日程管理

软件构架文档

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <04/07/2014> | <1.0> | 文档的初步完善 | 张蔚 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 2

1.1 目的 2

1.2 范围 2

1.3 定义、首字母缩写词和缩略语 2

1.4 参考资料 2

1.5 概述 2

2. 构架表示方式 2

3. 构架目标和约束 2

4. 用例视图 2

4.1 用例实现 2

5. 部署视图 2

6. 大小和性能 2

7. 质量 2

软件构架文档

# 简介

## 目的

本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

## 范围

本文档适用于张蔚单人小组暑假小学期开发的日程管理项目。

## 定义、首字母缩写词和缩略语

QT——C++图形类库**逻辑视图**

ScheduleManagement——日程管理

ToDoList——任务列表

CalenderMap——日历及相关功能主界面绘制类

CalenderTranser——对日历进行转换等一系列操作的类

Memorandum——备忘录

## 参考资料

暂无

# 构架表示方式

本文档将通过一下视图来说明日程管理软件项目的软件架构：用例视图、部署视图

所有视图都通过PowerDesigner实现

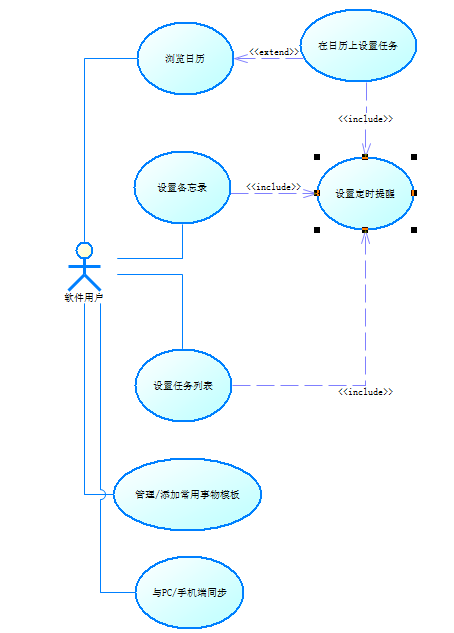
# 构架目标和约束

1. 项目的设计约束：开发语言现阶段为C++，使用MySql数据库存放数据。
2. 系统无须登陆，可以向所有用户提供服务。
3. 备忘录、任务列表的读取及储存时间不超过0.5秒。

# 用例视图

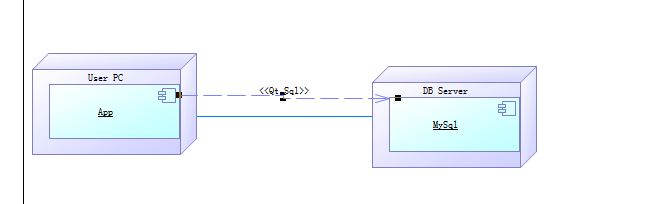
本章是对软件架构的用例视图的描述。可参考软件需求规约及答辩文档

## 用例实现



[本节说明将系统分解为轻量级进程（单个控制线程）和重量级进程（成组的轻量级进程）的情况。本节的内容按照各个通信或交互的进程组来进行组织。说明进程之间的主要通信模式，例如消息传递、中断和会合。]

# 部署视图

本章描述了部署和运行软件的物理配置。

6.1 User App

用户使用单机客户端来享受应用服务。

6.2 DB Server

数据库服务器运行MySql数据库

# 大小和性能

本系统采取的架构针对于单一用户的使用，可满足以下性能需求：

1. 用户普通操作响应时间在0.05s以内，储存操作在0.5s以内
2. 可以支持的日程管理期限不设置上限

# 质量

本系统由于独立于网络而单机运行，所以对质量方面的需求满足情况较好，可满足:

1. 对所有使用安装了该软件的用户提供服务，界面简单无须任何操作指南即可上手。
2. 系统对事件采用数据库存储，数据可靠。