

# 计算机科学系实验报告

课程名称	<u>UML 与可视化建模</u>	班级	<u>2 班</u>		
实验名称	<u>淘淘商城</u>	指导教师	<u>曾少宁</u>		
姓名	<u>陈保锋</u>	学号	<u>1414080901221</u>	日期	<u>2016.3.3</u>

---

## 一、实验目的

掌握基于 UML 2.0 的建模概念与方法，掌握各种 UML 图的概念与画法，其中包括用例图、活动图、类图、顺序图、组件图和状态图等。

## 二、实验设备与环境

操作系统：Windows 10；建模工具：StarUML。

## 四、实验要求

1. 实验及实验报告以增量方式完成，每次作业都在上一次作业的基础上完成，作业提交网站不提供报告下载，所以请同学们自行保管好自己的实验报告；
2. 请将实验报告中“占位符”信息替换为自己的实验相关信息；
3. 请认真撰写实验体会，**实验课结束时**立即上传实验报告：<http://zeng.shaoning.net/uml/>。

## 四、实验内容、程序清单及运行结果

淘淘商城

角色:用户,管理员

- 1、单点登录
- 2、下单
- 3、加入购物车

#### 4、实验一：需求建模 - 用例模型

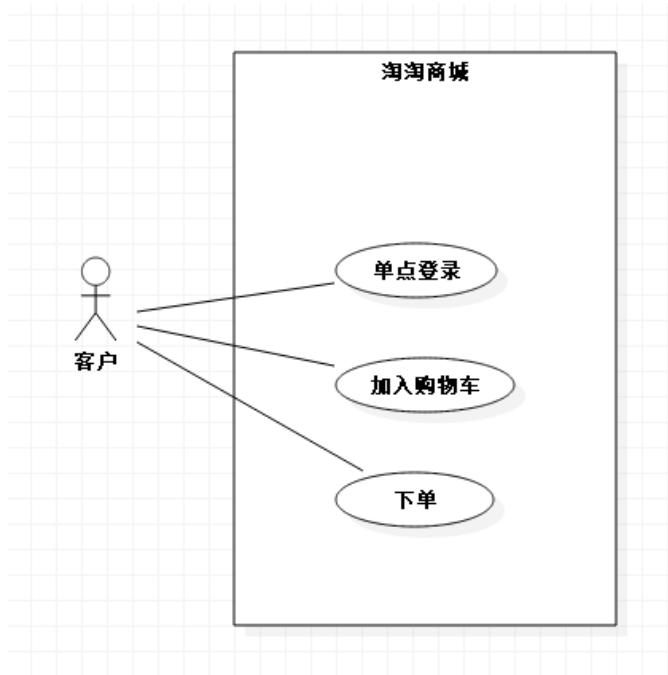


图 1：淘淘商城用例图

注：用例规约内容及项目可自行增加。

用例编号：	UC001
用例名称：	单点登录
用例描述：	分布式环境中,根据页面并发量的不同同一个系统将拆分为不同的模块部署到不同的服务器中,单点登录实现用户的 session 共享.
前置条件：	Redis 缓存
基本流程：	<ol style="list-style-type: none"> <li>1.用户访问登录页面</li> <li>2.后端拦截器拦截用户登录 URL, 从 Cookie 中获取 TOKEN_KEY 信息</li> <li>3.后端利用获取的 TOKEN_KEY 查询 Redis 服务器, 获取到用户信息</li> <li>4.向 RabbitMQ 的交换机发送用户登录的消息</li> <li>5.登录成功, 重定向到主页</li> </ol>
扩展流程：	<ol style="list-style-type: none"> <li>2.1 后端没有获取到 TOKEN_KEY 信息, 转发到登录页面</li> <li>2.2 用户填写账号密码, 提交 POST 表单</li> <li>2.3 后端查询用户表, 匹配账号密码</li> <li>2.4 后端生成 TOKEN_KEY,保存该用户信息到 Redis 服务器,向客户带有 TOKEN_KEY 信息的一级域名的 Cookie</li> </ol>

	<p>2.5 向 RabbitMQ 的交换机发送用户登录的消息</p> <p>2.6 用户登录成功,页面重定向到主页</p> <p>3.1 没有命中 TOKEN_KEY, 用户信息可能过期, 后端转发到登录页面</p> <p>2.3.1 账号密码匹配失败, 转发到登录页面, 做出“账号或密码错误”提示</p>
后置条件:	其他二级域名对应的系统在访问 TOKEN_KEY 信息的时候需要提供相应的服务程序给其他系统查询.

用例编号:	UC004
用例名称:	加入购物车
用例描述:	用户在离线或者登陆状态都能将商品加入购物车, 需要维护登陆与 cookie 缓存的购物车信息。
前置条件:	购物车系统消息接收队列需要注册接收用户登录成功的 RabbitMQ 交换机
基本流程:	<ol style="list-style-type: none"> <li>1、用户在商品详情页点击加入购物车,校验表单数据</li> <li>2、购物车系统接收到商品 id, 对所有 id 信息进行校验, 重新从商品表查询所有商品信息</li> <li>3、购物车系统查询单点登录系统的用户登录信息服务, 返回用户登录信息</li> <li>4、购物车系统根据商品信息和用户 id 将商品信息添加到购物车表</li> <li>5、购物车系统添加商品成功, 向客户端反馈成功添加信息</li> </ol>
扩展流程:	1.1 用户没有选择商品信息, 作出提示

	<p>2.1 没有查询到商品信息，用户请求参数不可靠，向用户返回错误信息</p> <p>3.1 购物车系统没有从单点登录系统中查询到用户登录信息，用户处于离线状态</p> <p>3.2 查询 cookie 中的 cart_token 信息</p> <p>3.3 根据 cart_token 信息查询 Redis 服务器，获取购物车信息</p> <p>3.4 将返回的购物车信息进行反序列化，将用户添加的商品加入到其中</p> <p>3.5 重新将所有购物车信息序列化并写入到 Redis 服务器</p> <p>3.6 购物车系统添加商品成功，返回添加成功信息给客户端</p> <p>3.2.1 没有查询到 cart_token 信息，生成一个 cart_token 唯一码</p> <p>3.2.2 将所有商品信息进行序列化，将 cart_token 和商品信息写入到 Redis 服务器</p> <p>3.2.3 生成 cart_token 的 cookie 信息，设置到响应头中</p> <p>3.2.4 购物车系统添加商品到购物车成功，返回添加成功信息</p> <p>3.3.1 用户有 cart_token 的 cookie 信息但是没有从 Redis 服务器中查询到商品信息，该 cookie 信息非法，将 cart_token 为 null 的 cookie 信息写入到响应头，即删除该 cookie 信息，重走 3.2.1 流程</p>
后置条件:	<p>1.购物车系统从消息队列中接收到用户登录成功消息</p> <p>2.从消息中获取所有 cookie 信息，获取 cart_token 信息</p> <p>3.根据 cart_token 信息查询 Redis 服务器</p> <p>4.反序列化所有商品信息</p> <p>5.将商品信息同步到数据库</p> <p>2.1 没有获取到 cart_token 信息，直接结束，不做任何操作</p> <p>3.1 没有查询到商品信息，直接结束，不做任何操作</p>

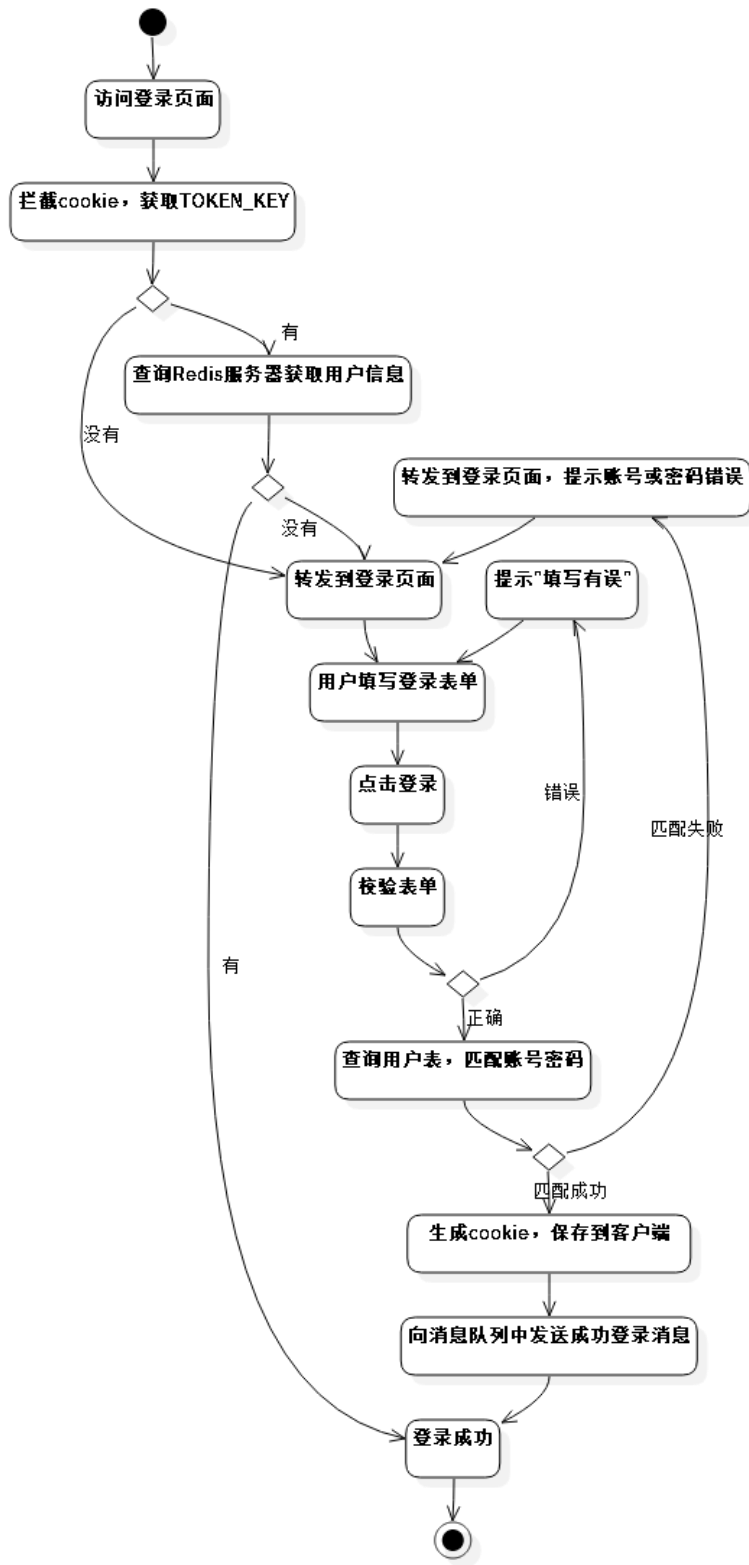
用例编号:	UC003
用例名称:	下单
用例描述:	下单

前置条件:	下单并利用第三方支付
基本流程:	<ol style="list-style-type: none"> <li>1.用户勾选需要下单产品点击下单，校验表单数据</li> <li>2.后端根据商品信息查询商品数据库，整合商品信息，计算商品总价，转向确认订单页面</li> <li>3.用户点击提交订单，校验用户是否选择收货地址</li> <li>4.后端接收订单信息，重新计算商品总价，保存订单信息到订单表，重定向到第三方支付</li> <li>5.客户支付完毕，第三方支付回调支付成功服务</li> <li>6.后端接收到支付成功信息，修改订单状态，页面重定向到支付完成页面</li> </ol>
扩展流程:	<ol style="list-style-type: none"> <li>1.1 用户没有选择商品，做出提示</li> <li>2.1 没有查询到商品信息，向客户端反馈“数据异常”</li> <li>3.1 用户没有选择收获地址，做出提示</li> <li>4.1 用户 30 分钟内没有支付,订单修改为过期状态</li> </ol>
后置条件:	

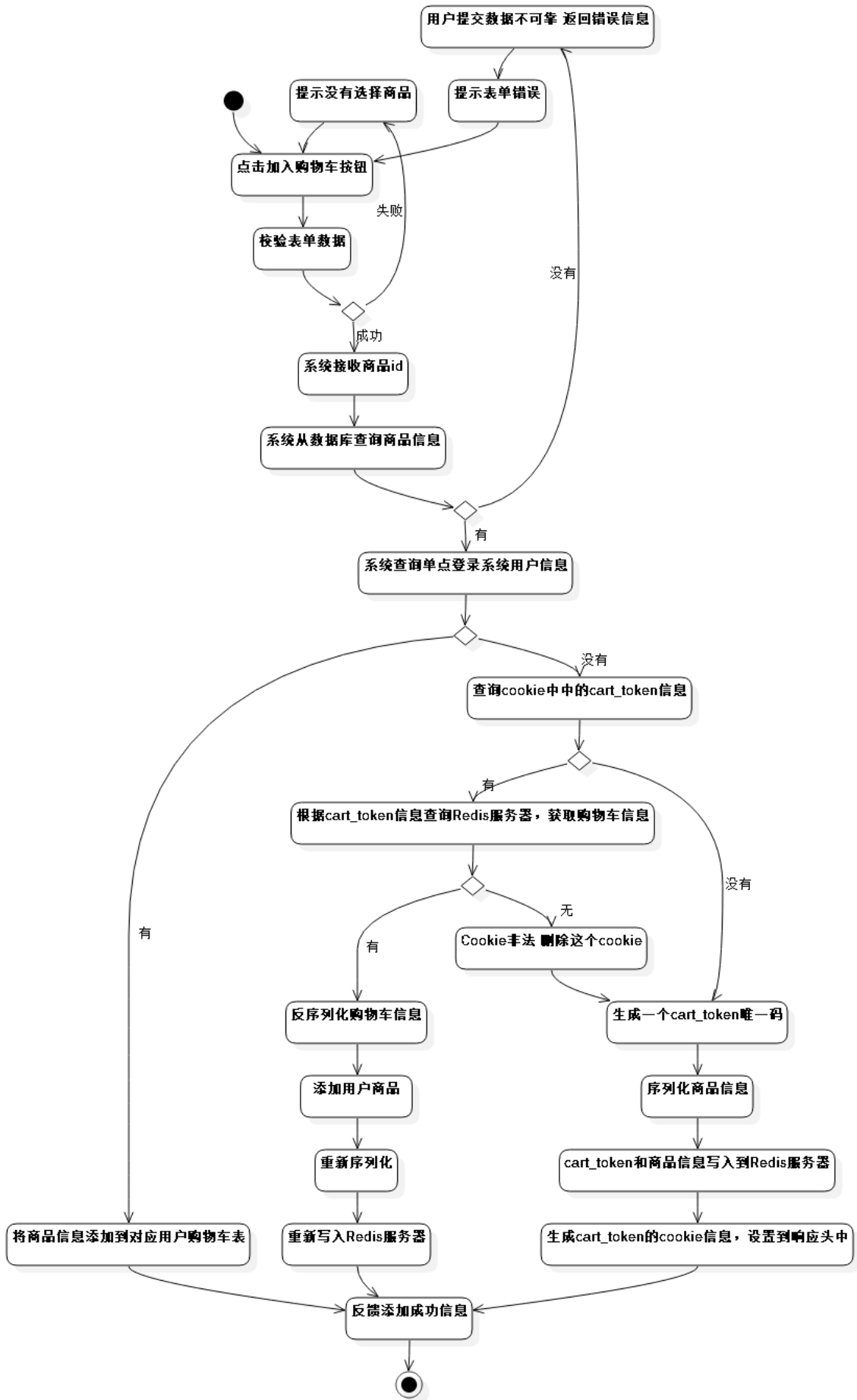
## 5、实验二：过程建模 – 活动模型

使用活动图描述系统的业务过程。

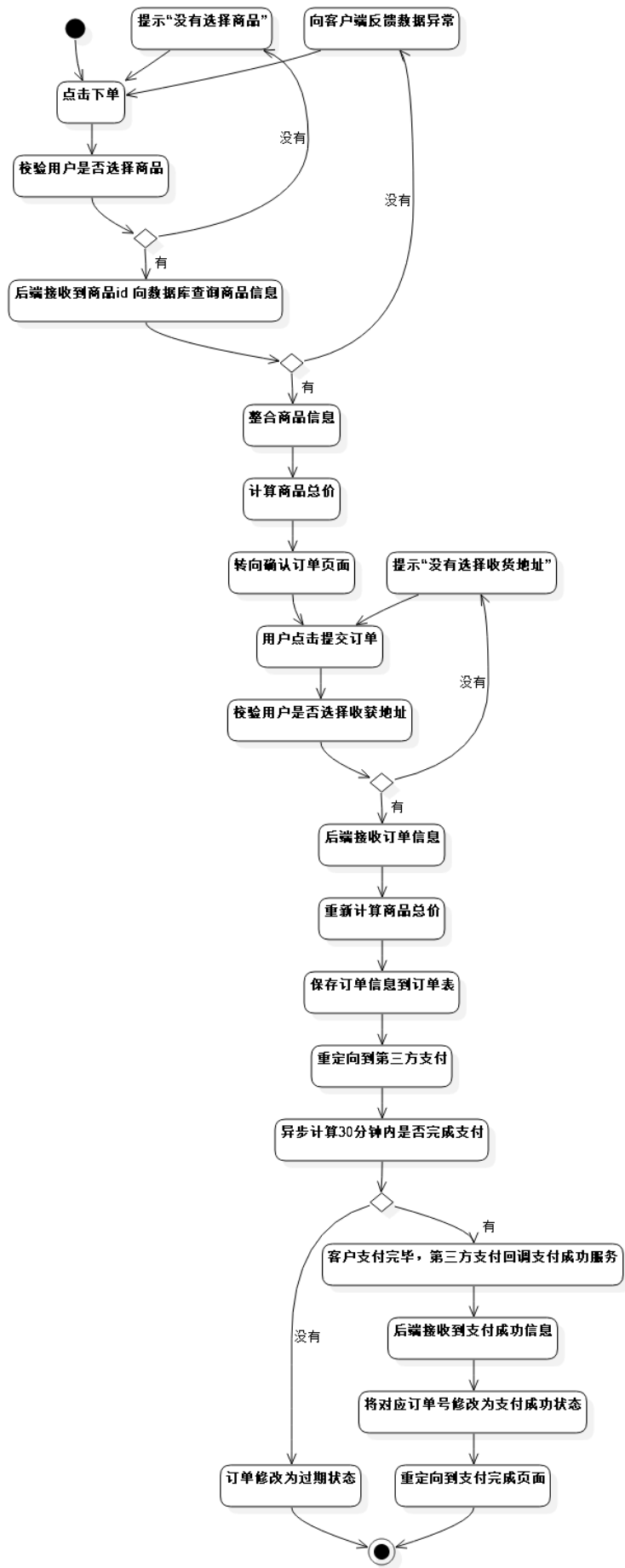
方法：将用例规约中的基本流程与扩展流程抽象为过程步骤（Action），画出对应的活动图。



单点登录活动图



加入购物车活动图



下订单活动图

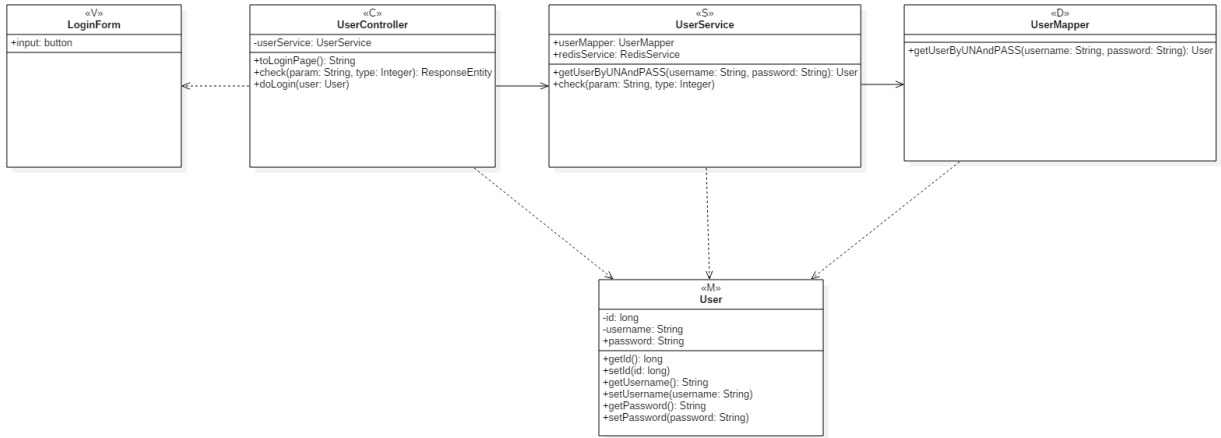


## 6、实验三：逻辑建模 – 类模型

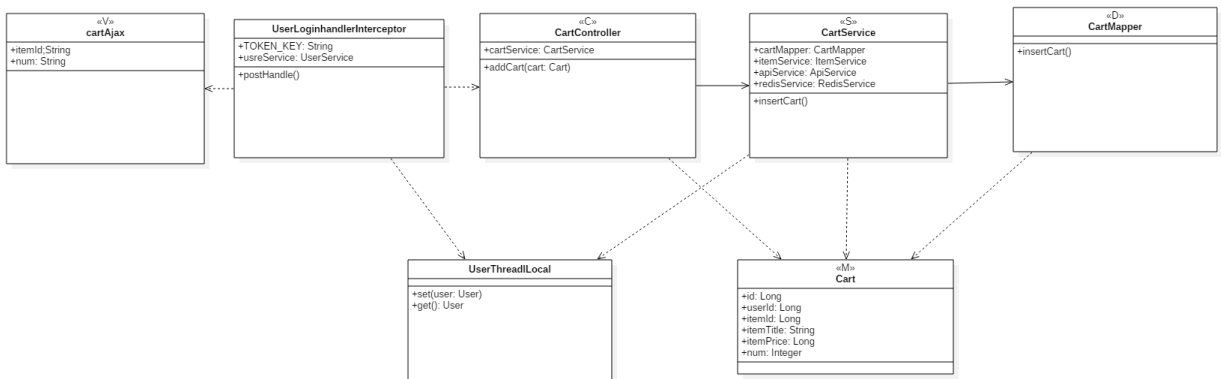
基于 MVC 设计模式找出实现用例的类。

方法：分别找出实现用例的模型（Model）、视图（View）和控制器（Controller）类，确定类之间的关系及其关键属性，画出类图。

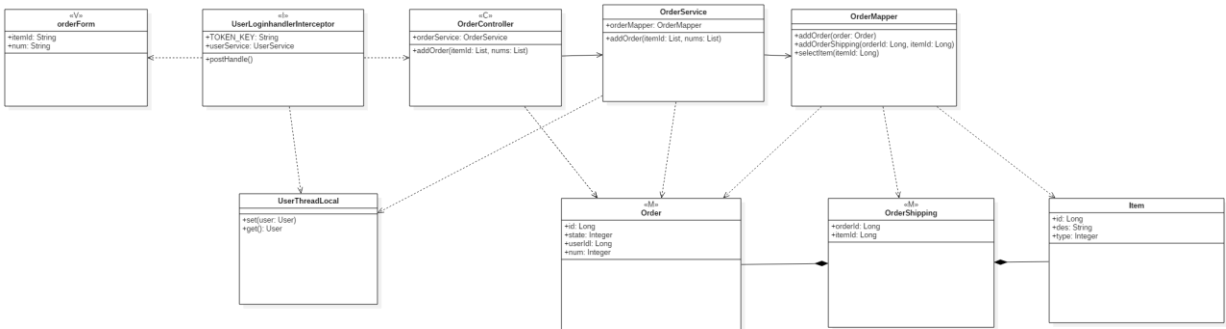
参考：讲义 P26 页。



单点登录类图



加入购物车类图



下单

## 7、实验四：交互建模 – 顺序模型

创建各个类（MVC 及 Actor）的对象，并描述对象之间的交互。

方法：分别创建参与者（Actor）、界面类（View）、控制器类（Controller）和模型类（Model）的对象，描述各个对象之间的消息及其顺序，画出顺序图。

参考：讲义 P33 页 8.7.2。

## 8、实验五：状态建模 – 状态模型

对系统中最重要的对象进行状态建模。

方法：选择一种对象，定义该对象的状态，描述状态之间的切换及条件，画出状态图。

参考：讲义 P9 和 P10 页。

# 五、实验体会

实验一：

实验二：

实验三：

实验四：

实验五：