Internship Report Out 2023

Paul Khayet, Intern

April, 2023

About Me

- **Grew up In El Dorado Hills, CA**
- **Went to Folsom Lake College**
- **Transferred to the University of California Irvine in 2021**
- **Expected to Graduate in 2024 with Computer Engineering Degree**

Snowboarding ↑

Hobbies

- **Traveling**
- **Snowboarding**
- **Photography**
- **Reading**
- **Lifting Weights**

Photography ->

First Project (PCN Mapping)

- Problem: We have a large folder filled with PCN PDF's and would like customers to be able to find the correct one on the website
- Solution: Create a script that searches through the PDF's to find the correct file using either the PCN, the SKU, or MMID. Then display it to the customer.

```python
# importing required modules
import PyPDF2
import os
import sys
import shutil

# Directories of PDF Files MAKE SURE TO CHANGE THESE!!!!
currentDir = r"place\original\folder\here"
newDir = r"place\destination\folder\here"

# Take in arguments from the command line
argument = sys.argv[1]
if len(sys.argv) > 2:
    i = 2
    while i < len(sys.argv):
        argument = argument + " " + sys.argv[i]
        i += 1

# Copies a list of the current directory
pdfDir = os.listdir(currentDir)

# The list of all the PCN's that have the currect drives
pdfList = []
destinationList = []

# User inputs what they want the script to search for
currentSearch = sys.argv[1]
fileCount = 0

# Finds the amount of files in the current directory
totalFiles = len(pdfDir)

# Loops through every file looking for the matching text
while fileCount < totalFiles:
    # Checks that file is a PDF
    pdfCheck = pdfDir[fileCount].find(".pdf")
    if pdfCheck != -1:
        pdf = pdfDir[fileCount]
        pdfFileObj = open(pdf, "rb")
        pdfReader = PyPDF2.PdfFileReader(pdfFileObj)
        pgNum = pdfReader.numPages
        allPages = ""

        x = 0
        while x < pdfReader.numPages:
            pageObj = pdfReader.getPage(x)
            allPages += pageObj.extractText()
            x += 1
        x = 0

        pdfFileObj.close()
```

In action

```
python3 script.py 978348
```

PCN's

Name	Date Modified	Size	Kind
CC 660P_976804_P...2)_August 23 2018.pdf	Jun 9, 2022 at 2:35 PM	158 KB	PDF Document
CC 660P_976804_P...0_January 13 2021.pdf	Jun 9, 2022 at 2:35 PM	87 KB	PDF Document
CC 660P_976804_P..._February 17 2021.pdf	Jun 9, 2022 at 2:34 PM	98 KB	PDF Document
CC 660P_976804_P...3)_August 30 2021.pdf	Jun 9, 2022 at 2:33 PM	96 KB	PDF Document
CC 660P_976804_P...4-00_July 14 2021.pdf	Jun 9, 2022 at 2:33 PM	55 KB	PDF Document
CC 660P_978348_P...0_August 23 2018.pdf	Jun 9, 2022 at 2:23 PM	158 KB	PDF Document
CC 660P_978348_PCN116943-00_May 15 2019.pdf	, 2022 at 2:23 PM	39 KB	PDF Document
CC 660P_978348_P...-01_June 24 2020.pdf	Jun 9, 2022 at 2:22 PM	135 KB	PDF Document
CC 660P_978348_P...0_August 19 2020.pdf	Jun 9, 2022 at 2:21 PM	169 KB	PDF Document
CC 660P_978348_P...November 23 2020.pdf	Jun 9, 2022 at 2:20 PM	180 KB	PDF Document
CC 660P_978348_P...7-00_April 27 2021.pdf	Jun 9, 2022 at 2:19 PM	260 KB	PDF Document
CC 660P_978348_P...01_August 30 2021.pdf	Jun 9, 2022 at 2:18 PM	96 KB	PDF Document
CC 660P_978350_PC...)_August 23 2018.pdf	Jun 9, 2022 at 2:30 PM	158 KB	PDF Document
CC 660P_978350_PC...(1)_May 15 2019.pdf	Jun 9, 2022 at 2:29 PM	39 KB	PDF Document
CC 660P_978350_PC...(1)_June 24 2020.pdf	Jun 9, 2022 at 2:28 PM	135 KB	PDF Document
CC 660P_978350_PC...)_August 19 2020.pdf	Jun 9, 2022 at 2:28 PM	169 KB	PDF Document
CC 660P_978350_PC...ovember 23 2020.pdf	Jun 9, 2022 at 2:27 PM	180 KB	PDF Document
CC 660P_978350_PC...(1)_April 27 2021.pdf	Jun 9, 2022 at 2:26 PM	260 KB	PDF Document
CC 660P_978350_PC...)_August 30 2021.pdf	Jun 9, 2022 at 2:25 PM	96 KB	PDF Document

destination

Name	Date Modified	Size	Kind
CC 660P_978348_P...01_August 30 2021.pdf	Jun 9, 2022 at 2:18 PM	96 KB	PDF Document

Lessons Learned:

- Learned to code in Python (no prior python experience)
- Created a better understanding of UNIX file management
- Learned how to manipulate PDF's and convert them to TXT files

```python
# importing required modules
import PyPDF2
import os
import sys
import shutil

# Directories of PDF Files MAKE SURE TO CHANGE THESE!!!!
currentDir = r"place\original\folder\here"
newDir = r"place\destination\folder\here"

# Take in arguments from the command line
argument = sys.argv[1]
if len(sys.argv) > 2:
    i = 2
    while i < len(sys.argv):
        argument = argument + " " + sys.argv[i]
        i += 1

# Copies a list of the current directory
pdfDir = os.listdir(currentDir)

# The list of all the PCN's that have the currect drives
pdfList = []
destinationList = []

# User inputs what they want the script to search for
currentSearch = sys.argv[1]
fileCount = 0

# Finds the amount of files in the current directory
totalFiles = len(pdfDir)

# Loops through every file looking for the matching text
while fileCount < totalFiles:
    # Checks that file is a PDF
    pdfCheck = pdfDir[fileCount].find(".pdf")
    if pdfCheck != -1:
        pdf = pdfDir[fileCount]
        pdfFileObj = open(pdf, "rb")
        pdfReader = PyPDF2.PdfFileReader(pdfFileObj)
        pgNum = pdfReader.numPages
        allPages = ""

        x = 0
        while x < pdfReader.numPages:
            pageObj = pdfReader.getPage(x)
            allPages += pageObj.extractText()
            x += 1
        x = 0

        pdfFileObj.close()
```

Second Project (Server Dashboard)

- Problem: Solidigm has dozens of servers used by CAT, SRT, etc. that are only available to those who know the specific IP of the server
- Solution: Create a dashboard available for all teams to see a list of servers and created a system in which to allocate time to individuals who want to use them

Frontend

Server List

Created by Khayet, PaulX SK, last modified just a moment ago

Reserving Server Time:

1. **Find the server that fits your requirements**
2. **Go to that servers page and find an available time in the calendar**
3. **Reserve the time slot with the ticket you are working on (if available)**
4. **When you are done with the server make sure to leave it as you found it**
5. **If you are done early please update the calendar to show that it's available**

Server 1	**CentOS 8** / U.2 / 24 Bay
Server 2	**CentOS 7** / U.2 / 24 Bay
Server 3	**CentOS 6** / U.2 / 24 Bay
Server 4	**Redhat 9** / E1.L / 12 Bay
Server 5	**Ubuntu 16.02** / U.2 / 12 Bay
Server 6	**Windows Server 10.7** / M.2 / 6 Bay

Server 1

Created by Khayet, PaulX SK, last modified just a moment ago

Use this Wiki page to automatically track what servers are available and the current configuration of the servers.

All entries need to be manually updated to keep track of systems

Server 1 (192.168.73.45):

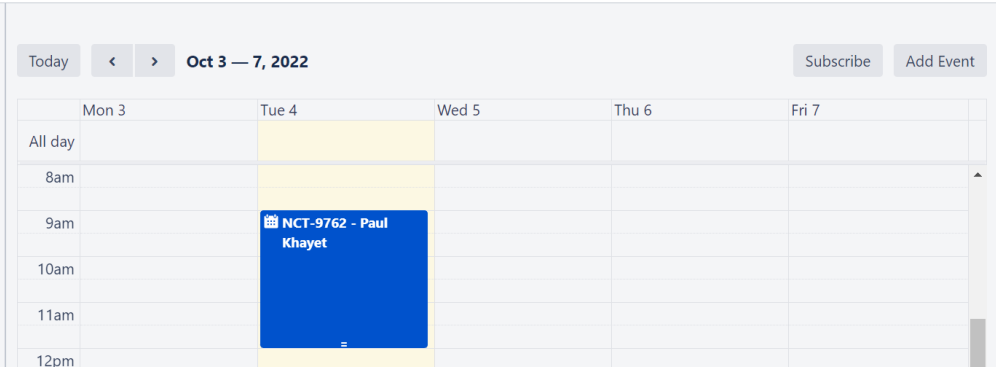
Status: AVAILABLE

OS: CentOS 7

Connection: U.2

Drive Bay's: 24

Last Used By: Paul Khayet



- Created a user-friendly front end so that anyone who needs to access the servers would have a simple and easy way of seeing what servers are available. They would also be able to reserve time if they know that they will need the server soon or if the server is used by multiple teams.

Backend

```python
#updates following global variables: IP, username, OS
def update_UserOSIP(server):
    global localIP, OS, username
    # This section of the code checks for the OS and runs the appropriate commands
    if platform.system() == "Linux" or platform.system() == "linux2":
        # This portion of the code returns the version of linux currently being run
        RELEASE_DATA = {}

        with open("/etc/os-release") as f:
            reader = csv.reader(f, delimiter="=")
            for row in reader:
                if row:
                    RELEASE_DATA[row[0]] = row[1]

        if RELEASE_DATA["ID"] in ["debian", "raspbian"]:
            with open("/etc/debian_version") as f:
                DEBIAN_VERSION = f.readline().strip()
            major_version = DEBIAN_VERSION.split(".")[0]
            version_split = RELEASE_DATA["VERSION"].split(" ", maxsplit=1)
            if version_split[0] == major_version:
                # Just major version shown, replace it with the full version
                RELEASE_DATA["VERSION"] = " ".join([DEBIAN_VERSION] + version_split[1:])

        OS = "{} {}".format(RELEASE_DATA["NAME"], RELEASE_DATA["VERSION"])

        username = getpass.getuser()
```

	A	B
1		Server 1
2	Status	Available
3	OS	Windows-10
4	Connection	U.2
5	Drive Bays	16
6	Last Used By	intel

- We loaded scripts onto the servers that would pull necessary information from the system and load them onto a Google Sheet that could be displayed on the NPSG Wiki. We also had a 3rd party server that would ping all the servers that had a script in order to confirm that they were online at any given time

Key Takeaways

- Developed stronger communication skills
- New technical skills (Python/Google API’s)
- Working with team members help reduce roadblocks
- Confidence in my ability to learn
- Problem Solving Skills
- Sleeping on a problem helps find the solution

Special Thanks to:

- Fred Khoury
- Felipe Martinez
- Marcus Winters
- Manvir Kaur
- Kate Nguyen
- Carson Jamieson

What's next

End Date: May 12th, 2023

June 2023

Graduate Spring 2024