

## 《互联网应用开发技术》课程大作业 “e-Book 系统” 要求

### 一、功能要求

---

1. 用户管理：管理员身份登录后，可以看到该功能；用户分为两种角色：顾客和管理员
  - 1) 管理员可以禁用/解禁用户，被禁用的用户将无法登录系统
2. 用户登录与注册
  - 1) 用户登录需要输入用户名和密码，未输入点登录按钮时，提示用户必须输入
  - 2) 被禁用用户无法登录系统，并且会提示用户“您的账号已经被禁用”
  - 3) 根据用户名来确认其为管理员还是顾客，不同角色的界面具有差异
  - 4) 新用户注册时需要填写用户名、密码、重复密码、邮箱
  - 5) 需要校验用户名是否重复、两次输入的密码是否相同、邮箱是否符合格式要求
3. 书籍管理：管理员身份登录后，可以看到该功能
  - 1) 管理员可以浏览数据库中已有的书籍，以列表形式显示，包括书名、作者、封面、ISBN 编号和库存量
  - 2) 在列表上方提供搜索功能，管理员可以用书名来过滤想要查找的书籍
  - 3) 管理员在列表中可以修改每本图书的上述各种属性，包括书名、作者、封面、ISBN 编号和库存量
  - 4) 管理员可以删除旧图书，可以添加新图书
4. 浏览书籍：此部分功能可以复用“书籍管理”中的功能
  - 1) 顾客和管理员都可以浏览数据库中已有的书籍，以列表形式显示，包括书名、作者、封面、ISBN 编号和库存量
  - 2) 提供搜索功能，用户可以用书名来过滤想要查找的书籍
  - 3) 选中某本书后，通过 Ajax 方式获取并显示书的详细信息
5. 购买书籍
  - 1) 当用户浏览书籍时，可以选择将某本书放入购物车
  - 2) 用户可以浏览购物车，看到自己放入购物车的所有书籍
  - 3) 在购物车中点击购买书籍之后，清空购物车，同时书籍库存相应地减少
  - 4) 购买书籍后，生成订单，展示给用户，并将订单存入数据库
6. 订单管理
  - 1) 顾客可以查看自己的所有订单，并且可以使用搜索功能来实现过滤，具体可以按照时间范围或书籍名称过滤

- 2) 管理员可以查看系统中所有的订单，并且可以使用搜索功能来实现过滤，具体可以按照时间范围或书籍名称过滤

## 7. 统计

- 1) 管理员可以统计在指定时间范围内各种书的销量情况，按照销售量排序，形成热销榜，以图或表的方式呈现
- 2) 管理员可以统计在指定时间范围内每个用户的累计消费情况，按照购书进行排序，形成消费榜，以图或表的方式呈现
- 3) 顾客可以统计在指定时间范围内自己购买书籍的情况，包括每种书购买了多少本，购书总本数和总金额

## 8. 移动端

- 1) 在微信小程序上实现登录、书籍列表浏览、书籍详情浏览、放入购物车和下订单完整流程
- 2) 在移动端 App 上实现登录、书籍列表浏览、书籍详情浏览、放入购物车和下订单完整流程

# 二、技术实现要求

---

## 1. Web 前端

- 1) React 架构 或 Vue 架构
- 2) 使用 WebPack 或类似工具打包管理

## 2. 移动前端

- 1) 微信小程序
- 2) React 架构、Flutter 架构或类似架构开发的移动端应用

## 3. 后端

- 1) 使用 Spring 来实现(Spring MVC/Bootstrap/ORM) or 使用类似框架实现
- 2) 使用 Maven 或类似工具打包管理

## 4. 开发工具

- 1) IntelliJ / WebStorm
- 2) MySQL & MongoDB
- 3) Android Studio & WeChat Studio

# 三、迭代评分标准

---

1. 第 1 次迭代：15 分，运用 HTML、CSS 和 React/Vue 开发前端静态页面的网站
  - 1) 编写完整的网站静态 HTML 页面：5 分
    - a) 你应该编写整个网站的页面, 如果将来计划使用 React 或 Vue 框架来生成页面中的元素, 那么你也应该将页面框架构建好, 例如包含登录页面、网站首页、书籍浏览页、书籍详情页、下订单页等。
  - 2) 运用 CSS 调整页面样式：3 分
    - a) 你应该使用外部自定义的 CSS
    - b) 你可以使用现成框架的 CSS, 不需要自己完全重新编写, 但是你要注意你引用不同的 CSS 时存在的冲突问题。
  - 3) React 或 Vue 框架开发书籍浏览功能：5 分
    - a) 你至少应该使用 React 或 Vue 这两个框架中的一种, 在你给出的 HTML 页面上开发像上课示例中所示的图书浏览、搜索和编辑的功能。你选中的框架将作为你后续开发基础。
  - 4) React 或 Vue 框架开发规范：2 分
    - a) 你的前端工程应该体现出构件化开发的思想。
2. 第 2 次迭代：20 分，运用 Java 的 ORM 框架，开发前后端集成的网站
  - 1) 数据库设计：5 分
    - a) 库结构和表结构合理，表与表之间的关系设计合理
    - b) 表中主键设计合理，字段类型设计合理
  - 2) 功能完备：5 分，至少应实现普通用户(非管理员)的下列功能
    - a) 登录：用户使用存储在数据库中的用户名和密码登录
    - b) 书籍列表主页：以列表形式展示所有存储在数据库中的书籍，具体格式参照上课给的示例
    - c) 书籍详情：点击书籍列表中的每一本书籍，进入该书籍的详情页
    - d) 加入购物车：在书籍详情页上添加加入购物车的功能，购物车信息可以存放数据库，也可以在内存中保存
    - e) 下订单：在浏览购物车页面时，可以下订单，完成交易
  - 3) 前后端集成：5 分
    - a) 前端页面操作直接反映到数据库中，即前后台集成，数据并非在前端页面写死
    - b) 使用你选择的前端框架的路由功能实现各个页面之间的跳转逻辑的控制
  - 4) 框架运用：5 分

- a) 应该使用 ORM 框架来访问数据库，并且 OR 映射的方式合理
- 3. 第 3 次迭代：25 分，运用 Java 的 IoC 框架，以及 Ajax 和 JSON 开发前后端集成的网站，网站架构合理
  - 1) Ajax 和 JSON 前后端集成：5 分
    - a) 运用 Ajax 和 JSON 实现前后端 HTTP 请求和响应的发送，以及响应数据的传递
  - 2) 系统架构：5 分
    - a) 后端工程应该体现出合理的分层架构，包括控制层、服务层、数据访问层、实体层等
    - b) 后端工程要体现接口与实现分离的原则
  - 3) 功能完备：10 分，至少应实现下列管理员功能，具体功能描述见第一节
    - a) 用户管理
    - b) 书籍管理
    - c) 订单管理
    - d) 统计
- 4. 第 4 次迭代：30 分，实现微信和手机前端应用程序
  - 1) 微信小程序：8 分，至少应实现普通用户(非管理员)的下列功能
    - a) 登录：用户使用存储在数据库中的用户名和密码登录，后台对用户进行认证
    - b) 书籍列表主页：以列表形式展示所有存储在数据库中的书籍，具体格式参照上课给的示例
    - c) 书籍详情：点击书籍列表中的每一本书籍，进入该书籍的详情页
    - d) 加入购物车：在书籍详情页上添加加入购物车的功能，购物车信息可以存放数据库，也可以在内存中保存
    - e) 下订单：在浏览购物车页面时，可以下订单，完成交易
    - f) 页面路由逻辑正确：能够在上述页面之间进行堆栈式路由
  - 2) 移动端 App：8 分，至少应实现普通用户(非管理员)的下列功能
    - a) 登录：用户使用存储在数据库中的用户名和密码登录，后台对用户进行认证
    - b) 书籍列表主页：以列表形式展示所有存储在数据库中的书籍，具体格式参照上课给的示例
    - c) 书籍详情：点击书籍列表中的每一本书籍，进入该书籍的详情页
    - d) 加入购物车：在书籍详情页上添加加入购物车的功能，购物车信息可以存放数据库，也可以在内存中保存

- e) 下订单：在浏览购物车页面时，可以下订单，完成交易
  - f) 页面路由逻辑正确：能够在上述页面之间进行堆栈式路由
- 3) 技术方案：10 分，下列各考核点针对整体工程，如只有个别功能点对应代码符合要求，而整体工程未全部遵循下列原则，则不能得分
- a) 后端系统架构：2 分，分层合理(1 分)，接口与实现分离(1 分)
  - b) 后端框架运用：2 分，ORM 运用合理(1 分)，依赖注入合理(1 分)
  - c) 后端数据存储：2 分，关系型数据库设计合理(1 分)，表结构设计合理(1 分)
  - d) 浏览器前端系统架构：2 分，构件化设计合理(1 分)，正确处理数据获取和状态更新(1 分)
  - e) 移动端系统架构：2 分，构件化设计合理(1 分)，正确处理数据获取和状态更新(1 分)
- 4) 界面友好：4 分，符合一般电子商务网站的操作习惯(2 分)，界面友好美观(2 分)

#### 四、平时作业评分

---

每周布置一次作业，除答辩验收外，共 10 次作业，每次 1 分，共 10 分，