SCAN: Smart Check-In and Analytics Network

Cory Brynds, Jonah Sprandel, and DeAndre Hendrix

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida 32816-2450

Abstract - At a large university like UCF, data collection is essential. SCAN, the Smart Check-in and Analytics Network, provides a network of check-in kiosks that can be used by the UCF Recreation and Wellness Center (RWC) and other facilities/events to collect user engagement data. SCAN presents users with an effortless check-in experience and supplies facilities administrators with analytics data. With an occupancy tracking system in place, the RWC could transform its daily operations. Staff could predict peak hours precisely, manage staffing levels throughout the week, and gather demographic data on the types of students utilizing the facility. By tracking this engagement data, the attendee experience is also enhanced by ensuring that resources are allocated when and where they're needed most. Overall, SCAN provides a product marketed towards facilities and event administrators to efficiently handle check-ins and gather real-time analytics.

I. INTRODUCTION

This section introduces the SCAN project, outlining its motivation, defining its scope, and providing a high-level overview of its technical components.

A. Project Motivation

At a large public university like UCF, data collection is not just important—it is essential. Graduation rates, enrollment statistics, and faculty-to-student ratios are carefully tracked each year and reported to institutions for accreditation and ranking. Another measurement often overlooked is engagement: How many students attend major university events? How long are they staying? What are their activity patterns week-to-week throughout the academic year? The answers to these questions directly influence key decisions on resource allocation, staffing, and campus planning. Without access to accurate, accessible engagement data, the university risks missing insights that could impact its future.

UCF students, faculty, and staff already use their UCF ID cards—embedded with a combination of magnetic stripe and RFID technology—for many daily functions, whether it's scheduling appointments at Student Health Services, accessing restricted buildings, or checking into locations like the UCF Recreation and Wellness Center (RWC). Due to their widespread use, these cards present an overlooked opportunity to streamline data collection and capture valuable insights in real time.

The UCF RWC is a prime example. With an occupancy tracking system in place, the RWC could transform and adjust its daily operations. Staff could predict peak hours precisely, manage staffing levels throughout the week, and even gather detailed demographic data on the types of students utilizing the facility. By tracking this engagement data, not only is the efficiency of the RWC's operations enhanced, but the experiences of individual attendees are improved by ensuring that resources are allocated exactly when and where they're needed most.

B. Project Overview

We present SCAN: a Smart Check-in and Analytics Network capable of tracking and analyzing user data. We have designed a low-cost check-in kiosk equipped with an NFC transceiver and interactive display to log check-ins and check-outs of users. Multiple kiosks can operate simultaneously to process authentications at multiple facilities or provide multiple access points at a single facility. Encoded NFC cards emulate UCFIDs to avoid any security concerns involving the use of actual IDs, and each kiosk includes a keypad for manual ID entry as a back-up authentication method. User data is stored in a secure database external to the kiosk, and the kiosk queries the database to ensure that the user has proper permissions to access the area. In addition, a web application displays analytics based on the check-in data, which is made available to administrators and students.

C. Technical Components

We divide the technical components of the SCAN project into three key areas, which will be overviewed in the following three sections.

1). Hardware Design: The hardware design section focuses on the design of the SCAN PCB, the hardware interfaces between our MCU and peripheral devices, and the topology of our power management system.

2). Embedded Firmware Design: This section overviews the embedded software development for the SCAN system in ESP-IDF. It includes implementation details of the primary state machine used to control the kiosk program flow and our interface with the database for user authentication. Taking advantage of ESP-IDF's native FreeRTOS support, we describe our approach to task scheduling. Finally, we overview the use of over-the-air updates to deploy new firmware patches directly over the network.

3). Web Stack Software Design: The web stack design section unveils the selection and implementation details for the frontend, backend, and database technologies used for creating our web app ecosystem.

Table 1: SCAN Requirement Specifications

No.	Requirement	Specification
1	Maximum size of the kiosk enclosure and stand	< 1’ x 1’ x 4’
2	Number of supported user authentication methods	2 methods
3	Minimum continuous operating time on battery power under typical operating conditions	> 2 hours
4	Minimum supported charging power of the kiosk	> 10W
5	Maximum proximity sensor detection range	> 1.5 meters
6	Number of check-in analytics that SCAN will display based on check-in data	> 3
7	Maximum time to authenticate and check in/out a user	< 10 seconds
8	Accuracy of user authentication	> 95%
9	Rate at which SCAN’s collected data will be analyzed and updated on the webpage	< 1 minute

II. ENGINEERING SPECIFICATIONS

This section provides an overview of the requirement specifications we have used to guide the design of the SCAN system, summarized in Table 1. This table includes specifications for enclosure size, power, and software responsiveness. We have selected three of these specifications to demonstrate in our final project demo: time to authenticate, authentication accuracy, and analytics refresh rate.

A. Time to Authenticate

This specification refers to the time from when a user taps their ID card or presses the enter key on the keypad to the time SCAN’s database returns a check-in decision. This amounts to the time required for an HTTP GET request to retrieve user info from the database and a follow-up HTTP PUT request to update their status to be checked in or out. To test the time-to-authentication, we used the ESP32’s built-in timer to record the duration between an ID being entered and the database returning a response following the HTTP PUT request and recorded the results of 20 trials with user IDs of varying credentials.

B. Authentication Accuracy

This specification refers to the kiosk’s ability to produce the desired check-in, check-out, or validation failed response based on the credentials of the tapped or entered user ID. Situations in which this accuracy would be less than 100% involve the NFC transceiver incorrectly parsing the data on a user ID card, a failed HTTP request to the database, or an incorrect check-in decision being made based on a user’s credentials or check-in status. To test this specification, we applied stimulus to the kiosk using different check-in methods and users of various credentials (valid user, active student, current check-in status). The required authentication accuracy is 95% (< 5% error rate).

C. Analytics Refresh Rate

Our final engineering requirement specification to be demonstrated sets an upper bound of one minute on the amount of delay time between new data being recorded, such as new data from a check-in or check-out event at a kiosk, and refreshed analytics like occupancy and average stay duration being displayed on SCAN’s web app dashboard. To ensure our system met this specification, we conducted 10 trials of measuring the elapsed time between a user swiping their keycard to check in and refreshed analytics being seen on the analytics dashboard.

III. SYSTEM COMPONENTS

The SCAN system consists of a carefully-selected set of hardware components to ensure reliability, efficiency, and seamless user interaction.

A. ESP32-C6

For the primary SoC interfacing all of kiosk peripherals and communicating with the SCAN database, we selected the ESP32-C6 SoC, which is a dual-core, 32-bit RISC-V SoC supporting 2.4 GHz Wi-Fi 6, Bluetooth 5, and Zigbee 3.0. Its low-power modes make it ideal for applications that require intermittent operation, such as the SCAN kiosk. The SoC we selected to integrate into our PCB, the ESP32-C6-WROOM-1-N8, has 23 GPIOs, 8MB of flash memory. and an on-board RF antenna. [1] For our purposes, the ESP32-C6’s affordability, large flash memory size, wide array of GPIO pins, and support for 2.4GHz WiFi make it perfect for our project. We can also utilize the ESP32-C6’s HP/LP dual-core architecture for efficient task scheduling using FreeRTOS.

B. Near-Field Communication Transceiver

Near-field communication, or NFC, is a subset of RFID communication specifically for short-range, high-frequency wireless communication between devices. It is used in applications like tap-to-pay, data exchange, and—in our case—access control and user authentication. To enable the reading of our NFC user ID “smart cards”, we selected the PN532 RFID transceiver. This is a highly versatile transceiver with a built-in 80C51 microcontroller for handling the PN532’s many operating modes. We use the PN532 in ISO/IEC 14443A/MIFARE Reader/Writer mode, which allows it to read and write to NFC forum smart cards. It supports data transfer speeds of up to 424 kbit/s and supports SPI, I2C, and HSU, of which we choose SPI to interface with our MCU. [2] The PN532 has a rich amount of documentation and library support on the

ESP-IDF platform, which was our primary reason for selecting this transceiver for our project.

C. Keypad Matrix

As a back-up authentication method to the NFC transceiver, the SCAN kiosk includes a keypad for manual entry of a user's ID. The keypad can also assist with administrative tasks such as writing a new ID card. We selected a 4x4 membrane keypad with a molded plastic enclosure for its tactility and aesthetic design. To avoid using eight of our MCU's GPIO pins in order to address the four rows and four columns of the keypad, we selected the PCF8574, which is a I2C IO expander for up to 8 GPIO pins. The PCF8574 has an interrupt pin and a sourcing current of 25mA. By using this I2C expander, we free 6 additional GPIO pins on our MCU and can share the I2C bus with SCAN's other I2C devices.

D. Passive Infrared Sensor

For creating an efficient and low-power design, a triggering mechanism is necessary to transition our kiosk from low power mode to active operation. This function is achieved using a proximity sensor. Passive infrared sensors (PIRs) detect motion by measuring changes in infrared radiation emitted by objects in the sensor's detected field. The AM312 PIR was selected for this project due to its extremely low operating current of 0.1mA, quick response time, and a detection range of 3-5m, meeting our requirement specification.

E. GC9A01 TFT Display

As the primary method of conveying information to the user throughout the check-in process, we selected a round 1.28-inch thin-film transistor (TFT) display. Specifically, we chose a 240x240 display featuring the GC9A01 graphics driver IC. The display communicates with our ESP32-C6 through an SPI interface, requiring five GPIOs for SCLK, MOSI, chip select, data control, and reset (MISO pin is not needed). Overall, the display's ease of programmability, low operating current of 20mA, and simple interface made it an ideal choice for our project.

F. User Interface Feedback Devices

Additionally, we selected the WS2812B addressable RGB LED and a 3V, 16Ω, 2KHz passive buzzer to integrate more status indicators into the kiosk's design. Each WS2812B LED integrates an RGB controller IC directly into the package, and these LEDs can be chained together to address the color and brightness of all LEDs with a single GPIO. Each LED operates at 5V and draws approximately 60mA at maximum brightness. The selected passive buzzer provides an effective auditory feedback system for different states of the check-in kiosk, as the kiosk firmware can control the frequency, duration, and pattern of the buzzer's output through PWM. By associating unique LED colors and sound patterns with kiosk states, such as initialization, check-in decisions, and error notifications, these two components enhance the SCAN kiosk's user interface.

G. Power Management Controller

To manage the kiosk's power delivery and battery charging, we selected the BQ25798 controller, a highly integrated buck-boost converter. It is designed to support USB, DC, and solar power inputs, making it ideal for the vast use cases of the kiosk. It features an autonomous battery charging routine, programmable settings through an I²C interface, narrow voltage DC power path management, making it ideal for our kiosk's seamless operation in different scenarios. The device supports battery chemistries including Li-ion and lithium iron phosphate up to 5A charge current, ensuring fast and efficient charging.

H. USB Power Delivery Controller

To enable faster battery charging and more efficient power operation, we selected the CYPD3177, a programmable USB Power Delivery (PD) controller from Infineon. The CYPD3177 is a single-port, sink-only PD controller, fulfilling the kiosk's battery charging and power supply needs. This controller enables the kiosk to negotiate power delivery up to 60 W, ensuring sufficient power for the ESP32C6, peripherals, and battery charging. Most importantly, the CYPD3177 acts as the bridge between the industry standard power cable (USB-C) and the SCAN kiosk. Using I²C and built-in hardware interrupt functionality, the PD controller offers significant customizability without requiring external firmware intervention. Additionally, integrated overvoltage, undervoltage, and overcurrent protection mechanisms help safeguard the system from power faults.

I. Talentcell 12V 6Ah $LiFePO_4$ Battery

For the SCAN kiosk, selecting a reliable and efficient battery is essential to ensure consistent operation in various environments. After careful consideration, we chose the Talentcell 12V 6Ah $LiFePO_4$ battery. Lithium iron phosphate ($LiFePO_4$) technology is well-suited for our application due to its high safety, long lifespan, and excellent energy density. Compared to other lithium-based chemistries, $LiFePO_4$ batteries are more stable, resistant to thermal runaway, and capable of deep discharge cycles without significant degradation. The Talentcell battery meets our key requirements, maintaining an operating voltage above 7V to support our power converters efficiently. It is compact enough to fit within the kiosk stand while offering ample capacity for prolonged usage. Additionally, its safety features make it suitable for deployment in public spaces, ensuring reliable performance without the risks associated with other battery technologies.

IV. HARDWARE DESIGN

A. *Hardware Architecture and Block Diagram*

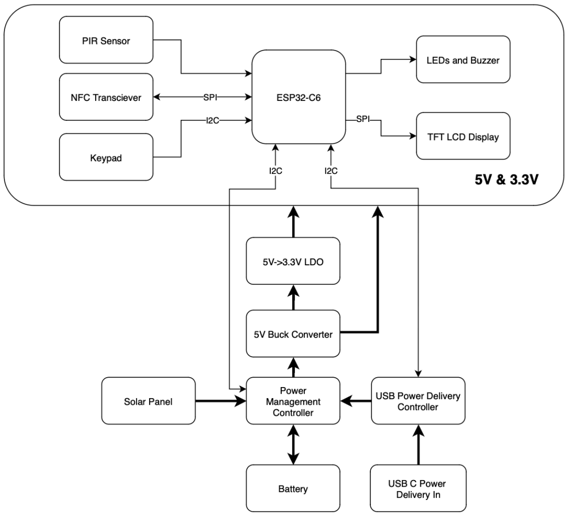

Figure 1: Hardware Block Diagram

The hardware block diagram in Figure 1 provides a high-level representation of the system's architecture, detailing how components are interconnected. It illustrates the power flow from USB-C Power Delivery, battery backup, and an optional solar panel, all regulated through a power management controller. The diagram also highlights communication interfaces between the ESP32-C6 and peripherals such as the NFC transceiver and keypad.

B. *Power Management Design*

The SCAN kiosk's power system is designed to manage multiple power sources while ensuring reliable operation in various environments. At the center of the power management design is the BQ25798, which dynamically regulates power input and output from three potential sources: USB-C (regulated by the CYPD3177), the 12V $LiFePO_4$ battery, and a solar panel. When an external USB-C source is available and capable of supplying sufficient power, the BQ25798 automatically charges the battery while simultaneously powering the system. If USB-C power is unavailable, such as in outdoor environments, then the battery seamlessly takes over to keep the kiosk running without interruption.

The system output power from the BQ25798 is then converted into a stable 5V rail using a TPS563203 buck converter, which supplies power to the USB-UART bridge and RGB LEDs. To provide power for the rest of the components, the 5V rail is further stepped down to 3.3V using a TLV76733 linear voltage regulator, ensuring stable operation of the kiosk.

C. *PCB Layout and Design*

The system is implemented on a single six-layer PCB, chosen to optimize power integrity, signal routing, and noise reduction. By using multiple layers, the design effectively reduces electromagnetic interference (EMI) and improves overall system stability. The PCB layer stack-up is organized as follows:

Layer 1 (Top) - Signal: High-speed signal traces, critical component placements, and routing for key peripherals
Layer 2 - Ground Plane: A solid ground plane to provide a stable reference for signals and reduce EMI
Layer 3 - 3.3V Power Plane: Power plane for the 3.3V logic rail, delivering consistent power to all peripherals
Layer 4 - Signal: Additional signal routing layer primarily to separate critical and less important traces
Layer 5 - Ground Plane: Another solid ground plane to improve signal integrity and reduce noise
Layer 6 (Bottom) - Signal: Lower-priority signal traces and passive component placements

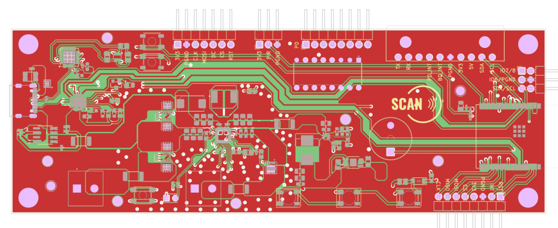

Figure 2: Layer 1 (red) and Ground Plane (green)

Figure 2 shows the primary signal layer, where most of the important traces and critical component placements are routed. This layer demonstrates the design style for all layers. For example, critical signals like the switching signals for the power converters are carefully routed with minimal vias. Large copper pours are carved out to carry power for circuits like the USB Power Delivery and battery charger. High frequency components, specifically like the BQ25798, are routed with the shortest trace lengths possible, following the layout guidelines from the datasheets.

D. Kiosk Enclosure Design

The requirement specification for the enclosure size puts the kiosk at 1'x1' wide and 4' tall. To meet these requirements, the enclosure is designed with a combination of 3D printing and other acquired materials. The base and side panels are made of cut PVC panels, and the stand is a cut 1.25" PVC pipe. The kiosk enclosure, sitting at the top of the stand, is 3D-printed PLA plastic and houses the kiosk peripherals, PCB, battery, and solar panel. The enclosure is positioned at an accessible height for card and keypad entry, around torso height for the average person. Overall, the kiosk is intentionally designed to be minimalistic and blend into its surroundings.

V. EMBEDDED SOFTWARE DESIGN

SCAN kiosk's embedded software was developed in ESP-IDF, the official C-based development framework for the ESP32. The software is structured around a finite state machine (FSM) running within a FreeRTOS task to manage system states, user authentication, and database interactions. Additional features such as over-the-air (OTA) updates and real-time database communication enhance system maintainability and reliability.

A. Control Logic

The primary control logic for the SCAN embedded firmware is in the form of a finite state machine (FSM) managed by the FreeRTOS state_control_task(), shown in Figure 3, which continuously evaluates the system state. On reset, following the initialization functions for each peripheral, the FSM will begin in STATE_WIFI_INIT by establishing a connection to the specified WiFi network, waiting for a semaphore that will signal an established WiFi connection and transition to STATE_WIFI_READY. After cleaning up the WiFi init task, the system will transition to STATE_IDLE, where it remains in a light sleep mode until proximity is detected by an interrupt attached to the PIR sensor. Upon detecting a user, the kiosk will awaken and accept user ID input via the NFC transceiver and keypad in STATE_USER_DETECTED.

When a user ID is successful read from the NFC transceiver or an ID of valid length is entered on the keypad (whichever comes first), the state machine will transition to STATE_DATABASE_VALIDATION, where the kiosk will retrieve and authenticate the user's credentials against the database. Based on the response from the database, the next state will either be a check-in decision, validation error, or the admin mode state. A check-in decision will transition to STATE_CHECK_IN or STATE_CHECK_OUT, where it will remain for a couple seconds, long enough to display a check-in status message to the screen, before transitioning back to STATE_IDLE. If the ID queried in the database does not exist, the FSM will transition to STATE_VALIDATION_FAILURE, where it will prompt the user to try again and go back to STATE_USER_DETECTED.

The purpose of kiosk administrator mode is to allow the administrator to write new ID cards for existing users in the database with valid credentials. The state machine will enter STATE_ADMIN_MODE if the user returned from the database has a role of "admin" instead of "student". From there, a second state machine running in another FreeRTOS task, admin_mode_control_task(), will be signaled by a task notification and transition from STATE_IDLE to STATE_ENTER_ID. This secondary state machine can be seen in Figure 4. First, in STATE_ENTER_ID, the admin will enter the ID of an existing user in the database. Upon entry of a valid length ID, the kiosk will query the database and determine whether the ID belongs to a valid user in STATE_VALIDATE_ID. If the user is not valid, it will inform the admin in STATE_ENTER_ID_ERROR before returning to STATE_ENTER_ID for another attempt.

If the user ID is valid, it will transition to STATE_TAP_CARD, where the admin will place a blank ID card on the NFC transceiver and the kiosk will format and write the previously-entered user ID onto the card. Upon a successful card write, the admin state machine will transition to STATE_CARD_WRITE_SUCCESS, notify

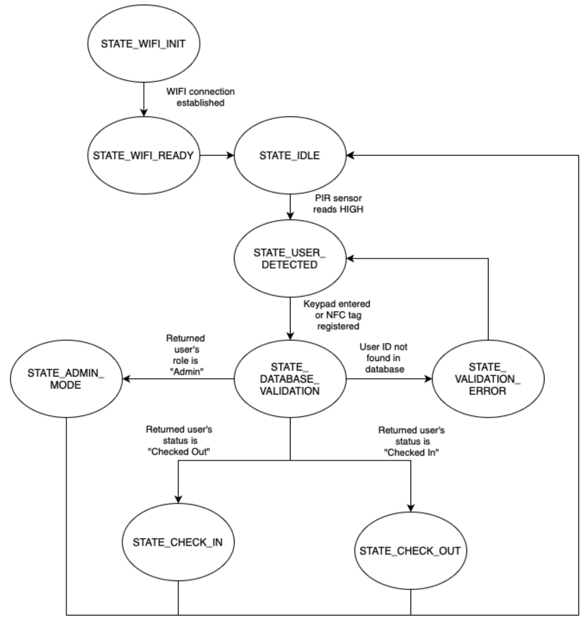

Figure 3: State Control Task FSM

the admin, and transition back to STATE_IDLE until the next time the main state machine enters admin mode. At this point, the main state machine will transition from STATE_ADMIN_MODE back to STATE_IDLE for regular operation.

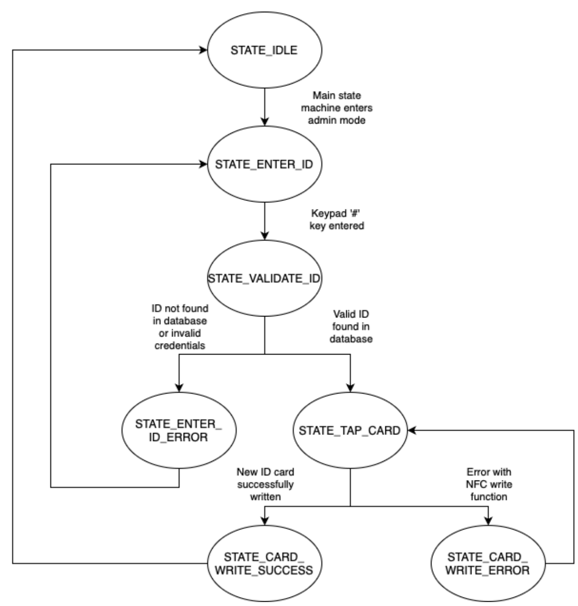

Figure 4: Admin Mode Control Task FSM

B. Database Interface

For interacting with the database via firmware, three simple database API functions were created using ESP-IDF HTTP handling libraries and example code including: getUserInfo(userID), checkInUser(userID), and checkOutUser(userID).

The first function getUserInfo(userID) is called within the control flow once a userID is entered on a kiosk either via keypad or NFC tag. If the user ID is valid, the function retrieves the user's information stored in the database as a struct. Otherwise, the function flags the state machine of an invalid userID and returns for proper error handling.

The checkInUser(userID) and checkOutUser(userID) functions provide the main control flow with high level check-in and check-out API calls that abstract away HTTP handling. The program flow invokes either of these functions once a user is validated using the getUserInfo(userID) function.

C. NFC Software Design

To interface the NFC transceiver and ESP32 via software, we utilized an ESP-IDF component library supporting an SPI interface with the PN532. This library provides the low-level protocols necessary to read and write raw bytes of data to the NFC cards but does not handle writing formatted text such as ID strings. To handle reading and writing IDs, we followed the NFC Data Exchange Format (NDEF), a widely used data form for NFC devices, which governs the structure of data stored on tags such as the NTAG213 smart card used for this project.

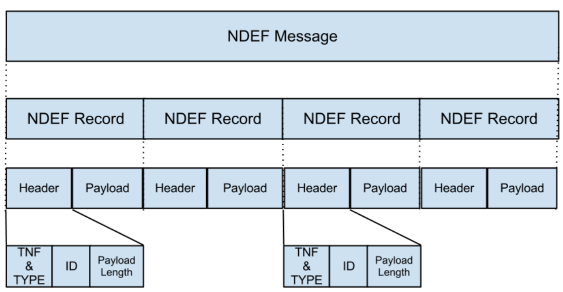

Figure 5: NDEF Message Structure

On each ID card, we write one NDEF message to block 2 of the NTAG213 smart card, with one NDEF record whose header follows the structure of an NDEF text record.

D. GUI Design

The SCAN kiosk's graphical user interface was designed using the ESP-IDF version of the Light and Versatile Graphics Library (LVGL), a popular open-source embedded graphics library that can be used to develop responsive UIs for MCU-driven displays. For SCAN, we developed display screens that correspond to different check-in states, such as STATE_USER_DETECTED, which prompts the users to tap their card or enter their ID on the keypad, or STATE_CHECK_IN_SUCCESS, which displays a check-in message to the user. At reset, as part of the initialization phase, the program will load all of the screens into SRAM. Then, upon transition to a new state, the program will load the corresponding screen to the display. The current screen on the display is periodically refreshed by a FreeRTOS task running in the background.

E. Over-the-Air Updates

Over-the-air (OTA) updates are a key feature for the SCAN kiosk, enabling seamless firmware updates without requiring physical connections or manual intervention by end-users. Kiosk administrators begin the process by selecting the desired kiosk software version on the SCAN website. This publishes an MQTT message with the software version number via HiveMQ, the MQTT broker. The ESP32 uses the built-in MQTT library to listen for these messages. When a message is received, the OTA update process is automatically triggered. The OTA system is implemented using ESP-IDF's built-in OTA library, which securely downloads new firmware from the GitHub server over WiFi. The OTA update firmware downloads the new binary into a secondary flash partition on the ESP32, ensuring a fail-safe mechanism in case of update failures. Once the update is verified, the system reboots and switches to the new firmware. By integrating OTA updates, the SCAN kiosk maintains reliability, security, and continuous software enhancements with minimal user intervention.

F. FreeRTOS Task Management

Our embedded firmware leverages FreeRTOS to achieve efficient multitasking, enabling multiple tasks to run concurrently with defined priority levels. Tasks interact using synchronization primitives such as semaphores, event groups, and task notifications to coordinate operations and manage conditional execution. SCAN's firmware uses dedicated FreeRTOS tasks for:

1). Managing the system states through a dedicated control task that handles FSM transitions.
2). Maintaining and monitoring WiFi network connectivity for the ESP32-C6.
3). Capturing and processing keypad input events in a non-blocking manner.
4). Routinely refreshing the active GUI displayed on the TFT display.
5). Performing over-the-air (OTA) firmware updates.

Running these tasks concurrently enhances system responsiveness, as task execution is not blocked by procedural code. For instance, during OTA firmware downloads, the kiosk continues normal operation, processing check-ins without delay, since the state control and user-interaction tasks remain unblocked. Additionally, assigning higher priorities to time-critical operations such as WiFi connectivity and OTA updates ensures these tasks receive immediate processor attention, further enhancing overall system reliability and performance.

VI. WEB APPLICATION DESIGN

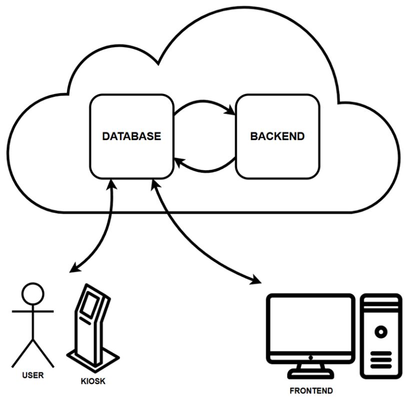

Figure 6: Web App Stack

The SCAN web app serves as the central interface for managing check-in data and user analytics, providing administrators with real-time insights and control. We selected ReactJS for the frontend, Firebase Cloud Functions for the backend, and Firebase Realtime Database to handle data storage and synchronization. This stack was chosen for its scalability, serverless architecture, and seamless real-time updates, allowing for efficient data processing, low-latency interactions, and easy integration across all system components.

A. Frontend Design

For our web app's frontend, we used ReactJS to design three web pages including the Analytics Dashboard, Activity Log, and User Management pages. The latter two of which contain protected user information, only accessible via an admin account.

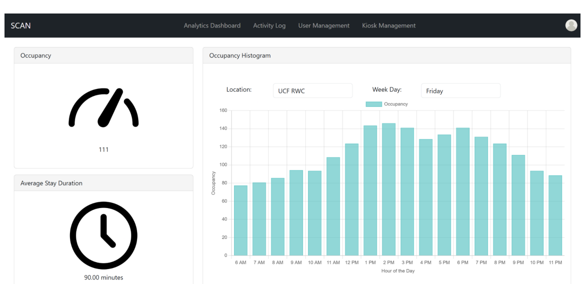

Figure 7: SCAN Analytics Dashboard Page

The Analytics Dashboard serves as the main landing page for all users, displaying statistics users can utilize to better plan their future visits. It contains three main statistical visuals, all synced to the database in real-time using the Firebase SDK. An occupancy counter shows the live number of people at any given location/time, the average stay duration statistic shows how long event goers are expected to stay, and the occupancy histogram shows hourly historical data trends at event spaces.

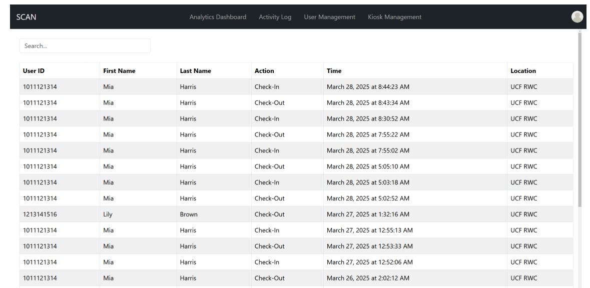

Figure 8: SCAN Activity Log Page

The activity log is a private page, only accessible to administrative users that shows an actively updated log of user check-in and check-out events. The search bar can be used to filter events by userID, name, action, time, and location.

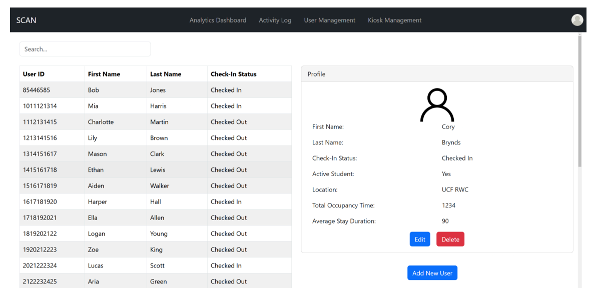

Figure 9: SCAN User Management Page

Finally, the User Management page provides admin users with the necessary capabilities for creating, viewing, updating, and deleting user profiles. It too contains a search feature for quickly retrieving user information.

B. Database Design

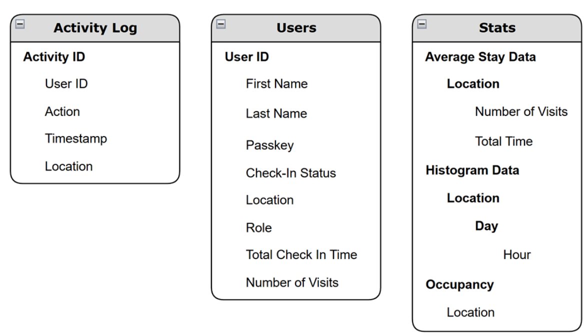

Figure 10: SCAN Database Schema

To connect our kiosk firmware and web app frontend we chose Firebase Realtime Database for its minimal cost, responsive real-time updates, and ease of integration with the frontend, backend, and Firebase hosting.

We settled on the database structure in Figure 10 consisting of three primary nodes including the activity log node which stores all user check-in and check-out event objects, a users node for storing all user information, and a stats node for storing statistics on average stay duration, histogram, and occupancy data. The kiosk firmware interacts with the Firebase Realtime Database via HTTP requests to Firebase's RESTful API. Our ReactJS web app interacts with the database using the JavaScript FirebaseSDK.

C. Backend Design

In order to convert raw check-in data to statistics displayed on our web app, we chose to use Firebase Cloud Functions as our backend. We constructed two primary Cloud Functions. The first Cloud Function we created automatically triggers every time a new activity event is logged in the database (either a check-in or check-out). It updates occupancy count and average stay duration statistics, and handles updating check in statuses and times for user profiles. The second Cloud Function operates hourly, updating the occupancy histogram to show a running average of occupancy with each passing hour.

VII. CONCLUSION

The SCAN (Smart Check-in and Analytics Network) system is designed to enhance user engagement data collection at UCF through a network of check-in kiosks and an intuitive web application. Our team developed a cost-effective kiosk for managing the check-in process, alongside a ReactJS web application for viewing and managing user statistics. Our design features a custom ESP32-based kiosk PCB with an integrated power management system, supporting battery, solar, and USB-C power sources. The kiosk's peripheral subsystem includes a keypad, NFC reader, display, proximity sensor, buzzer, and LEDs, all interfaced with the MCU via a FreeRTOS-based firmware architecture. This firmware handles the check-in process while seamlessly interacting with our Firebase Realtime Database to log check-in and check-out events. Additionally, Firebase Cloud Functions process these updates in real time for immediate display on our web application. Empowered with the analytics data that the SCAN system provides, the RWC and other facilities across UCF will be able to make the most informed decisions to ensure that resources are allocated precisely when and where they are needed most.

ACKNOWLEDGEMENT

The authors would like to thank our project review committee—Dr. Piotr Kulik, Dr. Sonali Das, and Dr. Mike Borowczak—for providing feedback and mentorship in the development of our project. We would also like to thank Dr. Weeks and Hailey Gorak for assisting in the development and troubleshooting of our PCB and Dr. Lei Wei for advising us throughout senior design. Finally, we would like to thank UCF and the ECE department for giving us the tools we need to succeed in senior design and beyond.

BIOGRAPHIES

Cory Brynds is a senior graduating with a BS in Computer Engineering on the Digital VLSI track. During his time at UCF, he has been an active member of IEEE UCF and has previously interned at AMD, nScrypt, and Limbitless Solutions. After graduation, he plans to pursue his masters in Computer Engineering and then enter the semiconductor design industry.

Jonah Sprandel is a senior electrical engineering student focused on power electronics and firmware development. He initially became interested in these areas during his internship at ApECOR. He has since interned at L3Harris Aerojet Rocketdyne and hopes to work on electric vehicle charging in the future. While in college, he was an involved member of IEEE UCF, where he led embedded programming workshops and managed the club's finances as Treasurer.

DeAndre Hendrix is a senior pursuing a BS in Computer Engineering. While at UCF, he has interned at Limbitless Solutions and L3Harris Aerojet Rocketdyne as an embedded software engineer intern, respectively. After graduation, he intends to pursue a masters in Computer Engineering with a focus in embedded software.

REFERENCES

[1] *ESP32-C6*: Espressif Systems, "ESP32-C6 Series SoC" v1.2 datasheet, 2023. [Online]. Available: https://www.espressif.com/documentation/esp32 -c6_datasheet_en.pdf

[2] *PN532*: NXP Semiconductors, "PN532/C1 Near Field Communication (NFC) controller," Rev. 3.6 datasheet, Nov. 28, 2017. [Online]. Available: https://www.nxp.com

[3] *BQ25798*: Texas Instruments, "BQ25798 I2C Controlled" SLUSDV2B datasheet, Rev. B, Jan. 2023. [Online]. Available: https://www.ti.com/lit/pdf/SLUSDV2

[4] *CYPD3177 (EZ-PD BCR)*: Infineon Technologies (formerly Cypress Semiconductor), "EZ-PD™ BCR USB Type-C Port Controller for Power Sinks," 002-25383 datasheet, Rev. *B, May 18, 2021. [Online]. Available: https://www.infineon.com