**Project Solution Report**

Prepared For

Professor Jacky Lau

SEP600(Embedded Systems)

By

Kannav Sethi and Arnav Nigam

SEP600(Embedded Systems)

**14th April 2024**

**Introduction**

The manual process of tracking attendance at large-scale events can often lead to inefficiencies and inaccuracies, plus it also requires a lot of manual hard work. We propose to develop an RFID-based attendance tracking system to solve the manual problem of taking attendance. This system will utilize RFID technology to automate attendance management, providing real-time data to event organizers.

**Problem Statement**

Manual attendance tracking at events could be more efficient and error-prone which also involves outdated methods like paper sign-in sheets and manual data entry that can lead to bad handwriting, missing entries and duplicate records. This creates discrepancies in attendance records and consumes substantial time and effort from organizers which leads to the distraction of the user from other vital event management tasks and reduces participant engagement.

**Summary of Existing Solution**

Existing solutions to combat the issue of manual attendance at large-scale events already exist as seen in the reports by Shrivastava et al (2023) and Harum et al (2023). Both reports have been seen utilizing RFID tags and RFID reader for the automation process and using ESP8266 to transmit that information to the backend server for further processing. These solutions, while innovative, have limitations in scalability and real-time data management. Moreover, they do not fully exploit IoT capabilities for continuous monitoring and instant data access.

**Proposed Solution**

The solution that we propose aims to further advance on the solution that already exists by including a physical Real Time Clock Module, this is done based on the fact that we want to have much more precise readings of the timestamps and also to reduce the latency between sending the request to RTC module and getting the timestamp and it also leads to independence from relying on secondary API’s to getting time, which can again lead to various security issues. The server that we will be using here will be made using **Flask** in **Python** and we will be storing our data records in a relational database made using **SQLite in Python.** The reason for this being is we would need the dynamic capabilities of Python to plot a graph and also fetch and store records in a database simultaneously in P.ython with good efficiency. We would also be using a buzzer, to provide auditory feedback to the user

A diagram of a computer system

Description automatically generated

**Fig 1 - System/Circuit Diagram**

**Issues Faced**

There were a lot of issues that we faced while we were developing the solution:

* Since our data was being transferred between 3 components in the system i.e. ESP32, FRDM K66 Microcontroller and Python Server, there were a lot of synchronization errors. Some of the time even after ESP32 was connected to the python server, the data was still not being accepted by it. The data was still in the buffer and this was interrupting the normal flow, because there were backlogs from previous transmissions, So we had to create a function to clear the buffer before beginning the normal transmissions.
* When we were setting up the Python server, our initial idea was to use a NoSQL-based database called MongoDB, but while implementing that we faced a lot of problems with establishing a relationship between the table that tracks the activity of a user and the one that contains all the user information, so after testing various situations we decided to switch to SQLite3, which is a relational database. This established clear boundaries between different relationships.
* Another issue that we faced was with the LCD’s, we were sending the right data to LCD, but even after clearing the LCD using **lcd.cls(),** the characters for the word were not appearing right, turns out this was due to us not having a delay between **lcd.cls()** and **lcd.printf()**, so we had to establish a small delay between that.
* One of the biggest issues that we faced while developing this solution was that we were trying to operate a buzzer that had an operating voltage of 8v-12v, we tried using a transistor with an external battery, but doing so fried one of the boards that we had. So, instead, we decided to switch the buzzer to the one that operates on 3.3v.

**Health and Environmental Consideration**

The RFID attendance system uses energy-efficient components to minimize power consumption and carbon footprint. Health risks are mitigated by ensuring that all electronic components are not in direct contact with human beings and operate within regulated radiation and noise levels to protect users.

**Future Advancements**

In the future, we could add machine learning capabilities to this project to predict the event scheduling trends based on the data gathered by the RFID reader. Also, we can include sensors that monitor the environment around the people such that we can improve the event safety and comfort based on the data coming from the system.Moreover, we can also create an app for the administrators to track and record the events statistics based on the user entering the events.

**References**

Shrivastava, A., Suji Prasad, S. J., Yeruva, A. R., Mani, P., Nagpal, P., & Chaturvedi, A. (2023). IoT Based RFID Attendance Monitoring System of Students using Arduino ESP8266 & Adafruit.io on Defined Area. *Cybernetics and Systems*, *ahead-of-print*(ahead-of-print), 1–12. <https://doi.org/10.1080/01969722.2023.2166243>

Harum, N. B., Ahmad Mahin, N. A., Hamid, E., Emran, N. A., Anawar, S., & Asnawi, A. L. (2023). RFID Attendance System with Contagious Disease Prevention Module using Internet-of-Things Technology. *International Journal of Interactive Mobile Technologies*, *17*(18), 4–15. <https://doi.org/10.3991/ijim.v17i18.41603>