# logo

# SCaMPy

Scan and Mail with Python

## Requirements

The application should allow a user who has no knowledge of commercial scanner software to scan any number of documents and send these scanned documents as a zip file attachment in an email message.

The email service to be used is GMail.

Here’s a scenario that explains the need:

The user receives documents he needs to fill in manually, or has another kind of document he wants to send to a mail recipient. As the user might not be aquanted with the burden of using a flatbed scanner, creating a zip file and send this zipfile as an attachment he needs to have a simple user interface.

This interface or GUI (Graphical User Interface) as we will call it from now on will assist hem in the above mentionned tasks.

## Tasks of the GUI

The GUI has several distinctive tasks it needs to help the user with. First of all, the GUI should remember previously created and emailed documents. The GUI will show a list of these ‘sent’ items in tabular view upon startup. This provides the user with an overview of previously performed tasks, it’s like looking at ‘Sent Items’ in an email program (or the Online GMAIL web app).

The other tasks are:

1. Start a new scan project
   1. Let the user give the project a name
   2. Let the user enter the email address of his correspondent(s)
   3. Let the user enter a subject and a message body
2. Scan document(s)
   1. Check if the scanner is online, and if not ask the user if the scanner is on, and that he is using the correct WIFI connection
   2. Guide the user in asking to place a document on the scanner
   3. Scan the document
   4. Save the scanned document
   5. Show the document
   6. Ask for more documents
3. Compose the email
4. Send the email
5. Save the project

## Technologies and libraries

* Programming Language Python 3.7
* PyQT5 for the Gui layer
* Pyinsane2 for the scanner layer
* Python SMTP library
* Windows 10 and Ubuntu Linux

## Use cases

#### Starting the Application

Once the user has decided he wants to send an email by using SCaMPy, he launches the application by double-clicking on it’s icon on the desktop. The application then starts up and loads the list of recently created email/scan projects in a table. This table is sorted by the date of the creation of the project in a descending order.

#### Creating a new project

The user has a button/menu option to start a new scan/email project. By selecting it, the system will require the following information:

* Name of the project. This information will be used as the name of the folder where the scanned documents will be stored.
* The email adress(es) of the person(s) he wishes to send the scanned documents to.
* The subject of the email.
* The body of the email message.
* Indicate if he wants to make a zip file with the scanned documents. The name of the zipfile will contain the name of the project.
* The application will inform the user if the scanner is online. If it can’t find a scanner a warning will be issued. It will state to turn on the scanner, and to verify if the correct WIFI connection is used.

#### Scan document(s)

Once the user has completed use case 2, he can start scanning documents. The system does not care how many documents there are. Each scanned document will simply be added to the project. He will have an option to preview the newly scanned documents. If this preview doesn't please the user, he can re-scan the document. This might happen if the document was not in the right position for example and the user wants to put the document in another position on the scanner.

#### Send Email

After use case 3 is completed, meaning the user has scanned all the documents he wants to include in the email, he can click on the Send Message button. The application will then automatically connect to the GMAIL server using the account linked to the user.

The system will then, if selected, compress the scanned images into a zipfile and then send this as an attachment to the recipient(s).

An option could be included here, as the files are scanned into a PNG format, it might be handy to convert them into a PDF file. Such a library exists and was already tested for another application.

The application will update the project information database by setting the following fields:

Date\_Sent, date when the email was sent

Time\_sent, time of email,

Nr of pages,

Size of the zipfile,

Success, ok or not