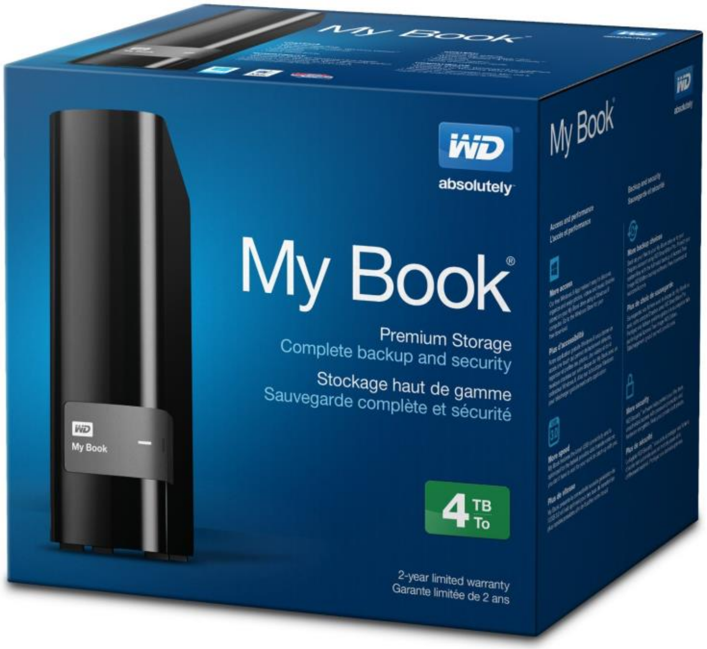
# Handling Archived Data: Big Data Team



Advisor: Erik Rozolis

Member: Jin Bai

# TABLE OF CONTENTS

## 1.0 Briefing into Big Data Team

This is Jin Bai's first semester on the Big Data Team. In the initial Sunday meetings, new members of Big Data were to first choose which group they wanted to work in for the semester. Jin choose to join the Web Interface and System/Cloud group.

The System/Cloud group ended up only meeting a handful of times and is not the focus of this report; from Appendix A, you can see what Jin has learned on the System/Cloud group such as installing and setting up Azure (Appendix A: figure 1,2), generating secure keys (Appendix A: figure 3), and creating Azure virtual machine instances as well as running virtual machine commands such starting and stopping a VM(Appendix A: figure 4).

In the earlier Thursday meetings of the semester from around 8/31 to 9/18, the Web Interface group mainly focused on teaching new members about how the website works, setting us up with access to the GitHub and Google Drive, helping us SSH into the server, and showed us how our database looked. These items were not delved into in major detail. Members took notes on these items.

Each member was then assigned a different task for the semester and beyond. Jin was assigned to handle archived data so that people could access and analyze the data we had in our archives.

## 2.0 Introduction to Handling Archived Data

We expect to have several hard disks in the future that store various types of archived data. Currently, we don’t have a way to utilize that valuable data. The idea is to eventually access them easily and incorporate that into the schema of our current website.

The long term goal is to be able to access archived data and allow people to analyze that. There are currently several hard disks which are several terabytes, each, full of archived data from national parks. The advisors of the Web Interface team, Erik and Everett, would like to be able to pull images, and eventually videos, from these disks and display them on the website for people to analyze. This is actually part of a much larger project, but Jin’s job is to focus on setting up the drives so that it is possible to retrieve any image from any one of them.

Even though Jin’s job was to simply set up the hard drives so that we can access the images, he went above and beyond that and is hopeful that we can already incorporate those image files into the website as a new prototype by next semester.

## 3.0 Initial Ideas, Problems and Analysis of Various Use Cases

There were a lot of problems initially. It wasn't even possible to access our archived images in the WD storage device. My Book, the directory with the data could not be cd into, permission was denied and change mode had no effect on it since it was an external device. The only way that My Book was able to be accessed was as super user (Appendix B: figure 1). Another issue was that we could not view the images even as super user.

This problem was finally solved with Ahmed's help by creating a new user called "archive" on the archive machine. By connecting the drive when logged in as archive, the new user was granted permission to access My Book. It is no longer necessary to be super user to access images (no need for permission changes either). Archive will now be the primary account used by the team for accessing the archived images (Appendix B: figure 2).

The initial idea was to set up a two way socket form of communication between the archive drive machine that is connected to the WD media storage drive and the development machine. Professor Lu informed the team that a socket was a very ambitious task that takes a ton of time so should look elsewhere perhaps and HTTP server. After that, Professor Lu suggested that we find a way to index our data to account for various use cases such as for past crime incidents, vacation location history, historical weather analysis, and past events.

## Implementation and Issues

Professor Lu suggested the use of mapping tables and after viewing the data with Michael, it was decided that mapping tables may not be necessary as our data is extremely organized (Appendix C: figure 1). I thought immediately of associating key to value through the use of an organized Python dictionary since our images are all named in a similar fashion (Appendix C: figure 2a, figure 2b). The format of the National Park Data (Appendix C: figure 3) is the basis of this idea.

Jin implemented this idea by first making a text file that mapped the park name abbreviation to the actual name of the park with the idea in mind that users can eventually select a park name which would map to our data where the name is simply an abbreviation (Appendix C: figure 4). Then Jin implemented this idea in Python by writing a script that parsed a user request for a park through the command line (Appendix C: figure 5). Then the script would cd into the correct folder with the text file, and create the dictionary. From there, the dictionary would be used to cd into the actual folder of the park that the user wanted.

Professor Lu thought this idea was too narrow and wouldn't cover all the use cases so Jin had to further think of another idea to store all the possible future data. He began to think about making another table in our database for National Park information for easy access and integration with the site. He created an archived_data folder (Appendix D: figure 1) in camcam directory of the development machine and found a way to manually transfer single images to that folder (Appendix D: figure 2) and set up a secure communication for file transferring with ssh keys. Eventually, the idea would be to send these images from the development machine to the site.

A new table called archived_data was also created in the cam2 database (Appendix D: figure 3). The fields park_name and snapshot_time are the main parts of the table. Many values were manually input into the table (Appendix D: figure 4). The idea is to eventually use this crucial information to find images requested from the development machine to get them from the archived data machine. The next step was to find a way to auto-populate these tables instead of manually inputting values. Erik helped with the process by showing Jin how to output all image filename data from a directory into a text file (Appendix D: figure 5). Jin utilized this on each individual archive national park directory to obtain text files of all the names of parks and verified the information was correct (Appendix D: figure 6). The issue is to use this in a Python script; Jin learned how to auto-populate the database from old scripts (Appendix D: figure 7, figure 8).

## Next Semester Goals

This semester was a huge success in the members all completed the task and went beyond what was expected from them. The next step is to incorporate our back end communications into the website seamlessly. Jin has provided his ideas in two Sunday presentations and many great discussions stirred from that.

The idea Jin presented is to have a separate category specifically for archived data on our site which lets users browse our archives. The idea is to have users select that they want to see historical data, then let them select what kind of historical data to see to account for all use cases. For now we have data on national parks, but in the future when the Big Data team receives even more archive data such as from NASA or other organizations, there should be a way for users to select which one they want to view. From the URL in (Appendix E: figure 1) the Appendix E, “natl_parks” could be just one option; NASA for example could be there instead. After the user selects, further options appear (Appendix E: figure 2). From there, date and time options are displayed which allow users to further specify their request (Appendix E: figure 3) with a loading bar that pops out. Lastly, the correct images requested are displayed on the page (Appendix E: figure 4).

## Summary and Conclusions

This semester Jin has definitely learned a lot through doing VIP with the Big Data team. Working with archived data has allowed Jin to work with both hardware and software. He has also learned a lot of valuable presentation skills from weekly Sunday presentations. He wants to work with the Big Data team again next semester.

The goal for archived data in the future has become much more clear due to the work done on it this semester. Once the back end is incorporated into the site, the team will have much more flexibility in determining how they specifically want this data to be visible to the user. Many ideas came from the Sunday meetings. Ahmed and Professor Lu suggested possibly encapsulating all the data until the data is requested on the map and then have the archived images appear on the map in a slightly different format as the cameras that show up to differentiate this previous historical data from our current video camera analysis selections.

Jin has learned how to work with an SQL database, practiced his Python coding proficiency, learned how hard drives work and researched them to provide accessibility for our team. He is looking forward to continuing his research on archive data.

## References

"Cyberduck Help". RetrievedNovember , 2014 Available: https://trac.cyberduck.io/wiki/help/en

"Django Documentation". RetrievedSeptember , 2014 Available: https://docs.djangoproject.com/en/1.7/intro/tutorial01/

"is not in the sudoers file. This incident will be reported.". Retrieved October , 2014 Available: http://www.linuxforums.org/forum/red-hat-fedora-linux/144428-solved-not-sudoers-file-incident-will-reported.html

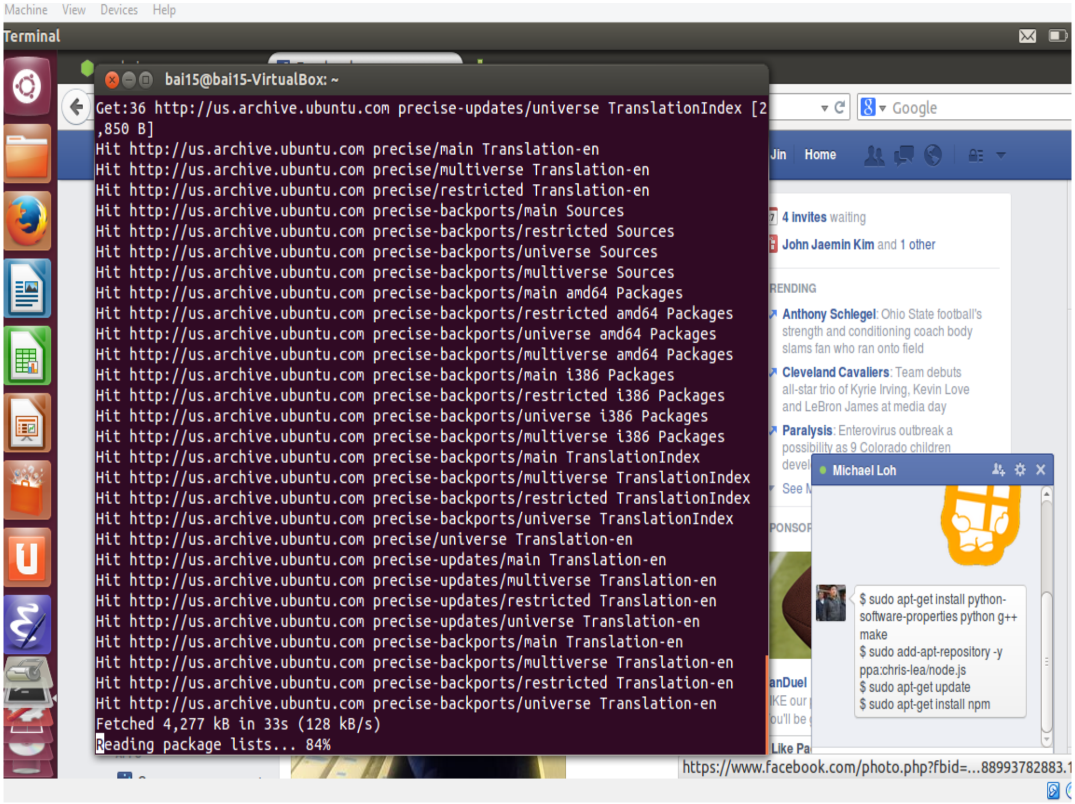
"Permission Denied when I try to access my MyBook". Retrieved October , 2014 Available: http://ubuntuforums.org/showthread.php?t=887771&page=2

Rozolis, E , "Twith2/Script.py". RetrievedOctober , 2014 Available: https://github.com/Erozolis/Twitch2/blob/master/Script.py

W , "Linux - Mount device with specific user rights". Retrieved October , 2014 Available: http://superuser.com/questions/320415/linux-mount-device-with-specific-user-rights

# Appendix A:

# System and Cloud



```
Get:36 http://us.archive.ubuntu.com precise-updates/universe TranslationIndex [2
,850 B]
Hit http://us.archive.ubuntu.com precise/main Translation-en
Hit http://us.archive.ubuntu.com precise/multiverse Translation-en
Hit http://us.archive.ubuntu.com precise/restricted Translation-en
Hit http://us.archive.ubuntu.com precise-backports/main Sources
Hit http://us.archive.ubuntu.com precise-backports/restricted Sources
Hit http://us.archive.ubuntu.com precise-backports/universe Sources
Hit http://us.archive.ubuntu.com precise-backports/multiverse Sources
Hit http://us.archive.ubuntu.com precise-backports/main amd64 Packages
Hit http://us.archive.ubuntu.com precise-backports/restricted amd64 Packages
Hit http://us.archive.ubuntu.com precise-backports/universe amd64 Packages
Hit http://us.archive.ubuntu.com precise-backports/multiverse amd64 Packages
Hit http://us.archive.ubuntu.com precise-backports/main i386 Packages
Hit http://us.archive.ubuntu.com precise-backports/restricted i386 Packages
Hit http://us.archive.ubuntu.com precise-backports/universe i386 Packages
Hit http://us.archive.ubuntu.com precise-backports/multiverse i386 Packages
Hit http://us.archive.ubuntu.com precise-backports/main TranslationIndex
Hit http://us.archive.ubuntu.com precise-backports/multiverse TranslationIndex
Hit http://us.archive.ubuntu.com precise-backports/restricted TranslationIndex
Hit http://us.archive.ubuntu.com precise-backports/universe TranslationIndex
Hit http://us.archive.ubuntu.com precise/universe Translation-en
Hit http://us.archive.ubuntu.com precise-updates/main Translation-en
Hit http://us.archive.ubuntu.com precise-updates/multiverse Translation-en
Hit http://us.archive.ubuntu.com precise-updates/restricted Translation-en
Hit http://us.archive.ubuntu.com precise-updates/universe Translation-en
Hit http://us.archive.ubuntu.com precise-backports/main Translation-en
Hit http://us.archive.ubuntu.com precise-backports/multiverse Translation-en
Hit http://us.archive.ubuntu.com precise-backports/restricted Translation-en
Hit http://us.archive.ubuntu.com precise-backports/universe Translation-en
Fetched 4,277 kB in 33s (128 kB/s)
Reading package lists... 84%
```

4 invites waiting
John Jaemin Kim and 1 other

RENDING

Anthony Schlegel: Ohio State football's strength and conditioning coach body slams fan who ran onto field

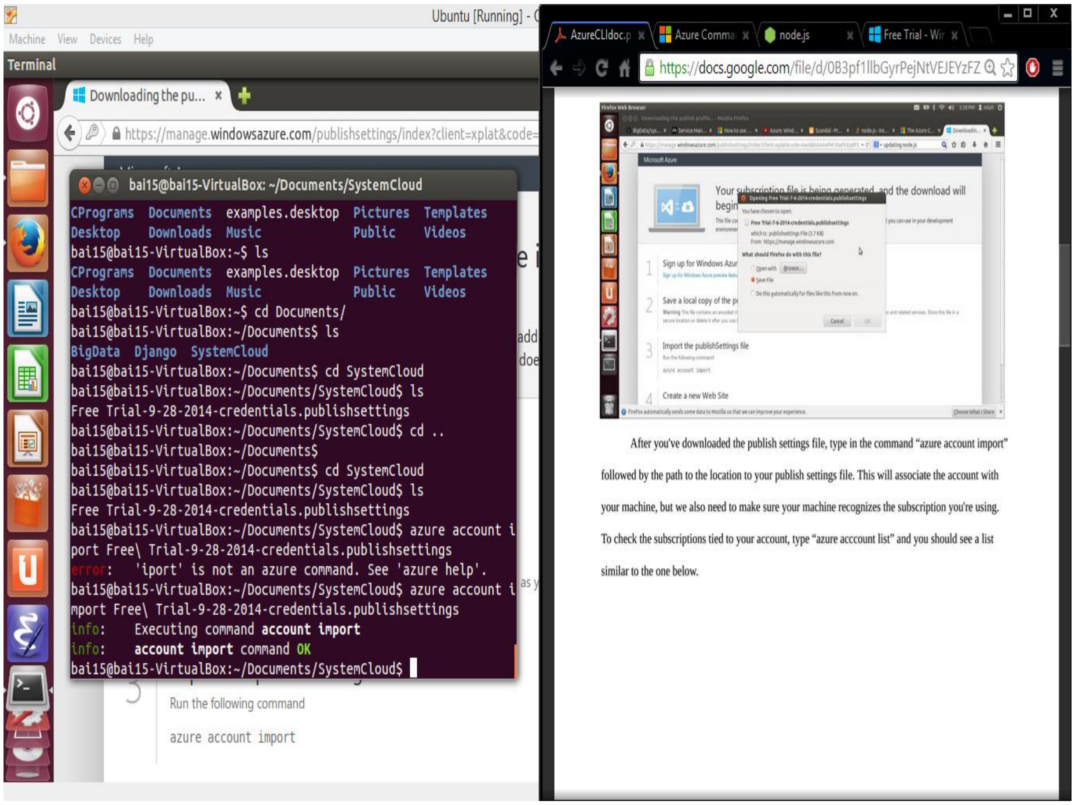
Cleveland Cavaliers: Team debuts all-star trio of Kyrie Irving, Kevin Love and LeBron James at media day

Paralysis: Enterovirus outbreak a possibility as 9 Colorado children devel

```
$ sudo apt-get install python-software-properties python g++ make
$ sudo add-apt-repository -y ppa:chris-lea/node.js
$ sudo apt-get update
$ sudo apt-get install npm
```

https://www.facebook.com/photo.php?fbid=...88993782883.1

Figure 1. Installing azure on Ubuntu virtual box



Ubuntu [Running]
Machine View Devices Help
Terminal
Downloading the pu...
AzureCLIdoc.p
Azure Comma
node.js
Free Trial - Wir
Microsoft Azure
Your subscription file is being generated, and the download will begin
Sign up for Windows Azure
Save a local copy of the p
Import the publishSettings file
Create a new Web Site
Cancel
OK

https://manage.windowsazure.com/publishsettings/index?client=xplat&code=

https://docs.google.com/file/d/0B3pf1lbGyrPejNtVEJEYzFZ

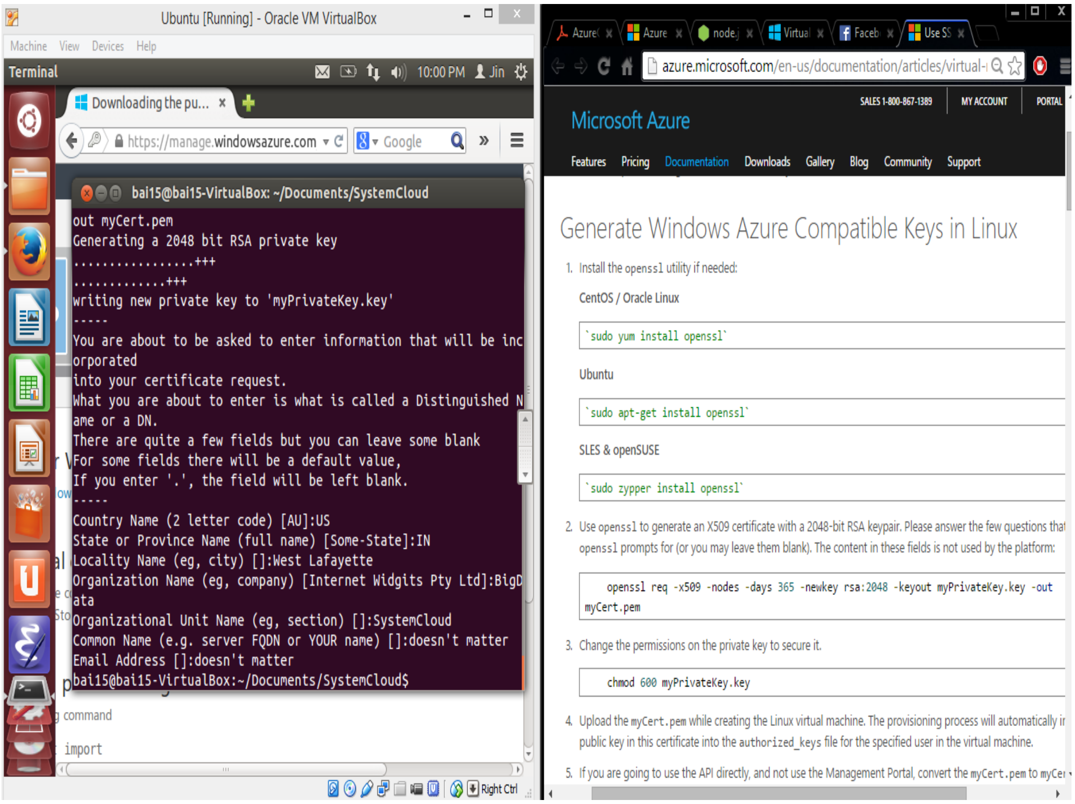
```
bai15@bai15-VirtualBox: ~/Documents/SystemCloud
CPrograms  Documents  examples.desktop  Pictures  Templates
Desktop    Downloads  Music             Public    Videos
bai15@bai15-VirtualBox:~$ ls
CPrograms  Documents  examples.desktop  Pictures  Templates
Desktop    Downloads  Music             Public    Videos
bai15@bai15-VirtualBox:~$ cd Documents/
bai15@bai15-VirtualBox:~/Documents$ ls
BigData  Django  SystemCloud
bai15@bai15-VirtualBox:~/Documents$ cd SystemCloud
bai15@bai15-VirtualBox:~/Documents/SystemCloud$ ls
Free Trial-9-28-2014-credentials.publishsettings
bai15@bai15-VirtualBox:~/Documents/SystemCloud$ cd ..
bai15@bai15-VirtualBox:~/Documents$
bai15@bai15-VirtualBox:~/Documents$ cd SystemCloud
bai15@bai15-VirtualBox:~/Documents/SystemCloud$ ls
Free Trial-9-28-2014-credentials.publishsettings
bai15@bai15-VirtualBox:~/Documents/SystemCloud$ azure account i
port Free\ Trial-9-28-2014-credentials.publishsettings
error:   'iport' is not an azure command. See 'azure help'.
bai15@bai15-VirtualBox:~/Documents/SystemCloud$ azure account i
mport Free\ Trial-9-28-2014-credentials.publishsettings
info:    Executing command account import
info:    account import command OK
bai15@bai15-VirtualBox:~/Documents/SystemCloud$
```

3 Run the following command

```
azure account import
```

After you've downloaded the publish settings file, type in the command "azure account import" followed by the path to the location to your publish settings file. This will associate the account with your machine, but we also need to make sure your machine recognizes the subscription you're using. To check the subscriptions tied to your account, type "azure acccount list" and you should see a list similar to the one below.

Figure 2: Associating Azure account with Big Data virtual machines


Ubuntu [Running] - Oracle VM VirtualBox
Machine View Devices Help
Terminal
10:00 PM
Jin
Downloading the pu...
https://manage.windowsazure.com
Google
bai15@bai15-VirtualBox: ~/Documents/SystemCloud
command
import
Right Ctrl
azure.microsoft.com/en-us/documentation/articles/virtual-
SALES 1-800-867-1389
MY ACCOUNT
PORTAL
Microsoft Azure
Features
Pricing
Documentation
Downloads
Gallery
Blog
Community
Support

```
out myCert.pem
Generating a 2048 bit RSA private key
................+++
............+++
writing new private key to 'myPrivateKey.key'
-----
You are about to be asked to enter information that will be inc
orporated
into your certificate request.
What you are about to enter is what is called a Distinguished N
ame or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:IN
Locality Name (eg, city) []:West Lafayette
Organization Name (eg, company) [Internet Widgits Pty Ltd]:BigD
ata
Organizational Unit Name (eg, section) []:SystemCloud
Common Name (e.g. server FQDN or YOUR name) []:doesn't matter
Email Address []:doesn't matter
bai15@bai15-VirtualBox:~/Documents/SystemCloud$
```

Generate Windows Azure Compatible Keys in Linux

1. Install the `openssl` utility if needed:

   CentOS / Oracle Linux

   ```
   `sudo yum install openssl`
   ```

   Ubuntu

   ```
   `sudo apt-get install openssl`
   ```

   SLES & openSUSE

   ```
   `sudo zypper install openssl`
   ```

2. Use `openssl` to generate an X509 certificate with a 2048-bit RSA keypair. Please answer the few questions that `openssl` prompts for (or you may leave them blank). The content in these fields is not used by the platform:

   ```
       openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout myPrivateKey.key -out
   myCert.pem
   ```

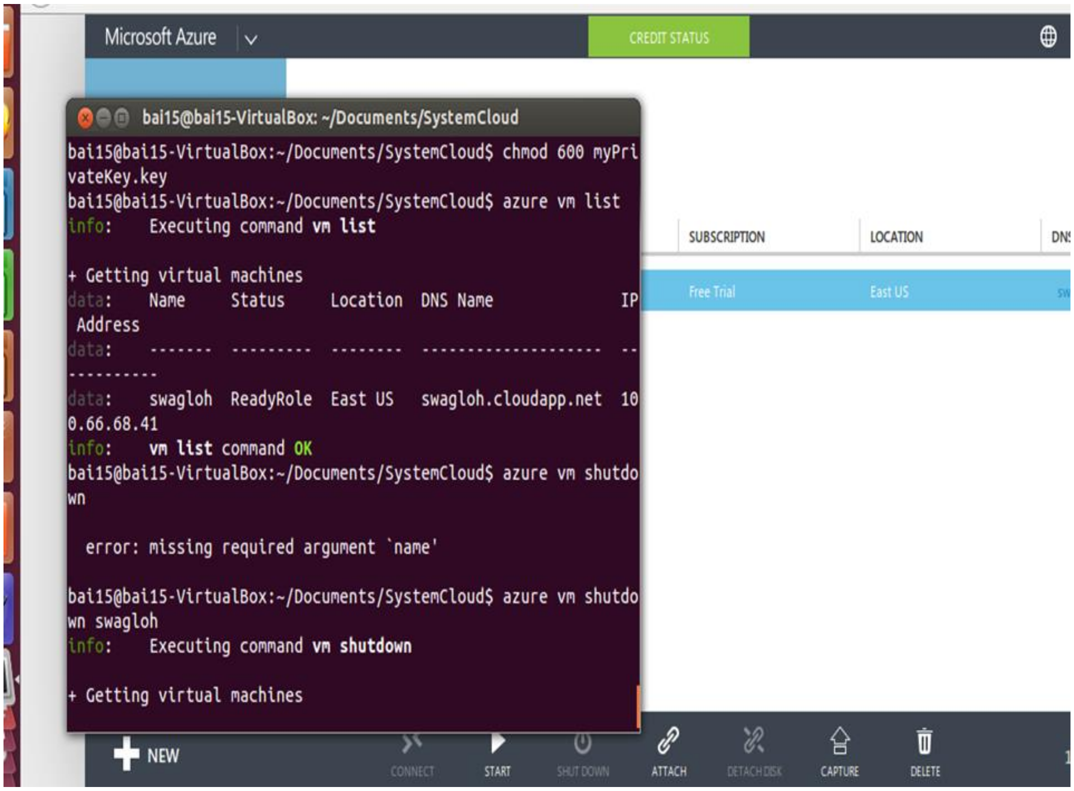
3. Change the permissions on the private key to secure it.

   ```
       chmod 600 myPrivateKey.key
   ```

4. Upload the `myCert.pem` while creating the Linux virtual machine. The provisioning process will automatically i public key in this certificate into the `authorized_keys` file for the specified user in the virtual machine.

5. If you are going to use the API directly, and not use the Management Portal, convert the `myCert.pem` to `myCer`

Figure 3. Generating secure keys for Azure



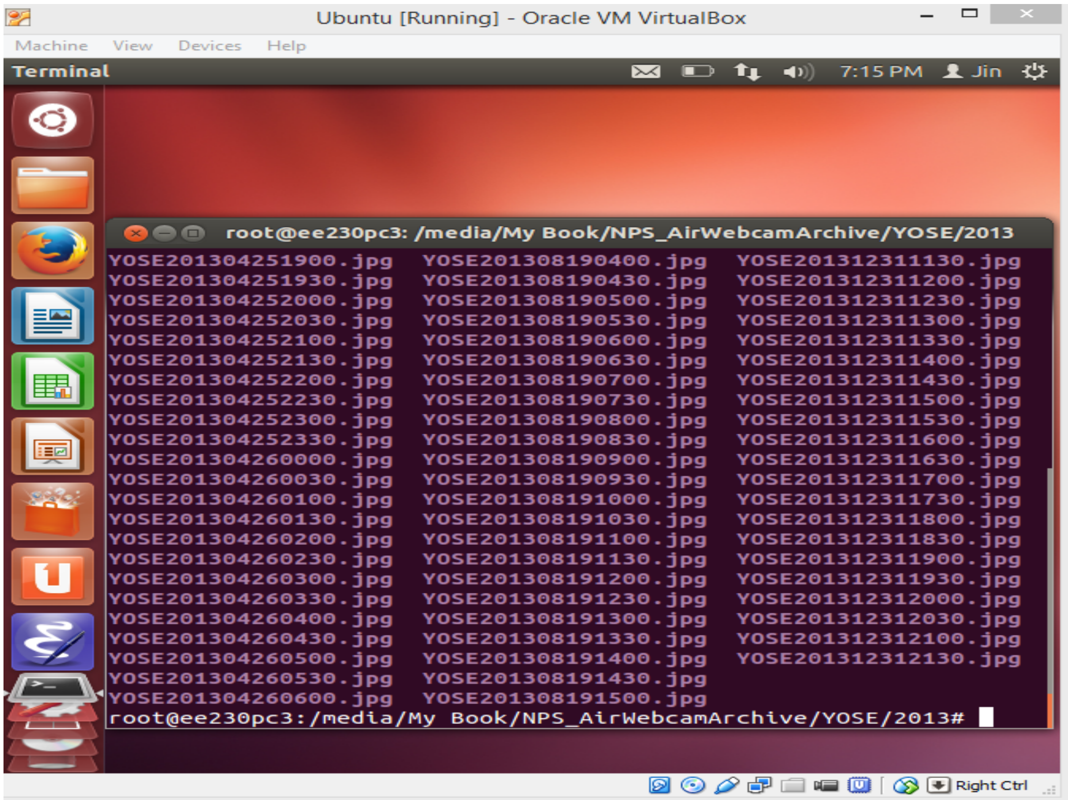
```
bai15@bai15-VirtualBox:~/Documents/SystemCloud$ chmod 600 myPri
vateKey.key
bai15@bai15-VirtualBox:~/Documents/SystemCloud$ azure vm list
info:    Executing command vm list

+ Getting virtual machines
data:    Name     Status     Location  DNS Name             IP
 Address
data:    -------  ---------  --------  -------------------  --
---------
data:    swagloh  ReadyRole  East US   swagloh.cloudapp.net  10
0.66.68.41
info:    vm list command OK
bai15@bai15-VirtualBox:~/Documents/SystemCloud$ azure vm shutdo
wn

  error: missing required argument `name'

bai15@bai15-VirtualBox:~/Documents/SystemCloud$ azure vm shutdo
wn swagloh
info:    Executing command vm shutdown

+ Getting virtual machines
```

| SUBSCRIPTION | LOCATION |
|---|---|
| Free Trial | East US |

Figure 4. Created an Azure instance, running various virtual machine commands

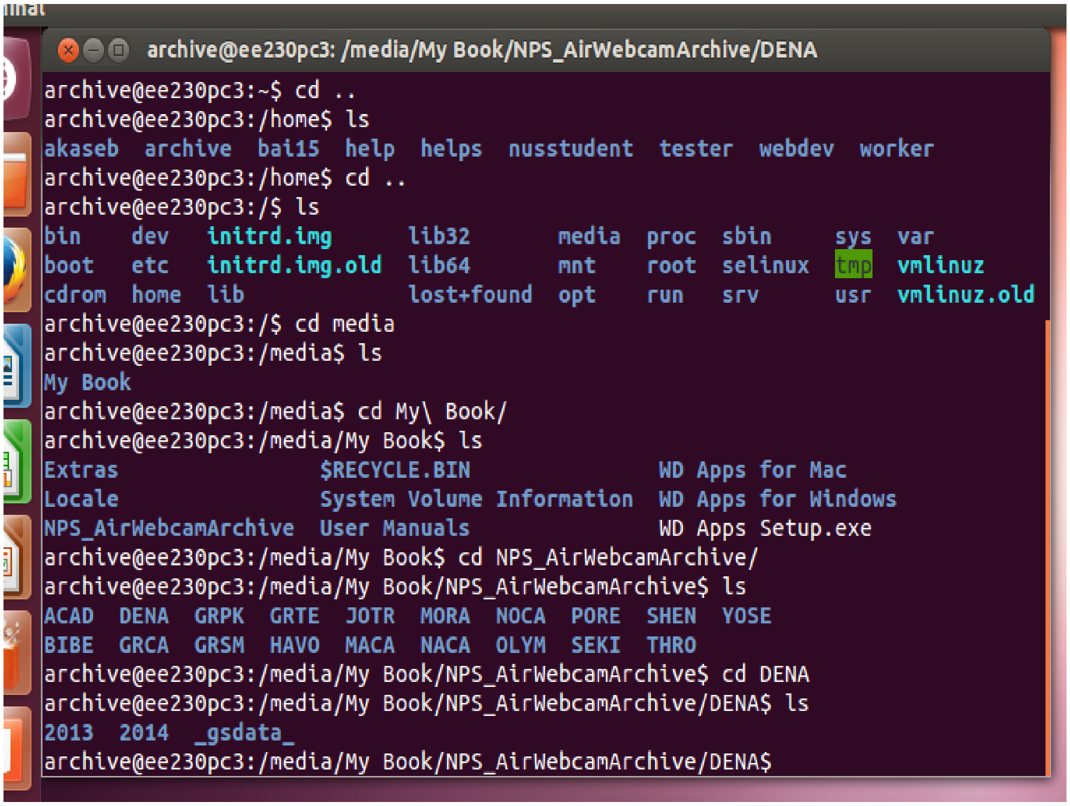
# Appendix B:

# Accessing Archived Data on My Book



```
root@ee230pc3: /media/My Book/NPS_AirWebcamArchive/YOSE/2013
YOSE201304251900.jpg  YOSE201308190400.jpg  YOSE201312311130.jpg
YOSE201304251930.jpg  YOSE201308190430.jpg  YOSE201312311200.jpg
YOSE201304252000.jpg  YOSE201308190500.jpg  YOSE201312311230.jpg
YOSE201304252030.jpg  YOSE201308190530.jpg  YOSE201312311300.jpg
YOSE201304252100.jpg  YOSE201308190600.jpg  YOSE201312311330.jpg
YOSE201304252130.jpg  YOSE201308190630.jpg  YOSE201312311400.jpg
YOSE201304252200.jpg  YOSE201308190700.jpg  YOSE201312311430.jpg
YOSE201304252230.jpg  YOSE201308190730.jpg  YOSE201312311500.jpg
YOSE201304252300.jpg  YOSE201308190800.jpg  YOSE201312311530.jpg
YOSE201304252330.jpg  YOSE201308190830.jpg  YOSE201312311600.jpg
YOSE201304260000.jpg  YOSE201308190900.jpg  YOSE201312311630.jpg
YOSE201304260030.jpg  YOSE201308190930.jpg  YOSE201312311700.jpg
YOSE201304260100.jpg  YOSE201308191000.jpg  YOSE201312311730.jpg
YOSE201304260130.jpg  YOSE201308191030.jpg  YOSE201312311800.jpg
YOSE201304260200.jpg  YOSE201308191100.jpg  YOSE201312311830.jpg
YOSE201304260230.jpg  YOSE201308191130.jpg  YOSE201312311900.jpg
YOSE201304260300.jpg  YOSE201308191200.jpg  YOSE201312311930.jpg
YOSE201304260330.jpg  YOSE201308191230.jpg  YOSE201312312000.jpg
YOSE201304260400.jpg  YOSE201308191300.jpg  YOSE201312312030.jpg
YOSE201304260430.jpg  YOSE201308191330.jpg  YOSE201312312100.jpg
YOSE201304260500.jpg  YOSE201308191400.jpg  YOSE201312312130.jpg
YOSE201304260530.jpg  YOSE201308191430.jpg
YOSE201304260600.jpg  YOSE201308191500.jpg
root@ee230pc3:/media/My Book/NPS_AirWebcamArchive/YOSE/2013#
```

Figure 1: Only able to access archive images as root user

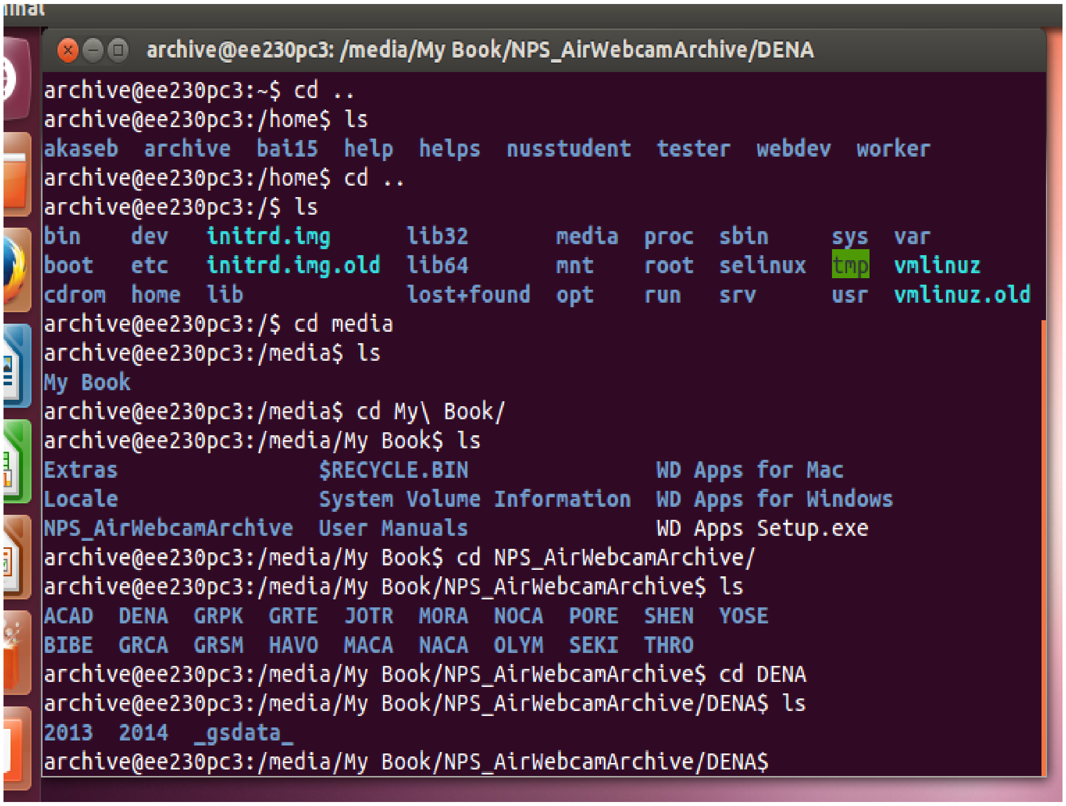


```
archive@ee230pc3: /media/My Book/NPS_AirWebcamArchive/DENA
archive@ee230pc3:~$ cd ..
archive@ee230pc3:/home$ ls
akaseb  archive  bai15  help  helps  nusstudent  tester  webdev  worker
archive@ee230pc3:/home$ cd ..
archive@ee230pc3:/$ ls
bin    dev   initrd.img      lib32       media  proc  sbin     sys  var
boot   etc   initrd.img.old  lib64       mnt    root  selinux  tmp  vmlinuz
cdrom  home  lib             lost+found  opt    run   srv      usr  vmlinuz.old
archive@ee230pc3:/$ cd media
archive@ee230pc3:/media$ ls
My Book
archive@ee230pc3:/media$ cd My\ Book/
archive@ee230pc3:/media/My Book$ ls
Extras               $RECYCLE.BIN               WD Apps for Mac
Locale               System Volume Information  WD Apps for Windows
NPS_AirWebcamArchive  User Manuals              WD Apps Setup.exe
archive@ee230pc3:/media/My Book$ cd NPS_AirWebcamArchive/
archive@ee230pc3:/media/My Book/NPS_AirWebcamArchive$ ls
ACAD  DENA  GRPK  GRTE  JOTR  MORA  NOCA  PORE  SHEN  YOSE
BIBE  GRCA  GRSM  HAVO  MACA  NACA  OLYM  SEKI  THRO
archive@ee230pc3:/media/My Book/NPS_AirWebcamArchive$ cd DENA
archive@ee230pc3:/media/My Book/NPS_AirWebcamArchive/DENA$ ls
2013  2014  _gsdata_
archive@ee230pc3:/media/My Book/NPS_AirWebcamArchive/DENA$
```

Figure 2: Archive user was created  while connected to archive drive which granted data access

# Appendix C:

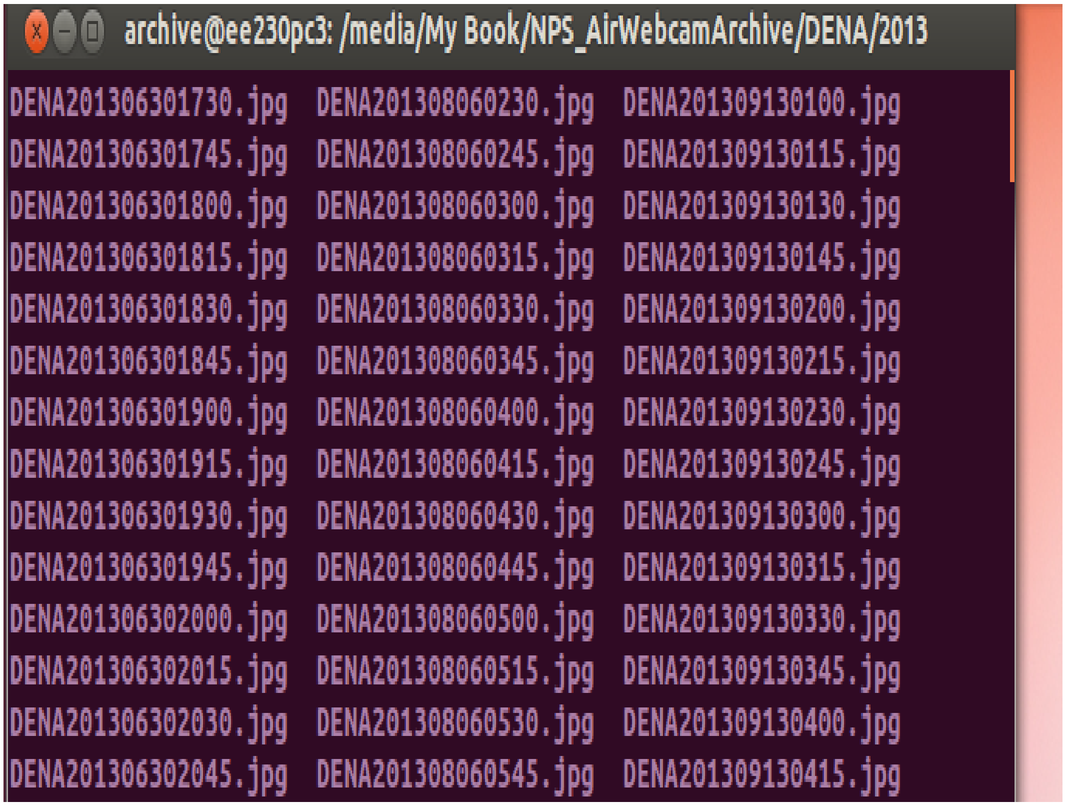
# Initial Ideas



archive@ee230pc3: /media/My Book/NPS_AirWebcamArchive/DENA

```
archive@ee230pc3:~$ cd ..
archive@ee230pc3:/home$ ls
akaseb  archive  bai15  help  helps  nusstudent  tester  webdev  worker
archive@ee230pc3:/home$ cd ..
archive@ee230pc3:/$ ls
bin    dev   initrd.img      lib32       media  proc  sbin     sys  var
boot   etc   initrd.img.old  lib64       mnt    root  selinux  tmp  vmlinuz
cdrom  home  lib             lost+found  opt    run   srv      usr  vmlinuz.old
archive@ee230pc3:/$ cd media
archive@ee230pc3:/media$ ls
My Book
archive@ee230pc3:/media$ cd My\ Book/
archive@ee230pc3:/media/My Book$ ls
Extras               $RECYCLE.BIN               WD Apps for Mac
Locale               System Volume Information  WD Apps for Windows
NPS_AirWebcamArchive User Manuals               WD Apps Setup.exe
archive@ee230pc3:/media/My Book$ cd NPS_AirWebcamArchive/
archive@ee230pc3:/media/My Book/NPS_AirWebcamArchive$ ls
ACAD  DENA  GRPK  GRTE  JOTR  MORA  NOCA  PORE  SHEN  YOSE
BIBE  GRCA  GRSM  HAVO  MACA  NACA  OLYM  SEKI  THRO
archive@ee230pc3:/media/My Book/NPS_AirWebcamArchive$ cd DENA
archive@ee230pc3:/media/My Book/NPS_AirWebcamArchive/DENA$ ls
2013  2014  _gsdata_
archive@ee230pc3:/media/My Book/NPS_AirWebcamArchive/DENA$
```

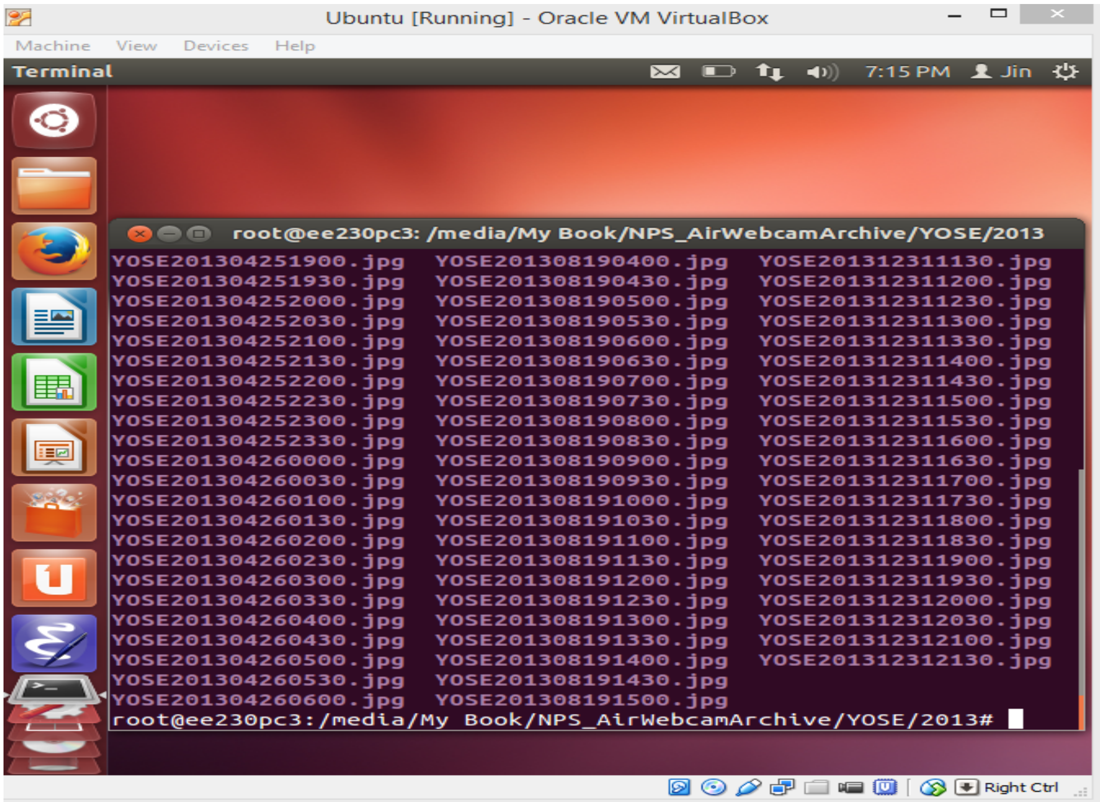
Figure 1: Structure of My Book storage directories are very well organized



archive@ee230pc3: /media/My Book/NPS_AirWebcamArchive/DENA/2013

```
DENA201306301730.jpg  DENA201308060230.jpg  DENA201309130100.jpg
DENA201306301745.jpg  DENA201308060245.jpg  DENA201309130115.jpg
DENA201306301800.jpg  DENA201308060300.jpg  DENA201309130130.jpg
DENA201306301815.jpg  DENA201308060315.jpg  DENA201309130145.jpg
DENA201306301830.jpg  DENA201308060330.jpg  DENA201309130200.jpg
DENA201306301845.jpg  DENA201308060345.jpg  DENA201309130215.jpg
DENA201306301900.jpg  DENA201308060400.jpg  DENA201309130230.jpg
DENA201306301915.jpg  DENA201308060415.jpg  DENA201309130245.jpg
DENA201306301930.jpg  DENA201308060430.jpg  DENA201309130300.jpg
DENA201306301945.jpg  DENA201308060445.jpg  DENA201309130315.jpg
DENA201306302000.jpg  DENA201308060500.jpg  DENA201309130330.jpg
DENA201306302015.jpg  DENA201308060515.jpg  DENA201309130345.jpg
DENA201306302030.jpg  DENA201308060530.jpg  DENA201309130400.jpg
DENA201306302045.jpg  DENA201308060545.jpg  DENA201309130415.jpg
```

Figure 2a: Image files are all named in a similar fashion


Ubuntu [Running] - Oracle VM VirtualBox
Machine
View
Devices
Help
Terminal
7:15 PM
Jin
root@ee230pc3: /media/My Book/NPS_AirWebcamArchive/YOSE/2013
Right Ctrl

```
YOSE201304251900.jpg  YOSE201308190400.jpg  YOSE201312311130.jpg
YOSE201304251930.jpg  YOSE201308190430.jpg  YOSE201312311200.jpg
YOSE201304252000.jpg  YOSE201308190500.jpg  YOSE201312311230.jpg
YOSE201304252030.jpg  YOSE201308190530.jpg  YOSE201312311300.jpg
YOSE201304252100.jpg  YOSE201308190600.jpg  YOSE201312311330.jpg
YOSE201304252130.jpg  YOSE201308190630.jpg  YOSE201312311400.jpg
YOSE201304252200.jpg  YOSE201308190700.jpg  YOSE201312311430.jpg
YOSE201304252230.jpg  YOSE201308190730.jpg  YOSE201312311500.jpg
YOSE201304252300.jpg  YOSE201308190800.jpg  YOSE201312311530.jpg
YOSE201304252330.jpg  YOSE201308190830.jpg  YOSE201312311600.jpg
YOSE201304260000.jpg  YOSE201308190900.jpg  YOSE201312311630.jpg
YOSE201304260030.jpg  YOSE201308190930.jpg  YOSE201312311700.jpg
YOSE201304260100.jpg  YOSE201308191000.jpg  YOSE201312311730.jpg
YOSE201304260130.jpg  YOSE201308191030.jpg  YOSE201312311800.jpg
YOSE201304260200.jpg  YOSE201308191100.jpg  YOSE201312311830.jpg
YOSE201304260230.jpg  YOSE201308191130.jpg  YOSE201312311900.jpg
YOSE201304260300.jpg  YOSE201308191200.jpg  YOSE201312311930.jpg
YOSE201304260330.jpg  YOSE201308191230.jpg  YOSE201312312000.jpg
YOSE201304260400.jpg  YOSE201308191300.jpg  YOSE201312312030.jpg
YOSE201304260430.jpg  YOSE201308191330.jpg  YOSE201312312100.jpg
YOSE201304260500.jpg  YOSE201308191400.jpg  YOSE201312312130.jpg
YOSE201304260530.jpg  YOSE201308191430.jpg
YOSE201304260600.jpg  YOSE201308191500.jpg
root@ee230pc3:/media/My Book/NPS_AirWebcamArchive/YOSE/2013#
```

Figure 2b: Image files are all named in a similar fashion

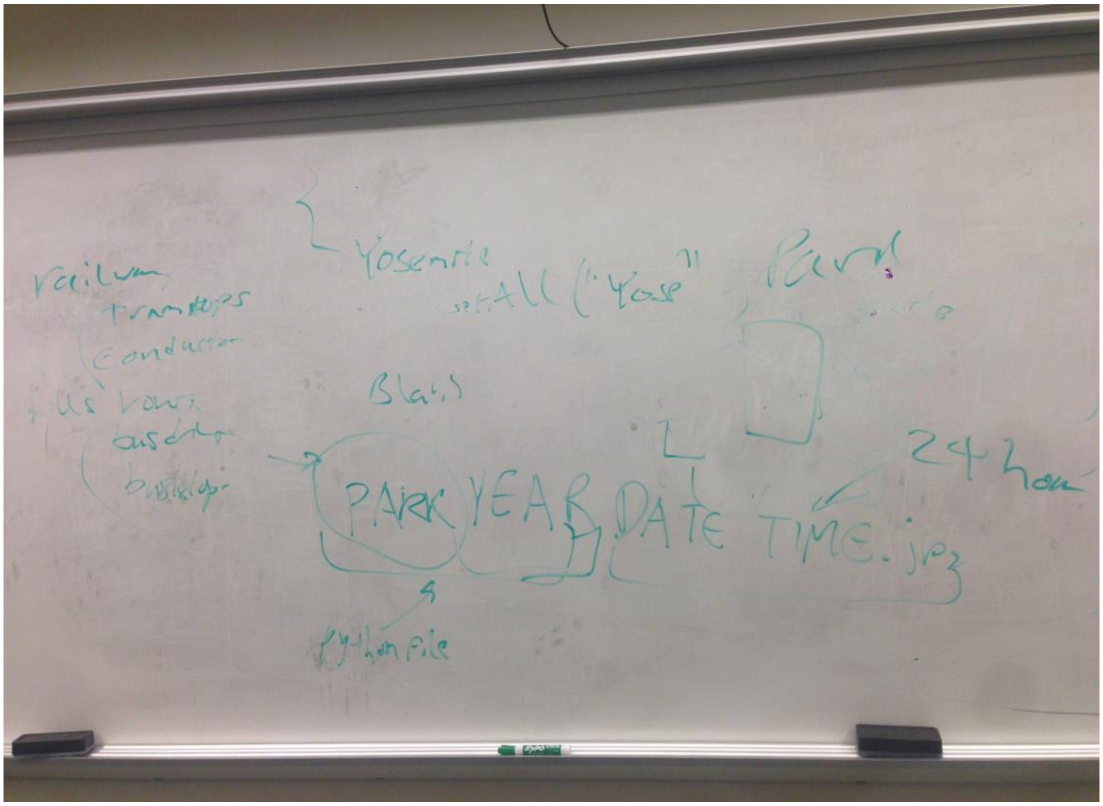
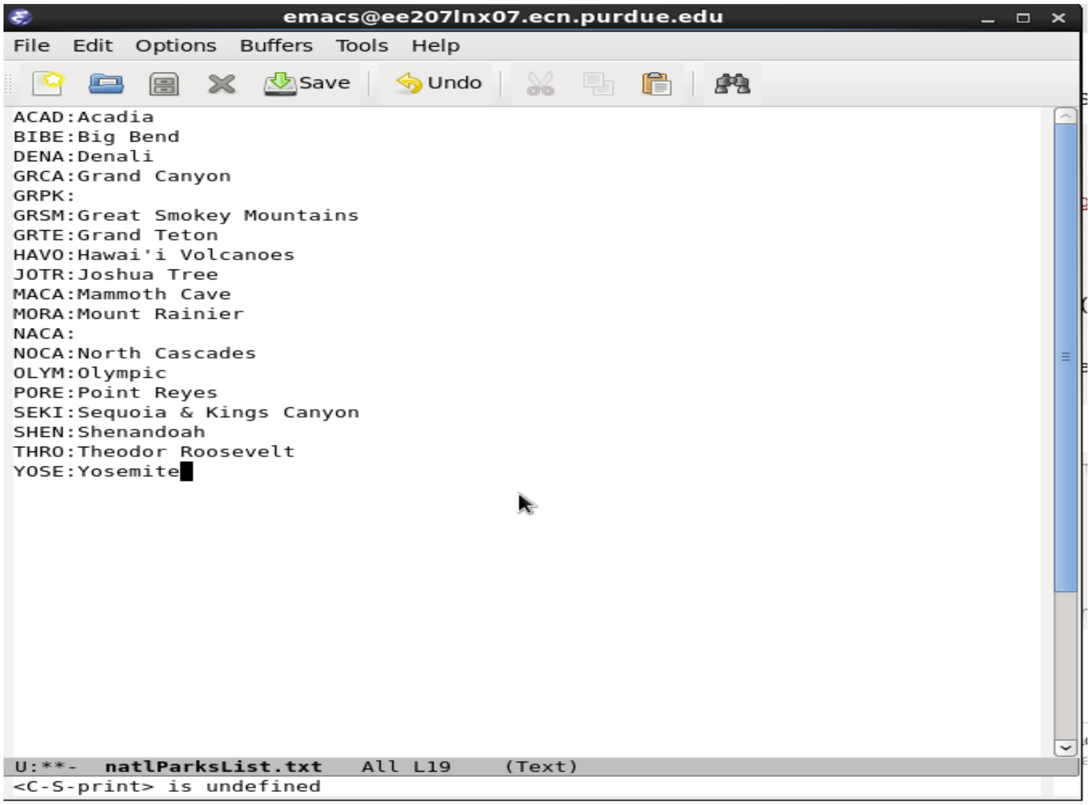
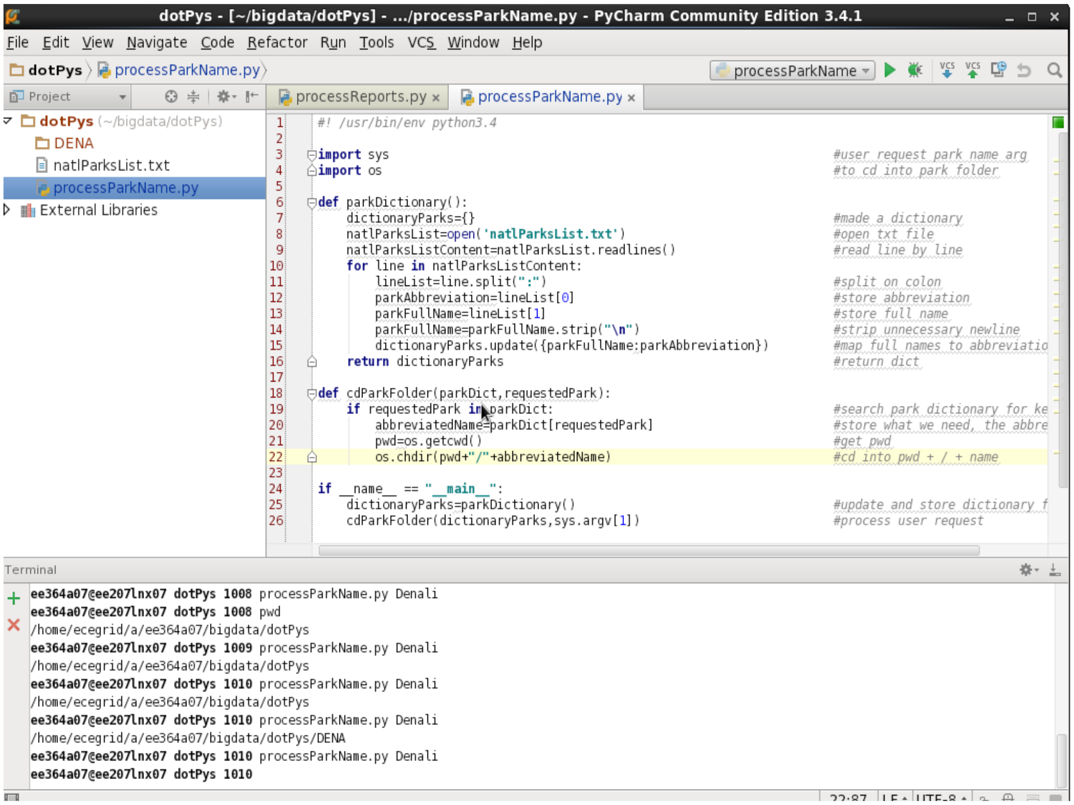


Figure 3: All names are in the format <Park Abbreviation><Year><Date><Time>


emacs@ee207lnx07.ecn.purdue.edu
File Edit Options Buffers Tools Help
Save
Undo

```
ACAD:Acadia
BIBE:Big Bend
DENA:Denali
GRCA:Grand Canyon
GRPK:
GRSM:Great Smokey Mountains
GRTE:Grand Teton
HAVO:Hawai'i Volcanoes
JOTR:Joshua Tree
MACA:Mammoth Cave
MORA:Mount Rainier
NACA:
NOCA:North Cascades
OLYM:Olympic
PORE:Point Reyes
SEKI:Sequoia & Kings Canyon
SHEN:Shenandoah
THRO:Theodor Roosevelt
YOSE:Yosemite

U:**-  natlParksList.txt    All L19    (Text)
<C-S-print> is undefined
```

Figure 4: Dictionary key to value mapping idea, text file to store data


dotPys - [~/bigdata/dotPys] - .../processParkName.py - PyCharm Community Edition 3.4.1
File Edit View Navigate Code Refactor Run Tools VCS Window Help
dotPys
processParkName.py
processParkName
Project
processReports.py
processParkName.py
dotPys (~/bigdata/dotPys)
DENA
natlParksList.txt
processParkName.py
External Libraries
Terminal

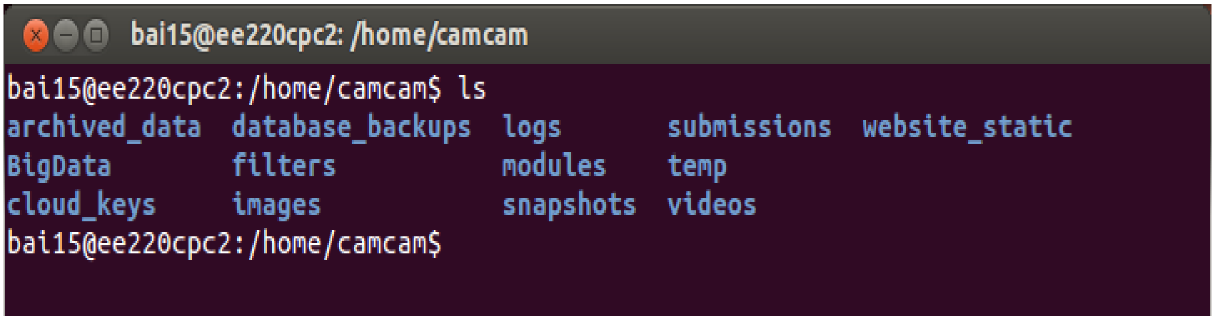
```python
#! /usr/bin/env python3.4

import sys                                                          #user request park name arg
import os                                                           #to cd into park folder

def parkDictionary():
    dictionaryParks={}                                              #made a dictionary
    natlParksList=open('natlParksList.txt')                         #open txt file
    natlParksListContent=natlParksList.readlines()                  #read line by line
    for line in natlParksListContent:
        lineList=line.split(":")                                    #split on colon
        parkAbbreviation=lineList[0]                                #store abbreviation
        parkFullName=lineList[1]                                    #store full name
        parkFullName=parkFullName.strip("\n")                       #strip unnecessary newline
        dictionaryParks.update({parkFullName:parkAbbreviation})     #map full names to abbreviatio
    return dictionaryParks                                          #return dict

def cdParkFolder(parkDict,requestedPark):
    if requestedPark in parkDict:                                   #search park dictionary for ke
        abbreviatedName=parkDict[requestedPark]                     #store what we need, the abbre
        pwd=os.getcwd()                                             #get pwd
        os.chdir(pwd+"/"+abbreviatedName)                           #cd into pwd + / + name

if __name__ == "__main__":
    dictionaryParks=parkDictionary()                                #update and store dictionary f
    cdParkFolder(dictionaryParks,sys.argv[1])                       #process user request
```

```
ee364a07@ee207lnx07 dotPys 1008 processParkName.py Denali
ee364a07@ee207lnx07 dotPys 1008 pwd
/home/ecegrid/a/ee364a07/bigdata/dotPys
ee364a07@ee207lnx07 dotPys 1009 processParkName.py Denali
/home/ecegrid/a/ee364a07/bigdata/dotPys
ee364a07@ee207lnx07 dotPys 1010 processParkName.py Denali
/home/ecegrid/a/ee364a07/bigdata/dotPys
ee364a07@ee207lnx07 dotPys 1010 processParkName.py Denali
/home/ecegrid/a/ee364a07/bigdata/dotPys/DENA
ee364a07@ee207lnx07 dotPys 1010 processParkName.py Denali
ee364a07@ee207lnx07 dotPys 1010
```

Figure 5: Implementation in Python

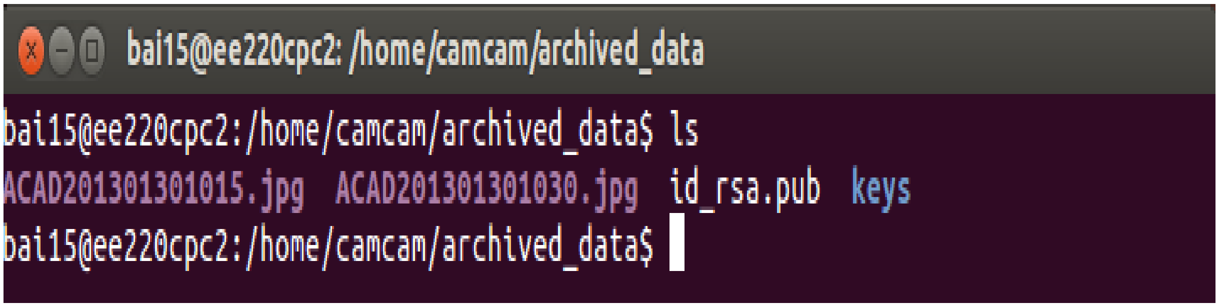
# Appendix D:

# Database Auto-population and Changes



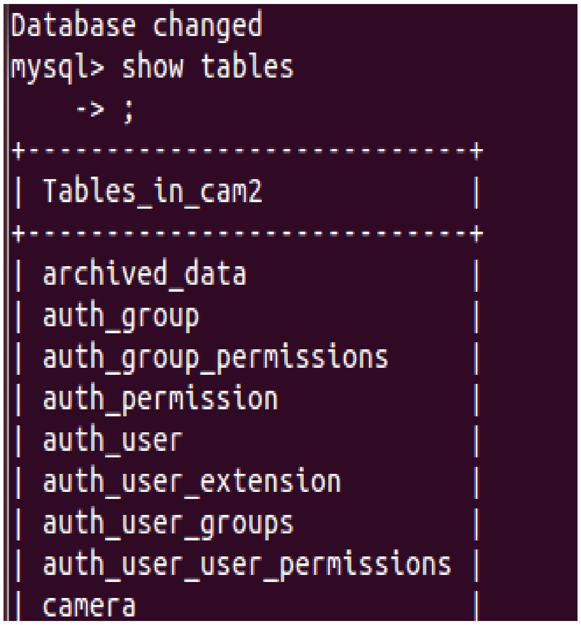
```
bai15@ee220cpc2:/home/camcam$ ls
archived_data  database_backups  logs       submissions  website_static
BigData        filters           modules    temp
cloud_keys     images            snapshots  videos
bai15@ee220cpc2:/home/camcam$
```

Figure 1: Added archived_data directory to camcam folder



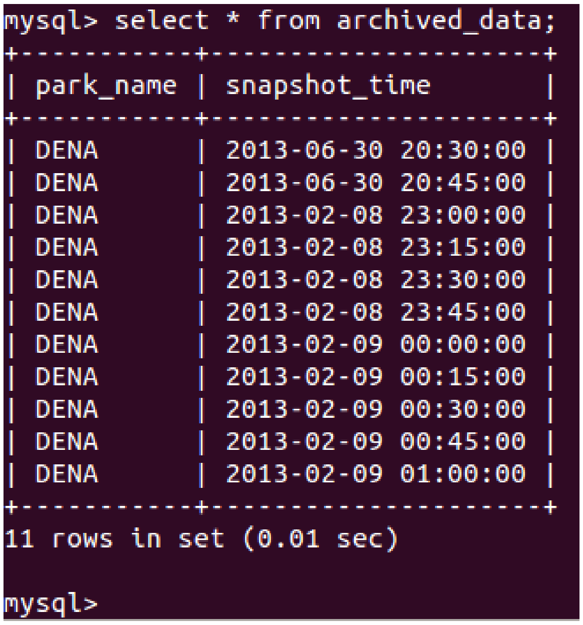
```
bai15@ee220cpc2:/home/camcam/archived_data$ ls
ACAD201301301015.jpg  ACAD201301301030.jpg  id_rsa.pub  keys
bai15@ee220cpc2:/home/camcam/archived_data$
```

Figure 2: Able to transfer single images from the archived machine into the dev machine



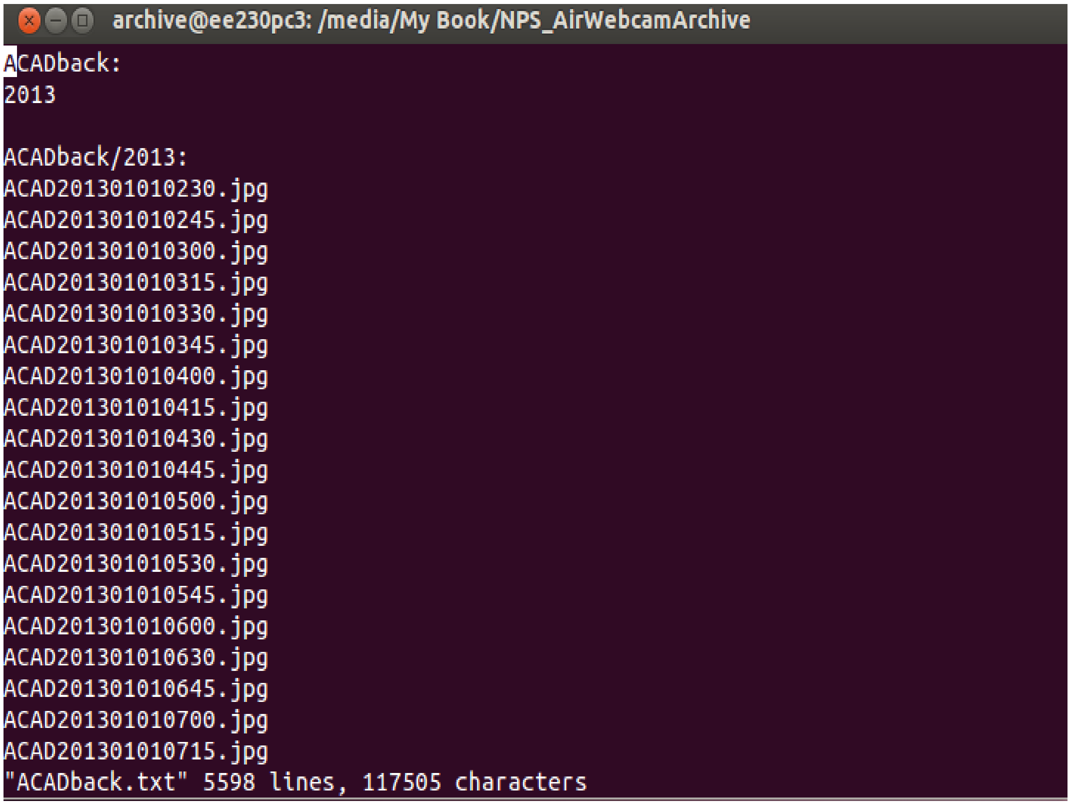
```
Database changed
mysql> show tables
    -> ;
+----------------------------+
| Tables_in_cam2             |
+----------------------------+
| archived_data              |
| auth_group                 |
| auth_group_permissions     |
| auth_permission            |
| auth_user                  |
| auth_user_extension        |
| auth_user_groups           |
| auth_user_user_permissions |
| camera                     |
```

Figure 3: Created archived_data table in DB



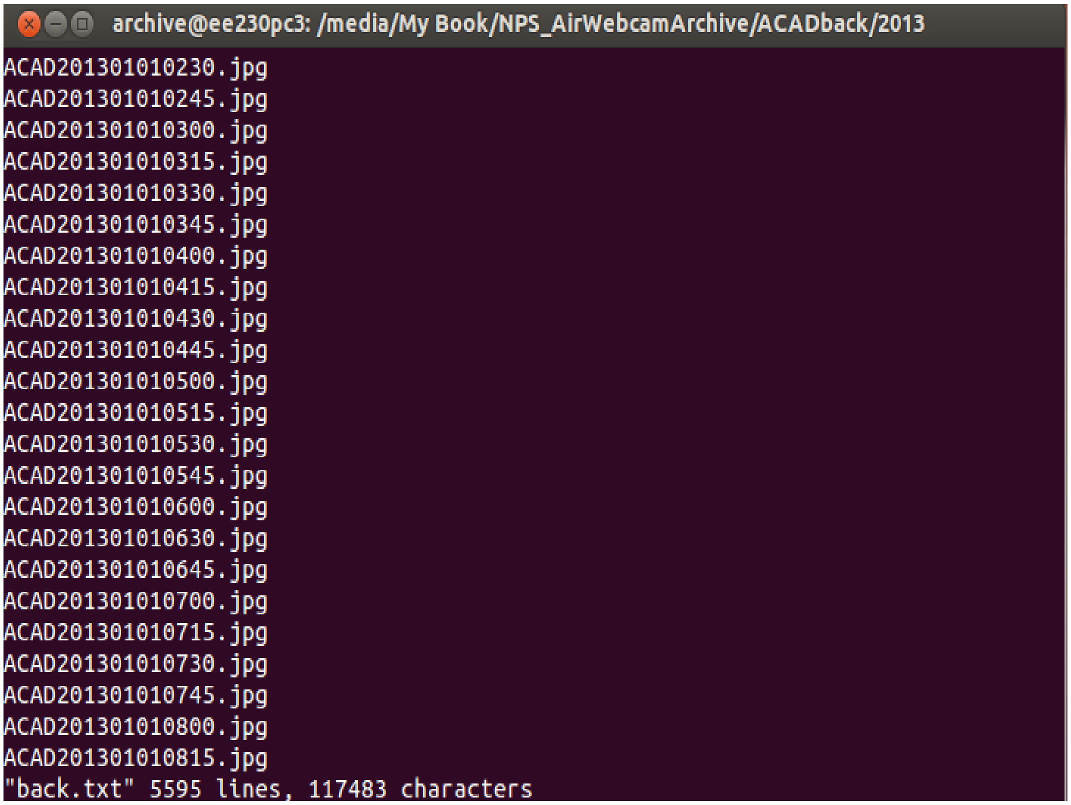
```
mysql> select * from archived_data;
+-----------+---------------------+
| park_name | snapshot_time       |
+-----------+---------------------+
| DENA      | 2013-06-30 20:30:00 |
| DENA      | 2013-06-30 20:45:00 |
| DENA      | 2013-02-08 23:00:00 |
| DENA      | 2013-02-08 23:15:00 |
| DENA      | 2013-02-08 23:30:00 |
| DENA      | 2013-02-08 23:45:00 |
| DENA      | 2013-02-09 00:00:00 |
| DENA      | 2013-02-09 00:15:00 |
| DENA      | 2013-02-09 00:30:00 |
| DENA      | 2013-02-09 00:45:00 |
| DENA      | 2013-02-09 01:00:00 |
+-----------+---------------------+
11 rows in set (0.01 sec)

mysql>
```

Figure 4: Manually input values into the table



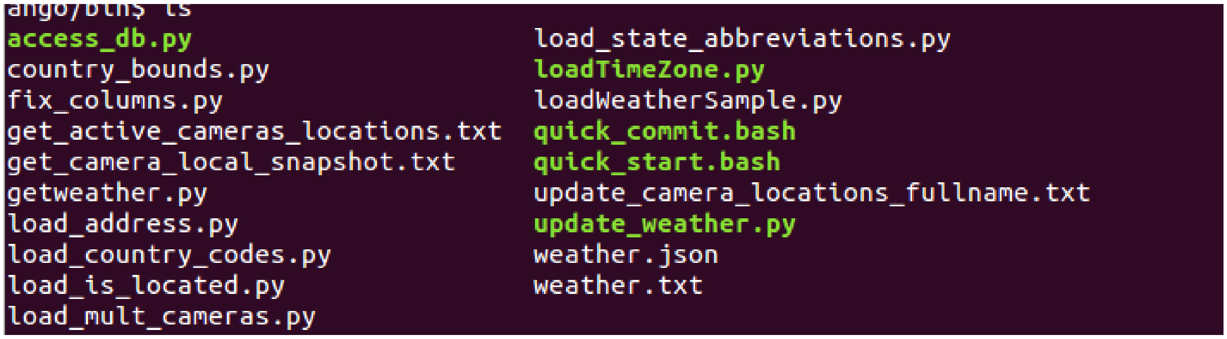
```
ACADback:
2013

ACADback/2013:
ACAD201301010230.jpg
ACAD201301010245.jpg
ACAD201301010300.jpg
ACAD201301010315.jpg
ACAD201301010330.jpg
ACAD201301010345.jpg
ACAD201301010400.jpg
ACAD201301010415.jpg
ACAD201301010430.jpg
ACAD201301010445.jpg
ACAD201301010500.jpg
ACAD201301010515.jpg
ACAD201301010530.jpg
ACAD201301010545.jpg
ACAD201301010600.jpg
ACAD201301010630.jpg
ACAD201301010645.jpg
ACAD201301010700.jpg
ACAD201301010715.jpg
"ACADback.txt" 5598 lines, 117505 characters
```

Figure 5: Outputting all the image filename data into a text file


archive@ee230pc3: /media/My Book/NPS_AirWebcamArchive/ACADback/2013

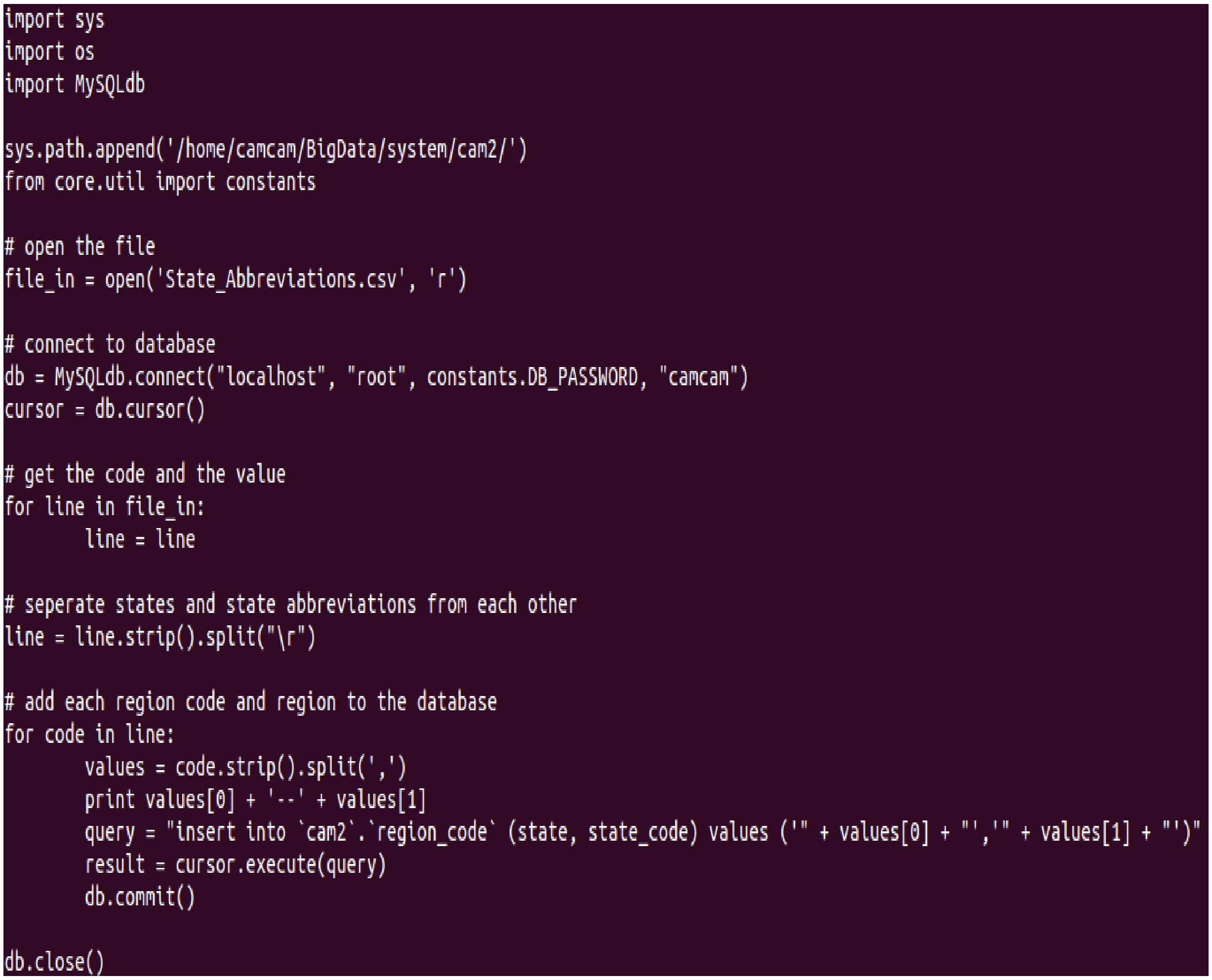
```
ACAD201301010230.jpg
ACAD201301010245.jpg
ACAD201301010300.jpg
ACAD201301010315.jpg
ACAD201301010330.jpg
ACAD201301010345.jpg
ACAD201301010400.jpg
ACAD201301010415.jpg
ACAD201301010430.jpg
ACAD201301010445.jpg
ACAD201301010500.jpg
ACAD201301010515.jpg
ACAD201301010530.jpg
ACAD201301010545.jpg
ACAD201301010600.jpg
ACAD201301010630.jpg
ACAD201301010645.jpg
ACAD201301010700.jpg
ACAD201301010715.jpg
ACAD201301010730.jpg
ACAD201301010745.jpg
ACAD201301010800.jpg
ACAD201301010815.jpg
"back.txt" 5595 lines, 117483 characters
```

Figure 6: Performed same ls –R > outfile command on inner directory to check lines to confirm that we have all the files



```
ango/bin$ ls
access_db.py                      load_state_abbreviations.py
country_bounds.py                 loadTimeZone.py
fix_columns.py                    loadWeatherSample.py
get_active_cameras_locations.txt  quick_commit.bash
get_camera_local_snapshot.txt     quick_start.bash
getweather.py                     update_camera_locations_fullname.txt
load_address.py                   update_weather.py
load_country_codes.py             weather.json
load_is_located.py                weather.txt
load_mult_cameras.py
```

Figure 7: Learn how to auto-populate the database through these files

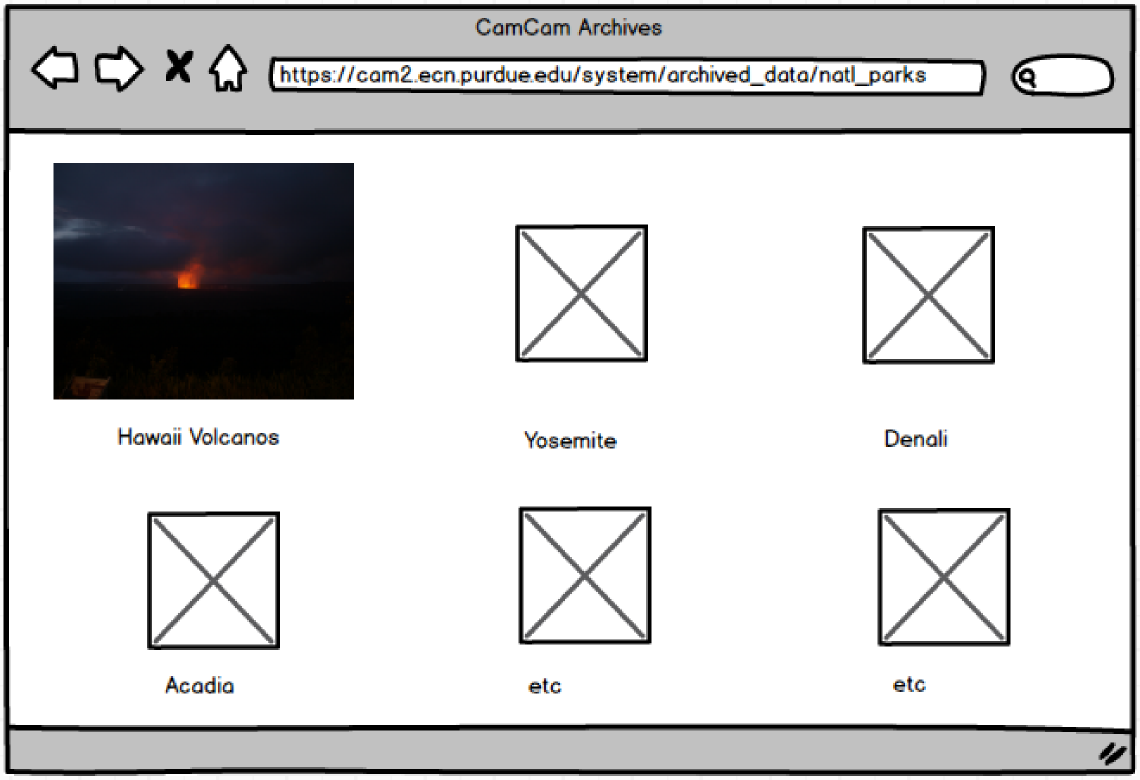


```
import sys
import os
import MySQLdb

sys.path.append('/home/camcam/BigData/system/cam2/')
from core.util import constants

# open the file
file_in = open('State_Abbreviations.csv', 'r')

# connect to database
db = MySQLdb.connect("localhost", "root", constants.DB_PASSWORD, "camcam")
cursor = db.cursor()

# get the code and the value
for line in file_in:
        line = line

# seperate states and state abbreviations from each other
line = line.strip().split("\r")

# add each region code and region to the database
for code in line:
        values = code.strip().split(',')
        print values[0] + '--' + values[1]
        query = "insert into `cam2`.`region_code` (state, state_code) values ('" + values[0] + "','" + values[1] + "')"
        result = cursor.execute(query)
        db.commit()

db.close()
```

Figure 8: State abbreviations load into DB file example

# Appendix E:

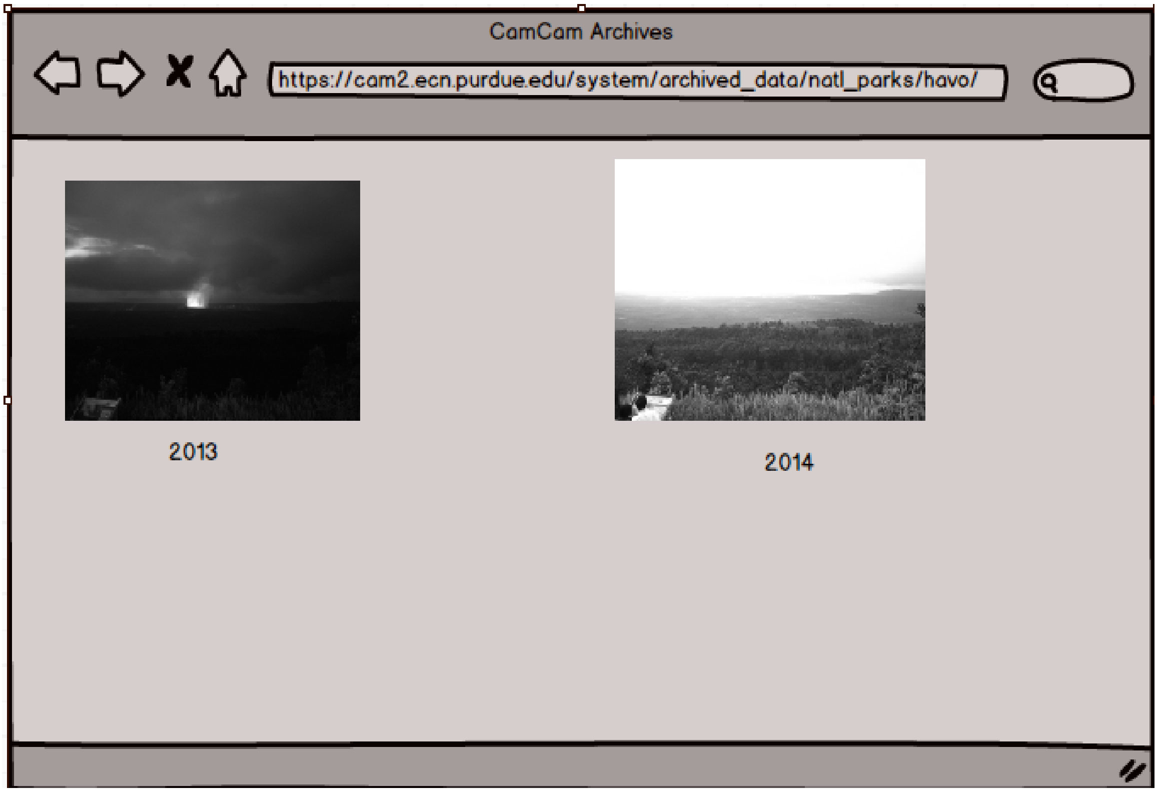
# Front End Design Possibilities



CamCam Archives

https://cam2.ecn.purdue.edu/system/archived_data/natl_parks

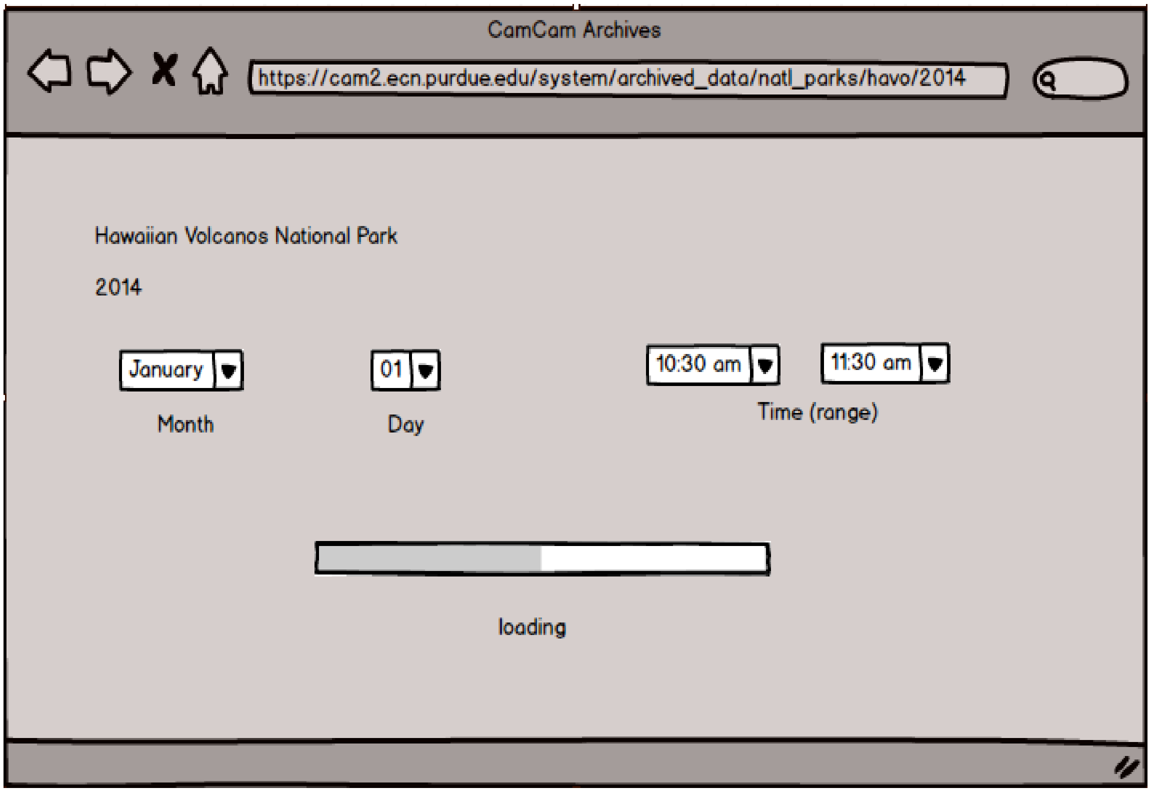
Figure 1: Front end idea, select category archived_data then nat'l parks, display parks



CamCam Archives

https://cam2.ecn.purdue.edu/system/archived_data/natl_parks/havo/

Figure 2: After selection of park, show year options



CamCam Archives

https://cam2.ecn.purdue.edu/system/archived_data/natl_parks/havo/2014

Hawaiian Volcanos National Park

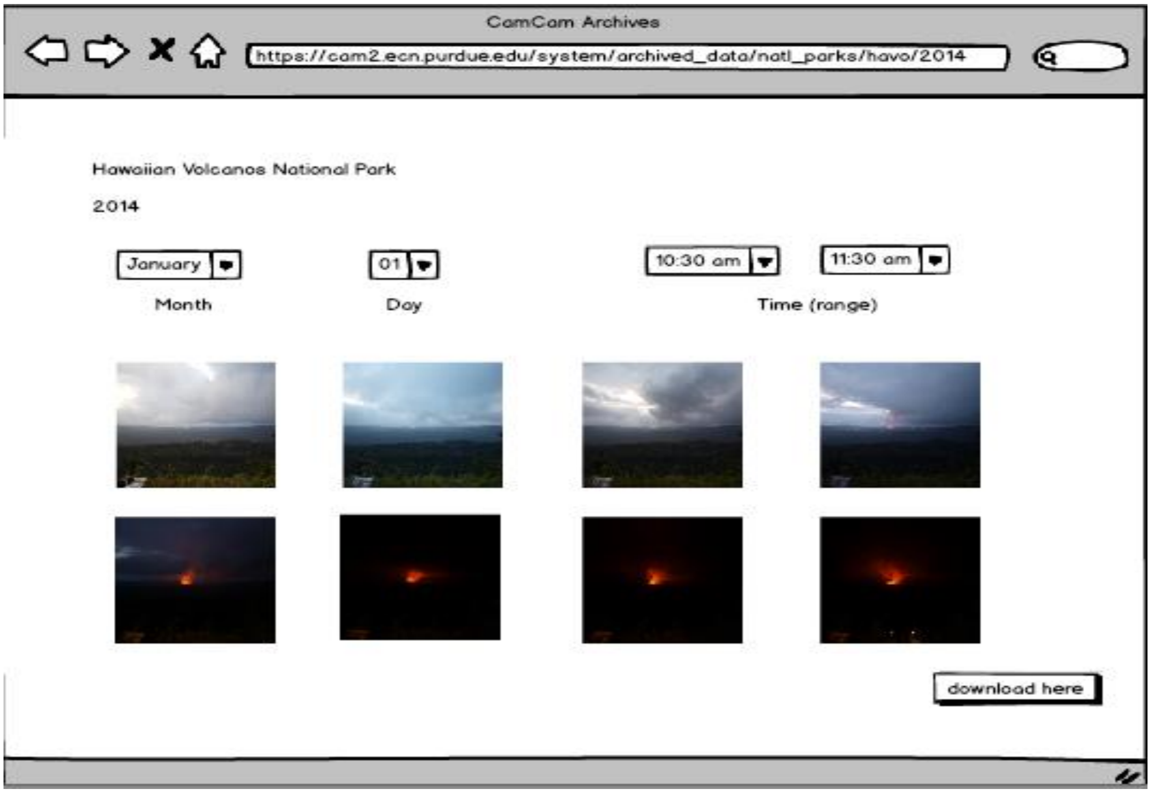
2014

Month: January

Day: 01

Time (range): 10:30 am – 11:30 am

Figure 3: After selection of year, show month, day, and time options



CamCam Archives

https://cam2.ecn.purdue.edu/system/archived_data/natl_parks/havo/2014

Hawaiian Volcanos National Park

2014

Month: January

Day: 01

Time (range): 10:30 am – 11:30 am

Figure 4: After desired options selected, show images