**Практическая работа 3**

**Цель практическое работы:**

Создать приложение по отправки email писем из приложения разработанного на Python.

**Задачи:**

* Доработать приложение по отправки email писем
* Необходимо сделать массовую отправку адресатам со вложениями.

**Ход работы**

В данной практической работы, нужно будет доработать приложение.

Для начала доработаем форму. форма должна содержать следующие элементы (Рис.1.)

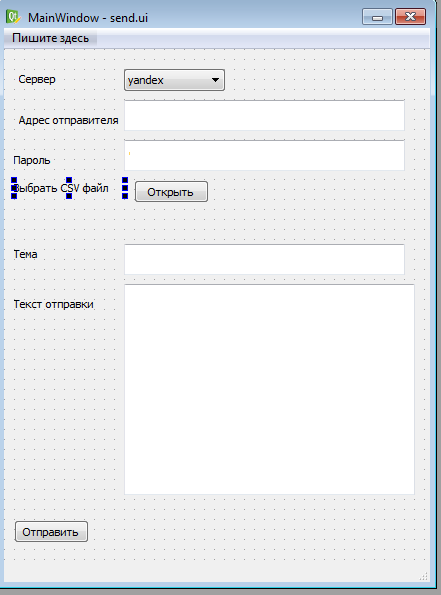


Рис.1 – Форма отправки email писем

Для того, чтобы была возможность отправлять массово email письма с вложениями, сделаем файл CSV. В котором первым значением будет «адресат», а последующие через «;»имена файлов(адрес файла). Пример содержания такого файла:

*Kulikov-mirea@yandex.ru;файл\_1.doc; файл\_1.mp4;*

*test@yandex.ru;файл\_3.doc; файл\_2.doc;*

Как видно из файла, у разных адресатов, может быть свой набор файлов.

Для этого создадим событие «click» на кнопку «Открыть»

self.connect(  
 self.ui.open\_file,  
 QtCore.SIGNAL(**'clicked()'**),  
 open\_file)

Функция **open\_file** – внешняя, вне класса **Win(QtGui.QMainWindow).**

**def** open\_file():  
 filename= QtGui.QFileDialog.getOpenFileName(  
 parent=win,  
 caption=**u'Открыть файл'**,  
 directory=PATH,  
 filter=**'csv (\*.csv);;AllFiles (\*)'** )  
 **if** filename:  
 win.path=filename  
 win.ui.label\_4.setText(filename)

Для выбора файлов, только в формате CSV, в файловом диалоге, необходимо указать фильтр, задается он так: **'csv (\*.csv);;AllFiles (\*)'.**

Запоминаем адрес файла в переменной ***path*** в объекте класса **Win**. В функции отправления сделаем чтение файла CSV.

f = open(unicode(win.path) , **'r'**)  
reader = csv.reader(f)

Объект reader, содержит массив строк из CSV файла.

Следующий шаг, это обработка массива строк, делается это следующим образом:

**for** row **in** reader:  
 addr\_to,file = row[0].split(**";"**)  
 send\_email(addr\_from,password,addr\_to,teame , msg, [file])

На каждой итерации цикла, будем отправлять письма адресатам с файлами. Функция ***send\_email***, уже была отписана в предыдущей практической работе, только в этой функции необходимо добавить параметр для передачи файлов, которые передаются в функцию:

process\_attachement(msg, files)

Для сохранения истории отправки адресатов, введем логирование в текстовый файл.

Создадим файл с до записью(если файл отсутствует, он создастся автоматически)

my\_file = open(**"send.txt"**, **"a"**)

Теперь логирование можно включить в цикл считывания CSV файла:

**for** row **in** reader:  
  
 addr\_to,file = row[0].split(**";"**)  
 send\_email(addr\_from,password,addr\_to,teame , msg, [file])  
 my\_file.write(**"\n Email {0}; тема:{1}; сообщение: {2}"**.format(addr\_to,teame,msg))

Таким образом, в файл ***send.txt*** будут записаны данные, кому отправлено, тема, и сообщение.

**Заключение**

В ходе практической работы были закреплены знания языка Python, а так же была освоена технология массовой отправки писем с вложениями. Чтение CSV файла. Было написано полностью рабочее приложение по отправки писем.