# Intelligent Poetry Tracing System Detailed Plan

**directory**

**[1. Introduction](#_Toc8569_WPSOffice_Level1)** **[2](#_Toc8569_WPSOffice_Level1)**

[1.1](#_Toc28483_WPSOffice_Level2) [Documentation purposes](#_Toc28483_WPSOffice_Level2) [2](#_Toc28483_WPSOffice_Level2)

[1.2 he document scope](#_Toc15_WPSOffice_Level2) [2](#_Toc15_WPSOffice_Level2)

**[2.](#_Toc28483_WPSOffice_Level1)** **[project overview](#_Toc28483_WPSOffice_Level1)** **[2](#_Toc28483_WPSOffice_Level1)**

[2.1 project brief](#_Toc12050_WPSOffice_Level2) [2](#_Toc12050_WPSOffice_Level2)

[2.2 project context](#_Toc261_WPSOffice_Level2) [3](#_Toc261_WPSOffice_Level2)

[2.3 Project objective function](#_Toc9386_WPSOffice_Level2) [3](#_Toc9386_WPSOffice_Level2)

[2.4 operating environment](#_Toc2542_WPSOffice_Level2) [3](#_Toc2542_WPSOffice_Level2)

[2.5 limiting condition](#_Toc21372_WPSOffice_Level2) [3](#_Toc21372_WPSOffice_Level2)

**[3. demand analysis](#_Toc15_WPSOffice_Level1)** **[3](#_Toc15_WPSOffice_Level1)**

[3.1 business requirements](#_Toc3313_WPSOffice_Level2) [3](#_Toc3313_WPSOffice_Level2)

[3.2 user demand](#_Toc22437_WPSOffice_Level2) [4](#_Toc22437_WPSOffice_Level2)

[3.3 functional requirement](#_Toc9733_WPSOffice_Level2) [4](#_Toc9733_WPSOffice_Level2)

[3.4 WBS and the gantt chart](#_Toc13407_WPSOffice_Level2) [4](#_Toc13407_WPSOffice_Level2)

[3.5 work cycle（milestone）](#_Toc19985_WPSOffice_Level2) [4](#_Toc19985_WPSOffice_Level2)

**[4. Scope of software](#_Toc12050_WPSOffice_Level1)** **[4](#_Toc12050_WPSOffice_Level1)**

[4.1](#_Toc24362_WPSOffice_Level2) [project purpose](#_Toc24362_WPSOffice_Level2) [4](#_Toc24362_WPSOffice_Level2)

[4.2 product description](#_Toc19459_WPSOffice_Level2) [5](#_Toc19459_WPSOffice_Level2)

[4.3 deliverable](#_Toc13567_WPSOffice_Level2) [5](#_Toc13567_WPSOffice_Level2)

[4.4 constraint](#_Toc31655_WPSOffice_Level2) [5](#_Toc31655_WPSOffice_Level2)

[4.5 assumed condition](#_Toc12870_WPSOffice_Level2) [5](#_Toc12870_WPSOffice_Level2)

[4.6 Project acceptance criteria](#_Toc80_WPSOffice_Level2) [5](#_Toc80_WPSOffice_Level2)

**[5. detailed design of system](#_Toc261_WPSOffice_Level1)** **[6](#_Toc261_WPSOffice_Level1)**

[5.1 Task Description](#_Toc19314_WPSOffice_Level2) [6](#_Toc19314_WPSOffice_Level2)

[5.2 summary on design](#_Toc27009_WPSOffice_Level2) [6](#_Toc27009_WPSOffice_Level2)

[5.3 System logic design](#_Toc29538_WPSOffice_Level2) [7](#_Toc29538_WPSOffice_Level2)

[5.4 module design](#_Toc16963_WPSOffice_Level2) [7](#_Toc16963_WPSOffice_Level2)

**[appendix](#_Toc9386_WPSOffice_Level1)** **[8](#_Toc9386_WPSOffice_Level1)**

## **Introduction**

## 1.1 Documentation 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 The 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.

# 2.project overview

## 2.1 project brief

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 project context

The intelligent poetry source tracing system comes into being as required by the software project management course. The main function of the system is to find the original text of the poem and picture uploaded by users through text recognition, and present the author, background, translation and appreciation, etc. The project cycle is 10 weeks.

## 2.3 Project objective function

The first iteration cycle: according to the pictures of a certain poem provided by the user, the corresponding whole poem and relevant data are given.

Second iteration cycle: expected to complete the function of retrieval according to author, poem name and dynasty.

## 2.4 operating environment

The overall offline development environment of the project is based on the window operating system; The development language used is python3, and the development tool is pycharm; MySQL Community Server version; Project management tool github.

## 2.5 limiting condition

1) certain requirements for photos uploaded by users:

The picture should contain at least one complete line of poetry.

Photos need to be sharp enough, or the words will be garbled or unrecognizable.

You need to shoot from the front of the poem to avoid the different shapes of the text caused by the Angle.

Do not include too much extraneous text in an image to avoid redundancy and overhead.

2) the running time of the system itself is limited. When the corresponding poem cannot be found in the background of the system, the system will provide the full-web search function to the user by default after 5 seconds, so as to improve the user experience.

3) system database limitation. The system database from the crawler technology collection of poetry, limited content, covering the vast majority of ancient poetry, but do not exclude omission.

# 3.Demand analysis

## 3.1 business 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

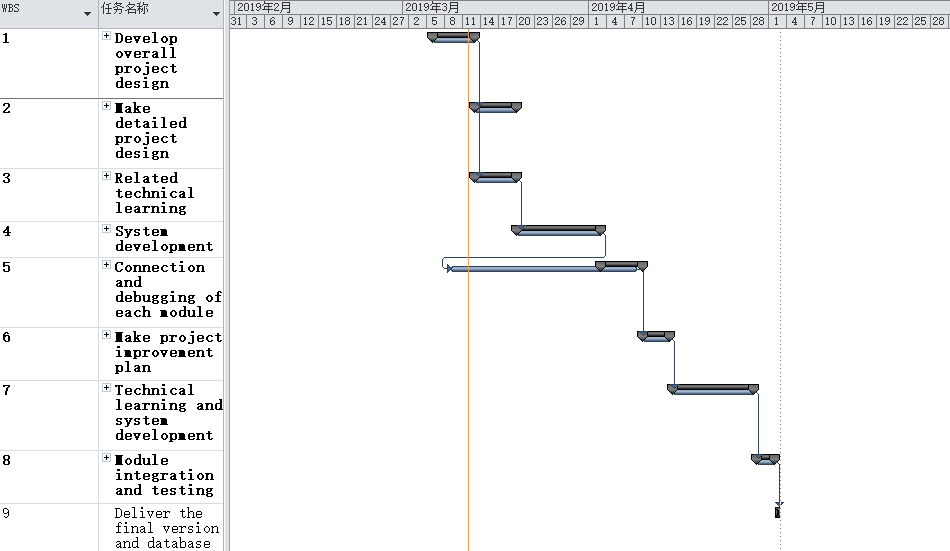
Display the query results on the web page

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 WBS and the gantt chart



## 3.5 work cycle（milestone）

The total cycle of the project is 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 milestones are as follows:

First iteration:

March 12→ submit the overall design (project plan)

March 19→ submit detailed design and technical study result

March 26→ submit the early code

April 2→ submit the middle code

April 9→ submit the post code

April 16→ deliver the initial system, make improvement plan

Second iteration:

April 23→ submit the early code

April 30→ submit the post code,system test result

May 1→ submit final product

# 4.Scope of software

4.1 project purpose

At present, there is a lack of tools to identify poetry in the market, especially the extraction of picture poetry. The system is dedicated to the extraction of picture poetry and poetry translation and appreciation. A simple picture will enable everyone to have a closer contact with the long-standing excellent cultural tradition of our Chinese nation, and let everyone have a closer taste of the beauty of poetry.

## 4.2 product description

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.

## 4.3 deliverable

The project eventually delivers a website product and a complete database, ensuring that users can submit photos and see full text and detailed information about the corresponding poems.

## 4.4 constraint

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.

## 4.5 assumed condition

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

It is assumed that the crawler successfully crawls to obtain relatively complete poetry data.

Assume that software requirements do not change.

## 4.6 Project acceptance criteria

It is required that only after the completion of acceptance test, if the unsolved defects meet the following requirements, can the acceptance pass: a) a-level defects: 0; B) class b defects: 0; C) class c defects: less than or equal to 3% of the total number of defects; D) grade d defects: less than or equal to 5% of the total number of defects; E) class e defects: less than or equal to 15% of the total number of defects.

The defect level is defined in the appendix.

# 5.Detailed design of system

## 5.1 Task Description

Project team needs to be done in the entire project cycle keep users to upload images, identification of poem words/phrases, according to the poem/query the database, display the query results in web pages, poetry retrieval functions (user query according to the author/dynasty/category), provides the poetry download function and the realization of the function of website to share information, and ensure the stability and efficiency of the system.

## 5.2 summary on design

### 5.2.1 Limits And Restraint

The overall offline development environment of the project is based on the window operating system; The development language used is python3, and the development tool is pycharm; MySQL Community Server version; Project management tool github.

At present, the technical conditions of the team are a little short, and we need to learn some new technologies, especially in image processing and website design. The project adopts the strategy of iterative development, so the time of the first round of development is relatively tight, and if necessary, the time cycle of the first round of development will be extended.

### 5.2.2 Design principles and requirements

The project adopts the method of modular design to decompose the software into multiple independent modules, and different modules have different functions and responsibilities. Each module can be independently developed, tested, and finally assembled into complete software.

In order to reduce the coupling of the system, the project used indirect coupling during the first iteration, and the connection between modules was completely realized through the control and invocation of the main module.

Table name specification: prefix: P; Body: plural form of an English word without a suffix.

Field name: prefix free, body: the singular form of the English word for the field.

Function naming convention: no prefix, function function of the English word, if the compound word is the first letter of each word underlined.

Page naming specification: adopts PASCAL naming method.

Naming conventions for class names: use the PASCAL nomenclature.

Naming conventions for method names: use the PASCAL nomenclature.

Naming conventions for variable names: use the hump method

The database design satisfies the second paradigm, that is, each row of data can be uniquely distinguished.

## 5.3 System logic design

### 5.3.1 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.

### 5.3.2 system interface design

First iteration：

## 

front-end

f

a

backstage

b

c

d

e

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

i

h

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

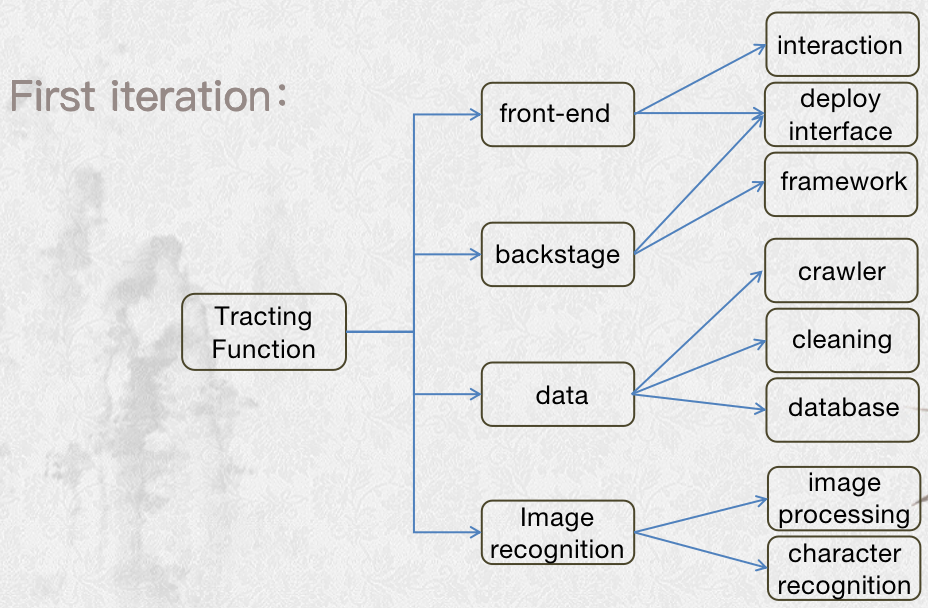
get\_mess\_background(background) ——output background

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

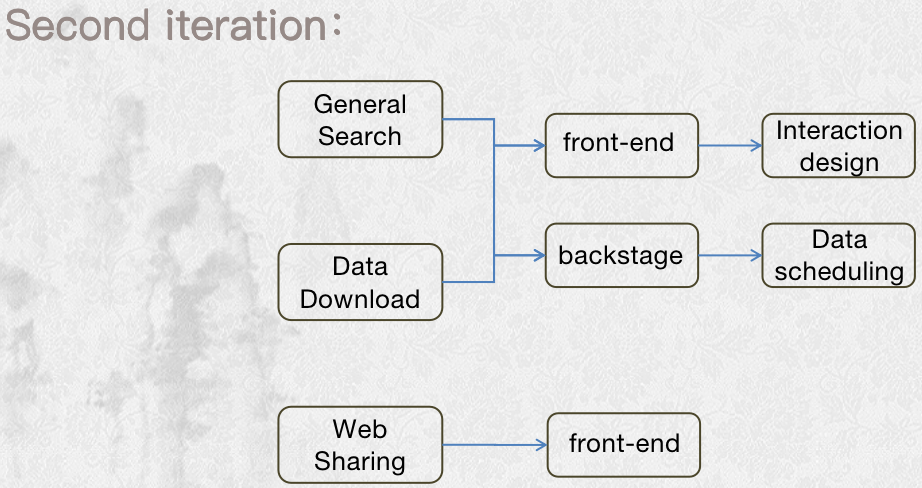
get\_mess\_appreciation(appreciation) ——output appreciation

## 5.4 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

## appendix

Defect level definition:

Defects are divided into five levels: A, B, C, D and E:

|  |  |
| --- | --- |
| A stage | OS crash Severe functional deficiency Program not running |
| B stage | Major functionality cannot be implemented  Program crashes  Text error on main page  The debug information was not cleared |
| C stage | The functional implementation is inconsistent with the requirements  Function can not be implemented but does not affect the use  Program logic error  Users are seriously inconvenient to use |
| D stage | Function implementation but use inconvenience  The prompt message is not uniform  The layout of the interface is not in keeping with user habits |
| E stage | Error message text  Negotiable page layout  Overall program tone |