

Министерство образования республики Молдова
Технический Университет Молдовы
Департамент Программной Инженерии и Автоматики

О т ч ё т

Лабораторная работа №4

По предмету: Programarea in retea

Выполнил студ. гр. SI-202

Абабий Эдуард

Проверил

Лях Аркадий

Кишинёв – 2023

Тема: Сă се creeze un program client de poștă electronică(MUA – Mail User Agent) apt să trimită și să citească mesajele prin intermediul unui cont de poștă electronică.

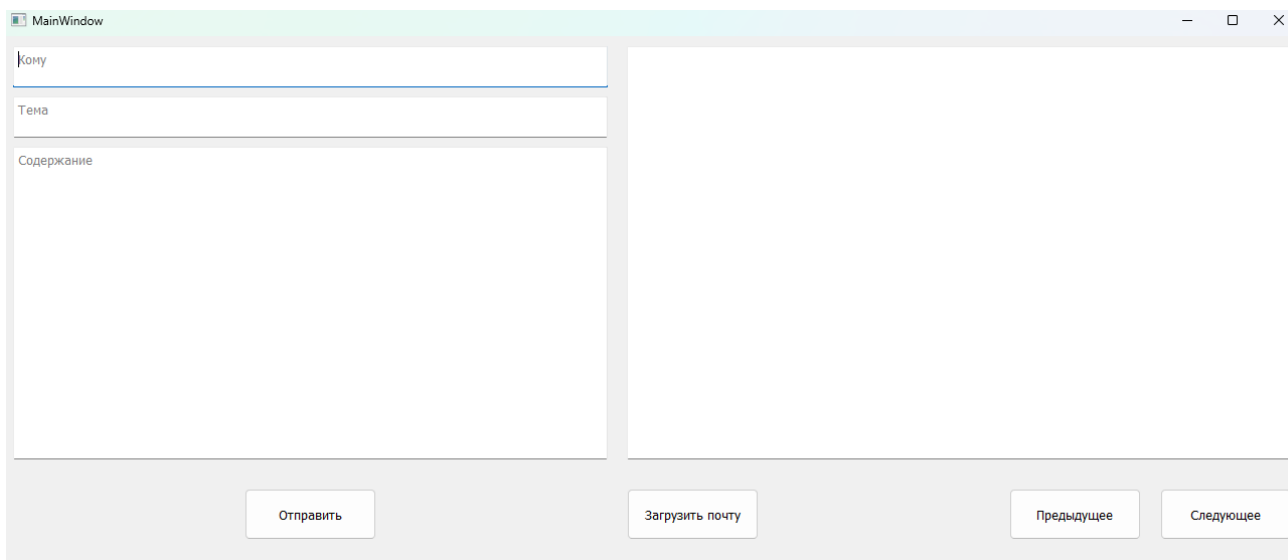


Рисунок 1 – интерфейс приложения

Здесь у нас есть поля для ввода текста:

- поле «Кому» для указания получателя
- поле «Тема» для указания темы сообщения
- поле «Содержание» для основной части сообщения

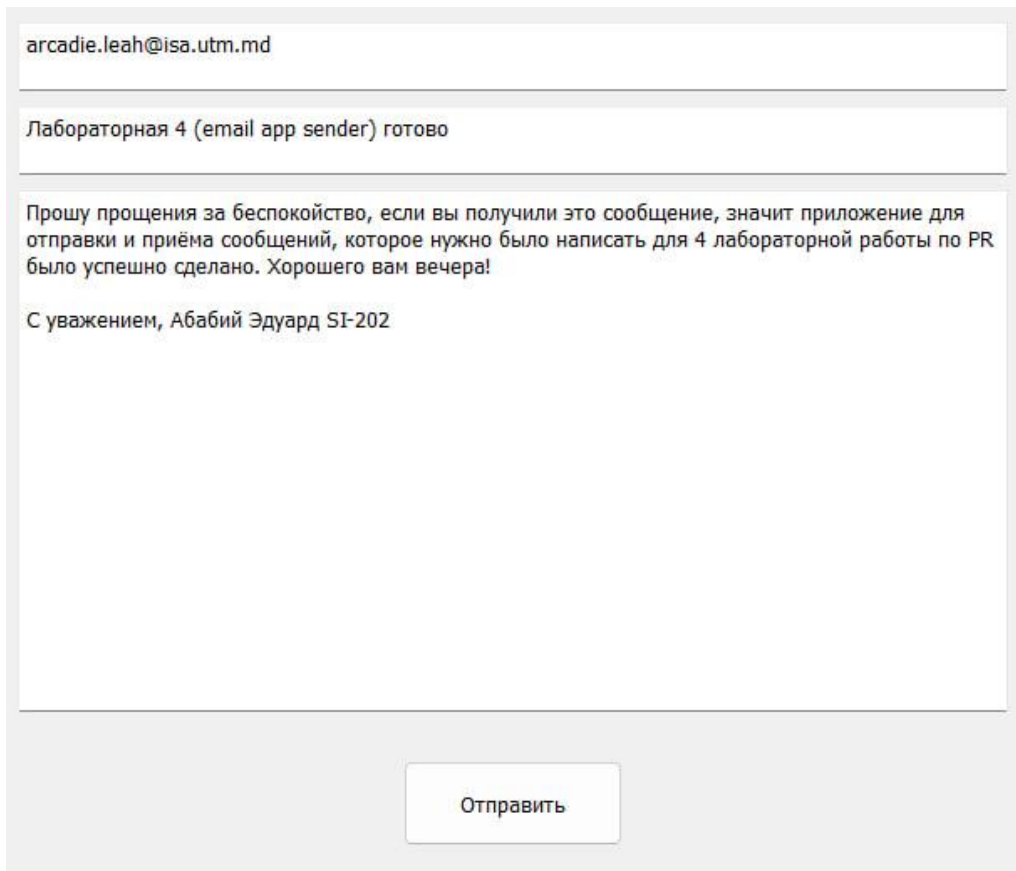
Поля для вывода текста:

- поле «Пустое», где будут отображаться загруженные сообщения

Кнопки:

- кнопка «Отправить», создаёт форму email сообщения и отправляет адресату
- кнопка «Загрузить почту», загружает почту с вашего аккаунта
- кнопка «Предыдущее» отображает в поле предыдущее сообщение по списку
- кнопка «Следующее» отображает в поле следующее сообщение по списку

Пример отправки сообщения:



arcadie.leah@isa.utm.md

Лабораторная 4 (email app sender) готово

Прошу прощения за беспокойство, если вы получили это сообщение, значит приложение для отправки и приёма сообщений, которое нужно было написать для 4 лабораторной работы по PR было успешно сделано. Хорошего вам вечера!

С уважением, Абабий Эдуард SI-202

Отправить

Рисунок 2 – отправка сообщения

```
def sendMessage(self):  
    messageTo = self.whoIAm.toPlainText()  
    theme = self.themeCont.toPlainText()  
    body = self.bodyCont.toPlainText()  
    if not re.match(r"^[^@]+@[^@]+\.[^@]+$", messageTo):  
        print("Неправильный формат email-адреса")  
    else:  
        mailServer.sendMessage(messageTo, theme, body)
```

Рисунок 3 – код frontend части приложения для отправки сообщения

Функция **sendMessage** вытаскивает все данные из полей, затем проверяет, что поле получается подходит под форму email адреса, если нет то выводит соответствующее сообщение. Если форма подходит, то мы отправляем данные в backend часть приложения.

```
def sendMessage(to, sub, body):  
    # Создаем объект SMTP  
    server = smtplib.SMTP('smtp.gmail.com', 587)  
  
    # Подключаемся к серверу и аутентифицируемся  
    server.starttls()  
    server.login("ababiy495@gmail.com", "ababiy495@gmail.com")
```

Рисунок 4 – backend часть для отправки сообщения

Здесь мы создаём объект SMTP и подключаемся к нужному порту 587

Далее подключаемся к серверу и проводим аутентификацию

```
# Создаем сообщение  
msg = MIMEText(body)  
msg['Subject'] = sub  
msg['From'] = 'ababiy495@gmail.com'  
msg['To'] = to  
  
# Отправляем сообщение  
server.sendmail("ababiy495@gmail.com", to, msg.as_string())  
  
# Закрываем соединение  
server.quit()
```

Рисунок 5 – backend часть для отправки сообщения

Создаём само сообщение, указываем тему, содержание, отправителя и получателя

Затем отправляем сообщение **sendmail**

И после чего закрываем сервер

Пример просмотра почты:

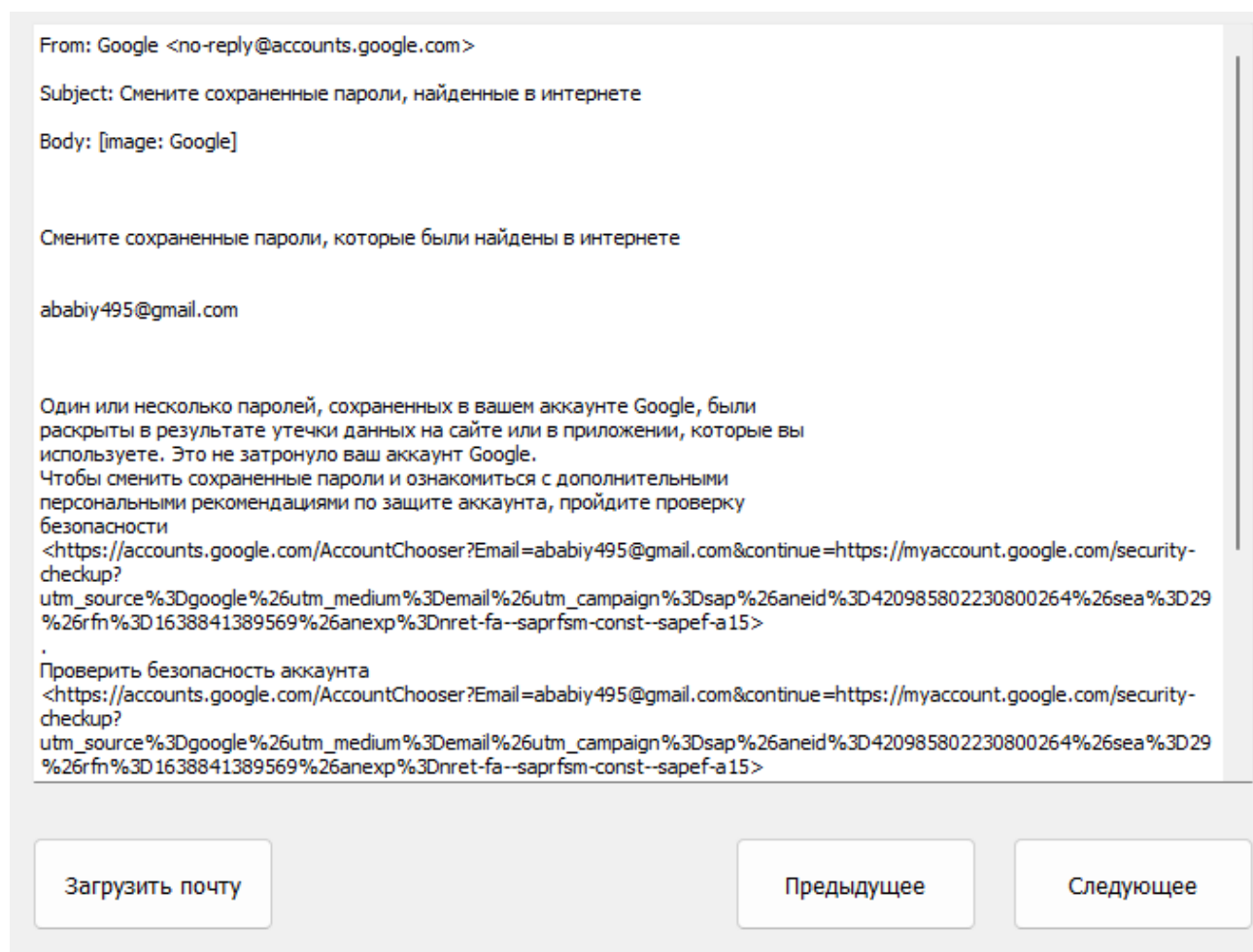


Рисунок 6 – загрузка почты

```
def takeMessages():  
    # Указываем данные для подключения к серверу POP3  
    pop_server = 'pop.gmail.com'  
    pop_port = 995  
    username = 'ababiy495@gmail.com'  
    password = 'orailbyarjvrotcr'
```

Рисунок 7 – бэк часть для получения почты

Здесь мы указываем все данные для подключения к серверу POP3

```
# Устанавливаем соединение с сервером
server = poplib.POP3_SSL(pop_server, pop_port)
server.user(username)
server.pass_(password)
```

Рисунок 8 – установка соединения с сервером

```
# Получаем количество сообщений в ящике и их общий размер
message_count, total_size = server.stat()
print(f'Сообщений в ящике: {message_count}, общий размер: {total_size} байт')

# Получаем список последних 30 сообщений
messages = [server.retr(i) for i in range(message_count, max(message_count - 30, 0), -1)]
list_of_emails = []
# Разбираем сообщения
for msg in messages:
    # создаем объект email.Message из байтового представления письма
    msg_str = b"\n".join(msg[1]).decode("utf-8")
    msg_obj = email.message_from_string(msg_str)
```

Рисунок 9 – получение сообщений и их последующая обработка

Вывод: В данной лабораторной работе я узнал о том, что такое SMTP и POP3, как с ними работать. Создал небольшое графическое приложение для отправки сообщений и загрузки почты.