**软件设计文档**

**目 录**

1 开发规划 2

1.1 开发人员 2

1.2 开发计划 2

1.3 开发语言、环境和工具 2

2 总体设计 3

2.1 基本设计描述

2.2 主要界面流程描述 4

**2.2.1** 实时效果预览 界面流程 4

**2.2.2** md文件转pdf文件 界面流程 4

2.4 模块划分 5

3 **软件设计技术**

4 软件选型理由

# 开发规划

## 开发人员

|  |  |
| --- | --- |
| **分 工** | 人 员 |
| 前端 | 欧光文、林锦涛 |
| 语法解析 | 庄嘉鑫 |
| 生成pdf | 刘继汉、欧光文 |
| 功能、文档 | 黄新伟、赖君秋 |
| 界面设计 | 黄新伟 |
| 测试 | 赖君秋 |

## 开发计划

5.12 完成第一次迭代，包括前端，语法解析

5.14 生成pdf

## 开发语言、环境和工具

开发语言

|  |  |
| --- | --- |
| **语言** | **作用** |
| HTML、CSS | 前端 |
| JavaScript | 后端 |

开发环境

|  |  |
| --- | --- |
| **环境** | **人员** |
| Windows系统 | 黄新伟、赖君秋 |
| Ubuntu系统 | 庄嘉鑫、林锦涛、刘继汉 |
| Mac OS系统 | 欧光文 |

开发工具

|  |  |
| --- | --- |
| **工具** | **作用** |
| Axure RP Pro | 设计软件界面 |
| Sublime Text | 编写软件代码 |
| IE浏览器 | 观察软件效果 |

# 总体设计

Markdown是一种可以使用普通文本编辑器编写的标记语言，通过简单的标记语法，它可以使普通文本内容具有一定的格式。这种写作方式最大的特点是用“符号”表示“格式”。例如，使用word等传统方式写作的时候，输入了一个标题，那就需要选中这个标题去格式栏中设置格式，而使用markdown方式写作的时候，只需要在标题前面加“##”这样的井号标记来表示标题的级别。可以明显的感觉到你输入两个“##”的过程完全不会打断你的写作，这要比单独设置格式方便的多。所以markdown写作方式，就是一种在码字的时候，把文章的格式也顺便“码”进去的写作方式。用户可以使用诸如“\*”、“#”等简单的标记符号以最小的输入代价生成极富表现力的md文档。

Markdown具有很多优点：

①写作中添加简单符号即完成排版，所见即所得，达到实时效果预览的效果，使用户能够专注于文字而不是排版。

②格式转换方便，Markdown的文本你可以轻松转换为pdf文档等。

③可以保存成纯文本。

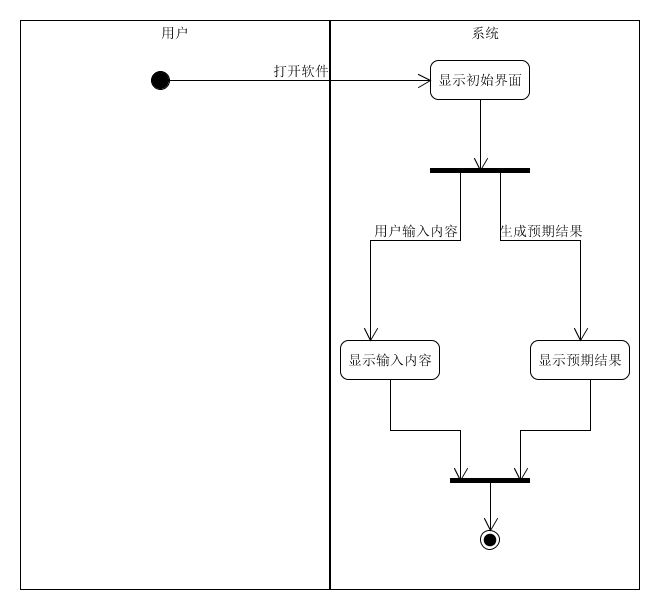
支持Markdown的编辑器太多，功能也不完全一致。本项目团队编写的marker，是使用markdown语言的编辑器，目前将提供两个功能：

①实时效果预览。marker主要界面分为两部分，左半边是编辑区域，用于markdown语言写作；右半边是预览区域，能够实时地对左半边编辑区域中的代码进行编译并实时展示。

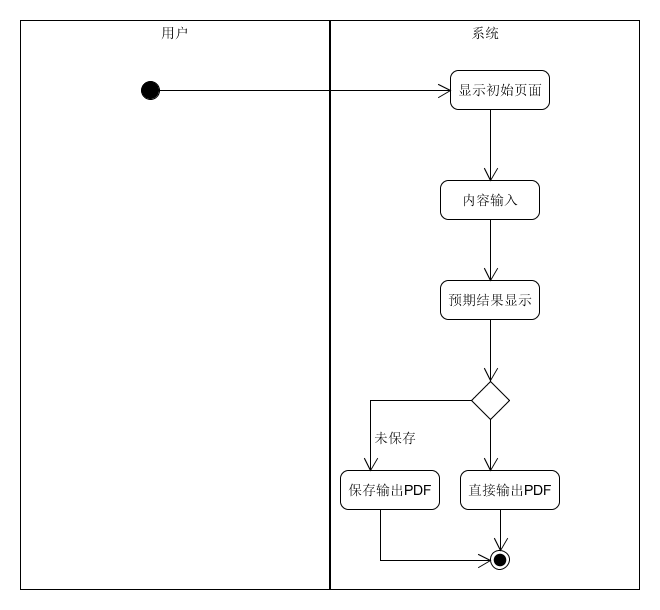
②md文件转pdf文件。将markdown语言文本经过编译之后得到的效果转换为pdf文件并保存到本地。

## 主要界面流程描述

### 实时效果预览 界面流程

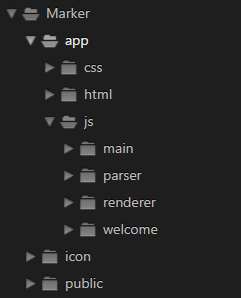


### md文件转pdf文件 界面流程



## 模块划分

程序主要分为CSS，HTML，JS，三个模块，JS又分为main，parser，renderer，welcome，四个部分。首先是main部分，主要是electron的主进程的所有JS代码。Parser部分是负责使markdown编辑器执行语法解析的功能。Renderer是负责处理渲染线程的，与html目录中index.html相关。Welcome部分是关于点击软件后，一开始弹出欢迎页面的功能，与html目录中welcome.html相关。icon文件夹是保存了欢迎页面的图标。Pubilc保存的是页面上需要用的图片。



# 软件设计技术

Design pattern：运用到了观察者模式，主要是通过在主进程监听渲染进程的通信消息。



下面是在渲染进程中监听主进程的通信消息：



# 技术选型理由

**Electron:**

我们选择了Electron是因为Electron 可以让你使用纯 JavaScript 调用丰富的原生 APIs 来创造桌面应用。你可以把它看作是专注于桌面应用而不是 web 服务器的，io.js 的一个变体。

这不意味着 Electron 是绑定了 GUI 库的 JavaScript。相反，Electron 使用 web 页面作为它的 GUI，所以你能把它看作成一个被 JavaScript 控制的，精简版的 Chromium 浏览器。

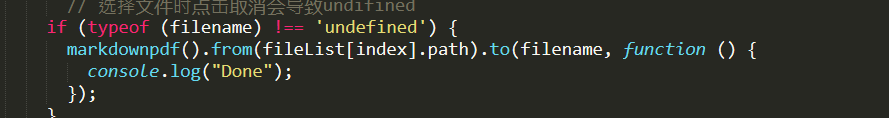
在 Electron 里，运行 package.json 里 main 脚本的进程被称为**主进程**。在主进程运行的脚本可以以创建 web 页面的形式展示 GUI。

由于 Electron 使用 Chromium 来展示页面，所以 Chromium 的多进程结构也被充分利用。每个 Electron 的页面都在运行着自己的进程，这样的进程我们称之为**渲染进程**。

在一般浏览器中，网页通常会在沙盒环境下运行，并且不允许访问原生资源。然而，Electron 用户拥有在网页中调用 io.js 的 APIs 的能力，可以与底层操作系统直接交互。

**markdown-pdf:**

用来将markdown文件转成pdf格式的一个依赖包。



**node-localstorage:**

用来将用户信息存储在Electron中，当用户再次打开时，软件就能将上次用户操作的信息读取出来。



**jquery:**

用来操作html里面的动操作。



**marked:**

用来将markdown文件转成html的一个依赖包。



**eslint:**

在开发时进行JS语法编辑规范。

