

软件工程课程报告

班 级：191171

学 号：20171001967

学生姓名：崔令祎

指导老师：王勇

日 期：2020 年 6 月

目录

- 摘要3
- 一．需求分析.....3
 - 1.1 数据模型-ERD3
 - 1.2 功能模型-数据流图4
 - 1.3 行为模型-状态转换图5
- 二．界面设计.....6
- 三．结构化设计.....9
 - 3.1 概要设计10
 - 3.2 详细设计10
 - 3.2.1 引言10
 - 3.2.2 屏幕区域获取模块设计说明11
- 四．面向对象设计.....14
 - 4.1 序列图.....15
 - 4.2 用例图.....15
 - 4.3 活动图.....16
 - 4.4 类视图.....17

摘要

二次压降检测系统的意义就是在电脑上用软件控制检测等一系列操作，把很多手动操作比如按按钮、整理数据交给软件来实现。可以说是增强了检测仪器的功能。

软件设计中用 ERD 、数据流图、状态转换图、模块结构图、UML...来描述用户，需求、系统功能之间的关系。这样的描述有助于站在用户角度表达软件要达到什么需求，抛弃程序员只考虑实现的固定思维，更直观明了的表达软件的需求。

一. 需求分析

使用结构化分析技术完成分析用户需求的工作。结构化是把软件系统功能当作一个大模块，根据分析与设计的不同要求，进行模块分解或者组合。把整个系统开发过程分成若干阶段，每个阶段进行若干活动，每项活动应用一系列标准、规范、方法和技术，完成一个或多个任务，形成符合需求的软件产品。

1.1 数据模型-ERD

ER 图有七个实体，其中（二次压降）检测器是指检测二次压降的组合，包括互感器和互感器二次压降检测仪；控制器是计算机的抽象，用于控制检测设备和与使用者交互；测试员是使用该系统的用户的抽象；记录器是用于拍照的仪器的抽象，可以是相机，也可以是手机。联系有七个，有控制器“控制”，使用者“传递信息”，控制器“发送命令”...每个实体都有属性。

检测结果（结果号，电压，比差，角差，压降误差，生成时间）

测试命令（命令号，结果号，命令数量）

检测设备（设备号，型号，生产厂商）

控制器（控制器编号，名称，接口类型，内存大小）

测试员（测试员号，姓名）

检测信息（设备号，测试员号，检测日期）

记录器（记录器编号，型号，名称）

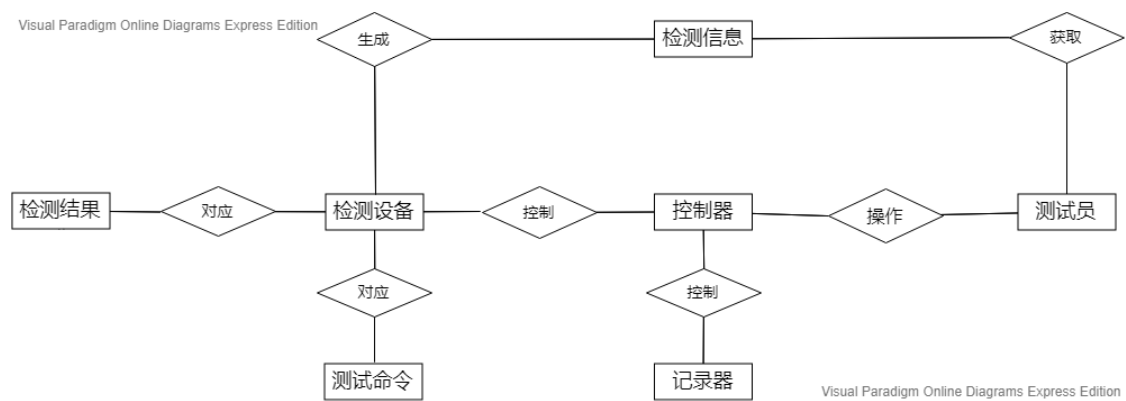


Figure 1 二次压降检测系统 ER 图

1.2 功能模型-数据流图

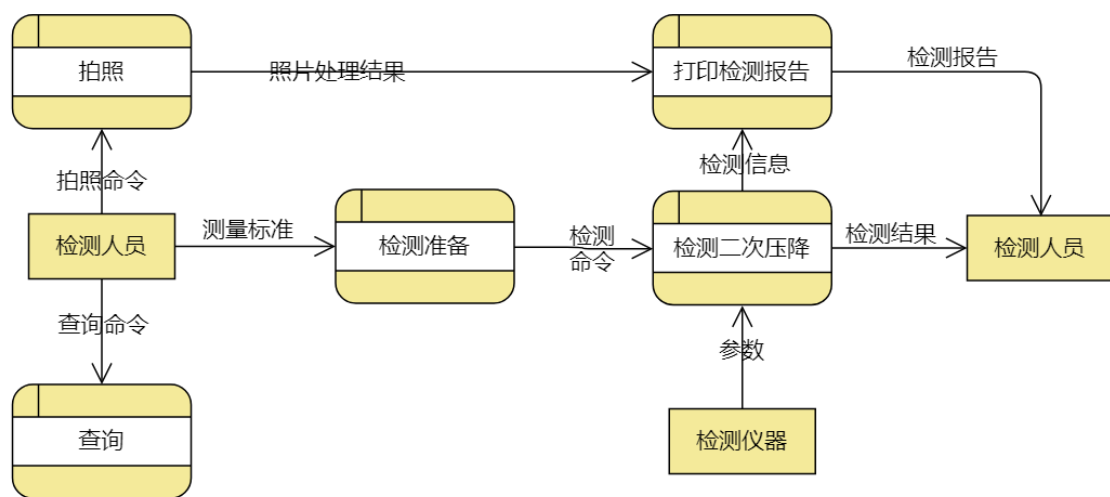


Figure 2 0 层图

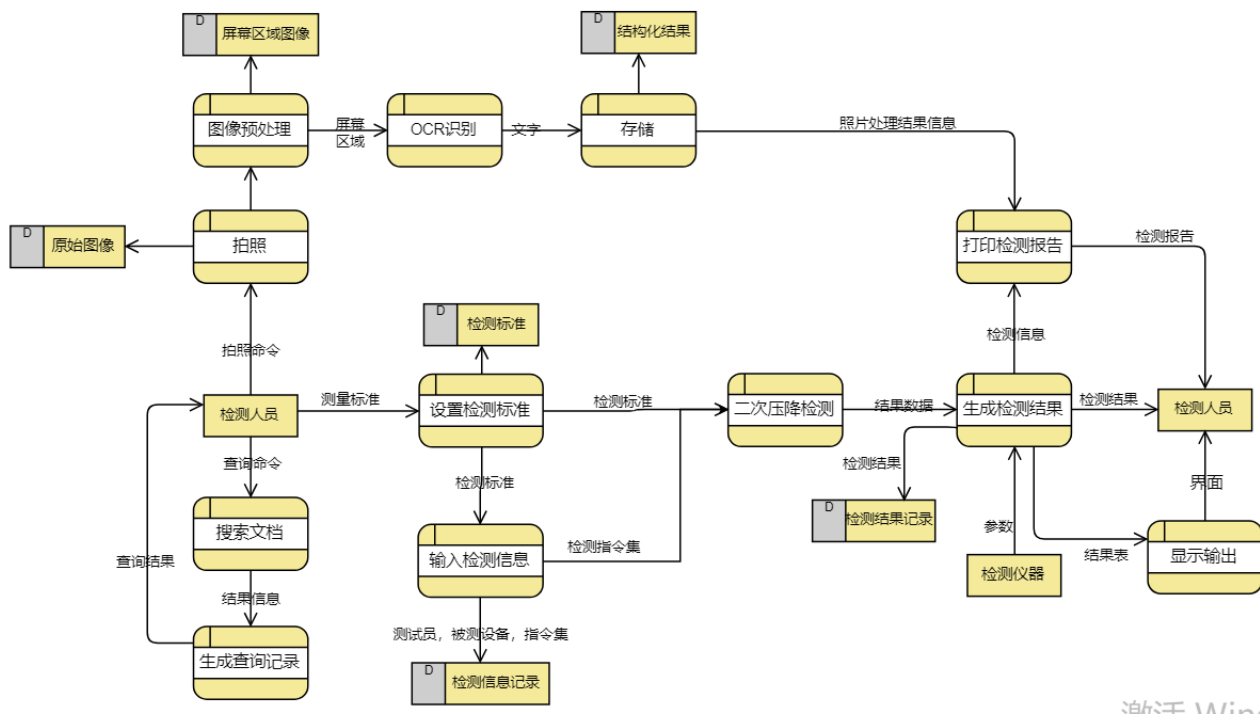


Figure 3 细化数据流图

1.3 行为模型-状态转换图

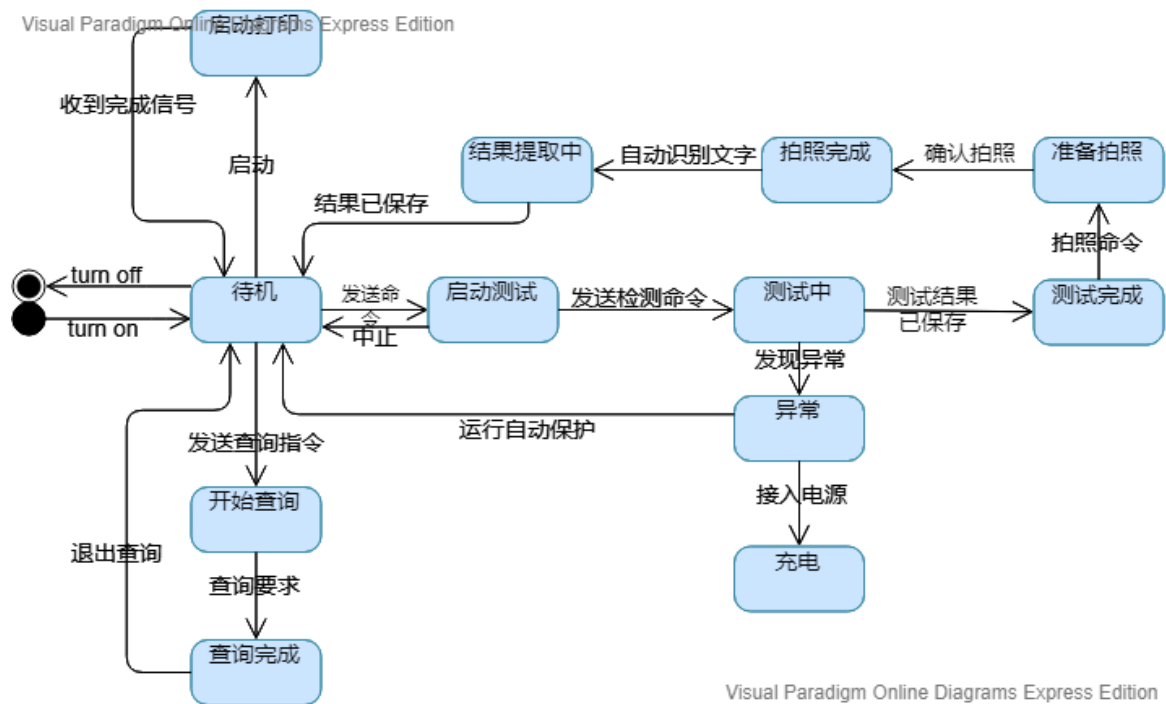
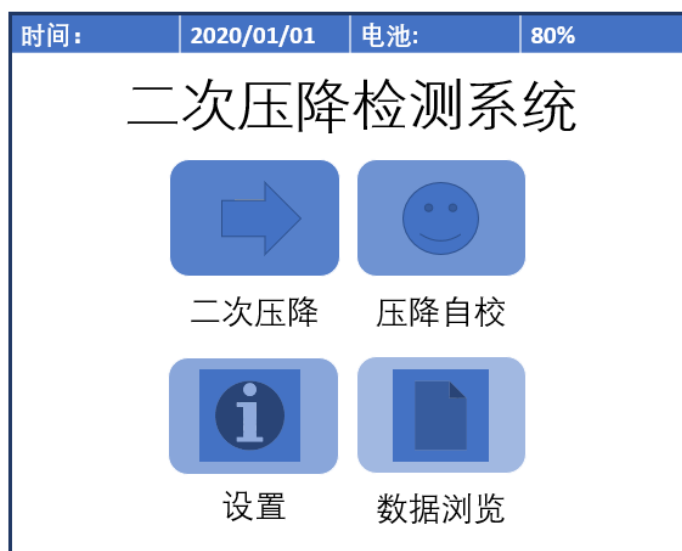


Figure 4 状态转换图

二. 界面设计



- 二次压降：点击此图标将进入电压互感器二次回路压降测量功能,其中包括三相三线、三相四线以及单相压降测试功能。还可以加入一个“停止”选项。

时间:	2020/01/01	电池:	80%
<h2>二次压降</h2>			
			
三相三线		三相四线	
		单相	

时间:	2020/01/01	电池:	50%
<h2>三相四线测试</h2>			
测试人: Tracy 测试编号: 28 测试时间: 2019/01/01			
	L1	L2	L3
电压
比差
角差
压降误差
综合误差	0.4938%		
<div> <div>测量</div> <div>保存</div> <div>停止</div> <div>屏幕快照</div> <div>返回</div> </div>			

- 压降自校: 点击此图标将进入三相三线、三相四线压降自校测试功能。

时间:	2020/01/01	电池:	80%
<h2>压降自校</h2>			
	A0	B0	C0
比差
角差
<div> <div>开始</div> <div>保存</div> <div>停止</div> <div>返回</div> </div>			

- 数据浏览：点击此图标将进入浏览仪器内部存储器中的各测试记录,可以进行数据的浏览以及删除、通讯、打印、查询等操作。

时间:	2020/01/01	电池:	50%
<h2>数据浏览</h2>			
测试人:	Tracy	测试编号:	28
测试时间:	2019/01/01		
被测设备号:	A309		
	L1	L2	L3
电压
比差
角差
压降误差
综合误差	0.4938%		

上一页

下一页

导出/打印

删除

返回

时间:	2020/01/01	电池:	80%	— □ ×
<div><<返回<div>历史记录</div><div>更多筛选</div></div>				
<h2>查询</h2>				
<div>输入关键字: <input type="text"/></div>				
<div>AI搜索</div>				

时间:	2020/01/01	电池:	80%	— □ ×
<div><<返回<div>查询</div></div>				
<div>查询时间</div>				
<div><div>当天</div><div>七天</div><div>一个月</div><div>自定义</div></div>				
<div>排序</div>				
<div><div>由近及远</div><div>由远及近</div></div>				
<div><div>重置</div><div>确定</div></div>				

- 系统设置：点击此图标将可以设置系统时间和液晶对比度。

时间：	2020/01/01	电池：	80%
-----	------------	-----	-----

系统设置


日期/时间


参数设置


显示设置

时间：	2020/01/01	电池：	80%
-----	------------	-----	-----

时间/日期设置

更改日期和时间

日期

2020年 ▾

12月 ▾

29日 ▾

时间

11 ▾

40 ▾


更改

取消


时间：	2020/01/01	电池：	80%
-----	------------	-----	-----

显示设置

更改背光亮度



更改屏幕对比度



完成

返回

三. 结构化设计

3.1 概要设计

结构化设计建模：用模块结构图表示。对于中等规模的软件，其深度约为 10 层左右；好的系统的平均扇出数通常是 3—4，顶层扇出数较高，中间层扇出数比较少，底层扇入数比较高。

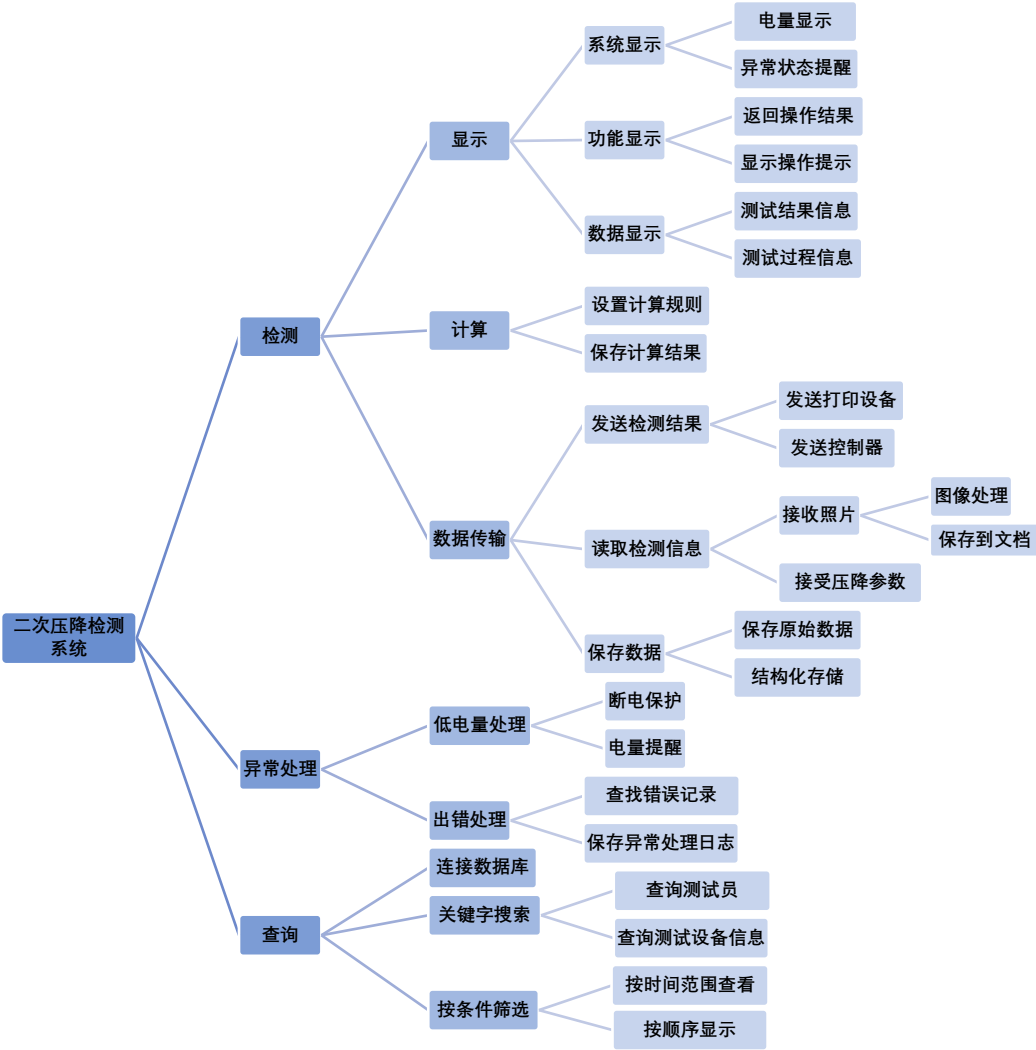


Figure 5 模块结构图

3.2 详细设计

3.2.1 引言

(1) 编写目的

详细设计的任务就是要决定各个模块的实现算法，并精确表达出这些算法。

本详细设计说明书是在二次压降检测系统概要设计的基础上进一步明确系统结构，详

细地介绍系统的各个模块，为进行后面的实现和测试作准备。预期读者为开发人员、系统测试维护人员。

(2) 背景

- 所建议开发的软件系统的名称：二次压降检测系统；
- 任务提出者：某公司；
- 开发者：崔令祎；
- 用户：检测人员及实现该软件的计算中心或计算机网络：个人笔记本电脑、公司机房

(3) 定义

- 主控仪器：计算机，可以是台式机，也可以是笔记本电脑。
- 检测仪器：二次压降检测仪和互感器。
- 记录器：带有拍照功能的手机或相机。
- OCR: Optical Character Recognition, 光学字符识别。

(4) 参考资料

[1]伊恩·萨默维尔.软件工程[M].机械工业出版社:北京,2018:50-200.

[2]张海藩.软件工程[M].清华大学出版社:北京,2013:20-200.

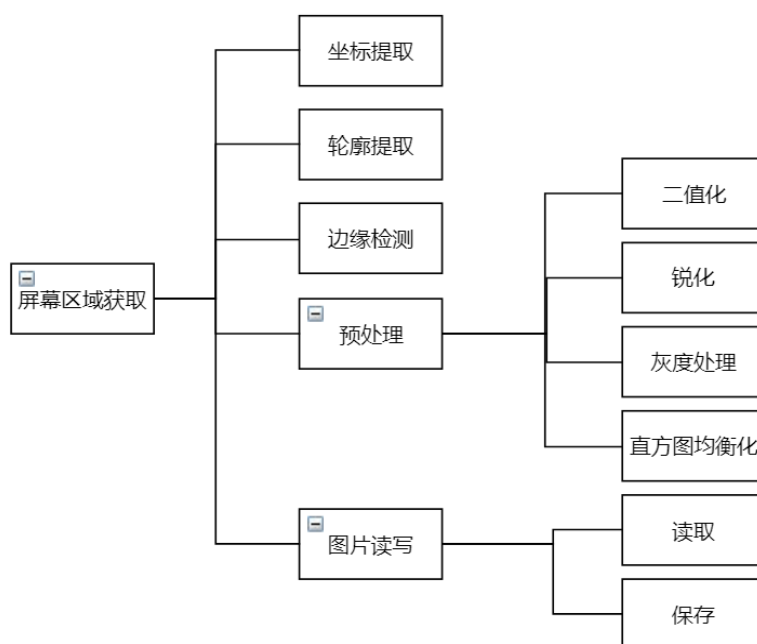
3.2.2 屏幕区域获取模块设计说明

(1) 程序简单描述

屏幕区域获取程序主要实现提取屏幕特征功能，可重用。只要电脑获取到相机拍摄的照片就可以在此程序中进行操作：打开摄像头拍照，采集屏幕区域数据。图像进行灰度变化,首先进行预处理,将背景噪声以及图像中的突刺变化去除,防止影响后面的图形计算精度；利用 C++语言，利用 OpenCv 库中的已有函数接口进行定位,将轮廓提取出来,这里面的用的方法：具有尺度不变性的模板匹配。

(2) 整体结构

使用者导入的是未经处理的原始图像，模块经过一系列处理得到屏幕图片，用于存档和提取文字。



(3) 性能

算法准确率 100%，平均耗时 0.1s。具有良好的实时性，在较小的噪声环境下具有一定的鲁棒性。

(4) 输入输出项

输入项				
名称	标识	数据类型	有效范围	输入方式
图片	Pic	Mat	0~65535	由参数传入
数量	Amount	Int	0-20	自动输入

输出项			
名称	标识	数据类型	有效范围
屏幕图片	Screen	Mat	20 位数字以内
结果数量	FinishedNo	INT	1-20

(5) 算法

把图像提取封装在 PictureProcessing 类中：继承基类 CameraControl 拍照类，需要实现接口函数：protected Bitmap processFrame(int flag, Mat capture)

构造函数中需要实现的是：获取图像资源，进行文件的读写，利用该文件进行分类器对象的初始化，分类器对象在 OpenCV 中是实现目标检测的对象，是由特征文件 xml 进行构造的。在该软件中需要定位屏幕，因此需要以下几个特定的 xml 文件（OpenCV 中自带的已

经训练好的分类器特征文件)。

重写 Changed()函数，主要是初始化了 mat 数据，mat==二维的保存图像的矩阵,OpenCV 中自带的数据格式。

实现 processFrame 接口函数（整个模块的核心函数）：

- 图像预处理：解析出照片的 RGB 以及灰度图像格式，进行预处理过程（直方图的均衡化、降噪、灰度处理、锐化...），OpenCV 自带接口 Imgproc.equalizeHist();

减小尺寸 pyrDown (srcPic 源图像 Mat 对象, shrunkedPic 目标图像 Mat 对象); //加快运算速度

转化为灰度图 cvtColor (shrunkedPic, greyPic, COLOR_BGR2GRAY); //

中值滤波 medianBlur(greyPic, greyPic, 7); //参数三为奇数，值越大滤布越强

阈值化 threshold(greyPic, binPic, 80, 255, THRESH_BINARY) ;

- 边缘检测

Canny (binPic, cannyPic, cannyThr, cannyThr*ratio); //Canny 边缘检测, ratio= 2.5

- 提取轮廓

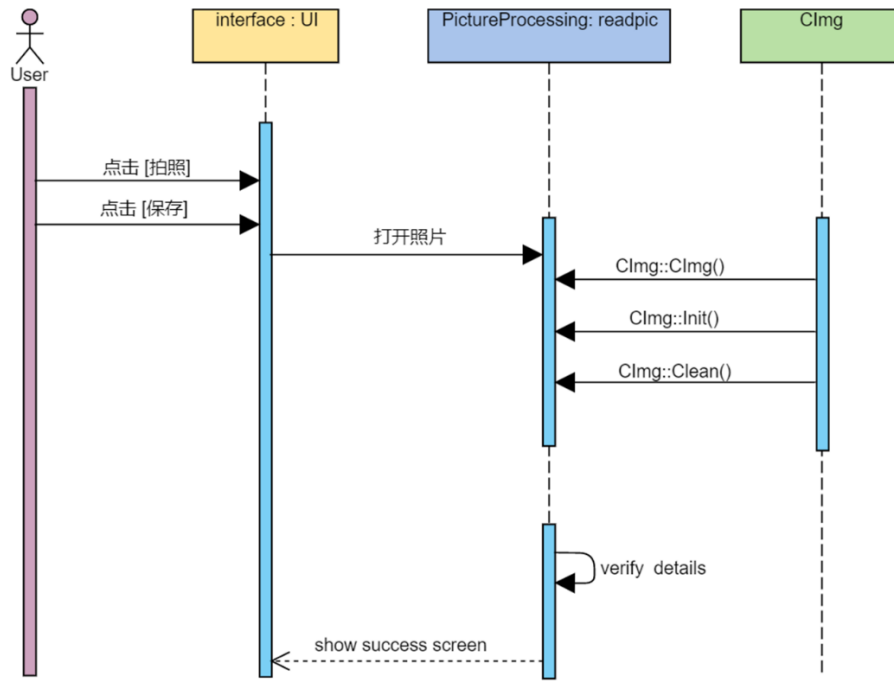
findContours (cannyPic, contours, hierarchy, RETR_CCOMP—检测所有的轮廓, CHAIN_APPROX_SIMPLE—仅保存轮廓的拐点信息);

//轮廓保存到 vector<vector<Point>> contours 轮廓集;

- 提取四个角的坐标

convexHull (polyContours[maxArea]—点集, hull—输出 xy 坐标, false—表示方向)

(6) 流程逻辑



(7) 接口

processFrame 接口函数为相机拍照模块和屏幕区域获取模块的接口。

(8) 注释设计

准备在本程序中安排的注释有：

- 模块首部注释说明本模块开始编写的时间，编写人员，及其基本功能；
- 加在各分枝点的注释说明获取屏幕区域所要具备的条件；
- 对图像变量进行说明，指出各参数的含义；
- 注释说明不同的情况对图像处理的规则不同，指出具体的计算方法。

(9) 限制条件

必须保证程序正常地连接到服务器；保证相机与控制器通信正常。

(10) 测试计划

进行不同亮度，不同格式的屏幕图片提取测试，验证结果的准确性。

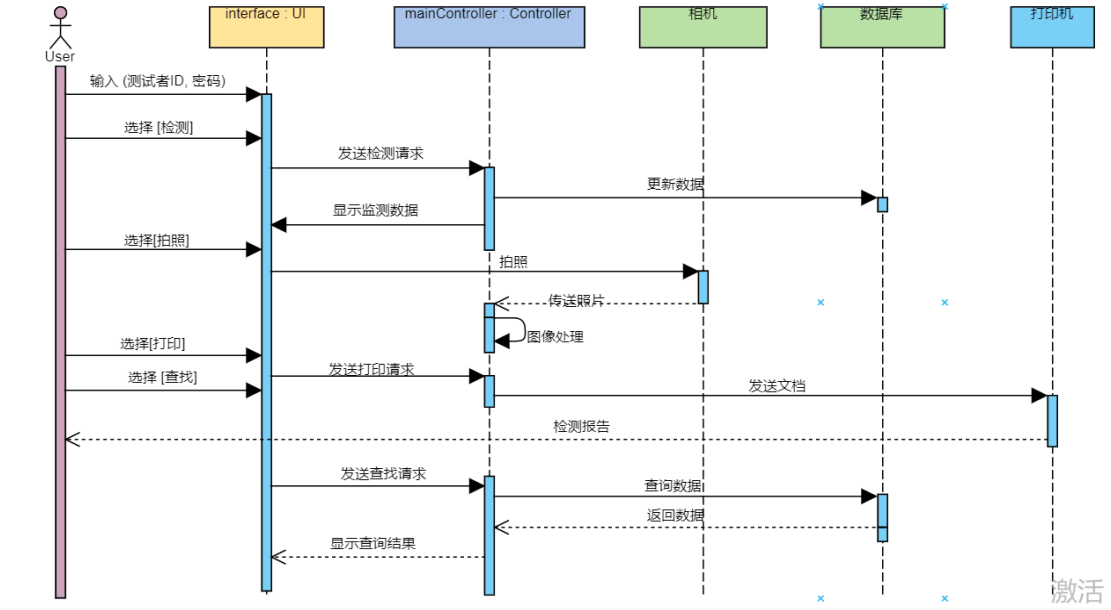
(11) 尚未解决的问题

算法需要不断改进，程序维护的难易程度未知。

四. 面向对象设计

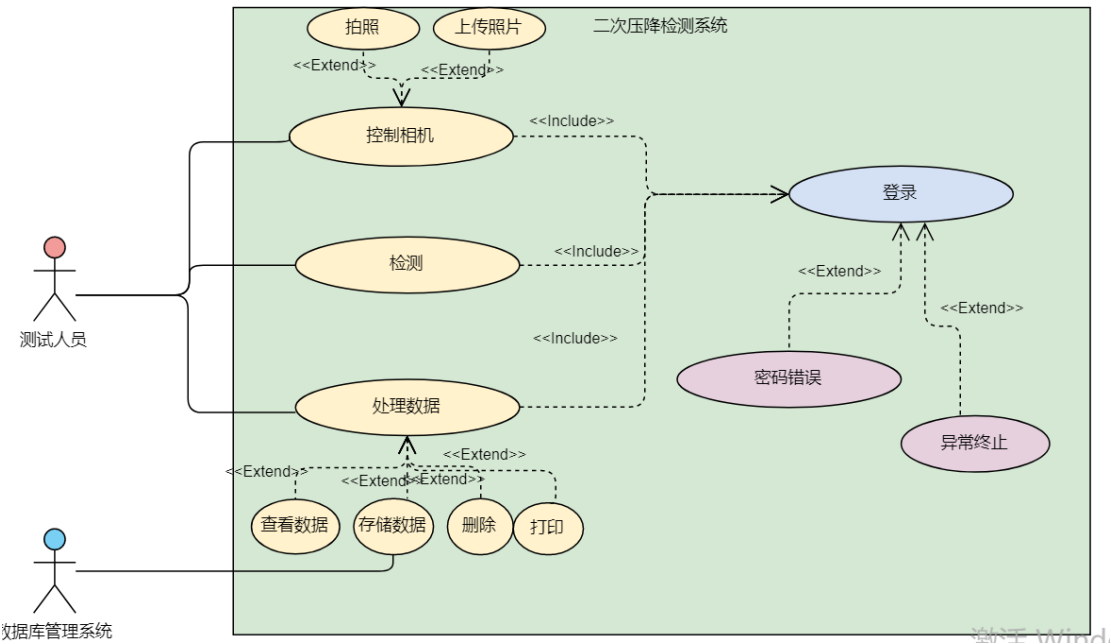
4.1 序列图

显示具体用例的详细流程和流程中不同对象之间的调用关系。



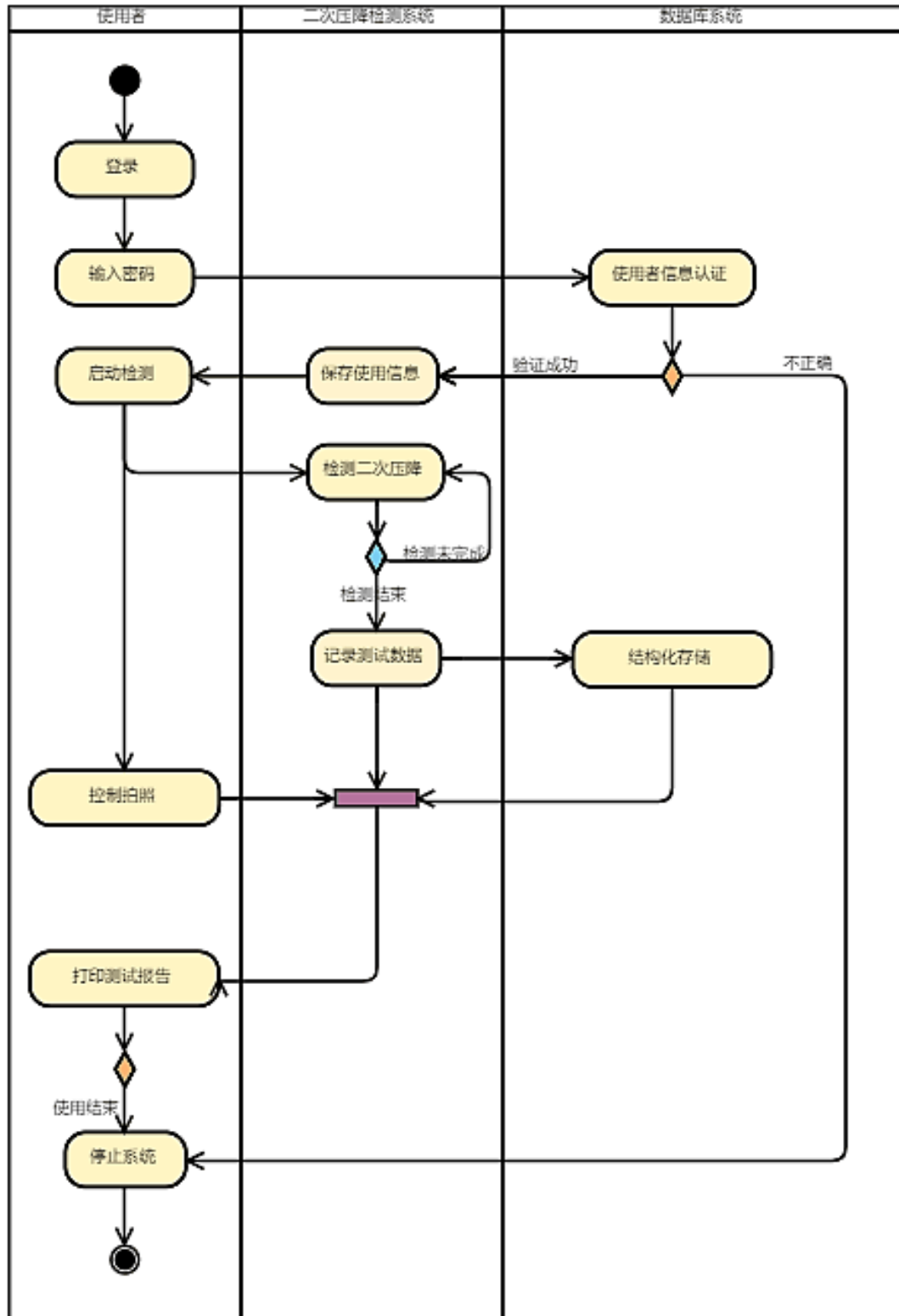
4.2 用例图

以一种可视化的方式理解系统的功能需求。二次降压检测系统与测试人员、组织或外部数据库系统进行交互的几种方案，帮助开发人员和用户从外部理解系统的功能



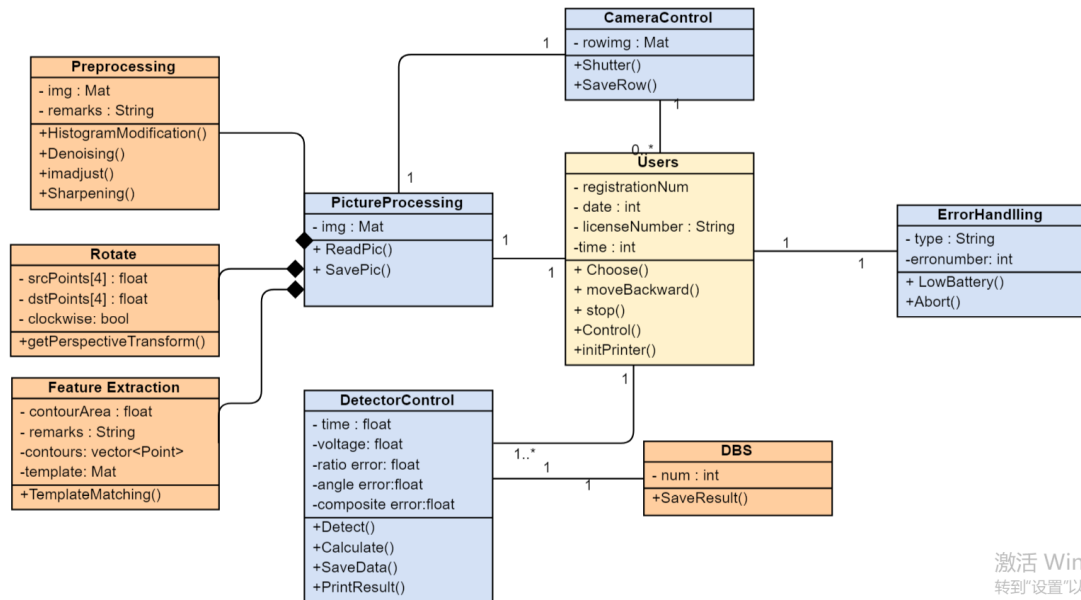
4.3 活动图

将二次压降检测进程以工作流的形式显示出来，操作由人、软件组件或计算机来执行。



4.4 类视图

显示了系统的静态结构。可用于表示逻辑类 CameraControl、ErrorHandling...,即业务人员所谈及的事物种类；实现类 Users, DBS, 即程序员处理的实体。



激活 Windows
转到“设置”以激活