项目计划书——WordKeeper(单词管家)

厦门大学第十三届软件设计大赛初赛材料

DAOO 参赛组

组员：周兴、林联辉、陈丰铧

目录

1.引言

1.1 编写目的

没有规矩不成方圆，无论什么事情，要顺利地完成，必须有一个统一的计划指导书。 软件项目开发也不例外。 这个计划书不仅能让参与项目的开发者们知道 如何进行，还明确了他们各自的职责、 保证项目团队之间的协作更加的有条不紊、 使得项目工作的各个过程能够合理有序地进行。

同时，计划书也能让团队内外的沟通起着向导作用、团队之间的工作范围、开发模块之间的关系， 以及对开发进度、经费预算、分配人力物力、风险等因素进行了大概的描述。本项目开发计划用于从总体上指导 WordKeeper 项目顺利进行并最终得到通过评审的项目产品。本项目开发计划面向项目组全体成员。

1.2 背景

随着国际化的发展，英语的学习也越发地重要。所以针对此现状，并以移动学习的核心参与群体———大学生为对象，对新型大学英语词汇学习网站的设计提出设想。

但是，近些年，英语学习类的软件应用特别词汇类数量不断增加，但是能够脱颖而出却很难，项目死亡率也在不断上升，整个应用领域的竞争越来越激烈。所以，进行软件的优化设计和深入创新是重中之重。而英语学习软件在大学生的学习生活中扮演着重要的角色，因为大学生的生活自由度比较高，一款能符合大学生英语学习特点的英语软件能够更好地帮助大学生提高自身英语水平。因此值得我们进行在该领域进行创新，所以我们想到利用人工智能技术结合传统的英语词汇学习网站的模式，进行合理创新，实现英语学习类软件的经济效益和最大程度发挥其学习作用的共赢。

1.3 定义

专业术语	定义与解释
MySQL	MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，属于 Oracle 旗下产品。
深度学习	深度学习(DL, Deep Learning)是机器学习(ML, Machine Learning)领域中一个新的研究方向，它被引入机器学习使其更接近于最初的目标——人工智能(AI, Artificial Intelligence)。
SpringBoot	Spring Boot 是由 Pivotal 团队提供的全新框架，其设计目的是用来简化新

	Spring 应用的初始搭建以及开发过程。
Bootstrap	Twitter 推出的一个用于前端开发的开源工具包。
Maven	Maven 项目对象模型(POM)，可以通过一小段描述信息来管理项目的构建，报告和文档的项目管理工具软件。
Mybaitis	MyBatis 本是 apache 的一个开源项目 iBatis，2010 年这个项目由 apache software foundation 迁移到了 google code，并且改名为 MyBatis 。

1.4 参考资料

《软件项目管理》 Rajeev T Shandilya 编著科学出版社。
软件工程国家标准文档
软件工程项目开发文档范例

2.项目概述

2.1 项目目标

本次的 WordKeeper（单词管家） 项目分三个重要过程，重要过程如下:
第一: 进行可行性分析，以及需求建模。并选择合适的开发环境与开发工具进行开发，并做好队伍人员分工，安排好进度。
第二: 在进度安排与明确分工下，尽快搭建起基本的程序框架，并根据需求工程得出的功能点，设计合适的 UI 界面，并完善后端的基本功能。
第三: 完善基于深度学习的特色功能点，完善相关算法，探索需求遗漏之处，做好软件测试。

2.2 功能需求

帮助学生记忆单词，以做选择题的模式达到沉浸式记忆的效果
为用户每次推送一篇优质的英文的文章，进行英语学习的扩展
词汇查询的功能
输入单词，利用神经网络生成一副图片，利用图像化抽象而具体，加深记忆。
根据用户行为，为用户指定独一无二、最懂用户的学习记忆方案
输入一篇文章，为用户分析出值得学习的高频词汇

口语与作文的评分功能
用户可以建立自己的生词库
账号的注册与注销

2.3 项目范围管理计划

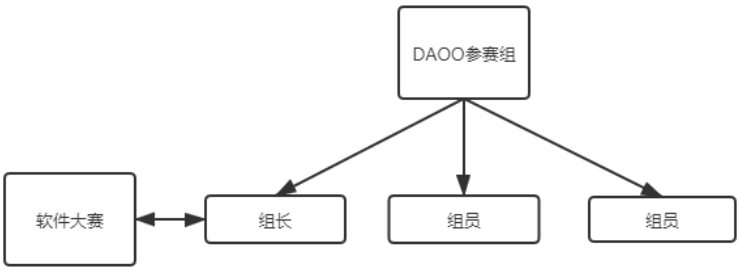
围绕着 WordKeeper(单词管家)的实现过程中，所进行的项目构建活动、学习活动、以及管理活动如进度安排等。

2.4 项目开发环境

操作系统：Windows 10
浏览器：Google Chrome
后端:SpringBoot+Mysql+Mybaitis 的 Maven 项目
前端：Bootstrap、jQuery
深度学习部分：Torch, CuDNN,

3.项目团队组织

3.1 组织结构



图：组织构架图

民主式组织结构，在这个结构中，小组成员完全平等，名义上的组长与其他成员没有任何区别。大家享有充分的民主，项目共作由全体人员讨论协商决定，并根据每个人的经验和能力进行适当的分配。充分 激发大家的创造力，有利于攻克技术难关，虽然缺乏明确的权威领导，但是出现意见分歧时大家都会尽量协商解决的。

3.2 角色与职责划分

角色	职责	负责人员
需求分析员	整理 WordKeeper 系统需求分析并以撰写需求分析分析文档	周兴、林联辉、陈丰铧
软件前端设计员	负责 WordKeeper 系统的前端设计并进行开发	林联辉、陈丰铧
软件后端开发人员	编写 WordKeeper 系统开发的后端代码	周兴、林联辉
算法实现人员	实现 WordKeeper 系统所需要的深度学习以及数据挖掘的算法	周兴、林联辉、陈丰铧
总结人员	负责最后的收尾工作并撰写项目文档	周兴

4.项目可行性

4.1 可行性分析

通常在做项目的时候，要求是客户的需求，只有我们要对客户的要求做到充分的理解， 理解到位，那么团队对项目的目标才是明确的，才对后期的项目制作很有帮助。虽然此次我们的项目是针对比赛，但是一样适用，我们也必须要进行可行性分析，将比赛委员会当作隐形的客户，做好可行性分析，才能够目标明确，顺利完成项目。

1.对自身要求：

明确自身方向，有针对性的大范围收集资料，了解市场上同类产品的现状，分析各个产品的优点与不足。提前熟悉开发流程，探索必须的准备工作，以及学习在后续开发过程中需要学习的东西。

2.完成期限：

由于赛制的要求，初赛要求在 2020 年 2 月 17 日晚进行初赛材料的提交。决赛的作品提交的截至时间是 2020 年 3 月 15 日晚。故根据赛制，我们进行以下严格的项目时间要求：

项目开始时间为 2020 年 1 月 25 日，第一阶段结束在 2020 年 2 月 16 日，整个项目工程完成在 2020 年 3 月 14 日前结束。

4.2 决定可行性的主要因素

1、技术人员：需要过硬的前后端开发的编程人员

2、UI 设计人员：一个好看且好用的 UI 是软件项目的门面
3、项目开发负责人：组织工程的推进，分工、进度管理都需要有人来负责。
4、时间：根本要素

5.软件设计

5.1 软件内容设计

5.1.1 数据字典与 ER 图

注：由于单词库在不断的扩充中，会有多个单词库，所以在此不对单词库进行数据字典的描述。

1. 关于单词的表
   由于采用的单词库的资源不同，建立不同的表。

2. 用户信息

字段名	类型	描述	其他约束
id	bigint(9)	用户 id（自增得来）	PRIMARY KEY AUTO_INCREMENT
name	varchar(31)	用户名	NOT NULL
user_photo	varchar(255)	头像 url	DEFAULT NULL
role_id	tinyint(1)	用户权限	0、1、
passw	varchar(255)	用户密码	密文存储 NOT NULL
email	varchar(255)	用户邮箱	NOT NULL
tel	varchar(11)	用户电话号码	NOT NULL
vocabulary	bigint(9)	词汇量	DEFAULT NULL
num_everyday	bigint(5)	每天单词学习数	DEFAULT NULL
books	varchar(255)	所选单词书	DEFAULT NULL
gmt_create	datetime(2)	表项创建时间	DEFAULT NULL
gmt_modified	datetime(2)	表项最后一次的修改时间	DEFAULT NULL
is_deleted	tinyint(1)	是否被逻辑删除	unsigned DEFAULT '0'

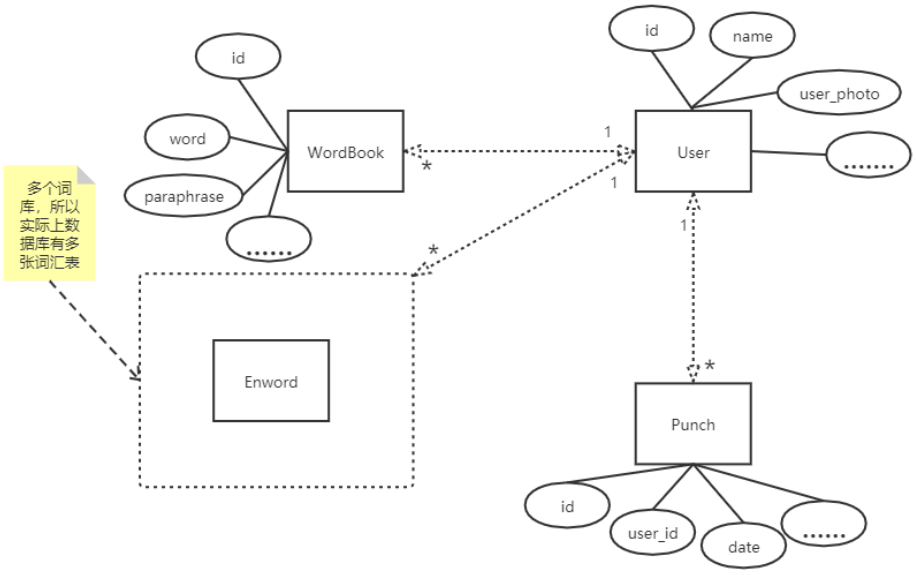
3. 打卡记录

字段名	类型	描述	其他约束
id	bigint(9)	记录 id（自增得来）	PRIMARY KEY AUTO_INCREMENT
user_id	bigint(9)	用户 id	NOT NULL

date	datetime(2)	打卡时间	NOT NULL
nums	bigint(5)	连续打卡的次数	NOT NULL
gmt_create	datetime(2)	表项创建时间	DEFAULT NULL
gmt_modified	datetime(2)	表项最后一次的修改时间	DEFAULT NULL
is_deleted	tinyint(1)	是否被逻辑删除	unsigned DEFAULT '0'

4. 自定义生词本

字段名	类型	描述	其他约束
id	bigint(9)	生词 id（自增得来）	PRIMARY KEY AUTO_INCREMENT
word	varchar(255)	单词或者短语	NOT NULL
paraphrase	varchar(255)	释意	NOT NULL
remark	varchar(255)	备注	DEFAULT NULL
user_id	bigint(9)	用户 id	NOT NULL
date	datetime(2)	创建时间	NOT NULL
gmt_create	datetime(2)	表项创建时间	DEFAULT NULL
gmt_modified	datetime(2)	表项最后一次的修改时间	DEFAULT NULL
is_deleted	tinyint(1)	是否被逻辑删除	unsigned DEFAULT '0'



图：实体关系图

5.1.2 软件功能点

- 注册登录、
- 选择题练题（利用深度学习根据单词生成图片）、
- 莎士比亚古典英语转化（娱乐功能）、
- 词典功能、
- 打卡记录功能、
- 创建生词本，
- 并记录生词、
- 给用户推送英文阅读、
- 输入文章词频统计并挑选合适的单词进行学习、
- 自定义学习计划、
- 邮箱定时提醒功能

5.1.3 典型用例

用例 1:

注册:

1. 用户输入注册信息
2. 系统检测信息是否合法
3. 用户注册成功并返回登录界面

用例 2:

登录:

1. 用户输入登录信息
2. 系统检测用户名密码是否正确
3. 用户登录成功进入个人主页

用例 3:

背单词:

1. 用户添加单词 (见用例 1a)
2. 系统制定复习计划
3. 用户复习单词 (见用例 3a)
4. 用户通过测试完成本轮复习 (见用例 4a)

1a. 添加单词:

场景 1:

1. 用户选择将一本单词书添加进生词本

场景 2:

1. 用户将自定义的生词添加进生词本

场景 3:

1. 用户输入一段文字
2. 系统检测文字中可能存在的生词

3. 系统将生词添加进生词本

3a. 复习单词：
1. 系统基于用户数据为用户制定复习计划
2. 用户根据自身情况对单词进行标记

4a. 测试：
1. 系统为用户每日复习的单词进行图文出题
2. 用户选择符合题意的图片
3-1. 用户通过测试进行打卡
3-2. 用户测试失败重新进行测试

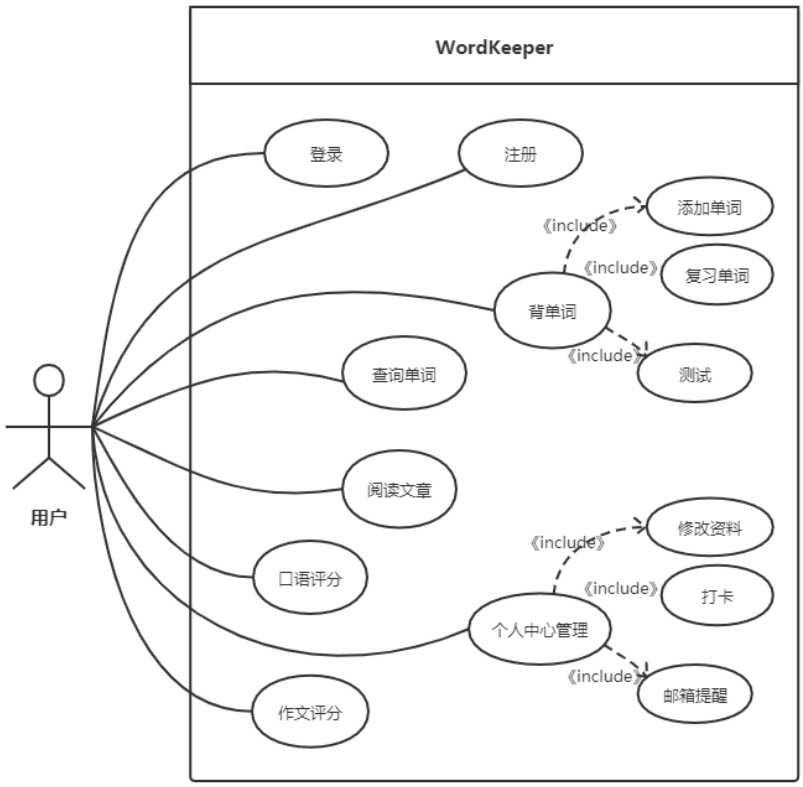
用例 4:
查询单词：
1. 用户输入单词进行查询
2. 系统显示查询到的单词词义并生成对应的图片

用例 5:
阅读文章：
1. 系统推送主流媒体文章
2. 用户选择文章进行阅读

用例 6:
口语评分：
1. 用户选择系统一篇文章
2. 用户上传该文章中的一段口语录音
3. 系统显示用户录音得分

用例 7:
作文评分：
1. 用户上传一段文字
2. 系统检测文字并给出该段文字的得分

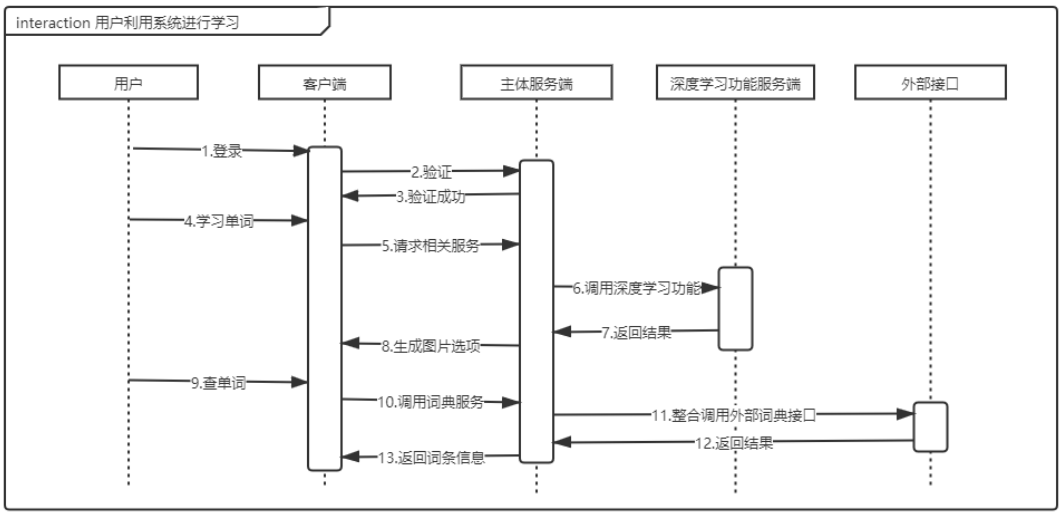
用例 8:
邮箱提醒：
1. 用户设置提醒功能开启
2. 系统检测到用户超时未进行学习复习，发邮件进行提醒。



图：系统功能用例图

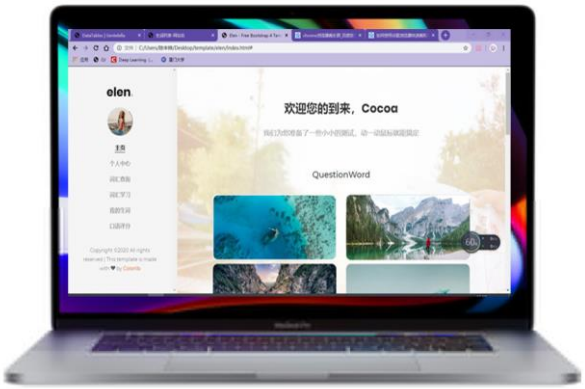
5.1.4 系统时序图

注：本部分以一个典型用户在使用系统的基础功能的场景画的时序图。



图：系统基础功能的时序图

5.1.5 用户界面设计

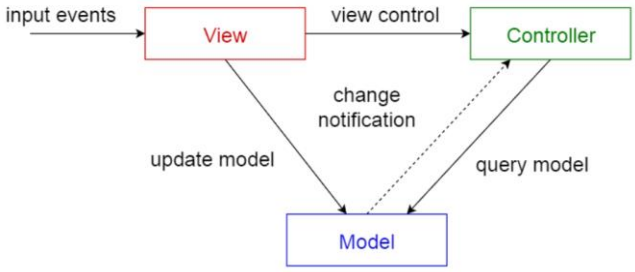


图：界面风格示意图

注：其他的界面设计详细内容见同材料包的截图

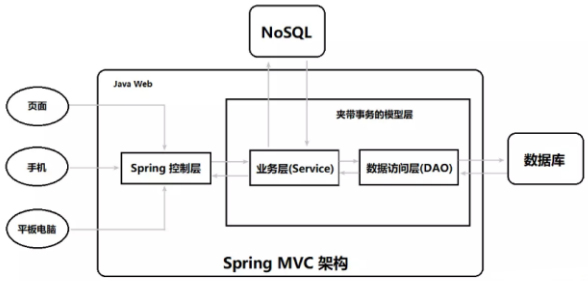
5.2 软件架构设计

由于本项目主体采用的 Spring MVC 进行开发的，故主体软件构件采用的是模型-视图-控制器模式，该模式也叫 MVC 模式，划分交互程序为 3 个部分：模型——包含核心功能和数据，视图——显示信息给用户（多个视图可被定义），控制器——处理用户输入。它通过分割用户信息的内部陈述和呈现、接受方式来实现，解耦组件并允许高效的代码复用。



图：MVC 构架示意图

相较于传统的 MVC,Spring MVC 使得传统的模型层被拆分为了业务层(Service)和数据访问层（DAO,Data Access Object)。在 Service 下可以通过 Spring 的声明式事务操作数据访问层，而在业务层上还允许我们访 NoSQL ，这样就能够满足异军突起的 NoSQL 的使用了，它可以大大提高互联网系统的性能。

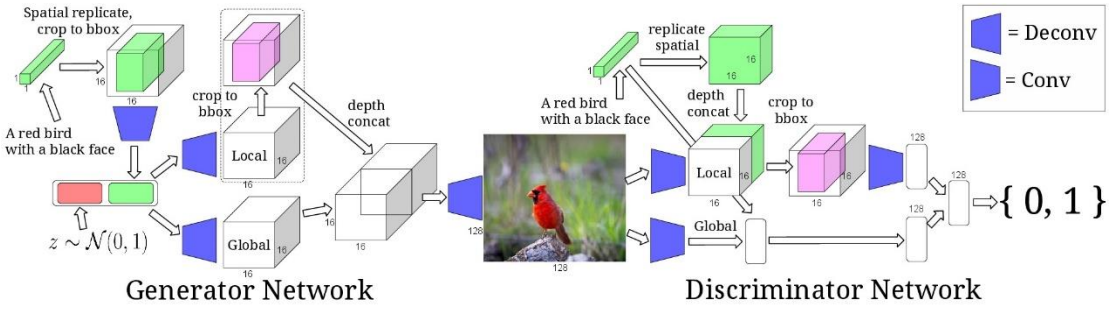


图：Spring MVC 的架构

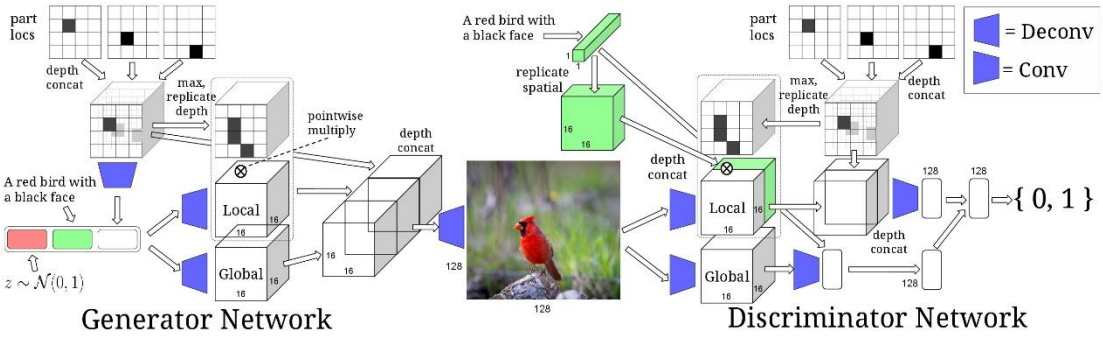
5.3 软件算法设计

nips2016（用来实现用英文描述生成图片功能）：

生成式对抗网络(GANs)最近展示了合成引人注目的真实世界图像的能力，例如房间内部、唱片封面、漫画、面孔、鸟类和花卉。虽然现有的模型可以基于全局约束(如类标签或标题)合成图像，但它们不能提供对姿态或对象位置的控制。该算法提出了一种新的模型，即生成式对抗性 what - where 网络(GAWWN)，它综合图像，给出描述在什么位置绘制什么内容的指令。该算法在加州理工学院和加州大学圣地亚哥分校的鸟类数据集中展示了高质量的 128×128 的图像合成，以非正式的文本描述和对象位置为条件。该算法的系统暴露了对鸟类及其组成部分的包围框的控制。通过对零件位置的条件分布进行建模，系统还可以对零件的任意子集(例如，只有鸟嘴和尾巴)进行条件设置，从而为挑选零件位置提供了一个有效的接口。在 MPII 人体姿态数据集上，我们还展示了在更具挑战性的文本和位置可控合成人类动作图像领域的初步结果。



图：算法示意图 1

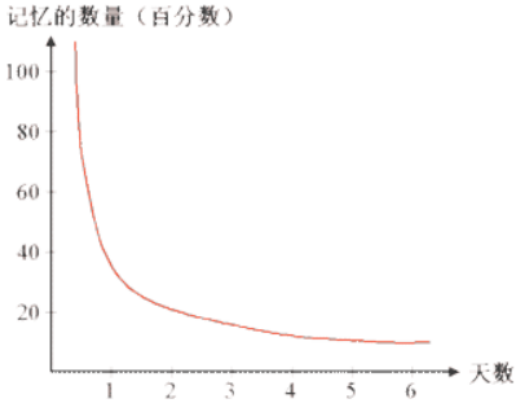


图：算法示意图 2

vid2speech（根据口型配音，实现趣味英语学习的娱乐功能）：
语音阅读是一项从视觉上观察到的发音面部动作来推断语音信息的任务，对于人类来说，这是一项非常困难的任务。此算法提出了一种基于卷积神经网络(CNN)的端到端模型，用于从说话人的无声视频帧中生成可理解的、听起来很自然的声学语音信号。其使用网格和 TCD-TIMIT 数据集对说话者进行训练，并使用常见的客观测量方法评估重构语音的质量和可理解度。结果表明，所提出的语音预测模型比现有的语音预测模型有明显的提高。此外，该算法还展示了从无约束字典重建语音的良好结果。

记忆曲线算法：

根据用户产生的数据，从而不断修正艾宾浩斯记忆曲线模型的参数，模拟出最符合用户的记忆曲线，并以此安排学习计划。在实际操作时设计好时间点，并定好时间单位，用以获得用户记忆数据，遗忘率以及背词次数等，进行周期性拟合，动态修正曲线，使之尽可能适合不同的用户，从而定制个人的记忆计划。



图：艾宾浩斯记忆曲线

口语评分与作文评分：
口语评分的主要实现逻辑，就是利用语音转文字算法（不考虑口音，标准英语的），将口语转成文字。再将生成的文字和原来的目标文字进行比对，最后得出口语评分。至于作文评分，由于自然语言处理技术的发展,自动分词、词性标注、句法分析等技术都取得了很大的进展，可以设计并实现了一个基于自然语言处理技术的英语作文自动评分系统。从语言质量和内容质量对英语作文进行评价。在语言质量中,分别从词汇质量和语句质量两个维度对作文进行评价。其中,词汇质量评价中,从词汇数、错误单词比例、平均词长、词汇丰富程度、词汇分布情况等方面对文章的词汇进行了评价；语句质量评价中,从标点符号个数、平均每个句子中的词汇数、短语数、语法错误数等方面对英语作文的语句质量进行评价。在语法检查中,本文采用了基于规则与统计相结合的方法。在内容质量评价方面,该算法采用了基于 TF-IDF 的方法进行了主题词抽取,这样,阅卷者可以根据抽取的主题词来判断该作文是否“跑题”。

6.管理过程

6.1 项目启动计划

每位组员既是积极的建言者，又是负责的合作者。决策应在充分的讨论基础上做出，并被及时有效的执行。按时按量完成项目的基本功能，按时推进产品的开发，遵循规范的项目运作标准，文档严谨完整，代码注释充分，便于后续维护。产品要运行稳定，界面友好易上手，能很好地实现 WordKeeper 的需求。开发软件过程中要注重团队建设，成员分工合理，合作默契，气氛融洽。 项目设计和开发商要有创新，有亮点可以吸引到人。

6.2 工作计划

参赛报名后的第一周：完成需求规格说明并撰写需求规格说明
参赛报名后的第二周：完成系统设计并撰写软件设计文档
参赛报名后的第三、四周：完成编码测试
参赛报名后的第五周：完成软件交付并撰写总结文档

6.3 控制计划

小组内各个成员以天为单位进行工作安排，并每天晚上 9：00 通过在线电话会议，汇报工程进度，根据赛制要求，及时更正进度安排。

6.4 风险管理计划

风险	标题	可能性	影响	优先级	规避或减轻策略
1	深度学习以及数据挖掘知识不熟练	80%	灾难性	高	提前指定学习计划，并适量放低设计难度
2	部分成员仍有其他项目在身	60%	严重的	中	合理组内任务与工作时间
3	需求较为不明确	50%	轻微的	中	花费足够的精力在项目的需求工程，以达到需求较为明确

风险的详细描述如下：
风险一：深度学习以及数据挖掘知识不熟练
组员在这方面的经验都不是很足，并且也没有系统上过相关的课程，所以对队伍来说是一个比较大的挑战。

风险二：部分成员仍有其他项目在身
其中队伍中有两名队员同时身处另一比赛的项目开发组中，所以如何合理在两个项目之间分配时间与精力，做到两个项目都不耽误也是一种挑战。
风险三：需求较为不明确
赛制并没有要求明确做何种功能的软件，所以整个项目的需求工程需要我们自己通过市场调研与资料收集来进行需求工程从而实现。

6.5 项目收尾计划

在开发阶段结束后，开发人员之间会进行代码走查，减少 bug，并在测试阶段更新源代码，测试人员根据测试文档进行软件测试，提高软件正确性。最终交付 WordKeeper（单词管家）软件给赛委会。

7.计划过程

7.1 过程模型

应用快速原型模型，用原型辅助软件开发的一种新思想。经过简单快速分析，快速实现一个原型，用户与开发者在试用原型过程中加强通信与反馈，通过反复评价和改进原型，减少误解，弥补漏洞，适应变化，最终提高软件质量。快速原型模型允许在需求分析阶段对软件的需求进行初步而非完全的分析和定义，快速设计开发出软件系统的原型，该原型向用户展示待开发软件的全部或部分功能和性能；用户对该原型进行测试评定，给出具体改进意见以丰富细化软件需求；开发人员据此对软件进行修改完善，直至用户满意认可之后，进行软件的完整实现及测试、维护。

7.2 方法、工具和技术

本小组的团队组织结构为主程序员式组织结构；编程语言为 java；采用面向对象的分析设计方法；深度学习等方面使用的语言为 python;利用 UML 进行系统建模；统一文件命名、代码版式、注释等编码规范；编码人员进行代码走查后再进行代 码编译；测试人员根据测试文档进行单元测试；最后实现软件的交付。

7.3 基础设施

个人 PC，笔记本、GitHub 云端仓库

8.支持过程

8.1 工作包

工作包	子工作包	预期完成时间	负责人	最终交付产物	简单描述说明
需求分析	初步描述	2020/1/26	周兴、林联辉、陈丰铧	需求规格说明	收集资料、市场调研、组内讨论
	规格说明	2020/1/28			
	进一步修改	2020/1/29			
	最终确定	2020/2/1			
系统设计	概要设计	2020/2/2	林联辉、周兴	软件设计文档	可根据需求规格说明的局部调整进行相应改变
	详细设计	2020/2/3			
	模型确定	2020/2/4			
编码测试	编码开发	2020/3/1	周兴、林联辉、陈丰铧	源代码	为了克服技术不熟的缺陷，建议在此之前加强相关知识的学习
	编码测试	2020/3/10			
	模型确定	2020/3/13			
软件交付	系统交付	2020/3/15	周兴、陈丰铧	总结文档	负责最后的收尾工作并撰写总结文档
	总结	2020/3/15			

8.2 依赖关系

1） 组织团队是完成软件项目的前提，明确分工负责；
2） 配置管理贯穿于整个软件开发和测试过程；
3） 需求分析是软件项目进入开发阶段的重要标志；
4） 系统设计是基于需求分析的基础上，又是编码的原理依据；
5） 编码测试是软件开发进展的重要过程；
6） 交付阶段是可以提交决赛材料的标志，是软件开发结束的标志；

8.3 资源需求

人员：小组软件项目开发成员
支持软件： Office 、PressOn 在线画图、Navicat、IDEA、Postman
计算机硬件：服务器等

办公室：家
工程设备：个人 PC 机、笔记本
项目资源维护需求的数目和类型： 3 台个人电脑（ Pentium III 800 以上 CPU，1G 以上内存）

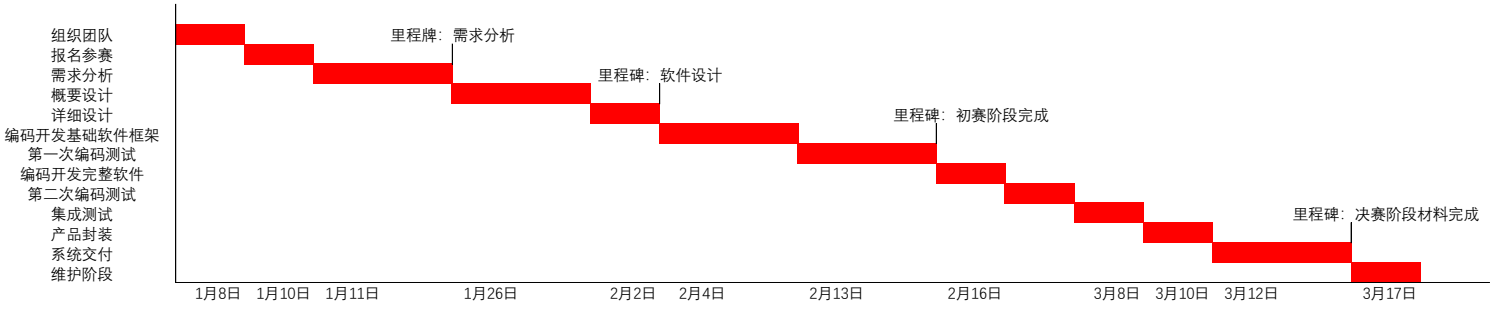
8.4 预算和资源分配

预算：本次软件开发没有涉及到任何经济方面的预算，仅花费小组成员假期时间。
资源分配：各自使用各自的个人 PC。

8.5 进度表

进度事件	时间（2020 年）
组织团队	1/8
报名参赛	1/10
需求分析	1/11
概要设计	1/26
详细设计	2/2
编码开发基础软件框架	2/4
第一次编码测试	2/13
编码开发完整软件	2/16
第二次编码测试	3/8
集成测试	3/10
产品封装	3/12
系统交付	3/15
维护阶段	3/17

以下为进度安排的甘特图：



图：甘特图