

日程管理系统

设计报告

文件状态： [] 草稿 [] 正式发布 [√] 正在修改	文件标识：	
	当前版本：	0.1.0
	作 者：	陈金龙
	完成日期：	2023-04-12

版 本 历 史

版本/状态	作者	审核人	起止日期	备注
v.0.1.0	陈金龙		2022-03-19/ 2022-03-21	

目 录

1	文档介绍	4
1.1	文档编写目的	4
1.2	项目背景	4
1.3	文档范围	4
1.4	参考文献	4
2	设计要点	5
2.1	设计报告范围	5
2.2	关键业务功能	5
2.3	技术实现难点	5
3	功能实现设计	6
3.1	注册功能实现	6
3.1.1	实现简述及约定	6
3.1.2	功能实现流程	6
3.2	登录功能实现	6
3.2.1	实现简述及约定	6
3.2.2	功能实现流程	7
3.3	新加日程功能实现	7
3.3.1	实现简述及约定	7
3.3.2	功能实现流程	7
3.4	编辑日程功能实现	7
3.4.1	实现简述及约定	7
3.4.2	功能实现流程	7
3.5	删除日程功能实现	8
3.5.1	实现简述及约定	8
3.5.2	功能实现流程	8
3.6	统计日程数据功能实现	8
3.6.1	实现简述及约定	8
3.6.2	功能实现流程	8
4	接口设计	9
4.1	外部接口	9
4.2	内部接口	9
5	数据结构设计	10
5.1	对于用户管理表	10
5.2	对于日程管理表	11

1 文档介绍

1.1 文档编写目的

在编写《日程管理系统》软件之前，对现今日程管理系统行了调查。然后基于《需求分析说明书》编写《概要设计报告》，该概要设计报告将介绍系统的基本架构和设计原则，包括所需的技术和软件组件。

1.2 项目背景

本概要设计报告的目的是介绍一个用于管理日程的系统的设计和实现方案。该系统的主要目的是帮助用户更好地管理和安排日常任务和活动。当前，许多人在日常生活中面临着繁重的任务和多样的时间安排，因此需要一个可靠的系统来协助管理日常安排。

该系统旨在提供一种集成的平台，使用户可以方便地创建任务、安排日程、设置提醒，并通过多种方式查看任务和活动的时间表。此外，该系统还将提供一些高级功能，如自动化任务调度和分析任务执行的效率等功能。

希望通过这个日程管理系统，提供一个简单易用的工具，帮助用户更好地管理和安排任务和活动，提高效率和生产力。

1.3 文档范围

本文档包含以下几个部分：

1. 系统功能
2. 设计要点
3. 数据结构

1.4 参考文献

Craig Walls, Spring in Action, Sixth Edition 6th Edition(Manning Publications Co., 2022)

2 设计要点

2.1 设计报告范围

系统报告将涵盖以下范围：

1. 系统设计：确定系统的架构、组件和 workflows，以满足用户需求和期望。
2. 系统开发：使用现代的软件开发工具和技术，实现系统的功能和特性。
3. 数据库设计：该报告不会涵盖系统的数据库设计和实现，如数据库架构、表结构等。
4. 接口设计：确定系统的一些接口设计，介绍内外部接口。

该报告的范围是确定系统设计和开发过程中的关键要素，并概述系统的基本功能和特性。通过本报告，读者将能够了解系统的基本功能和特性，并了解系统设计和开发的关键方面。

2.2 关键业务功能

日程管理系统将提供以下关键业务功能，以帮助用户有效地管理时间和任务：

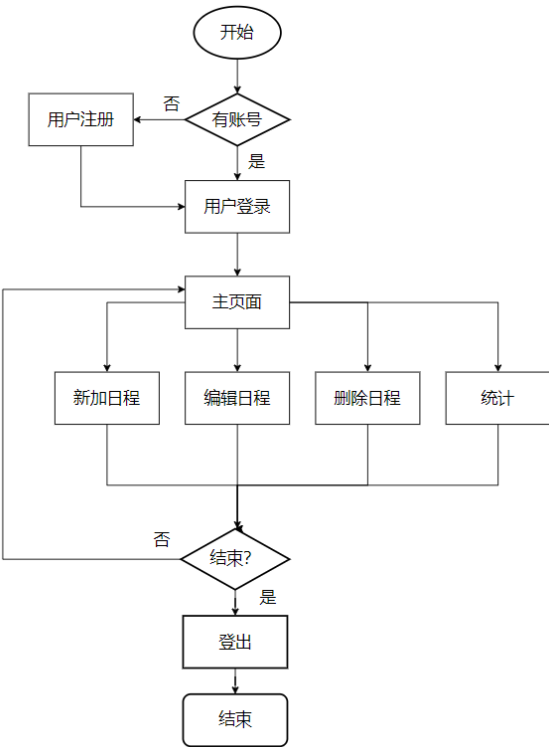
1. 任务管理：允许用户创建、编辑、删除和标记任务，设置任务优先级和状态，并为任务添加截止日期和提醒。
2. 协作功能：允许用户创建共享任务和日历，并邀请其他用户加入协作，以便团队协作工作。
3. 统计和分析：为用户提供任务和时间的统计和分析功能，帮助用户了解时间分配和任务完成情况，以优化时间管理。

2.3 技术实现难点

在日程管理系统的开发过程中，我们将面临以下技术实现难点：

1. 数据库设计和性能优化：由于系统需要存储大量的用户数据，我们需要设计一个高效的数据库结构，并进行性能优化，以确保系统的响应速度和可扩展性。
2. 使用 API 通信，把数据显示在用户界面：视图提供一个方案使得能有效地将从数据库取得的数据，显示在用户界面。

3 功能实现设计

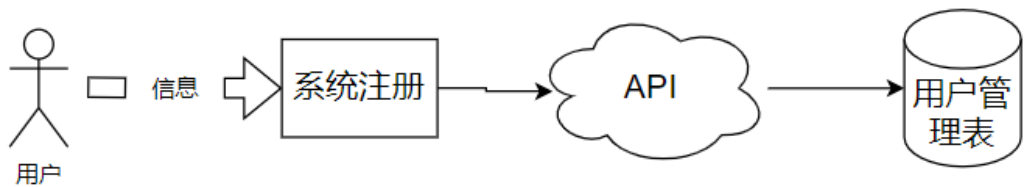


3.1 注册功能实现

3.1.1 实现简述及约定

用户注册是对于所有没注册过的用户的一个强制要求，必须填写个人信息，包括姓名、电话号、邮件和设定一个密码，对于密码数据库会使用 SHA1 来保护用户的密码

3.1.2 功能实现流程



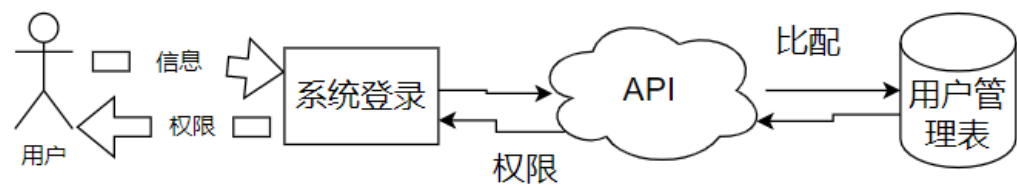
3.2 登录功能实现

3.2.1 实现简述及约定

用户登录，每个用户输入自己的账号及密码，有后端向数据库匹配账号信息和复制对用

户分配权限

3.2.2 功能实现流程

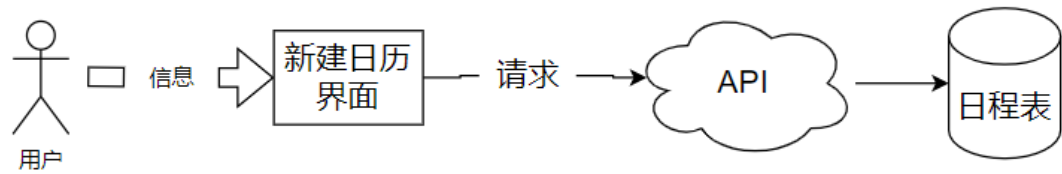


3.3 新加日程功能实现

3.3.1 实现简述及约定

用户向系统提供新日程的信息，系统执行请求，接着将新日程信息存到数据库的日程表

3.3.2 功能实现流程

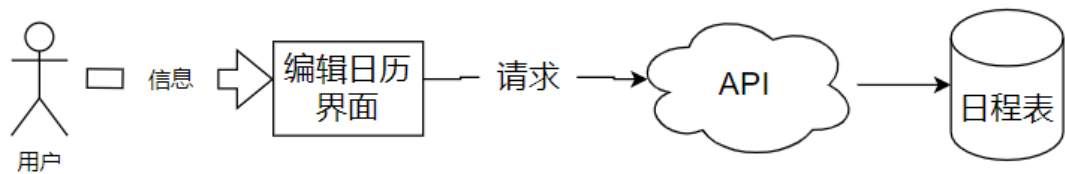


3.4 编辑日程功能实现

3.4.1 实现简述及约定

用户向系统提出编辑请求，系统执行请求，将新的数据送给数据库，更新旧的数据

3.4.2 功能实现流程

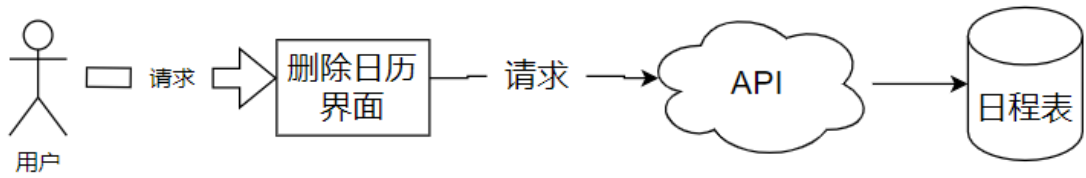


3.5 删除日程功能实现

3.5.1 实现简述及约定

用户对系统提出对某个日程的删除请求，系统向数据库执行删除请求

3.5.2 功能实现流程

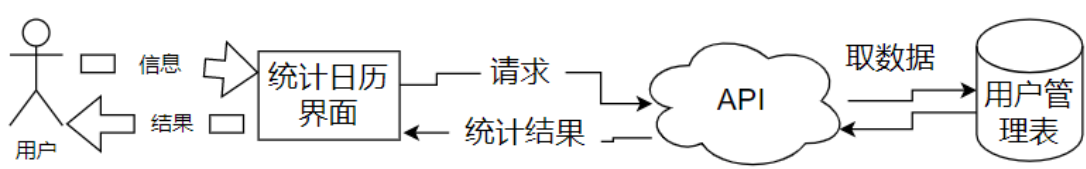


3.6 统计日程数据功能实现

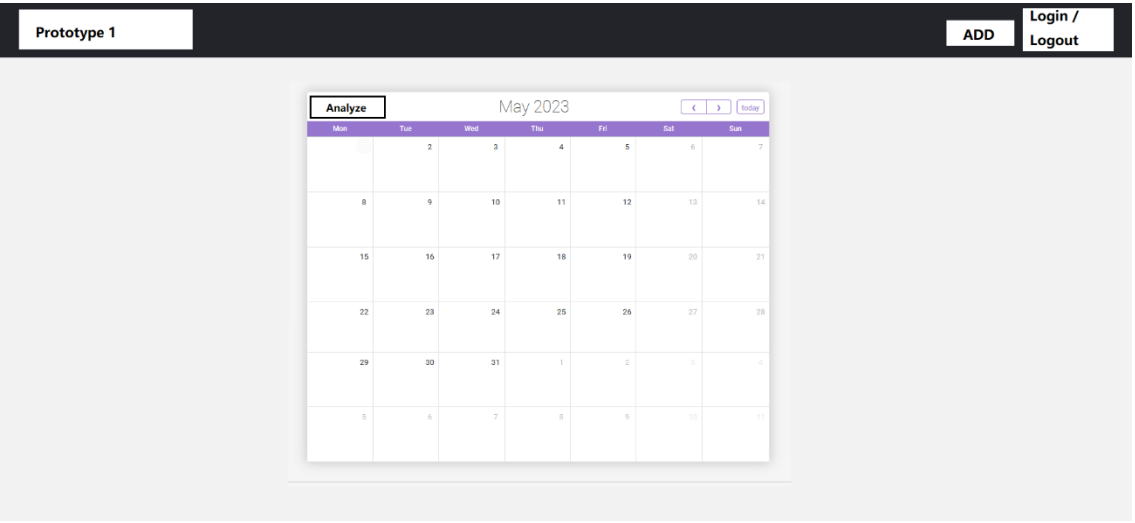
3.6.1 实现简述及约定

用户提出对一个日立的统计请求，系统收到请求后向数据库取数据，做相应的计算，有了统计结果，将统计结果显示在用户界面

3.6.2 功能实现流程



4 接口设计



4.1 外部接口

4.1.1. 用户界面

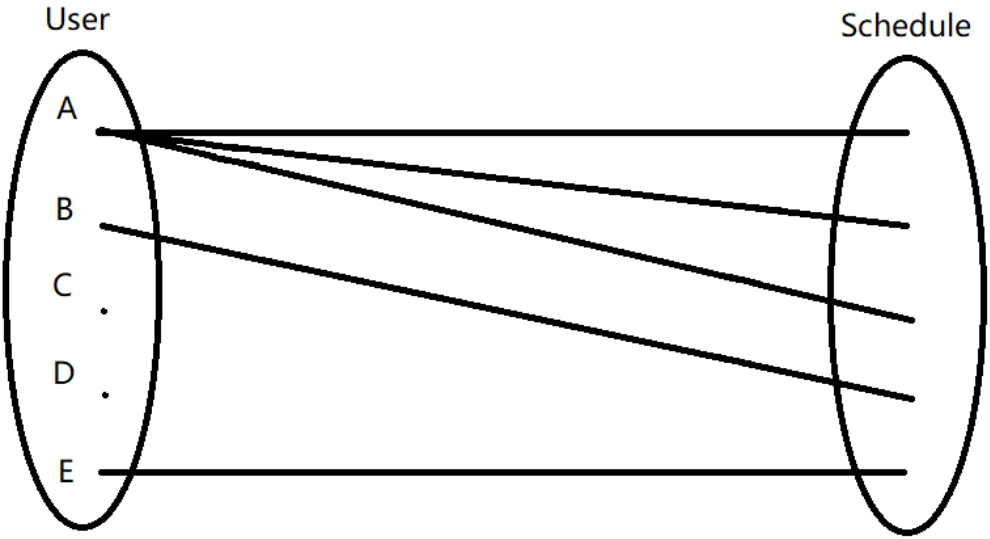
- 1) 功能：编辑按钮，删除按钮，显示日程描述按钮，新加日程按钮，登录/登出按钮，统计按钮
- 2) 编辑按钮：
通过选定某一日的日程，按“编辑”按钮，跳到编辑页面，进行编辑
- 3) 删除按钮：
通过选定某一日的日程，按“删除”按钮，进行删除
- 4) 显示日程描述按钮：
通过选定某一日的日程，显出日程描述
- 5) 新加日程按钮：
在上部，点击“新加”按钮，跳到添加信息页面，进行编辑
- 6) 登录、登出按钮：
在上部，点击“Login”或“Logout”按钮，进行登录或登出
- 7) 统计按钮：
在一个日历顶部，点击“统计”按钮，进行统计并显示一个月内的数据

4.2 内部接口

因为使用框架会提供接口，通过参数表传递数据，交换信息

5 数据结构设计

一下是对于设计的两个表的关系：



5.1 对于用户管理表

字段名字	数据类型	字段属性			
		字 段 大 小	Unique	Primary	说明
teacher_id	Int	30	是	是	用户编号
first_name	Varchar	30	否	否	用户名字
last_name	varchar	30	否	否	用户性
created_date	Timestamp				用户账号制造时间
email	Varchar	30	是	否	用户邮件
phone	Varchar	20	否	否	用户电话号
pwd	Varchar	30	否	否	用户账号密码

5.2 对于日程管理表

字段名字	数据类型	字段属性			
		字段大小	Unique	Primary	说明
sch_id	Int	30	是	是	日程编号
user_id	Varchar	30	否	否	用户编号
sch_name	varchar	30	否	否	日程名称
created_date	Timestamp				日程制造时间
sch_desc	Varchar	100	是	否	日程描述
year	Varchar	4	否	否	日期格式-年
month	Varchar	2	否	否	日期格式-月
day	Varchar	2	否	否	日期格式-日