<戒烟辅助系统>

软件需求规约

版本 <1.1>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <4/11/2023> | <1.0> | <第一版软件需求规约> | <丁灿、张浩宇> |
| <10/11/2023> | <1.1> | <第二版软件需求规约> | <丁灿、张浩宇> |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 定义、首字母缩写词和缩略语 4

1.3 参考资料 4

2. 整体说明 4

3. 功能需求 4

3.1 <Use case 图> 5

3.2 <Use case1 规约> 5

3.3 <Use case2 规约> 5

4. 非功能需求 5

4.1 易用性 5

4.2 可靠性 5

4.3 性能 5

4.4 可支持性 6

4.5 设计约束 6

5. 其它产品需求 6

5.1 联机用户文档和联机帮助的需求 6

5.2 接口需求 6

5.2.1 用户界面 6

5.2.2 硬件接口 6

5.2.3 软件接口 6

5.2.4 通信接口 7

5.3 适用的标准 7

软件需求规约 (简化版)

# 简介

## 目的

SRS是项目成功的关键文档之一，它为项目的各个阶段提供了方向和依据，确保项目交付符合质量标准和干系人的期望。SRS明确了应用系统或子系统的外部行为，以下是SRS的主要目的：

1. 明确需求。包括功能性需求和非功能性需求，确保开发团队和利益相关者对系统的预期有清晰的了解。
2. 合同基础。SRS通常被视为合同的一部分，约定了开发团队需要交付的内容，确保各方达成一致。
3. 项目管理。包括制定项目计划，分配项目内容，跟踪进度和成本等，有助于确保项目按时交付并在预算内完成。
4. 维护更新。SRS有助于系统将来的维护和更新工作，以便在必要时进行修改和改进。

## 定义、首字母缩写词和缩略语

## 参考资料

# 整体说明

戒烟辅助系统的总体效果是帮助用户成功戒烟，改善其健康和生活质量。产品应具有有效的戒烟支持功能，鼓励用户坚持戒烟计划，提供戒烟信息和资源，以减少吸烟次数和烟草依赖。

• 产品功能

产品需要提供一系列功能，具体功能如下：

1. 戒烟支持和辅导：系统应提供戒烟支持，为用户制定个性化的戒烟计划，提供戒烟技巧和资源，鼓励用户坚持戒烟目标。它可以提供实时反馈和监控，以帮助用户追踪他们的进度。
2. 教育和信息：系统应该为用户提供戒烟相关的教育和信息，帮助他们了解吸烟对健康的影响，提供戒烟策略和方法。这有助于用户做出明智的戒烟决策。
3. 用户社交互动：系统可以提供用户社交互动的机会，例如参加支持小组、在线社区或与其他用户互动。这有助于用户分享经验、提供支持和建立互相激励的社交网络。
4. 数据分析和报告：系统可以记录用户上传的戒烟数据，生成报告和统计信息，以帮助用户了解他们的进度和改进机会。这有助于用户量化他们的戒烟成就。
5. 可持续性和长期效果：系统应该追求长期的戒烟效果，而不仅仅是短期成就。它应该为用户提供支持，以帮助他们在戒烟过程中克服挑战，预防复吸。

• 用户特征

用户特征包括吸烟数据、戒烟动机、戒烟意愿、吸烟习惯、数字素养、社交互动意愿、目标设定、地理位置、年龄、性别、健康状况、语言偏好等。产品应考虑不同用户的特征，提供个性化的支持和建议。

• 约束

约束是产品开发和实施的限制因素。包括预算、时间、技术能力、合规性要求等，影响产品的开发和交付计划。以下是一些可能的约束：

1. 法律法规：系统需要遵循与医疗信息、隐私和数据安全相关的法律法规，这可能设计用户数据的合法处理、隐私政策的明确说明、数据存储和传输加密等。
2. 隐私约束：系统需要确保用户上传的戒烟数据得到妥善保护，以防止未经授权的访问和数据泄露。隐私政策和数据安全措施应该得到充分的考虑。
3. 安全约束：系统需要具有适当的安全措施，以防范潜在的网络攻击、数据泄漏和恶意行为。这包括网络安全、身份验证控制、数据备份等。
4. 资源约束：项目可能受到预算、时间和人力资源的限制。这需要在项目计划中予以考虑，以确保项目能够按时交付。
5. 技术约束：系统的设计和实施可能受到技术约束的限制，包括可用技术栈、开发工具、硬件设备等。这需要在项目计划中进行评估。

• 假设与依赖关系

假设是关于产品运行环境和用户行为的猜测。依赖关系表示产品可能依赖其他系统、数据或服务。

假设：

1. 用户数据准确性：系统假设用户上传的戒烟数据是准确的，包括吸烟频率、吸烟数量等。如果用户提供不准确的数据，系统的建议和监控可能会受到影响。
2. 用户自愿性：系统假设用户是自愿使用该系统的，并且有戒烟的意愿。如果用户不愿意坚持戒烟计划，系统的效果可能会减弱。
3. 可访问性：系统假设用户有互联网访问和设备，以便上传戒烟数据。对于没有互联网访问的用户，系统的可用性可能受到限制。

依赖关系：

1. 数据库：系统依赖数据库来存储和管理用户的戒烟数据。数据库需要提供数据的持久性和安全性
2. 互联网连接：用户上传数据的过程依赖于互联网连接。如果用户没有稳定的互联网连接，数据上传可能会受到影响。
3. 数据分析工具：如果系统提供数据分析功能，它将依赖适当的数据分析工具和算法来生成有用的戒烟建议和报告。

• 需求子集

1. 用户注册和登录需求：

* 用户能够通过提供相关信息并创建账户。
* 用户能够使用用户名和密码登录
* 用户可能忘记密码，允许用户重置密码。

1. 个性化需求：

* 用户能够设计个人戒烟目标，吸烟习惯，戒烟计划等信息。
* 用户能随时编辑或查看这些信息。

1. 信息和资源需求：

* 系统应提供戒烟信息和资源，如健康效益、戒烟技巧、烟草替代品等。
* 用户应能够访问文章、视频和其他教育性材料。

1. 数据分析和报告需求：

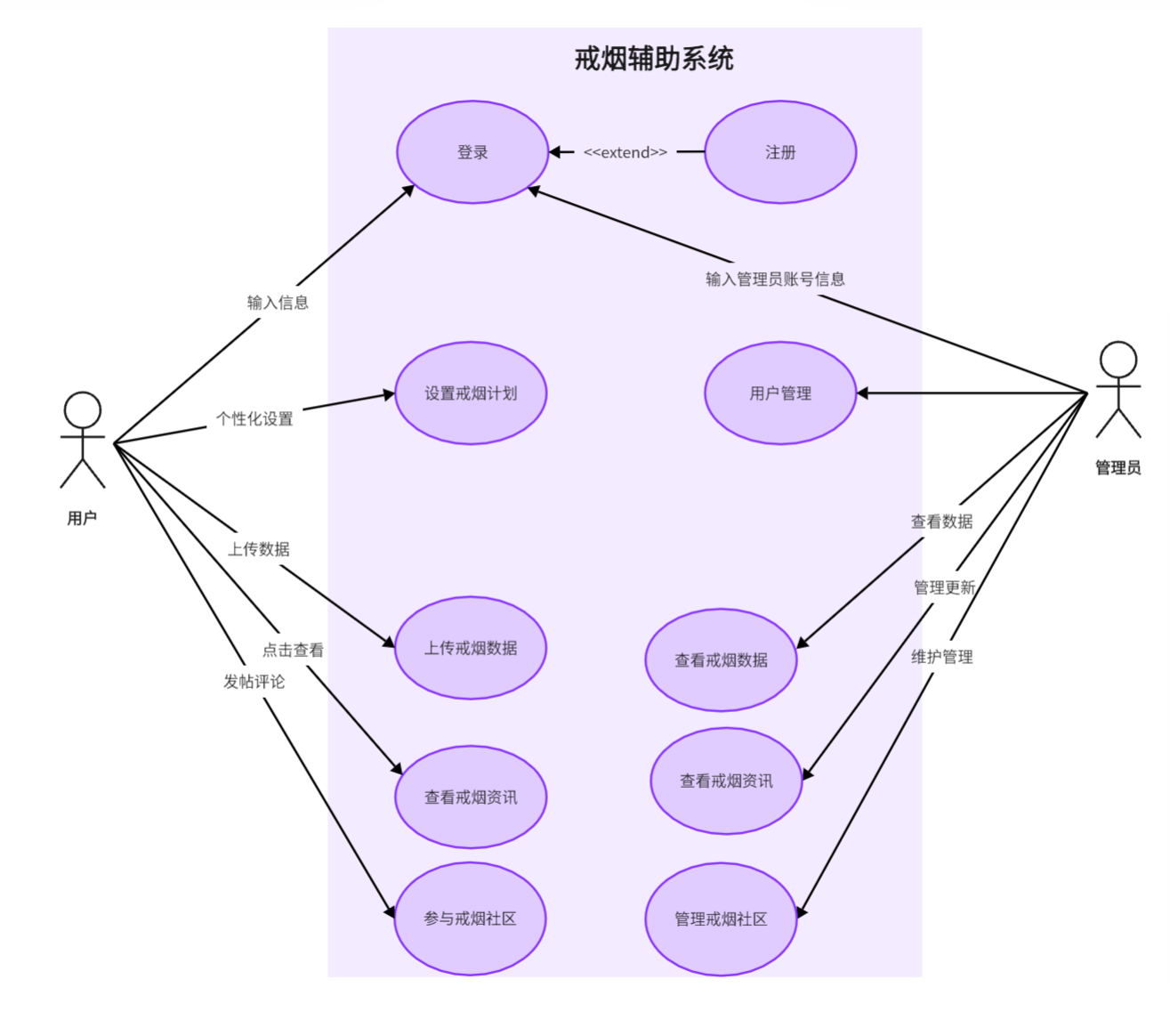
* 系统应收集和分析用户的戒烟数据，生成报告和统计信息。
* 提供用户可以理解的可视化报告并可随时查看。

1. 安全性需求：

* 用户数据应受到适当的保护，包括密码加密、数据传输的安全性等。
* 防止未经授权的访问和数据泄漏。

# 功能需求

## <Use case 图>



**主要参与者：**

* 用户：使用系统来上传戒烟数据、设置戒烟计划和查看进度。
* 管理员：负责管理和维护系统，包括用户帐户和数据管理。

**Use Case 列表：**

* 用户注册：用户可以注册一个新帐户。
* 用户登录：已注册的用户可以登录到系统。
* 上传戒烟数据：用户可以上传自己的戒烟数据，包括戒烟日期、吸烟频率等信息。
* 设置戒烟计划：用户可以设置个性化的戒烟计划，包括目标戒烟日期和方法。
* 查看戒烟进度：用户可以查看他们的戒烟进度、统计数据和图表。
* 查看戒烟资讯：用户可以从网站获取戒烟知识和相关资讯。
* 参与社区：用户可以参与社区、进行讨论。
* 用户管理：管理员可以管理用户帐户，包括创建、禁用或删除帐户。
* 数据管理：管理员可以管理用户上传的戒烟数据，以确保数据的准确性和完整性。
* 社区管理：管理员可以管理用户参与的社区，包括禁言、删帖等操作。
* 资讯管理：管理员可以更新并管理戒烟资讯的信息。

## <用户注册 规约>

|  |  |
| --- | --- |
| 项目 | 内容描述 |
| 用例名称 | 用户注册 |
| 用例编号 | UC01 |
| 参与者 | 用户 |
| 用例简述 | 用户可以注册一个新帐户，以便使用系统的功能。 |
| 事件流 | 基本事件流：   * 用户选择“注册”。 * 系统显示注册表单。 * 用户填写所需信息，如用户名、密码、电子邮箱等。 * 用户提交表单。 * 系统验证信息的有效性。 * 系统创建新用户账户。 * 系统显示注册成功消息。   备选事件流：   * 如果信息验证失败（如用户名已存在或密码不符合要求），系统显示错误消息并要求用户重新输入。 |
| 前置条件 | 用户访问注册页面。 |
| 后置条件 | 用户成功注册并可以登录到系统。 |

## 3.3<上传戒烟数据 规约>

|  |  |
| --- | --- |
| 项目 | 内容描述 |
| 用例名称 | 上传戒烟数据 |
| 用例编号 | UC03 |
| 参与者 | 用户 |
| 用例简述 | 用户可以上传自己的戒烟数据，包括戒烟日期、吸烟频率等信息。 |
| 事件流 | 基本事件流：   * 用户选择“上传戒烟数据”。 * 系统显示数据上传界面。 * 用户输入戒烟数据，如戒烟日期、吸烟频率等。 * 用户提交数据。 * 系统保存数据。 * 系统显示上传成功消息。   备选事件流：   * 如果用户未输入必要的数据或数据格式不正确，系统显示错误消息并要求用户重新输入。 |
| 前置条件 | 用户成功登录到系统。 |
| 后置条件 | 用户的戒烟数据已上传并保存在系统中。 |

## 3.4 <设置戒烟计划 规约>

|  |  |
| --- | --- |
| 项目 | 内容描述 |
| 用例名称 | 设置戒烟计划 |
| 用例编号 | UC04 |
| 参与者 | 用户 |
| 用例简述 | 用户可以设置个性化的戒烟计划，包括目标戒烟日期和方法。 |
| 事件流 | 基本事件流：   * 用户选择“设置戒烟计划”。 * 系统显示戒烟计划设置界面。 * 用户设定戒烟目标（如戒烟日期）和方法。 * 用户提交戒烟计划。 * 系统保存戒烟计划。 * 系统显示设置成功消息。   备选事件流：   * 如果用户未完成所有必要步骤或输入格式不正确，系统显示错误消息并要求用户重新输入。 |
| 前置条件 | 用户成功登录到系统。 |
| 后置条件 | 用户的戒烟计划已设置并保存在系统中。 |

## 3.5 <用户管理 规约>

|  |  |
| --- | --- |
| 项目 | 内容描述 |
| 用例名称 | 用户管理 |
| 用例编号 | UC08 |
| 参与者 | 管理员 |
| 用例简述 | 管理员可以管理用户帐户，包括创建、禁用或删除帐户。 |
| 事件流 | 基本事件流：   * 管理员选择“用户管理”。 * 系统显示用户列表。 * 管理员选择一个或多个用户进行管理操作（创建、编辑、禁用或删除）。 * 系统执行操作并更新用户状态。 * 系统显示操作成功消息。   备选事件流：   * 如果操作失败（如试图删除不存在的用户），系统显示错误消息。 |
| 前置条件 | 管理员成功登录到系统。 |
| 后置条件 | 用户帐户被管理，相应操作完成。 |

# 非功能需求

## 易用性

1. 培训时间要求：

* 普通用户应该能够在系统中执行常见任务，如获取戒烟知识和查看实时反馈，而无需长时间的培训。培训时间应限制在30分钟内。
* 高级用户或有经验的用户（如已经使用了戒烟系统的用户）应该能够快速上手，不需要培训，或者只需非常有限的培训时间。

1. 可评测任务次数需求：

* 系统应该定期进行用户界面可用性测试，以确定常见任务的可评测任务次数。这有助于发现和修复潜在的用户界面问题。
* 用户应该能够在系统中执行重复性任务，如上传戒烟数据、查看戒烟进度等，而无需多次重复相同的步骤。

1. 符合易用性标准需求：

* 系统应该符合公认的易用性标准，如IBM的通用用户界面体系结构（CUA）标准和Microsoft的图形用户界面（GUI）标准。
* 用户界面应具有一致性，符合用户的期望，采用常见的界面元素和交互模式。

1. 导航和信息可访问性需求：

* 用户应该能够轻松找到和访问系统的各个部分，包括设置、信息资源、实时反馈和通信功能。
* 系统应提供清晰的导航菜单和标签，以便用户可以快速找到所需信息和功能。

## 可靠性

1. 可用性：系统应该具有高可用性，可用时间百分比应不低于99%。系统维护窗口应在最低流量时间段内进行，以减少对用户的影响。在维护窗口期间，应提供维护访问权，以便维护人员能够进行必要的维护工作。
2. 平均故障间隔时间：系统的平均故障间隔时间（系统不可用的平均时间间隔）应该超过6个月。
3. 平均修复时间：当系统发生故障时，平均修复时间（从故障发生到系统恢复正常运行的时间）应该尽可能短，以减少中断时间。系统在发生故障后应能够在3小时内重新运行或提供降级模式操作。
4. 最高错误或缺陷率需求：系统开发过程中应设置最高的错误或缺陷率标准，每千行代码的错误数目应低于3（bugs/KLOC）。严重错误应该被明确定义，如数据丢失或系统无法提供某些重要功能。

## 性能

1. 对事物的响应时间需求：

* 平均事务响应时间：系统应该在平均情况下以不超过1秒的响应时间来处理用户请求。
* 最长事务响应时间：最长的单个事务处理时间不应超过10秒。

1. 吞吐量需求：

系统应该能够每秒处理不少于5000个事务，以确保用户在高负载时仍能够获得及时的响应。吞吐量应根据用户访问模式来定义。

1. 容量需求：

系统应该能够容纳不少于500个同时在线用户，应该能够处理不少于10000个戒烟数据上传事务。

## 可支持性

1. 编码标准和质量需求：

* 所有系统代码必须遵循明确定义的编码标准，以确保代码的一致性、可读性和可维护性。
* 代码应定期进行静态代码分析，以检测潜在的问题和错误。

1. 命名约定需求：

* 所有命名应该具有描述性，清晰地反映其用途和功能。
* 命名约定应遵循一致性，以便开发人员容易理解和维护代码。

1. 类库和框架要求：

* 系统应使用适当的类库和框架，以减少开发工作和提高可维护性。
* 所选的类库和框架应得到维护和更新，以确保它们与最新的技术和标准保持一致。

1. 维护访问权要求：

* 维护人员应具有足够的维护访问权，以能够执行必要的维护操作，如修复错误、升级系统、备份数据等。
* 访问权应受到适当的权限和安全控制的约束。

1. 维护实用程序需求：

* 系统应提供维护实用程序，用于监控系统性能、日志记录、错误报告和数据备份。
* 维护人员应能够轻松访问和使用这些实用程序，以支持系统的运行和维护。

1. 文档需求：

* 系统应提供详细的技术文档，包括系统架构、数据库结构、API文档、用户手册和维护指南。
* 文档应保持更新，以反映系统的当前状态和功能。

1. 版本控制需求：

* 所有系统代码和配置应使用版本控制系统进行管理，以跟踪变更并确保版本的一致性。
* 版本控制应包括版本历史记录和版本标签，以便回滚和恢复操作。

## 设计约束

1. 系统的开发必须使用特定的编程语言和技术栈，基于web的戒烟辅助系统使用JavaScript作为编程语言、前端框架采用Vue.js，后端采用Node.js以便实现网站的各种功能和服务。
2. 数据库技术的制定：数据存储必须使用特定的数据库技术，基于web的戒烟辅助系统使用MySQL、Redis作为数据库。
3. 开发工具：开发团队必须使用特定的集成开发环境（IDE）和版本控制工具，基于web的戒烟辅助系统使用Visual Studio Code提高开发效率和代码质量、使用Git作为版本控制工具。
4. 构架及设计约束：系统的架构必须遵循特定的模式或架构，基于web的戒烟辅助系统可以使用基于web的客户端-服务器架构，确保用户可以通过web浏览器访问系统。