# 2. Approach

The Quicket System reduces wait times and increases throughput by implementing a passive, contactless, and continuous entry system for nearly any event or venue. The system improves the queueing experience by using a thermal camera to detect the number of people entering the gate and an UHF RFID reader to scan tickets. The inputs from the thermal camera and RFID reader are then passed into the microcontroller, where the tickets are cross checked with a database. Then, the number of scanned tickets is compared with the number of people detected by the entry count subsystem. The microcontroller holds the system in a “pass” state until a mismatch occurs between the two numbers, at which point the system switches states and signals for human intervention.

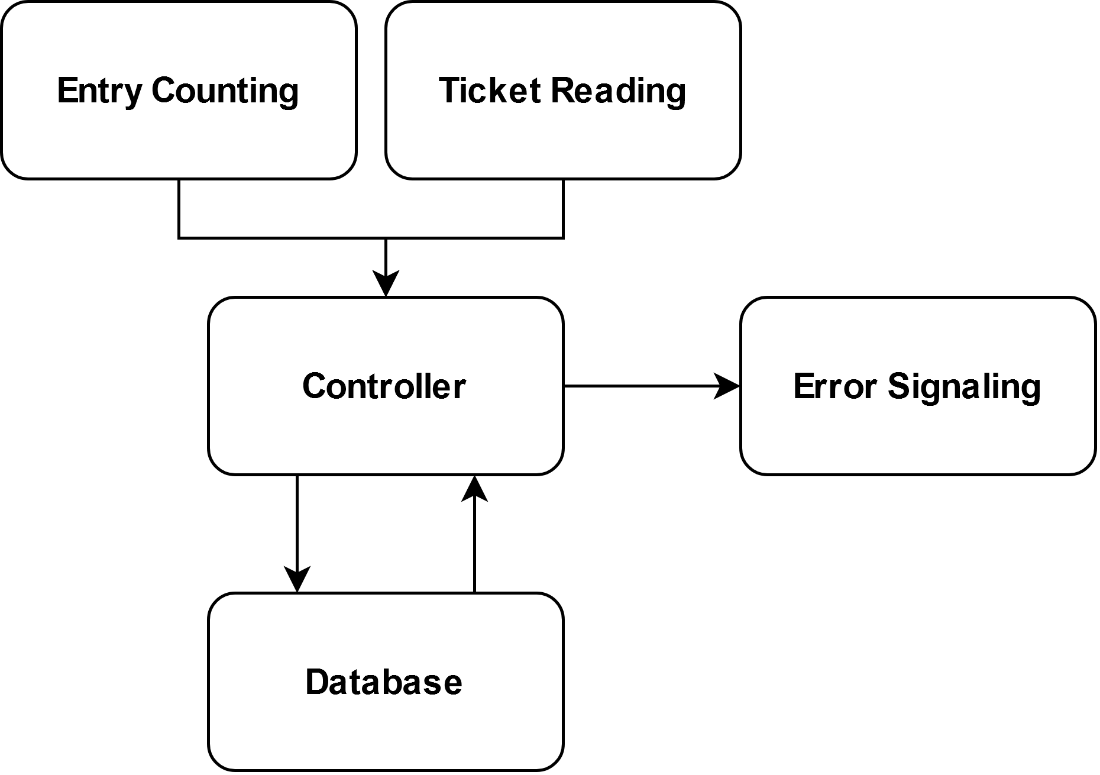


Figure 2.1: The Quicket System Subsystems

## 2.1 Hardware

The Quicket System relies on four major hardware subsections: the ticket reader subsystem, the controller subsystem, the security subsystem, and the error signaling subsystem. These subsystems are further detailed in the following sections.

**2.1.1 Ticket Reader**

The first subsystem involves the ticket reader itself, including an emphasis on the signal type, the reader model, and its antenna characteristics. This system handles detecting tags in the gate area, processing the tags’ identification data, and sending those data to the system controller.

**2.1.1.1 Signal Type**

Numerous types of signals are used throughout daily life, and several of them already handle ticket processing or hold such potential. After initial research, the design team selected four signal types, shown in Table 2.1, for consideration in The Quicket System.

Table 2.1 - Signal Types

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Technology | Requirements | UHF RFID | Bluetooth | NFC | UWB |
| 2 Meter Range | Yes | Yes | Yes | No | Yes |
| Active/Passive | Passive | Passive/Active | Active | Passive | Active |
| Tag Handling | Multiple | Multiple | Multiple | Single | Multiple |

The most important requirement comes from a sufficient read range for the system. In order to comply with ADA required clearances, the reader must be installed a minimum of 80 inches above the ground. With the assumption that tags will not be located less than 0.3 meters (1 foot) from the floor, the design team specified a 2 meter (78.74 inches) read range as one of constrains of The Quicket System. This range enables to provide adequate coverage while meeting industry standards.

RFID is a common technology based off the principle of a reader that interacts with tags over a variety of radio wave frequencies. These frequencies are broken into three main categories: low frequency (30 KHz to 300 KHz), high frequency (300 MHz to 30MHz), and ultra-high frequency (300 MH to 3 GHz) [6]. The general idea is that a lower frequency means smaller read distance and slower data transfer but less interference from metals and liquids. The crucial differences in these separate categories stem from the available read range. The low frequency signals offer the shortest read range, almost exclusively under 10 cm. This range increases to a full meter with high frequency RFID. However, ultra-high frequency signals greatly surpass the other two types in read range. These signals routinely create maximum read ranges from 10 meters up to 100 meters when paired with the proper tags and antennas [6]. Given the differences between the three categories, the design team focused on ultra-high frequency RFID options.

Within UHF RFID systems, two major types of tags exist. Passive tags use the energy transmitted by the reader to power their own internals and radiate a return signal. Active tags contain a built-in power supply that emits its own signal for the reader to pick up. Active tags contribute to a much longer read range than passive tags, but they are more expensive to produce and have a shelf life dependent on their power source. Additionally, passive tags still offer a range that easily covers the required 2-meter range of the system. As such, passive UHF tags offer the best solution from RFID options for The Quicket System.

Bluetooth signals mainly focus on linking communications between two devices. This idea could be applied for ticketing measures, especially when paired with a Bluetooth beacon. Bluetooth beacons constantly emit signals picked up by Bluetooth compatible devices, such as smartphones [7]. These signals contain “push” data that prompts the user of the device to allow a certain action. A possible implementation for ticketing involves the user storing tickets on the device. As the device nears a Bluetooth beacon, the user receives a prompt to “activate” the valid ticket. A system of beacons triangulates the device and determines when it passes through the gate. However, this design heavily relies on user input and requires app development.

NFC technology is already prevalent for ticketing as well as for a variety of other uses. An NFC device focuses on linking compatible devices together similarly to a Bluetooth pairing but provides a much more secure connection. This technology also uses a passive approach based on using magnetic induction to supply power to chips. However, this secure and passive connection comes at the expense of the signal range. The maximum range of traditional NFC devices is only around 0.1 meters [8].

Ultra-wide band (UWB) technology is another option that has seen increased popularity over the past few years, even being integrated into the newest models of Apple’s iPhones. UWB operates similarly to the other options by transmitting and decoding pulses of radio waves. The two biggest advantages of a UWB-based system are resistance to interference and accurate distance measuring. These measuring capabilities are due to UWB’s “Time of Flight (ToF)” basis, a timing system that records how long it takes to send and receive signals from one device to another [9]. However, these tags and devices require active components dependent on power supplies.

Given the current ticketing options available on the market, the design team agreed to focus on two main attributes for The Quicket System: multi-ticket processing and a passive approach that removes user dependability. Both ideas offer opportunities to increase efficiency and throughput compared to existing systems. As such, Bluetooth is an inappropriate signal base for the design. Furthermore, NFC ticketing kiosks already exist, but this technology does not currently enable a read range to cover the needs of the system. Therefore, the final decision lies between UHF RFID and UWB signals. While both meet the system requirements, a significant difference exists between the options and price range of UHF RFID tags compared to UWB tags. While the distance measuring aspect of UWB could be impactful for error detection, it does not outweigh the system flexibility available through UHF RFID.

**2.1.1.2 RFID Reader**

Given the information above, the design team researched various UHF RFID readers. The design team did not consider readers over $300 as they are over budget and overly equipped for The Quicket System’s needs. The design team narrowed down options to the three readers in Table 2.2.

Table 2.2 - RFID Reader Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Requirements | Fonkan FM-503 [10] | Sparkfun M6E Nano [11] | LPSECURITY UHF-105 [12] |
| Price | < $300 | $85 | $225 | $92 |
| Maximum Range | ≥ 2 m | 4 m | 4.9 m | 7 m |
| RF Output Power | < 30 dBm | 0-25 dBm | 0-27 dBm | 0-30 dBm |
| Antenna | N/A | Yes | Yes | Yes |
| Gain/Polarization | Circular | 5.5 dBi, Circular | Not specified | 8 dBi, circular |
| Range with Antenna | ≥ 2 m | 3 m | 0.6 m | 5 m |

The Fonkan FM-503 reader offers a middle-ground option, supplying an ample range at a price within the design team’s budget. Additionally, it gives the ability to adjust output power and comes with an antenna that provides enough range for the system requirements. Overall, this reader meets all of The Quicket System’s constraints at a reasonable price.

The Sparkfun M6E Nano offers the best processing power at the highest price point of the three options. The read range with the integrated antenna does not meet the design requirements, which would require an additional antenna. Given the steep price point and the need for an additional antenna, this reader was not chosen.

The LPSECURITY UHF-105 offers the strongest signal of the three options and meets the system requirements. However, too much signal strength causes problems within the system as the reader begins to detect tickets outside the proper range, thus increasing the likelihood of false error reports. This signal strength also nears the FCC radiated power limits of 1 W without offering many benefits. As such, the reader is not a good fit for the system.

**2.1.2 Antenna**

A crucial component for the ticketing system is the RFID antenna. The antenna radiates the signal supplied by the reader and picks up the backscatter emitted from the UHF tags. One of the largest impacts from the antenna depends on its polarization. Two major types of polarization exist: linear and circular. Linearly polarized antennas produce signal waves that oscillate within one plane. Signals from circularly polarized antennas oscillate in two, perpendicular planes. Linearly polarized antennas typically possess longer ranges but are more susceptible to tag orientation problems. Circularly polarized antennas show shorter read ranges but compensate by allowing higher degrees of error in tag orientation.

Other important antenna characteristics are radiation pattern and beam width. Combined with polarization, these traits greatly affect the effective zone of the antenna. Too narrow of a beam width misses tags that pass on the edge of coverage while too large of a beam width activates excessive or improper tags. The FM-503 antenna is circularly polarized and possesses an effective radiation pattern and beam width. As such, the supplied antenna sufficiently covers the required read range while providing proper margin of error for tag orientation and location within its coverage area.

**2.1.3 System Controller**

The Quicket System must communicate and process information relatively quickly to meet technical design constraints. The design team considered three options for the system controller listed in Table 2.3 below.

Table 2.3 - Processor Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Specification | Requirements | Arduino Uno [13] | BeagleBone Black [14] | Raspberry Pi 4 B [15] |
| Price | < $70 | $23 | $63.48 | $35 |
| Processor | 1.5 GHz | 16 MHz, 8-bit | 1 GHz, 32-bit | 1.5 GHz, 64-bit |
| RAM | 4 GB | 2 KB | 512 MB | 4 GB |
| GPIO Headers | Yes | Yes | Yes | Yes |
| Networking | Wi-Fi/Ethernet | None | Ethernet | Wi-Fi, Ethernet |
| Display | 1 port | None | HDMI | 2x Micro HDMI, Camera and Display MIPI ports |

Arduino boards, while less expensive than the latest Raspberry Pi, lack some hardware specifications that make the Arduino less efficient within the scope of The Quicket System. Arduinos used for RFID scanning would be limited in networking capabilities and processing speeds. An Arduino uses an 8-bit microcontroller that better suits larger amounts of control I/O [13]. While the board can handle larger currents, these currents are not necessary for The Quicket System’s applications. Most importantly, the Arduino only has 2 kilobytes of RAM available, making it more of a small-scale controller than a multi-functional computer [13].

The BeagleBone and the Raspberry Pi 4 have similar hardware specifications. The BeagleBone Black utilizes a 1 GHz, 32-bit processor and 512 MB of RAM with 4 GB of flash storage [14]. These features, along with its 3D graphics accelerator and dual 32-bit microcontrollers [14], make the BeagleBone capable of delivering similar performance to the Raspberry Pi. Though the Raspberry Pi 4 has higher clock speed, RAM, and overall output, the BeagleBone would meet The Quicket System design constraints.

The chosen computer or microprocessor for The Quicket System provides reliable network communication with a server database for tag identification. To relay this information between the server and the computer, the RAM and CPU clock speeds must be substantial. Industrial RFID applications, such as the ItemSense by Impinj, require at least 1.5 GHz CPU speeds and, at minimum, 4 GB of memory [16]. While The Quicket System does not need to scan tags as fast as the ItemSense might, the Raspberry Pi handles a slightly smaller volume of tags and then simultaneously communicates those data to a remote database. The database communication, while smaller in computational demand, is aided by a higher clock cycle on the CPU. Network communication can be challenging on other platforms as well. The Arduino Uno lacks network cards by default and needs more adaptation, unlike the Raspberry Pi, which already has on-board Wi-Fi/Ethernet connectivity.

Most applications that suit the Arduino pertain to specific subsystems or sensors that are measured or operated systematically. Applications for PLCs and simpler control logic conform better to the Arduino’s designed purpose.

Though comparable in performance, the major drawbacks of the BeagleBone are lack of community support and hardware integration. These aspects combined with the slight dip in performance make the Raspberry Pi more useful for testing and design implementation. Additionally, the BeagleBone does not offer the same out-of-the-box integration with other components chosen for the thermal imaging and RFID reader subsystems as the Raspberry Pi.

The Raspberry Pi provides more robust technical specifications. With a 64-bit processor and up to 8 GB of RAM on the latest Raspberry Pi 4 B, it can handle more processes at one time than the Arduino. With a clock speed of 1.5 GHz and a quad-core CPU [15], the Pi’s CPU surpasses the Arduino’s 16 MHz CPU by around 837% and the BeagleBone’s 1 GHz speed by around 50%. These increases in clock speed and available RAM allow the Pi to handle more demanding computational loads. The additional computational power of the Raspberry Pi allows for easier full-scale integration with complex programs needed for the database interaction and thermal camera subsystems.

The design team also considered the interfacing methods between the primary computer and the peripherals, then determined the additional hardware required to split or join cabling to the various hardware components. The chosen computer handles multiple digital and analog inputs and higher voltage outputs, as well as different protocols for the thermal camera and the RFID reader. Each of the 40 GPIO pins on the Raspberry Pi 4 allows for up to 3.3 V and a maximum current of 16 mA output [15]. Such pins can power light emitting diodes (LEDs) and control outputs, in addition to the thermal camera operation. The GPIO headers provide TX/RX pins for the RS232 DB9 connector that joins the data stream from the RFID reader to the computer. The Raspberry Pi also offers accessible display outputs for two HDMI ports and MIPI camera/display ports [15], allowing for the use of a touchscreen/monitor display for real-time troubleshooting.

The Raspberry Pi offers major advantages by working more independently of connected devices. Some hardware components for both the RFID reader/antenna subsystem and the thermal camera subsystem can be programmed in multiple languages to communicate with the Raspberry Pi, whereas the Arduino is primarily coded in C++ [17]. The added configurability of the Raspberry Pi allows for handling unforeseen errors or behaviors. The system can be adapted to suit the future problem needs with a wide variety of user-made content.

Another consideration of the controller choice was cost. The design team already owns a Raspberry Pi 4 B with appropriate power connections, which removes the cost of this product from the budget entirely. Any alternatives would reduce available funds needed for higher quality thermal cameras and UHF RFID reader systems.

Overall, the Raspberry Pi’s availability at no cost, combined with its superior hardware specifications and component compatibility, made it the best choice for The Quicket System’s primary computer.

**2.1.4 Error Signaling**

The Quicket System’s final hardware stage is the error signaling subsystem, which calls for human intervention upon any error during the ticketing process. The system operates on a pass until fail basis. The green LED signaling no problems at the gate remains on until a difference between the number of people entering and the number of valid tickets scanned is detected. Upon detecting a difference between these numbers or any other error during the scanning process, the Raspberry Pi switches the output to the red LED. The 4 possible states of the system are listed below in Table 2.4.

Table 2.4 - Error Signaling States

|  |  |
| --- | --- |
| Inputs | Outputs |
| T = P = 0 | Pass |
| T > P | Fail |
| T < P | Fail |
| T = P | Pass |

T – Tickets

P – People

The system indicates the error and calls for human intervention when the number of people entering the gate does not match the number of valid tickets scanned by the RFID reader or if a valid ticket is scanned more than once.

Two main systems were considered for error signaling: individual high-power LEDs and an LED strip. The Raspberry Pi can provide 3 V and 16 mA maximum current on pins 1 and 17, which could then be switched on and off to change the output of the system. This current easily powers a small LED but falls short as a practical error signaling method since such an LED could be overlooked or misinterpreted. High-power LEDs offer better solutions but require an external power supply and an n-channel metal oxide silicon field effect transistor (MOSFET) to control the grounds. An LED strip also requires an external power supply but would be a much larger and more noticeable indicator of a problem. Due to the pre-existing need for an external power supply to power the indicator and the opportunity for greater noticeability, the design team chose an LED strip to provide the visual status of the system.

The Raspberry Pi controls the activation of the error signaling; therefore, LED strips that were not compatible with the Raspberry Pi were not considered. The design team narrowed the options down to three LED strips that would fulfill the needs of The Quicket System and are listed in Table 2.5.

Table 2.5 - LED Strip Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Requirements | WS2812B [18] | WS2801 [19] | WS2813 [20] |
| Price | < $30 | $10.88 | $37.99 | $26.30 |
| Voltage | 5 V | 5 V | 5 V | 5 V |
| LEDs | > 30 | 60 | 160 | 150 |
| Length | >1 m | 1 m | 5 m | 5 m |
| Wattage | < 50 W | 18 W | 40 W | 45 W |
| Waterproof | IP56 | Yes | No | Yes |

An LED strip with at least 1 m in length would provide ample visibility for an error and a minimum of 30 LEDs would lower the risk of an LED burning out and making the error indication unnoticeable. Most LED strips are powered at 5 V; this requirement reduces the risk of needing a non-standard power supply. The LED strips must be IP56 compliant at minimum, this ensures the design is able to be used outdoors as per the requirements in the design constraints. A maximum cost of $30 was assessed because this was significant room for a strip with enough LEDs to provide noticeable error indication.

The WS2801 and WS2813 are only sold at the 5 m length with 160 and 150 LEDs, respectively. This length and number of LEDs called for substantial power and came with an increased cost. Additionally, the WS2801 did not have the option to meet IP56 standards and was eliminated.

Unlike the other two options, the WS2812B is sold at 1 m and 5 m lengths. The 1 m length with 60 LEDs meets the visibility needs of The Quicket System and comes at a lower cost. The reduced cost and power consumption made the WS2812B the best option for The Quicket System.

As mentioned above, the LED strip requires an external power supply. The power supply must be able to power the WS2812B, which runs off of 5 V and draws 4 A. The chosen power supply is the ALITOVE 5 V power supply adapter [21], which works with a standard wall outlet and supplies 5 V at 10 A of maximum current, which meets needs of the chosen LED strip.

**2.1.5 Thermal Camera**

The Quicket System uses two separate data streams for admitting users. This idea, in and of itself, is not unique to The Quicket System. Almost every conventional ticketing system uses a secondary “data stream,” of sorts, usually in the form of human intervention rather than a literal stream of data. The Quicket System differs in that the secondary data stream stems from a thermal camera. A visible light camera was considered as a secondary data stream, but this approach was rejected due to the increased computational strain that machine-learning based image recognition algorithms entail.

The Quicket System utilizes thermal imagery to count the number of persons attempting entry and compare that to those who have been authorized entry. If discrepancy occurs between the number of tags detected by the RFID subsystem and the number of people counted by the thermal imaging subsystem, then an unauthorized entry has occurred.

The three thermal cameras considered for their applicability to these general needs are listed and compared below in Table 2.6.

Table 2.6 - Thermal Camera Options

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Requirements | MLX90640 Thermal Camera Breakout with PIM366 [22] | MLX90640 Thermal Camera Breakout with PIM365 [23] | FLIR ONE Pro [24] |
| Price | <$100 | $60.84 | $60.84 | $399.00 (Provided by MSU) |
| Lens Angle | >70° in width | 110°x75° | 55°x75° | 55°x 45° |
| Thermal Resolution | >20x20 | 32x24 | 32x24 | 160x120 |
| Frame Rate | >30 FPS | 64 FPS | 64 FPS | 8.7 FPS |
| GPIO | Yes | Yes | Yes | No |
| Compatible with Raspberry Pi | Yes | Yes | Yes | No |

The Quicket System makes use of the MLX90640 Thermal Camera Breakout, which uses the PIM366 thermal camera. The PIM366 has a resolution of 32x24, and a wide-angle (110°x75°) lens. While this resolution is low, the MLX90640 fits the needs of The Quicket System.

The temperature detection capabilities of this camera exceed the needs of the thermal imagery processing software, at a range of –30 °C to 400 °C and an accuracy of 1 °C, all while maintaining a framerate of 64 FPS.

The MLX90640 natively supports the Raspberry Pi, utilizing the I2C protocol for communication over GPIO pins via a utility library.

The wide-angle variant (PIM366) of this camera is used so the system can detect people attempting to enter the RFID reader’s range. The standard, non-wide-angle variant (PIM365) would only detect people directly below the camera and therefore already within the range of the RFID reader.

The FLIR ONE Pro is the most powerful camera in the comparison and is provided at no cost to the design team. However, this camera does not fit the needs of The Quicket System. The FLIR ONE Pro provides no GPIO compatibility and is only operable only in conjunction with an Android smartphone. If efforts were made to make the FLIR ONE Pro compatible with a Raspberry Pi, the camera would still not meet the framerate or lens angle requirements of The Quicket System. More specifically, the FLIR ONE Pro does not have a lens angle wide enough to detect approaching people. With respect to frame rate, the FLIR ONE Pro falls short of the high-fidelity motion-tracking needs of The Quicket System. If the time difference between each frame is too high, the software may be unable to distinguish between an already existing person moving and a new person entering the thermal camera’s view.

## 2.2 Software

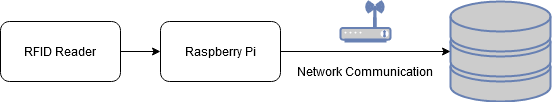
The Quicket System contains two major software subsystems. The thermal imaging subsystem sends the number of people attempting to pass through the RFID reader’s range to the database interaction subsystem. The database interaction subsystem checks the data received from the RFID scanner against both the database of valid tickets and the data received from the thermal imaging subsystem. If a discrepancy between the number of people detected and the number of tickets is found, an error is raised and reflected in the error signaling subsystem.

**2.2.1 Database Interaction**

The Quicket System uses MariaDB (a fork of MySQL) to store the IDs associated with each RFID ticket. MariaDB was chosen because it is an easily accessible, non-proprietary database software that can be hosted on a consumer computer. The latter feature simplifies database interactions and reduces the costs associated with the project. Additional benefits offered by MariaDB include the extensive documentation provided for the system, simple vertical scalability, and ability to add security to the system through data encryption [25].

PostgreSQL is an open-source, MySQL-based system with support for a wide variety of custom data types, such as JSON and XML. However, this system was eliminated because of its poor documentation, where many details are either inconsistent or missing altogether. The system also suffers from a lack of tools that show the state of the database, which would make keeping the database stable difficult [25].

MySQL was the third database management system considered for the project. This system was considered due to it being one of the most popular [25] relational database systems and working well with a wide variety of computer systems. MySQL is also easy to learn with simple syntax. However, MySQL was not chosen for The Quicket System because it does not scale easily, imposes further cost for features that may be needed, and struggles with large volumes of data.

  
Figure 2.2: Overview of Reader/Database Interaction

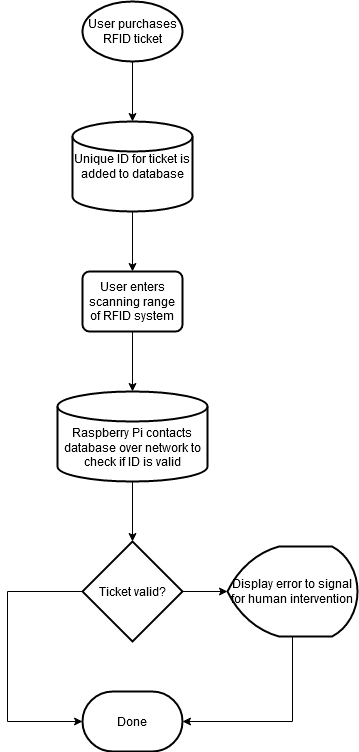


Figure 2.3: Tag Reading and ID Checking Logic Overview

When a ticket is purchased, its unique identifier is stored in the MariaDB database. This unique identifier is programmed into the RFID tag held by the end user. When the end user enters scanning range of the RFID reader in the gate, the reader sends the recorded identifier to the Raspberry Pi. The Raspberry Pi checks the identifier against the database after transmitting it over the network. If the identifier is found in the database and has not been marked as “used” for a given event, then the ticket is considered valid and no intervention is required; however, if the identifier is not found in the database or has been marked as used, the ticket is invalid. The database returns an “OK” or “not OK” flag to the Raspberry Pi. If an “OK” flag is received, then the Raspberry Pi continues supplying power to the green “OK” LED. If a “not OK” flag is received, then a signal is sent to the LED subsystem, which illuminates a red light. This red light signals the need for human intervention.

The system continues the scanning of RFID tickets even while in the “not OK” state. Therefore, communication between the Raspberry Pi and the database ceases only when the system is powered off.

**2.2.2 Thermal Imaging**

Video taken from the thermal imaging hardware is fed into a designed program that detects the number of people attempting to pass through the detection area. This program leverages several libraries in a Python environment to this end: MLX90640, a utility library for the system’s thermal camera (provided by the manufacturer); OpenCV, a computer vision and machine learning library; NumPy, a scientific computing library; and imutils, a convenience library that streamlines parts of the OpenCV and NumPy pipeline.

These libraries are used to render counts of people currently passing through and those approaching the RFID reader’s range. The count of people passing through the RFID reader’s range is then compared against the number of tags detected by the RFID reader. The count of people approaching but outside the RFID reader’s range is used later in the program and is detailed later in this section.

The first step in the process performed by the thermal imaging subsystem is image extraction and correction. Images extracted from the thermal camera’s video stream via the MLX90640 library are processed and corrected for various environmental and technical factors, such as fisheye lens distortion, sensor inaccuracies, and thermal noise. OpenCV is used to correct fisheye lens distortion, and NumPy is used for output calibration and thermal noise correction. The thermal output is calibrated such that the average human temperature is close to 37.0 °C (98.6 °F). This temperature is chosen because it serves as a dependable ground truth against which to calibrate. Finally, the matrix of temperature information is normalized using NumPy.

OpenCV’s blob detection methods are used to identify people using the image and temperature data to find the number of people. By comparing the blobs detected from one frame of video to that of the next, the velocity of each person can be detected. With this information, each person is tracked individually and is assigned a unique identifier until they leave the thermal camera’s range.

The position of each person is used to determine their distance from the RFID reader. Each image received from the camera is then mapped onto a Cartesian plane where the Y axis runs from the top of the image to the bottom. People below a calibrated offset on the Y axis of the image are considered to be currently passing through the RFID reader’s range and those above that offset are considered to be members of the next group of people to enter the RFID reader’s range. Once a group of people enters the RFID reader’s range, they are counted, and their unique identifiers are destroyed. This process in its entirety repeats on the next group of people in line.

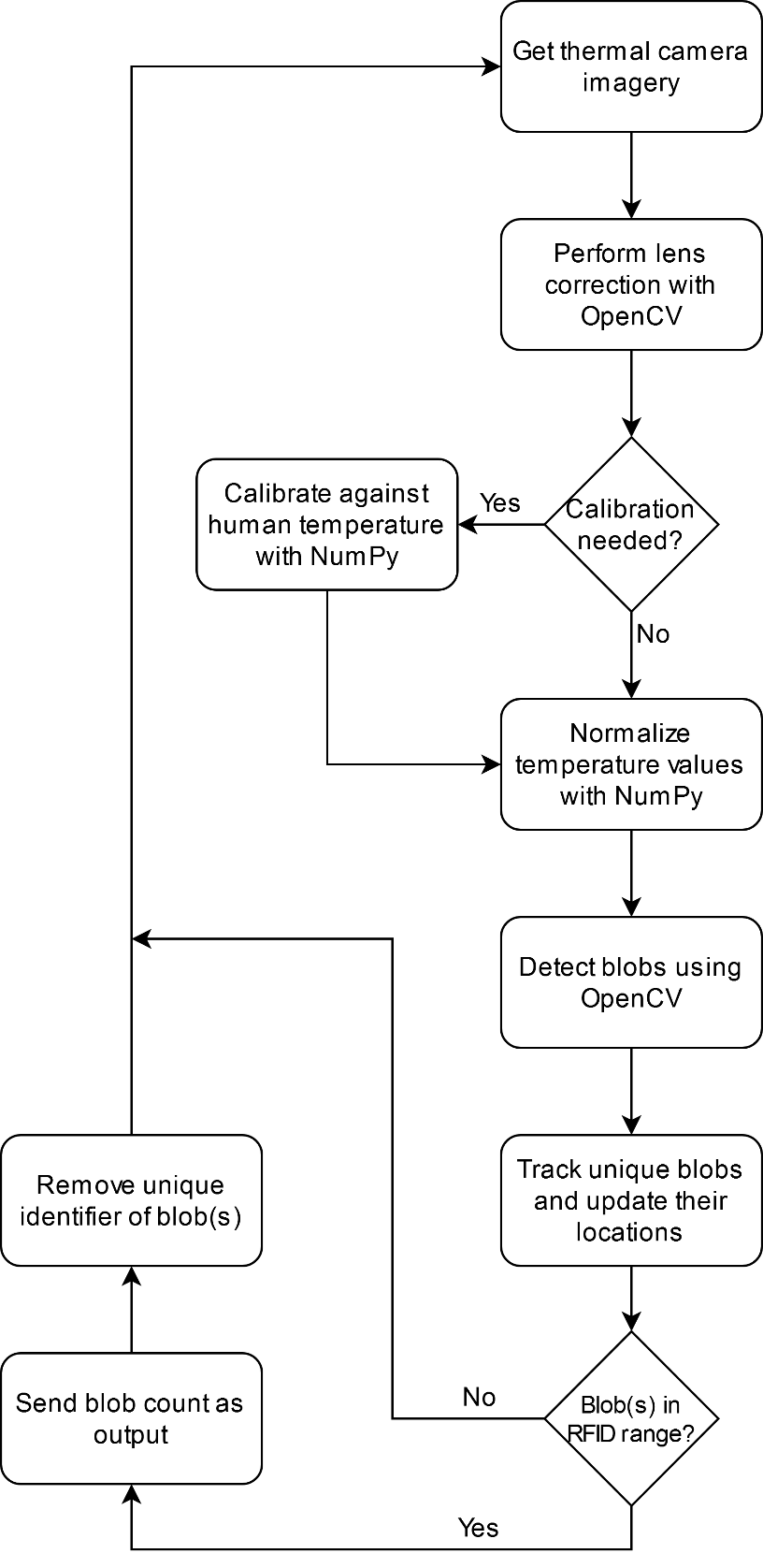


Figure 2.4: Thermal Imaging Software Process Flowchart

The data are then sent to the database interaction subsystem to be compared against data from the RFID subsystem. The number of people currently passing through the RFID reader’s range is used for error or non-ticketholder entry.

References:

[6] “What Are the Different Types of RFID Technology?” *lowrysolutions.com,* Dec. 12, 2014. [Online]. Available:<https://lowrysolutions.com/blog/what-are-the-different-types-of-rfid-technology/> [Accessed Sept. 2, 2020].

[7] “What is a Bluetooth beacon?” *beaconstac.com*, [Online]. Available: <https://www.beaconstac.com/what-is-a-bluetooth-beacon> [Accessed Sept. 12, 2020].

[8] N. Chandler, “What’s an NFC tag?” *howstuffworks.com*, [Online]. Available: <https://electronics.howstuffworks.com/nfc-tag.htm> [Accessed Sept. 13, 2020].

[9] “Ultra-Wideband (UWB),” *bleesk.com*, [Online]. Available: <https://bleesk.com/uwb.html> [Accessed Sept. 13, 2020].

[10] FM-503. [Online]. *AliExpress*. Available: <https://www.aliexpress.com/item/4000133201826.html?spm=a2g0s.9042311.0.0.27424c4dpZPuIN> [Accessed 10 Sept. 2020].

[11] Sparkfun M6E Nano. [Online]. *Sparkfun*. Available: <https://www.sparkfun.com/products/14066> [Accessed 10 Sept. 2020].

[12] UHF-105. [Online]. *AliExpress*. Available: <https://www.aliexpress.com/item/32325789868.html?spm=2114.12057483.0.0.49ca74d4EMgOvq> [Accessed 12 Sept. 2020].

[13] Arduino Uno Rev3. [Online]. *Arduino*. Available: <https://store.arduino.cc/usa/arduino-uno-rev3#:~:text=Arduino%20Uno%20is%20a%20microcontroller,header%20and%20a%20reset%20button>. [Accessed 14 Sept. 2020].

[14] BeagleBone Black. [Online]. *BeagleBoard*. Available: http://beagleboard.org/black [Accessed 17 Sept. 2020].

[15] Raspberry Pi 4 B. [Online]. *RaspberryPi*. Available: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/ [Accessed 14 Sept. 2020].

[16] “Server Hardware Requirements,” *impinj.com,* [Online]. Available: https://platform.impinj.com/site/docs/itemsense/requirements/index.gsp

[17] F. Copes, “Arduino vs Raspberry Pi,” *flaviocopes.com*, [Online]. Available: https://flaviocopes.com/arduino-vs-raspberry-pi/#:~:text=The%20Raspberry%20Pi%20comes%20with%20a%2064%2Dbit%20microprocessor.&text=In%20terms%20of%20I%2FO,set%20of%20I%2FO%20pins.&text=Raspberry%20Pi%20runs%20an%20operating,Arduino%20is%20much%20more%20simple. [Accessed 15 Sept. 2020].

[18] RGBW 5VDC LED Strip. [Online]. *BTF-*Lighting. Available: https://www.btf-lighting.com/products/1-sk6812-rgbw-4-in-1-pixels-individual-addressable-led-strip-dc5v?variant=25757793812580

[19] HKBAYI 5-meter RGB LED Strip Light [Online]. *HKBAYI.* Available: https://www.amazon.com/HKBAYI-WS2801-addressable-32Leds-non-Waterproof/dp/B00JY2P672

[20] WS2813 Waterproof LED Strip. [Online]. *BTF-Lighting.* Available: https://www.btf-lighting.com/products/ws2813-dc5v-led-pixel-strip-dual-signal?variant=25780951482468

[21] 5V/10A Power Supply. [Online] *ALITOVE.* Available: https://www.alitove.net/product/5v-10a-bk/

[22] PIM366. [Online]. *Digi-Key*. Available: https://www.digikey.com/product-detail/en/pimoroni-ltd/PIM366/1778-1232-ND/9606192 [Accessed 10 Sept. 2020].

[23] PIM365. [Online]. *Digi-Key*. Available: https://www.digikey.com/product-detail/en/pimoroni-ltd/PIM365/1778-1231-ND/9606191 [Accessed 10 Sept. 2020].

[24] FLIR ONE Pro. [Online]. *FLIR*. Available: https://www.flir.com/products/flir-one-pro [Accessed 10 Sept. 2020].

[25] “Comparing Database Management Systems: MySQL, PostgreSQL, MSSQL Server, MongoDB, Elasticsearch and others,” altexsoft.com, Jun. 20, 2019. [Online].Available: https://www.altexsoft.com/blog/business/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/ [Accessed 15 Sept. 2020].