智能简历解析系统

详细设计文档

2023年 7月

目 录

[1.引言 3](#_Toc140349629)

[2 系统架构设计 3](#_Toc140349630)

[3 模块设计 4](#_Toc140349631)

[3.1 简历信息录入模块 4](#_Toc140349632)

[3.2 简历解析模块 5](#_Toc140349633)

[3.3 关键信息提取模块 5](#_Toc140349634)

[3.4 人才画像构建模块 5](#_Toc140349635)

[3.5 简历信息统计与展示模块 5](#_Toc140349636)

[3.6 视图模块 5](#_Toc140349637)

[3.7 控制器模块 6](#_Toc140349638)

[4. 接口设计 6](#_Toc140349639)

[4.1 简历信息录入模块接口 6](#_Toc140349640)

[4.2 简历解析模块接口 7](#_Toc140349641)

[4.3 关键信息提取模块接口 7](#_Toc140349642)

[4.4 人才画像构建模块接口 7](#_Toc140349643)

[4.5 简历信息统计与展示模块接口 7](#_Toc140349644)

[4.6 视图模块接口 7](#_Toc140349645)

[4.7 控制器模块接口 8](#_Toc140349646)

[4.8 总结 8](#_Toc140349647)

[5. 数据结构和算法模型 9](#_Toc140349648)

[5.1 数据结构 10](#_Toc140349649)

[5.2算法设计 10](#_Toc140349650)

[6 界面设计 11](#_Toc140349651)

[6.1简历信息录入 11](#_Toc140349652)

[6.2自动提取简历关键信息 13](#_Toc140349653)

[6.3人才画像构建与岗位匹配 13](#_Toc140349654)

[6.4简历信息统计与展示 16](#_Toc140349655)

[7.安全设计 18](#_Toc140349656)

[8.性能设计 18](#_Toc140349657)

# 1.引言

本文档主要介绍智能简历解析系统的详细设计，包括系统架构、模块划分、接口设计、数据结构和算法等内容。本详细设计文档旨在为开发团队提供详细的技术指导，以便顺利进行系统开发和实现。

# 2 系统架构设计

智能简历解析系统采用MVC（Model-View-Controller）设计模式，主要包括以下三个部分：

Model（模型）：负责简历解析、提取关键信息和人才画像构建等业务逻辑处理。

View（视图）：负责用户界面的展示，以PyQt实现客户端的GUI界面。

Controller（控制器）：负责接收用户输入，调用模型处理数据，并更新视图的显示。

系统架构如下图所示：

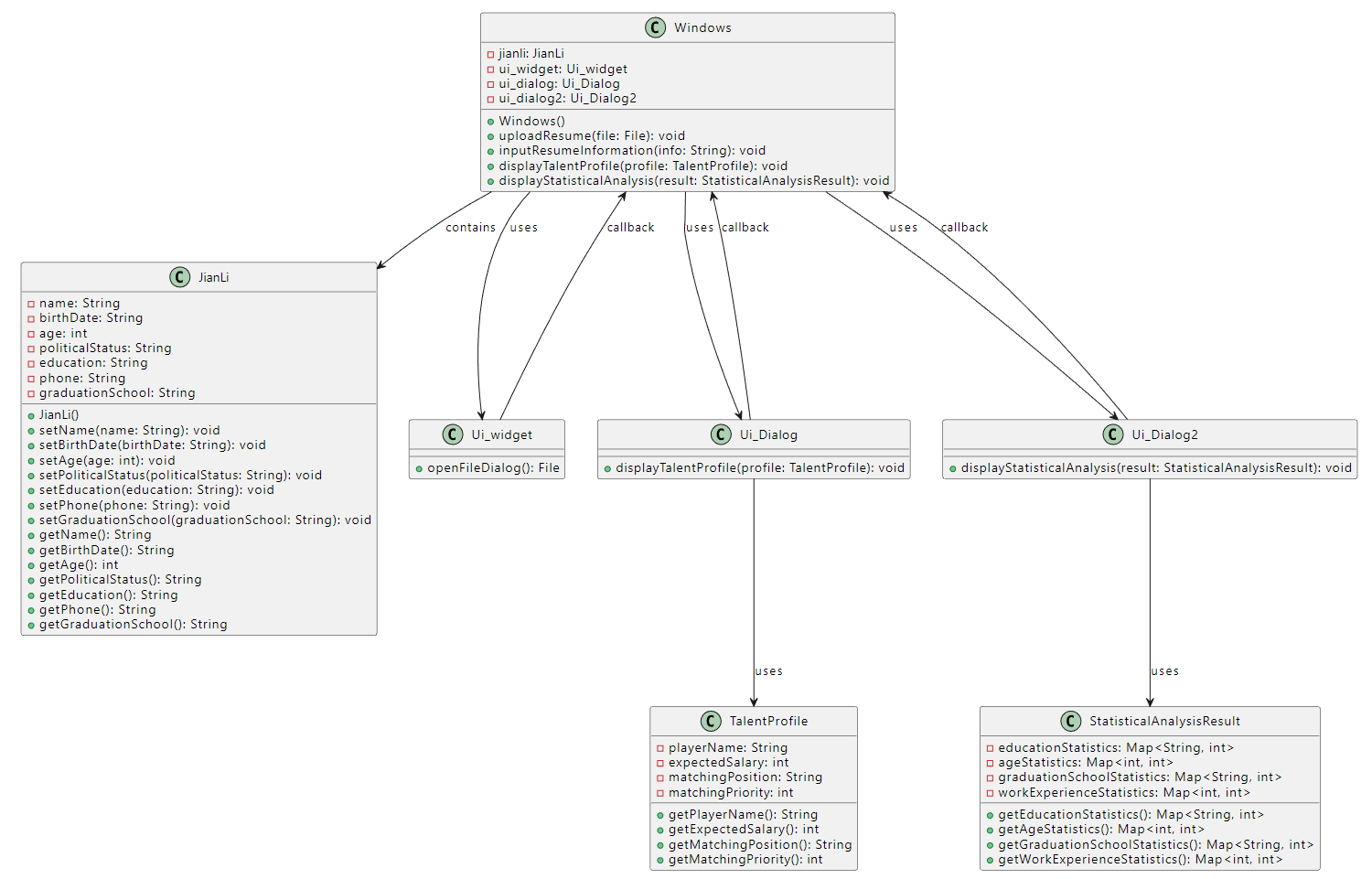


图 2-1 系统架构图

服务端采用Python语言，使用PaddleNLP进行简历解析和关键信息提取，主要包括以下内容：

文本预处理：对输入的简历文本进行预处理，将上传的Docx，PDF，PNG，TXT转化为待识别的文本。

实体识别：根据预处理后的文本，使用PaddleNLP的实体识别功能，提取出简历中的关键信息，例如姓名、性别、年龄、教育背景、工作经验等。

人才画像构建：根据简历中提取出的关键信息，对求职者的能力和特点进行分析和构建，形成人才画像。

客户端采用Python语言，使用PyQt库进行界面设计和交互实现，主要包括以下模块：

简历信息录入模块：用户可以通过客户端界面，输入简历信息并上传简历文件，系统会自动进行简历解析和关键信息提取。

简历信息统计和展示模块：用户可以通过客户端界面，查看已经上传的简历信息，并进行统计和展示。用户可以根据不同的维度，例如学历、工作年限等进行查看。

人才画像构建模块：用户可以通过客户端界面，查看求职者的人才画像构建结果，并进行分析和比较，为招聘决策提供科学依据。

# 3 模块设计

根据系统功能需求和MVC设计模式，将系统划分为以下几个模块：简历信息录入模块，简历解析模块，关键信息提取模块，人才画像构建模块，简历信息统计与展示模块，视图模块，控制器模块，下面将对每个模块进行详细说明。

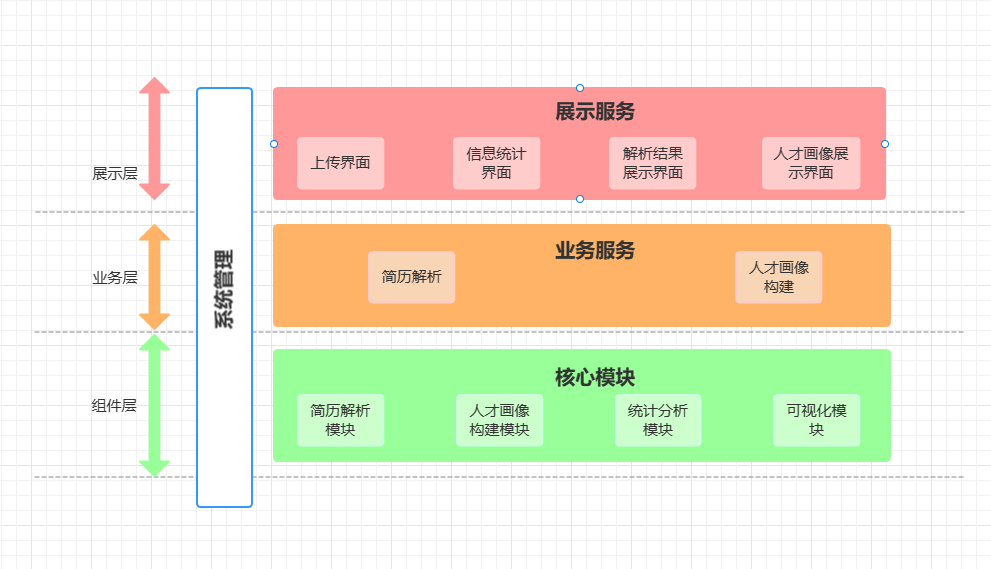


图3-1 系统模块图

## 3.1 简历信息录入模块

该模块主要负责用户通过GUI界面上传简历文件，支持的文件格式包括DOCX、PDF、PNG等。模块实现以下功能：

1.通过文件选择对话框让用户选择简历文件，通过文本框让用户输入简历信息。

2.将用户选择的简历文件保存至本地。

3.将简历文件的路径传递给控制器模块。

## 3.2 简历解析模块

该模块负责解析用户上传的简历文件，将简历内容转换为结构化数据。模块实现以下功能：

1.根据简历文件的格式，调用相应的解析工具（如：pdf2image、pymupdf、移动云OCR等）解析文件内容。

2.将解析后的简历内容转换为结构化数据（python的字典），再将结构化数据传递给关键信息提取模块。

## 3.3 关键信息提取模块

该模块负责从结构化的简历数据中提取关键信息，如姓名、年龄、最高学历、毕业院校、工作年限等。模块实现以下功能：

1.使用PaddleNLP处理结构化简历数据，识别并提取关键信息。

2.将提取到的关键信息传递给人才画像构建模块。

## 3.4 人才画像构建模块

该模块根据提取出的关键信息构建求职者的人才画像，并进行岗位匹配。模块实现以下功能：

1.根据关键信息，对求职者进行画像，包括技能特长、行业背景、工作经验等方面的分析。

2.根据岗位信息，对求职者与岗位进行匹配，计算匹配度。

3.将简历解析结果传递给视图模块。

## 3.5 简历信息统计与展示模块

该模块负责汇总和展示简历解析结果，包括关键信息、人才画像和岗位匹配度等。模块实现以下功能：

1.对提取到的关键信息进行统计和汇总。

2.将人才画像和匹配度信息进行整合。

3.将整合后的简历解析结果传递给视图模块。

## 3.6 视图模块

该模块负责通过PyQt实现客户端的GUI界面，展示简历解析结果。模块实现以下功能：

1.设计和实现简历解析系统的主界面和各个子界面。

2.接收简历信息统计与展示模块传递的数据，并在相应界面进行展示。

3.获取用户输入，如文件上传、按钮点击等操作，并将操作信息传递给控制器模块。

## 3.7 控制器模块

该模块负责接收视图模块的用户输入，调用相应的模型处理数据，并更新视图的显示。模块实现以下功能：

1.接收视图模块传递的用户操作信息。

2.根据用户操作，调用相应的模型处理数据。

3.将处理后的数据传递给视图模块更新显示。

# 4. 接口设计

本节将详细介绍各个模块之间的接口设计，包括输入参数、输出参数和调用方式等信息。

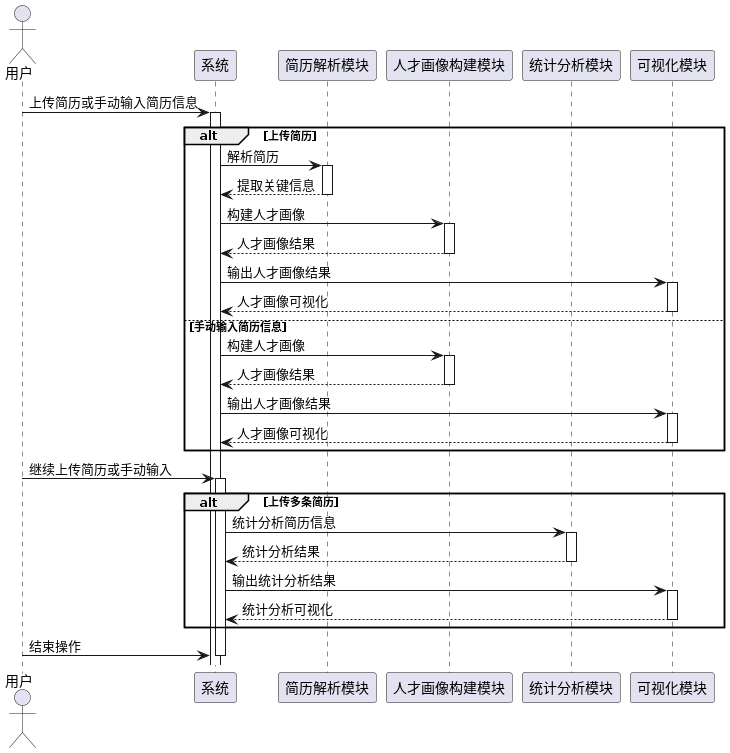


图4-1 系统时序图

## 4.1 简历信息录入模块接口

函数名：load\_file

输入参数：无

输出参数：简历文件路径（字符串）

调用方式：控制器模块调用

## 4.2 简历解析模块接口

函数名：parse\_resume

输入参数：简历文件路径（字符串）

输出参数：结构化简历数据（python字典）

调用方式：控制器模块调用

## 4.3 关键信息提取模块接口

函数名：get\_info

输入参数：结构化简历数据（python字典）

输出参数：关键信息（python字典或excel表格）

调用方式：控制器模块调用

## 4.4 人才画像构建模块接口

函数名：process\_result

输入参数：关键信息（python字典）

输出参数：人才画像和匹配度信息（JSON对象）

调用方式：控制器模块调用

## 4.5 简历信息统计与展示模块接口

函数名：summarize\_resume\_info

输入参数：关键信息（JSON对象或excel表格）

输出参数：整合后的简历解析结果（python字典）

调用方式：控制器模块调用

## 4.6 视图模块接口

函数名：update\_view

输入参数：整合后的简历解析结果（python字典）

输出参数：无（用户界面）

调用方式：控制器模块调用

## 4.7 控制器模块接口

函数名：handle\_user\_operation

输入参数：用户操作信息（字典或JSON对象）

输出参数：无

调用方式：视图模块调用

## 4.8 总结

简历解析类的类图

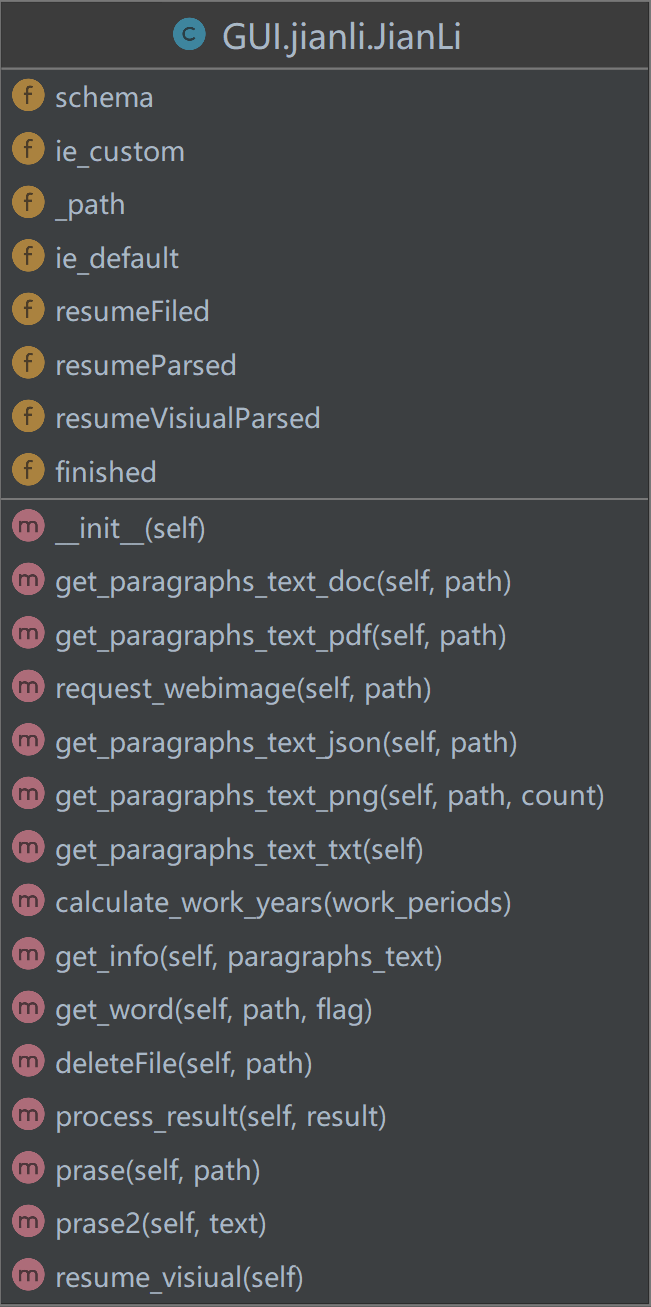


图4-8-1 简历解析类的类图

视图控制类的类图

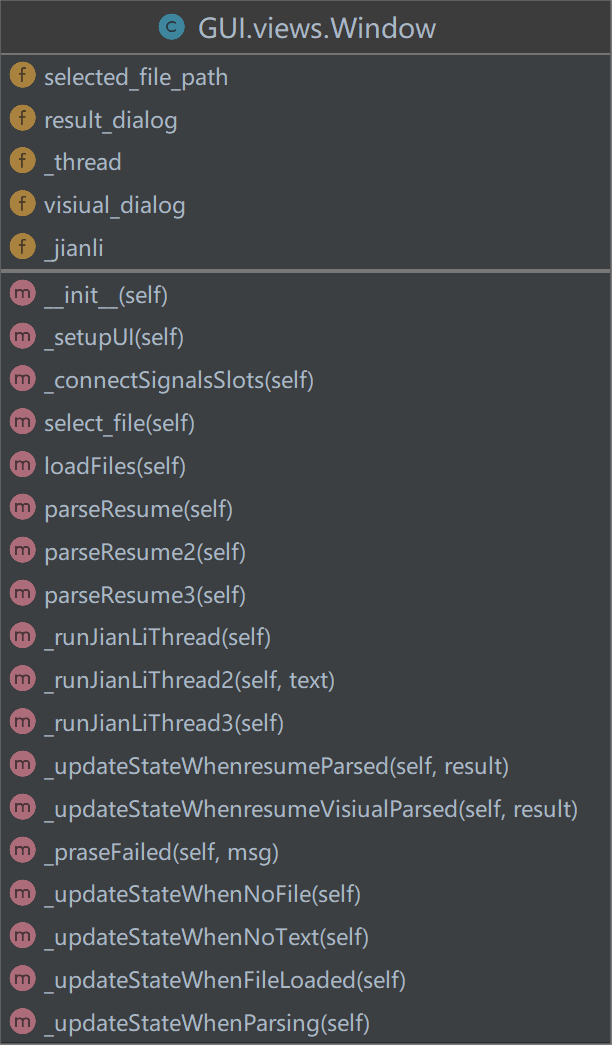


图4-8-2 视图控制类的类图

# 5. 数据结构和算法模型

本节将简要介绍系统中涉及的数据结构和算法模型。

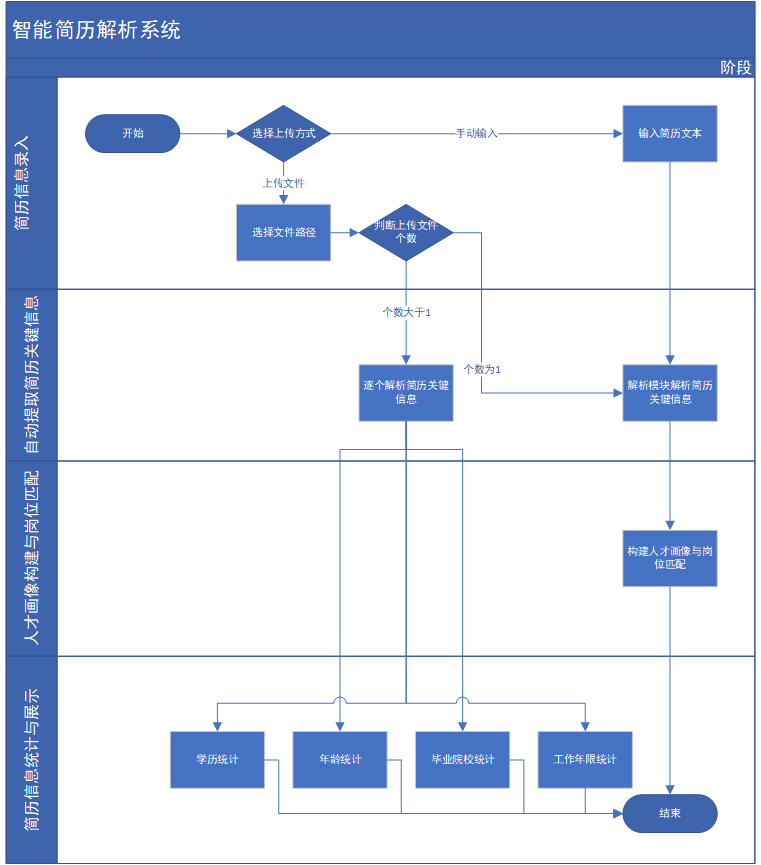


图5-1 系统总体流程图

## 5.1 数据结构

1.简历数据：字符串格式，将这种格式的简历信息转换为字符串。

2.关键信息：使用python字典，键值对包括‘姓名’，‘出生年月’，‘年龄’,‘政治面貌’，‘学历’，‘电话’，‘毕业院校’，‘工作年限’。

3.人才画像：使用python字典，键值对包括‘选手标签1’，‘选手标签2’，‘预计薪酬’,‘岗位匹配’，‘匹配优先级’。

4.简历信息统计可视化：pandas格式，matplotlib图表包括学历统计条形图，年龄统计扇形图，毕业院校统计条形图，工作年限统计扇形图。

## 5.2算法设计

1.简历信息录入：用户可以上传单个或多个文件，当用户上传单个文件时，调用个人简历解析，当上传多个文件时，调用简历信息统计，简历识别完后删除用户的文件。也可以使用文字录入，直接进入关键信息提取步骤。

2.文件解析：通过判断文件格式，对应不同的解析策略。如果是docx格式，使用BeautifulSoup提取信息；如果是png格式，调用移动云OCR进行识别，将识别后的结果存储起来；如果是pdf格式，使用pdf2image将每一页转化为图片，再按照png的格式识别；如果是txt格式，直接进入关键信息提取步骤。流程图如下：

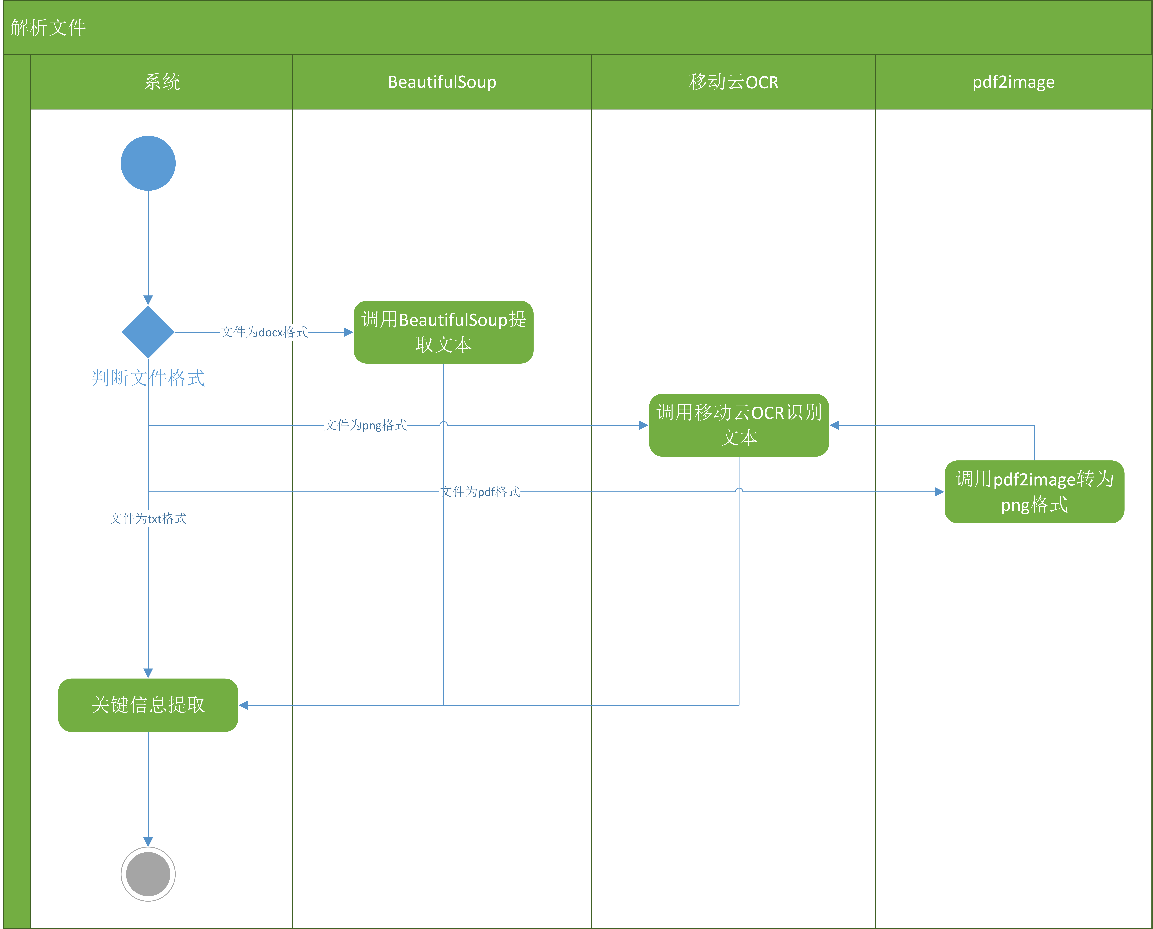


图5-2-1 文件解析流程图

3.关键信息提取：根据前面提取出的简历信息，调用飞桨paddle模型与自己的微调模型进行命名实体识别，提取出简历的性别，学历，工作年限等信息。

4.人才画像与岗位匹配：根据提取的关键信息，使用ai算法构建选手标签，预计薪酬和岗位匹配。

5. 简历信息统计与展示：调用matplotlib，结合前面的数据，完成学历统计条形图，年龄统计扇形图，毕业院校统计条形图，工作年限统计扇形图。

# 6 界面设计

## 6.1简历信息录入

首先定义了一个名为Ui\_widget的类，该类包含一个名为setupUi的方法，该方法用于设置UI界面的各种元素，例如进度条、标签和文本框等。在这个UI界面中，包含以下组件：QProgressBar：进度条，用于显示任务的进度。QLabel：标签，用于显示文件路径和标题文本。QWidget：窗口部件，用于包含其他组件。

文件上传简历原型界面（图6-1-1）



图6-1-1 文件上传简历原型界面

手动输入简历原型界面（图6-1-2）



图6-1-2 手动输入简历原型界面

简历信息录入类图（图6-1-3）

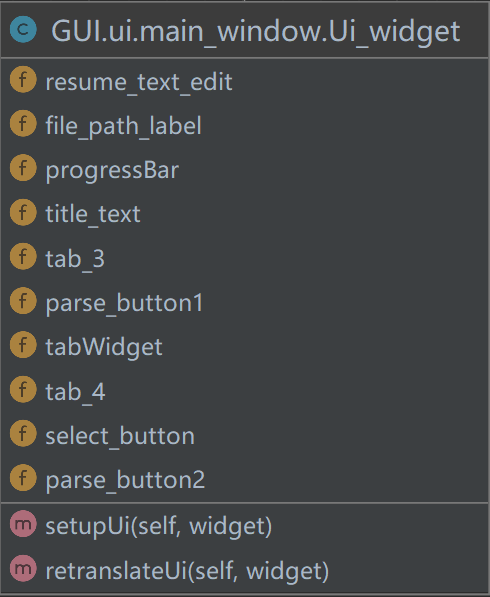


图6-1-3 简历信息录入类图

## 6.2自动提取简历关键信息

首先创建一个名为"parse\_result\_dialog"的对话框。这个对话框包含一个QTabWidget，其中包含两个QWidget，分别用于展示个人信息和工作经历。在个人信息的QWidget中，有姓名、出生日期、年龄、政治面貌、学历、电话、毕业院校和工作年限等信息。在人才画像的QWidget中，有求职者的选手标签，预计薪酬，岗位匹配和优先级等信息。还有一个QPushButton，用于关闭对话框。

简历解析结果原型界面（图6-2-1）



图6-2-1 简历解析结果原型界面

## 6.3人才画像构建与岗位匹配

人才画像构建与岗位匹配原型界面（图6-2-2）



图6-2-2 人才画像构建与岗位匹配原型界面

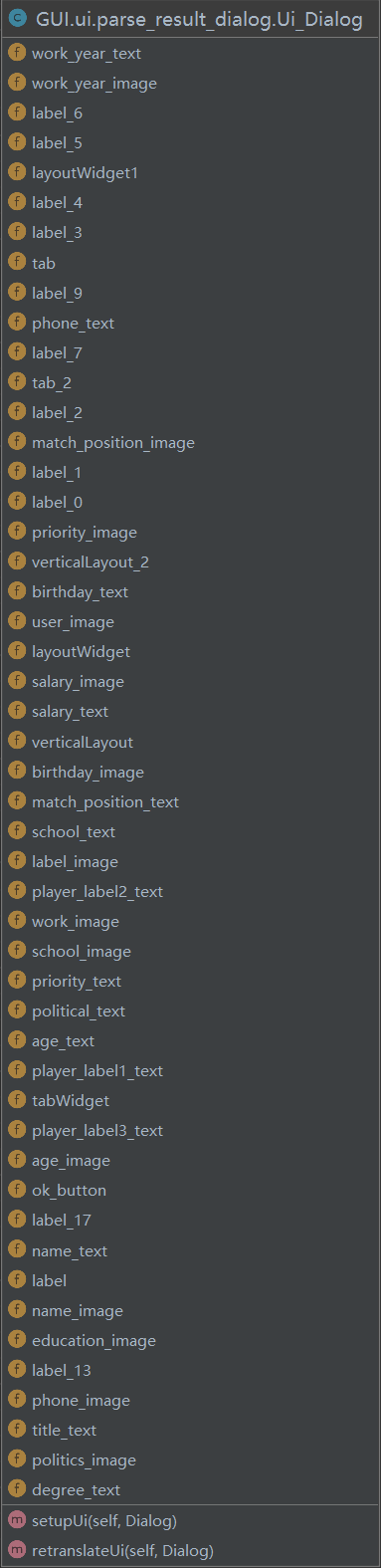


图6-2-3 简历解析结果类图

## 6.4简历信息统计与展示

首先创建一个名为"resume\_visual\_dialog"的对话框。这个对话框包含一个QTabWidget，其中包含4个QWidget，分别用于展示学历统计，年龄统计，毕业院校统计和工作年限统计。每个QWidget中都包含一个QWidget，用于包含matplotlib绘制的图，还有一个QPushButton，用于关闭对话框。

学历统计原型界面（图6-4-1）

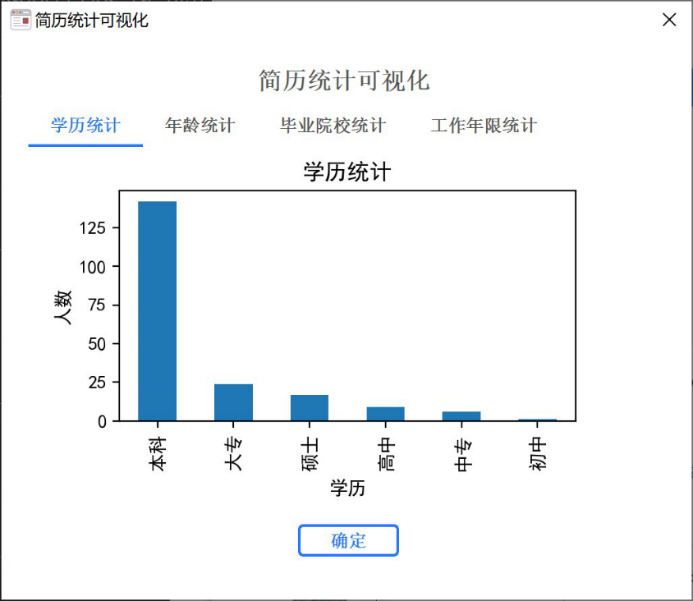


图6-4-1学历统计原型界面

学历统计原型界面（图6-2-2）

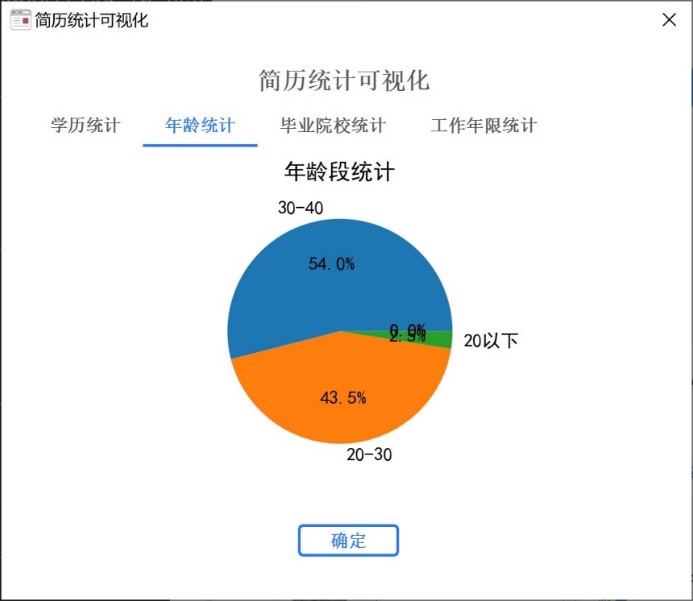


图6-4-2 年龄统计原型界面

工作年限统计原型界面（图6-4-3）

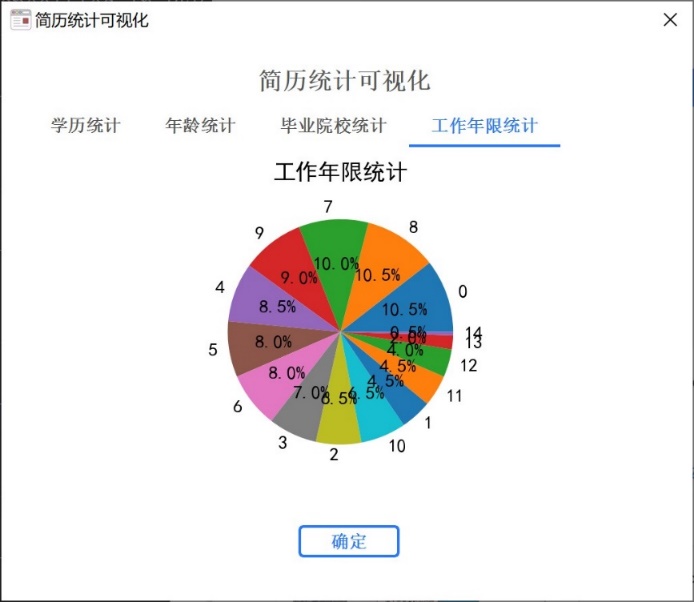


图6-4-3工作年限统计统计原型界面

毕业院校统计原型界面（图6-4-4）

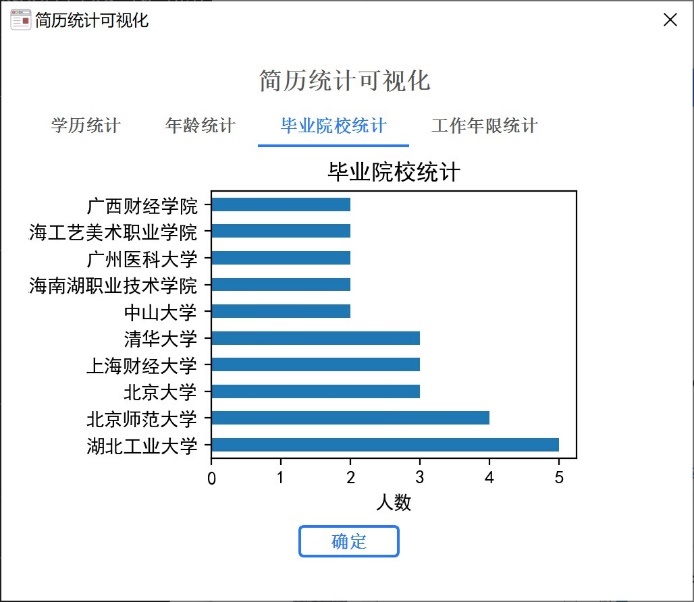


图6-4-4 毕业院校统计原型界面

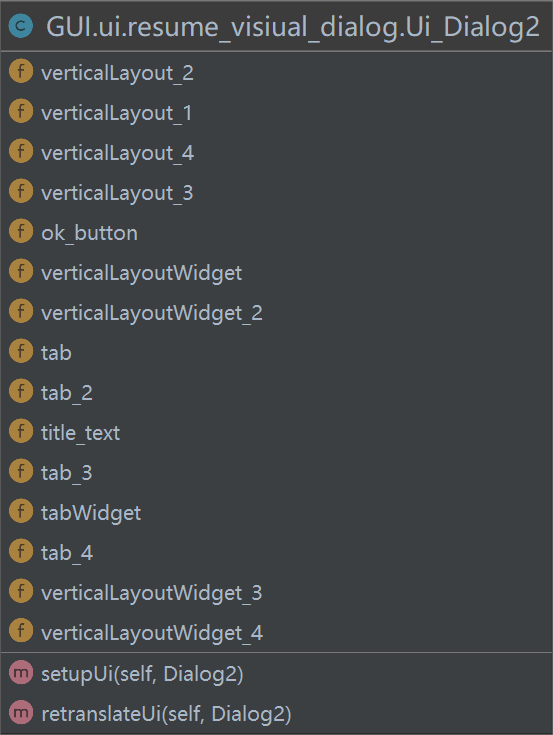


图6-4-5 简历统计可视化类图

# 7.安全设计

系统安全模块主要负责保证系统的安全和稳定性，防止系统遭受恶意攻击和破坏。

该系统的安全设计主要体现在数据安全，当用户上传简历后，简历暂时存在本地，当简历解析完成后，自动删除用户的简历，保护用户的隐私。

# 8.性能设计

性能设计是指在设计产品、系统、软件等时，为了满足特定的性能要求而进行的设计过程。

该简历解析系统由于需要调用人工智能模型，对显卡需求较大。对于纯cpu解析一份简历大概需要1分钟，对于一般显卡，解析大约需要2秒钟。

对于多条简历解析，将在九天•毕昇平台上解析，将结果保存下来，然后再本地进行展示，以此来优化性能，加快响应时间。