

网络 GIS 课间实习

实 习 报 告

专业： 遥感科学与技术

学号： 2017302590132

姓名： 姜屿涵

遥感信息工程学院

2020 年 5 月

1. 实习目的

本次实习旨在学习和掌握网络 GIS 开发的基本流程和基础技术，在实践中学习，完成以下四个任务：

任务一：编写网页，用到框架、背景图片、表单、表格、列表、链接等。

任务二：参考购物车.html 编写一个网页。

任务三：根据参考样例，编写一个“静态”地图网页，具体功能自定义。

任务四：熟悉 XML 和 JSON，尝试编写 XML 和 JSON 的读写、相互转换（转 csv）。

结合校园的跳蚤市场和供需交流平台等信息，我将以上四个小任务整合为一个基于百度地图 API 的“校内供需交易平台”。一是构建商品列表，并将发布的所有二手物品在地图上显示，考虑到二手物品的唯一性，所有的商品只能售出一次，售出后的商品在地图上为灰色图标，不可购买或添加到购物车；未售出的商品在地图中为蓝色图标，可以购买或添加到购物车，一旦售出更新为灰色；用户在列表中点击浏览的商品以“小人”的图标突出显示。二是用户可以将商品添加到购物车，之后选择删除或购买；三是购买商品后，选择收货或退货；四是用户可以提交表单发布自己的二手物品，平台根据输入的文本地址信息，调用百度 API 实现经纬度获取，实时在地图中定位显示，进行售卖。初始数据有 csv 文件导入 MySQL 数据库中，通过后台接口转换为 JSON 数据，在前台利用 AJAX 方法完成调用和显示。

2. 实习环境

2.1 开发工具

1. IntelliJIDEA

IDEA 全称 IntelliJIDEA，是 java 编程语言开发的集成环境。IntelliJ 在业界被公认为最好的 java 开发工具，尤其在智能代码助手、代码自动提示、重构、J2EE 支持、各类版本工具(git、svn 等)、JUnit、CVS 整合、代码分析、创新的 GUI 设计等方面的功能可以说是超常的。它的旗舰版本还支持 HTML，CSS，PHP，MySQL，Python 等。

2. MySQL

关系型数据库管理系统。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL 是最好的 RDBMS(RelationalDatabaseManagementSystem，关系数据库管理系统)应用软件之一。

3. Navicat

一套快速、可靠并价格相当便宜的数据库管理工具，拥有图形用户界面，是专为简化数据库的管理及降低系统管理成本而设的。

4. JDK

JDK 是 Java 语言的软件开发工具包,主要用于移动设备、嵌入式设备上的 java 应用程序。JDK 是整个 java 开发的核心,它包含了 JAVA 的运行环境 (JVM+Java 系统类库) 和 JAVA 工具。

5. Maven

Maven 是一个项目管理工具,可以对 Java 项目进行构建、依赖管理。

6. Postman

Postman 是用于 API 开发的协作平台。Postman 的功能简化了构建 API 的每个步骤并简化了协作,因此您可以更快地创建更好的 API。在本项目中主要用于,调试后端代码,查看 http 和 https 的 GET/POST/PUT/DELET 等请求的直接结果。

7. Google Chrome

Google Chrome 是由 Google 开发的一款设计简单、高效的 Web 浏览工具。在本项目中主要是使用其自带的开发者选项,调试模块,进行前端代码的调试。

2.2 开发语言

1. HTML+CSS

超文本标记语言 (HTML) 是一种用于创建网页的标准标记语言。层叠样式表是一种用来表现 HTML (标准通用标记语言的一个应用) 或 XML (标准通用标记语言的一个子集) 等文件样式的计算机语言。CSS 不仅可以静态地修饰网页,还可以配合各种脚本语言动态地对网页各元素进行格式化。在本项目中,主要用于编写前端 Web 样式。

2. JavaScript

JavaScript 是一种属于网络的脚本语言,已经被广泛用于 Web 应用开发,常用来为网页添加各式各样的动态功能,为用户提供更流畅美观的浏览效果。在本项目中,主要用于编写动态效果以及实现 HTTP 请求,获取后台提供的数据。

3. Java

Java 是一门面向对象编程语言,不仅吸收了 C++ 语言的各种优点,还摒弃了 C++ 里难以理解的多继承、指针等概念,因此 Java 语言具有功能强大和简单易用两个特征。Java 可以编写桌面应用程序、Web 应用程序、分布式系统和嵌入式系统应用程序等。在本项目中,主要用于后台服务器的搭建。

4. SQL

结构化查询语言 (Structured Query Language) 简称 SQL, 是一种特殊目的的编程语言,是一种数据库查询和程序设计语言,用于存取数据以及查询、更新和管理关系数据库系统。

2.3 开发环境

2.3.1 前端开发

1. Bootstrap

Bootstrap 是美国 Twitter 公司的设计师 Mark Otto 和 Jacob Thornton 合作基于 HTML、CSS、JavaScript 开发的简洁、直观、强悍的前端开发框架，使得 Web 开发更加快捷。Bootstrap 提供了优雅的 HTML 和 CSS 规范，它即是由动态 CSS 语言 Less 写成。

2. jQuery

具有独特的链式语法和短小清晰的多功能接口；具有高效灵活的 css 选择器，并且可对 CSS 选择器进行扩展；拥有便捷的插件扩展机制。

3. AJAX

AJAX=异步 JavaScript 和 XML。AJAX 是一种用于创建快速动态网页的技术。通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

本次实习使用 Bootstrap 模板和 jQuery 搭建前端的基本样式，利用 AJAX 实现与服务器交换数据。

2.3.2 后端开发

1. Spring Boot

Spring Boot 是由 Pivotal 团队提供的全新框架，其设计目的是用来简化新 Spring 应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。另外 Spring Boot 通过集成大量的框架使得依赖包的版本冲突，以及引用的不稳定性等问题得到了很好的解决。

2. Spring Data JPA

Spring Data 是 Spring Framework 的一部分。Spring Data 存储库抽象的目标是显著减少为各种持久性存储实现数据访问层所需的代码量。Spring Data JPA 不是 JPA 提供者。它是一个库/框架，它在我们的 JPA 提供程序（如 Hibernate）的顶部添加了一个额外的抽象层。其致力于减少数据访问层（DAO）的开发量，开发者唯一要做的，就只是声明持久层的接口，其他都交给 Spring Data JPA 来完成。

本实习基于以上后端开发技术，完成与数据库的连接，实现对 MySQL 数据库的增、删、改、查，并将从数据库中获取的数据以 Json 的格式传递给前端，进行展示或响应前端发起的表单提交等更新数据库的操作。

3. HTML+CSS 编程实践

3.1 实习内容

3.1.1 开发流程图

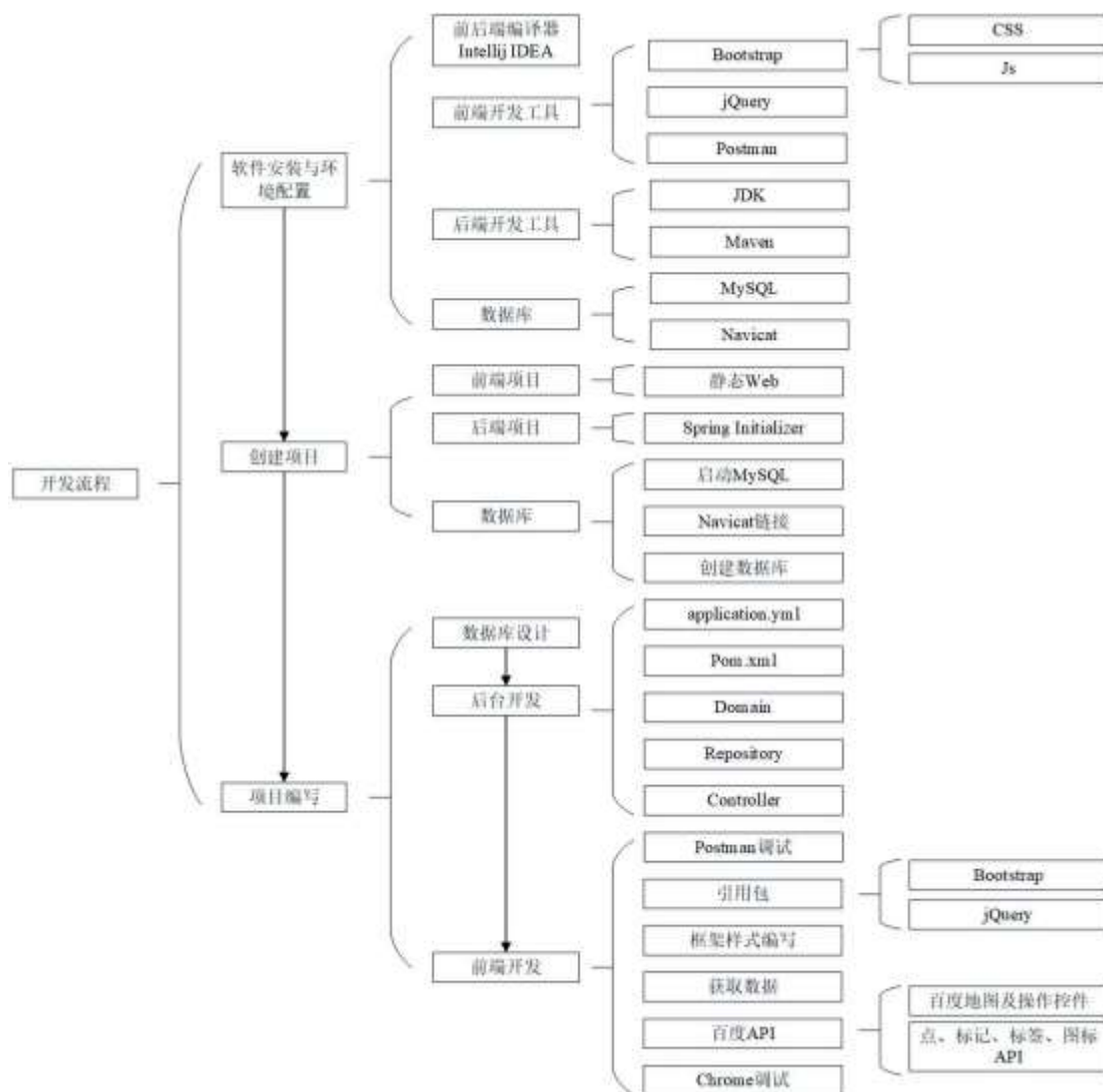


图 1 开发设计流程图

平台的开发流程如上图所示，箭头处为流程顺序。主要是三大块内容，首先是相关软件的安装和相关框架包的下载，安装和下载的顺序随意。其次是项目的创建，创建的顺序如图所示，前端是静态 Web，后端是 Springboot 项目，数据库是 MySQL 数据库，具体细节见下文。最后是项目的编写，项目的编写按照如图所示的顺序，首先设计数据库，再开发对应的后台接口，最后是前端的开发。

3.1.2 开发流程

1. 软件安装与环境配置

此处省略相关的编译器、调试工具、JDK、Maven 等的安装，一是注意环境变量的配置；二是用到的 JDK 版本为 1.8.0，Maven 版本为 2.1.3，MySQL 版本为 5.7，注意不同的版本可能会有冲突。此处具体介绍用到的模板的下载与安装。

- 1) Bootstrap: 从模板网站下载，解压后放入工程文件下即可。

<http://www.cssmoban.com/tags.asp?page=3&n=bootstrap>

- 2) jQuery: 从官网下载 jQuery 库，下载 Developmentversion，用于测试和开发版本（未压缩，是可读的代码），解压后放入工程文件下即可。

<https://jquery.com/>

2. 创建项目

- 1) 前端项目

- 打开 IntelliJ IDEA，选择创建项目 2.选择创建静态 Web 项目
- 选择项目存放地址并命名
- 新建一个 HTML 页面
- 将 Bootstrap 模板的库文件，jQuery 包，与 HTML 文件放在同一目录下

- 2) 后端项目

- 打开 IntelliJ IDEA,选择创建项目
- 选择创建 Spring Initializer 项目。
- Project SDK 选择之前下载安装好的 JDK。
- Choose Initializer Service URL 选择 Default:http://start.spring.io。
- 填写 Group 和 Artifact 信息，注意 Aritifact 不能包含大写字母，其他按照默认即可。
- Type: Maven Project。
- Packaging: 项止打包方式，可以选择 jar 包或 war 包，由于 spring-boot 已集成 tomcat，故可以用 java-jar 方式运行。
- 选择主体功能，Web，选择 Spring Boot 版本 2.1.3。

- 修改工程名和路径选择。
- 进入项目，将.mvn、mvnw、mvnw.cmd 删除。
- 查看创建的项目信息：

Spring Boot 版本：

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.1.3.RELEASE</version>
  <relativePath/> <!-- lookup parent from repository -->
</parent>
```

Java 版本：

```
<properties>
  <java.version>1.8</java.version>
</properties>
```

Web 后台项目：

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

Maven 打包插件：

```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>
```

3) 创建数据库

- 在服务中启动 MySQL
- NavicatforMySQL 链接数据库
 - a. 点击文件->新建连接->MySQL
 - b. 输入密码，查看端口号是否正确
 - c. 确定，链接成功

- 创建数据库，设置数据库名，字符集选择 utf8mb4--UTF-8Unicode。

3. 项目编写

1) 数据库编辑

- 建立数据库中的表，平台涉及三个表，分布式购物车表、购物清单表（购买过的所有商品）、商品表。所有表的引擎设置为 InnoDB，字符集为 utf-8。
- 设计每一个表的属性字段和数据类型，设置主键和外键。

a. 商品表

product_id 为主键，唯一标识每一个商品。name 为商品的名称。price 为商品的售价。discription 为商品描述，比如该商品为 9 成新。address 为商品的文本地址。lng 和 lat 为地址对应的经纬度。type 为商品的类型，平台设计了 5 类商品，分布式书籍类、衣物类、化妆品类、生活用品类和其它。available 用于标识该二手商品是否已经卖出，如果卖出标记为“0”，未卖出标记为“1”。

名	类型	长度	小数点	不是 null	
product_id	int	16	0	<input checked="" type="checkbox"/>	1
name	varchar	255	0	<input type="checkbox"/>	
price	float	0	0	<input type="checkbox"/>	
discription	varchar	255	0	<input type="checkbox"/>	
address	varchar	255	0	<input type="checkbox"/>	
lng	double	0	0	<input type="checkbox"/>	
lat	double	0	0	<input type="checkbox"/>	
type	varchar	16	0	<input type="checkbox"/>	
available	varchar	16	0	<input type="checkbox"/>	

图 2 商品表设计

b. 购物车表

cart_id 为主键，唯一标识每一条购物车数据，product_id 为外键，参考商品表中的 product_id，获取购物车中商品的具体信息。add_date 为商品加入购物车的时间，由用户点击“加入购物车”时实时获取。注意相同的商品不能重复加入购物车。

名	类型	长度	小数点	不是 null	
cart_id	int	11	0	<input checked="" type="checkbox"/>	1
product_id	int	11	0	<input type="checkbox"/>	
add_date	datetime	0	0	<input type="checkbox"/>	

图 3 购物车表设计

名	栏位	参考数据库	参考表	参考栏位	删除时	更新时
c_product_id	product_id	supdem2	product	product_id	CASCADE	CASCADE

图 4 购物车表外键

c. 购物清单表

checkboxlist_id 为主键，唯一标识购买过的所有商品。product_id 为外键，参考商品表中的 product_id，获取购物车中商品的具体信息。purchase_date 为商品被购买的时间，由用户点击“立即购买”时实时获取。status 用于标识商品

被购买后的状态，平台设计了三种状态，“1”代表用户购买后商品正在发货中，“2”代表用户确认收货，“3”代表退货，当且仅当买家确认收货后，交易结束，同步更新商品列表的 available 属性为“0”，代表商品已售出不可再购买，但是退货后的商品若 available 为“1”，可以再次购买。注意相同的商品不能重复加入购买清单。

名	类型	长度	小数点	不是 null	
checklist_id	int	16	0	<input checked="" type="checkbox"/>	1
product_id	int	16	0	<input type="checkbox"/>	
purchase_date	datetime	0	0	<input type="checkbox"/>	
status	varchar	16	0	<input type="checkbox"/>	

图 5 购物清单表设计

名	栏位	参考数据库	参考表	参考栏位	删除时	更新时
product_id	product_id	supdem2	product	product_id	CASCADE	CASCADE

图 6 购物清单表外键

2) 后台开发

➤ 后台总体框架

通过使用 Spring Boot 框架，搭建后端。总体框架如下：

pom.xml: Maven 的配置文件，在该文件中引入各类依赖。本平台主要是 Jpa 方法，用于操作数据库。

SupdemApplication: 项目启动文件。

Application.yml: 项目配置文件，一是设置与前端交互的根端口号；二是设置与数据库连接的参数（数据库端口号、密码、url）；三是设置 jpa 方法，操作数据库，此处设置为“update”，表示更新数据库，即保留数据库原有数据，向其中增、删、改数据。

```
server:
  port: 8082
  servlet:
    context-path: /supdem

spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://localhost:3306/supdem2?useUnicode=true&characterEncoding=utf8
    username: root
    password: 123456

  jpa:
    hibernate:
      ddl-auto: update
      show-sql: true
  mvc:
    view:
      prefix: /WEB-INF/
      suffix: .jsp
```

图 7 项目配置信息

Controller: 编写前端与后台的接口 url，前端 ajax 获取数据的 url 在此处编写，相关的增、删、改、查和算法在 Controller 中实现。每个对象对应一个 Controller 文件。

Domain: 对象，与数据库中的表对应。每个表是一个对象。

Repository: 对象与数据库的接口，继承 Jpa 方法，实现数据库操作。

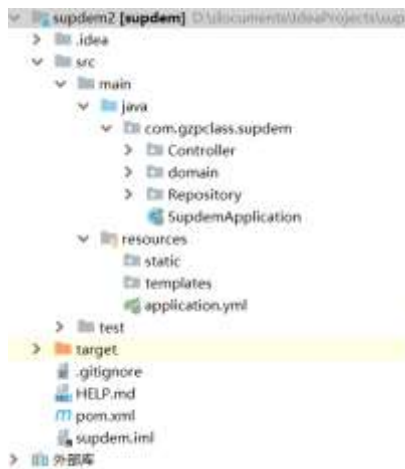


图 8 后台总体框架

➤ 后台详细开发

a. 创建对象 Domain

后台中的对象，对应于数据库中的表，对象的属性即为表的属性字段，对象的方法为通过 Get 方法获取属性值和 Set 方法修改属性。数据库中的独有的表为 hibernate_sequence, 为后台控制数据库的 id 序号（若插入数据，序号依次递增，反之则反）。以 cart 购物车对象为例，Java 代码结构如图 10 所示。



图 9 对象总和

```
public class cart {  
    @GeneratedValue  
    private Integer cartId;  
    private Integer productId;  
    private Date addDate;  
  
    public Integer getCartId() { return cartId; }  
    public void setCartId(Integer cartId) { this.cartId = cartId; }  
  
    public Integer getProductId() { return productId; }  
    public void setProductId(Integer productId) { this.productId = productId; }  
  
    public Date getAddDate() { return addDate; }  
    public void setAddDate(Date addDate) { this.addDate = addDate; }  
}
```

图 10 后端对象及其属性方法

b. 创建 Repository 继承 Jpa 方法

建立每一个对象与数据库对应的接口，通过继承 JPA 的方法实现对数据库的操作，增、删、改、查。图 11 以 product 商品对象继承的方法为例，继承后会直接有默认的查找所有数据和增删改的基础方法，此处扩展继承了按照商

品类型 type、商品是否可售 available、按照 id 查找三类方法。

```
Repository
├── cartRepository
├── checklistRepository
├── HistoryOrderRepository
├── productRepository

public interface productRepository extends JpaRepository<product,Integer> {
    public List<product> findByType(String type);
    public List<product> findByAvailable(String available);
    public product findFirstByOrderByProductIdDesc();
}
```

图 11 对象与数据库的接口及 Jpa 方法的继承

c. 创建 Controller 编写接口

编写 url，为前端访问数据提供接口。以 product 商品对象为例，后台为其提供了 6 种接口，分别是获取所有的商品信息、查询最新添加的商品信息、按照 Id 查询商品信息、添加新商品、更改已经发布的商品的信息以及删除商品。其中百度 API 的方法再添加新商品时使用，用于解析地址的经纬度信息，当无法定位到用户输入的地址时，默认为武大校内。请求类型包括“POST”“PUT”“GET”“DELETE”。

```
Controller
├── cartController
├── checklistController
├── HistoryOrderController
├── productController

@RequestMapping("/product")
public class productController {

    @Autowired
    private productRepository repository;

    //加载所有数据
    @GetMapping
    public List<product> all() { ... }

    //查询最新添加的数据
    @GetMapping
    public product latest() { return repository.findFirstByOrderByProductIdDesc(); }

    //Id查询
    @GetMapping
    public Optional<product> findById(@PathVariable("productId") Integer id) { return repository.findById(id); }

    //添加商品
    @PostMapping
    public product create(@RequestParam("name") String name, @RequestParam("price") String price,
        @RequestParam("discription") String discription, @RequestParam("address") String address, @RequestParam("type") String type
        ) { ... }

    //更新商品
    @PutMapping
    public product productUpdate(@PathVariable("productId") Integer id, @RequestParam("name") String name, @RequestParam("price") String price,
        @RequestParam("discription") String discription, @RequestParam("address") String address, @RequestParam("type") String type,
        @RequestParam("available") String available) { ... }

    //删除商品
    @DeleteMapping
    public void productDelete(@PathVariable("productId") Integer id) { repository.deleteById(id); }

    // 调用百度地图API根据地址，获取坐标
    public static double[] getCoordinate(String address) { ... }
    public static String loadJSON(String url) { ... }
}
```

图 12 前端访问接口及使用 Jpa 方法操纵数据库

➤ 端口调用调试

在 Postman 中输入端口号，设置请求方法，判断前端是否能正常通过接口请求数据，后台是否能正确的操作数据库，查看 url 返回的数据格式（JSON）。下图为“PUT”方法的请求示例，以及获取的返回的 JSON 数据。按照相同的方法在 Postman 中，检查所有接口是否全部编写好。

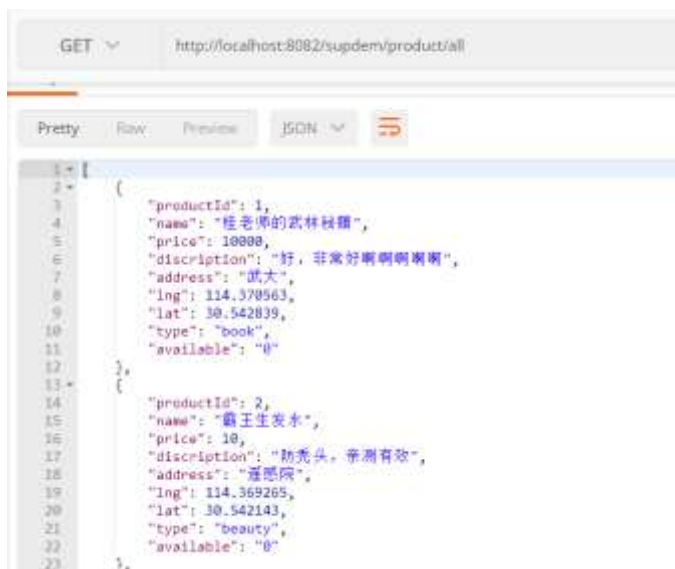


图 13 前端获取的 JSON 数据

3) 前台开发

➤ 欢迎页面开发

在欢迎页面主页完成图片背景和链接的使用。使用不同的图片动态播放形成欢迎页面的背景，设置按钮链接跳转到平台的主页。此处仅展示实现图片动态播放的 JavaScript 代码，timeInterval 是用于控制图片切换速度的参数。

```
setInterval(changeImg, timeInterval);  
function changeImg()  
{  
    var obj=document.getElementById("obj");  
    if (currentIndex==arr.length-1)  
    {currentIndex=0;}  
    else  
    {currentIndex+=1;}  
    obj.src=arr[currentIndex];  
}
```

图 14 图片切换

➤ 主页开发

主页的开发主要是在百度地图中定位显示商品、实现用户添加商品、购物车收藏商品、购买商品的功能。此次仅介绍核心内容。执行任何操作时，通过表单提交和 AJAX 的方法，地图上的 Marker、购物车的数据状态、数据库中

的数据同步更新。

a. 引入 Bootstrap 模板、jQuery 的库、百度 API

```
<!--bootstrap-->
<link rel="stylesheet" href="https://cdn.bootcss.com/bootstrap/3.3.7/css/bootstrap.min.css">
<link rel="stylesheet" href="https://cdn.bootcss.com/bootstrap/3.3.7/css/bootstrap-theme.min.css">
<script src="bootstrap.min.js"></script>
<!--jquery-->
<script src="jquery-3.4.1.js"></script>
<script type="text/javascript" src="jQuery.1.7i.js" language="javascript"></script>
<!--百度地图API-->
<script type="text/javascript" src="http://api.map.baidu.com/api?v=2.0&services=true&ak=4tPcrZcaduiSx5hSHGRvdAF088gEKwKc"></script>
```

图 15 引入 css 和 Javascript

b. 样式框架

参考 Bootstrap 模板的示例 HTML，布局项目的 Web 前端样式框架，在适当的位置，使用<div id=" XXX">标签留出容器的位置，用于添加百度地图、表单、表格，不同 id 对应不同容器，在 Javascript 中获取数据后，向容器中中添加项目内容。图 16 为商品列表，不同的中是不同类别的商品。图 17 为功能键，通过点击后弹出模态框，实现添加新的商品、加入购物车、购买商品。至此 Web 前端样式的框架搭建完成。

```
<ul class="nav sidebar-nav">
  <li class="sidebar-brand">
    <a href="#" class="sidebar-header" style="color: #ffffff">供需订单</a>
  </li>
  <li class="dropdown dropdown-title">
  <li class="dropdown dropdown-title">
  <li class="dropdown dropdown-title">
  <li class="dropdown dropdown-title">
  <li class="dropdown dropdown-title">
</ul>
```

图 16 供需订单列表

```
<!-- 按钮 -->
<div style="...">
  <button type="button" class="btn btn-circle btn-lg" data-toggle="modal" data-target="#exampleModal1"><i class="glyphicon glyphicon-star">
  <button type="button" class="btn btn-circle btn-lg" data-toggle="modal" data-target="#exampleModal2"><i class="glyphicon glyphicon-shoppi
  <button type="button" class="btn btn-circle btn-lg" data-toggle="modal" data-target="#exampleModal3"><i class="glyphicon glyphicon-plus">
</div>

<!-- 购物车表格 -->
<div class="modal fade overlap2" id="exampleModal1" tabindex="-1" role="dialog" data-backdrop="false" aria-labelledby="exampleModalLabel">
<!-- 购物车清单框 -->
<div class="modal fade overlap2" id="exampleModal2" tabindex="-1" role="dialog" data-backdrop="false" aria-labelledby="exampleModalLabel">
<div class="modal-dialog" role="document">
</div>
<!-- 提交表单，添加商品 -->
<div class="modal fade overlap2" id="exampleModal3" tabindex="-1" role="dialog" data-backdrop="false" aria-labelledby="exampleModalLabel">
<div class="modal-dialog" role="document">
</div>
```

图 17 功能键

c. 百度地图底图

添加百度地图底图，并更改控件的位置。


```

<!-- 百度地图 -->
<script type="text/javascript">
    var map = new BMap.Map("allmap");
    map.centerAndZoom(new BMap.Point(114.37292, 30.543804), 16); //初始中心点为武汉大学
    map.enableScrollWheelZoom(); //启用滚轮放大缩小
    map.enableKeyboard(); //启用键盘上下左右键移动地图
    map.addControl(new BMap.NavigationControl({anchor: BMAP_ANCHOR_TOP_RIGHT, type: BMAP_NAVIGATION_CONTROL_LARGE, offset: new BMap.Size(0, 60)}));
    map.addControl(new BMap.OverviewMapControl({anchor: BMAP_ANCHOR_BOTTOM_RIGHT, isOpen: 1}));
    map.enableScrollWheelZoom();
    map.addControl(new BMap.MapTypeControl());
    map.addControl(new BMap.ScaleControl({anchor: BMAP_ANCHOR_BOTTOM_LEFT}));
</script>

```

图 18 百度地图底图

d.通过 id 向 HTML 的 body 中添加信息

平台涉及商品列表的实时更新、购物车的表格实时更新、购物清单实时更新、地图 Marker 实时更新。以更新商品列表为例，介绍在 JavaScript 中实现对 HTML 页面的更新，同时设置 onclick 事件，实现实时响应“加入购物车”和“立即购买”事件，从而实现数据库的更新。

```

//创建列表
function createOption(data, type) {
    var li=document.createElement('li');
    li.innerHTML="<a onclick='addMarker('+(data.My_id)+')' datastopPropagation='true' +'>"+data.name+'</a>';
    document.getElementById(type).appendChild(li);
}

```

图 19 HTML 更新

d.AJAX 方法

平台涉及多类 AJAX 方法的数据交换和更新，以获取所有数据的接口为例，介绍前端与后台的数据交换。此处使用 ajax 的“GET”方法获取后端 url 接口的数据，数据的格式为 JSON。访问成功则将数据储存在变量中，访问失败则在网页中弹框提示。此外平台还实现了“PUT”方法用于更新数据，如修改商品的售出状态；“POST”方法用于添加数据，发布新的二手商品；“DELETE”方法删除购物车数据，表示商品已被购买。

```
$.ajax({
    type: "get",
    url: "http://localhost:8082/supdem/product/all",
    data: {},
    dataType: "json",
    async: false,
    success: function (data) {
        for (var i = 0; i < data.length; i++) {
            mJson.push({
                My_id: i,
                productId: data[i].productId,
                name: data[i].name,
                price: data[i].price,
                discription: data[i].discription,
                address: data[i].address,
                type: data[i].type,
                available: data[i].available,
                lng: data[i].lng,
                lat: data[i].lat
            });
            createOption(mJson[i], mJson[i].type);
        }
    },
    error: function (errorMsg) {
        // 请求失败时执行该函数
        alert("请求数据失败!");
    }
});
```

图 19 AJAX 数据获取

3.2 成果展示

1. 欢迎页面

图 20 为欢迎的效果图，以及自动切换背景图的效果，中间的按钮链接到交易平台页面。



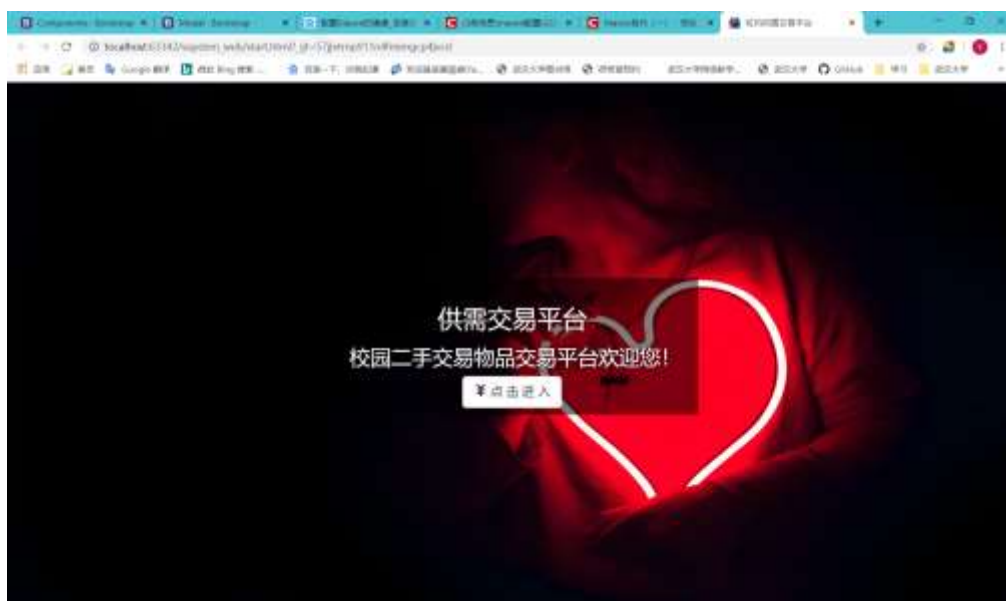


图 20 欢迎页面自动切换背景图片

2. 平台主页

图 21 为平台的主页，左上角为拉出商品列表的按钮，地图上有两种样式的标记，灰色代表该商品已经售出，不可购买，蓝色代表商品还未售出可购买。由于校内的二手交易，没有快递，都是同学们私下交易，路程是在交易中需要考虑的重要因素。标记的位置即为发布二手商品时，在地图中展示商品的位置，可以帮助大家判断商品距离自己的远近。在图 22 中的信息框中，点击相应的标记可以弹出消息框。灰色标记对应左图信息窗口，购物车和购买按钮均不可点击，蓝色标记对应右图信息窗口，购物车和购买按钮可点击。

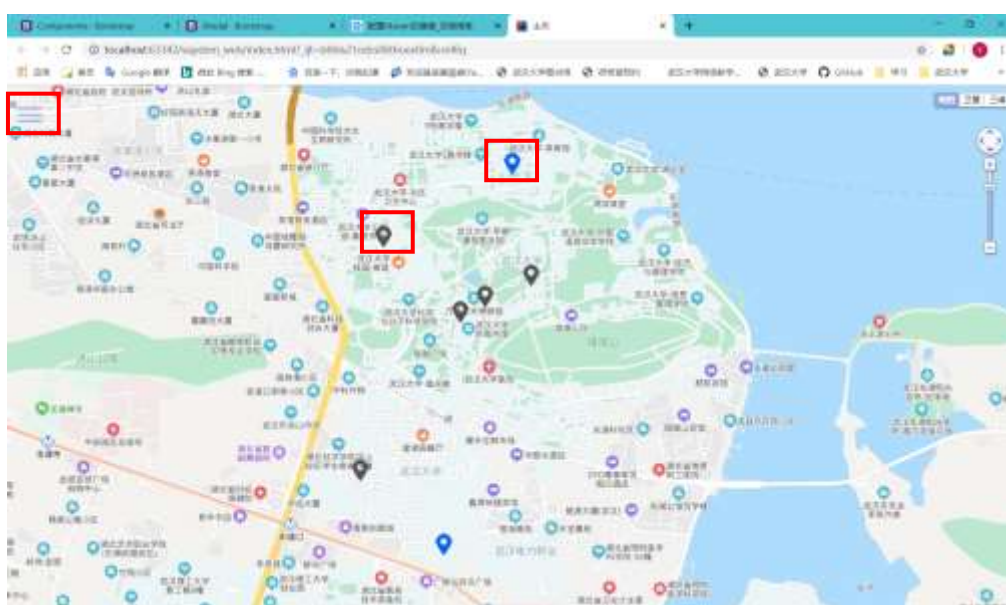


图 21 平台主页



图 22 商品信息弹出框

3. 商量列表

图 23 为展开的商品列表，当鼠标停留在某个商品上时，会以背景颜色突出，下方三个按钮依次代表购物车、购物清单、发布新商品。点击商品列表中的某个商品时，更换原本的图标，以红色小人突出显示用户选中的商品，并自动展示该商品的信息，商品消息框与上一小节中展示的相同，按钮可点击则可购买，反之则反。

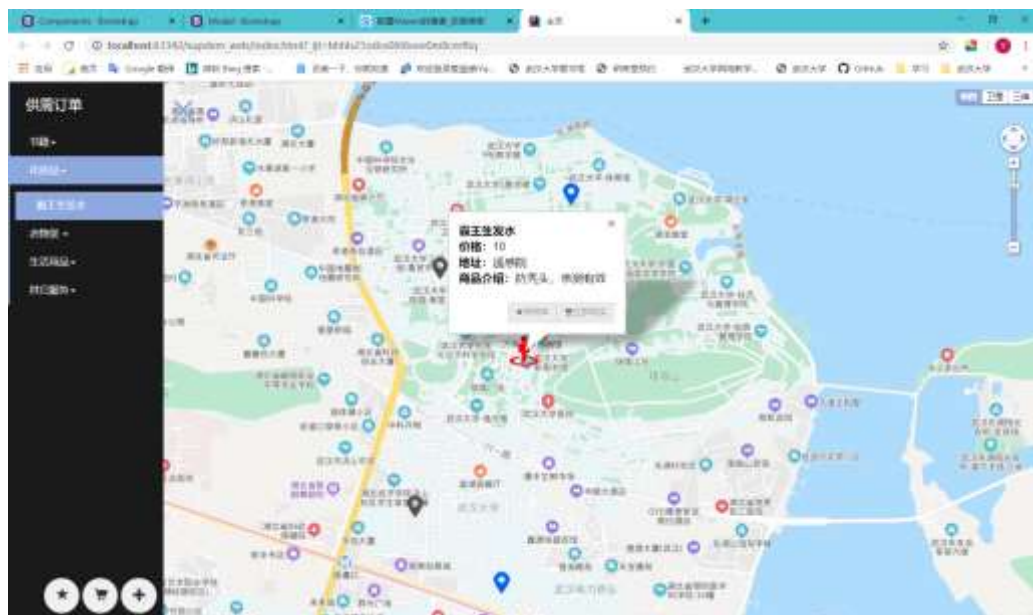


图 23 商品列表

4. 购物车

点击榨汁机加入购物车，平台自动刷新。打开购物车表格（即愿望清单），可以看到新加入的商品，背景色为黄色，代表商品正在销售中。用户可以继续选择从购物车中删除该商品或购买该商品，若选择购买则商品会被加入到购物清单中，若删除则从愿望清单中删除该商品。



图 24 加入购物车

5. 购物清单

用户有两种购买商品的接口，一是从地图上点击商品后直接购买；二是从购物车中购买，此处展示从地图中点击“购买”来购买商品，此时愿望清单和购物清单中均有“榨汁机”这个商品（商品可以同时存在在愿望清单和购物清单中，但是同种商品不能同时加入愿望清单或购物清单）。所有刚购买的商品状态都为“发货中”，背景色为黄色。当交易双方碰面后，买家可以选择收货或退货，收货和退货只能点击一次，点击后按钮不可以再次点击。退货和收货后购物清单的状态如图 25 所示，收货为绿色背景，退货为红色背景。

当选择收货时，商品不再可销售，因此平台同步更新了地图中的标记和愿望清单的数据。点击榨汁机收货，如图 26 所示，在地图上商品以红色小人定位，“购物车”和“购买”按钮不可点击，当选择其它商品时，该标记更新为灰色，代表商品不可购买。在愿望清单中状态显示为售罄，背景色为红色，购买按钮不可点击，删除按钮可点击。



图 25 购物清单



图 26 商品收货同步更新状态

6. 添加商品

发布新的二手物品进行售卖，以表单的形式将数据传递给数据库。表单中，用户需要输入以下商品的描述信息，也可以选择空缺。其中地址信息若为填写，则自动定位到武汉大学校园中心为默认的交易地点。图 28 和图 29 展示了数据库和地图中同步更新的商品信息。

The figure shows a '发布供需订单' (Post Supply and Demand Order) form with the following fields and options:

- 商品名称** (Product Name): 网络GIS课间实习
- 价格** (Price): 1000
- 地址** (Address): 武汉大学信息学部温感楼
- 商品类型** (Product Type):
 - ☒ 书籍
 - ☐ 化妆品
 - ☐ 衣物
 - ☐ 生活用品
 - ☐ 其它服务
- 商品描述** (Product Description): 课间实习

At the bottom right are buttons for '取消' (Cancel) and '提交订单' (Submit Order).

图 27 发布商品表单

图 28 数据库同步更新商品列表



图 29 商品列表和地图同步更新

4. 实习体会

本次实习，结合四个任务，完成了“供需交易平台”的开发，学习和掌握了基本网络GIS开发技术。

在开发过程中，最重要的是程序的调试工作。由于涉及数据交换和同步更新，用户每次进行操作时，要同步更新数据库、前端储存数据的变量以及页面的展示效果，并且更改的信息也不止一处，比如在购物清单中购买后，愿望清单和地图标记都要同步更新，很容易出现缺漏，一定要耐心的去调试程序，跟逻辑。

平台也还存在一些缺点有待改善，比如百度API定位识别不准确，如何增加校验功能提高定位精度。发布的商品信息可以添加图片，但目前在MySQL数据库中存储图片的方式还未解决，因此尚未实现。此外基于所有发布过的商品以及商品的类别可以扩展完成更多的分析操作，分析同学们在二手交易中的商品特征。

总而言之，此次实习收货很多，完整的完成了前后端开发，以及数据的更删改查操作，前端对JSON数据的操作。对百度API的浏览器端的和服务端的应用都有使用，对框架的使用和控件排版布置能力也有得到提高。