
<SJTU>

<水町语库>
软件架构文档

版本 <1.1>

<水町语库>	Version: <1.1>
软件架构文档	Date: <21/11/12>

修订历史记录

日期	版本	说明	作者
<11/11/2021>	<1.0>	完成底稿	李卓壕
<11/12/2021>	<1.1>	增加逻辑视图、进程视图、部署视图。	夏潇恒
<11/12/2021>	<1.1.1>	增加数据视图	王徐忆

<水町语库>	Version: <1.1>
软件架构文档	Date: <21/11/12>

目录

1.	简介	4
1.1	目的	4
1.2	参考资料	4
2.	用例视图	4
3.	逻辑视图	5
3.1	概述	6
3.2	在构架方面具有重要意义的设计包	6
4.	进程视图	错误! 未定义书签。
5.	部署视图	7
6.	实现视图	错误! 未定义书签。
7.	技术视图	7
8.	数据视图	8
9.	核心算法设计	8
10.	质量属性的设计	8

<水町语库>	Version: <1.1>
软件架构文档	Date: <21/11/12>

软件架构文档

1. 简介

1.1 目的

本文档将从构架方面对“虫出没”项目组的软件工程项目“水町语库”进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

该文档主要面向本项目的开发人员以供架构参考，说明了软件的设计层次和逻辑，指导本项目的开发与部署过程，有助于项目后期的维护。

1.2 参考资料

[1] 软件工程原理.2013年2月版.高等教育出版社.沈备军，陈昊鹏，陈雨亭编著

[2] System Design (Part1/2/3)，陈国兴，软件工程（课号：EE458）

2. 用例视图 (Use Case Diagram , based on UML 2.4)

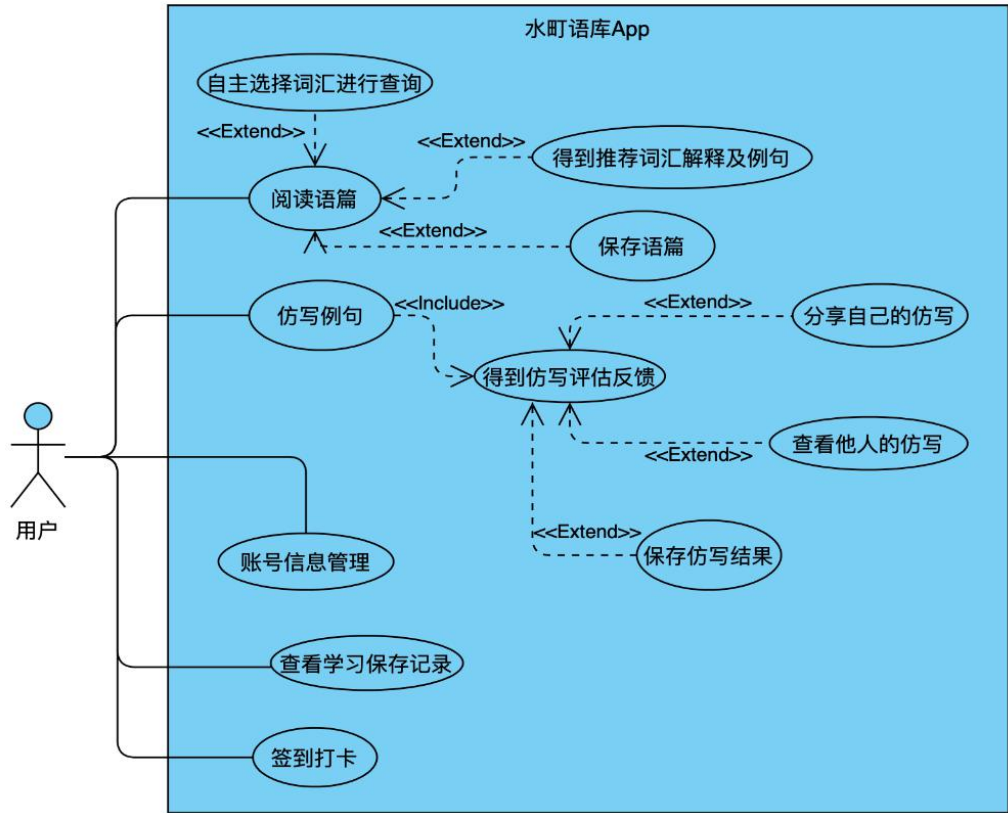
该视图列举并描述了该项目的一组重要的场景和用例，这些场景和用例代表了该项目将要实现的中心功能，这些用例的实现是每次迭代的工作焦点和中心。

它主要包括以下用例：

- 阅读语篇 - 自主选择词汇进行查询
- - 得到推荐词汇解释例句
- - 保存语篇
- 仿写例句 - 得到仿写评估反馈 - 分享自己的仿写
- - 查看他人的仿写
- - 保存仿写结果
- 账号信息管理
- 查看学习和保存记录
- 签到打卡

2.1 Use-Case 展示与说明

<水町语库>	Version: <1.1>
软件架构文档	Date: <21/11/12>



- 2.1.1 阅读语篇:

在进入软件界面后，用户需要阅读语篇。阅读结束后，用户需要自主选择词汇进行查询，软件根据选择的词汇得到推荐词汇解释及例句，同时保存语篇。
- 2.1.2 仿写例句:

用户仿写例句时，软件会向用户提供仿写评估反馈。用户可以在这个窗口分享自己的仿写、查看他们的仿写，并保存仿写的结果。
- 2.1.3 账号信息管理:

软件提供用户的账号信息管理。
- 2.1.4 查看学习保存记录:

用户可以在软件查看学习保存记录。
- 2.1.5 签到打卡:

软件也提供签到打卡的功能，用户每天学习后可以在此模块进行打卡。

3. 逻辑视图

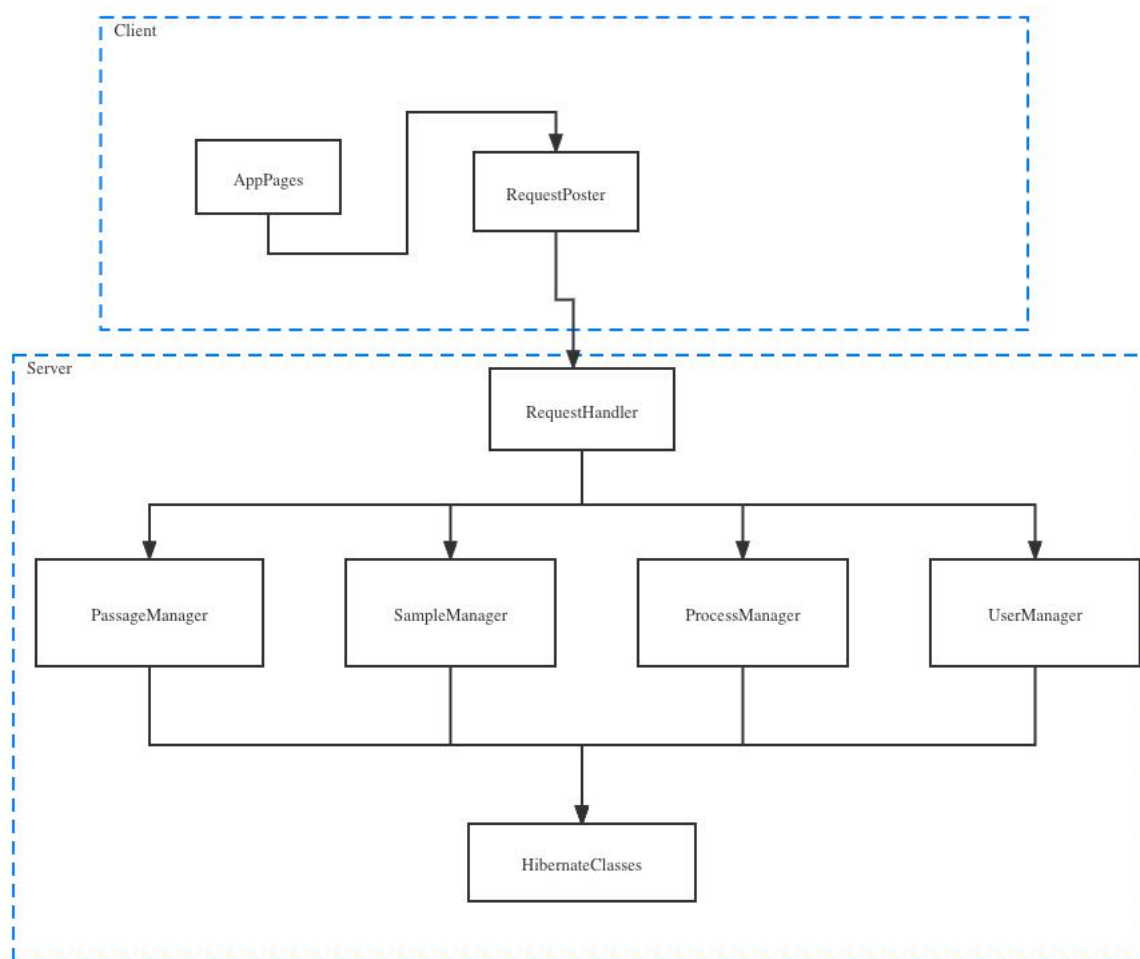
[本节说明设计模型在构架方面具有重要意义的部分，例如设计模型被分解为多个子系统和包。而每个重要的包又被分解为多个类和类实用程序。您应该介绍那些在构架方面具有重要意义的类，并说明它们的职责，以及几项非常重要 的关系、操作和属性。]

<水町语库>	Version: <1.1>
软件架构文档	Date: <21/11/12>

3.1 概述

逻辑视图用来描述软件面向用户端的功能性描述。本项目整体采用 C/S 的架构风格：

用户浏览 Android App 界面即 AppPages 包中的类，通过交互由 RequestPoster 中的各类发送请求至 Server 端，Server 端的 RequestHandler 包中的类诸如 Servlet 等类接收到请求并将相关信息根据内容转发给 PassageManager、SampleManager、ProcessManager、UserManager 包。其中 PassageManager 中的各类负责从后端数据库选择阅读文章，词汇学习的管理；SampleManager 用来管理仿写句子的评估和反馈，包括通过 HTTPS 和 web 对外的分享，查看他人的分享，用 NLP 和人工智能技术评价仿写例句的质量；ProcessManager 用来保存用户的学习进程，查看和保存学习记录等；UserManager 负责维护用户信息相关业务。



3.2 在构架方面具有重要意义的设计包

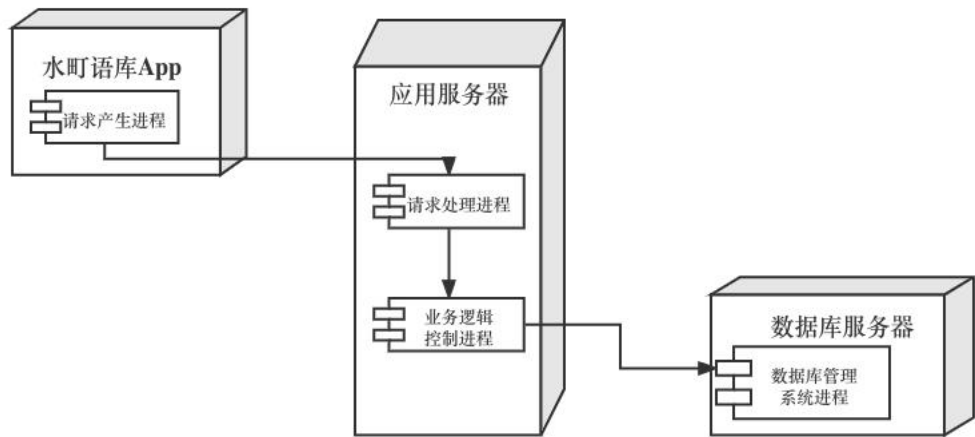
[对于每个重要的包，都用一个小节来加以说明，其中应包括该包的名称、简要说明以及显示该包中所有重要的类和包的图。]

对于该包中的每个重要类，应包括其名称、简要说明，还可选择包括对其部分主要职责、操作和属性的说明。]

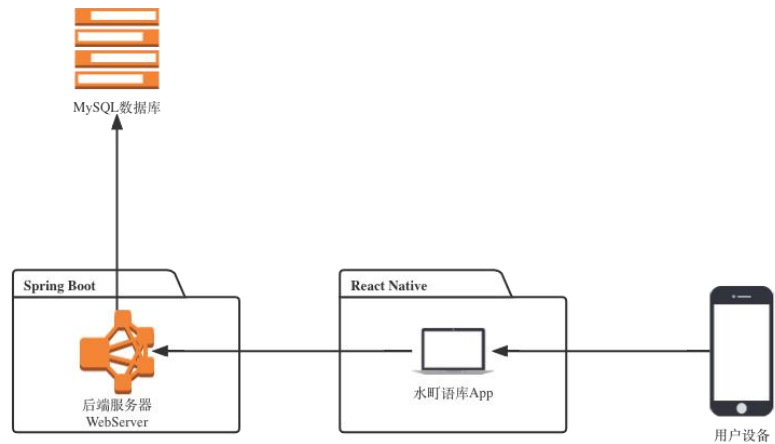
<水町语库>	Version: <1.1>
软件架构文档	Date: <21/11/12>

待定

4. 部署视图



5. 技术视图



软件语言： 前端开发使用 React-Native+ expo 框架， 语言使用 JavaScript， 后端使用 Spring 框架， 语言选择 Java 语言。

组件和依赖库： 前端开发使用 React 与 React-Native 组件库以及 ant-design 的组件库； 后端开发使用同一版本的 Java 库以及 Spring JPA。

开发工具的指定用途： Windows 使用 Idea IntelliJ 或者 Android Studio 进行软件开发， Mac 使用 Idea IntelliJ 结合 Xcode 进行软件开发， 利用 PowerDesigner 进行系统的建模， 统一使用华为云以及 Github 进行项目管理。

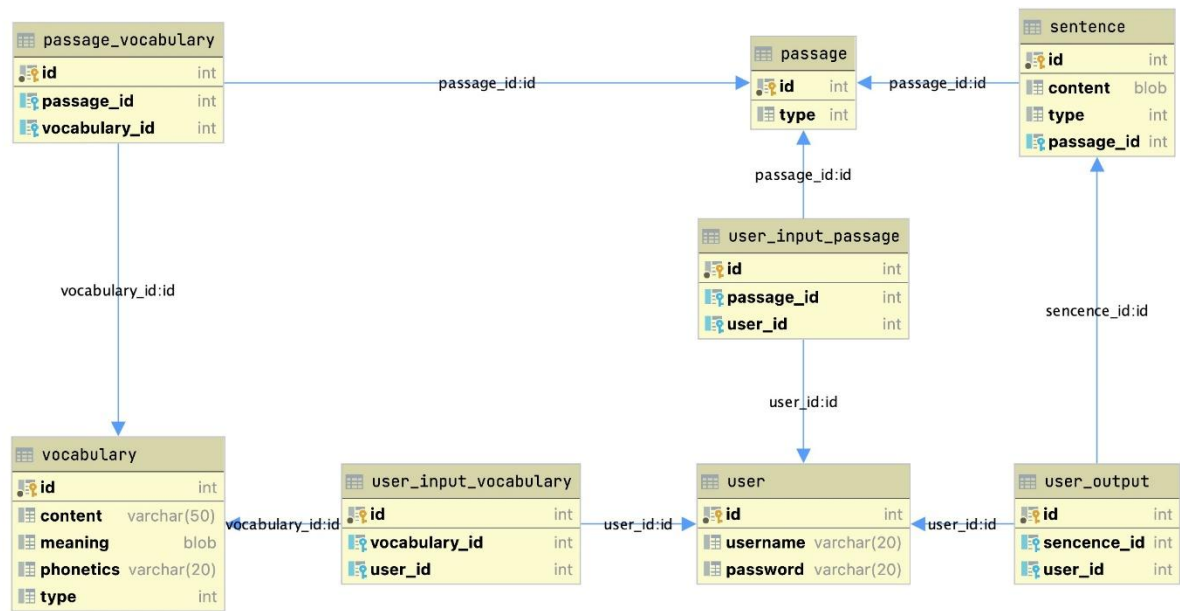
购买的构件： 待定。

数据库： MySql

<水町语库>	Version: <1.1>
软件架构文档	Date: <21/11/12>

中间件: Tomcat

6. 数据视图



7. 核心算法设计

待定

8. 质量属性的设计

8.1 易用性

8.1.1 用户快速上手使用

设计一个【开始使用】视频教程，使得使用水町语库的用户可以即刻上手，降低用户的学习成本。（参考：Notability）

8.1.2 用户分享例句社区

用户可以在例句分享社区和广大软件使用者分享仿写的例句

8.1.3 方便合理地得到软件例句反馈

不像市面上主流软件一样，用户主导整个学习过程，发挥主观能动性，主动学习。

<水町语库>	Version: <1.1>
软件架构文档	Date: <21/11/12>

8.2 可靠性

8.2.1 可靠性需求

可用性：用户可以正常阅读软件甄选出来的阅读语篇和推荐词汇，用户能够正常根据选出的词汇进行仿写和例句分享。用户能够正常参与打卡、学习词汇、仿写例句、收到例句反馈、分享例句、查看他人例句、保存用户的学习进度。在无软件升级或维护等特殊情况下保证 7*24 小时使用

平均故障间隔时间：24h

平均修复时间：1h

精准度：无

小错误：阅读语篇调出出错（2/1000 行代码）

大错误：仿写反馈评分故障（0/1000 行代码）

严重错误：数据丢失、软件界面弹出无效，点击无效（0/1000 行代码）

8.3 性能

8.3.1 性能需求

事务响应时间：

1. 用户普通的点击事件响应与页面跳转响应：平均：0.01s，最长：0.1s
2. 仿写例句的分享和查看：（根据网络因素决定）平均：0.5s，最长：1s
3. 得到推荐例句和阅读语篇：平均：0.5s，最长：1s
4. 软件反馈评分：平均：1s，最长：2s
5. 冷启动响应：平均：0.5s，最长：1s

吞吐量：

MIPS: 6000/s

容量：

1. 客户：可容纳客户数：100，并发服务客户数：20
2. 可容纳事务数：100 件

降级模式：

维护软件多个可运行版本，当高版本运行故障，亟待修复时，可以先及时降级到较低版本，保证软件的正常运行，同时进行故障的修复工作

资源利用：待定（受经费制约）

8.4 可支持性

8.4.1 可支持性需求

编码标准：采用 Google 编程规范

命名约定：统一采用驼峰命名法标准，并且利用版本管理工具与项目管理工具保证成员间开发过程中不会出现重名情况。

类库：前端开发使用 React 与 React-Native 组件库以及 ant-design 的组件库等；后端开发使用同一版本的 Java 库以及 Spring JPA。

维护访问权：系统服务器可以识别不同用户的权限类型，并给予不同的访问或维护的权限。

维护实用程序：待定