In-Processing Kiosk for JBER Professional Military Education Center

Connor Kreis & Mika Brassfield
University of Alaska Anchorage
CSCE A470: Computer Science and Engineering Capstone Project
Dr. Frank Witmer
05/02/2024

TABLE OF CONTENT

Abstract

With collaboration from the JBER Professional Military Education Center (PME) and direct oversight from Technical Sergeant Sherita Ariola, the goal of this project was to expedite the check-in process for hundreds of students attending leadership training. It prioritized security, user-friendliness, efficiency, and error minimization. Initial plans for a two-part system were revised due to security concerns, leading to the development of a website for student input and a kiosk interface for QR code scanning. The website, built using Python Flask, generates QR codes containing essential student information, while the kiosk app, developed with PyQt5, validates and processes the scanned QR codes, ensuring security and accuracy. The integrated system significantly reduces administrative burdens, allowing instructors to focus on essential duties during student in-processing.

1. Introduction

This project was created in collaboration with the JBER Professional Military Education Center (PME), our direct contact being Technical Sergeant Sherita Ariola. The PME center seeks to sharpen technical and foundational leadership skills that are fundamental to the overall competence of the Airmen attending. The objective of this initiative was to streamline the in-processing procedures for the hundreds of students attending classes.

The need for this project arose from the growing demand for a more automated system due to the steady influx of students attending courses at the PME center. Manual check-in procedures take a long time and only serve to remove the instructors from their core job of teaching. By developing a simple, web-based data input system for the students and secure kiosk-based validation, we sought to improve workflow and create a better solution to in-processing procedures. Through close collaboration and feedback from the PME center staff, we were able to identify the key points and requirements of the project so that our solution aligned with the PME center's goals of improving an already strong learning environment to ensure the mentorship of strong aspiring leaders in the Air Force.

2. Project Overview

The goal of this project was to create a software program, utilizing the kiosk purchased by the PME Center, that would allow the instructors to automate the check-in process of each student arriving for leadership training at the center. This would remove a large portion of repetitive administrative tasks and allow the instructors to focus on other important duties for the first day. To ensure the delivery of a desired product, all requirements came from TSgt Sherita Ariola herself. Other instructors at the PME center offered their recommendations to her which were then relayed to us when appropriate.

3. Project Requirements

The basic requirements for this project were to have a way for the students' information to be gathered into Excel spreadsheets since the instructors were most familiar with the specific layout; display the class information to the students; and build a recall roster for the instructors to know who had checked in for class and who were still absent. Along with those criteria, we had to keep the following in mind:

3.1. Security

The program would handle the personal information of students attending leadership training, such as names, contact information, medical details, or DoD numbers. The Center's main focus was on the security of the information, given the location of the kiosk is open to the public. Security issues were heavily considered throughout development to safeguard the students' private information.

3.2. User Friendliness

The primary users of the program are the instructors and students at the PME Center. Few assumptions were made about the level of technological proficiency of our users, so a user-friendly design was one of our primary focuses. The design needed to be intuitive and straightforward, ensuring that instructors could easily use the software and retrieve data without technical assistance.

3.3. Efficiency

Given the high volume of students arriving for leadership training, the check-in process needs to be efficient. A user-friendly interface streamlines the process, enabling them to quickly check in students without delays or complications.

3.4. Minimizing Errors

Simplifying the user interface reduces the likelihood of user errors during the check-in process. By presenting clear instructions and limiting user input, the software helps ensure that accurate information is collected from each student without mistakes.

With these aspects in mind, we were tasked to create a program that would take user input in the form of a QR code and produce output in the form of spreadsheets containing key information about the students. As will be discussed in the *Second Implementation* section of this paper, our final functional specifications were for us to create a website and a graphical user interface for the kiosk portion of the project. It was specified to us that the website should generate QR codes containing key information about each student, and the kiosk should accept only those QR codes as input from the students.

4. Project Architecture

The kiosk is a tablet-sized touch screen on a podium structure, currently running Windows 11 operating system. It is equipped with a Honeywell fix mount HF52X scanner, capable of scanning barcodes and QR codes [1]. The barcode scanner is capable of taking input in the form of barcodes or QR codes and turning them into keyboard input that can then be interpreted by the kiosk. The kiosk can be programmed to process this data and turn it into a more usable format.

4.1. First Implementation

The first design idea we attempted was a two-part structure with an interface for both the instructors and the students. The instructors would continue their process of entering the students' information into their Excel spreadsheets. The students would scan their Common Access Card (CAC) using the scanner on the kiosk and their class information, such as class number and designated "flight", would be displayed on the screen. Figures 1 and 2 show the

layout of the CAC and what information can be gathered from it. The kiosk program would then prompt the student to provide a contact number, generating a recall roster for the instructors. Figure 3 is a visual representation of the flow of the design structure.

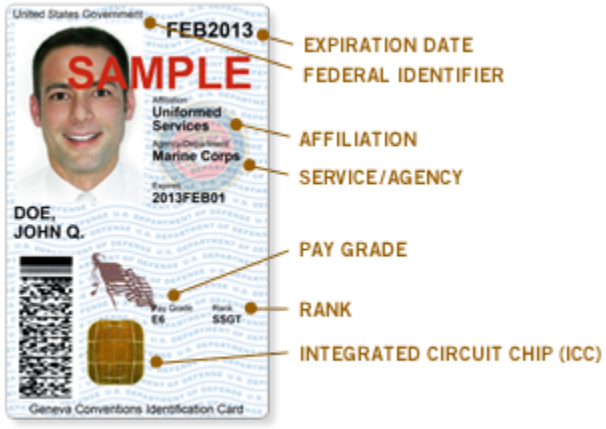

Figure 1: Front view of a Common Access Card

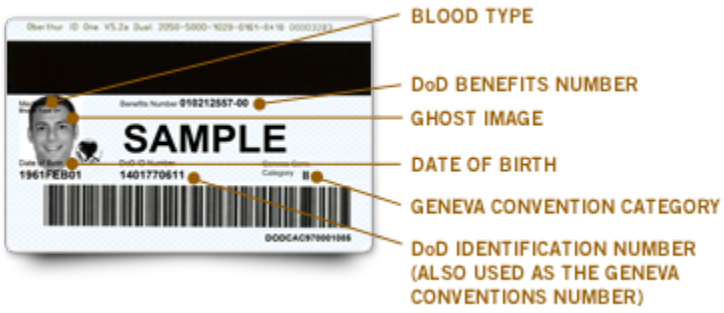

Figure 2: Reverse view of a Common Access Card

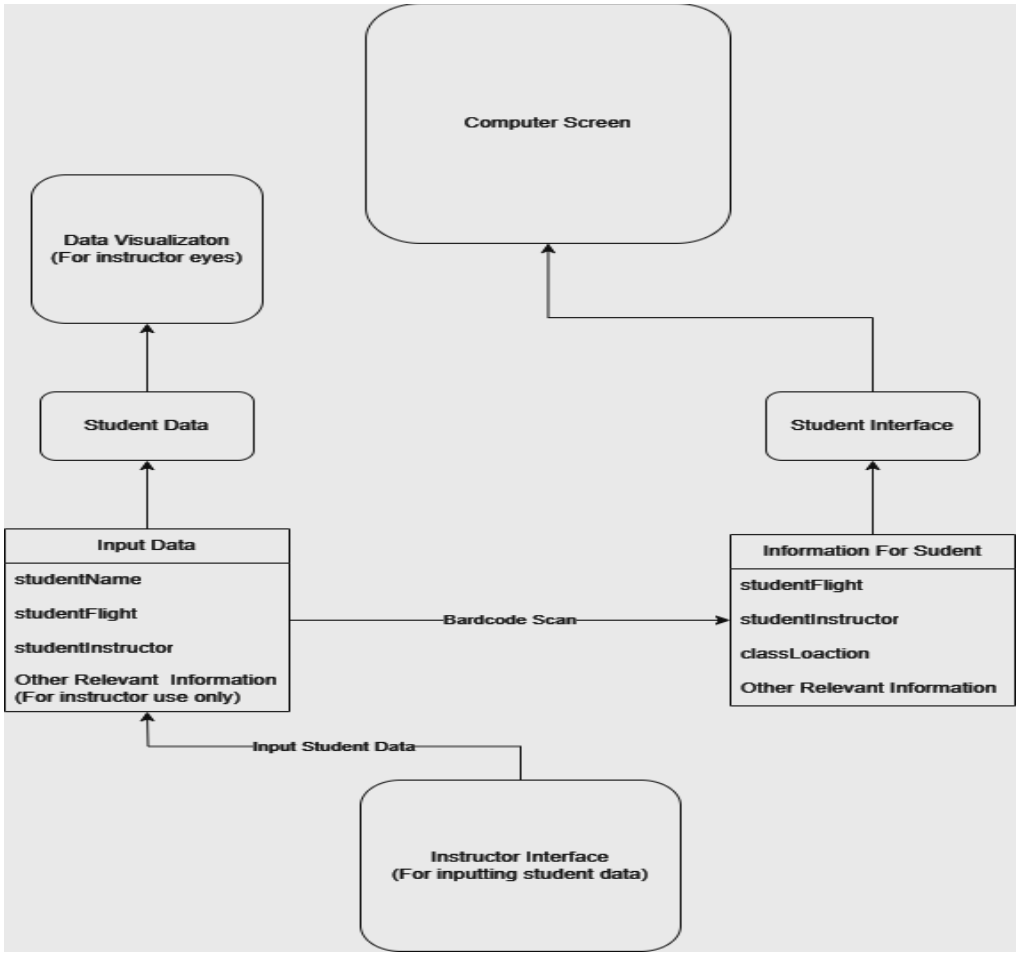

Figure 3: Initial Design Layout

This approach mostly failed due to concerns regarding the security of the system. More details of this can be found below in the *Second Implementation* section. In addition, the implementation wouldn't improve the overall workflow. A bigger quality of life increase was needed due to the high throughput of students. It was decided that this approach should be abandoned in favor of a more secure and transformative process that worked well for both the students and the instructors.

4.2. Second Implementation

Due to a change in security requirements, many parts of our initial design had to be scrapped. It's important to understand that the DoD uses NIPRNet, or non-classified internet protocol

router network, to access the internet. Communications from NIPRNet to the internet are heavily controlled and monitored. This is to prevent the spillage of personally identifiable information that is often stored on these computers, such as DoDID numbers or Social Security numbers. The kiosk itself is connected to the internet through a public WiFi connection, meaning it does not undergo the same monitoring and control that computers on the NIPRNet are subject to. This means that transfer between the two is difficult and should not involve the exchange of personally identifiable information. Due to the recognition of this issue, a new design was devised and it consists of two main components: a website and a kiosk interface.

The website is essentially an e-form that prompts the student to enter essential information for in-processing. This information includes their first and last name, rank, unit, phone number, and a couple of questions regarding their fitness status. After the student submits, the web app generates a QR code that the student is then asked to bring with them to their check-in day. The web app is built using Python Flask and is deployed on the Heroku platform as a service. This allows the website to utilize the Python library "qrcode" to generate QR codes containing information inputted by the user on the form. The form is rendered according to the specifications defined in the HTML/CSS files. A rendering of the user interface is shown in Figure 4. There is a large focus in the code on ensuring that the information the user submits is a valid input that follows the expected format of each data field. This means checking for invalid characters in their phone number, name, or any other data field and clearly outputting any errors for the user to correct issues. This is to ensure that each user input follows an expected structure so no problems are introduced in the end product. The website then prompts the student to save the QR code for their check-in day. Realistically, as long as the student has internet access in the lobby of the PME center, the form is simple enough for them to do right there if need be.

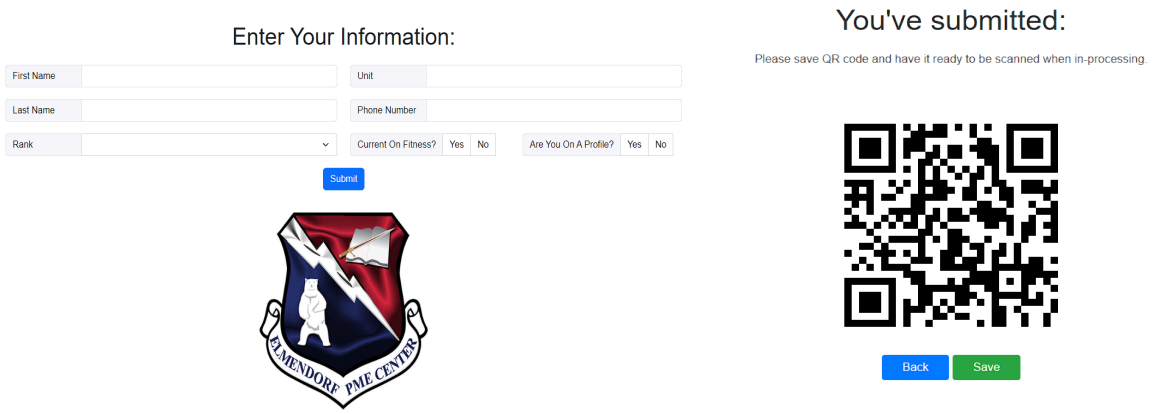

Figure 4: Website screenshots showing the form (left) and the submitted screen (right)

The second main aspect of this project involves the kiosk interface. The kiosk app is again implemented using Python, with the GUI being implemented using PyQt5. PyQt5 is a GUI toolkit that allows for the use of methods such as setStyleSheet() which allows us to mimic the exact style of our website's interface. It also allows us to override the function keyPressEvent() which was integral to our implementation. keyPressEvent is called every time an input from the keyboard is detected. Since most barcode scanners are interpreted by computers as a form of keyboard input, this makes it really easy to capture the signal from the scanner. Inside the keyPressEvent function, we used the state machine shown below in Figure 5 as a guide for the code. A scan is considered complete once either the terminal symbol is reached (for our machine it is '$') or the scan times out due to it not reaching the terminal symbol (5 seconds). After the scan is completed, the program will then validate the given input against the same parameters enforced on the website to ensure that the given QR code contains the needed information. Any unexpected formats are immediately rejected. This means any malicious QR codes that are meant to interrupt the operability of the kiosk are rejected. Once the final accepting state is reached, a checkmark GIF is played to show the student that a successful scan has been completed, and their information is then written to spreadsheets that the instructor may reference later. Other additional bits of user feedback are shown during the process such as a loading screen to indicate the program is running and error messages to show a negative scan. Figure 6 shows each state the kiosk screen can be in depending on which state it is in on the mock state machine shown in Figure 5.

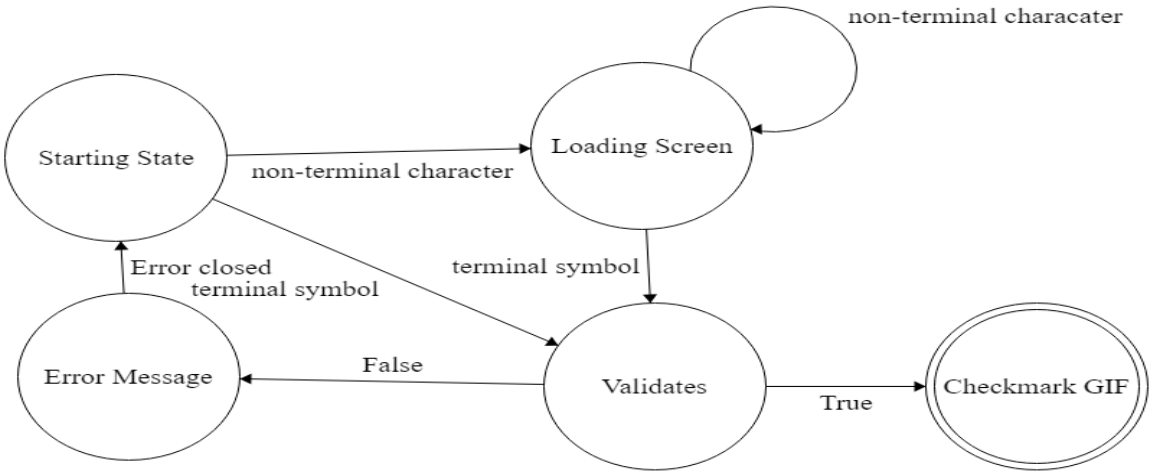

Figure 5: State Machine

Figure 6: Kiosk screenshots showing idle screen (top left) loading screen (top right) accepted screen (bottom left) and error message (bottom right)

5. Development Process

This project had a lot of new, unfamiliar territory for us to learn through. Arranging meetings with our client and working to avoid schedule conflicts was different from the solo projects we have completed in classes before. Once we figured out how the kiosk worked, the next big step was learning how to develop GUIs, as we had limited knowledge of app designs. This again differed from our experience in classes. In classes, the emphasis is often on the solution to technical problems. We realized quickly that this would not be the case for this project. Many people care more about how it looks rather than how it works, or the two are intertwined with them; a product that looks cheaply made will have the perception that it does not work as well. This means that we had to place a larger emphasis than we ever have had to on the looks and the design of our project. Unlike other academic problems, there is not a definite solution to good GUI design. While there are resources available that help you achieve better results, there is no definite "answer" so to speak. This means we had to draw on all our experiences interacting with websites and let that inform our design.

The security issue in our first implementation was the largest setback we faced throughout the project, and having to go back to the drawing board did make our time for completion shorter. Though drastic changes were made, some of the code we had written was still salvageable. Encountering this issue also framed the whole project in a different light. It made us realize what the stakes were and again just how different this project was compared to our academic experience to this point. Things like security are often discussed in classes with some pointers on how to make programs secure, but there is no set checklist that deems a program risk-free. This caused us to think more consciously about potential security problems and address them before they could become issues. For this project, this mostly came in the form of ensuring the scanned inputs in the kiosk follow the same format as those generated by our website. This is how we can make sure that no malicious QR codes are scanned.

6. Results

Both of our implementations have met their specified requirements and demonstrated successful functionality. While the development was successful overall, we acknowledge that there is room for improvement regarding optimizing the instructor experience with the program. This might include more effective ways for the instructors to pull stored information on the kiosk for their own purposes, but no efficient solution could be identified in the timeframe we were given.

Our testing process involved the validation of the kiosk's robustness and security. It was an emphasis to identify any potential attack vectors and find any shortcomings of the system. We designed tests utilizing mock data to simulate various scenarios to find any weaknesses. Both valid inputs and malicious invalid inputs were tested to spot any weaknesses. Malicious inputs may include any QR code that was made to intentionally disrupt the kiosk and distract it from its overall purpose. This included testing QR codes that could be anything from URLs to codes that mimicked the format generated by the website but were wrong in subtle ways. The kiosk's timeout feature, which allows each scan a maximum of five seconds to complete, effectively mitigates potential issues with incomplete or non-standard QR codes. The timeout and validation features ensure users are promptly notified of any difficulties, maintaining a smooth and reliable check-in process.

7. Conclusion

In conclusion, the development of our application aimed at expediting the student check-in process at the JBER PME center has been successfully completed. Through close collaboration with Technical Sergeant Ariola and other members at the PME center, a security conscious approach has been successfully accomplished. This approach was realized through the creation of a web application and a kiosk interface, which work in tandem by adhering to the same data standards encoded in the QR code.

This approach streamlines the check-in process by removing the need to manually input information into spreadsheets for each student. By automating data capture through QR code scanning, instructors can redirect their focus towards their teaching duties rather than repetitive

administrative work. This not only improves the efficiency of the PME center, but it also enhances their overall productivity by allowing instructors to allocate more time and resources towards enhancing the education experience of their students. This potential increase in productivity will produce better leaders for the Air Force at large, turning a moderate quality-of-life improvement into a much greater overall contribution.

This project tested our limits and required a lot of growth. Our largest takeaway is how to effectively deal with customers to deliver a desirable and realistic product. Our requirements changed drastically late in the project's life, and that was mostly due to us realizing a problem and addressing it too late. This served as a valuable lesson in drafting requirements and doing thorough research earlier in the project. This would have allowed us to detect the issue earlier and deliver an even stronger product. Outside of dealing with customers, our user interface skills were tested and honed during this project. Learning how to make a product look clean and presentable for users was a skill we had not yet developed. Development challenged us to find clean and elegant ways to provide users with feedback to ensure the proper execution of the program. It cannot be understated how valuable and impactful these lessons have been, even over the course of one short semester.

Lastly, we would like to thank Technical Sergeant Ariola for her patience and willingness to collaborate with us, as well as the rest of the PME center staff. Without their support, this project would not have been possible, and we will be forever grateful for their contribution.

References

[1] Honeywell International. (2021). HF52X fixed Mount Barcode Scanner - User Guide. https://prod-edam.honeywell.com/content/dam/honeywell-edam/sps/siot/en-gb/products/barcode-scan-engines-modules-and-decoding-software/barcode-scan-modules/hf521-2d-imager-module/documents/sps-siot-optical-sensing-hf52x-2d-imager-module-user-guide-en.pdf

Actions Taken

Severity	Line Numbers / Heuristic Issue	Issue Description	Actions Taken
H	10 / Heuristic	More instructions on what to do with the QR code	Added short instructions to the submitted QR code page
L	N/A	No docstrings for any of the functions	Some docstrings were added for clarity
M	1-16	Special characters in names are not allowed for input	Included special characters in name regex
M	Multiple	Fallbacks for if statements (check if else is necessary?)	Added cases where False is returned if no error is encountered for clarity
1	8 / Heuristic	Summary of submission information is not well formatted	Removed summary for development purposes
H	155-156	Check for malicious Excel macro in the QR code & other issues; verify the QR code	Input re-validated before being added to spreadsheets, removing the likelihood of malicious macros
M	4, 13, 27, 34	Magic numbers	Eliminated magic numbers where possible
M	19-21	Confirm that the rank matches something in the expected list	Now confirms it is in the list
L	Multiple	Change function names to reflect error-checking	Docstring makes it clear that functions return errors