# 项目开发总结报告

目录

[项目开发总结报告 1](#_Toc124597599)

[1.引言 3](#_Toc124597600)

[1.1编写目的 3](#_Toc124597601)

[1.2项目背景 3](#_Toc124597602)

[1.3定义 3](#_Toc124597603)

[1.4参考资料 3](#_Toc124597604)

[2.开发结果 4](#_Toc124597605)

[2.1产品 4](#_Toc124597606)

[2.2主要功能及性能 6](#_Toc124597607)

[2.3所用工时 7](#_Toc124597608)

[2.4所用机时 7](#_Toc124597609)

[2.5进度 7](#_Toc124597610)

[2.6费用 7](#_Toc124597611)

[3.评价 9](#_Toc124597612)

[3.1生产率评价 9](#_Toc124597613)

[3.2技术方案评价 9](#_Toc124597614)

[3.3产品质量评价 9](#_Toc124597615)

[4.经验与教训 10](#_Toc124597616)

## 1.引言

### 1.1编写目的

现今，一般学校通过点名的方式对学生进行考勤。但随着学生人数的增多，这种考勤系统显得工作量很大，并且经常造成错误。很多学校迫切需要种新的考勤系统来提高工作效率，减少考勤开支，增加考勤灵活性。

### 1.2项目背景

（1）本系统全称为“人脸识别考勤系统”

（2）本项目由张小勤老师提出，由学生组成的小组自行开发。用户为需要对学生进行考勤的学校

（3）本系统会与存有学生信息，教师信息，课程信息，考勤信息的数据库进行交互，也会与教师提供的学生考勤情况进行交互

### 1.3定义

暂无

### 1.4参考资料

1. 项目的计划任务书，合同或批文；
2. 项目开发计划；
3. 需求规格说明书
4. 概要设计说明书；
5. 详细设计说明书；
6. 用户操作手册;

[1]王权汉.基于人脸识别的考勤系统设计与实现[J].电子技术与软件工程,2022,(19):167-170.

## 2.开发结果

### 2.1产品

各部分的程序名称包括由java编写的权限系统与基础信息管理系统和由python编写的考勤系统。

**下列1.1-1.5是java文件**

1.1controller层下的servlet文件：

ACourseServlet共135行，用于对课程进行增删改查。

AFunctionServlet共131行，用于对功能进行增删改查。

ARoleFuncServlet共130行，用于对角色-功能联系增删改查。

ARoleServlet共132行，用于对角色进行增删改查。

AttendanceServlet共149行，用于新建、删除一个考勤表、修改特定学生的考勤信息、查询已到和未到人数。

AUserRoleServlet共130行，用于对用户-角色联系增删改查。

AUserServlet共291行，用于学生注册和对用户、学生用户、教师用户。增删改查。

LoginServlet共47行，用于登录。

1.2dao层及对应的实现层：

CourseDaoImpl共154行，执行sql语句，用于对课程进行增删改查。

FunctionDaoImpl共154行，执行sql语句，用于对功能进行增删改查。

RoleFuncDaoImpl共148行，执行sql语句，用于对角色-功能联系增删改查。

RoleDaoImpl共159行，执行sql语句，用于对角色进行增删改查。

AttendanceDaoImpl共159行，执行sql语句，用于新建、删除一个考勤表、修改特定学生的考勤信息、查询已到和未到人数。

UserRoleDaoImpl共148行，执行sql语句，用于对用户-角色联系增删改查。

UserDaoImpl共335行，执行sql语句，用于学生注册和对用户、学生用户、教师用户。增删改查。

1.3service层及对应的实现层：

CourseServiceImpl共24行，调用dao层的函数，用于对课程进行增删改查。

FunctionServiceImpl共27行，调用dao层的函数，用于对功能进行增删改查。

RoleFuncServiceImpl共27行，调用dao层的函数，用于对角色-功能联系增删改查。

RoleServiceImpl共27行，调用dao层的函数，用于对角色进行增删改查。

AttendanceServiceImpl共28行，调用dao层的函数，用于新建、删除一个考勤表、修改特定学生的考勤信息、查询已到和未到人数。

UserRoleServiceImpl共27行，调用dao层的函数，用于对用户-角色联系增删改查。

UserServiceImpl共35行，调用dao层的函数，用于学生注册和对用户、学生用户、教师用户。增删改查。

1.4entity（model）层：

Course共64行，确定课程的成员并定义构造函数，获取和更新成员的函数。

Function共45行，确定功能的成员并定义构造函数，获取和更新成员的函数。

RoleFunc共47行，确定角色-功能的成员并定义构造函数，获取和更新成员的函数。

Role共45行，确定角色的成员并定义构造函数，获取和更新成员的函数。

Attendance共54行，确定考勤表的成员并定义构造函数，获取和更新成员的函数。

UserRole共48行，确定用户-角色的成员并定义构造函数，获取和更新成员的函数。

User共82行，确定用户的成员并定义构造函数，获取和更新成员的函数。

1.5DBUtil：

连接数据库，定义了查询、更新数据库的具体函数

**下面是jsp文件：**

Login.jsp共109行，包含学生注册、登录等内容。

Index.jsp共28行，包括考勤系统以外的布局。

Top.jsp共39行，单纯用于显示。

Left.jsp共340行，罗列了系统所有的功能。

Right.html共40行，除考勤拍照功能外的所有功能在left.jsp中被点击后就会在这个页面显示。

StudentReigister.jsp用户注册个人信息，默认赋予学生角色。

接下来是left.jsp下的subtree会跳转的页面：

UserRegister.jsp页面用于添加用户

UserList.jsp显示所有用户，也可以根据用户名模糊查询

UserUpdate.jsp用于修改选中的用户的个人信息

StudentList.jsp显示所有角色为学生的用户

StudentUpdate.jsp用于修改选中的学生用户的个人信息

TeacherList.jsp显示所有角色为教师的用户

TeacherUpdate.jsp用于修改选中的教师用户的个人信息。

UserRoleRegister.jsp添加用户-角色联系

UserRoleList.jsp显示所有用户-角色联系，也可按联系号查询，UserRoleUpdate.jsp修改用户-角色联系信息。

RoleRegister.jsp添加角色

RoleList.jsp显示所有角色，也可按角色名查询

RoleUpdate.jsp修改角色信息。

RoleFuncRegister.jsp添加角色-功能联系

RoleFuncList.jsp显示所有角色-功能联系，也可按联系号查询，RoleFuncUpdate.jsp修改角色-功能联系信息。

FunctionRegister.jsp添加功能

FunctionList.jsp显示所有功能，也可按功能名查询

FunctionUpdate.jsp修改功能信息。

CourseRegister.jsp添加课程

CourseList.jsp显示所有课程，也可按课程名查询

CourseUpdate.jsp修改课程信息。

AttendanceRegister.jsp有新建包含所有学生的考勤表和删除现有考勤表，AttendanceList.jsp显示所有考勤信息，也可按已到/未到查询，AttendanceUpdate.jsp修改考勤信息。

**下列为python文件：**

Train.py共63行,主要功能是对模型进行训练

preprocess.py共56行,主要功能是对人脸照片进行预处理

postprocess.py共28行，功能是对照片预测模型进行分开输出，验证结果后处理

inference.py共33行，功能是创建一些预测器供后续使用

upload.py共48行，功能是将upload文件夹中指定图片经过人脸检测后存入img文件夹中

facecapture.py共68行，功能是电脑摄像头进行实时拍摄后，经过人脸检测后存入img文件夹中，命名为temp.jpg

facerecognize.py共155行，功能是对img中temp.jpg与其他照片进行人脸识别。

上述程序整合进入Django框架的view.py文件中，共456行

url.py共14行，记录了Django框架的所有url地址

**下列为javascript文件：**

Ajax.js共100行，用来进行前端与后端的通信

### 2.2主要功能及性能

每个产品功能已经在2.1中进行过详细的阐述。

**以下是产品的相关功能总结**

人脸采集：通过电脑摄像头获取考勤人员的人脸信息。

人脸比对：用户点击考勤按钮之后，对人脸进行采集之后，与服务器数据库中学生存储的人脸信息进行比对，并返回学生姓名。

记录数据：确定学生身份后，并记录下学生考勤情况，并通过软件处理考勤形成报表。

课程信息管理：教务处管理员通过系统录入课程信息，包括课程名，学分等。

学生信息管理：学生注册为系统的用户，由教务处人员为其分配相应的权限。

教师信息管理：教师注册为系统的用户，由教务处人员为其分配学生信息管理权限。

**以下是产品的相关性能总结**

速度：点击人脸考勤按钮之后，系统会在2s内识别出人脸信息，并录入考勤信息数据库。

容量：系统可以储存至少10万个学生的信息。

方便性：每个人的人脸都是唯一的，不会出现学生忘记带学生证找老师进行补签的情况

高效性：每个人可以在光线较暗的情况下进行人脸识别比对，降低了对环境的要求。

### 2.3所用工时

所用工时：权限系统和信息管理系统初步开发用了两天，后续系统功能优化和与考勤系统对接用了五天。

人脸识别考勤功能开发用了五天，其中三天时间用来开发人脸识别算法相关，两天时间进行人脸识别算法与网页的连接，主要采用Django框架，Ajax技术。

从整个系统开发到部署至华为云上一共10天。

### 2.4所用机时

使用Lenovo82LM型号系统开发了20个小时

使用OMEN by HP Laptop 15-dc1xxx型号系统开发了50个小时

### 2.5进度

计划在8号完成权限系统和信息管理系统的初步开发，实际进度在7号完成。

计划在11号完成权限系统、信息管理系统和考勤系统的对接，实际在11号完成。

计划在12号完成权限系统、信息管理系统的界面优化，实际在14号完成。

### 2.6费用

按照人脸识别算法工程师平均工资收入每月29.2k来计算，本系统开发周期为10天，所以产生相关费用约为9k。

需求工程分析师平均工资收入每月15k, 所以产生相关费用约为5k。

测试分析工程师平均工资收入每月15k, 所以产生相关费用约为5k。

华为云部署产生相关花费约200元

包括采购、开发和安装下列各项所需的费用，如:学校提供计算机设施，计算机重新加8台以及原电脑更新总计1,000元，云服务器总计1,600元，数据通讯设备1000元，安全与保密设备5,000元，操作系统，应用软件和数据库管理软件5,000元。所以总基建投资需要13,600元

免费维护一年，后期如果还需要维护，每月系统维护一次，每次1,000元。

对服务器定期更新和维护，1,000 元一个月。

宽带网络1,000元一年

2个管理员每月1,000元一个

## 3.评价

### 3.1生产率评价

每人每月生产的源程序行数约为：10000\*3/8=3750行

代码行数保持在100行/天以上的人均水平，生产率非常可观。

总文档字数约为60000字，每人每月生产的源程序行数约为7500行，同样十分可观

### 3.2技术方案评价

Jsp+java+tomcat开发权限和信息管理系统

Python+Django开发考勤系统

Django 是一个开放源代码的 Web 应用框架，由 Python 写成。

Django 遵守 BSD 版权，初次发布于 2005 年 7 月, 并于 2008 年 9 月发布了第一个正式版本 1.0 。

Django 采用了 MVT 的软件设计模式，即模型（Model），视图（View）和模板（Template）。

### 3.3产品质量评价

经过测试计划，本系统产品质量有如下保证：

（1）可执行软件与需求规格说明书、设计说明书是一致的；

（2）测试覆盖率应达到100%（注：对于白盒测试，覆盖率指的是代码覆盖率，对应的是函数覆盖率；对于黑盒测试，覆盖率指得是需求覆盖率）

（3）测试用例通过率要达到95%；

（4）软件缺陷终结率达到100%

（5）系统页面风格符合规范化要求，程序代码编写以及各种命名符合规范化要求。

（6）各模块正确衔接。

（7）对异常数据应有相应的提示信息，并能安全终止异常操作。

但是，系统界面不够美观，有待改进，在一些模块中用户操作不够人性化。

## 4.经验与教训

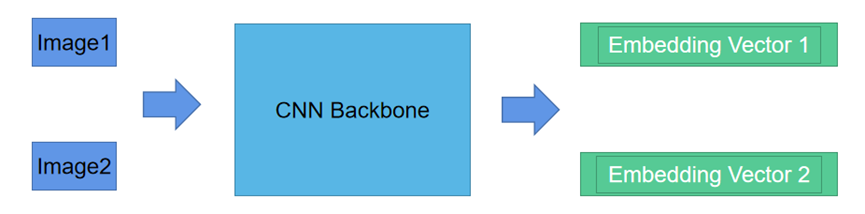
1. 权限、信息系统和考勤系统对接时的经验教训，在页面分框架的情况下，超链接跳转到另一个架构写的web时，target需要改为parent，以此实现整个页面的跳转变化。

2.本系统人脸识别主要采用python进行编写，采用了paddle这个包来进行模型的训练与调用，使用了cuda，导致部署的时候较难部署，可能会出现相关问题

3.对于每张照片，人脸的位置是不确定的，因此无法准确提取出人脸信息。因此通过深层的卷积神经网络(CNN)搭配目标检测结构的模型来实现，具体的技术细节这里就先不展开了。本实验采用飞浆平台中的PaddleHub包，其中已经包含了不少人脸检测的预训练模型，可以直接使用来进行简单的人脸检测。

4.人脸验证任务如果采用直接分类，分辨是准确的哪一个人，输出其标签会导致当模型训练完成后无法随时加入新的人。因此将其转换为一个二分类问题，即分辨两张人脸照片组成的图片对中的人脸是否来自同一个人，输出为同一个人的置信度。

一般使用孪生网络（Siamese Network）实现。



原理：

通过同一个CNN网络将人脸图片进行相同的编码，嵌入一个高维的向量空间

使用softmax loss作为损失函数直接对两个样本嵌入向量的拼接做二分类训练，使模型能够直接输出两个样本之间的相似度，当相似度达到一定的阈值后即判断为同一个人的人脸

或者使用triplet loss、contrastive loss、center loss等损失函数对CNN网络进行优化，使其最后编码的高维的向量空间中，同类样本的距离缩小，不同类样本的距离扩大

进而通过两个样本嵌入向量的距离来衡量两个样本之间的相似度，同样当相似度达到一定的阈值后即判断为同一个人的人脸

5.html的file上传标签只能接收到文件名字等相关信息，file 无法获得文件绝对路径，因此上传照片只能从upload文件夹中进行上传。

6.一开始在view.py中采用了import来调用其他模块，但是一个模块不会重复载入。多个不同的模块都可以用 import 引入同一个模块到自己的 Local 名字空间，其实背后的 PyModuleObject 对象只有一个。所以后续改为了函数调用。