MINISTERUL EDUCAȚIEI
AL REPUBLICII MOLDOVA
UNIVERSITATEA TEHNICĂ
A MOLDOVEI

DEPARTAMENTUL INGINERIA
SOFTWARE ȘI AUTOMATICĂ

MINISTRY OF EDUCATION
OF THE REPUBLIC OF MOLDOVA
TECHNICAL UNIVERSITY
OF MOLDOVA

Network Programming

Laboratory work 4

Email Client

Author:
Stanislav Bîzdîga

Supervisor:
Alexandr Gavrișco

Chișinău - 2018

Laboratory work 4

1 General purpose

Use application-level protocols (POP3/IMAP and SMTP) to implement an email client with basic functionality.

2 Prerequisites

– Read about OSI model (definition, basic info)

– Read about (de)serialization,parsing

– Client-Server Architecture

3 Specific Tasks

I. Base task (5 - 6)

There are some prerequisites:
- pick up an e-mail service (it should support IMAP/POP3 and SMTP clients)

Implement a simple mail client which offers the following features:
a. Log in;
b. Get number of unread messages;
c. Get last N received messages (display subject, date, sender), ordered by date;
d. Send a message. Next fields must be available - subject, recipient, CC, body;

II. Message Preview and MIME-types (7)

a. Display additional info for messages list:
a1. If there are attachments in the e-mail, display number of attached files;
a2. If message is plain text, display a short preview of the message (e.g. first sentence);
b. Allow to select and read a message (just display content)

III. IMAP, HTML and attachments (8)

a. If you implemented previous task using POP3 protocol, migrate the app to IMAP;
b. Allow to send e-mail with attachments (any file, at least 1 attachment);
c. If an e-mail is in HTML format, either display it using a WebView or send as file and open in browser;
d. Implement drafts (synchronized with the e-mail server)

IV. Notifications and search (9)

a. Check in background for new messages and display a notification (show pop-up and/or play a sound) when there's a new message;

b. Add search feature (e.g. find messages which contain a (keyword) in subject or sender).

<u>c.</u> Add "remember me" option to the login screen (store passwords securely).

V. UX improvements (10)

a. "Undo" button - allow to undo sending a message after user sent it (implementation is up to you);

b. Implement pagination for messages (e.g. display 10 msg per page with next/prev buttons)

c. Allow basic formatting (bold, italic etc) for a new email.

d. If there's no internet connection, enqueue messages and send them later.

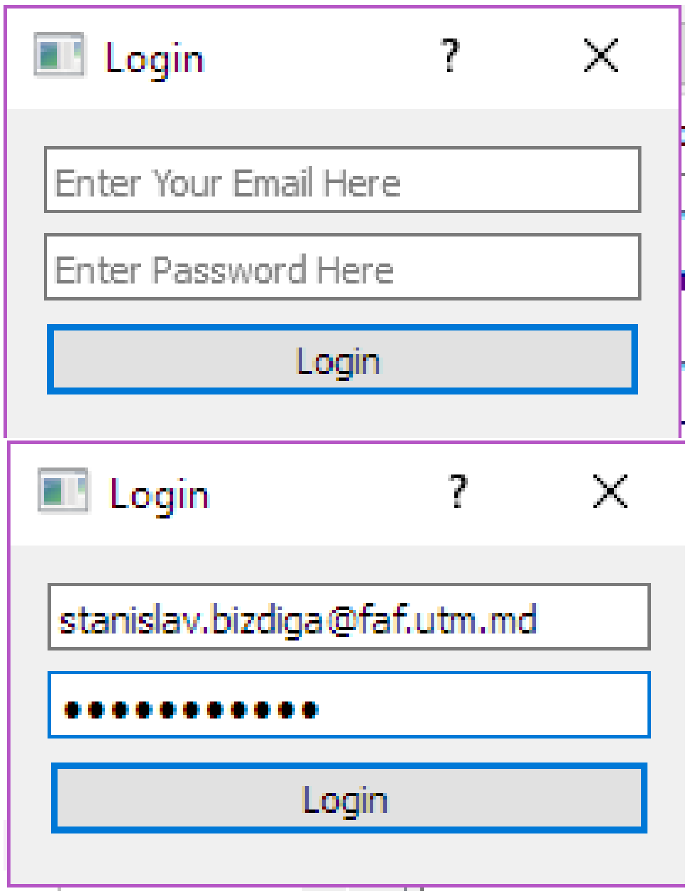
4 Task Realization

NOTE: Task letters(enumerations above) noted as BOLD are fully completed. Underlined - tackled a bit, & could have easily been implemented if had more time, i.e. there is a general understanding of how to do it and a feel of grasp of the concept.

(GUI took way too much time to design and implement.)

Task I.a. - Log in

With this task, I've added a dialog window to take care about the logging in. There are the email and password text fields (the latter is hidden so that it's safe to introduce it), that get the data from the user and input it into the logging function, which tries to connect and fetch the email server data. If it fails, then it prompts the user to retry.



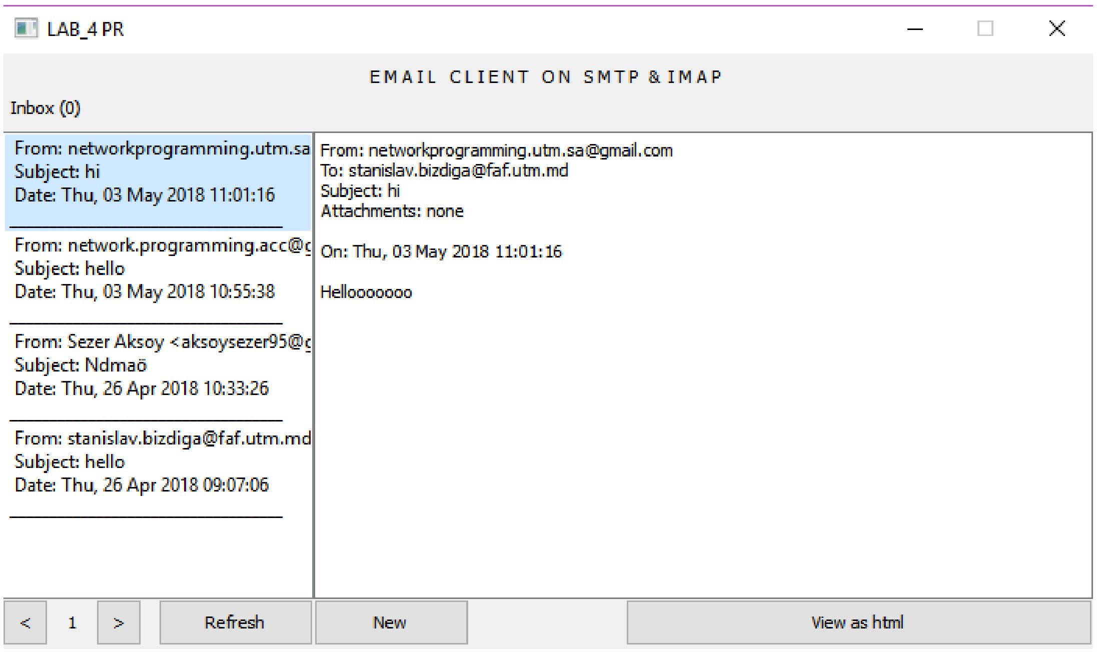
stanislav.bizdiga@faf.utm.md

●●●●●●●●●●●●

Figure 4.1 – Login dialog and filling in the personal data

Task I.b. - Get number of unread messages

With this task, the implementation is simple, as the fetching of data is complete, you search for emails that are tagged "unseen" and count them, updating the number in the GUI.



From: networkprogramming.utm.sa
Subject: hi
Date: Thu, 03 May 2018 11:01:16

From: network.programming.acc@ç
Subject: hello
Date: Thu, 03 May 2018 10:55:38

From: Sezer Aksoy <aksoysezer95@ç
Subject: Ndmaö
Date: Thu, 26 Apr 2018 10:33:26

From: stanislav.bizdiga@faf.utm.md
Subject: hello
Date: Thu, 26 Apr 2018 09:07:06

From: networkprogramming.utm.sa@gmail.com
To: stanislav.bizdiga@faf.utm.md
Subject: hi
Attachments: none

On: Thu, 03 May 2018 11:01:16

Helloooooooo

Figure 4.2 – Unread message count is beside "Inbox", top left

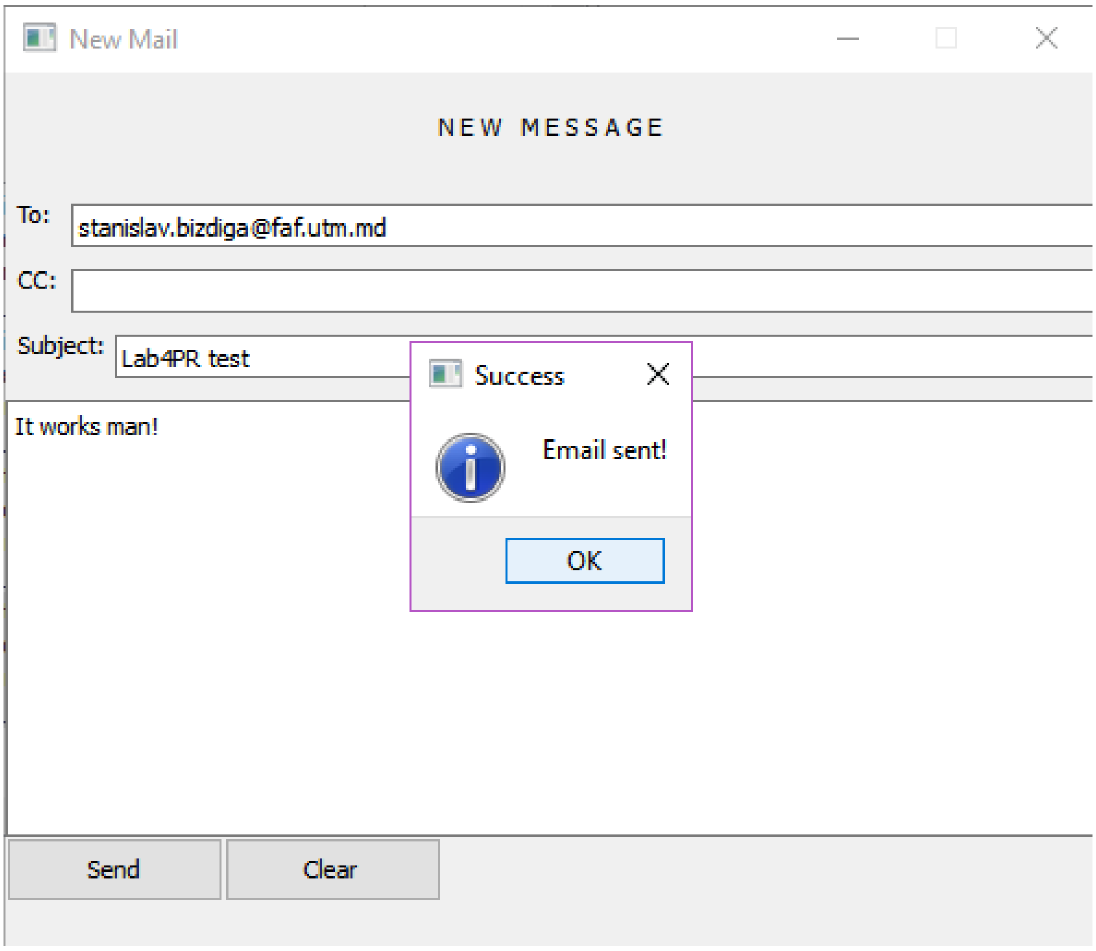
Task I.c. - Get last N received messages (display subject, date, sender), ordered by date

The messages are organized in a list (4 per page), with the Subject, date, sender info. It is also scroll-able, and click-able *(see task II.b)*.

The list can be seen on the left-side, in the picture above.

Task I.d. - Send a message. Next fields must be available - subject, recipient, CC, body

Using the smtp connection, there is no trouble sending an email if you have the data parsed correctly in the headers. SMTP stands for Simple Mail Transfer Protocol, therefore it should be a simple, straight-forward connection with easy implementation. A message is sent as follows below, being able to see the email arrived in the next picture, setting the inbox unread count to 1:


New Mail

NEW MESSAGE

To: stanislav.bizdiga@faf.utm.md

CC:

Subject: Lab4PR test

It works man!

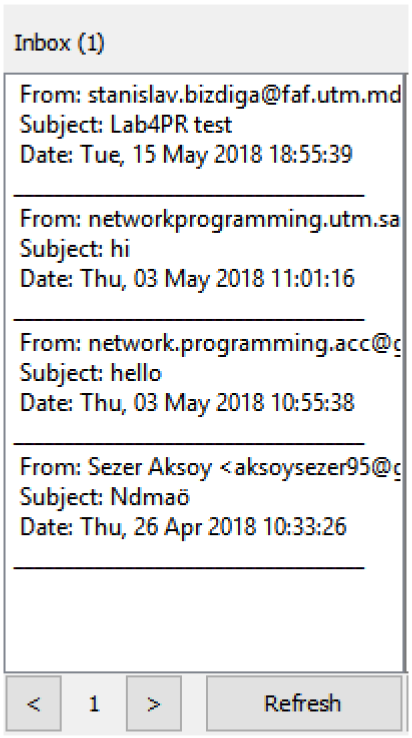
Success

Email sent!

OK

Send Clear

Figure 4.3 – Filling in the blanks, and sending the email


Inbox (1)

From: stanislav.bizdiga@faf.utm.md
Subject: Lab4PR test
Date: Tue, 15 May 2018 18:55:39

From: networkprogramming.utm.sa
Subject: hi
Date: Thu, 03 May 2018 11:01:16

From: network.programming.acc@
Subject: hello
Date: Thu, 03 May 2018 10:55:38

From: Sezer Aksoy <aksoysezer95@
Subject: Ndmaö
Date: Thu, 26 Apr 2018 10:33:26

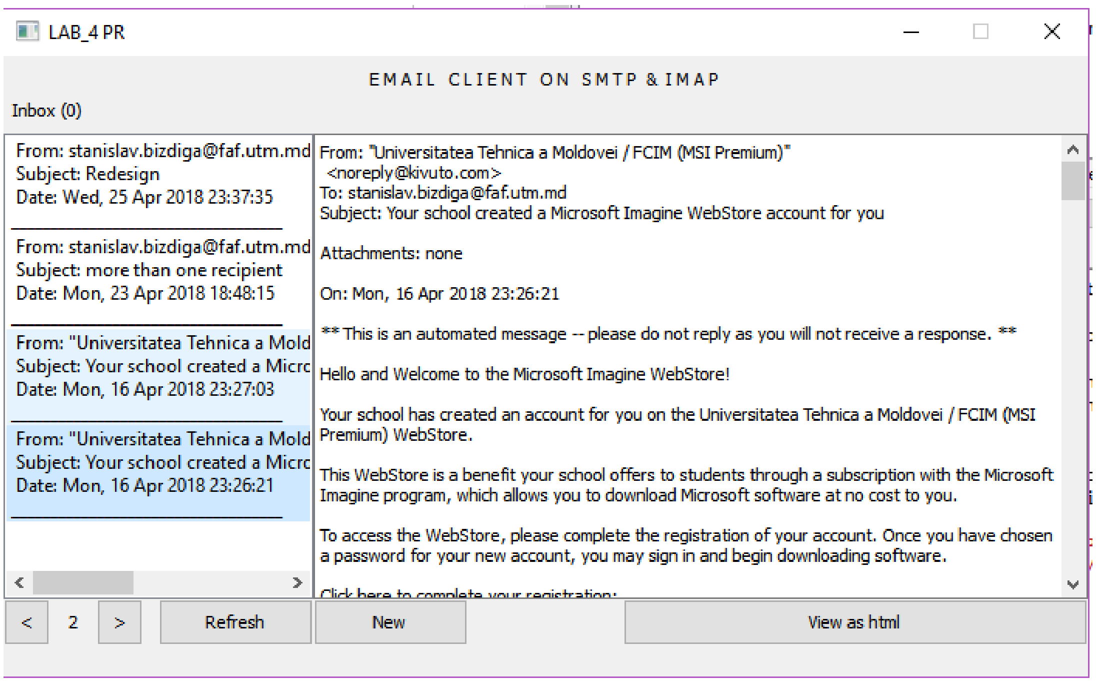
< 1 > Refresh

Figure 4.4 – Inbox List

Task II.a. - Display additional info for messages list

a1 - If there are attachments in the e-mail, display number of attached files.
So far, the approach here was very straight-forward, and silly - and it only detects if there ARE attachments, and not the exact amount of those. As you click any of the emails from the inbox list, it will include a row below the headers saying: "Attachments: none" or "Attachments: yes"
a2 - If message is plain text, display a short preview of the message (e.g. first sentence).
Basically, since I like my gui compact, it was simply displayed to the right in the message preview. But there's no problem with showing the text in the list, it just has no space for that matter. The problem here was that as the email is being loaded in the list, it would have already been counted as "Read" but my workaround is to change the way fetching is made from using the 'RFC822' to 'BODY.PEEK[HEADER]' which solved the issue.

Task II.b. - Allow to select and read a message (just display content)



From: stanislav.bizdiga@faf.utm.md
Subject: Redesign
Date: Wed, 25 Apr 2018 23:37:35

From: stanislav.bizdiga@faf.utm.md
Subject: more than one recipient
Date: Mon, 23 Apr 2018 18:48:15

From: "Universitatea Tehnica a Mold
Subject: Your school created a Micro
Date: Mon, 16 Apr 2018 23:27:03

From: "Universitatea Tehnica a Mold
Subject: Your school created a Micro
Date: Mon, 16 Apr 2018 23:26:21

From: "Universitatea Tehnica a Moldovei / FCIM (MSI Premium)" <noreply@kivuto.com>
To: stanislav.bizdiga@faf.utm.md
Subject: Your school created a Microsoft Imagine WebStore account for you

Attachments: none

On: Mon, 16 Apr 2018 23:26:21

** This is an automated message -- please do not reply as you will not receive a response. **

Hello and Welcome to the Microsoft Imagine WebStore!

Your school has created an account for you on the Universitatea Tehnica a Moldovei / FCIM (MSI Premium) WebStore.

This WebStore is a benefit your school offers to students through a subscription with the Microsoft Imagine program, which allows you to download Microsoft software at no cost to you.

To access the WebStore, please complete the registration of your account. Once you have chosen a password for your new account, you may sign in and begin downloading software.

Click here to complete your registration:

Figure 4.5 – Clicking and previewing the email

Task III.a. - If you implemented previous task using POP3 protocol, migrate the app to IMAP

The whole development was initially decided upon SMTP and IMAP.

Task III.c. - If an e-mail is in HTML format, either display it using a WebView or send as file and open in browser

Implemented a skeleton button for that, but haven't managed to with the functionality. The idea is simple though. I'd have created a WebView in QtGui, and just gave it the html data from the email.

Task IV.c - Add "remember me" option to the login screen (store passwords securely).

There was a few attempts working with this feature. I've been saving my password and email in code for quicker debugging so there was a temporary autologin feature. Storing it could be secure by using some encryption algorithm but a quick work-around was a mere encoding in base64. Also the passwords would've been saved locally into a file. However, due to time insufficience, the feature was given up on.

Below is a snippet of what was in use during the auto-login phase. (Also there are some useful global variables)

```
EMAIL_ACCOUNT=""
EMAIL_PASSWORD = ""

#def decode_password():
#     return base64.b64decode(EMAIL_PASSWORD).decode()
#def encode_password(password):
#     return base64.b64encode(password)
```

Task V.b - Implement pagination for messages (e.g. display 10 msg per page with next/prev buttons).

As previously noted, there is an inbox list that allows 4 messages per page, and so, the pagination is there.

Extra note:
Extreme amounts of time was spent on perfecting the GUI and designing the layout to have a comfortable and nice feel. Due to inexperience in gui programming it took additional time to handle it and learn the PyQt library.

Conclusions

This laboratory work is useful because of the intense study concerning the basics of network programming. Specifically, protocols, and in this lab, we've touched on mailing protocols like IMAP and SMTP.

Also, an incredibly useful learning experience was the creation and designing of the GUI elements and layouts - an activity never before done in python.

Thanks to this lab, working with networks doesn't seem as scary as it used to but rather an interesting activity that requires a bunch of research and preparation before actually making some progress.

With this knowledge, I am eager to try dig deeper into network programming, as it has become now a little clearer.

Bibliography

GUI programming in Python: PyQt4 Objects and their attributes
`http://pyqt.sourceforge.net/Docs/PyQt4/`

SMTP in python:
`https://docs.python.org/2/library/smtplib.html`

IMAP in python:
`https://docs.python.org/2/library/imaplib.html`

Email package:
`https://docs.python.org/3/library/email.examples.html#email-examples`