User Guide

Thanks for the guidance from Dr. Antoniadis
Thanks also to Abdullah Alhussainy

Luyuan Zeng

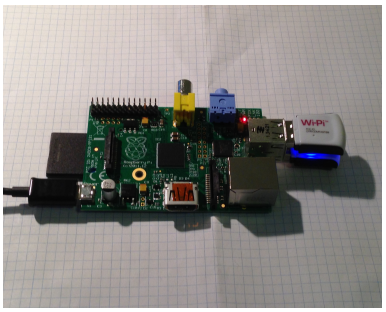
Welcome to the Applications for Offline Networks

Thanks for the source code from the two websites:
`http://www.foliopages.com`
`http://www.phpkobo.com/ajax_poll.php`
which I modified to suit this local application.

According to the installation guide of setting up a rapsberry pi as a captive portal made by Dr. Antoniadis and Abdullah Alhussainy, the raspberry pi is set up as a captive portal and is able to communicate the data from the local area to the server with two wifi adapters.

If you would like to know more about setting up a raspberry pi as a captive portal, please contact Dr. Antoniadis

Please find what we need in the GitHub Repository
`https://github.com/nethood/localapps`,



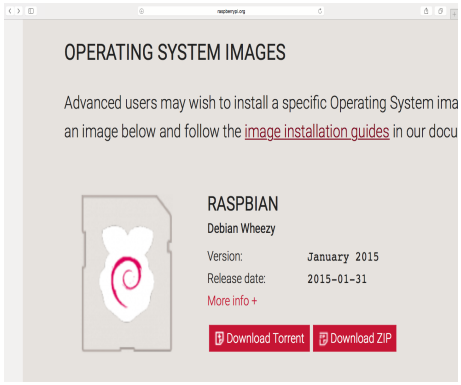
Here is the equipments we need: one raspberry pi, one ethernet cable, one USB cable for the power supply, one SD card and two wifi adapters.

Note: We are not responsible for the photos that have been uploaded, the questions chosen by the ones who have taken the role as the administrator of this application, and the messages that have been left on this local applications as well.

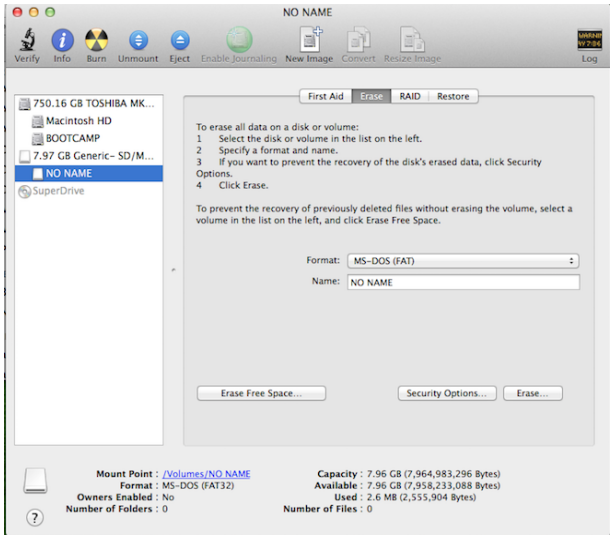
Setting up raspberry pi as a captive portal

Note: This covers the basic stuff we need to set up a raspberry pi as a captive portal with two wifi adapters. If you have further needs, for example limiting the speed and quota for the users connected, please connect Dr. Antoniadis

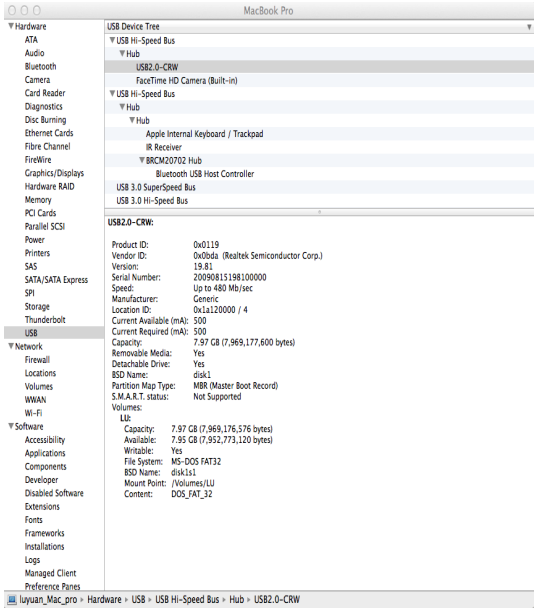
First of all, we should download the image of the operating system for the new raspberry pi from `http://www.raspberrypi.org/downloads/`, we choose Raspbian. Then we click the image installation guides on the web page to choose the correct way of installing the Operating System Image according to our Operating System.



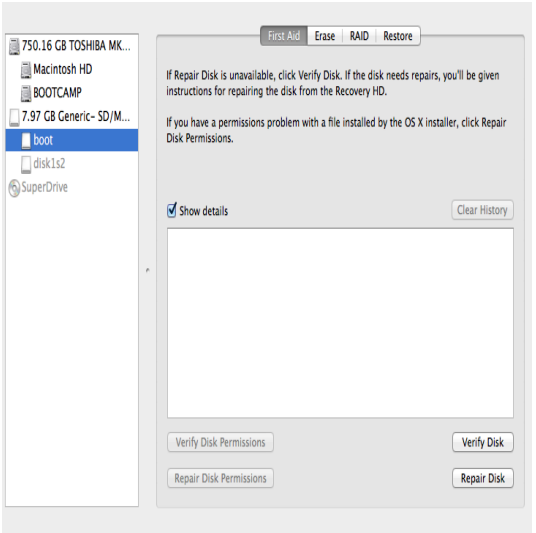
I made an example of Flash and Format the Operating System Image using Mac OS X by following the guide from `http://www.raspberrypi.org/documentation/installation/installing-images/mac.md`. First, insert the new SD card into the slot where it should be and then go to Disk Utility of the system. We will see our new SD card appears on the left column and then choose the format MS-DOS(FAT).



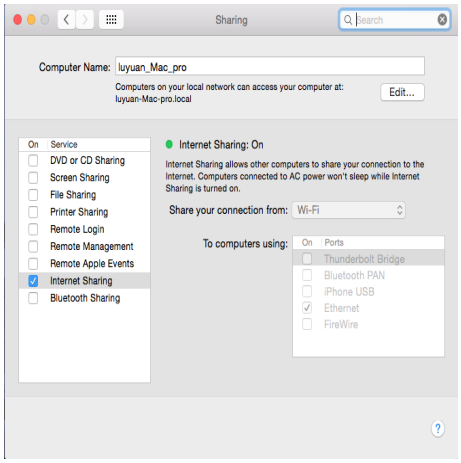
From the Apple menu, choose About This Mac, then click on System Report. Click on USB and search for the SD card on the right side of the window. Please search for the BSD name which must be like diskn(n is a number) and make sure that you take the note of this number. For example, disk1.



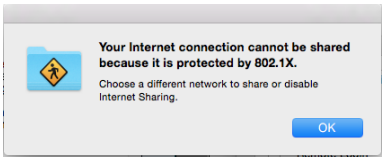
Go back to Disk Utility and then unmount it so that it is allowed to overwrite the disk. Finally, we should go back to the terminal, run sudo dd bs=1m if=(path of your image).img of=/dev/diskn Please note that diskn is the BSD name we have remembered before. We have to wait for a long time, if you would like to see the process, press Ctrl+T. When we finish, we will see boot from Disk Utility.



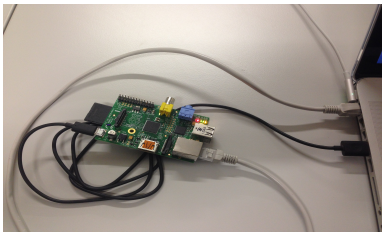
After installing the operating system image to the SD card, we **boot our raspberry pi**. In this case, we connect to our raspberry pi via SSH, so we do not have to use additional screen or keyboard. Firstly, we plug our raspberry pi to an energy source and then connect our rapsberry pi with laptop using an ethernet cable. In this way, we could share the Internet access of the computer from wifi to ethernet so this Internet connection can be used to communicate with the raspberry pi. For the MAC OS X user, please go to **System Preference→ Sharing → Internet Sharing → share the Internet Access from the WIFI to Ethernet**.



Note: For the MAC OS user, it is very important to know that we should choose the WIFI connection that is not protected by 802.1X, otherwise the internet connection can not be shared from the WIFI to the Ethernet. For example, in ETH, instead of choosing the WIFI eth or eduroam, we should choose the wifi Public to share the connection.



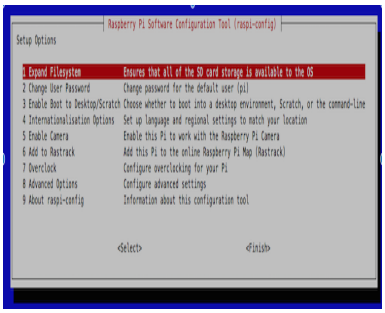
Now, we will have our computer and the raspberry pi connected like the following image.



Therefore, we can use ssh to connect the raspberry pi through the command line interface. First of all, we need to find the IP address of the raspberry pi. We use the command **arp -a**. From the graph below, we can figure out the IP address of the raspberry pi is **192.168.2.14**.Then we use the command **ssh pi@192.168.2.14**, **the password for a new raspberry pi is raspberry**

```
wlan-docking-hg-1-0293:~ luyuanzeng$ arp -a
rou-hg-1-wlan-docking-hg-1.ethz.ch (172.30.24.1) at d8:67:d9:e:73:c1 on en1 ifscope [ethernet]
wlan-docking-hg-1-1293.ethz.ch (172.30.29.13) at 8c:2d:aa:5d:b8:7f on en1 ifscope [ethernet]
wlan-docking-hg-1-2001.ethz.ch (172.30.31.209) at 60:3:8:70:86:64 on en1 ifscope [ethernet]
? (172.30.31.255) at ff:ff:ff:ff:ff:ff on en1 ifscope [ethernet]
? (192.168.2.14) at b8:27:eb:75:84:cc on bridge100 ifscope [bridge]
wlan-docking-hg-1-0293:~ luyuanzeng$
wlan-docking-hg-1-0293:~ luyuanzeng$
wlan-docking-hg-1-0293:~ luyuanzeng$
wlan-docking-hg-1-0293:~ luyuanzeng$ ssh pi@192.168.2.14
```

After accessing the raspberry pi successfully, we can do the reconfiguration of the new raspberry pi by the command **sudo raspi-config**. Then we should see something similar to the following image.

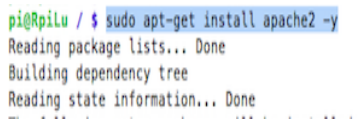


If you are able to ping a website, please skip the short instruction below. Otherwise, please type **cd /etc** to go to the /etc directory and add this line : **nameserver 8.8.8.8** to the **resolv.conf** file. Type **sudo nano resolv.conf** to edit this file, then press **Ctrl+x**, **click y** to save the editing.

```
nameserver 192.168.2.1
nameserver 8.8.8.8
```

Please type **sudo apt-get update** in the terminal to update the 'apt-get' catalog.

When the update finishes, we start installing Apache Webserver and PHP on the raspberry pi to allow it to host and serve web pages. Let us install **apache2** first by the following command **sudo apt-get install apache2 -y**



By default, the Apache server puts HTML files in the web folder, which is /**var**/**www**, so let us go to this folder by the command **cd** /**var**/**www** and then we list the information about the files using **ls -al**.

```
pi@raspberrypi /var $ cd www
pi@raspberrypi /var/www $ ls -al
total 12
drwxr-xr-x  2 root root 4096 Sep 18 21:32 .
drwxr-xr-x 12 root root 4096 Sep 18 21:32 ..
-rw-r--r--  1 root root  177 Sep 18 21:32 index.html
```

From the image above I find that the file **index.html** is owned by the root user so we need to get the root permission to read and write this file.

```
pi@raspberrypi /var/www $ cat index.html
<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
</body></html>
```

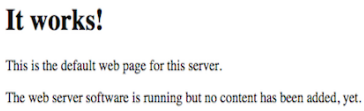
Therefore, we need to change the owner, the group and the permissions of the directory /var/www by the following command shown on the picture below.

```
pi@RpiLu / $ sudo chown www-data:www-data /var/www
pi@RpiLu / $ sudo chmod 775 /var/www
pi@RpiLu / $ sudo usermod -a -G www-data pi
```

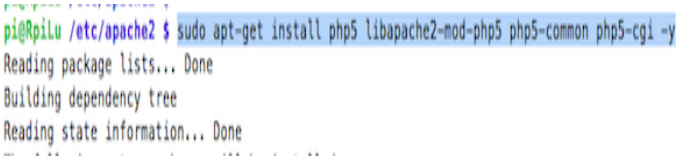
The commands above change directory owner and group of the directory /var/www to www-data, allow the group to write the directory and add the user pi to the www-data group. Please have a look at the Chown, Chmod, Chgrp, and How to add user to group to have a better understanding of the commands above.

In order to get the apache server to run correctly, we have to add **Server-name localhost** at the **Global configuration** section in the file **apache.conf**, which is located at /**etc**/**apache2**/ If we have modified the **apache2.conf** file. We can restart the server by running **sudo service apache2 restart**.

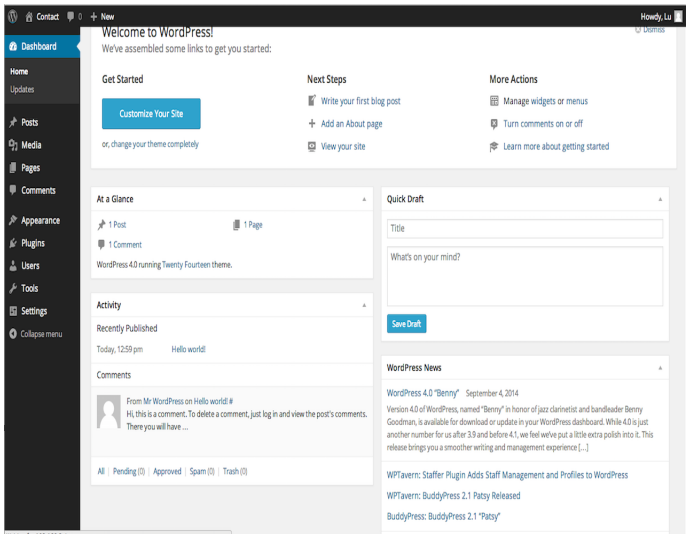
If we browse the IP address of your raspberry pi with the browser, we will see the default web page on the raspberry pi. For searching the IP address of the raspberry pi, we can use **hostname -I**.



In order to allow the apache server to serve the php files, we should install php5 and the relating models.Then let us install **PHP5** by the following command **sudo apt-get install php5 libapache2-mod-php5 php5common php5cgi –y**



By following the Raspberry Pi Learning Resource, we could also build our own blog by wordpress.



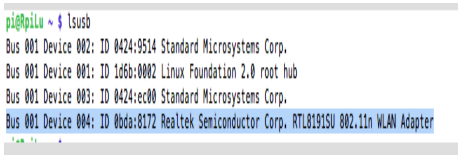
Let us start building the raspberry pi as a **wireless router with two interfaces wlan0 and wlan1** respectively. First of all, we should take a look at RPi USB Wi-Fi Adapters to know the wifi adpaters that are working on the raspberry pi. Please note that different wifi adapters may need different configuration files to make it work. In this case, we choose the wifi adapter **ASUS USBN13**.

hostapd is a user space daemon for wireless access point and authentication servers. Therefore, we need to install hostapd to create a wireless network

zone for the nearby potential users. First, we go to the terminal and type **sudo apt-get install hostapd**.

```
pi@RpiLu / $ sudo apt-get install hostapd
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Then plug in the wifi adapter to the USB port of the raspberry pi. The command **lsusb** allows you to a list of recognized devices on the raspberry pi. If you use the wifi adapter **ASUS USBN13**, you should see the **USB-N13** on the list.



From the image above, we can observe that I used another kind of wifi adaper instead of the **ASUS USBN13**, but I didn't make it work after trying a lot of ways of configuration. Finally, I used the **ASUS USBN13** wifi adapber and followed the installation guide provided by Dr. Antoniadis and Abdullah Alhussainy. There are many kinds of wifi adapters for us to choose, but we have to use the correct configuration to make it work.

Then we need to edit the interfaces file by the executing the following command in the terminal. First let us go to the **/etc/network** directory by typing **cd /etc/network**, then type **sudo nano interfaces** to edit this file. Firstly, we comment three lines of code as shown by the image below.

```
#auto lo
iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0

#iface wlan0 inet manual
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf

iface default inet dhcp
```

Then let us add the following lines to the bottom of this file as shown by the image below.

```
auto wlan0
iface wlan0 inet static
address 10.0.0.1
netmask 255.255.255.0
```

The IP address of wlan0 is 10.0.0.1 with netmask 255.255.255.0. We finishing edit this file then press **Ctrl+x, y, Enter** to save the file and exit.

The next step is to go to the directory **/etc/default** to edit the **hostapd** file. First, we go to the directory by **cd /etc/default**, then edit this file by **sudo nano hostapd**. We should uncomment the line **DAEMON_CONF = ""** by **removing the sign # that appears before the line** and type **/etc/hostapd/hostapd.conf** inside the quotes. In this way, the operating system will read the configuration file of the hostapd. This screenshot is an example.

```
# Defaults for hostapd initscript
#
# See /usr/share/doc/hostapd/README.Debian for information about alternative
# methods of managing hostapd.
#
# Uncomment and set DAEMON_CONF to the absolute path of a hostapd configuration
# file and hostapd will be started during system boot. An example configuration
# file can be found at /usr/share/doc/hostapd/examples/hostapd.conf.gz
#
DAEMON_CONF="/etc/hostapd/hostapd.conf"

# Additional daemon options to be appended to hostapd command:-
#       -d   show more debug messages (-dd for even more)
#       -K   include key data in debug messages
#       -t   include timestamps in some debug messages
#
# Note that -B (daemon mode) and -P (pidfile) options are automatically
# configured by the init.d script and must not be added to DAEMON_OPTS.
#
#DAEMON_OPTS=""
```

Let us write the configuration file of the hostapd in the directory **/etc/hostapd**. First, we go to the directory by **cd /etc/hostapd**, then write this file by **sudo nano hostapd.conf**. The screenshot below is an example of this file. We can change the **ssid** by changing the name assigned to it. For example, **ssid=free**.

```
interface=wlan0
ssid=localApps
hw_mode=g
channel=6
beacon_int=100
auth_algs=3
wmm_enabled=1
```

Dnsmasq is a DNS server operated on Unix-Like systems for small networks, which can be downloaded freely from the Internet. Let us install the **dnsmasq** by the command **sudo aptget install dnsmasq**, then go to the directory **/etc** by the command **cd /etc** and edit the **dnsmasq.conf** file, which is the dnsmasq configuration file. We add the following lines as the screenshot shown below to the top of this file. We use the command **sudo nano dnsmasq.conf** to edit this file and press **Ctrl+x, Y, Enter** when we finish editing this file.

```
# Configuration file for dnsmasq.

interface=wlan0
dhcp-range=10.0.0.2,10.0.0.10,255.255.255.0,12h
dhcp-option=3,10.0.0.1
```

we could start the service of hostapd and dnsmasq by the following command as shown in the screenshot below.

```
pi@raspberrypi ~ $ sudo service hostapd start
[ ok ] Starting advanced IEEE 802.11 management: hostapd.
pi@raspberrypi ~ $ sudo service dnsmasq start
[ ok ] Starting DNS forwarder and DHCP server: dnsmasq[....] (already running).
```

We could connect to the raspberry pi within the wireless network coverage by connecting the **wireless access point labeled by the ssid we have configured before** with a computer, a smart phone and so on. Then we could go to the terminal using ssh to connect our raspberry pi by the following command **ssh pi@10.0.0.1**, so we don't have to use the ethernet cable now.

Now we could think about how to forward all of the wireless traffic to the captive portal. Firstly, we have to enable the packet forwarding by uncommenting following line as shown in the screen shot below in the file

/**etc/sysctl.conf**, which is around line 28 . We use the command **sudo nano -c /etc/sysctl.conf** to edit this file and press **Ctrl+x, Y, Enter** when we finish editing this file.(nano -c could see the number of the lines when we are editing this file.)

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

After modifying the file sysctl.conf to enable the packet forwarding, we have to run the command **sudo sysctl -p** to re-read this file.

Firstly, let us try to turn this raspberry pi into a router meaning that the raspberry pi forwards all of the traffic from its wireless adapter(wlan0) to the ethernet cable connected to the computer. In order to achieve this goal, we could create an **iptable script** and this script has to run at the startup to turn the raspberry pi into a router. So we have to download **iptables** first by the command **sudo aptget install iptables** if we don't have it on our rapsberry pi.

Therefore, we go to the directory **/etc/network/if-up.d** to create a file called **router**. In this way, this script will run when the raspberry pi is online. So we type **cd /etc/network/if-up.d** to go to that directory and **sudo nano router** to create and edit the file. The commands are shown in the screenshot below without the sign# at the beginning, but I suggest to type all of the lines in the file for future use.

```
#!/bin/sh

iptables -F
iptables -X
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A INPUT -i wlan0 -j ACCEPT
iptables -A OUTPUT -o wlan0 -j ACCEPT
iptables -A FORWARD -i wlan0 -j ACCEPT

#sudo iptables -A POSTROUTING -t nat -o wlan1 -j MASQUERADE
sudo iptables -A POSTROUTING -t nat -o eth0 -j MASQUERADE
```

When we finish editing this file, press **Ctrl+x, Y, Enter** to save it and type this command: **sudo chmod +x /etc/network/if-up.d/router** to make it executable.

Let us analyze the rule at the last line carefully.

```
sudo iptables -A POSTROUTING -t nat -o eth0 -j MASQUERADE
```

The rule adds the POSTROUTING chain rule for NAT and uses the NAT packet matching table on the firewall's external networking device (-o eth0), which means that when the packet departures the firewall, the IP address of the packet will be specified with the external IP address of the firewall. Therefore, we could change the external networking device to wlan1. By configuring another interface wlan1 and pluging in another wifi adapter, we can forward the packet from the raspberry pi to wlan1. The screenshot below describes how to do it.

```
sudo iptables -A POSTROUTING -t nat -o wlan1 -j MASQUERADE
```

The last step is to consider how to forward the wireless traffic to the captive portal. By adding these rules (the ones without the # before) on the screenshot below to the router script file **/etc/network/if-up.d/router**, we can make it. Type **sudo nano router** to edit this file and press **Ctrl+x, Y, Enter** to save it when we finish.

```
#!/bin/sh

#iptables -F
#iptables -X
#iptables -A INPUT -i lo -j ACCEPT
#iptables -A OUTPUT -o lo -j ACCEPT
#iptables -A INPUT -i wlan0 -j ACCEPT
#iptables -A OUTPUT -o wlan0 -j ACCEPT
#iptables -A FORWARD -i wlan0 -j ACCEPT

sudo iptables -A POSTROUTING -t nat -o wlan1 -j MASQUERADE
#sudo iptables -A POSTROUTING -t nat -o eth0 -j MASQUERADE

sudo iptables -t mangle -N internet
sudo iptables -t mangle -A PREROUTING -i wlan0 -p tcp -m tcp --dport 80 -j internet
sudo iptables -t mangle -A internet -j MARK --set-mark 99
sudo iptables -t nat -A PREROUTING -i wlan0 -p tcp -m mark --mark 99 -m tcp --dport 80 -j DNAT --to-destination 10.0.0.1
```

We have analyzed the first rule and the rest means that all the HTTP requests will be forwarded to the captive portal, which is the local IP address of the apache server.

So we are going to configure another interface wlan1 for the raspberry pi. Therefore, we have to go back to the directory /**etc**/**network**, and type **sudo nano interfaces** to edit the **interfaces** file by adding the configuration of wlan1 at the end.

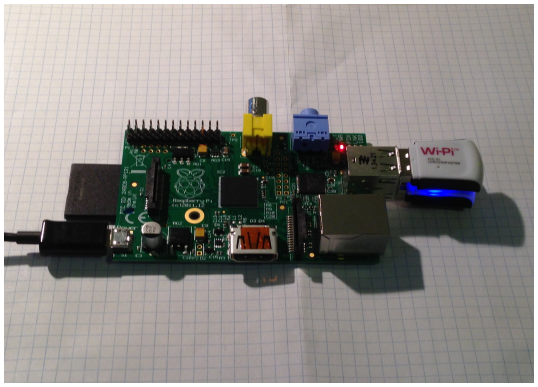
If we have an open wireless network, the screenshot below shows what we need to add. Make sure that the ssid name is added inside the quotes.

```
#auto lo
iface lo inet loopback
iface eth0 inet dhcp

allow-hotplug wlan0

#iface wlan0 inet manual
#wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf

iface default inet dhcp

auto wlan0
iface wlan0 inet static
address 10.0.0.1
netmask 255.255.255.0

#open network
auto wlan1
allow-hotplug wlan1
iface wlan1 inet dhcp
wpa-ssid "SSID_NAME"
wpa-key-mgmt NONE
wpa-scan-ssid 1
```

If we have a wireless network with a preshared key, the screenshot below shows what we need to add. Make sure that the ssid name is added inside the quotes.

```
#wireless network with a pre-shared key PSK
auto wlan1
allow-hotplug wlan1
iface wlan1 inet dhcp
wpa-ssid "SSID_NAME"
wpa-key-mgmt WPA-PSK
wpa-scan-ssid 1
wpa-psk network_key

#For example, network_key could be 12345678
```

Now, we can plug in another wifi adapater to the available USB port on the raspberry pi and then type **sudo reboot** to restart our raspberry pi. If everything goes well, we could see the blue light flickering like the screenshot below, which means that the interface wlan1 is connected to the internet.



If you would like to close the raspberry pi, you could type **sudo halt -h** at the terminal.

```
pi@raspberrypi ~ $ sudo halt -h

Broadcast message from root@raspberrypi (pts/0) (Mon Feb  2 21:51:24 2015):
The system is going down for system halt NOW!
pi@raspberrypi ~ $ Connection to 192.168.2.14 closed by remote host.
Connection to 192.168.2.14 closed.
wlan-docking-hg-1-0293:~ luyuanzeng$
```

Now we finish the basic setting up of our raspberry pi, then we start installing our applications.

The folder **photo_upload** contains the application for the users to put comments on the whiteboard and upload photos with extension jpg, jpeg, png and gif. The folder **questionnaire** contains the application for the users to vote the answers for the questions the administrator set and then send the

data by FTP to the FTP server. In addition, the administrator can also customize the questionnarie. The folder **server** contains the application for adding and configuring the raspberry pi on the map where the administrator needs to collect the data , so that the administrator can use the map to visualize the data collected where there is internet access.

Please download the applications above in the GitHub Repository
`https://github.com/nethood/localapps`

Please go to the folder of the applications on your terminal and copy the php file **index.php** , the photo **background.jpg** , the folder **questionnaire** and the folder **photo_upload** to the **/var/www** directory of the raspberry pi by the commands below:
scp index.php pi@10.0.0.1:/var/www
scp background.jpg pi@10.0.0.1:/var/www
scp -r photo_upload pi@10.0.0.1:/var/www
scp -r questionnaire pi@10.0.0.1:/var/www

In order to see the uploaded pictures when the user connected to this interface, we should install PHP GD image processing library for our raspberry pi by the following command **sudo apt-get update sudo apt-get -y install php5-gd**

```
pi@raspberrypi ~ $ sudo apt-get -y install php5-gd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libapache2-mod-php5 php5-cgi php5-cli php5-common
```

First, we go to the directory **/var/www** and change the permissions of the **background.jpg**
cd /var/www
sudo chmod 777 background.jpg
Then we go to the **photo_upload** folder.
cd /var/www/photo_upload

Then we change the permissions of some folders and files:
sudo chmod 777 forum
sudo chmod 777 forum/*
sudo chmod 777 album

We go back to the directory /**var**/**www** by the command **cd ..**
We change the permissions of the folder questionnaire:
sudo chmod 777 quesionnaire
sudo chmod 777 questionnaire/*
Then we go to the **questionnaire** folder.
cd questionnaire

We have to add three txt files in the folder questionnaire by the commands below:
touch id.txt
touch title.txt
touch questions.txt

In order to send the information of this raspberry pi to the FTP server by FTP, we should add the FTP parameters to the php file **admin.php** in the position shown by the screenshot below.

```
//Here is the parameters of the ftp server
//Please put your ftp_server, user_name and password in the " " below.
$ftp_server = "";
$ftp_user_name="";
$ftp_user_pass="";
```

Pay attention to the path on the FTP server that this data file is uploaded to. Please have a look at the screenshot below because you may need to adjust it according to your situation.

```
//Change to the correct directory.
//Please customize this path.
//Here is only an example.
ftp_chdir($conn_id, 'www/server/votes/');
```

Therefore, we should edit this file by the command **sudo nano admin.php** and press **Ctrl+x, y, Enter** to save the file when we finish editing it.

Then we change the permissions of some folders and files:
sudo chmod 777 id.txt
sudo chmod 777 title.txt
sudo chmod 777 questions.txt
sudo chmod 777 app.data
sudo chmod 777 app.data/*

The app.data contains the data file of this application. When we finish configuring the raspberry pi, we have to give the write permissions to the folders and files.
cd app.data
sudo chmod 777 poll-multi-choice.def
sudo chmod 777 poll-multi-choice.def/*

Then we go back to the **questionnaire** folder
cd ..

For uploading the data file to the FTP server by FTP, we should add the parameters of our own FTP server in the **upload_datafile.php** at the position indicated by the screenshot below. So we have to go to the **poll-multi-choice** directory.
cd poll-multi-choice
sudo nano upload_datafile.php
press **Ctrl+x, y, Enter** to save the file when we finish editing it.

```
//Here is the parameters of the ftp server
//Please put your ftp_server, user_name and password in the " " below.
$ftp_server = "";
$ftp_user_name="";
$ftp_user_pass="";
```

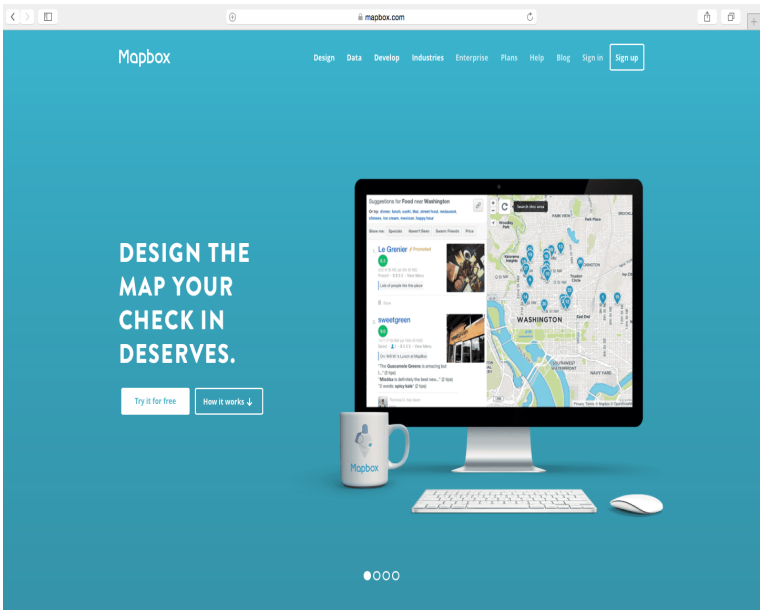
Pay attention to the path on the FTP server that this data file is uploaded to. Please have a look at the screenshot below because you may need to adjust it according to your situation.

```
//Change to the correct directory.
//Please customize this path.
//Here is only an example.
ftp_chdir($conn_id, 'www/server/votes/');
```

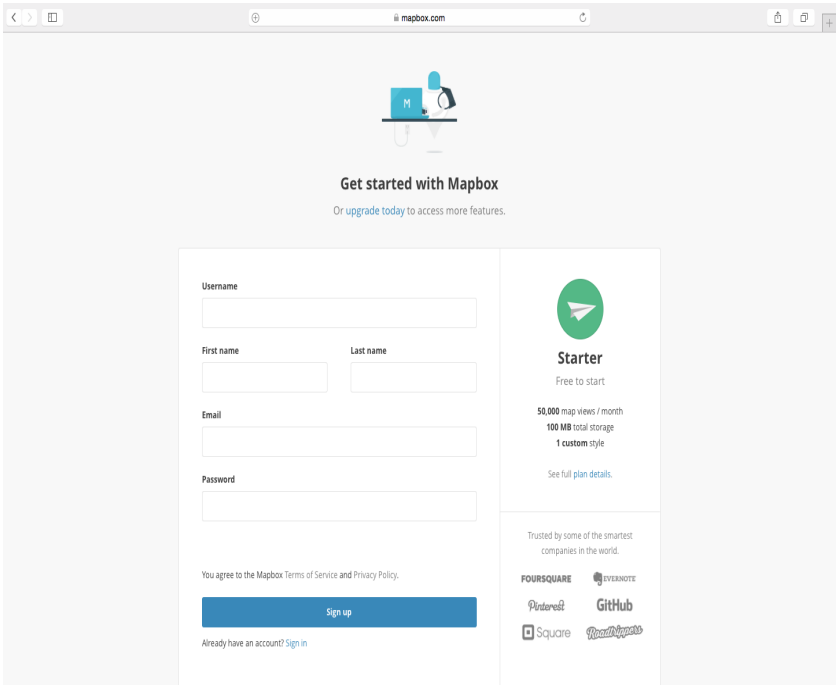
Please put the folder **server** at a standard shared web hosting space. If we have a raspberry pi as an administrator, we can add the rapsberry pi on the map and configure it then put it at the same place on the map. For example, the city center. The applications contained in the **server** folder enables the people who can access this server to visualize the data at the place where we put the raspberry pi.

The most important code is contained in **map_server_gjson.php**. **Please make sure that this php file can read,write and execute the contents**

in the folder votes and votes_new. In order to use this php file, we need to creat an account at **Mapbox** `https://www.mapbox.com`



Then we **click sign up** to create an account. The screenshot below is the page for signing up on Mapbox.



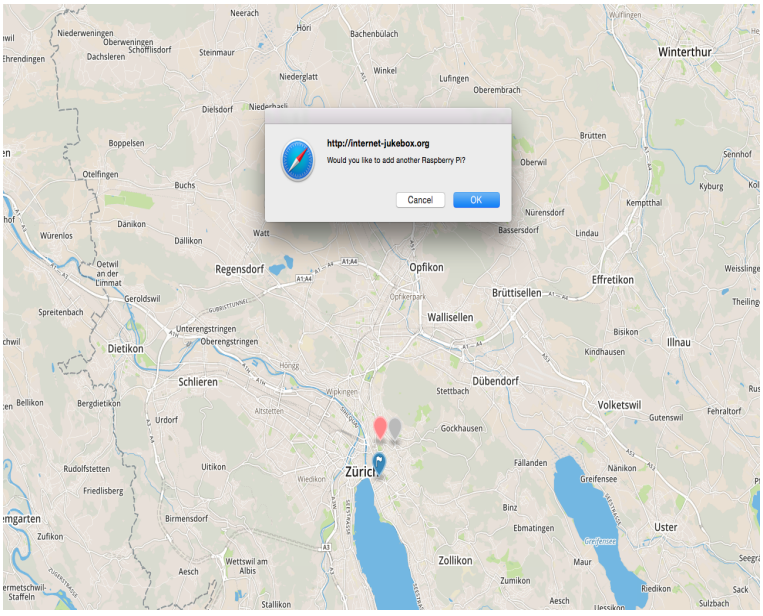
After we finished creating an account, we sign in with our username and

password. Then we click **Project** to get the parameters we need for using this application.

We could find our **accessToken** and **Map ID** in this page. Then we copy and paste them into **map_server_gjson.php** at the position indicated by the screenshot below.

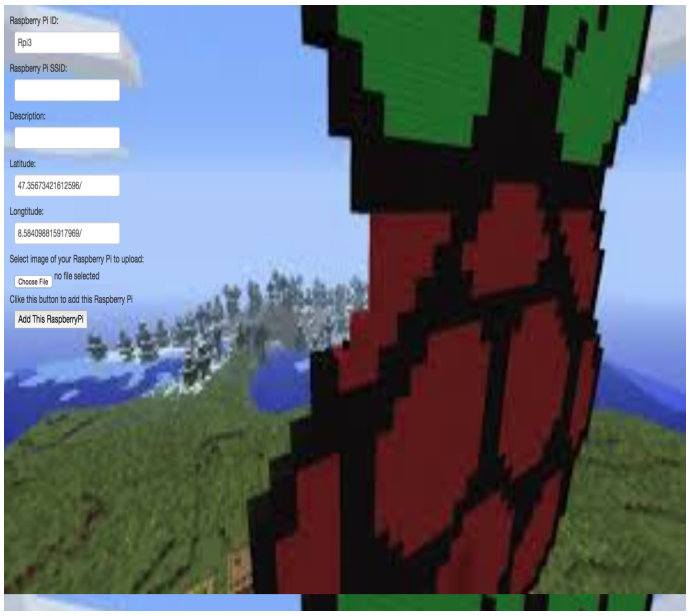
```
//Firstly, go to your MapBox Page
//Please put the your accessToken and your map ID of your account in the correct place


L.mapbox.accessToken = 'Your accessToken';
var map = L.mapbox.map('map', 'Your Map ID')
```

Let us start to assign a raspberry pi on the map by clicking the point that we want to add the raspberry pi on the map.

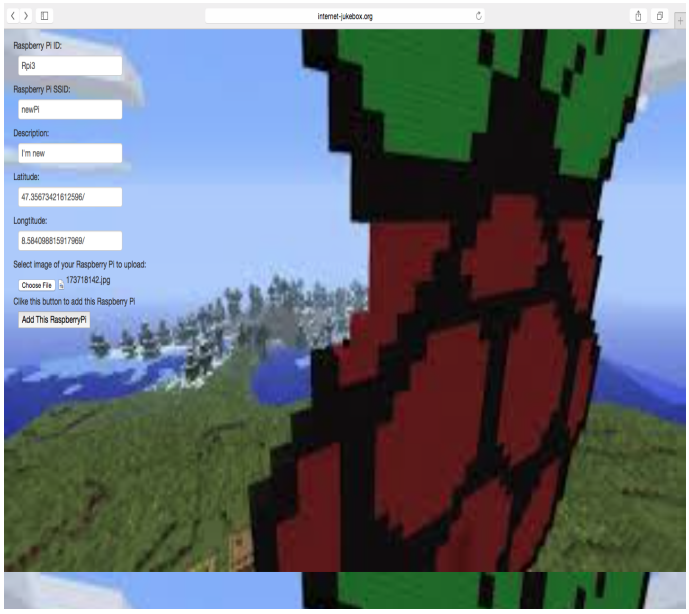


After clicking Ok, we will be redirected to a page to insert the parameters for the raspberry pi. If you can not see the background picture when you would like to insert the raspberry pi, please check the permissions of the picture **insert_rpi.jpg**. Pay attention that the id of the raspberry pi, the latitude and the longtitude are assigned automatically to the raspberry pi, we only have to insert the SSID, description and put a photo for this raspberry pi.

We should make sure that the php file **insert _rpi.php** has the permission to write into the file **raspberry_pis.txt**

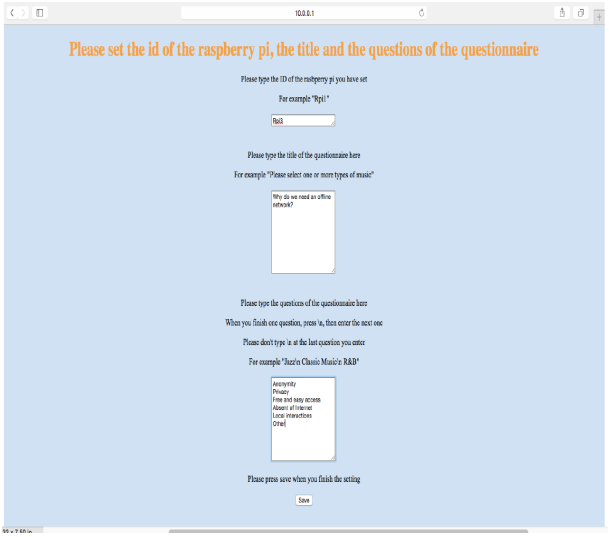


Let us insert the parameters. The example is provided by the screen shot below.



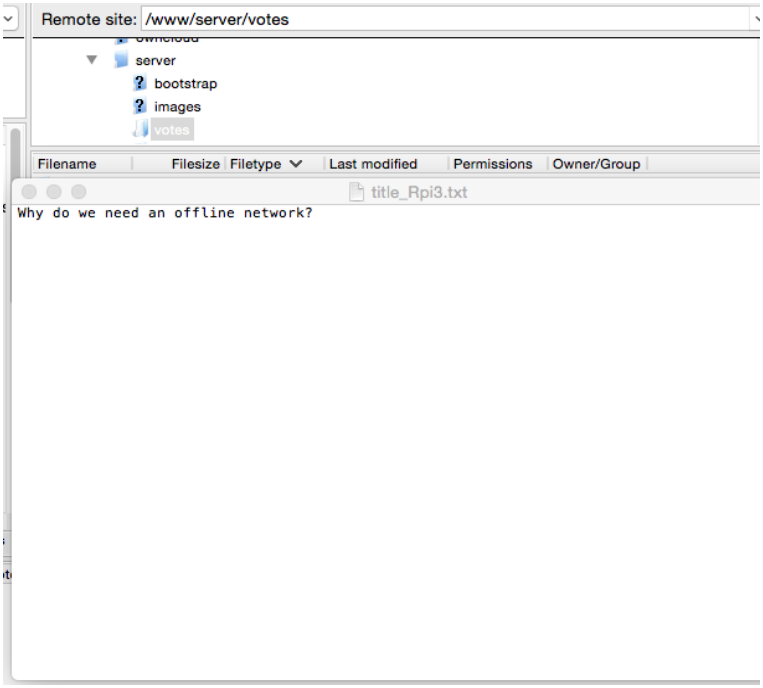
After clicking **Add this raspberry pi**, we will be redirected to another page that shows the parameters we have configured to this raspberry pi. We could see our configuration from the screenshot below.

Now let us take the raspberry pi. First we connect to the WIFI provided by the raspberry pi and then go to the terminal of the computer. Please use **ssh pi@10.0.0.1** to connect the raspberry through ssh. We could change the ssid of our raspberry pi according to the ssid we just configured. Another way is that we could assign the raspberry pi on the map at the very beginning and configure the raspberry pi with the ssid we just assigned on the computer. Now we start configuring the raspberry pi by typing **10.0.0.1/questionnaire/admin.php** on our browser, we could also type **www.google.com/questionnaire/admin.php**, but not some URL with **https** because the captive portal does not work with **htttps**. We will see a page for us to insert the id of the raspberry pi, the title of the questionnaire and the questions of the questionnaire. We have to fill it by following the instructions on it.

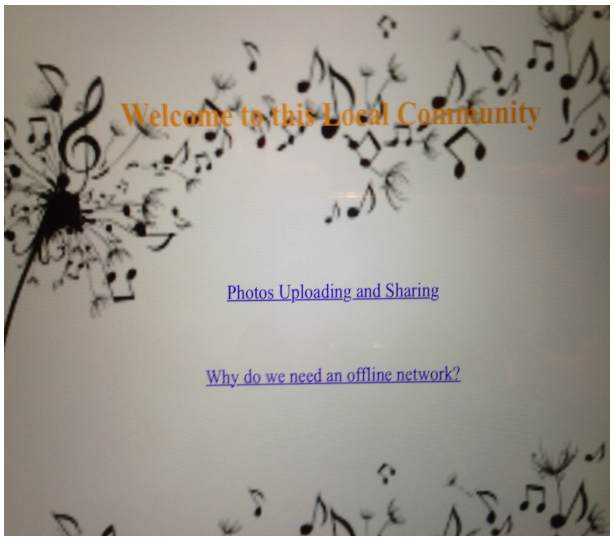


When we finish configuring, **click the Save button**. The configuration results will be saved and sent to a FTP server by FTP. If we would like to

check it, we have to connect to the internet now and then go to the FTP server. The screenshot below is an example using FileZila as an FTP server.

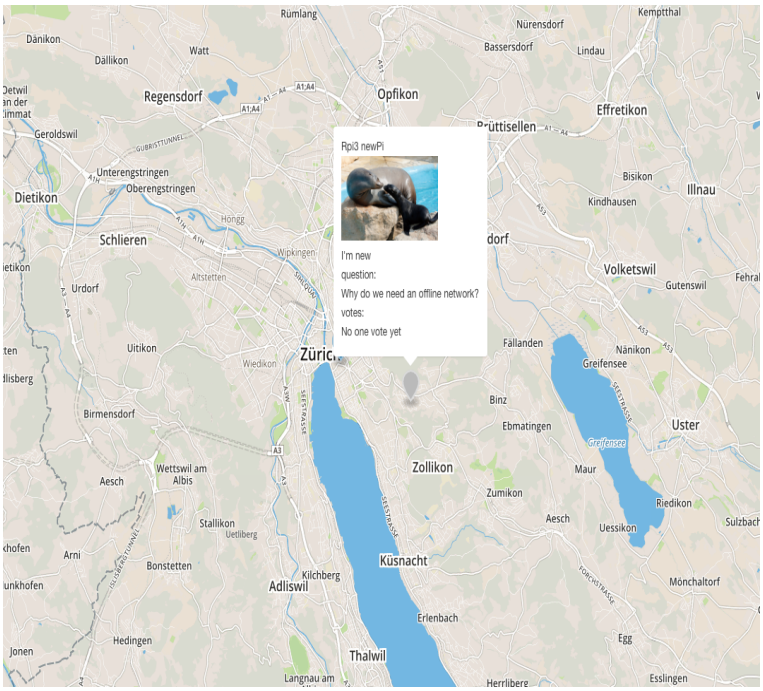


For example. if we connect the captive portal by typing **10.0.0.1** or **www.google.com** on the brower of our mobile phone, we could see the interface like the screenshot below, the last row is exactly what the administrator entered as the title of the questionnaire.



We could connect our computer with the internet and then press **go back**

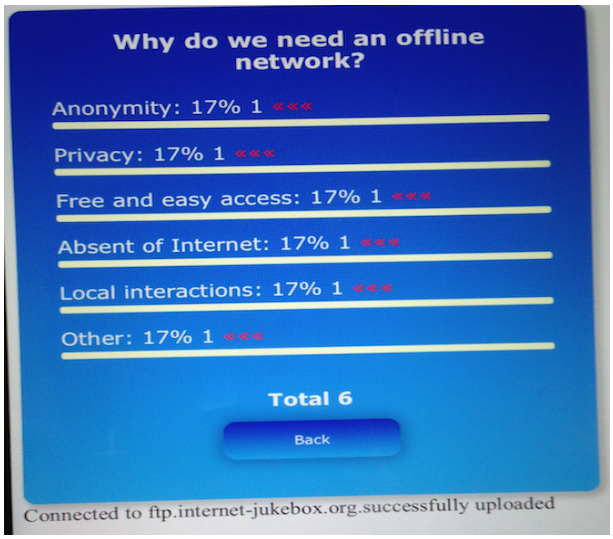
to the map on the page that shows the configuration of the raspberry pi. Then we could see our new raspberry pi on the map.



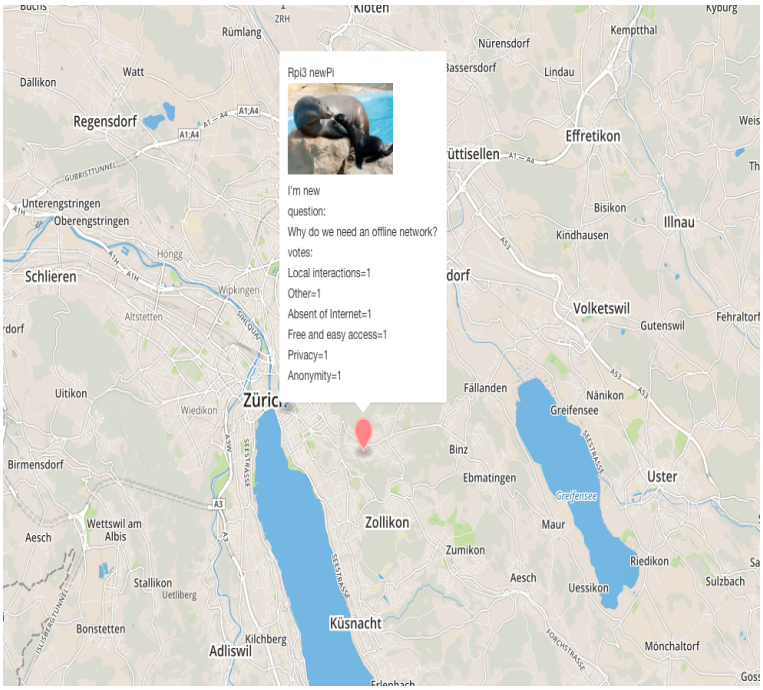
We could have a look at this question by clicking the question on our mobile phone if the mobile phone is connected to the captive portal of this raspberry pi.

Then we make the choice and click vote. Few seconds later, we can see the result. This time depends on the speed of the internet connection wlan1

because during this period, the data file is also uploaded to the FTP server.

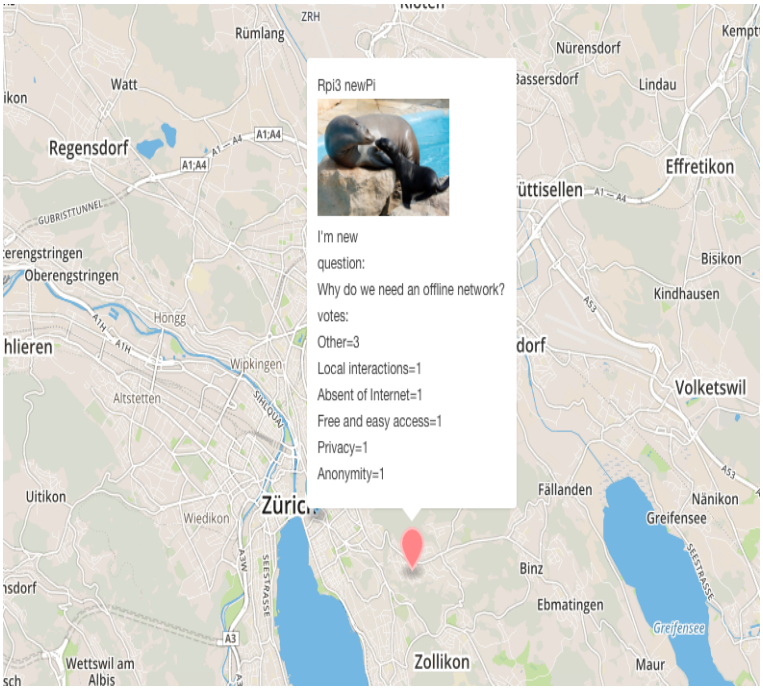


We could go back to our computer and check the results on the map by refreshing the webpage. And also, the point which represents the raspberry pi has become red because some one voted there and the data has been changed.

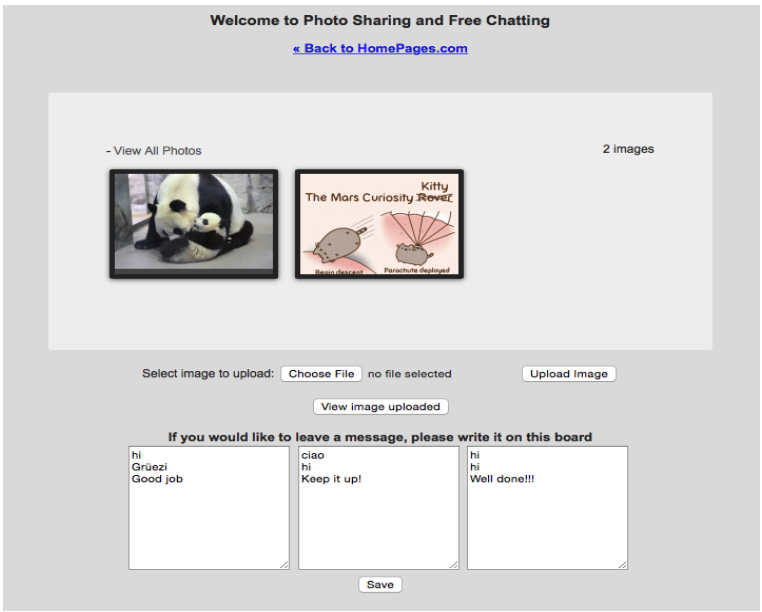


We could compare that the results on the map are the same as the results on the mobile phone. **In addition, this application also sorts the results**

according to decreasing order. We could vote for the other item other twice more to see the results so there are three votes for the other item. I made this test of my applications using the open wireless network at my room.

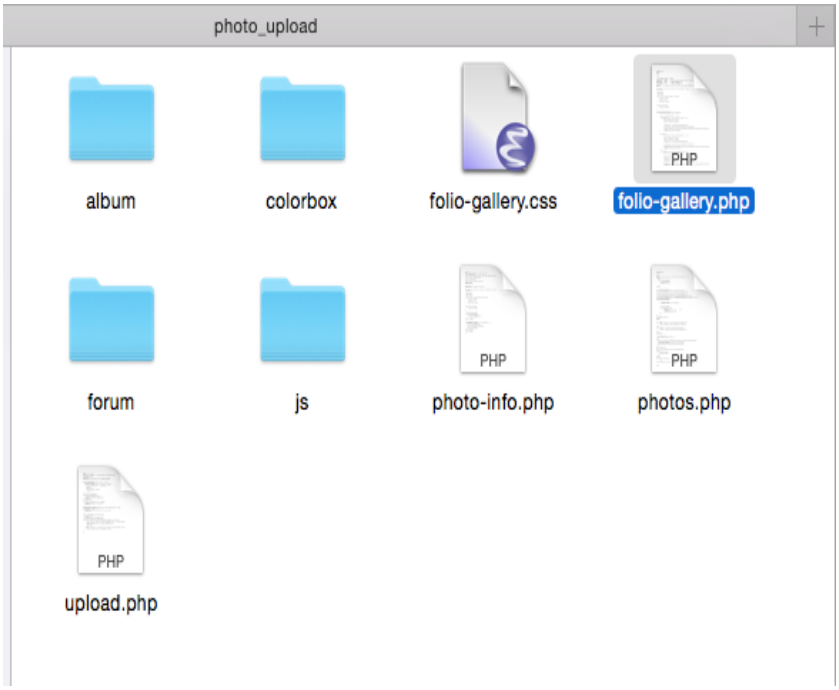


The **photo_upload** application also works well.



Additional Notes about the applications

Let us have a look at the folders and files contained in the folder **photo_upload** from the screenshot below.

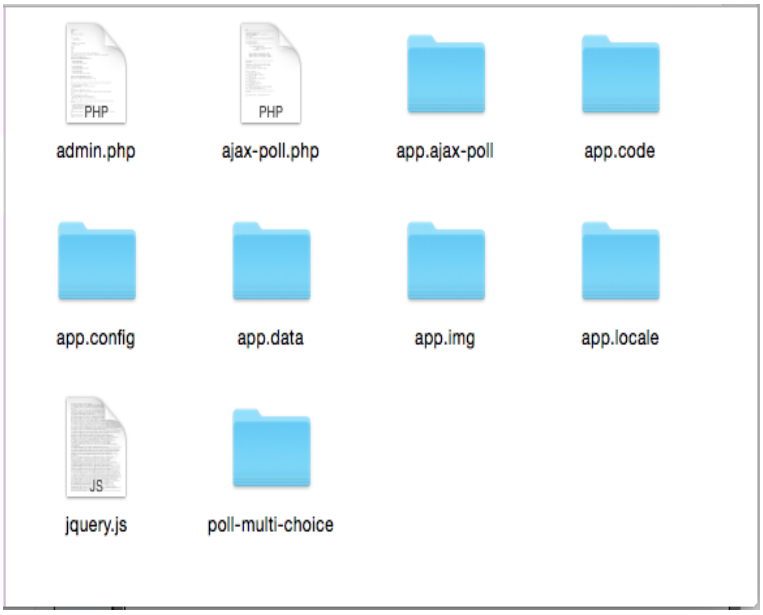


We could also adjust the maximum number of the images and the maximum size of each photo that can be uploaded by the users connected to the captive portal. From the screenshots below, we know that the maximum number is set to 30 and the maximum size of the photo can be uploaded is 2MB. To change the maximum number of the photos to be uploaded, we should edit the file **folio-gellary.php**. To change the maximum size of the photo to be uploaded, we should edit the file **upload.php**. For editing the file, we could use nano text editor.

```
// photo gallery settings
$mainFolder    = 'album';   // folder where your albums are located - relative to root
$itemsPerPage  = '30';      // number of images per page
```

```
// Check file size
if ($_FILES["fileToUpload"]["size"] > 2000000) {
    echo "Sorry, your file is too large.";
    $uploadOk = 0;
}
```

Let us have a look at the folders and files contained in the folder **questionnaire**.



There are four absolute paths in the questionnaire application. If you don't put the **questionnaire** folder in the directory /**var**/**www**, you have to adjust them accrording to your situation. More information about paths in terms of computing can be find on this website: `http://en.wikipedia.org/wiki/Path_%28computing%29`

One is located in the file **CPoll.inc.php**, which can be find in the directory /**var**/**www**/**questionnaire**/**app.ajax-poll**/**include**. The absolute path contained in the code can be seen from the screen shot below.

```
function getDataFilePath() {
        $id=file_get_contents('/var/www/questionnaire/id.txt');
        $data_file="votes_".$id.".txt";
        chmod ($data_file,0777);
        return $this->prt->getDataFolderPath() . "$data_file";
}
```

Another two are located in the file **class.inc.php**, which can be found in the directory /**var**/**www**/**questionnaire**/**poll-multi-choice**. The absolute paths contained in the code can be seen from the screen shot below.

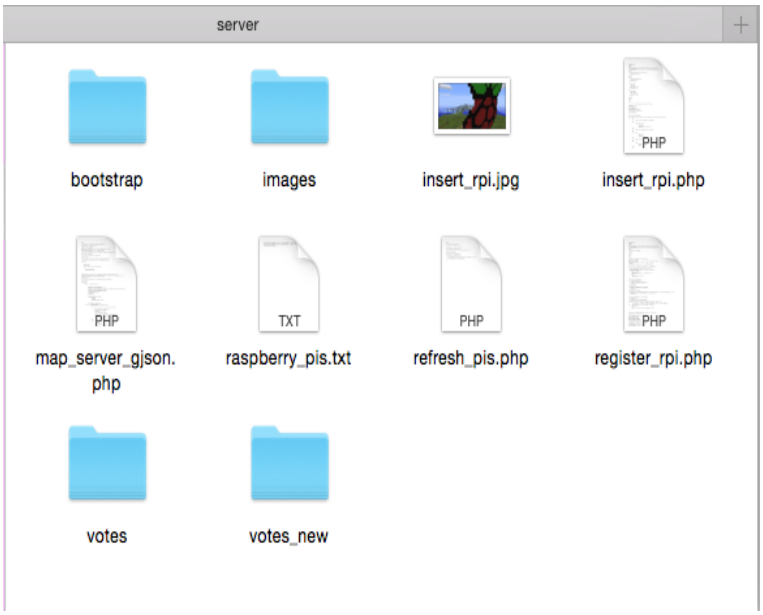
```
$title=file_get_contents('/var/www/questionnaire/title.txt',true);
$poll->attr( "title", "$title" );
//-- Poll Title
//$poll->attr( "title", "Please make one or more choices " );
```

```
$file = fopen("/var/www/questionnaire/questions.txt", "r");
while(!feof($file)){
        $line = fgets($file);
        if(strstr($line, "\n")) {
                $line=trim($line);
        }
        $poll->addItem($line);
}
fclose($file);
```

The last one is located in the file **upload_datefile.php**, which can be found in the directory **/var/www/questionnaire/poll-multi-choice**. The absolute path contained in the code can be seen from the screen shot below.

```
// A simple PHP/FTP upload to a remote site
$id=file_get_contents('/var/www/questionnaire/id.txt');
```

Let us have a look at the folders and files contained in the folder **server** from the screenshot below.



For the three applications above, the code contained in the file includes comments to describe the function of the corresponding code. When you read the code with the comments, you will understand the corresponding code. Please feel free to customize the applications to suit your needs.

Summary

This user guide demonstrates an example of setting up a local area network, deploying local applications for it and collecting data from local interactions. Ease of use and customization are the some of the most important characteristics of these applications. From the tools used (see the screenshoot below), we can conclude that the applications is not complicated.

Therefore, if you have basic knowledge of computers and the Internet, you can manage setting up a local area network and installing the applications by following this user guide.

Thanks for reading and feel free to customize these applications to suit your own local area network.

Luyuan Zeng
20 01 2015