# “任务规划软件”开发文档

1. 总体需求
2. 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。
3. 使用此软件需要登录。
4. 关闭软件后数据能够存储。
5. 代码量2000行。
6. 具体实现
7. 登录和主界面

功能：

1. 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。
2. 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



图1 登录界面

主界面布局：如图2



图2 主界面

登录设计:

1. LoginView用springlayout实现布局，并且设置“登录”为默认按钮。
2. 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：

1. MainView实现布局。
2. MainHandle实现事件跳转处理。
3. 发布任务

功能：

1. 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。
2. 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。
3. 提供关键词查找和排序。

布局：如图2

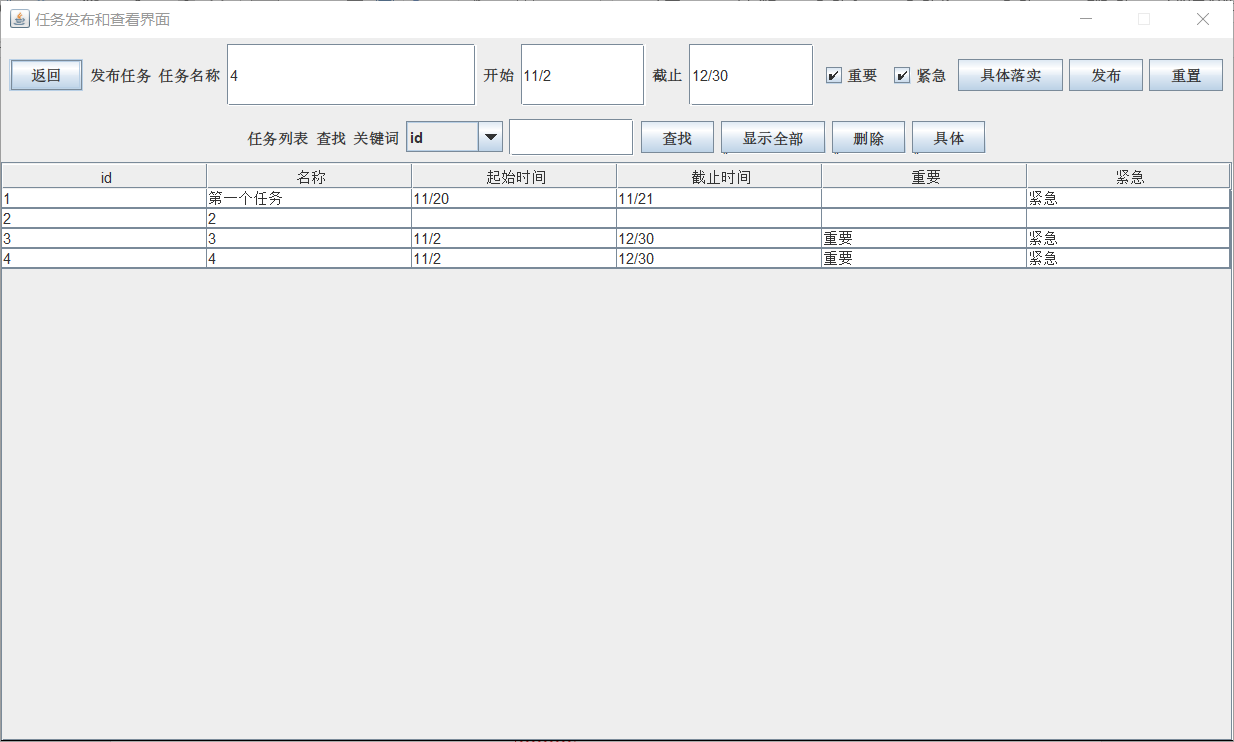


图3 任务界面



图4 SetView

设计：

1. 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。

1. 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。
2. 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。

1. 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

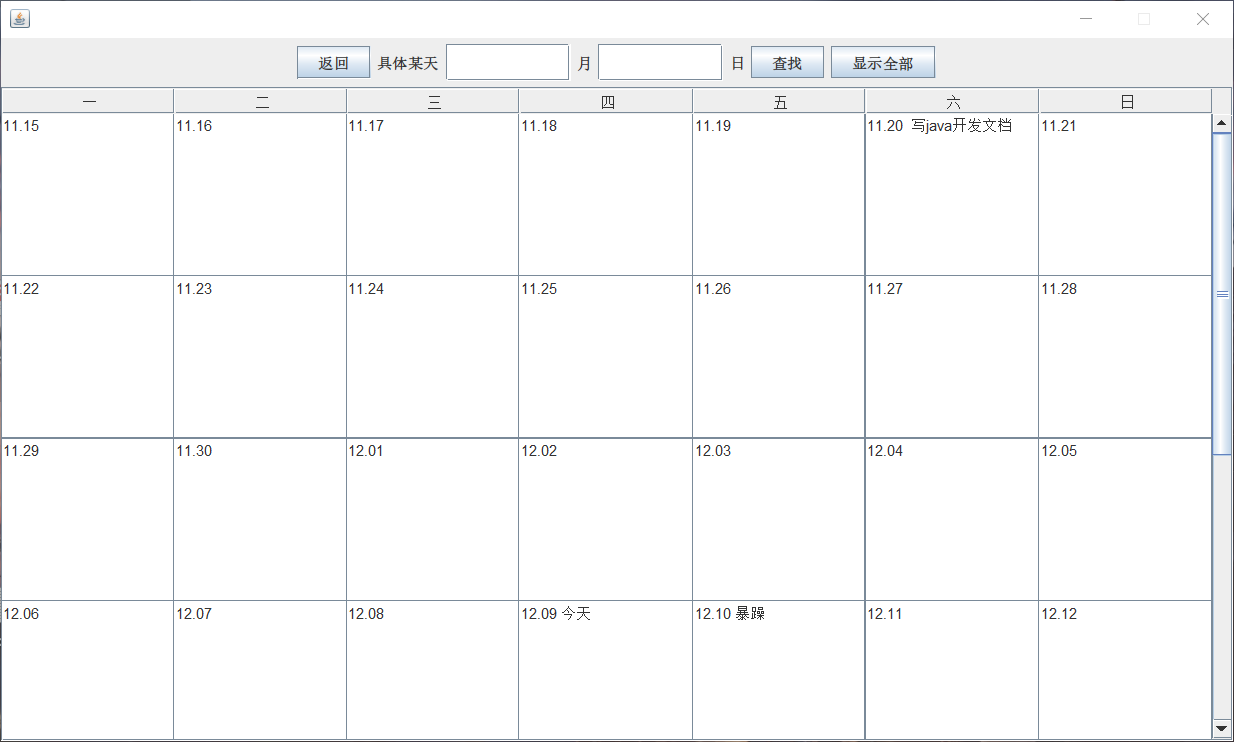


图5 日程界面

设计：

1. 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。

1. Day类实现日期类。有day、mouth、content三个属性。
2. 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。

1. 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

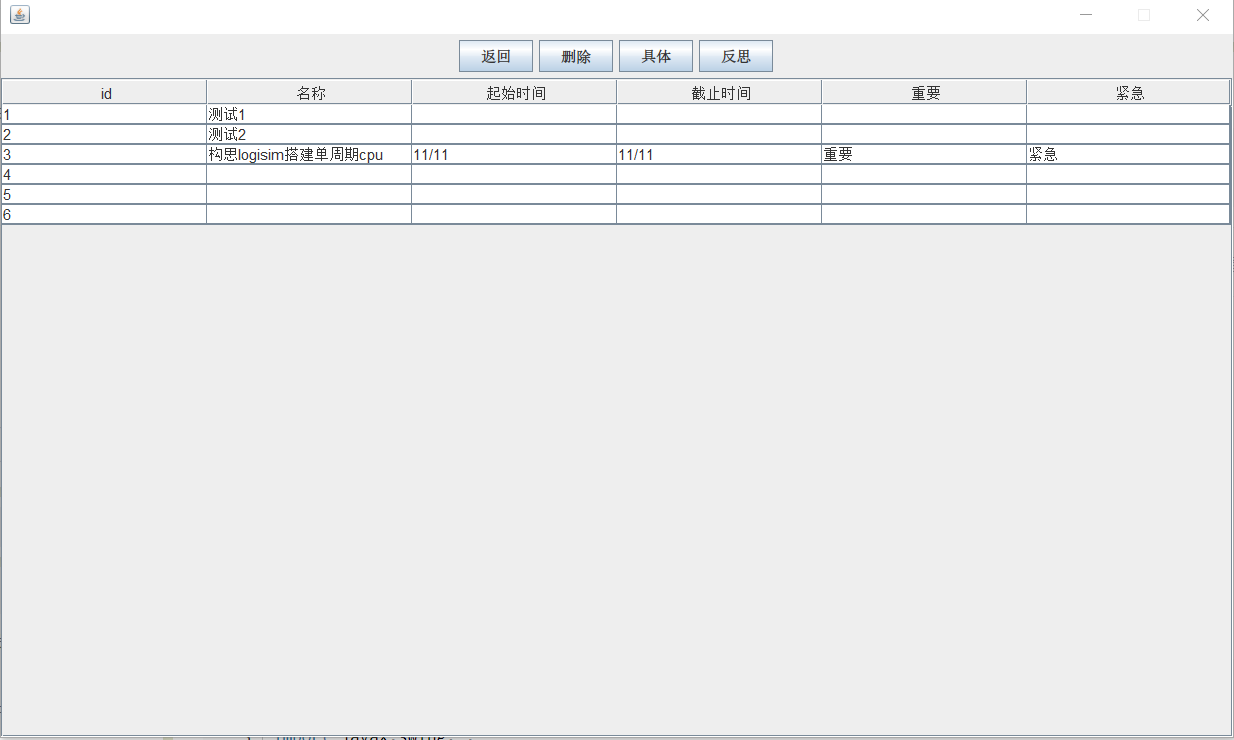


图6 反思界面

设计：

1. 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。

1. 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。

1. 发布笔记

功能：发布、查找、删除、更新笔记

布局：如图7

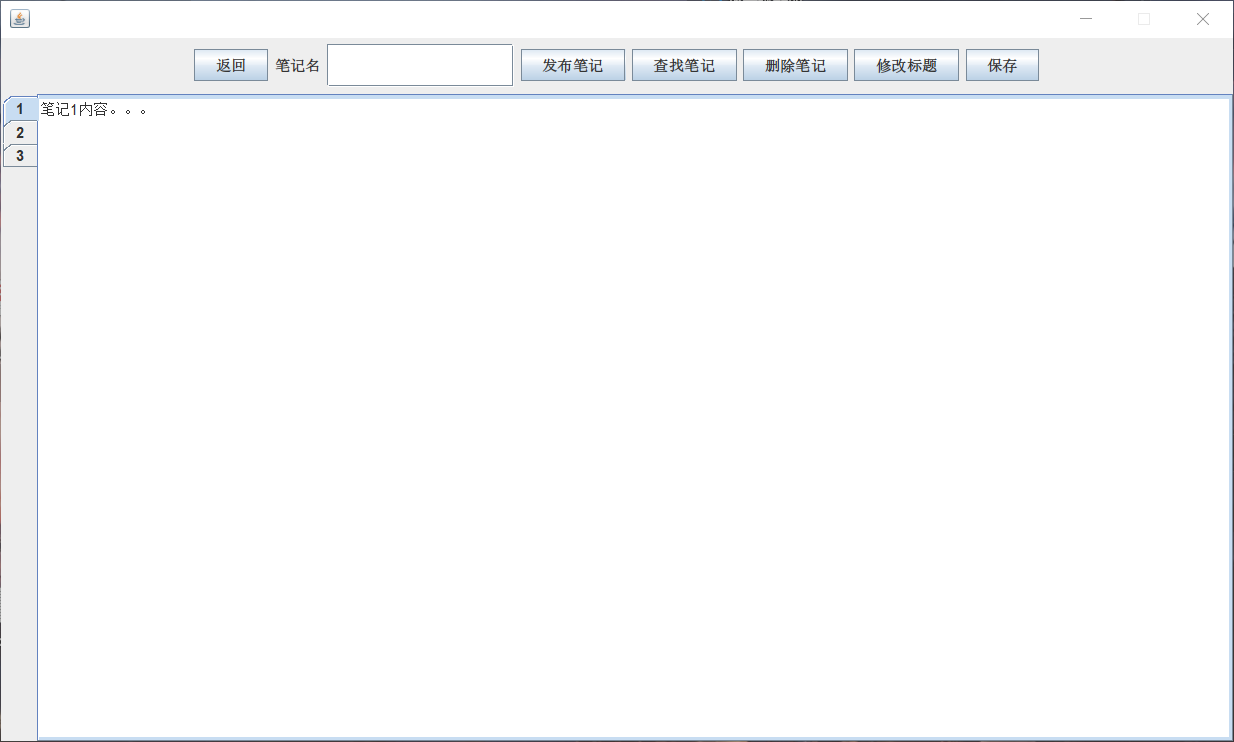


图7 笔记界面

设计：

1. 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。

1. Note类实现笔记本类。有title和content两个属性。
2. 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。

1. 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：

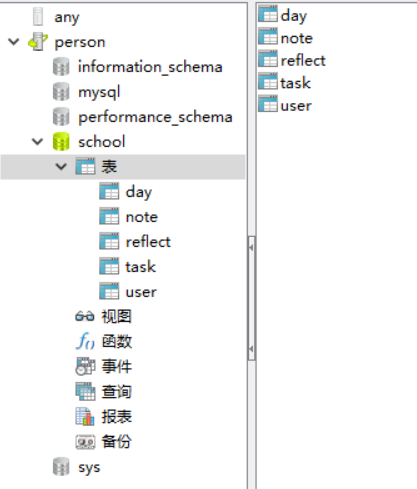


图8 navicat数据库界面

1. 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。