# Poetry Tracing System

# Detailed Design

# Project name: Poetry Tracing System

Team members: Tu Xiaozhao 201830310209

Che Jingang 201830310245

Wang Ziyue 201830310208

Yu Li 201830310210

March 19, 2019

# 1. Introduction

## 1.1 Document purposes

In order to ensure that the project team completes the project objectives on time with good quality, easy to project team members to better understand the project situation, enables each project work to carry out a reasonable and orderly process, it is necessary, in the form of a documented in the project life cycle of the scope of work tasks, each work task decomposition, project team organization structure, between the inner and outer communication team members work responsibilities, team collaboration mode, progress, budget, project content such as internal and external environment condition, risk countermeasure, in the form of a written description, as members of the project team and project stakeholders between consensus and agreement, The action basis for all project activities during the project life cycle, and the basis for the project team to carry out and check the project work.

The project development plan is used to guide the project of intelligent poetry tracing system to proceed smoothly and finally get the project products that pass the review.

## 1.2 Document scope

The document generally described the beginning and end of the project, including the role of the project plan, software project overview, requirements analysis, software scope and overall system design, and detailed the software requirements specification, SOW, WBS, gantt chart, milestone and delivery.

## 1.3 Reader object

This document is mainly for developers, software testers, project managers, maintenance personnel and other relevant personnel to provide access.

## 1.4 Explanation and abbreviations

Technical terms:

(1) Django：It is an open source web application framework written in Python. MVC framework pattern is adopted, namely model M, view V and controller C.

1. Tesseract-OCR：It is an open source OCR (Optical Character Recognition) engine developed by HP laboratory and maintained by Google

Abbreviations:

1. Tu: Tu xiaozhao (201830310209)

Yu: Yu li (201830310210)

King: Che jingang (201830310245)

Wang:Wang ziyue (201830310208)

1. SQL：Structured Query Language

# 2. Overview

## 2.1 System introduction

With the increasing power of data, people are more and more enthusiastic in the pursuit of technology. However, they gradually forget the traditional Chinese culture and the foundation of the Chinese nation -- Chinese language, especially ancient poetry. However, ancient poetry is very large and has different sentence patterns. Learning only from paper books is extremely inefficient and limited in ability, which requires an intelligent and efficient system with a huge amount of data to retrieve ancient poetry. In addition, there will be some scattered and incomplete poems and sentences, and even some rare words, which causes great difficulties in the search of poems and essays. The intelligent ancient poetry source tracing system can find out the source of this sentence and a series of information based on the pictures of poems uploaded by users, which greatly reduces the threshold for learning and popularizing ancient poetry and improves the learning efficiency.

The intelligent poetry tracing system adopts advanced image and text recognition technology and crawler technology. Pycharm is used as a development tool, and the background database management system is MySql community version. Finally, the website is used as a carrier for users to use.

## 2.2 System environment

### 2.2.1 development environment

|  |  |  |
| --- | --- | --- |
| **category** | **name** | **version** |
| OS | Windows | Windows 10 |
| Database | MySQL | Community Server version |
| Develop language | Python | 3.4 |
| Project management too | GIT | / |
| Develop tool | PyCharm | 2018.1.2 |

### 2.2.2 Runtime environment

|  |  |  |
| --- | --- | --- |
| **category** | **name** | **version** |
| OS | Windows/Mac | / |
| Platform | browser | / |

# 3.Demand analysis

## 3.1 Time requirements

The project of this intelligent poetry tracing system lasts for 10 weeks. The first stage is the first 5 weeks (March 12, solstice, April 9), and the second stage is the last 5 weeks (April 9, solstice, May 14). The specific schedule is as follows:

March 5~March 12: complete the overall design (project plan)/personnel arrangement of the project

March 12 ~March 19: complete detailed system design and demonstration/technical study/system development

March 19 ~ March 26: system development phase

March 26 ~ April 2: system development phase

April 2 ~ April 9: the modules are basically completed, the modules are connected and tested, and the basic functions are realized

April 9 ~ April 16: deliver the initial system/start the second iteration, make improvement plan/system development

April 16 ~ April 23: system development phase

April 23 ~ April 30: the second iteration task is basically completed/system test

April 30 - May 7: delivery of the final version

## 3.2 user demand

The user provides the system with clear pictures or screen shots of the poem. The system displays a series of information about the poem on the page for the user to browse or download. The overall response time shall not exceed 5 seconds, and the whole network query function shall be provided when the response time exceeds 5 seconds, so as to improve the user experience.

## 3.3 functional requirement

First iteration: save the image uploaded by the user

Identify the poem/words in the picture

Query the database by poem/phrase

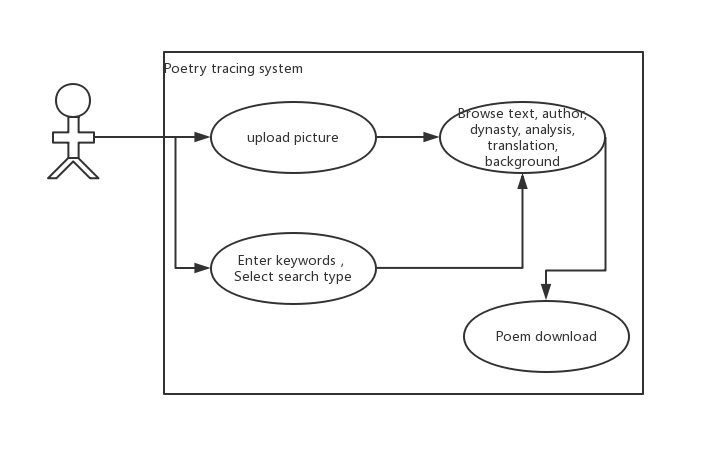
Display the query results on the web page

Second iteration: realize the function of poetry retrieval (users can query according to author/dynasty/category)

Provide poetry information download function

Website sharing function

## 3.4 use case diagram

****

## 3.5 Constraints and assumptions

### 3.5.1 Constraints

Time and budget constraints: the first phase of the project will end on April 9, and the main functions of the system (poetry tracing) will need to be completed, which will take 4 weeks. If the delivery is over time, the course performance will be affected. No budget, staff reward self-funding.

Technical constraints: the project needs to use image recognition technology, crawler technology and python web front and rear end of a number of technologies, basically starting from scratch, the technology is not mature, but the network is rich in educational resources, technical feasibility.

Personnel constraints: there are 4 people in the project team, with graduate degree, little practical experience and limited skills. The temporary task is divided into four modules -- image recognition module, crawler module, front-end module and website architecture module. Each of them is responsible for one module and carries out cooperation.

Organizational constraints: none.

### 3.5.2 assumptions

Assume that the actual working time per day is one hour and efficiency is normal.

Assume that the project developer does not leave the team and follows the planned activities.

Assume that the crawler successfully crawls to obtain relatively complete poetry data.

Assume that client requirements do not change.

# 4. Architecture design

## 4.1 Design concept and process flow

First of all, users need to upload a picture of a contains a poem, type can be a JPG, jpeg, and PNG, etc.After receiving the image, the system preprocesses the image (image size conversion, channeling, etc.), recognizes the poem and converts it into a string.Then the system searches the pre-built database and returns a complete set of information (poem name, verse, author, dynasty, background, translation and appreciation) of the corresponding poem.

Users can also input keywords and select the type (poem name, verse, author) by themselves. The system will search the corresponding information and return a complete set of information (poem name, verse, author, dynasty, background, translation, appreciation) of the corresponding poem.

## 4.2 Software modules

### 4.2.1 Module summary

* Data module: collect data, build database and provide query interface.

Involving technology：crawler

Workload：14 hours(days)

Director:Tu

Inspector:Yu

* Verse recognition module: convert the image into text after processing, and provide conversion interface.

Involving technology：Tesseract\_OCR

Workload：14 hours(days)

Director:Yu

Inspector:King

* Website architecture module: Django architecture website is used to call interfaces.

Involving technology：Django

Workload：21+7 hours(days)

Director:King

Inspector:Tu

* Website front-end module: design and implement the website front-end, provide data reception and display interface.

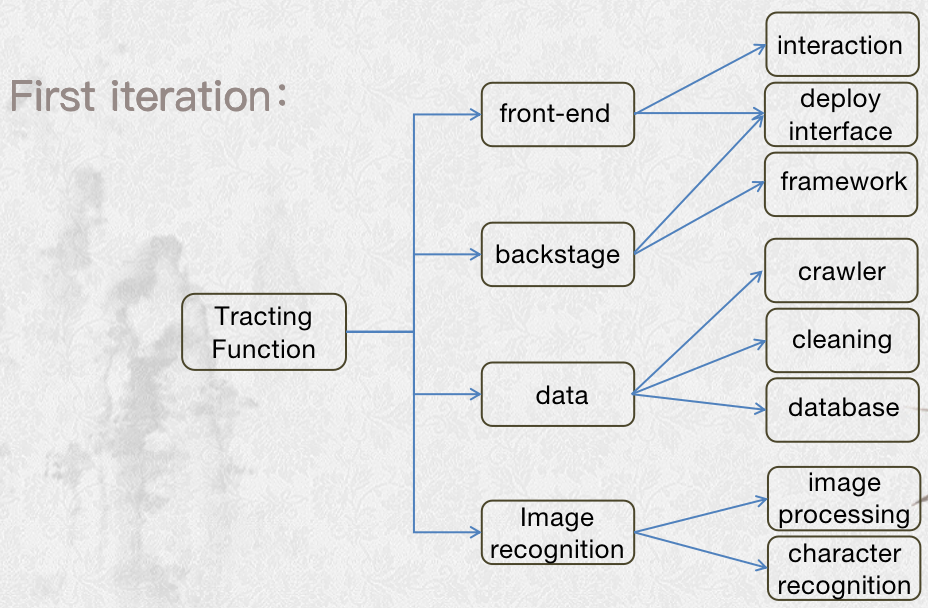
Workload：14+7hours(days)

Director:Wang

Inspector:Yu

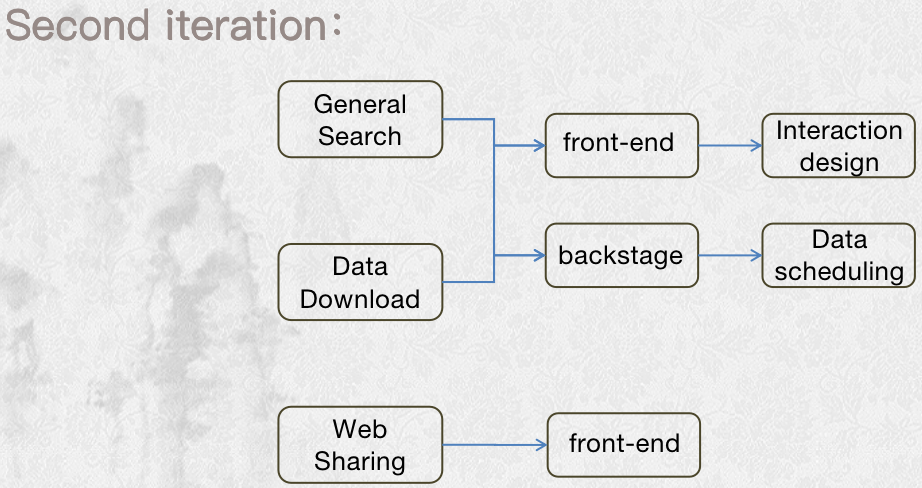
### 4.2.2 module design

The first round of iteration: users upload pictures, and identify the poems and texts in the pictures through the picture recognition function of the system. The system sends the poems and texts to the background, and the background returns the complete information of poems and texts to the users.



The first iteration of the system use case diagram

The second iteration: it is roughly divided into three modules -- poetry and prose retrieval module (users can search according to author/dynasty/category), poetry and prose information download module and website sharing module.

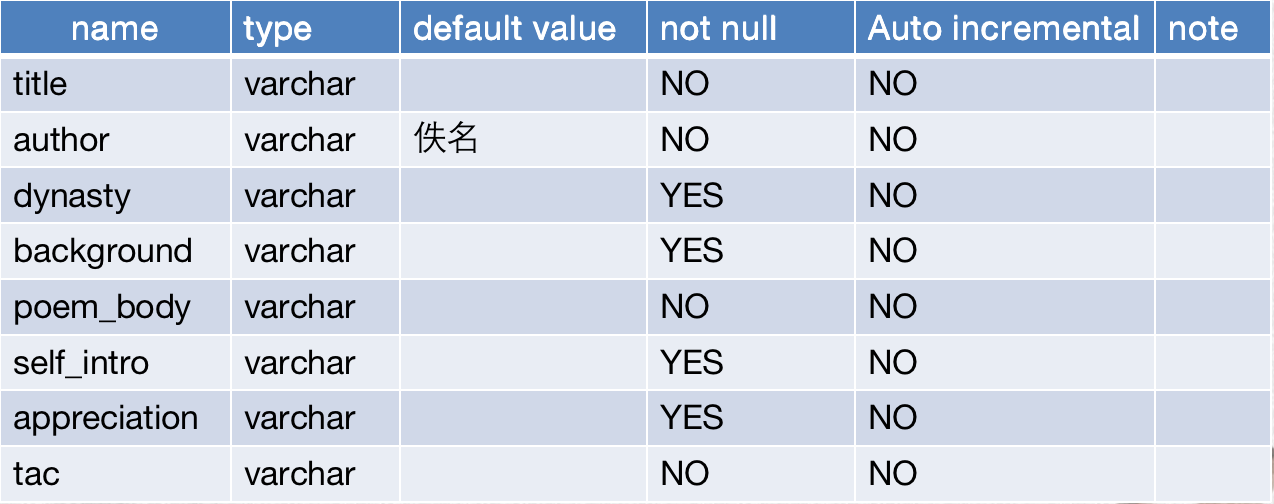


The second iteration of the system use case diagram

### 4.2.3 Module technical requirements

|  |  |
| --- | --- |
| Name | Requirements |
| front-end interactive | Html、css、jsp |
| framework | Django |
| crawler | / |
| cleaning | / |
| database | Mysql |
| image processing | / |
| character recognition | Tesseract\_OCR |

## 4.3 Database design



## 4.4 front-end layout

Design sketches for the first iteration



Design sketches for the second iteration：

## 屏幕快照 2019-03-19 下午10.21.57

# 5. Data structure design

## 5.1 System organization design

The system is composed of image recognition module and poetry query module, but these two modules are unified regulation through the main system. The poetry recognized by the image recognition module is returned to the customer's poetry source and all contents of poetry through the response of poetry query module.

## 5.2 interface design

First iteration：

backstage

front-end

f

a

e

d

b

c

data base

Image recognition

**a** my\_upload(address) ——output picture address

**b、c** feedbackWord(address) ——input image location

return verse ——output verse

**d、e** select\_for\_args(sql) ——input database operation command

return list ——output data

**f** get\_mess\_title(title\_name) ——output title

get\_mess\_author(author\_name) ——output author

get\_mess\_dynasty(dynasty\_name) ——output dynasty

get\_mess\_poem(poem) ——output poem\_body

get\_mess\_tac(tac) ——output tac

get\_mess\_background(background) ——output background

get\_mess\_self\_intro(self\_intro) ——output self\_intro

get\_mess\_appreciation(appreciation) ——output appreciation

Second iteration：

front-end

j

g

backstage

h

i

Image recognition

data base

**g** my\_search(input,checkbox) ——input search\_type and text

**h、i** select\_for\_args(sql) ——input database operation command

return list ——output data

**j**  get\_mess\_title(title\_name) ——output title

get\_mess\_author(author\_name) ——output author

get\_mess\_dynasty(dynasty\_name) ——output dynasty

get\_mess\_poem(poem) ——output poem\_body

get\_mess\_tac(tac) ——output tac

get\_mess\_background(background) ——output background

get\_mess\_self\_intro(self\_intro) ——output self\_intro

get\_mess\_appreciation(appreciation) ——output appreciation

## 5.3 interface introduction

|  |  |
| --- | --- |
| Interface name | my\_upload(address) |
| function | pull the image user submit |
| input | / |
| output | the address of the image |
| Inspection standard | be able to use it to find the right image |
| Time | 2 |
| Director | King |
| Inspector | Yu |

|  |  |
| --- | --- |
| Interface name | feedbackWord(address) |
| function | identify text in image |
| input | Address of the image |
| output | verse in the image |
| Inspection standard | Accuracy > 0.7 |
| Time | 7 |
| Director | Yu |
| Inspector | Tu |

|  |  |
| --- | --- |
| Interface name | select\_for\_args(sql) |
| function | select data in the database |
| input | a database operation command |
| output | a list of data |
| Inspection standard | data should be accurate and complete |
| Time | 1 |
| Director | Tu |
| Inspector | King |

|  |  |
| --- | --- |
| Interface name | get\_mess\_\*(\*) |
| function | organize the data acquired in the background onto the page |
| input | Title,author,dynasty,poem,tac,background,appreciatio |
| output | / |
| Inspection standard | Beautiful and complete |
| Time | 7 |
| Director | Wang |
| Inspector | Yu |

|  |  |
| --- | --- |
| Interface name | my\_search(input,checkbox) |
| function | provide the keywords and types entered by the user to the background |
| input | Input，checkbox |
| output | / |
| Inspection standard | accurate and complete |
| Time | 2 |
| Director | Wang |
| Inspector | King |

# 6. System error handling design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Error description** | **Error module** | **Module leader** | **Reason** | **Solution or remedy** |
| Upload image search no output | Image recognition | Yu | Abnormal character recognition function or low recognition rate | When the query fails, the user is prompted to upload again on the page |
| data base | Tu | The website database is not rich enough | The feedback mechanism is triggered when several attempts fail and data is added manually |
| Upload pictures and search results are inconsistent | Image recognition | Yu | Low rate of character recognition | Strengthen the input of image processing to improve the recognition accuracy |
| Page display error | backstage | King | Web site or server exception | Set up a feedback mechanism and deal with it as soon as possible |
| Multi-times search has no feedback | data base | Tu | Database exception | Data heavy guide |
| backstage | King | Database service not open or connection abnormal | Database reconnection |

# 7. System maintenance design

* Database maintenance: when there is insufficient poetry database, manually add dynamic, rich database.
* Website architecture maintenance: when there are problems such as website crash maintenance.
* Maintenance time: week 13 (before the end of the software project management course) to provide maintenance services.