"RFID-uino Logon"

A Real-time Arduino System

Allowing Users to Log into Computers

Using RFID Devices & Including

Automatic Email Notifications & Logging

Created By: Michael Huber

JWU Computer Science Major

Class of 2020

<u>Product Description:</u>

Arduinos are an amazing microcontroller solution for makers, tinkerers, and product creators of all kinds and skill levels. Using the libraries and tools provided by both the creators and communities of Arduino, there are thousands of projects already completed, as well as unlimited potential to complete brand new projects for embedded, realtime, and interactive systems of all kinds.

This project is no different in the respect that it is both relatively unique in its total execution but also that the work of the community was integral to the success of the total project.

My "RFID-uino Logon" project (not to be confused with a shield product of the same name) is based off of the MFRC522 RFID Transmitter/Receiver controller that allows RFID cards and FOBs to transmit data directly to a microcontroller, such as an Arduino, or microcomputer with GPIO I/O such as a Raspberry Pi. In the case of the Arduino, code must be uploaded to the microcontroller and then it is connected via USB to the target computer to act as the conduit between the RFID signal to digital data that Windows will then read as login information and log into the specified account with that given RFID card or FOB. The computer then send an email to a specified email address over encrypted SSL/SMTP which uses the email address automatic settings specified in the email account settings to categorize and log each login attempt. I used a Python program to send the email from my main gmail address already logged in to itself so that the settings are authenticated, and listed as "ADMIN", which allows for logging and categorization based on the controls already within the gmail settings.

Project Programs and Designs:

For the setup of the Arduino MFRC522 RFID, as evidenced by the code shown later on in this section, the RFID controller can be attached either through breadboard/circuit or directly to the Arduino in the following order:

RFID RC522	ARDUINO
Vcc	3.3V
GND	GND
RST	D9
MISO	D12
MOSI	D11
SCK	D13
SDA/NSS	D10
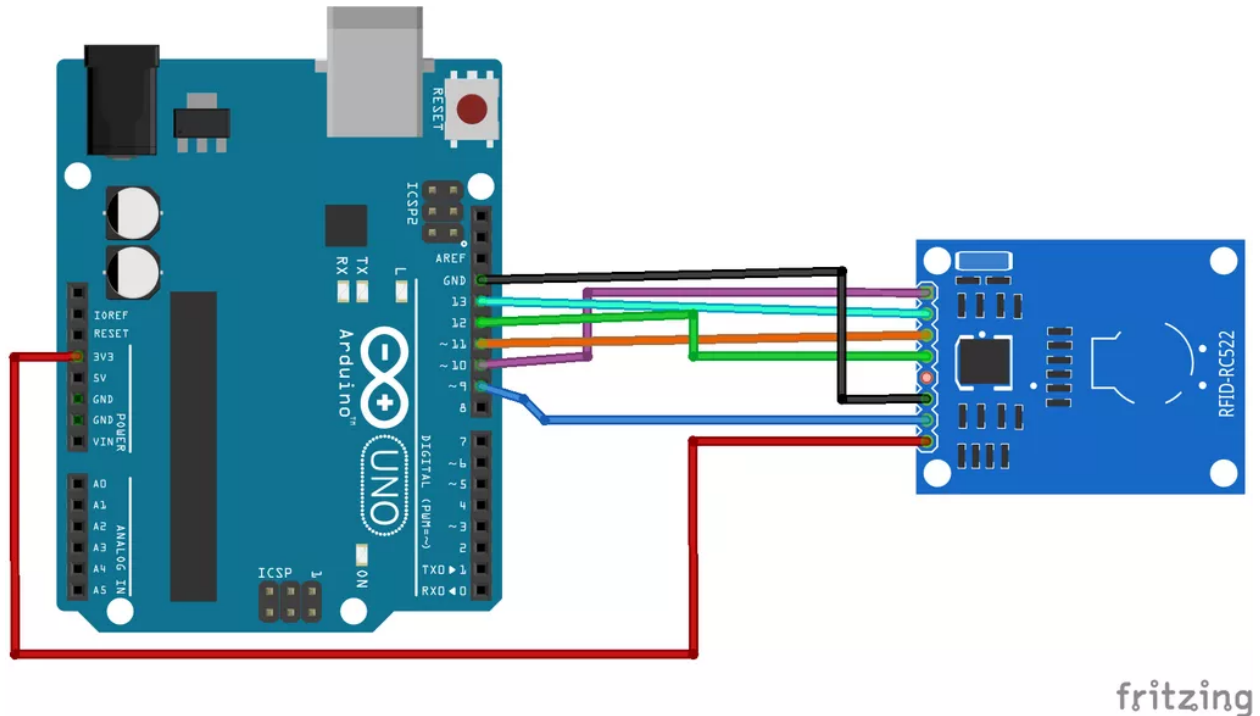
Before coding, the Arduino must be properly set up, and the MFRC522 Arduino library

must be downloaded and installed from https://github.com/miguelbalboa/rfid. The rest

of the programs and code can be found at the GitHub page for this project.

Below are the flowcharts and basic code (some excluded for security reasons).

Here is the flowchart and the code for the Arduino RFID program and the Windows

Registry:

```cpp
#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 10
#define RST_PIN 9
MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance.

void setup()
{
```
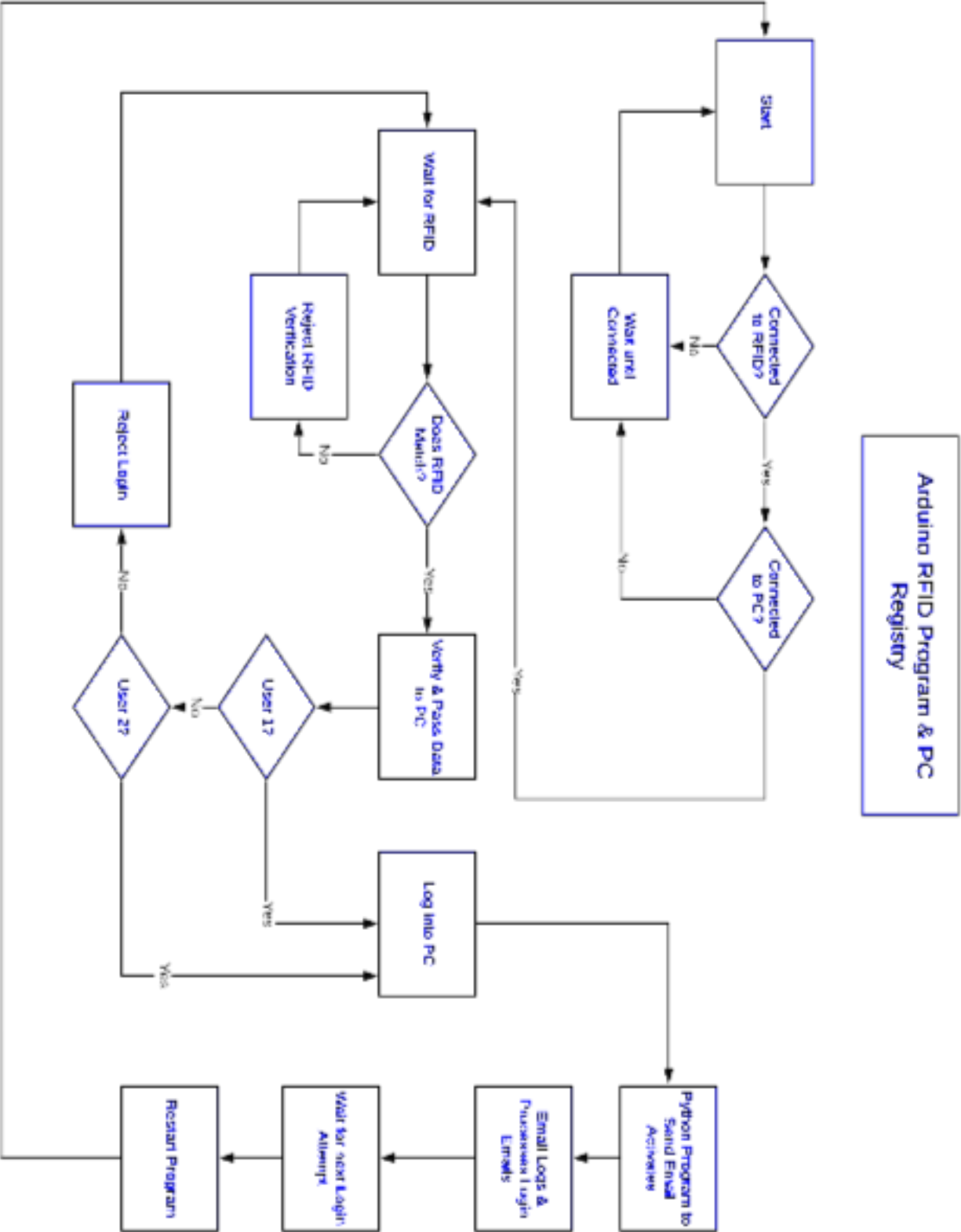
```cpp
  Serial.begin(9600);    // Initiate a serial communication
  SPI.begin();        // Initiate  SPI bus
  mfrc522.PCD_Init();    // Initiate MFRC522
  Serial.println("Approximate your card to the reader...");
  Serial.println();
}
void loop()
{
  // Look for new cards
  if ( ! mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }
  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  //Show UID on serial monitor
  Serial.print("UID tag :");
  String content= "";
  byte letter;
  for (byte i = 0; i < mfrc522.uid.size; i++)
  {
     Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
     Serial.print(mfrc522.uid.uidByte[i], HEX);
     content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
     content.concat(String(mfrc522.uid.uidByte[i], HEX));
  }

  Serial.println();
  Serial.print("Message : ");
  content.toUpperCase();
  if (content.substring(1) == "Replace with your card UID") //change
here the UID of the card/cards that you want to give access
  {
    Serial.println("Authorized access");
    Serial.println();
    delay(3000);
  }
 else   {
    Serial.println(" Access denied");
    delay(3000);
  }
}
```

Arduino RFID Program & PC Registry

Start

Connected to RFID? — No → Wait until Connected

Connected to PC? — No → Wait until Connected

Wait for RFID

Reject RFID Verification

Does RFID Match? — No → Reject RFID Verification

Verify & Pass Data to PC

User 1? — No → User 2? — No → Reject Login

Reject Login — No → Wait for RFID

User 1? — Yes → Log Into PC
User 2? — Yes → Log Into PC

Log Into PC

Python Program to Send Email Activated

Email Logs & successes / Login Limits

Wait for next Login Attempt

Restart Program

Below are the the basic code & flowchart for the Python Email application (though information and code has been left out to prevent security issues, such as passwords, addresses, and settings).

```python
import smtplib
from email.utils import formatdate
from email.mime.text import MIMEText

def send_mail(self):
    import smtplib
    from email.MIMEMultipart import MIMEMultipart
    from email.MIMEText import MIMEText

    gmailUser = 'myemail@gmail.com'
    gmailPassword = 'P@ssw0rd'
    recipient = 'sendto@gmail.com'
    message='your message here '

    msg = MIMEMultipart()
    msg['From'] = gmailUser
    msg['To'] = recipient
    msg['Subject'] = "Subject of the email"
    msg.attach(MIMEText(message))

    mailServer = smtplib.SMTP('smtp.gmail.com', 587)
    mailServer.ehlo()
    mailServer.starttls()
    mailServer.ehlo()
    mailServer.login(gmailUser, gmailPassword)
    mailServer.sendmail(gmailUser, recipient, msg.as_string())
    mailServer.close()

def main():
    send_mail()
```
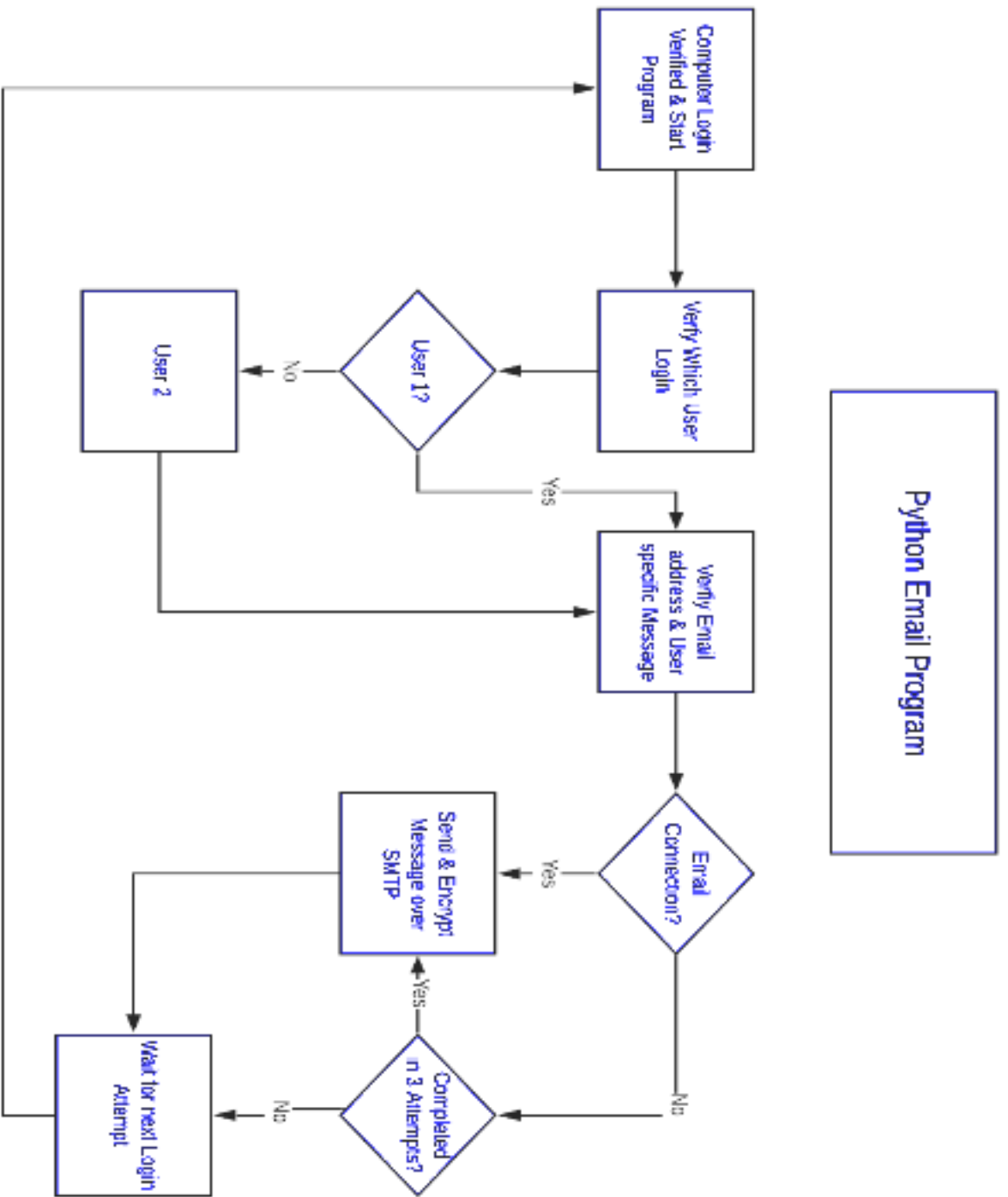
Python Email Program

Computer Login Verified & Start Program

Verify Which User Login

User 1?

No → User 2

Yes → Verify Email address & User specific Message

Email Connection?

Yes → Send & Encrypt Message over SMTP

No → Completed in 3 Attempts?

Yes

No → Wait for next Login Attempt

Setup & Operation:

Due to the experimental nature of this project, this does require the user to have some extra steps in order to get the project working. This requires some understanding of how to work with computers, the registry, and coding, though this is to be expected, especially due to the need for extended changes to prototypes, code, and registry entries.

1.) Set up your Arduino & RFID card reader:

    A. There is extensive documentation as to how to set up an Arduino. This particular project used an Arduino Uno, though it is essentially the same with all other versions of the Arduino microcontroller. You must download the Official Arduino IDE program for drivers and to upload the code from the files included in this project to your Arduino. Tutorials on how can do this can be found on the Arduino website, as well as included with the documentation included with your Arduino.

    B. Download and install the included RFID library to the Arduino Libraries included with the IDE, as shown on the GitHub page for the RFID library. Then Build the circuit as shown in previous diagram,

    C. After having the circuit ready, go to File > Examples > MFRC522 > DumpInfo and upload the code. This code will be available in your Arduino IDE (after installing the RFID library). Then, open the serial monitor. You should see something like the figure below:
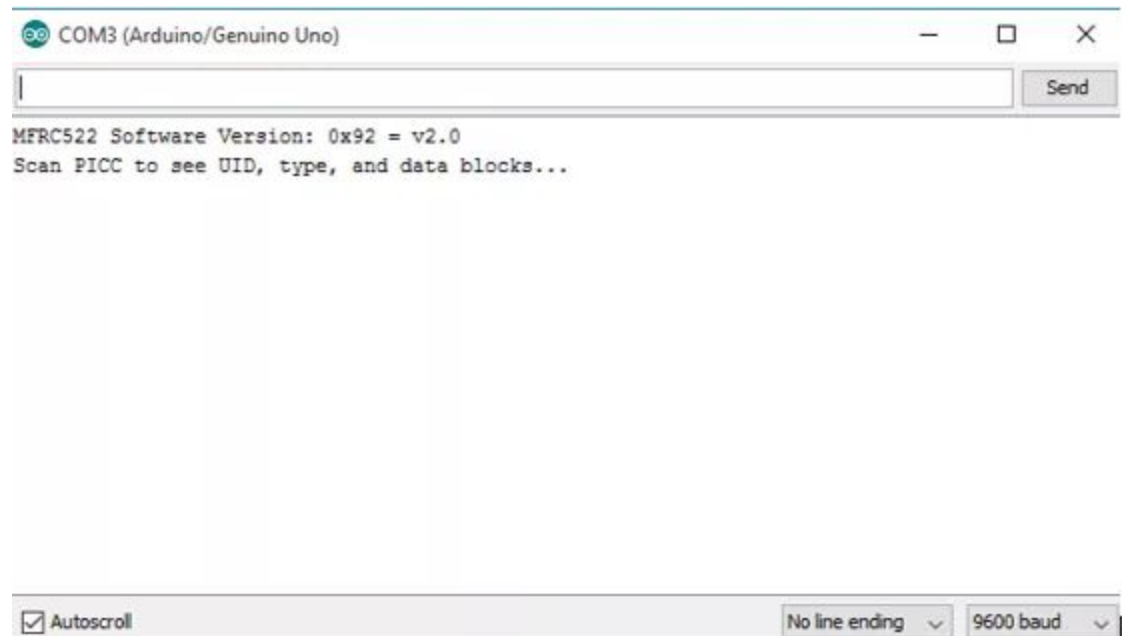
D. Approximate the RFID card or the keychain to the reader. Let the reader and the tag closer until all the information is displayed. Write down the card UID for later, as that is what you will insert into the Arduino code and also the Windows Registry Entry.

```
COM3 (Arduino/Genuino Uno)
|

MFRC522 Software Version: 0x92 = v2.0
Scan PICC to see UID, type, and data blocks...
Card UID: BD 31 15 2B
PICC type: MIFARE 1KB
Sector Block   0  1  2  3    4  5  6  7    8  9 10 11   12 13 14 15   AccessBits
  15     63   00 00 00 00   00 00 FF 07   80 69 FF FF   FF FF FF FF   [ 0 0 1 ]
         62   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
         61   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
         60   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
  14     59   00 00 00 00   00 00 FF 07   80 69 FF FF   FF FF FF FF   [ 0 0 1 ]
         58   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
         57   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
         56   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
  13     55   00 00 00 00   00 00 FF 07   80 69 FF FF   FF FF FF FF   [ 0 0 1 ]
         54   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
         53   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
         52   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
  12     51   00 00 00 00   00 00 FF 07   80 69 FF FF   FF FF FF FF   [ 0 0 1 ]
         50   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
         49   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
         48   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
  11     47   00 00 00 00   00 00 FF 07   80 69 FF FF   FF FF FF FF   [ 0 0 1 ]
         46   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
         45   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
         44   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
  10     43   00 00 00 00   00 00 FF 07   80 69 FF FF   FF FF FF FF   [ 0 0 1 ]
         42   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
         41   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
         40   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
   9     39   00 00 00 00   00 00 FF 07   80 69 FF FF   FF FF FF FF   [ 0 0 1 ]
         38   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00   [ 0 0 0 ]
```

E. Edit this Arduino code with your card UID:

```
if (content.substring(1) == "Replace with your card
UID") //change here the UID of the card/cards that you
want to give access

  {
```

2.) Setup of the registry entry for Windows 10:

    A. Download the Windows RFID Login zip file from the GitHub page, and unzip it to a folder such as Documents or Downloads, but not a system folder. It will look like this:


    Note: I would like to note that I have already changed the aspects of this registry entry so that it works properly with the most recent Windows 1903 update, as the update caused issues as the original code was originally made for Windows 7 and support has ended for that program You should not have to edit the registry entries "Register" or "Unregister" in any way.

    B. Open your Arduino IDE and look for the "Tools" menu. Select that menu and go down to the "Port" button. Note which COM Port it is using, as you will need this for the registry entry. It should look like this:
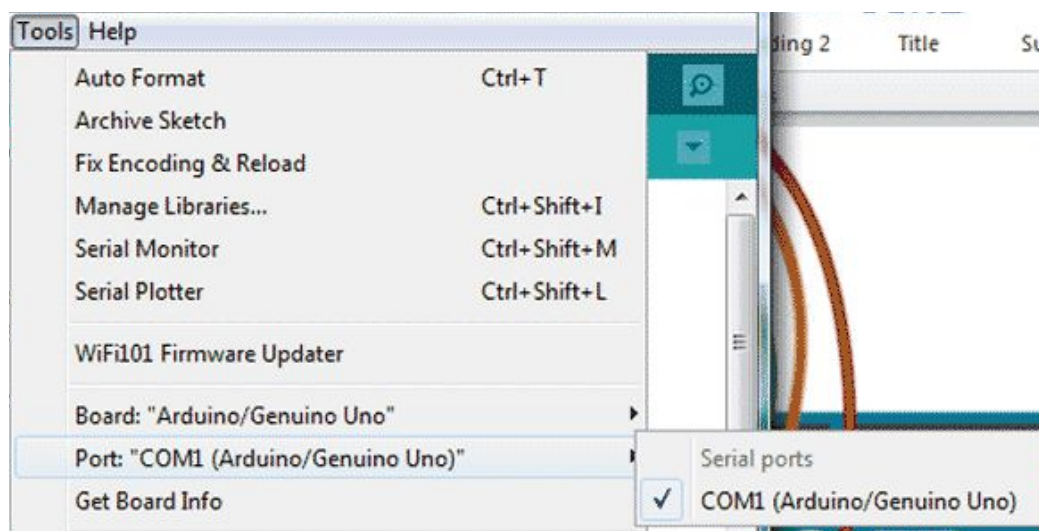
C. After extracting the ZIP file get into the folder named 32 bit or 64-bit folder (according to your operating system) and open the notepad named RFIDcredentials.txt. Paste the RFID values and update the system user name and password. If you want to add two cards then add the same credential in the second row as shown below. Where it says "user", you must enter the user account that RFID UID is being used to unlock. The "pass" is the password for that given account. You can use this to allow for multiple accounts to be logged in on a system with separate UIDs, or one account to be able to be logged in using multiple UIDs. However, the same UID cannot be used to log into two accounts at once.
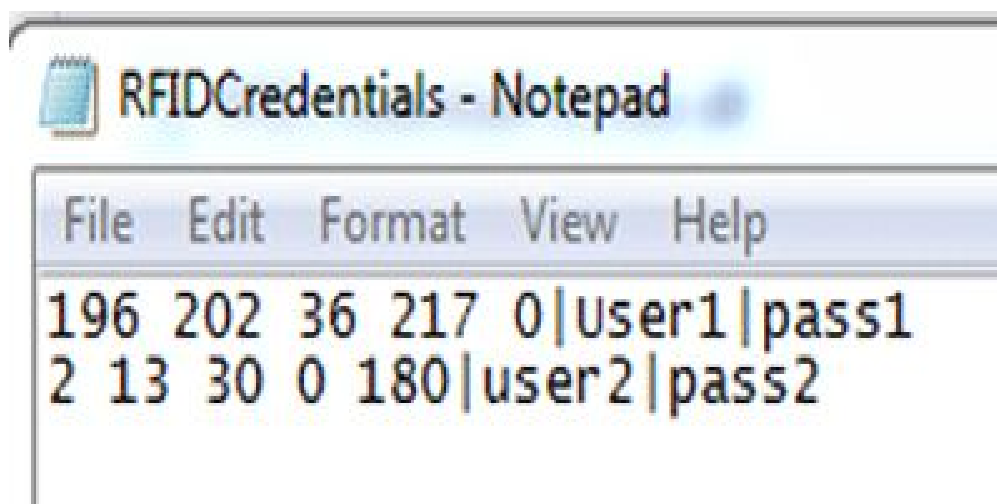

Then save and close this file. Now come back and open the RFIDCredSettings notepad and update the Arduino port in it then save and close. Again my COM port number is 1, update it with your COM port number. Leave the rest to default values as shown below.
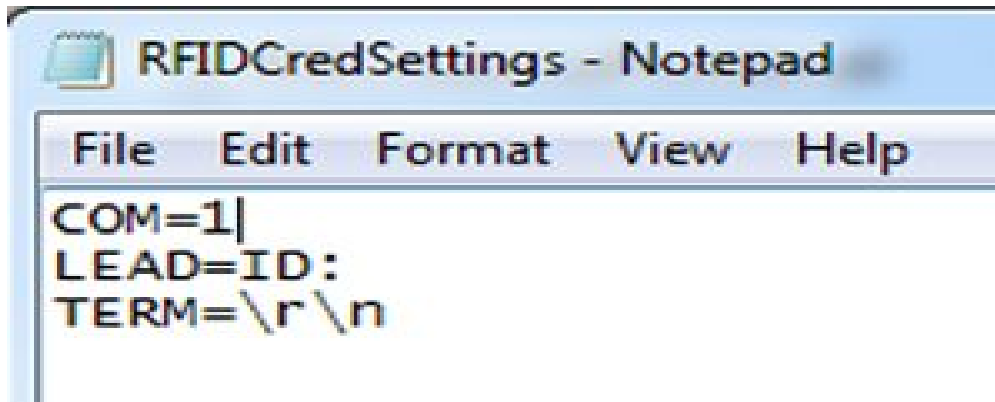
D. Now copy all four items and paste them in C:\Windows\System32. If it asks for any permission just give or click on yes. Now run the "Register" file to register the changes. Agree to all windows that come up asking if you wish to make this choice, and it should operate flawlessly.

E. If you ever wish to change this setting to no longer allow for RFID login, you can simply run the "Unregister" file as well.

3.) Creating and Implementing Email Notifications and Logging:

A. You must edit the Python (.py) file with your email addresses, passwords, and the content of the email message that you intend to send each time you log in via the RFID function of your computer.

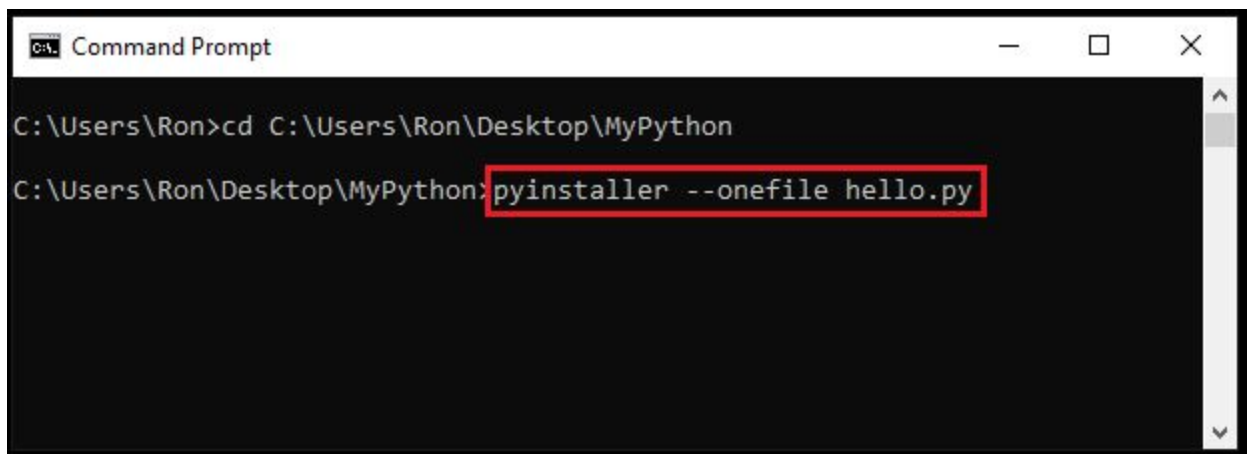B. Set your Python Distribution to your PATH variables and then install pyinstaller via the command prompt (CMD.exe)
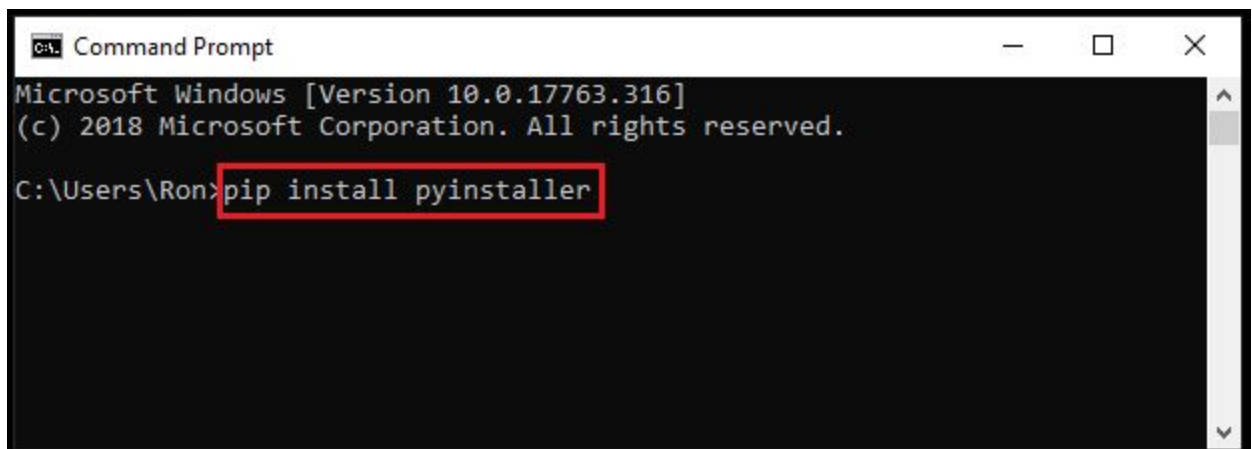
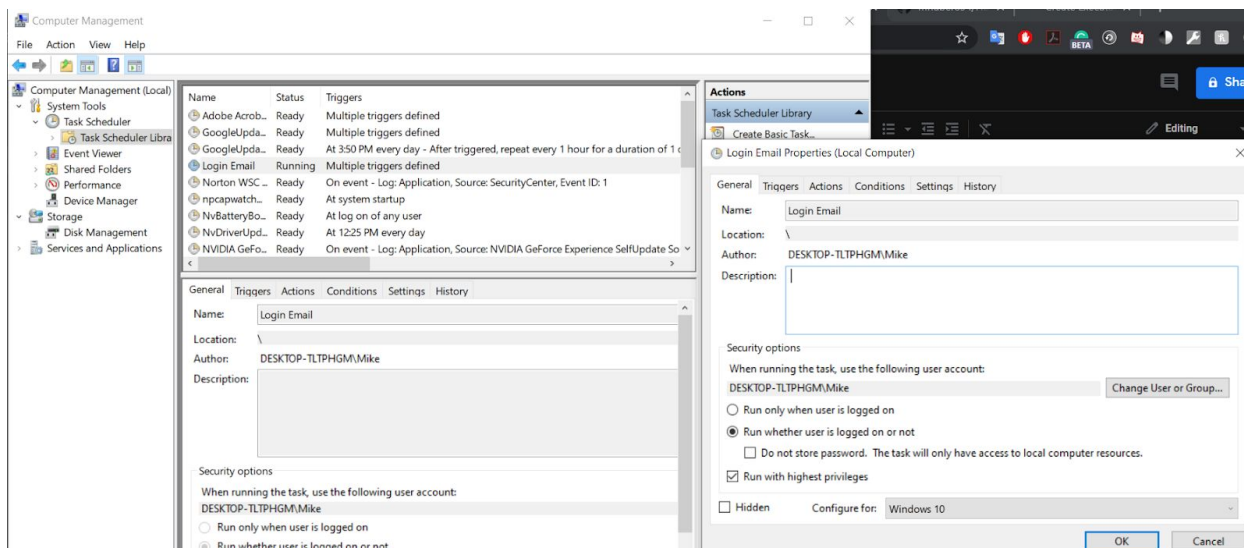C. Find the Python file that you edited and then go the that location through the `cd` command in your command prompt.

D. Then, enter the following command into the command prompt in order to transfer it into an executable .exe program

E. After that, go into your task scheduler and create a specialized login task based on the registry entry change you made earlier

F. Under the "Actions" tab, select the .exe program you created from the Python program you made before so that every time the computer is logged in through the RFID account option, and will send an email to the specified email address letting you know it logged into the computer.

Help & Support

If you have issues or concerns, please consult the various documentation about the specific tools and programs used in this project. If those documents do not address your issues, please reach out to me though the contact options on my GitHub account with images, specific error messages, and any pertinent information related to the errors with your programs and options. Please be as thorough.

Please reach out if you have any questions or concerns, please reach out to me without hesitation. All feedback is welcome and taken into consideration for future updates and changes to the project. Thank you for Choosing this project!