1. Coding report

Class *Mail*:

* Variable within class object:
  + **sender**: a string to store the sender's email address.
  + **cmd\_list**: a list storing the command data extracted from a received email.
  + **attachment**: a string to store the path of the return file (in case there is one)
  + **body**: a string to prepare the answer for the email sent back to the requester.
  + **subject**: a fixed string of the email title.
  + **email\_message**: a MIMEMultipart object to compress the email content that will be sent.
* Member functions of the mail class:
  + The `**fetch\_mail**` function is designed to retrieve unread emails from a Gmail inbox with the subject "Mail Control". It logs into the mail server, selects the inbox, searches for unread emails with the specified subject, and processes each email to extract the sender and a command list using the `**decode\_mail**` function.
  + The `**send\_mail**` function is designed to send an email using the SMTP protocol via the Gmail SMTP server (*smtp.gmail.com*). It establishes a connection, logs in using predefined global constants (*USERNAME* and *APP\_PASS*), sends an email to a specified recipient (sender), and prints a message indicating the successful mailing operation.
  + The `**screenshot**` function captures a screenshot of the user's PC, attaches it to an email, and populates the email message with relevant information. The function utilizes an external function, `**capture\_SS**`, to obtain the screenshot file path. It also incorporates auxiliary functions, **current\_time**, and **set\_info**, to include a timestamp and set additional information in the email.
  + The **webcam** function captures an image using the computer's webcam, attaches it to an email, and sets up the email message with relevant information. The function utilizes an external function, **capture\_webcam**, to obtain the webcam capture details. It also includes auxiliary functions, such as **set\_info**, to configure the email subject and additional information.
  + The **keylogger** function initiates a keylogging operation, configures email information, and prepares an email with details about the keylogging duration. Leveraging the external function **key\_logger**, it captures the keylogger log file path and the specified duration for keystroke recording. Additionally, the function incorporates **set\_info** to establish the email subject and initialize the email message object. The resulting email body contains information about the program recording keystrokes for a specific duration. The keylogger file is attached to the email for the sender's retrieval.
  + The **logout** function composes an email body indicating that the user has been logged out of their PC and attaches this message to the email. The function utilizes the **send\_mail** method to send the email, notifying the sender of the logout forward. Subsequently, the function employs the *os.system("shutdown -l")* command to initiate a system logout.
  + Similarly, the shutdown function composes an email body indicating the imminent shutdown of the user's PC and then notifying the sender of the impending shutdown via email. The function proceeds to execute a countdown, displaying the remaining seconds until shutdown in the console. Finally, it initiates a system shutdown using the *os.system("shutdown /s /t now")* command.
  + The **list\_app** function retrieves the list of currently running applications on the user's PC using the **list\_running\_application** function. Then the **set\_info** support function is called to set additional information. The running applications list is attached as a text file named *"RunningApplications.txt"* to the email for the sender's reference.