Bachelor Thesis

“Development of attendance tracking system using computer vision based AI on an embedded system”

Anmol Singh, 24244

Supervised by:

Prof. Dr. Ronny Hartanto – 1st Examiner
Prof. Dr. Matthias Krauledat – 2nd Examiner

Presentation Date: 03.11.2022

HOCHSCHULE RHEIN-WAAL
Rhine-Waal University of Applied Sciences

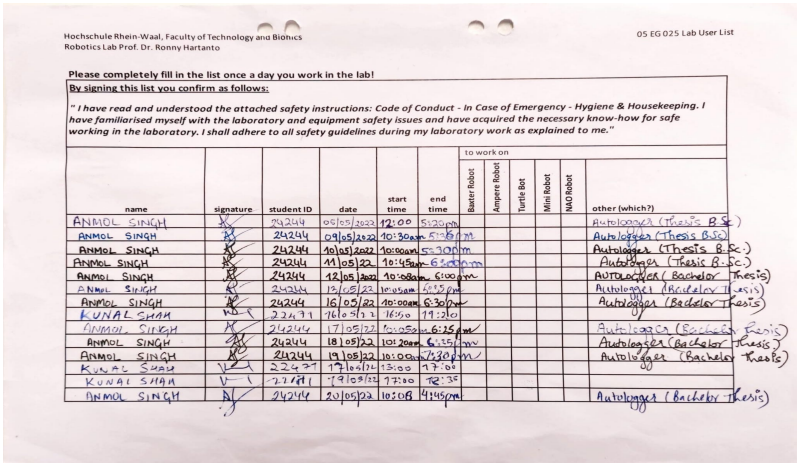
Colloquium Agenda

- **Thesis Objective**
- **Background**
- **Hardware**
- **Implementation**
- **Optimisation**
- **Results**
- **Conclusion**
    1. **Limitations**
    2. **Future Development**
- **Open Questions and Practical Demonstration of the Device**

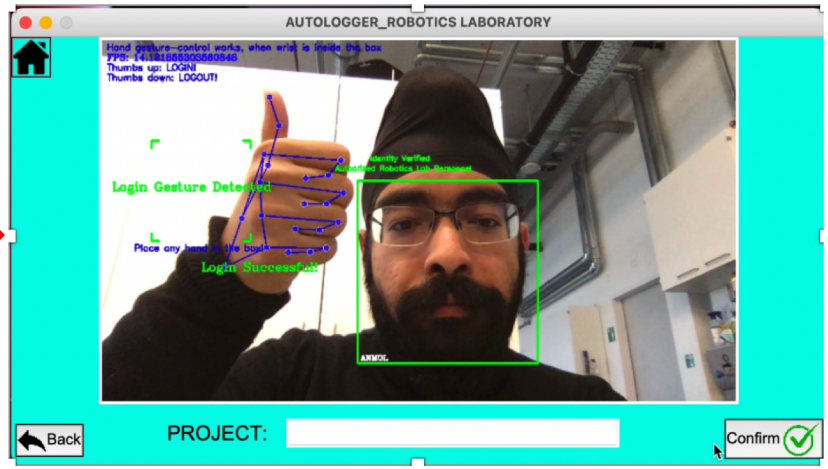
Thesis Objective

- Automate the attendance monitoring system which is more secure than the conventional method.
- Get rid of pen and paper based attendance user list.
- To log in/out of the laboratory with “as contact free as possible” technique.
- A complete software able to run smoothly using an embedded system + camera + touchscreen.
- Long term objective: Create a smart high tech lab, facilitating remote control.

Attendance Tracking (Old)



Attendance Tracking (New)



Software executed and videos recorded on: MacBook
Attendance userlist Credits: Robotics Laboratory

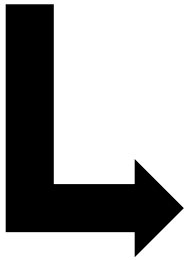
HOCHSCHULE RHEIN-WAAL
Rhine-Waal University of Applied Sciences

Background

Face Detection

Face Recognition

Gesture Recognition



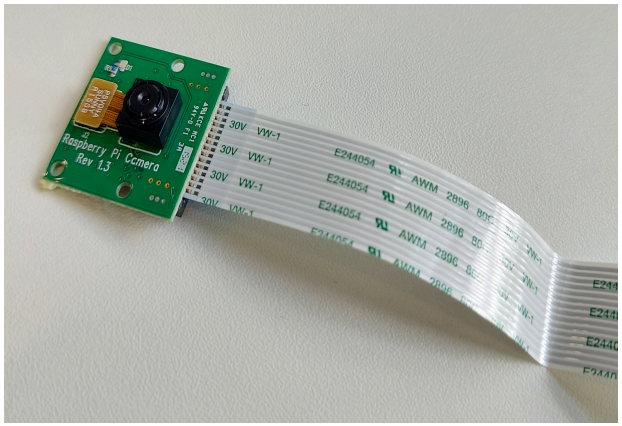
Palm Detection

Hand Landmark Detection

Hardware

Raspberry Pi model 4B

Raspberry Pi Touchscreen



Raspberry Pi Camera v1.3

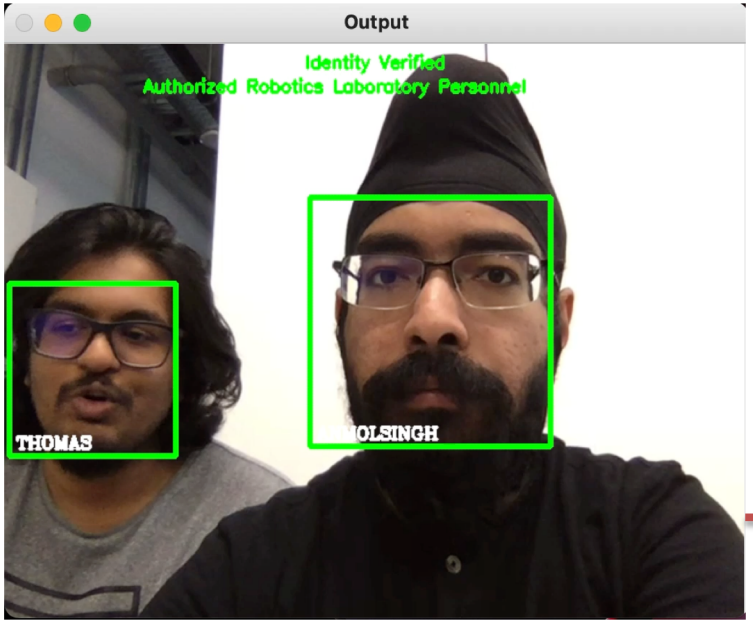
Other available options in lab:

- RPi 3B
- Jetson Nano 4GB

But they were not used!

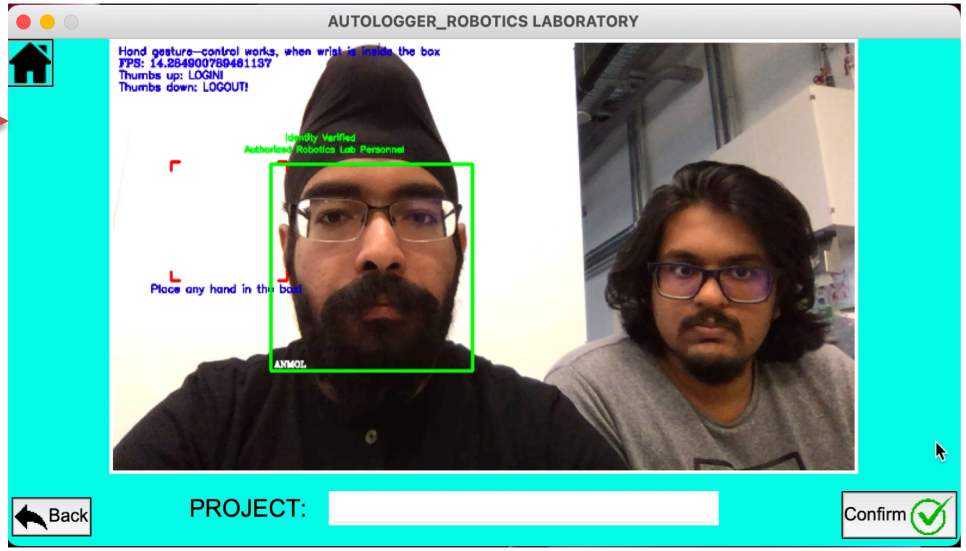
Implementation

Face Detection and Face Recognition



Initial output of face detection
And face recognition model

Final output of face detection
And face recognition model

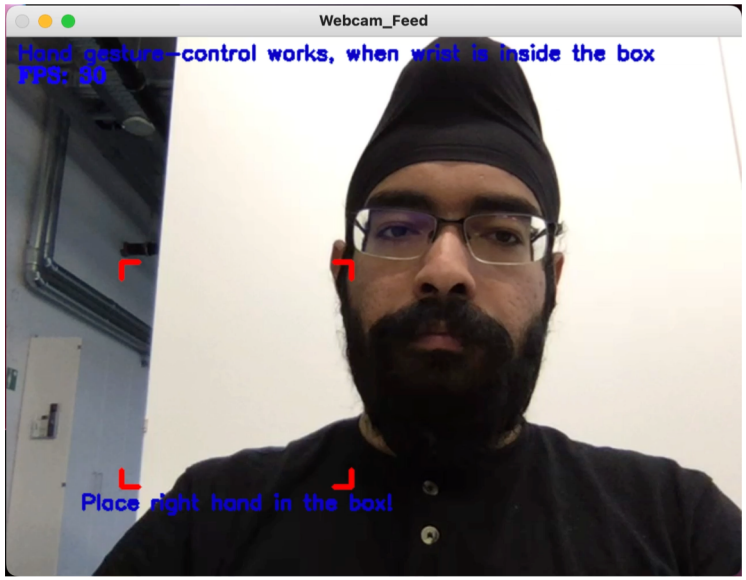


Logic: Highest area enclosed by a bounding box is processed.

Software executed and videos recorded on: MacBook
Video Credits: Mr. Mamen Thomas Chembakasseril

Implementation

Gesture Recognition

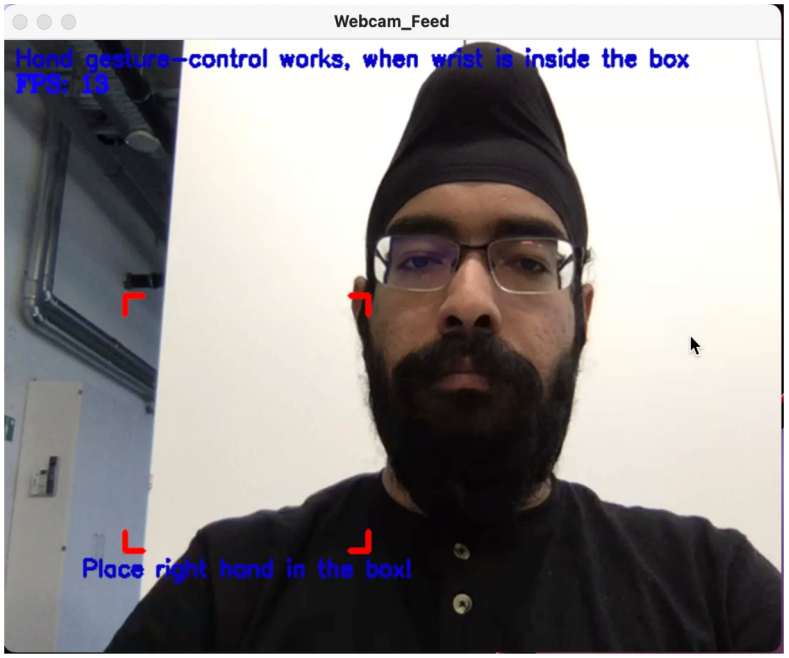


Recognized working gestures

Model predicts all hands but in this case one hand is taken into consideration.

Gesture recognized only inside the box.

Other gestures don't work

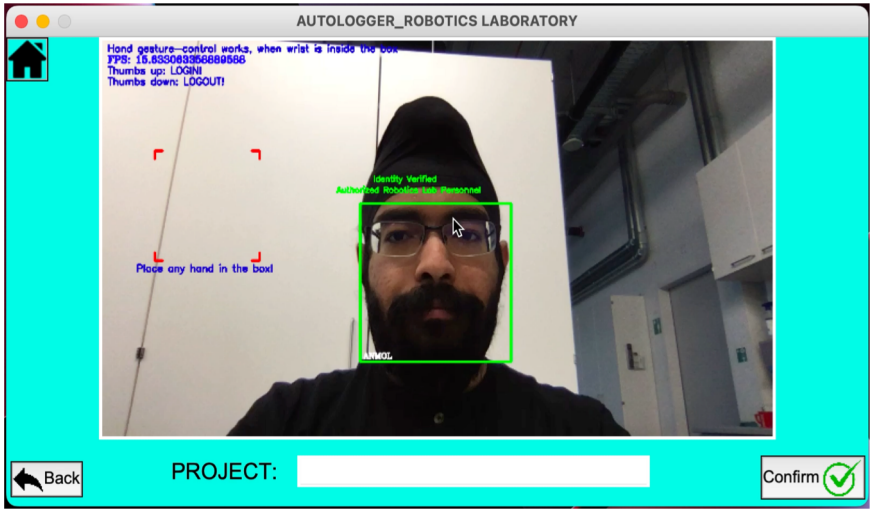


Login: Thumbs-Up
Logout: Thumbs-Down

The idea of implementation for login is that the tip of the thumb should always be above the rest of the detected hand key points while the opposite in logout gesture.

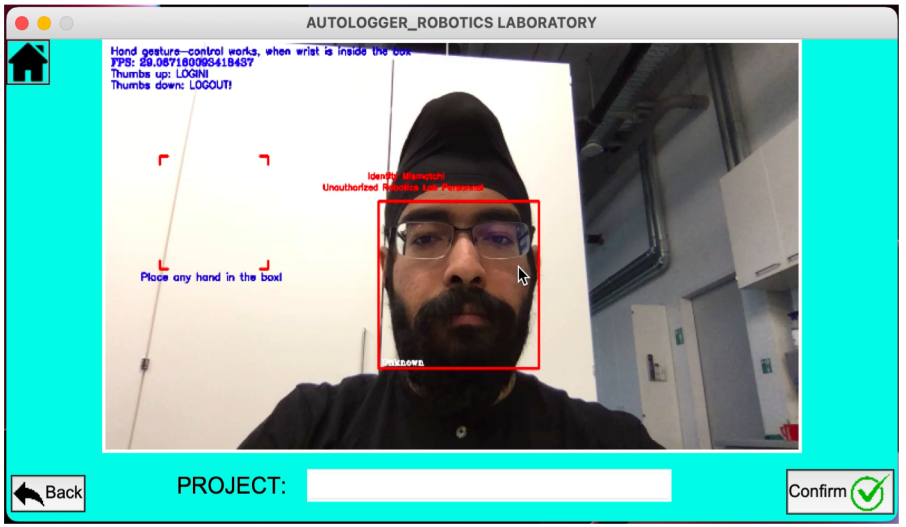
Software executed and videos recorded on: MacBook

Implementation



Gesture recognition works only after face authentication.

This video shows face not recognized, hence gesture doesn’t work.

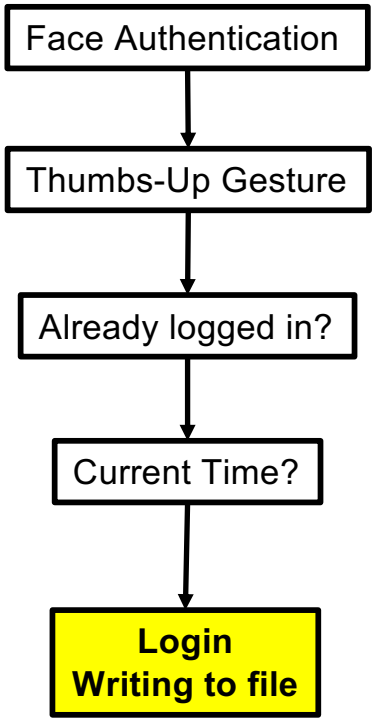


Software executed and videos recorded on: MacBook

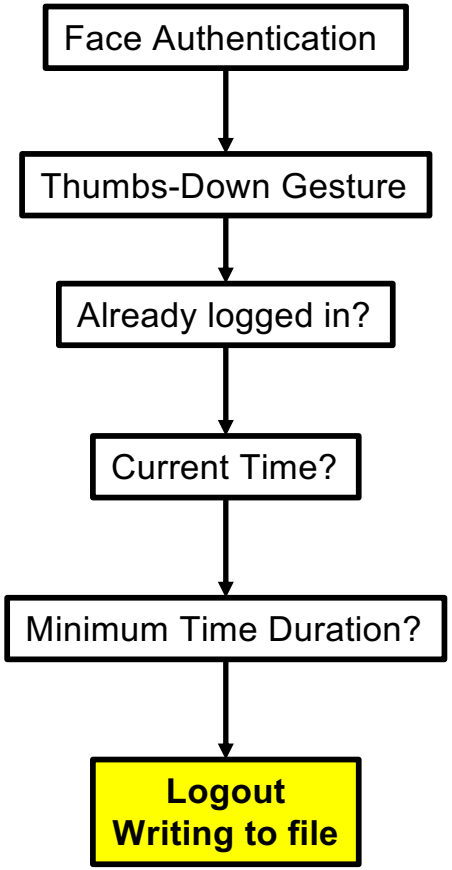
Implementation

Condition Check:

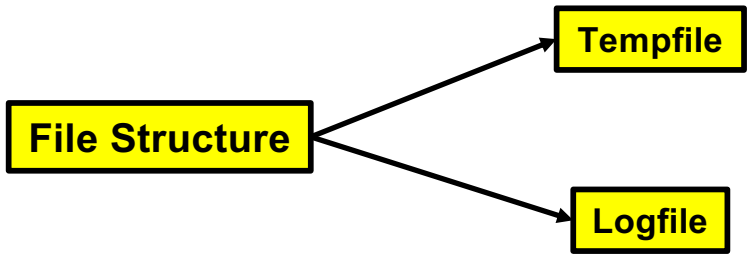
User Login



User Logout



Implementation



Storage on a daily basis (Actions: Read, Modify and Append)

Permanent Storage (Actions: Only Append)

File Writing Format: DATE: LOGIN STATUS, NAME, TIME STAMP, PROJECT

How saving the attendance record work?

Single Login/Logout Mode

Login: Tempfile and logfile are written with a predefined format

Logout: Logfile is written with the predefined format but the particular entry in the tempfile is rewritten as:

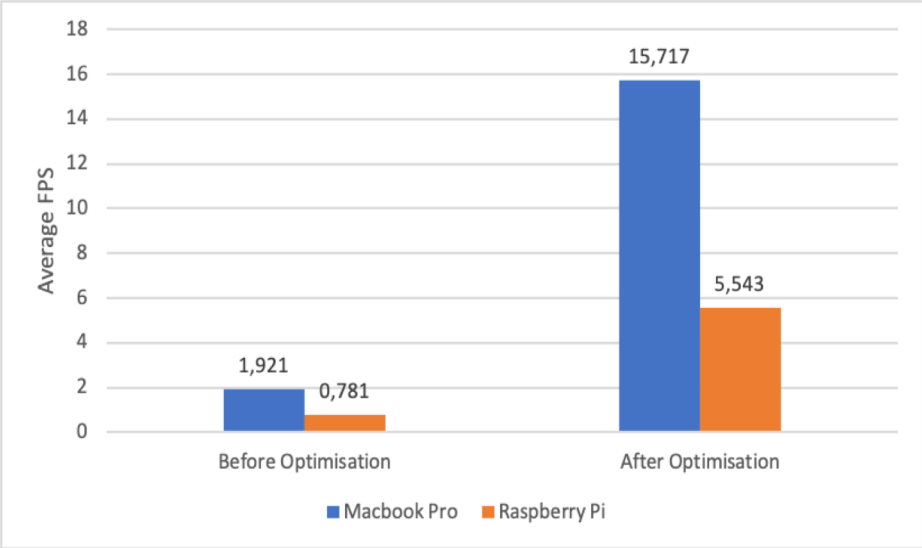
DATE: NAME

Multiple Login/Logout Mode

Login: Tempfile and logfile are written with a predefined format

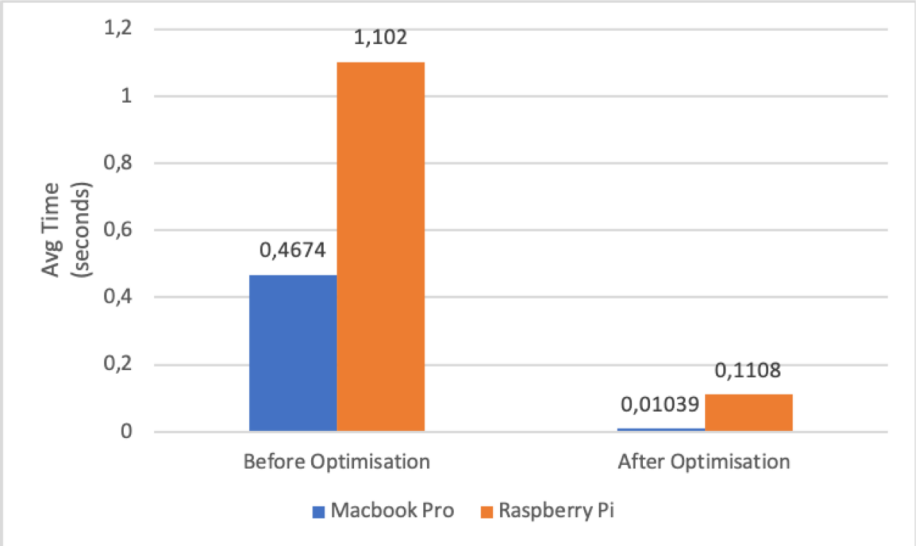
Logout: Logfile is written with the predefined format but the particular entry in tempfile is erased off.

Optimisation



Performance Analysis Results

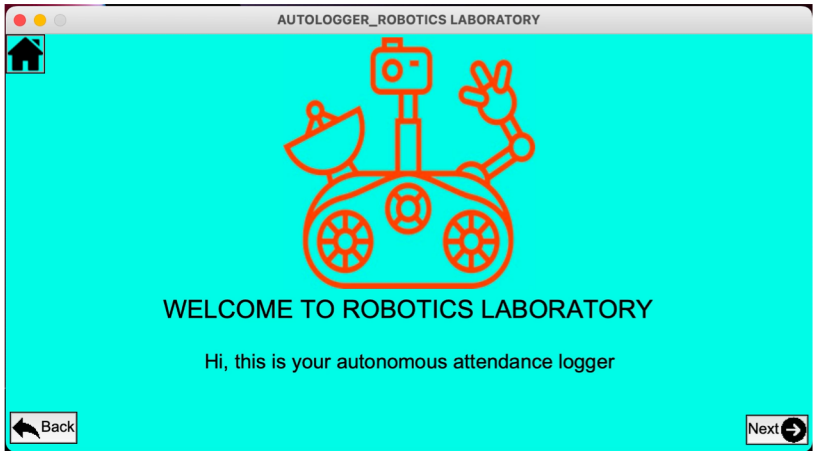
Time Analysis Results



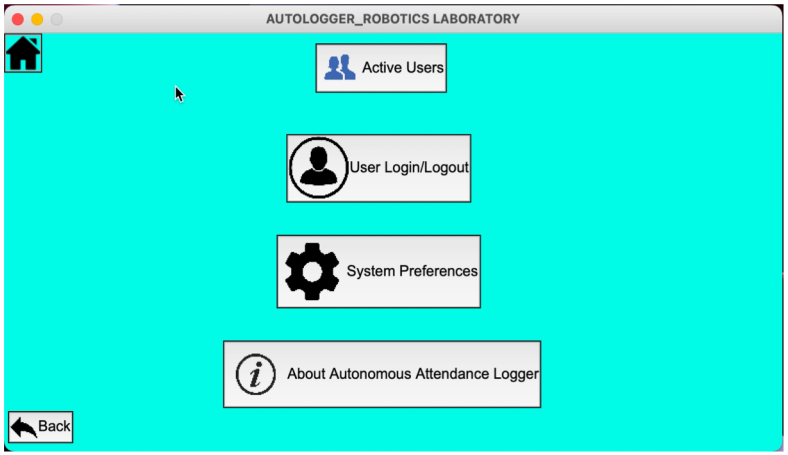
HOCHSCHULE RHEIN-WAAL
Rhine-Waal University of Applied Sciences

Results

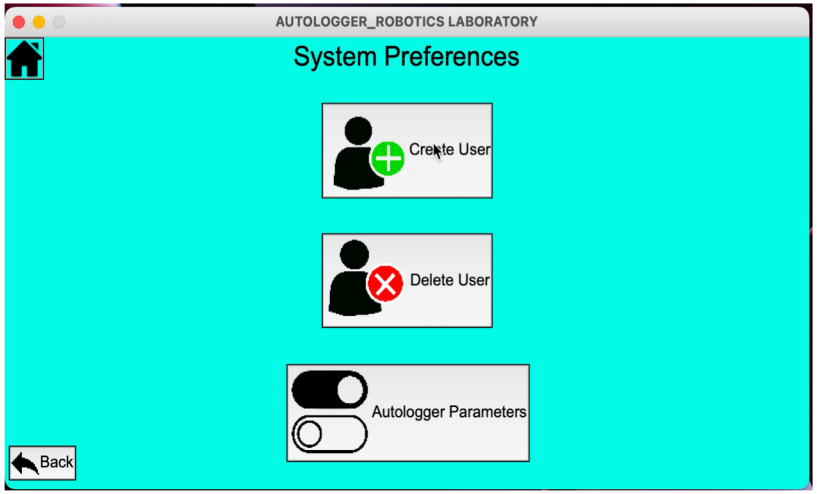
User Login + Active Users



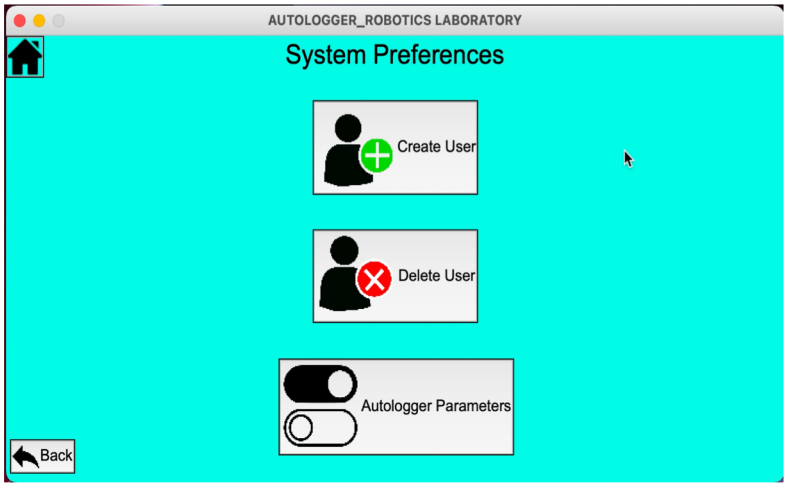
User Logout + Active Users



Create User



Delete User

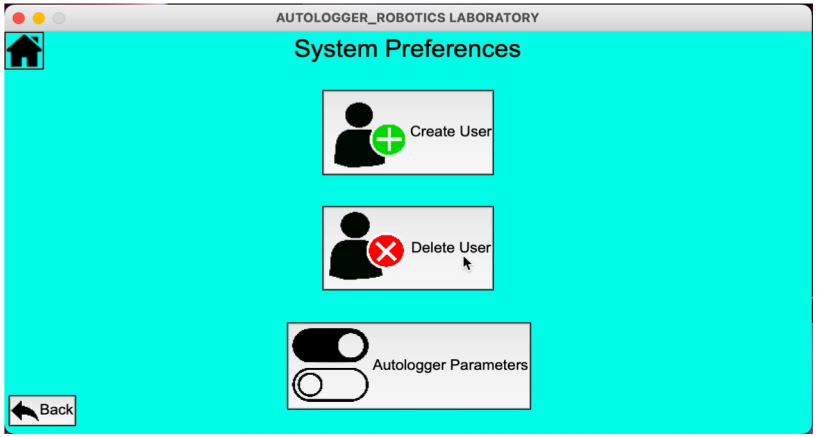


Software executed and videos recorded on: MacBook

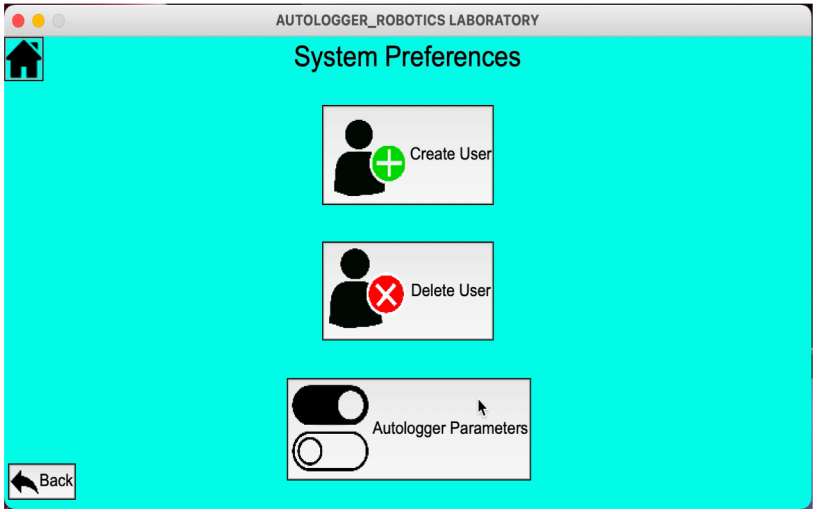
HOCHSCHULE RHEIN-WAAL
Rhine-Waal University of Applied Sciences

Results

Archive User



Autologger Parameters

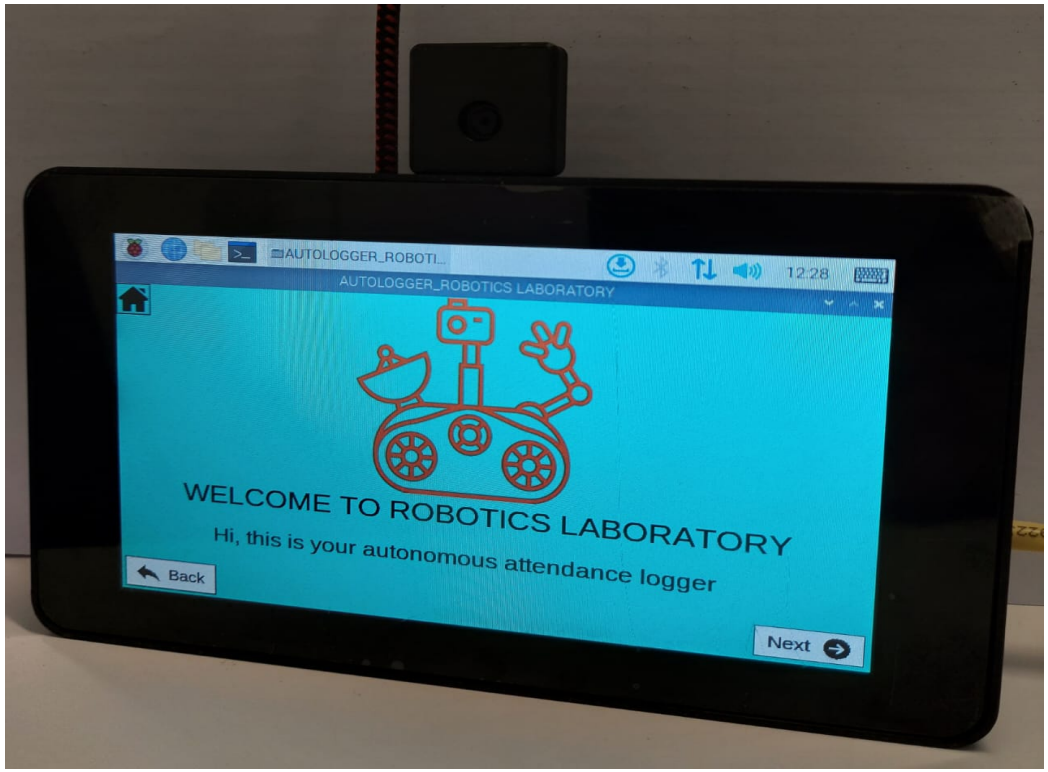


Software executed and videos recorded on: MacBook

Conclusion

Development of attendance tracker system deployed on raspberry pi 4 as an embedded system has been successful using face detection, face recognition and gesture recognition.

The device looks as follows:



Working Device

Functionalities Delivered:

- User Login/Logout
- Active Users
- About Software
- Create User
- Delete/Archive User
- Changing the parameter settings

Limitations

- Computer Vision fails to work properly in low environment light conditions.
- Face recognition model isn’t able to identify the person wearing a mask or if the person has an extreme candid pose.
- After optimisation, for the software to recognize a face, the previous face has to go out of the frame first and then the new user can enter the frame. Optimisation adds to the robustness of the system.
- The system can be fooled easily by showing a 2D picture of an authorized user, thus logging in/out of the system. This is primarily due to lack of depth information from the frame while recognizing a face. Also, the face recognition model has not been trained upon using 3D information which makes the system vulnerable to false authorizations.
- Regarding the hardware, the system needs an internet connection to keep a track of time as the hardware lacks an RTC module.

Future Development

- The most important factor to be improved is to make the system fool proof by using depth information which could be established using upgraded hardware and 3D face recognition technique.
- Second most important factor to be considered is to verify if the gesture is given out from the same person who is being recognized in the frame. This could be approached using human pose estimation model.
- Carry out intensive optimisation in the areas left untouched like memory and CPU optimisation.
- Design the code in a more OOP based approach.
- Testing out the hardware on a better embedded system if available could also yield better results.
- To gain remote access to the device.

Thank you for your attention!

☺

Questions & Practical Demonstration