# Markdown编辑器需求说明书

## 1. 引言

### 1.1 目的

本需求说明书旨在明确Markdown编辑器的基本功能、用户界面设计、性能要求以及非功能性需求，为开发团队提供明确的开发方向和目标。

### 1.2 目标用户

* **普通用户**：需要快速记录笔记、撰写文档的用户。
* **开发者**：需要编写技术文档、博客文章的开发者。
* **内容创作者**：需要撰写长篇文章、书籍的作者。

### 1.3 项目范围

本项目将开发一个跨平台的Markdown编辑器，使用Qt框架进行开发，支持Windows、macOS和Linux操作系统。编辑器将提供基本的Markdown语法支持，并具备实时预览、导出功能等。

## 2. 功能需求

### 2.1 基本编辑功能

* **文本输入**：支持标准的键盘输入，包括字母、数字、符号等。
* **Markdown语法支持**：支持常见的Markdown语法，如标题、列表、链接、图片、代码块等。
* **撤销/重做**：支持撤销和重做操作，方便用户修改内容。
* **查找/替换**：支持文本的查找和替换功能。

### 2.2 实时预览

* **实时预览**：在编辑器右侧或下方显示Markdown文本的实时渲染效果。
* **同步滚动**：编辑区和预览区同步滚动，方便用户查看内容。

### 2.3 文件管理

* **新建文件**：支持创建新的Markdown文件。
* **打开文件**：支持打开本地Markdown文件。
* **保存文件**：支持保存当前编辑的Markdown文件。
* **另存为**：支持将文件另存为其他格式（如HTML、PDF等）。

### 2.4 导出功能

* **导出为HTML**：将Markdown文件导出为HTML格式。
* **导出为PDF**：将Markdown文件导出为PDF格式。
* **导出为Word**：将Markdown文件导出为Word格式。

### 2.5 主题和样式

* **主题切换**：支持切换编辑器的主题（如浅色、深色、自定义主题等）。
* **字体设置**：支持自定义编辑器字体大小和样式。

### 2.6 插件支持

* **插件系统**：支持第三方插件的安装和使用，扩展编辑器功能。
* **内置插件**：提供一些常用的内置插件，如表格生成器、数学公式支持等。

## 3. 用户界面设计

### 3.1 主界面

* **编辑区**：左侧为Markdown文本编辑区，支持语法高亮。
* **预览区**：右侧或下方为实时预览区，显示渲染后的HTML效果。
* **工具栏**：顶部工具栏包含常用的编辑功能按钮（如加粗、斜体、标题、列表等）。

### 3.2 菜单栏

* **文件菜单**：包含新建、打开、保存、另存为等功能。
* **编辑菜单**：包含撤销、重做、查找、替换等功能。
* **视图菜单**：包含切换主题、调整字体大小等功能。
* **插件菜单**：包含插件管理、插件设置等功能。

### 3.3 快捷键

* **常用快捷键**：支持常用的快捷键操作，如Ctrl+S（保存）、Ctrl+Z（撤销）等。
* **自定义快捷键**：支持用户自定义快捷键。

## 4. 性能需求

### 4.1 响应速度

* **启动时间**：编辑器启动时间不超过3秒。
* **编辑响应**：文本输入和编辑操作的响应时间不超过0.5秒。
* **预览渲染**：实时预览的渲染时间不超过0.2秒。

### 4.2 内存占用

* **内存使用**：编辑器在正常使用情况下的内存占用不超过500MB。

### 4.3 稳定性

* **崩溃率**：编辑器在正常使用情况下的崩溃率不超过0.1%。

## 5. 非功能性需求

### 5.1 安全性

* **数据加密**：支持对本地文件进行加密存储，防止数据泄露。
* **权限管理**：支持对插件的权限管理，防止恶意插件的运行。

### 5.2 可扩展性

* **插件扩展**：支持第三方插件的开发和集成，方便用户扩展功能。
* **API接口**：提供API接口，方便开发者进行二次开发。

### 5.3 国际化

* **多语言支持**：支持多种语言界面，方便不同语言用户使用。
* **本地化**：支持根据用户所在地区自动切换语言和日期格式。

## 6. 项目计划

### 6.1 开发阶段

* **需求分析**：2周
* **UI设计**：2周
* **功能开发**：8周
* **测试与优化**：4周

### 6.2 发布计划

* **Alpha版本**：第10周发布，包含基本编辑功能和实时预览。
* **Beta版本**：第14周发布，包含文件管理、导出功能和插件支持。
* **正式版本**：第16周发布，包含所有功能和优化。

## 7. 附录

### 7.1 术语表

* **Markdown**：一种轻量级标记语言，用于格式化文本。
* **实时预览**：在编辑器中实时显示Markdown文本的渲染效果。
* **插件**：扩展编辑器功能的独立模块。

### 7.2 参考资料

* [Markdown官方文档](https://daringfireball.net/projects/markdown/)
* [Qt框架](https://www.qt.io/)

## 8. 使用Qt框架的优点

### 8.1 跨平台支持

* **跨平台开发**：Qt框架支持Windows、macOS和Linux等多个操作系统，使用Qt开发的Markdown编辑器可以在不同平台上保持一致的用户体验。

### 8.2 丰富的UI组件

* **丰富的UI组件**：Qt提供了丰富的UI组件，如文本编辑器、工具栏、菜单栏等，可以快速构建复杂的用户界面。

### 8.3 强大的信号与槽机制

* **信号与槽机制**：Qt的信号与槽机制使得事件处理更加直观和高效，方便实现实时预览、同步滚动等功能。

### 8.4 国际化支持

* **国际化支持**：Qt内置了强大的国际化支持，可以轻松实现多语言界面和本地化功能。

### 8.5 性能优化

* **性能优化**：Qt框架经过多年的优化，性能表现优异，适合开发高性能的桌面应用程序。

### 8.6 社区支持

* **强大的社区支持**：Qt拥有庞大的开发者社区，提供了丰富的文档、教程和第三方库，方便开发者快速解决问题和扩展功能。

以上是使用Qt框架开发的Markdown编辑器需求说明书，希望为开发团队提供清晰的开发方向和目标。