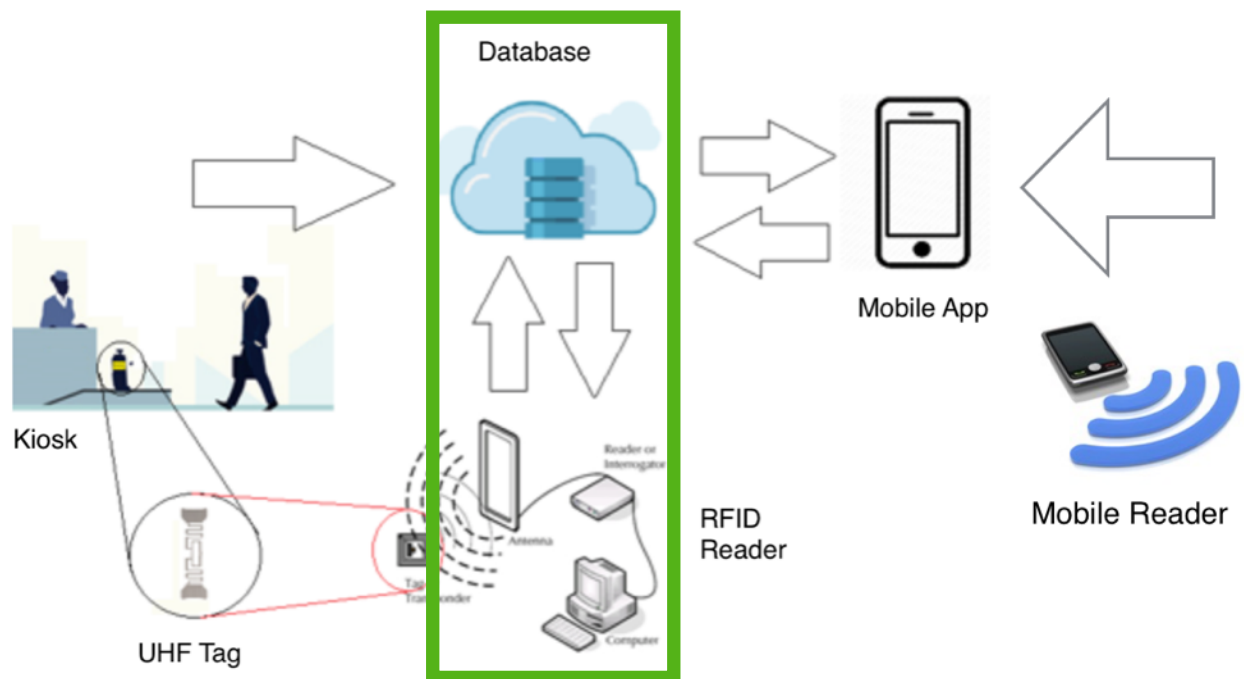


RFID READER/DATABASE INTEGRATION



Introduction

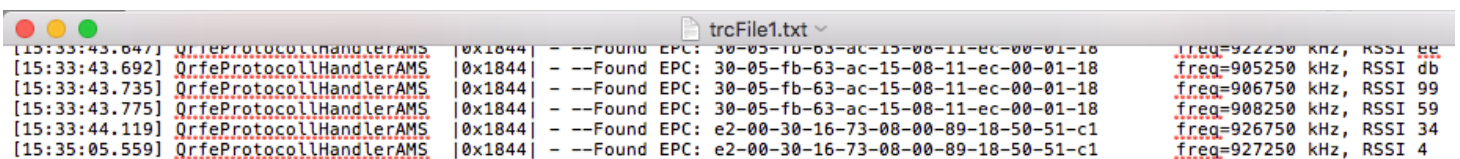
Design Team 1's RFID Luggage Tracking project consisted of two main components —the RFID scanner operations, and the iOS Application and data management. The operation of the RFID scanners results in data logs detailing the status of the scanners' reads while the web server hosts the MySQL database from which the iOS Application pulls information. The relay of the information from the RFID reader to the database will be detailed in this Application Note.

Objective

This Application Note is a user guide detailing the scripts used throughout the project to manage the database in which the travelers' information is stored. From check-in at a kiosk until the traveler retrieves his or her bag, proper tracking of the luggage must be ensured. By updating the status of the luggage at intervals throughout the handling process with the use of Python scripts, the likelihood of loss and misplacement decreases significantly.

The Operations

The LinkSprite IDE associated with the RFID reader includes scanning and handling functions which facilitated the process of correctly identifying a unique tag. By clicking the "Start Scan" button within the IDE, the microcontroller activates the RFID reader module, emitting a signal for UHF tags to pick up. Upon receipt of the reader's signal, the tag will respond with a unique code which is then written to a file in the format below in Figure 1.



```
[15:33:43.647] QrfeProtocolHandlerAMS | 0x1844 | --Found EPC: 30-05-fb-63-ac-15-08-11-ec-00-01-18 | freq=922250 kHz, RSSI 99
[15:33:43.692] QrfeProtocolHandlerAMS | 0x1844 | --Found EPC: 30-05-fb-63-ac-15-08-11-ec-00-01-18 | freq=905250 kHz, RSSI 99
[15:33:43.735] QrfeProtocolHandlerAMS | 0x1844 | --Found EPC: 30-05-fb-63-ac-15-08-11-ec-00-01-18 | freq=906750 kHz, RSSI 99
[15:33:43.775] QrfeProtocolHandlerAMS | 0x1844 | --Found EPC: 30-05-fb-63-ac-15-08-11-ec-00-01-18 | freq=908250 kHz, RSSI 59
[15:33:44.119] QrfeProtocolHandlerAMS | 0x1844 | --Found EPC: e2-00-30-16-73-08-00-89-18-50-51-c1 | freq=926750 kHz, RSSI 34
[15:35:05.559] QrfeProtocolHandlerAMS | 0x1844 | --Found EPC: e2-00-30-16-73-08-00-89-18-50-51-c1 | freq=927250 kHz, RSSI 4
```

Figure 1 - Scan log

When a tag is identified, trcFile1.txt is appended with respect to the time of the scan, the protocol used to handle the signal, the size of the scan, the unique tag epc code, the frequency of the signal, and the strength of that signal. Because this project revolves entirely around the proper recognition of an epc code, it is important that that small bit of information be extracted from the text file.

To simulate the characteristics of the scanner for software development, the script in Figure 2 on the next page, named write_trc.py, was created. The script infinitely loops in two second intervals, appending trcFile1.txt with a portion of the text above. The epc code in this script, 30-05-fb-63-ac-15-08-11-ec-00-01-12, will be referenced throughout this report.

As the luggage progresses through the airline system, its status will be reflected in the database according to the reader it most recently passed. Each reader throughout the routing process would be hardcoded to reflect the position associated with it (Checked In, On Plane, At Destination, On Carousel X, etc). For demonstration purposes, the reader will be hardcoded to show that a bag has just passed the reader associated with its location. The reader would likely be placed above the belt before the luggage is moved to the revolving carousel, detailed in Figure 6 below.

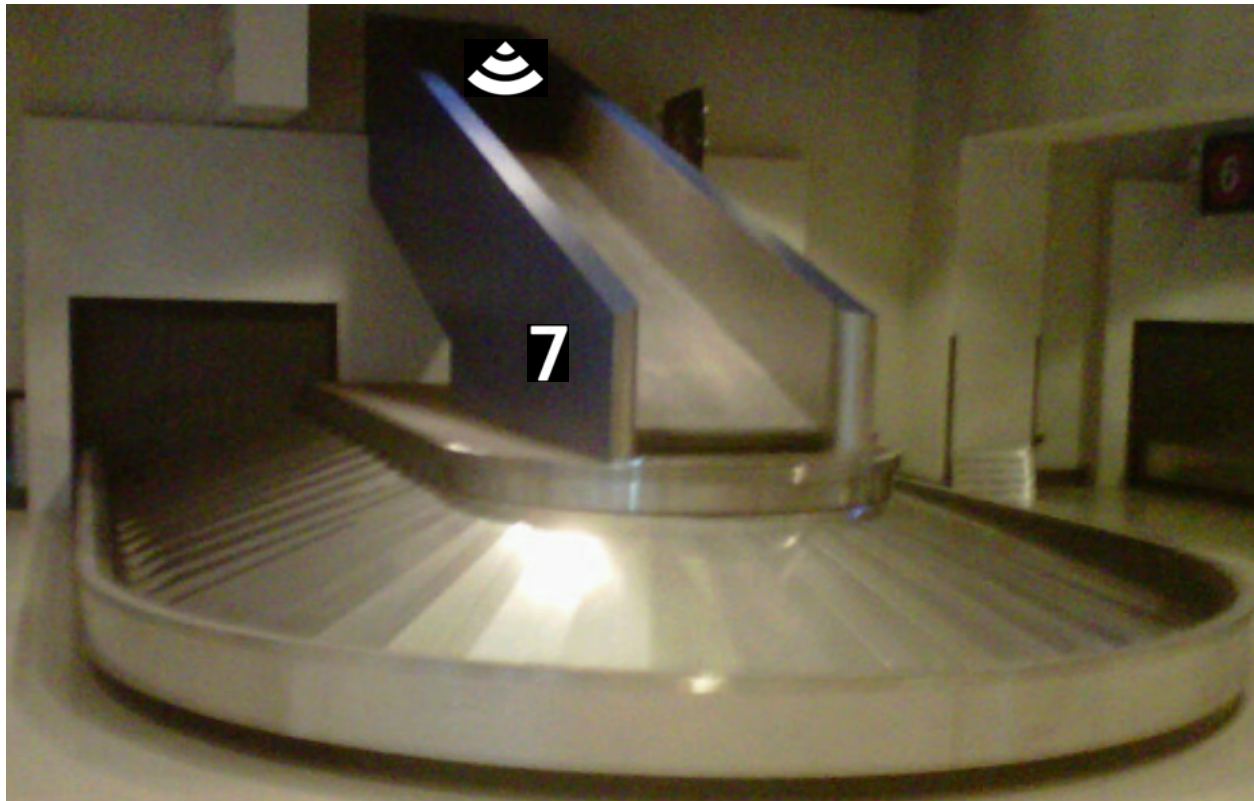


Figure 6 - Final Reader Above Carousel

The final update would be sent to the database for the traveler to pull from, indicating that his or her bag may now be retrieved from a specific carousel. This information can be seen in the database in Figure 7 below.

name	departure	destination	status ▾	flight	tag_id	bag_status
John Smith	DTW	LAX	On Time	4840A	30-05-fb-63-ac-15-08-11-ec-00-01-12	On Carousel 7

Figure 7 - Final update of bag status

Conclusion

By mounting scanners throughout an airport to carry out location-specific tasks, airlines can greatly improve their efficiency and therefore their customers' satisfaction. For greater accuracy, many scanners should be installed, however once in place, little to no maintenance will be required. The code used throughout this Application Note and in the Appendix can be used to update the status of luggage in an airline's system and can be used to automate the routing conveyor belts. Automation is the future of nearly every industry due to the high reliability and low labor costs.