

上海大学

计算机工程与科学学院

计算机应用大型作业

实践报告



课题名称： Beist 背单词系统

组长： 学号 14122342 姓名 张艳丽 成绩：

组员： 学号 14122431 姓名 朱寒燕 成绩：

学号 14122340 姓名 金昞昞 成绩：

学号 14122458 姓名 王琳琳 成绩：

指导教师签名： _____

2017 年 9 月

目录

第一章 目的、要求和内容	1
第二章 详细设计	2
第三章 程序模块	6
第四章 测试用例	18
第五章 用户手册	29
第六章 个人体会	
组长：张艳丽	31
组员：金昞昀	33
王琳琳	34
朱寒燕	35

第一章 目的、要求和内容

1. 目的

计算机应用大型作业是整个计算机教学过程中的一个实践环节，目的是使学生巩固知识、增强能力、提高素质，促进学生的全面发展。通过计算机应用大型作业这一实践环节，培养学生系统回顾检查系列课程学习的基本理论、基本知识，使所学的理论知识系统化、网络化、融会贯通，培养学生独立思考、独立分析、独立解决问题的能力，通过作业小组成员的集体合作，培养学生合作共事、协同工作的能力，为学生今后更好地胜任计算机应用环境的开发打好扎实的基础。

2. 要求

- 1) 学生 3-5 人自由组合为一组，到生产业务单位参加实际工作，并作调查研究；
- 2) 课题来自实际调查，不能采用虚拟课题；
- 3) 组内必须明确分工，密切配合；
- 4) 要求按照软件工程的规范开展各阶段工作，各阶段必须提交相应文档。

3. 内容

在生产实习的基础上进行如下工作：

- 1) 详细设计，得到各模块的数据结构、输入输出和具体处理的说明。
- 2) 编制程序进行模块调试。
- 3) 设计测试用例，进行系统调试，进一步完善系统。

第二章 详细设计

1. 架构设计

1.1. 总体结构

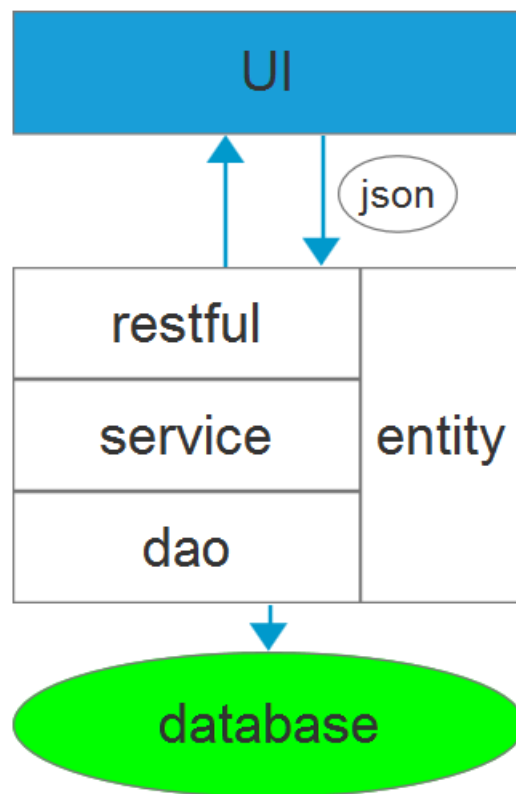
```
beist
|_ beist_boot      -- 后端代码
|_ beist_ui        -- 前端代码
|_ beist_spider    -- 爬虫代码
|_ document        -- 项目文档
|_ database        -- 数据库文件
|_ readme.md
```

1.2. 后端

1.2.1. 代码结构

```
beist_boot
|_ pom.xml
|_ src
|_ test
|_ main
|_ java
|_ com.beist
|_   BeistBootApplication.java -- 后端启动入口
|_   entity
|_   vo
|_   dao
|_   service
|_   impl
|_   restful
|_   util
|_ resources
|_ application.properties
```

1.2.2. 技术分层



Restful 层：提供 UI 交互接口，使用 Spring Data Rest 框架。

Service 层：封装业务逻辑代码。

Dao 层：数据访问层，封装查询或维护数据库的代码，使用 Spring Data JPA 框架，继承其 `CurdRepository` 类进行数据库快速查询，也可以使用 HQL 和 SQL 语句查询数据库。

Entity 层：领域模型，跨多层次使用。其中与 UI 的交互使用 JSON 格式序列化、反序列化，若需要自定义表单，写在 vo 包内。

1.3. 前端

1.3.1. 代码结构

```
|_ beist_ui
  |_ package.json      -- 配置文件
  |_ readme.md
  |_ node_modules      -- 包模块
  |_ public
    |_ index.html      -- 主页面
  |_ src
```

_ App.js	
_ styles	-- 存放.css等样式文件
_ pages	-- 存放页面
_ images	-- 存放图片
_ component	-- 存放组件

2. 接口设计

2.1. 前端访问接口

前端运行在 3000 端口，后端运行在 8080 端口，通过跨域 ajax 请求，使得前端能够访问后端的接口。

前端根据不同的模块，访问后端不同的 URL 地址，分别为：

User 模块：api/user/

Word 模块：api/word/

Article 模块：api/article/

2.2. 后端返数据接口

后端以以下形式给前端返回数据，分别为状态、信息、结果集：

```
public class JsonResult {
    public static final Integer STATUS_OK = 0;
    public static final Integer STATUS_FAIL = 1;

    public static final String MESSAGE_OK = "SUCCESS";
    public static final String MESSAGE_FAIL = "FAILED";

    public static String fillResultString(Integer status, String
message, Object result){
        JSONObject jsonObject = new JSONObject() {{
            put("status", status);    // 状态
            put("message", message); // 错误信息
            put("result", result);    // 结果集
        }};
        return jsonObject.toString();
    }
}
```

	}
}	

第三章 程序模块

1. 用户模块

1.1. 登录子模块

1) 获取前端表单数据，判断该用户是否在数据库中

```
// Ajax - Fetch API
// 调用后端接口，验证登录是否正确，正确转 myPage
fetch("/api/user/login", {
  method: "POST",
  headers: {
    'Cache-Control': 'no-cache',
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    'userTele': this.state.userInfo.userTele,
    'password': this.state.userInfo.userPass
  })
})
// JSON.stringify 序列化
}).then(function (response) {
  return response.json()
}).then((json) => {
  console.log('parsed json', json)
  if (json.status === 0) {
    this.setState({
      loginLabel: ''
    });
    let token = json.result.token;
    // 存入 localStorage 中
    localStorage.setItem("token", token);
    let userTele = json.result.userTele;
    localStorage.setItem("userTele", userTele);
    let userName = json.result.userName;
    localStorage.setItem("userName", userName);
```



```

        window.location.href = '/myPage';
    }
    else {
        this.setState({
            loginLabel: "手机号或密码错误"
        });
    }
}).catch(function (ex) {
    console.log('parsing failed', ex)
})

```

- 2) 判断是否是用户一天内的第一次登录，如果是，则选取今天需要背的单词（旧词和新词以一定比例分配），并修改最近登录日期
- 3) 后端返回给前端 token 串，使前端获取访问 myPage 的权限

```

// 生成 Token
JWTHelper jwtHelper = new JWTHelper();
String token = null;
try {
    token = jwtHelper.createJWT(userTeleConfirmed,
JWTHelper.SHORT_JWT_TTL);
// 自定义 jwtHelper 类以用户手机号生成 token、其中 JWT_TTL 为 token
有效时间，jwtHelper 类调用第三方 jjwt 包
} catch (Exception e) {
    e.printStackTrace();
}

```

1.2. 注册子模块

- 1) 判断前端表单数据是否填写完整，判断该手机号用户是否已经存在在数据库中，如果不存在，允许用户注册
- 2) 初始化用户背单词的信息（注册时用户只有新词）
- 3) 后端返回给前端 token 串，使前端获取访问 myPage 的权限

1.3. 用户设置子模块

- 1) 用户通过 UI 界面修改用户昵称
- 2) 前端将修改后的昵称通过 json 传至后端
- 3) 后端判断用户是否拥有 token 权限，若没有则拒绝修改请求

```

JSONObject jsonObject = new JSONObject(jwtCheck(token, userTele));
// 调用 jwtCheck 函数判断是否拥有权限, jwtCheck 函数将 token 串进行解析, 若解析结果与 userTele 相等, 则拥有权限
if (jsonObject.getInt("status") == 1) {
    result.put("errorMessage", "您没有权限, 请登录");
    return JsonResult.fillResultString(JsonResult.STATUS_FAIL,
        JsonResult.MESSAGE_FAIL, result);
}

```

4) 修改用户表的昵称字段

1.4. 背诵设置子模块

- 1) 用户通过 UI 界面修改用户难度设置
- 2) 前端将修改后的昵称通过 json 传至后端
- 3) 后端判断用户是否拥有 token 权限, 若没有则拒绝修改请求
- 4) 修改用户表的难度字段, 由于登录时才重新载入当天要背的单词, 因此修改第二天之后才会生效

2. 单词模块

2.1. 单词读入子模块

- 1) 根据按类型存储的单词列表从网页中爬取单词信息, 部分代码如下

```

public static void SearchInformation(String word) {
    .....
    String tag1 = "<span class=\"def\">";    //英英词意
    String tag2 = "<span class=\"phonetic\">";    //音标
    String tag3 = "<div class=\"trans-container\">"; //词性+中文词意
    String tag4 = "<div class=\"examples\">"; //例句
    try{
        flag1 = 0; //标记是否已记录相关信息
        flag2 = 0;
        flag3 = 0;
        flag4 = 0;
        word_information = word;
        //生成一个 URL 对象
    }
}

```

```

        url=new URL("http://dict.youdao.com/w/"+ word
+ "/"#keyfrom=dict2.top");
        //打开 URL
        urlConnection =
(HttpURLConnection)url.openConnection();
        //获取服务器响应代码
        responsecode=urlConnection.getResponseCode();
        if(responsecode==200)
        {
            //得到输入流，即获得了网页的内容
            reader = new BufferedReader(new
InputStreamReader(urlConnection.getInputStream(),"UTF-8"));
            while((line = reader.readLine())!=null)
            {
                if(flag1 == 1 && flag2 == 1 && flag3 == 1 && flag4 == 1)
                    break;
                System.out.println(line);
                if(flag1 == 0 && line.indexOf(tag1)!=-1)
                {
                    flag1 = 1;
                    information = getTeacherList(line);
                    //去除其中中的标签
                    word_information = word_information + " | " +
information.substring(1,information.length()-1);
                }
                else if(flag2 == 0 && line.indexOf(tag2)!=-1)
                    .....
            }
            PrintWriter out = null;
            FileWriter fw=null;
            try{
                fw = new FileWriter(file,true);
                out = new PrintWriter(fw);
                out.println(word_information);
            }

```

```

        } catch (IOException e) {
            System.out.println(e.getMessage());
        } finally {
            out.close();
            fw.close();
        }
        .....

```

2) 将爬取到的信息存入数据库，部分代码如下

```

// 行数据转换成数组
String[] custArray = new String[] { "", "", "", "", "a" };
//System.out.println(custArray[4].trim());
custArray = line.split("\\|");
if (custArray.length == 4)           //若未获取到例句信息
{
    String[] custArray1 = new String[] { "", "", "", "", "Not found." };
    custArray1[0] = custArray[0];
    custArray1[1] = custArray[1];
    custArray1[2] = custArray[2];
    custArray1[3] = custArray[3];
}
else
    insertCustInfo(custArray);
    .....

public static void insertCustInfo(String[] strArray) {
    try {
        String sql = "INSERT INTO W VALUES (?, ?, ?, ?, ?, ?, ?)";
        //System.out.println(strArray[4].trim());

        pstmt = conn.prepareStatement(sql);
        idx = 1;
        pstmt.clearParameters();
        pstmt.setInt(1, i);
        i++;
        pstmt.setString(2, strArray[0].trim());

```

```

        pstmt.setString(3, strArray[1].trim());
        pstmt.setString(4, strArray[4].trim());
        pstmt.setString(5, strArray[3].trim());
        pstmt.setString(6, strArray[2].trim());
        pstmt.setString(7, "六级");           //设置单词类型
        pstmt.executeUpdate();
        //System.out.println(strArray[0]);
    } catch (SQLException e) { .....

```

2.2. 查单词子模块

- 1) 用户通过 UI 界面传回要查的单词
- 2) 后端对用户要查的单词进行模糊查询，返回单词列表

2.3. 测试单词子模块

- 1) 用户点击“去测试”按钮，前端向后端请求测试单词
- 2) 后端随机从数据库中选择不等难度（5类）的单词各8个，共40个。
- 3) 前端根据用户回答情况判断用户水平，并提示后端修改用户水平

2.4. 背单词子模块

- 1) 用户点击“背单词”按钮，前端向后端请求返回今日要背的单词
- 2) 当用户在仅出现单词和音标的时候点击认识，则说明用户已掌握该单词，单词无需再背，后端相应修改该单词熟练度
- 3) 否则这些单词将循环出现，直至所有单词都能够第一遍认识为止

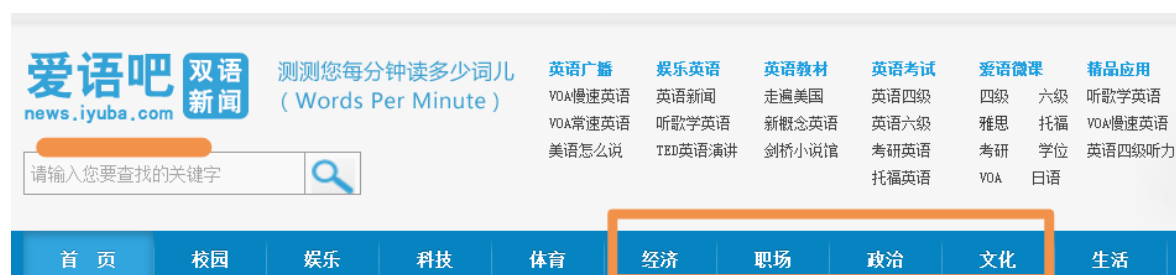
3. 文章模块

3.1. 爬虫子模块

- 1) 爬虫环境：PyCharm
- 2) 爬取对象：

网站：<http://news.iyuba.com/>

经济、职场、政治、文化四类型的新闻



3) 爬虫设计思路

先从文章正文页面出发爬取出文章的标题、日期、单词数、正文内容，并将其存在特定的 txt 文件当中，外层从目录的 URL 出发，在目录页找到文章详情页面的 URL 进行一个循环，得到所有的文章信息。

4) 具体实现：

1) 文章正文页面获取文章标题

```
for news in soup2.select('.title_cn'):
    new_data['title'] = news.select('h1')[0].text
    #print(new_data['title'])
    str_res = str_res + new_data['title']
```

2) 文章正文页面获取文章日期

```
for news in soup2.select('.span1'):
    timesource = news.text
    #print(timesource)
    new_data['date'] = datetime.strptime(timesource, '时间:%Y-%m-%d')
    # 将取得的日期字符串类型转化为 date 类型
```

3) 文章正文页面获取文章单词数

```
for news in soup2.select('#wordcount'):
    new_data['count'] = int(news.text)          # 将字符型的数据
转化为 int 型
    #print(new_data['count'])
    str_res = str_res + ' | ' + str(new_data['count'])
```

4) 文章正文页面获取文章正文、类型和路径

```
for news in soup2.select('.box'):
    new_data['article'] = []
    for x1 in news.select('.p1'):
        #print(x1.text)
        new_data['article'].append(x1.text)
    str_de = new_data['title'] + '\n' +
'\n'.join(new_data['article'])
```

```

        if x2 < 59: # 根据我们爬取新闻的顺序，给类别
new_data['type'] 赋值
            new_data['type'] = '职场'
        elif x2 < 118:
            new_data['type'] = '政治'
        elif x2 < 177:
            new_data['type'] = '经济'
        else:
            new_data['type'] = '文化'

# 通过文章的爬取顺序赋予不同的文件名，得到每篇文章的路径
new_data['path']
    str_txt = str(x2)+".txt"
    new_data['path'] = str_txt
    str_res = str_res + ' | ' + new_data['path'] + '\n'

5) 将信息都写入 txt 文件
    fout = open(str_txt, 'w')      # 将每篇文章的内容写到对应的 TXT
文件中
    fout.write(str_de)
    fout.close()
# 格式为 'Job search site apologizes | 2017-08-12 00:00:00 | 163 |
职场 | 6.txt'
    fout = open('result.txt', 'w')      # 将所有文章信息写到
'result.txt' 文件中
    fout.write(str_res)
fout.close()

6) 获取文章页面的 URL
# 遍历 link_ca 列表的 url
for i in range(0,15):
    url1 = link_ca[i]
    res1 = requests.get(url1)
    res1.encoding = 'utf-8'

```

```

        soup1 = BeautifulSoup(res1.text, 'html.parser')
# 取得目录页面 c_left2 类下 <a>标签中的 'href' 属性
"essay/2017/09/03/57656.html"
# 由于其属性值中的 url 信息是不完整的, 所以在其之前添加
'http://news.iyuba.com/'
        for news in soup1.select('.c_left2'):
            for x1 in news.select('a')[:7]:
                link_ar.append('http://news.iyuba.com/' + x1['href'])

```

3.2. 文章分词与难度设置子模块

- 1) 将进行过预处理的 txt 文件读入设置文章信息除难度外的参数并存入数据库

```

line = line.trim();
String[] articleInfo = new String[] {"", "", "", "", ""};
String articleLevel = ""; //记录文章难度类型
//articleInfo[0]--文章标题
//articleInfo[1]--时间
//articleInfo[2]--字数
//articleInfo[3]--type 类型
//articleInfo[4]--路径
//读入文章信息行
articleInfo = line.split("\\|");
Article article = new Article();
article.setArticleTitle(articleInfo[0].trim());
article.setArticleType(articleInfo[3].trim());
article.setArticlePath(articleInfo[4].trim());
article.setArticleWordNumber(Integer.parseInt(articleInfo[2].trim()));
article.setArticleDate(new Date());
articleService.save(article);

```

- 2) 对文章内容进行分词处理

```

article_file = new FileInputStream(articlePath + "/article" +
articleInfo[4].trim());
String article_body = "";
BufferedReader reader2 = new BufferedReader(new

```



```

ArrayList<Map.Entry<String, Integer>>(map.entrySet());
for (int i = 0; i < infoIds.size(); i++) {    //输出
    Entry<String, Integer> id = infoIds.get(i);
    //System.out.println(id.getKey()+":"+id.getValue());
    Word w = wordService.findByWord(id.getKey());
    if(w != null) {
        if (w.getWordLevel().equals("基础")) {
            level_jichu += id.getValue();
            //写入文章单词表操作
            wordService.saveArticleWord(article,w,id);
        }
        else if (w.getWordLevel().equals("初中")) {
            level_chuzhong += id.getValue();
            //写入文章单词表操作
            wordService.saveArticleWord(article,w,id);
        }
        else if (w.getWordLevel().equals("高中")) {
            level_gaozhong += id.getValue();
            //写入文章单词表操作
            wordService.saveArticleWord(article,w,id);
        }
        else if (w.getWordLevel().equals("四级")) {
            level_siji += id.getValue();
            //写入文章单词表操作
            wordService.saveArticleWord(article,w,id);
        }
        else if (w.getWordLevel().equals("六级")) {
            level_liuji += id.getValue();
            //写入文章单词表操作
            wordService.saveArticleWord(article,w,id);
        }
    }
    else {
        //写入文章单词表操作, 单词编号为 0

```

```

        w = wordService.findByWord("not found");
        wordService.saveArticleWord(article, w, id);

    }
}

```

4) 计算文章难度并修改数据库文章难度参数

```

int total = level_jichu + level_chuzhong + level_gaozhong + level_siji
+ level_liuji;
if(level_liuji > total*0.1) //六级单词占比超过 10%
    articleLevel = "六级";
else if(level_siji > total*0.1) //四级单词占比超过 10%
    articleLevel = "四级";
else if(level_gaozhong > total*0.2) //高中单词占比超过 20%
    articleLevel = "高中";
else if(level_chuzhong > total*0.2) //初中单词占比超过 20%
    articleLevel = "初中";
else
    articleLevel = "基础";
//写入文章难度信息
article.setArticleLevel(articleLevel);
articleService.save(article);

```

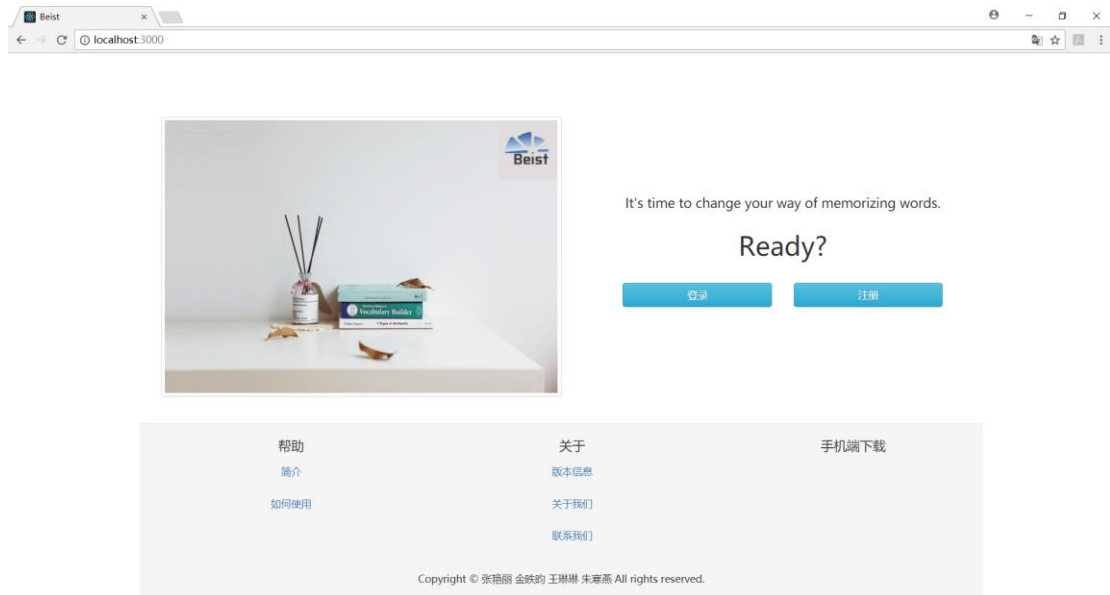
3.3. 阅读文章子模块

- 1) 自动给用户随意推荐符合用户难度的文章，其中有一篇为主打，显示概要。
- 2) 系统将用户待读的文章与推荐的文章都传给前端
- 3) 前端选择阅读的文章，此时若文章未曾被点击过，则加入用户文章表，并返回文章的单词给用户，让用户先背单词
- 4) 若文章曾被点击过，但单词还未背完，则仍然返回文章的单词给用户，让用户先背单词
- 5) 若文章曾被点击过，单词已背完，但仍未完成阅读，则直接返回文章给用户继续阅读
- 6) 若文章已阅读完成，则直接返回文章，用户可以再浏览一遍文章

第四章 测试用例

1. 用户模块

1.1. 主页



1.2. 点击简介



1.3. 登录

登录

手机号

15821379810

密码

..

还未注册?

登录

1.4. 注册（提示信息）

注册

手机号

手机号

手机号必须为11位

昵称

昵称

密码

密码

确认密码

确认密码

难度

六级

验证问题

验证问题

验证问题答案

验证问题答案

获取验证码

验证码

手机号不能为空

已有账号?

注册

手机号

15821379810



手机号必须为11位

昵称

Admin

密码

..

确认密码

...

难度

六级

验证问题

验证问题

验证问题答案

验证问题答案

获取验证码

验证码

密码与确认密码不一致！

已有账号？

注册

1.5. 注册后进入个人主页

Beist 背单词 读文章

查单词

Admin

设置

帮助

注销

单词

今日已背	0 个
今日需背	10 个
总共已背	0 个
背诵进度	0 %

背单词

文章

熟悉单词	0 篇
准备阅读	0 篇
已经阅读	0 篇
阅读进度	0 %

读文章

用户信息

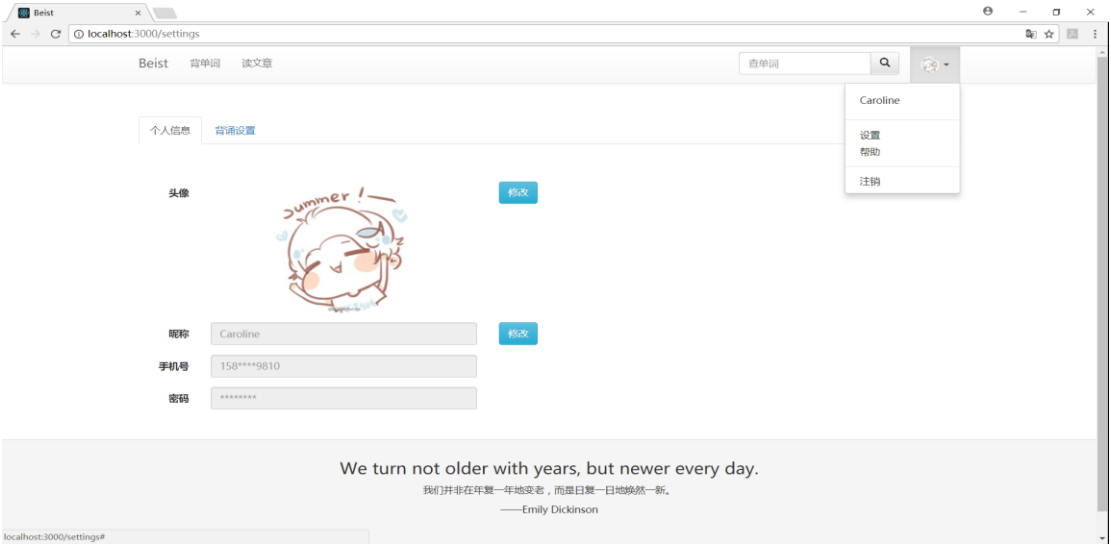
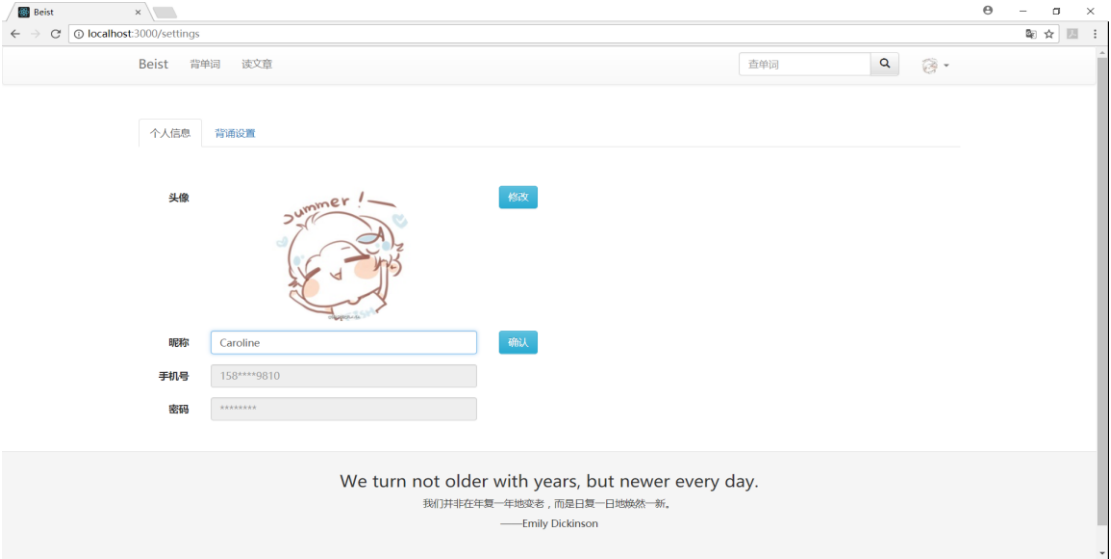
水平测试	零基础
目标难度	四级
已背天数	0 天
总共需背	1321 个

去设置

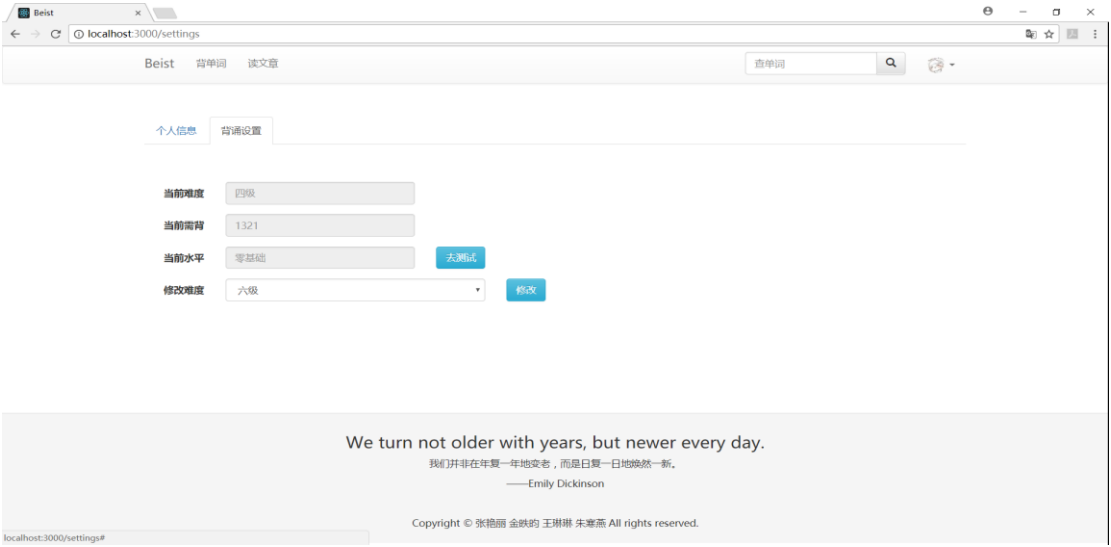
We turn not older with years, but newer every day.
我们并非在年复一年地变老，而是日复一日地焕然一新。
——Emily Dickinson

Copyright © 张艳丽 金朕的 王琳琳 朱章燕 All rights reserved.

1.6. 修改用户名

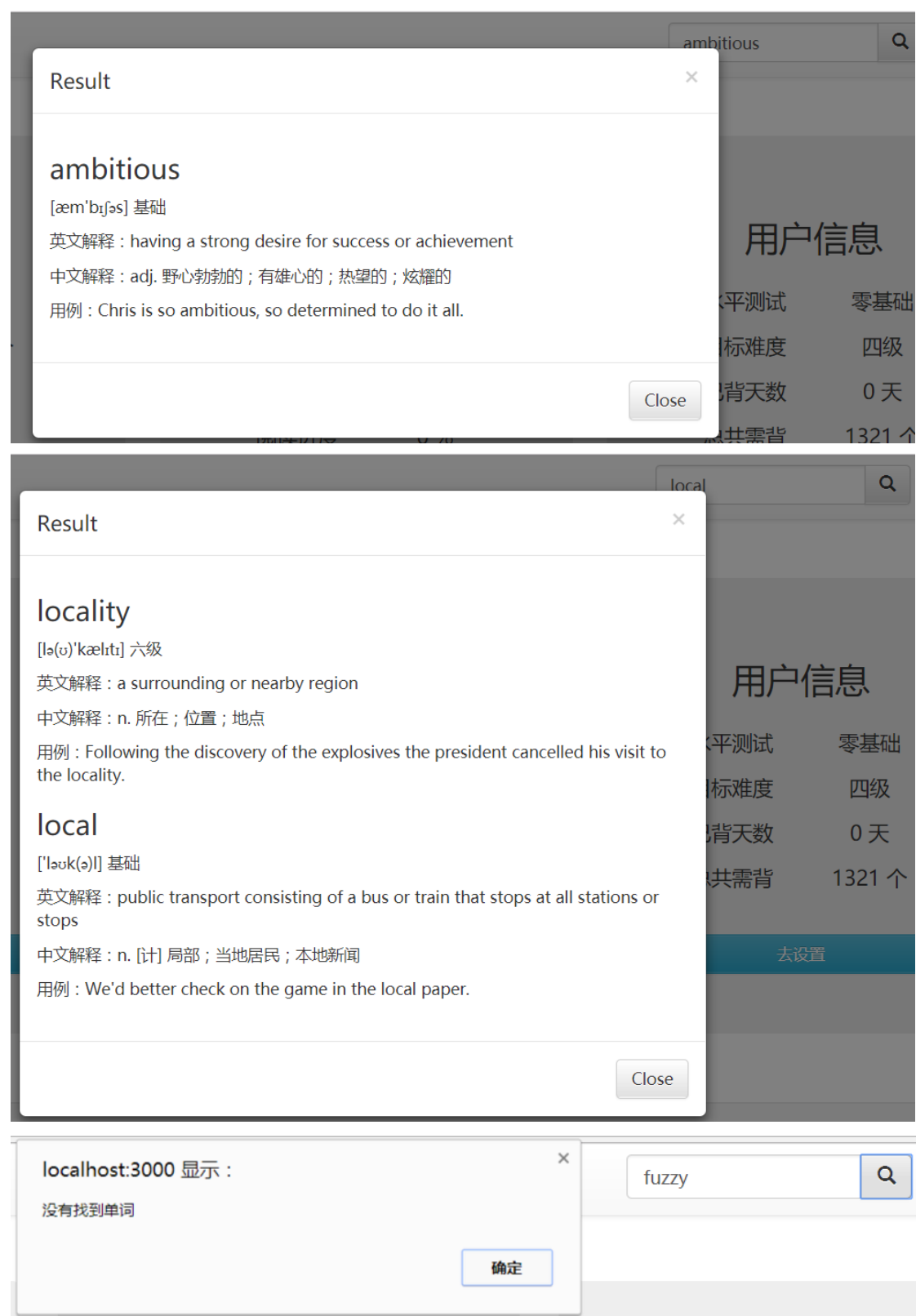


1.7. 修改难度



2. 单词模块

2.1. 查单词



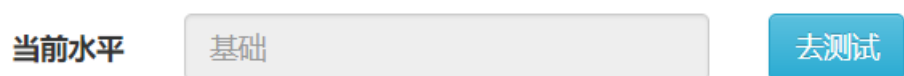
2.2. 测试单词（中途防止误触退出测试）



2.3. 测试单词



2.4. 测试单词后水平更改



2.5. 背单词 (初始界面)



2. 6. 点击不认识后，逐一跳出解释项



×

absolutely

['æbsəlu:tli]

例句: Joan is absolutely right.

英文解释: completely and without qualification; used informally as intensifiers

中文解释: adv. 绝对地; 完全地

认识

查看详情

2. 7. 点击查看详情出现单词详情

×

absolutely

['æbsəlu:tli]

例句: Joan is absolutely right.

英文解释: completely and without qualification; used informally as intensifiers

中文解释: adv. 绝对地; 完全地

撤销认识

下一个

2. 8. 点击下一个进入下一个单词

×

accordingly

[ə'kɔːdɪŋli]

认识

不认识

2.9. 点击认识，直接进入单词详情，下次不会再出现，允许撤销

×

accordingly

[ə'kɔːdɪŋli]

例句: We have a different background, a different history.
Accordingly, we have the right to different futures.

英文解释: (sentence connectors) because of the reason given

中文解释: adv. 因此，于是；相应地；照著

撤销认识

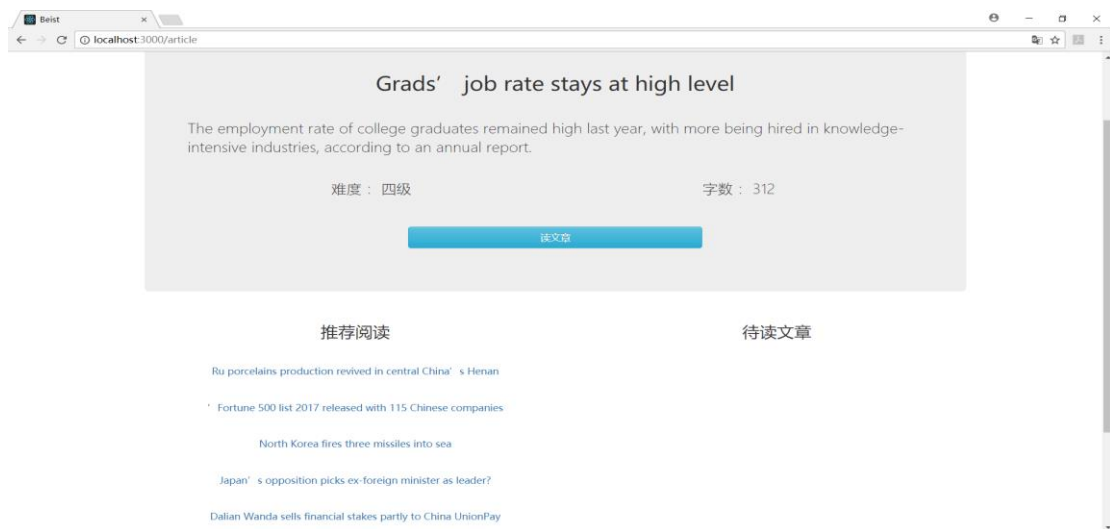
下一个

2.10. 单词模块信息

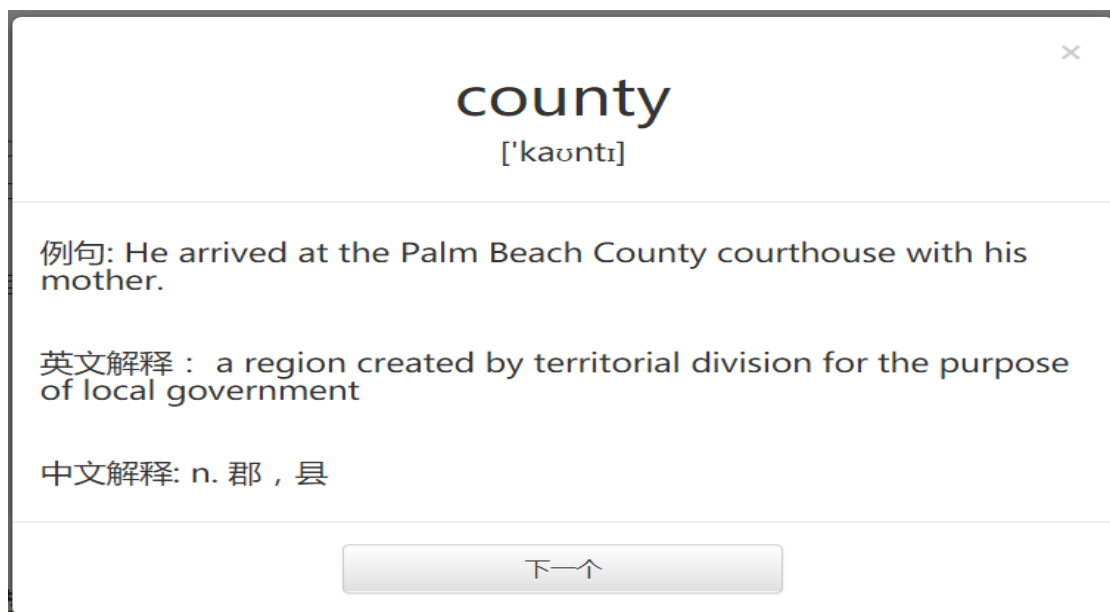


3. 文章模块

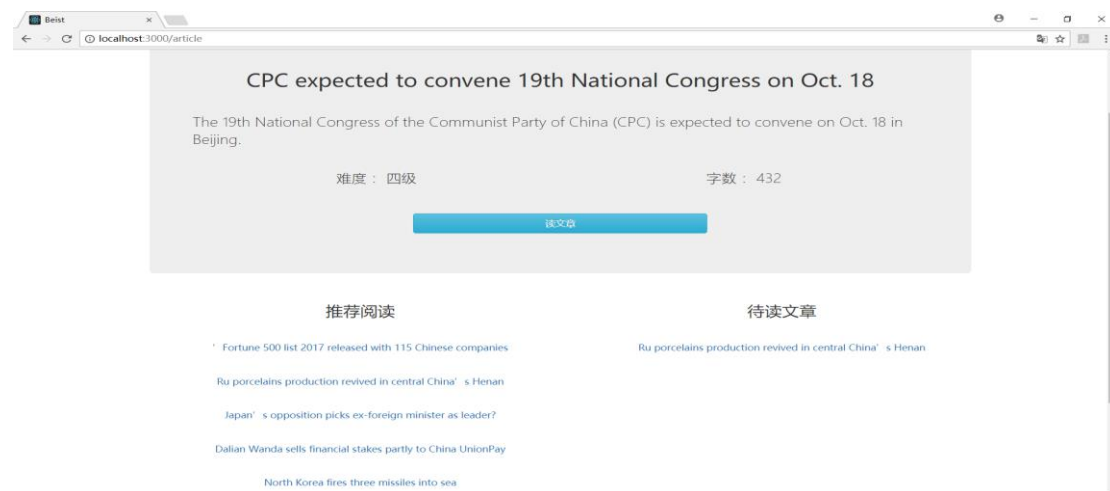
3.1. 文章页面（推荐阅读和待读文章）



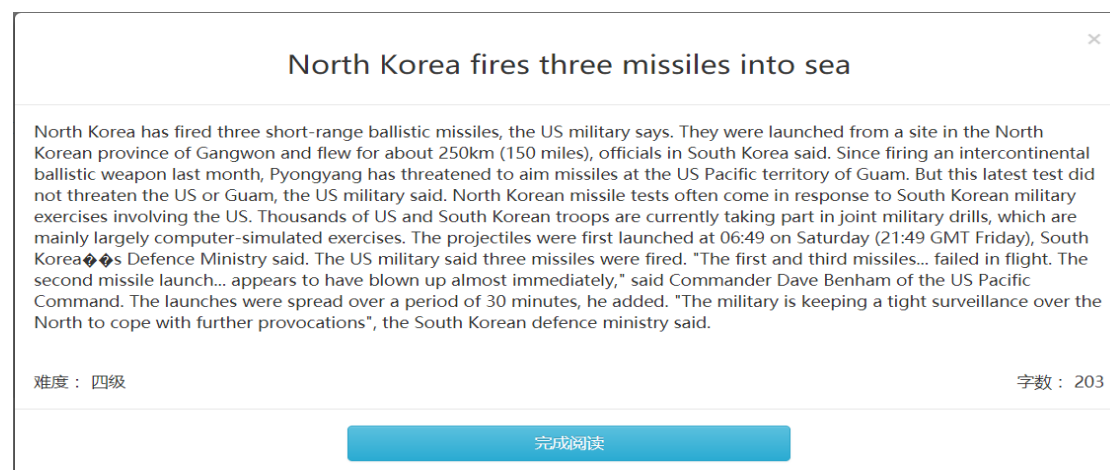
3.2. 点击未阅读过的文章，进入单词界面



3.3. 中途退出，加入待读文章列表



3.4. 点击已背完单词的文章，进入文章详情页阅读



3.5. 文章详情页



第五章 用户手册

1. 安装说明

1.1. 语言和集成开发环境

- 1) Java : IntelliJ IDEA 2017.1
- 2) Javascript & html & css : WebStorm 2017.2.2
- 3) Python : Pycharm 2017.1.2
- 4) SQL : SQL Server 2016

以下介绍仅针对于以上开发环境

1.2. 安装说明

1.2.1. beist-ui

安装 NPM(Node Package Management): 一款包管理工具, 随 Node.js 一起安装, 安装包可在 document 文件夹下找到;

进入 beist-ui 文件夹, 启动 cmd, 输入 npm start;

打开浏览器, 输入网址 localhost:3000 即可看到网页界面。

1.2.2. beist-boot

安装 Maven: 一款项目管理工具, 安装包可在 document 文件夹下找到, 需配置环境变量, 并打开 Maven\apache-maven-3.5.0\conf\settings.xml 文件修改 LocalWarehouse 地址;

打开 Idea IDE 设置 Maven 安装目录和 Settings 文件位置;

进入 SQL Server, 附加数据库;

打开 beist-boot 项目;

修改 src\main\resources 下的 application.properties 文件中 spring.datasource.username 和 spring.datasource.password 的值;

修改 src\main\java\com\beist\util 下的 PathConstants.java 文件中的地址;

点击 Run 'BeistBootApplication' 即可运行后端代码。

1.2.3. beist-spider

使用 Pycharm 打开程序

2. 使用说明

2.1. 用户

- 1) 用户进入后，可查看帮助信息
- 2) 老用户可选择登录进入个人主页，新用户可选择注册一个账号进入个人主页。
- 3) 用户进入个人主页后，可选择设置修改用户名和难度

2.2. 单词

- 1) 用户进入个人主页后，可选择背单词。在仅出现单词及音标后点击认识，则该单词今日不会再出现，否则该单词今日会反复出现，直到仅出现单词音标后确认认识。
- 2) 用户可以模糊查询单词的意思
- 3) 用户可以进入设置页面，对自己的水平进行测试
- 4) 用户可以在个人主页上查看自己的背单词进度

2.3. 文章

- 1) 用户进入个人主页后，可选择读文章。新选择的文章会先进入单词的背诵，直到所有该难度的单词背诵完成后，才会进入文章阅读页。曾经背完过单词的文章再次点击，会直接跳出文章
- 2) 用户可以在文章页面查看推荐的文章和曾经阅读过的文章
- 3) 用户可以在个人主页上查看自己的读文章进度

第六章 个人体会

1. 张艳丽

这是大学阶段最后的一个小组项目。在暑假前，我们组员四人都卯足了劲想要做出一个完美的背单词系统。时至至今，结果稍有些不尽如人意，我们只做到了最原始的基本功能，大多拓展功能都因为时间、难度等方方面面的问题被阉割了。

但好在，我们为项目扩展做了不少准备，这体现在我们最初规划功能、设计数据库、甚至是选用的框架上。我们采用了前后端分离的架构，后端由于只提供 REST 接口，几乎不需要改动，而前端的 React 是近年来最火的前端框架之一，其衍生的 React Native 可以用来开发 App，React 的代码只需稍微修改即可用在手机上，当然手机 App 的用户体验和网页肯定是不同的。

此次项目，作为组长，我主要负责任务分配、架构的搭建、前后端的交互、用户权限等任务。在架构的选择上，我们这次选择的架构都非常热门、非常新颖，也因此踩了不少的坑。例如，我们搭建 React 前端是采用了 facebook 官方提供的脚手架，其版本是最新的，以至在安装 React-router 的时候，我们不知道其选用的 4.0 版本将 React-router 分成了两个模块包，而网上几乎没有人碰到过类似的问题，只能通过仔细阅读官方文档得到解答。又比如，网上大多数使用 Spring Boot + React 的架构教程，React 最终也只是生成一个页面存入到 webapp 文件夹下，而我们并不想将前后端本分的好好的两个文件夹又重新打乱在一起，因此又尝试了许多方法。这些框架给我们带来了不小的学习成本，前前后后搭建整个架构用了 2-3 周的时间，我想我也需要为拖慢整个项目的进度负很大的责任。但新框架的新特性总能让人眼前一亮，这样穷途末路的体验反倒比项目本身更让我感到跃跃欲试，我想我是不后悔的吧。

用户权限上，我也摒弃了传统的 session 和 cookie，而选用了 token 串的形式，他是 REST 模式下的一种新方式，也是移动端大热的今天越来越流行的一种方式。唯一遗憾的是，由于时间关系，尝试了许久的 spring security 还是失败了，可能我们使用 token 串做的验证仍然非常浅显。前后端的交互我们使用了 REST 模式，前端使用 Fetch 来进行 ajax 请求，也取代了普遍使用的 jQuery。可以说，我们的项目，就是在不断的和新事物磨合的过程中，磨出来的结果。

此外，与以往做的其他项目不同，这次的项目从功能开始，都是我们自己设计的。业务逻辑上难免会出现很多磕磕绊绊，而模块之间又不能做到彻底独立，使得我们不断地进行返工。比如，当 User 模块做完后，Word 模块的设计中又需要修改 User 的流程和逻辑，用户改了背单词的难度，Word 表也要相应做修改。

Web 项目毕竟不像普通只有一个 main 函数的程序，他没有确定的输入，用户会做些什么几乎是不可控的，也因此很难有确定的输出。在项目测试的过程中，我们常常发现很多结果不能重现，也就无从改起，但事出必有因，这些 bug 埋藏在微小的细节处，宛如一颗定时炸弹，很可能会引发后续的 bug。一整个暑假我都在企业中做开发的实习，bug 几乎是习以为常如同呼吸一般正常的事，有的 bug 改好了，回归却引发了新的错误……这带给我很深刻的印象。

这个暑假即将结束，但我不希望这个项目也随着验收的结束而结束。它是我花了数个夜晚思考的逻辑功能，它是我切实分析可行性的成果。它还很不成熟，它还有很多 bug 没有改，它还有很多拓展功能没有做，这都让我对它放不下心，好在大四的我没有什么课，继续完成这个项目已经成为了我的规划，我想将它变得更好一些。而即使哪天我将所有的功能都做完了，将所有的 bug 都找到了（虽然几乎不可能），它使用的框架也不再是最流行的那个了，它还会继续被重构，被优化……

大概计算机就是这样的东西吧。

2. 金昞昞

这次的项目比起以往参与完成的都要更为繁杂，很多功能，在实现才发现其实远没有当初设想时的那么简单。真正代码实现时有许许多多要考虑的东西，特别是很多细节方面，稍不注意，最终效果就会产生偏差。

项目中涉及到了很多我之前没有接触过的技术，比如爬虫、框架等。像是在建立数据库时，就参考了爬虫的一些实现方式，并结合对字符串的处理，来得到所需的数据。而在学习和使用框架时，虽然看上去十分难以理解和掌握，但真正使用时还是感受到了这一技术的一些方便之处。可以说对于规模比较大的项目来说，在较好地掌握了框架的前提下，它能够保证其结构的清晰和操作的便利，可以说还是有很大的优越性的。但也正是由于对这种不同于以往写程序的方法，我们对它的很多使用方式其实并不太熟悉，掌握度不够使得我们在完成和测试时话费了比预计更多的时间。这也让我深刻体会到了完成一个项目时难度预期和时间规划的重要性。

整个项目的完成过程中遇到了不少问题，虽然总体上实现了最早所设计的基本功能，但也留下了不少待完善的部分。通过这次项目，我对其中所涉及的基本知识进行了巩固，对于之前没接触过了技术应用也有了一个基本的了解。虽然由于时间等问题对很多内容其实还处于一知半解的状态，希望在以后的学习中能逐步巩固与掌握这些知识。

3. 王琳琳

本次实习我们所做的项目是 beist 背单词软件，整体的感觉是这个项目从一开始的构思，到最后完成都走得很扎实。首先，在设计阶段我对项目的认识更加深刻，在此之前，我对做项目的概念基本为零，如果让我一个人去想，我一定是毫无思路，无从下手。但是小组里有两名同学很有经验，她们通过划分功能模块，将每个模块的功能进行细化，并讨论功能实现的操作和相关算法，做成文档。通过阅读文档几乎整个项目的实现都清晰了。这给了我很大的启发。其次，在实现编码的阶段，我意识到自己的能力跟其他组员的差距非常大，我根本跟不上她们的节奏。在编码时，我被分配的任务是爬虫，得到我们需要的文章。根据我对爬虫的初步了解，使用 python 语言会比较简单，我在网上找到一些相关的实例教程，由于对 python 语言不了解，只能根据教程来做，我以为爬虫的代码是有模板的，只要弄出了那个实例就离成功不远了，所以在第一个模板没成功时，我又找了非常多的实例教程，不断地套模板，但是由于爬取内容的网站结构不同，爬的内容不同，代码都是不一样的。我的思路从一开始就是错的，意识到这个问题，我才又开始学习 python 语言，然后一步步自己摸索，这个过程是困难的。整个爬虫的过程我走了很多弯路，浪费了太多时间，进度被我一拖再拖。自己对组员很愧疚，对自己也很失望。不过最后能成功也一定程度上给了我些许鼓舞，总而言之，自己的能力需要提升，实践经验也需要进一步积累。

4. 朱寒燕

此次项目我参与开发是前端，由于之前尚未接触过前端的语音，对此比较陌生，需要在短时间内获取大量的知识，好在前端基础内容相对简单易懂，刚接触的时候，感觉不是很难，因此也产生了一点兴趣，刚开始就要先看 html, css, javascript 的内容，这些语言的基础的部分还是很容易理解的，就是后面复杂的理解能够理解，但是并不能反应很快的记住，在看完这些的基础上，后来又学习了 Bootstrap 模板，并尝试这在此模板上改一些自己想要的设计的东 西，例如登陆，注册等界面，再后来，组长决定让我们学习使用 github，我又去网上边看教程边下载，每次下载时，总是下载到一半会停住，卡住不动，百思不得其解，思考是电脑的原因还是网络的原因，后来发现使因为浏览器的缘故，换了个浏览器，下载软件也就顺利了。

再之后，最终决定使用 React，因其高效灵活的特点，使我们非常欣赏它。网上又有 React-Bootstrap，于是在做网页设计时会对着这些组件看，由于接触时间不长，刚开始做时，边看边做，也在做界面的同时，把 React 的一些组件变得渐渐熟悉起来，慢慢地有了印象。

前端一开始使用的软件使 Textlime，这个软件权当使练练手了，后来为了决定用了 Webstorm，装上此软件后，会发现，缺少好多插件，于是就百度一个个按照解决方法解决问题。

当前端完成，看着自己设计的界面，感觉非常满意，也非常感谢组长督促我的学习，由于个人原因，拖了一点点小组进度，但是组长并没有因此责怪，组长非常认真负责，让我在此次项目中收获良多。