



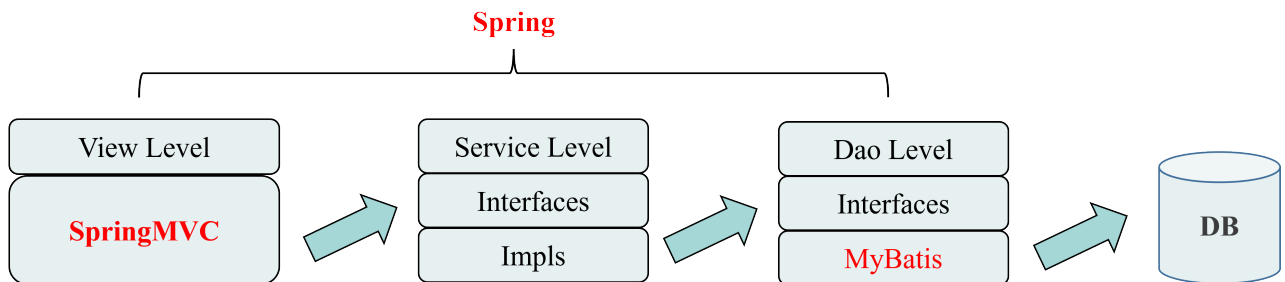
系统设计

1.1 SSM框架

SSM框架是Spring、SpringMVC、MyBatis的简称。

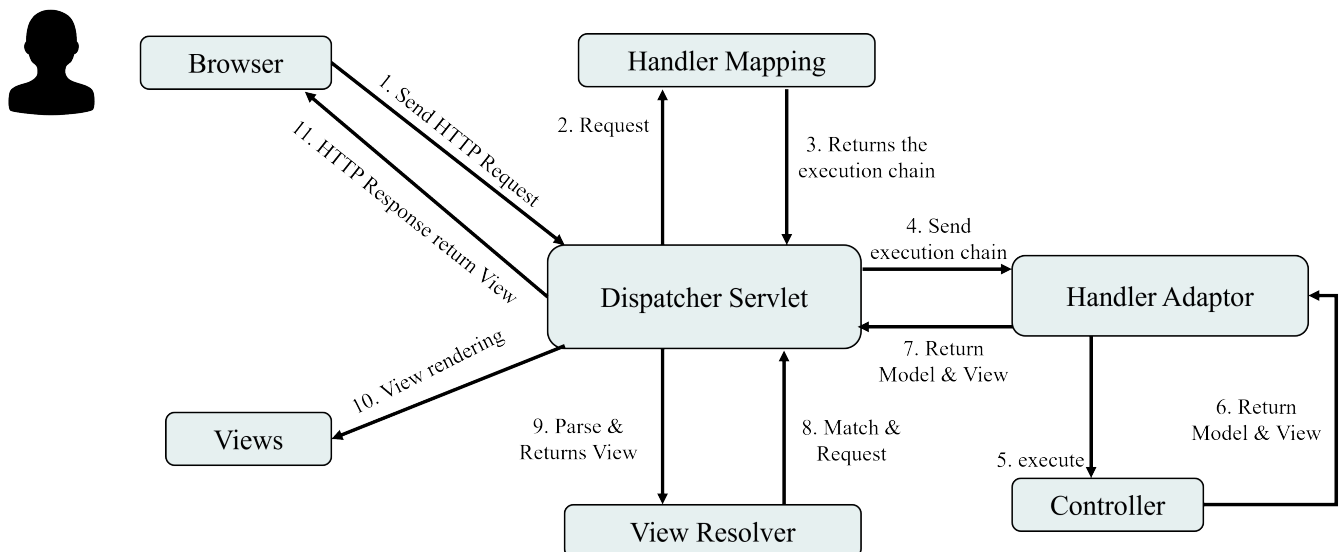
它的架构设计如下图：

SSM Framework



如图所示，整个系统采用SSM架构，SpringMVC作为View的实现，它的Controller层（Handler层）作为整个项目的控制器，完成用户的请求响应。SpringMVC的主要工作流程如下图所示：

Spring MVC Framework

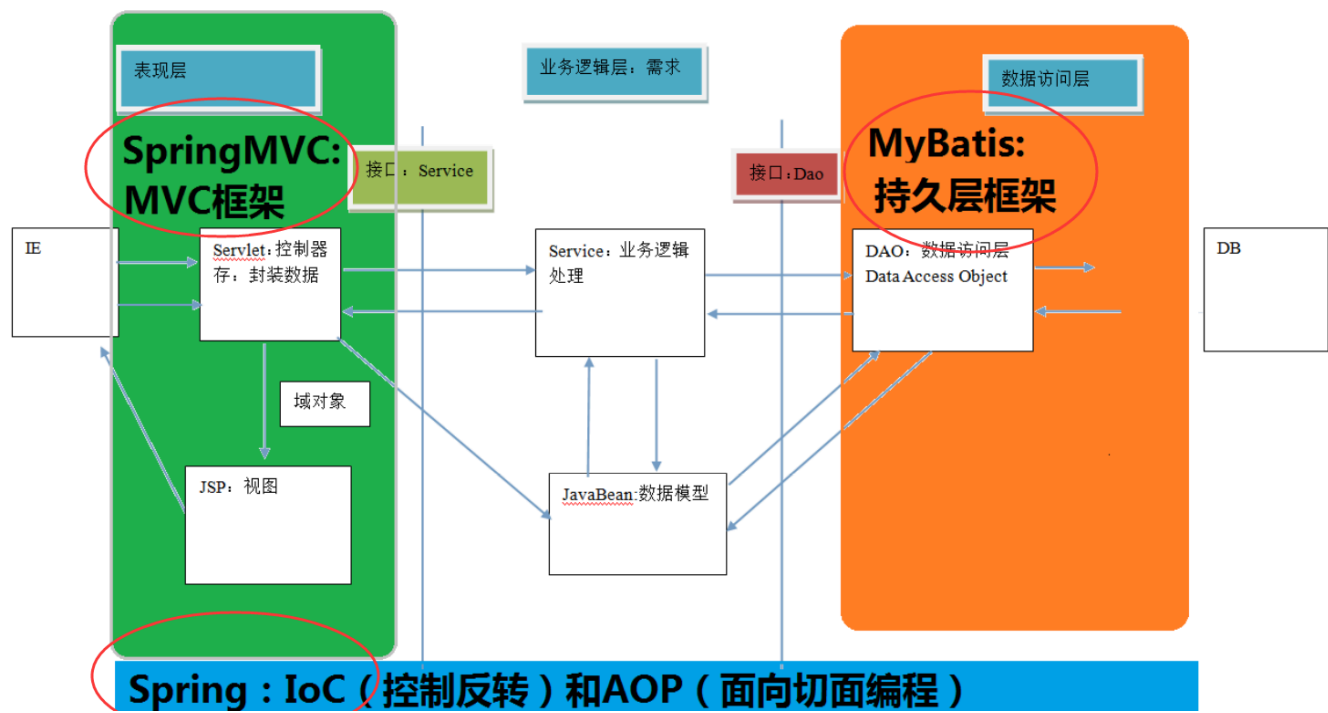


Service层主要负责业务模块的逻辑应用设计。首先设计接口，再设计其实现的类接着再在Spring的配置文件中配置其实现的关联。这样我们就可以在应用中调用Service接口来进行业务处理。

MyBatis作为Dao层的实现，完成了数据库的增删改查操作。DAO层，Service层这两个层次都可以单独开发，互相的耦合度很低，完全可以独立进行，这样的一种模式在开发大项目的过程中尤其有优势。Controller，View层因为耦合度比较高，因而要结合在一起开发，但是也可以看作一个整体独立于前两个层进行开发。

Spring 作为整个应用的管理者，整个系统的Bean都由Spring来管理，SpringMVC和MyBatis都是Spring的子容器。

下图说明各个容器的关系：



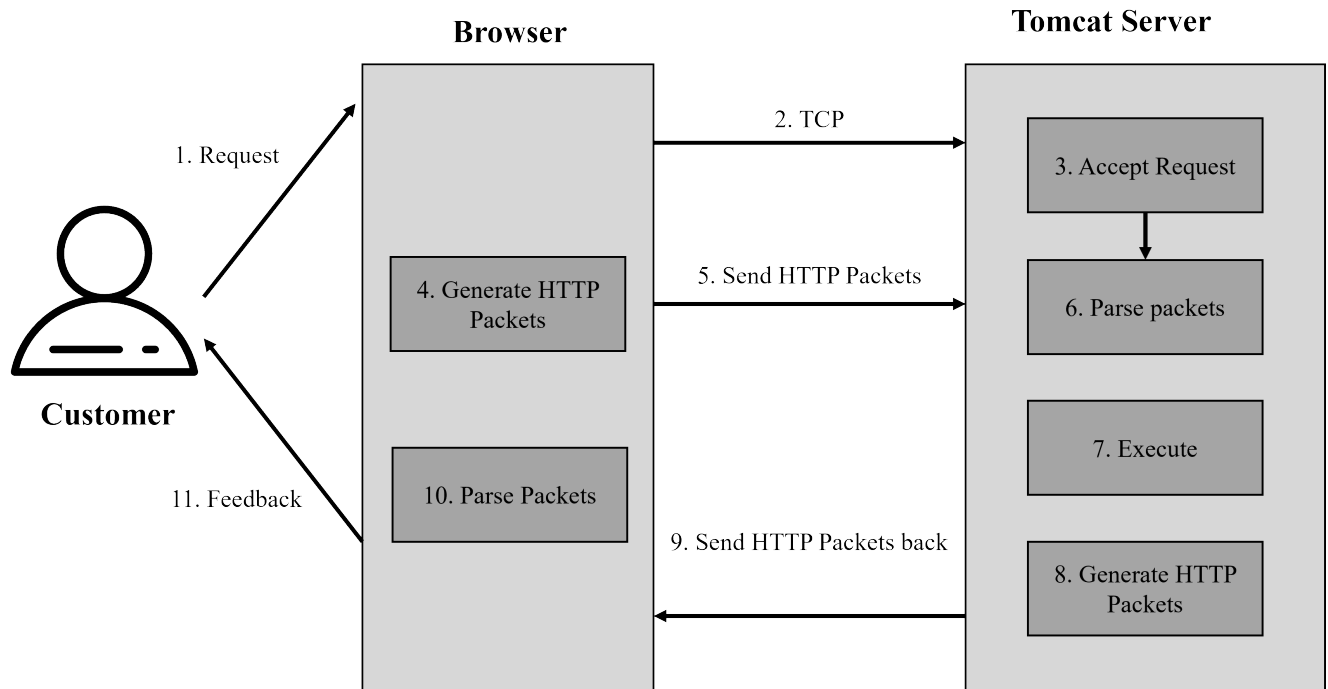
本项目使用Navicat for MySQL 作为数据库管理工具，使用Maven作为项目管理工具，使用Tomcat作为服务器，使用IntelliJ IDEA作为开发工具。

1.2 Tomcat

Tomcat是一个免费的开放源代码的Web应用服务器，属于轻量级应用服务器。

Tomcat的主要功能是执行Servlet和JSP，它是一个Servlet容器，它可以执行Java Servlet和JavaServer Pages (JSP) 程序，提供了一个“容器”来运行这些程序。它的工作流程如下图所示：

Tomcat Framework



1.3 系统功能设计

1.3.1 管理员功能

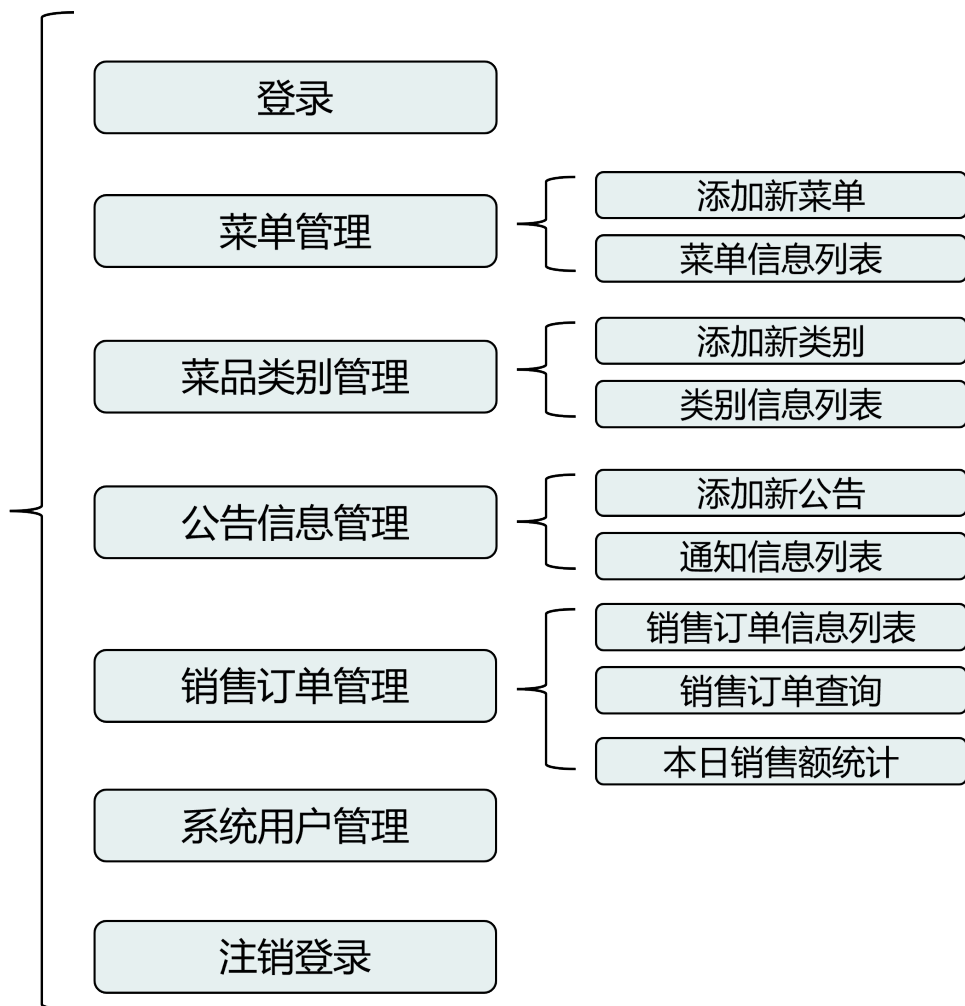
管理员需要进入管理员后台登录地址：<http://localhost:8080/admin/> 进行登录，可申请5天内免登录。

登陆成功后，管理员可对应的功能如下：

管理员的功能



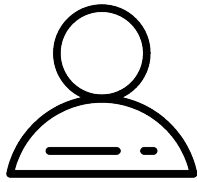
管理员



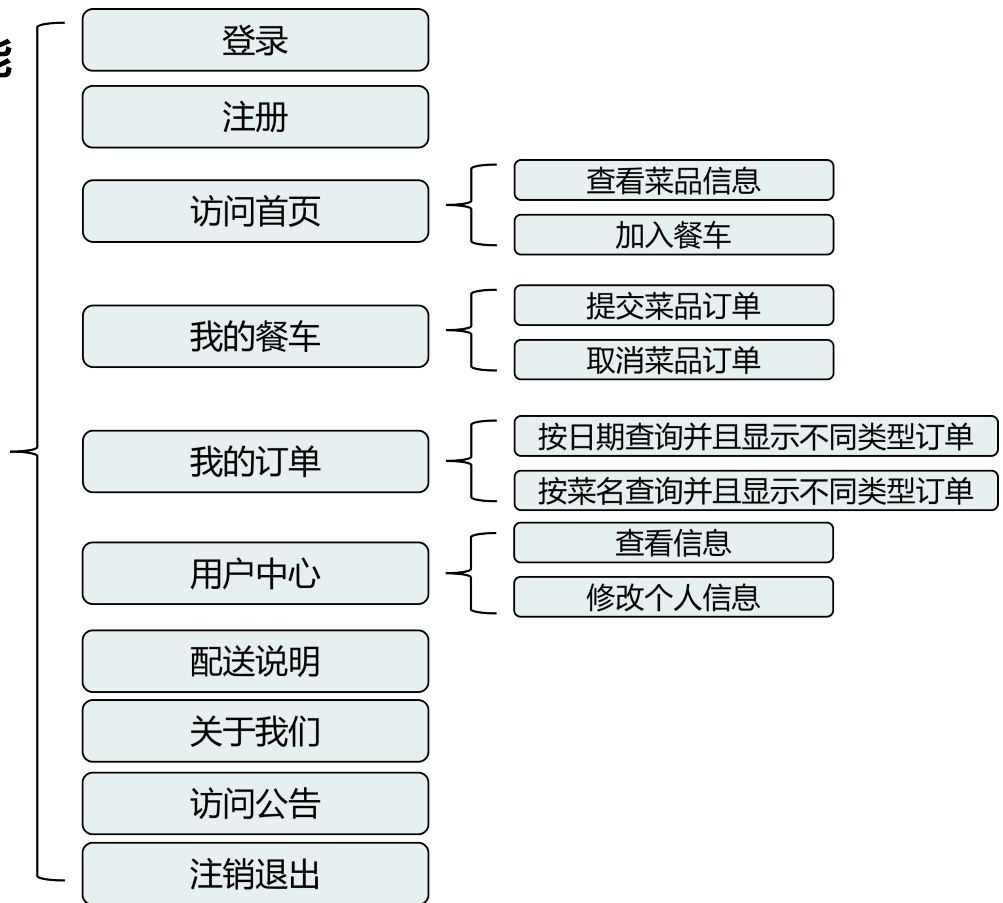
1.3.2 用户功能

项目启动后，普通已注册的用户可以正常访问点菜系统，应具有如下功能：

注册用户的功能



注册用户



1.3.2 菜品展示

数据库初始化了6个菜品，应展示菜品菜名，市场价格与会员价格，菜品图片，菜品的配料与类型，并且提供给用户加入餐车的功能。

1.3.3 公告功能

管理员应该具有修改公告的权限，公告应该在首页展示，用户可以查看公告。

1.3.4 餐车功能

餐车应该展示在菜品展示，用户可以在菜品展示页面对餐车进行操作，包括增加菜品，删除菜品，清空餐车。并且实时显示当前餐车菜品的总价。

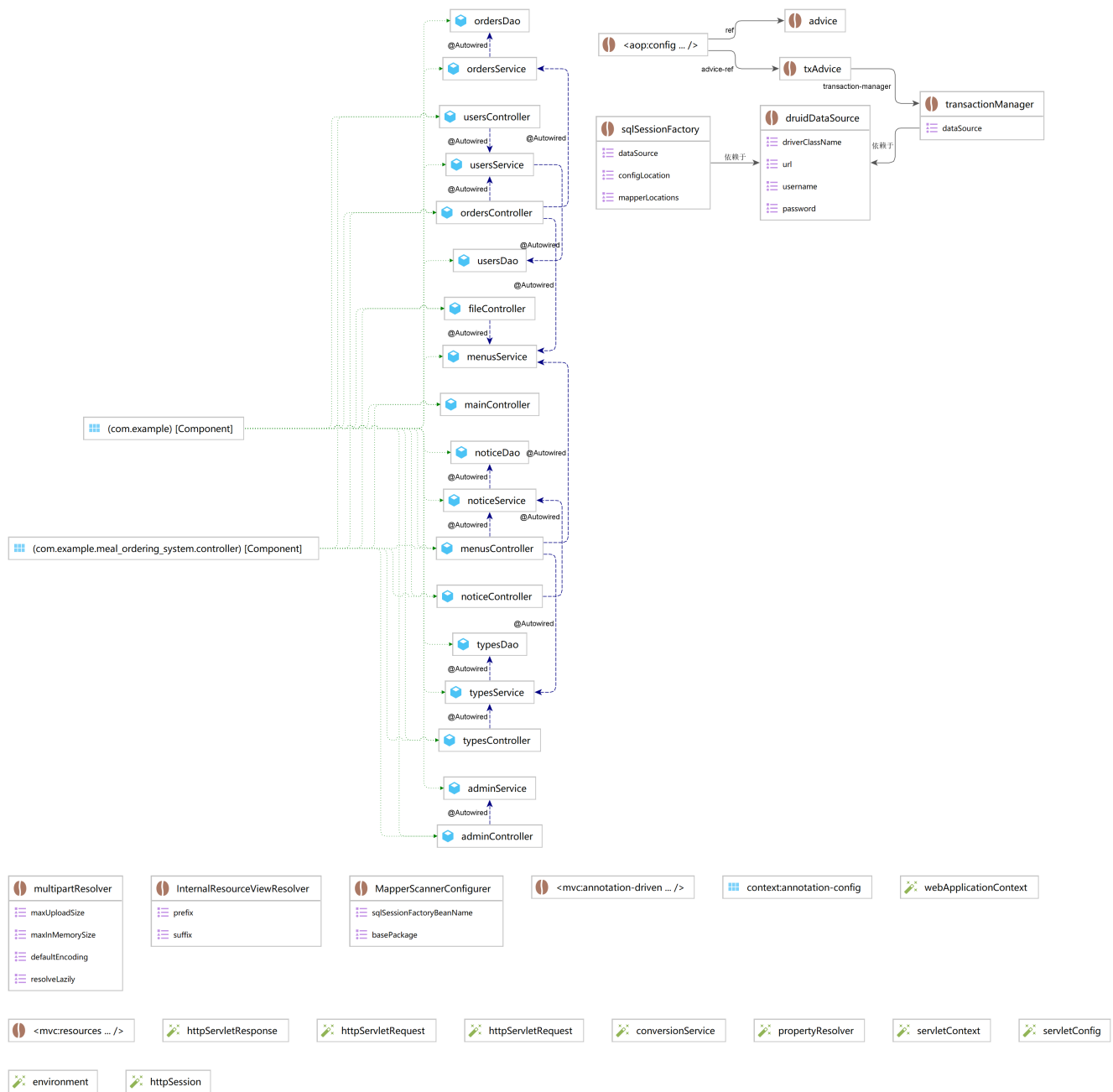
1.3.5 销售排行榜功能

在首页显示销售排行榜，以菜品名按照销售数量进行排序，并且展示菜品销售次数。

1.3.6 依赖关系与UML类图

系统主要设计了：Admin、Menus、Notice、Orders、Page、Pages、ShoppingCart、Types、User等实体对象，对应的Controller方法有：AdminController、MenusController、NoticeController、OrdersController、FileController、MainController、TypesController、UserController，

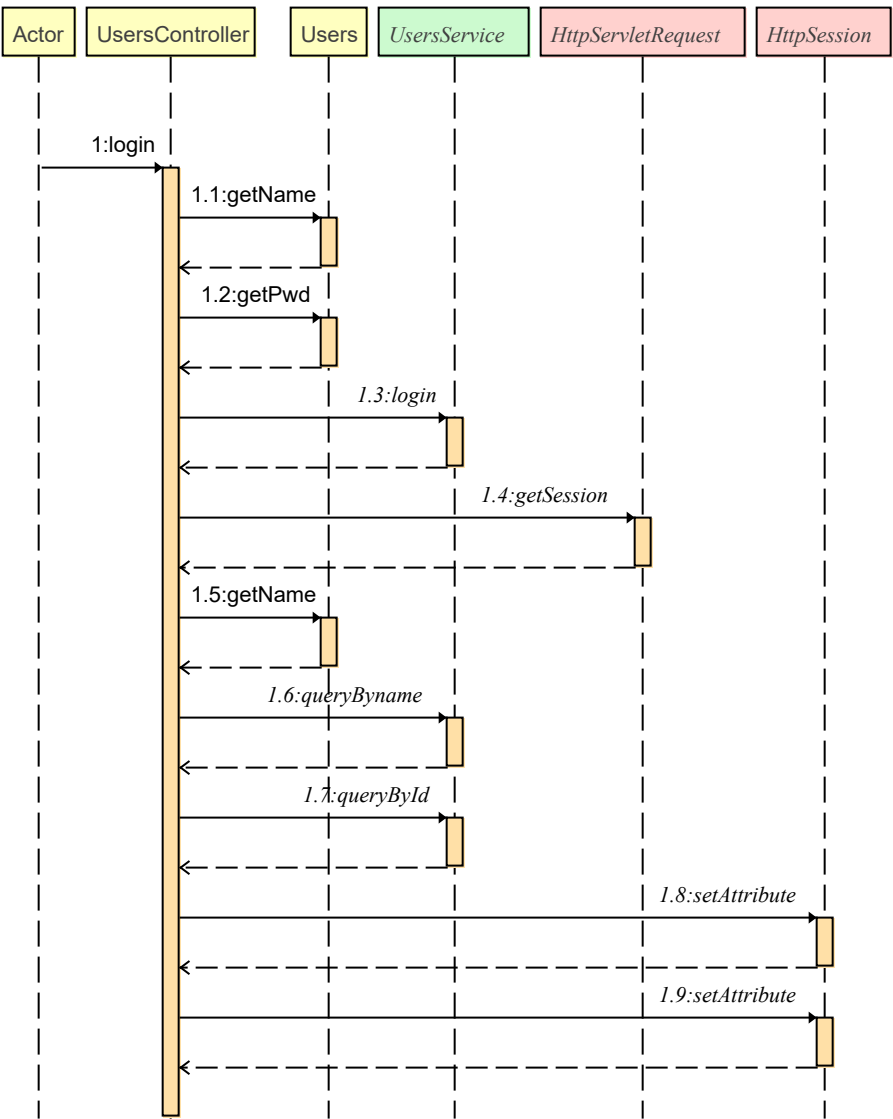
UML类图与依赖关系如下：



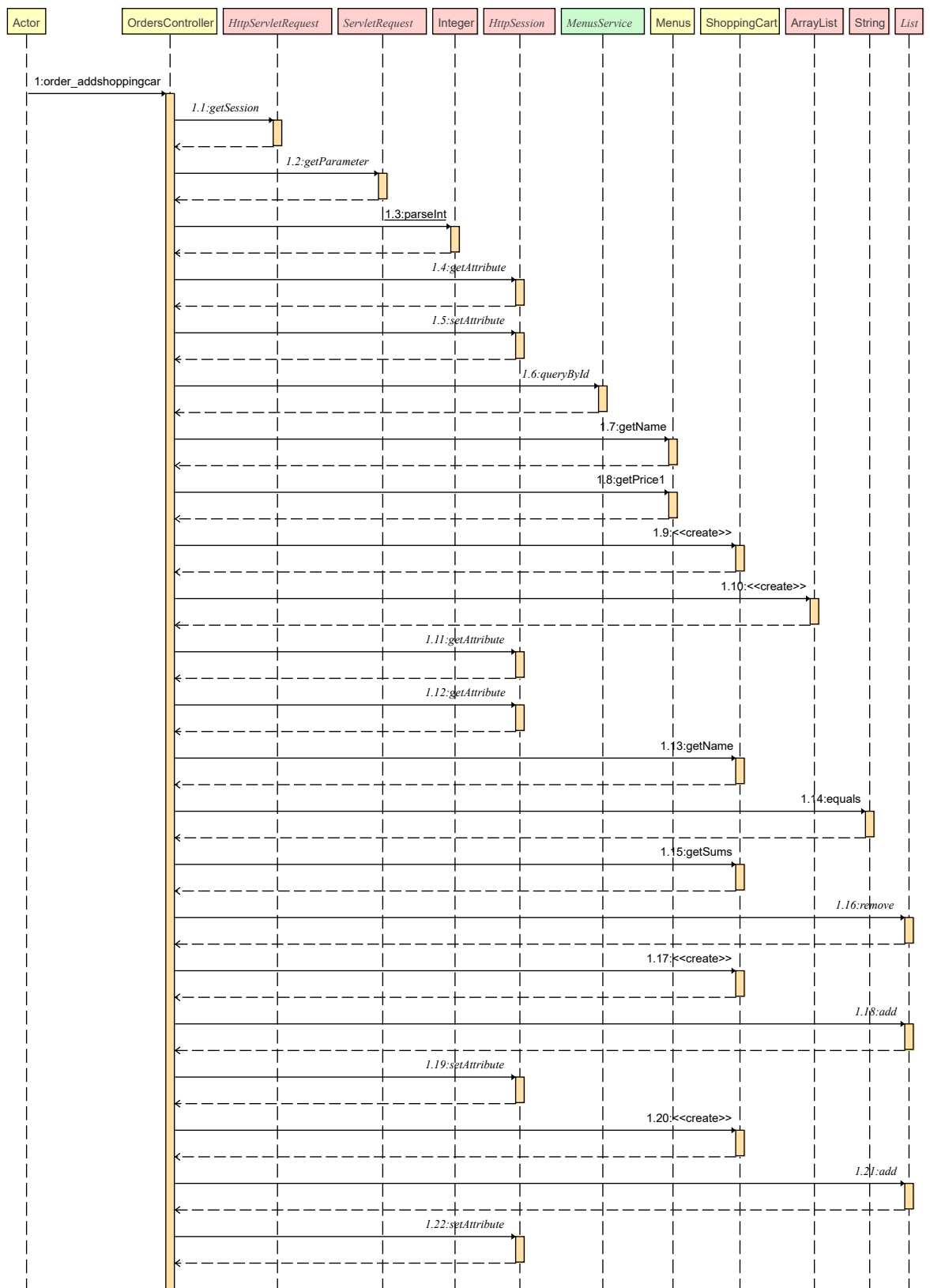
1.3.7 UML时序图

系统每个阶段都可以使用UML时序图来解读，本文只对其中一部分流程进行解读，其他流程类似。

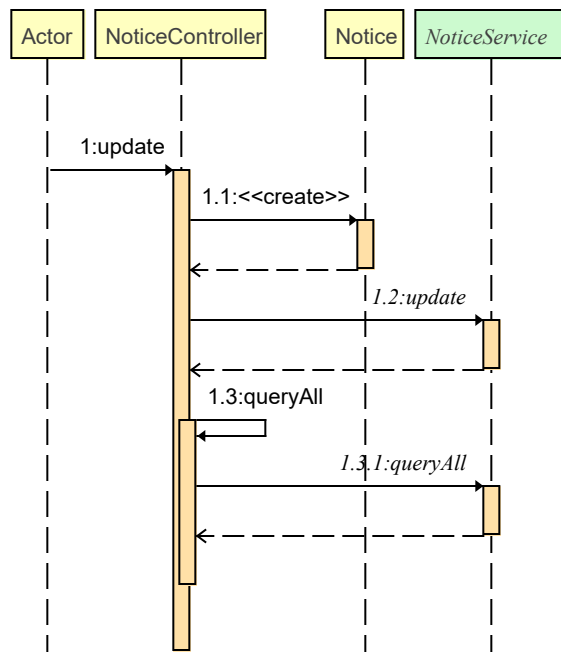
- 1. 用户登录的时序图，行为触发login方法，在UserController中的login方法中，使用getName与getPwd方法判断用户名与密码是否存在，如果存在，则进行HTTP请求，得到Session，再进行查询与访问。



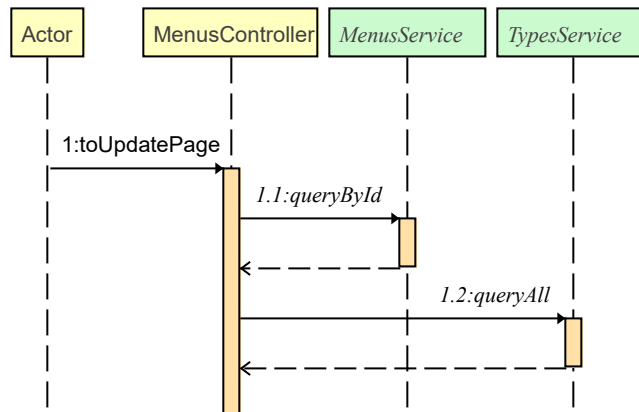
- 2. 点单添加到购物车，行为触发OrdersController，该组件控制整个流程，getSession请求HTTP获取Session，在对Service层进行调用，得到Params，再进行查询与访问。再在MenusService层依据id查询菜品Name与Price，在ShoppingCartService层进行添加购物车操作。并且重复该请求操作得到SumPrice，最后返回给前端。



- 3. 更新公告，行为触发update方法，再查询公告信息，再返回给前端。



- 4. 更新菜单页面，行为触发MenusController，再依据id查询菜品信息，再返回给前端。



1.4 数据模型设计

系统采用MySQL数据库，使用Navcat for MySQL进行数据库管理。

1.4.1 数据库设计

数据库包含admin.....等数据模型，

以下是数据库数据结构设计展示：

1.4.2 接口设计

本系统主要设计了xx，xx，xx接口，分别对应xx，xx，xx功能。

详细设计如下表：