

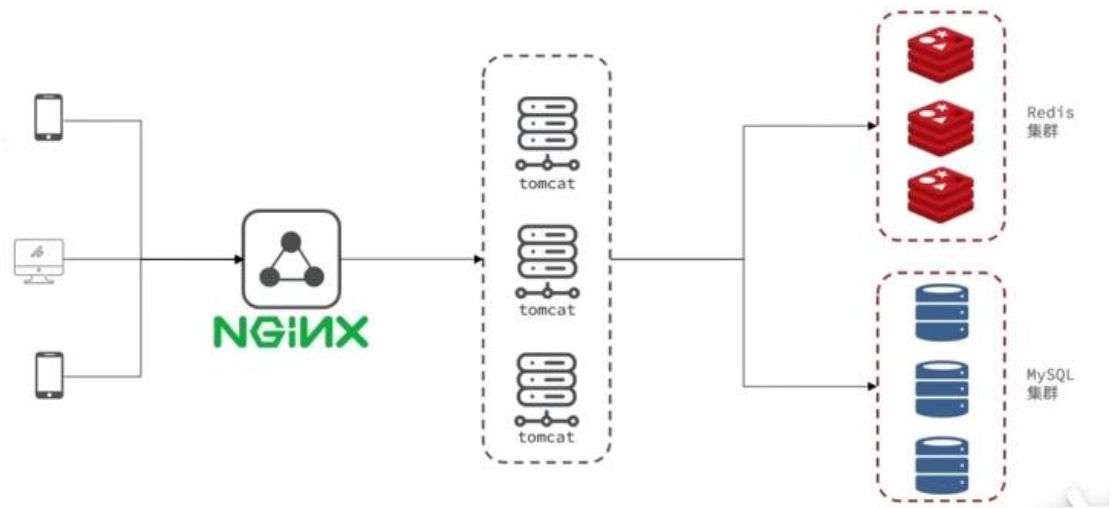
WEEKIE 时间规划 APP 开发计划

目录

- 1. 项目概述 2
 - 1.1 项目流程 2
 - 1.2 技术选型 2
- 2. 工作计划 3
- 3. 人员安排计划 4
- 4. 时间控制计划 5
- 5. 风险管理计划 7

1. 项目概述

1.1 项目流程



流程图如上所示。

用户在手机端或 PC 端浏览网页，访问前端代理服务器（图中为 NGINX）的动态页面。Html 经浏览器渲染后，用户在页面中操作，仍是访问前端代理服务器，代理服务器对请求进行负载均衡，分发给后端服务器集群（图中为 tomcat），调用后端服务器的接口。在后端接口中处理前端请求，修改对应的数据库，然后将结果返回给代理服务器，进而返回给前端页面。

1.2 技术选型

项目所使用到的技术栈如下：

用户层	node.js	VUE.js	ElementUI				
网关层	Nginx						
应用层	Spring Boot	Spring MVC	Spring Task	httpClient	JWT	Swagger	lombok
数据层	Mysql	Redis	Mybatis	Mybatis-plus	pagehelper	Mybatis-plus	

2. 工作计划

前端：

①动态页面的编写。要能正确反映项目所实现的外部接口并保证用户方便使用。

②接口文档的编写。根据动态页面中调用的后端接口，整理一份接口文档。对于每个接口应明确规定请求路径、请求参数、返回数据、接口功能。

后端：

后端接口的编写。应严格按照接口文档中的说明实现每个接口。

数据库：

参考接口文档，设计合理的数据库表。

服务器部署：

前端代码部署到前端代理服务器 NGINX 上。

后端代码采用 spring-boot 框架，内嵌 tomcat 后端服务器，最终以 jar 包的方式启动，不需要额外配置后端服务器。

数据库采用 Mysql 和 Redis，需要配置 Mysql 服务器和 Redis 服务器。

在开发阶段，服务器在本地运行。测试及生产阶段，服务器统一部署在云端。

其他：

实现 NLP 功能，完成用户日程的自动规划。

在我们的 weekie 时间规划 APP 中，具体的任务如下：

任务 A：初步实现前端代码中必要的组件；完成接口文档的初步编写，包括一些基础接口，如注册/登录接口、用户日程的增删改查接口。

任务 B：完善前端代码，调整页面布局；完善接口文档。

任务 C：根据接口文档初步设计数据库表，如：用户表，日程表。

任务 D：完善数据库表的设计。

任务 E：开发环境下服务器集群的部署。

任务 F：根据接口文档初步实现后端接口。

任务 G：完善后端接口。

任务 H：NLP 功能的实现。

任务 I：本地测试

任务 J：项目云部署，测试

3. 人员安排计划

田子昂、陈宇楠：完成前端页面的代码编写。

易森权：协助前端页面的编写，并整理前端页面中对后端接口的请求，汇总成一份接口文档。

唐鸢羽：设计数据库表；实现 NLP 功能。

龚厚旺：完成后端代码的编写；部署服务器集群。

对方小组 TNT 课评宝成员：协助对产品进行测试验收，监督产品的质量

角色-人员责任矩阵表如下：

角色	人员
前端开发小组	田子昂，陈宇楠，易森权
后端开发小组	唐鸢羽，龚厚旺
软件测试小组	对方小组 TNT 课评宝成员，易森权，龚厚旺
需求分析小组	易森权，陈宇楠，龚厚旺
软件设计小组	陈宇楠，唐鸢羽
软件项目负责人	田子昂
用户方负责人	对方小组 TNT 课评宝成员
前端开发负责人	陈宇楠
后端开发负责人	龚厚旺
质量保证小组	易森权

软件开发活动-角色责任矩阵表：

软件开发活动	执行	负责	评审	批准
任务 A	前端开发小组	前端开发负责人	需求分析小组	软件项目负责人
任务 B	前端开发小组	前端开发负责人	需求分析小组	软件项目负责人
任务 C	后端开发小组	后端开发负责人	需求分析小组	软件项目负责人
任务 D	后端开发小组	后端开发负责人	需求分析小组	软件项目负责人
任务 E	后端开发小组	后端开发负责人	需求分析小组	软件项目负责人
任务 F	后端开发小组	后端开发负责人	需求分析小组	软件项目负责人
任务 G	后端开发小组	后端开发负责人	需求分析小组	软件项目负责人
任务 H	后端开发小组	后端开发负责人	需求分析小组	软件项目负责人
任务 I	用户方负责人	软件项目负责人	质量保证小组	软件项目负责人
任务 J	后端开发负责人	软件项目负责人	质量保证小组	软件项目负责人

4.时间控制计划

在工作计划中，我们有任务 A 至任务 F 一共 10 个任务，在这里为了并行开展任务，提高效率，我们将任务细化，同时合理安排时间，得到的任务网络图如下：

5. 风险管理计划

风险信息表如下：

风险名称	开发技术选型不当
风险类型	技术风险
影响值	高
风险因素	缺乏对新技术的了解和掌握，包括前沿的前端，后端开发经验
发生概率	中
影响	延误项目进度，导致功能不稳定，增加后续维护成本
监控方法	定期进行技术评估和学习，与专家进行沟通，及时调整技术选型
应急计划	咨询专家，同时投入更多时间

风险名称	团队成员流失
风险类型	人力资源风险
影响值	中
风险因素	学业压力、个人因素
发生概率	低
影响	影响项目进度，可能导致任务分配不均，沟通效率下降
监控方法	定期进行团队成员状态评估，建立良好的团队氛围和沟通机制
应急计划	额外时间用于团队建设

风险名称	需求变更频繁
风险类型	需求管理风险

影响值	高
风险因素	用户需求不明确、变动频繁
发生概率	高
影响	影响项目进度，导致频繁调整开发计划，增加开发成本
监控方法	建立灵活的需求管理机制，与用户保持密切沟通，明确需求变更影响
应急计划	额外时间用于需求调整

风险名称	安全漏洞
风险类型	安全风险
影响值	高
风险因素	数据泄露、系统被攻击
发生概率	中
影响	可能导致用户信息泄露、系统瘫痪，严重影响用户信任和软件可靠性
监控方法	定期进行安全漏洞扫描和修复，加强团队安全意识培训
应急计划	额外的安全扫描工具和安全专家咨询费用

风险名称	兼容性问题
风险类型	技术风险
影响值	中
风险因素	不同平台、设备、浏览器兼容性
发生概率	中
影响	可能导致部分用户无法正常使用或体验差，影响用户满意度
监控方法	定期进行跨平台、跨设备、跨浏览器测试，及时修复兼容性问题
应急计划	额外的测试设备和人力成本

风险排序表如下：

风险名称	风险概率	损失	危险度
需求变更频繁	0.8	8	6.4
安全漏洞	0.6	6	3.6
开发技术选型不当	0.7	5	3.5
兼容性问题	0.5	3	1.5
团队成员流失	0.2	2	0.4