

# 内窥镜Console系统软件项目报告

田钰栋 负责报告1、2、5部分

何牧航 负责报告4部分

冯烁芸 负责报告3部分

## 1.引言

### 1.1 编写目的

本项目旨在开发一款基于PyQt5的内窥镜Console系统软件，用于实时显示内窥镜传感器的图像并进行相关处理和管理。该软件具备以下主要功能：内窥镜图像实时预览、患者信息输入、图像采集与存储、图像防篡改处理以及定时发送图像到目标系统。

### 1.2 背景

内窥镜检查是一种常见的医学检查方法，广泛应用于消化道、呼吸道等疾病的诊断和治疗。传统的内窥镜系统通常只能提供实时图像展示，缺乏患者信息管理、图像存储和安全性保障等功能。因此，开发一款集成了这些功能的内窥镜Console系统软件对于提高内窥镜检查的效率和安全性具有重要意义。

本软件基于PyQt5框架开发，结合openCV库实现内窥镜传感器的接入，并实时显示图像。通过与用户界面的交互，实现患者信息输入、图像采集与存储、图像防篡改处理以及定时发送图像到目标系统的功能。通过将图像转换为DICOM格式并建立图像数据库，方便管理和查询已采集的图像信息。同时，引入图像防篡改处理技术，增强图像的安全性和完整性。

通过这份报告，我们将详细介绍软件的各项功能和实现方式，展示如何利用PyQt5和相关技术开发出一款功能强大的内窥镜Console系统软件。软件的开发将提高内窥镜检查的效率，便于医务人员进行患者信息管理和图像处理，同时保障图像的安全性和完整性，为医疗工作提供可靠的支持。

### 1.3 系统功能概述

内窥镜Console系统软件具备以下主要功能：

- 内窥镜图像实时预览：通过openCV库接入内窥镜传感器，实时显示每秒25帧的动态图像，提供内窥镜图像预览页面，给予医务人员全面的视觉展示。
- 患者信息输入：通过鼠标点击主页面的姓名栏，弹出患者检查信息输入对话框。用户可输入患者姓名、性别、年龄、检查医师和评论信息，并点击确认按钮后，将其更新到图像预览页面的对应字段，方便患者信息管理和标记。
- 图像采集与存储：用户可以通过鼠标点击图像的方式采集图像，并将采集的图像按DICOM SC（二次采集图像）图像格式转换为DICOM格式进行存储。每次采集图像后，软件会在图像数据库内保存图像索引信息，方便后续查询和管理。
- 图像防篡改处理：软件具备图像防篡改处理功能，能够在DICOM SC图像中插入相关字段，实现图像分区域篡改检测。系统能够辨别上左、上右、下左、下右部分像素矩阵是否被篡改，以及患者姓名、性别是否被篡改，增强图像的安全性和完整性。
- 定时发送图像到目标系统：系统会定时检查图像数据库，将状态为“未发送”的图像通过DICOM传输协议发送到目标系统。发送后，将图像状态设定为“已发送”，确保图像及时传输到目标系统，方便医务人员进行远程查看和诊断。

通过上述功能，内窥镜Console系统软件为医务人员提供了一种高效、安全和可靠的内窥镜图像管理和处理工具。软件的开发将提高内窥镜检查的效率，方便医务人员对患者信息管理和图像处理，同时保障图像的安全性和完整性，为医疗工作提供可靠的支持。在接下来的报告中，我们将详细介绍软件的具体实现方式和技术细节，以及UI设计的规范要求。我们将按照软件开发的步骤，逐一介绍每个功能的实现方法，并展示软件的界面设计和交互流程。同时，我们将介绍图像处理和安全性保障的相关技术，以及如何利用PyQt5框架实现用户界面和与图像传感器的交互。通过本报告，我们旨在提供一份清晰的软件开发指南，帮助开发人员了解内窥镜Console系统软件的设计和实现，为医疗行业提供更好的技术支持。

## 2.技术与工具

### 2.1 PyQt5框架简介

PyQt5是一个基于Python语言的GUI（图形用户界面）开发框架，用于创建功能强大且交互性良好的桌面应用程序。它是对Qt库的Python封装，Qt是一个跨平台的C++应用程序开发框架，具有丰富的功能和广泛的应用领域。相较于其他的GUI开发框架PyQt5具备许多独特的优点：首先，PyQt5具有强大的跨平台性，可以在各种操作系统上运行，包括Windows、MacOS、Linux等，使得开发的应用程序具备良好的可移植性；并且PyQt5提供了丰富的GUI组件和工具，可以实现各种各样的用户界面设计，满足不同应用的需求；此外，PyQt5支持与其他Python库和工具的集成，使得开发者可以方便地利用Python生态系统中的丰富资源来增强应用程序的功能；最后，PyQt5拥有完善的官方文档和活跃的开发者社区，开发者可以轻松地获取支持和解决问题。综上所述，我们最终选择了PyQt5作为我们软件的开发框架。

在开发软件的过程中，我们使用到了以下的PyQt5的核心组件：

- QtCore：提供了核心的非图形相关功能，如事件处理、定时器、线程和文件IO等。
- QtGui：包含了图形相关的功能，如窗口、按钮、标签、绘图等。
- QtWidgets：建立在QtGui之上，提供了更高级的UI组件，如对话框、菜单、工具栏等，是开发桌面应用程序的主要模块。

使用PyQt5进行开发的一般流程是先使用Qt Designer工具进行界面设计，通过可视化方式拖拽组件、设置布局 and 属性，生成.ui文件。然后，使用pyuic工具将.ui文件转换为对应的Python代码，生成用于界面初始化的代码。接下来，开发人员可以编写Python代码来处理用户交互和实现业务逻辑，通过信号与槽机制将界面与代码连接起来。在调试和测试阶段，可以使用Python的调试工具和单元测试框架来确保应用程序的正确性和稳定性。最后，使用工具如PyInstaller或cx\_Freeze将Python代码和依赖库打包成可执行文件或安装包，方便应用程序的发布和部署。总之，PyQt5是一个功能强大且受欢迎的GUI开发框架，在接下来我们将使用它来实现内窥镜Console系统的图形界面和交互功能。

### 2.2 openCV库

OpenCV（Open Source Computer Vision）是一个开源计算机视觉库，旨在提供丰富的图像和视频处理算法及工具。它由一系列的C++函数和少量的Python接口组成，可以在多个平台上运行，包括Windows、Linux、macOS和Android。OpenCV提供了广泛的功能，涵盖了计算机视觉领域的各个方面，包括图像处理、特征检测与描述、目标识别、物体跟踪、摄像机标定、三维重建等。它还支持多种编程语言，包括C++、Python、Java和MATLAB，使开发者能够根据自己的喜好和需求选择适合的编程语言进行开发。同时，OpenCV具有多个特点和优势，包括跨平台性、丰富的图像处理算法、强大的计算机视觉功能、高效的图像和视频处理以及丰富的社区支持和资源。通过OpenCV，开发人员可以在多个操作系统上运行，并利用其丰富的图像处理算法来进行滤波、边缘检测、形态学操作等任务。此外，OpenCV还提供了特征检测与描述、目标识别与跟踪、摄像机标定与姿态估计等强大的计算机视觉功能。它使用并行计算和硬件加速等技术，以提高图像和视频处理的效率和性能。同时，OpenCV拥有庞大的开发者社区，提供了丰富的文档、教程和示例代码，为开发者提供了支持和解决问题的资源。总之，OpenCV是一款功能强大的计算机视觉库，我们将利用它来实现内窥镜Console系统中的图像处理和计算机视觉功能。

## 2.3 DICOM格式和DICOM数据模型

DICOM (Digital Imaging and Communications in Medicine) 是医学图像和相关数据的国际标准格式。它定义了一种用于存储、传输和共享医学图像和相关信息的文件格式和通信协议。DICOM格式结合了图像数据和元数据,使得医学图像可以在不同的设备和系统之间进行可靠和一致的交换和处理。

DICOM数据模型是DICOM格式的基础,它描述了如何组织和表示医学图像和相关数据。DICOM数据模型基于对象的概念,其中每个对象由一组属性 (Attributes) 组成。属性包括标签 (Tag)、值 (Value) 和值的表示 (Value Representation, VR) 等信息。标签是唯一标识属性的数值,例如 (0010,0010) 表示患者姓名。值是属性的实际数据,可以是文本、数字、图像等。值的表示定义了值的数据类型和格式,如字符串、整数、日期时间等。DICOM数据使用层次结构组织属性,最小的单位是元素 (Element),每个元素由标签、VR和值组成。元素可以通过标签进行唯一标识和访问。元素可以组合成一组,形成一个更大的对象,称为数据集 (Dataset)。数据集可以包含多个元素,用于描述医学图像的各个方面,如患者信息、图像序列、图像像素数据等。它还定义了一些特殊的对象,如SOP (Service-Object Pair) 类和IOD (Information Object Definition)。SOP类是一种具有特定功能和特性的对象,例如CT图像、MRI图像等。每个SOP类都有唯一的标识符 (SOP Instance UID),用于区分不同的实例。IOD定义了SOP类的属性和行为,规定了SOP类的特定要求和限制。DICOM数据模型的灵活性和标准化使得不同设备和系统能够共享和解析医学图像和相关数据。它提供了一致的数据表示和交换方式,使得医学图像可以在不同厂商的设备之间进行无缝传输和解析。同时,DICOM数据模型还支持扩展,允许根据具体应用需求定义和使用自定义的属性和对象。

总之,DICOM格式和DICOM数据模型提供了一种标准化的方式来组织、存储和传输医学图像和相关数据。它基于对象和属性的概念,定义了医学图像的组织方式和表示方法。通过DICOM格式和数据模型,医学图像可以在不同设备和系统之间进行无缝交换和解析,促进了医学图像的共享和应用。

## 2.4 数据库

数据库是用于存储和管理数据的系统。数据库系统基于一组原理和概念,旨在提供可靠、高效和安全的数据管理。下面是数据库原理的简要介绍:

1. 数据库模型: 数据库模型定义了数据在数据库中的组织方式和表示方法。常见的数据库模型包括层次模型、网络模型、关系模型和对象模型。其中,关系模型是最广泛应用的模型,它使用表格形式的关系 (表) 来表示数据,并使用关系代数和关系演算进行数据操作和查询。
2. 数据库管理系统 (DBMS): 数据库管理系统是用于管理数据库的软件系统。它提供了数据定义语言 (DDL) 和数据操作语言 (DML) 等接口,用于定义数据库结构、操作数据和执行查询。DBMS负责数据的存储、索引、事务管理、并发控制、安全性和数据完整性等方面。
3. 数据库事务: 事务是数据库中的一个逻辑单位,它由一系列操作组成,要么全部执行成功,要么全部回滚。事务具有ACID属性: 原子性 (Atomicity) 保证事务中的操作要么全部执行成功,要么全部回滚; 一致性 (Consistency) 保证事务将数据库从一个一致状态转换到另一个一致状态; 隔离性 (Isolation) 保证并发执行的事务相互隔离,互不干扰; 持久性 (Durability) 保证事务一旦提交,其结果将永久保存。
4. 数据库索引: 索引是一种数据结构,用于加快对数据库表中数据的访问速度。索引通过提供快速的数据查找和排序功能来优化查询性能。常见的索引类型包括B树索引、哈希索引和全文索引等。
5. 数据库范式: 范式是一组规则,用于设计关系数据库中的表结构。它通过消除数据冗余和提高数据一致性来优化数据库设计。常见的范式包括第一范式 (1NF)、第二范式 (2NF)、第三范式 (3NF) 等。
6. 数据库安全性: 数据库安全性涉及保护数据库中的数据免受非授权访问、恶意操作和数据丢失等威胁。数据库安全性的措施包括访问控制、身份认证、授权管理、加密和备份与恢复等。
7. 数据库优化: 数据库优化旨在提高数据库的性能和效率。优化技术包括索引优化、查询优化、表设计优化、缓存策略和存储优化等。优化可以减少查询时间、提高并发性能和降低系统资源消耗。

总之，数据库原理涉及数据库模型、数据库管理系统、事务管理、索引、范式、安全性和优化等方面。了解这些原理有助于设计和管理高效、可靠的数据库系统，并提供可靠的数据存储和访问功能。

## 2.5 图像防篡改

本项目的放篡改算法主要基于哈希算法来实现，哈希算法是一种将任意长度的数据转换为固定长度哈希值的算法，其具有唯一性、不可逆性、一致性能特点，在本系统设计中，可以在图像上传时对DICOM图像进行哈希计算，并将计算得到的哈希值存储在系统中或与图像相关的记录中。当需要验证图像的完整性时，可以重新计算图像的哈希值，并与之前存储的哈希值进行比对。如果两个哈希值一致，则说明图像没有被篡改；如果哈希值不一致，则表示图像可能遭到了篡改。常用的哈希算法包括MD5、SHA-1、SHA-256等。在选择哈希算法时，需要考虑算法的安全性、计算效率和哈希值的长度等因素。较新的算法如SHA-256通常较为安全，但计算复杂度较高。通过应用哈希算法，系统能够提供一种可靠的方式来验证DICOM图像的完整性，防止图像数据被篡改或损坏。此外，为增强系统的安全性和防护能力，本系统采用数据加密和数字签名技术。通过对DICOM图像进行加密，保护图像数据不被非法访问。同时，使用数字签名来验证图像的完整性和真实性，确保图像未被篡改。

## 3.系统功能介绍

### 3.1 图像预览与患者信息收集界面

#### 图像预览

- 图像预览界面包括患者姓名、性别、年龄、日期、时间、检查医师信息和评论信息栏，以及图像显示区域。通过点击姓名栏弹出的对话框，用户可以更新姓名、性别、年龄、检查医师信息和评论信息。日期和时间实时显示，每秒刷新一次。图像显示区域的罩层形状可以通过修改配置文件进行设置，支持 circles、squares 和 octagons，并且图像更新帧率为30fps。

#### 患者信息收集

Name :

gender:


age:

2024-01-06

21:38:26

Physician:

Comment:



编辑用户信息

Name :

Age :

Gender :

Physician:

comment:

确认

取消

- 患者信息收集模块通过鼠标点击内窥镜图像预览页面的姓名栏，弹出患者检查信息输入对话框来实现。
- 这一模块使用了PyQt5库创建了一个窗口应用程序，用于显示图像预览和收集患者信息。主要实现过程是：创建了一个UserInfoDialog类，继承自QDialog，作为窗口的主要界面。
- 该界面中包含以下组件：
  - QLabel用于显示标题和标签，例如"Name"、"Age"、"Sex"、"Physician"、"Comment"。
  - QLineEdit用于接收用户输入的患者信息，例如姓名、年龄等。
  - QRadioButton用于选择性别，有"Male"和"Female"两个选项。
  - QPushButton用于提交和取消按钮。
- 该界面的功能包括：
  1. 点击患者姓名栏目，弹出用户信息输入模式对话框
  2. 模式对话框内，可输入患者姓名、性别、年龄、检查医师信息和评论信息
  3. 点击确认按钮后，模式对话框退出，并将用户输入的患者姓名、性别，更新到主界面相应栏目。

## 3.2 数据采集与存储

### 图像数据结构

- 首先需要初始化一个字典ImageSopInsInfo，用于存储输入的信息以及传输。
- 该数据结构包含患者（Patient）、检查（Study）和系列（Series）等层次结构。
- 该数据结构将实现存储数据信息（姓名，年龄，性别，DICOM文件相关的信息等），以及更新数据库两个功能。

### 数据采集与存储

- 在数据采集模块中，通过鼠标点击图像预览页面，触发图像采集过程，同时将数据保存在数据库以及本地指定路径中，这一过程以DICOM格式为基础进行数据存储。以下是数据采集的主要步骤：
  1. 通过鼠标点击获取图像，包括整个图像预览页面的内容。
  2. 通过读取系统配置文件获取图像存储路径（ImageRepositoryPath）。
  3. 生成唯一的SOP Instance UID（SOPInsUID）和SOP Class UID（SOPClaUID）。
  4. 创建DICOM文件，并设置文件元信息（FileMetaDataset）。
  5. 从图像预览界面获取患者的基本信息，包括姓名、性别、年龄、检查医师信息和评论信息。
  6. 将患者信息和图像数据添加到DICOM文件中，包括设置像素数据、图像尺寸等。
  7. 以DICOM格式保存文件，并加入数据传输协议，保证文件的正确性和完整性。
  8. 对DICOM文件进行防篡改加密处理。
  9. 将加密后的DICOM文件以及图像数据保存到指定的输出路径。
  10. 更新图像数据结构以及数据库，确保数据库中的信息与实际采集到的信息一致。

## 3.3 数据防篡改处理

该模块的主要功能是加密DICOM文件，以确保数据的安全性。以下是加密过程的详细步骤和功能：

### 加密过程步骤

#### 1. 读取DICOM文件：

使用 `dcmread` 函数读取指定路径下的DICOM文件，并获取其中的像素数据（`pixel_data`）以及关键元信息。

#### 2. 加载公钥和私钥：

通过 `load_private_key` 和 `load_public_key` 函数加载私钥和公钥，用于数字签名和验证。

### 3. 添加时间戳和公钥信息：

向DICOM文件中添加时间戳和公钥信息，以便后续的数字签名和验证。时间戳记录当前加密操作的时间，而公钥则作为加密过程的一部分保存在DICOM文件中。

### 4. 定义四个区域并进行数字签名：

定义图像的四个区域（上左、上右、下左、下右），分别提取像素数据，并计算哈希值。使用私钥对哈希值进行数字签名，生成数字签名（`signature`）和哈希摘要（`digest`）。

### 5. 生成患者姓名和ID的数字签名：

将患者姓名和ID转换为字节串，使用私钥对其进行数字签名，生成数字签名和哈希摘要。

### 6. 创建数字签名Sequence：

创建一个空的Sequence（`sequence`），用于存储区域和患者信息的数字签名。将每个数字签名信息（区域名称、哈希摘要、数字签名）添加到Sequence中。

### 7. 将Sequence添加到DICOM文件：

将Sequence添加到DICOM文件中，确保数字签名信息与相应的区域和患者信息关联。

### 8. 设置DICOM文件的Transfer Syntax：

设置DICOM文件的Transfer Syntax为ExplicitVRLittleEndian，以保证数据的正确解析和传输。

### 9. 保存加密后的DICOM文件：

将加密后的DICOM文件保存到指定路径，完成加密过程。

## 功能说明

- **数字签名：**

通过使用私钥对特定区域和患者信息的哈希值进行数字签名，确保数据的完整性和来源可信。

- **加密认证部分：**

在DICOM文件中添加时间戳和公钥信息，同时保存了图像像素数据。这些信息将在后续的验证过程中用于确认数据的完整性和真实性。

- **区域定义和签名：**

通过定义四个区域，将图像分割，并对每个区域的像素数据进行数字签名。这一步骤增强了对图像各部分的安全验证。

- **Sequence的创建与添加：**

创建一个空的Sequence，用于存储数字签名信息。每个数字签名信息包含区域名称、哈希摘要和数字签名，确保数字签名与相应的数据部分关联。

- **设置Transfer Syntax：**

设置DICOM文件的Transfer Syntax为ExplicitVRLittleEndian，以确保DICOM文件的正确解析和传输。

通过以上步骤，加密过程将保障图像数据的安全性，防范数据篡改和伪造，同时为后续验证提供了关键信息。

## 3.4 数据传输

### 数据传输步骤

#### 1. 定时图像发送：

- `imageSendProc` 方法通过定时器每20秒执行一次，调用了 `sendImagesToTargetSystem` 方法，实现定时图像发送的功能。

#### 2. 连接数据库：

- 在 `sendImagesToTargetSystem` 中，首先建立与SQLite数据库的连接，使用 `sqlite3.connect` 方法。

#### 3. 查询符合条件的图像记录：



- 执行SQL查询语句，筛选出数据库中符合条件（`storagePath`与传入的 `filepath` 相同，且 `sendFlag` 为0）的图像记录。

#### 4. 发送DICOM图像：

- 如果找到符合条件的图像记录，调用 `dicom_send_image` 函数，将DICOM图像发送至目标系统。
- `dicom_send_image` 函数使用 `pydicom` 库建立一个DICOM Application Entity (AE)，读取DICOM文件，添加请求的呈现上下文，与目标系统建立关联，并使用C-STORE服务发送DICOM数据集。

#### 5. 更新发送标志：

- 如果发送成功，通过SQL更新数据库中的图像记录，将 `sendFlag` 设置为1，表示该图像已成功发送。

#### 6. 处理异常情况：

- 异常处理：如果在任何步骤中发生错误，捕获异常并打印错误信息。

#### 7. 关闭数据库连接：

- 在最终的 `finally` 块中，关闭与SQLite数据库的连接，确保良好的资源管理。

## 功能说明

### • 定时图像发送：

- 定时执行 `imageSendProc` 方法，实现图像定期发送的功能，可以按需调整定时器的时间间隔。

### • 查询与发送：

- 根据指定的 `filepath` 查询数据库中是否存在待发送的图像记录，若存在则进行发送操作。

### • DICOM图像发送：

- 通过 `dicom_send_image` 函数，建立DICOM Application Entity，与目标系统建立连接，并使用C-STORE服务将DICOM图像数据集发送至目标系统。

### • 更新发送标志：

- 在发送成功后，将数据库中相应图像记录的 `sendFlag` 标志更新为1，表示该图像已成功发送。

### • 异常处理：

- 捕获任何可能发生的异常，防止程序中断并记录错误信息。

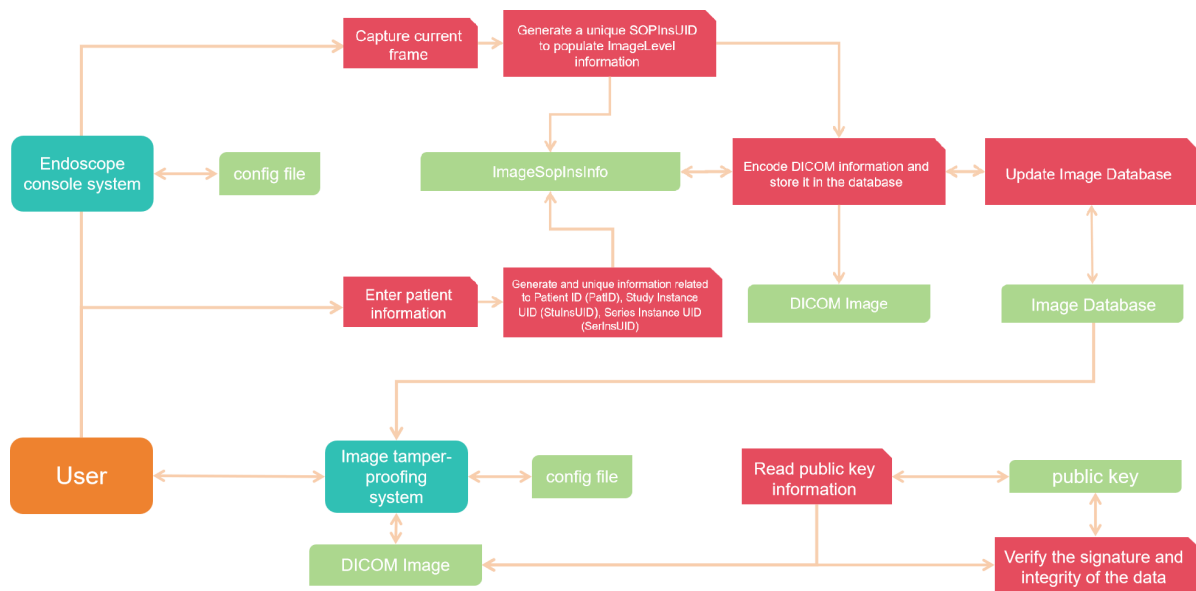
### • 数据库连接管理：

- 通过 `connection.close()` 确保在任何情况下都关闭了数据库连接，以防止资源泄漏。

通过以上步骤和功能，实现了一个简单的定时图像发送机制，确保了图像数据的及时传输，并在发送成功后更新数据库，实现了数据传输的自动化管理。

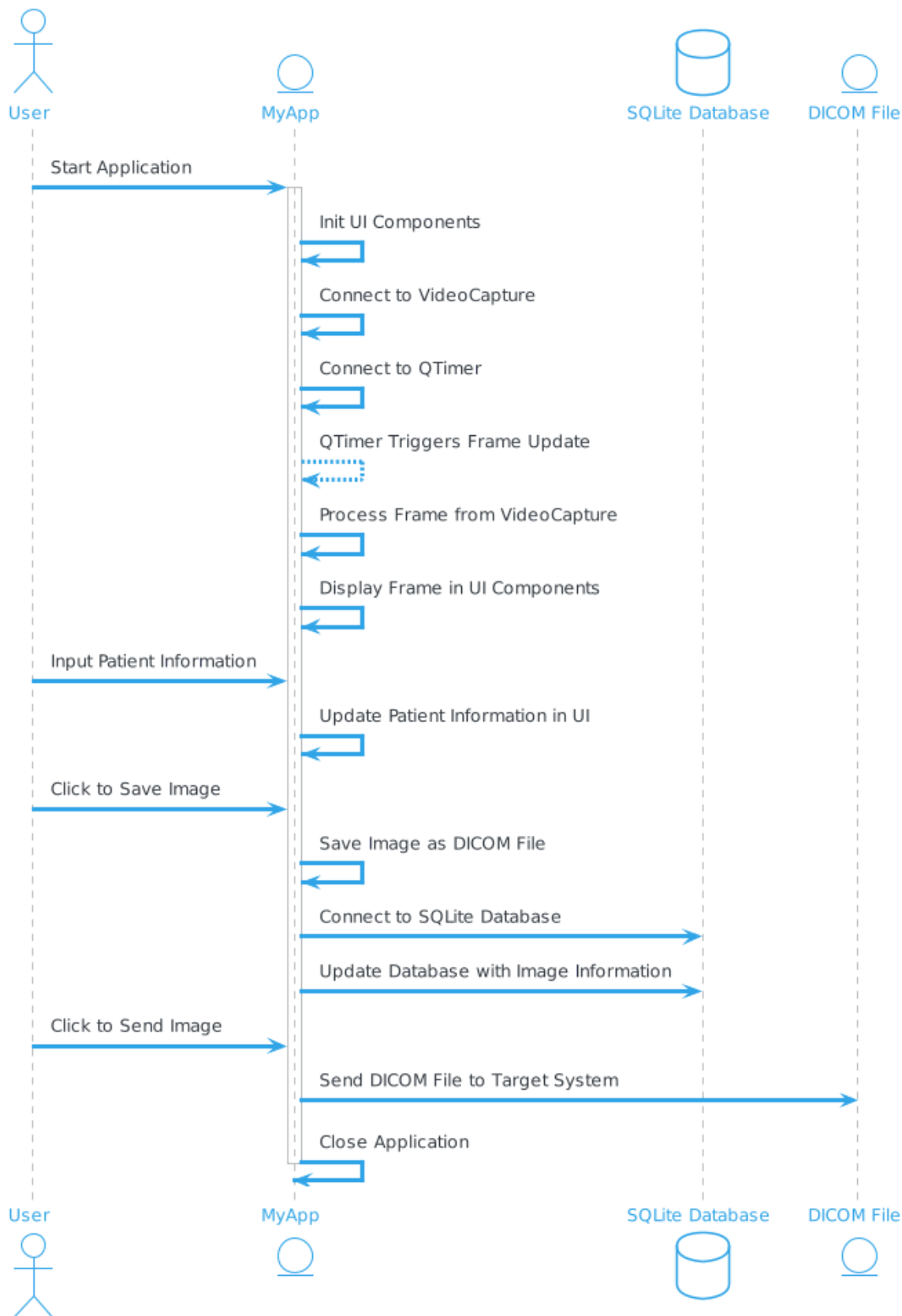
## 4.软件运行流程

### 运行功能流程图



数据流图





## 5.总结与展望

在本项目中，我们成功地开发了一款基于PyQt5的内窥镜Console系统软件，该软件具备实时图像显示、患者信息输入、图像采集和存储、图像数据库建立和索引、图像防篡改处理以及图像传输等功能。通过本项目，我们成功地实现了内窥镜图像的实时显示和预览功能，并通过openCV库实现了内窥镜传感器的接入。患者信息输入模块允许用户方便地输入患者的检查信息，并将其与图像预览页面关联。图像采集和存储模块能够按照DICOM SC图像格式采集图像，并将其转换为DICOM格式进行存储。通过图像数据库的建立和索引，我们能够高效地管理和检索大量的内窥镜图像数据。图像防篡改处理模块的

实现使得我们能够对图像进行分区域篡改检测，增强了图像的安全性和完整性。最后，通过DICOM传输协议，我们能够将图像发送到目标系统，并实现了图像状态的管理。

在项目完成过程中，我们深入理解了PyQt5框架和openCV库的使用，以及DICOM格式和数据模型的规范。我们通过良好的团队协作和任务分配，按计划完成了项目各个阶段的开发和测试工作。同时，我们也遇到了一些挑战和问题，例如图像处理算法的选择和优化，以及与实际硬件设备的兼容性问题。这些经验和教训将对我们未来的项目开发和团队合作有所裨益。

尽管我们的内窥镜Console系统软件已经实现了基本的功能，并取得了满意的结果，但仍有一些改进和扩展的空间。未来，我们可以考虑加入更多的图像处理和分析功能，例如图像增强、病变检测和测量功能，以提高系统的临床应用价值。此外，我们还可以进一步优化系统的性能和用户界面，以提供更好的用户体验。另外，我们也可以考虑与其他医疗信息系统进行集成，以实现更高效的数据交换和共享。综上所述，通过本项目，我们成功地开发了一款功能齐全的内窥镜Console系统软件，并取得了满意的成果。我们在项目中积累了宝贵的经验和教训，并为未来的工作提供了有益的展望和方向。这个软件的开发为医疗行业提供了一种高效、安全和可靠的内窥镜图像管理和分析解决方案。

## 6.附录

---

在本次课程设计中，完成了一项基于PyQt5的内窥镜Console系统软件的开发任务。通过这个项目，我们学到了许多关于软件开发和图像处理方面的知识，并且基本上完成了整个软件的开发。通过这一学期的课设教学，我深入了解了软件开发的实际过程，也学习了PyQt5框架和openCV库的使用，掌握了图像处理算法和DICOM格式的知识。通过实际的开发任务，我对软件架构、模块设计和团队协作有了更深入的理解。此外，也提高了问题解决能力和调试技巧，对软件开发的整体流程有了更清晰的认识。整体来说，在这门课程中我学习到了许多有用的地方，一个小建议就是软件开发通常是一个团队合作的过程，因此在课程中可以加强团队协作和沟通能力的培养。例如可以在课程一开始就让学生进行组队，通过小组项目或团队合作作业，让学生学会有效地分工合作、交流协调，并体验团队合作中的挑战和解决方案。