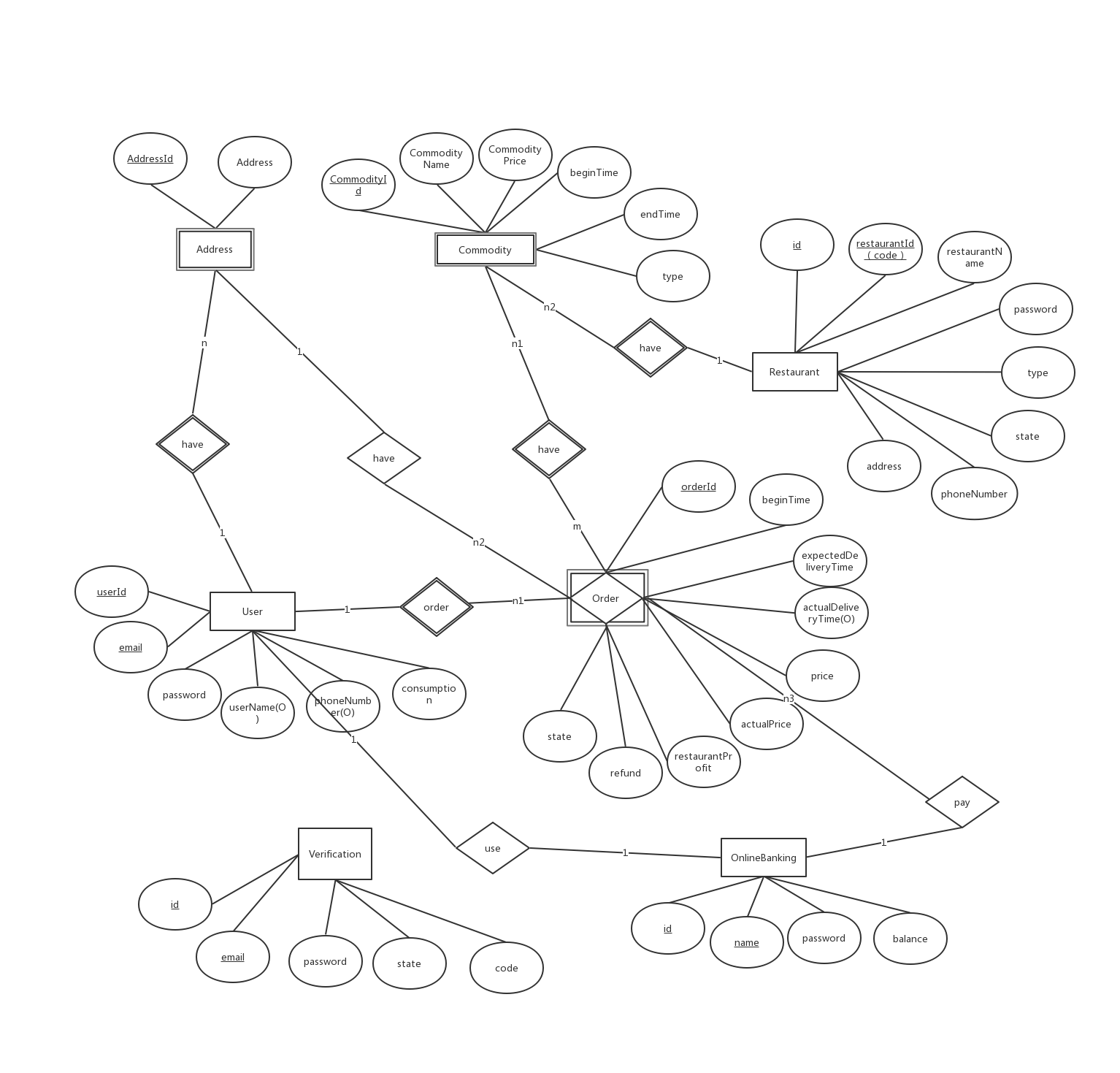
# Yummy说明文档

**161250041 侯韵晗**

一、数据库设计

### 1、ER图



2、数据库表

**Table:Address**

Columns:

addressId int

address varchar(30)

userId int

position int

ForignKey:

userId reference User userId

**Table:CommodityInfo**

Columns:  
 commodityId int

restaurantId int

commodityName varchar(20)

commodityPrice double

commodityNum int

beginTime timestamp

endTime timestamp

type varchar(10)

**Table:OnlineBanking**

Columns:

id int

name varchar(20)

password varchar(20)

balance double

**Table:Order**

Columns:

orderId int

email varchar(30)

commodities varchar(100)

addressId int

beginTime timestamp

expectedDeliveryTime timestamp

actualDeliveryTime timestamp

price double

actualPrice double

state int

refund double

restaurantProfit double

restaurantId double

**Table:Restaurant**

Columns:

id int

restaurantId varchar(7)

restaurantName varchar(15)

password varchar(20)

phoneNumber varchar(20)

address varchar(30)

type varchar(10)

state int

**Table:User**

Columns:

userId int

email varchar(30)

password varchar(20)

userName varchar(15)

phoneNumber varchar(20)

consumption double

**Table:Verification**

Columns:

id int

email varchar(30)

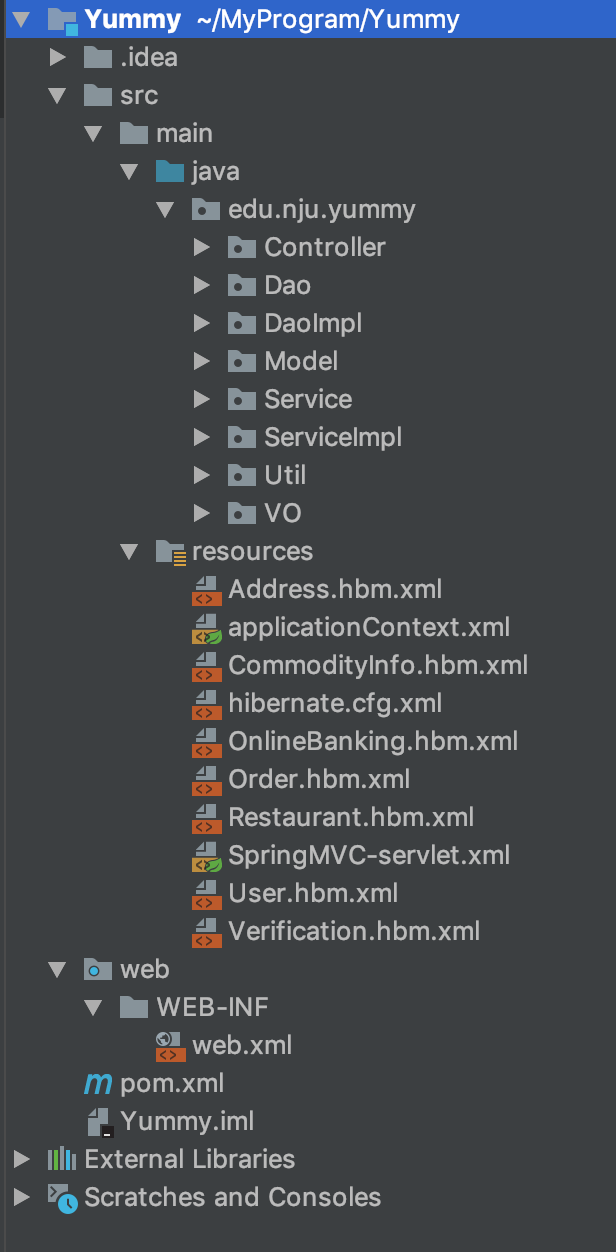
password varchar(20)

state int

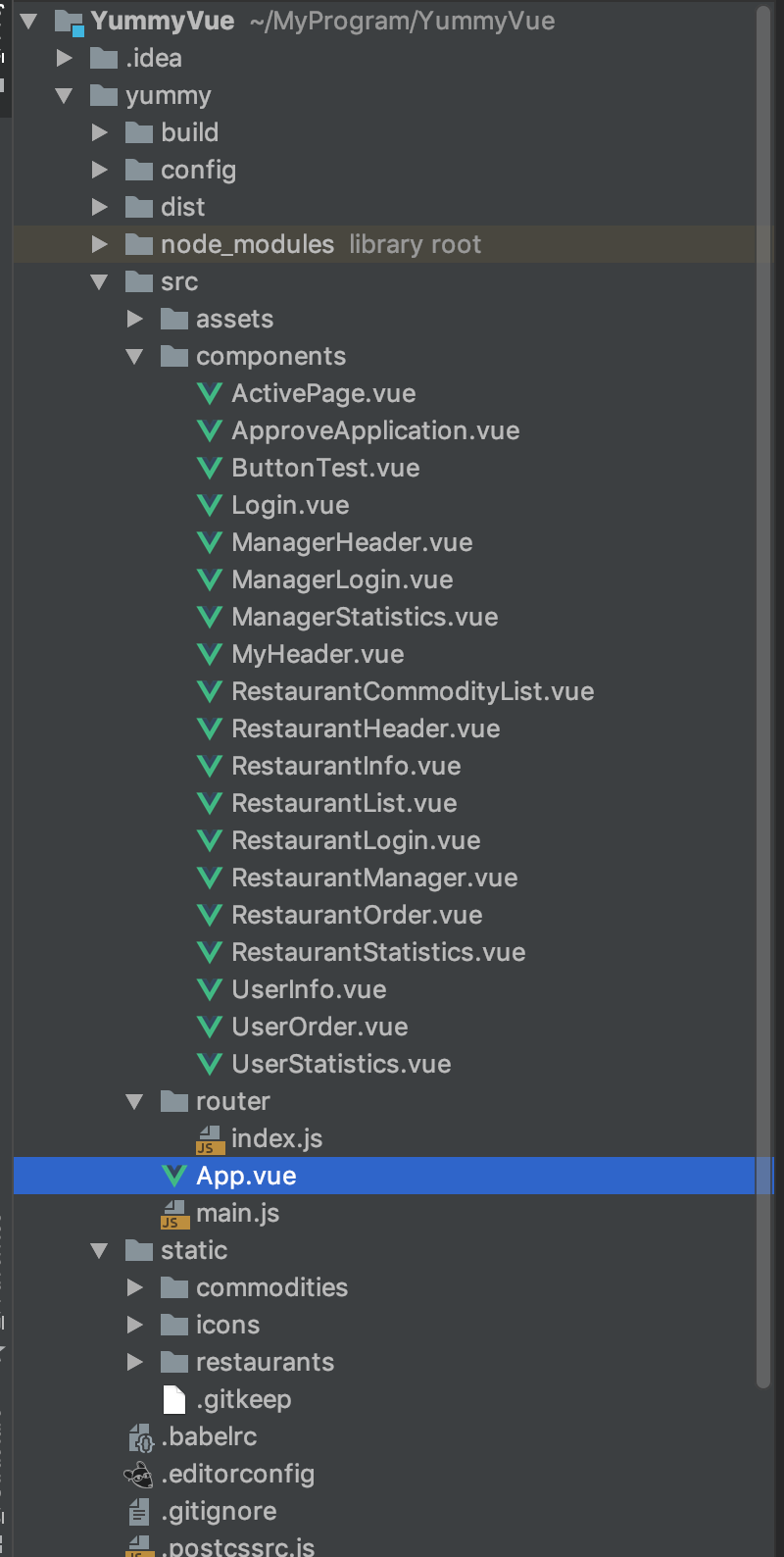
code varchar(4)

### 架构设计1、工程的项目结构截图

后端：



前端：



### 后端框架

SpringMVC+Hibernate

### 前端框架

Vue （使用组件库：iview）

## 三、类设计

### 1、各包的类：名称及职责

**Controller：**

ApproveApplication：经理审批商家信息（当餐厅刚注册和修改信息时）

Restaurantlnfo：商家信息修改

RestaurantManager：商家发布商品/套餐

RestaurantOrder：获取商家的订单情况

RestaurantSignin：商家登陆

Signin：用户登录

RestaurantList：用户获取商家列表/商家商品列表

TakeOrder：用户下单

UserInfo：用户信息修改

UserOrder：获取用户的订单情况

Statistics：各种统计数据获取

**（以下业务层接口 数据层接口皆不作说明，具体作用见其实现类）**

**Service(接口)/ServiceImpl：**

ManagerServicelmpl：实现经理的所有职责

RestaurantServicelmpl：实现商家的所有操作

UserServicelmpl：实现用户的所有操作

**Dao(接口)/DaoImpl：**

CommoditylnfoDaolmpl：关于CommodityInfo表的增删改查操作

OnlineBankingDaolmpl：关于OnlineBanking表的增删改查操作

OrderDaolmpl：关于Order表的增删改查操作

RestaurantDaolmpl：关于Restaurant表的增删改查操作

UserDaolmpl：关于User表的增删改查操作

VerificationDaolmpl：关于Verification表的增删改查操作

**Model：**

Address：用户地址对象 user与address是一对多关系

Commoditylnfo：商品信息对象

OnlineBanking：模拟的网上银行账户对象

Order：订单对象

Restaurant：商家对象

User：用户对象

User\_ ResStatistic：方便数据库查询所建，用来保存查询到的某个用户在某个餐厅的消费总额。

Verification：验证信息对象

**VO：（与前端交互所使用的对象）**

CommodityVO：商品信息

OrderVO：订单信息

ResTimeVO：用于传输商家的统计页面的时间筛选区间

UserTimeVO：用于传输用户统计页面的时间筛选区间

UserVO：用户登录信息

**Util：(工具包）**

CodeUtil：随机生成四位验证码

MailUtil：发送验证邮件

TimestampUtil：解决时区不同导致的时间偏差问题

### 各前端的页面：名称及功能

**Components：**

ManagerLogin.vue：经理登录页面

ManagerHeader.vue：经理主页面header组件

ApproveApplication.vue：经理审批商家信息页面

ManagerStatistics.vue：经理的统计页面

RestaurantLogin.vue：商家登陆页面

RestaurantHeader.vue：商家主页面的header组件

RestaurantManager.vue：商家管理商品/套餐信息的页面

RestaurantOrder.vue：商家订单查看页面

Restaurantlnfo.vue：商家信息（修改）页面

RestaurantStatistics.vue：商家的统计页面

Login.vue：用户登录页面

ActivePage.vue：用户点击邮箱链接进入的的验证成功/失败页面

MyHeader.vue：用户主页面的header组件

RestaurantList.vue：商家列表

RestaurantCommodityList.vue：商品列表

UserOrder.vue：用户订单查看页面

Userlnfo.vue：用户信息（修改）页面

UserStatistics.vue：用户的统计页面

## 四、其他

### 1、开发环境

数据库 mysql

### 开发心得体会

1. 把前端框架运用好，可以加快整个项目的开发。
2. 后期数据库改动代价大，前期对数据库的设计要考虑仔细。
3. 接口的设计需要多加考虑，避免代码重复和耦合过重。

### 其他说明

前端用到了iview组件库 ，需要在/YummyVue/YummyVue中

执行

[npm](https://www.baidu.com/s?wd=npm&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd) install iview --save 在项目中引入iview

npm install iview-loader --save-dev

使用了axios与后端交互，

执行

npm install vue-axios --save

npm install qs.js --save

使用了echarts来显示图表，

执行

npm install vue-echarts --save