# 作品介绍

## 简介

REMENBER是一个交互式英语学习网站。REMENBER针对广大学生群体记忆英语单词的需求，解决记单词不全面、背过却写不出等问题。REMENBER使用.NET框架，使用Visual Studio 2013的开发平台和Microsoft SQL Server Management Studio 17数据库平台。REMENBER所涉及的开发语言有C#，HTML，CSS，jQuery，JavaScript，SQL。REMENBER英语学习网站为广大学子提供一个全面背诵词书的科学高效记忆单词的平台，能极大地提高单词拼写能力。

## 创作思想

创作这样一个网站的起因是由于自己本身就有这方面的需求，无论是学校的英语课程，还是考四六级，或是考研出国，都有背单词的需要。且根据自身背单词的经验发现主要问题——背过的单词拼写不出来或是写错。这样的问题在考试中致命的，因此也是在复习时最需要注意的。正是针对这样的需要，我就想做一个能帮助提高单词拼写能力的学习工具，REMEMBER应需而生。

## 设计说明

界面——简介、干净、易操作

色彩——明亮、和谐、活力

# 系统设计

## 系统功能设计

### 功能结构图

**图2-1 功能结构图**

### 主要功能说明

* **单词测试（REMEMBER-首页-学习模块-单词测试）**

系统选取用户个人词库中的学习次数最少的30个单词提供给用户学习并测试。测试采用看中写英的默写模式，致力于提高用户的单词拼写能力。30词共分6页进行答题，每完成一页会立即批改显示答题情况和正确答案，及时的反馈能帮助用户立刻纠正错误，加深印象，提高学习效率。在答题完成后，最终会显示用户本次测试的答题情况总汇，帮助用户了解自己的整体掌握情况。

* **我的词库（REMEMBER-首页-学习模块-我的词库）**

此功能模块用于方便用户查看所添加的词库和每个单词的学习情况。

该模块分为五个子模块：

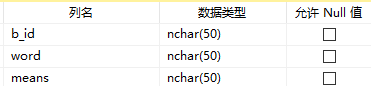
1. 所有单词：显示用户所添加的词书中的所有单词。
2. 已学单词：在所有单词中筛选出已进行测试过的单词。
3. 未学单词：在所有单词中筛选出未进行测试过的单词。
4. 易错单词：在所有单词中筛选出已进行测试过的单词，并对其按错误次数从高到低排序。
5. 分类检索：系统提供的分类选项有词书、首字母、学习次数和错误次数，点击“检索”按钮，系统在所有单词中筛选出符合用户选择条件的单词。

## 数据库设计

### 表结构设计

表book的结构设计如表2-1所示。表book用于存储每本词书的单词信息，存储内容包括词书编号b\_id，英文单词word及单词的中文含义means。

**表2-1 表book的结构设计**



表LoginRecord的结构设计如表2-2所示。表LoginRecord用于存储用户的登录信息，存储内容包括用户名u\_name以及登录时间loginTime。

**表2-2 表LoginRecord的结构设计**



表TestRecord的结构设计如表2-3所示。表TestRecord用于存储用户的单词测试信息，存储内容包括用户名u\_name，测试时间t\_date,测试用时useTime,测试正确率correctRate。

**表2-3 表TestRecord的结构设计**



表users的结构设计如表2-4所示。表users用于存储用户的个人信息，存储内容包括用户名u\_name，密码u\_pwd，生日birth，性别gender,头像head，邮箱mail，学校school，专业major，家乡home，注册日期date。其中，用户名u\_name为主键；生日birth、性别gender、头像head、邮箱mail、学校school、专业major、家乡home允许Null值。

**表2-4 表users的结构设计**



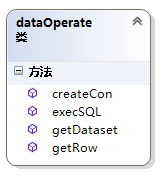
表words的结构设计如表2-5所示。表words用于存储用户添加的单词信息，存储内容包括用户名u\_name,词书编号b\_id，单词word，单词中文含义means,学习次数learn，测试错误次数errorNum，测试时间w\_time。

**表2-5 表users的结构设计**



## 框架结构设计

本作品的框架结构设计使用了公共类dataOperate类。dataOperate类包含的方法有createCon,execSQL,getDataset,getRow，如图2-2所示。



**图2-2 dataOperate类的类图**

### createCon

方法createCon用于连接数据库，具体实现如代码清单2-01所示

**代码清单2-01**

public static SqlConnection createCon()

{

SqlConnection con = new SqlConnection("server=(local);database=REMEMBER;uid=sa;pwd=123456;");

return con;

}

现分步骤具体阐述如下 ：

1. 创建SqlConnection对象
2. 返回SqlConnection对象

### execSQL

方法execSQL用于执行数据库添加、删除、更新等操作，具体实现如代码清单2-02所示

**代码清单2-02**

public static bool execSQL(string sql)

{

//创建数据库连接

SqlConnection con = createCon();

//打开数据库连接

con.Open();

//创建SqlCommand对象

SqlCommand com = new SqlCommand(sql, con);

try

{

//执行SQL语句

com.ExecuteNonQuery();

}

catch (Exception e)

{

//返回布尔值False

return false;

}

finally

{

//关闭数据库连接

con.Close();

}

//返回布尔值True

return true;

}

现分步骤具体阐述如下 ：

1. 创建数据库连接
2. 打开数据库连接
3. 创建SqlCommand对象
4. 执行SQL语句
5. 关闭数据库连接
6. 返回执行结果（若SQL语句执行中断将返回布尔值false，执行成功将返回布尔值true）

### getDataset

方法getDataset用于查询记录，以DataSet对象形式返回,具体实现如代码清单2-03所示

**代码清单2-03**

public static DataSet getDataset(string sql)

{

//创建数据库连接

SqlConnection con = createCon();

//打开数据库连接

con.Open();

//创建DataSet对象

DataSet ds = new DataSet();

//创建DataAdapter对象

SqlDataAdapter sda = new SqlDataAdapter(sql, con);

//填充DataSet数据集

sda.Fill(ds);

//返回DataSet对象

return ds;

}

现分步骤具体阐述如下 ：

1. 创建数据库连接
2. 打开数据库连接
3. 创建DataSet对象
4. 创建DataAdapter对象
5. 填充DataSet数据集
6. 返回DataSet对象

### getRow

方法getRow用于查询记录，以SqlDataReader对象形式返回，具体实现如代码清单2-04所示

**代码清单2-04**

public static SqlDataReader getRow(string sql)

{

//创建数据库连接

SqlConnection con = createCon();

//打开数据库连接

con.Open();

//创建SqlCommand对象

SqlCommand com = new SqlCommand(sql, con);

//返回ExecuteReader方法返回的DataReader对象

return com.ExecuteReader();

}

现分步骤具体阐述如下 ：

1. 创建数据库连接
2. 打开书记库连接
3. 创建SqlCommand对象
4. 返回ExecuteReader方法返回的DataReader对象

# **系统实现**

系统的功能模块共分三大部分——登录模块，单词模块和用户模块。其中，单词模块为主要功能模块。

登录模块分三个子模块：注册模块，登录模块和找回密码模块。单词模块分四个子模块：学习单词模块，添加词书模块，我的词库模块和单词进度模块。用户模块分两个子模块：用户设置模块和用户退出模块。

接下来，详细介绍各个重要模块部分，介绍内容包括界面，功能，使用和实现。

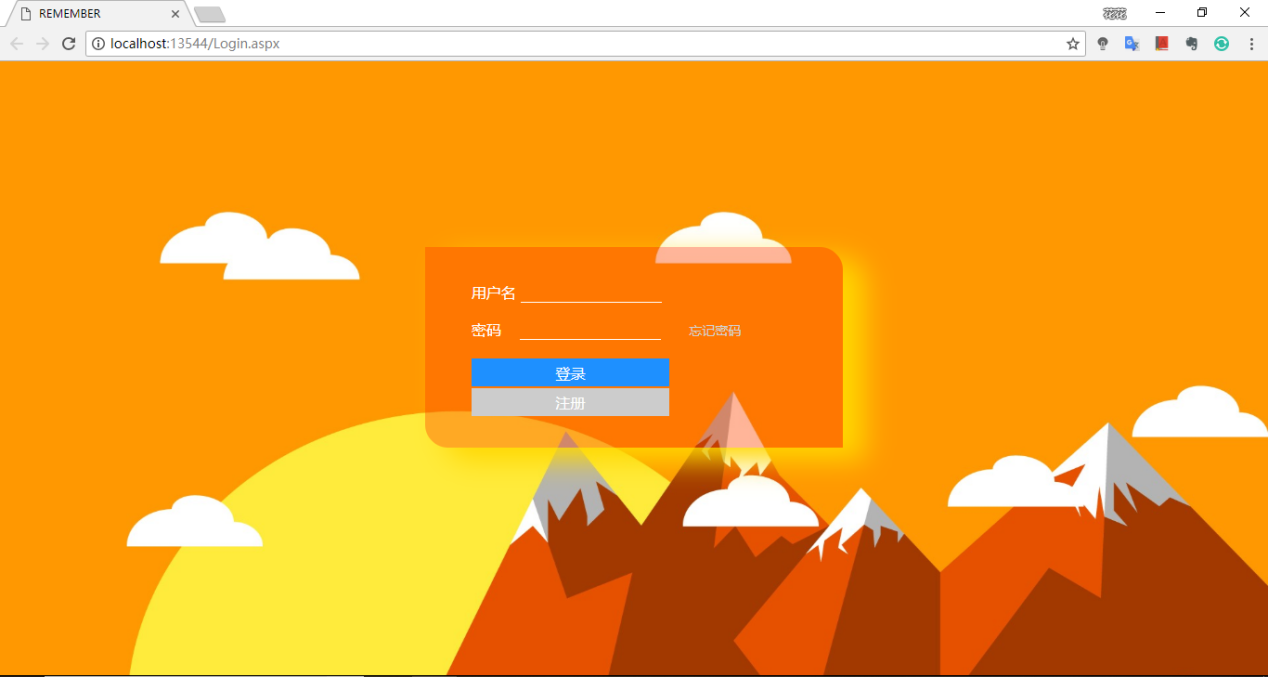
## 登录模块功能介绍

登录模块的功能有用户注册，用户登录，用户找回密码三个部分。

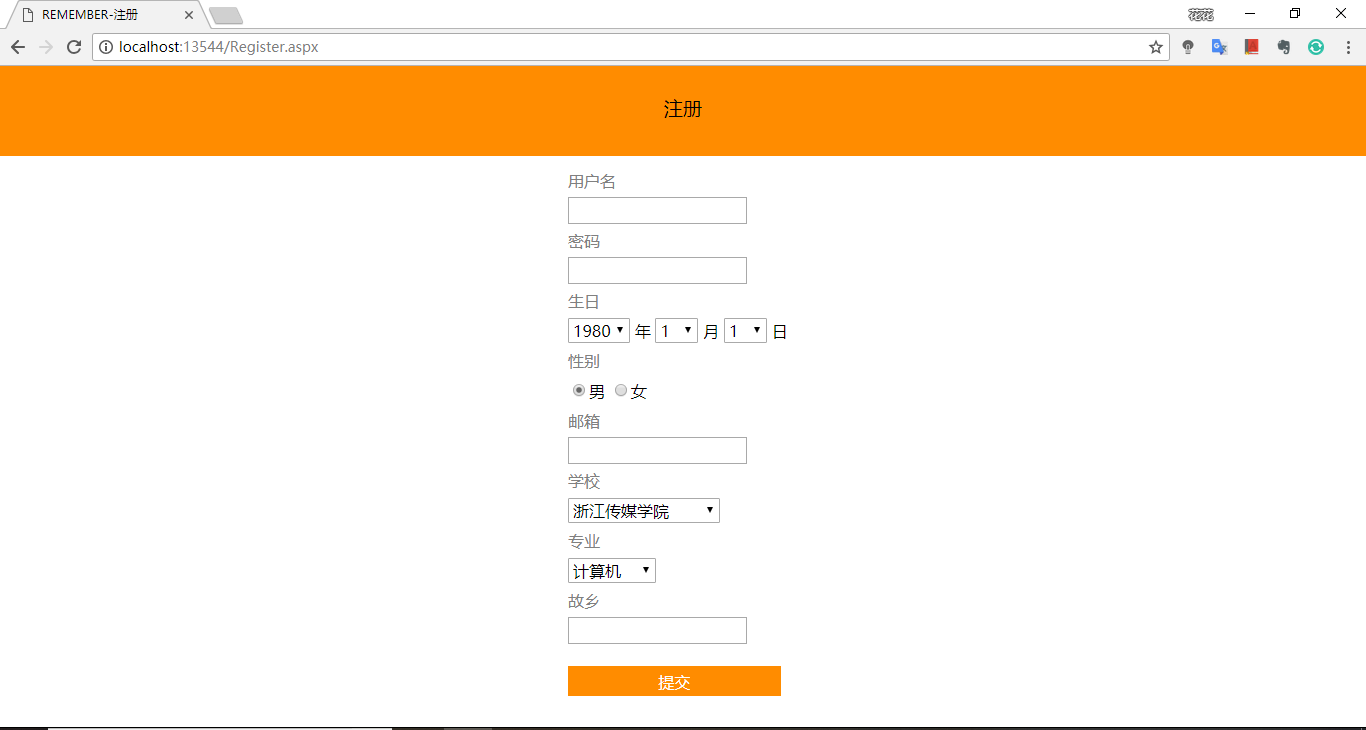
* 用户注册

使用步骤：

1. 在登录页面（如图3-1）点击“注册”按钮，页面将跳转到注册页面（如图3-2）。
2. 在注册页面填写完信息，点击“提交”按钮即可完成注册，并自动跳转回登录页面。



**图3-1 登录页面**



**图3-2 注册页面**

* 用户登录

使用步骤：

1. 在登录页面（如图3-1）填写已注册完成的用户名和密码，点击“登录”按钮即可登录。

代码实现：

“登录”按钮触发事件如代码清单3-01所示

**代码清单3-01**

protected void btnSignIn\_Click(object sender, EventArgs e)//登录

{

username = txtUsername.Text.Trim();//用户名

pwd = txtPwd.Text.Trim();//密码

//验证用户合法性

sqlStr = "select \* from users where u\_name='" + username + "' and u\_pwd='" + pwd.GetHashCode() + "'";

SqlDataReader Dr = dataOperate.getRow(sqlStr);

if (Dr.Read())

{//有记录，登录成功

Session["u\_id"] = username;

//添加该用户登录记录

sqlStr = "insert into LoginRecord (u\_name,loginTime) values('" + username + "','" + DateTime.Now +"')";

dataOperate.execSQL(sqlStr);

//跳转到用户界面

Response.Redirect("Main.aspx");

}

else

{//无记录，登录失败

lblTip.Text = "登录失败！请输入正确的用户名与密码！";

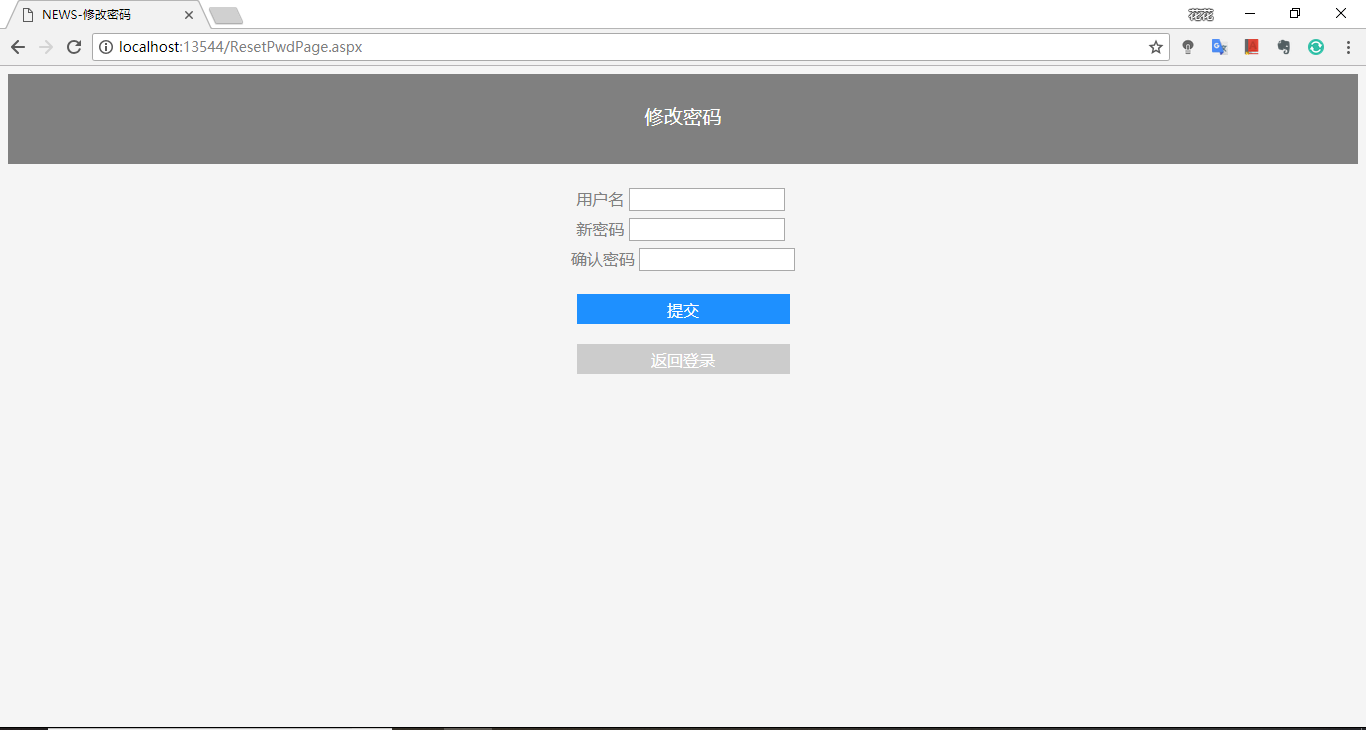
}

}

* 找回密码

使用步骤：

1. 在登录页面（如图3-1）点击“忘记密码”按钮，页面将跳转到修改密码页面（如图3-3）。
2. 填写所需信息，点击“提交”按钮即可完成密码修改。



**图3-3 修改密码页面**

## 单词学习模块功能介绍

单词学习模块的功能是进行单词的学习和测试。

使用步骤：

1. 进入单词学习模块（如图3-4）后，进行30个单词的学习。



**图3-4 单词学习页面**

1. 点击“开始学习”按钮开始进行测试（如图3-5）。



**图3-5 单词学习页面中的答题页面**

1. 30词共分6题，每题5个单词。输入完成点击“提交”。系统会立刻批改，显示正确答案，并用不同颜色显示答题情况，红色为错误，绿色为正确。（如图3-6）



**图3-6 答题页面中的局部——显示答案**

1. 点击“返回”按钮（或完成所有题目），系统会给出答题情况的总汇。（如图3-7）



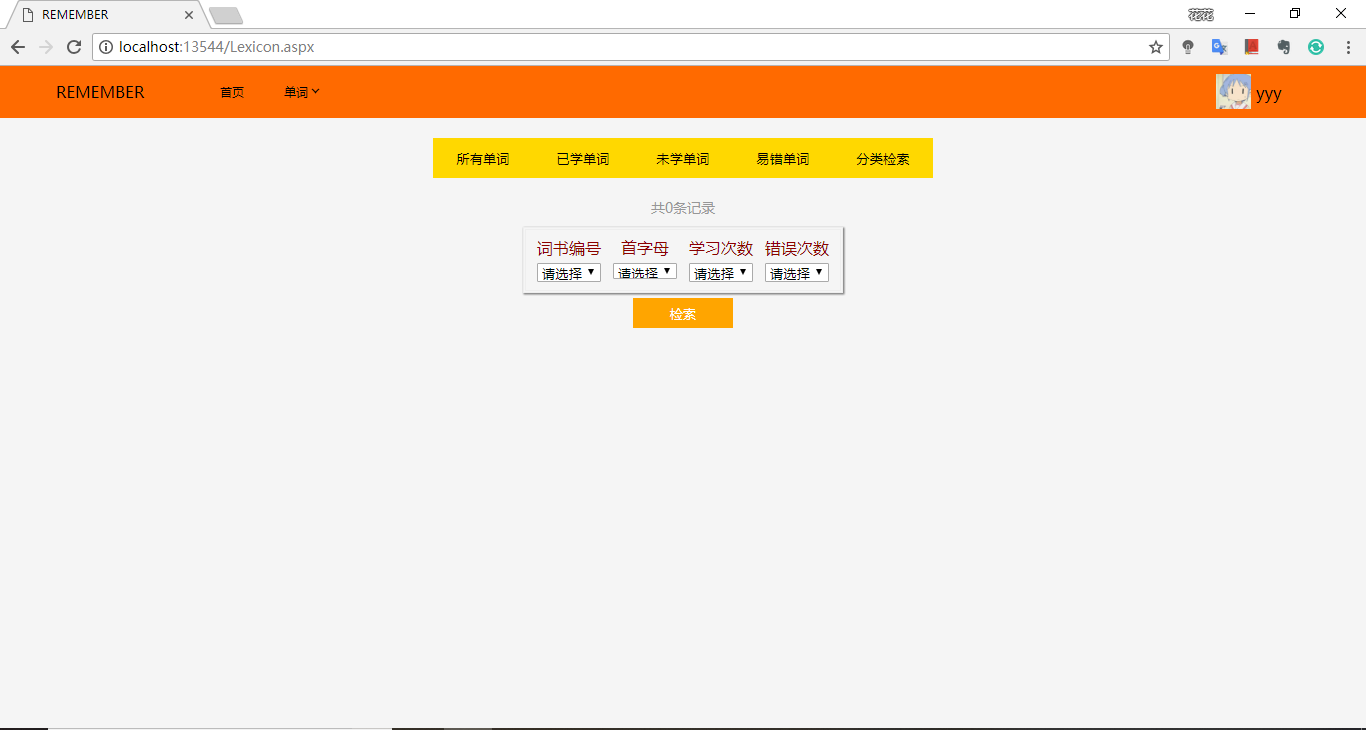
**图3-7 答题页面——答题情况总汇**

## 我的词库模块功能介绍

我的词库模块用于方便用户查看所添加的词库和每个单词的学习情况。其功能包括：所有单词，已学单词，未学单词，易错单词，分类检索。

使用步骤：

1. 进入我的词库模块后，界面如图3-8



**图3-8 我的词库页面**

1. 点击“所有单词”，查看用户添加的所有单词，如图3-9



**图3-9 我的词库页面局部——所有单词**

1. 点击“已学单词”，查看进行过学习测试的单词，如图3-10



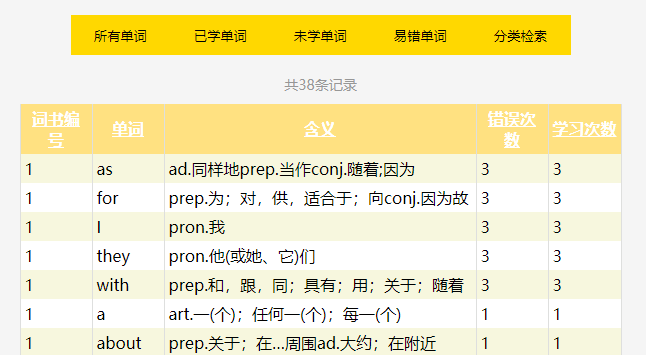
**图3-10 我的词库页面局部——已学单词**

1. 点击“未学单词”，查看未进行测试的单词，如图3-11



**图3-11 我的词库页面局部——未学单词**

1. 点击“易错单词”，查看进行学习过且答错的单词，并按错误次数从多到少进行排序，如图3-12



**图3-12 我的词库页面局部——易错单词**

1. 点击“分类检索”，按需查看用户所需要的单词信息，如图3-13



**图3-13 我的词库页面局部——分类检索**

代码实现：

主要使用的技术就是从数据库中筛选所需要的信息，显示到页面上，以显示“已学单词”为例。

使用asp:SqlDataSource控件从数据库中读取数据，如代码清单3-02所示

**代码清单3-02**

<asp:SqlDataSource runat="server" ID="sds\_learned" ConnectionString="<%$ ConnectionStrings:REMEMBERConnectionString %>"

SelectCommand="SELECT [b\_id], [word], [means], [learn], [errorNum] FROM [words] WHERE (([learn] &lt;&gt; @learn) AND ([u\_name] = @u\_name)) ORDER BY [learn] DESC, [b\_id], [word]">

<SelectParameters>

<asp:Parameter DefaultValue="0" Name="learn" Type="Int32" />

<asp:SessionParameter Name="u\_name" SessionField="u\_id" Type="String" />

</SelectParameters>

</asp:SqlDataSource>

# **总结**

REMEMBER是一个辅助用户背英语单词的系统，重点解决背单词过程中“认识却不会写”，“因拼写错误，考场失分严重”等常见问题。针对这些问题，作者设计对单词的测试学习采用“看中写英”的默写模式。这样设计的目的在于提高用户的单词拼写能力，保证考场上小分不失，也一定程度上提高用户的翻译能力。在单词学习测试部分，作者还增加了题目小题化，批改及时化，交互动态化的设计，致力于帮助用户能更好的掌握所学习的内容，能科学高效且有趣味地学习。

此系统还有一大亮点在于信息反馈的丰富化和个性化。无论是我的词库模块还是单词进度模块，都为用户实时的反馈当前的学习情况。系统所反馈的信息包括用户自己按需添加的词书信息，学习的单词信息，单词测试的反馈信息，用户自身的个人信息等。且为保护用户的个人信息，在设置中也添加了对各部分信息的清除修改操作，用户可根据自身需要进行修改、清除、重置。

我认为系统还存在的不足有：学习测试方法单一；词书资源少；缺少激励学习的机制；缺少社区机制等。

对我个人而言，我觉得本次的项目对我的帮助极大。从设计到实现，从前端到后台，从作品到文案，每一部分都有一定程度的提高。并且此次作品的创作背景正是因着自己有这方面的需要，因此也更能明确自己创作的方向与目标，也一定程度上给了我很大的动力要做好这个项目。

在项目的初期，主要工作内容在于整个系统的功能设计及实现。这个系统初始的定位是辅助英语学习，因此所涉及的面比较，要包括语法、单词、句子、文章等等。这样的工作量过于庞大且复杂，也不易于在三周内做精美，反而会使这个作品变得华而不实。因此，在做这部分的设计时，为了保证系统的实用性，我重点选择了一个方向，即针对背单词中的拼写问题进行设计。所以，纵观整个作品，其实都是围绕这一个问题进行设计的——除单词拼写测试以外，其他的功能都是它的辅助功能。这使得本作品的使用对象针对性强，且目标明确——克服单词拼写问题。

在项目的后期，我就主要关注在界面的观赏性，系统的交互性及系统设计的合理性上面。后期所做的大量工作是关于界面设计的调整，功能的微调和文案的准备。

总体，这次的项目做的还是满意的，时间过的还是充实的，是一次有意义的经历。