****

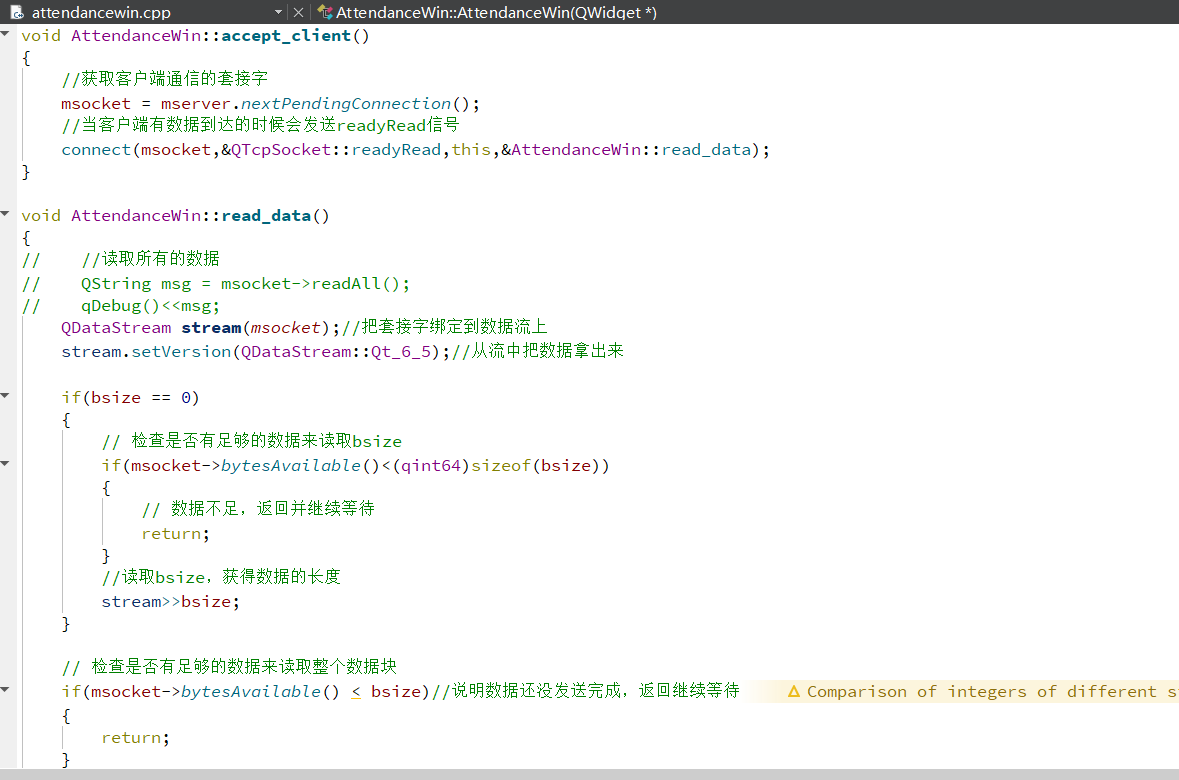
**课题名称：基于Qt框架的人脸识别门禁系统**

**系统界面：**

**信息管理平台：**

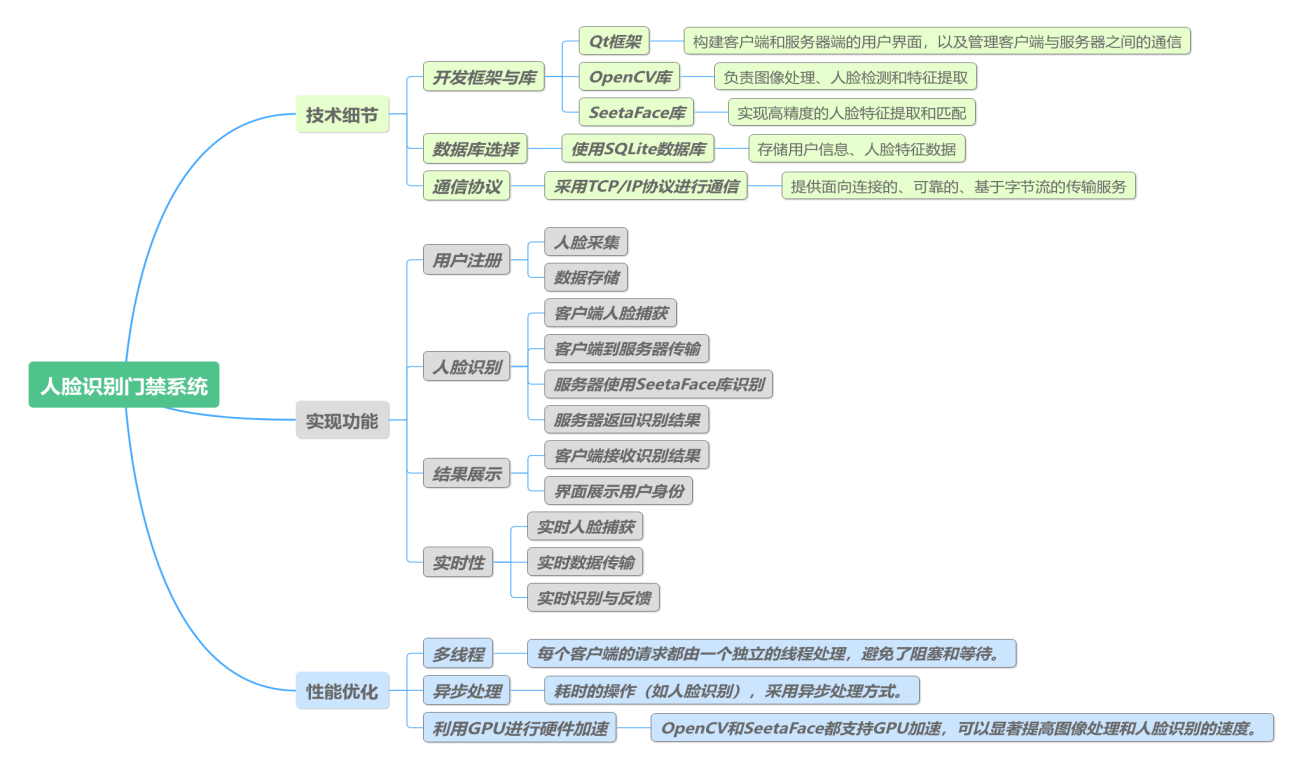
****

**C++(Qt＋SQLite)后台**



# 项目概述

本次实训旨在开发一个基于Qt框架的宿舍人脸识别门禁系统，通过人脸识别技术提高门禁管理的安全性和效率。实训周期为五周，涵盖了从**系统需求分析**、**环境搭建**、**功能实现**到**系统测试和优化**的全过程。



# 项目任务

表1：任务分解列表

|  |  |  |  |
| --- | --- | --- | --- |
| **任务分解** | **任务描述** | **达到指标** | **时间安排** |
| 任务一：系统设计 | 系统需求分析 | 完成开题报告、课程报告相应字段 | 第一周（5h） |
| 任务二：环境搭建 | 搭建Qt开发环境、完成opencv与seetaface库源码编译与配置。 | 环境搭建完成，可以在Qt项目中进行两个库的使用 | 第一周（5h） |
| 任务三：学习关键技术 | 学习使用opencv与seetaface库的功能 | 掌握OpenCV和SeetaFace基本使用 | 第一周  （3h） |
| 任务四：GUI界面 | GUI设计 | Qt的初步GUI界面完成 | 第二周、第三周（2h，2h） |
| 任务五：摄像头功能实现 | 在客户端实现摄像头功能 | 在客户端可以成功进行人脸采集 | 第三周  （2h） |
| 任务六：通信模块实现 | 实现基于TCP/IP协议的网络通信功能 | 客户端与服务器端能成功建立连接并相互发送数据 | 第二周、第三周  （4h，4h） |
| 任务七：数据库设计 | 设计、创建信息表 | 学生个人信息表与刷脸信息表被成功创建 | 第二周  （3h） |
| 任务八：功能整合 | 整合客户端和服务器端的功能 | 整合成功，完成了人脸识别流程 | 第四周  （3h） |
| 任务九：信息管理功能 | 实现信息的存储与查询功能 | 管理员可成功进行学生信息与刷脸信息的查询 | 第四周  （3h） |
| 任务十：性能优化 | 优化系统性能，提升识别准确率和实时性 | 系统人脸识别准确率和识别性得到提高 | 第五周  （4h） |
| 任务十一：系统界面完善 | 对用户界面进行调整与修改 | 系统界面与用户交互体验得到完善 | 第五周  （2h） |
| 任务十二：代码整理 | 整理代码，为实训报告编写做准备 | 客户端与服务器端代码整理完成 | 第五周  （2h） |

# 软件需求分析

本次实训旨在开发一个基于Qt框架的宿舍人脸识别门禁系统。系统主要包括两个部分：**服务器端**和**客户端**。该系统需求如下：

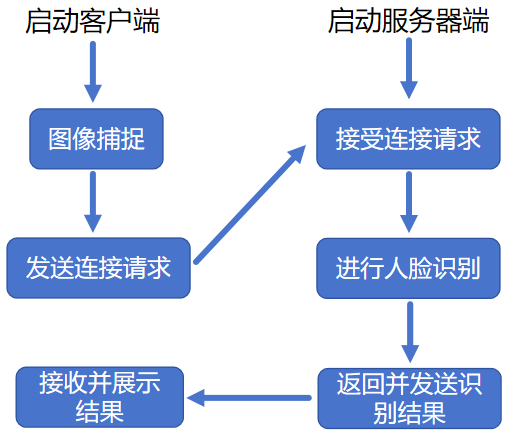
* **服务器端需求：**
  + **学生注册管理：**支持管理员录入学生信息，采集人脸图片。
  + **人脸识别：**接收客户端发送的人脸图片，利用OpenCV和SeetaFace库进行识别，并返回结果。
  + **刷脸信息管理：**记录每次刷脸信息，提供查询和管理功能。
* **客户端需求：**
  + **人脸识别验证：**实时采集学生的人脸图片，并发送给服务器进行识别。
  + **结果展示：**展示服务器返回的识别结果。

# 软件设计

**1、系统设计的详细描述**

* **系统架构：**采用客户端-服务器端模型，分为客户端和服务器端两个主要部分。
* **客户端设计：**
  + **用户界面：**基于Qt Widgets，提供清晰的操作界面，包括注册、刷脸验证等功能。用户可以通过这个界面与系统交互。
  + **摄像头模块：**客户端实现了摄像头模块，负责实时捕捉学生的人脸图像。
  + **数据处理：**使用OpenCV和SeetaFace库对采集的图像进行处理。
  + **通信模块：**客户端通过TCP/IP协议与服务器端进行通信，通过QTcpSocket实现与服务器的数据交换，发送人脸图像并接收识别结果。
* **服务器端设计：**
  + **注册管理模块：**提供管理员界面，用于输入学生信息和采集、上传人脸图像。
  + **人脸识别处理模块：**服务器端集成了OpenCV和SeetaFace库，负责接收客户端图像，执行人脸识别操作，并返回结果。
  + **数据库管理模块：**使用数据库存储学生信息和刷脸记录。
* **数据流设计**
  + 当学生在客户端进行刷脸时，摄像头模块捕捉到人脸图像，并通过通信模块发送到服务器。
  + 服务器端的人脸识别模块接收图像，利用人脸识别技术进行处理，并将识别结果存储在数据库中。
  + 识别结果随后通过服务器的通信模块发送回客户端，并在用户界面上展示给学生。

**2、系统的处理流程图**



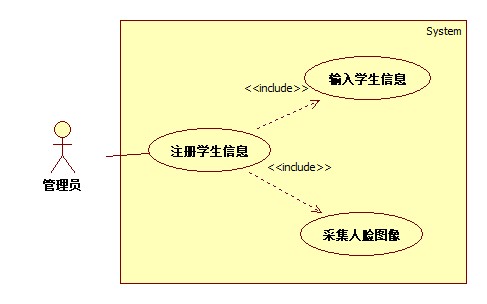
**3、系统用例图**

**宿舍人脸识别门禁系统的主要用例：**

1. **管理员注册学生信息**

* **主要参与者：**管理员
* **目标：**在系统中创建新的学生记录，并采集学生的人脸图像。
* **触发条件：**管理员需要添加新学生到门禁系统。
* **主要成功场景：**
  + 管理员登录到服务器端。
  + 管理员输入学生的姓名、学院、宿舍号等信息。
  + 管理员通过文件选择或摄像头拍照采集学生的人脸图像。
  + 系统保存学生信息和人脸图像，并确认注册成功。

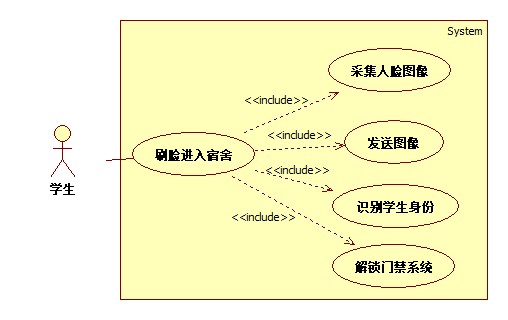
用例图如下：



1. **学生刷脸进入宿舍**

* **主要参与者：**学生
* **目标：**通过人脸识别验证学生身份，允许进入宿舍。
* **触发条件：**学生试图进入宿舍。
* **主要成功场景：**
  + 学生站在门禁系统前。
  + 客户端摄像头实时采集学生的人脸图像。
  + 图像被发送到服务器进行识别。
  + 服务器端确认学生身份，并返回识别结果。
  + 门禁系统解锁，允许学生进入。

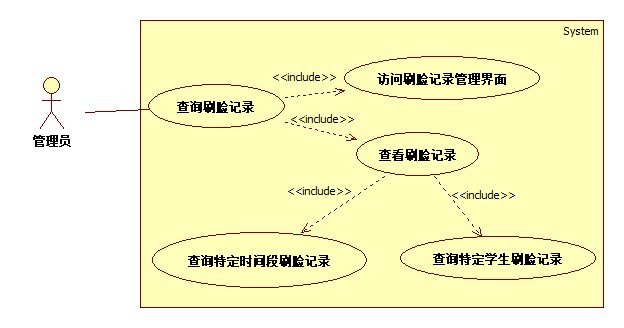
用例图如下：

****

1. **查询刷脸记录**

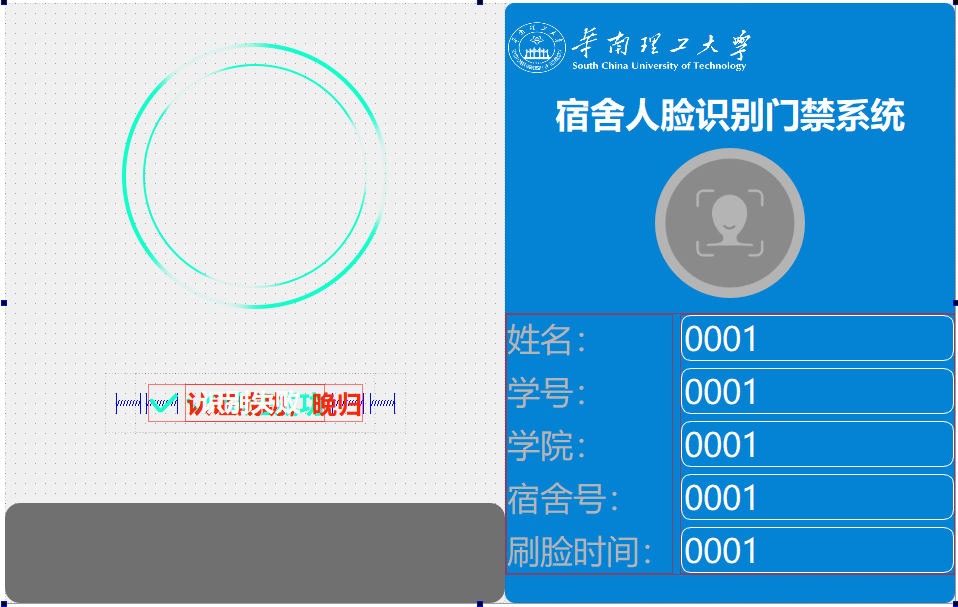
* **主要参与者：**管理员
* **目标：**查看和管理学生的刷脸记录。
* **触发条件：**管理员需要审查刷脸记录。
* **主要成功场景：**
  + 管理员访问刷脸记录管理界面。
  + 系统展示所有刷脸记录，包括学生姓名、刷脸时间等信息。
  + 管理员可以查询特定时间段或学生的刷脸记录。

用例图如下：

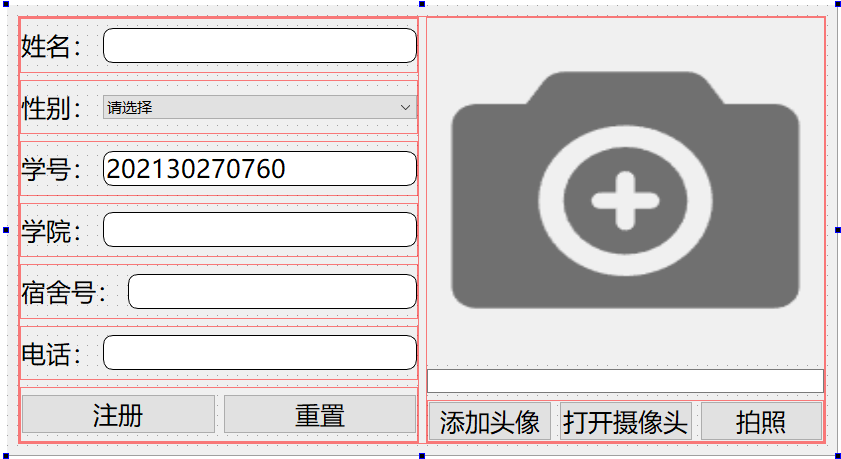


**4、用户界面图**

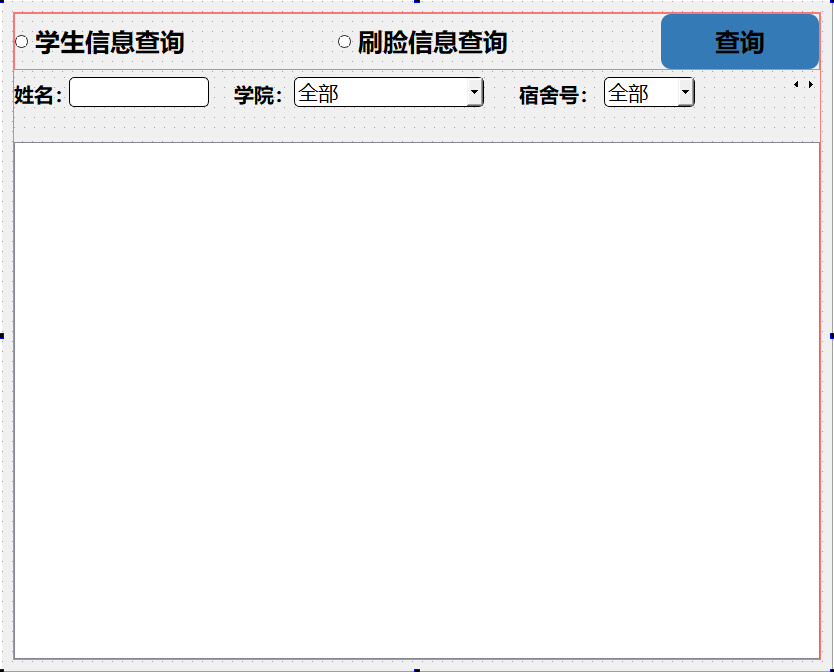
* 1. **客户端界面设计：**



* 1. **服务器端注册页面设计：**



* 1. **服务器端信息管理界面设计：**



# 技术实现

我采用了**Qt框架**开发客户端和服务器端，利用**OpenCV和SeetaFace库**进行人脸识别。通过**TCP/IP协议**实现了客户端与服务器端的数据通信。**数据库**用于存储学生信息和刷脸记录。**多线程技术**和标志变量被用于优化系统性能。

* **环境搭建：**配置了Qt开发环境，集成了OpenCV和SeetaFace库。
* **功能实现：**
  + 完成了客户端和服务器端的用户界面设计。
  + 实现了摄像头功能，支持实时采集人脸图片。
  + 实现了客户端与服务器端的通信功能。
  + 整合了客户端和服务器端的功能，完成了人脸识别门禁流程。
  + 实现了刷脸信息的存储与查询功能。
* **性能优化：**
  + 采用**多线程技术**优化耗时操作（即图像识别任务），提高系统响应性。
  + 引入标志变量避免短时间内对同一张人脸的多次识别。

# 完成情况总结

在本次实训中，我成功实现了一个功能完整的宿舍人脸识别门禁系统。我设计了友好的用户界面、实现了高效的人脸识别流程、系统测试与系统性能优化过程，从中我发现并解决了多个问题，如识别准确度和系统卡顿问题。用户界面和交互体验的改进使得系统更加友好易用，多线程技术也提高了系统的响应性。