**桌面日程表**

1. **前言该概要设计是以概要设计文档为基础，沿用思路继续设计进而得到。**

**概要设计文档中已经包含**

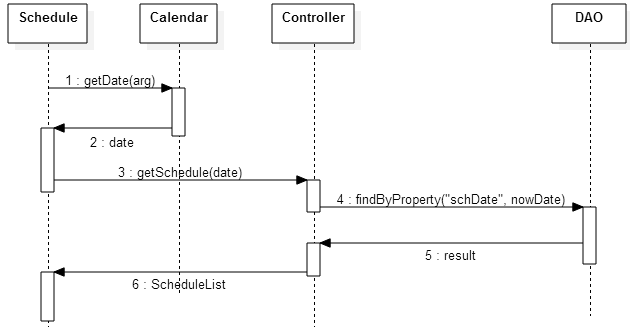
1. **领域模型**
2. **用例解释**
3. **架构设计**
4. **数据库设计**
5. **UI设计**
6. **其他要求**

**且已经较为完善。**

**文档通过对领域模型、架构设计、数据库设计以及考虑其他要求进一步分析得出，在原有设计的基础上给出了更为详细的设计方案， 直接用于程序编写。**

1. **架构设计**

**用户获取对应日期日程表的顺序图：**

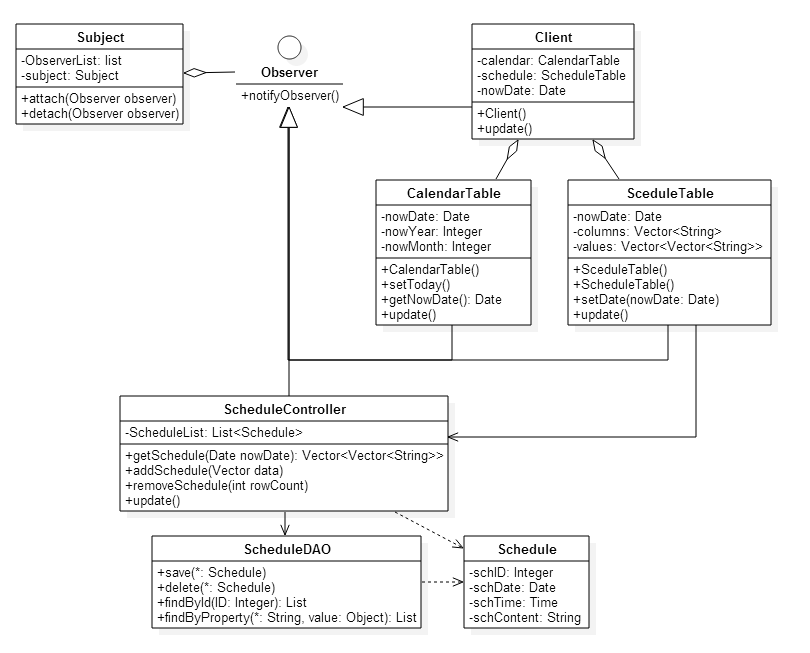
****

**用户选定日期后ScheduleTable类自Calendar类获取该日期，然后将改入Controller。Controller自Hibernate层DAO中使用findByProperty方法获取日程表然后返回至Controller封装后传入Schedule类显示。**

**(添加删除日程基本相同)**

1. **完整类图**

**根据需求分析、概要设计、详细设计等文档，最终给出如下类图。**

****

**项目采用Observer模式，其中Controller以及所有UI均属于Observer，每当选择一个新的日期时调用Subject中notifyObservers()刷新Observer**

1. **函数解释**

**Observer.update()**

**该函数接口用来刷新所有需要刷新的(实现了Observer接口)的类**

**所有UI以及Controller均属于Observer且会使用update()函数刷新数据。**

**Subject.getSub().attach()**

**将类放入观察者列表中**

**Subject.getSub().notifyObservers()**

**通知所有观察者刷新**

**Client.Client()**

**调用主界面，同时创建日历和日程并刷新。**

**CalendarTable:**

**CalendarTable() 创建日历表**

**setToday() 将返回今日日期**

**Date getNowDate() 获取选中的日期**

**SceduleTable**

**ScheduleTable() 创建日程表**

**setDate(Datenow Date) 设置日程表中日期**

**ScheduleController**

**Vector<Vector<String>> getSchedule(Date nowDate) 获得日程表表格数据**

**addSchedule(Vector data) 添加新日程**

**removeSchedule(int rowCount) 按行号移除日程**

**ScheduleDAO**

**save(Schedule \*) 保存日程**

**delete(Schedule \*) 删除日程**

**List findById(Integer ID) 按ID查询返回结果列表**

**List findByProperty(String \*, Object value) 按属性查询**