

# “任务规划软件” 开发文档

## 一、 总体需求

- 1) 开发一个能发布任务、发布日程、发布反思、发布笔记的桌面小软件。
- 2) 使用此软件需要登录。
- 3) 关闭软件后数据能够存储。
- 4) 代码量 2000 行。

## 二、 具体实现

### 1) 登录和主界面

功能：

- a) 登录：用户输入用户名和密码，点击登录或回车即可进入主界面，点击重置即可重新输入。
- b) 主界面：可以选择进入任务、日程、反思、笔记 4 个页面。

登录布局：如图 1



The image shows a login window for a task planning software. The window has a title bar with the text '登录页面' and standard window controls. The main content area has a large title '任务规划软件' in a stylized font. Below the title, there are two input fields: one for '用户名' (Username) and one for '密码' (Password). At the bottom of the form, there are two buttons: '登录' (Login) and '重置' (Reset).

图 1 登录界面

主界面布局：如图 2



图 2 主界面

登录设计:

- a) LoginView 用 springlayout 实现布局，并且设置“登录”为默认按钮。
- b) LoginHandle 实现事件处理。点击登录，将获取的用户名和密码传入 Data.login 方法，在数据库中进行检索。点击重置，则将用户名和密码清空。

主界面设计:

- a) MainView 实现布局。
- b) MainHandle 实现事件跳转处理。

## 2) 发布任务

功能:

- a) 发布一个具有名称、起止事件、重要紧急程度、具体落实的任务，提供重置处理。
- b) 在下方显示已经发布的任务（初始化时和添加任务时），增加 id 栏，可以修改、删除任务。
- c) 提供关键词查找和排序。

布局：如图 2

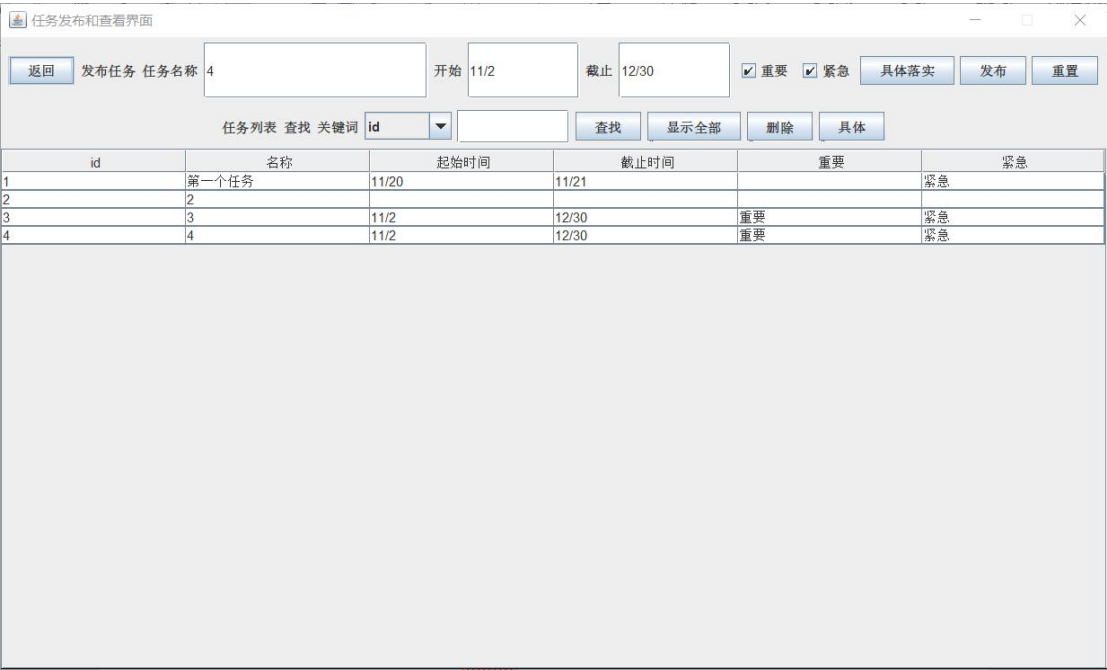


图 3 任务界面

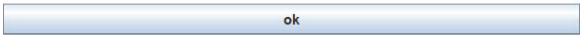


图 4 SetView

设计：

a) TaskView 实现布局和初始化。

在北部用 northPanel 实现 flowlayout 布局。

在南部，用 BorderLayout 布局的 southPanel 实现。southNorthPanel 实现按钮，southSouthPanel 实现 Jtable。

最后将 northPanel 和 southPanel 添加到 View 中。

初始化：调用 Data.getAllTasks 方法，初始化 task 数组，并进行渲染。

- b) Task 类实现任务类，具有 id、名称、起止时间、重要紧急程度、内容等属性。
- c) TaskHandle 实现事件处理。

发布时，将新的 task 加入 task 数组中，并用 Data.updateAll 更新数据库，再将此 task 加入渲染的 data 向量，进行渲染。

删除时，操作类似发布。

点击“具体落实”和“具体”，跳出一个新页面，用 SetView(图 4)实现。

前者是空白的，是在发布任务时编写具体落实的，点击“ok”则将数据返回。后者则是会显示此任务的具体落实，并可编辑，点击 ok 后会更新数据库。

点击“查找”，会根据关键词确定查找的栏目，进而将所有匹配的进行渲染。

编辑单元格时，也需要对 task 数组和数据库进行更新。

### 3) 发布日程

功能：实现一个日历，可以在上面添加日程，并可以查找具体的日期。

布局：如图 4

一	二	三	四	五	六	日
11.15	11.16	11.17	11.18	11.19	11.20 写java开发文档	11.21
11.22	11.23	11.24	11.25	11.26	11.27	11.28
11.29	11.30	12.01	12.02	12.03	12.04	12.05
12.06	12.07	12.08	12.09 今天	12.10 暴躁	12.11	12.12

图 5 日程界面

设计：

a) ScheduleView 实现布局。布局类似 TaskView。

并利用 Data.initialDay 进行具体日程数据的读入 dayList。

之后计算出表格开头的日期（依据使用此软件时的日期必须显示在第一行），并根据此开头日期通过 scheduleFromNow 进行初始化渲染，即将具体的月日添加到每个单元格的头部。scheduleFromNow 是根据 dayList[mouth][day]数组里面存放的 Day.content 来得到每个日期的具体日程。

b) Day 类实现日期类。有 day、mouth、content 三个属性。

c) ScheduleHandle 实现事件处理。

“查找”时，同样计算好开头日期，并根据 scheduleFromNow 得到新的日程 data 进行渲染。

修改单元格时，将修改好的单元格内容更新到 dayList 中，并且用数据库

Data.updateDay 更新。

#### 4) 发布反思

功能：将删除过的任务显示在反思界面，并可以对选中任务进行反思。

布局：如图 5

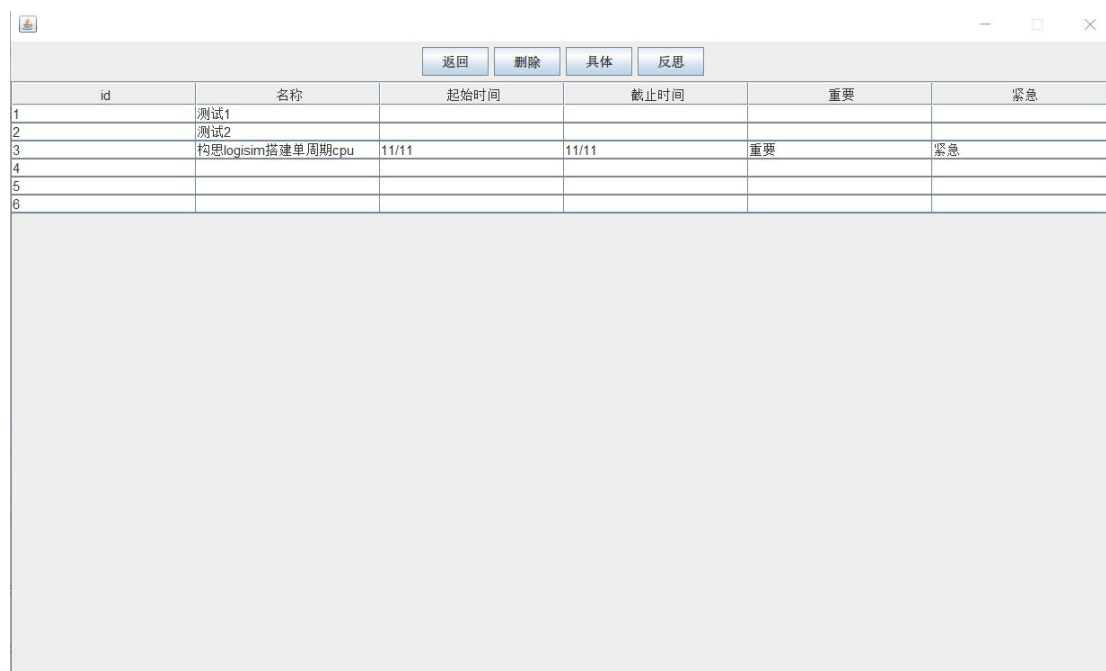


图 6 反思界面

设计：

a) Review 实现界面布局。同日程界面。

并且从数据库中用 Data.getAllReflects 得到具体的反思数据，进行初始化渲染。

b) ReflectHandle 实现事件处理。

“反思”，就是调用 SetView（图 4）界面，显示反思，并且可以修改，要更新 reflect 数组和调用 Data.updateReflect 更新数据库。

“具体”，同样调用 SetView 界面，只不过是不可编辑的状态。

“删除”，用 Data.deleteReflect 更新数据库，并且要更新 reflect 数组和 jtable 的 data 数组。

## 5) 发布笔记

功能：发布、查找、删除、更新笔记

布局：如图 7



图 7 笔记界面

设计：

a) NoteView 实现布局。JTabbedPane 实现笔记本的效果。

用 Data.getAllNotes 读取数据库中已有的笔记并初始化。

b) Note 类实现笔记本类。有 title 和 content 两个属性。

c) NoteHandle 实现事件处理。

“发布笔记”时，检测笔记名是否有重名，有则弹窗提醒，无则更新 note 数组和用 Data.addNote 更新数据库，并用 JTabbedPane.addTab 增加笔记，同时将新增笔记选中。

“删除笔记”时，更新 note 数组且用 Data.deleteNote 更新数据库，并用 JTabbedPane.removeTabAt(index)来删除笔记。

“修改标题”时，同样检测重名，并用 Data.updateNote 更新数据库，其他操作类似。

“保存”时，注意用 Data.updateNote\_content 来更新数据库。

“查找笔记”时，以笔记名为 key。

创建笔记布局时，用 createTab，返回 JTextArea，并且其添加键盘监听，

当监听到 ctrl+s 时，调用 save（），等同于点击保存按钮。

## 6) 数据库存取

功能：对该软件的数据进行持久化存储，并在初始化时取用。

设计：用 Data 类实现数据库类，并且用 mysql 和 navicat 软件进行数据的存取。下表是该类中的方法汇总。

表 1 数据库类方法

Name	Function
login	检测用户名和密码
getAllTasks	将数据库中存放的所有 task 传至 task 数组
deleteAll	删除数据库中所有的 task
updateAll	用 task 数组更新数据库中的 task
initialDay	将数据库中存放的所有 day 传至 dayList 数组
updateDay	更新数据库中具体某月某日的内容
addReflect	添加反思到数据库中
getAllReflect	将数据库中存放的所有 reflect 传至 reflect 数组
deleteReflect	删除数据库中的某个反思
updateReflect	更新数据库中的某个反思



getAllNotes	将数据库中存放的所有 note 传至 note 数组
deleteNote	删除数据库中的某个笔记
addNote	添加笔记至数据库中
updateNote	更新数据库中的某个笔记的标题
updateNote_content	更新数据库中的某个笔记的内容

下图是 navicat 数据库界面：

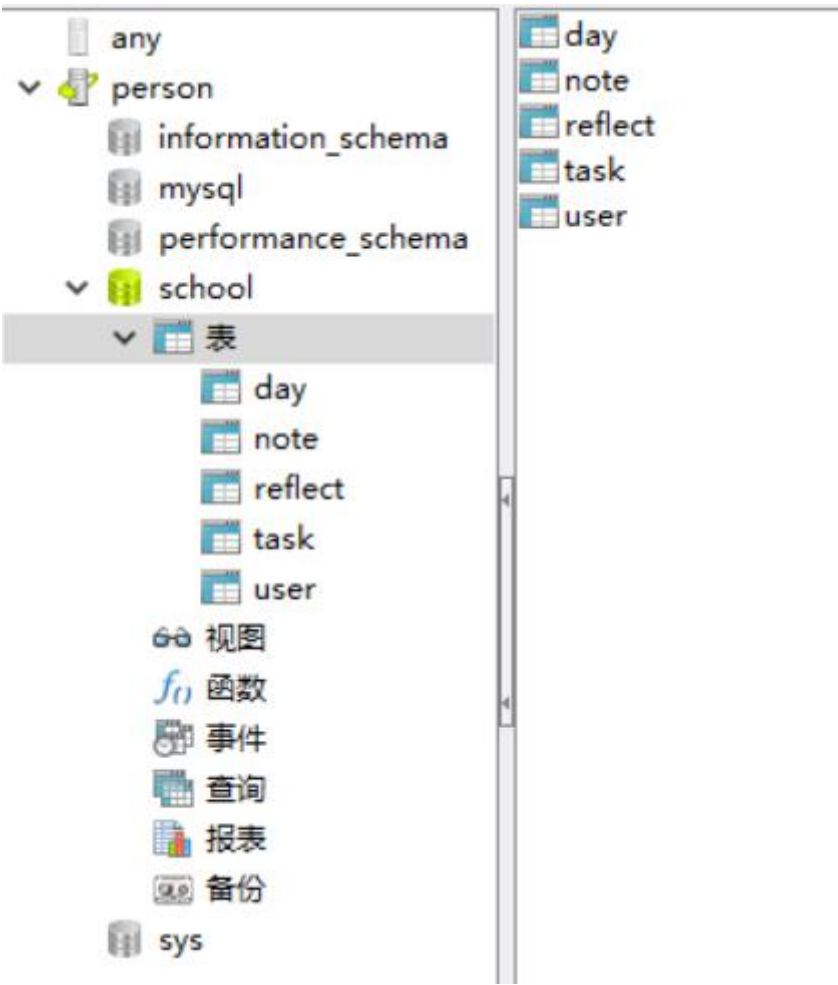


图 8 navicat 数据库界面

### 三、 完成情况

总代码量 2000 行左右，基本完成了 4 个主要功能的设计，用到 mysql 数据库。