

校园订餐服务系统分析与设计

(The requirement analysis and logical design of ordering service system)

需求概述:

本服务系统的运营者为校园网运营单位（如学校的网路中心）或学校负责餐食及为校园师生日常生活提供服务的管理部门（如后勤部），旨在通过校园网为校园师生提供更好的生活服务，服务形式为将校园餐厅、食堂以及入驻校园的各类实体经营服务者（超市、水吧等等）以及校园周边1公里范围经认证的餐饮经营者纳入服务系统，向校园师生，特别是学生提供各类网上餐饮服务，如线上预订，支付，预订餐位等等。该系统的运营的目标在于提供学生更便捷更好的生活保障服务，而非以盈利为目的。该系统与美团，饿了么等线上服务非排他关系，但存在竞争。

1.范围定义与问题分析

1.识别并指出本系统的用户都包括哪些角色？

order placer, order taker, platform operator, food deliver

找到需求概述中的角色，一般包括用户，后台管理人员以及这里的服务人员等。

2.概述本系统的基本功能

Place order / Payment

Take order / Fulfill order / Delivering / accounting

Handling complains / Tracking and auditing / Monitoring / Platform management

一般都有用户注册登录，平台运营维护等基础功能，在这里还包括服务人员与用户的交互以及订单追踪处理等。

3.系统能够持续运营下去关键在于提供学生独一无二的服务，你是否同意这一观点？如果同意，请简述达成这一点的 关键问题在于哪些？这些问题的解决思路是什么？如果不同意，请指出你所认为的关键及其解决办法？

1, Key points: faster, cheaper, more convinient

2, a statement of problems decomposition and their corresponding possible solutions

关键问题一般就是相较其他竞争对手的优势，按照pieces这几个方面去答就好



2.需求分析与需求定义

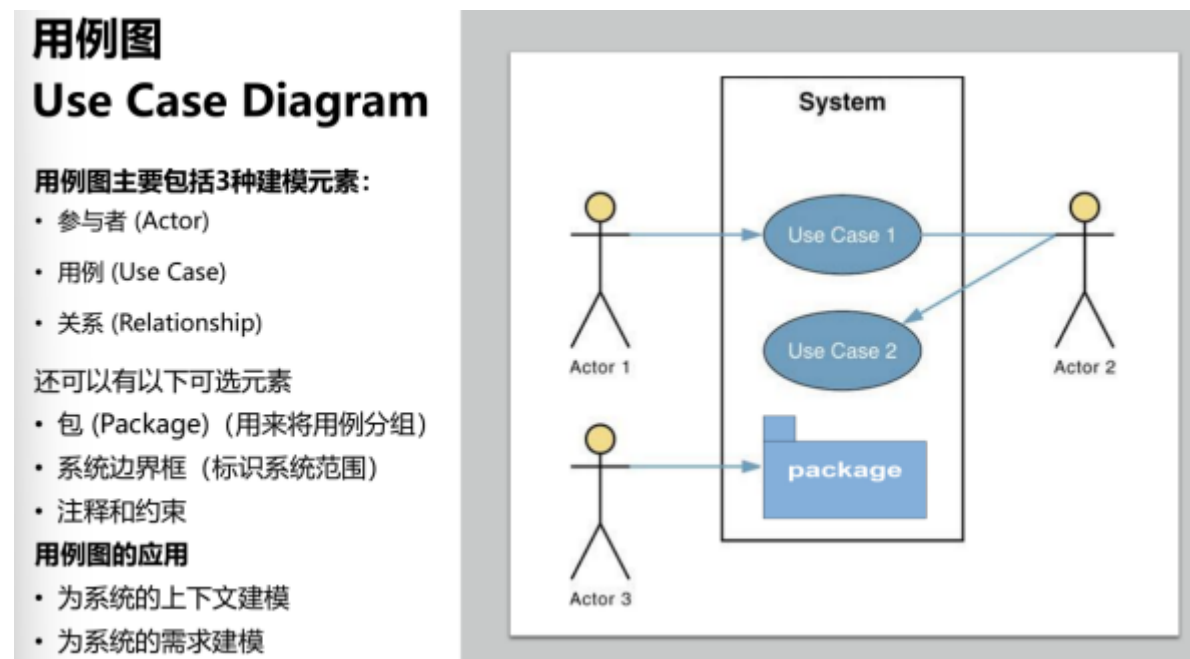
1.请提供餐饮商家的用户故事/用户功能

- 1, As a service provider (e.g, resturant, drink bar, canteen), one can take order from student, and then fullfill the order.
- 2, As a service provider, get and do order accounting.
- 3, complains
- 4, handing over to deliverer and tracking the order

针对某一角色进行更深的功能描述

2.提供商家处理学生订单的用例图及用例描述

用例图

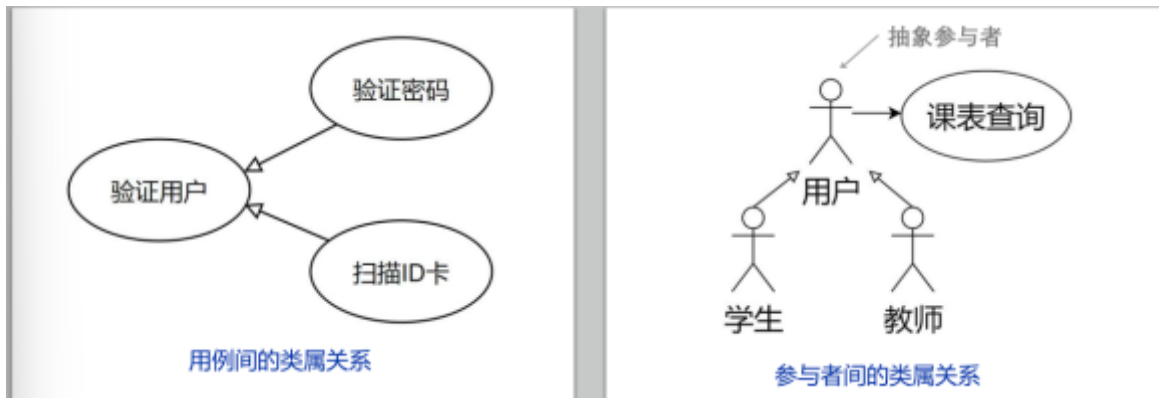
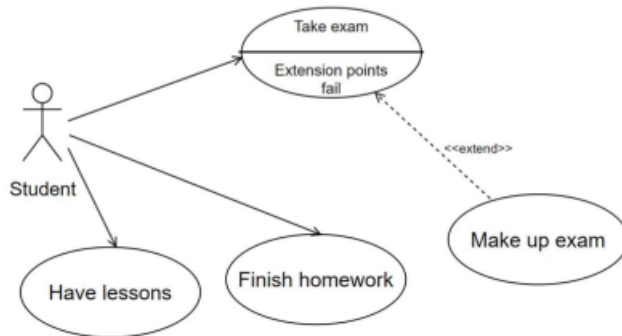
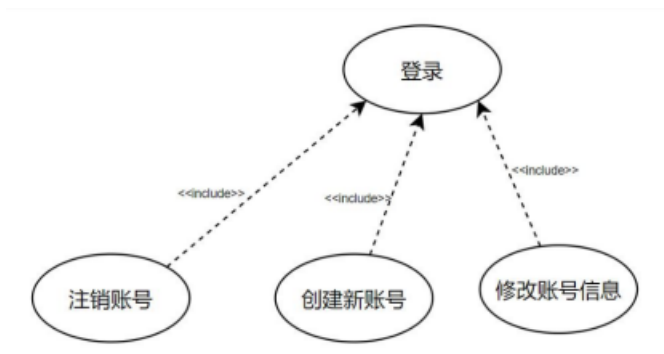


关联关系不用多说就是有交互

包含关系include通常将具有相同功能步骤的用例，放在一个单独的抽象用例中

扩充关系extend特定的用例部分，只有在特定条件下运行的系统行为

类属关系继承父类，也可添加或覆盖新的行为



用例描述

使用结构化叙述的文本描述用例图，填充每个业务事件，并说明用户如何同系统交互的细节。应该包括下列内容：

- 用例什么时候开始，怎样开始
- 用例什么时候结束，怎样结束
- 用例和参与者之间有什么样的交互作用
- 用例需要什么数据
- 用例的标准事件顺序是什么
- 替代或例外事件流如何描述

用例描述模板

Use Case Narratives Template

X.	用例XX（用例名）的事件流
X.1	前置条件 (Pre-Conditions)
X.2	后置条件 (Post-Conditions)
X.3	扩充点 (Extension Points)
X.4	事件流
X.4.1	基流 (Basic Flow)
X.4.2	分支流 (Subflows) (可选)
X.4.3	替代流 (Alternative Flows)

用例描述 Use Case Narratives

- **前置条件**——在用例执行之前，关于系统状态的约束条件。
一般指需要首先执行的另一个用例
- **后置条件**——可能影响用例实现的非功能需求（数据量、接口需求等）
- **扩充点**——只在特定条件下用例执行的事件。通常是一种异常情况。

以下为事件流

- **基流**——参与者和系统为了满足用例目标执行的常规活动序列，包括：系统与参与者的交互和为相应交互系统执行的活动
- **分支流（可选）**——执行附加需求来完善、辅助完成基流的用户行为
- **替代流**——典型过程出现异常或变化时的用户行为

“注册课程”用例描述

“Register for courses” Example

1 用例“注册课程”的描述	
1.1前置条件	在用例“注册课程”开始之前，用例“维护课程信息”的分支流“创建选修课程”必须完成。
1.2 后置条件	如果这个用例成功，学生的选修计划表则被创建。否则，系统的状态没有变化。
1.3 扩充点	无
1.4 事件流	
1.4.1 基流	当学生输入密码登录课程管理系统时，用例“注册课程”开始。如果密码验证是正确的(E-1)，则提示学生选择想要的动作：添加、查看、打印、退出。 如果所选的活动是添加，执行分支流S-1；添加所选课程。 如果所选的活动是查看，执行分支流S-2；查看所选课程。 如果所选的活动是打印，执行分支流S-3；打印所选课程。 如果所选的活动是退出，用例结束并退出。
...	...
1.4.2 分支流	S-1：添加所选课程 系统提示含有课程名和课程代号的域，学生输入希望选修的课程名和课程代码(E-2)，并建立该课程与该学生的连接(E-3)。用例重新开始。 S-2：查看所选课程 系统检索(E-4)并显示出学生所选的所有课程的信息，包括课程名、上课时间地点等。当学生查看完毕，用例重新开始。 S-3：打印所选课程 系统打印出学生所选的课程信息(E-5)。用例重新开始。
1.4.3 替代流	E-1 如果输入的密码无效，用户可以重新输入密码或终止用例。 E-2 如果输入的课程名或代码无效，用户可以重新输入或终止用例。 E-3 如果学生和课程间的连接不能建立，信息会被存储，晚些时候再次建立连接。用例继续。 E-4 如果系统不能检索课程选修信息。用例重新开始。 E-5 如果系统不能打印课程选修信息，学生会得到信息表示该选项目前无法使用。用例重新开始。

3.商家浏览学生订单的功能规格说明

This requirement specification description, can either be a text based description or a use case diagram + narratives.

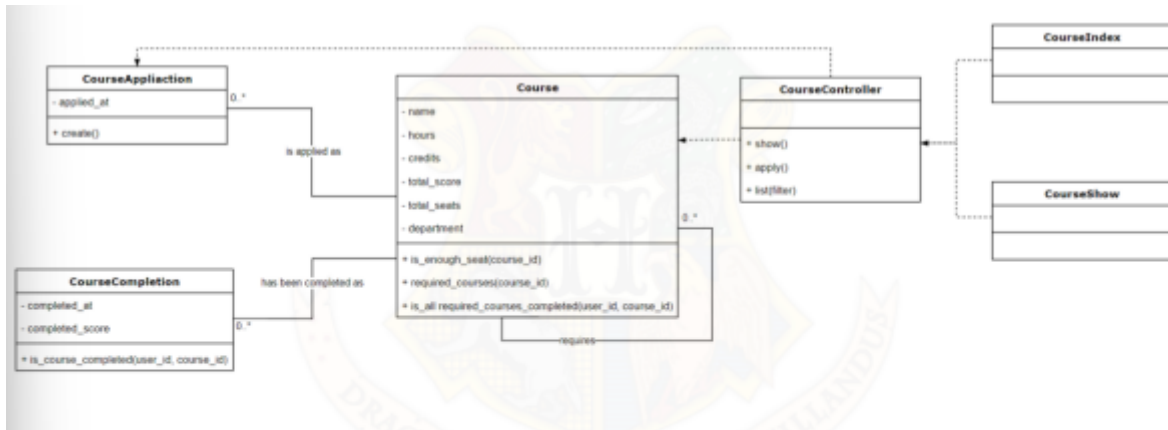
Description need cover all essential steps of the order function.

纯文本描述功能或画出用例图和用例描述

3.建模与设计

1.请用UML类图或ERD图描述本系统的主要业务实体以及关联关系

UML类图



设计类图——一个面向对象系统至少被结构化为三种不同类型的对象类

OOA to OOD

1. 向图中添加设计对象
2. 考虑图的大小和可读性，应该只包含主要的接口对象
3. 为设计对象添加属性、方法和可见性
4. 添加类之间的依赖关系

泛化 继承关系的一种，子类继承父类的所有行为和属性，子类可以新增新的功能或者重写父类功能

关联 使类可以感知其他类的行为和属性

聚合 聚合主要描述整体与部分直接的关系。需求陈述中常使用“包含”、“组成”、“分为...部分”等字句。

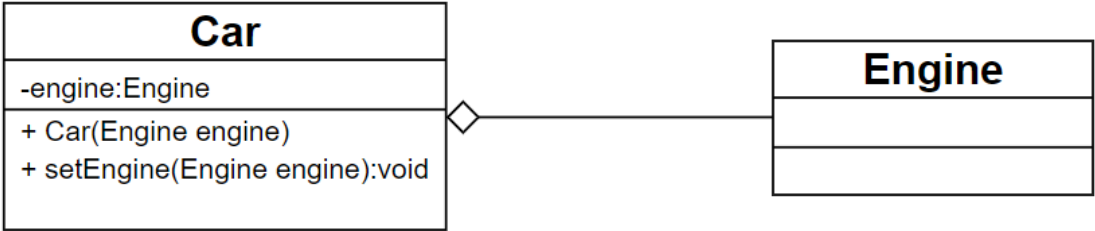
组合 组合聚合是聚合的一种特殊情况，成员对象与整体对象之间具有同生共死的关系

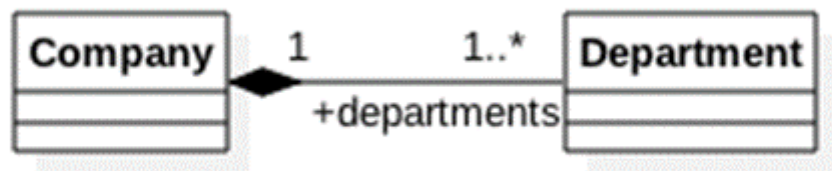
依赖 依赖即一个类的实现需要其他类的协助，通常代码表现为方法参数，局部变量，静态方法调用，util类调用

类的关系

类的关系	符号
泛化关系 (Generalization)	
关联关系 (Association)	
聚合关系 (Aggregation)	
组合关系 (Composition)	
依赖关系 (Dependency)	

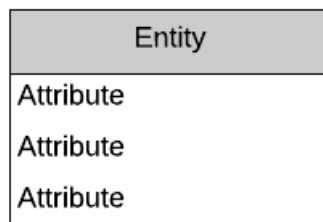
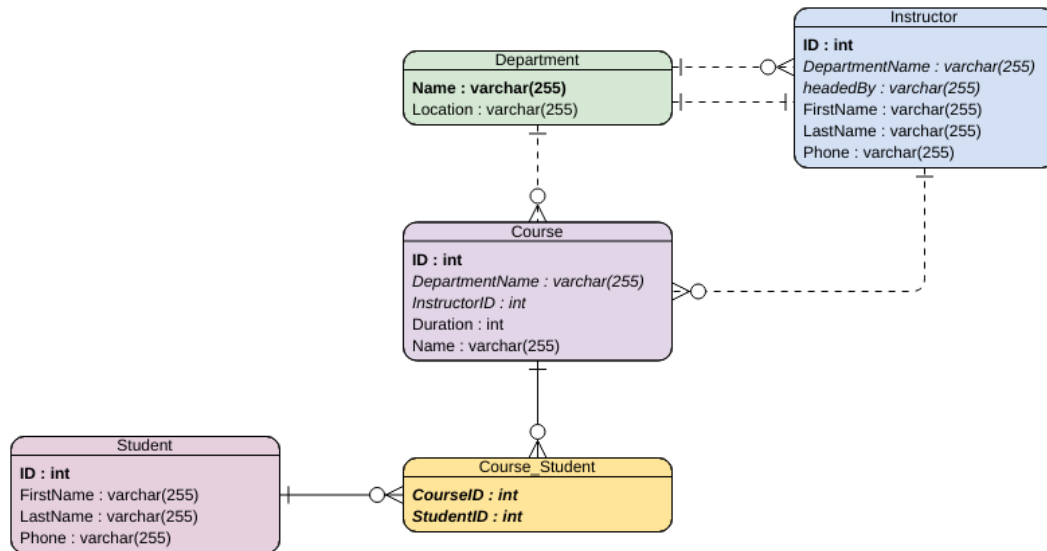
多重性 Multiplicity	多重性记号 UML Notation	带有多重性的关联关系	关联的含义
正好一个	1 或 留白		一个雇员为一个且仅一个部门工作
零个或一个	0..1		一个雇员有一个配偶，或没有配偶
零个或多个	0..* 或 *		一个客户可以没有进行支付，也可以进行多次支付
一个或多个	1..*		一个大学至少提供一门课程，也可提供多门课程
特定范围	<u>m..n</u>		一支队伍可以安排7、8或9场比赛





CSDN @chls

ERD图不用细看



Zero or Many

One or Many

Many

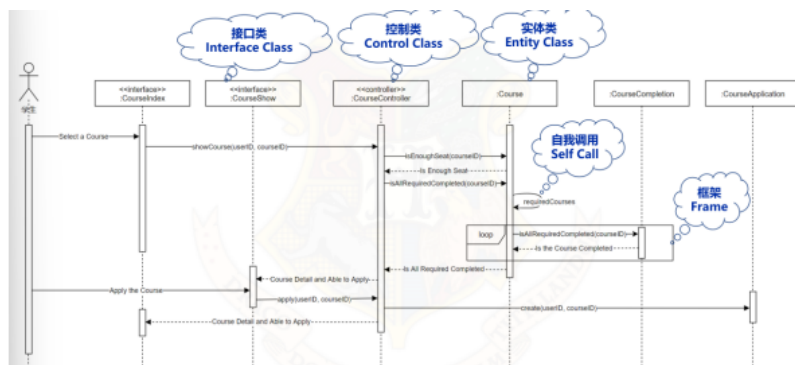
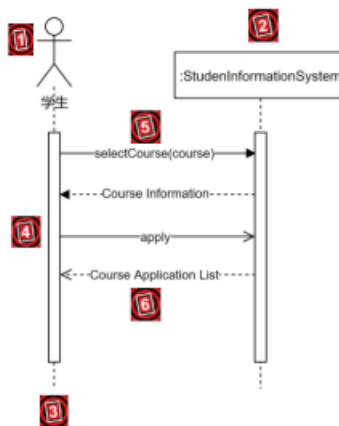
One

Zero or One

2. 请用时序图或交互图描述系统处理订单的过程

顺序图 Sequence Diagram

- ① 角色—用例的发起角色使用用例参与者符号表示
- ② 系统—盒子表示系统作为一个“黑盒子”或一个整体。冒号(:)用来表示系统的一个运行“实例”
- ③ 生命线—从角色和系统符号向下延伸的垂直虚线,表示生命顺序
- ④ 活动线—放置在生命线上的条形表示参与者进行交互活动的一段时间
- ⑤ 输入消息—从角色到系统的水平箭头表示消息输入
第一个单词字母小写,后续单词首字母大写,单词间无空格。
括号内包含了要传递的参数,逗号分隔每个参数。
- ⑥ 输出消息—从系统到角色虚线的水平箭头。
不需要使用标准的命名规范。(但是想用也可以用)



顺序图—描述了一个用例场景参与者和系统之间的消息交互,用来说明实现用例时调用的方法

3. 订单的状态图

状态图(Statechart Diagram)是描述一个实体基于事件反应的动态行为,显示了该实体如何根据当前所处的状态对不同的事件做出反应。通常我们创建一个UML状态图是为了以下的研究目的:研究类、角色、子系统、或组件的复杂行为。状态图用于显示状态机(它指定对象所在的状态序列)、使对象达到这些状态的事件和条件、以及达到这些状态时所发生的操作。

I 状态(State)

对象在生命周期中某一时刻的状态 (对象属性在某一个时刻的值)

I 状态转换事件(State transition event)

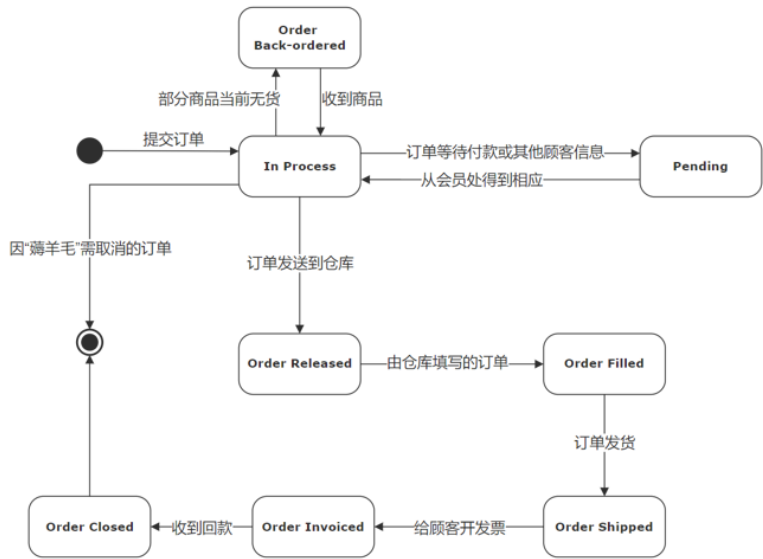
通过更新一个或多个属性的值来触发对象状态变化的事件

I 状态机图(State machine diagram)

描述了对象具有的不同状态,引起对象改变状态的事件,以及决定对象在状态间转换的规则。

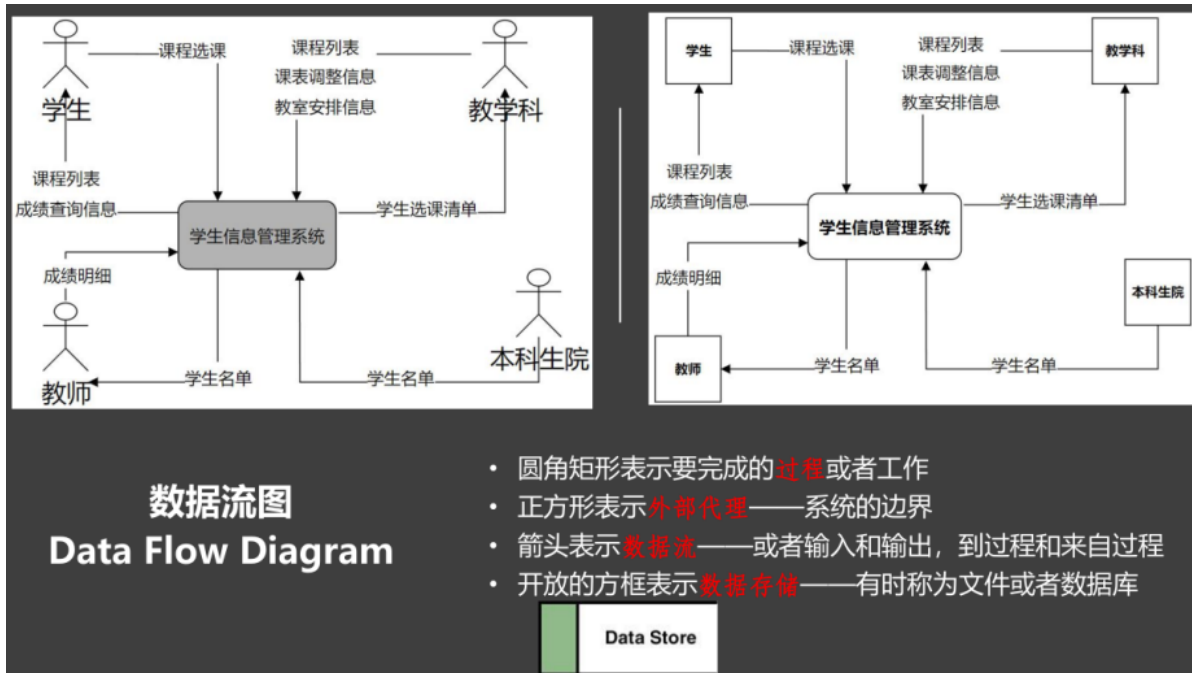
又称“状态图(state diagram)”或“状态转换图(state transition diagram)”

- 初始状态(实心圆)
- 在不同状态(圆角矩形)之间转换
- 最后进入结束状态(空心圆内的实心圆)
- 每个箭头表示触发订单对象从一个状态转换到另一个状态的事件



4.其他图

1.上下文数据流图

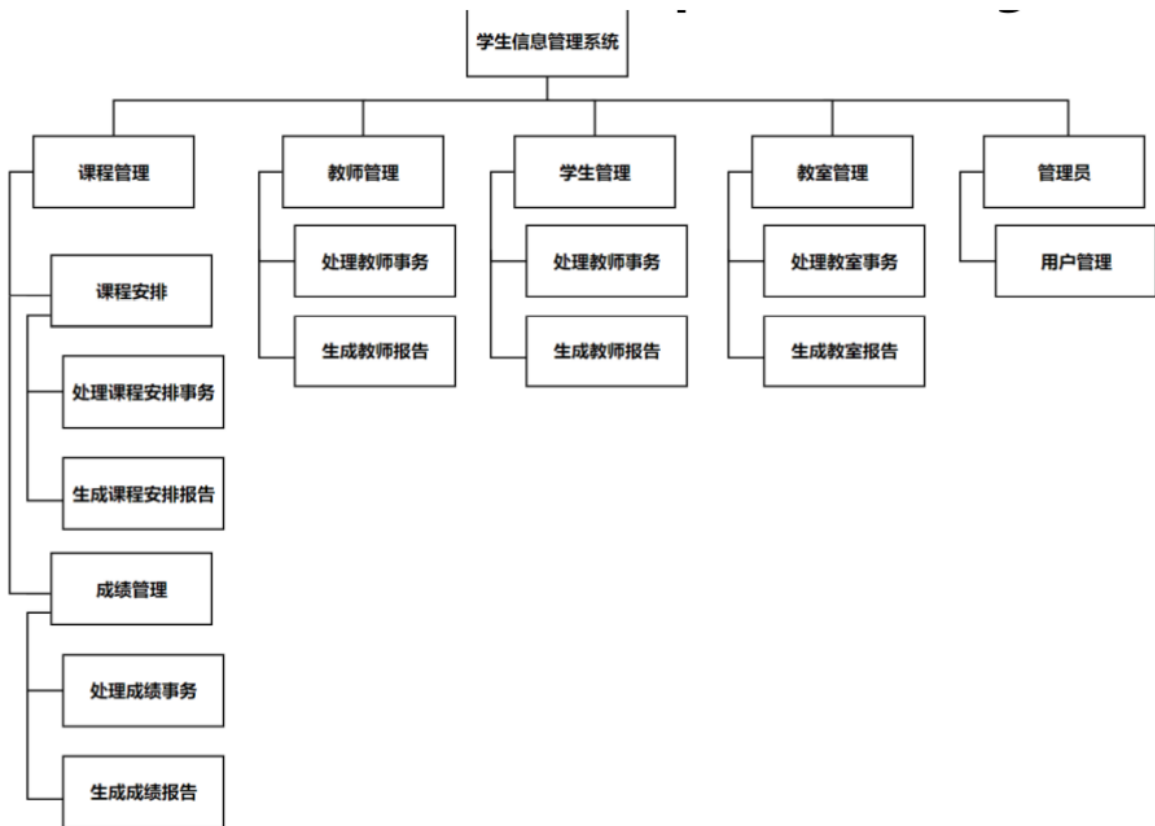


2.分解图

分解图 Decomposition Diagram

分解图，又称**层次图**，显示了一个系统自顶向下的功能分解和结构。

- 本质上是一种规划工具，用于更详细的过程模型
- 每个过程或者是**父过程**，或者是**子过程**，或者二者都是
- 父进程**必须**有两个或多个子过程
- 图中的连线不含箭头且没有命名



3.事件图

就是一个事件的上下文图

事件图是一个事件的上下文图，显示了事件的输入、输出和数据存储交互。

- ```

graph LR
 S1[学生] -- 已修课程 --> P[处理新课程申请]
 C[课程] -- 必修课程 --> P
 S2[学生] -- 新课程申请 --> P
 P -- 提交的课程申请 --> A[申请]
 A -- 课程申请确认 --> S2
 A -- 剩余席位 --> P

```

## 活动图 Activity Diagram

- 

- 
- ```

graph TD
    Start(( )) --> A(完成订单)
    A --> B[9]
    B --> C(发送订单确认)
    B --> D(释放结束的订单)
    C --> E[10]
    D --> E
    E --> End(( ))
  
```

Activity Diagram

