

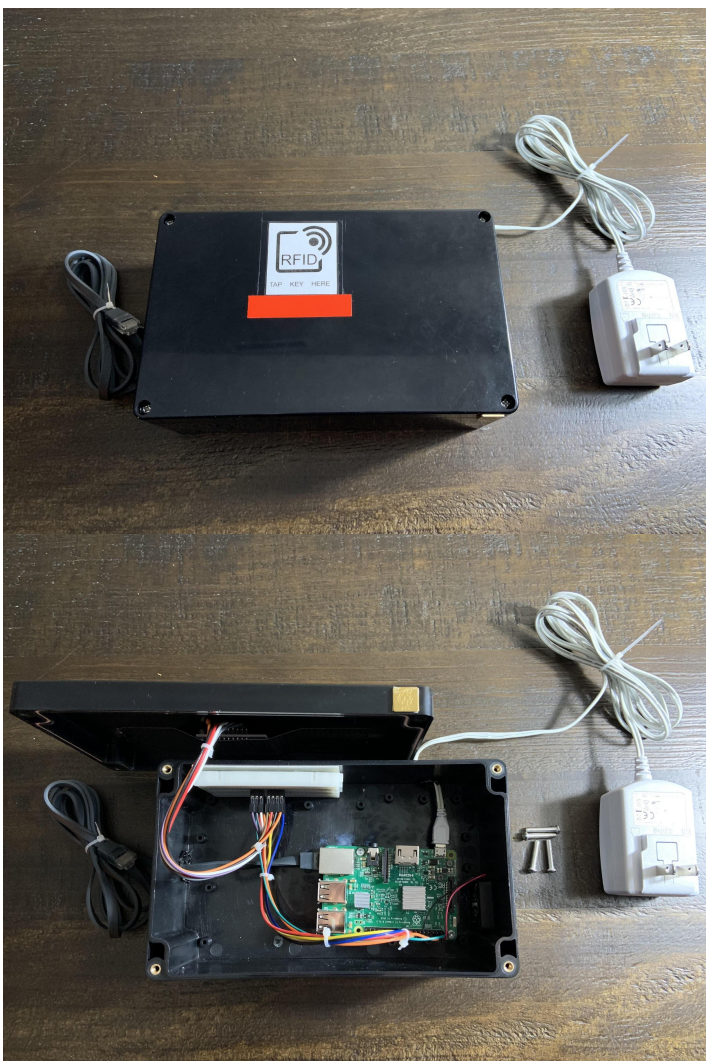
# Raspberry Pi Attendance System

Assembled and Designed By: John Hall  
Created By: Pi up my life

# Description of Product

This attendance system will log data that is tapped to the RFID sensor. Once data is created for a user on the Raspberry Pi, the user will tap his/her card to it upon entering the designated room. Data will consist of the date and time at which the student arrived or left, the name of the student with their Tag's unique ID number and any other relevant minimal data that may be associated with the user. This data is then sent to an SQL database where it is displayed on a simple web interface. This web interface will allow you to check the user's attendance and view when they tap their RFID card to the reader.

Task	Activity Description
Task 1	RFID Setup
Task 2	Prepare Raspbian
Task 3	Build RFID RC522 Reader Circuit and Enable SPI
Task 4	Enable the correct Kernel Module
Task 5	Create Python to Read/Write data to RFID Tags
Task 6	Test the RFID reader
Task 7	Database Creation
Task 8	Install MySQL
Task 9	Setup Secure Installation
Task 10	CREATE DATABASE attendance;
Task 11	CREATE USER
Task 12	Creating and Recording Multiple Users
Task 13	Create Python to write data to database
Task 14	Check Database
Task 15	Check Attendance
Task 16	Database Maintenance and Design
Task 17	NGINX and PHP for Web Interface
Task 18	GitHub Cloning
Task 19	URL testing
Task 20	Troubleshooting
Task 21	Advanced Troubleshooting
Task 22	Design Hardware Housing
Task 23	Order Hardware Housing
Task 24	Create Hardware Housing
Task 25	Design User Manual
Task 26	Create and Organize User Manual
Task 27	User Testing
Task 28	Implementation



< System Overview

Hardware Setup >

< Hardware Overview

RC522 Setup >



## check\_attendance.py

## Python Code (Need notes for each line.)

## save\_user.py

```

import time
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522
import mysql.connector

db = mysql.connector.connect(
    host="localhost",
    user="attendanceadmin",
    passwd="astate1",
    database="attendancesystem"
)

cursor = db.cursor()
reader = SimpleMFRC522()

try:
    while True:
        print('Place Card to\nrecord attendance')
        id, text = reader.read()

        cursor.execute("Select id, name FROM users WHERE rfid_uid="+str(id))
        result = cursor.fetchone()

        if cursor.rowcount >= 1:
            print(" " + result[1])
            cursor.execute("INSERT INTO attendance (user_id) VALUES (%s)", (result[0],))
            db.commit()
        else:
            print("User does not exist.")
            time.sleep(2)

finally:
    GPIO.cleanup()

```

```

import time
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522
import mysql.connector

db = mysql.connector.connect(
    host="localhost",
    user="attendanceadmin",
    passwd="astate1",
    database="attendancesystem"
)

cursor = db.cursor()
reader = SimpleMFRC522()

try:
    while True:
        print('Place Card to\nregister')
        id, text = reader.read()
        cursor.execute("SELECT id FROM users WHERE rfid_uid="+str(id))
        cursor.fetchone()

        if cursor.rowcount >= 1:
            print("Overwrite\nexisting user?")
            overwrite = input("Overwrite (Y/N)? ")
            if overwrite[0] == 'Y' or overwrite[0] == 'y':
                print("Overwriting user.")
                time.sleep(1)
                sql_insert = "UPDATE users SET name = %s WHERE rfid_uid=%s"
            else:
                continue;
            else:
                sql_insert = "INSERT INTO users (name, rfid_uid) VALUES (%s, %s)"
            print('Enter new name')
            new_name = input("Name: ")

            cursor.execute(sql_insert, (new_name, id))

            db.commit()

            print("User " + new_name + "\nSaved")
            time.sleep(2)

finally:
    GPIO.cleanup()

```

# SQL Commands

```
create table attendance(  
  id INT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE,  
  user_id INT UNSIGNED NOT NULL,  
  clock_in TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY ( id )  
);
```

```
create table users(  
  id INT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE,  
  rfid_uid VARCHAR(255) NOT NULL,  
  name VARCHAR(255) NOT NULL,  
  created TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY ( id )  
);
```