ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Intelligent Shared Item Cabinet

Team #23

NIHAOXUAN RUAN
(ruan14@illinois.edu)
YIHONG YANG
(yihongy3@illinois.edu)
XIAOTONG CUI
(xcui15@illinois.edu)
YANXIN LU
(yanxinl4@illinois.edu)

TA: Luozhen Wang

May 18, 2025

Abstract

This document provides an outline and LaTeX template for report formatting in Senior Design. This document does not teach you what to include, or how to use LaTeX. Assumes a workable level of LaTeX proficiency.

Contents

1 Introduction

1.1 Purpose

In modern campus residential college environments, students often need to borrow small items such as scissors, glue, or thermometers during non-office hours. However, the lack of access outside standard front-desk hours frequently causes inconvenience and delays. Manual borrowing systems are typically limited by fixed working hours, lack real-time tracking, and offer minimal security or accountability. This gap leads to inefficient usage and reduces overall user satisfaction.

Our project aims to address this issue by developing an Intelligent Shared Item Cabinet that provides 24/7 access to shared items through campus card authentication and automated item recognition. The system combines hardware and software components such as electromagnetic locks, weight sensors, cameras, and a touchscreen interface. These components work together to enable secure, real-time borrowing and returning of items. Each transaction is logged automatically to ensure traceability and system reliability. The cabinet is designed to be scalable, user-friendly, and suitable for daily student use.

1.2 Functionality

This project is designed to provide a smart and reliable way for students to borrow and return shared items, such as scissors, glue, or thermometers. The system includes several main functions that together support the overall goal of making the borrowing process more convenient, secure, and easy to manage.

- **Simple and user-friendly borrowing process.** Students with the proper access rights can use their campus card to start the borrowing process. Once a valid card is scanned, the cabinet unlocks automatically within 2 seconds and the user can take the item. The touchscreen gives clear instructions to guide users through each step. This function ensures that students can easily borrow what they need without needing staff assistance.
- **Accurate item recognition.** To confirm that the returned item is the same one that was borrowed, the system uses three methods: weight detection, image recognition, and NFC verification. The load cell checks if the item's weight matches the recorded value. The camera captures the item's image to assist with visual confirmation. Each item also has an NFC sticker, which is scanned to verify identity. If any mismatch is found, the system notifies the user through the screen. These methods together help improve accuracy and reduce the chance of errors.
- **Administrator control through user interface.** In addition to regular borrowing functions, the system includes a management interface for administrators. Through the touchscreen, an admin can add or remove users and update item type without using an external computer. This feature improves system management and makes it easier to handle changes in users or items over time.

1.3 Subsystem Overview

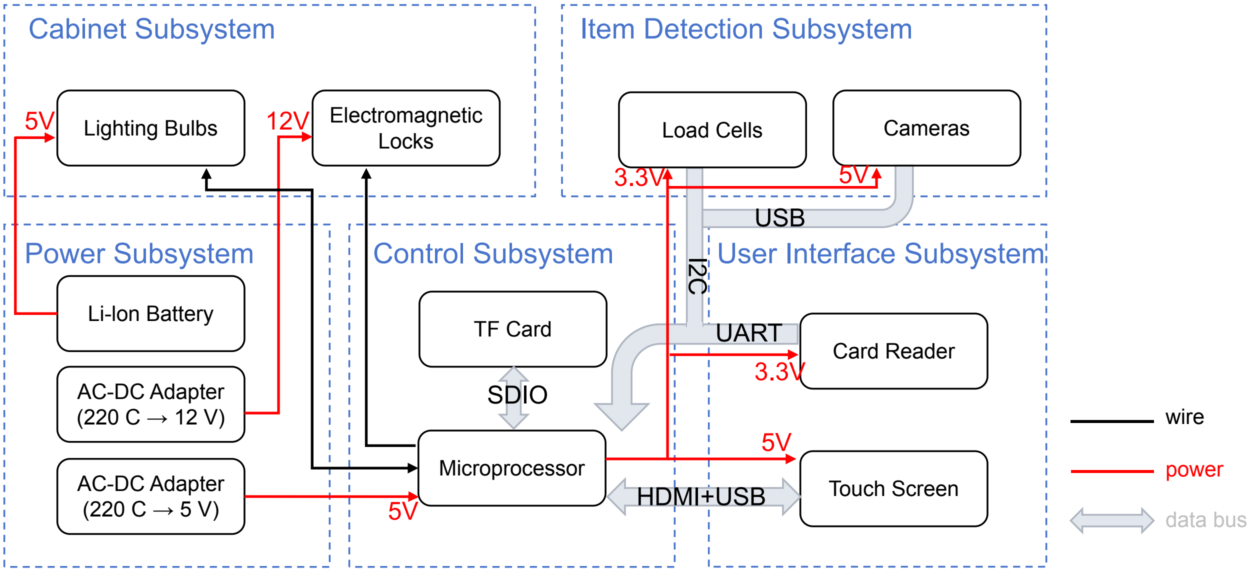


Figure 1: Top-level Block Diagram.

The system is composed of five main subsystems: the Power Subsystem, Control Subsystem, Cabinet Subsystem, Item Detection Subsystem, and User Interface Subsystem. These subsystems are connected through both power lines and data communication interfaces, as shown in Figure 1.

The Power Subsystem is responsible for supplying stable power to the entire system. It directly powers the Cabinet Subsystem and the Control Subsystem using 12V and 5V lines, respectively. The Control Subsystem serves as the central hub that manages the operation of all other subsystems. It sends control signals and receives status feedback from the other modules.

The Cabinet Subsystem manages the physical status of the cabinet, including the electromagnetic locks and lighting components. It communicates door status information back to the Control Subsystem to ensure proper access control.

The Item Detection Subsystem collects data from load cells, cameras, and NFC tags to help identify whether the correct item has been returned. This information is sent to the Control Subsystem for analysis and decision-making.

Finally, the User Interface Subsystem handles interactions with the user through the touchscreen and NFC card reader. It passes user input and authentication results to the Control Subsystem, which processes the commands and coordinates the system response accordingly.

Through this architecture, the Control Subsystem functions as the core controller, while the other subsystems act as functional modules that provide data and execute actions based on centralized commands.

2 Design

2.1 Equations & Simulations

We validate some of our design via Python-based simulations.

Lighting Bulbs Three-watt white LEDs at 5 V are installed to maintain internal illuminance above 150 lx for reliable camera recognition. The power consumption is

$$P = V \times I = 5\,\mathrm{V} \times 0.6\,\mathrm{A} = 3\,\mathrm{W}.$$

With a 400 mAh backup pack (energy $E = V \times Q = 5\,\mathrm{V} \times 0.4\,\mathrm{Ah} = 2\,\mathrm{Wh}$), the theoretical runtime is

$$t = \frac{E}{P} = \frac{2\,\mathrm{Wh}}{3\,\mathrm{W}} \approx 0.67\,\mathrm{h}.$$

However, since lighting is only enabled during the door-open recognition phase (typically less than 30 s per cycle), this capacity supports hundreds of operations between charges.

2.2 Design Alternatives

We evaluated multiple design options and addressed issues through quantitative analysis.

Lock Mechanisms We compared traditional mechanical key locks against electromagnetic locks (12 V, 2.4 A). Mechanical key locks offer low cost and zero power draw when idle but require manual operation, provide no electronic status feedback, and cannot integrate into the automated access-control system. Electromagnetic locks, actuated by a 12 V pulse, unlock within $1\,\mathrm{s}$ and then automatically relock via a spring-loaded latch upon power removal—mimicking real-life locker behavior. They include an internal door-status switch for real-time feedback, support remote unlocking, and feature a manual override lever for emergency access. Given the need for automated control, feedback, and user convenience, electromagnetic locks were selected.

Processing Unit Initially, we planned to use an STM32 microcontroller (e.g., STM32F407VG) for control tasks. Benchmark testing revealed that the STM32 could not handle multi-class YOLO inference, with times exceeding 5 s per frame. Consequently, we selected the Raspberry Pi 5 (2.4 GHz quad-core ARM, 4 GB RAM)[1], achieving inference latency below 100 ms per 640 × 480 frame.

Touch Display The original design included a capacitive SPI 3.5-inch MHS display requiring custom driver integration. During prototyping, after installing multiple software packages, the screen failed to display content (remaining white) due to driver conflicts with the OS environment. We switched to a plug-and-play 7″ USB touchscreen, eliminating compatibility issues and significantly reducing development time.

2.3 Design Description & Justification

To meet the high-level requirements and enable independent development, testing, and future extensibility, the design is organized into five modular subsystems: Cabinet, Power, Control, Item Detection, and User Interface. Each module interfaces via standardized power rails (3.3 V, 5 V and 12 V) and digital buses (SPI, I2C, UART). All design decisions are supported by analysis and prototype testing.

2.3.1 Cabinet Subsystem

The Cabinet Subsystem provides the structural enclosure and security mechanisms for individual storage compartments, ensuring safe, reliable containment and controlled access.

- **Lighting bulbs.** Three-watt white LEDs at 5 V are installed to maintain internal illuminance above 150 lx for reliable camera recognition. The microprocessor enables the LEDs only when the door is open or the system is recognizing items.
- **Electromagnetic locks.** XG-07 Series locks (12 V, 2.4 A) provide a 600 kgf holding force. Each lock integrates an internal door-status switch and a manual release. Accelerated life-cycle tests (1,000 cycles/min for 5 h) project service life >300,000 cycles, with an average unlock time $t_{\text{unlock}} = 1.2\,\text{s} < 2\,\text{s}$.

2.3.2 Power Subsystem

The Power Subsystem delivers stable, regulated voltage rails and backup supply to guarantee uninterrupted operation across all modules.

- **Li-Ion battery.** A 5 V, 400 mAh Li-Ion pack supports emergency lighting for up to 40 min and can be recharged.
- **AC–DC adapter (220 V →12 V).** Supplies electromagnetic locks with ripple <100 mV_{pp} and efficiency $\eta > 85\%$.
- **AC–DC adapter (220 V →5 V).** Powers the Raspberry Pi and peripherals at up to 4 A.

2.3.3 Control Subsystem

The Control Subsystem hosts the primary processing unit and manages all data communication, storage, and real-time control tasks.

- **Microprocessor.** Raspberry Pi 5 (4 GB) executes real-time control and runs YOLOv5 at 15 fps on 640 × 480 frames. SPI, I2C, and UART interfaces achieve data rates $\geq 2\,Mbps$ and function well when integrated together.
- **TF card.** A 32 GB TF card on the SDIO bus sustains 20 MB/s write speeds and remains error-free over 1,000 cycles, meeting data logging requirements.

2.3.4 Item Detection Subsystem

The Item Detection Subsystem combines weight measurement and visual analysis to identify items placed within the cabinet.

- **Load cells.** The PN532 module communicates over UART and is interfaced via a TTL-to-USB FT232 converter, completing card read/write in 100 ms with a bit-error rate below 0.1% across 500 trials.
- **Cameras.** USB 1080p @ 30 fps cameras operate on 5 V. End-to-end capture-to-recognition latency is less than 100 ms; YOLOv5 inference on scaled frames achieves 15 fps, exceeding the 10 fps real-time requirement.

2.3.5 User Interface Subsystem

The User Interface Subsystem enables secure user authentication and intuitive interaction through NFC and touch display.

- **NFC card reader.** The PN532 module communicates over UART and is interfaced via a TTL-to-USB FT232 converter, completing card read/write in 100 ms with a bit-error rate below 0.1% across 500 trials.
- **Touch screen.** A 7″ IPS display (1024×600, 60 Hz) via HDMI renders updates in $<$16 ms. USB touch interface reports events with 80 ms latency.

2.4 Subsystem Diagrams & Schematics

2.4.1 Cabinet Subsystem

We employ a control circuit to drive the electromagnetic lock. Testing has shown that our lock requires 12V and at least 1.5A to operate reliably, so we chose a 12V, 3A output voltage regulator as the circuit's power supply. As illustrated, a MOSFET (30N06) switches the lock: when the Raspberry Pi outputs its control voltage, the MOSFET conducts, and the lock releases. At the same time, we wire the lock's feedback circuit in parallel to relay its status back to the Raspberry Pi. Since the Pi's GPIO can suffer irreversible damage from input voltages above about 3.6V, we protect both the MOSFET and the Pi by placing an SR560 diode as a flyback (freewheeling) diode. We also add a series resistor and a pull-down resistor to limit current and prevent overheating or accidental short circuits. Finally, we selected the 30N06 MOSFET and the SR560 diode specifically for their voltage and current ratings, which exceed the circuit's maximum possible values, ensuring safe operation without component breakdown.

Similarly, we use MOSFET to control the light bulbs in our cabinet as well. When the Raspberry Pi outputs its control voltage, the corresponding light bulb will be on.

The circuit schematic is shown in Figure 2, and the PCB layout is shown in Figure 3.

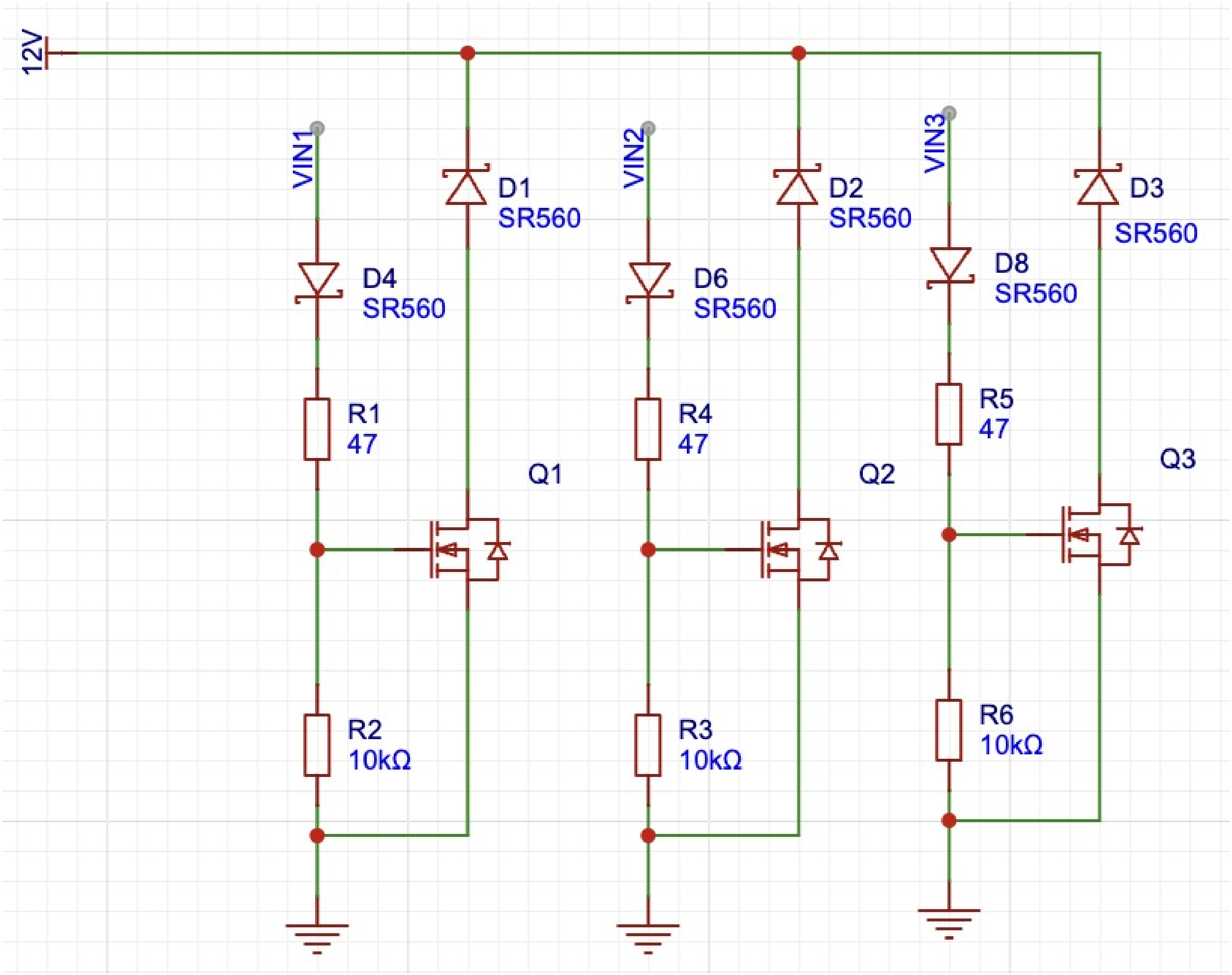


Figure 2: Circuit Schematic.

2.4.2 Item Detection Subsystem

Our item-detection workflow begins by cataloguing every object destined for the cabinet and capturing high-resolution photographs of each under consistent lighting and background conditions. We then apply a suite of data-augmentation techniques to expand the effective size and variability of our dataset. Next, we train a YOLO object-detection model exclusively on these augmented images. Since the training set contains only the specific items in our cabinet, the model learns to identify "this exact pair of scissors," "this exact screwdriver," etc., and will not recognize visually similar but unauthorized objects. This tailored approach guarantees that only our own items are detected and returned. The prediction results are shown in Figure 4.

2.4.3 User Interface Subsystem

We provide a fully interactive touchscreen UI running on the Raspberry Pi, through which users can perform both "Borrow" and "Return" operations with a few taps. When a user

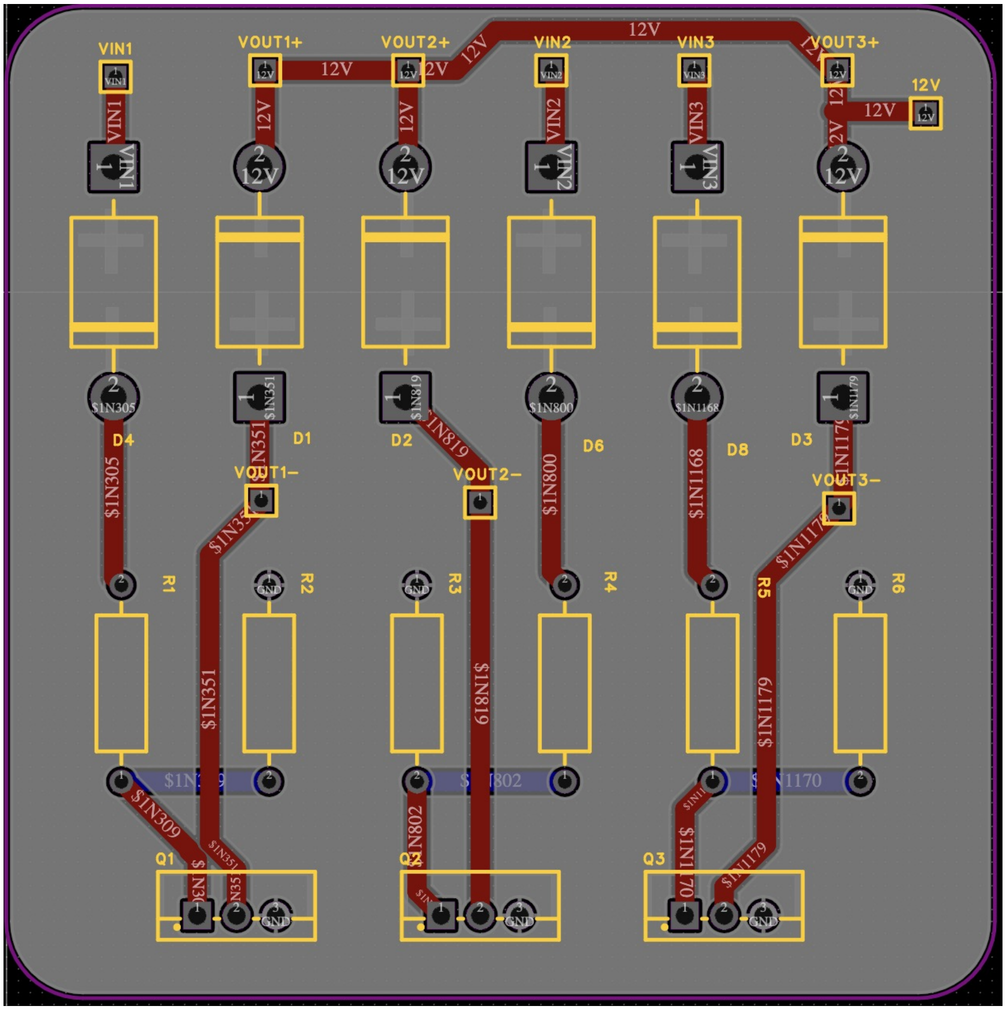


Figure 3: PCB Design.

taps the button, the system immediately prompts for authorization via proximity card. The user holds their card to the PN532 reader; once the PN532 returns a valid UID, the Pi verifies it against the on-device user database.

Upon successful authorization, the UI transitions to an item-selection screen: for borrowing, it lists all available cabinet items; for returning, it shows only the items currently checked out to that card. Users tap the icons of the items they wish to borrow or return, then the locker will unlock.

If card authorization fails (unknown or expired UID), an error dialog appears and the UI returns to the main menu. All screens are built in Python using the Tkinter framework for reliable interactions.

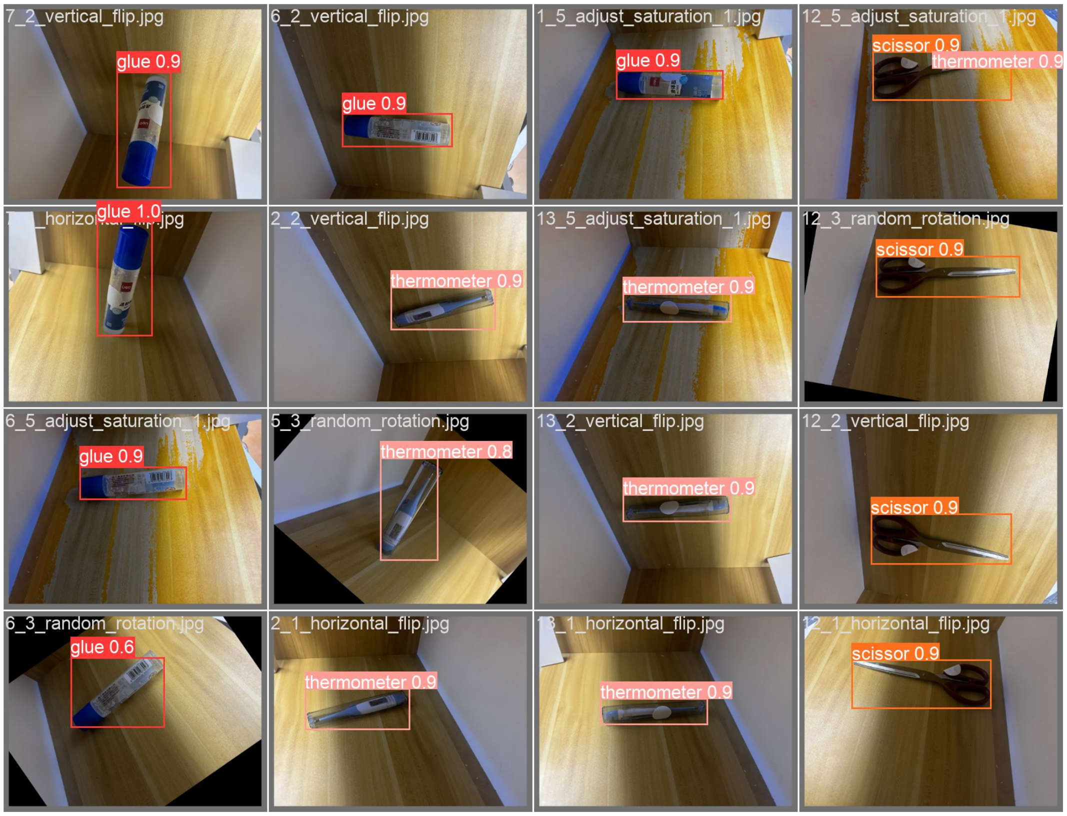
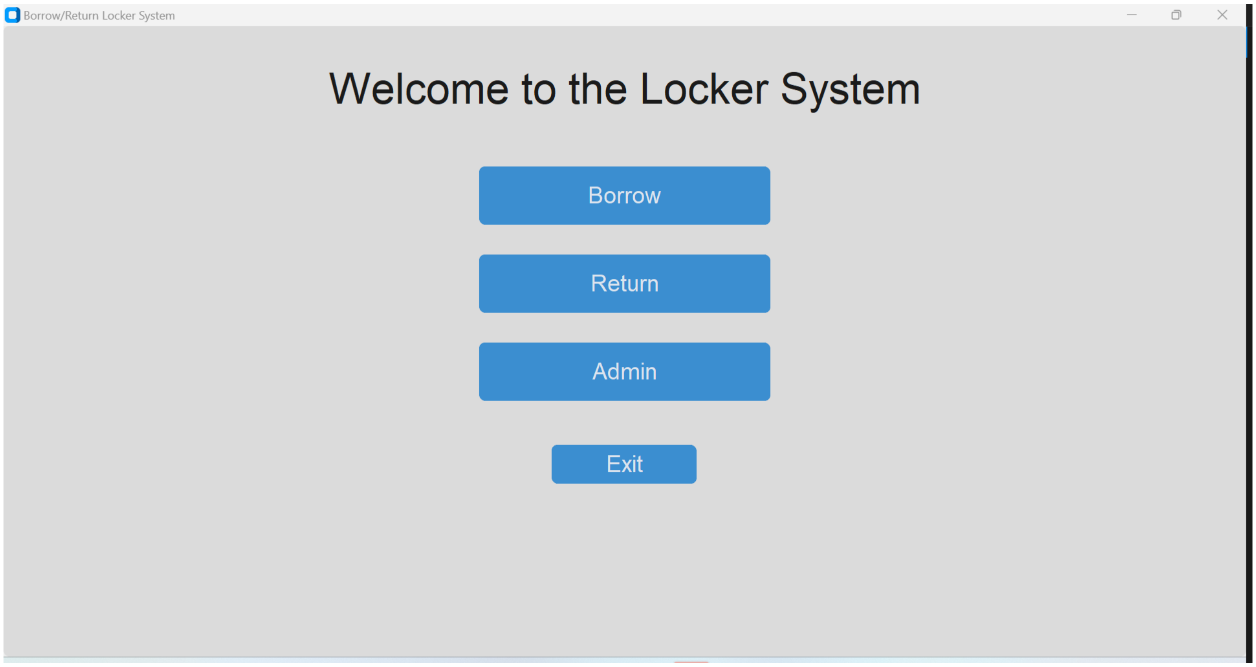


Figure 4: YOLO model prediction results on cabinet item, displaying bounding boxes and labels for each specifically trained object instance to ensure precise identification of authorized items.


Borrow/Return Locker System

Welcome to the Locker System

Figure 5: Main Menu.

3 Cost & Schedule

3.1 Cost

According to UIUC ECE department, the average starting salaries of Electrical Engineering and Computer Engineering graduate are 87769 and 109176 dollars [7]. So the total labor for all partners is:

$$\text{Labor} = 4 \cdot \frac{(87\,769 + 109\,176) dollars}{2} \cdot \frac{1\,\text{year}}{2080\,\text{hours}} \cdot (10\,\text{hours/week}) \cdot (12\,\text{weeks})$$
$$raft \approx 22\,724.4\ \text{dollars}.$$

Part #	**Mft**	**Price (¥)**	**Qty**	**Total (¥)**
Small Storage Cabinet Lock	Guangzhou Sai Rui Factory	15.5	4	62
4-door, lockable storage cabinet	Xuzhou Seven-Colored Fox Furniture Co., Ltd.	130	1	130
HX711 Load Cell	Unknown	14.25	1	14.25
External Keyboard Expansion	Unknown	3.77	1	3.77
Raspberry Pi 3B+/4B N	Raspberry Pi Ltd.	115.69	1	115.69
Raspberry Pi 5 (5B) 3.5-inch 50fps Display	Raspberry Pi Ltd.	58.69	1	58.69
Raspberry Pi 5th 5B/4B Development Board with Camera	Raspberry Pi Ltd.	598.6	1	598.6
			Total	**983**

Table 1: Bill of Materials

3.2 Schedule

Table 2: Weekly Project Schedule

Week	Niahoaxuan Ruan	Yihong Yang	Xiaotong Cui	Yanxin Lu
3/17	Research suitable electromagnetic lock modules and analyze their power/control requirements for cabinet use.	Review basic circuit needs and draft initial wiring plan for pressure sensors and lock control.	Set up development environment for Raspberry Pi and study its GPIO and serial interfaces.	Study Raspberry Pi touchscreen libraries and prepare simple demo code for UI testing.
3/24	Collect materials for cabinet construction and begin assembling cabinet frame and internal mounting structure.	Identify suitable components and sketch first draft of schematic layout.	Prepare GPIO interface test scripts and verify initial communication with Raspberry Pi.	Test card reader and display connection with Raspberry Pi using example scripts; confirm device recognition.
3/31	. . .	. . .	. . .	. . .
4/7	. . .	. . .	. . .	. . .
4/14	Begin drafting design modifications for the lock mechanism and plan integration of additional sensors and camera into the cabinet body.	Start designing the PCB layout and control circuit to integrate pressure sensors, SD card data, and ensure robust data transfer.	Research efficient computer vision models for Raspberry Pi and identify acceleration methods.	Interface the display with Raspberry Pi, verify hardware connections, and produce basic output.
4/21	Physically modify the lock and cabinet, add sensor mounts, and secure the camera optimally.	Finalize PCB schematic and build a breadboard prototype, testing sensor, camera, and Pi communications.	Refine vision model selection and begin code for inference acceleration on Raspberry Pi.	Develop storage/retrieval logic, integrate with display to show cabinet status.

Continued on next page. . .

Continued from previous page

Week	Niahoaxuan Ruan	Yihong Yang	Xiaotong Cui	Yanxin Lu
4/28	Perform integrated testing of lock, sensors, and camera; adjust physical setup based on results.	Build and test a physical PCB prototype, validate sensor data, SD operations, and inter-component communication.	Optimize and test vision model on Raspberry Pi for required speed and accuracy.	Debug display interface, ensure dynamic status updates (e.g., storage/retrieval, cabinet numbers) respond correctly.
5/5	Final mechanical tweaks to lock and cabinet, confirm sensors and camera are secure and functional.	Integrate PCB with the system, run full-system tests: data flow from sensors to Pi to remote PC and back to lock.	Complete performance tuning of on-Pi vision system, verify real-time recognition and inference.	Final debug of display and storage/retrieval interface, ensure UI accurately reflects system status.
5/12	...	...	...	...
5/19	**Final testing and preparation for the demo!**			

4 Requirements & Verification

4.1 Completeness of Requirements

All high-level requirements are defined with reasonable tolerances and have corresponding verification activities. Specifically, the system must unlock a compartment within $2\,\mathrm{s}$ of scanning a valid campus card, identify returned items with $\geq 95\%$ accuracy, and log every borrow/return event with 100% reliability. No critical requirements were omitted, and every lower-level requirement traces directly back to these objectives.

4.2 Appropriate Verification Procedures

Verification combined standard and custom methods:

- Stopwatch and oscilloscope measurements for card-swipe-to-unlock timing and power-rail stability under load.
- Automated test scripts driving 30 borrow/return cycles (60 events) with database queries to confirm 100% event logging, including simulated network outages to exercise local queuing.
- Iterative return trials (50 items) integrating load-cell and vision-based sensors in a "voting" scheme, logging both weight readings and image classifications to validate item identification accuracy.
- Multimeter validation of electrical parameters (voltage regulator ripple, MOSFET drive levels) and repeated door-sensor open/close trials to verify feedback accuracy.

All procedures yield quantitative, reproducible results and are fully documented in the appendix.

4.3 Quantitative Results

- **Unlock Timing:** Average delay $1.5\,\mathrm{s}$ (worst-case $1.8\,\mathrm{s}$) vs. $2\,\mathrm{s}$ requirement.
- **Item Detection:** $48/50$ correct identifications (96% success rate) vs. $\geq 95\%$ requirement.
- **Event Logging:** $60/60$ borrow/return events logged (100% reliability) with zero omissions, including under simulated network interruptions.
- **Power Stability:** DC-DC regulator output within $\pm 5\%$ tolerance (ripple $< 50\,\mathrm{mV}$) under 0–$2\,\mathrm{A}$ load; battery sustained $\sim 8.5\,\mathrm{h}$ at $400\,\mathrm{mA}$ draw vs. $8\,\mathrm{h}$ requirement.

All measured values fall within or exceed specified tolerances; no requirement failed verification.

5 Conclusion

We have developed a Raspberry Pi–based self-service cabinet for everyday tools—scissors, thermometers and glue sticks—that operates fully autonomously. By combining NFC authentication, load-cell weight sensing, YOLOv5s visual checks and per-item NFC tags, the system guarantees that the exact same item is borrowed and returned.

5.1 Accomplishments

Our cabinet seamlessly integrates NFC-based user authentication, load-cell weight sensing, YOLOv5's computer-vision checks and per-item NFC tags to guarantee that the same tool is borrowed and returned. All hardware components—the NFC reader, weighing sensor, camera and electromagnetic lock—run smoothly on a single Raspberry Pi. The administrator touch-screen interface enables easy addition or removal of users and tool types. In field tests, the vision model delivers end-to-end detections in under 0.5 s with over 98 % accuracy, while the weight sensor and NFC tag combine for near-perfect item verification. A comprehensive local log records every transaction for full traceability.

5.2 Uncertainties

Weight measurements can drift when tools share similar masses, suggesting a need for more precise calibration or signal filtering. Under poor lighting or partial occlusion, the vision system still produces occasional misclassifications. The long-term adhesion and durability of NFC stickers remain untested over thousands of cycles. Finally, the Raspberry Pi's performance under bursty, simultaneous access by multiple users requires further stress testing to ensure responsiveness does not degrade.

5.3 Future Work

We will investigate model-acceleration frameworks such as TensorRT or ONNX Runtime to improve detection speed and robustness. Additional sensing modalities—color or shape analysis, RFID readers—will be explored to strengthen multi-factor verification. A cloud-based dashboard will provide real-time monitoring, usage analytics and automated alerts. We also plan to add reservation and overdue-reminder features, and to conduct larger pilot deployments to collect user feedback for interface and ergonomic refinements.

5.4 Ethical Considerations

All user authentication and transaction records are encrypted at rest, adhering to data-minimization principles. We intend to integrate cryptographic counters and challenge-response protocols to defend against NFC cloning or relay attacks. To ensure inclusive access, we will provide clear guidance on system limitations and offer fallback procedures. Throughout development, we commit to transparent disclosure of potential risks and mitigation strategies, in line with professional engineering ethics.

References

[1] pidoc.cn, *Processors*, https://pidoc.cn/docs/computers/processors/, 2024.

Appendix A Example Appendix

An appendix can go here! Make sure you use the `\label{appendix:a}` above so that you can reference this section in your document.