User Manual

Student: Connor Scanlan – C00226867

Supervisor: James Egan

Cybercrime & IT Security – CW_KCCYB_B

Institute of Technology Carlow

Table of Contents

System Requirements

Operating System:

- Windows 10

Programming Language

- Python 3.9.0

Python Modules required

Cryptography	TKInter
OS	Shutil
Logging	Platform
Subprocess	Register
HashLib	Json
Fernet	

Installation Instructions

Make sure all the requirements above are met, and that all the Python Modules that are required are also installed.

Th next step will be to download the application files into the desired destination. Once the main application files are downloaded you can simply go to that location and run the application.

System Usage

Register

When the application is first loaded you will be brought to a registration page. Once here you will register, and it will create an account for the application on your local device. You will require a registered account to be able to access the functions of the application. (See **Appendix A, Appendix B, Appendix C**)

Login

Once you have registered or if you already have a registered account, you can click the login button and will be brought to the login page. The next step is to use your login details to gain access to the application. (See **Appendix D, Appendix E**)

Open a File

The open file button will allow you to open any file on the device. This button will open the local file explorer. (See **Appendix F, Appendix G**)

Copy a directory

The copy directory button allows you to go into file explorer and choose which directory you wish to work on and copies it to another location for testing. (See **Appendix G, Appendix H**)

Hash Feature

This feature allows you to get a hash of a file, this can be used to test if a file has been edited as the hashes will not be the same. (See **Appendix I**)

Logging

The logging feature will record the time an individual logs in, and it will also record which features they use. This is to record the time of events. These logs will be stored in a folder on your device for access later by an admin. (See **Appendix J**)

Hashing

The Hashing Feature will encrypt a file and create a key which can be used to then decrypt the file. (See **Appendix K, Appendix L**)

Bibliography

Van Rossum, G., 2007, June. Python Programming Language. In *USENIX annual technical conference* (Vol. 41, p. 36).

Sanner, M.F., 1999. Python: a programming language for software integration and development. *J Mol Graph Model*, *17*(1), pp.57-61.

Agarwal, S., Bharti, P.K. and Pathak, R.K., Implementation of DES Algorithm in Python.

Martin, D., 2020. *CryptoConfig (ITS Data Management)*. National Renewable Energy Lab.(NREL), Golden, CO (United States).

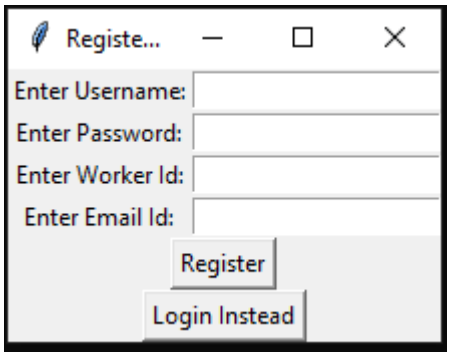
Sukumaran, J. and Holder, M.T., 2010. DendroPy: a Python library for phylogenetic computing. *Bioinformatics*, *26*(12), pp.1569-1571.

Holmgren, W.F., Andrews, R.W., Lorenzo, A.T. and Stein, J.S., 2015, June. PVLIB python 2015. In *2015 ieee 42nd photovoltaic specialist conference (pvsc)* (pp. 1-5). IEEE.

Hunt, J., 2019. Logging in Python. In *Advanced Guide to Python 3 Programming* (pp. 311-322). Springer, Cham.

Appendix

Appendix A



Appendix B

```python
def _register(self):
    name = self.username_username.get()
    password = self.password_username.get()
    worker = self.worker_username.get()
    email = self.email_username.get()

    register.add_user({"name": name, "password": password,
                       "worker_id": worker, "email_id": email})

    self.root.destroy()

    window = MainWindow(worker)
    window.add_elements()
    window.run()
```

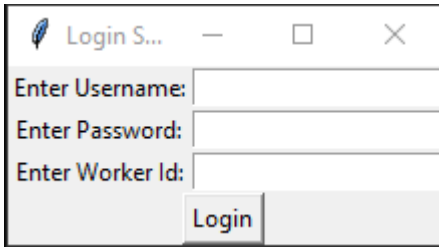
Appendix C

```python
import json
import os

mydir = 'C:/Users/New User/Desktop/Projects 4th Year/Tkinter File Traverser/logs'
myfile = 'admin.json'
admin_path = os.path.join(mydir, myfile)

def add_user(info: dict):
    with open(admin_path) as f:
        data = json.load(f)

    data.append(info)

    with open(admin_path, "w") as f:
        json.dump(data, f, indent=4)


def check_user(info: dict):
    mydir = 'C:/Users/New User/Desktop/Projects 4th Year/Tkinter File Traverser/logs'
    myfile = 'admin.json'
    admin_path = os.path.join(mydir, myfile)

    with open(admin_path) as f:
        data = json.load(f)

    for user in data:
        del user["email_id"]

        if info == user:
            return True

    return False
```

Appendix D

```
def _login(self):

    #mydir = 'C:/Users/New User/Desktop/Projects 4th Year/Tkinter File Traverser/logs'
    #myfile = 'admin.json'
    #admin_path = os.path.join(mydir, myfile)
    #self.decrypt(admin_path)
    name = self.username_username.get()
    password = self.password_username.get()
    worker = self.worker_username.get()

    result = register.check_user({"name": name, "password": password,
                                  "worker_id": worker})

    if result:
        self.root.withdraw()

        window = MainWindow(worker)
        window.add_elements()
        window.run()
        #self.encrypt(admin_path)
    else:
        messagebox.showerror("Wrong Credentails",
                             "The credentails you provided were wrong")
```

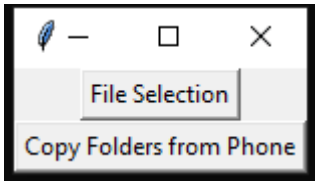
Appendix E



Appendix F

```
def _file_selection_handler(self):
    self.root.withdraw()
    filepath = askopenfilename()
    self.root.deiconify()

    if platform.system() == 'Darwin':
        subprocess.call(('open', filepath))  # For Mac OS
    elif platform.system() == 'Windows':
        os.startfile(filepath)  # For Windows
    else:
        subprocess.call(('xdg-open', filepath))  # For Linux

    logging.info(f'{self.user} has opened {filepath}')
```

Appendix G



Appendix H

```
def _transfer_files(self):
    src = askdirectory()
    dest = askdirectory()

    try:
        shutil.copytree(src, os.path.join(dest, os.path.split(src)[-1]))
    except Exception:
        pass
    else:
        logging.info(
            f'{self.user} has copied {src} to {os.path.join(dest, os.path.split(src)[-1])}')
```

Appendix I

```
file =  file_2_hash
BLOCK_SIZE = 65536

file_hash = hashlib.sha256()
with open(file, 'rb') as f:
    fb = f.read(BLOCK_SIZE)
    while len(fb) > 0:
        file_hash.update(fb)
        fb = f.read(BLOCK_SIZE)



print (file_hash.hexdigest())
```

Appendix J

```
def __init__(self, user):
    self.user = user

    self.root = Tk()
    self.root.title("Options Screen")

    with open(f'C:/Users/New User/Desktop/Projects 4th Year/Tkinter File Traverser/logs/{self.user}.log', 'w') as _:
        pass

    logging.basicConfig(filename=f'C:/Users/New User/Desktop/Projects 4th Year/Tkinter File Traverser/logs/{self.user}.log', level=logging.INFO,
                        format='%(asctime)s:%(levelname)s:%(message)s')
    logging.info(f'{self.user} has logged in')
```

Appendix K

```
#this just opens the 'key.key' and assings the key stored there as 'key'
file = open('key.key','rb')
key = file.read()
file.close()

# value of key is assigned to a variable
f = Fernet(key)

#this opens the json and reads its data into a new variable called 'data'
with open(admin_path,'rb') as f:
    data = f.read()

#this decrypts the data read from the json
fernet = Fernet(key)
decrypted=fernet.decrypt(data)

#this writes the new, encrypted data into a new JSON file
with open('admin.json','wb') as f:
    f.write(decrypted)
```

Appendix L

```
#this generates a key and opens a file 'key.key' and writes the key there
key = Fernet.generate_key()
file = open('key.key','wb')
file.write(key)
file.close()

#this just opens the 'key.key' and assings the key stored there as 'key'
file = open('key.key','rb')
key = file.read()
file.close()

#this opens the json and reads its data into a new variable called 'data'
with open(admin_path,'rb') as f:
    data = f.read()

#this encrypts the data read from the json and stores it in 'encrypted'
fernet = Fernet(key)
encrypted=fernet.encrypt(data)

#this writes the new, encrypted data into a new JSON file
with open('admin.json','wb') as f:
    f.write(encrypted)
```