Centralisation of Entry Registration Process Using Real-Time Cloud Environment

Pulkit Jain: 19ucc038@lnmiit.ac.in Utkarsh Pancholi: 19ucc039@lnmiit.ac.in

Abstract

This paper aims to discuss implementing and designing an automatic entry registration system based on a Real-Time Cloud Environment. A prototype for managing and controlling student registration is developed using IDLE and the Firebase Real-Time Database. However, the GUI Application is designed to help administrators check their presence details, such as the In/Out Status and their arrival/departure time and date. Automating registering process using Bluetooth technology will reduce the requirement of physical logs, which are hard to maintain and store. Moreover, using the Firebase Real-Time Database Cloud will be costless because it imparts data centralisation and prevents the cost of setting up huge infrastructure performing a similar task.
Moreover, building and designing a desktop and Cross-Platform GUI Application to control and follow the entry registration process and log out the relevant records. However, the benefits of using a real-time database in education institutions will cost less than a traditional data centre. The proposed registration system takes less time to record student attendance with high accuracy. Apart from that, not just time, many exercises are done to preserve these physical records, which accounts for a lot of paper wastage. As per the World Wildlife Fund, The pulp and paper industry, which makes office and catalogue paper, glossy paper, tissue and paper-based packaging, uses between 33 - 40% of all industrial products wood traded globally.

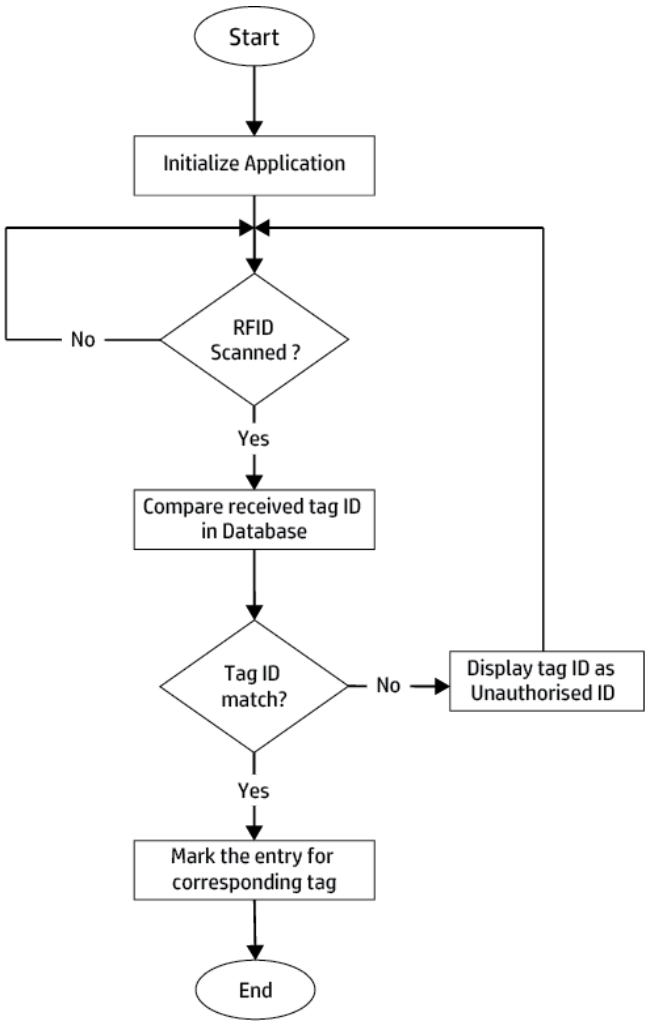
1.1 Introduction

Access control verifies a user's claimed identity and gives or denies access. The submitted project is a secure access control system to monitor students' entry through a checkpoint or a passage using barcode detection technology.
At the most superficial level, Barcode Detection technologies allow the transmission of a unique identification number(e.g. 19UCC000) provided by the institute using OpenCV with a Camera Sensor Device. The two critical parts of the system needed to do this are the RFID Card and the Camera Sensor. The main applications of Barcode Detection Technology include automated entry registration, attendance registration, monitoring systems etc. The cost criteria associated with implementing the RFID system are expected to be overcome soon. The flexibility of our offered access control system makes itself appropriate for various applications.

1.2 Methodology

Three Python Scripts from which the project is built. The first script is main.py which uses the OpenCV library to extract the data of the scanned QR code from the incoming frames by the Camera sensor. The next script is firebaseDataSearch.py which takes the extracted ID from main.py as input and searches for that ID in the firebase real-time database. If the ID matches, it returns a dictionary containing the student's record. If not, it returns -1. The third and final script is the push_to_csv.py which takes input as the output of firebaseDataSearch.py and logs the data in a CSV file named today.

The stand-alone application(main.exe) requires only the firebase authentication credentials to work. Similarly, it provides an easy interface to handle. The application window contains three buttons. First is the enter, then the exit and display log button. The enter button is used to scan the QR in the video frame. The relevant output after scanning the QR is displayed in the text box corresponding to the scanned ID button, and the exit button is used to exit the application. In the end, the display log button is used to display the contents of the CSV file in an interactive format. For security, two more features are implemented. The first is to ensure that the barcode is present in the frame, so if there is no barcode present in the frame and one tries to scan, the output will be reflected as “Improper Scan”. In the following case, if a barcode is present in the frame and contains the ID not registered in the database, the output is reflected as an “unauthorised ID”.



1.2.1 Components and Technologies Used

Camera Sensor: An camera or image sensor is a sensor that detects and conveys information used to make an image. It accomplishes so by converting the varying attenuation of light waves into signals, small bursts of current that give the information.

Google Firebase Real-Time Database: The Firebase Realtime Database is a cloud-hosted NoSQL database that allows you to store and sync data between users in real time. Realtime syncing makes it easy for your users to access their data from any device: web or mobile.

IDLE (Integrated Development Environment for Python): IDLE can be used to execute a single statement like Python Shell and create, modify, and execute Python scripts. IDLE provides a fully-featured text editor to create Python scripts. It also has a debugger with stepping and breakpoints features.

OpenCV (Open Source Computer Vision Library): OpenCV is a huge open-source library for computer vision, machine learning, and image processing. It plays a significant role in real-time operation, which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even the handwriting of a human.

PySimpleGUI: PySimpleGUI is a python library that wraps Tkinter, wxPython and Remi (for browser support), allowing swift and simple-to-learn GUI programming. PySimpleGUI defaults to using Tkinter, but the user can change to another supported GUI library by just changing one line.

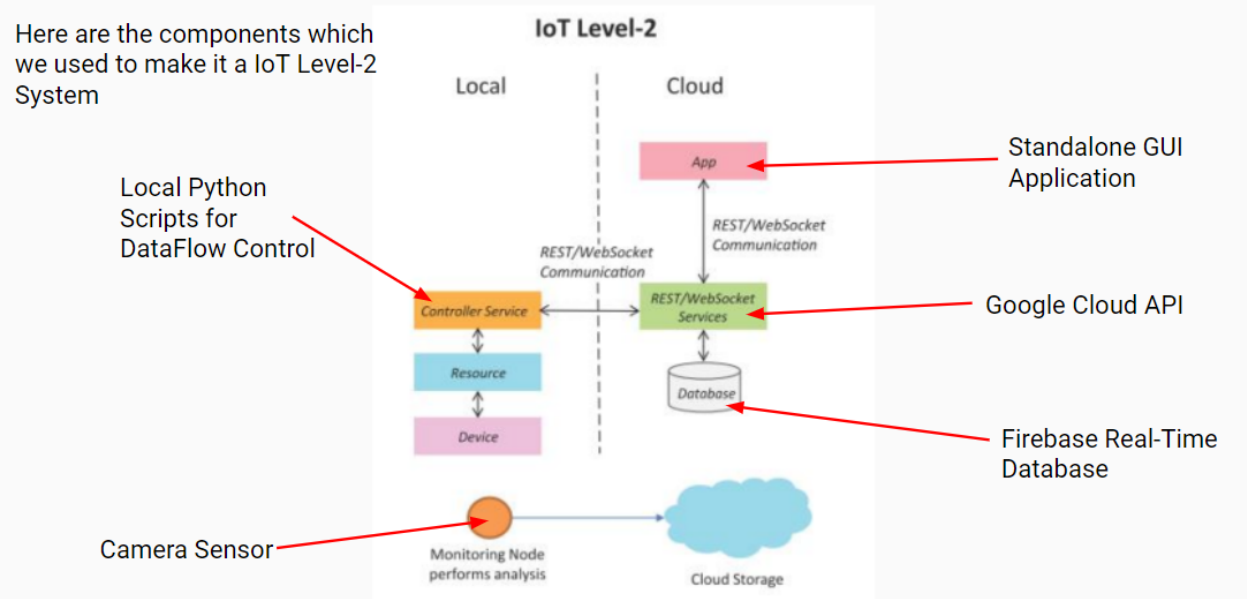
PyInstaller: PyInstaller works by reading your Python program, analysing all of the imports it makes, and bundling copies of those imports with your program. After analysing your code and discovering all of the libraries and modules it uses, PyInstaller generates a “spec file” that details how your Python app needs to be packed up.

Visual Studio Code: Visual Studio Code is a streamlined code editor supporting development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle.

Git/Github: Git is a distributed version control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. GitHub is a web-based Git repository hosting service that offers all of the distributed revision control and source code management (SCM) functionality of Git and adds its features.

1.2.2 Project Model (As an IoT Level-2 System)

The illustration below shows that the camera sensor is working as a monitoring node, and the local Python scripts work as controller services between the database and the node. The Firebase Realtime Database, connected to Google Cloud, serves as a data storage unit for reading, writing, and updating the data centrally. Google Cloud Firebase API is used to read and write data to the database in real-time. We used a GUI application to log the output.

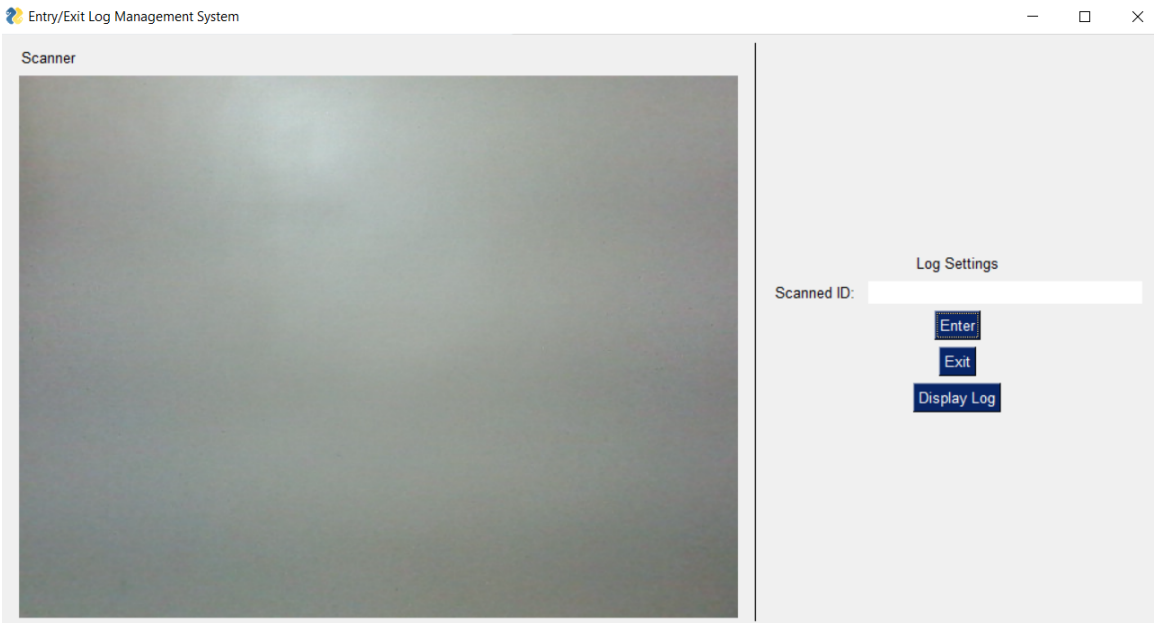


1.3 Application Working

- **Setting Up with the Application**: **(Consider a device with an inbuilt camera sensor)**
  - For Windows Users
    Download the main.exe file and the scanner-database-firebase-adminsdk-npg5h-956503dab3.json file from the Github repository. Put both the files at the same location and open the main.exe file. Wait for the application window to appear.

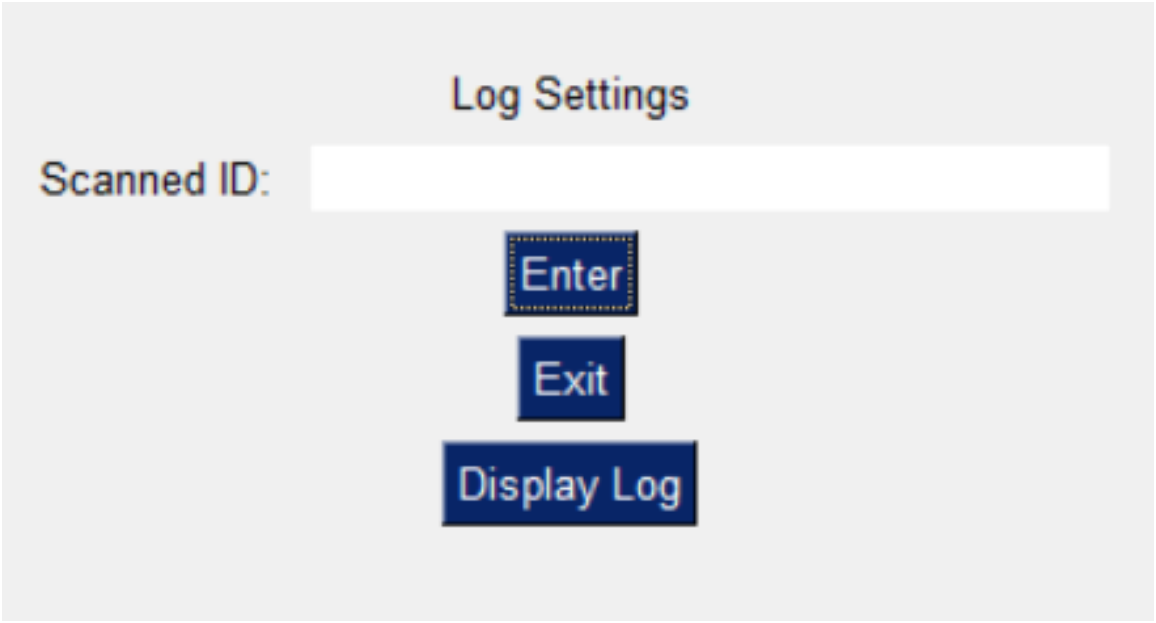
  - For Non-Windows Users
    Download the python scripts, namely:
    main.py, scanner.py, firebaseDataSearch.py, push_to_csv.py, table_example.py and scanner-database-firebase-adminsdk-npg5h-956503dab3.json from the Github repository. Put all the files at the same location and run/execute the main.py file. Wait for the application window to appear.

- **Application Window**: It is the home window of the standalone GUI application. It has five components distributed over two sections: scanner and log settings. The five components are 1 Display, 3 Button and 1 Label Component.

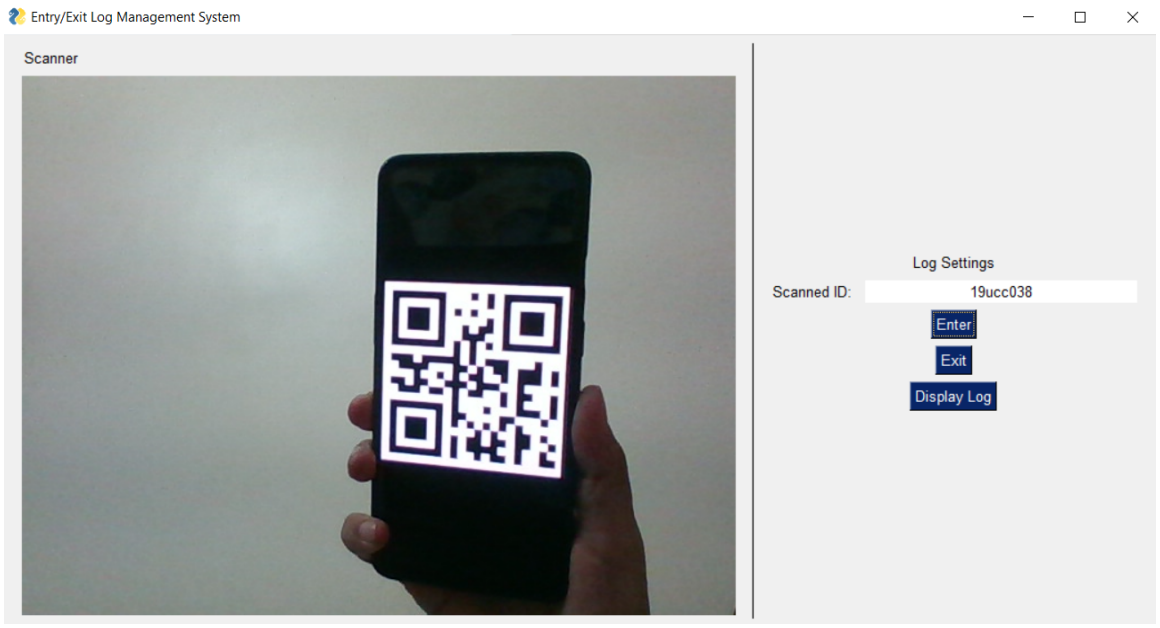


- The scanner section includes the Display component. The display component shows the frames captured by the camera sensor. It deals with scanning the data from barcodes. The local system receives the data from the camera sensor in grids of pixels converted to bytes and then sent to the display component.

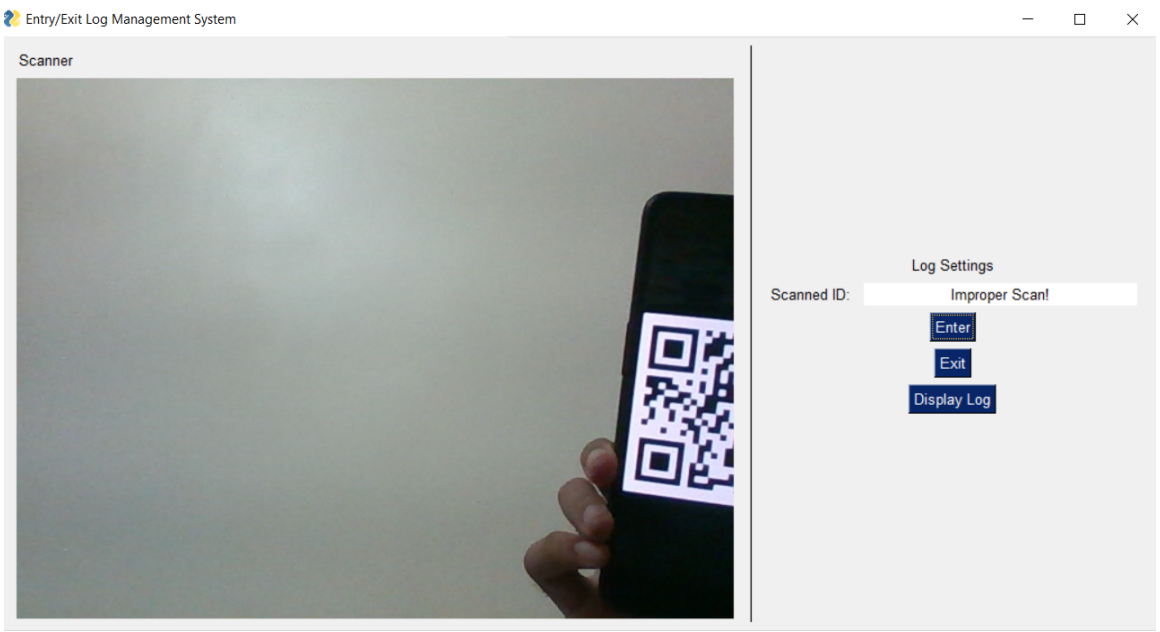
- The log settings section includes three buttons Enter, Exit and Display Logs. The functionality of each button is displayed below:
    - Enter Button: Serves as a trigger to scan for barcode/QRcode.
    - Exit Button: Used to close the application.
    - Display Log Button: Is used to display the logs recorded on a particular day by selecting the appropriate record(.csv) file.



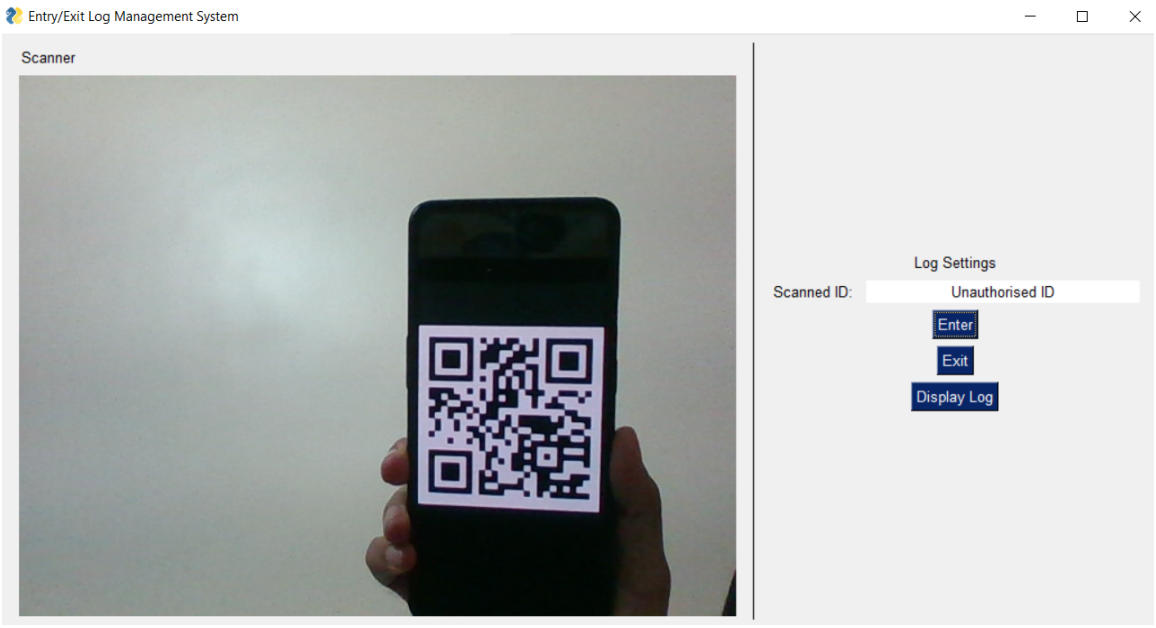
- **Scanning for QR/barcode**: Click the enter button on the application. Alternatively, the same functionality can be invoked by pressing the “Enter” key from the keyboard. If the ID from the scanned barcode is valid and present in the database, then the Scanned Id Label displays the scanned ID, as illustrated in the image below.



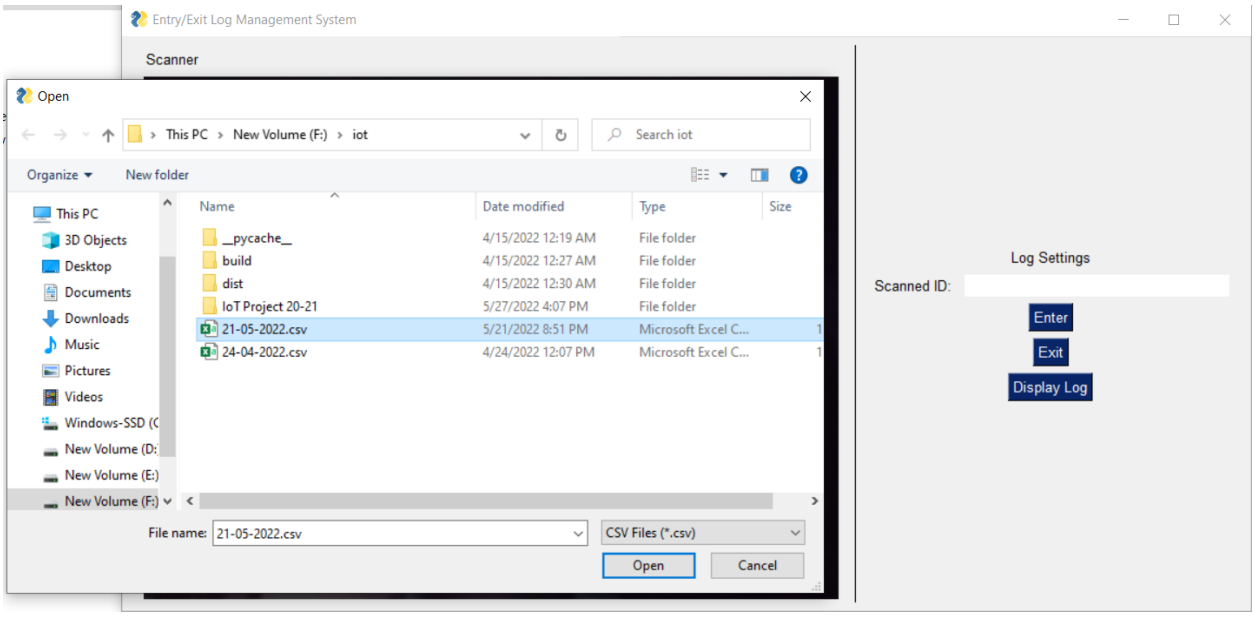
  - If the Scanned Id is not placed or scanned correctly, the Scanned Id Label displays ‘Improper Scan’, as illustrated below.



  - If the Scanned Id is invalid or not present in the database, the Scanned Id Label displays ‘Unauthorised ID’, as illustrated below.

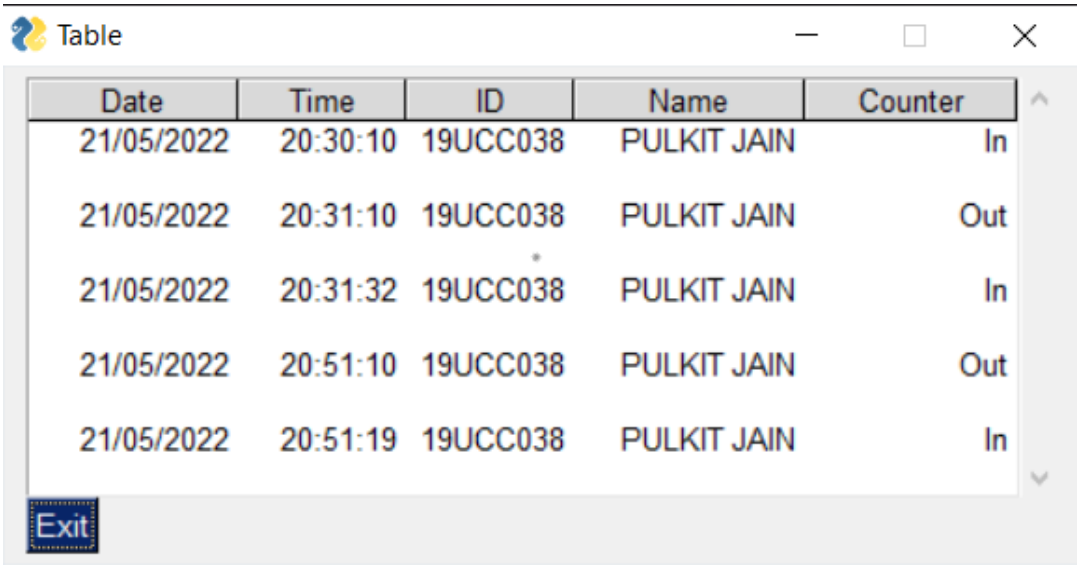


- **Displaying the Recorded Logs**: The user needs to click the display button from the main application window to indicate the recorded entries. The user then needs to choose the file named the required date as per the requirement from the popped-up window.



  - After selecting the required data(.csv) file, a new window will open consisting of all the entries present in the selected data file displayed in a tabular format. This window can be closed by pressing the Exit Button present at the bottom left corner of the window.

  - After pressing the Exit Button the user returns to the application window.



1.4 Conclusion

The problem can be solved by an IoT level 2 system. It is an affordable design that allows convenience and security for institutes in maintaining student entry records. The design is relatively short and easy enough to install with just a couple of steps. RFID can identify everyone by the barcode printed on the cards uniquely. Moreover, using the RFID barcode for entry registration based on the cloud will be costless because all students have their RFIDs issued and authorised by the institute. This feature has been used to make entries by scanning the code on the RFID using OpenCV. With the centralisation of data on a real-time database on google firebase, the entry needs to be done at only one location, saving both time and cost. However, using cloud computing and mobile technology in education institutions costs less than a traditional data centre.

In many cases, entry recording is inaccurate because it relies on manual monitoring and manual recording. Therefore, the proposed system helps to complete recording students' movements through automated methods based on the recognition of the RFIDs. The proposed model will help to reduce errors and reduce the time and effort required to complete the student entry registration process. Moreover, the standalone desktop application is based on plug and play. This executable can be installed on any system and handled by individuals with varied technical backgrounds.

1.4.1 Code

https://github.com/Utkarsh212/IoT-Project-20-21

1.4.2 Result

The following video explains the project:

https://drive.google.com/file/d/14Zw1I2xxGVjDwFsw6wOjm1V_Eu4T2E4T/view?usp=sharing

Bibliography

[1] Arshdeep Bahga, Vijay Madisetti: "Internet Of Things: A Hands-On Approach," 2014

[2] I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, "A brief introduction to OpenCV," 2012 Proceedings of the 35th International Convention MIPRO, 2012, pp. 1725-1730.

[3] Iranna M Shettar, "Quick Response (QR) Codes in Libraries: Case study on the use of QR codes in the Central Library, NITK," TIFR-BOSLA National Conference on Future Librarianship-2016At: Tata Institute of Fundamental Research, Mumbai, 2016

[4] Chunnu Khawas, Pritam Shah, "Application of Firebase in Android App Development-A Study," International Journal of Computer Applications, 2018, 179(46):49-53

[5] Krithivasan Ramamritham, "Real-time databases," Distributed and Parallel Databases, 1993, 1(2):199-226

[6] S. Goyal, S. Yadav and M. Mathuria, "Exploring concept of QR code and its benefits in digital education system," 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2016, pp. 1141-1147, doi: 10.1109/ICACCI.2016.7732198.

[7] S. H. Son, "Real-time database systems: present and future," Proceedings Second International Workshop on Real-Time Computing Systems and Applications, 1995, pp. 50-52, doi: 10.1109/RTCSA.1995.528750.

[8] S. Samaiya and M. Agarwal, "Real time database management system," 2018 2nd International Conference on Inventive Systems and Control (ICISC), 2018, pp. 903-908, doi: 10.1109/ICISC.2018.8398931.

[9] Suchandra Datta, "How to Get Started with Firebase Using Python," freeCodeCamp, 2021

[10] Murtaza's Workshop - Robotics and AI, "How to Detect QRCode and BarCode using OpenCV in Python," Online Murtaza's Workshop, 2020

[11] World Wildlife Fund, 1250 24th Street, N.W. Washington, DC 20037 | https://www.worldwildlife.org/industries/pulp-and-paper