**毕业设计说明书**

**基于NLP的可视化评价信息收集与分析系统的设计与实现**

|  |  |
| --- | --- |
| **专 业** | **网络工程** |
| **学 生 姓 名** | **左明** |
| **班 级** | **B网络191** |
| **学 号** | **190107151003** |
| **指 导 教 师** | **仲冰** |
| **完 成 日 期** | **2023年 5月11日** |

**基于NLP的可视化评价信息收集与分析系统的设计与实现**

**摘 要**： 自然语言处理（NLP）是人工智能的一个重要领域，可以用来构建一个强大的可视化评估信息收集和分析系统。 在本文中，我们提出了一种基于 NLP 的系统的设计和实现，该系统利用 Spacy 进行文本分析，并利用数据可视化库 Echarts 进行数据呈现。

拟议的系统由两个主要部分组成：数据收集模块和分析模块。数据收集模块使用网络抓取技术从社交媒体平台和电商平台等各种来源收集数据。分析模块使用 Spacy 处理文本数据以提取有价值的信息，例如情绪、产品特性和用户偏好。 然后使用Echarts提供的可视化工具（例如条形图、饼图和折线图）呈现提取的信息。

该系统可应用于电子商务、社交媒体监控和公共安全等各个领域。 例如，在电子商务领域，该系统可用于收集客户对产品的反馈并分析情绪以提高产品质量。 在社交媒体监控中，该系统可用于检测不当内容并将其标记以供审查。

为了评估所提出系统的性能，我们对我们系统收集的文本数据集进行了实验。 结果表明，该系统可以准确地从文本数据中提取信息，并使用 Echarts 可视化呈现，为一些应用提供有价值的见解。

总的来说，所提出的系统提供了一个强大的工具，用于使用 NLP 技术收集和分析文本数据、用于数据收集的网络抓取以及用于数据分析的 Echarts 可视化，这可以帮助组织做出更好的决策并改进他们的服务。

**关键词**：自然语言处理；数据可视化；文本分析；网页抓取；情感分析

**Design and implementation of NLP-based visual evaluation information collection and analysis system**

**Abstract**: Natural Language Processing (NLP) is a major field in artificial intelligence that can be leveraged to build a robust visualizable evaluation information collection and analysis system. In this paper, we propose a design and implementation of an NLP-based system that utilizes Spacy for text analysis and Echarts, a data visualization library, for data presentation.

The proposed system consists of two main components: the data collection module and the analysis module. The data collection module uses web scraping techniques to gather data from various sources such as social media platforms and online marketplaces. The analysis module processes the textual data using Spacy to extract valuable information such as sentiment, product features, and user preferences. The extracted information is then presented using Echarts provided visualization tools such as bar charts, pie charts, and line graphs.

The proposed system can be applied in various domains such as e-commerce, social media monitoring, and public safety. For instance, in the e-commerce domain, the system can be used to collect customer feedback on products and analyze the sentiment to improve product quality. In social media monitoring, the system can be used to detect inappropriate content and flag it for review.

To evaluate the performance of the proposed system, we conducted experiments on a dataset of textual data collected by our system. The results show that the system can accurately extract information from textual data and present it using Echarts visualizations, providing valuable insights for various applications.

Overall, the proposed system offers a powerful tool for collecting and analyzing textual data using NLP techniques, web scraping for data collection, and Echarts visualizations for data analysis, which can help organizations make better decisions and improve their services.

**Key words**: Natural Language Processing; data visualization; text analysis; web scraping; sentiment analysis

**目 录**

[1 概述 2](#_Toc132445821)

[1.1 开发背景 2](#_Toc132445822)

[1.2 开发意义 3](#_Toc132445823)

[1.3 课题开发的工具及技术 4](#_Toc132445824)

[2 系统分析 7](#_Toc132445825)

[2.1 需求分析 7](#_Toc132445826)

[2.2 可行性分析 8](#_Toc132445827)

[2.3 数据库选择 9](#_Toc132445828)

[2.4 系统目标 9](#_Toc132445829)

[3 系统的概要设计 10](#_Toc132445830)

[3.1 总体设计 10](#_Toc132445831)

[3.2 系统架构 11](#_Toc132445832)

[3.3 系统的模块功能划分 12](#_Toc132445833)

[3.4 系统的接口设计 13](#_Toc132445834)

[3.5 系统涉及的数据结构 13](#_Toc132445835)

[3.6 系统的错误处理 14](#_Toc132445836)

[4 系统的详细设计 15](#_Toc132445837)

[4.1 系统接口设计 15](#_Toc132445838)

[4.2 系统登录模块设计 15](#_Toc132445839)

[4.3 系统数据收集模块设计 18](#_Toc132445840)

[4.3 普通用户模块设计 21](#_Toc132445841)

[5 系统实现 22](#_Toc132445842)

[5.1 系统接口的实现 22](#_Toc132445843)

[5.2 系统数据收集模块的实现 22](#_Toc132445844)

[5.3 系统数据展示模块的实现 22](#_Toc132445845)

[5.4 系统数据分析可视化模块的实现 22](#_Toc132445846)

[5.5 系统通知模块的实现 22](#_Toc132445847)

[5.6 系统Docker部署的实现 22](#_Toc132445848)

[6 系统测试 23](#_Toc132445849)

[6.1 系统登录模块测试 23](#_Toc132445850)

[6.2 系统数据收集模块测试 23](#_Toc132445851)

[6.3 系统数据展示模块测试 23](#_Toc132445852)

[6.4 系统数据分析可视化模块测试 23](#_Toc132445853)

[6.5 系统通知模块测试 23](#_Toc132445854)

[7 系统的发展方向及应用前景 23](#_Toc132445855)

[8 结束语 23](#_Toc132445856)

[参考文献 25](#_Toc132445857)

[致 谢 26](#_Toc132445858)

[附 录 27](#_Toc132445859)

[附录1 源程序清单 27](#_Toc132445860)

[附录1.1 系统登录模块 27](#_Toc132445861)

[附录1.2 接警人模块 27](#_Toc132445862)

[附录1.3 普通用户模块 27](#_Toc132445863)

**基于NLP的可视化评价信息收集与分析系统的设计与实现**

# 1 概述

## 1.1 开发背景

随着互联网和社交媒体的普及，人们越来越依赖于在线评价来做出决策，如选择商品、预定旅游行程、选购服务等。这些评价通常包含了大量的信息和情感，然而，由于人工处理这些信息耗时且容易出现误判，因此自动化地分析和处理这些信息变得尤为重要。

基于NLP的可视化评价信息收集与分析系统是一种可以自动地收集和分析在线评价的工具，它可以将大量的文本数据转换成易于理解和分析的形式，并可以使用可视化技术来帮助用户更好地理解和分析这些数据。

基于NLP的可视化评价信息收集与分析系统的开发背景就是为了满足人们处理和分析在线评价的需求，帮助用户更好地理解和分析这些数据，为他们做出更加明智的决策提供支持。

### 1.1.1 NLP技术的发展

自20世纪50年代以来，自然语言处理（NLP）技术在计算机科学、人工智能和语言学等领域得到了广泛应用和研究。随着计算机性能的提高和数据量的增加，NLP技术也得到了快速的发展和普及。

NLP技术的发展主要经历了以下几个阶段：

基于规则的方法：早期的NLP技术主要采用人工编写规则的方式进行文本分析和语言处理。这种方法需要耗费大量的人力和时间，且规则的编写和维护十分复杂。

统计方法：20世纪90年代，随着机器学习技术的兴起，NLP技术也开始采用统计方法进行文本分析和语言处理。这种方法利用大量的语言数据进行训练，可以自动学习语言规律和模式。

深度学习方法：近年来，深度学习技术的发展也给NLP技术带来了重大的进展。利用深度学习算法，可以构建更加复杂和准确的NLP模型，例如文本分类、命名实体识别、机器翻译等。

### 1.1.2 可视化技术的发展

可视化技术的发展主要经历了以下几个阶段：

基于手工绘图的方法：早期的可视化技术主要依靠人工绘制图表和图形来展示数据。这种方法需要耗费大量的人力和时间，且绘制的图表通常较为简单和粗糙。

统计图表方法：随着计算机技术的发展，出现了许多自动生成统计图表的软件工具，例如Excel和SPSS等。这种方法可以方便地生成各种类型的图表，但通常需要人工进行进一步的调整和优化。

交互式可视化方法：近年来，交互式可视化技术得到了快速发展。这种方法可以让用户通过交互操作来探索和分析数据，例如缩放、滚动和筛选等操作。交互式可视化技术通常采用Web技术和可视化库来实现。

### 1.1.3 爬虫技术的发展

爬虫技术是指利用程序自动获取和抓取互联网上的数据的技术。随着互联网的发展和数据量的增加，网络爬虫技术也得到了快速的发展和应用。

网络爬虫技术的发展主要经历了以下几个阶段：

手工抓取：早期的爬虫技术主要是手工抓取网页数据，需要人工浏览和提取数据，工作效率低下。

基于规则的自动化抓取：随着计算机技术的发展，出现了一些自动化抓取工具，例如wget和curl等。这种方法需要用户自己编写规则来指定要抓取的网页和数据，存在一定的局限性。

基于爬虫框架的自动化抓取：近年来，出现了许多开源的爬虫框架，例如Scrapy和BeautifulSoup等。这种方法可以更加灵活和高效地抓取数据，且具有很好的可扩展性和稳定性。

智能化爬虫技术：最近几年，随着人工智能和机器学习技术的发展，智能化爬虫技术得到了快速的发展。这种方法可以通过机器学习算法来识别网页的结构和内容，以便更准确地抓取数据。

智能化爬虫技术的发展将会进一步推动网络爬虫技术的发展和应用。随着数据量的不断增加，网络爬虫技术将会在许多领域得到广泛的应用，例如市场调研、舆情监测和数据挖掘等。同时，网络爬虫技术也需要注意数据安全和隐私保护等问题，以保证用户的数据安全和隐私不受侵犯。

## 1.2 开发意义

主要体现在以下几个方面：

提高评价信息的收集和分析效率：通过使用NLP和可视化技术对于评价信息进行处理和展示，可以大大提高评价信息的收集和分析效率，从而帮助企业、政府和个人等更好地了解用户的需求和反馈。

优化产品和服务质量：通过对于评价信息进行分析，可以得到用户对于产品和服务的反馈和建议，帮助企业和政府优化产品和服务质量，提高用户的满意度和忠诚度。

促进企业竞争力的提升：通过分析竞争对手的评价信息，可以了解其产品和服务的优势和劣势，从而帮助企业制定更好的竞争策略，提高企业的竞争力。

推动相关技术的研究和应用：该课题涉及到文本处理、情感分析和可视化技术等方面，可以促进相关技术的研究和应用，从而推动人工智能和可视化技术的发展。

服务于社会公共利益：通过对于政府公共服务的评价信息进行分析，可以帮助政府更好地了解公众对于政府服务的需求和反馈，从而提高公共服务的质量和效率，为社会公共利益服务。

该课题具有重要的实际意义和社会价值，可以为企业、政府和个人等提供有力的支持和帮助，同时也可以推动相关技术的研究和应用。

## 1.3 课题开发的工具及技术

### 1.3.1 开发工具

PyCharm 2023.1（专业版）

PyCharm 是专门为 Python 编程语言设计的集成开发环境 (IDE)。 它提供了广泛的工具和功能，例如代码完成、语法突出显示、代码分析、调试和测试。 PyCharm 还支持各种 Python 框架，如 Django、Flask 和 FastAPI，以及各种版本控制系统。

MongoDB Compass

MongoDB Compass 是 MongoDB 的图形用户界面工具，MongoDB 是一种 NoSQL 面向文档的数据库系统。 它允许开发人员以可视化和直观的方式与 MongoDB 数据进行交互，从而更轻松地创建、修改和分析数据。 借助 MongoDB Compass，开发人员还可以查询和分析数据，以及创建和管理索引。

Visual Studio Code

Visual Studio Code 是微软开发的一款流行的开源代码编辑器。 它支持包括 Python 在内的多种编程语言，并提供语法高亮显示、代码完成、调试和 Git 集成等广泛的功能。 它还支持扩展，方便开发者自定义和扩展编辑器的功能。

Git

Git 是一种分布式版本控制系统，用于在软件开发过程中跟踪源代码的更改。 它允许多个开发人员同时协作处理一个项目，同时跟踪更改和管理冲突。 Git 提供了范围广泛的命令和工具，用于管理代码更改、合并代码以及在必要时回滚更改。

Chrome

Chrome 是谷歌开发的网络浏览器。 它被开发人员广泛用于测试和调试 Web 应用程序。 它提供各种开发人员工具，例如开发人员控制台，允许开发人员查看和修改网页元素、运行 JavaScript 代码以及分析网络流量。 它还支持扩展，使开发人员可以轻松地向浏览器添加自定义功能。

### 1.3.2 技术选择

爬虫模块使用 Scrapy 来爬取评价信息，并将数据存储到 MongoDB 中。

NLP 模块使用 Spacy 来处理评价信息，提取关键信息并将其存储到 Elasticsearch 中。

可视化模块使用 Vue.js 来设计和实现前端界面，用户可以在界面上查看评价信息和相关分析结果。

后端使用 FastAPI 来实现 API，包括用户认证、数据查询和分析等功能。

任务调度和通知模块使用 Celery 和 RabbitMQ 来处理任务调度和通知。当爬虫爬取到新的评价信息时，会将其发送到 RabbitMQ 的队列中，Celery 会从队列中取出任务并调用相应的函数进行处理。

Apache Spark 用于处理大规模数据，可以对 Elasticsearch 中的评价信息进行分析和处理，并将结果存储到 MySQL 中。

Redis 用作缓存和消息代理，可以缓存一些常用的数据以加速访问，并作为 RabbitMQ 的消息代理。

Docker 和 Docker Swarm 用于部署和管理各个模块，可以将每个模块打包成 Docker 镜像，并使用 Docker Swarm 进行部署和管理。可以使用 Docker Compose 来定义和管理多个容器应用程序。

### 1.3.3 技术简介

Scrapy

Scrapy是一个Python的网络爬虫框架，可以自动抓取和处理网页数据。它提供了丰富的API，可以让开发者快速定义和运行爬虫，同时支持异步IO和并发处理等特性，可以高效地处理大量数据。使用Scrapy，开发者可以定义爬虫规则和提取数据的方式，将抓取的数据存储到本地或者远程数据库中，同时支持多种爬虫任务调度策略。

MongoDB

MongoDB是一个NoSQL数据库，采用文档型存储方式，可以快速处理大量数据。它支持高可用性和可扩展性，可以方便地进行分布式计算和数据处理。MongoDB具有简单易用的API和查询语言，同时支持全文搜索和地理位置查询等功能。在基于NLP的可视化评价信息收集与分析系统中，MongoDB可以作为数据存储和管理平台，将从爬虫中获取的评价信息存储到数据库中。

Spacy

Spacy是一个高性能的自然语言处理库，可以用于文本分类、实体识别等任务。它使用Cython编写，性能优异，同时提供了简单易用的API和预训练的模型，可以方便地进行文本处理和分析。在基于NLP的可视化评价信息收集与分析系统中，Spacy可以作为NLP模块的核心，对爬虫获取的评价信息进行清洗、处理、分类和实体识别等任务，并将处理结果存储到数据库中。

Elasticsearch

Elasticsearch是一种基于Lucene的搜索引擎，可以支持全文搜索和数据可视化。它采用分布式架构，可以高效地处理大规模数据，并提供了丰富的查询API和聚合函数等特性，可以方便地进行数据分析和可视化。在基于NLP的可视化评价信息收集与分析系统中，Elasticsearch可以作为可视化模块的数据源，将处理后的评价信息进行可视化展示，并提供数据查询和聚合等功能。

Vue.js

Vue.js是一种流行的JavaScript框架，可以快速构建复杂的单页应用程序。它具有简单易用的API和组件化开发方式，同时支持虚拟DOM和响应式数据绑定等特性，可以提高开发效率和应用程序性能。在基于NLP的可视化评价信息收集与分析系统中，Vue.js可以作为可视化模块的前端界面，实现数据展示和用户交互等功能。

FastAPI

FastAPI是一个快速构建API的Python框架，具有高性能、易用、文档友好等特点。它采用异步IO和类型注解等特性，可以提高API的性能和开发效率，同时支持OpenAPI和Swagger等文档自动生成工具，可以方便地进行API文档生成和测试。在基于NLP的可视化评价信息收集与分析系统中，FastAPI可以作为后端模块的核心，实现用户认证、数据查询和分析等功能，并提供API文档和测试工具。

Celery

Celery是一个Python的分布式任务队列，可以支持异步任务调度和消息队列。它采用消息传递和事件驱动的方式，可以方便地处理异步任务和并发计算，同时支持多种消息中间件和存储后端，可以快速地进行任务调度和分布式计算。在基于NLP的可视化评价信息收集与分析系统中，Celery可以作为任务调度和通知模块的核心，实现爬虫任务调度、数据处理和通知等功能。

RabbitMQ

RabbitMQ是一个开源的消息代理软件，可以支持分布式计算和异步任务调度。它采用AMQP协议和消息队列方式，可以高效地处理消息传递和事件驱动，同时支持多种客户端和后端存储，可以方便地进行分布式计算和异步任务调度。在基于NLP的可视化评价信息收集与分析系统中，RabbitMQ可以作为消息代理和任务队列的核心，实现爬虫任务调度、数据处理和通知等功能。

Apache Spark

Apache Spark是一种大规模数据处理引擎，可以支持分布式计算和处理大规模数据。它采用内存计算和RDD抽象层，可以快速地进行数据处理和分析，并提供了丰富的API和算法库，可以方便地进行数据挖掘和机器学习等任务。在基于NLP的可视化评价信息收集与分析系统中，Apache Spark可以作为大规模数据处理和分析的引擎，实现对Elasticsearch中的评价信息进行分析和处理，并将结果存储到MySQL中。

Redis

Redis是一个内存数据库，可以作为高速缓存和消息代理使用。它采用键值存储和发布订阅模式，可以方便地进行数据缓存和消息传递，同时支持多种数据结构和存储方式，可以满足不同的应用场景。在基于NLP的可视化评价信息收集与分析系统中，Redis可以作为缓存和消息代理，缓存一些常用的数据以加速访问，并作为RabbitMQ的消息代理。

Docker

Docker是一个开源的容器化平台，可以打包应用程序和服务到容器中，实现快速部署和管理。它采用轻量级的容器技术，可以快速地构建、运行和部署应用程序，同时支持多种操作系统和云平台，可以方便地进行应用程序的扩展和迁移。在基于NLP的可视化评价信息收集与分析系统中，Docker可以作为部署和管理平台，将每个模块打包成Docker镜像，并使用Docker Swarm进行部署和管理，可以方便地进行应用程序的扩展和迁移。

Docker Swarm

Docker Swarm是Docker官方提供的容器编排和管理平台，可以支持集群管理和应用程序自动伸缩。它采用集群管理和负载均衡技术，可以方便地进行容器的部署和管理，同时支持服务发现和自动伸缩等特性，可以提高应用程序的可用性和性能。在基于NLP的可视化评价信息收集与分析系统中，Docker Swarm可以作为集群管理和应用程序自动伸缩的平台，可以实现高可用性和可扩展性。

# 2 系统分析

在软件需求分析过程中，计划阶段初步确定的软件范围将得到提炼且具体化，并分析各软件部件可能采用的解决办法。完善的软件需求说明是软件开发项目得以成功的基础。不管设计如何精心或者编码如何巧妙，如果对软件需求不加以明确规定，将使用户感到失望，并给软件开发带来严重后果。

## 2.1 需求分析

### 2.1.1 系统实现的功能

基于NLP的可视化评价信息收集与分析系统的功能主要包括：

爬虫模块：实现自动抓取目标网站的评价信息，并将数据存储到MongoDB中。

NLP模块：对爬虫获取的评价信息进行清洗、处理、分类和实体识别等任务，并将处理结果存储到Elasticsearch中。

可视化模块：将Elasticsearch中的评价信息进行可视化展示，并提供数据查询和聚合等功能。

后端模块：提供API接口，实现用户认证、数据查询和分析等功能。

任务调度和通知模块：实现爬虫任务调度、数据处理和通知等功能。

### 2.1.2 系统的功能需求

基于NLP的可视化评价信息收集与分析系统的功能需求包括：

爬虫模块需求：

（1）自动抓取目标网站的评价信息；

（2）支持多个目标网站的抓取；

（3）支持对抓取规则的动态配置；

（4）将数据存储到MongoDB中。

NLP模块需求：

（1）支持对评价信息的清洗、处理、分类和实体识别等任务；

（2）支持对NLP模型的动态更新和配置；

（3）将处理结果存储到Elasticsearch中。

可视化模块需求：

（1）支持对Elasticsearch中的评价信息进行可视化展示；

（2）支持数据查询和聚合等功能；

（3）支持对可视化界面的动态更新和配置。

后端模块需求：

（1）提供API接口，实现用户认证、数据查询和分析等功能；

（2）支持对API接口的动态更新和配置。

任务调度和通知模块需求：

（1）实现爬虫任务调度、数据处理和通知等功能；

（2）支持对任务调度和通知的动态更新和配置。

## 2.2 可行性分析

基于NLP的可视化评价信息收集与分析系统使用的技术包括FastAPI、Vue.js、Scrapy、Celery、RabbitMQ、Apache Spark、Spacy、Redis、MongoDB、Elasticsearch和Docker Swarm等。

### 2.2.1 技术可行性

这些技术都是经过广泛应用和验证的成熟技术，可以满足系统的需求，并且具备足够的性能和可扩展性。FastAPI是一个快速构建API的Python框架，具有高性能、易用、文档友好等特点；Vue.js是一种流行的JavaScript框架，可以快速构建复杂的单页应用程序；Scrapy是一个Python的网络爬虫框架，可以自动抓取和处理网页数据；Celery和RabbitMQ可以协同工作，使得任务调度和消息队列变得简单；Apache Spark是一种大规模数据处理引擎，可以支持分布式计算；Spacy是一个高性能的自然语言处理库，可以用于文本分类、实体识别等任务；Redis和MongoDB是NoSQL数据库，可以快速处理大量数据；Elasticsearch是一种基于Lucene的搜索引擎，可以支持全文搜索和数据可视化；Docker Swarm是一个容器编排和管理平台，可以使得系统的部署和管理更加方便和高效。

### 2.2.2 经济可行性

这些技术都是开源的或者有免费版本，可以减少系统的开发和运营成本。同时，这些技术也有商业版本和服务，可以支持更高级别的功能和支持。

### 2.2.3 数据可行性和安全可行性

数据可行性需要考虑以下几个方面：

数据来源：系统需要从多个渠道收集评价信息，例如社交媒体、评论和问卷调查等。需要评估这些数据来源的可靠性和数据量是否足够，以及数据是否易于清洗和处理。

数据清洗和处理：系统需要对收集到的评价信息进行清洗和处理，例如去除重复数据和无效数据，以便更准确地分析数据。需要评估清洗和处理数据的成本和效率。

数据存储和管理：系统需要建立数据存储和管理机制，以保证数据的安全和可靠性。需要评估数据存储和管理的成本和效率。

安全可行性需要考虑以下几个方面：

数据安全：系统需要采取合适的技术措施保护数据的安全，例如加密和访问控制等。

隐私保护：系统需要保护用户的隐私不受侵犯，例如匿名化处理和数据脱敏等。

系统安全：系统需要保护系统的安全，防止遭受黑客攻击和恶意软件侵入等，需要采取合适的安全措施，例如身份验证和防火墙等。

基于NLP的可视化评价信息收集与分析系统使用的这些技术在数据处理和安全方面都具有一定的优势。例如，MongoDB和Elasticsearch具有高性能、可扩展和数据可视化等特点，可以满足系统的数据处理需求；Redis可以作为高速缓存提高数据访问速度；Spacy可以支持文本处理和语言分析等任务。同时，这些技术也具有相应的安全特性，例如可以支持数据加密和身份验证等措施，保护用户的数据安全和隐私不受侵犯。

## 2.3 数据库选择

在为基于 NLP 的视觉评估信息收集和分析系统选择数据库时，需要考虑几个因素。 基于系统的需求和特点，MongoDB是较适合的数据库选择，原因如下：

面向文档：MongoDB 是一个面向文档的数据库，这意味着它将数据存储在类似 JSON 的文档中，这些文档可以具有不同的结构。这使得它非常适合存储非结构化和半结构化数据，这是基于 NLP 的系统的典型特征。

可扩展性：MongoDB 具有高度可扩展性，可以轻松处理大量数据。它支持分片，可以跨多个服务器进行水平扩展。这对于需要频繁更新和实时分析数据的系统来说很重要。

灵活性：MongoDB 在模式设计方面提供了高度的灵活性，允许随着需求的发展改变数据结构。这对于处理不断变化和演变的数据的基于NLP 的系统特别有用。

查询能力：MongoDB 拥有强大而灵活的查询语言，可以实现复杂的查询和聚合操作。这对于需要对文本数据进行复杂分析和处理的系统来说非常重要。

与其他工具集成：MongoDB 与基于 NLP 的系统中常用的其他工具和技术很好地集成，例如 Python、NLTK 和 spaCy。

## 2.4 系统目标

基于NLP的可视化评价信息收集与分析系统的目标是：

使用自然语言处理（NLP）技术收集和分析评价信息。该系统的主要目标是通过从基于文本的评估（例如评论或评论）中提取有意义的见解并将其转换为可视化的指标来自动分析文本数据。

该系统可用于多种应用，例如分析客户对产品的反馈、评估用户对网站的参与度，或监控社交媒体对品牌或活动的情绪。通过自动分析文本数据，系统可以节省时间和资源，同时提供对用户行为和偏好的宝贵见解。

提供一个集成化的平台，支持从多个渠道收集评价信息，进行数据清洗和处理，并进行可视化分析和展示。

提供多种数据可视化方式，例如词云图、柱状图、折线图等，以及文本可视化分析，例如词向量图、情感图等，帮助用户更直观地理解和分析数据。

具备足够的性能和可扩展性，能够快速处理大量的评价信息，并支持分布式计算和云计算平台。

保证数据的安全和用户隐私的保护，采取合适的技术措施，例如身份验证、数据加密和访问控制等。

不断优化和改进系统功能和性能，以满足用户需求和市场竞争。

# 3 系统的概要设计

## 3.1 总体设计

### 3.1.1 需求规定

主要的输入输出项目：

该系统应该能够从多个来源收集评估信息，包括基于文本的评论和评论。然后，它应该使用自然语言处理技术来分析数据并以可视化指标和可视化的形式生成有意义的见解。

功能性能要求：

1）输入：系统应该能够处理文本格式的输入数据。它还应该能够同时处理来自多个来源的输入。

2)处理：系统应使用自然语言处理技术来分析文本数据并生成有意义的见解。

3)输出：系统应提供可量化的指标和可视化效果，以总结分析产生的见解。它还应提供用于编辑和自定义输出的工具，例如过滤和排序选项。

### 3.1.2 操作环境

硬件：系统应该能够在标准计算机硬件上运行，包括台式机和笔记本电脑。

系统环境：系统应能与Python、NLTK、OpenCV等自然语言处理和视觉分析中常用的其他软件工具集成。它还应该为用户提供文档和支持资源。

### 3.1.3概念和处理流程

基于NLP的可视化评价信息采集与分析系统的基本设计理念是利用自然语言处理技术对文本数据进行自动化分析。该系统从各种来源收集视觉评估信息，然后使用 NLP 技术对其进行处理以提取有意义的见解。然后，这些见解以可量化指标和可视化的形式呈现，可以根据需要进行定制和编辑。

系统的处理流程如下图所示：

[输入数据] --> [数据预处理] --> [NLP分析] --> [输出生成] --> [自定义和编辑] --> [共享和广播]（需要将其转换成图片）

系统首先对输入数据进行预处理，以确保其采用适合分析的格式。然后它对文本数据执行 NLP 分析以提取有意义的见解。生成的见解随后用于生成可量化指标和可视化形式的输出。在共享和广播给他人之前，用户可以自定义和编辑输出。

在编程概念方面，系统利用模块和子程序将分析过程分解为更小、更易于管理的组件。 它还利用并发性和并行性来优化性能，并在必要时跨多台机器分配处理任务。

## 3.2 系统架构

基于NLP的可视化评价信息收集与分析系统的架构图如下：

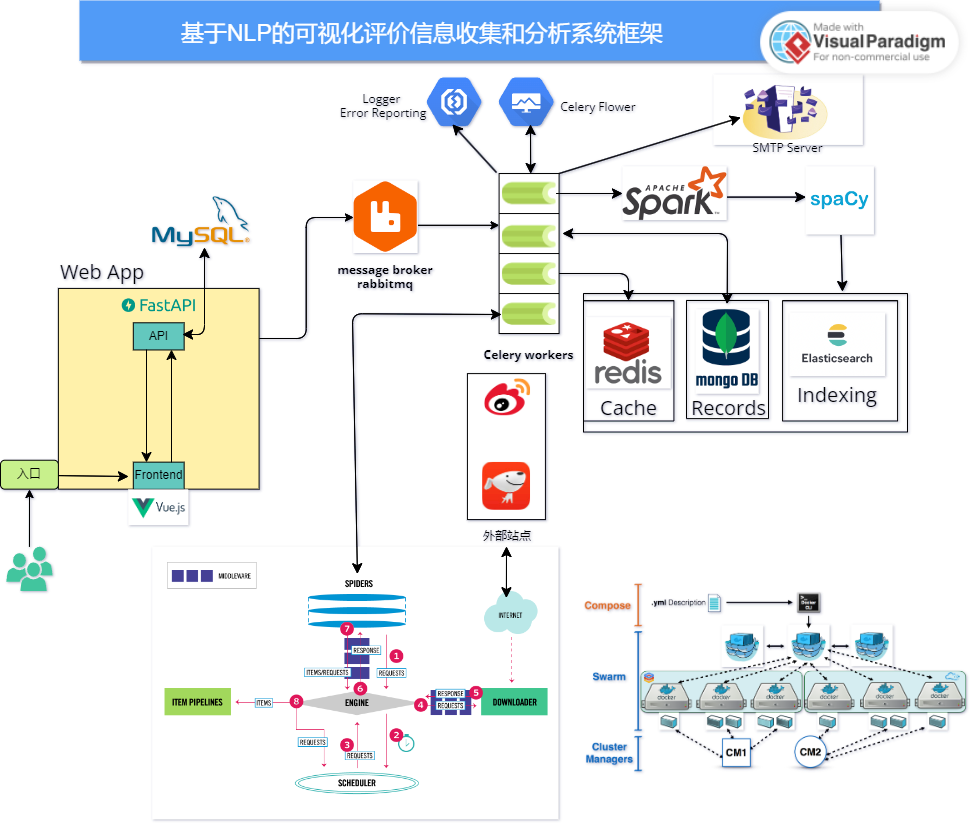


图3-2 系统架构设计图

基于NLP的可视化评估信息采集与分析系统采用了一系列不同的技术，包括FastAPI、Vue.js、Scrapy、Celery、RabbitMQ、Apache Spark、spaCy、Redis、MongoDB、Elasticsearch和Docker Swarm。系统的层次结构可以分解为若干层，每一层都有自己的一套功能和技术。

用户界面层：

用户界面层负责为用户提供与系统交互的基于网络的界面。它使用Vue.js 构建，并使用RESTful API 调用与后端API 层通信。

后端API 层：

后端API层提供了用户界面层与系统各种处理组件之间的接口。它使用FastAPI 构建，并利用Celery 和 RabbitMQ 进行任务管理和消息队列。

数据收集层：

数据收集层负责从各种来源收集评估数据，例如社交媒体或电商平台。它使用基于Python的网络爬虫框架Scrapy 实现，并利用RabbitMQ 进行消息队列。

数据处理层：

数据处理层负责使用自然语言处理和机器学习技术分析收集到的数据。它利用 spaCy 进行 NLP 处理，利用 Apache Spark 进行分布式处理。Redis用作缓存以提高性能，MongoDB用作存储处理后数据的主数据库。

数据可视化层：

数据可视化层负责以视觉上吸引人且易于理解的格式呈现分析后的数据。它利用 Elasticsearch 对处理过的数据进行实时搜索和索引，并使用 D3.js 等前端库生成可视化效果。

基础设施层：

基础设施层负责管理系统组件的部署和扩展。它使用容器编排工具 Docker Swarm 实现，在需要的情况下可以利用 AWS 和 GCP 等各种云服务进行托管。

系统各层之间的调用关系可以描述如下：

用户界面层调用后端API层发起处理任务，获取分析结果。

后端API层与数据采集层通信发起数据采集任务，与数据处理层通信发起数据处理任务。

数据处理层与数据可视化层通信以生成可视化效果，并与数据库层通信以存储处理后的数据。

基础架构层管理所有系统组件的部署和扩展。

实现技术类型：

在实现技术类型上，系统利用了多种开源工具和框架，如基于Python的Web框架（FastAPI、Flask）、网络爬虫框架（Scrapy）、任务队列系统（Celery、RabbitMQ）、大数据处理等框架（Apache Spark）、NLP 处理库（spaCy）、数据库管理系统（MongoDB、Redis、Elasticsearch）、前端库（Vue.js、D3.js）和容器编排工具（Docker Swarm）。

## 3.3 系统的模块功能划分

用例图（Use Case Diagram）是由软件需求分析到最终实现的第一步，它描述人们希望如何使用人们希望如何使用一个系统。用例图显示谁将是相关的用户、用户希望系统提供什么服务，以及用户需要为系统提供的服务，以便使系统的用户更容易地理解这些元素的用途，也便于软件开发人员最终实现这些元素。用例图在各种开发活动中被广泛地应用，但是它最常用来描述系统以及子系统。

接警人进行系统操作的用例图。接警人的操作有接警记录，出警安排，案件总结，案件查询。

图3-3 接警人接警处理用例图

普通用户参与的用例图。普通用户可以进行注册，案件查询，修改密码的工作。



图3-4 普通用户参与的用例图

## 3.4 系统的接口设计

用户身份验证：提供用户身份验证的API，接受用户的登录请求，检查用户提供的用户名和密码，验证用户身份的合法性。

数据采集和管理：提供数据采集和管理的API，包括数据的采集、分类、编辑和删除等操作，方便用户管理数据。

数据分析和处理：提供数据分析和处理的API，通过NLP技术对系统收集的数据进行分析和处理，并返回分析结果。

数据查询和搜索：提供数据查询和搜索的API，支持多条件搜索和数据的快速定位，方便用户找到所需的数据。

用户设置和管理：提供用户设置和管理的API，包括用户信息的修改等操作，方便用户管理个人信息。

## 3.5 系统涉及的数据结构

设计要点：

基于NLP的可视化评价信息采集与分析系统利用多种数据结构对评价数据进行存储和处理。这些包括：

原始数据结构：该数据结构存储从各种来源收集的原始评价数据，例如基于文本的评论。此数据结构中的每条记录都包含数据项，例如数据源、时间戳和原始格式的实际数据。

处理后的数据结构：该数据结构存储使用 NLP 和 ML 技术分析后处理过的视觉评估数据。该数据结构中的每条记录都包含数据项，例如数据的来源、时间戳和结构化格式的分析数据。

可视化数据结构：此数据结构存储用于根据处理后的数据生成可视化的数据。此数据结构中的每条记录包含数据项，例如可视化类型、数据范围以及可视化所基于的处理数据。

物理结构要点：

系统中使用的每个数据结构中的每个数据项都有特定的存储要求、访问方法和物理关系。例如,原始数据结构中的数据项可能存储在MongoDB等面向文档的数据库中，具有CRUD操作等访问方式。 访问单元可以在文档级别，通过唯一标识符访问单个记录。该数据结构的存储区域可能是特定的数据库服务器，设计时考虑了数据分区和索引。

类似地，处理后的数据结构中的数据项可能存储在与原始数据结构相同的数据库服务器中，但具有不同的访问方法，例如用于复杂分析操作的 SQL 查询。 访问单元可以在记录级别，通过唯一标识符访问单个记录。该数据结构的存储区域可能是 MongoDB 数据库中的一个特定集合，设计时考虑了数据建模和索引。

可视化数据结构可以存储在单独的数据库中，例如Elasticsearch，具有搜索查询等访问方法。访问单元可能处于可视化级别，通过唯一标识符访问各个可视化。该数据结构的存储区域可能是 Elasticsearch 数据库中的特定索引，设计考虑了数据建模和查询性能。

与程序的关系：

每个数据结构都由系统中的软件程序通过适当的API和数据访问库进行访问和操作。例如，系统的数据收集组件可以访问原始数据结构，而数据处理和分析组件可以访问处理后的数据结构。可视化数据结构可以被系统的可视化组件访问，其基于处理的数据生成可视化。可以通过标准数据库查询和 API 调用以及特定于系统的自定义数据访问功能来访问这些数据结构。

## 3.6 系统的错误处理

错误处理和错误信息输出：

基于NLP的视觉评价信息采集分析系统在数据采集、处理、可视化等各个运行阶段都可能遇到错误或故障。 为确保用户能够有效地识别和解决这些问题，系统应提供清晰且信息丰富的错误消息以响应每个可能的错误或故障。

以下是系统可能输出的错误消息示例：

数据收集错误：

表格：“错误：无法从 [来源] 收集数据。请检查您的互联网连接，然后重试。”

含义：由于连接问题或其他技术问题，系统无法从特定来源收集视觉评估数据。

处理方法：系统会向用户显示错误信息，提示用户检查网络连接或采取其他适当措施解决问题。

数据处理错误：

表格：“错误：无法处理数据。请重试或联系支持人员。”

含义：系统在使用 NLP 和 ML 技术处理视觉评估数据时遇到错误。

处理方式：系统会向用户显示错误信息，提示用户重新尝试分析或联系支持寻求进一步帮助。

可视化错误：

表格：“错误：无法生成可视化。请检查您的数据并重试。”

含义：由于数据或可视化设置存在问题，系统无法基于处理后的数据生成可视化。

处理方法：系统会向用户显示错误信息，提示用户检查数据和可视化设置后再试。

授权错误：

表单：“错误：用户名或密码无效。请重试。”

含义：用户在尝试登录系统时输入了错误的用户名或密码。

处理方法：系统会向用户显示错误信息，提示用户重新输入用户名和密码。

服务器错误：

表格：“错误：内部服务器错误。请联系支持人员。”

含义：系统遇到意外错误或问题，导致其无法正常运行。

处理方式：系统会向用户显示错误信息，提示用户联系客服寻求进一步帮助。

一般来说，错误消息应该清晰、简洁且信息丰富，为用户提供足够的信息来理解问题并采取适当的措施。 系统还应向用户提供有关他们为解决问题而采取的任何行动的反馈，例如在数据收集任务成功完成时显示成功消息。

# 4 系统的详细设计

详细设计的目标不仅仅是逻辑上正确地实现每个模块的功能，更重要的是设计出的处理过程应该尽可能简明易懂。对于交互式系统来说，人机界面设计和数据库设计、体系结构设计一样重要。近年来，人机界面在系统中所占的比例越来越大，在个别系统中人机界面的设计工作量甚至占总设计量的一半以上。系统界面的设计质量，直接影响用户对软件产品的评价，从而影响软件产品的竞争力和寿命，因此，必须对系统界面设计给予足够重视。

这里的详细地解释了“怎样做”的问题。

## 4.1 系统接口设计

在个别系统中人机界面的设计工作量甚至占总设计量的一半以上。系统界面的设计质量，直接影响用户对软件产品的评价，从而影响软件产品的竞争力和寿命，因此，必须对系统界面设计给予足够重视。

这里的详细地解释了“怎样做”的问题。

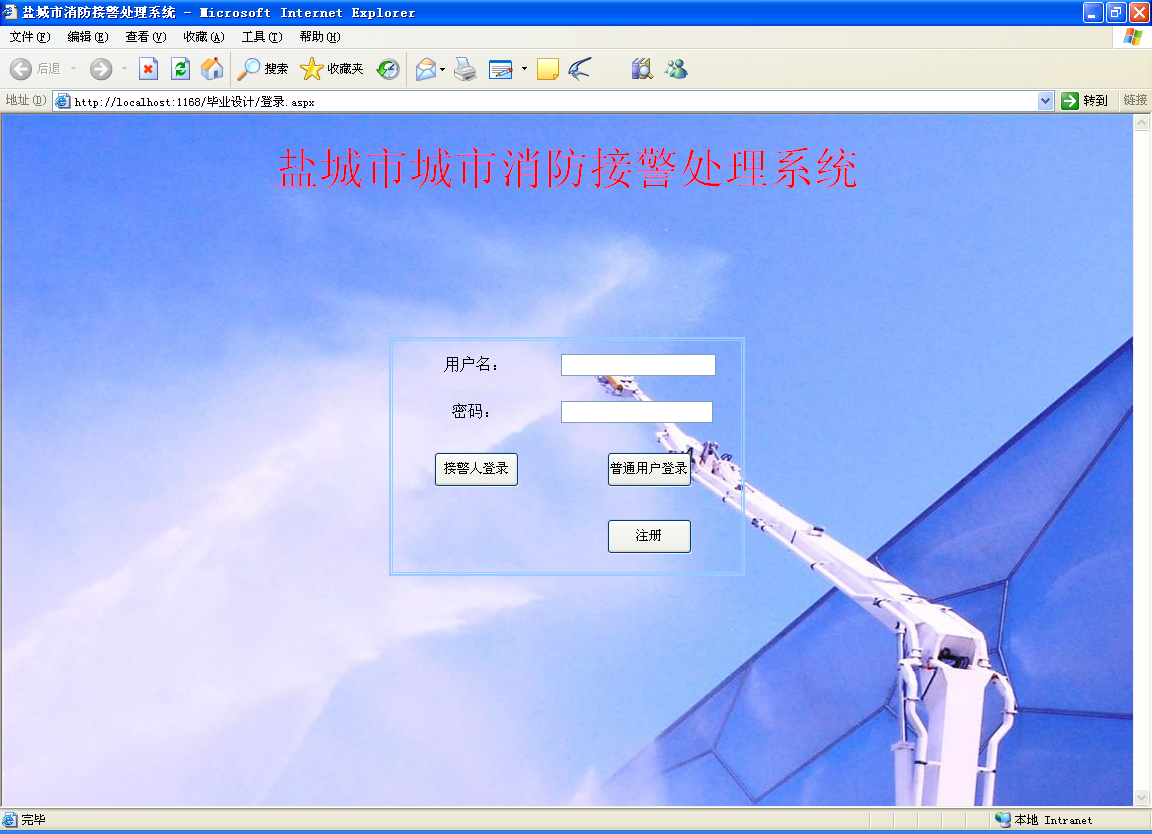
## 4.2 系统登录模块设计

系统登录模块分为用户登录模块和用户注册模块。

### 4.2.1 用户登录模块设计

用户登录模块包含了接警人登录和普通用户登录。接警人是系统的主要使用人，所以用户名和密码是系统赋予的，而普通用户可以看作系统的额外使用人，所以他们的用户名和密码是需要注册得到的。两种登录的用户采用不同的登录方式，用不同的按钮控制，所以登录界面上有两个登录按钮。还为了方便没有用户名的警员登录，还有一个注册按钮。如图4-1用户登录界面所示。

该界面可用于多用户的登录。

图4-1 用户登录界面

在整个登录界面的设计中，有两个Table控件。一个在头部分，中间插入一个Label控件，用来输入系统名称。另一个在中间部分，分几行几列。对应插入了两个Label控件，两个TextBox控件和三个Button控件。分别用来表示用户名和密码，用户名和密码的输入及接警人登录按钮、普通用户登录按钮、注册按钮。其中，用来输入密码的TextBox的TextMode属性要设置成password，这样输入密码的时候就可以不显示出来了。

Visual Studio 2005中设计部分分为设计和源，可以在设计中使用控件，在属性表中修改属性。也可以在源中填写代码。系统所用到的图片可以放在项目下的image文件夹中。界面的背景设置可以在设计中选中样式，在image中选择所需的图片。也可以在源中的body 后加background="image/200807141044577817222.jpg"，引号中是图片路径。字体的属性可以在Label中设置。Table中的控件基本上设置为居中，只要在td后加上 align="center"，也可以设置居左或者居右。

登录系统时，系统要与数据库连接。输入的用户名和密码要与数据库中相应的用户名和密码一致，否则会弹出警示对话框提醒错误。当用户名的输入为空时，也会弹出提示框，这时要重新输入。

用户登录作为系统的第一步，在登录界面中就使用数据控件SqlDataSource建立了数据源，在Web.Config中加入连接数据源语句，如下：

<add name="XFJJConnectionString"connectionString="Data Source=HH01;Initial Catalog=XFJJ;Persist Security Info=True;User ID=sa;Password=123456" providerName="System.Data.SqlClient" />

在登录界面的后台，写上数据库连接语句，再用SQL语句进行具体的数据验证。数据库连接语句如下：

SqlConnection conn = new SqlConnection(ConfigurationManager.ConnectionStrings["XFJJConnectionString"].

ConnectionString);

### 4.2.2 用户注册模块设计

新用户登录的时候要先注册，再登录。把注册按钮的PostBackUrl属性设置为“~/注册.aspx”，在登录界面点击注册按钮就能进入注册界面。

注册界面也是用Table控件来设计的，跟登录界面一样，也用了两个Table控件。一个在头部，用作系统名称的输入；一个在中间插入多个Label和TextBox控件，用作注册输入。注册界面的系统名称没有用Label控件输入，而是在Table中插入了一张背景图片，插入的方式与登录界面的背景插入一样。背景图片是PS过的，系统名称直接在图片中，作为图片的一部分出现。注册信息输入的Table中，Label控件同样作为要注册的信息名，TextBox控件用作信息输入。还有两个Button按钮，一个是注册提交按钮，一个是返回按钮。各个控件位置如登录界面中一样，可以设置为居左，居右，或者居中，还可以设置其他一些属性。在几个必填的信息后面还加入了RequiredFieldValidator验证控件，提示不能为空，加入CompareValidator验证控件，提示两次输入密码要一致。

注册用户名的时候，若遇到用户输入的用户名已被注册的情况。在提交按钮的后台代码中写入一段这样的代码。

yanzheng();if (falg)

Response.Write("<scrilanguage='javascript'>alert('该用户名已存在！

');location='注册.aspx';</script>");

yanzheng()函数如下：

protected void yanzheng()

{string sqlstr = "select user\_name from putongyonghubiao where user\_name='" +

TextBox1.Text + "'";

SqlCommand cmd = new SqlCommand(sqlstr, conn);

SqlDataReader dr = null;

try

{conn.Open();dr = cmd.ExecuteReader(CommandBehavior.CloseConnection);

}catch (SqlException ex)

{throw new Exception(ex.Message, ex);}

if (dr.Read())

{falg = true;}}

falg 的定义放在public partial class 注册 : System.Web.UI.Page中。

具体的注册界面如图4-2普通用户注册界面所示。



图4-2 普通用户注册界面

## 4.3 系统数据收集模块设计

接警人模块包含了接警记录模块，出警安排模块，案件总结模块和案件查询模块，是整个系统的主要功能模块。接警人模块中各个界面的创建都调用了一个母版页，具体界面的设计则是放在内容页中。这样的设计使得整个系统的界面设计变得简单，修改方便，大大节省了设计时间，也让系统的界面看起来更加的协调，美观。

接警人模块中调用的母版页的设计和前面的界面设计类似，在头部的Table控件插入带标题的图片，其他Table中设置了背景色。其他的就是一些Label控件和Button控件的设置。最后要插入内容页。

从登录界面登录的时候，接警人首先进入的是接警人主界面。所以，接警人模块的具体设计从接警人主界面开始。

### 4.2.1 接警人主界面设计

添加一个新项，名字为接警人主界面，调用设计好的母版页，然后在内容页中进行设计。接警人主界面作为登录和实现功能之间的缓冲，采用了一个div加css的设计，插入一个javascript的菜单脚本，用来展示一些介绍和说明。在每个菜单中加入一个TextBox控件，并将其的TextMode属性设置为MultiLine，在其中加入

一个Text="",引号中写入的就是可以展示的文本。Css代码放在了母版页中，div及javascript的代码就在接警人主界面的源中。部分代码如下：

<script type="text/javascript">

function $(id){return document.getElementById(id);}

function clickNav(n){

for(i=1;i<=4;i++){

$('boxNav'+i).className='normaltab';

$('boxContent'+i).className='nosee';

}

$('boxNav'+n).className='hovertab';

$('boxContent'+n).className='see';

} </script>

具体界面如图4-3接警人主界面所示。

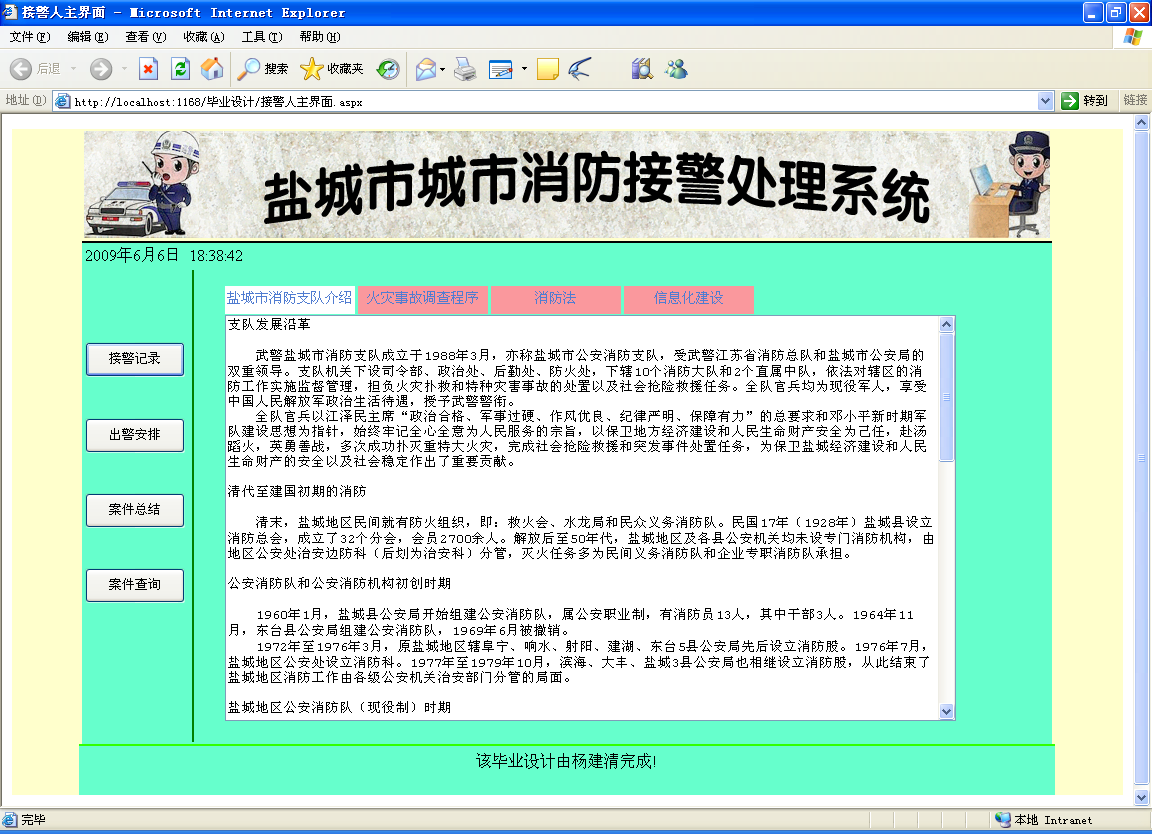


图4-3 接警人主界面

### 4.2.2 接警记录模块设计

接警记录模块的建立，也采用了母版页，所以母版页的内容不变，只是在内容页中设计。接警记录模块以记录为主，所以仍然采用Table控件，插入Label控件和TextBox控件，再加上Button控件。接警记录中有四项内容需要记录，分别是：

案件编号，案件类型，案件地点，接警时间。为了更快的出警，接警记录所要记录

的信息应该是对案件最具有相关性的。而且，作为案件的最初信息输入，不允许为空

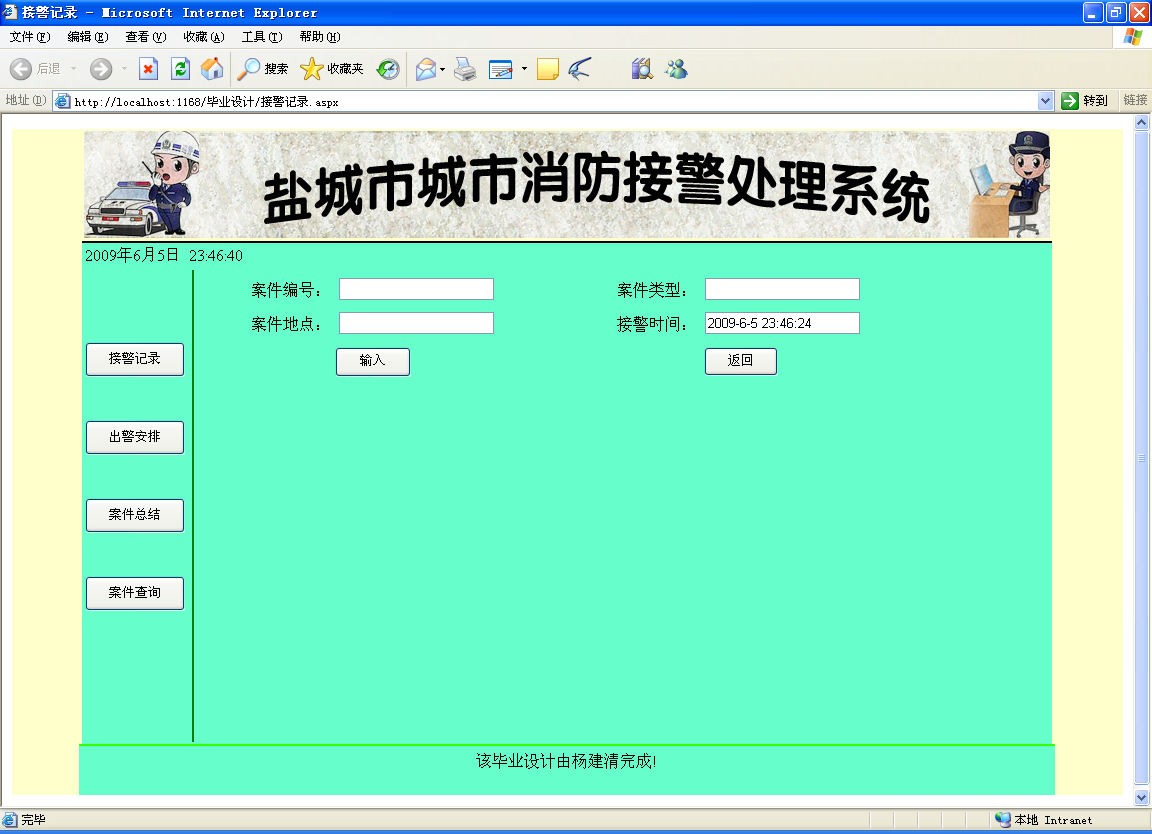


图4-4 接警记录界面

点击输入按钮后，完成记录，会在下方将表中的信息显示。主要是使用了

### 4.2.3 出警安排模块设计

的PostBackUrl属性设置到需直接跳转的界面。具体界面如图4-5出警安排界面所示。

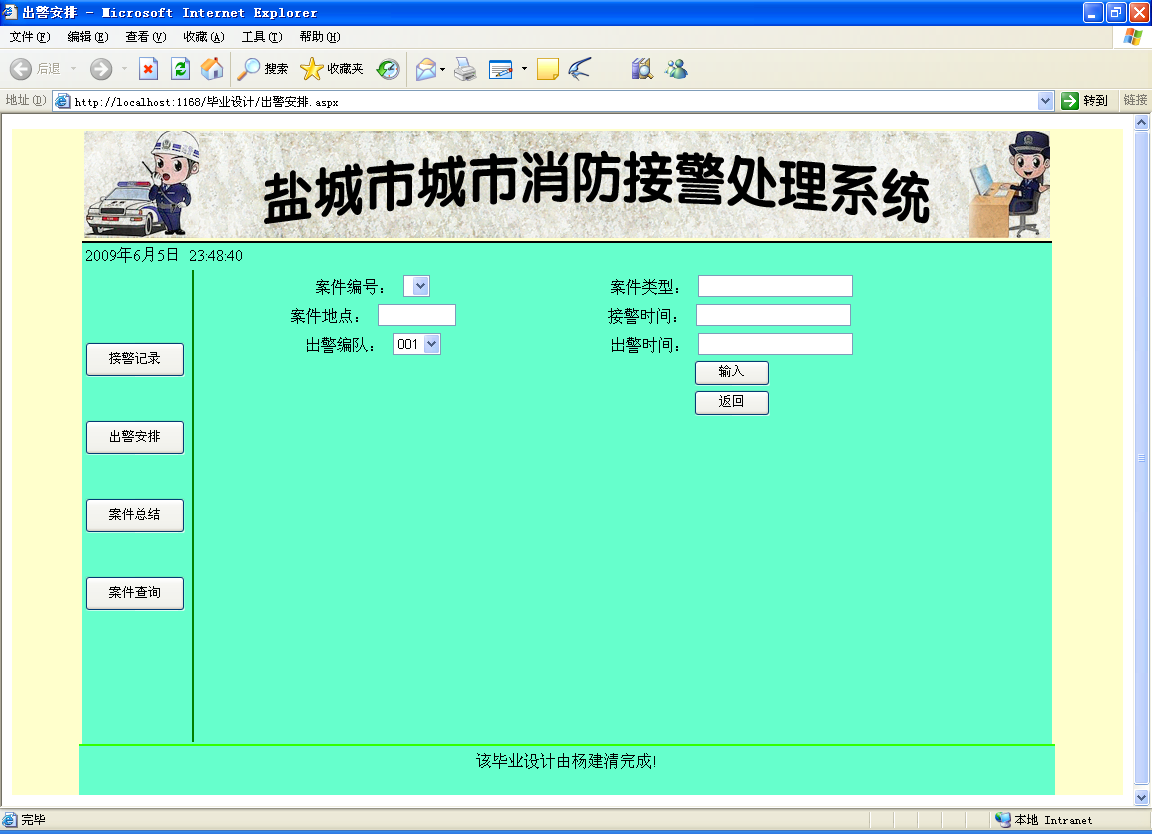


图4-5 出警安排界面

如图4-5所示，内容页中有6个要输入的信息，案件编号和出警编队的信息输入

## 4.3 普通用户模块设计

普通用户模块只包含案件查询和修改密码两个模块。普通用户模块和接警人模块一样，也是采用母版页的设计方式，两个母版页也是相似的。而且，普通用户主界面的内容页与接警人主界面的内容页一样。两种案件查询也大体相似，只是普通

用户的案件查询比接警人的案件查询少一种查询方式。

### 4.3.1 修改密码模块的设计

用户登录后才可以修改密码，所以采用了旧密码，新密码和重复密码的修改密码方式，用验证控件验证，相关代码如下：

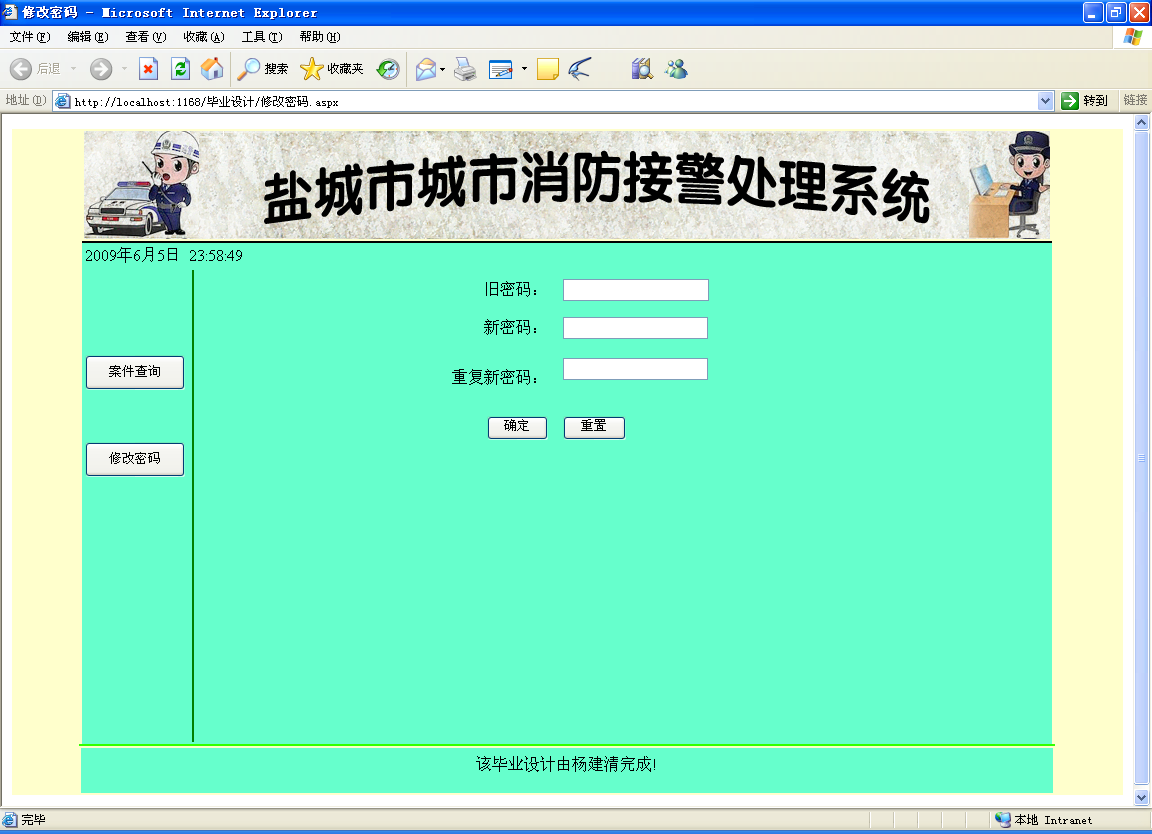
string cmdstr = "update putongyonghubiao set password=" + "'" + xinmima + "'" + "where user\_name=" + "'" + name + "'";

SqlCommand cmd = new SqlCommand(cmdstr, conn);

conn.Open();

cmd.ExecuteNonQuery();

具体界面如图4-7普通用户修改密码界面所示。



4-7 普通用户修改密码界面

# 5 系统实现

## 5.1 系统接口的实现

## 5.2 系统数据收集模块的实现

## 5.3 系统数据展示模块的实现

## 5.4 系统数据分析可视化模块的实现

## 5.5 系统通知模块的实现

## 5.6 系统Docker部署的实现

# 6 系统部署与测试

## 6.1 系统部署

## 6.2 系统登录模块测试

## 6.3 系统数据收集模块测试

## 6.4 系统数据展示模块测试

## 6.5 系统数据分析可视化模块测试

## 6.6 系统通知模块测试

数据分析和挖掘：随着数据规模的不断增大和数据分析需求的不断增加，数据分析和挖掘的市场需求也在不断扩大。该系统可以为用户提供更丰富的数据分析和挖。

# 7 系统的发展方向及应用前景

数据分析和挖掘：随着数据规模的不断增大和数据分析需求的不断增加，数据分析和挖掘的市场需求也在不断扩大。该系统可以为用户提供更丰富的数据分析和挖掘功能，满足用户不同的分析需求，并可以应用于多个领域，如电商、金融、医疗等。

基于大数据的智能应用：随着人工智能技术的不断发展和应用，基于大数据的智能应用也将越来越受到关注。该系统可以为用户提供智能化的数据分析和挖掘功能，帮助用户更好地理解和利用评价信息。

大规模数据处理：随着数据规模的不断增大和数据处理需求的不断增加，大规模数据处理技术的市场需求也在不断扩大。该系统可以使用 Apache Spark 等大规模数据处理技术来处理评价信息和其他大规模数据，提高数据处理的效率和性能。

多模态数据分析：除了文本数据，还有图像、音频、视频等多种数据类型需要分析和处理。该系统可以扩展到多模态数据分析和处理领域，为用户提供更多样化的数据分析和挖掘功能。

# 8 结束语

经过这几个月的毕业设计的学习，我收获了很多。我充分体会到了一个系统的完成是需要不断的学习，不断的完善。经过这次对盐城市消防接警处理系统的设计，充分利用了软件工程的思想，使得我对大学四年所学到的知识有了实践性的提高。

通过这次毕业设计，我得出了以下几点体会：

a) 对于软件专业的学生来说，在实践中学习远比抱着书本死读书要收获的多的多。要在做的过程中，不断尝试，不断实践。虽然会遇到很多的难题，但是你每解决一个难题，就收获了一个知识点，遇到的困难越多，解决的越多，进步的就越快。

b) 对于一个软件的设计，一定要遵循软件工程的步骤。按正确的步骤，不仅能使软件的实现更容易，也是实现步骤更清晰。

c) 编写代码不仅要认真仔细，还要遵循代码编写原则，比如：

代码缩进。一般遵循的原则是，使用流程控制语句IF…ENDIF、SELECT CASE、 FOR…NEXT 、DO…LOOP时，流程控制语句之中的程序代码部分要用[Tab]键进行一次缩进，遇到嵌套时，亦遵循此规则。

代码换行。一行的程序代码如果太长，那么就需要换行以增加其可读性。换行时应尽量按照程序执行的逻辑进行换行，遇到标点时标点应放在前一行的末尾处，且所下移的行要尽量缩进。

代码模块化。对代码进行模块化处理，可以提高代码的复用性以及提高开发速度。代码的模块化处理，包括将需要进行复用的代码编写成一个子程序或者函数，或者一个类，以便使用时进行调用。将代码太长的函数分解成几个函数的调用，这样都会大大提高程序的可读性和可维护性。

总的来说，由于水平还不高，以及毕业设计时间较短，该系统只是基本达到毕业设计的要求，还有许多不尽如人意的地方，比如数据库的设计欠缺，用户界面不够美观，自动化程度不够等多方面问题。这些都有待进一步改善。

通过这次毕业设计，我发现自己的知识还很不够，还不能完全适应当前的需求。在今后的学习和工作中，我将努力弥补，不断地扩充各方面的知识，吸取失败的教训。通过不断的学习和改进，希望能够进一步提高自己的水平。

# 参考文献

[1] 张海藩.软件工程导论（第四版）[M].北京：清华大学出版社，2003.

[2] 周涛,吕伟臣,廉颖.SQL Server 2005 数据库企业级应用[M].北京：清华大学出版社，2008.4.

[3] 王敬栋,施伟.ASP.NET 2.0数据库通用模块开发与系统移植[M].北京：清华大学出版社，2007.5.

[4] 龙马工作室.Visual C# 2005 从入门到精通[M].北京：人民邮电出版社，2008.4.

[5] 李德奇.ASP.NET程序设计[M].北京：人民邮电出版社，2007.9.

[6] 启明工作室.ASP.NET+SQL Server网络应用系统开发与实例[M].北京：人民邮电出版社，2005.2.

[7] 蔡敏,徐慧慧,黄炳强.UML基础与Rose建模教程[M].北京：人民邮电出版社，2006.1.

[8] 郑霞,赵辉,徐慧.ASP.NET 2.0编程技术与实例[M].北京：人民邮电出版社，2007.7.

[9] 杨云,王毅.ASP.NET 2.0典型项目开发[M].北京：人民邮电出版社，2007.10.

[10]郑阿奇.SQL Server实用教程[M].北京：电子工业出版社，2005.7.

[11]ASP.NET页面事件：顺序与回传详解[DB/OL].www.csdn.net，2008.4.

[12]胡吉林,姜波,高丙朋.[存储过程在ASP．NET数据库系统开发中的应用](http://mag.shudoo.com/qk/96730X/200812/28859873.html)[J].电脑开发与应用,2008，（12）：66-68.

(>=10个参考文献，近3年的。书写按照我PPT中给的格式标准。)

# 致 谢

千言万语，不如一笺谢意。

# 附 录

# **附录1 源程序清单**

## 附录1.1 系统登录模块

### 附录1.1.1 用户登录模块

using System;

using System.Data;

### 附录1.1.2 普通用户注册模块

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

## 附录1.2 接警人模块

### 附录1.2.1 接警记录模块

using System;

using System.Data;

### 附录1.2.2 出警安排模块

using System;

using System.Data;

using System.Configuration;

using System.Collections;

using System.Web;

using System.Web.Security;

### 附录1.2.3 案件总结模块

using System;

using System.Data;

using System.Configuration;

### 附录1.2.4 案件查询模块

ntPlaceHolder1" Runat="Server">

<table style="width: 736px; height: 150px">

## 附录1.3 普通用户模块

### 附录1.3.1 普通用户案件查询模块

using System;

### 附录1.3.2 普通用户修改密码模块