



CSE 476/575

Mobile Communication Networks

AI-Based Smart Door Lock

Project Report

Berke Yılmaz

Furkan Ayan

Furkan Taşkın

Project Objective

Our goal was to change old door security. We used face checking to make getting in easier and safer. We wanted to stop using keys, cards, or number codes. These items can be lost or shared. Instead, people can get in just by showing their face.

We did more than just control who enters. We made a full system to manage security. When allowed people come in, the system notes the exact time. If someone who is not allowed tries to get in, the system takes their picture. It then quickly sends a message to the administrator's phone.

We also developed a mobile app. This app gives the administrator full control. With the app, they can:

- Add new people and their face information.
- See a list of who came in and when.
- Quickly change who is allowed to come in.

Our system finds a good balance. It is secure, easy to use, and respects people's privacy. We check faces right at the door. This means less private information needs to be sent to other places. At the same time, administrators can still manage everything from one main place using safe connections.

Finally, we have shown that new AI tools can solve common security problems in a useful way. We built a system that is easy to understand and feels new. This is what people now expect from smart building tools.

Links

Github Repositories:

Backend: <https://github.com/mobilecommunicationsnetwork/backend>

Raspberry PI: <https://github.com/mobilecommunicationsnetwork/rpifacedetection>

Mobile: <https://github.com/mobilecommunicationsnetwork/mobile-side>

Youtube link to our demonstration of Project:

<https://www.youtube.com/watch?v=com9eOA3wi8>

Project Implementation

We've created a complete door security solution that brings together the best of hardware design, AI technology, and practical security thinking.

Our implementation begins with custom-designing and 3D-printing parts that copy standard door locks. We wanted something that not only functions reliably but also look great and reusable aswell—after all, this is going at the entrance to homes in our concept project.

The main processor of our operation is a Raspberry Pi 5 paired with a specialized Hailo8L AI accelerator. This combination gives us enough processing power for working for all the time. The Hailo8L is particularly used to run face recognition AI right at the door with literally no waiting time.

Here's how our system works in practice: The Raspberry Pi camera constantly monitors the doorway area, capturing video frames. These images go directly to our first AI model Scrfd running on the Hailo8L chip, which scans constantly for human faces.

When a face is detected, our system focuses on just that portion of the image and passes it to our second AI program which is Arcface. This specialized software converts the face into what's essentially a mathematical "fingerprint"—a unique pattern that captures the distinctive features that make each person's face different from everyone else's, which is called vector data, and then we convert that vector data into an array of floating point numbers to store it in our backend.

We then compare this vector data against our secure database of authorized users. Our comparison algorithm calculates a confidence percentage that tells us how likely this person is someone who should have access. If they score above our predetermined threshold which is around 0.6, we activate the servo motor we've installed, which turns the lock mechanism and grants entry.

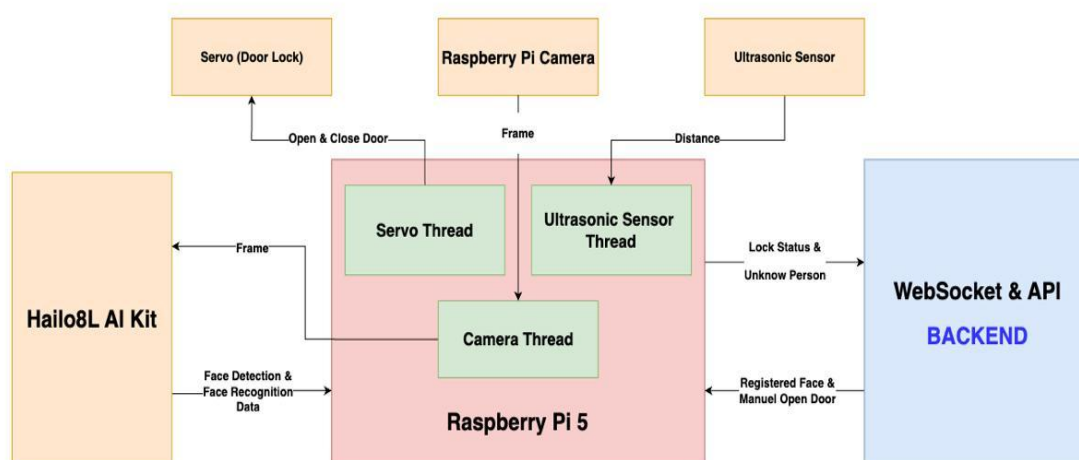
For people who don't match anyone in our system (unknown user as in possible intruders), the door stays locked, and we automatically save their image in our database for the actual registered user to review. Proposing a system to grant or deny access to anyone to your home with the use of AI technology and 3D printed pieces combined with hardware such as raspberry pi that is connected via backend to an easy to use mobile application in all platforms was our imagination.

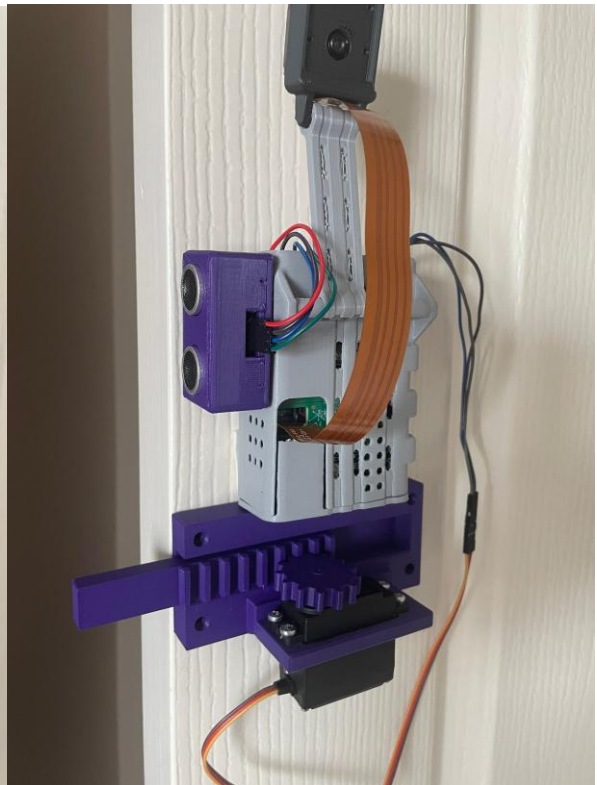
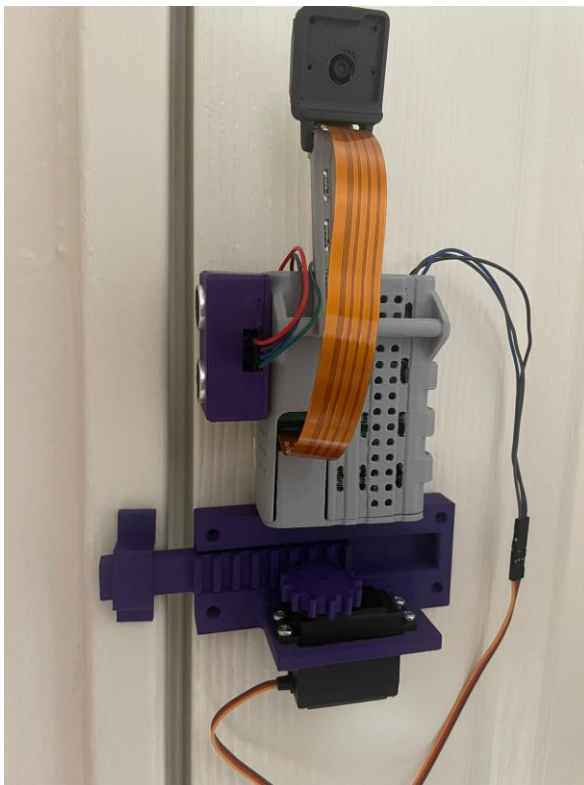
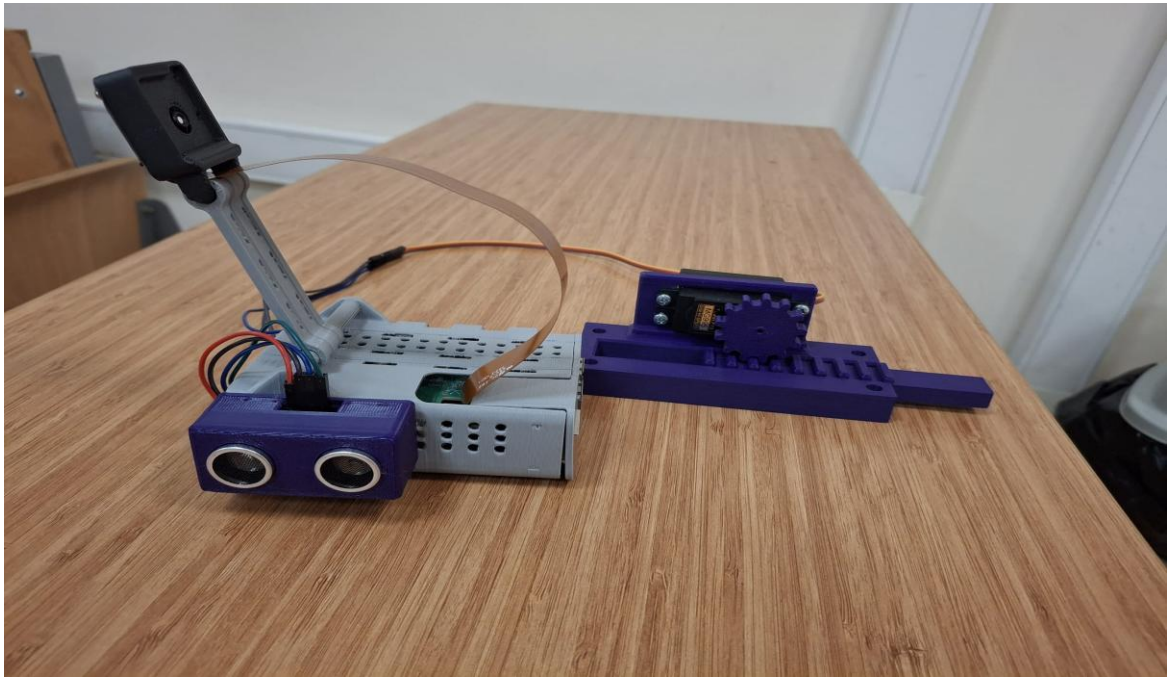
Safety was a major consideration too. We integrated an ultrasonic sensor that constantly monitors whether the door is physically open or closed. If someone has been granted access and the door is still open, the system keeps the lock disengaged to prevent it from accidentally relocking while someone's using the doorway. Once the door closes properly, our system waits 10 seconds and then automatically secures the lock to make sure if the actual physical door is closed the lock is closed as well.

The whole system is tied together with our software design. We built the backend using Django REST API with Python, making it both secure and scalable. For user interaction, we developed our mobile app using React Native that works smoothly on both iOS and Android devices.

One of our additions was implementing WebSocket connections throughout the system. This creates instant, two-way communication between the door hardware, our server, and the mobile app. When someone approaches the door, administrators can receive notifications in real-time, and any changes to access permissions take effect immediately. The WebSocket connections also work as in a way to prevent race conditions in a way, so that it controls mobile commands with automated actions from raspberry pi.

System Architecture Diagram:





WebSocket request from Mobile to backend to broadcast to raspberry pi the new face to get access:

```
elif message_type == 'face_recognition_request':
    face_image_base64 = text_data_json.get('face_image_base64')
    request_id = text_data_json.get('request_id', str(uuid.uuid4()))
    name = text_data_json.get('name')

    if not face_image_base64:
        await self.send(text_data=json.dumps({
            'type': 'error',
            'message': 'face_image_base64 field is required',
            'timestamp': str(timezone.now())
        }))
        return
    if ';base64,' in face_image_base64:
        base64_data = face_image_base64.split(';base64,')[1]
    else:
        base64_data = face_image_base64

    self.active_requests[request_id] = {
        'source_client_id': self.client_id,
        'timestamp': str(timezone.now())
    }

    await self.channel_layer.group_send(
        self.door_group_name,
        {
            'type': 'broadcast_face',
            'message_type': 'face_recognition_request',
            'name': name,
            'face_image_base64': base64_data,
            'request_id': request_id,
            'source_client_id': self.client_id,
            'timestamp': str(timezone.now())
        }
    )
    websocket_logger.info(f"Face recognition request from client [{self.client_id}] broadcasted to group {self.door_group_name}")
    # İstenen face_recognition_result mesajını beklemeden direk yanıt döndür,
    # böylece client beklemeden devam edebilir ve bağlantıyı kapatabilir
    result_message = {
        'type': 'face_recognition_result',
        'result': 'in_progress',
        'name': name,
        'request_id': request_id,
        'timestamp': str(timezone.now()),
        'message': 'Face recognition request is being processed'
    }
    await self.send(text_data=json.dumps(result_message))
    websocket_logger.info(f"Sent in_progress response for request {request_id} to client [{self.client_id}]")
```

- Validates that the face image data is present and properly formatted
- Generates a unique request ID if one wasn't provided
- Stores the request details in an 'active_requests' dictionary to track it
- Forwards the request to all connected devices in the door group (likely including the door controller with the camera)
- Immediately responds to the client with an "in_progress" message

MOBILE APPLICATION

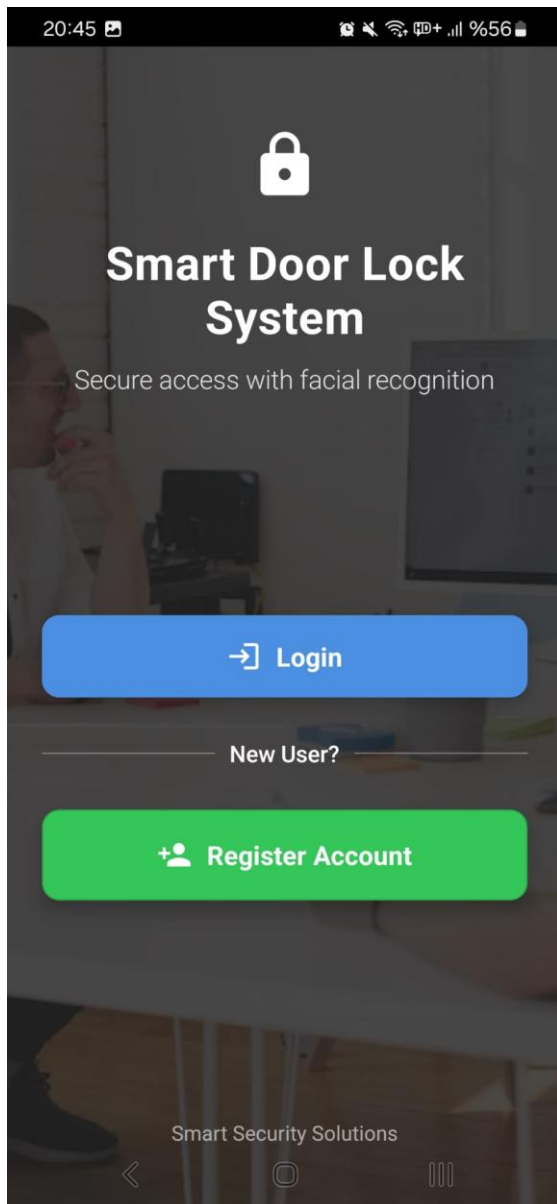
This mobile app helps you use your smart door lock. When you open it, the first screen calls the system Smart Door Lock System. It says the system gives safe access by checking faces. From this screen, you can log in. Or you can make a new account. To log in, use your email and password. To make a new account, give your full name. Also give your email address and a new password.

After you log in, you can control the lock. This is key if you are an admin. A good tool is Register Face. This adds a person's face to the lock system. An admin can see all users who signed up. This list shows their names, like Furkan Ayan and Furkan Taşkın. It also shows their face pictures and sign-up dates. You can remove people from this list. There is a delete button for this. If no faces are added, it says no users were found.

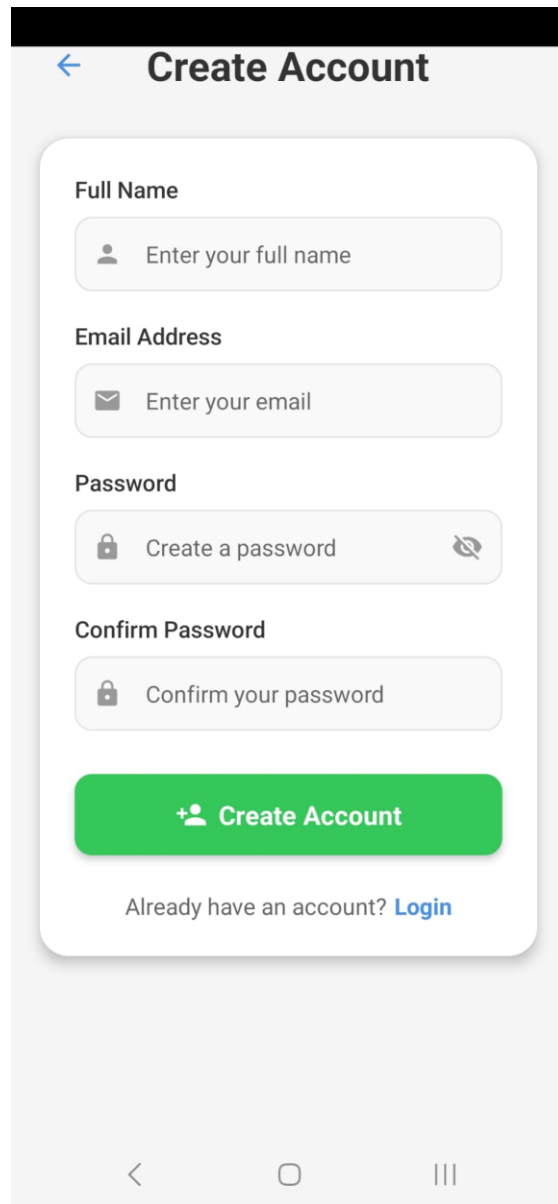
The app has Smart Door Control. This lets you control the lock from far away. You can see if the door is shown as closed. You can also see if its status is Locked. A simple button opens the door when you need it. The app also logs unknown actions for safety. This log shows if an unknown person tried to enter. It includes their picture and the time of the attempt. It also shows details like an IP address.

In sum, this app puts all main lock controls in one place. It helps you add new people. It adds their faces for the lock. It shows you these faces. You can open the door from far off. And you can see who tried to get in.

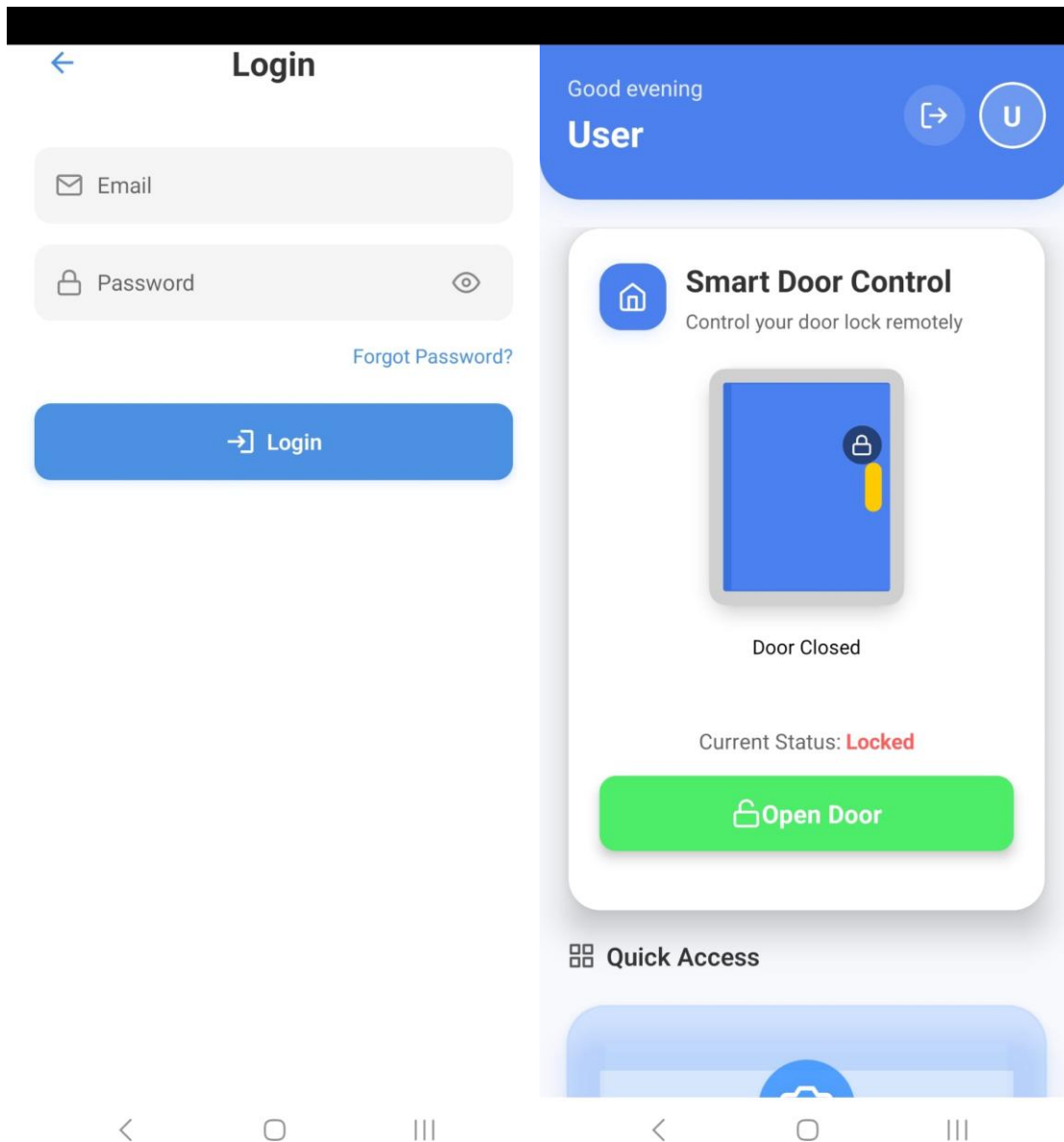
Mobile Application Screenshots



Opening page

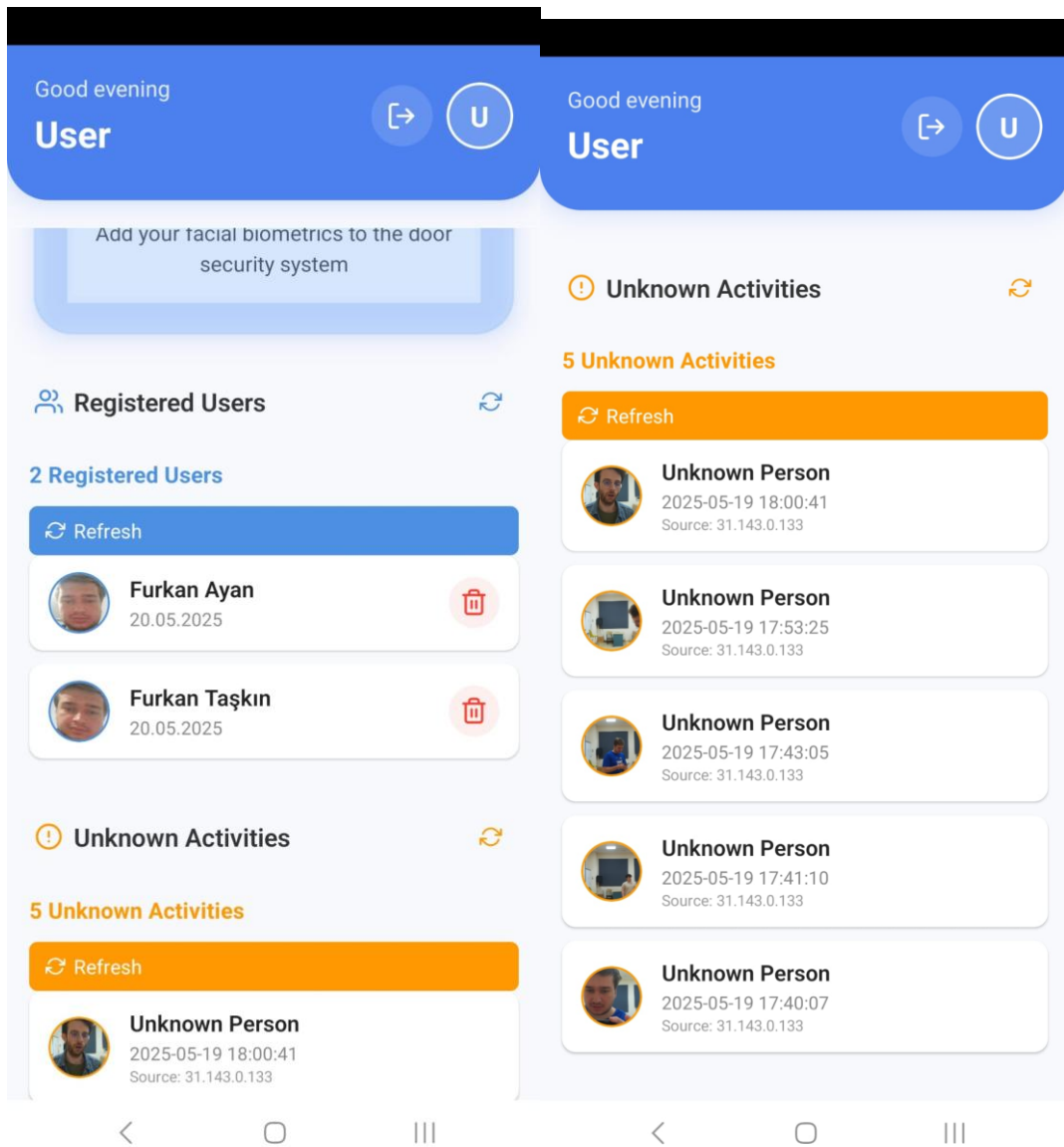


Register Page



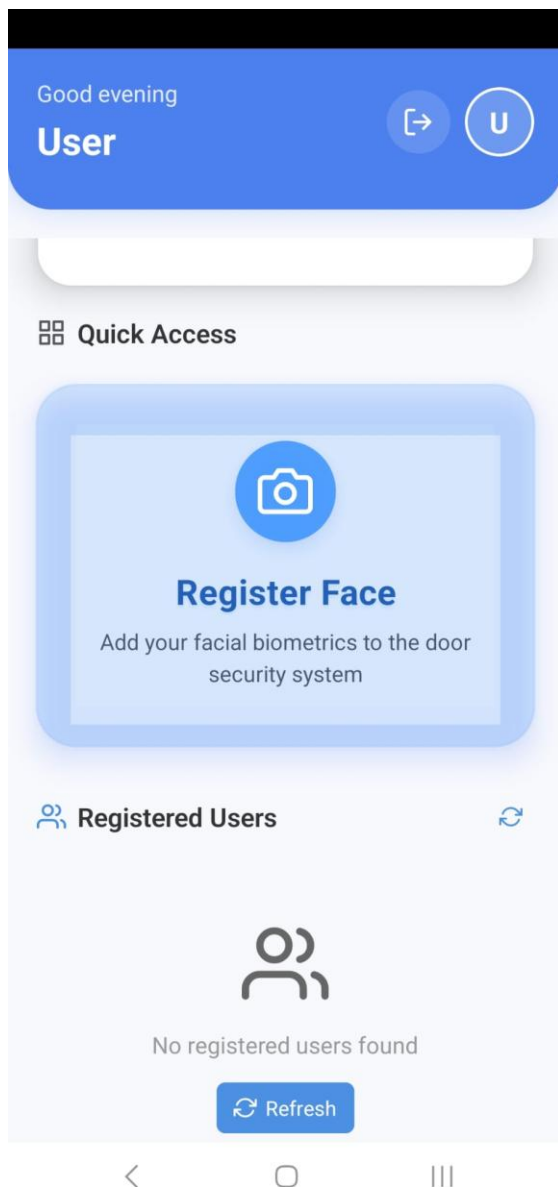
Login Page

Opening the Door



Registered Users

Unknown Users



Register Face