|  |
| --- |
| 哈尔滨工业大学（威海）计算机科学与技术学院软件工程系 |
| 软件开发实践报告 |
| 2019级 |
|  |

2021年07月12日-2021年07月23日

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 角色 | 学号 | 姓名 | 组内排序 | | 小组成绩 | 个人成绩 |
| 组长 | 2191110527 | 罗雯 | A | -0 |  |  |
| 组员 | 2191110529 | 张丽 |  |  |  |
| 组员 | 171110116 | 莫璧嘉 |  |  |  |
| 组员 | 2191110619 | 许智超 | D | -9 |  |

|  |  |
| --- | --- |
| 指导教师 | 🞏辛国栋 🞏王大顺 🞏韩希先 🞏朴学峰 ✓王金宝 🞏张 华 |

|  |  |
| --- | --- |
| 教师评语 |  |

目录

[1 哈尔滨工业大学（威海）软件开发实践任务书 II](#_Toc77812015)

[2 软件开发实践报告 I](#_Toc77812016)

[2.1 需求获取与分析 1](#_Toc77812017)

[**2.1.1** **产品结构** 1](#_Toc77812018)

[**2.1.2** **需求清单** 1](#_Toc77812019)

[**2.1.3** **领域模型 (lw)** 1](#_Toc77812020)

[**2.1.4** **用例图 (xzc)** 2](#_Toc77812021)

[**2.1.5** **原型 (lw)** 2](#_Toc77812022)

[2.2 系统设计 3](#_Toc77812023)

[**2.2.1** **功能列表 (xzc)** 3](#_Toc77812024)

[**2.2.2** **时序图 (zl)** 4](#_Toc77812025)

[**2.2.3** **技术栈** 4](#_Toc77812026)

[**2.2.4** **前后端api接口** 4](#_Toc77812027)

[**2.2.5** **数据架构 (mbj)** 4](#_Toc77812028)

[2.3 系统功能测试例及测试证据 (zl and mbj) 4](#_Toc77812029)

[**2.3.1** **已经测试的功能列表** 4](#_Toc77812030)

[**2.3.2** **测试用例及证明** 4](#_Toc77812031)

# 哈尔滨工业大学（威海）软件开发实践任务书

|  |
| --- |
| 组长姓名：  组长学号：  院（系）： 计算机科学与技术暨软件学院 专业：软件工程/服务科学  任务起止日期：2021年07月12日-2021年07月23日 |
| 课程设计题目： |
| 格式要求：中文标题采用黑体小四号，单倍行距，段前0.5行，段后0.5行；  正文采用宋体五号，英文Times New Roman 五号字体，单倍行距，段前0行，段后0行；  不得任意加空行；  蓝色字体属于提示信息，正式文档中请删除。  1. 系统描述  请用300字左右介绍系统背景和系统解决的问题。  2. 系统基本功能描述  请用介绍系统的基本功能。  3. 各类角色功能  请按照角色详细介绍每个角色需要的功能。  4.成员分工  按照指导书中进行纵向任务分工，说明每个成员分配的模块。  5.进度计划  请参考指导书中的时间节点，细化完成本课程实践需要的各项活动。 |

# 软件开发实践报告

内容要求：写必要的文档。自行安排章节，报告内容包含但不限于以下：

* 必须有需求获取和分析过程，包括
  + 产品结构的思维导图及说明
  + 给出版本需求清单
  + 给出版本需求清单限定的功能的领域模型
  + 给出版本清单限定功能的系统用例图（包括所有角色，及每个角色能够使用的功能，注意不得截图，用staruml直接粘贴拷贝，否则打印出来看不清楚，另外，注意extend和include的关系）
  + 给出系统用例图中的每一个用例的用例说明（也可以采用流程图或者活动图的方式表达该用例的业务流程，注意，此时一定将系统看出黑盒，是用户和系统的交互过程）
  + 给出原型（可以低保真，也可高保真。但必须与版本需求清单，系统用例图及用例说明高度一致）
  + 原型中，需要给出页面跳转关系。
  + 每页原型的交互说明。
  + 其他必要的，能够对需求进行分析和说明的内容，均可补充。
* 必须有系统设计内容
  + 功能列表（可分成多个层级并进行功能说明）
  + 针对每一个用例及用例说明，用时序图分析该用例对应的时序图。（即，用例实现。此时需要将系统展开，分析哪些对象交互才能完成该用例，因此，如果在本用例实现中，出现System字样的对象，必然是错的，要的就是把System分解开）
  + 软件的体系结构及其说明
  + 软件的技术栈（包括前端，后端，数据库，也可以包括Devops以及对外的接口，这些内容所需要的技术体系，参考指导书）
  + 前后端api接口（如有，可以通过swagger等工具导出，也可以自己设计表格，说明输入，输出，返回值等相关参数的格式要求）
  + 数据架构（ER图或json格式定义）
* 系统功能测试例及测试证据
  + 所有已经测试的功能列表（需要说明测了哪个功能，测试结果）
  + 针对已经测试的功能列表，对每一个功能，编写测试用例，并在测试用例后加上测试证明。（如果一个功能跨多个页面，需要把多个页面按照流程粘上，否则就是假的。）
* 报告总结
  + 有感而发，别扯淡。实在没啥说的，就别写。

格式要求：严格按照指导书要求撰写。请各位注意，格式是影响老师对报告质量判断的重要因素，部分小组具有因格式问题不及格的风险。

## 需求获取与分析

### **产品结构**

图示

描述已自动生成

**说明：①**生词管理系统可提供给游客**注册登录**功能。

**②**用户可以创建自己的词书，并且查找自己的词书，删除词书，或者修改词书内容。

**③**用户可查看自己的词书里的单词内容，或者通过模糊查询查找词书里的单词，增加或删除词书内单词，并可以查看词书里单词的复习情况。

**④**用户统计数据：可以查看自己词书的复习情况。

### **需求清单**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 分类 | 主要功能点 | 功能点拆分 | 功能特性描述 | 优先级 |
| 用户模块 | 注册、登陆 | 提取信息 | 从数据库提取用户信息。 | P0 |
| 数据库存储 | 将用户信息存入数据库 | P0 |
| 信息验证 | 从获取的用户信息验证合法性 | P1 |
| 修改用户信息 | 搜索用户 | 通过用户名查询到用户id | P0 |
| 修改信息 | 修改数据库该用户信息 | P1 |
| 词书模块 | 新建词书 | 数据库建立新词条 | 用户新建一个词书 | P1 |
| 修改词书 | 更新词书数据库 | 词书信息被修改更新 | P0 |
| 获取某个用户的词书 | 在数据库检索该用户数据库 | 用户查看自己的词书列表 | P0 |
| 删除词书 | 在数据库删除词条 | 删除掉用户的一个词书 | P0 |
| 搜索词书 | 在数据库索引词条 | 用户根据关键词查找特定词书 | P0 |
| 添加词书封面 | 为词书封面建立数据库 | 用户 | P0 |
| 获取词书复习情况 | 获取还没复习的单词数量，内容 | 用户可知道哪些单词还没复习 | P0 |
| 获取已经复习的单词数量，内容 | 用户可知哪些单词已经复习 | P0 |
| 单词模块 | 获取某个词书全部单词 | 检索词书内容 | 用户获得词书的全部单词 | P0 |
| 根据单词查询单词内容（模糊查询） | 以词书为id检索数据库查单词 |

### **领域模型 (lw)**

### **用例图 (xzc)**

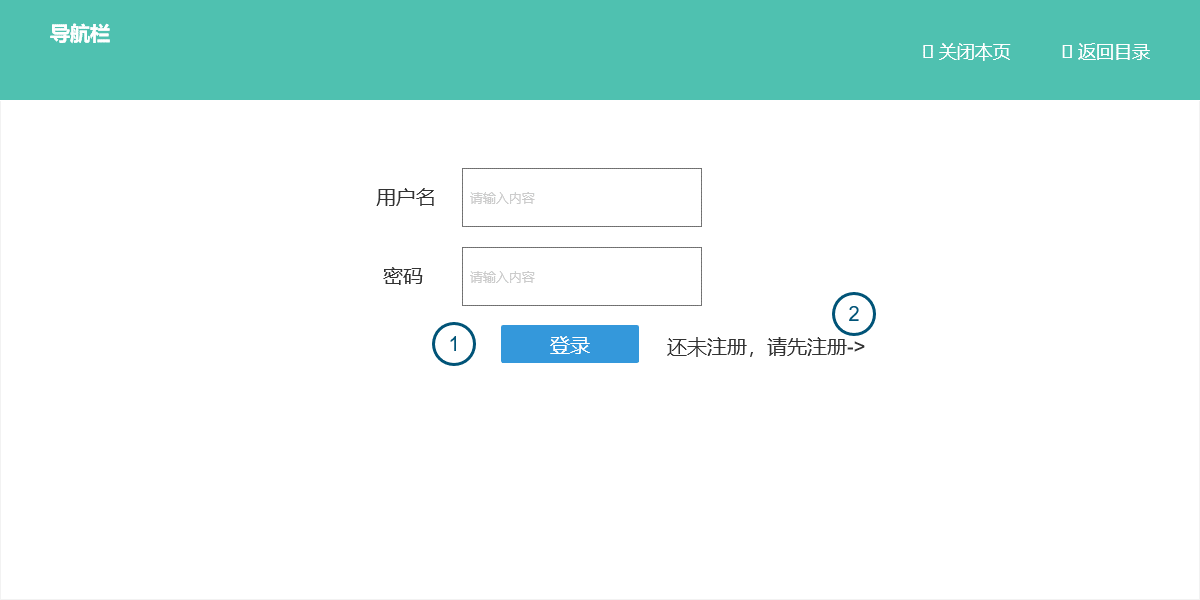


给出版本清单限定功能的系统用例图（包括所有角色，及每个角色能够使用的功能，注意不得截图，用staruml直接粘贴拷贝，否则打印出来看不清楚，另外，注意extend和include的关系）

用例图及用例图说明（黑盒，用户和系统的交互）

### **原型 (lw)**

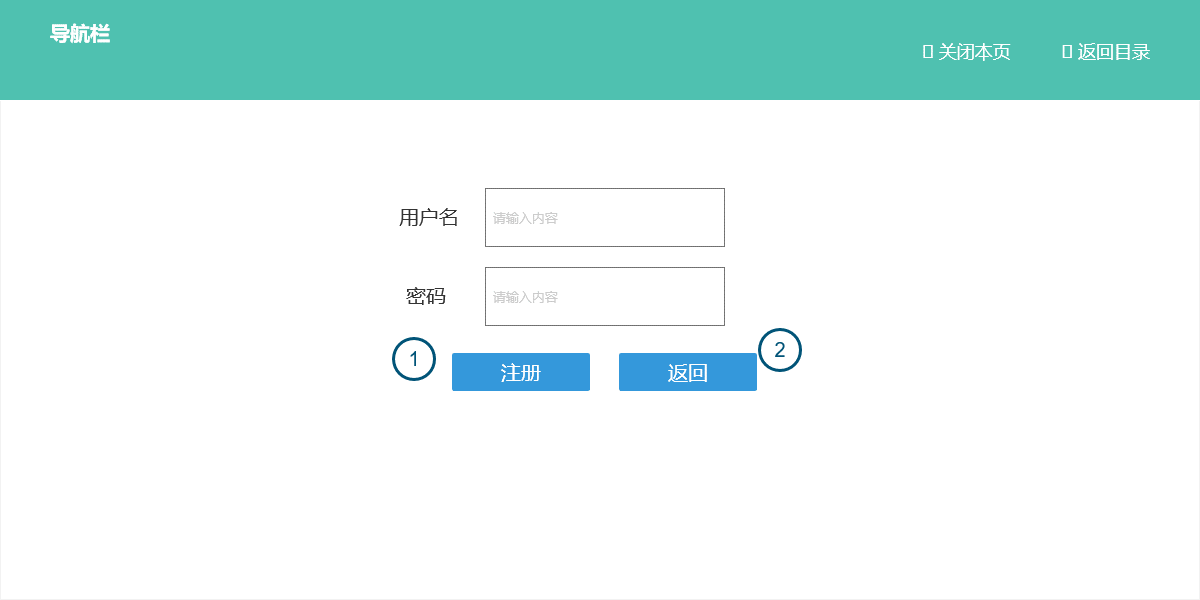
#### 登录界面



**①登录按钮：填写完毕用户名和密码之后，点击登录按钮，如果正确则跳转进词书主页面**

**②注册链接：点击后进入注册页面**

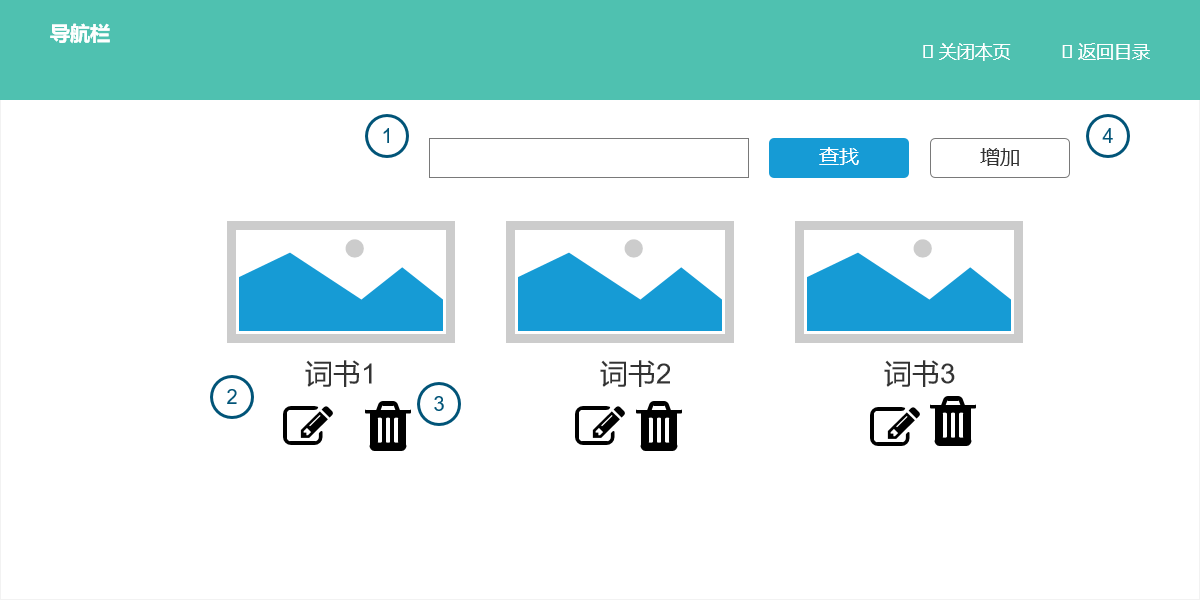
#### 注册页面



**①注册按钮：填写完毕用户名和密码之后，点击注册按钮，判断用户名是否存在，若无则注册成功，跳转至登录页面**

**②返回：点击后返回登录页面**

#### 词书主页面



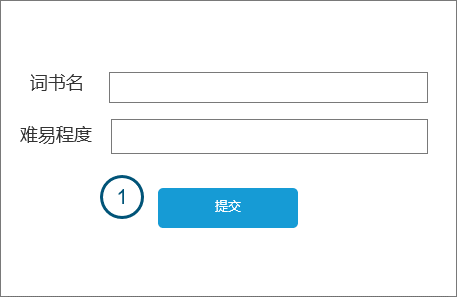
**①查找：填写查找的词书名，并点击查找按钮后，返回词书列表**

**②修改按钮：点击修改按钮，弹出修改词书页面**

**③删除按钮：点击删除按钮，将词书删除**

**④增加按钮：点击增加按钮，进入增加词书页面**

#### 修改词书页面



**①提交按钮：填写所需修改的内容后，点击提交按钮，进行词书信息的修改**

## 系统设计

### **功能列表 (xzc)**

### **时序图 (zl)**

* + 针对每一个用例及用例说明，用时序图分析该用例对应的时序图。（即，用例实现。此时需要将系统展开，分析哪些对象交互才能完成该用例，因此，如果在本用例实现中，出现System字样的对象，必然是错的，要的就是把System分解开）

### **技术栈**

前端：vue + elementUI

后端：SpringBoot + Mybatis-plus

数据库：Mysql

### **前后端api接口**

因内容较多，见压缩包内接口文档.docx文件

### **数据架构 (mbj)**

* + 数据架构（ER图或json格式定义）

## 系统功能测试例及测试证据 (zl and mbj)

**这个部分张丽丽和莫璧嘉分别测自己的部分吧~**

### **已经测试的功能列表**

* + 所有已经测试的功能列表（需要说明测了哪个功能，测试结果）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 功能名称 | 预置条件 | 测试项 | 输入 | 预期结果 | 测试结果 | 状态 |
| 登录功能测试 | 无 | 登录页面显示及登录功能 | 用户信息，包括用户名及密码 |  | 和预期相符 | P |
|  |  |  |  |  |  | P |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

### **测试用例及证明**

* + 针对已经测试的功能列表，对每一个功能，编写测试用例，并在测试用例后加上测试证明。（如果一个功能跨多个页面，需要把多个页面按照流程粘上，否则就是假的。）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能名称 | 预置条件 | 测试项 | 输入 | 测试用例 |
| 登录功能测试 | 无 | 登录页面显示及登录功能 | 用户信息，包括用户名及密码 | 用例1：正确的用户名及密码(“admin”, ”root”)  用例2：密码错误(“admin”, “123”)  用例3：用户名错误(“ad”, “root”)  用例4：未填写用户名(“”, ”123”)  用例5：未填写密码(“admin”, “”) |

测试用例1结果：

测试用例2结果：

测试用例3结果：

测试用例4结果：

测试用例5结果：