纯心找茶

概

要

设

计

|  |  |
| --- | --- |
| **项目名称** | 纯心找茶 |
| **文档撰写** | 桂云飞 |
| **建立日期** | 2021年2月14日 |

目录

[1 概述 3](#_Toc64448110)

[2 系统架构图 3](#_Toc64448111)

[3 系统逻辑架构图 4](#_Toc64448112)

[4 物理拓扑图 4](#_Toc64448113)

[5 部署结构图 5](#_Toc64448114)

[6 关键技术 5](#_Toc64448115)

[6.1 Echarts图可视化展示 7](#_Toc64448116)

[6.2 分页与搜索 9](#_Toc64448117)

[6.3 后台管理的登录拦截 10](#_Toc64448118)

[7 关键设计模块 11](#_Toc64448119)

[7.1 部分功能界面 11](#_Toc64448120)

[7.1.1 首页展示 11](#_Toc64448121)

[7.1.2 推荐商品展示 11](#_Toc64448122)

[7.1.3 可视化数据展示 12](#_Toc64448123)

[7.2 数据库设计 12](#_Toc64448124)

[7.2.1 管理员表 12](#_Toc64448125)

[7.2.2 商品表 13](#_Toc64448126)

[7.2.3 商品类型表 13](#_Toc64448127)

[7.2.4 商品评论表 14](#_Toc64448128)

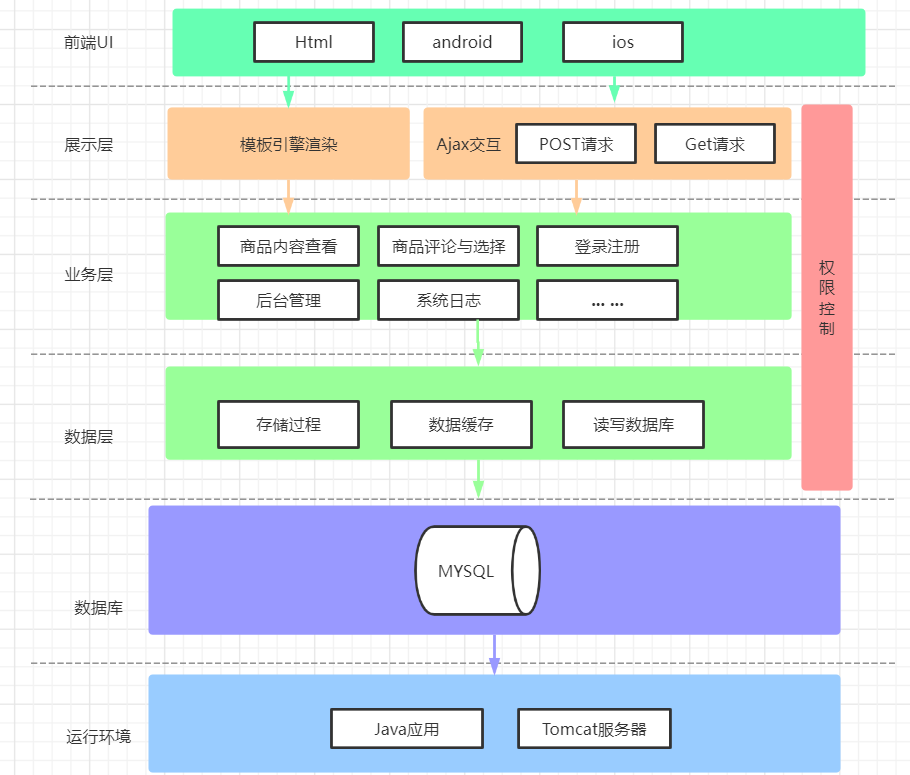
[7.3 关键功能模块 14](#_Toc64448129)

[7.4 业务逻辑 14](#_Toc64448130)

# 概述

“纯心找茶”主要任务是让用户浏览获取平台的一些热销产品以及可视化的数据展示。从而引导客户进行消费以及管理员后台商品管理。该文档旨在帮助开发人员了解系统的整体架构，关键技术和关键模块的设计，帮助开发人员了解系统的设计概况，为详细设计做铺垫。

# 系统架构图



# 物理拓扑图

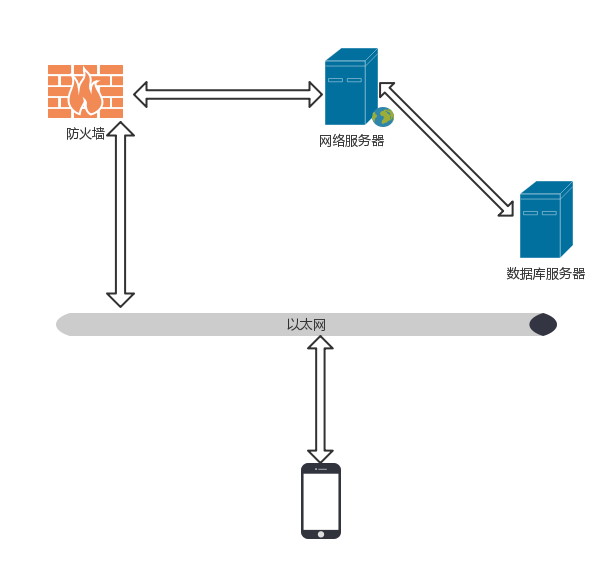


图4-1

# 部署结构图

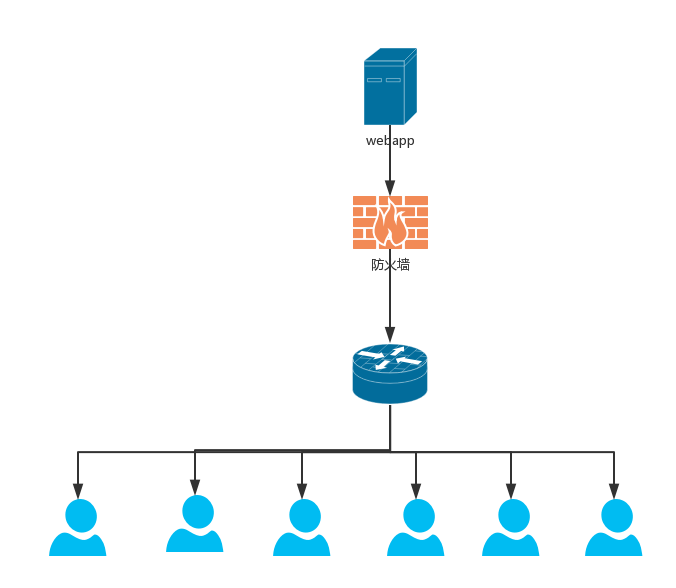


图5-1

# 关键技术

本项目主要使用SSM框架的个人开发项目，SSM即(SpringMVC Spring Mybatis)其中

**Mybatis框架的优点主要为：**

1、最简单的持久化框架、小巧简单易学

2、SQL代码从程序代码中彻底分离出来，可重用

3、提供XML标签，支持编写动态SQL

4、提供映射标签，支持对象与数据库的ORM字段关系映射

**Spring框架的优点主要为：**

1、方便解耦，简化开发 （高内聚低耦合）

Spring就是一个大工厂（容器）,用于创建对象(bean)和维护对象间的依赖关系.

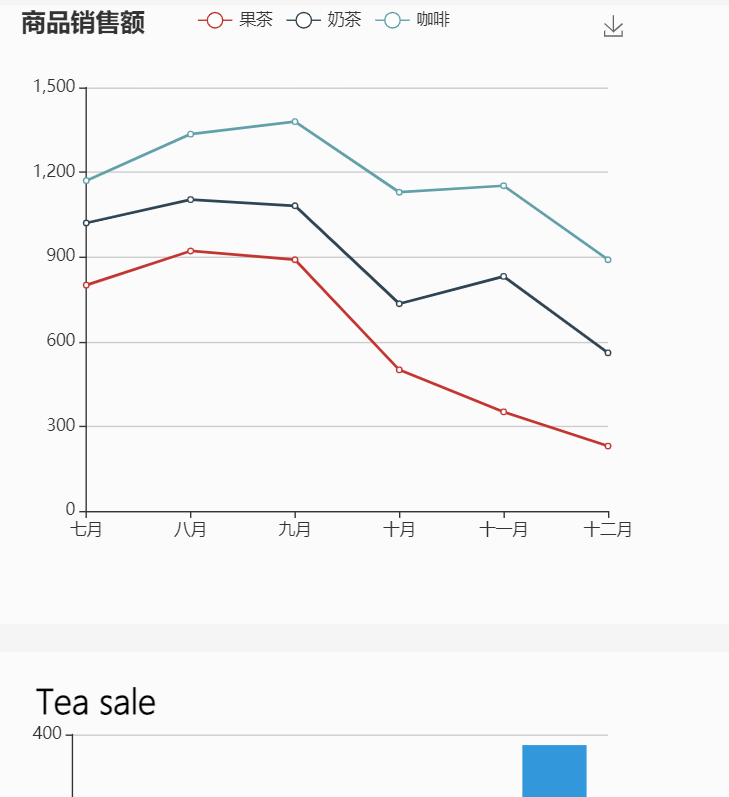
2、AOP编程的支持

Spring提供面向切面编程,可以方便的实现对程序进行权限拦截、运行监控等功能.

声明式事务的支持,只需要通过配置就可以完成对事务的管理,而无需手动编程

**SpringMVC的主要优点：**  
1、封装代码，维护成本低，耦合性低；  
2、有利于开发中的分工，提高开发效率；  
3、组件重用，有利于代码复用，重用性高

## Echarts图可视化展示



案例

|  |
| --- |
| $(function () {  var myCharts1 = echarts.init(***document***.getElementById('main1'));  var option1 = {  title: {  text: '商品销售额'  },  tooltip: {  trigger: 'axis'  },  legend: {  data: ['果茶', '奶茶', '咖啡']  },  grid: {  left: '3%',  right: '4%',  bottom: '3%',  containLabel: true  },  toolbox: {  feature: {  saveAsImage: {}  }  },  xAxis: {  type: 'category',  boundaryGap: false,  data: ['一月', '二月', '三月','四月','五月','六月']  },  yAxis: {  type: 'value'  },  series: [  {  name: '果茶',  type: 'line',  stack: '总量',  data: [100, 121, 213, 300, 321, 400, 584]  },  {  name: '奶茶',  type: 'line',  stack: '总量',  data: [320, 450, 412, 359, 413, 342, 350]  },  {  name: '咖啡',  type: 'line',  stack: '总量',  data: [220, 232, 298, 395, 321, 330, 410]  }  ]  };  myCharts1.setOption(option1);  }) </script> |

使用Echarts其底层依赖轻量级Canva+库 ZRender，提供直观，生动，可交互，可高度个性化定制的数据可视化图表。通过非常直观的图形化的界面，从而直观的看出哪些商品的销售额较大。用户可以更直观的做出判断。其数据管理员可通过后台数据管理，对数据进行修改，进而使普通客户端能够查看更客观的数据。

Echarts的使用也非常简单。

1.通过 var myCharts1 = echarts.init(document.getElementById('main1'));初始化一个容器

2.通过 var options 通过数组的方式从数据库获取将数据放置到该文本中

3.通过 myCharts1.setOption(option1);将数据放在容器中即可。

## 分页与搜索

分页显示，也就是讲数据库中的结果集，一段一段显示出来怎么分段，当前在第几段（每页有几条，当前再第几页）。当后台管理员对商品的添加等逐渐增多时、超过一定数量时，分页与搜索功能的愈发重要。分页使得界面不会过于冗长，影响整体美感；搜索让用户可以快速找到商品，节约时间。





该分页主要特点还有，可以进行展示一页有多少个数据，从而进行相应的数据展示选择。

## 后台管理的登录拦截

因为后台管理仅仅需要的只是我们的本地的管理员一个小团队进行操作。我们并不希望外界用户或者普通用户进行访问。所以我们对管理员的登录操作进行了一系列的拦截操作用以保证管理员登录的安全性。

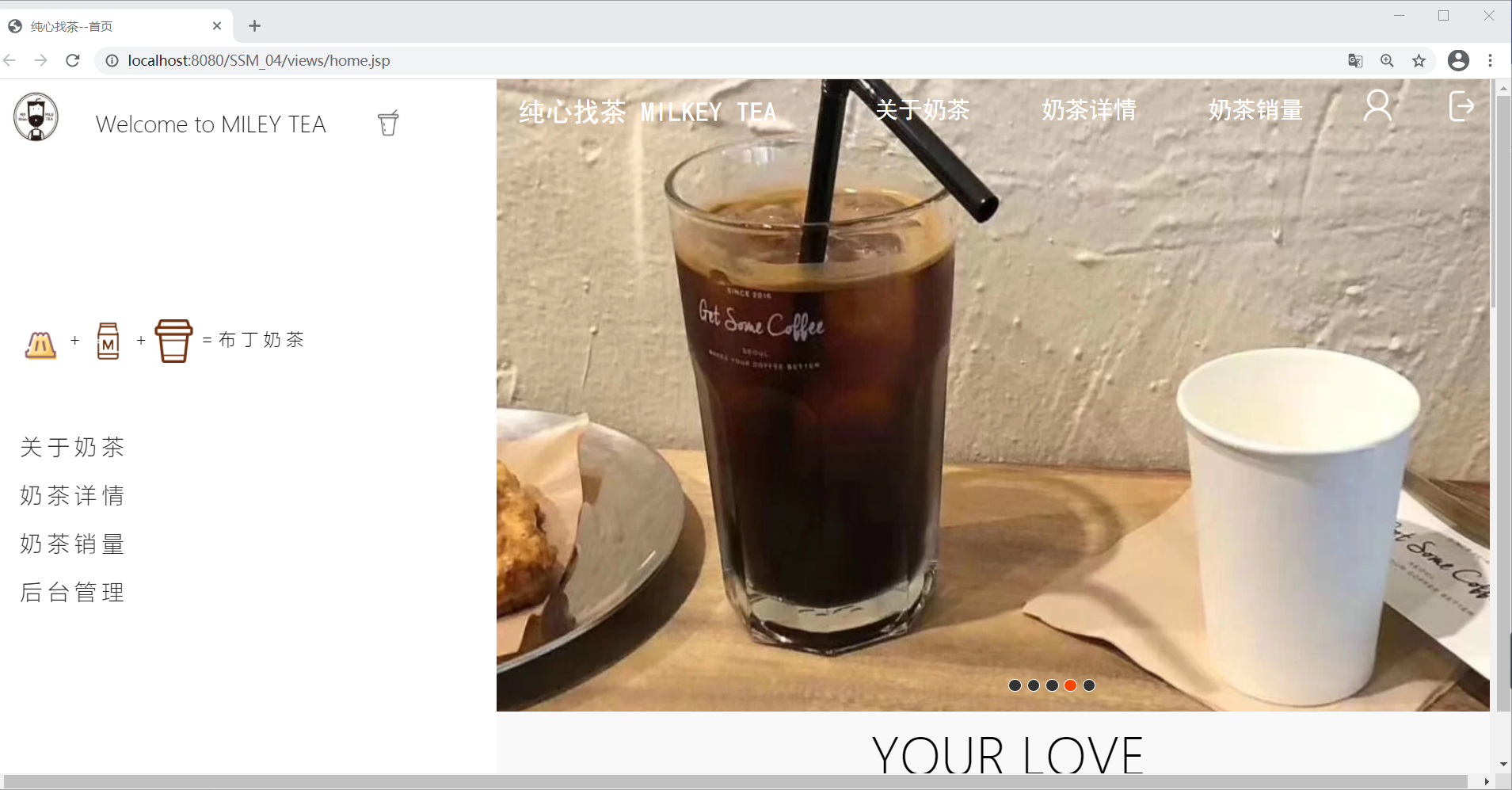
|  |
| --- |
| //验证失效时间  String expireTime = user.getExpireTime();  String currentTime = DateTimeUtil.*getSysTime*();  if(expireTime.compareTo(currentTime)<0){  throw new LoginException("账号已失效");  }  //判断锁定状态  String lockState = user.getLockState();  if("0".equals(lockState)){  throw new LoginException("账号已锁定");  }  //判断ip地址  String allowIps = user.getAllowIps();  if(!allowIps.contains(ip)){  throw new LoginException("ip地址受限");  }  return user; } |

我们通过我们数据库允许设置的IP地址，限制管理员登录的局域网。必须在相应的局域网内，进行用户登录。并且考虑到员工入职或离职的情况，将管理员的账号通过冻结或者解冻的两种状态。用来保证后台数据的严谨性。不干扰正常工作或由账号异常等问题出现的数据冗余，数据不正常的现象。

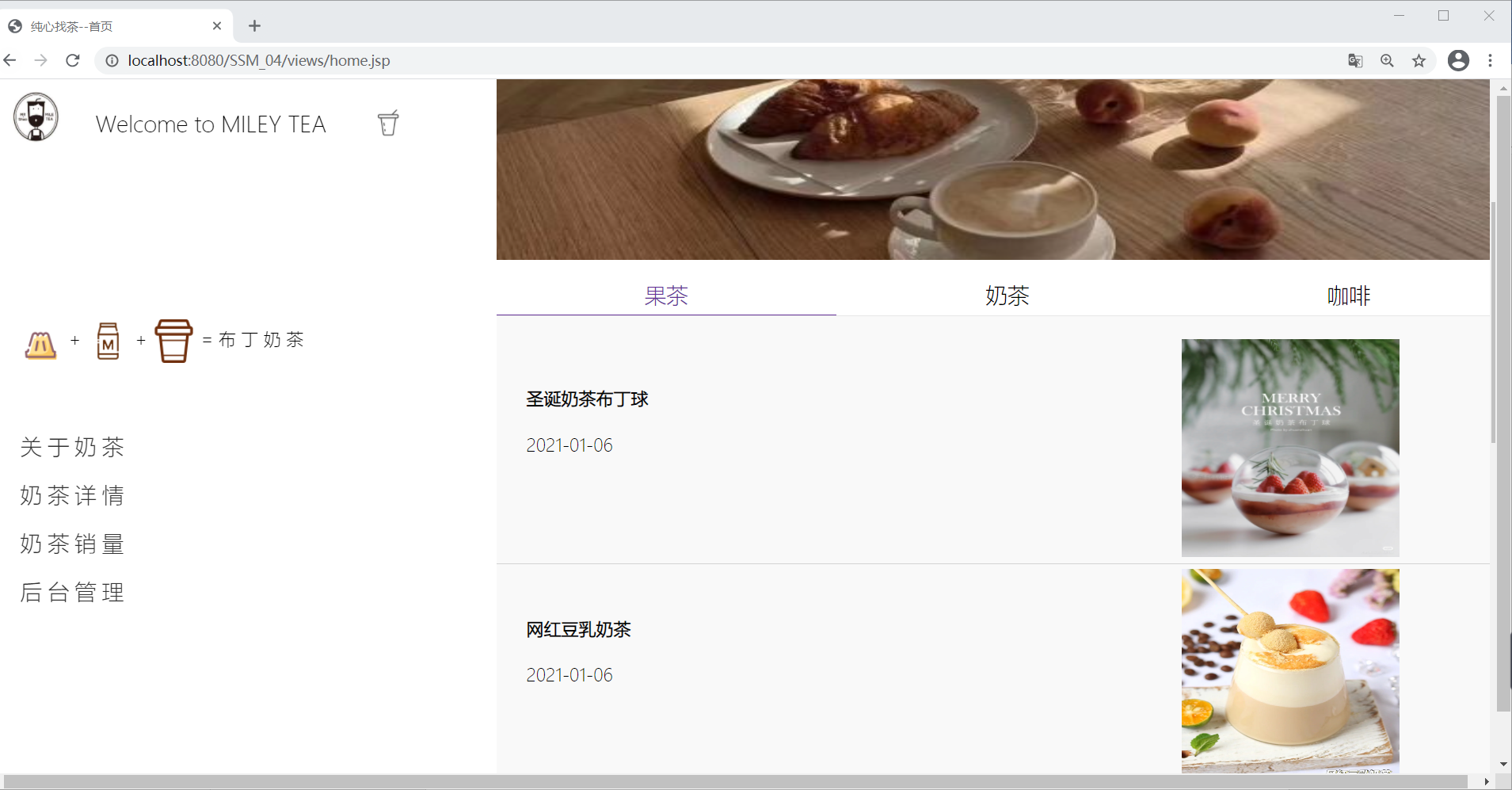
# 关键设计模块

## 部分功能界面

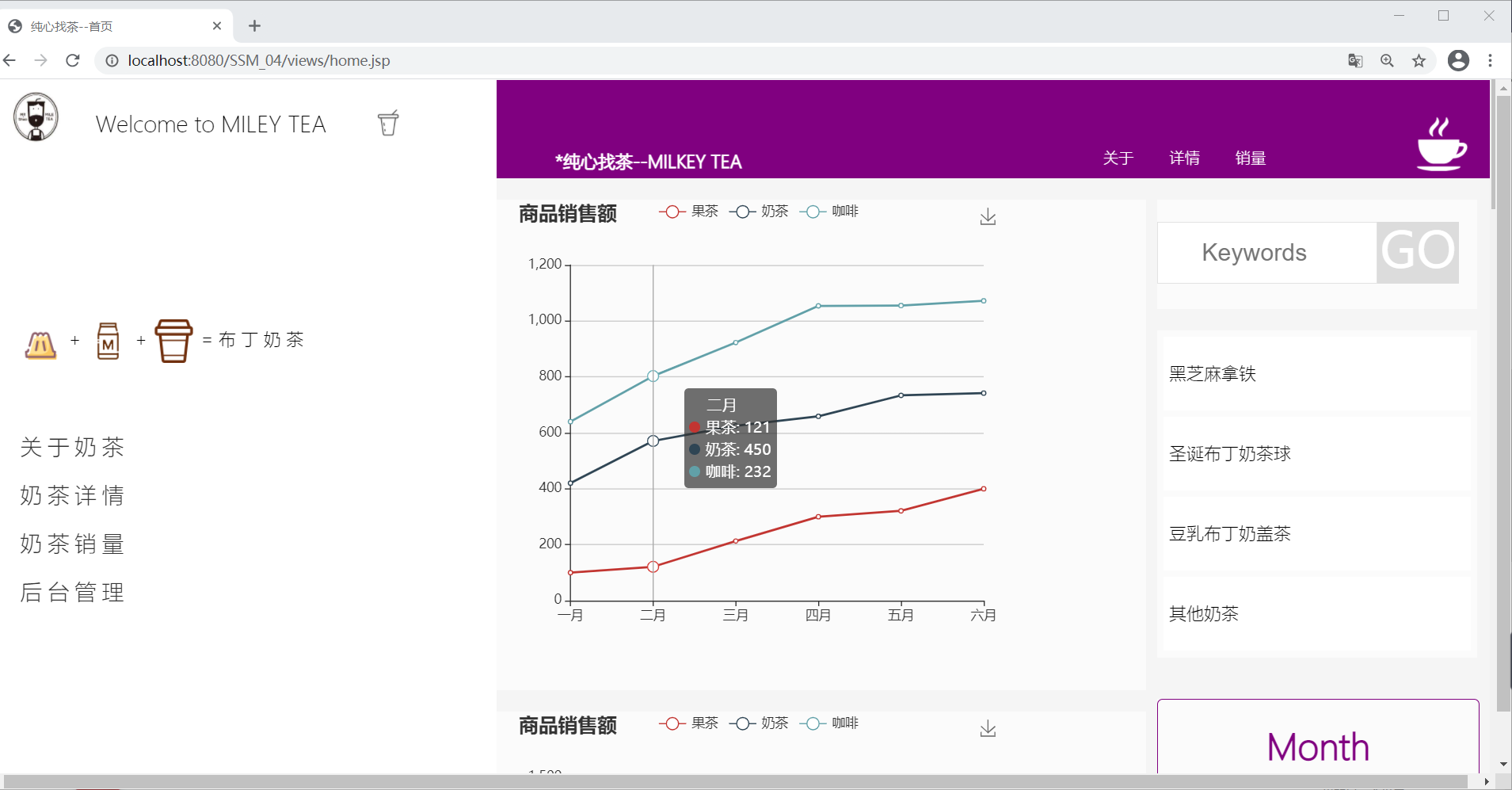
### 首页展示



### 推荐商品展示

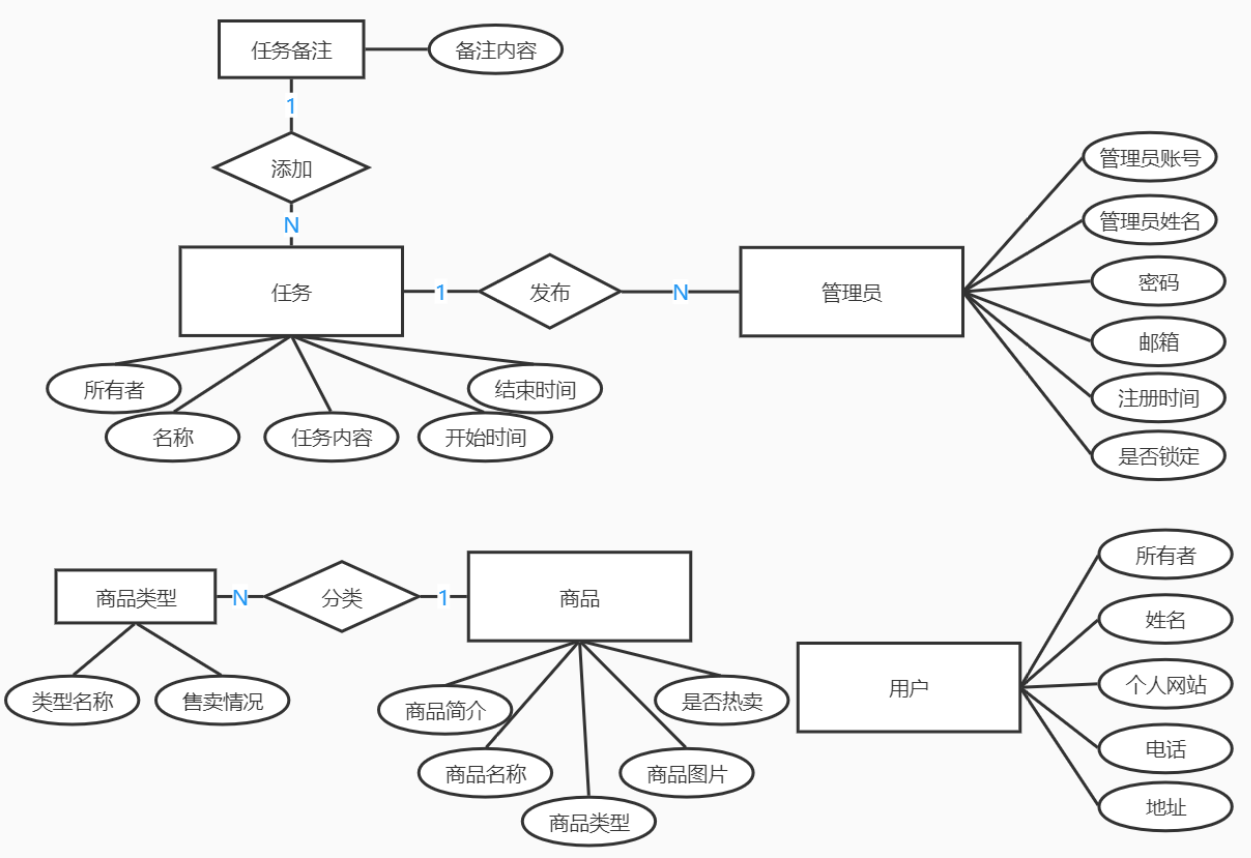


### 可视化数据展示



## 数据库设计

根据功能界面分析，得到系统的ER图。



## 关键功能模块

* 普通用户管理： 主要进行浏览数据并对商品进行评价。
* 管理员管理：管理员发布每日任务，其他管理员可以对商品的材料进行购入。
* 商品信息管理：管理员和普通用户均可通过页面进行查看，普通用户能更直观的查看到热销商品，以及其他用户的评论。
* 任务管理：商家可以对每日的任务进行增删改的操作，定期查看任务和历史任务可以方便对商品材料的管理。

## 业务逻辑

* 普通用户：可以通过主页面浏览推荐的商品，并对其进行评价以及通过可视化数据从而进行引导性消费。
* 商品：商品主要包括增删改查，商品评论，商品备注的一些内容管理。商品与商品备注，商品评论均是一对多的关系。
* 管理员：管理员通过对每日任务的发布与其他管理员对用户的备注，通过评论的方式，较为醒目。其中每日任务与每日任务评论是一对多的关系。并且管理员可以对每日任务以及每日任务的评论进行增删改查。