C.L.A.I.M - Computerized Luggage and Information Messenger

Ernest Jackman, Adrian McGrath, Tomasz Pytel

DEPT. OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE, UNIVERSITY OF CENTRAL FLORIDA, ORLANDO, FLORIDA, 32816-2450

***Abstract* — Airport terminals are typically crowded in baggage claim areas while passengers stand around waiting for their luggage. This group's project would help alleviate that burden by attaching RFID tags to luggage and scanning bags when they are loaded on the carousel. Once the owner of the luggage is identified, they are notified via either a text or mobile application notification. A television display mounted next to the baggage carousels also shows information of luggage being unloaded as well as those on the carousel. This would result in a more orderly luggage retrieval process, compared to the current chaotic process.**

***Index Terms* — Airport, Android, Raspberry Pi, RFID.**

I. INTRODUCTION

There are no products on the market which address the specific situation our group chose to address, however some similar products do focus on improving the airport's baggage handling system accuracy rather than improving the passenger user experience. Our project will instead focus on enhancing the passenger experience while simultaneously providing tracking possibilities for the airport.

The goal of the system is to read data from RFID chips located on tags attached to the luggage to communicate with the airline. The information for identifying the owner of the bag is found on the RFID chip. The passenger's airline is then asked to notify the passenger that their bag has arrived. This is accomplished by a RFID transceiver scanning the RFID tags, and the data is sent to a microcontroller which then sends it over to another device over a wireless connection. This new device then communicates with the airline's web services to request that they notify the appropriate owner of the bag. The notification of the owner is handled by the airline service, and the received data is then transmitted to a connected monitor. This monitor is wirelessly connected and displays the real-time luggage data to passengers waiting in the baggage claim area. The system is robust enough to handle communication with different airlines to accurately identify the luggage owners. This system is mounted so as to utilize the empty space above existing carousel systems. The system is powered by a plug to an outlet in the setup environment.

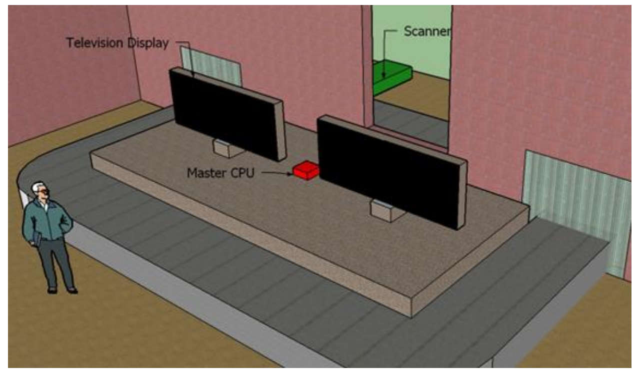


Fig. 1. This is a model depiction of the system in use at an airport carousel.

Figure 1 portrays the overall system setup of all required components necessary for function. The scanner unit located in the back communicates with the master CPU unit placed in the outside public area. The master CPU sends the data information to the display unit and sends a notification to the luggage owner either via a mobile app push notification.

II. RELATED WORK

Even though RFID technology stems from the 1980s, it has not been as widely or prominently utilized in modern times as one would expect. There are several noteworthy products that are similar to this group's project solution. However, even though there are a few similar systems being worked on, those systems are tailored more for the benefits of the airports and airlines rather than the benefits of the passengers. This group decided to focus on a solution more suited for the passenger's benefit.

The Hong Kong International Airport (HKG) has started using only RFID Baggage tags as early as 2009 for all of the bags that leave the airport every day. This RFID system has been proven to yield an improved handling capacity compared to the standard paper tag bar code system used in place today [2]. These RFID tags are encoded with unique ID numbers, 3 letter IATA code describing the destination airport, as well as the date and flight number. This specific ID number links to the back end baggage handling systems the passenger's information.

Air France-KLM launched an eTag in 2014 which includes two e-ink displays. Partnered with the eTrack system which makes use of GSM, GPS, and Bluetooth technology enables the luggage to be easily tracked by any smartphone. This system was designed to benefit the airlines as well as the passengers and allows linking of different luggage bags to an account which can then be used for a streamlined check-in progress [3]. A company in the United Kingdom by the name of ReboundTAG has created permanent RFID luggage tags for airlines [4]. This tag has two microchips, where one tag is used for identification, which remains permanent and can be used to help automate the check-in process. The other one is re-writable so that airlines can write new flight information on each trip. This tag can also be tracked with Worldtracer to locate the luggage and identify owner of its whereabouts. This is a pretty similar product to the group's approach; however it includes an additional tag that houses the passenger's personal information. For security reasons the group wanted to limit the tags to one microchip and make sure they are re-writable so they could be reused as a smarter economical option.

III. APPROACH

The idea of using RFID technology to improve the customer satisfaction at the airport, comes from the tags ability to hold information on a portable device that does not require maintenance or internal power. These tags can be used to contain the needed information to update a visible display and provide a message to the customer's personal device when the luggage is ready for pickup. The current demonstration represents a proof of concept of this system; to scan luggage and notify the passengers of the luggage arrival. Modifications from a full scale device to a small demonstration model were done by reducing power output and frequency of the scanning unit.

In this demonstration, a High Frequency Near Field Communication device is used as the scanning unit with external antenna. At these frequencies, the range of tag scanning is limited to a few inches from the antenna surface. Power output from this demonstrations device is also low enough to satisfy FCC regulation on intentional radiating devices, and requires no additional safety measures that a full scale system would require.

A. RFID Tags

The RFID tags used are passive tags. This allows for cheap mass production, and can be thrown away after the consumer has retrieved their luggage. By actively powering the tags with the scanning unit, each tag is able to transmit its unique ID, along with any other relevant information the airline wishes to include on the tag. The unique ID is made up of 10 hexadecimal values which allow for the creation of many tags that can be active in the system across multiple airlines and flights. The additional information that can be include on the tag can include, consumers name, airline, and flight number. Contact information can also be included on the tag or through a service connection with the airport to send a notification directly to the consumer that their luggage is present on the conveyer system.

B. Microcontroller Scanning Unit

The scanning unit consists of three subsections. The RFID chip that decodes tag information, a Bluetooth transceiver, and the microprocessor that will send control signals to the RFID chip, along with information exchange between the RFID chip and the Bluetooth.

The RFID scanning chip is the TRF7970A from Texas Instrument. It communicates with the MSP430 microcontroller through a Serial Peripheral Interface. This allows for bi-directional communication as the control signals from the MSP430 are sent to trigger the turning on/off of the RF power output for passive tag activation. Any tags in range will be powered up, and its data decoded by the RF demodulation circuitry and passed through the SPI line back to the MSP430.

The MSP430 itself is controlling the communication protocols between the TRF and the Bluetooth LMX unit. It initializes both units for UART and SPI communication protocols, and send a boot command to the TRF to initialize all memory addresses into read mode. The MSP will also initialize the LMX into transparent Bluetooth mode, where the Bluetooth will wait for a valid connection, and send all information present on the UART line directly to the connected device as raw data.

The LMX Bluetooth transceiver is set up to pass all information presented to it on the output of the MSP430 and pass it to the Master CPU though a valid Bluetooth connection. This data will include any pertinent information stored on the tag that is relevant to update the output display with a customer's information and send out a notification to the customer.

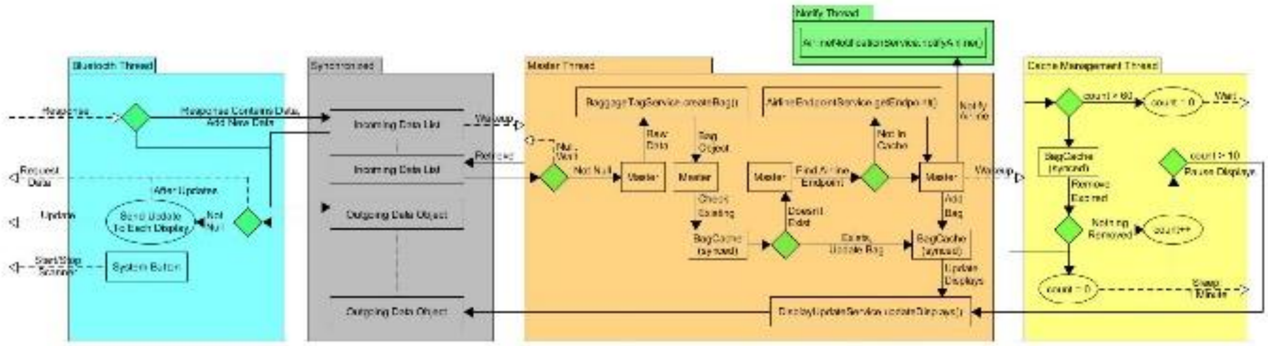


Fig. 2. This is the full threading process diagram

C. Master CPU

The Master CPU is tasked with managing and caching all of the baggage data, as well as managing all of the communications between devices and telling the airlines when to send the notifications to the passengers. The Master CPU communicates with the Scanning Units and the Display Devices with a Bluetooth communication.

A Raspberry Pi B+ 2 is used for the group's Master CPU. The Raspberry Pi was chosen because it offered a 900MHz quad-core processor, 1 GB of RAM, 4 USB Ports, and has a large support community. The Raspberry Pi is running the Raspbian Wheezy Operating System, which is a subset of Linux. The code for the project is written in Java 8 and runs on the pre-existing Java Virtual Machine. The Master CPU requires a Bluetooth Connection and an Internet Connection.

For the project, the group used the TRENDnet Micro-Bluetooth Micro USB Adapter. The group originally planned on using the Edimax Wifi Micro USB Adapter for wireless dynamic Wifi, however the group was unable to get the Dynamic Wifi to work correctly on the Raspbian Operating System. Instead of Wifi, the group is using Ethernet for Internet Access.

The Master CPU contains a configuration file to specify many of the settings for the system. This configuration file contains the name of the airport the system is in, the carousel the system is attached to, the internet credentials needed to access the airport's provided internet, the list of Bluetooth MAC Address for the Scanning Units, the list of Bluetooth MAC Addresses for the Display Devices, and the timeout value for the bags. The timeout value should be configured to be 1.5 times the amount of time it takes for the bag to travel the entire length of the carousel.

The largest concern the group had was ensuring the Bluetooth Communication was not blocked by the Baggage Processing or the Passenger Notifications. The rate at which bags would be scanned is unknown, and the rate at which the monitor needs to be updated is also unknown. Due to these concerns, the group designed a complex multi-threaded design for the Master CPU, which the Raspberry Pi's CPU would be more than capable of processing. The system was split up into three primary threads. The threads are periodically put into the wait state when they are not needed. Certain events in the system will notify the threads to continue running.

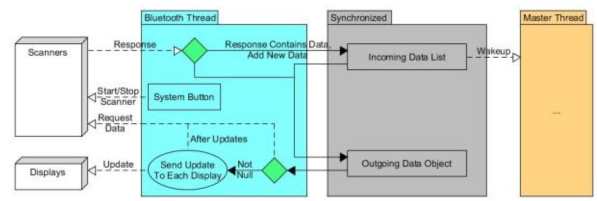


Fig. 3. This is a model depiction of the system in use at an airport carousel.

The first primary thread was the Bluetooth Thread, which is in charge of managing all of the Bluetooth Communications. Figure 3 shows the diagram of the Bluetooth Thread. The Bluetooth Thread establishes a client connection with the Scanning Units and waits for scanned tag data. The tag data is then passed into a synchronized Incoming Data List. When data is passed into the synchronized Incoming Data List, the Master Thread is notified.

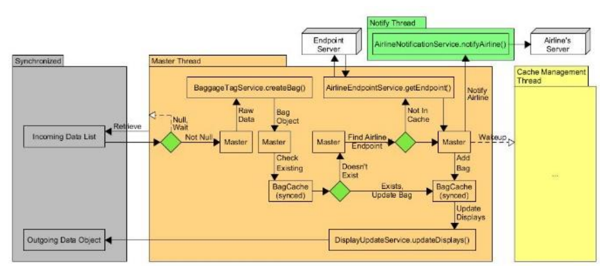


Fig. 4. This is a model depiction of the system in use at an airport carousel.

The synchronized Incoming Data List is consumed by the Master Thread, which is the second primary thread. The Master Thread consumes the synchronized Incoming Data List until it is empty, at which point the Master Thread puts itself into the wait state. Figure 4 shows the diagram of the Master Thread. The Master Thread converts the raw scanned tag data into a Baggage object. The Baggage Cache is then checked to see if the Baggage object already exists. If the Baggage object already exists in the Baggage Cache, it means the Bag has been rescanned. The Bag's scan and timeout values are updated and a new message for the Display Devices is created. If the Baggage object doesn't already exist in the Baggage Cache, it means this is the first time the Bag has been scanned. The new Baggage object is given a timeout value, and a message is sent the Airline Web services to notify the passenger that their bag has arrived on the carousel. The new Baggage object is added to the Baggage Cache, and a new message for the Display Devices is created. Additionally, the Cache Management Thread is also notified when a new Baggage object is added to the Baggage Cache.

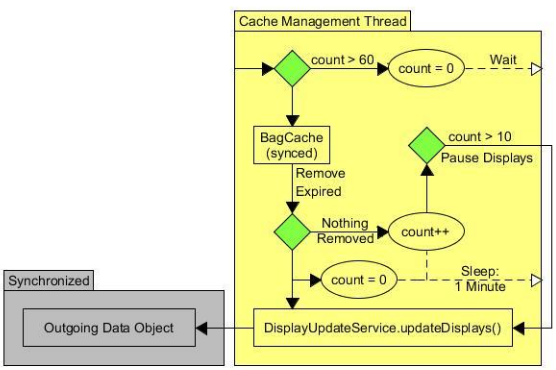


Fig. 5. This is a model depiction of the system in use at an airport carousel.

The Cache Management Thread is the third primary thread. Figure 5 shows the diagram of the Cache Management Thread. It contains a counter, which represents the number of time increments since the last bag was added to the system. The thread checks the Baggage Cache for bags which have passed their timeout value. If the bag has passed its timeout value, it can be assumed that the bag has been removed from the carousel. If no bags need to be updated when the thread's process runs, the counter value is incremented. The thread then waits for one minute. If the counter value reaches 10, indicating ten minutes since the last bag timeout, a message is sent to the Display Devices to tell them to go into Standby Mode. If the counter value reaches 60, indicating an hour since the last bag timeout, the system is assumed idle and the Cache Management Thread puts itself into the wait state.

The created message for the Display Devices is a Json Message and is sent to the synchronized Outgoing Data Object. The Bluetooth Thread checks the synchronized Outgoing Data Object between retrievals from the Scanning Devices. If it finds a Json Message waiting to be sent to the Display Devices, it passes that value on to each Display Device and clears the value in the synchronized Outgoing Data Object.

D. Web Services

The airline's implemented web service runs whatever server-side language they wish to implement with. It is up to the airline to determine what information should be sent to their webservice. They determine this information by how they set up their tag data. The purpose of this web service is to send notifications to the passengers that their bag is available on the carousel to be picked up. It is the host airline's choice as to whether they would like to implement their notifications as a text message, as a mobile application push notification, or in some other fashion. The host airline can also use this service for updating their internal baggage systems and databases.

```
{
    "identification": "bagId",
    "flight": "flightId",
    "arrivalAirport": "airportIATA",
    "carousel": "carouselNumber",
    "time": "time",
    "contact": "contactInfo"
}
```

Fig. 6. This is a model depiction of the system in use at an airport carousel.

The web service the airline is expected to implement should consume a Json Message. An example of this Json Message can be seen in Figure 6. This web service is not expected to return a result to the Master CPU. The identification field should contain the bag's identification number. The flight field should contain the flight identification number. The arrival airport should be the three alphanumeric IATA code for the airport. The carousel field should contain the carousel number the bag was scanned at. The time field should contain the time the bag was scanned. The contact field should contain the contact information for the passenger.

For the group's web service, a Java RESTful web service was created. The web service is hosted on www.livingbucket.com, which is a domain owned by one of the members of the group. Two web services were implemented for demoing and testing. The first web service was for the test airline with the IATA code "FL". This web service can be found at: http://www.livingbucket.com/webservices/claim/notifyFLPassenger. The second web service was for the test airline with the IATA code: "US". This web service can be found at: http://www.livingbucket.com/webservices/claim/notifyUSPassenger. The Json Messages are sent to the web services using the POST http method. The group passed the Android Device Token in the contact field to the web service. The group's web service then uses the Google Cloud Messaging Service to send Mobile Application Push Notifications to the passenger's registered device.

Additionally, the distributor of the system would be expected to implement a web service to verify that a given airline is supported by the system. This service simply needs to accept the airline's IATA code in the url path. The group's example web service can be found at: http://www.livingbucket.com/webservices/claim/FL or http://www.livingbucket.com/webservices/claim/US.

E. Display

The display device needed to be able to receive data via a wireless serial device from the single board computer, prepare the data for display, and send the information to the TV to be displayed. After the tag data is properly cached on the Raspberry Pi it is sent to the display unit to be displayed. The group decided to utilize existing monitors already in place at airport carousels. This decision removes the need for unknown lengths of cable to wire additional monitors required for data to be displayed. In order to repurpose existing monitors being used for advertisements, the Android Stick hardware device was chosen. Specifically, the MK808b Plus device was used. Using these devices allows multiple monitors to be used to display the information when the Raspberry Pi broadcasts the json data information file.


```
{"Bags:"
 {
  {"airline":"FL","flight":"744","paxName":"J. Smith"},
  ...
 }
}
```


Fig. 7. This is an example of the contents of the Json data file being sent to the Android stick from the Raspberry Pi.

This Json file, as seen in Figure 7, contains a list of bags from the latest update with the display information specified by the airlines. Each line depicts one bag object with the corresponding IATA airline code, the flight number, and the passenger name. After the file is parsed, the android stick will update its cache and display the new information accordingly.

The monitor requirement consists of any HDMI based television to allow the android stick to transmit its information to the video receiver. Taking size in to consideration, the resolution size is only required to be high enough to accommodate the font size needed for displaying tag information at distance.

The Mk808b Plus Android Stick receives the data information from the Raspberry Pi via the built-in Bluetooth receiver. Depending on the command contents being sent from the Raspberry Pi, the Android Stick will either proceed to cache the new information it and send it to the monitor to display, or enter standby mode. Standby mode displays advertisements on the screen and returns to displaying luggage data after the next prompt from the Raspberry Pi. This mode is also turned on when the system is idling when not in use.

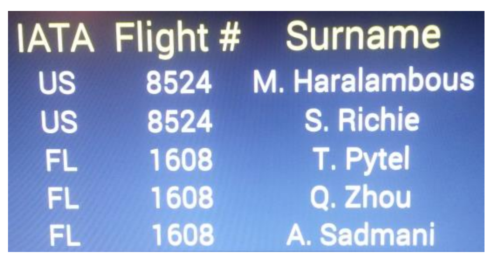


Fig. 8. This is depiction of the data being displayed on the monitor at the airport carousel.

The data displayed on the monitor is dependent on what the airlines choose to display and would be defined for passengers before they opt into the service. The group is going to display the IATA airline code, the flight number, and the passenger surname accompanied with the first initial of their first name. This can be seen in Figure 8.

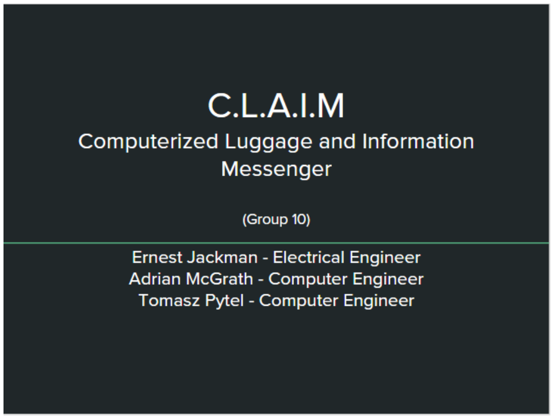


Fig. 9. This is depiction of the standby mode being displayed on the monitor at the airport carousel.

During the standby mode of the display feature, any existing advertisement could go in the place of the group information placeholder as seen in Figure 9. Simply adding this additional feature ensures that the monitors are still being used for advertisement time as well as displaying the C.L.A.I.M. project information for the extra service for the passengers. This form of notification coupled with the mobile app push notification allows passengers freedom to partake in other activities while they wait for their bags to appear.

IV. EXPERIMENTAL RESULTS

The group ran ten tests with the system. The group collected different bits of data during those tests. The total amount of time it takes to scan the baggage tag to when the bag was displayed on the screen was measured.

Test Attempt	Time (Seconds)
1	1.1
2	1.53
3	1.25
4	1.21
5	1.26
6	1.2
7	1.11
8	1.1
9	1.15
10	1.15

Table 1 This table depicts the total amount of time it took to scan the baggage tag and display the data.

The time for each test was recorded. The results of each test can be found in Table 1 above. The average time for these tests came to about 1.2 seconds. The total amount of time it takes to scan the baggage tag to when the passenger was notified that their baggage is on the carousel.

Test Attempt	Time (Seconds)
1	1
2	1.43
3	1.15
4	1.1
5	1.16
6	1.19
7	1.1
8	1.1
9	1.09
10	1.13

Table 2 This table depicts the total amount of time it took to scan the baggage tag to when the owner of the bag was notified.

The time for each test was recorded. The results of each test can be found in Table 2 above. The average time for these tests came to about 1.5 seconds. The group also kept note of how many bags were successfully identified. The group saw a 100% success rate for bag identification.

During testing, the group also made several other discoveries. The group found that the maximum length the scanner was capable of reading the baggage tag from is 11.5 centimeters. The group also found that the tag data would occasionally be incomplete. After a lengthy investigation, the group found that the reason for the incomplete tag data was the tag being on the edges of the RF radiation zone. Since the tag was not completely in the RF radiation zone, only some of data was correctly read by the Scanner Unit.

The group also noticed the Master CPU was requesting tag information from the Scanning Unit too quickly and the Scanning Unit was frequently not working. The group implemented a delay in the Master CPU between scan times. The group first tried a delay of one second, but then found that to be way too large. The group eventually found that a ten millisecond delay yielded the best result.

V. CONCLUSION

The project has been a valuable learning experience for the group during the two semesters spent designing and prototyping. The group learned how to work in a group setting, design and build a functional system, and how to properly document everything related to the project. During the first semester dedicated to system design, the group spent the majority of time researching and contributing ideas via discussions which provided a new experience for the group. Group members had no prior experience with any elements of the project technology and utilized outside sources to research the different aspects. Each group member had their specified role in the project completion process with each member working on a specific part of the system, and working together on the integration of the full system.

Due to this demonstration being a small scale prototype for proof of concept, the full system design will contain an Ultra High Frequency scanning unit, and increased power output. This will allow for better scanning range so that proper clearance of luggage on the conveyer tracks can be achieved. Additionally, depending on the type of conveyer system used by the airport, two optional mounting styles will be needed. These systems are shown in Figure 10, and Figure 11. They include a single layer flat track system where the scanning unit is placed over the surface or the track, preferably in the back working area for safety reasons. If the single layer flat track system is maintained on the consumer side only, it will be placed in an area with proper no entry radius placed near the scanning unit hanging over the track. For multi-layered tracks, all baggage is loaded in from a single conveyer feeder, and the scanning unit can be placed within the bumper that the entire luggage must pass by. This will also require a safety radius to warn of the active scanning device.

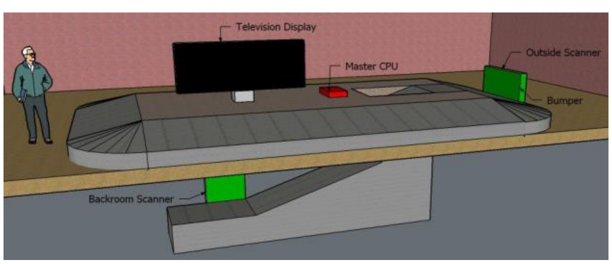


Fig. 10. This is a multi-layer conveyer system setup environment with all project components.

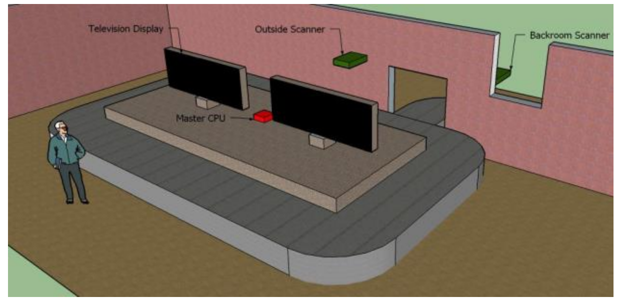


Fig. 11. This is a single layer conveyer system setup environment with all project components.

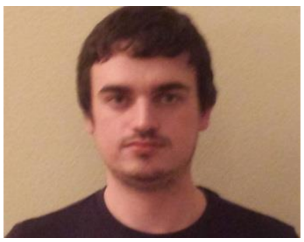
There were some initial difficulties with the microcontroller board design for the scanning unit, considering if the unit should be powerful enough to be self-contained, requiring large memory access and computing power, or separated into two processing devices. There were additional concerns regarding the Bluetooth communication between the Master CPU and the other devices, consisting of determining if each device would be connected independently though its own Bluetooth connection, or constructed into a Piconet. Each of these issues were resolved over time as the project progressed.

The complete design and prototype process of the project helped the group members gain necessary experience in developing and implementing possible future projects in the workforce. This includes the ability to design full electrical schematics, the conversion of these schematics into a PCB layout, along with the logistics of ordering parts for system construction. The group learned about how to use all of the different Bluetooth Protocols to communicate between multiple devices. Android development knowledge was greatly expanded upon during the development of the display device. The wisdom gained from working with the new technology gave wise insight to potential products that could be worked on in a workplace.

Overall teamwork and time management was needed to complete a working prototype within a set deadline. The group hopes the work presented in the project inspires future groups that wish to look in to this relatively untapped RFID technology and potentially improve on the design.

BIOGRAPHIES

Ernest Jackman is currently a senior of the Electrical Engineering department at the University of Central Florida. He has previous work experience in communication, and radar repairs as an Electrical technician in the U.S. Navy. He is currently working towards a Bachelor of Science in Electrical Engineering.



Adrian McGrath is currently a senior student of the Computer Engineering department at the University of Central Florida. He will receive his Bachelors of Science in Computer Engineering in December of 2015 and is going to pursue a working career as a software developer. He is hoping to get his own startup company running.

Tomasz Pytel is currently a senior student of the Computer Engineering department at the University of Central Florida. He will receive his Bachelors of Science in Computer Engineering in December of 2015 and is going to pursue a working career in the Computer Engineering profession.

ACKNOWLEDGMENT

The group members responsible for C.L.A.I.M would like to acknowledge the help, information and support that was provided by everyone involved in the success of the project. The assistance and time provided by the mentors and consultants was a valuable asset for the group. Special thanks go out to the following individuals: Steve Antonakakis, and Dr. Samuel Richie.

Since one of the group members had a patent on a potential new product, the group decided to meet with the company which held the patent rights [7]. The group communicated with NCR, National Cash Register, for some insight on possible implementations of the project creation. During the meeting, it was concluded what the best way to communicate with the airlines was, and how to handle the notification system for the customers. On top of the discussion about the project at hand, the group also managed to see how current single board computer systems are used in modern Kiosk construction and the different types of RFID cards currently in use.

The group also extends thanks and gratitude to the professors who have so kindly agreed to review the project that the group has worked so hard on.

REFERENCES

[1] Department of Transportation, 'Department of Transportation', 2015. [Online]. Available: https://www.transportation.gov/sites/dot.gov/files/docs/2015MayATCR_1.pdf. [Accessed: 05- Aug- 2015].

[2] R. Journal, 'Hong Kong Airport Says It Now Uses Only RFID Baggage Tags - RFID Journal', *Rfidjournal.com*, 2015. [Online]. Available: http://www.rfidjournal.com/articles/view?4885. [Accessed: 05- Aug- 2015].

[3] Future Travel Experience, 'Air France-KLM unveils new permanent bag tag and bag tracking device', 2014. [Online]. Available: http://www.futuretravelexperience.com/2014/03/air-france-klm-new-permanent-bag-tag-and-tracking-device-can-benefit-the-entire-industry/. [Accessed: 05- Aug- 2015].

[4] Reboundtag.com, 2015. [Online]. Available: https://www.reboundtag.com/. [Accessed: 05- Aug- 2015].

[5] Vanguardid.com, 'ViewTag', 2015. [Online]. Available: http://www.vanguardid.com/PermanentRFIDLuggagetags.aspx. [Accessed: 05- Aug- 2015].

[6] Qantas.com.au, 'Next Generation Check-in Regional Implementation Begins', 2015. [Online]. Available: http://www.qantas.com.au/travel/airlines/media-releases/mar-2011/5084/global/en. [Accessed: 05- Aug- 2015].

[7] McGrath; Adrian, "Baggage Delivery Notification System and Method" U.S. Patent 20140210623, issued July 31, 2014