全部内容总结

开发模式

敏捷开发模式

两人配合？

开发过程控制

结合开发模式，功能实现过程控制

版本控制

代码管理

配合交流

知识和经验总结共享

开发设计和实现

发布与运营

应用设计总结

1. 总体功能

为在校学生提供日常的课表功能，可同时管理多个课表。

同时课表提供作业和笔记的记录功能，按上课时间与课程记录一起管理。

1. 详细功能（要以客户面的功能语言来描述）

周课表查看功能

详细课程信息

多课表管理

作业和笔记管理

1. UCD设计

心得：应用的设计应该从UCD开始，根据详细功能要求设计整个应用的交互方案。这是用户需求实现为应用功能的转接点：将功能设计到应用界面中。之后才能根据应用界面，来做接下来的设计，包括架构和模块内容等。

UCD的设计同时还应包括界面主题和风格的，包括主体的大小和颜色，各控件界面的设计等等。

UCD设计完之后，应该尽量固定不变，一旦改变将为底层带来极大的变化！

1. UCD设计原则：

* 简洁：界面力求简洁精致，数量简但内容要精
* 易用：无需太多额外的学习即能上手
* 风格主题统一：（可在实现过程中不断优化）
* 减少操作：易用强调易上手，减少操作强调用户实现相同功能的操作次数
* 减少界面层级：层级越多感觉越乱，最好不要超过3层。
* 考虑可扩展性：留点后路

1. UCD设计输出

* 总体设计方案
* 与功能对应的用户交互操作流程描述

1. 架构设计

一般采用MVC，但不拘泥，视具体功能而定！

这里就需要把整个应用的类框架同时定下来，比如哪个功能要用哪个类来做，而不是简单画三个MVC的框框。以SchoolLife为例，MVC的架构设计结果可以如下：

（图和表）

1. 实现设计

针对架构设计中结果的功能类及描述，做实现上的设计和编码实现。

实现时要额外考虑如下内容：

* 类、方法与变量说明，用于生成文档
* 问题定位方便（考虑Log打印方案）
* 测试方便