

背单词辅助系统 软件需求规格说明书

组长姓名：郑文涛

小组成员：沈大为

小组成员：喻悦

小组成员：张自成

小组成员：把硕

武汉大学计算机学院

目录

1	引言	4
1.1	编写目标	4
1.2	读者对象	4
1.3	文档概述	4
1.4	术语定义	4
1.5	参考文献	5
2	软件系统概述	5
2.1	软件产品概述	5
2.2	用户特征	5
2.3	设计和实现约束	6
2.4	假设与依赖	6
3	功能性需求描述	6
3.1	软件功能描述	6
3.2	软件需求的用例模型	6
3.3	软件需求的分析模型	8
3.3.1	软件需求的动态建模	8
3.3.2	软件需求的类建模	9
4	非功能性需求	9
5	界面需求	10
6	接口定义	10
6.1	登录接口	10
6.2	单词列表接口	11
6.3	添加单词接口	11
6.4	修改单词接口	12
6.5	删除单词接口	13
7	进度需求	13
8	交付要求	13
9	何种形式来交付	14

目 录	3
-----	---

10 验收要求	14
10.1 验收标准	14
10.2 验收流程	14

1 引言

1.1 编写目标

本报告，即背单词辅助系统软件需求规格说明书，明确了本软件（即“背单词辅助系统”）的开发的目与开发环境，介绍了本软件的所有功能以及适用范围，可以说是在正式开发之前对提前须知的软件需求进行定义与规范。

本报告明确阐释了本软件最终要实现的非功能需求、性能需求、运行环境需求等需求中，并在之前得到的初步的软件需求基础上进一步细化，建立了以面向对象为基础的软件需求分析模型。

这样一方面是为了向软件开发商提供系统的开发依据；另外则是为了以后相关的系统使用人员能够迅速熟悉该系统的具体业务功能，并作为今后系统的验收标准。

1.2 读者对象

本文档的预期读者是：软件工程授课老师，软件开发人员、软件测试人员、软件评审人员，软件用户。

1.3 文档概述

该文档描述了四六级背单词软件项目所使用的各类软件需求模型、非功能性需求、界面需求、接口定义、进度需求、交付要求和验收要求等内容。并且，本文档对该软件的功能、预期效果和需求分析所使用的方法进行了说明。

该软件项目由软件工程第 12 小组，郑文涛（组长），把硕，沈大为，喻悦，张自成完成。

1.4 术语定义

用例模型 定义一个参与者要用到的系统功能，同时描述系统为实现参与者价值所开展的行为序列，然后对参与者与系统之间的交互活动进行建模。从特定的用户角度出发，是完整的、实现特定用户价值的事件流。

状态转换图 状态转换图，即 STD 图 (State Transform Diagram)，表示行为模型。STD 通过描述系统的状态和引起系统状态转换的事件，来表示系统的行为，指出作为特定事件的结果将执行哪些动作（例如处理数据等）。STD 描述系统对外部事件如何响应，如何动作。

类的 UML 表示 类的 UML (Unified Modeling Language, 统一建模语言) 是一种用于图形化表示类、对象和它们之间关系的标准化符号表示方法。它可以用于软件设计和开发过程中, 用于描述系统的静态结构和动态行为。

SQLite SQLite 是一个进程内的库, 实现了自给自足的、无服务器的、零配置的、事务性的 SQL 数据库引擎。它是一个零配置的数据库, 这意味着与其他数据库不一样, 使用者不需要在系统中配置。

QT QT 是一个跨平台的 C++ 图形用户界面应用程序框架。它为应用程序开发者提供建立艺术级图形所需的所有功能。它是完全面向对象的, 容易扩展, 并且允许真正的组件编程。

1.5 参考文献

1. 《软件工程导论》第六版, 清华大学出版社, 2013
2. 《数据库系统开发》文东, 北京科海电子出版社, 2009
3. 《数据库系统实践》第二版, 机械工业出版社, 2012
4. GB 9385 《计算机软件需求说明编制指南》

2 软件系统概述

2.1 软件产品概述

该软件是一个独立的软件, 内容全部自含。

由于目前市面上的背单词软件存在联网和占内存较大等问题, 大学生需要一款单机轻量的背单词软件, 以方便他们在电脑上随时进行背单词练习, 同时解决易错单词的记忆问题。

该软件系统可以为用户提供单词背诵、易错词保存、自测模式、导入单词库等功能, 还可以实现单词的添加、删除、编辑等功能。用户可以自定义单词库文件。系统可以进行单词测试, 并提供易错词保存等功能。

2.2 用户特征

使用人员 本软件的用户主要包括在校学生用户为主的各类有语言学习需求的社会人士, 他们只需要简单的计算机操作知识且经过软件使用说明引导, 熟练图形界面的操作即可。

维护人员 维护人员只需要精通数据库相关知识，掌握整个系统的业务逻辑即可。本系统为大学生提供单词库和小测，将可能在一段时间内，每天都会使用，每次使用时间在 1 小时以内。

2.3 设计和实现约束

技术约束、资源约束、时间约束等如下所示：

运行环境：windows10 及以上

数据库：SQLite(64-bit DLL x64) for SQLite version 3.41.2

集成开发环境：Visual Studio 2022

高级编程语言：C++

图形用户界面（GUI）：QT

运行寿命：至少 3 年

开发期限：2 个月

硬件环境：本系统仅在 PC 端上运行，暂未考虑移动端

2.4 假设与依赖

假设 windows10 及以上版本的系统支持数据库 SQLite、编码工具 visual studio 2022、QT 前端框架的运行环境。

3 功能性需求描述

3.1 软件功能描述

轻量级 即点即用，不需联网，本地化运行，主打一个轻便高效。

单词记忆及巩固功能 可以跳过单词、判断单词、回退单词以及生成巩固单词，方便大学生查看掌握单词和温习易错单词。

背诵计划的生成功能 可以设定背单词的范围，词库选择多样化，既可以使用系统备用的单词库也可以自己导入新的单词库，方便大学生制定符合自身情况的计划，具有高灵活度和响应度。

单词测试功能 为大学生提供自我测试的功能，方便温故知新。

3.2 软件需求的用例模型

场景描述如下所示：

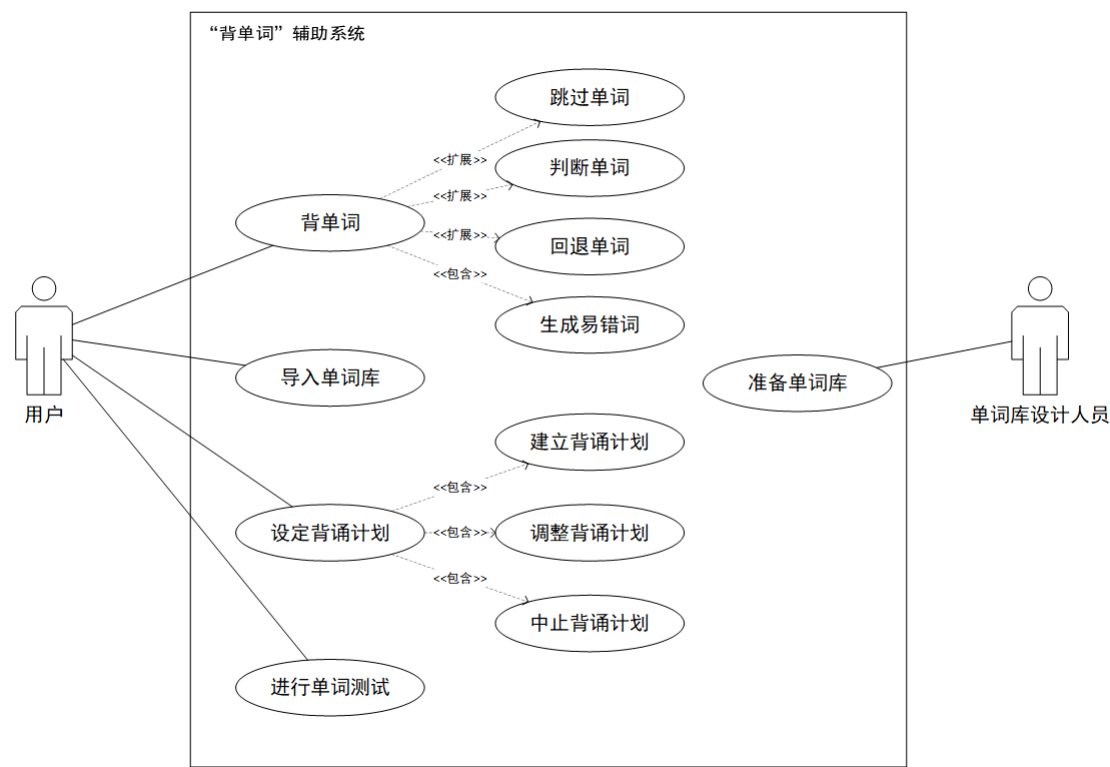


图 1 背单词辅助系统的用例图

Fig. 1 Use case diagram of word memorization assistance system

准备单词库 软件设计人员在设计软件时准备了一个标准的四六级单词库。

导入单词库 学生不使用软件自带的单词库，选择导入自己准备的单词库文件背单词。

背诵单词 学生导入单词库，或者使用系统准备的单词库，并且制定好背单词计划后开始背单词。在背单词过程中，学生可以选择回退到上一个单词，或者跳过当前单词。过程中，软件还会总结学生背单词时发现的易错词，并生成易错词本。

设定背诵计划 学生可以建立量身的背诵计划，并且可以选择调整或终止背诵计划。

进行单词测试 学生在开始背单词计划后，每隔一周时间就检验一次背单词的成果。检验方法是通过检查学生已经背过的单词。

3.3 软件需求的分析模型

3.3.1 软件需求的动态建模

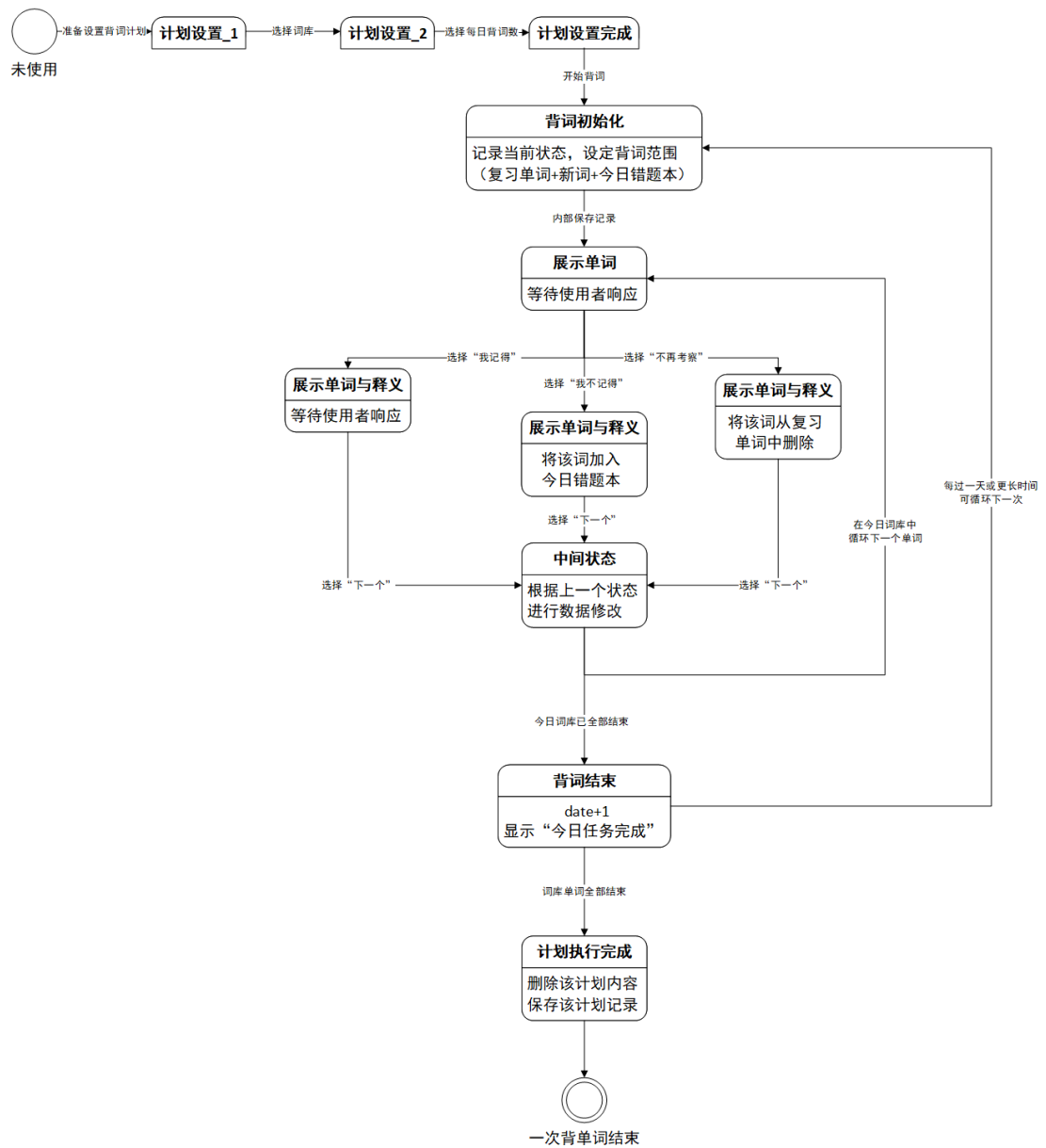


图 2 背单词用例的状态转换图

Fig. 2 State transition diagram for word memorization use case

最初为未使用状态，当用户进行计划设置（选择词库及背词数目）后进入背词初始化状态，通过系统内部保存的记录展示单词，用户进行选择（我记得或我不记得或不再考虑）后进入中间状态，随后循环背词直至单词背诵目标完成，进入背词结束状态。

3.3.2 软件需求的类建模

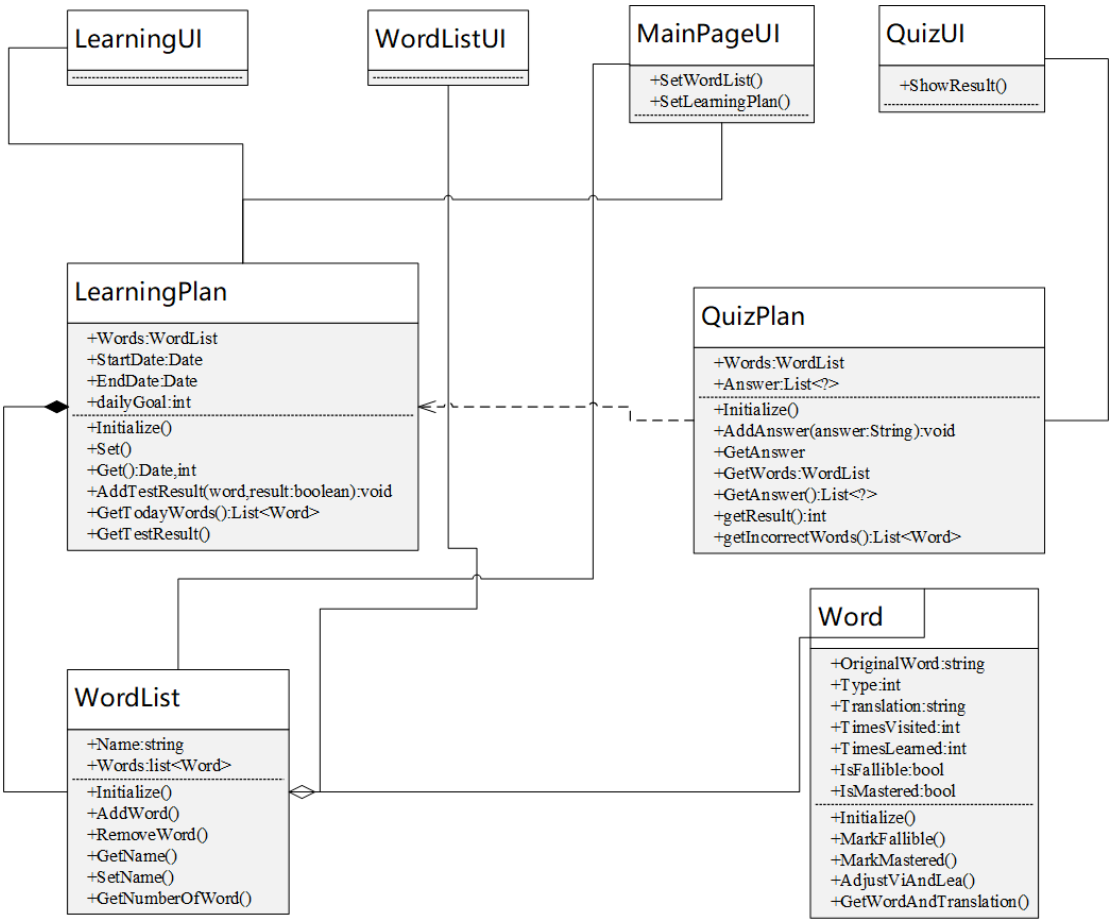


图 3 背单词辅助系统中的类图

Fig. 3 Class diagram in word memorization assistance system

我们设计的这些类的释义如上图中的英语所示，这些类的关系如下：

“WordList”类与“MainPageUI”类相关联，“LearningPlan”类、“LearningUI”类和“MainPageUI”类互相关联，“QuizUI”类和“QuizPlan”类互相关联，“Word”类和“WordListUI”类聚合于“WordList”类，“QuizPlan”类依赖于“LearningPlan”类，“WordList”类是“LearningPlan”类的组成部分。

4 非功能性需求

- 1. 使用体验好，界面友好，操作简单易学；
- 2. 软件占用空间小，加载速度快，离线应用不会弹出广告等；
- 3. 对于背单词效果具有较好的检查效果，可回退或跳过等操作，可保存易错词。

除此之外，该软件也具有有一些一般性的性能要求：

安全性 用户的单词库和学习记录应该得到保护，只有授权用户可以访问和修改。每次使用更改单词库时需要检查是否为用户主动更改。

可靠性 软件应该稳定运行，不应该因为程序崩溃或其他原因导致数据丢失或破坏。

易用性 软件应该易于使用，用户可以快速上手并理解如何使用各个功能，且易用性要求强于兼容性等要求。

高效性 软件应该快速响应用户的操作，应能在用户操作 1s 内给出结果，特别是在背单词和检查单词时。

兼容性 软件应该兼容 PC 端平台，在未来考虑向移动端方向移植。

5 界面需求

首页 该系统的门面。展示相关重要信息，例如用户的学习进度，推荐的单词列表等等。

单词查询页面 这个页面允许用户输入一个单词，并返回单词的释义、例句等信息。

词汇表管理页面 这个页面允许用户创建、编辑、删除自己的词汇表。

学习计划管理页面 这个页面允许用户创建、编辑、删除学习计划。

测试评估页面 这个页面允许用户进行测试评估，例如填空、选择、连线等测试形式。

6 接口定义

6.1 登录接口

URL: /api/login

Method: POST

Request Body:

```
{  
  "username": "string",
```

```
"password": "string"
}
Response Body:
{
  "token": "string",
  "user_id": "string"
}
```

6.2 单词列表接口

```
URL: /api/words
Method: GET
Request Headers:
{
  "Authorization": "Bearer token"
}
Response Body:
{
  "total": 100,
  "words": {
    {
      "word": "string",
      "meaning": "string",
      "pronunciation": "string",
      "example": "string",
      "category": "string",
      "learned": true
    }
  }
}
```

6.3 添加单词接口

```
URL: /api/words
Method: POST
Request Headers:
{
  "Authorization": "Bearer token"
```

```
}  
Request Body:  
{  
  "word": "string",  
  "meaning": "string",  
  "pronunciation": "string",  
  "example": "string",  
  "category": "string"  
}  
Response Body:  
{  
  "word_id": "string"  
}
```

6.4 修改单词接口

```
URL: /api/words/:word_id  
Method: PUT  
Request Headers:  
{  
  "Authorization": "Bearer token"  
}  
Request Body:  
{  
  "word": "string",  
  "meaning": "string",  
  "pronunciation": "string",  
  "example": "string",  
  "category": "string",  
  "learned": true  
}  
Response Body:  
{  
  "word_id": "string"  
}
```

6.5 删除单词接口

URL: /api/words/:word_id

Method: DELETE

Request Headers:

```
{  
  "Authorization": "Bearer token"  
}
```

Response Body:

```
{  
  "word_id": "string"  
}
```

7 进度需求

需求分析阶段 在四月九日之前完成需求分析和规格说明书的编写。

设计阶段 在两周内（或老师安排的计划时间内）完成系统架构和模块划分的设计。

编码和测试阶段 在二至三周内完成代码编写、单元测试以及相关的优化测试。

验收和交付阶段 一周内完成系统验收测试、用户手册和操作指南的编写以及最终的系统部署和交付。

8 交付要求

软件交付 提供完整的源代码和相关文档；需要包含详细的开发文档、部署文档和用户手册；提供部署程序和相关脚本，方便用户进行部署和安装。

软件保证 确保软件功能完整、可靠、稳定，符合需求规格说明书中的要求；确保软件运行在常见的操作系统和浏览器上，并与其他相关系统兼容。

软件维护 提供一定时间的软件维护和技术支持服务；在用户反馈问题时，及时响应并解决问题；对软件进行升级和维护，确保软件长期稳定运行。

软件培训 在交付软件后，提供相关的培训服务，帮助用户了解软件的使用方法；在软件使用过程中，为用户提供一定的技术支持和帮助。

9 何种形式来交付

交付形式 交付形式可以可执行文件或源代码的形式交付软件；将软件代码和文档打包成压缩文件，或者使用 Git 或 SVN 等版本控制工具进行交付；如果需要，可以通过网络或邮件等形式进行交付。

部署方式 可以将软件部署在云服务器或本地服务器上；可以将软件打包成 Docker 容器进行部署；如果需要，可以提供详细的部署文档和脚本，方便用户进行部署和安装。

10 验收要求

10.1 验收标准

软件的功能符合需求规格说明书中的要求；软件运行稳定，不存在明显的漏洞和错误；软件的性能符合用户的要求和预期；软件的界面友好、易于使用。

10.2 验收流程

开发人员在完成开发后进行初步测试，确保软件符合基本功能和性能要求；用户进行验收测试，测试软件的功能、性能和用户界面等方面；针对用户反馈的问题和建议，开发人员进行修复和改进；确认软件的交付和部署，结束验收流程。