党未福2022年终总结

概述

2022年的工作即将告一段落，回顾这一年来的工作，我在公司领导及各位同事的支持与帮助下，严格要求自己，按照公司的要求，努力提高业务、工作水平，较好地完成了本职工作。现将工作情况总结如下：

工作内容（如期上线）

1. 涉及到的项目（8个）
2. 功能点（18个）

* 体检计划功能（身高体重测量、营养状态评价、报表统计）
* 每日三检功能（晨检测量、检查项配置、报表统计）
* 访客模块（访客单动态配置）
* 巡检模块（功能优化、任务单异常检查）
* 班级圈模块（业务bug修复）
* 亲子活动模块（功能优化，支持录音资源）
* 食谱模块（食谱优化，增加图片式食谱）
* 教学计划模块（教学计划优化，增加图片式教学计划）
* 宝宝监控功能（幼儿看宝宝开通、关闭；园所需求维护）
* 文件服务（对接阿里鉴黄鉴恐）
* 用户必要项管理（扫描园所、教师、学生、家长相关信息并向用户提醒系统必须的一些数据项）
* 体温统计（幼儿体温统计）
* 数据大屏报表统计（监管机构及其下级园所、教职工、学生相关报表统计）
* 客户端运营（支持用户自定义首页常用应用、应用管理（全部、常用、推荐应用））
* 客户端运营（广告位支持自定义）
* 客户端运营（客户端协议支持动态发布）
* 客户端运营（客户端版本支持动态管理）
* 客户端运营（运营后台支持IP白名单）
* bug修复处理
* 线上问题排查修复

已具备的工作能力

* 业务功能开发（95%）；
* 线上问题排查与修复（80%）；
* 项目瓶颈优化（50%）；
* 独立项目的功能实现(分析、方案、开发) （85%）

自我评估工作能力的依据：

技术方面：

熟知jdk常用API

熟知第三方工具类库

熟知中间件：Redis、RocketMQ、MongoDB、OSS、XXL-JOB、MySQL、Nacos

熟知的框架：SpringBoot、Mybatis、Mybatis-plus

项目管理方面：

熟知项目开发流程：调研、评审、开发、审查、测验、发布

项目考虑点：

1. 明确用户的核心需求及产研结果价值
2. 明确已拥有的可供调配的资源(人力、物力、财力、第三方资源配置)
3. 投入产出比，越小越好
4. 风控：
   1. 内外因素导致产研结果未达预期的风险
   2. 各种因素至资源调配不足使投入成本过大的风险
   3. 项目可行性调研不足至目标未能达到的风险

尚未具备或欠缺的能力

信息收集、汇总、分析、表述能力欠佳。满分10分，自我评价6.5分。

部分项目瓶颈优化。例如：

1. mongo日志查询过慢
2. MySQL主从延迟导致查询过慢
3. 考勤CPU、磁盘占用过高

缺陷不足的原因

1. 信息收集、汇总、分析、表述相关不足的原因：
   * 1. 对信息敏感度不够。收集信息的途径方式不够。
     2. 对信息收集不够充分导致无法完整准确的分析
     3. 脑海中对信息的处理大多数时间都处于非紧张状态，瞬时的应变及时处理能力不够。
     4. 对事物的口头表述常常词不达意，词穷，对事物无法简明扼要的阐述清楚，嘴笨。
2. 项目瓶颈优化能力不足的原因：
   * 1. 对项目环境熟悉程度不够，技术栈理解程度太浅。

改进的措施

1. 信息收集、汇总、分析、表述相关能力改进提升的措施：
   * 1. 多与同事沟通，了解更多信息检索的途径，不过分依赖百度。了解更多信息检索的方式方法，快速提升信息收集的能力。强迫自己提升信息敏感度、接收能力。
     2. 多与同事沟通事务（工作上、生活上）提升自己的口头表述能力
2. 项目瓶颈优化能力方面改进提升的措施：
   * 1. 寻找技术栈详解视频文档相关资料，深入了解熟悉技术栈的工作原理及不同运行环境下可能产生的问题和解决方式。
     2. 寻找并学习CPU、磁盘监控相关的工具，能够充分使用工具快速定位问题原因。
     3. 多看看阿里云文档中心的相关介绍资料，熟悉云服务存在的一些问题及解决方案。

预期的目标

1. 能够独立自主的排查、定位、分析问题，查找相关资料提出解决方案并快速响应解决。
2. 能够独立自主的承接项目，确定需求边界，评估投入成本、项目风险，把控项目进度，能够如期保质保量的完成交付。

经验与教训

1. MySQL主从同步延迟导致数据查询不出来问题
2. 未注册事务而使用Spring事务管理器进而导致程序异常问题
3. 日志输出不完善对排查问题未起到正向引导作用
4. 业务逻辑不够清晰完善，存在代码逻辑漏洞问题
5. 程序执行效率考虑不够，存在性能问题
6. 程序结构设计不够充分，存在大量冗余代码
7. 事务控制的粒度不够细致，导致部分大事务提交过慢问题
8. 锁的粒度控制的不够准确，存在锁竞争过于激烈、无竞争锁增加性能。
9. 无锁的事务操作在并发环境下出现重复数据，导致下游业务报错问题。
10. 循环中操作数据库触发IO报警
11. API运用不够准确，存在过分运用的情况，譬如：Optional判空最好是运用在链式判断中，只有一两个的判断使用它则不合适；线程池的运用不够规范，同一个项目中存在多个线程池，表面上看起来是异步提升接口性能，实际上多个线程池的任务都是在同一台机器上，资源都是固定的，原本一次线程调度就能完成的任务，为了提前响应硬生生拆成多个线程，用户感知是快了实际上增加了服务器的压力。这种操作并不是性能优化，实则是性能压力转移