PageEye页面监视工具

软件项目计划

版本 0.1

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2024.6.25 | 0.1 | 首次创建软件项目计划 | 张倍宜 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 定义、首字母缩写词和缩略语 4

2. 项目概述 4

2.1 项目的目的、规模和目标 4

2.2 假设与约束 4

2.3 项目的可交付成果 4

3. 项目组织 4

4. 项目计划 4

4.1 风险分析 4

4.2 方法和工具 5

4.3 开发计划 5

4.4 质量保证计划 5

4.5 项目沟通计划 5

软件项目计划

# 简介

## 目的

明确制定“PageEye页面监视工具”的开发计划，包括迭代计划、任务分解、风险管理等；确保项目按时、按质高效完成，满足用户的功能和性能需求；为项目团队提供明确的工作指引，并为项目管理提供依据。

## 范围

该计划仅针对“PageEye页面监视工具”的开发，不涉及其他项目。

## 定义、首字母缩写词和缩略语

元素(Element)：指用户选择监视的目标，可以是网站中的HTML元素，如文字、图片、按钮等。

Watcher：指系统中的监视进程，用户可以创建多个Watcher来实现不同的监视目标。Watcher会持续监测指定元素的状态，并记录监测结果。用户可以查看Watcher的运行记录和状态。

# 项目概述

## 项目的目的、规模和目标

本项目旨在开发一个网页监视工具，帮助用户监控指定网页上的元素变化，并通过邮件、短信等方式及时通知用户。该工具主要包括以下功能：

1. 用户登录注册
2. 创建Watcher
3. 查看当前所有 Watcher 的运行状态
4. 接收监视到的页面元素变化并通知用户

通过这个工具，用户可以方便地监控自己关注的网页内容，追踪存档网页变化，及时获取感兴趣的更新信息，提高工作效率和生活质量。

本项目的主要目标如下：

1. 设计并开发一个简单易用的网页监视工具
2. 实现用户登录注册、创建 Watcher、查看状态等核心功能
3. 提供邮件、短信等多种通知方式，确保用户能及时获得监视结果
4. 确保系统的安全性和可靠性，满足用户的隐私和安全需求
5. 优化系统性能，提高用户体验

总之，该项目旨在为用户提供一个方便高效的网页监视解决方案，帮助他们更好地管理和获取感兴趣的信息。

## 假设与约束

本项目计划基于以下假设：

1. 用户需求和功能需求已明确定义，不会发生重大变更。
2. 开发团队拥有足够的 Web 开发和数据分析技能。
3. 项目相关的 API 和第三方服务(如邮件、短信)能够正常提供支持。
4. 开发过程中不会出现重大的技术障碍。
5. 预算足够购买满足性能需求的云服务器。

本项目受到以下约束：

1. 预算约束：项目总预算为900元
2. 人员约束：项目团队由5名开发人员组成，团队成员的平均技术经验为2年
3. 设备约束：项目开发人员使用个人电脑开发，Web应用程序运行在华为云服务器上
4. 时间约束：整个项目的预计总工期为9周，每个迭代周期为3周
5. 质量约束：系统整体可用性目标不低于99%

## 项目的可交付成果

|  |  |
| --- | --- |
| **类型** | **内容** |
| 文档 | 《软件项目计划》 |
| 《迭代计划》 |
| 《迭代评估报告》 |
| 《SRS文档》和用例模型 |
| 《软件架构文档》和基于UML的分析设计模型 |
| 《系统测试用例》和《系统测试报告》 |
| 《项目总结报告》 |
| 验收答辩PPT |
| 代码 | 源代码 |
| 单元测试用例和单元测试覆盖报告 |
| 软件 | Web应用程序 |

# 项目组织

|  |  |  |
| --- | --- | --- |
| **姓名** | **角色** | **分工** |
| 张倍宜 | 组长 | 搭建项目框架 |
| 陈瑞涵 | 组员 | 搭建数据库 |
| 顾一帆 | 搭建测试网站 |
| Jason Young | 实现核心算法 |
| Juliano | 搭建邮箱、短信通知服务 |

# 项目计划

## 风险分析

|  |  |  |  |
| --- | --- | --- | --- |
| **优先级** | **风险名称** | **风险描述** | **风险缓解措施** |
| 1 | 技术风险 | 由于开发团队缺乏相关技术经验，或所使用的技术存在局限性，可能会在开发过程中遇到技术障碍。 | 1. 提前评估技术可行性，选择合适的技术方案 2. 制订合理的技术学习计划 3. 与技术专家保持沟通，寻求技术支持 |
| 2 | 进度风险 | 项目进度可能会因各种原因（如需求变更、资源调配不足等）而延误，影响最终交付时间。 | 1. 制定合理的项目进度计划，设置适当的缓冲时间 2. 密切监控项目进度，及时发现并解决问题 3. 优化任务分配，确保资源合理利用 |
| 3 | 安全隐私风险 | 系统可能存在安全漏洞或用户隐私数据泄露的风险，影响系统的可信度。 | 1. 采用行业标准的安全防护措施 2. 建立完善的安全管理机制和应急预案 3. 加强用户隐私保护，遵守相关法规要求 |
|  |  |  |  |

## 方法和工具

版本管理工具：GitHub

项目管理工具：CodeArts

建模工具：PowerDesigner

IDE：WebStorm，IntelliJ IDEA Ultimate，Visual Studio Code

## 开发计划

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **迭代名称** | **起止时间** | **所缓解的风险** | **需完成的任务** | **需提交的成果** |
| 迭代1 | 6.25-7.9 | 技术风险，进度风险 | 1. 确定技术栈，学习技术 2. 需求分析和开发计划制定 3. 软件架构设计和基础功能开发 | 《迭代计划》、《迭代评估报告》、《SRS文档》和用例模型、《软件架构文档》和基于UML的分析设计模型、项目代码 |
| 迭代2 | 7.10-7.19 | 进度风险，安全隐私风险 | 1. 剩余核心功能开发 2. 云服务器上线部署 3. 系统测试与Bug修复 4. 完善安全与隐私保护机制 | 《迭代计划》、《迭代评估报告》、《系统测试用例》和《系统测试报告》、项目代码 |
| 迭代3 | 7.20-9.13 | 安全隐私风险 | 1. 额外功能开发 2. 系统优化与文档编写 3. 最终系统测试与Bug修复 | 《迭代计划》、《迭代评估报告》、《系统测试用例》和《系统测试报告》、《项目总结报告》、项目代码 |
|  |  |  |  |  |

## 质量保证计划

|  |  |  |
| --- | --- | --- |
| 计划 | 时间 | 人员 |
| 需求评审 | 6.25-6.28 | 全体成员 |
| 设计评审、代码评审 | 每次迭代完成后 | 全体成员 |
| 单元测试 | 模块开发完成后 | 模块开发者 |
| 集成测试 | 每次迭代完成后 | 全体成员 |
| 系统测试 | 迭代2、3完成后 | 全体成员 |

## 项目沟通计划

每日立会：周一到周五上午9：30