XPlanner

软件架构文档

版本 <3.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <30/08/2018> | <1.0> | 初步编写了软件架构文档 | 汪喆昊 |
| <06/09/2018> | <2.0> | 补全了缺失的部分 | 李琥、汪喆昊 |
| <09/9/2018> | <3.0> | 审核了文档 | 同睿哲 |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 参考资料 4

2. 用例视图 4

3. 逻辑视图 5

3.1 概述 5

3.2 在构架方面具有重要意义的设计包 6

4. 进程视图 6

5. 部署视图 7

6. 实现视图 7

软件架构文档 （简化版）

# 简介

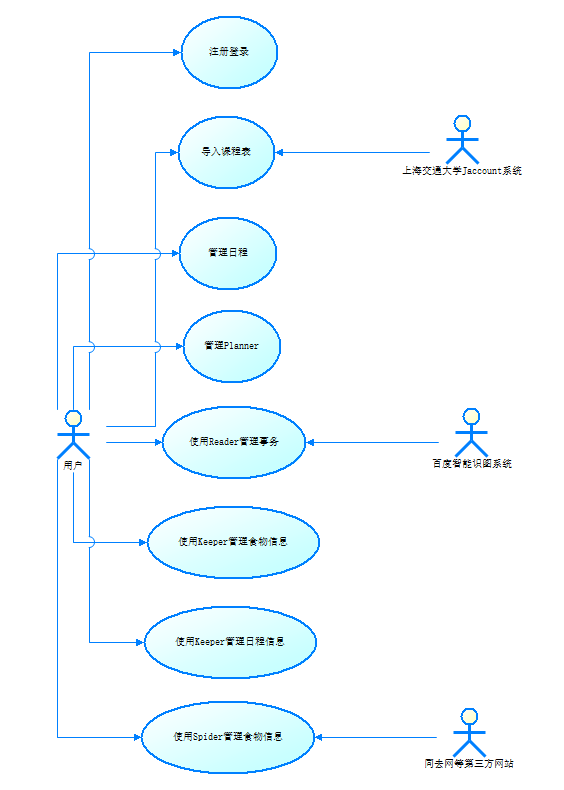
## 目的

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。本文档用于说明XPlanner项目的软件架构，帮助开发人员在维护代码时快速了解项目架构，并且帮助审核人员了解项目大致情况。文档的参考资料见1.2节，用例视图见2节，逻辑视图见3节，进程视图见4节，部署视图见5节，实现视图见6节，数据视图见7节，核心算法见8节。

## 参考资料

无外部引用资料。

# 用例视图



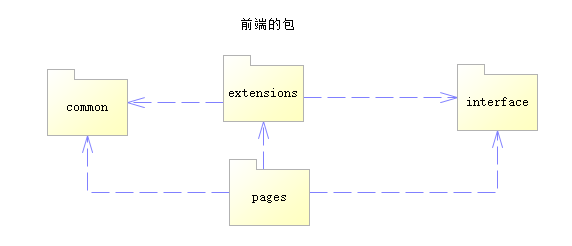
# 逻辑视图

## 概述

项目分为前端和后端两大子系统。前端有common，extensions，interface，pages这4个包，Java后端有DAO，Config，Converter，Model，Service，Security，Web，此外还有python后端的QRCode包。

前端的所有包包含运行在微信客户端中的文件，主要逻辑由javascript文件负责，界面由wxml、wxss文件负责。微信小程序框架的语法本身已经包含了类的思想和清晰的结构，没有太大必要定

义类，所以前端包中没有显示地使用类的语法，但是在逻辑上仍有类的存在。



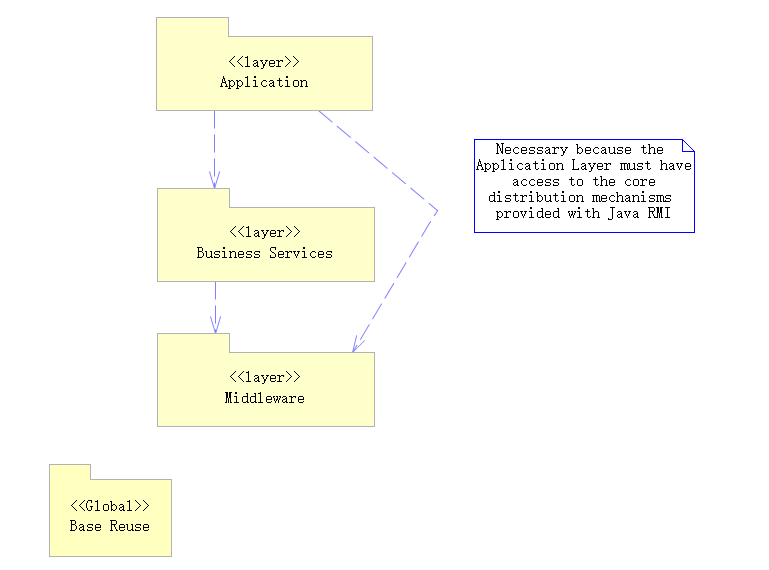
common包包含了extension、schedule、spider、time4个类。分别封装了处理扩展、日程、爬虫获取事务、时间的函数。

extension包包含项目中实现的3个扩展，分别为reader、keeper、spider，负责实现各个扩展自身的界面和功能。

interface包包含wrapper一个类，用于封装前后端通信使用的函数。

pages包包含项目的4个主要页面，分别为scheduler、store、me，负责实现各个页面及其子页面的界面和功能。

Java后端的DAO包实现了各种对数据库的访问功能，用来底层对数据库进行增删改查。



Config包括了各种后端代码的全局变量，系统设置等等。

Converter是一个实现了前后端数据传输转化的包。

Model包对数据表的各张表、各种数据库进行了建模，使它变成了Java类，让开发人员可以面向对象编程。

Service使后端逻辑的主要实现的地方，实现了各种逻辑代码，例如Spider、Parse等，是逻辑最复杂的地方。

Security建立了用户的认证登录体系，并实现了权限控制、用户认证等。

Web主要了实现了各种Controller，建立endPoints和RestfulAPI.

Python后端只有一个包，即QRcode包，主要实现了二维码的获取，相对Java后端来说，功能比较单一。

## 在构架方面具有重要意义的设计包

前端中比较重要的包为extension和pages。后端中比较重要的包为DAO、Service、Web。extension包包含reader、keeper、spider三个模块（逻辑上可以看作类的实现）。reader负责reader页面的界面和逻辑，用于从用户输入的文字或图片中提取时间信息，生成日程事务。keeper负责keeper页面的界面和逻辑，用于管理、计算用户所吃食物信息，生成体育运动事务。spider负责spider页面的界面和逻辑，用于爬取同去网、教学信息服务网的信息，生成日程事务。需要注意的是，reader、keeper、spider生成日程事务的功能并不是由他们本身实现的，而是由pages中的addSchedule模块实现的，它们只是传递了一些参数给相应的模块。三者均为边界类和控制类的混合体。

pages包包含scheduler、store、me四个模块。scheduler负责日程表界面的逻辑，是项目前端部分的核心，用于日程的管理（增删改查）。scheduler包含两个子模块，addSchedule和scheduleDetails，分别负责添加、修改日程的界面的实现和显示日程详情页面的实现。store模块实现扩展商城，提供使用扩展（扩展入口）的功能。me中包含了登录界面和逻辑的实现。以上4个模块均为边界类和控制类的混合体。

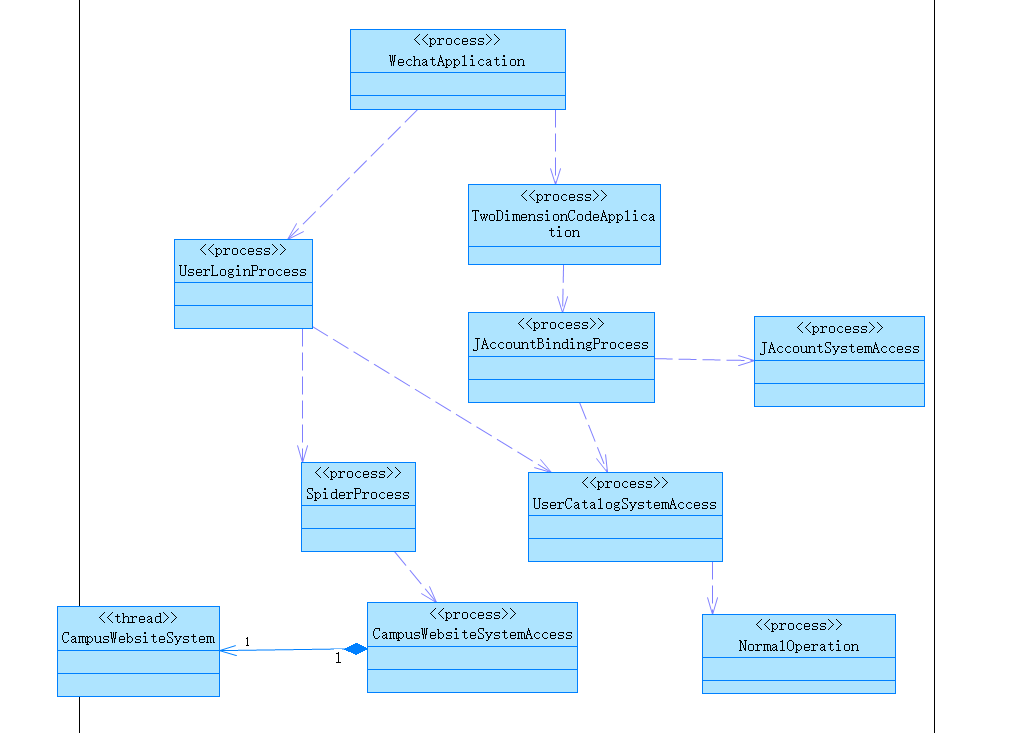
Java后端的三个包即DAO、Service、Web体现了经典的三层架构。

DAO实现了数据库的访问，并且有很多对数据库的底层操作，即增删改查等等。其处在整个架构最底层的位置，为三层架构提供了支撑。

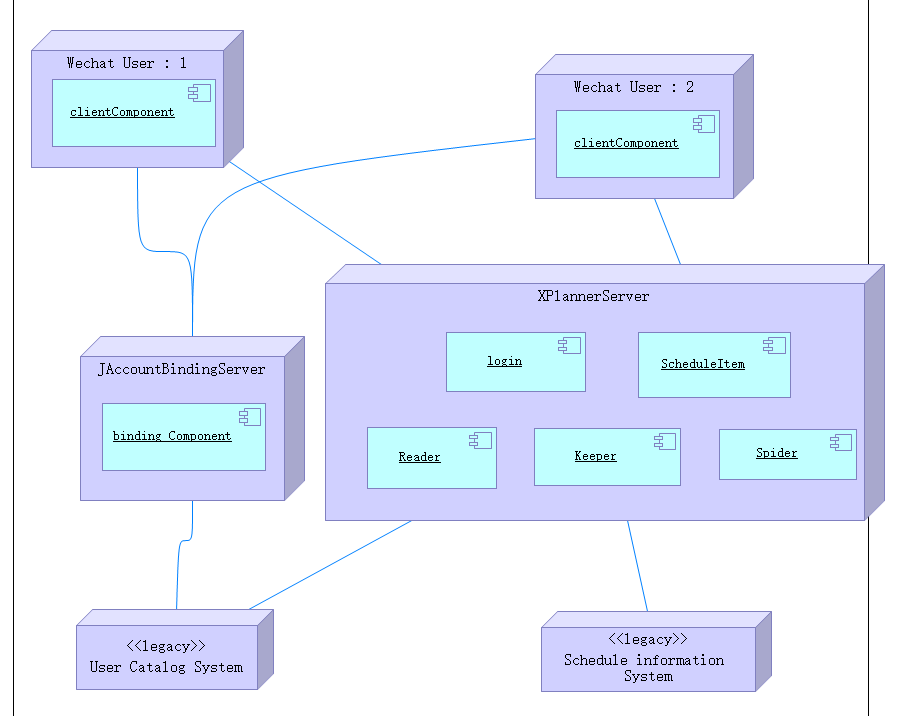
Service层实现了大量的逻辑代码和功能性代码。其处在中间层。在三层架构中处在中流砥柱的位置，例如爬取Spider的功能的实现、转化用户课程表的功能的实现、推荐算法的实现，都在其中。

Web层实现了大量控制类，用于协助其他对象实现用例的功能，其处于三层架构中的最高层，具有良好的隔离作用，系统的其他部分与用例的具体执行逻辑形成松散耦合。

# 进程视图



# 部署视图



# 实现视图

