JBER PME Kiosk

Connor Kreis and Mika Brassfield

JBER Professional Military Education

- Place where airmen learn supervisory skills
  - Leadership development
  - Networking and collaboration
- Every prospective supervisor must attend
- In process thousands of students a year
- Important stage of your career

Problem Recap

Instructors want to focus on more time teaching rather than doing administrative work

Prior Process

- Students would give instructors information on their first day
- This information would be manually entered and stored on excel spreadsheets
- Every time new students come in, they get added
- Repeated for thousands of students every year
- Very tedious, time-consuming process

Kiosk or Doormat?

- The kiosk is a very simple device
- Essentially a Dell computer with barcode scanner
- Windows 11 operating system
- Touch screen with very limited I/O

Our Goal

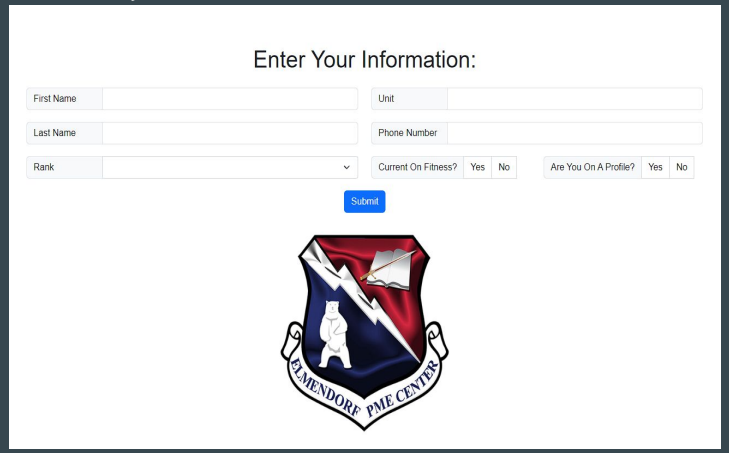
- Some complications at the beginning
  - Security is a high priority
  - Requirements changed
- Broken down into two separate but connected components
  - Website
  - Kiosk interface
- Needs to be user friendly
- Limited instructor interaction

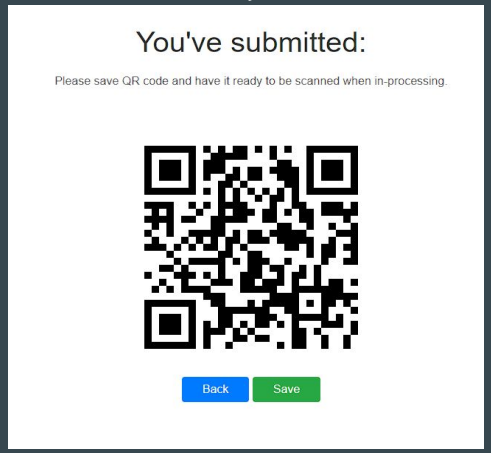
Solution

- Have students produce QR code before they come to class
- Have students scan QR code at kiosk
- Have that check them in and record down the relevant information
  - Need a way for format to be consistent
  - Can’t trust the students to produce QR code on their own
- This is where the website comes in

Website

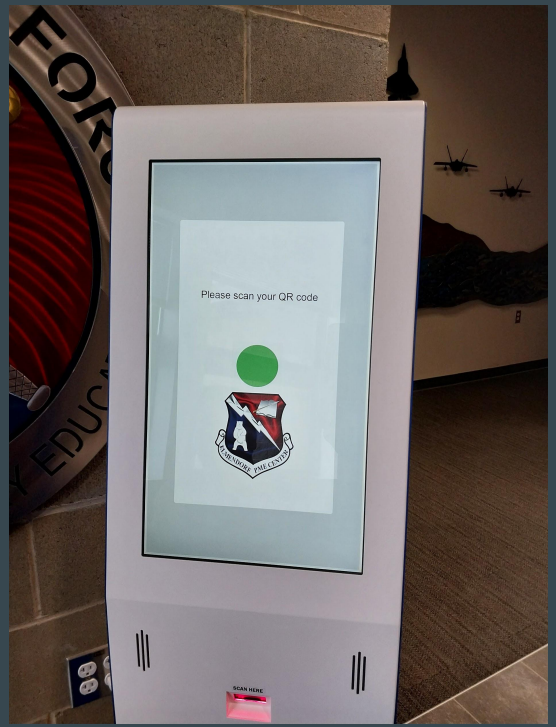
- Essentially an E-Form for students to fill out
- Input is validated to ensure nothing malicious
- Records down the essential information for recall rosters
- Produces a QR code that they can save for check-in
- Easy to fill out, could even be done in the lobby





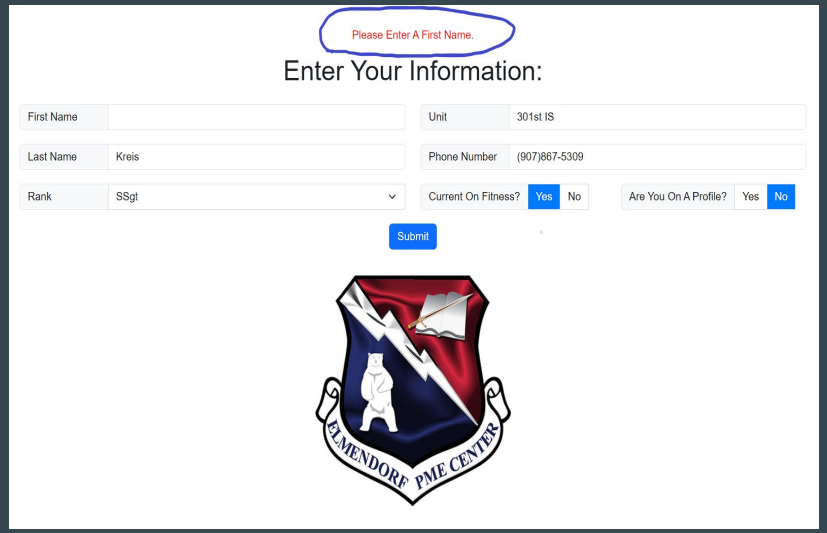
Kiosk Interface

- QR code scanned by barcode scanner
- Kiosk validates the input again
- Rejects any QR code not formatted from the website
- Prevent against malicious QR codes
- Information is written to a spreadsheet, accessed by instructors



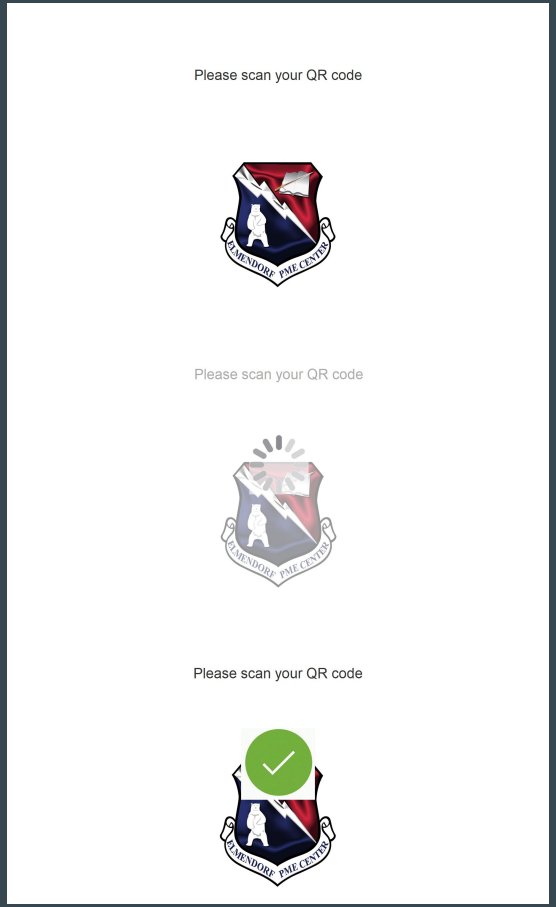
Website Development

- Hosted by Heroku
  - Allows customer to take ownership after project
  - Some budget for them
- Logic implemented using Python Flask
- HTML/CSS for page rendering
  - Bootstrap libraries used to make it look pretty
- Had to make it unbreakable
- Implement user feedback/errors
  - Make it look natural and easy to understand

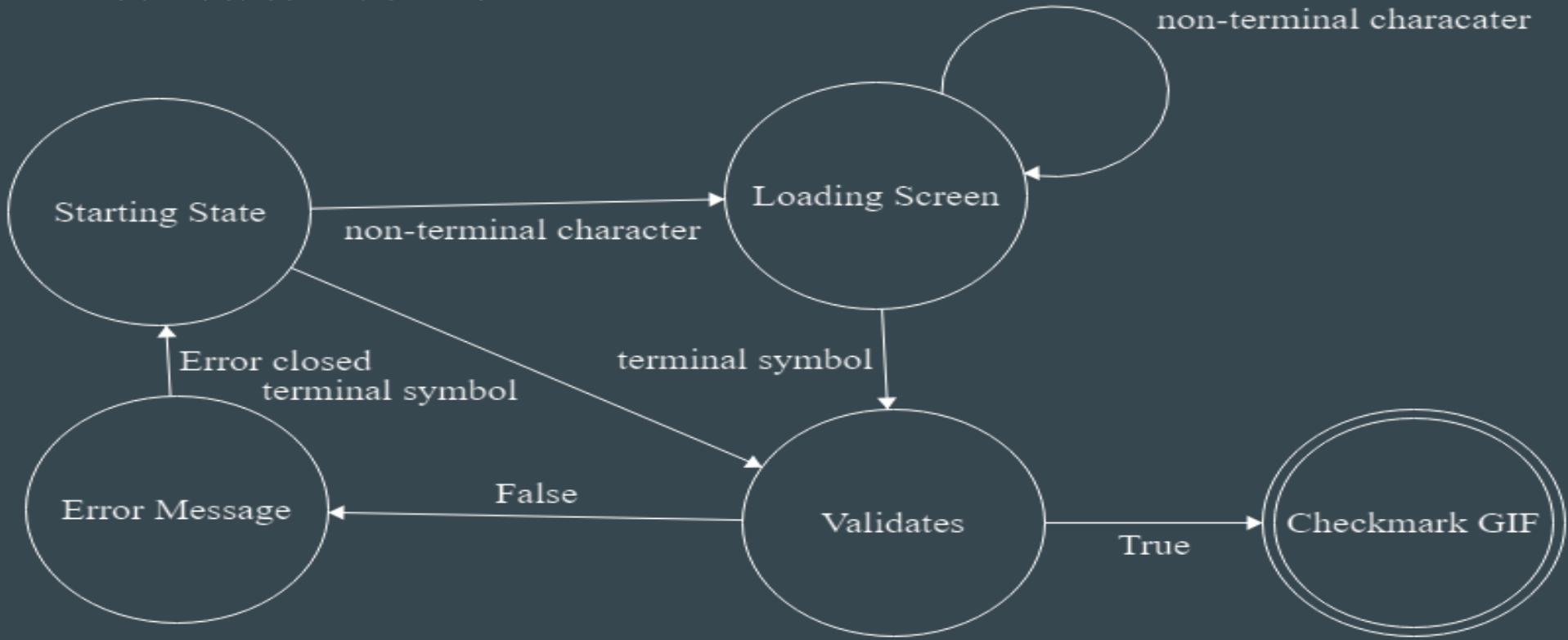


Kiosk Development

- PyQt5 GUI toolkit
  - Allowed use of .setStyleSheet() to mimic design of website
- Barcode scanner treated as keyboard input
  - Makes obtaining data from scans easy
  - keyPressEvent() to capture scan data
- Each scan validated against website parameters
- Visual feedback to inform user
  - Loading screen when scanned
  - Checkmark GIF when accepted
  - Error popup when bad QR code scanned



Kiosk state machine



Development

- New, unfamiliar territory for us
- Never really worked for a customer
  - Work with the requirements given to us
- Had to learn a lot about GUI design
  - Visually appealing
  - Feedback for the user
- Large emphasis on user-friendliness
  - Contrasts what we've learned

Final Remarks