摘要

近年来随着电商网络行业的新兴发达，越来越多的流媒体软件被开发出来，尤其以课程直播软件尤为数量众多，但是其中的质量良莠不齐，因此需要做更多的功能性测试，性能测试，稳定性测试等等；但由于人工操作繁复且效率不高，因此本课题围绕自动化测试系统展开，该系统主要包含了教室刷新，切换摄像头，麦克风，混音效，日志保存显示这几大功能。应用Python pywinauto的主要技术设计了测试系统，解决了人工测试的繁琐矛盾，以及考量了被测试系统的稳定性。真正做到了一键自动化，让测试不在成为枷锁的目标。本文主要介绍了系统的整个详细实现过程，包括工具的选型，界面的设计，功能模块的划分，系统的可行性研究，需求的设计拆分，系统技术实现，数据的配置，测试计划制定，系统测试。本系统较为简单，使用方便，运行较为稳定，实现了自动化测试的目标，减轻了负担。

第二章 系统需求分析

2.1. 用户需求的初步分析与实习课题的目标与任务

2.1.1. 功能需求分析

该系统主要对被测试软件进行自动化的测试，首先被测试系统主要包含登录，环境配置，教室刷新，切换摄像头，混音效的这几大功能，因此测试系统主要围绕着这几大功能展开，另外加运行时日志的写入，以及在测试系统上的实时显示。对于不同的用户来说他们选择的配置也可能不一样，因此对于不同的用户需要以数据驱动的方式来运行，因此就有了数据配置功能，因此对上述功能的实现是本系统的主要目标。

2.1.2. 软件运行环境要求

硬件环境要求

处理器：Intel Pentium V 以上

内存：4048 M以上推荐4048 M或更高 硬盘空间：50 G

软件环境要求：

操作系统：win7及以上系统

2.1.3. 性能需求分析

（1）系统的快速响应性

所开发的系统应该能及时响应用户的请求，对于运行繁多的测试用例来说不应占用过多的系统资源并能在用户允许的时间内快速响应，已达到性能良好的目标。

1. 系统易用性

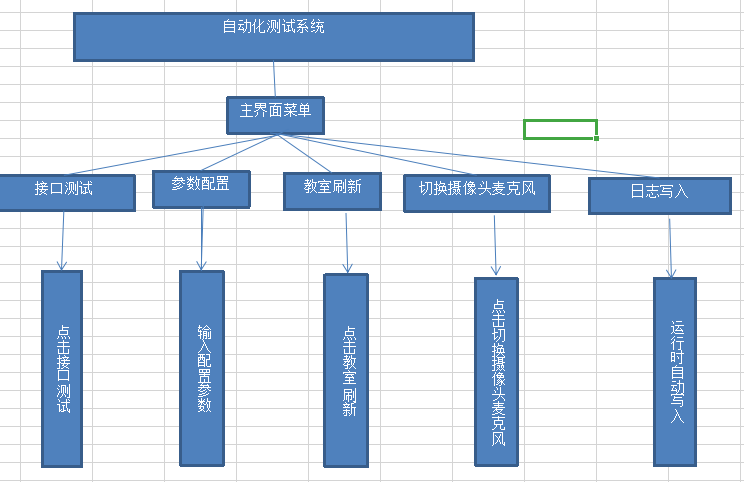
对于很多人用的测试系统而言， 所开发的系统应做到操作简便，尽量使系统操作不受用户对电脑知识水平的限制，以及软件的安装步骤尽量减少,做到一键安装。

1. 系统的可维护性

由于系统涉及的信息比较广，数据库中的数据需定期修改，系统可利用的空间及性能也随之下降，为了使系统更好地运转，可以对系统数据及一些简单的功能进行独立的维护及调整

2.1.4 业务流程分析

经过仔细的分析考量可将整个测试系统的流程划分为以下的流程：



该流程图为整个测试系统的总体流程图，用户进入界面后可以看到接口测试，参数配置，教室刷新，切换摄像头麦克风，日志写入这几大功能菜单。用户可以根据自己的需求来方便的做出自己的选择。

2.2. 可行性研究

2.2.1. 技术可行性

本测试系统win7系统，并采用Python pyqt5 pywinauto py xautogui 技术实现， 作者对于Python语言，界面设知识计都有一定的技术基础，由于是面向广大用户的因此本系统采用pc客户端的方式进行开发，因此实现系统从技术层面来说是完全可行的。

2.2.2. 经济可行性

由于Python以及相应的软件包都是开源的因此并不需要耗费经济费用，。另外从系统的复杂程度上看也不需要更多的人力耗费，综上所述从经济方面来看还是比较容易实现的，完全满足经济上的需求。

第三章 系统设计与实现

3.1. 系统设计

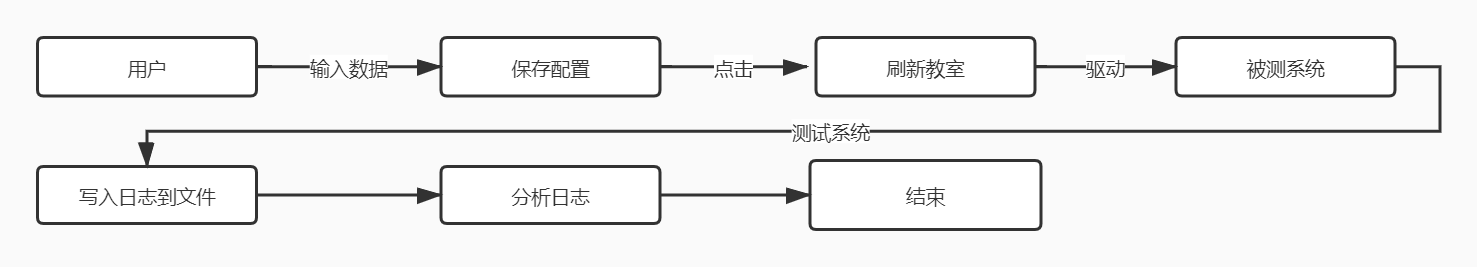
### 3.1.1系统工具语言选型

* [Winium.Desktop](https://link.zhihu.com/?target=https://github.com/2gis/Winium.Desktop/" \t "https://zhuanlan.zhihu.com/p/_blank) - 测试Windows应用(主要是基于WinForms和WPF平台)的自动化测试工具. 实现了Selenium Remote WebDriver协议
* [PyAutoGUI](https://link.zhihu.com/?target=https://pypi.python.org/pypi/PyAutoGUI" \t "https://zhuanlan.zhihu.com/p/_blank) - 跨平台的GUI测试工具，支持通过python脚本控制键盘和鼠标
* [robotframework-autoitlibrary](https://link.zhihu.com/?target=https://code.google.com/p/robotframework-autoitlibrary/" \t "https://zhuanlan.zhihu.com/p/_blank) - Robot Framework的windows GUI测试扩展
* [autopy](https://link.zhihu.com/?target=https://github.com/msanders/autopy" \t "https://zhuanlan.zhihu.com/p/_blank) - 简单跨平台的GUI测试工具集
* [UISoup](https://link.zhihu.com/?target=https://pypi.python.org/pypi/UISoup/" \t "https://zhuanlan.zhihu.com/p/_blank) - 支持windows和MacOS平台的UI自动化(仅工作在x86平台)
* [pywinauto](https://link.zhihu.com/?target=http://pywinauto.github.io/" \t "https://zhuanlan.zhihu.com/p/_blank) - 非常有python面向对象风格的GUI测试库，现已支持64位机器以及py2和py3
* [SikuliX](https://link.zhihu.com/?target=http://sikulix.com/" \t "https://zhuanlan.zhihu.com/p/_blank) - 基于OpenCV的 GUI 测试框架, 使用图片识别技术，支持python2.7

从以上的工具上看pywinauto 尤其适合桌面软件自动化，故而选择其做自动化。

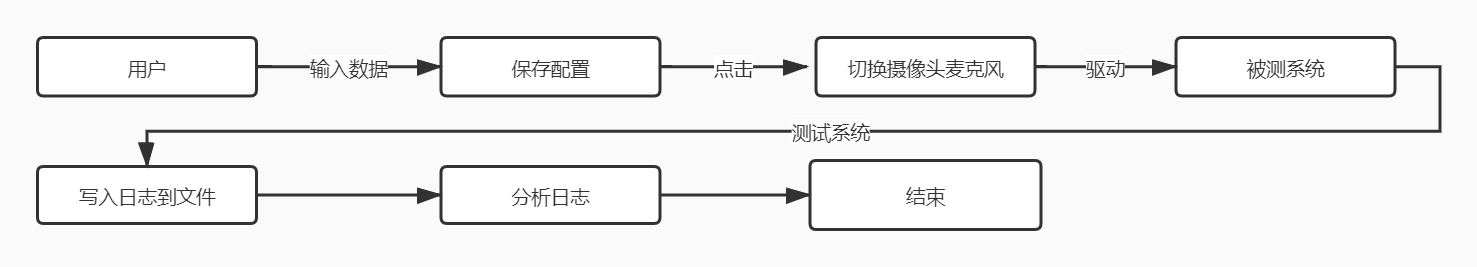
3.1.2刷新教室功能设计

刷新教室主要流程设计下图：



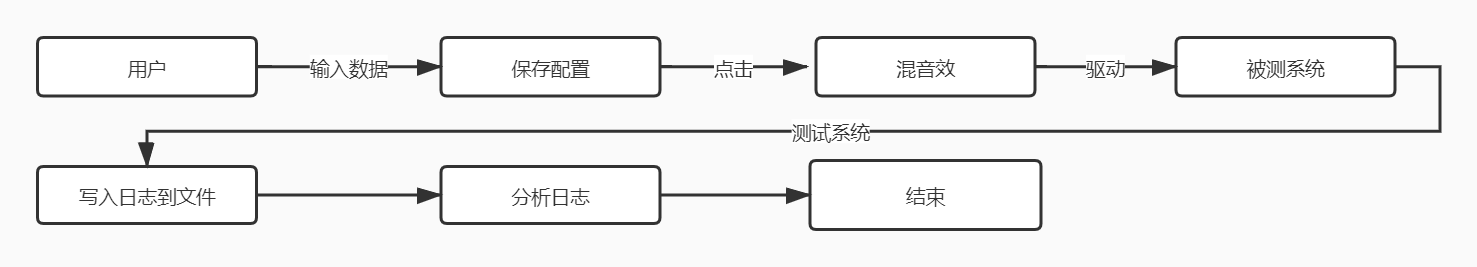
3.1.3切换功能设计

切换功能流程设计如下：



3.1.4混音效功能设计

混音功能流程设计如下：



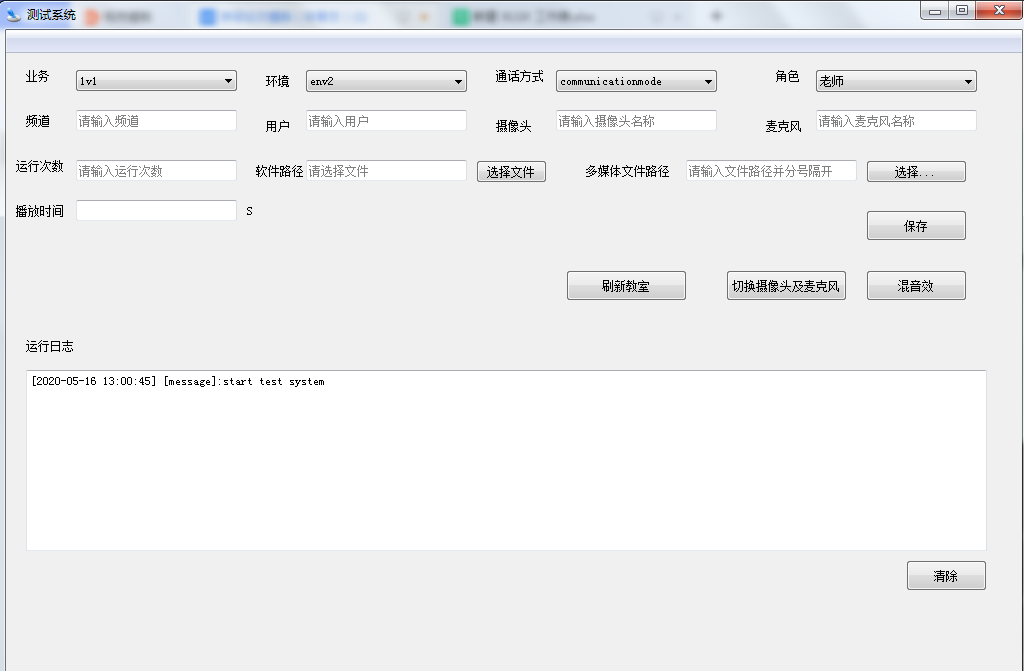
3.2.系统实现

3.2.1模块拆分

根据其用户需求分析主要分为参数配置，教室刷新，切换摄像头麦克风，日志体统这几大模块

3.2.2 界面布局

界面布局如下所示：



其中实现布局代码如下所示：

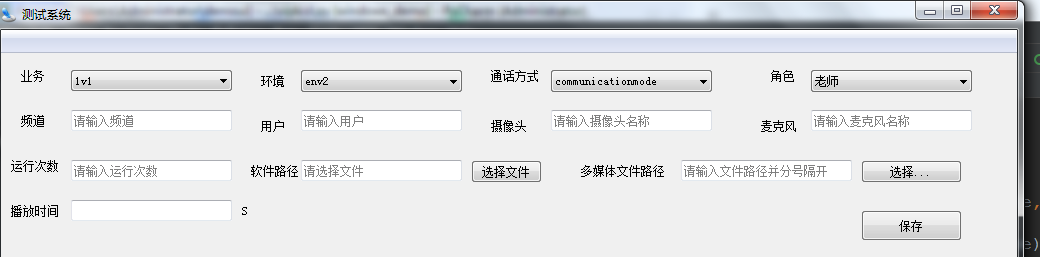
class Ui\_MainWindow(QWidget):  
  
 def \_\_init\_\_(self,parent=None):  
 super(Ui\_MainWindow, self).\_\_init\_\_(parent)  
 self.Widget=QtWidgets.QWidget()  
 self.setupUi(self.Widget)  
 sys.stdout = Stream(newText=self.onUpdateText)  
  
 def setupUi(self, MainWindow):  
 MainWindow.setObjectName("MainWindow")  
 MainWindow.resize(1016, 898)  
 self.centralwidget = QtWidgets.QWidget(MainWindow)  
 self.centralwidget.setObjectName("centralwidget")  
 self.label\_19 = QtWidgets.QLabel(self.centralwidget)  
 self.label\_19.setGeometry(QtCore.QRect(20, 310, 54, 12))  
 self.label\_19.setObjectName("label\_19")  
 self.frame = QtWidgets.QFrame(self.centralwidget)  
 self.frame.setGeometry(QtCore.QRect(0, 0, 1011, 221))  
 self.frame.setStyleSheet("")  
 self.frame.setFrameShape(QtWidgets.QFrame.StyledPanel)  
 self.frame.setFrameShadow(QtWidgets.QFrame.Raised)  
 self.frame.setObjectName("frame")  
 self.label\_28 = QtWidgets.QLabel(self.frame)  
 self.label\_28.setGeometry(QtCore.QRect(10, 0, 54, 12))  
 self.label\_28.setObjectName("label\_28")  
 self.label\_25 = QtWidgets.QLabel(self.frame)  
 self.label\_25.setGeometry(QtCore.QRect(10, 120, 61, 31))  
 self.label\_25.setObjectName("label\_25")  
 self.label\_27 = QtWidgets.QLabel(self.frame)  
 self.label\_27.setGeometry(QtCore.QRect(770, 30, 31, 31))  
 self.label\_27.setObjectName("label\_27")  
 self.comboBox\_6 = QtWidgets.QComboBox(self.frame)  
 self.comboBox\_6.setGeometry(QtCore.QRect(550, 40, 161, 22))  
 self.comboBox\_6.setObjectName("comboBox\_6")  
 self.comboBox\_6.addItem("")  
 self.comboBox\_6.addItem("")  
 self.lineEdit\_13 = QtWidgets.QLineEdit(self.frame)  
 self.lineEdit\_13.setGeometry(QtCore.QRect(550, 80, 161, 21))  
 self.lineEdit\_13.setObjectName("lineEdit\_13")  
 self.lineEdit\_13.setPlaceholderText("请输入摄像头名称")  
  
 self.comboBox\_11 = QtWidgets.QComboBox(self.frame)  
 self.comboBox\_11.setGeometry(QtCore.QRect(810, 40, 161, 22))  
 self.comboBox\_11.setObjectName("comboBox\_11")  
 self.comboBox\_11.addItem("")  
 self.comboBox\_11.addItem("")  
 self.comboBox\_11.addItem("")  
 self.lineEdit\_5 = QtWidgets.QLineEdit(self.frame)  
 self.lineEdit\_5.setGeometry(QtCore.QRect(300, 80, 161, 21))  
 self.lineEdit\_5.setObjectName("lineEdit\_5")  
 self.lineEdit\_5.setPlaceholderText("请输入用户")  
  
 self.label\_8 = QtWidgets.QLabel(self.frame)  
 self.label\_8.setGeometry(QtCore.QRect(260, 80, 41, 31))  
 self.label\_8.setObjectName("label\_8")  
 self.label\_23 = QtWidgets.QLabel(self.frame)  
 self.label\_23.setGeometry(QtCore.QRect(490, 80, 51, 31))  
 self.label\_23.setObjectName("label\_23")  
 self.lineEdit\_4 = QtWidgets.QLineEdit(self.frame)  
 self.lineEdit\_4.setGeometry(QtCore.QRect(70, 80, 161, 21))  
 self.lineEdit\_4.setObjectName("lineEdit\_4")  
 self.lineEdit\_4.setPlaceholderText("请输入频道")  
  
 self.lineEdit\_15 = QtWidgets.QLineEdit(self.frame)  
 self.lineEdit\_15.setGeometry(QtCore.QRect(70, 130, 161, 21))  
 self.lineEdit\_15.setObjectName("lineEdit\_15")  
 self.lineEdit\_15.setPlaceholderText("请输入运行次数")  
  
 self.lineEdit\_14 = QtWidgets.QLineEdit(self.frame)  
 self.lineEdit\_14.setGeometry(QtCore.QRect(810, 80, 161, 21))  
 self.lineEdit\_14.setObjectName("lineEdit\_14")  
 self.lineEdit\_14.setPlaceholderText("请输入麦克风名称")  
  
 self.label\_11 = QtWidgets.QLabel(self.frame)  
 self.label\_11.setGeometry(QtCore.QRect(490, 30, 61, 31))  
 self.label\_11.setObjectName("label\_11")  
 self.label\_20 = QtWidgets.QLabel(self.frame)  
 self.label\_20.setGeometry(QtCore.QRect(20, 30, 41, 31))  
 self.label\_20.setObjectName("label\_20")  
 self.comboBox\_10 = QtWidgets.QComboBox(self.frame)  
 self.comboBox\_10.setGeometry(QtCore.QRect(300, 40, 161, 22))  
 self.comboBox\_10.setObjectName("comboBox\_10")  
 self.comboBox\_10.addItem("")  
 self.comboBox\_10.addItem("")  
 self.comboBox\_10.addItem("")  
 self.label\_9 = QtWidgets.QLabel(self.frame)  
 self.label\_9.setGeometry(QtCore.QRect(20, 80, 31, 21))  
 self.label\_9.setObjectName("label\_9")  
 self.label\_21 = QtWidgets.QLabel(self.frame)  
 self.label\_21.setGeometry(QtCore.QRect(260, 40, 41, 21))  
 self.label\_21.setObjectName("label\_21")  
 self.label\_24 = QtWidgets.QLabel(self.frame)  
 self.label\_24.setGeometry(QtCore.QRect(760, 80, 51, 31))  
 self.label\_24.setObjectName("label\_24")  
 self.pushButton\_4 = QtWidgets.QPushButton(self.frame)  
 self.pushButton\_4.setGeometry(QtCore.QRect(860, 180, 101, 31))  
 self.pushButton\_4.setObjectName("pushButton\_4")  
  
 self.comboBox\_12 = QtWidgets.QComboBox(self.frame)  
 self.comboBox\_12.setGeometry(QtCore.QRect(70, 40, 161, 21))  
 self.comboBox\_12.setObjectName("comboBox\_12")  
 self.comboBox\_12.addItem("")  
 self.comboBox\_12.addItem("")  
 self.comboBox\_12.addItem("")  
 self.comboBox\_12.addItem("")  
 self.comboBox\_12.addItem("")  
 self.label\_10 = QtWidgets.QLabel(self.frame)  
 self.label\_10.setGeometry(QtCore.QRect(250, 130, 51, 21))  
 self.label\_10.setObjectName("label\_10")  
  
 self.lineEdit\_6 = QtWidgets.QLineEdit(self.frame)  
 self.lineEdit\_6.setGeometry(QtCore.QRect(300, 130, 161, 21))  
 self.lineEdit\_6.setObjectName("lineEdit\_6")  
 self.lineEdit\_6.setPlaceholderText("请选择文件")  
  
 self.pushButton\_5 = QtWidgets.QPushButton(self.frame)  
 self.pushButton\_5.setGeometry(QtCore.QRect(470, 130, 71, 23))  
 self.pushButton\_5.setObjectName("pushButton\_5")  
 self.pushButton\_5.clicked.connect(self.getFiles)  
  
  
  
 self.lineEdit\_7 = QtWidgets.QLineEdit(self.frame)  
 self.lineEdit\_7.setGeometry(QtCore.QRect(680, 130, 171, 21))  
 self.lineEdit\_7.setObjectName("lineEdit\_7")  
 self.lineEdit\_7.setPlaceholderText("请输入文件路径并分号隔开")  
 self.label\_12 = QtWidgets.QLabel(self.frame)  
 self.label\_12.setGeometry(QtCore.QRect(580, 130, 91, 21))  
 self.label\_12.setObjectName("label\_12")  
  
 self.label\_26 = QtWidgets.QLabel(self.frame)  
 self.label\_26.setGeometry(QtCore.QRect(10, 170, 61, 21))  
 self.label\_26.setObjectName("label\_26")  
 self.lineEdit\_16 = QtWidgets.QLineEdit(self.frame)  
 self.lineEdit\_16.setGeometry(QtCore.QRect(70, 170, 161, 21))  
 self.lineEdit\_16.setObjectName("lineEdit\_16")  
 self.label\_29 = QtWidgets.QLabel(self.frame)  
 self.label\_29.setGeometry(QtCore.QRect(241, 170, 21, 21))  
 self.label\_29.setObjectName("label\_29")  
 self.pushButton\_8 = QtWidgets.QPushButton(self.frame)  
 self.pushButton\_8.setGeometry(QtCore.QRect(860, 130, 101, 23))  
 self.pushButton\_8.setObjectName("pushButton\_8")  
 self.pushButton\_8.clicked.connect(self.getMusicfiles)  
 self.pushButton = QtWidgets.QPushButton(self.centralwidget)  
 self.pushButton.setGeometry(QtCore.QRect(560, 240, 121, 31))  
 self.pushButton.setObjectName("pushButton")  
 self.pushButton.clicked.connect(self.FreshRoomRoom)  
  
 self.pushButton\_2 = QtWidgets.QPushButton(self.centralwidget)  
 self.pushButton\_2.setGeometry(QtCore.QRect(720, 240, 121, 31))  
 self.pushButton\_2.setObjectName("pushButton\_2")  
  
  
 self.pushButton\_3 = QtWidgets.QPushButton(self.centralwidget)  
 self.pushButton\_3.setGeometry(QtCore.QRect(860, 240, 101, 31))  
 self.pushButton\_3.setObjectName("pushButton\_3")  
 self.textEdit = QtWidgets.QTextEdit(self.centralwidget)  
 self.textEdit.setGeometry(QtCore.QRect(20, 340, 961, 181))  
 self.textEdit.setAutoFillBackground(False)  
 self.textEdit.setObjectName("textEdit")  
 self.pushButton\_6 = QtWidgets.QPushButton(self.centralwidget)  
 self.pushButton\_6.setGeometry(QtCore.QRect(900, 530, 81, 31))  
 self.pushButton\_6.setObjectName("pushButton\_6")  
  
  
 self.menubar = QtWidgets.QMenuBar(MainWindow)  
 self.menubar.setGeometry(QtCore.QRect(0, 0, 1016, 23))  
 self.menubar.setObjectName("menubar")  
  
 self.statusbar = QtWidgets.QStatusBar(MainWindow)  
 self.statusbar.setObjectName("statusbar")  
  
 self.retranslateUi(MainWindow)  
 QtCore.QMetaObject.connectSlotsByName(MainWindow)  
  
 def retranslateUi(self, MainWindow):  
 \_translate = QtCore.QCoreApplication.translate  
 MainWindow.setWindowTitle(\_translate("MainWindow", "测试系统"))  
 MainWindow.setWindowIcon(QIcon(':/images/image.ico'))  
 self.label\_19.setText(\_translate("MainWindow", "运行日志"))  
 self.label\_28.setText(\_translate("MainWindow", "通用配置"))  
 self.label\_25.setText(\_translate("MainWindow", "运行次数"))  
 self.label\_27.setText(\_translate("MainWindow", "角色"))  
 self.comboBox\_6.setItemText(0, \_translate("MainWindow", "communicationmode"))  
 self.comboBox\_6.setItemText(1, \_translate("MainWindow", "livemode"))  
 self.comboBox\_11.setItemText(0, \_translate("MainWindow", "老师"))  
 self.comboBox\_11.setItemText(1, \_translate("MainWindow", "学生"))  
 self.comboBox\_11.setItemText(2, \_translate("MainWindow", "监课"))  
 self.label\_8.setText(\_translate("MainWindow", "用户"))  
 self.label\_23.setText(\_translate("MainWindow", "摄像头"))  
 self.label\_11.setText(\_translate("MainWindow", "通话方式"))  
 self.label\_20.setText(\_translate("MainWindow", "业务"))  
 self.comboBox\_10.setItemText(0, \_translate("MainWindow", "env2"))  
 self.comboBox\_10.setItemText(1, \_translate("MainWindow", "env5"))  
 self.comboBox\_10.setItemText(2, \_translate("MainWindow", "env6"))  
  
 self.label\_9.setText(\_translate("MainWindow", "频道"))  
 self.label\_21.setText(\_translate("MainWindow", "环境"))  
 self.label\_24.setText(\_translate("MainWindow", "麦克风"))  
 self.pushButton\_4.setText(\_translate("MainWindow", "保存"))  
 self.pushButton\_4.clicked.connect(self.save\_setting)  
 self.comboBox\_12.setItemText(0, \_translate("MainWindow", "1v1"))  
 self.comboBox\_12.setItemText(1, \_translate("MainWindow", "双优class"))  
 self.comboBox\_12.setItemText(2, \_translate("MainWindow", "蜂校测试"))  
 self.comboBox\_12.setItemText(3, \_translate("MainWindow", "小班class"))  
 self.comboBox\_12.setItemText(4, \_translate("MainWindow", "微米测试"))  
 self.label\_10.setText(\_translate("MainWindow", "软件路径"))  
  
 self.pushButton\_5.setText(\_translate("MainWindow", "选择..."))  
 self.label\_12.setText(\_translate("MainWindow", "多媒体文件路径"))  
 self.label\_26.setText(\_translate("MainWindow", "播放时间"))  
 self.label\_29.setText(\_translate("MainWindow", "S"))  
 self.pushButton\_8.setText(\_translate("MainWindow", "选择..."))  
  
 # 打开file 选择框  
 self.pushButton\_5.setText(\_translate("MainWindow", "选择文件"))  
 self.pushButton.setText(\_translate("MainWindow", "刷新教室"))  
 self.pushButton\_2.setText(\_translate("MainWindow", "切换摄像头及麦克风"))  
 self.pushButton\_2.clicked.connect(self.switchcaOrMicrophone)  
 self.pushButton\_3.setText(\_translate("MainWindow", "混音效"))  
 self.pushButton\_6.setText(\_translate("MainWindow", "清除"))  
 self.label\_12.setText(\_translate("MainWindow", "多媒体文件路径"))  
 self.pushButton\_3.clicked.connect(self.audioEffectMix)  
 self.pushButton\_6.clicked.connect(self.logClear)  
 def onUpdateText(self, text):  
 *"""Write console output to text widget."""* cursor = self.textEdit.textCursor()  
 cursor.movePosition(QTextCursor.End)  
 cursor.insertText(text)  
 self.textEdit.setTextCursor(cursor)  
 self.textEdit.ensureCursorVisible()

其中用pyqt5的开发步骤如下：

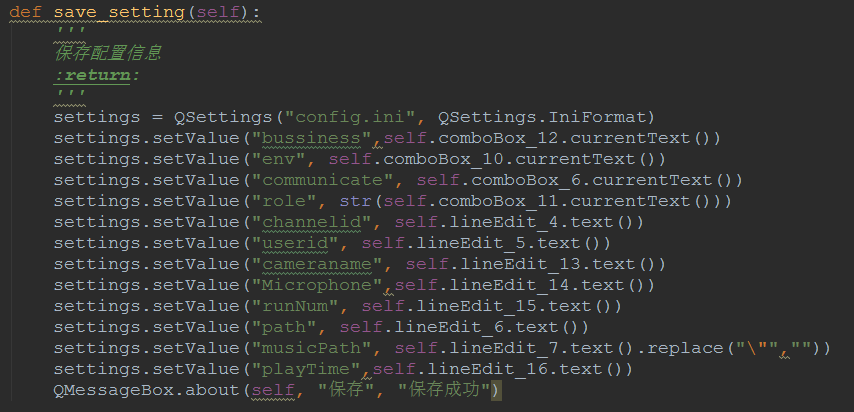
1. 利用qtdesjgner 创建gui；
2. 在文件中添加各种需要的控件，并修改控件的属性，保存为ui文件；
3. 通过pyuic5将ui文件转换为Python脚本，并对其进行修改；其中命令为：pyuic5 -o xxx.py xxx.ui
4. 对于ui上的控件编写绑定函数，触发信号以及槽函数，利用connect 进行绑定；

3.2.3参数配置实现

用户打开软件后，直接填写相关的参数配置，点击保存即可。具体实现界面如下所示：



用户可以填写相关的业务参数，相关数据保存在目录下的config.ini,如下部分代码所示

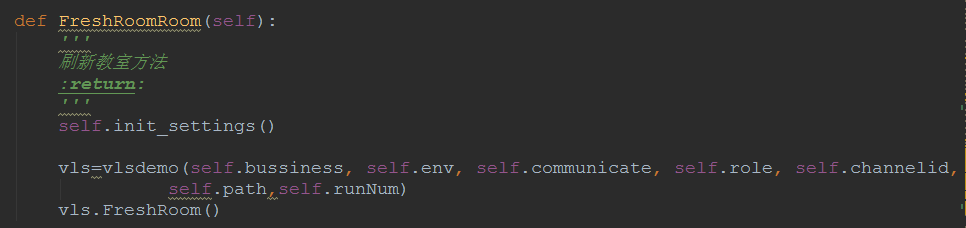


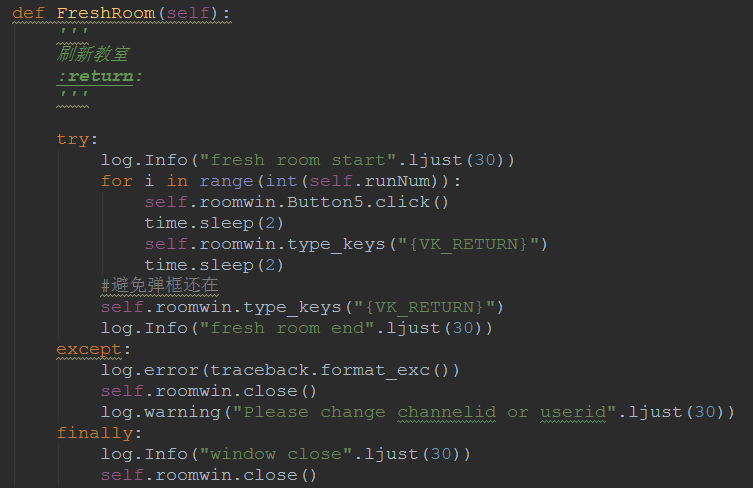
这段代码就是获取用户输入的参数，保存成功后给用户提示。

3.2.4教室刷新

用户打开软件后，直接填写相关的参数配置并保存后，点击教室刷新就可以进行对被测试系统的测试了，其中界面设计如下：

其中部分逻辑代码实现如下所示：



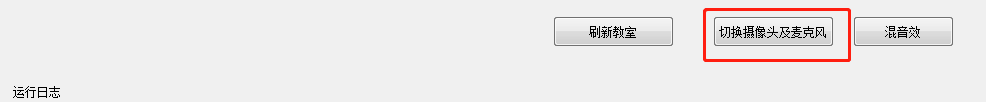


这段代码就是首先加载config.ini里边的配置信息，然后将必要的参数传给FreshRoom的方法，然后运行此方法已达到对被测试软件的刷新教室的目标。另外运行次数不宜设置的过大，考虑到被测试软件系统的稳定性，设置过大容易崩溃。

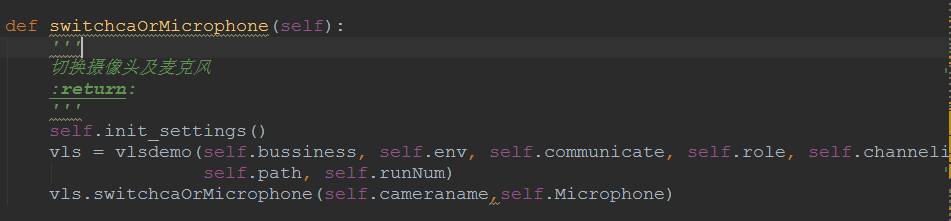
3.2.5切换摄像头麦克风

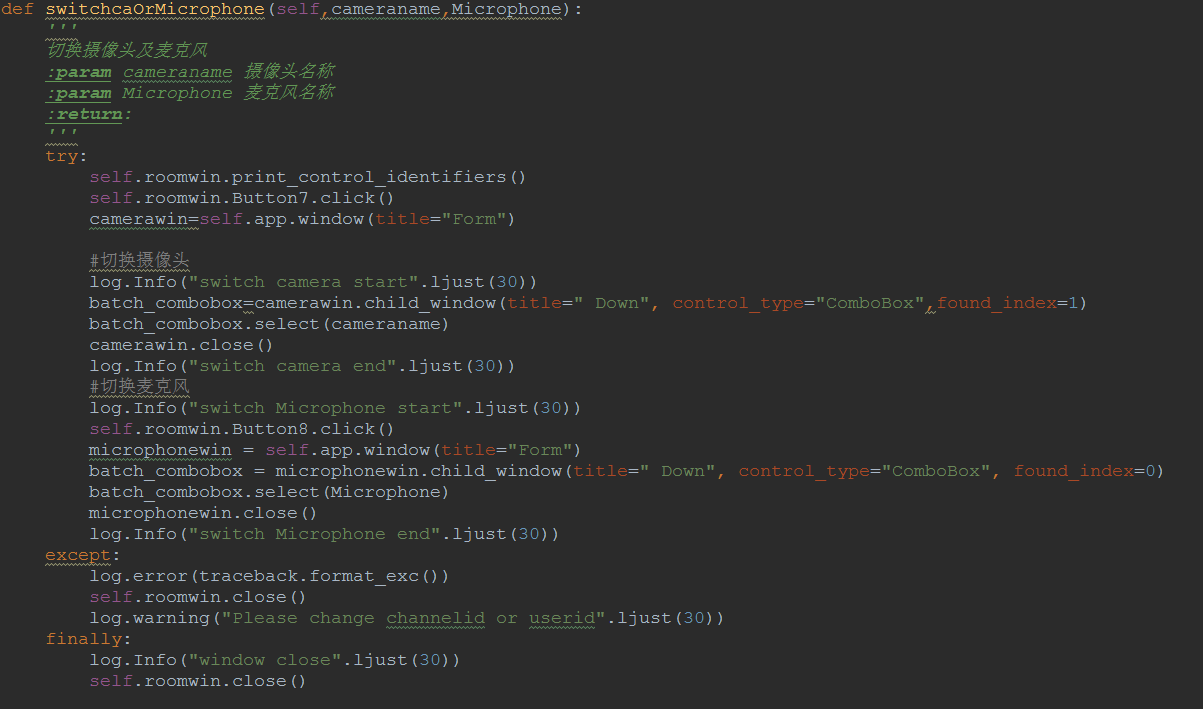
用户打开软件后，直接填写相关的参数配置并保存后，点击切换摄像头麦克风

就可以进行对被测试系统的测试了，其中界面设计如下：



其中部分逻辑代码实现如下所示：



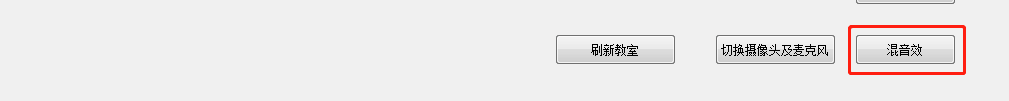


这段代码也是首先加载config.ini里边的配置信息，然后将必要的参数传给switchcaOrMicrophone的方法，然后运行此方法已达到对被测试软件摄像头麦克风的切换。

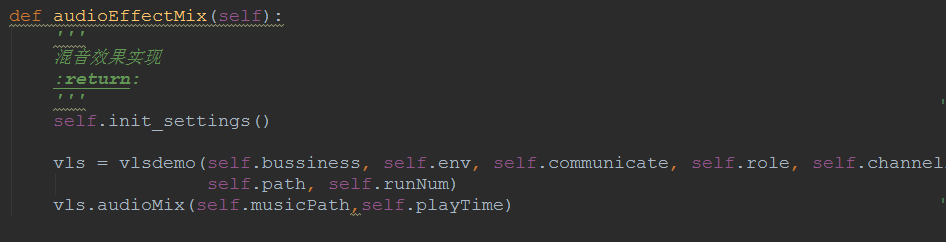
3.2.6 混音效的实现

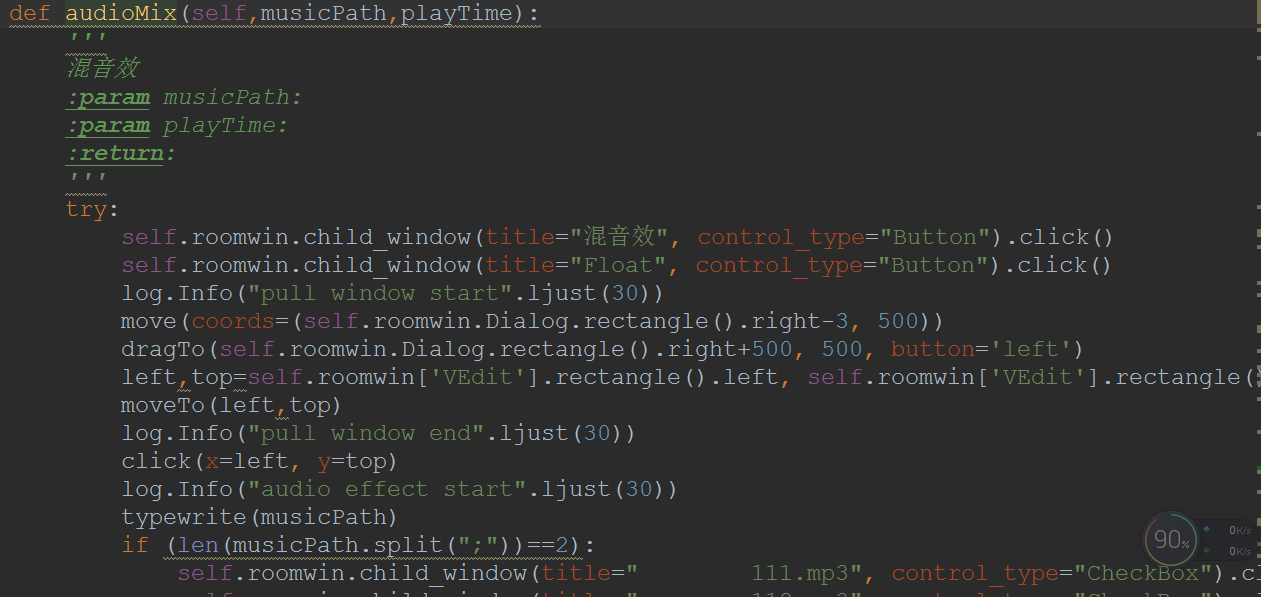
用户打开软件后，直接填写相关的参数配置并保存后，点击混音效

就可以进行对被测试系统的混音测试了，其中界面设计如下：



其中部分代码实现如下：

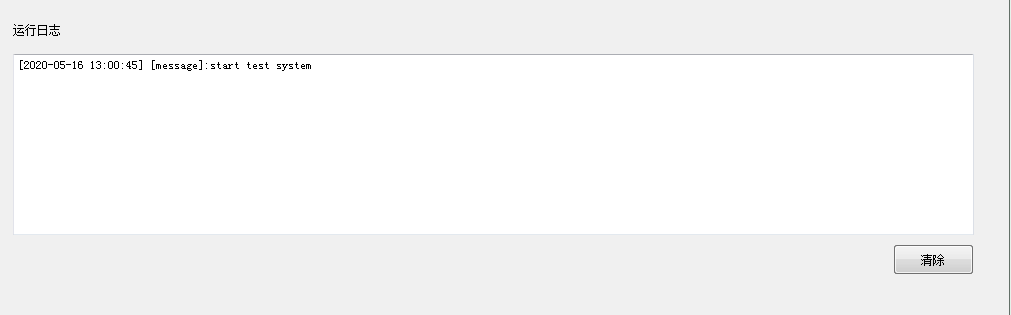




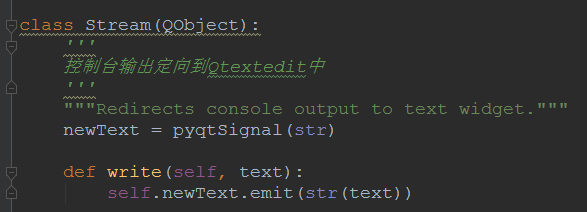
这段代码也是首先加载config.ini里边的配置信息，然后将必要的参数传给audioMix的方法，然后运行此方法已达到对被测试软件混音效的实现，其中在调试过程中pywinauto多文本框的拉伸不支持是个难点，最后借助了pyaugui支持拖拽拉伸的方法得以实现效果。

3.2.6日志实现

用户打开软件后，直接填写相关的参数配置并保存后，地啊你其中某个功能其中的运行日志会实时写入到界面中和日志文件中，其中界面设计如下：



日志系统的部分逻辑代码如下：





在此用到了pyqt5的槽函数，用于将打印的日志直接给显示在文本框中。

3.3.系统测试

软件产业不过短短的几十年，整个行业对软件测试的依赖性也越来越大，但由于各个软件的质量并不那么完美因此需要我们对软件进行测试以保证满足用户的需求。一般来说软件测试的方式主要分为以下几种方式：

1.公司内部专业测试人员进行的功能性测试和性能测试，这种方式由于内部人员经过软件需求评审对其功能非常熟悉，因此更能发现其中的问题，相对来说是一种经济的方式；

2.一种是将发动大量的用户一起进行测试，用于发现软件中存在的问题；此种方式目的性就没那么强，主要是会搜集用户的反馈意见以便对软件进行必要的修改，优化等。

3.这个是将软件测试这个任务发送给第三方，第三方进行软件测试，功能性能压力等方面的验证。不过这种方法来说由于第三方对于软件功能的不熟悉，所以在沟通上会有较大的花费，另外对于公司的经济也是一种考验。

本系统是纯粹的客户端的方式，客户端运行于win7 及以上系统环境，目前采用的测试方式主要有：

1. 基本的功能性测试

客户端采用基本的测试方式（黑盒测试，对于代码层级的可以进行单元测试覆盖）

1. 易用性测试

由于本系统是客户端开发，因此对于GUI的操作测试尤为重要 ，下面是列举的测试中要考虑的问题：

1.**鼠标和菜单操作** 菜单条，工具条在合适的环境下是否状态正常显示和工作；菜单是否会随着当前的操作变亮或者置灰；界面的各个控件是否有序排列正常显示；其中每个控件的绑定时间能否正常触发测试；如果多次单击鼠标点击控件是否会出现卡顿现象；

2.**窗口的操作** 窗口变大，缩小、移动、拉伸拖拽；能否正常响应相关的输入或菜单命令；窗体中的数据能否用鼠标，键盘，功能键，方向箭头操作；相关的下拉工具条，滚动条是否能够正常使用，如果进行多次的点击是否会出现崩溃现象。

3.**数据操作** 数据输入控件是否正常显示，对于数据的提示是否正常；对于输入的不合理数据是否有判断识别，图形界面是否正常工作；数据输入消息是否得到正常理解。

1. 性能测试

主要是进行对软件的压力测试,以用来确定软件最大承受能力。压力测试一般暴露的问题有内存泄漏，性能问题，系统资源消耗过大问题以及磁盘空间不足的问题

### 3.3.1 测试环境

测试环境主要是运行客户端，其在Windows系统上，具体如下表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 名称 | 台数 | 型号 | cpu | 内存 | 操作系统 |
| 客户端 | 1 | 华硕 | i5 3.2GHz | 8G | Win7 UItimate |

### 3.3.2 测试用例

3.3.2.1 教室刷新测试

此功能开始经过几次调用日志并无异常显示，功能正常，但将运行次数曾大以后被测试软件会出现崩溃现象，经过仔细的查看以及调试，加一些线程等待，功能稳定运行无异常

3.3.2.2 切换摄像头测试

此功能测试需要安装外接摄像头，麦克风设备，进行测试时必须输入外接摄像头，麦克风名称，在测试过程中功能较为稳定并无异常。

3.3.2.3 混音功能测试

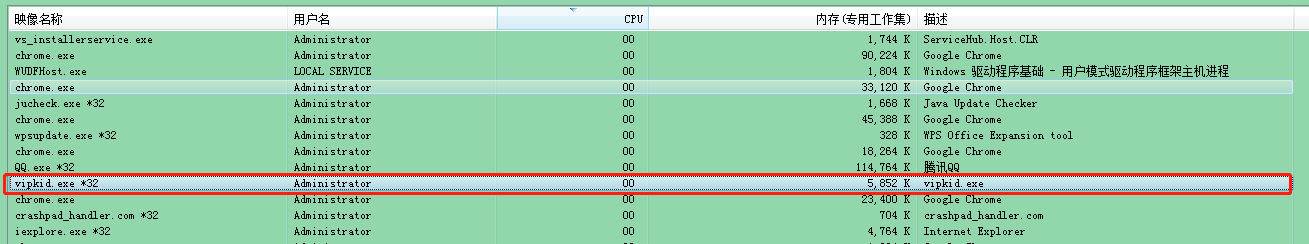
此功能需要下载MP3文件，在进行混音测试时，需要进行选择，测试过程中，由于原先对于弹框的定位坐标设置固定，这就导致了在不同分辨率的机器上时，窗口无法拉伸，输入数据，查找原因是pywinauto是对支持窗口的拉伸不支持，因此经过多方查找资料利用pyautogui配合其做自动化，因此解决了问题，在此后的测试过程中功能也达到稳定的标准。

### 3.3.3 测试结果分析

测试结果主要从以下几个方面的分析：

1. cpu、内存占用情况

检查后台应用进程占用系统内存的比例，由于是在Windows机器上，故而直接查看任务管理器即可，如下图所示，测试系统占用内存约为5M,系统资源占用较少，符合使用要求。



1. 各个模块运行情况

通过查看写入的日志，查看运行过程是否

第五章 项目管理

## 5.1 项目计划

## 5.2 项目执行

## 5.3项目管理能力培养

参考文献

1. Dorothy Graham．REQUIREMENTS：Requirements and testing：Seven missing—link myths．IEEE Software，2002， 19(5)：15—17
2. 朱少民.软件测试方法和技术.北京：清华大学出版社，2010
3. 王蕾. 工业软件研发，测试与质量管理论丛 .上海：社会科学院出版社，2012
4. 许育诚.软件测试与质量管理.水利电力出版社，2004
5. 谢敏，戴金龙.软件测试系列报道之二——追踪每一个软件缺陷.计算机世界，2005
6. 陈能技.软件测试技术大全.北京：人民邮电出版社, 2008
7. James A. Whittaker 《实用软件测试指南》电子工业出版社 2003