总体设计

1. 引言
   1. 编写目的

该系统是为了平时个人日程比较细碎且繁多，希望能有效系统地管理个人日程，并及时的提醒日程的人群而设计的。

* 1. 设计思路

该系统的设计基于日历模块，在日历模块，用户可通过单击某一天日期进行日程事件的添加；此时系统将跳转到日程添加模块，在日程添加模块，用户可以添加并编辑日程。此外该系统还有日程总览模块，即在该模块，可以看到所有按时间排序的已添加的日程，并且可以通过点击该日程，从而进行对该日程的编辑和删除。该系统的另一模块为分析模块，用户点击分析按钮，即可设置按周或月分析每一周或每一月的日程完成度等。

2 系统概述

2.1 模块划分

该系统主要包括四个功能模块：日历模块、日程添加模块、日程总览模块、分析模块

2.2.1 日历模块

此模块可按月或查看示日期，并通过单击某一天进行日程的添加

|  |
| --- |
| 日期 |
| +年：int=year  +月：int={1,2…,12}  +日：int={1,2,…,31} |
| +获取日期（）  +添加日程（） |

2.2.2 日程编辑模块

|  |
| --- |
| 备忘录 |
| +事件：string=name  +添加日期：int=data  +截止日期：int=data |
| +添加（） |

2.2.3 日程总览模块

可以总览所添加的日程，并删除某一日程，或者修改某一日程的内容，如：日程标签、日程时间、日程名称等

|  |
| --- |
| 查看事件 |
|  |
| +删除日程（）  +修改日程（） |

2.2.4 分析模块

用户可以查看每个“标签”一周或者一个月的完成度

|  |
| --- |
| 分析 |
| +属性[0..\*]:string={学习，运动，工作}  +事件完成情况[0..\*]:Boolean={0,1}  +完成度[0..\*]:integer={10%，50%，100%} |
| +修改属性名(属性:string=新属性)  +修改完成情况(事件完成情况；Boolean={0,1}) |

3 软件结构

3.1顺序图

该系统采用面向对象的设计方法

查看

分析

备忘录

日历

用户

1.查看日期

2.返回日期

3.添加备忘录 5.查看、删除、修改

4.返回添加成功 6.返回信息

7.查看分析数据

8.返回数据

3.2 结构图

日程管理系统

日历

添加模块

查询模块

分析模块

修改属性名及事件完成情况

查询所有未完成事项

查询所有完成事项

删除待办事项

修改待办事项

查询所有待办事项

查询特定日期待办事项

查询日期

属性

事项

日期

4.开发平台

该系统基于Java语言开发。