

目录

| | |
|---|----|
| 梦呓单词..... | 2 |
| 1 项目介绍..... | 2 |
| 1.1 项目背景..... | 2 |
| 1.2 项目定义..... | 2 |
| 1.3 参考资料..... | 2 |
| 2 需求概述..... | 3 |
| 2.1 总用例图..... | 3 |
| 2.1.1 用户注册 | 4 |
| 2.1.2 用户登录 | 5 |
| 2.1.3 选择单词书并制定背诵计划..... | 6 |
| 2.1.4 背诵新词 | 7 |
| 2.1.5 复习单词 | 8 |
| 2.1.6 定时推送通知提醒..... | 9 |
| 2.1.7 单词考试 | 10 |
| 2.2 条件与限制..... | 11 |
| 3 总体设计..... | 11 |
| 3.1 软件结构..... | 11 |
| 3.2 功能设计..... | 12 |
| 4 接口设计..... | 17 |
| 4.1 服务器 API 接口 | 17 |
| 4.2 数据库表结构..... | 18 |
| 4.2.1 User..... | 18 |
| 4.2.2 Word..... | 18 |
| 4.2.3 Learnt..... | 18 |
| 4.2.4 Love | 18 |
| 5 服务端逻辑..... | 19 |
| 5.1 总体逻辑图..... | 19 |
| 5.2 关键代码分析..... | 20 |
| 5.2.1 登录拦截器..... | 20 |
| 5.2.2 实体 Bean (以 User 为例)..... | 21 |
| 5.2.3 dao 层 (数据库交互) | 21 |
| 5.2.4 控制器@Controller(以 review 界面为例) | 22 |
| 6 开发结果..... | 23 |
| 6.1 产品..... | 23 |
| 6.1.1 梦呓单词网站系统..... | 23 |
| 6.1.2 Word Bank | 24 |
| 6.2 主要功能..... | 24 |
| 6.2.1 选择单词书并制定背诵计划..... | 24 |
| 6.2.2 查看当日计划背诵的单词 | 25 |
| 6.2.3 通过看英文选中文的方式复习单词 | 25 |
| 6.2.4 用户注册 | 25 |
| 6.2.5 用户登录并将学习进度与网站同步 | 25 |
| 6.2.6 使用图表展现用户学习历史..... | 25 |

梦呓单词

1 项目介绍

1.1 项目背景

如今背单词已经成为许多人的日常，开发一给好用顺手的背单词 WEB 系统能大大提高人们背诵单词的效率。另外，在移动端进行单词背诵，首先能利用用户的碎片化时间进行学习，比如在坐公交时、在餐厅等菜时等等。其实，用户还能在应用中定制自己的学习计划、系统也能定时推送复习通知，这都大大提高了用户的学习效率。此外，用户还能通过复习考核的功能巩固自己已学知识。在给予用户成就感的同时还能促进他们的学习兴趣。因此，开发一款功能相对完备的背诵单词网站系统和 APP 是十分有意义的。

在这一大背景下，梦呓单词应运而生。本系统的开发意义便是提供以用户一个爱不释手的背单词网站系统，提高用户的学习效率，增加用户的学习兴趣以达到提升其英语水平的目的。

1.2 项目定义

该系统使用 Spring Boot 框架进行开发，使用 Maven 进行包管理，整合使用了模板引擎 Thymeleaf，持久层框架 Mybais，PageHelper 等插件，数据库使用的是 Mysql80。

该系统的定位是一款学习类的网站系统，提供用户丰富的功能以提高其学习英语的兴趣和效率。

1.3 参考资料

[1] Spring Boot: <http://spring.io/projects/spring-boot>

[2] Thymeleaf: <https://www.thymeleaf.org/>

[3] Mybais: <http://www.mybatis.org/mybatis-3/>

2 需求概述

2.1 总用例图

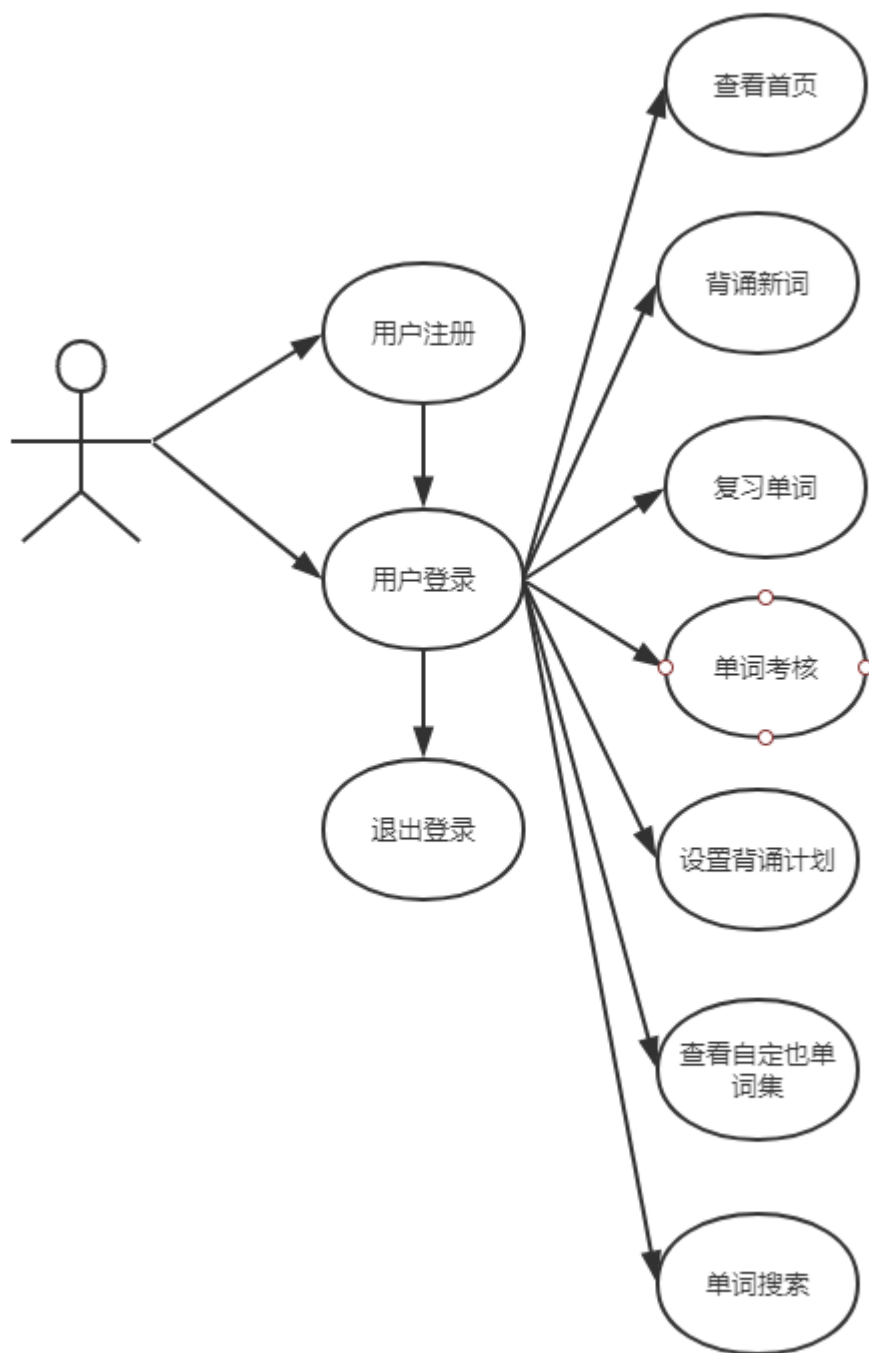


图 1 总体用例图

2.1.1 用户注册

表格 1 用户注册用例

| | | | |
|-------|--|------|------|
| 用例编号 | UC--01 | 用例名称 | 用户注册 |
| 角色 | 用户 | 需求来源 | 用户代表 |
| 主要参与者 | 用户 | | |
| 描述 | 进入网页后，通过注册选项注册自己的账号 | | |
| 前置条件 | 1. 数据库正常运行 2. 系统正常运行 | | |
| 触发器 | 点击注册按钮 | | |
| 后置条件 | 用户的注册记录写入数据库 | | |
| 输入 | 1. 用户名 2. 密码 3. 注册邮箱 | | |
| 主干过程 | 1. 选择设置 2. 在设置页面选择注册 3. 输入注册信息 | | |
| 分支过程 | 无 | | |
| 异常 | 1. 用户名已存在 处理：系统提示错误 “用户名已存在” 2. 两次输入密码不匹配 处理：系统提示错误 “两次输入密码不匹配” 系统结束用例 | | |
| 输出 | 页面跳转至首页 | | |
| 包括用例 | 无 | | |
| 使用频率 | 低 | | |
| 假设 | 无 | | |
| 备注与问题 | 无 | | |

2.1.2 用户登录

表格 2 用户登录用例

| | | | |
|-------|--|------|------|
| 用例编号 | UC--02 | 用例名称 | 用户登录 |
| 角色 | 用户 | 需求来源 | 用户代表 |
| 主要参与者 | 用户 | | |
| 描述 | 进入网页后，通过登录选项登录自己的账号 | | |
| 前置条件 | 1. 数据库正常运行 2. 系统正常运行 | | |
| 触发器 | 点击登录按钮 | | |
| 后置条件 | 用户的登录记录写入服务端端记录日志 | | |
| 输入 | 1. 用户名 2. 密码 | | |
| 主干过程 | 1. 选择设置 2. 在设置页面选择登录 3. 输入用户名和密码 | | |
| 分支过程 | 无 | | |
| 异常 | 1. 用户名不存在 处理：系统提示错误 “用户名不存在” 2. 密码错误 处理：系统提示错误 “密码错误” 系统结束用例 | | |
| 输出 | 页面跳转至首页 | | |
| 包括用例 | 无 | | |
| 使用频率 | 高 | | |
| 假设 | 无 | | |
| 备注与问题 | 无 | | |

2.1.3 选择单词书并制定背诵计划

表格 3 选择单词书并制定背诵计划用例

| | | | |
|-------|------------------------|------|------|
| 用例编号 | UC--03 | 用例名称 | 制定计划 |
| 角色 | 用户 | 需求来源 | 用户代表 |
| 主要参与者 | 用户 | | |
| 描述 | 登录后，制定背诵计划 | | |
| 前置条件 | 1. 用户已登录 | | |
| 触发器 | 点击修改计划 | | |
| 后置条件 | 在服务端数据库更新用户的背诵计划 | | |
| 输入 | 1. 选择单词书 2. 每日背诵单词数 | | |
| 主干过程 | 1. 选择设置 2. 选择修改计划 | | |
| 分支过程 | 无 | | |
| 异常 | 无 | | |
| 输出 | 页面跳转至首页 | | |
| 包括用例 | 无 | | |
| 使用频率 | 低 | | |
| 假设 | 无 | | |
| 备注与问题 | 无 | | |

2.1.4 背诵新词

表格 4 背诵新词

| | | | |
|-------|---------------------------|------|------|
| 用例编号 | UC--04 | 用例名称 | 背诵新词 |
| 角色 | 用户 | 需求来源 | 用户代表 |
| 主要参与者 | 用户 | | |
| 描述 | 登录后，进行背诵单词 | | |
| 前置条件 | 用户已登录 | | |
| 触发器 | 进入背诵页面 | | |
| 后置条件 | 用户的背诵记录写入服务端数据库 | | |
| 输入 | 无 | | |
| 主干过程 | 1. 选择背诵 2. 进行背诵 | | |
| 分支过程 | 无 | | |
| 异常 | 未选择背诵计划 系统提示：“未选择背诵计划” | | |
| 输出 | 无 | | |
| 包括用例 | 无 | | |
| 使用频率 | 高 | | |
| 假设 | 无 | | |
| 备注与问题 | 无 | | |

2.1.5 复习单词

表格 5 复习单词用例

| | | | |
|-------|---------------------------|------|------|
| 用例编号 | UC--05 | 用例名称 | 复习单词 |
| 角色 | 用户 | 需求来源 | 用户代表 |
| 主要参与者 | 用户 | | |
| 描述 | 登录后，进行复习单词 | | |
| 前置条件 | 用户已登录 用户背诵过单词 | | |
| 触发器 | 在复习页面点击开始测验 | | |
| 后置条件 | 用户的复习记录写入服务端端记录日志 | | |
| 输入 | 点击单词选项 | | |
| 主干过程 | 1. 选择复习 | | |
| 分支过程 | 无 | | |
| 异常 | 不存在复习单词 系统提示：“已完成所有复习” | | |
| 输出 | 无 | | |
| 包括用例 | 无 | | |
| 使用频率 | 高 | | |
| 假设 | 无 | | |
| 备注与问题 | 无 | | |

2.1.6 定时推送通知提醒

表格 6 定时提醒用例

| | | | |
|-------|----------------------|------|------|
| 用例编号 | UC--06 | 用例名称 | 定时提醒 |
| 角色 | 用户 | 需求来源 | 用户代表 |
| 主要参与者 | 用户 | | |
| 描述 | 为用户定时推送通知，提醒其进行今日的背诵 | | |
| 前置条件 | 用户已制定背诵计划 | | |
| 触发器 | 到达时间点并且用户未背诵 | | |
| 后置条件 | 无 | | |
| 输入 | 无 | | |
| 主干过程 | 1. 应用通知用户进行今日的计划 | | |
| 分支过程 | 无 | | |
| 异常 | 无 | | |
| 输出 | 无 | | |
| 包括用例 | 无 | | |
| 使用频率 | 高 | | |
| 假设 | 无 | | |
| 备注与问题 | 无 | | |

2.1.7 单词考试

表格 7 定时提醒用例

| | | | |
|-------|------------------------------------|------|------|
| 用例编号 | UC--07 | 用例名称 | 单词考试 |
| 角色 | 用户 | 需求来源 | 用户代表 |
| 主要参与者 | 用户 | | |
| 描述 | 登录后，进行单词考试 | | |
| 前置条件 | 用户已进行过背诵单词 | | |
| 触发器 | 点击考试 | | |
| 后置条件 | 无 | | |
| 输入 | 无 | | |
| 主干过程 | 1. 点击进行考试 2. 输入正确答案 3. 点击下一题 | | |
| 分支过程 | 无 | | |
| 异常 | 无 | | |
| 输出 | 1. 答案错误 系统提示：“答案错误” | | |
| 包括用例 | 无 | | |
| 使用频率 | 高 | | |
| 假设 | 无 | | |
| 备注与问题 | 无 | | |

2.2 条件与限制

需要实现的基本功能如下：

实现用户注册、登录功能，用户注册时需要填写必要的信息并验证，如用户名、密码要求在 6 字节以上，email 的格式验证，并保证用户名和 email 在系统中唯一。

用户登录后可以设置需要背的单词集，如 4 级、6 级等。单词集可以从网上收集，数量多少不影响评分。

用户可以维护自己的自定义单词

实现基本的背诵计划、复习、考核等功能，记录进度。

界面样式需要适配 PC 和手机的浏览器

增强功能：

实现一个 Android 或 iphone 客户端软件，功能同网站，支持离线使用，并能实现背诵计划的通知提醒

具体一定的学习能力，能根据记忆曲线或用户的使用习惯调整背诵的内容，此项功能在界面上表现不明显时，可以在文档中详细说明

为了提交作业方便，如有数据库，建议使用 mysql 或 mangodb，提交作业时同时附带 SQL 脚本文件。

3 总体设计

3.1 软件结构

| | |
|------|-------------------|
| 开发环境 | Mac OS(Win 10) |
| 开发语言 | Java(Spring boot) |
| 服务器 | Tomcat |
| 数据库 | Mysql |

3.2 功能设计

在需求概述的基础上，对功能进行阐述：

(1) 用户注册

通过该功能创建用户账号。

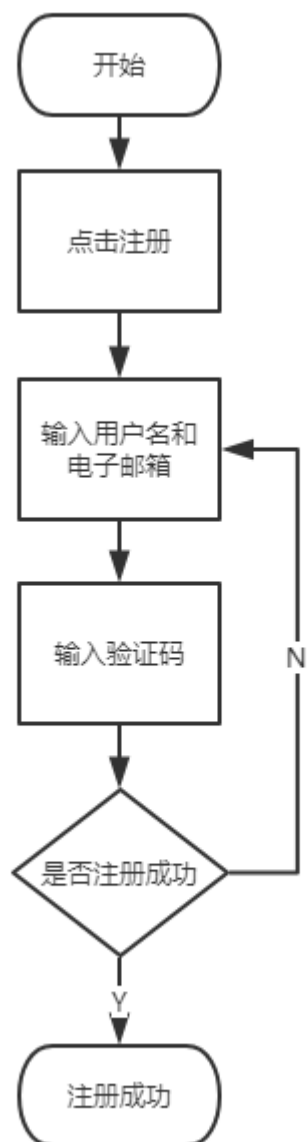


图 2 用户注册流程图

(2) 用户登录

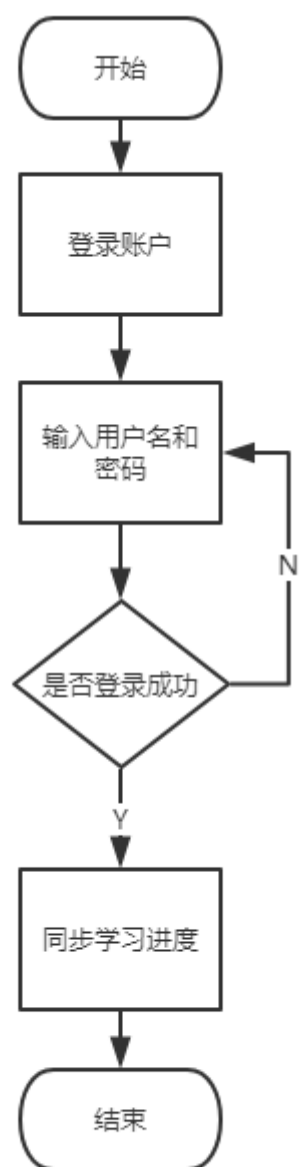


图 2 用户登录流程图

(3) 选择背诵计划（包括四级、六级、托福、雅思等）

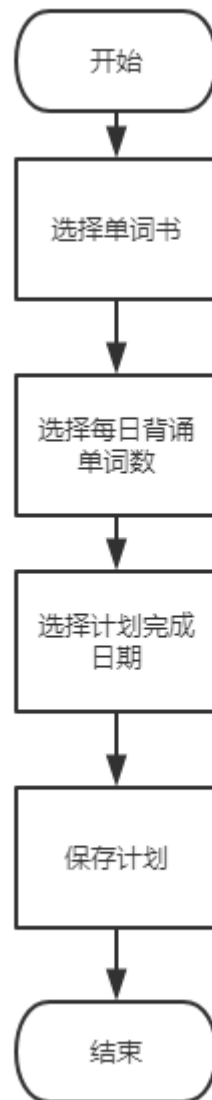


图 3 选择背诵计划流程图

(4) 进行每日的单词背诵

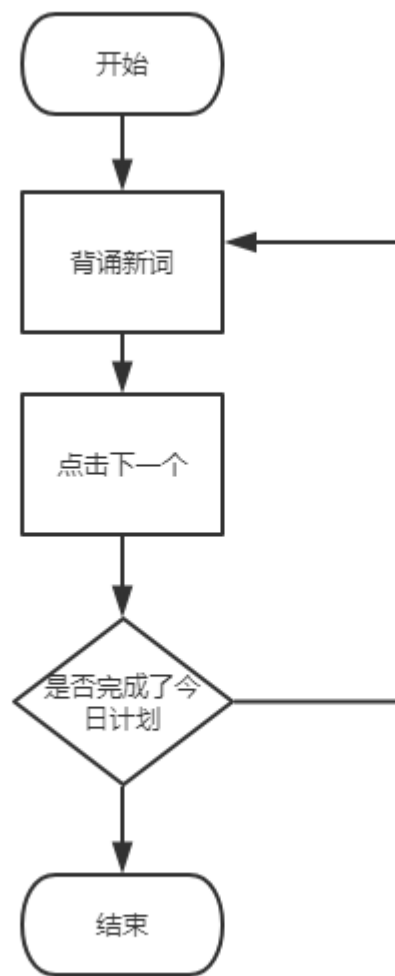


图 4 背诵新词流程图

(5) 复习单词

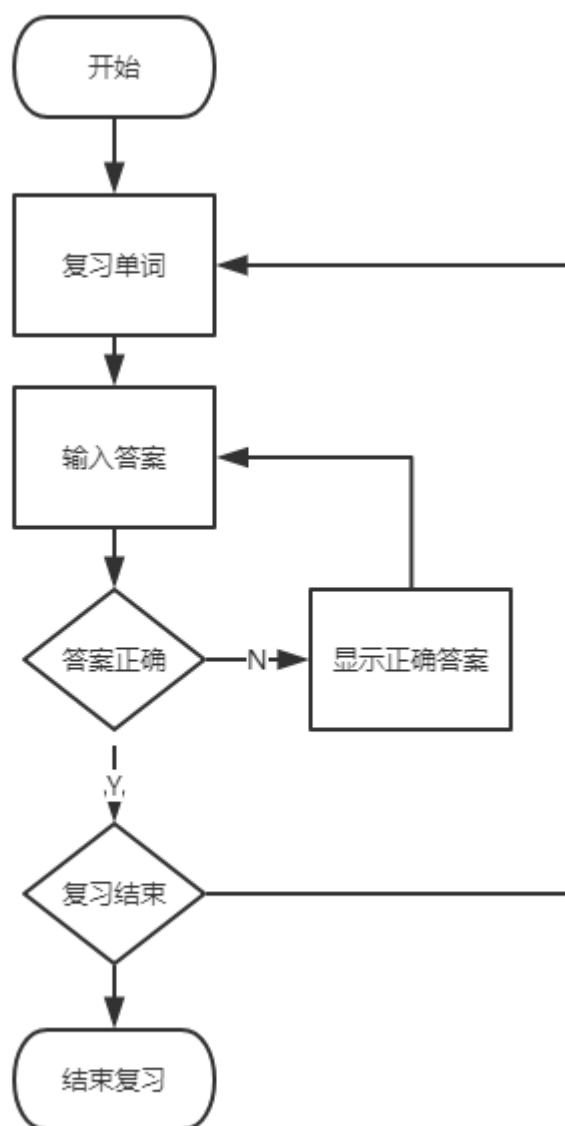


图 5 复习单词流程图

4 接口设计

4.1 服务器 API 接口

服务器 API 接口设计如下：

| 接口 | 方法 | 备注 |
|---------------|------|-----------|
| register | POST | 注册新账户 |
| login | POST | 登录账户 |
| guide | GET | 设置向导界面 |
| learntype | POST | 修改背诵计划 |
| word | GET | 单词显示页面 |
| review | GET | 显示复习单词 |
| exam | GET | 单词考核页面 |
| love | GET | 显示收藏单词 |
| search | GET | 搜索单词 |
| nextlearnword | GET | 获取新的背诵单词 |
| examget | GET | 获取下一个考核单词 |
| knownget | GET | 获取下一个复习单词 |
| addlove | POST | 添加自定义单词 |
| deletelove | POST | 删除自定义单词 |

4.2 数据库表结构

4.2.1 User

| 字段名 | 类型 | 意义 | 限制 |
|------------|---------|---------|----|
| username | varchar | 用户名 | 主键 |
| password | varchar | 密码 | 非空 |
| email | varchar | 邮箱 | 非空 |
| level | varchar | 单词集 | |
| num | int | 每日学习单词数 | |
| havelearnt | int | 今日已学单词数 | |

4.2.2 Word

| 字段名 | 类型 | 意义 | 限制 |
|---------|---------|-------|----|
| ID | int | 单词 ID | 主键 |
| word | varchar | 单词英文 | 非空 |
| meaning | varchar | 中文解释 | |
| lx | varchar | 例句 | |

4.2.3 Learnt

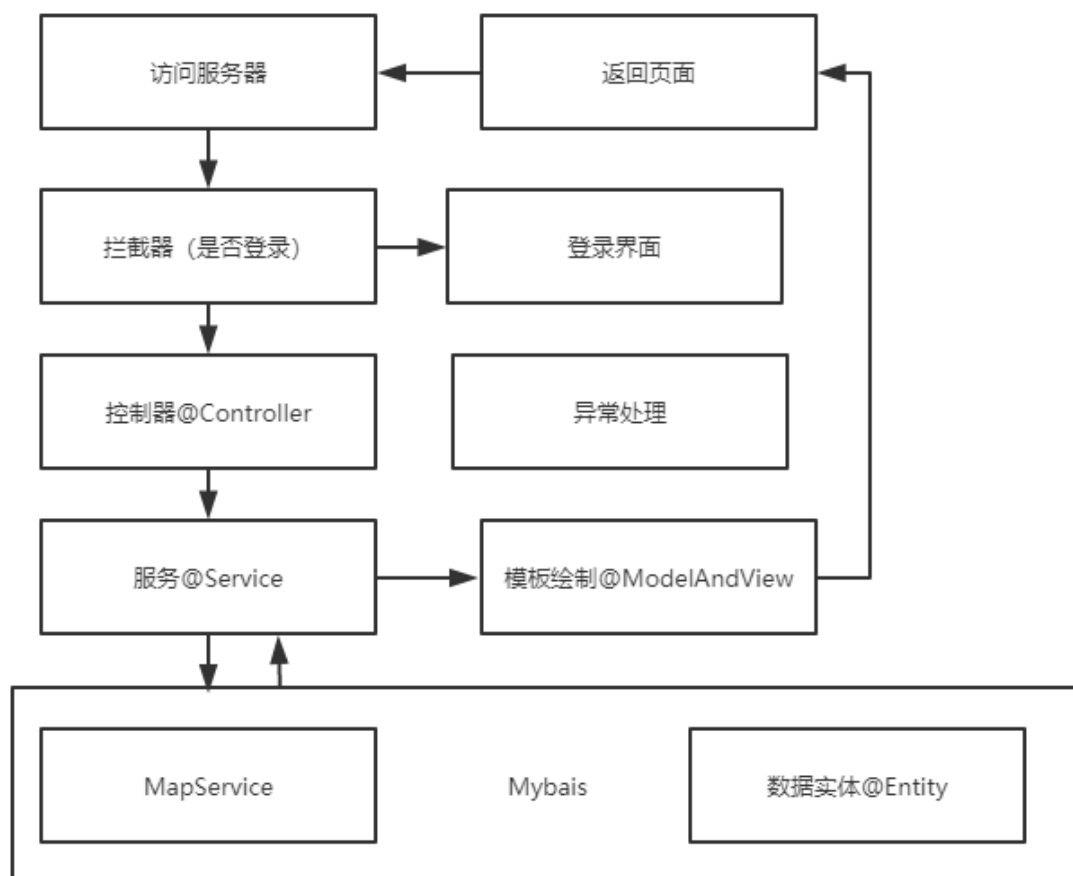
| 字段名 | 类型 | 意义 | 限制 |
|----------|---------|-----------|----|
| username | varchar | 用户名 | 外键 |
| ID | varchar | 单词 ID | 外键 |
| known | int | 复习时认识的次数 | |
| unknown | int | 复习时不认识的次数 | |

4.2.4 Love

| 字段名 | 类型 | 意义 | 限制 |
|----------|---------|-------|----|
| username | varchar | 用户名 | 外键 |
| ID | varchar | 单词 ID | 外键 |

5 服务端逻辑

5.1 总体逻辑图



5.2 关键代码分析

5.2.1 登录拦截器

```
public class LoginInterceptor implements HandlerInterceptor {  
    @Override  
    public boolean preHandle(HttpServletRequest request, HttpServletResponse  
response, Object handler) throws Exception {  
        boolean flag =true;  
        User user=(User)request.getSession().getAttribute("user");  
        if(null==user){  
            response.sendRedirect("/");  
            flag = false;  
        }else{  
            flag = true;  
        }  
        return flag;  
    }  
  
    @Override  
    public void postHandle(HttpServletRequest request, HttpServletResponse  
response, Object handler, ModelAndView modelAndView) throws Exception {  
    }  
  
    @Override  
    public void afterCompletion(HttpServletRequest request, HttpServletResponse  
response, Object handler, Exception ex) throws Exception {  
  
    }  
}
```

判断用户是否已经登录，若未登录，则重定向到登录界面。

5.2.2 实体 Bean (以 User 为例)

```
public class User {  
  
    private String username;  
    private String password;  
    private String email;  
    private String level;  
    private int num;  
  
    public int getHavelearnt() {  
        return havelearnt;  
    }  
  
    public void setHavelearnt(int havelearnt) {  
        this.havelearnt = havelearnt;  
    }  
    .....  
}
```

5.2.3 dao 层 (数据库交互)

以 WordMapper 为例:

```
@Component  
@Mapper  
public interface WordMapper {  
    @Select("select ID, Word, meaning, lx from user natural join learnt natural  
join words where username = #{username} order by  
known/(known+unknown)+(known+unknown) limit 1")  
    Word findLearntByName(String username);  
  
    @Select("select ID, Word, meaning, lx from user natural join learnt natural  
join words where username = #{username} order by rand() limit 1")  
    Word findLearntByNameRandom(String username);  
  
    @Select("select ID, Word, meaning, lx from user natural join love natural join  
words where username = #{username}")  
    List<Word> findLoveByName(String username);  
  
    @Insert("insert into learnt(username, ID) values(#{username},#{ID})")  
    int addLearnt(@Param("username")String username, @Param("ID")int ID);  
    ...  
}
```

5.2.4 控制器@Controller(以 review 界面为例)

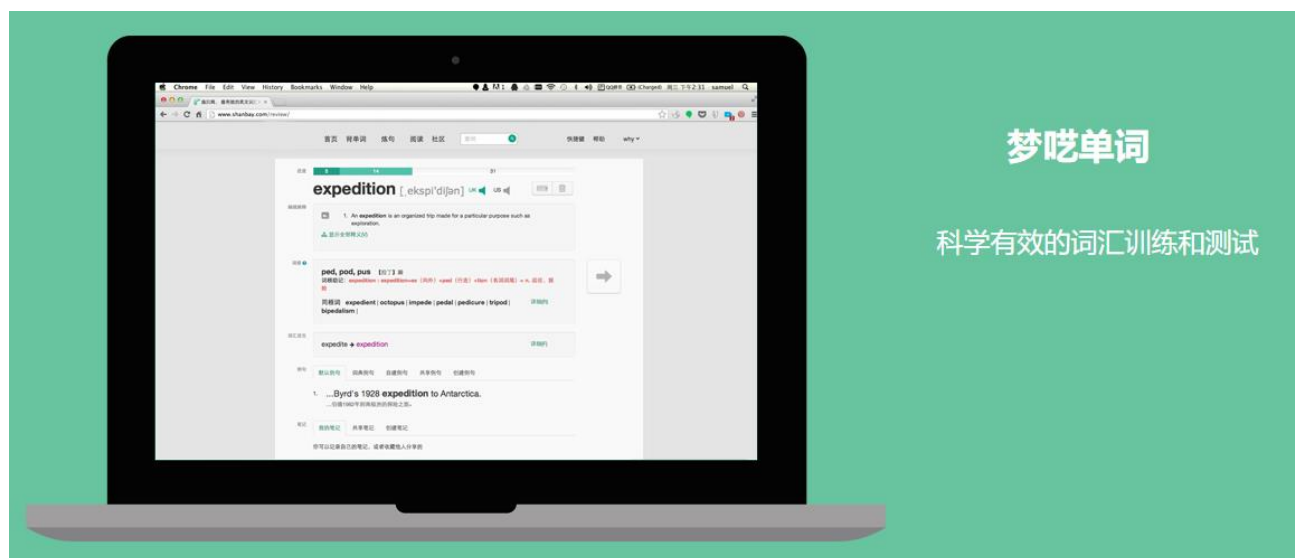
先通过 wordservices 在数据库中获取用户的复习单词，再通过 Thymeleaf 绘制模板并返回。

```
@RequestMapping("review")
public ModelAndView review(HttpServletRequest request, HttpServletResponse
response){
    ModelAndView modelAndView = new ModelAndView("/learnword/review");
    User user =(User)request.getSession().getAttribute("user");
    modelAndView.addObject("user",user);
    Word word = wordService.findLearntByName(user.getUsername());
    if (word==null)
        return modelAndView;
    request.getSession().setAttribute("reviewID",word.getID());
    String[] lx=word.getLx().split("/r/n");
    if(lx.length<2) {
        lx = new String[2];
        lx[0]=" ";
        lx[1]=" ";
    }
    modelAndView.addObject("word",word);
    modelAndView.addObject("lx",lx);
    return modelAndView;
}
```

6 开发结果

6.1 产品

6.1.1 梦呓单词网站系统



产品名称：梦呓单词

产品类型：单词学习网站

产品简介：梦呓单词是一款多功能集成的、智能的英语单词学习网站。只要您挑选好单词书，制定好背诵计划，它就可以按计划每日对您进行监督与提示，每日为您推送生词，帮您复习熟词，甚至帮助您进行自我测试，您的所有成绩和学习记录都可以全网同步，能让您一目了然地感受到自己的进步，梦呓单词，让您在不知不觉中，学会英语。

产品开发者：薛圣杰

产品完成时间：2018 年 6 月

6.1.2 Word Bank



图 “Word Bank” 图标

产品名称：Word Bank

产品类型：多功能智能英语单词学习 app

产品简介：Word Bank 是一款多功能集成的、智能的英语单词学习 app。只要您挑选好单词书，制定好背诵计划，它就可以按计划每日对您进行监督与提示，每日为您推送生词，帮您复习熟词，甚至帮助您进行自我测试，您的所有成绩和学习记录都可以全网同步，图表化的呈现，也能让您一目了然地感受到自己的进步，Word Bank，让您在不知不觉中，学会英语。

产品开发者：赖家乐，薛圣杰

产品完成时间：2018 年 6 月

备注：该 APP 为我和赖家乐同学共同开发完成，赖家乐同学为主程序员，负责 APP 整体的架构构建、开发，我作为副程序员进行协助开发。该 APP 使用 React-Native 进行开发，因此能跨平台进行使用，同时支持 Android 和 IOS 系统。

6.2 主要功能

主要功能如下：

6.2.1 选择单词书并制定背诵计划

后端数据库中存储有多本单词集，比如四级考试英语、高考英语等，这些单词集任君选择，选择后，可以针对性地进行背诵计划的设定，设定每日的背诵个数，系统会自动计算出需要的时间以及完成计划的日子。之后应用就会根据计划，每日给用户推送合适的单词进行学习。

6.2.2 查看当日计划背诵的单词

根据制定的计划，应用会推送数量固定的不同单词给用户进行学习，学习中断后，也会及时保存用户的学习情况，可以随时进行单词的浏览和查询，保证每天的学习量合适并且不重不漏。

6.2.3 通过看英文选中文的方式复习单词

为了增强学习的趣味性和互动性，避免枯燥的死记硬背，应用还特地设计了选择题的形式，呈现英语，让用户选择中文解释，进行自我测验，从而在成就感和反思中学习。

6.2.4 用户注册

每个用户都有一个独立的账号，每个账号都有独立的学习计划，用户可以通过电子邮箱进行验证后完成注册。

6.2.5 用户登录并将学习进度与网站同步

用户根据自身账号登录以后，会立刻将后端数据库的用户信息同步过来，并实现多端同步，包括网站、app、后端等，每时每刻数据都一致。

6.2.6 使用图表展现用户学习历史

对于用户每天的学习成果都有记录，用户可以随时查看自己一直以来的学习情况数据，并且可以通过图表的方式直观展现，让用户清晰地了解到自己的进步与不足。

7 总结与体会

整体的开发流程是比较充实的，里面涉及的内容和心得也不少。

在写该网站系统前，首先要考虑的问题是应该使用什么后端语言和框架来写。应该先前使用过 PHP 进行过几次的网页开发，所以如果使用 PHP 进行开发的话肯定会得心应手许多。但是最好还是使用了没有接触过的 JAVA Spring Boot 进行开发。

Spring Boot 给我的感受就是功能强大、使用方便。功能强大毋庸置疑，Spring Boot 支持各种第三方框架、插件，如 Mybatis、Thymeleaf 等等。并且，比起 Spring 繁杂的配置文件操作，Spring Boot 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。

但有一点值得吐槽的是，thymeleaf 模板引擎是真的很严格，任何标签，包括、<meta>、<link>等标签都必须有对应的闭合标签匹配，否则就会各种报错。

在前端的设计上，其实还是显得挺粗糙的。缺少前端设计的经验，第一次做的网页美观还是不足。在以后的工作中，还是应该对这一块进行改进。

在移动端 APP，Word Bank 的开发上，在赖家乐同学带领开发下，完成了一项功能相对完备的背单词 APP 软件。整个项目过程中，对 React Native 的印象还是十分深刻的。虽然 React Native 方案本身的性能肯定不如原生的开发，但就目前来说，我们还没有看出有多大性能的差别，并且开发成本降低了很多。可以说是实现了 Learn once, write anywhere 和 Write Once, run anywhere。所以从性价比来说，这绝对是一个优秀的解决方案，尤其是对于一些追求高效开发、高速开发的业务场景。