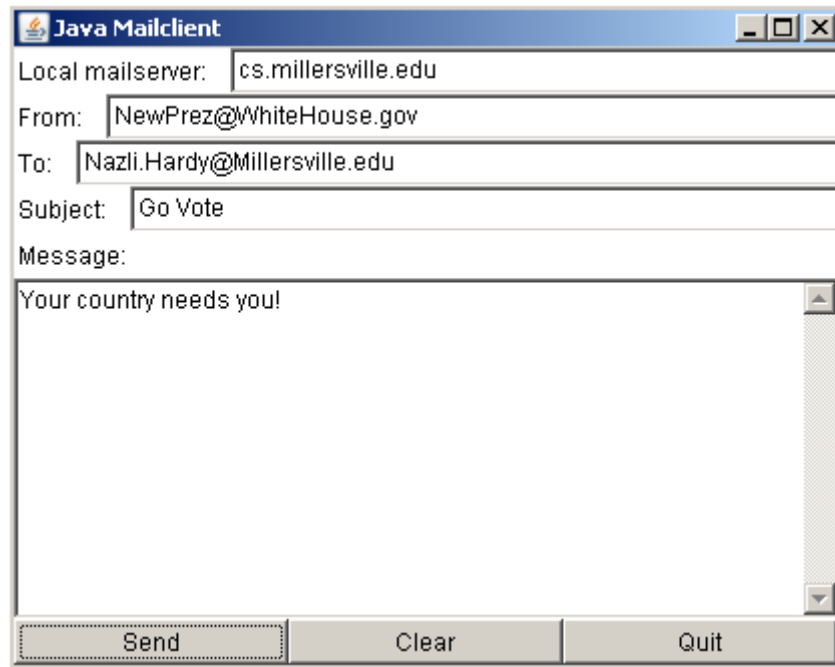


## CSCI 395: Computer Networks (W) Mail User Programming Assignment Due 10/19/08 by 11:59 PM

### Goal

In this lab you will implement a **mail user agent** that sends mail to other users. Your task is to complete the **SMTPConnection** interaction between the MUA and the local SMTP server. The **MailClient** provides a graphical user interface containing fields for entering the sender and recipient addresses, the subject of the message and the message itself. See below:



With this interface, when you want to send a mail, you must fill in complete addresses for both the sender and the recipient, i.e., `Nazli.Hardy@Millersville.edu`, not just simply `user`. You can send mail to only one recipient. You will also need to give the name (or IP address) of your local mailserver.

When you have finished composing your mail, press *Send* to send it.

### The Code

The program consists of four classes:

<b>MailClient</b>	The user interface
<b>Message</b>	Mail message
<b>Envelope</b>	SMTP envelope around the Message
<b>SMTPConnection</b>	Connection to the SMTP server

You will need to complete the code in the **SMTPConnection** class so that in the end you will have a program that is capable of sending mail to any recipient. The code for the **SMTPConnection** class and the other three classes is provided on my website ([Computer Networks/ Resources/ Mail User Lab](#)).

The places where you need to complete the code have been marked with the comments **[in pink]**. Each of the places may require one or more lines of code.

1. The **MailClient** class provides the user interface and calls the other classes as needed.

2. When you press *Send*, the `MailClient` class constructs a `Message` class object to hold the mail message. The `Message` object holds the actual message headers and body.
3. Then the `MailClient` object builds the SMTP envelope using the `Envelope` class. This class holds the SMTP sender and recipient information, the SMTP server of the recipient's domain, and the `Message` object.
4. Then the `MailClient` object creates the `SMTPConnection` object which opens a connection to the SMTP server and the `MailClient` object sends the message over the connection.

The sending of the mail happens in 3 phases:

1. The `MailClient` object creates the `SMTPConnection` object and opens the connection to the SMTP server.
2. The `MailClient` object sends the message using the function `SMTPConnection.send()`.
3. The `MailClient` object closes the SMTP connection.

Note: the `Message` class contains the function `isValid()` which is used to check the addresses of the sender and recipient to make sure that there is only one address and that the address contains the @-sign. The provided code does not do any other error checking.

### Reply Codes

For the basic interaction of sending one message, you will only need to implement a part of SMTP. In this lab we will only need to implement the following SMTP commands (**look at detailed class notes from the board**):

Command	Reply Code
DATA	354
HELO	250
MAIL FROM	250
QUIT	221
RCPT TO	250

The above table also lists the accepted reply codes for each of the SMTP commands you need to implement. For simplicity, you can assume that any other reply from the server indicates a fatal error and abort the sending of the message.

In addition, when you open a connection to the server, it will reply with the code 220.

### The Output

Once you are done, send **me** an email from **you** – with this message, “**skinamarinki dinki di, skinamarinki doo**”. For example, the message I received from the Prez is below:

