文件编号：[TEAMNAME]-SWC2018-[TEAMNUMBER]

受控状态：■受控 □非受控

保密级别：□公司级 □部门级 ■项目级 □普通级

采纳标准：CMMI DEV V1.2





文字图片匹配项目

**Phords**

项目开发文档

**Version [Number]**

[YYYY.MM.DD]

**Written by SBBStream**

[Team LOGO]

**All Rights Reserved**

目录

[1 引言 1](#_Toc526945116)

[1.1 编写目的 1](#_Toc526945117)

[1.2 项目概述 1](#_Toc526945118)

[1.3 项目背景 1](#_Toc526945119)

[1.4 术语和缩略语 1](#_Toc526945120)

[1.5 参考资料 1](#_Toc526945121)

[1.6 项目定位 2](#_Toc526945122)

[1.6.1 应用场景 2](#_Toc526945123)

[1.6.2 目标人群 2](#_Toc526945124)

[1.7 项目目标 2](#_Toc526945125)

[1.8 项目价值 2](#_Toc526945126)

[1.9 创新点 2](#_Toc526945127)

[1.9.1 功能性创新 2](#_Toc526945128)

[1.9.2 非功能性创新 2](#_Toc526945129)

[1.10 解决思路 2](#_Toc526945130)

[1.11 系统亮点 2](#_Toc526945131)

[1.11.1 功能性亮点 2](#_Toc526945132)

[1.11.2 非功能性亮点 2](#_Toc526945133)

[2 开发计划 2](#_Toc526945134)

[2.1 最终呈现形式 2](#_Toc526945135)

[2.2 主要功能描述 2](#_Toc526945136)

[2.3 运行环境 2](#_Toc526945137)

[2.4 验收标准 2](#_Toc526945138)

[2.5 关键问题 3](#_Toc526945139)

[2.6 进度安排 3](#_Toc526945140)

[2.7 开发预算 3](#_Toc526945141)

[3 可行性分析 3](#_Toc526945142)

[3.1 市场可行性分析 3](#_Toc526945143)

[3.1.1 与现行软件对比 3](#_Toc526945144)

[3.2 技术可行性分析 3](#_Toc526945145)

[3.2.1 功能简述 3](#_Toc526945146)

[3.2.2 技术要素 3](#_Toc526945147)

[3.3 资源可行性分析 3](#_Toc526945148)

[4 需求分析 3](#_Toc526945149)

[4.1 数据描述 3](#_Toc526945150)

[4.1.1 静态数据 3](#_Toc526945151)

[4.1.2 动态数据 3](#_Toc526945152)

[4.1.3 数据词典 3](#_Toc526945153)

[4.1.4 数据采集 3](#_Toc526945154)

[4.2 功能需求 3](#_Toc526945155)

[4.2.1 功能模块结构图 4](#_Toc526945156)

[4.2.2 核心功能模块描述 4](#_Toc526945157)

[4.3 非功能性需求 4](#_Toc526945158)

[4.3.1 扩展性和可维护性 4](#_Toc526945159)

[4.3.2 易用性 4](#_Toc526945160)

[4.4 性能需求 4](#_Toc526945161)

[4.4.1 时间特性 4](#_Toc526945162)

[4.4.2 适应性 4](#_Toc526945163)

[4.5 运行需求 4](#_Toc526945164)

[4.5.1 用户界面 4](#_Toc526945165)

[4.5.2 硬件接口 4](#_Toc526945166)

[4.5.3 软件接口 4](#_Toc526945167)

[4.5.4 故障处理 4](#_Toc526945168)

[4.6 用例图 4](#_Toc526945169)

[4.7 核心模块用例规约 5](#_Toc526945170)

[4.8 其他需求 6](#_Toc526945171)

[5 概要设计 6](#_Toc526945172)

[5.1 处理流程 6](#_Toc526945173)

[5.2 总体结构和模块设计 6](#_Toc526945174)

[5.3 功能分配 6](#_Toc526945175)

[5.4 接口设计 6](#_Toc526945176)

[5.4.1 外部接口 6](#_Toc526945177)

[5.4.2 内部接口 6](#_Toc526945178)

[5.4.3 用户界面设计 6](#_Toc526945179)

[5.5 数据结构设计 6](#_Toc526945180)

[5.5.1 逻辑结构设计 6](#_Toc526945181)

[5.5.2 物理结构设计 6](#_Toc526945182)

[5.5.3 数据结构与程序的关系 6](#_Toc526945183)

[5.6 运行设计 6](#_Toc526945184)

[5.6.1 运行模块关系 6](#_Toc526945185)

[5.6.2 运行控制 6](#_Toc526945186)

[5.6.3 运行时间 6](#_Toc526945187)

[5.7 错误/异常处理设计 6](#_Toc526945188)

[5.7.1 错误/异常输出信息 7](#_Toc526945189)

[5.7.2 错误/异常处理对策 7](#_Toc526945190)

[5.8 系统配置策略 7](#_Toc526945191)

[5.9 系统部署方案 7](#_Toc526945192)

[5.10 代码规范 7](#_Toc526945193)

[5.11 其他相关技术与方案 7](#_Toc526945194)

[6 数据库设计 7](#_Toc526945195)

[7 详细设计 7](#_Toc526945196)

[7.1 \*\*功能模块 7](#_Toc526945197)

[7.1.1 功能描述 7](#_Toc526945198)

[7.1.2 性能描述 7](#_Toc526945199)

[7.1.3 输入 7](#_Toc526945200)

[7.1.4 输出 7](#_Toc526945201)

[7.1.5 算法 7](#_Toc526945202)

[7.1.6 程序逻辑 7](#_Toc526945203)

[7.1.7 接口 7](#_Toc526945204)

[7.1.8 存储分配 7](#_Toc526945205)

[7.1.9 限制条件 7](#_Toc526945206)

记录更改历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **更改原因** | **版本** | **作者** | **更改日期** | **备 注** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 引言

## 编写目的

该项目开发文档的编写目的是全面深入地介绍并分析本项目的建设情况。该文档将从项目背景到项目搭建再到重点实现进行介绍，以便于对各个模块的实现方案进行详细分析和探讨。我们通过编写开发文档，让您了解我们整个项目架构的设计、以深度学习为核心的算法设计、面向用户的界面设计以及用户第一的人性化人机交互设计等设计理念。这些方案支撑整个项目的开发和高效实现。该项目开发文档重点介绍了项目的背景、概述、需求分析、概要设计、可行化分析、数据库设计及其他详细设计，采用图片、文字、表格等方式前面展示项目设计理念，突出我们项目的可行性、实用性、创新性。

## 项目概述

本项目的基本功能，就是给用户提供一个图片选择平台，用户可以在这个平台上输入一段文字，然后获得一些与文字相关的图片，用户可以浏览这些图片并选择下载。经过预测，该项目将会应用于各个领域：需要做讲义的老师、即时聊天急需表情包的用户、写文档的博主、网页设计师等等，都会有相关的需求，该项目有很高的市场前景。该项目运用tensorflow.js前端深度学习技术，以及前后端web开发等成熟的技术，可以很好解决项目开发的技术问题。

## 项目背景

现在我们正处于一个网络普及、计算机技术发展极其迅速的时代，信息一直在无时不刻更新着，人们不断地上传、分享、刷新、浏览着各种各样的信息：网页新闻、网站信息、在线文档、演示文稿、即时聊天等等。这些都显然离不开文字。而有时候，即时聊天的一张表情图片、演示文稿一个页面的插图、文档关键字或术语解释的一张图片以及新闻资讯的一张现场图片，都胜过千言万语，让人一目了然。所以，为适应信息的迅速更新，及时传递准确的、易于理解的信息以及相关图片，就尤为关键。

如果在写文字或读文字的同时，再分析文字的内容，然后再通过关键字或者自己的理解，从网上搜索图片，不仅费时费力，而且准确性也有待考察。虽然现在也有一部分软件功能也有类似的功能，不过大多都是靠词语或者单个字匹配图片，且准确率并不令人满意。所以，为一段文字迅速匹配合适的高质量的图片，包括表情包、插图等，是有必要的。

而机器学习的迅速发展，以python、tensorflow等为代表的深度学习、神经网络等技术日益成熟、流行，则为文字关键字识别以及图片匹配提供了足够的、必要的技术支持，作为web端程序，已成熟的数据库技术和前后端框架技术，例如spring+mybatis，也保证了项目的开发，使其具有一定的可行性。在此需求以及背景下，我们准备研发该项目。

## 术语和缩略语

## 参考资料

## 项目定位

### 应用场景

1. 新闻平台：即时更新新闻文章需要匹配图片
2. 撰写文档：相关名词、术语需要匹配图片进行解释

例如：博主想要写教学博客，相关知识、公司的图片就可以匹配并使用

1. 演示文档：ppt页面需要匹配相关图片

例如：教师编写教学课件，相关知识、方法、数据的图片就可以匹配使用

1. 网页更新：根据网页内容变化匹配相关图片
2. 即时聊天：为用户聊天时的名词、情感等方面提供表情包图片
3. 总结：需要快速为文字匹配图片的其他大多数场景都可适用，该项目都可以为这些场景快速地、准确地提供一些相关图片可供用户选择。而且不仅仅适用于单个名词、情感等词汇匹配，还可以根据一段文字各个部分综合分析考虑匹配更加准确的图片。

### 目标人群

撰写文章、文档的博主，需要制作讲义的教师，以及商演的人士，或者日常网上聊天的用户都可以使用这个项目，及时、快速地为文字匹配相关的图片。在迅速、及时的同时我们会尽量保持准确、美观，一些设计人员或专业人士，以及并不急需图片的人士也可以借鉴该平台。

## 项目目标

旨在通过深度学习、神经网络算法以及web相关技术，为用户提供实现图片匹配文字的一个web平台。该平台将通过实用性、创新性设计来保证图片的迅速、多元以及准确。该平台还保证设计的人性化，为用户提供注册、登录等基本功能的同时，还有密码修改、手机邮箱认证、历史浏览、下载保存等功能，来提升用户体验，并且后台可以定时更新图片，让该项目能用、好用。

## 项目价值

(1) 应用广泛，适用于撰写文章或文档的博主、需要制作讲义的教师、商演的人士、日常网上聊天的用户等在短时间内需要匹配图片的人士；适用于需要快速为文字匹配图片的大多数场景都可适用。

(2) 可以提高生产效率：大大提高撰写文档，更新新闻、网页，编写课件，在线聊天等活动效率。

(3) 成本较低：基于web开发的项目，基于tensorflow.js的前端深度学习技术，mysql数据库技术，降低了项目开发的难度，且易于维护，开发、维护成本较低。且用户只需申请账号、输入文字就可以找到相关图片，使使用成本也控制在一个比较低的范围内。

(4) 有一定市场价值：设置部分功能处于体验、开发状态，部分额外功能会员可以体验，加上其比较大的市场前景，会有一定的市场效益。

## 创新点

### 功能性创新

将多个服务互联成系统，功能新颖全面，可在实际场景中投入使用，并吸引一定量的用户。功能包括文本关键字识别、情绪识别、智能图片匹配等等，有着独特的人文关怀。现代人使用电子产品编辑文本的场景越来越常见，针对这部分人群，我们的产品具有极强的实用性，因而能够吸引大量用户。

### 非功能性创新

软件效率高，匹配时间比较快，延迟小于2s；软件稳定可靠，因为有多个角度解析文本并配图；软件简单易上手，只需要把文字粘贴到指定位置并点击匹配按钮即可完成图片匹配；软件可维护性极强，因为软件每个模块进行过单独专业测试；软件的可移植性很强，可放在任意服务器上提供服务。

## 解决思路

## 系统亮点

### 功能性亮点

### 非功能性亮点

# 开发计划

## 最终呈现形式

基于前后端分离技术，前端采用vue框架 + iview组件库 + nginx反向代理服务器、后端采用spring框架 + mysql数据库 + mybatis + tomcat代理服务器的web开发应用

## 主要功能描述

## 运行环境

主流浏览器：如下图所示的浏览器以及操作系统的兼容性



考虑到web应用，综合来看运行环境为除IE等浏览器以外的主流浏览器

## 验收标准

随机输入五端有感情的文字，配图准确率到四张以上

## 关键问题

(1) 浏览器版本是否支持:给用户推荐一个能支持的浏览器；

(2) 深度学习判断泛化正确率是否较高：增大训练数据集大小以及模型精度；

(3) 图片库来源：爬虫程序爬取

## 进度安排

## 开发预算

# 可行性分析

## 市场可行性分析

### 与现行软件对比

## 技术可行性分析

### 功能简述

### 技术要素

## 资源可行性分析

# 需求分析

## 数据描述

### 静态数据

### 动态数据

### 数据词典

### 数据采集

## 功能需求

### 功能模块结构图

### 核心功能模块描述

表 核心功能模块描述

|  |  |  |  |
| --- | --- | --- | --- |
| **功能模块** | **功能** | **功能描述** | **优先级** |
|  |  |  |  |
|  |  |  |
|  |  |  |

## 非功能性需求

### 扩展性和可维护性

### 易用性

## 性能需求

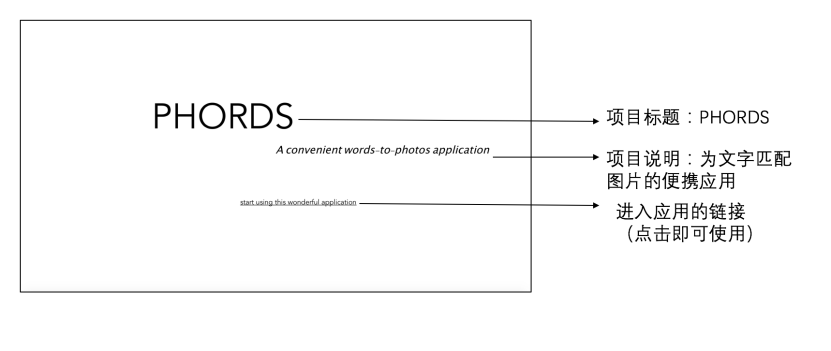
### 时间特性

### 适应性

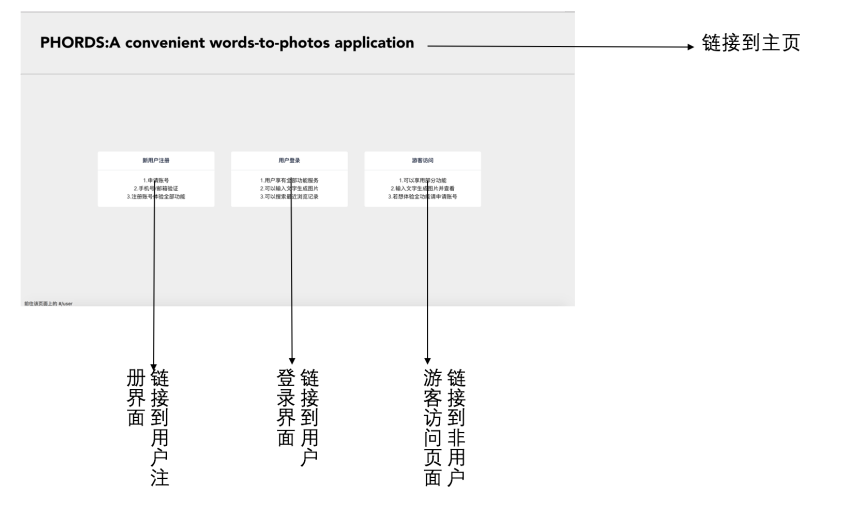
## 运行需求

### 用户界面

(1) 主页：



(2) 用户选择界面：



(3) 游客访问页面：



[1]返回主页的链接 [2]输入框：用户输入一段文字 [3]按钮：用户输入完成后点击按钮实现功能 [4]与文字匹配的图片列表生成 [5]使用教程说明

(4) 用户注册界面



[1]返回用户选择界面的链接 [2]输入框：用户输入昵称

[3]输入框：用户输入邮箱 [4]输入框：用户输入密码

[5]输入框：用户确认密码 [6]按钮：用户点击完成注册

1. 用户登录页面



[1]返回用户选择界面的链接 [2]输入框：用户输入注册邮箱

[3]输入框：用户输入密码 [4]按钮：用户点击登录，若登录成功则跳转到用户功能界面

(6)用户功能页面



[1] 返回主页页面 [2] 下拉列表：可以选择浏览历史或注销账号

[3] 历史浏览：点击查看自己最近浏览的图片以及输入的文字

[4] 注销账号：登出账号（退出登录状态）

[5] 综合设置：可以对用户信息进行设置 [6] 图片缓存：查看或删除自己下载的图片

[7] 输入框：用户输入一段文字 [8] 按钮：用户输入完成后点击按钮实现功能

[9] 与文字匹配的图片列表生成

### 硬件接口

### 部署在基于 Intel CPU 的设备（例如使用 Intel CPU 的laptop）上的在浏览器中能够流畅运行的 Web 应用程序

### 软件接口

(1) 用户注册接口：前端页面，用户输入昵称、邮箱、密码(按照格式)、确认密码，通过对昵称、邮箱、密码输入格式的正确性判断(昵称不为空、没有非法符号、密码由大小写字母及数字组成、邮箱格式正确），然后将正确格式的邮箱进行判断其存在性，若邮箱地址正确且输入格式正确，就将密码加密并和其他用户信息存入数据库。

(2) 用户登录接口：前端页面，用户输入邮箱、密码，先判断其输入合法性，若合法就传入后端对比数据库信息，若一致就返回登录成功。

(3) 用户/游客功能接口：前端页面输入一段文字，利用tensorflow.js匹配图片，并将图片呈现在前端页面供用户浏览，数据库并储存此时该用户输入的文字以及查看的图片

(4) 用户历史浏览接口：点击按钮，后端数据库查找该用户的浏览历史，并将信息返回给前端页面进行显示

(5) 用户账号注销接口：点击按钮退出账户

(6) 用户下载图片接口：点击下载按钮，通过接口后台将点击的图片储存到数据库中

(7) 用户密码修改接口：通过邮箱验证的方式，用户在邮箱收到邮件，来修改密码、重复密码，若格式符合要求，就显示修改成功并将新的加密密码存到数据库。

### 故障处理

(1) 后端采用异常处理机制，会对不同的异常问题进行分析、分类，并将问题返回前台告知用户。

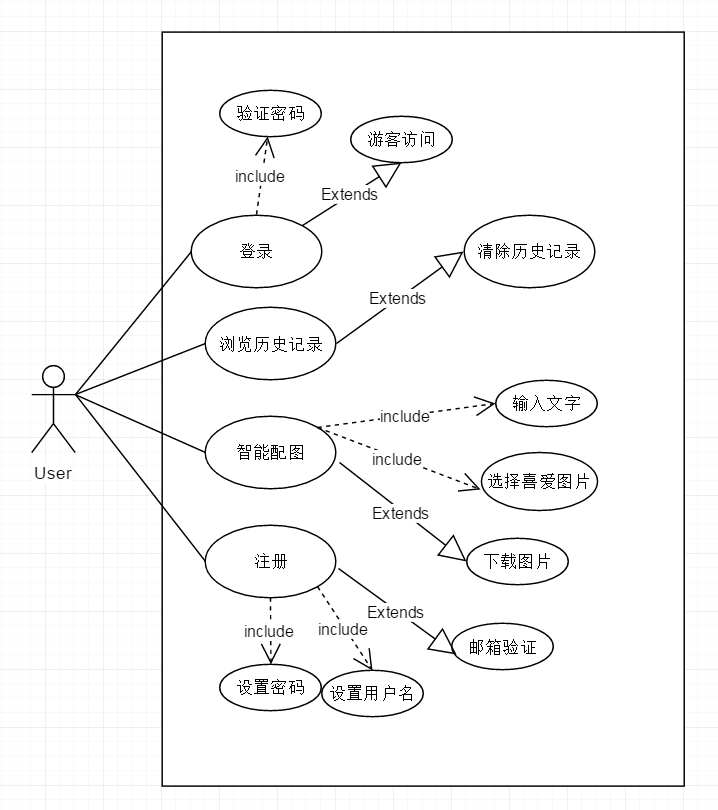
(2) 若是网络问题，则提示用户重新检查网络。

(3) 前端故障，大多是输入的问题，可以将问题储存，交给管理员及时处理。

(4) 用户可以反映问题，管理员收集问题并及时解决。

(5) 管理员定期管理数据库，清理垃圾、冗余数据

## 用例图



## 核心模块用例规约

表 \*\*\*\*用例规约

|  |  |
| --- | --- |
| **用例名称** |  |
| **功能简述** |  |
| **用例编号** |  |
| **执行者** |  |
| **前置条件** |  |
| **后置条件** |  |
| **涉众利益** |  |
| **基本路径** |  |
| **扩展路径** |  |
| **字段列表** |  |
| **设计规则** |  |
| **未解决的问题** |  |
| **备注** |  |

## 其他需求

1. 安全性：密码加密、输入非法字符时的处理、数据库安全、并行操作以及网络安全问题，是需要考虑并解决的

(2) 可维护性：定期进行数据库清理、维护；及时收集用户的反映，进行多次全面的、系统的修复和测试；编码时加上注释，变量符合命名规范，格式简单、清晰，采用模块化、结构化设计，有助于维护

(3) 可移植性：具有良好的可移植性，可以随着处理器的更新正常使用，当条件有变化时，程序无需作很多修改就可运行。

# 概要设计

## 处理流程

## 总体结构和模块设计

## 功能分配

## 接口设计

### 外部接口

### 内部接口

### 用户界面设计

## 数据结构设计

### 逻辑结构设计

### 物理结构设计

### 数据结构与程序的关系

## 运行设计

### 运行模块关系

### 运行控制

### 运行时间

## 错误/异常处理设计

### 错误/异常输出信息

### 错误/异常处理对策

## 系统配置策略

## 系统部署方案

## 代码规范

## 其他相关技术与方案

# 数据库设计

# 详细设计

## \*\*功能模块

### 功能描述

### 性能描述

### 输入

### 输出

### 算法

### 程序逻辑

### 接口

### 存储分配

### 限制条件