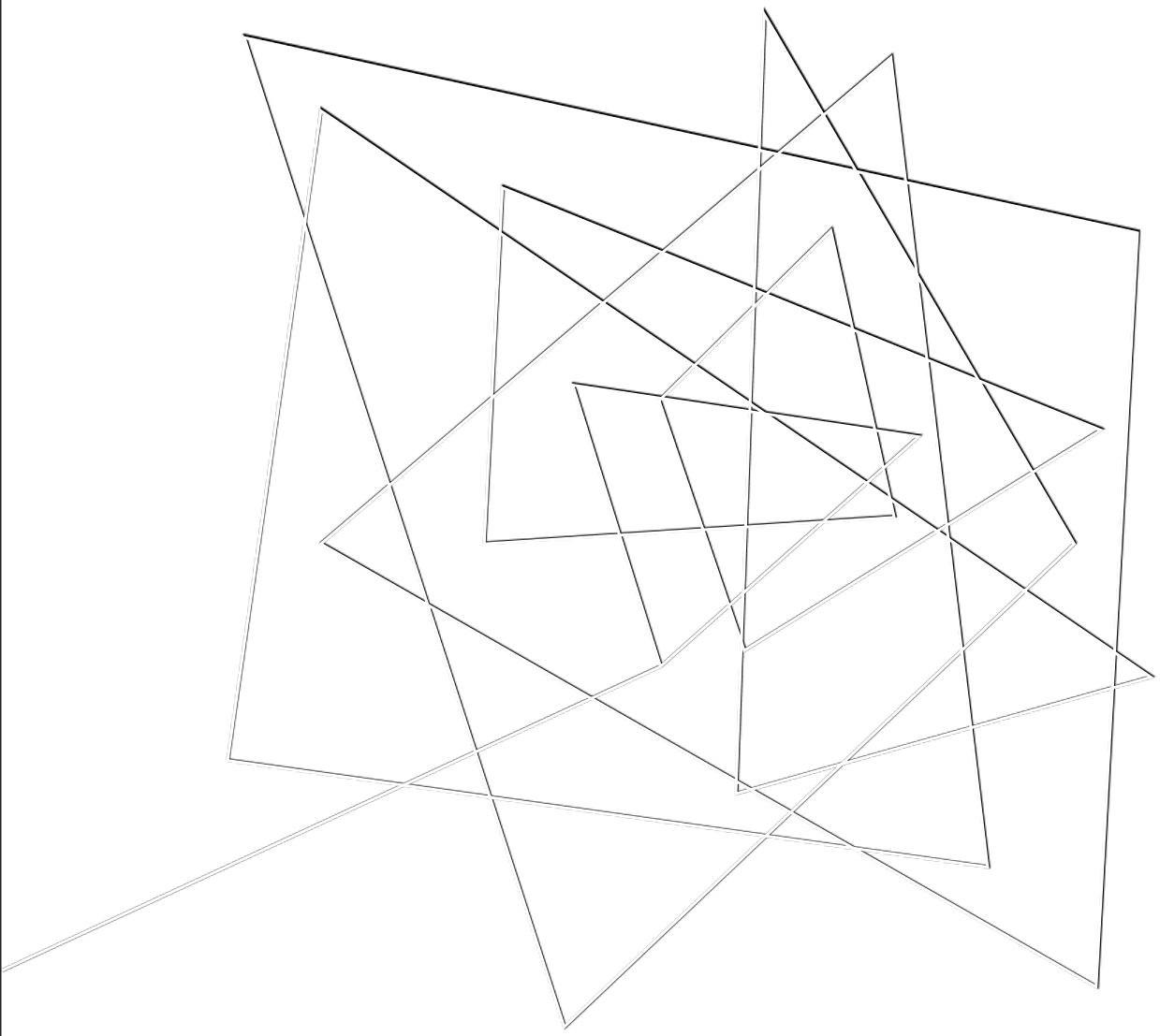


DR. PILL

Automatic Pill Dispenser

TEAM AND RESPONSIBILITIES

Jonathan Cerniaz

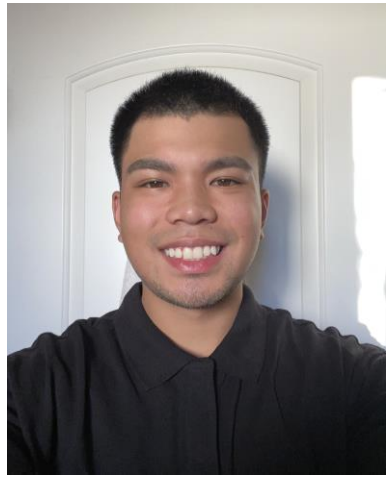
- Circuit
- Power Supply and Regulation
- Fingerprint and Facial Recognition

Jehmel Espiritu

- Software Development of Interface

Jeremy Espiritu

- Software Development of Interface

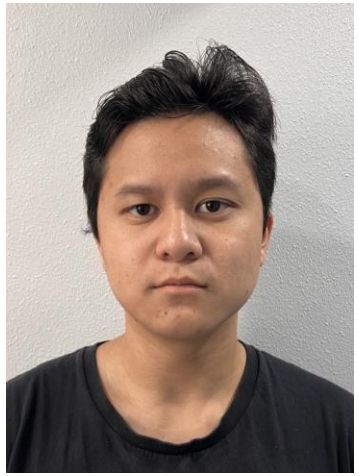


Joseph Guzman

- Enclosure & Framework
- 3D Design and Build
- Motor Testing & Calibration

Afzal Hakim

- Fingerprint and Facial Recognition
- Log tracking and Security

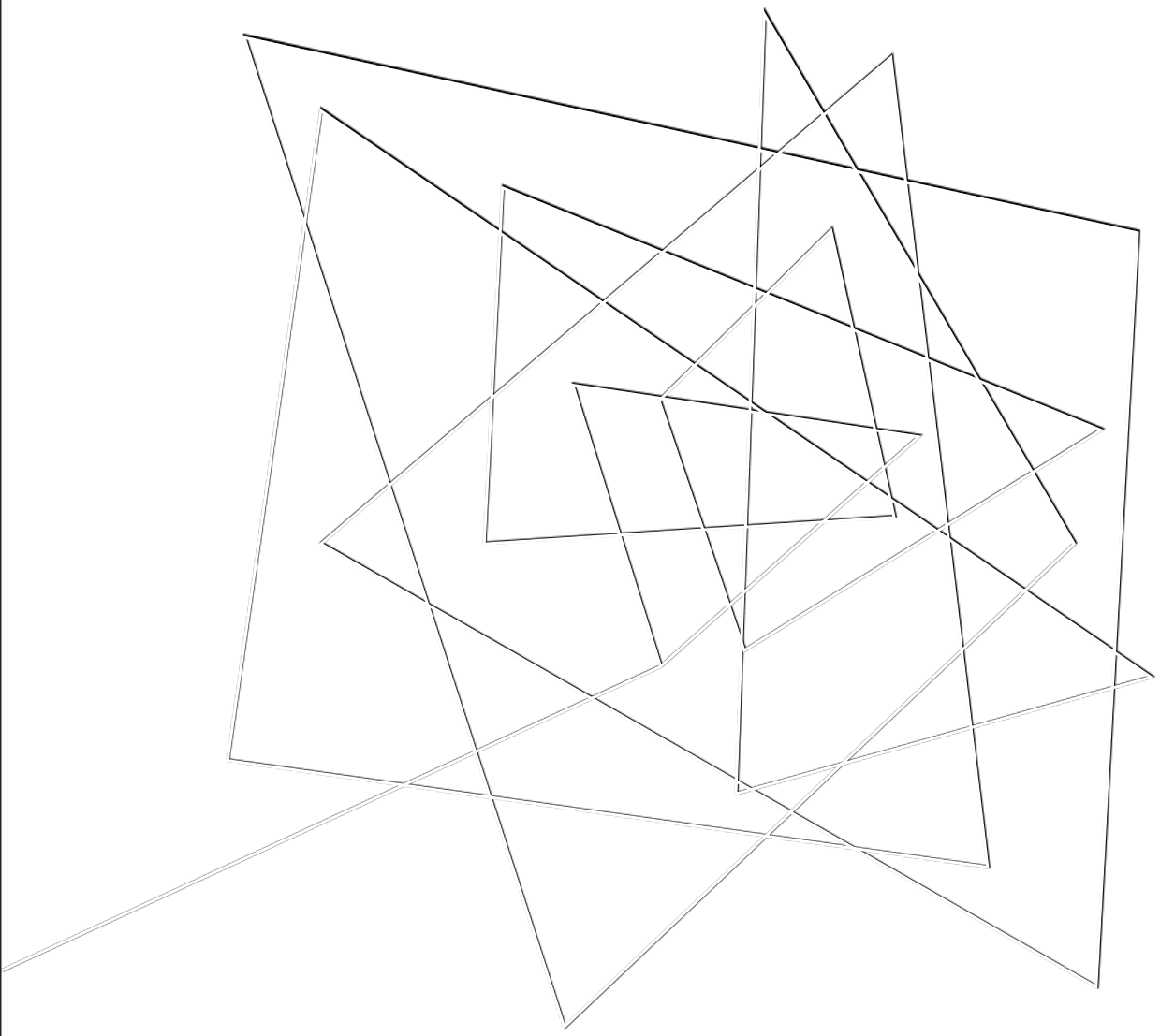


Lee Roger Ordinario

- Pill Storage System and Organization
- Soldering

Executive Summary

Dr. Pill is an automatic pill dispenser that simplifies medication management. Our device aims to facilitate the process by freeing up more time for doctors and allowing patients to take their medication accurately by eliminating the common issue of medication errors and missed dosages. Designated for pharmacies, nursing homes, personal households.



OUR 3 DEMOS

1. Dispense Mechanism & Storage

2. Displaying User Interface

3. Authentication and Board to Board Communication

Final: 'Dr. Pill'
The Automatic Pill Dispenser

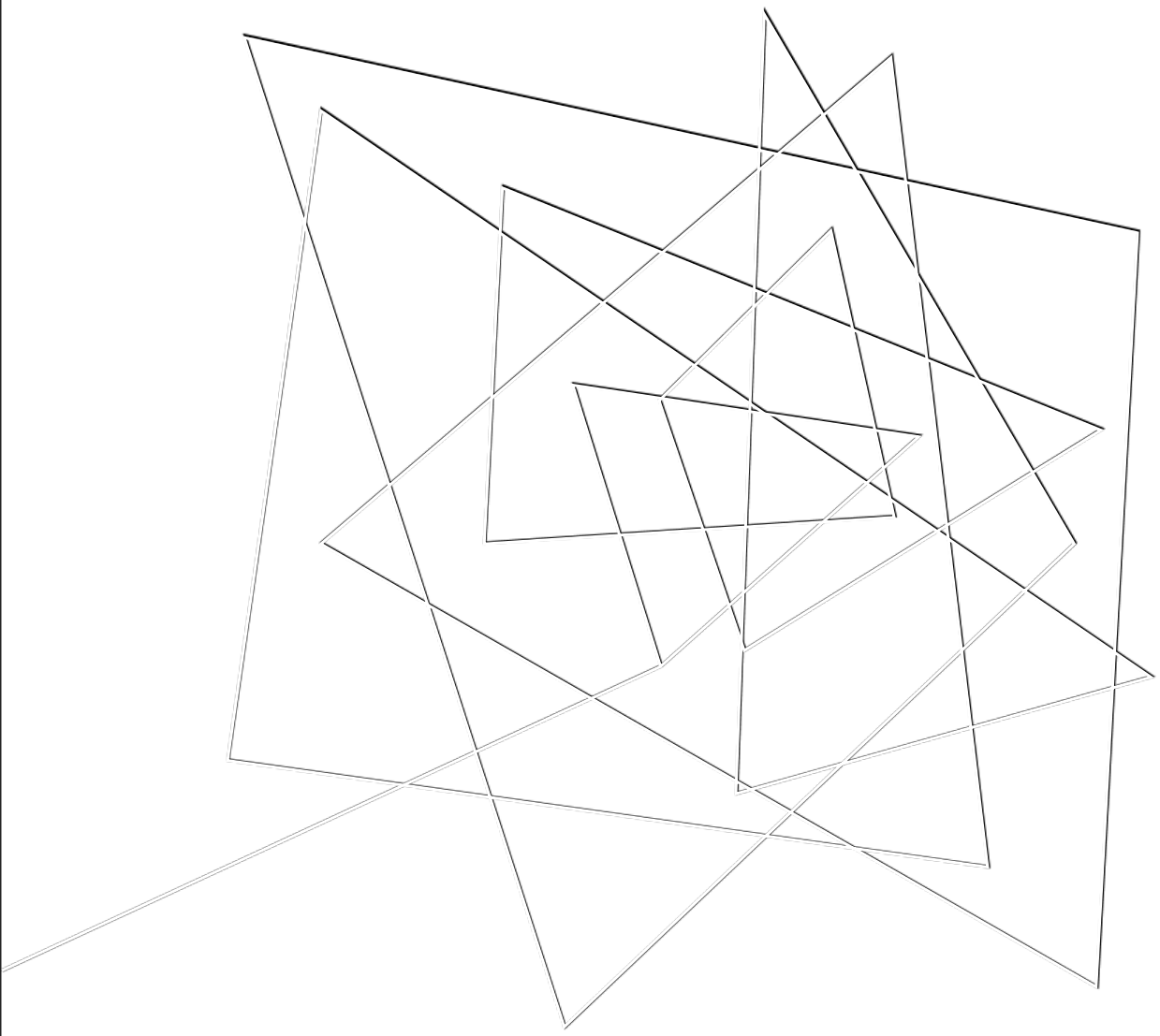
FINAL DEMONSTRATION

'Dr. Pill' The Automatic Pill Dispenser

Requirements: Accumulation of all demos + Real-Time Logging, Add/Delete User, Conveyor Belt

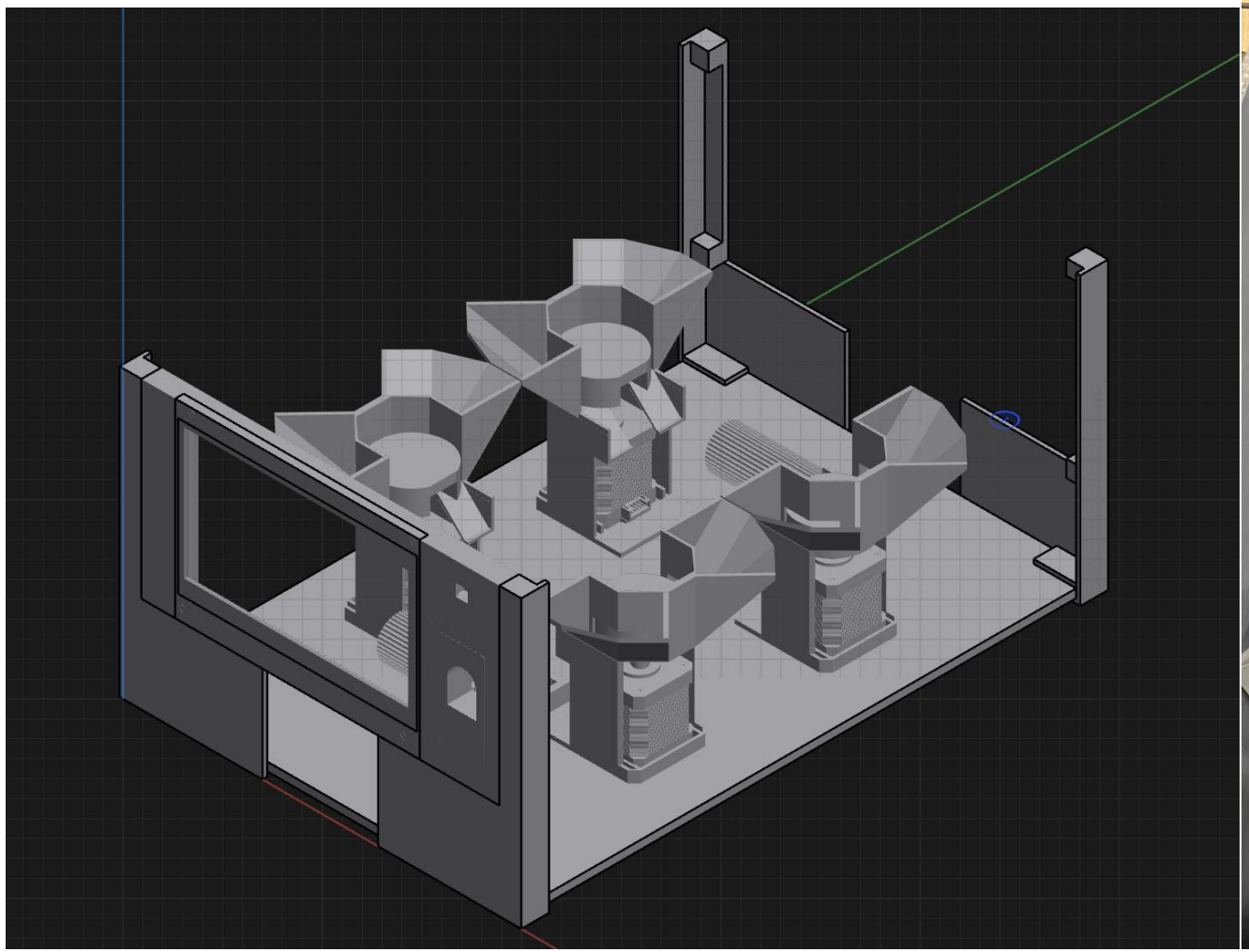
Constraints: Log accuracy, facial and fingerprint recognition accuracy

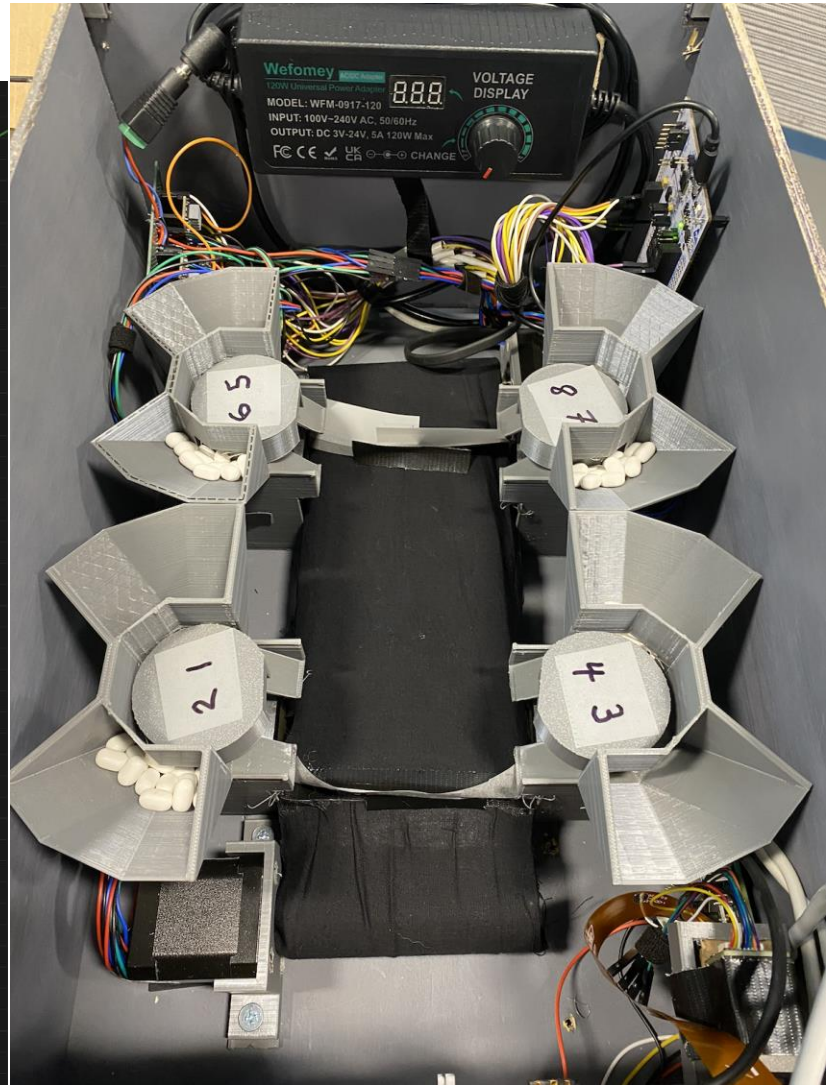
Success: Logs show real-time activity, conveyor belt brings pills to the front, adding/deleting user



DEMO TIME

TECHNICAL DETAILS





TECHNICAL DETAILS

```python
def add_user_pin():
    global current_popup

    # Destroy existing popup if one is already open
    if current_popup:
        current_popup.destroy()

    # Create a new popup box for entering PIN information
    current_popup = Box(app, layout="vertical", width=400, height=600, align="top")

    # Prompt label asking for the user's name
    prompt_label = Text(current_popup, text="Enter Name for PIN:", size=text_size)

    # TextBox for user to enter their name
    name_box = TextBox(current_popup, width=30)

    # Hidden TextBox for entering PIN (initially invisible)
    pin_box = TextBox(current_popup, width=30, hide_text=True)
    pin_box.visible = False

    # Box to hold on-screen keyboard layout (initially empty)
    keyboard = Box(current_popup, layout="vertical", align="top")

    # Box to hold the numeric pin pad (initially hidden)
    pin_pad = Box(current_popup, layout="grid", visible=False)
```

TECHNICAL DETAILS

```
def add_user_fingerprint():
    global current_popup

    # Check if the fingerprint sensor is working properly
    if finger.read_sysparam() != adafruit_fingerprint.OK:
        error("Error", "Failed to get system parameters")
        return

    # Check how many fingerprint templates already exist
    if finger.count_templates() != adafruit_fingerprint.OK:
        error("Error", "Failed to count existing fingerprints")
        return

    # Close any currently open popup window
    if current_popup is not None:
        current_popup.destroy()

    # Create a new popup box for entering user name before fingerprint enrollment
    current_popup = Box(app, layout="vertical", width=400, height=600, align="top")

    # Label prompting user to enter their name
    Text(current_popup, text="Enter User Name for Fingerprint:", size=text_size)

    # TextBox for entering the user name
    name_box = TextBox(current_popup, width=30)
```

TECHNICAL DETAILS

```
def add_user_facial_recognition():
    # Hide all other screens to prevent UI conflicts
    hide_all_screens()

    global current_popup

    # Close the current popup if one exists
    if current_popup:
        current_popup.destroy()

    # Create a new popup box for facial recognition user name entry
    current_popup = Box(app, layout="vertical", width=400, height=600, align="top")

    # Label prompting user to enter their name for facial recognition
    Text(current_popup, text="Enter New User Name:", size=text_size)

    # TextBox for entering the user name
    name_box = TextBox(current_popup, width=30)
```

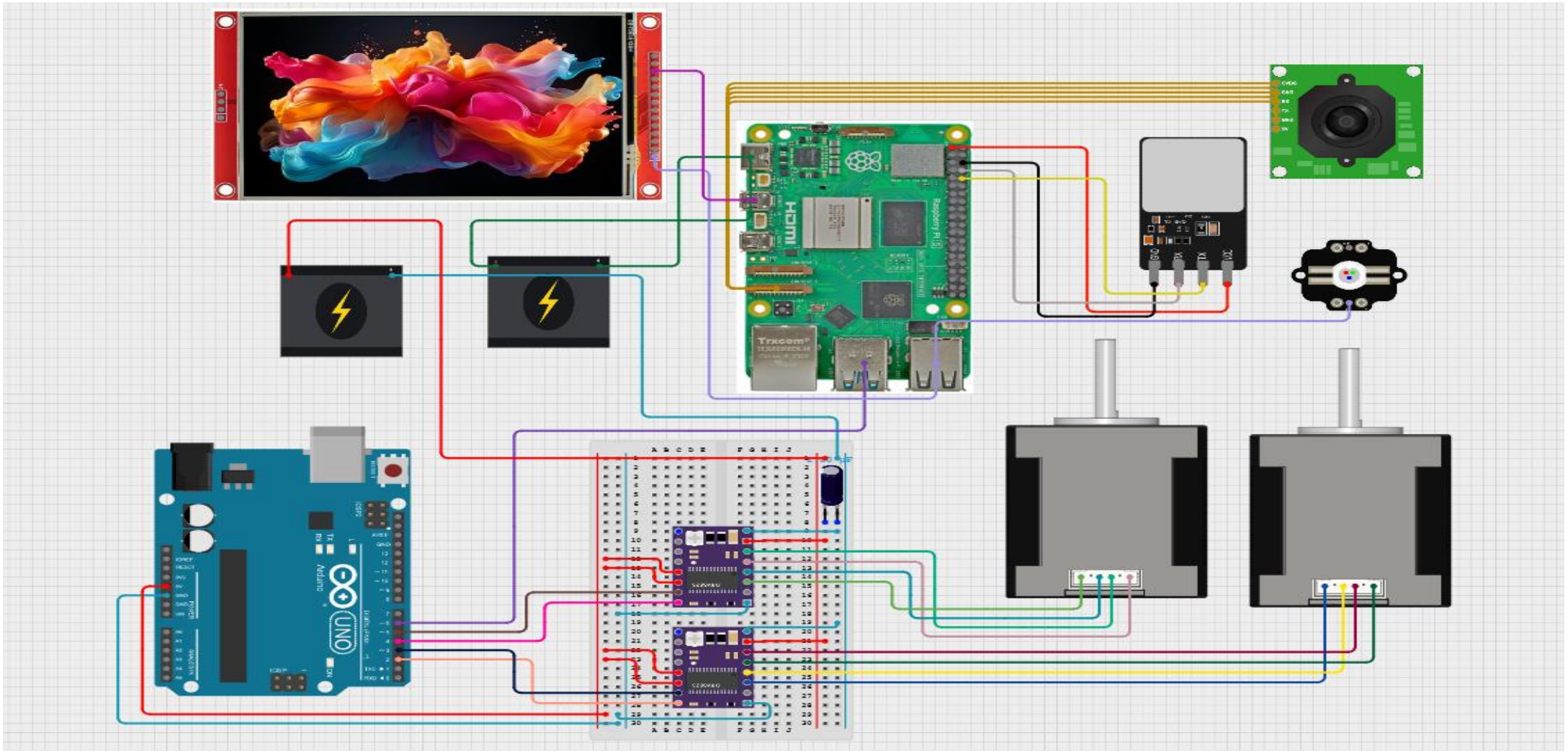
TECHNICAL DETAILS

```
current_method = "Unknown"

db = mysql.connector.connect(
    host="10.0.0.42",
    user="useradmin",
    password="12345678",
    database="fingerprint_logs"
)
cursor = db.cursor()

current_method = "Fingerprint"

sql = "INSERT INTO login_records (user_id, confidence, status, method) VALUES (%s, %s, %s, %s)"
values = (uid, confidence, "Success", current_method)

cursor.execute(sql, values)
db.commit()

print(f"[DEBUG] Fingerprint ID: {uid}, Name: {name}, Role: {role}, Confidence: {confidence}")
```

TECHNICAL DETAILS

id	user_id	login_time	confidence	status	method	pill_dispensed
93	0	2025-05-05 18:47:26	0	Failed	Numpad	NULL
92	0	2025-05-05 18:46:09	0	Success	Numpad	NULL
91	0	2025-05-05 18:45:27	0	Failed	Fingerp...	NULL
90	0	2025-05-05 18:40:38	0	Failed	Fingerp...	NULL
89	0	2025-05-05 18:40:21	0	Success	Numpad	NULL
88	0	2025-05-05 18:37:56	0	Failed	Fingerp...	NULL
87	0	2025-05-05 18:35:51	0	Success	Numpad	NULL
86	0	2025-05-05 18:34:47	0	Success	Numpad	Pill 2 x1
85	0	2025-05-05 18:34:29	0	Success	Numpad	NULL
84	0	2025-05-05 18:33:37	0	Success	Numpad	NULL
83	0	2025-05-05 18:33:26	0	Failed	Fingerp...	NULL

TECHNICAL DETAILS



CHALLENGES AND HOW WE SOLVED IT

- 3D printing issues (clogging, broken nozzel, poor print quality, etc.)
- Wi-fi connectivity (Mango Wi-fi router)
- Dispensing (Object recognition, jamming, pill size, and unpredictability)
- GUI (Crashing, quality of interface, and multiple menus)

REFLECTION

- We all learned a lot from this class like learning how 3D print, creating logs with MySQL, Fingerprint and Facial Recognition, creating a custom GUI from scratch.

- We also learned the fundamentals of working as a team (team communication, time management, planning, etc.)

- Since we were only missing the object recognition that leaves us with 95% of the project being complete. If we were to continue this project there it a lot we could improve on in the future (Cleaner GUI, object recognition, better enclosure, etc.)

THANK YOU