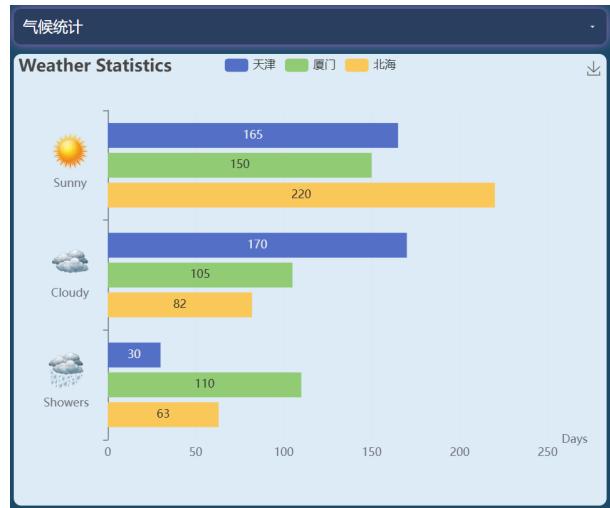


图 3.23: 海洋牧场天气统计

鱼类数据显示 在本项目中，我们通过数据可视化模块对海洋牧场的鱼类数据进行了全面展示。鱼类数据展示模块主要包含以下几个方面：

- 鱼类种类数量、分布展示
- 鱼类数量随时间的变化趋势
- 未来鱼类数量预测

首先我们通过动态柱状图以及动态扇形图展示不同种类的鱼在近年的数量以及占比变化：

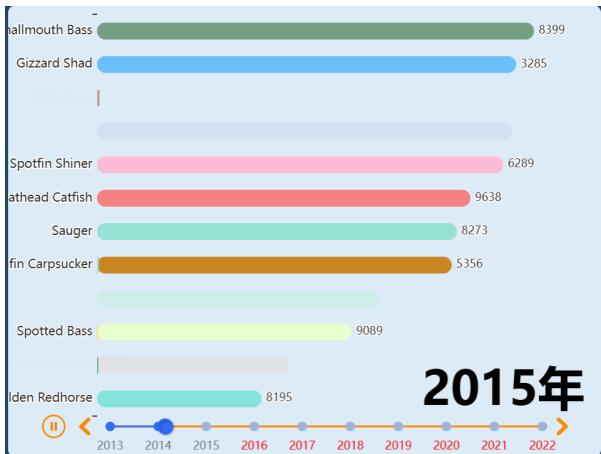


图 3.24: 鱼类数量变化

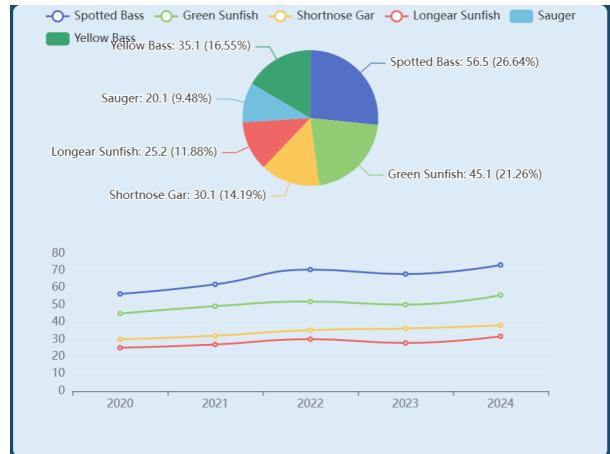


图 3.25: 鱼类占比变化

除此之外，我们对每种鱼都进行了详细的统计分析：

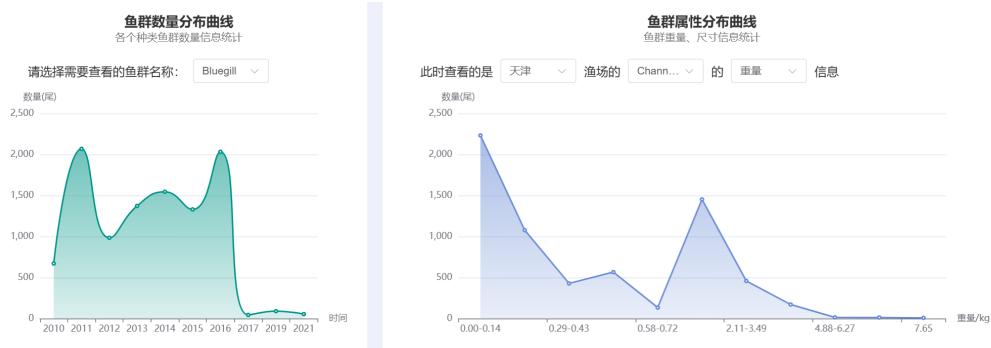


图 3.26: 不同鱼类展示

为了指导管理员和养殖户对海洋牧场的管理，我们通过 LSTM 模型对鱼群数量以及营业额进行了预测，让用户可以了解到海洋牧场的未来近期反正趋势。

未来预测

预测海洋牧场未来十天的营业额与物种数量

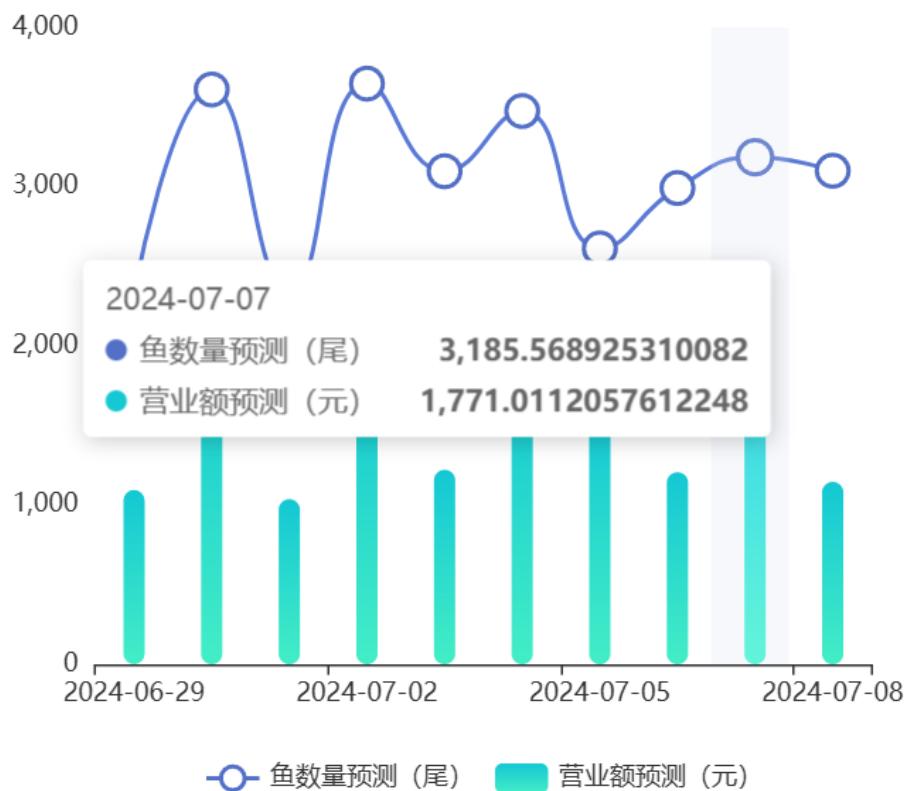


图 3.27: 预测结果展示

系统数据展示 海洋牧场管理系统中，我们也对系统数据进行展示，主要涉及到设备、系统数据、访客数据等。通过下面的图，我们为访客呈现了该管理系统的数据分布以及主要数据类型：



图 3.28: 数据占比

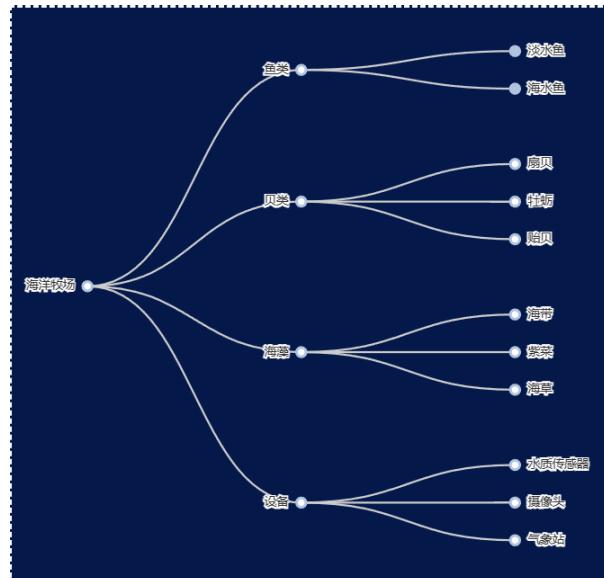


图 3.29: 主要数据

我们也通过表格和动态图展现了设备情况：

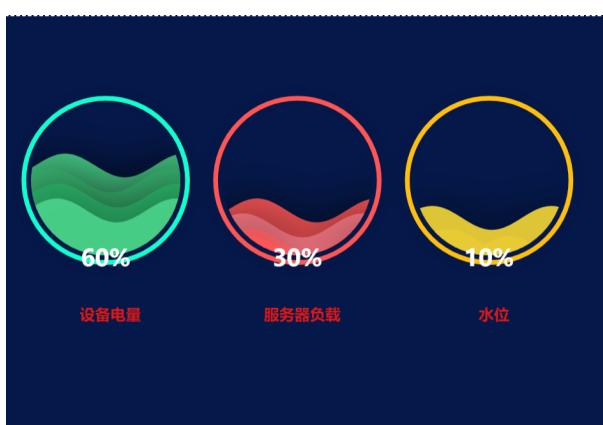


图 3.30: 系统运行状态

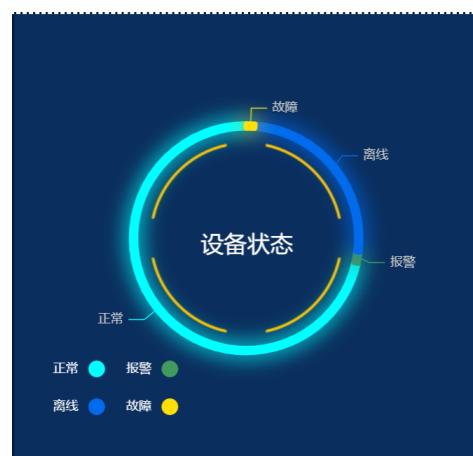


图 3.31: 设备运行状态

我们通过日志记录了海洋牧场的动态，并用组合图展示了访客数量：



图 3.32: 牧场动态

时间线

最新的牧场动态

- 数量波动 20只xwf出生啦! 2024-06-29 07:23
- 养殖活动 捕捞了20只Bluegill 2024-06-29 07:23
- 养殖活动 投放了10条Channel Catfish 2024-06-29 07:23
- 数量波动 20只xwf出生啦! 2024-06-29 07:23
- 环境动态 天黑了 2024-06-29 07:22

图 3.33: 访客数量

地图显示 为了增加图标多样性我们使用地图来展示海洋牧场的一个月内订单销售地：

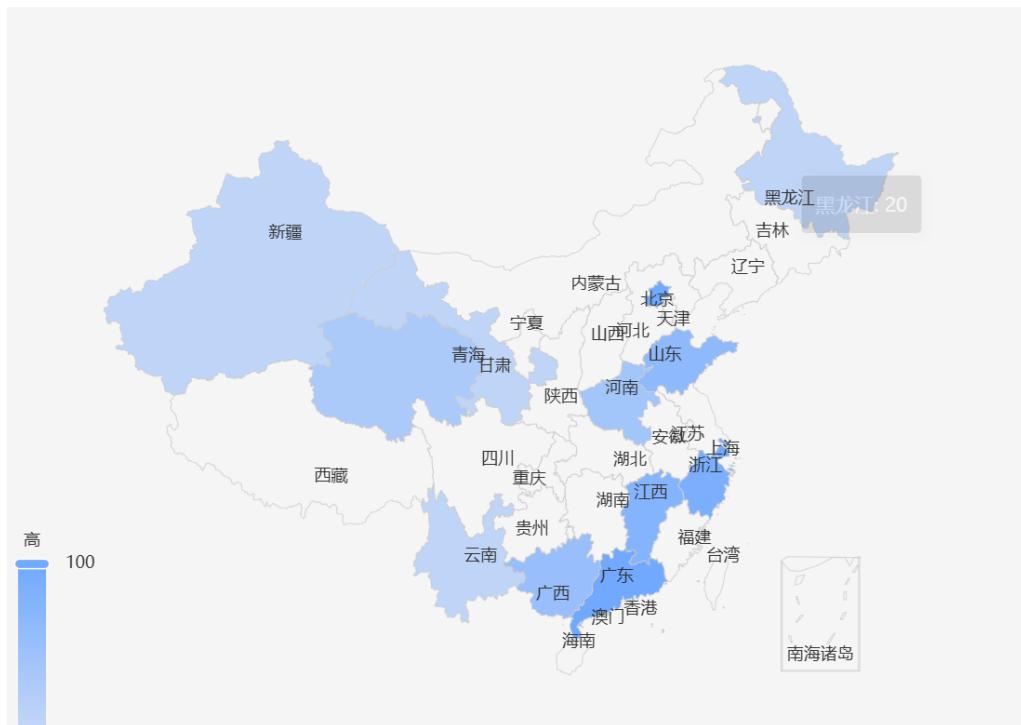


图 3.34: 订单销售

视频播放显示 在海洋牧场中我们通过视频传感器对牧场进行监控，在管理系统中可以同步不同传感器的视频：

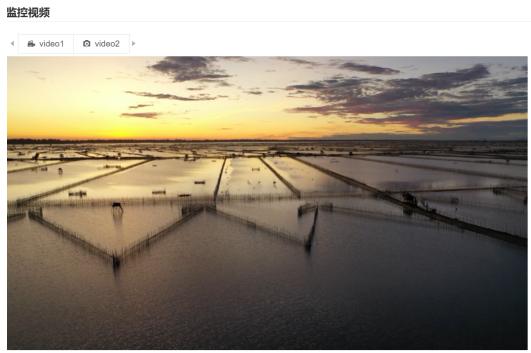


图 3.35: video1

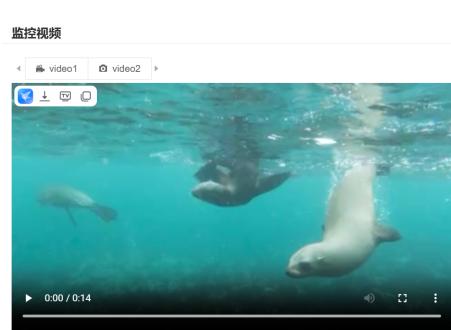


图 3.36: video2

3.2.4 报警与通知模块

在报警与通知模块，由于我们缺少实时数据，因此我们通过让四个指标即盐度（%），溶解氧（mg/L），pH与水温（°C）在每隔一段时间进行自动变化（通过随机数的方式），然后每间隔一段时间去检测四个指标是否超出了阈值。一旦超出阈值，数据将进行实时更新。系统每 30 秒监测一次上述四个指标，超过阈值的情况将自动向所有用户发送警报消息，并将报警的数量、最新时间和信息同步至报警监管模块。

1. 发送报警消息

首先是发送报警消息，这部分实际上我们是结合用户中心的消息管理实现的，首先以用户 farmer123 为例：

The screenshot shows the MySQL Navicat Cloud interface. On the left, there's a profile for 'farmer123' with a photo, bio (@XXX FE Developer), and follower/following counts (1,800 and 666). The main area shows a '消息通知' (Message Notification) panel with a red box highlighting a message from '给farmer123的消息' dated '2024-2-31'. Below it is a '删除全部' (Delete All) button. To the right, a database table named 'message' is displayed with two rows of data:

msg_id	user_id	message	status	date
2	3	给farmer123的消息	read	2024-2-31
10	9	给farmer123的消息	read	2024-2-31

图 3.37: farmer123 用户初始消息通知

如图3.37所示，最开始 farmer123 用户初始消纬度消息没有，已读消息只有一条。

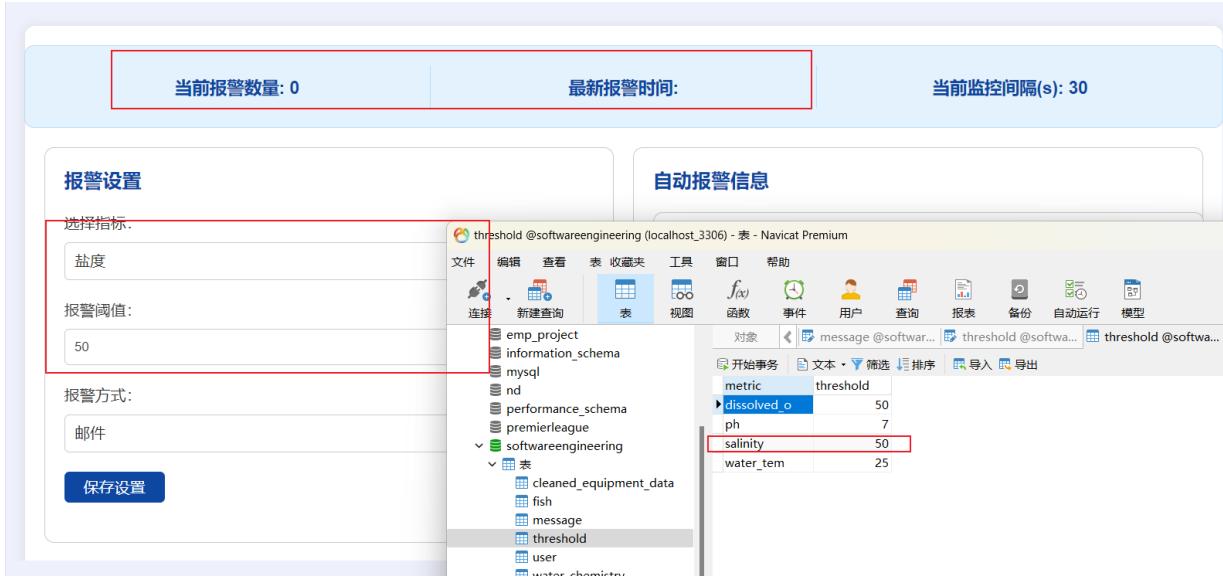


图 3.38: 报警监管初始界面

如图3.38所示，我们使用开发者打开报警监管页面，可以看到左侧的可以调整监控阈值，通知方式。上面最开始由于此时还没开始报警，因此当前报警数量和最新报警时间都是没有的，右侧显示了报警时间间隔为30s。此时的阈值以盐度为例，和数据库中是一致的，后续会进行修改阈值同步更新。

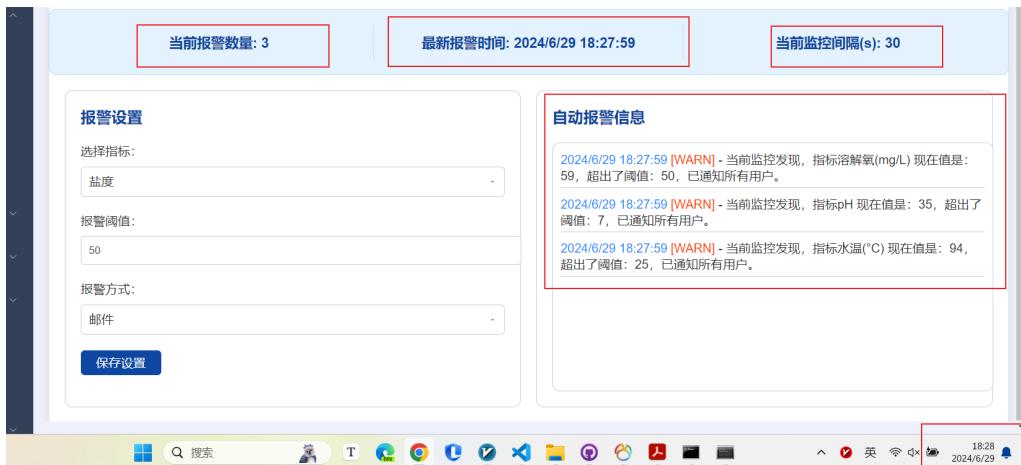


图 3.39: 报警监管

如图3.39所示，可以看到监管界面在相隔了一段时间后，右侧的自动报警消息提示了最新的报警信息，并且上方正确显示了最新的报警数量和最新报警时间。

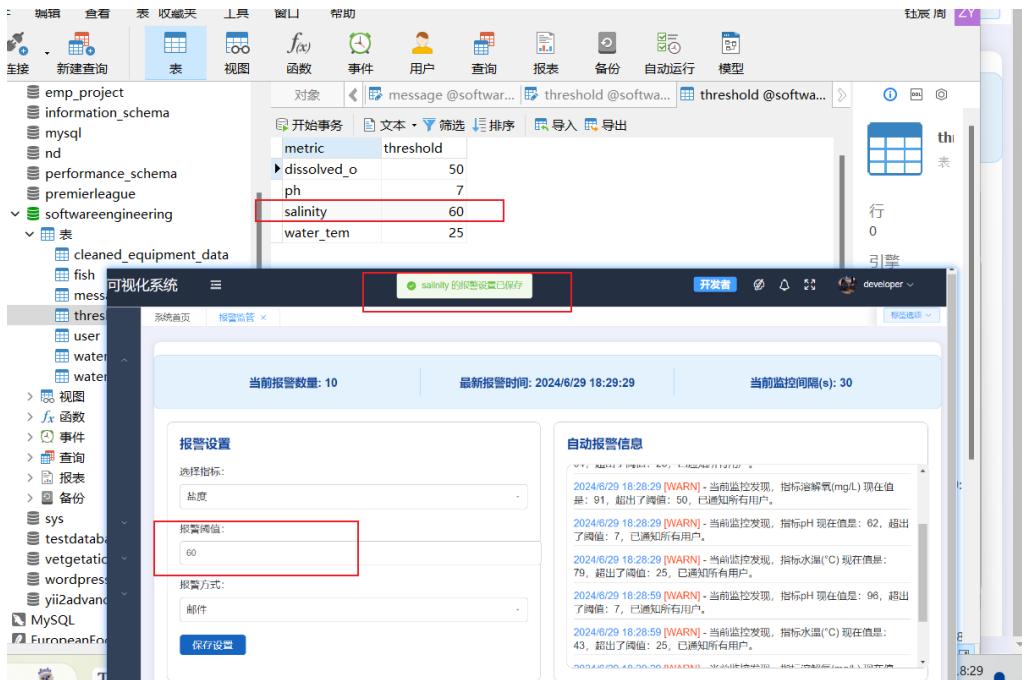


图 3.40: 报警监管

如图3.40所示，可以看到在界面端将盐度阈值直接从 50 调整为 60，数据库也同步进行了更新。

2. 接受报警消息

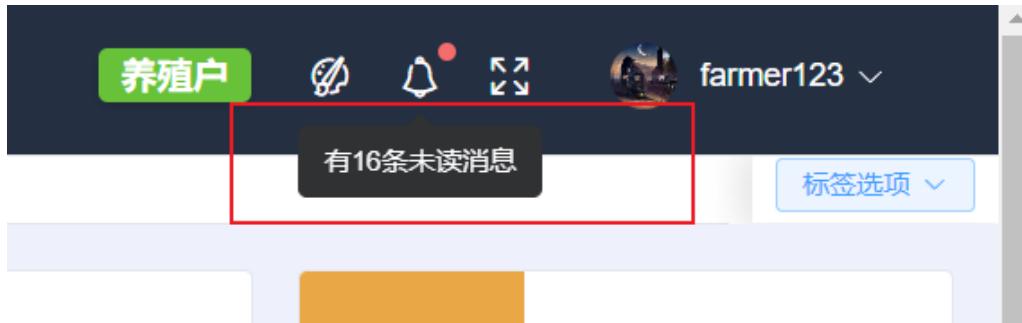


图 3.41: 接受报警信息 1

如图3.41所示，可以看到再次来到 `farmer123` 用户登录，此时最上面的 bar 显示了此时有 16 条未读消息。

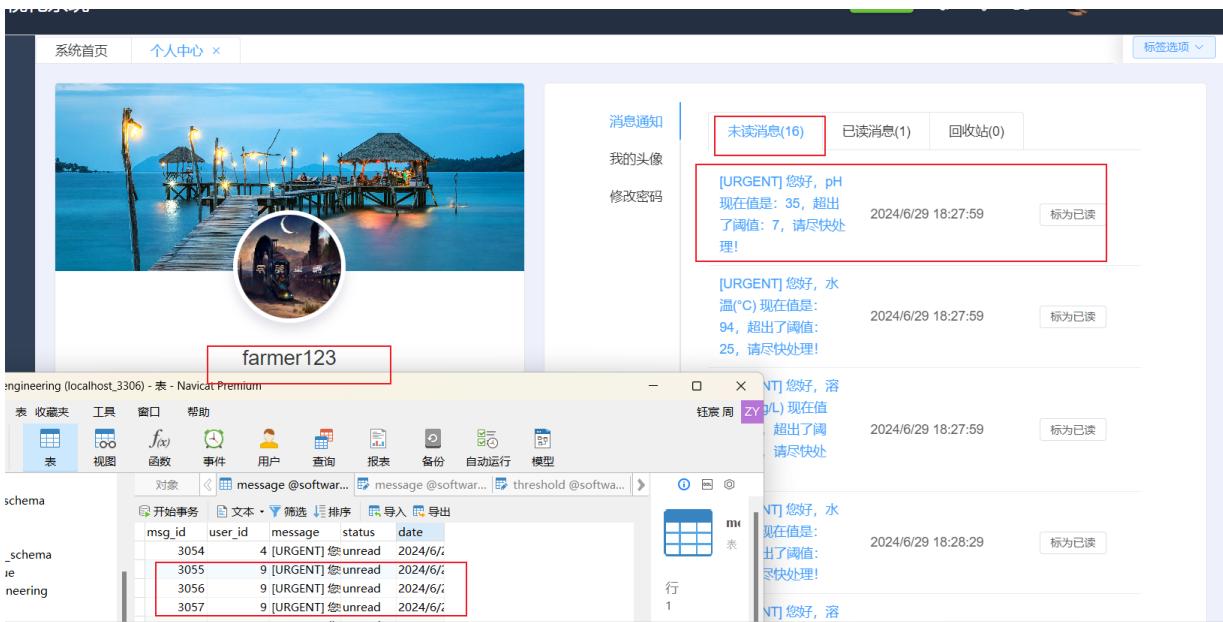


图 3.42: 接受报警信息 2

如图3.42所示，点开具体的消息框，可以看到用户已经收到了许多报警消息的未读消息，同时后端也进行了同步的更新。证明了自动报警通知功能的完整性。

3. 报警监控图表

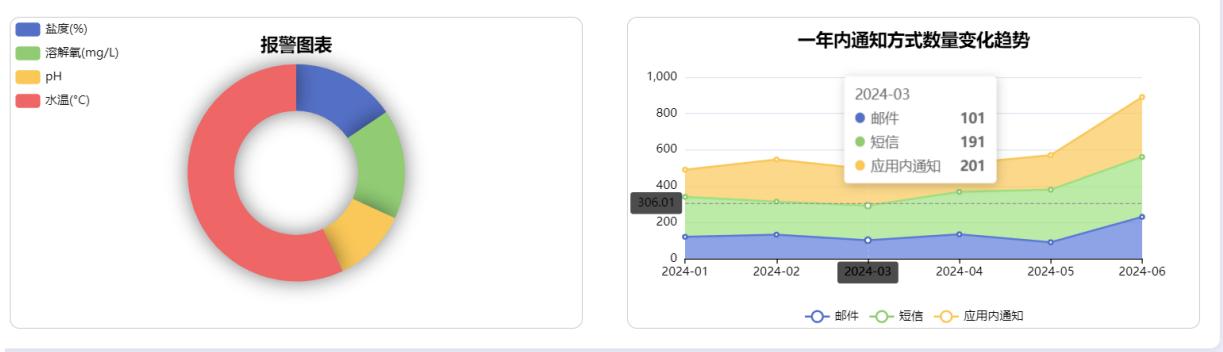


图 3.43: 报警图表

如图3.43所示，可以看到在报警监管页面还有着两个图表分别显示了我们报警的四个阈值即盐度（%），溶解氧（mg/L），pH 与水温（°C）的实时变化情况，以及一年内不同通知方式的数量变化趋势。

3.2.5 智能化模块

在智能化模块，我们需要完成如鱼类图像识别等功能。为了确保智能化模块的丰富性和完整性，我们做了如下几个模块：

AI 智能对话 AI 智能对话模块旨在通过自然语言处理技术，实现人机之间的自然交流。该模块利用先进的对话生成模型，能够理解并回答用户的问题。无论是基础知识查询，还是复杂问题探讨，AI 智能对话模块都能提供准确且及时的回复，提升用户的互动体验和满意度。

我们使用了大语言模型 (LLM) 作为自然语言技术的后端，来实现我们的对话功能，如下图所示：



图 3.44: 智能 AI 问答

基于 YOLO 的图像识别 基于 YOLO (You Only Look Once) 算法的图像识别模块主要用于实时鱼类图像识别。YOLO 是一种高效的目标检测算法，能够快速准确地识别出图像中的目标物体。通过将 YOLO 应用于鱼类图像识别，我们可以实现对鱼类的实时检测，帮助用户快速获取有关鱼类的信息，提高管理和监控效率。

我们采用最新的 YOLO v10 模型作为架构，使用了 YOLOv10-X 的高精度较大模型来进行图像分类，但鉴于数据集的缺少以及算力资源的不足，我们的 YOLO 模型可以对鱼、人等物体进行分类，但暂时无法对具体的鱼类进行细分。我们使用一个图片上传模块，供用户上传需要检测的图片，来达到图像分割识别的效果，如下图所示：



图 3.45: 用户上传的图片



图 3.46: YOLO 识别效果

基于 YOLO 的动态追踪 渔场需要动态实时监测信息，而动态信息通常是采用视频流的方式提供的，因此我们有必要对视频进行动态追踪，这里我们仍然采用的是 YOLO v10 架构，YOLO 在视频流的帧追踪表现出良好的性能和效果，如下图所示：

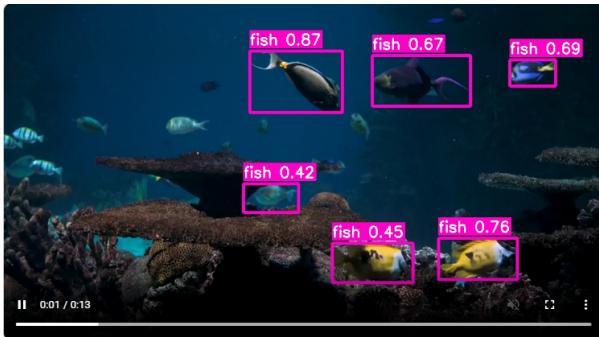


图 3.47: 水下牧场动态识别

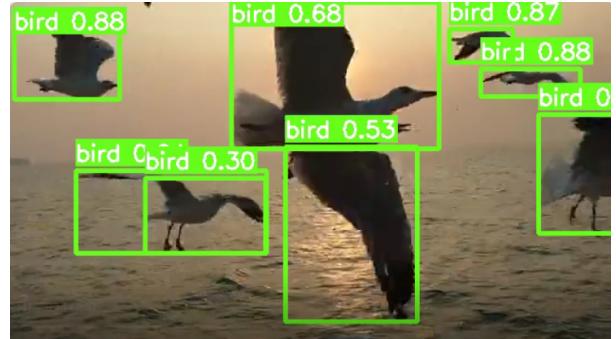


图 3.48: 水上牧场动态识别

可以看到，我们可以实时地对鱼的轨迹进行捕捉追踪，并且识别到是鱼，是鸟，亦或是人等生物，这为海洋牧场的监控提供了有力的技术支撑。

基于机器学习的指标预测 基于机器学习的指标预测模块旨在对海洋生态牧场的环境指标进行精准的预测。通过对历史数据的分析和建模，该模块能够预测出未来一段时间内的关键环境参数，如温度、盐度和 pH 值等。这些预测结果对于科学管理和优化养殖策略具有重要意义。

在模型构建过程中，我们采用了长短期记忆网络（LSTM）这一先进的时序预测算法。LSTM 模型具有处理长时间依赖数据的优势，能够有效捕捉环境参数的变化趋势和规律，从而提供高精度的预测结果。

为了更直观地展示预测结果，我们在系统界面中集成了图表展示功能。用户可以通过图表清晰地看到各项环境参数的变化趋势，便于进行决策和调整。以下折线图展示了对温度、盐度和 pH 值的预测结果。这些数据是通过运行 LSTM 模型生成的，并在前端界面中进行可视化显示。通过这种可视化手段，用户可以快速掌握环境变化情况，并依据预测结果及时采取相应措施，确保生态牧场的稳定和高效运营。

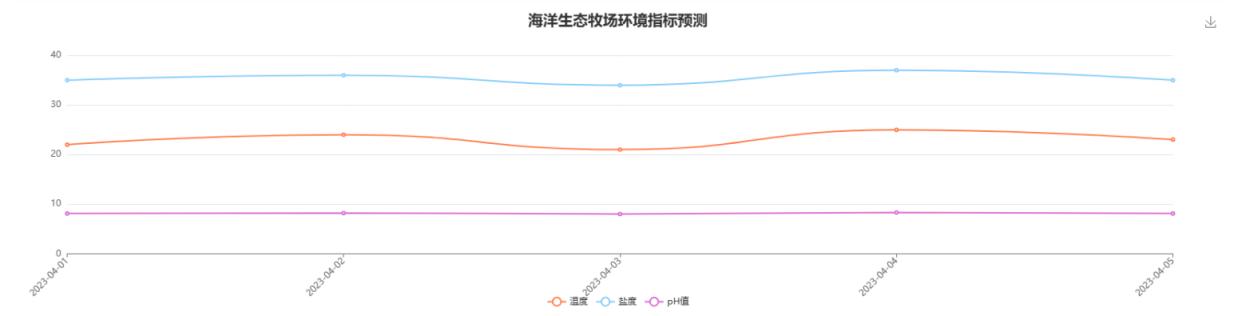


图 3.49: 未来指标预测

3.2.6 用户中心

在这部分我们主要实现了三个用户中心功能，包括消息通知，我的头像查看与修改，以及修改密码。接下来依次测试三个功能。

1. 消息通知

首先是消息通知，在此我们通过将表 8 所示的消息数据直接存入后端，并通过用户在前端不同的操作对应修改数据库后端中消息的状态等，来实现。具体而言：

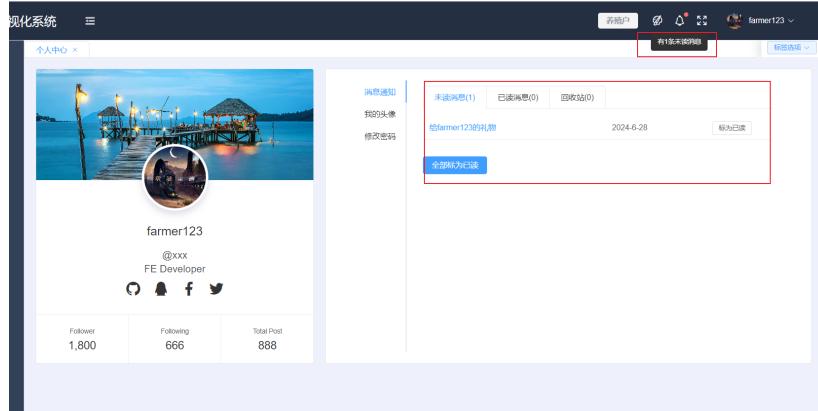


图 3.50: 消息通知测试 1

首先如图3.50所示，可以看到此时用户在未读消息位置有着一个消息。点击右侧的读，或者下方的全部标记为已读后：



图 3.51: 消息通知测试 2

如图3.51所示，此时消息就到了我已读消息的地方，我们再将其删除后：



图 3.52: 消息通知测试 3

如图3.52所示，删除消息后刚才的消息就到了已删除的界面，此时如果点击清空回收站就会将消息彻底删除，从数据库中直接删除。

2. 上传头像

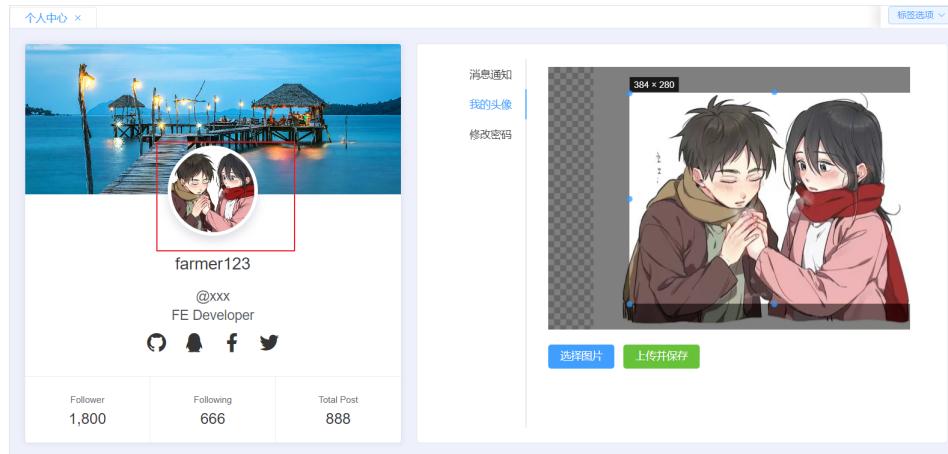


图 3.53: 上传头像

如图3.53所示，可以看到也能够实现对用户头像的上传，左侧的用户头像就发生了变化了。

3. 修改密码

最后是修改密码，在下面进行功能测试：

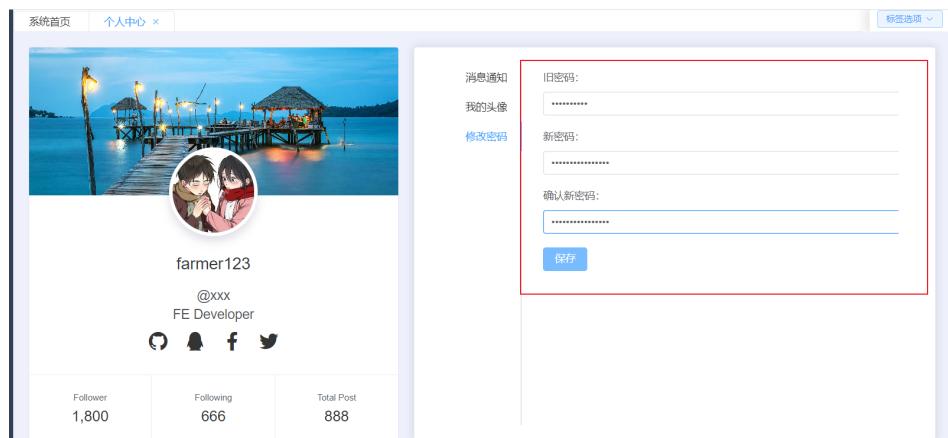


图 3.54: 修改密码

如图3.54所示，可以看到我们将密码改为了另一个较长的版本，现在重新登陆测试新的密码：

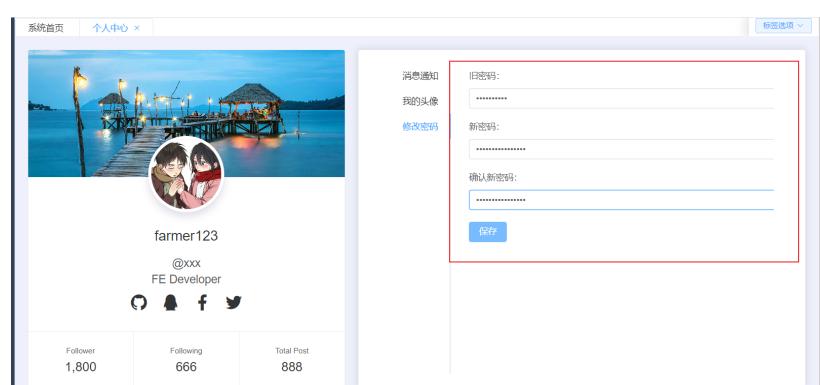


图 3.55: 新密码测试 1

```

Permiss 0,001,002 change to 0,001,002
Loading permissions for role: farmer
Default list before loading: ▶ Proxy(Object)
Loaded permissions for role farmer: ▶ Proxy(Array)
User logged in as farmer with permissions:
  ▼ Proxy(Array) i
    ▶ [[Handler]]: MutableReactiveHandler
    ▶ [[Target]]: Array(3)
    [[IsRevoked]]: false
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJmYXJtZXIxMjM1LCj1eHA10jE3 login.vue:101
MTk3NjI5ODF9.-uC1gIJGj188BQ4vXfjyikoZ09Y4Pg8tjpcPizzFw6A
User Store.role: farmer
Permissions: ▶ Proxy(Array)
header's role: farmer

```

图 3.56: 新密码测试 2

如图3.55和3.56所示，可以从日志和登陆界面看到新的密码也登录成功，证明了用户修改密码后登录成功。

3.2.7 用户管理模块

管理员在用户管理模块中可以对用户信息进行增删改查。

(a) User Data View

序号	用户名	邮箱	角色	手机号	姓名	性别	年龄	操作
1	admin123	1980437320@qq.com	admin					编辑 禁用 删除
2	2	admin123	user					编辑 禁用 删除
3	3	analyt123	analyst					编辑 禁用 删除
4	4	farmer123	farmer					编辑 禁用 删除

(b) User Data Delete

序号	用户名	邮箱	角色	手机号	姓名	性别	年龄	操作
1	2	admin123	user					编辑 禁用 删除
2	3	analyt123	analyst					编辑 禁用 删除
3	4	farmer123	farmer					编辑 禁用 删除

(c) User Data Modify

序号	用户名	邮箱	角色	手机号	姓名	性别	年龄	操作
1	2	admin123	user	110119				编辑 禁用 删除
2	3	analyt123	analyst					编辑 禁用 删除
3	4	farmer123	farmer					编辑 禁用 删除

(d) User Data Add

序号	用户名	邮箱	角色	手机号	姓名	性别	年龄	操作
1	2	admin123	user	110119				编辑 禁用 删除
2	3	analyt123	analyst					编辑 禁用 删除
3	4	farmer123	farmer					编辑 禁用 删除
4	5	cteide	管理员	111@qq.com				编辑 禁用 删除

图 3.57: 不同用户管理功能

上述四张图片完整展示了用户管理模块的操作方式以及功能实现。

3.3 性能测试

在此我们测试系统的性能，即测试多个用户同时登录的话是否仍能保证正常运行。我们首先分别注册了普通用户，养殖户，数据分析人员，系统管理人员四个用户，然后再打开开发者。同时登陆后打开其中一个页面：

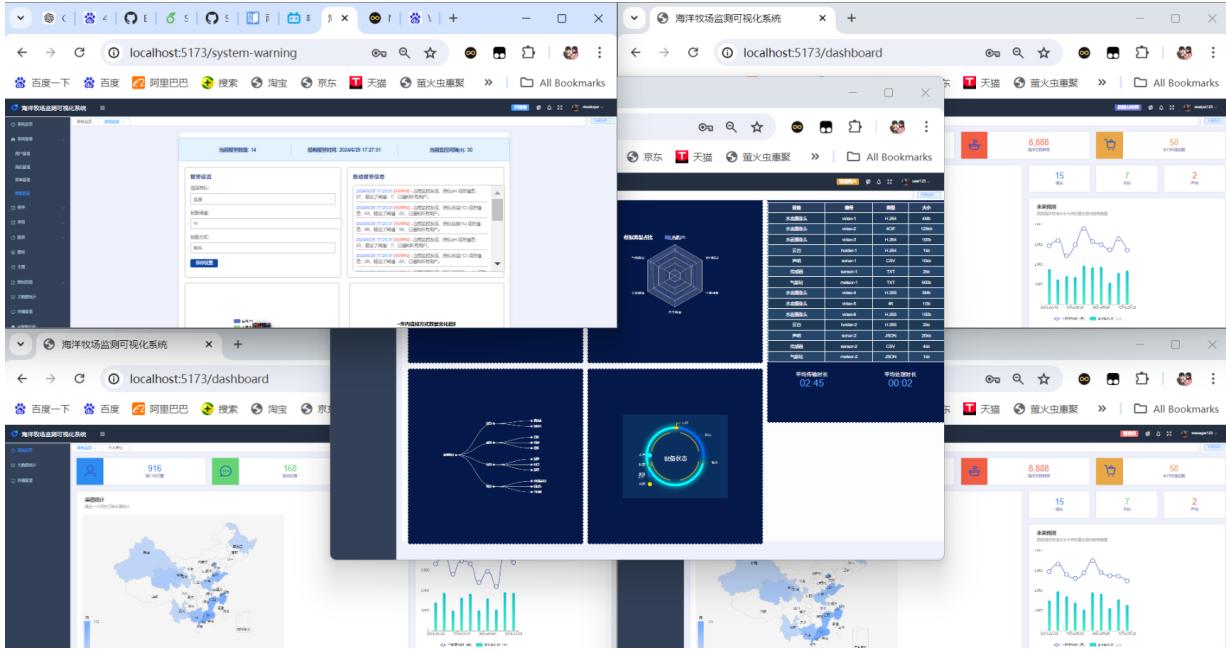


图 3.58: 性能测试结果

如图3.58所示，可以看到此时即使多开同时登陆多个用户，也没有出现明显的卡顿，左上角的报警功能还可以正常地向所有用户发送报警消息。

对象	message @softwareengineering	message @softwareengineering	threshold @softwareengineering	threshold @softwareengineering
开始事务	文本	筛选	排序	导入
msg_id	user_id	message	status	date
2137	9 [URGENT]	您unread		2024/6/1
2138	9 [URGENT]	您unread		2024/6/1
2139	1 [URGENT]	您unread		2024/6/1
2140	1 [URGENT]	您unread		2024/6/1
2141	1 [URGENT]	您unread		2024/6/1
2142	1 [URGENT]	您unread		2024/6/1
2143	3 [URGENT]	您unread		2024/6/1
2144	3 [URGENT]	您unread		2024/6/1
2145	3 [URGENT]	您unread		2024/6/1
2146	3 [URGENT]	您unread		2024/6/1
2147	4 [URGENT]	您unread		2024/6/1
2148	4 [URGENT]	您unread		2024/6/1
2149	4 [URGENT]	您unread		2024/6/1
2150	9 [URGENT]	您unread		2024/6/1
2151	4 [URGENT]	您unread		2024/6/1
2152	9 [URGENT]	您unread		2024/6/1
2153	9 [URGENT]	您unread		2024/6/1
2154	9 [URGENT]	您unread		2024/6/1
2155	1 [URGENT]	您unread		2024/6/1
2156	1 [URGENT]	您unread		2024/6/1
2157	1 [URGENT]	您unread		2024/6/1

文件	编辑	查看	窗口	帮助
开始事务	文本	筛选	排序	导入
user_id	username	password	name	gender
1	xwfwjn	\$2b\$12\$E4dv	谢斐菲	female
3	manager	\$2b\$12\$R1u	(Null)	(Null)
4	aaa123	\$2b\$12\$DPg	(Null)	(Null)
9	farmer123	\$2b\$12\$YLR	(Null)	(Null)
10	analyst123	\$2b\$12\$MD3	(Null)	(Null)
11	manager123	\$2b\$12\$bkX	(Null)	(Null)
12	user123	\$2b\$12\$Ozux	(Null)	(Null)

图 3.59: 性能测试准备

如图3.59所示，数据库中存在着对应的新的用户以及发送的新的报警消息。证明本智慧海洋牧场系统在负载较高的时候也具有一定的稳定性。

4 项目管理

4.1 参与人员及分工

本系统参与开发人员如下表所示。

姓名	学号
吴静	2113285
周钰宸	2111408
齐明杰	2113997
杨浩甫	2113824
谢雯菲	2110803

本小组由**周钰宸**担任组长，对于本次项目的小组分工，我们在项目的不同推进阶段对任务进行了不同的具体划分。

- 在项目的初始代码构建部分，由齐明杰、周钰宸、谢雯菲负责后端的代码构建、项目框架搭建、服务器部署后端接口测试与调整，从而保证后端系统的稳定性以及功能的完善性。由齐明杰、杨浩甫、吴静负责前端页面搭建、前端接口构造，以设计出用户友好、界面简洁美观的前端页面，并与后端接口进行有效的数据交互。
- 在项目的测试完善部分，由齐明杰、杨浩甫、周钰宸负责数据表的维护与更新、环境测试、性能测试，以保证数据的准确性和系统的高性能。由谢雯菲、吴静负责数据约束、异常处理实现、功能测试与完善、页面布局优化，以确保系统在各种情况下的稳定性以及良好的用户体验。
- 在项目的最终提交部分，由五人共同负责项目说明文档的撰写与 demo 展示视频等工作，以保证尽可能详尽地展示本项目的成果和亮点，确保本项目的完整性。

4.2 项目进展记录

在项目开展的四个月内，我们的项目推进记录如下。

- 在第一个月中，我们小组各自学习了 python、html 等语言，对项目进行了理论上的初步构造，完成了各式 uml 图的构造，为后续项目的进一步推进奠定了理论和技术基础。
- 在第二个月内，我们结合课程的具体开展情况，逐步推进，开始进行基本的代码撰写，完成了项目的初步构造。通过完成课堂上老师部署的需求分析、软件设计等测试作业，我们对大作业中的可行性分析、需求分析与系统设计、系统测试等模块，有了更深的理解。
- 在第三个月中，我们对项目进行了充分的测试。我们运用测试工具和技术对系统进行了各种测试包括功能测试、性能测试和异常处理测试。并通过测试结果的分析和设想的运行效果的对比，我们对项目进行了进一步的优化，修复了存在的问题，对项目的各式扩展功能进行了具体实现。
- 在最后一个月，我们小组共同着手编写项目说明文档并进行了成果汇报展示与总结。我们详细记录了项目的整体架构、功能模块的实现细节、遇到的问题和解决方案，以及项目的成果和亮点。我们还进一步优化了前后端代码，努力达到最佳呈现效果。

总体而言，通过四个月的努力和合作，我们小组在项目的不同阶段取得了不断的进展。我们从理论学习到具体实践，逐步完成了项目的各个方面。各个成员都充分发挥了自己的专长和能力，共同推动项目的发展。我们面对项目完成过程中遇到的各个挑战，积极解决问题，不断改进和优化，最终取得了良好的项目成果。**具体的我们项目推进记录可以见我们的前后端仓库：**即前端 github 仓库<https://github.com/Starlight0798/Software-Engineering-frontend>与后端 github 仓库<https://github.com/Starlight0798/Software-Engineering-backend>。

4.3 项目管理工具

在本项目中，我们团队使用了以下工具来管理和协作开发工作，以提高效率并确保项目的顺利进行。

- **Visual Studio Code (VS Code):** 我们主要使用 VS Code 作为代码编辑器。它提供了强大的代码编辑和调试功能，并支持多种编程语言的插件，极大地提升了我们的开发效率。
- **Git 及 GitHub:** 为了进行版本控制和团队协作，我们使用了 Git 和 GitHub。通过 Git，我们可以追踪代码的历史变更，便于发现和回退到之前的版本。GitHub 则提供了一个平台，使我们可以共享代码、提交 Pull Request、进行 Code Review，并通过 Issue 跟踪项目中的问题和任务。

通过这些工具的综合使用，我们有效地管理了项目的各个方面，从代码开发到版本控制，再到团队协作。每个工具都在不同的环节发挥了重要作用，帮助我们更好地完成了项目。

5 用户手册

5.1 引言

5.1.1 编写目的

本用户手册提供了海洋牧场管理系统的全面指南，旨在帮助用户有效利用系统的各项功能，实现高效的海洋牧场管理。系统集成了多个模块，涵盖了数据管理、设备监控、环境监测、智能分析等多方面功能。

数据管理模块负责存储和管理海洋牧场的所有数据，包括用户数据、环境物理数据、环境化学数据、鱼类数据、消息数据以及阈值数据。系统采用结构化的数据库设计，确保数据的完整性和可追溯性。每个数据表格都具有清晰的字段定义和类型说明，用户可以通过直观的界面进行增删查改操作。

设备监控模块用于实时监控海洋牧场中的各类设备。用户可以查看设备的运行状态、地理位置信息、历史维护记录以及保修到期时间等详细信息。系统提供了多种数据展示形式，包括柱状图、折线图和地理分布图，帮助用户全面了解设备的运行状况并及时进行维护和管理。

环境监测模块主要负责监测海洋牧场的环境参数，包括水温、水质、空气质量和气压等。系统通过不同类型的传感器采集数据，并实时显示在监控界面上。用户可以通过系统设置报警阈值，当环境参数超出正常范围时，系统会自动发出预警，帮助用户及时采取措施。

智能分析模块结合了先进的 AI 技术，为用户提供数据分析和辅助决策支持。系统可以对历史数据进行分析，预测未来趋势，并生成详细的分析报告。用户还可以通过系统上传鱼类图像，利用 AI 技术识别鱼类种类，从而更好地管理鱼类资源。

用户管理模块用于管理系统的所有用户和角色。管理员可以在该模块中添加、删除、修改用户信息，分配用户角色和权限，确保系统的安全性和可操作性。同时，用户管理模块还支持对用户操作日志的记录和查询，方便管理员进行审计和追踪。

报警管理模块用于设置和管理各类报警信息。用户可以根据实际需求设置不同的报警阈值，当监测数据超出设定范围时，系统会自动生成报警信息，并通过消息中心通知相关用户。用户可以在该模块中查看和处理所有报警信息，确保及时应对各种异常情况。

5.1.2 项目背景

海洋牧场管理系统项目旨在为现代化的海洋牧场管理提供全面的信息化解决方案。随着全球水产养殖业的迅速发展，传统的管理方式已无法满足当前高效、精准和可持续发展的需求。为了提升海洋牧场的管理效率和生产力，本项目结合了物联网、数据分析、人工智能等前沿技术，开发了一套集数据采集、实时监控、智能分析于

一体的综合管理平台。系统采用了 CSS+HTML+JavaScript+Vue3 作为前端技术，后端使用 Python 技术栈和 FastAPI 框架。

5.1.3 术语和缩略词

- 智慧水产养殖场：智慧水产养殖场在普通水产养殖场的基础上，可通过系统实时观察到水产养殖场的情况，并且可以通过系统对水产养殖场进行实时地监控与分析，是结合市场需要与新时代科技的一体化软件。
- 养殖场管理员：管理水产养殖场的人，有权限通过系统对水产养殖场进行实时的管理。
- 数据分析师：对得到的具体数据进行分析，给出建设性建议的专业人员。
- 可视化界面：指将科学的数据转换为饼图或条状图等更直观、更便于理解的图像，并在系统界面显示。
- 大数据分析：指借助生成式人工智能，对已有数据进行分析；或是借助图像识别人工智能，对已有图像进行识别，从而执行更可靠、更高效的规划与决策。

5.2 软件概述

5.2.1 软件目标

智慧水产养殖系统的目标是提供一个统一管理水产养殖场中的设备、环境、生物的系统，使养殖场管理员与数据分析师能够更直观地观察养殖场情况并进行分析，同时提供给普通用户对水产养殖场的直观了解。

5.2.2 软件功能

- 用户信息录入
- 养殖场主要信息可视化
- 实时数据处理与分析
- 养殖设备实时管理
- 实时报警与通知
- 大数据分析
- AI 智能识别

5.2.3 软件性能

智慧水产养殖可视化系统具有以下性能特点。

- 高并发性：能够同时处理大量用户的注册、登录、查询和上传下载等操作。
- 数据安全性：采用安全的数据传输协议和加密技术，保护用户的个人信息和养殖场数据的安全。
- 系统稳定性：具备良好的系统稳定性和容错能力，能够处理异常情况并进行相应的恢复。

5.3 软件使用说明

5.3.1 安装

海洋牧场可视化系统是基于 Web 的应用程序，无需进行安装。只需确保您的计算机连接到互联网，并使用兼容的 Web 浏览器即可访问系统。

5.3.2 使用

在系统界面用户可以进行多项操作：

1. 注册与登录：根据用户类型选择注册或登录功能，填写相应信息即可完成注册或登录操作。
2. 查看主要信息：在主界面可对水产养殖场的主要信息进行查看，并跟进实时消息。
3. 修改个人信息：在个人信息界面可对个人信息进行修改，并且可以接收到系统的推送通知。
4. 养殖管理操作：具有管理权限的用户可浏览养殖管理界面，对养殖场的相关设备和数据进行管理。
5. 数据分析操作：具有数据分析权限的用户可浏览数据分析界面，调用大数据模型对相关数据进行分析，并且可以实现数据的上传和下载。
6. 报警操作：系统会对相应的数值进行监测，如果某项数值超出一定阈值，则会向管理人员发出预警信息。
7. 人工智能识别：具有相应权限的用户可以访问大数据界面，调用 AI 模型对相关鱼类图像进行识别，并得到相应的建议。

5.3.3 出错与恢复

如果在使用过程中遇到错误或系统异常，请尝试以下方法进行恢复：

1. 重新登录系统。
2. 清除浏览器缓存并刷新页面。
3. 检查网络连接是否正常。

5.3.4 求助查询

如果您遇到问题无法解决，请查阅系统提供的常见问题解答页面。如仍无法解决，您可以联系系统管理员或相关技术人员以获取帮助。

5.4 运行说明

5.4.1 运行步骤

- 打开您的 Web 浏览器。
- 输入系统的 URL 地址。
- 在系统首页进行相应操作，如注册、登录等。

5.4.2 操作命令一览表

下表中列出了系统中常见的操作。

操作	描述
注册	注册为新用户
登录	用户登录系统
编辑个人信息	修改个人信息
查看首页	可以查看相关信息
养殖管理	对养殖场相关设备进行监控和操作，查看相关数据
大数据分析	查看系统详细数据，进行分析
AI 识别分析	调用 AI 接口对数据进行分析，识别相关鱼类图像
数据上传下载	下载 AI 分析数据，上传数据到后端数据库
接收预警信息	接收系统推送的预警信息

表 12: 常见操作

5.5 用户操作示例

5.5.1 注册

1. 在系统首页点击“立即注册”按钮
2. 填写个人信息，包括用户名、邮箱、密码以及注册的角色类型（包括普通用户、养殖户、系统管理员、数据分析师四类），每一类角色对应不同的权限
3. 点击“注册”提交注册信息并完成注册



图 5.60: 注册界面

user_id	username	password	name	gender	age	phone_num	email	role
5	w1234567	\$2b\$12\$mvu(Null)	(Null)	(Null)	(Null)	(Null)	1052767108	admin

图 5.61: 注册完成

这里在数据库中存储的数据是经过密文存储的，无法直接获取，有一定安全性保障。

5.5.2 登录

1. 在系统首页进行登录
2. 输入用户名、密码和角色
3. 点击登录，即可进入对应角色的界面



图 5.62: 登录界面

5.5.3 查看首页

由于不同角色用户拥有的权限不同，于是不同角色用户能够查看的主界面不同，即系统首页不一样，核心权限如下：

1. 系统管理员
 - 系统管理
 - 主题设置
 - 大数据统计
2. 数据分析师
 - 大数据统计

- AI 智能分析

3. 养殖户

- 大数据统计
- 养殖管理

4. 普通用户

- 大数据统计

页面呈现如下：

The figure consists of four sub-screenshots labeled (a) through (d), each showing a different user's view of the system's main dashboard.

- (a) 系统管理员 (System Administrator):** Shows the full dashboard with sections for 管理员 (Administrator), 大数据统计 (Big Data Statistics), and AI智能分析 (AI Intelligent Analysis). It includes a map of China with regional data overlays, a line chart for future prediction, and a bar chart for sensor counts.
- (b) 数据分析师 (Data Analyst):** Shows the same dashboard structure but with a simplified navigation bar at the top.
- (c) 养殖户 (Farmer):** Shows the dashboard with a simplified navigation bar at the top.
- (d) 普通用户 (General User):** Shows the dashboard with a simplified navigation bar at the top.

图 5.63: 不同用户登录界面

在首页部分，我们实现了多个关键性数据的展示，包括当前用户访问量、系统消息、海洋生物种类和今日环境指数，最近一个月的订单来源统计，海洋牧场未来十天的营业额与物种数量预测，三种传感器数量的展示，最新牧场动态，海洋牧场销量最高的品种 Top5，以及最近一个月的访客统计。



图 5.64: 首页部分界面

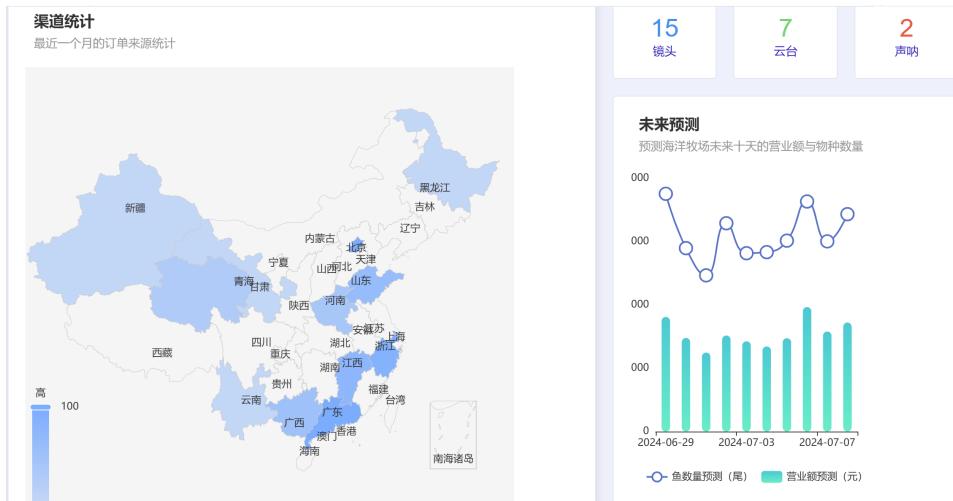


图 5.65: 首页部分界面



图 5.66: 首页部分界面

5.5.4 养殖管理

在养殖管理界面，我们致力于以更清晰明了的界面将各个核心信息呈现给养殖户。

1. 监控视频界面呈现了当前各渔场的监控视频，养殖户还可以选择调整摄像头、灯光、清洁刷以及设置是否开启预警；
2. 水文气象界面将盐度、溶解氧、pH 和水温四个变量的实时情况呈现在面前；
3. 定位部分则是将当前渔场位置在地图上显示了出来；
4. 鱼群数量分布曲线则可以查看各个不同种类鱼群随着时间的变化趋势；
5. 鱼群属性分布曲线可以查看不同渔场不同种类鱼群的不同属性的信息，同时还支持导入和导出数据，更加方便统计。

各个部分界面如下：

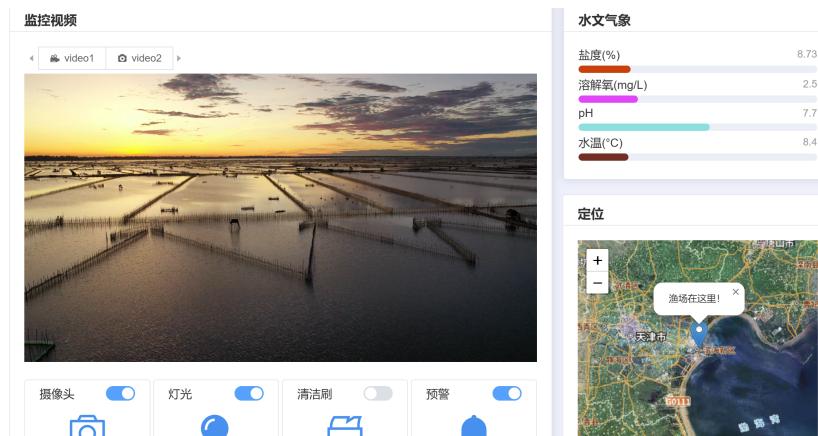


图 5.67: 监控视频等界面展示



图 5.68: 数据展示部分

5.5.5 大数据分析

在大数据分析部分，我们使用了多种分析工具对数据进行深入挖掘和可视化展示。我们通过这些工具绘制直观的图表，展示数据的趋势、分布和关系。此外，我们还对部分数据进行了人工智能（AI）预测，为用户提供未来趋势的参考。在数据分析页面，可以直观了解海洋牧场的生物、设备、气象信息。

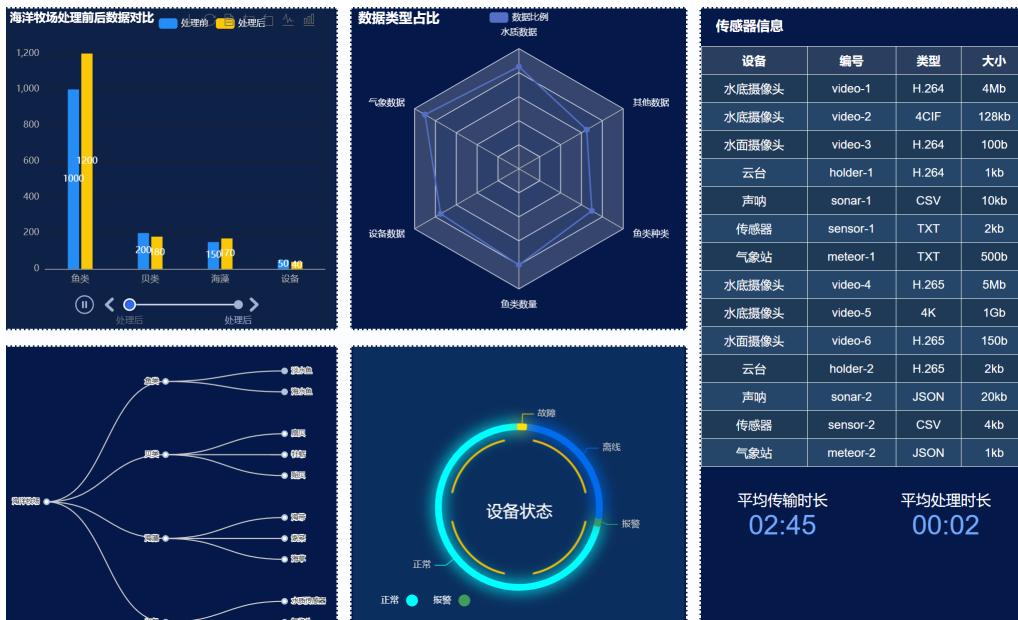


图 5.69: 数据分析



图 5.70: 数据分析 1



图 5.71: 数据分析 2

5.5.6 AI 识别分析

在智能化 AI 处理界面，我们致力于使用多种角度的 AI 工具来辅助用户进行决策，或观测等工作，以提高工作效率。我们把使用的 AI 工具分为以下几个部分：

AI 智能对话 用户可以输入需要回答的问题，下面也已经列出了几个常见的问题，来让大模型提供回答：

点击发送按钮后，用户可以收到大模型的回复，作为参考：

图像识别 用户可以选择上传一张图片来识别（图片中可以有如鱼，鸟，人，动物等物体，均可识别），点击上传后，后端会自动调用 YOLO 高精度模型来进行识别，并且传回识别出的图像，如下图所示：



图 5.72: 大模型回复



识别结果:



识别结果:

图 5.73: 识别人类

图 5.74: 识别鱼类

动态轨迹追踪 我们使用了视频动态追踪技术，可以对实时监控视频进行动态追踪。但限于数据和缺乏真实的实时监控设备，我们这里采用录屏的方式作为源文件，使用 YOLO 技术生成的动态追踪视频作为展示，分为水

上和水下两部分，分别如下所示：



图 5.75: 动态轨迹追踪

5.5.7 数据上传下载

数据上传 在养殖管理界面，用户可上传指定内容结构的 csv 文件至网站。点击发送数据按钮后，网站将读取该文件传递给后端数据库，将用户的文件数据添加进数据库中。

1	年,月,日,鱼,范围[(长度cm)],范围[(重量kg)],数量
2	2025,1,6,Bluegill,69-72,1.85-1.92,13
3	2025,1,6,Bluegill,39-42,3.00-3.22,20
4	2025,1,6,Bluegill,33-36,0.58-0.65,10

(a) 用户上传文件内容格式

(b) 用户选择上传内容

(c) 用户上传结果

2025-01-10	... Bluegill	33-36	0.58-0.65	10
2025-01-10	Bluegill	39-42	3.00-3.22	20
2025-01-10	Bluegill	69-72	1.85-1.92	13

(d) 数据库后端更新

图 5.76: 用户上传数据

数据下载 同样在养殖管理界面，用户可以点击导出数据按钮导出当前波形图的数据，浏览器将把数据下载为 csv 文件。



图 5.77: 用户下载数据

5.5.8 发送与接收预警信息

发送预警消息 其中在系统管理人员和开发者可见的预警监管页面中，用户可以设置不同指标的报警阈值，同时页面下方也有实时的指标数值方便查看随时对针对性地对阈值进行调整。监控默认是 30 秒对四个指标进行一次，若出现问题会自动向所有当前已经注册的用户发送报警消息，同时将报警结果，报警次数，最新报警时间都第一时间显示在界面里面，方便系统管理人员和开发者进行管理。



图 5.78: 监管与自动发送预警消息

如图5.78所示，当系统管理人员和开发者在预警监管界面开启后，可以通过左侧选择对应的指标进行阈值设置，右侧的会在发送给所有用户报警信息后也通知一下当前正在看该报警界面的用户。另外上面的信息框也能

看到当前报警数量和最新报警事件等，都是实时同步的。

接收预警消息 在这部分接收预警信息，我们主要结合了用户中心实现的接收预警功能。在用户中心部分，用户可以查看自己会收到的新消息，包括已读，未读和已删除。因此刚才报警的消息就会第一时间发送给对应的用户，用户就可以在系统管理界面查看对应消息，由此实现对预警消息的接收。

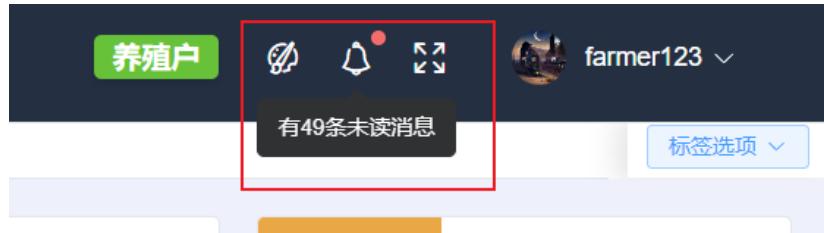


图 5.79: 未读消息提示

如图5.79所示，其它用户可以在登陆后在自己的上方消息上看到对应的未读消息提示。点击后来到个人中心。

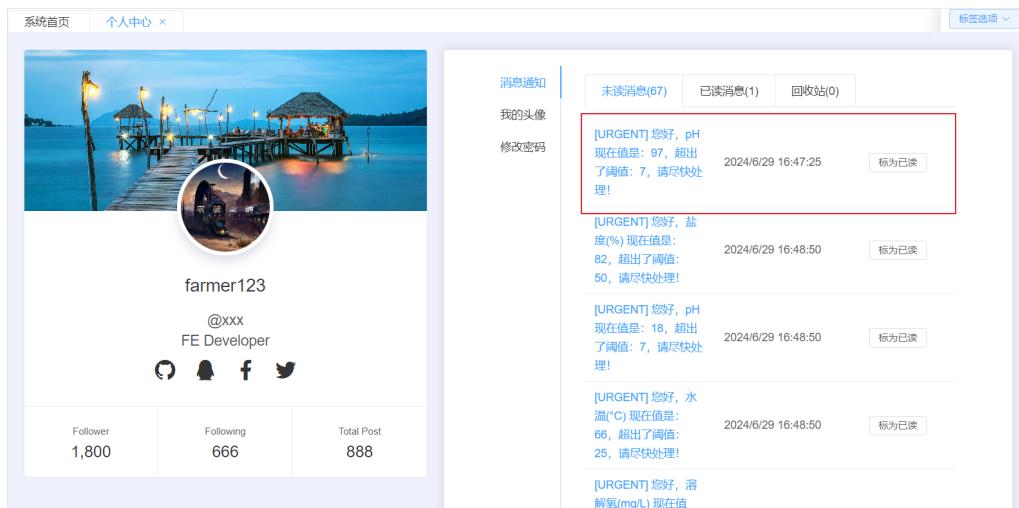


图 5.80: 接收预警消息

如图5.80所示，点进个人中心的消息界面就可以看到刚刚超过预警的信息提示了包括报警的时间，报警的内容，可以向正常的消息处理一样对其进行读取，删除回复等的处理。

5.5.9 用户个人中心

在用户的个人中心中，用户可以对自己的头像进行修改：

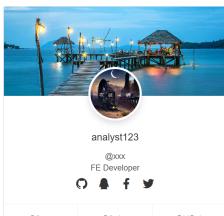


图 5.81: 修改之前

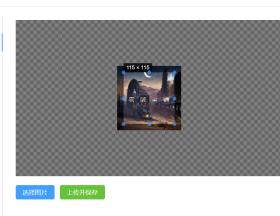


图 5.82: 修改之后

我们也可以对密码进行修改：



图 5.83: 密码修改

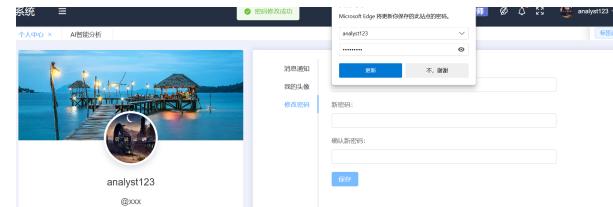


图 5.84: 修改成功

5.6 总结

海洋牧场管理系统通过多模块协同工作，为用户提供了一站式的管理解决方案。系统采用先进的技术和友好的用户界面，帮助用户高效管理海洋牧场的各项资源，实现数据的集中管理、设备的实时监控、环境的全面监测以及智能化的决策支持。希望本用户手册能帮助您充分利用海洋牧场管理系统的各项功能，为您的工作带来更多便捷和高效。

致谢

本学期软件工程课程到此结束了，感谢刘老师一直以来的讲解，刘老师一直在课上耐心的为我们传递这一门十分实用的课程知识，并通过一次次实验课让我们掌握了实用的工程工具以及习惯，让我们受益匪浅。

我们还想要感谢助教韩学姐，学姐一直对我耐心协调和答疑，组织实验课程。让我们本学期的课程学习十分轻松愉快。

最后想要感谢我们自己，在最后的智慧海洋牧场可视化监测系统的开发中，我们团队协作同心，动用了上课中所学到的各个开发方法，解决了一个又一个的困难，作为本科阶段最后一门大作业，我们对自己的成功十分满意，是一份无愧于心的交待。

再次感谢老师的耐心指导，收获颇丰！希望老师学姐们未来生活工作一切顺利。

学姐人美心善！毕业快乐呀！