ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

UNIVERSITY OF WEST ATTICA

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Καταγραφή σημείων αναφοράς και διαχείριση βιομετρικών δεδομένων

Κετσεμενίδης Ελευθέριος
Αριθμός Μητρώου: 18390282

Επιβλέπων: Γιαννακόπουλος Ηρ. Παναγιώτης
Καθηγητής

Ακαδημαϊκό έτος 2022-2023

Αιγάλεω, Σεπτέμβριος 2023

ΕΠΙΤΡΟΠΗ ΕΞΕΤΑΣΗΣ

Γ. Πρεζεράκος

Καθηγητής

Georgios Prezerakos

Digitally signed by Georgios Prezerakos
Date: 2024.01.15 21:13:16 +02'00'

Panagiotis Giannakopoulos

Digitally signed by Panagiotis Giannakopoulos
Date: 2024.01.10 20:50:13 +02'00'

Π. Γιαννακόπουλος

Καθηγητής

Stavros Fatouros

Digitally signed by Stavros Fatouros
Date: 2023.12.29 10:16:11 +02'00'

Σ. Φατούρος

Αναπληρωτής Καθηγητής

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ και Κετσεμενίδης Ελευθέριος, Σεπτέμβριος, 2023

και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις θέσεις του επιβλέποντος, της επιτροπής εξέτασης ή τις επίσημες θέσεις του Τμήματος και του Ιδρύματος.

Δήλωση Συγγραφέα Διπλωματικής Εργασίας

Ο κάτωθι υπογεγραμμένος Κετσεμενίδης Ελευθέριος του Γεωργίου, με αριθμό μητρώου 18390282, φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών, δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της πτυχιακής/διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς παραφρασμένες, αναφέρονται στο σύνολο του, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένων χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από εμένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση των πτυχίων μου.»

Κετσεμενίδης Ελευθέριος

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες προς όλους τους καθηγητές του Πανεπιστημίου Δυτικής Αττικής για την ανεκτίμητη συνεισφορά τους στην πορεία μου στο προπτυχιακό Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών. Οι γνώσεις και οι δεξιότητες που απέκτησα κατά τα χρόνια σπουδών μου δεν θα ήταν δυνατές χωρίς την εμπειρογνωμοσύνη, την υποστήριξη και την ενθάρρυνσή σας. Η πτυχιακή εργασία που ολοκληρώνω τώρα είναι αποτέλεσμα της πολύτιμης σας καθοδήγησης, και αισθάνομαι τυχερός που είχα την ευκαιρία να μαθαίνω από εσάς. Σας είμαι ευγνώμων για την εμπειρία που μοιραστήκαμε και για τη συνεισφορά που θα με συνοδεύει στον επόμενο κεφάλαιο της ζωής μου.

Επιπλέον, θέλω να ευχαριστήσω του γονείς μου, Ιορδάνη και Μαρία, για την πολύτιμη υποστήριξη τους σε όλο μου το βίο.

Τέλος, ευχαριστώ πολύ τον επιβλέποντα καθηγητή Δρ. Γιαννακόπουλο Παναγιώτη για την καθοδήγηση του.

Περίληψη

Στην παρούσα μελέτη, καταγράφεται αναλυτικά η δημιουργία ενός συστήματος ελέγχου ταυτότητας χρηστών, κάνοντας χρήση βιομετρικών δεδομένων και συγκεκριμένα του δακτυλικού αποτυπώματος. Κατασκευάστηκε υλικό μέρος υπό την μορφή αισθητήρα δαχτυλικού αποτυπώματος Raspberry Pi 4 και UART Capacitive Fingerprint Sensor, ενώ για την ανάγκες διαχείρισης των δεδομένων δημιουργήθηκε διαδικτυακή εφαρμογή με βάση την Python και το Flask. Στα κεφάλαια της παρούσας εργασίας, παρουσιάζονται έννοιες όπως: η επαλήθευση της ταυτότητας των χρηστών ενός συστήματος και η συνεισφορά τους στην ασφάλεια αυτού, ο βιομετρικός έλεγχος ταυτότητας και η διαχείριση των βιομετρικών δεδομένων και ειδικότερα στο ζήτημα της συλλογής, της επεξεργασίας και της ασφαλούς αποθήκευσής τους. Επιπλέον, παρουσιάζεται το αποτέλεσμα βιβλιογραφικής αναζήτησης για τα απαραίτητα χαρακτηριστικά που πρέπει να έχει το software και το hardware προκειμένου η λειτουργία του ενός συστήματος βιομετρικού ελέγχου ταυτότητας να είναι ασφαλής και χρηστική. Τέλος, παρουσιάζεται η υπό εξέταση εφαρμογή με ανάλυση για το software και το hardware που χρησιμοποιήθηκε.

Λέξεις Κλειδιά: Βιομετρικά δεδομένα, έλεγχος ταυτότητας, αισθητήρας δακτυλικού αποτυπώματος, Python , Flask, Raspberry Pi 4

Abstract

In this dissertation, we detail the creation of a biometric data user authentication system specifically with the use of fingerprint scanner. The hardware part was built with the use of a Raspberry Pi 4 and a UART Capacitive Fingerprint Sensor, while a web application based on Python and Flask was created for data management needs. In the following chapters concepts such as: the verification of the identity of the users of a system and their contribution to its security, biometric authentication and the management of biometric data and in particular the issue of their collection, processing and secure storage are presented. In addition, the result of a bibliographic search is presented for the necessary characteristics that the software and hardware must have in order for the operation of a biometric authentication system to be secure and usable. Finally, the application under consideration is presented with an analysis of the software and hardware used.

Keywords: Biometrics, Authentication, Fingerprint Sensor, Python, Flask, Raspberry Pi 4

Περιεχόμενα

Εικόνες

Εισαγωγή

Η ταχεία εξέλιξη της τεχνολογίας έχει καταστήσει τα συστήματα επεξεργασίας ψηφιακών βιομετρικών δεδομένων μέρος της καθημερινότητάς μας, αντικαθιστώντας τις παραδοσιακές μεθόδους ελέγχου ταυτότητας. Συγκεκριμένα, η αναγνώριση ατόμων μέσω των μοναδικών χαρακτηριστικών τους, όπως για παράδειγμα τα δακτυλικά αποτυπώματα ή η χαρακτηριστική δομή των ματιών τους, αποτελεί μια προφανή βολική μέθοδο επαλήθευσης της ταυτότητας τους. Τόσο ο αριθμός όσο και η ποικιλία των εφαρμογών για αυτές τις τεχνολογίες συνεχίζει να αυξάνεται.

Σκοπός της παρούσας μελέτης είναι η δημιουργία ενός συστήματος ελέγχου ταυτότητας χρηστών, το οποίο θα κάνει χρήση των βιομετρικών τους δεδομένων. Ειδικότερα, μέσα από τη διαθέσιμη βιβλιογραφία θα εξετασθούν και θα αναλυθούν οι διαθέσιμες λύσεις προκειμένου να δημιουργηθεί μια χρηστική και ασφαλής εφαρμογή η οποία θα επαληθεύει την ταυτότητα των χρηστών μέσα από το δακτυλικό τους αποτύπωμα. Συγκεκριμένα, θα δημιουργήσουμε μια εφαρμογή η οποία θα πιστοποιεί την ταυτότητα και την παρουσία των φοιτητών ενός πανεπιστημίου και στη συνέχεια θα τους κατηγοριοποιεί στα αντίστοιχα τμήματα. Για την υλοποίηση θα αξιοποιηθεί η γλώσσα προγραμματισμού της Python, η βιβλιοθήκη της Flask και SQLAlchemy. Αυτές οι βιβλιοθήκες προσφέρουν ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών καθώς έχουν δυνατότητα ενσωμάτωσης η μια της άλλης. Το διαχειριστικό της βάσης μας είναι το PostgresSQL του οποίου οι δομές δεδομένων είναι πολύ συμβατές με αυτές της Python. Τέλος για την επικοινωνία της Python με το σύστημα δακτυλικών αποτυπωμάτων στο Raspberry χρησιμοποιούμε την βιβλιοθήκη GPIO. Το Hardware που χρησιμοποιήσαμε είναι το Raspberry Pi 4 Model B 8GB , το FT232 USB UART Board (micro) και ο αισθητήρας αναγνώρισης δακτυλικού αποτυπώματος Round-shaped All-in-one UART Capacitive Fingerprint Sensor.

Η δομή της εργασίας έχεις ως εξής:

Στο πρώτο κεφάλαιο θα γίνει αναφορά στην έννοια της επαλήθευσης της ταυτότητας των χρηστών ενός συστήματος ή μιας εφαρμογής. Ο έλεγχος ταυτότητας είναι μια διαδικασία όπου οι χρήστες ενός συστήματος προσδιορίζονται χρησιμοποιώντας

διαφορετικούς μηχανισμούς ελέγχου ταυτότητας. Η διαδικασία αυτή αποτελεί το πρώτο βήμα για τη βελτίωση της ασφάλειας. Η βασική λειτουργία του ελέγχου ταυτότητας είναι να διαχειρίζεται την αναγνώριση των χρηστών και να τους παρέχει κατάλληλο έλεγχο πρόσβασης για απρόσκοπτη λειτουργία και ασφάλεια. Επιπλέον, στην ενότητα αυτή θα περιγραφούν οι διαφορετικές μέθοδοι ελέγχου ταυτότητας, ενώ θα αναλυθούν τα πλεονεκτήματα και τα μειονεκτήματα της κάθε μίας αντίστοιχα.

Στο δεύτερο κεφάλαιο θα γίνει αναφορά στην έννοια και τα κύρια χαρακτηριστικά που παρουσιάζουν τα βιομετρικά δεδομένα. Ο βιομετρικός έλεγχος ταυτότητας χρησιμοποιεί τα φυσικά χαρακτηριστικά των ατόμων, τα οποία αποθηκεύονται σε μια βάση δεδομένων και ελέγχονται σε σχέση με τα δεδομένα που περιέχονται στη βάση δεδομένων κάθε φορά που ένας χρήστης θέλει να αποκτήσει πρόσβαση σε οποιαδήποτε σύστημα ή εφαρμογή. Θα γίνει εκτενής αναφορά στον μηχανισμό λειτουργίας των πιο δημοφιλών μεθόδων ελέγχου ταυτότητας μέσω βιομετρικών δεδομένων, ενώ θα παρουσιαστούν οι περιορισμοί και τα ευάλωτα σημεία αυτών των μεθόδων. Τέλος, θα παρουσιαστούν κάποιες προκλήσεις, οι οποίες σχετίζονται με το κόστος, την αξιοπιστία, το βαθμό ασφάλειας και το υλικό που χρησιμοποιείται για τη λειτουργία της κάθε μεθόδου και οι αντίστοιχες τάσεις για την επίλυση αυτών.

Στο τρίτο κεφάλαιο θα γίνει αναφορά στη διαχείριση των βιομετρικών δεδομένων και ειδικότερα στο ζήτημα της συλλογής, της επεξεργασίας και της ασφαλούς αποθήκευσής τους. Επιπλέον, θα γίνει μια εκτενής ανάλυση στις διάφορες μεθόδους-στρατηγικές αποθήκευσης των βιομετρικών δεδομένων, καθώς οι βασικές αδυναμίες των τεχνολογιών βιομετρικής επαλήθευσης ταυτότητας έχουν να κάνουν με ζητήματα ασφάλειας πληροφοριών. Θα αναφερθούν τα πολυάριθμα ζητήματα ασφάλειας πληροφοριών που επηρεάζουν τα βιομετρικά συστήματα ελέγχου ταυτότητας για μια πιο αντικειμενική αξιολόγηση των κινδύνων που σχετίζονται με τη χρήση υπαρχόντων εφαρμογών.

Στο τέταρτο κεφάλαιο θα γίνει μια εκτενής βιβλιογραφική αναζήτηση στα απαραίτητα χαρακτηριστικά που πρέπει να έχει το software και το hardware προκειμένου η λειτουργία του ενός συστήματος βιομετρικού ελέγχου ταυτότητας να είναι ασφαλής και χρηστική. Όσον αφορά το software θα γίνει μια εκτενής περιγραφή στις διαθέσιμες γλώσσες προγραμματισμού, στα συστήματα διαχείρισης βάσεων

δεδομένων, στις βιβλιοθήκες, στο πλαίσιο για την ανάπτυξη ιστοσελίδων και στο λογισμικό ORM Επιπλέον, θα παρουσιαστούν οι διάφοροι τύποι σαρωτών που κυκλοφορούν στο εμπόριο και τα αναλυθούν τα ιδιαίτερα χαρακτηριστικά τους. Μέσα από τη σύγκριση των πτυχών θα καταλήξουμε στο υλικό που θα βασιστεί η υπό εξέταση εφαρμογή.

Στο πέμπτο κεφάλαιο θα παρουσιαστεί η υπό εξέταση εφαρμογή. Ειδικότερα θα αναλύσουμε το software που θα χρησιμοποιήσουμε, αλλά και το hardware.

Κεφάλαιο 1ο

Έλεγχος Ταυτότητας Χρήστη

1.1 Μέθοδος επαλήθευσης ταυτότητας

Η μέθοδος επαλήθευσης ταυτότητας (authentication process) είναι μια διαδικασία κατά την οποία επικυρώνεται η ταυτότητα ενός χρήστη που ζητά πρόσβαση σε ένα σύστημα, δίκτυο, διακομιστή, εφαρμογή, ιστότοπο ή συσκευή μέσα από διαφορετικούς μηχανισμούς ελέγχου ταυτότητας. Ο πρωταρχικός στόχος του ελέγχου ταυτότητας είναι να διασφαλίσει ότι ο χρήστης είναι αυτός που ισχυρίζεται ότι είναι (Lal, Prasad, & Farik, 2016).

Σε ένα σύστημα ασφαλείας, η διαδικασία ελέγχου ταυτότητας ελέγχει τις πληροφορίες που παρέχονται από τον χρήστη με τα στοιχεία που υπάρχουν σε μια βάση δεδομένων. Εάν οι πληροφορίες ταιριάζουν με τις πληροφορίες της βάσης δεδομένων, ο χρήστης έχει πρόσβαση στο σύστημα ασφαλείας. Με αυτόν τον τρόπο βελτιώνεται η ασφάλεια και η διαχείριση επιτρέποντας σε οποιονδήποτε να διαχειρίζεται την ταυτότητα και την πρόσβαση κάθε μεμονωμένου χρήστη (Rajarajeswari & Stella, 2019; miniOrange Security Software Pvt Ltd, 2023).

Σε ένα ικανοποιητικό σύστημα επαλήθευσης ταυτότητας πρέπει να υπάρχει η σωστή αντιστάθμιση μεταξύ ασφάλειας και ευκολίας χρήστη, η οποία μπορεί να επιτευχθεί με την προσαρμογή του επιπέδου ελέγχου ταυτότητας βάσει μιας συνεχούς αξιολόγησης κινδύνου. Ένα σύστημα μπορεί να έχει ισχυρή ασφάλεια εάν ζητά με συστηματικό τρόπο πολλαπλούς παράγοντες ελέγχου ταυτότητας. Αντίθετα, αυτό το είδος ελέγχου ταυτότητας χρήστη μπορεί να έχει αντίθετα αποτελέσματα καθιστώντας το μη χρηστικό προς τους χρήστες (Thales, 2023).

Τέλος, το ενδιαφέρον για ισχυρό έλεγχο ταυτότητας χρήστη είναι αυξημένο, καθώς όλο και περισσότεροι κανονισμοί, όπως ο GDPR[1], η Ευρωπαϊκή Οδηγία Πληρωμών[2]

1 Γενικός Κανονισμός για την Προστασία των Δεδομένων

2 Αναθεωρημένη οδηγία για τις υπηρεσίες πληρωμών

(PSD2) και τα πρότυπα, όπως το PCI DSS[3], επιβάλλουν αυστηρές απαιτήσεις ασφαλείας για την προστασία των προσωπικών πληροφοριών, για τις ηλεκτρονικές πληρωμές και την προστασία των οικονομικών δεδομένων των καταναλωτών.

1.2 Διαφορετικοί τύποι ελέγχου ταυτότητας

Ο έλεγχος της ταυτότητας ενός χρήστη μπορεί να γίνει με αρκετές μεθόδους. Σημαντικοί συντελεστές για την επιλογή τους είναι η ασφάλεια της κάθε μεθόδου και η χρηστικότητα της (miniOrange Security Software Pvt Ltd, 2023). Το κλασικό παράδειγμα για τα συστήματα ελέγχου ταυτότητας προσδιορίζει τρεις παράγοντες ως ακρογωνιαίους λίθους του ελέγχου ταυτότητας (Lal, Prasad, & Farik, 2016; Rajarajeswari & Stella, 2019):

- Παράγοντας γνώσης ("κάτι που γνωρίζει ο χρήστης"): Το σύστημα τον αποδέχεται εάν δείξει ότι γνωρίζει ορισμένες πληροφορίες. Παραδείγματα περιλαμβάνουν τους κωδικούς PIN, απαντήσεις σε ερωτήσεις ασφαλείας, στοιχεία φορολογικής δήλωσης κ.λπ.
- Παράγοντας κατοχής ("κάτι που έχει στην κατοχή του ο χρήστης "): Το σύστημα δέχεται τον χρήστη εάν μπορεί να αποδείξει ότι έχει μια συγκεκριμένη φυσική συσκευή πάνω του. Στα παραδείγματα περιλαμβάνονται συσκευές όπως οι κάρτες, τα κινητά τηλέφωνα και τα USB.
- Εγγενής παράγοντας ("ένα χαρακτηριστικό του χρήστη που τον προσδιορίζει"): Το σύστημα αποδέχεται τον χρήστη χρησιμοποιώντας μια βιομετρική σύγκριση. Παραδείγματα περιλαμβάνουν τους σαρωτές δακτυλικών αποτυπωμάτων, τους σαρωτές αμφιβληστροειδούς, την αναγνώριση φωνής και τη βιομετρία συμπεριφοράς.

Επιπλέον ο έλεγχος ταυτότητας μπορεί να γίνει μέσω πολλαπλών παραγόντων, όπου γίνεται χρήση περισσότερων του ενός από τους παραπάνω παράγοντες. Η ισχύς των συστημάτων ελέγχου ταυτότητας καθορίζεται σε μεγάλο βαθμό από την τεχνολογία

---

3 Πρότυπο Ασφάλειας Δεδομένων Βιομηχανίας Πληρωμών

ελέγχου ταυτότητας που αναπτύσσεται και τον αριθμό των παραγόντων που ενσωματώνονται από το σύστημα. Όσο περισσότεροι παράγοντες χρησιμοποιούνται, τόσο πιο ισχυρό είναι το σύστημα ελέγχου ταυτότητας (Thales, 2023).

1.3 Πλεονεκτήματα-Μειονεκτήματα των Μεθόδων

1.3.1 Είσοδος βάσει κωδικού πρόσβασης

Αυτός ο τύπος ελέγχου ταυτότητας απαιτεί από τον αιτούντα την πρόσβαση να εισαγάγει έναν συνδυασμό του ονόματος χρήστη/του και ενός κωδικού πρόσβασης. Υπάρχουν δύο μέρη σε αυτή τη μέθοδο. Πρώτον, ο αιτών εισάγει το όνομα χρήστη και δεύτερον, τον κωδικό πρόσβασης. Ο κωδικός πρόσβασης είναι ο μυστικός συνδυασμός λέξεων και αριθμών που γνωρίζει ο αιτών. Το άτομο εξουσιοδοτείται μόνο όταν έχουν επαληθευτεί και τα δύο αυτά στοιχεία (miniOrange Security Software Pvt Ltd, 2023; Lal, Prasad, & Farik, 2016).

Ένα από τα δυνατά σημεία αυτής της μεθόδου είναι ότι οι μεγάλοι κωδικοί πρόσβασης είναι πολύ δύσκολο να σπάσουν. Ένας ισχυρός κωδικός αποτελείται έναν συνδυασμό κεφαλαίων, πεζών, αριθμών και μοναδικών χαρακτήρων. Ένας κωδικός πρόσβασης 12 χαρακτήρων με 94 πληθάριθμο και εντροπία 78,7 bit θα χρειαστεί 55 ημέρες για να σπάσει χρησιμοποιώντας υπερυπολογιστές και 3018 χρόνια χρησιμοποιώντας υπολογιστή (Farik & Shawkat, 2015).

Οι πλειοψηφία των χρηστών χρησιμοποιούν πολλές διαδικτυακές υπηρεσίες (εφαρμογές και ιστότοποι), με αποτέλεσμα να είναι δύσκολο να παρακολουθούν όλα τα ονόματα χρήστη και τους κωδικούς πρόσβασης τους. Ως συνέπεια αυτού, οι χρήστες ξεχνούν τους κωδικούς πρόσβασης και χρησιμοποιούν τον ίδιο κωδικό πρόσβασης για πολλές υπηρεσίες. Οι επιτήδειοι του κυβερνοχώρου εισέρχονται σε αυτό το σημείο και ξεκινούν ενέργειες όπως phishing, παραβιάσεις δεδομένων και ούτω καθεξής. Αυτός είναι ο βασικός λόγος για τον οποίο ο τυπικός έλεγχος ταυτότητας βάσει κωδικού πρόσβασης μειονεκτεί και οι περισσότεροι οργανισμοί στρέφονται σε προηγμένους πρόσθετους παράγοντες ελέγχου ταυτότητας ασφαλείας (miniOrange Security Software Pvt Ltd, 2023).

1.3.2 Είσοδος βάσει κάρτας

Μια έξυπνη κάρτα είναι μια κάρτα μεγέθους πιστωτικής κάρτας που έχει ενσωματωμένο πιστοποιητικό που χρησιμοποιείται για την αναγνώριση του κατόχου. Ο χρήστης μπορεί να εισάγει την κάρτα σε μια συσκευή ανάγνωσης έξυπνων καρτών για να ελέγξει την ταυτότητα του ατόμου. Οι έξυπνες κάρτες χρησιμοποιούνται συνήθως με ένα PIN που παρέχει έλεγχο ταυτότητας πολλαπλών παραγόντων. Με άλλα λόγια, ο χρήστης πρέπει να έχει κάτι (την έξυπνη κάρτα) και να γνωρίζει κάτι (το PIN).

Ένα από τα δυνατά σημεία της έξυπνης κάρτας είναι ότι συνοδεύεται από κάρτα μνήμης με δεδομένα αποθήκευσης με έλεγχο ταυτότητας δύο παραγόντων. Επιπλέον, συνοδεύεται από μικροεπεξεργαστή, καθιστώντας ισχυρότερο τον έλεγχο ταυτότητας δύο παραγόντων. Η έξυπνη κάρτα κλειδώνεται εάν το Pin έχει εισαχθεί λανθασμένα μετά από πολλές προσπάθειες, αποτρέπει επιθέσεις με χρήση λεξικού[4] (dictionary attacks), είναι φορητή και μεταφέρεται εύκολα από τους χρήστες (Lal, Prasad, & Farik, 2016).

Μερικά από τα μειονεκτήματα που αφορούν την έξυπνη κάρτα είναι ότι ορισμένοι χρήστες δυσκολεύονται να θυμηθούν το PIN και η κάρτα μπορεί να κλειδωθεί μετά από συγκεκριμένο αριθμό λανθασμένων προσπαθειών. Δεδομένου ότι είναι φορητή μπορεί επίσης να κλαπεί. Επιπλέον, ορισμένοι χρήστες που πραγματοποιούν συχνά αγορές μέσω διαδικτύου μπορεί να πέσουν θύματα phishing (Lal, Prasad, & Farik, 2016).

1.3.3 Είσοδος βάσει Βιομετρικών στοιχείων

Ο έλεγχος ταυτότητας χρήστη μέσω βιομετρικών δεδομένων είναι μια μέθοδος που επαληθεύει την ταυτότητά του χρήστη με βάση τη μέτρηση των μοναδικών φυσιολογικών χαρακτηριστικών ή των χαρακτηριστικών συμπεριφοράς του. Ο έλεγχος ταυτότητας που βασίζεται στα φυσιολογικά χαρακτηριστικά του χρήστη

[4] Επίθεση που χρησιμοποιεί ένα περιορισμένο υποσύνολο ενός χώρου κλειδιών για να νικήσει έναν μηχανισμό κρυπτογράφησης ή ελέγχου ταυτότητας προσπαθώντας να προσδιορίσει το κλειδί αποκρυπτογράφησης.

γίνεται μέσω των δακτυλικών αποτυπωμάτων, την αναγνώριση προσώπου, τη σάρωση ίριδας, τη γεωμετρία χεριού και τη σάρωση του αμφιβληστροειδούς. Ο έλεγχος ταυτότητας που βασίζεται σε συμπεριφορικά χαρακτηριστικά γίνεται μέσω της αναγνώρισης φωνής, των βηματισμών, τη σάρωση πληκτρολόγησης και τη σάρωση της υπογραφής. Τα δακτυλικά αποτυπώματα και τα αποτυπώματα χεριών είναι η πιο ευρέως χρησιμοποιούμενη βιομετρική μέθοδος που χρησιμοποιείται σήμερα (Lal, Prasad, & Farik, 2016).

Οι βιομετρικοί έλεγχοι ταυτότητας χρησιμοποιούνται ευρέως και έχουν μεγάλη ισχύ καθώς δεν απαιτούν την χρήση κωδικών πρόσβασης καθώς τα βιομετρικά στοιχεία είναι μοναδικά και είναι πολύ δύσκολο να αναπαραχθούν, δεν μπορούν να χαθούν και έχουν χαμηλό κόστος (σάρωση δακτυλικών αποτυπωμάτων). Από την άλλη μεριά, τα βιομετρικά στοιχεία είναι ευαίσθητα σε σφάλματα. Ένα σφάλμα ψευδούς απόρριψης (ονομάζεται επίσης σφάλμα τύπου 1) εμφανίζεται όταν ένα σύστημα απορρίπτει ψευδώς έναν χρήστη και υποδεικνύει ότι ο χρήστης δεν είναι γνωστός. Ένα σφάλμα ψευδούς αποδοχής (ονομάζεται επίσης σφάλμα τύπου 2) εμφανίζεται όταν ένα σύστημα προσδιορίζει ψευδώς έναν άγνωστο χρήστη ως γνωστό χρήστη. Τα βιομετρικά συστήματα μπορούν συνήθως να ρυθμιστούν για ευαισθησία, αλλά η ευαισθησία επηρεάζει την ακρίβεια. Ένα άλλο πρόβλημα είναι ότι δεν υπάρχει ενιαίο πρότυπο λόγω της ποικιλίας των προμηθευτών (Lal, Prasad, & Farik, 2016).

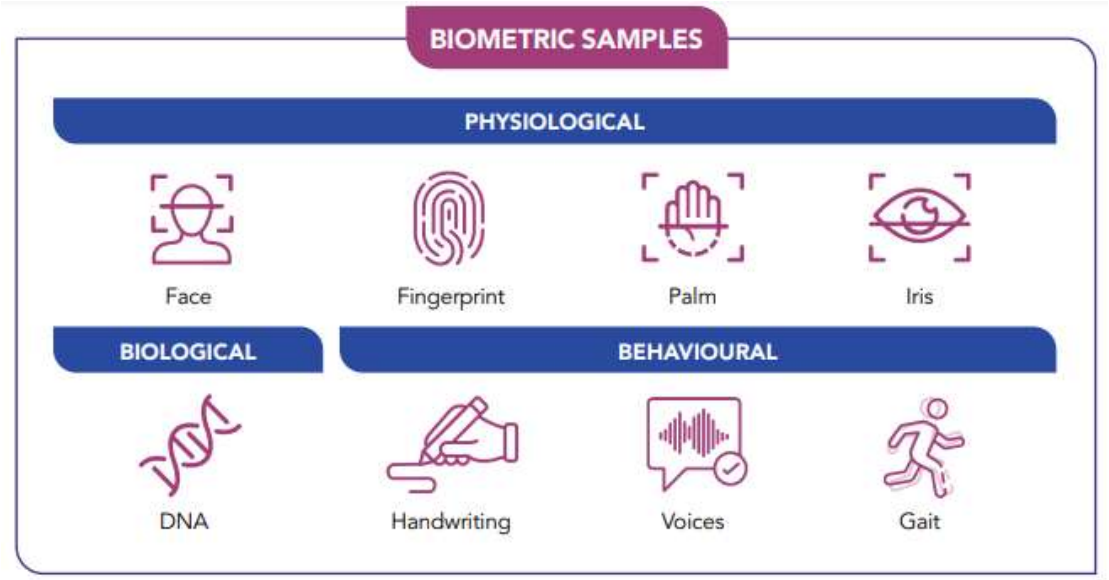
Κεφάλαιο 2ο

Βιομετρικός έλεγχος ταυτότητας

2.1 Βιομετρικά Δεδομένα

Η Βιομετρία είναι η στατιστική και μαθηματική μέτρηση μοναδικών φυσικών ή βιολογικών χαρακτηριστικών για σκοπούς αναγνώρισης. Στην ασφάλεια στον κυβερνοχώρο, ο ορισμός των βιομετρικών στοιχείων αναφέρεται στη χρήση μοναδικών βιολογικών χαρακτηριστικών για ψηφιακό έλεγχο ταυτότητας και έλεγχο πρόσβασης (Molinaro, 2022).

Τα βιομετρικά δεδομένα είναι κάθε είδους πληροφορία σχετικά με τα φυσικά χαρακτηριστικά ενός ατόμου. Τα βιομετρικά δεδομένα αναφέρονται σε βιομετρικά δείγματα (δηλαδή δεδομένα που σχετίζονται με τα φυσιολογικά, βιολογικά ή συμπεριφορικά χαρακτηριστικά ενός ατόμου) ή βιομετρικά πρότυπα που δημιουργούνται μέσω τεχνικής επεξεργασίας βιομετρικών δειγμάτων. Παραδείγματα βιομετρικών δειγμάτων περιλαμβάνουν εικόνες προσώπου, δακτυλικά αποτυπώματα και ηχογραφήσεις φωνής. Τα βιομετρικά δείγματα καταγράφονται μέσω αισθητήρων όπως αισθητήρες εικόνας και ήχου (Personal Data Protection Commission Singapore (PDPC), 2022).



Εικόνα 1: Παραδείγματα βιομετρικών δειγμάτων, πηγή: (Personal Data Protection Commission Singapore (PDPC), 2022)

Τα βιομετρικά δεδομένα όταν συσχετίζονται με άλλες πληροφορίες για ένα άτομο αποτελούν μέρος των προσωπικών δεδομένων αυτού του ατόμου. Τα βιομετρικά πρότυπα (από μόνα τους, χωρίς να συσχετίζονται με πληροφορίες ταυτοποίησης) θεωρούνται ανώνυμα δεδομένα, καθώς είναι δυαδικές αναπαραστάσεις που προέρχονται από την εφαρμογή αλγορίθμου σε βιομετρικά δείγματα (Personal Data Protection Commission Singapore (PDPC), 2022).

Για να είναι χρήσιμα, τα βιομετρικά δεδομένα πρέπει να είναι μοναδικά, μόνιμα και να μπορούν να συλλεχθούν, επομένως η καταλληλότητα διαφορετικών τύπων βιομετρικών δεδομένων ποικίλλει ανάλογα με την εφαρμογή. Για παράδειγμα, πολλοί χρήστες χρησιμοποιούν βιομετρικές πληροφορίες με τη μορφή τεχνολογίας ψηφιακών δακτυλικών αποτυπωμάτων και αναγνώρισης προσώπου για να ξεκλειδώσουν γρήγορα και εύκολα τα κινητά τους τηλέφωνα. Τέτοιες εφαρμογές είναι φαινομενικά αδύνατες με ένα φυσικό δείγμα DNA (Molinaro, 2022).

Το είδος των δεδομένων και ο τρόπος που χρησιμοποιούνται αλλάζει καθώς προχωρά η τεχνολογία. Τα ευρέως χρησιμοποιούμενα βιομετρικά δεδομένα μπορεί τελικά να καταστούν παρωχημένα καθώς μπορούν να παραποιηθούν, με χαρακτηριστικά παραδείγματα τα ρομπότ φωνής που μπορούν να υποκλέψουν κωδικούς πρόσβασης και τα bot αμφιβληστροειδούς και δακτυλικών αποτυπωμάτων (Molinaro, 2022).

Η βιομετρία έχει ένα ευρύ φάσμα εφαρμογών, ειδικά για το μέλλον της κυβερνοασφάλειας και του ψηφιακού απορρήτου. Από τη μία πλευρά, η βιομετρική τεχνολογία καθιστά τη σύνδεση σε λογαριασμούς και άλλα πρωτόκολλα ασφαλείας ταχύτερη και ευκολότερη για τους χρήστες, καθώς και πολύ πιο δύσκολη την παραβίαση τους. Από την άλλη πλευρά, αφήνει τους χρήστες πιο ευάλωτους εάν μια παραβίαση δεδομένων εκθέσει τα βιομετρικά τους διαπιστευτήρια. Μια επιτυχημένη επίθεση μπορεί να έχει σοβαρές επιπτώσεις στη ζωή και το απόρρητο των χρηστών. Η ευρεία χρήση βιομετρικών δεδομένων εγείρει μια σειρά από ζητήματα ιδιωτικότητας και ασφάλειας με αποτέλεσμα να αυξάνεται ο κίνδυνος κλοπής ταυτότητας και απάτης (Molinaro, 2022; Pagnin & Mitrokotsa, 2017).

2.2 Βιομετρικές μέθοδοι ελέγχου ταυτότητας

Ο βιομετρικός έλεγχος ταυτότητας είναι ένα γρήγορο, ακριβές και φιλικό προς το χρήστη εργαλείο που προσφέρει μια αποτελεσματική και αξιόπιστη λύση σε συστήματα ελέγχου πολλαπλής πρόσβασης. Η βάση για τον βιομετρικό έλεγχο ταυτότητας είναι η εξαγωγή ενός βιομετρικού χαρακτηριστικού από το σώμα ή την συμπεριφορά του ανθρώπου (Pagnin & Mitrokotsa, 2017).

Σε γενικές γραμμές, ένα βιομετρικό σύστημα ελέγχου ταυτότητας λειτουργεί με τον ακόλουθο τρόπο. Πρώτον, ένας χρήστης εγγράφεται στο σύστημα παρέχοντας την ταυτότητά του μαζί με το βιομετρικό του πρότυπο που γίνεται το πρότυπο αναφοράς του (φάση εγγραφής). Στη συνέχεια, ο χρήστης μπορεί να πιστοποιηθεί στο σύστημα (φάση ελέγχου ταυτότητας) υποβάλλοντας μια ταυτότητα και ένα βιομετρικό πρότυπο. Το σύστημα εκτελεί μια διαδικασία αντιστοίχισης, η οποία στοχεύει να ελέγξει εάν το παρεχόμενο νέο πρότυπο είναι αρκετά κοντά σε αυτό που είναι αποθηκευμένο για τον συγκεκριμένο χρήστη, όπου ο χρήστης γίνεται πιστοποιημένος/αποδεκτός ή απορρίπτεται (Pagnin & Mitrokotsa, 2017).

Κοινά βιομετρικά χαρακτηριστικά που χρησιμοποιούνται για έλεγχο ταυτότητας είναι το δακτυλικό αποτύπωμα, η υπογραφή, η ίριδα, το σχήμα του προσώπου και η φωνή.

2.2.1 Αναγνώριση Δακτυλικού αποτυπώματος

Επειδή είναι ένα από τα πιο φιλικά προς τον χρήστη και ακριβή βιομετρικά συστήματα, ο έλεγχος ταυτότητας με δακτυλικά αποτυπώματα είναι επί του παρόντος η πιο κοινή βιομετρική τεχνολογία για τους απλούς χρήστες (miniOrange Security Software Pvt Ltd, 2023). Τα ανθρώπινα δακτυλικά αποτυπώματα είναι λεπτομερή, σχεδόν μοναδικά, δύσκολα αλλάζουν και ανθεκτικά, καθιστώντας τα κατάλληλα ως μακροπρόθεσμους δείκτες προσωπικής ταυτότητας. Είναι επομένως φυσικό να υπάρχει ένα ευρύ φάσμα συστημάτων αναγνώρισης δακτυλικών αποτυπωμάτων για χρήση σε εφαρμογές υψηλής ασφάλειας και για την αυτοματοποιημένη αναγνώριση ατόμων (Security Industry Association, 2019). Ο έλεγχος ταυτότητας με δακτυλικό αποτύπωμα ταιριάζει με το μοναδικό μοτίβο του αποτυπώματος ενός ατόμου. Σε ορισμένα προηγμένα συστήματα ελέγχου ταυτότητας με δακτυλικά αποτυπώματα, γίνεται επίσης αισθητή η αγγειακή δομή του δακτύλου. Τα δεδομένα που

χρησιμοποιούνται για την αποθήκευση του δακτυλικού αποτυπώματος κάθε ατόμου πρέπει να είναι αρκετά μικρού μεγέθους ώστε να μπορούν να χρησιμοποιηθούν για τον έλεγχο ταυτότητας. Αυτά τα δεδομένα είναι μια μαθηματική αναπαράσταση των λεπτομερειών των δακτυλικών αποτυπωμάτων, που περιλαμβάνουν συγκεκριμένες λεπτομέρειες των ραβδώσεων τριβής επάνω στο δέρμα, όπως οι σπείρες, οι ραβδώσεις και οι διακλαδώσεις που σχηματίζονται μεταξύ άλλων (Conrad, Misenar, & Feldman, 2017).

2.2.2 Αναγνώριση Γραφικού Χαρακτήρα

Η επαλήθευση του βιομετρικού χαρακτηριστικού του γραφικού χαρακτήρα πιστοποιεί την ταυτότητα των υπογράφων μετρώντας τις χειρόγραφες υπογραφές τον τρόπο που γράφουν, συνήθως μέσω της υπογραφής τους. Η υπογραφή περιέχει μοναδικά βιομετρικά δεδομένα, όπως ο ρυθμός γραφής, η επιτάχυνση και η πίεση. Σε αντίθεση με άλλες μεθόδους καταγραφής ηλεκτρονικής υπογραφής, η επαλήθευση βιομετρικής υπογραφής δεν αντιμετωπίζει την υπογραφή ως γραφική εικόνα. Με γραφικές εικόνες, όπως οι σαρωμένες υπογραφές που συχνά επισυνάπτονται στα έγγραφά, δεν είναι δυνατός ο εντοπισμός της δυναμικής μέσα στην υπογραφή κάθε ατόμου και, ως εκ τούτου, οι υπογραφές μπορούν εύκολα να αντιγραφούν. Αντίθετα, η επαλήθευση βιομετρικής υπογραφής μετρά ακριβώς τον τρόπο που παράγεται η υπογραφή. Επειδή η μέθοδος αυτή μπορεί να παρακολουθεί τις φυσικές διακυμάνσεις κάθε ατόμου με την πάροδο του χρόνου, μπορεί εύκολα να προσδιορίσει την πλαστογραφία σε διάφορα έγγραφα (NAMIRIAL GmbH, 2019).

2.2.3 Αναγνώριση Ίριδας

Κατά τον έλεγχο ταυτότητας, ο σαρωτής ρίχνει ένα δυνατό φως στο μάτι και αναζητά διακριτικά μοτίβα στον πολύχρωμο δακτύλιο γύρω από την κόρη του ματιού. Αυτά τα μοτίβα είναι μοναδικά για όλους και δεν επηρεάζονται από αλλαγές στο φωτισμό ή την έκθεση. Στη συνέχεια, το σαρωμένο μοτίβο συγκρίνεται με δεδομένα που έχουν καταγραφεί σε μια βάση δεδομένων. Όταν ένα άτομο φοράει γυαλιά ή φακούς επαφής, ο έλεγχος ταυτότητας με βάση την ίριδα μπορεί να είναι ανακριβής. Επιπλέον, η μέθοδος αυτή έχει μεγαλύτερο κόστος σε σχέση με άλλους τύπους βιομετρικών στοιχείων (Kinzer, 2022; miniOrange Security Software Pvt Ltd, 2023).

2.2.4 Αναγνώριση Προσώπου

Στον έλεγχο ταυτότητας προσώπου, σαρώνονται πολλαπλές πτυχές του προσώπου ενός ατόμου. Χρησιμοποιεί χαρακτηριστικά του προσώπου, όπως το σχήμα των ματιών, της μύτης και των αυτιών, για να αναγνωρίσει το χρήστη. Όταν ελέγχεται ένα πρόσωπο από διαφορετικές οπτικές γωνίες ή όταν κάποια πρόσωπα μοιάζουν, όπως τα μέλη μίας οικογένειας, τα αποτελέσματα της αναγνώρισης προσώπου μπορεί να είναι ασυνεπή. Επιπλέον, οι μάσκες, τα γυαλιά ηλίου ή ο κακός φωτισμός μπορεί να δημιουργήσουν προβλήματα στην αναγνώριση (Kinzer, 2022).

2.2.5 Αναγνώριση φωνής

Η αναγνώριση φωνής είναι ένα βιομετρικό σύστημα που χρησιμοποιεί τον ήχο της φωνής κάποιου και μια συγκεκριμένη φράση ή φωνητικό μοτίβο για να τον αναγνωρίσει. Αυτό το σύστημα χρησιμοποιείται συχνά για σκοπούς ελέγχου ταυτότητας, όπως κατά τη σύνδεση σε έναν υπολογιστή ή το ξεκλείδωμα ενός τηλεφώνου. Σε αυτή τη διαδικασία, ο τόνος της φωνής αποθηκεύεται με έναν τυποποιημένο μυστικό κωδικό. Ο βιομετρικός έλεγχος ταυτότητας φωνής επιβεβαιώνει ότι ένα άτομο είναι αυτό που ισχυρίζεται ότι είναι αντιστοιχίζοντας τα δείγματα φωνής με το αρχικό δείγμα (Aware, Inc., 2022).

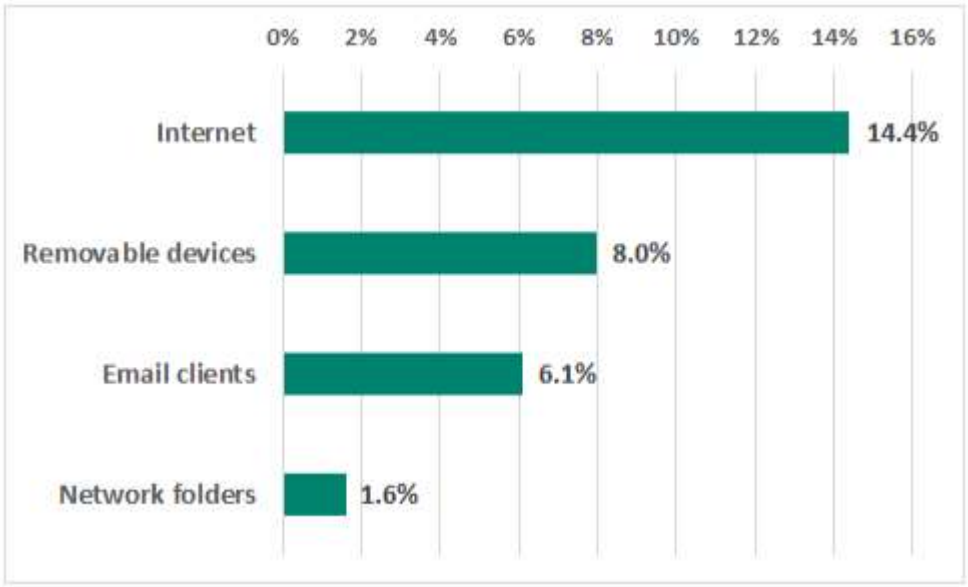
2.3 Περιορισμοί του βιομετρικού ελέγχου ταυτότητας

Ο βιομετρικός έλεγχος ταυτότητας μπορεί να είναι πιο αξιόπιστος από τις παραδοσιακές μεθόδους, όπως οι κωδικοί πρόσβασης, αλλά παρουσιάζει κάποιους περιορισμούς.

Αρχικά, ο βιομετρικός έλεγχος ταυτότητας δεν είναι απόλυτα ακριβής. Όλες οι μέθοδοι βιομετρικού ελέγχου ταυτότητας έχουν ποσοστό σφάλματος, που σημαίνει ότι ένα ορισμένο ποσοστό χρηστών θα απορριφθεί ή θα γίνει ψευδώς αποδεκτό. Αυτό το ποσοστό μπορεί να ποικίλλει ανάλογα με τη μέθοδο που χρησιμοποιείται, αλλά γενικά, όσο πιο ακριβής είναι ο βιομετρικός έλεγχος ταυτότητας, τόσο υψηλότερο είναι το κόστος εφαρμογής (Frąckiewicz , 2023).

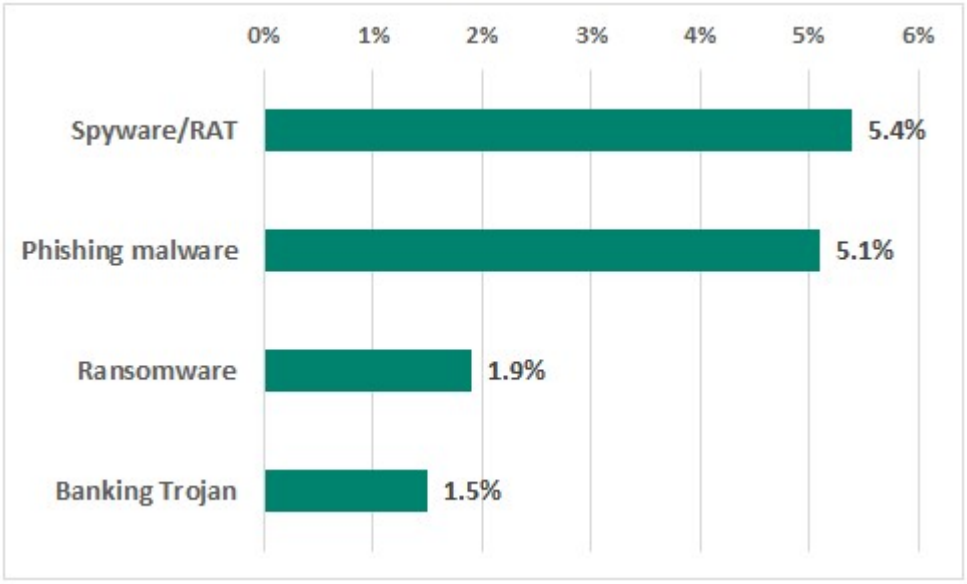
Επιπλέον, η μέθοδος αυτή μπορεί να είναι ευάλωτη σε επιθέσεις που έχουν ως σκοπό την υποκλοπή των δεδομένων. Σε μια τέτοια επίθεση, ένας κακόβουλος παράγοντας

μπορεί να δημιουργήσει ένα αντίγραφο των βιομετρικών δεδομένων ενός ατόμου και να το χρησιμοποιήσει για να το πλαστοπροσωπήσει. Με τον τρόπο αυτό θα μπορέσει να αποκτήσει πρόσβαση σε ένα σύστημα που προστατεύεται από έλεγχο ταυτότητας με το συγκεκριμένο βιομετρικό δεδομένο. Επιπλέον, τα βιομετρικά δεδομένα μπορούν να αποθηκευτούν σε μορφή που είναι εύκολα προσβάσιμη στους χάκερς. Σύμφωνα με δεδομένα του Kaspersky Security Network (KSN), το τρίτο τρίμηνο του 2019 μπλοκαρίστηκε κακόβουλο λογισμικό στο 37% των υπολογιστών που εκτελούν τις λειτουργίες συλλογής, επεξεργασίας και αποθήκευσης βιομετρικών δεδομένων – με άλλα λόγια, ένας στους τρεις υπολογιστές κινδύνευε να προσβληθεί από κακόβουλο λογισμικό. Μια ανάλυση των πηγών απειλών έδειξε ότι το Διαδίκτυο είναι η κύρια πηγή απειλών για τα συστήματα βιομετρικής επεξεργασίας δεδομένων (Kruglov, 2019).



Εικόνα 2: Κύριες πηγές απειλών για συστήματα επεξεργασίας και αποθήκευσης βιομετρικών δεδομένων, 3ο τρίμηνο 2019, πηγή: (Kruglov, 2019)

Μεταξύ των απειλών που αποκλείστηκαν στα συστήματα επεξεργασίας και αποθήκευσης βιομετρικών δεδομένων, το spyware, το κακόβουλο λογισμικό που χρησιμοποιείται σε επιθέσεις phishing (κυρίως προγράμματα λήψης και droppers), το ransomware και τα τραπεζικά Trojans αποτελούν τον μεγαλύτερο κίνδυνο για τέτοια συστήματα.



Εικόνα 3: Τύποι κακόβουλου λογισμικού που έχουν αποκλειστεί σε συστήματα επεξεργασίας και αποθήκευσης βιομετρικών δεδομένων-2019, πηγή: (Kruglov, 2019)

Ο βιομετρικός έλεγχος ταυτότητας μπορεί να παρουσιάσει προβλήματα χρηστικότητας. Για παράδειγμα, ορισμένα βιομετρικά συστήματα ελέγχου ταυτότητας απαιτούν από τους χρήστες να τοποθετούν το δάχτυλό τους ή το πρόσωπό τους με συγκεκριμένο τρόπο, προκειμένου να αναγνωρίζονται με ακρίβεια. Αυτό μπορεί να είναι άβολο, ιδιαίτερα σε περιπτώσεις όπου είναι απαραίτητος ο γρήγορος έλεγχος ταυτότητας (Frąckiewicz , 2023).

Τέλος, ένα σύστημα βιομετρικού ελέγχου ταυτότητας μπορεί να έχει υψηλό κόστος. Επιπλέον, το κόστος αντικατάστασης ή αναβάθμισης του υλικού ή του λογισμικού που χρησιμοποιείται στα βιομετρικά συστήματα ελέγχου ταυτότητας μπορεί επίσης να είναι δαπανηρό (Frąckiewicz , 2023).

2.4 Προκλήσεις και τάσεις

Ο βιομετρικός έλεγχος ταυτότητας παρουσιάζει κάποιες σημαντικές προκλήσεις, οι οποίες σχετίζονται με το κόστος, την αξιοπιστία, το βαθμό ασφάλειας και το υλικό που χρησιμοποιείται για τη λειτουργία της μεθόδου.

Αρχικά, ο βιομετρικός έλεγχος ταυτότητας βασίζεται σε εξειδικευμένο υλικό, όπως κάμερες και σαρωτές δακτυλικών αποτυπωμάτων, για την καταγραφή των φυσικών χαρακτηριστικών ενός ατόμου. Αυτό το υλικό μπορεί να είναι ακριβό και σε

ορισμένες περιπτώσεις, να μην είναι εύκολα προσβάσιμο. Η αναγνώριση δακτυλικών αποτυπωμάτων, προσώπου και φωνής αποτελούν τις πιο ελκυστικές επιλογές λόγω του χαμηλού κόστους εφαρμογής τους καθώς τα μικρόφωνα, οι κάμερες και οι σαρωτές είναι ήδη ευρέως διαδεδομένα (Kinzer, 2022). Επιπλέον, οι λύσεις cloud στη βιομετρία ή το Biometrics-as-a-Service (BaaS), θα κάνουν αυτήν την τεχνολογία ακόμη πιο προσιτή και πιο επεκτάσιμη. Το BaaS παρέχει βιομετρικές δυνατότητες ενσωμάτωσης και ελέγχου ταυτότητας σε πλατφόρμα cloud και εξαλείφει το κόστος που σχετίζεται με τη βάση δεδομένων, το δίκτυο και τα στοιχεία αποθήκευσης. Το μόνο στοιχείο υλικού που απαιτείται είναι η συσκευή βιομετρικής καταγραφής για την καταγραφή της μεμονωμένης βιομετρικής εισόδου, γεγονός που διευκολύνει την ανάπτυξη αυτών των λύσεων (Barra, και συν., 2018).

Οι προκλήσεις που σχετίζονται με την ανάπτυξη και τη χρήση των υπηρεσιών cloud αφορούν τη διασφάλιση του απορρήτου των δεδομένων με παράλληλη παροχή έγκαιρης και ασφαλούς πρόσβασης σε περιβάλλον υπολογιστικού νέφους. Ιδιαίτερα σε περιβάλλον πολλαπλού νέφους, μπορεί να είναι εξαιρετικά δύσκολη. Αυτό οφείλεται εν μέρει στους διαφορετικούς κανόνες απορρήτου και τους σχετικούς κανονισμούς και απαιτήσεις σχετικά με τη διαχείριση και την αποθήκευση δεδομένων μεταξύ των δικαιοδοσιών. Αυτό απαιτεί τη σχεδίαση συστήματος ελέγχου ταυτότητας που διασφαλίζει ότι τα δεδομένα είναι προσβάσιμα μόνο από εξουσιοδοτημένους χρήστες (Barra, και συν., 2018).

Επιπλέον, ο βιομετρικός έλεγχος ταυτότητας απαιτεί υψηλό επίπεδο ακρίβειας και αξιοπιστίας. Εάν η ακρίβεια δεν είναι υψηλή, μπορεί να οδηγήσει σε ψευδώς θετικά ή ψευδώς αρνητικά, με αποτέλεσμα την έλλειψη εμπιστοσύνης στην τεχνολογία (Frąckiewicz , 2023). Η αλληλουχία DNA είναι ο πιο ακριβής τύπος βιομετρικών δεδομένων, ακολουθούμενος από φυσιολογικά χαρακτηριστικά όπως τα μοτίβα αμφιβληστροειδούς, τα δακτυλικά αποτυπώματα και τη δομή του προσώπου. Οι λιγότερο ακριβείς τύποι βιομετρικών δεδομένων περιλαμβάνουν τα φωνητικά αποτυπώματα, τη γεωμετρία χεριών και τα ηλεκτροκαρδιογραφήματα (Molinaro, 2022). Στο πλαίσιο αυτό έχουν αναπτυχθεί συστήματα 3D αναγνώρισης προσώπου, τα οποία βασίζονται σε αισθητήρες για την αποτύπωση των χαρακτηριστικών του προσώπου με μεγαλύτερη λεπτομέρεια,

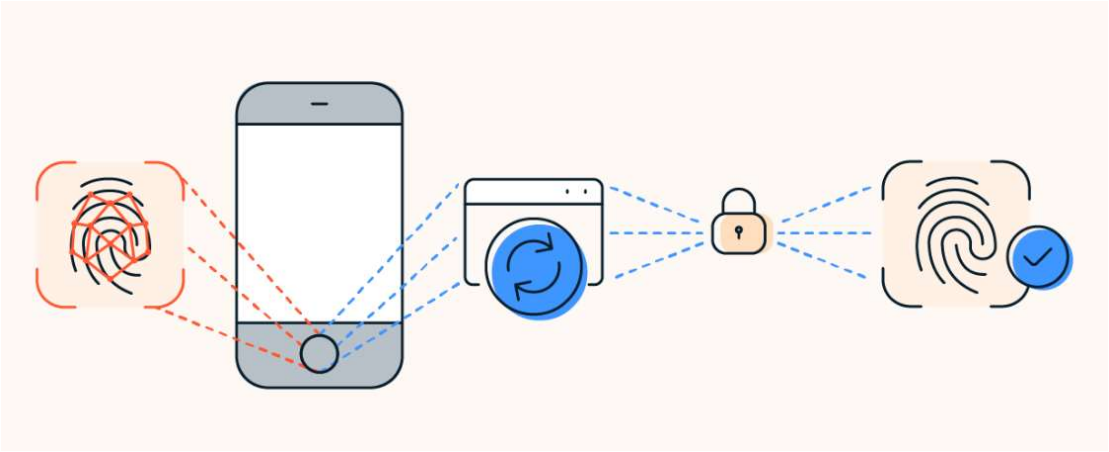
Μια άλλη πρόκληση με τον βιομετρικό έλεγχο ταυτότητας είναι ότι μπορεί να είναι ευάλωτος σε παραβιάσεις δεδομένων. Εάν τα βιομετρικά δεδομένα κλαπούν, δεν μπορούν εύκολα να αλλάξουν ή να επαναφερθούν (όπως ο κωδικός πρόσβασης), γεγονός που καθιστά δύσκολη την προστασία από μη εξουσιοδοτημένη πρόσβαση. Επιπλέον, τα βιομετρικά δεδομένα αποθηκεύονται συχνά σε κεντρικές βάσεις δεδομένων, οι οποίες μπορεί να είναι ευάλωτες σε κακόβουλες επιθέσεις (Frąckiewicz , 2023). Ο έλεγχος ταυτότητας δύο παραγόντων είναι μια διαδικασία ασφαλείας που απαιτεί δύο διαφορετικές μορφές αναγνώρισης από τον χρήστη για να συνδεθεί. Η μέθοδος αυτή χρησιμοποιεί τη βιομετρική αναγνώριση ως έναν παράγοντα και είτε ένα κωδικό πρόσβασης είτε άλλο διακριτικό ασφαλείας ως δεύτερο παράγοντα. Αυτός ο συνδυασμός θεωρείται αρκετά πιο ασφαλής. Γενικότερα, η πιο ασφαλής μορφή βιομετρικής ταυτοποίησης εξαρτάται από τη συγκεκριμένη εφαρμογή και το περιβάλλον στο οποίο χρησιμοποιείται (Kinzer, 2022).

Κεφάλαιο 3ο

Επεξεργασία & Αποθήκευση Βιομετρικών Δεδομένων

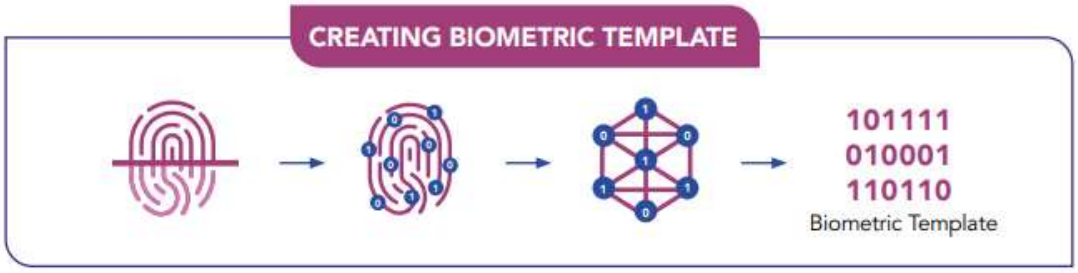
3.1 Συλλογή & Επεξεργασία Βιομετρικών Δεδομένων

Η συλλογή βιομετρικών δεδομένων περιλαμβάνει τρία στοιχεία: έναν αισθητήρα για τη συλλογή δεδομένων εισόδου, έναν υπολογιστή για την επεξεργασία και αποθήκευση τους και το λογισμικό που λειτουργεί ως ενδιάμεσος. Ο αισθητήρας καταγράφει τις βιομετρικές πληροφορίες, οι οποίες στη συνέχεια μετατρέπονται σε ψηφιακή μορφή, κρυπτογραφούνται και αποστέλλονται για μακροπρόθεσμη αποθήκευση. Όταν ένας χρήστης παρέχει τα βιομετρικά διαπιστευτήριά του στο σύστημα, το λογισμικό συγκρίνει τα δεδομένα με αυτά που είναι αποθηκευμένα στη βάση δεδομένων για τον έλεγχο ταυτότητας (Molinaro, 2022).



Εικόνα 4: Διαδικασία αποθήκευσης βιομετρικών δεδομένων, πηγή: (Molinaro, 2022)

Τα βιομετρικά δείγματα που καταγράφονται από τον αισθητήρα υποβάλλονται σε επεξεργασία για τη δημιουργία βιομετρικών προτύπων που χρησιμοποιούνται από τα συστήματα εφαρμογών.



Εικόνα 5: Μετατροπή βιομετρικού δείγματος (π.χ. δακτυλικού αποτυπώματος) σε βιομετρικό πρότυπο, πηγή: (Personal Data Protection Commission Singapore (PDPC), 2022)

Κατά την αποθήκευση, την επεξεργασία και τη χρήση βιομετρικών δεδομένων, το πρώτο στάδιο είναι η καταγραφή του βιομετρικού αναγνωριστικού ενός ατόμου. Από τη στιγμή που συλλαμβάνεται ένα κομμάτι βιομετρικών δεδομένων, δεν μπορεί να τροποποιηθεί. Αφού συλλεχθούν αυτά τα δεδομένα, στη συνέχεια αναλύονται και μετατρέπονται σε βιομετρικό πρότυπο, το οποίο συνιστά μια δυαδική μαθηματική αναπαράσταση του αρχικού βιομετρικού αναγνωριστικού. Ένα βιομετρικό πρότυπο δεν είναι ένα ακριβές αντίγραφο των βιομετρικών δεδομένων, αλλά ένα αρχείο που αντιπροσωπεύει μοναδικά αριθμητικά σημεία των δεδομένων που μετατρέπεται με έναν μυστικό, αποκλειστικό αλγόριθμο. Αυτό το πρότυπο δεν μπορεί να σχεδιαστεί αντίστροφα σε μια εικόνα δακτυλικού αποτυπώματος, προσώπου ή ίριδας. Ως εκ τούτου, τα ψηφιακά βιομετρικά δεδομένα είναι σημαντικά πιο ασφαλή από ένα ακριβές αντίγραφο ή μια φωτογραφία, καθώς χωρίς τον αποκλειστικό αλγόριθμο, κανείς δεν μπορεί να αποκωδικοποιήσει το βιομετρικό πρότυπο (NEC , 2022).

3.2 Στρατηγικές Αποθήκευσης Βιομετρικών Δεδομένων

Ένα κοινό βιομετρικό σύστημα ελέγχου ταυτότητας στοχεύει στον έλεγχο ταυτότητας των χρηστών ανεξάρτητα από το τι μπορεί να διαρρεύσει το σύστημα σχετικά με τα βιομετρικά διαπιστευτήρια του χρήστη σε τρίτους. Τέτοιες διαδικασίες προστατεύουν το απόρρητο στο στάδιο του σχεδιασμού αντί η προστασία να παρέχεται μέσω μιας μεταγενέστερης ενέργειας που υιοθετείται ως πρόσθετη υπηρεσία σε επόμενα στάδια. Αντίθετα, ένα σύστημα που διατηρεί το απόρρητο μετατρέπει τα βιομετρικά χαρακτηριστικά σε φορείς δεδομένων σε ασφαλείς τομείς, με τέτοιο τρόπο ώστε το σύστημα να μπορεί να εγγυηθεί την ανωνυμία του κατόχου του βιομετρικού

χαρακτηριστικού, ενώ μπορεί να διακρίνει μεταξύ των πελατών του συστήματος (Pagnin & Mitrokotsa, 2017).

Τα βιομετρικά πρότυπα είναι δυαδικά αρχεία και περιλαμβάνουν μοναδικά χαρακτηριστικά των βιομετρικών δεδομένων ενός ατόμου. Τα αρχεία αυτά είναι δυσανάγνωστα χωρίς τον σωστό αλγόριθμο. Υπάρχουν πολλές στρατηγικές αποθήκευσης για βιομετρικά δεδομένα με διαφορετικά χαρακτηριστικά η κάθε μία. Αυτές περιλαμβάνουν:

3.2.1 Αποθηκευτικός χώρος στη συσκευή

Τα βιομετρικά δεδομένα μπορούν να αποθηκευτούν στη συσκευή ενός τελικού χρήστη. Αυτό είναι πιο συνηθισμένο σε smartphone που χρησιμοποιούν αισθητήρες δακτυλικών αποτυπωμάτων αφής. Η αποθήκευση στη συσκευή μπορεί να χρησιμοποιηθεί για την αποθήκευση βιομετρικών δεδομένων μέσω ενός τσιπ που διατηρεί τα δεδομένα χωριστά στο δίκτυο της συσκευής. Κατά την αποθήκευση των δεδομένων στην ίδια τη συσκευή ελέγχου ταυτότητας, ο οργανισμός που εφαρμόζει τη διαδικασία βιομετρικής επαλήθευσης δεν έχει τον έλεγχό τους. Αυτός ο τύπος βιομετρικής αποθήκευσης είναι ιδιαίτερα ασφαλής επειδή δεν αποθηκεύει ευαίσθητα δεδομένα σε διακομιστές σε μεγάλες βάσεις δεδομένων (NEC , 2022).

3.2.2 Αρχιτεκτονική με επίκεντρο κάποιο διακομιστή

Σε αυτήν τη ρύθμιση, το βιομετρικό πρότυπο εγγράφεται και αποθηκεύεται κεντρικά σε έναν ασφαλή διακομιστή. Η αντιστοίχιση σε μια προσπάθεια ελέγχου ταυτότητας εκτελείται κεντρικά, σε αντίθεση με κάθε μεμονωμένη συσκευή. Κάθε φορά που ο χρήστης εκτελεί μια προσπάθεια επαλήθευσης, το δείγμα αποστέλλεται στην κεντρική μηχανή αντιστοίχισης, όπου υποβάλλεται σε επεξεργασία και αντιστοιχίζεται με το εγγεγραμμένο πρότυπο που είναι αποθηκευμένο κεντρικά. Κατά καιρούς, η τοπική αποθήκευση σε συσκευή δεν είναι εφικτή. Για παράδειγμα, οι μεγάλες εταιρείες που χρησιμοποιούν βιομετρικό έλεγχο ταυτότητας για την παραχώρηση ειδικής πρόσβασης και αδειών χρήστη, ενδέχεται να προτιμούν τη βιομετρική αποθήκευση σε βάση δεδομένων σε αντίθεση με την πρόσβαση μόνο σε τοπική συσκευή. Ένας διακομιστής είναι μία από τις πιο οικονομικές μεθόδους αποθήκευσης βιομετρικών δεδομένων, αν και είναι πιο επιρρεπής σε απειλές στον

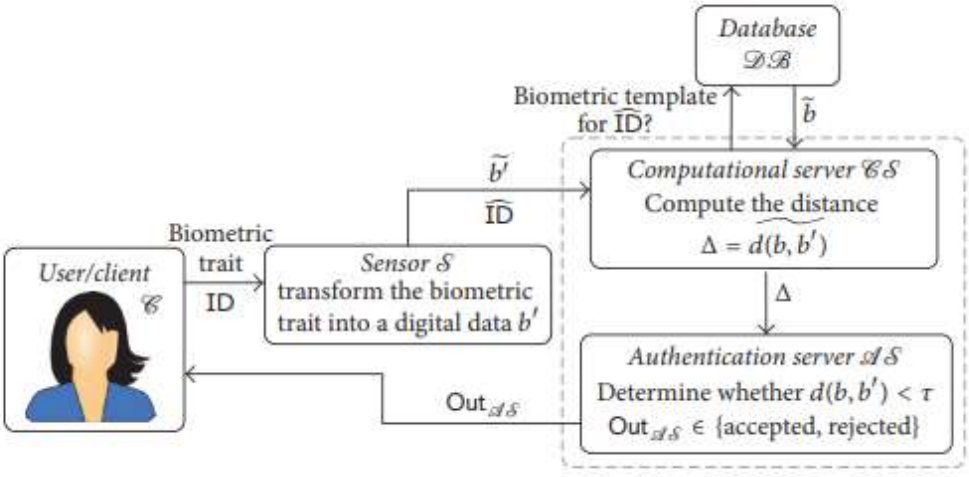
κυβερνοχώρο λόγω της προσέγγισης που βασίζεται στο δίκτυο. Καθώς τα δεδομένα διατηρούνται σε έναν εξωτερικό διακομιστή, ένα από τα πλεονεκτήματά του είναι ότι επιτρέπει μια διαδικασία επαλήθευσης πολλαπλών τοποθεσιών. Για να μειωθεί ο κίνδυνος παραβίασης των δεδομένων, θα πρέπει να είναι κρυπτογραφημένα κατά τη μεταφορά τους μέσω του δικτύου. Το πρόβλημα με την κρυπτογράφηση είναι το μέρος πού θα αποθηκευτούν τα κλειδιά κρυπτογράφησης και σε ποιον θα είναι αξιόπιστη η πρόσβαση. Ακόμη και αν το σύστημα παραβιαστεί και τα βιομετρικά στοιχεία «συλλεχθούν», χωρίς τον μυστικό και αποκλειστικό αλγόριθμο, τα βιομετρικά δεδομένα δεν μπορούν να ερμηνευθούν (NEC , 2022; Aware, Inc., 2023).

3.2.3 Κατανεμημένη αρχιτεκτονική αποθήκευσης δεδομένων

Η κατανεμημένη αρχιτεκτονική αποθήκευσης δεδομένων είναι μια μέθοδος που αποθηκεύει τα βιομετρικά πρότυπα σε έναν διακομιστή και μια συσκευή. Πρόκειται για μια μέθοδο που είναι ουσιαστικά μια λύση αποθήκευσης με διπλή υποστήριξη. Τα βιομετρικά δεδομένα θα χωριστούν σε μικρότερα, κρυπτογραφημένα αρχεία και θα αποθηκευτούν χωριστά στον διακομιστή και στο κέντρο αποθήκευσης της συσκευής ελέγχου ταυτότητας. Κάθε οντότητα που εμπλέκεται στη διαδικασία ελέγχου ταυτότητας εκτελεί μόνο μία εργασία. Πιο συγκεκριμένα, με την υιοθέτηση μιας κατανεμημένης αρχιτεκτονικής στη διαδικασία βιομετρικής επαλήθευσης ταυτότητας (π.χ. υπολογιστικός διακομιστής (Computational Server), διακομιστής ελέγχου ταυτότητας (Authentication Server), βάση δεδομένων (DataBase), είναι δυνατό να περιοριστεί ο όγκος των πληροφοριών που έχει στη διάθεσή της κάθε οντότητα και έτσι να αποφευχθεί μια συνολική αποτυχία του συστήματος η ο οποία θα προκληθεί από ένα σημείο. Επιπλέον, μια κατανεμημένη αρχιτεκτονική παρέχει υψηλότερες εγγυήσεις απορρήτου, καθώς καμία οντότητα δεν έχει πρόσβαση σε όλα τα ευαίσθητα δεδομένα. Αποθηκεύοντας τα δεδομένα με αυτόν τον τρόπο, καθιστά πιο δύσκολη την πρόσβαση ενός επιτήδειου στα δεδομένα, καθώς θα πρέπει να μπει και στα δύο σημεία. Αυτή η μέθοδος προσφέρει ασφάλεια και απόρρητο χωρίς να θυσιάζει τη χρηστικότητα ή την επεκτασιμότητα (NEC , 2022; Pagnin & Mitrokotsa, 2017).

Ειδικότερα, η διαδικασία έχει ως εξής (εικόνα 2): Ο χρήστης παρέχει το βιομετρικό του χαρακτηριστικό (μέσω του αισθητήρα S) μαζί με την ταυτότητά του. Στη

συνέχεια, αυτές οι δύο πληροφορίες επεξεργάζονται από τον αισθητήρα και μεταδίδονται στον υπολογιστικό διακομιστή CS, ως b' (π.χ. κρυπτογράφηση του νέου προτύπου) και $\widehat{ID}$ (π.χ. ψευδώνυμο). Ο υπολογιστικός διακομιστής CS υποβάλλει ερώτημα στη βάση δεδομένων DB για το αποθηκευμένο πρότυπο $\tilde{b}$ που συνδέεται με το $\widehat{ID}$. Μετά τη λήψη του $\tilde{b}$, το CS υπολογίζει την (πιθανώς κρυπτογραφημένη) απόσταση d μεταξύ b' και b. Έστω $\Delta = (b, \tilde{b'})$ το αποτέλεσμα που στέλνει το CS στο AS. Ο διακομιστής ελέγχου ταυτότητας χρησιμοποιεί το Δ για να εξαγάγει την πραγματική απόσταση μεταξύ b' και b και τη συγκρίνει με το τ, το όριο του συστήματος. Το όριο τ μπορεί να θεωρηθεί ως το επίπεδο ακρίβειας του συστήματος. Πράγματι, εάν τα πρότυπα είναι αρκετά κοντά (δηλ. $d(b, b') < \tau$) ο χρήστης επαληθεύεται, διαφορετικά ο χρήστης απορρίπτεται.



Εικόνα 6: Η φάση ελέγχου ταυτότητας σε ένα βιομετρικό σύστημα ελέγχου ταυτότητας με κατανεμημένη αρχιτεκτονική, πηγή: (Pagnin & Mitrokotsa, 2017)

3.2.4 Αποθήκευση δεδομένων Blockchain

Τα κρυπτοσυστήματα στην αναγνώριση με βιομετρία μπορούν να κατηγοριοποιηθούν ευρέως σε εκείνα που αντλούν το κλειδί απευθείας από το βιομετρικό χαρακτηριστικό που αποκτάται εν κινήσει και σε αυτά που δημιουργούν το κλειδί δεσμεύοντας το βιομετρικό χαρακτηριστικό και ένα τυχαίο δυαδικό κλειδί. Και στις δύο περιπτώσεις, το βιομετρικό χαρακτηριστικό δεν χρειάζεται να αποθηκευτεί (εκτός κατά την εγγραφή όταν τα αποκτηθέντα βιομετρικά δεδομένα χρησιμοποιούνται για τη δημιουργία του κρυπτογραφημένου κλειδιού). Μόλις εγγραφεί επιτυχώς ένας

χρήστης, οι πληροφορίες από το αποκτηθέν αρχικό βιομετρικό χαρακτηριστικό δεν χρησιμοποιούνται πλέον ούτε αποθηκεύονται (Barra, και συν., 2018).

Ωστόσο, όταν ο βιομετρικός έλεγχος ταυτότητας πραγματοποιείται σε αρχιτεκτονικές νέφους, ενδέχεται να προκύψουν πιθανές επιθέσεις στο απόρρητο κατά τη μετάδοση του αποκτηθέντος βιομετρικού χαρακτηριστικού μέσω του δικτύου. Για παράδειγμα, οι επιθέσεις πλαστογράφησης μπορούν να οδηγήσουν σε κλοπή ταυτότητας, που είναι ιδιαίτερα κρίσιμο στα βιομετρικά, λόγω της αδυναμίας αλλαγής του χαρακτηριστικού των χρηστών. Μια εγγραφή βίντεο ή, σε ορισμένες περιπτώσεις, μια φωτογραφία του εξουσιοδοτημένου ατόμου, μπορεί να χρησιμοποιηθεί για πρόσβαση σε προστατευμένα δεδομένα (Barra, και συν., 2018).

3.2.5 Σύστημα αναγνώρισης υλικού

Ένα σύστημα αναγνώρισης που βασίζεται σε υλικό είναι όπου τα δεδομένα αποθηκεύονται σε ένα συγκεκριμένο υλικό και το οποίο συνεργάζεται με τη συσκευή για να αναγνωρίσει τα δεδομένα, χωρίς να αποθηκεύει τα δεδομένα στην ίδια τη συσκευή. Αυτό προσφέρει γρήγορη απόκριση κατά τον έλεγχο ταυτότητας χρήστη, καθώς τα βιομετρικά πρότυπα αποθηκεύονται τοπικά και το σύστημα αναγνώρισης δεν απαιτεί εξωτερική απόκριση (NEC , 2022).

3.2.6 Φορητό διακριτικό

Τα βιομετρικά στοιχεία που είναι αποθηκευμένα σε φορητές μάρκες κάρτες ασφαλείας ή μονάδες USB, λειτουργούν σχεδόν με τον ίδιο τρόπο όπως η βιομετρική αποθήκευση στη συσκευή. Οι βιομετρικές πληροφορίες αποθηκεύονται σε μία μόνο συσκευή και αυτή η συσκευή πρέπει να παρουσιάζεται κατά τον έλεγχο ταυτότητας για σκοπούς επαλήθευσης. Τα βιομετρικά διακριτικά τείνουν να είναι λίγο πιο δαπανηρά στην εφαρμογή τους, επειδή απαιτούν τόσο το διακριτικό όσο και έναν ξεχωριστό βιομετρικό σαρωτή, αν και το πρόσθετο βήμα προσθέτει επίσης μια άλλη γραμμή ασφάλειας. Ένα άλλο πλεονέκτημα της αποθήκευσης βιομετρικών δεδομένων σε ένα φορητό διακριτικό είναι ότι δεν χρειάζεται να μεταφερθούν μέσω δικτύου για σκοπούς επαλήθευσης, και έτσι αυτό μειώνει τους κινδύνους που μπορεί να προκύψουν από ευπάθειες που σχετίζονται με το δίκτυο (NEC , 2022).

3.3 Στάδια στη σάρωση δακτυλικών αποτυπωμάτων

Τα δακτυλικά αποτυπώματα είναι φτιαγμένα από σειρές ραβδώσεων και αυλακιών στην επιφάνεια του δακτύλου και έχουν έναν πυρήνα γύρω από τον οποίο σχηματίζονται μοτίβα σαν τόξα, ώστε να διασφαλίζεται ότι κάθε αποτύπωμα είναι μοναδικό. Το τόξο είναι ένα σχέδιο όπου οι κορυφογραμμές εισέρχονται από τη μία πλευρά του δακτύλου, ανεβαίνουν στο κέντρο σχηματίζοντας ένα τόξο και μετά βγαίνουν από την άλλη πλευρά του δακτύλου. Οι κορυφογραμμές και τα αυλάκια χαρακτηρίζονται από ανωμαλίες, οι οποίες αποτελούν το βασικό χαρακτηριστικό στο οποίο βασίζονται οι τεχνολογίες σάρωσης δακτυλικών αποτυπωμάτων (Leonard & Ezeonyi, 2020).

Υπάρχουν πέντε στάδια που εμπλέκονται στην επαλήθευση με δακτυλική σάρωση και στον εντοπισμό πληροφοριών σε ένα βιομετρικό σύστημα. Περιλαμβάνουν (Leonard & Ezeonyi, 2020):

1. Την απόκτηση της εικόνας δακτυλικών αποτυπωμάτων.
2. Την επεξεργασία της εικόνας.
3. Τον εντοπισμό των διακριτικών χαρακτηριστικών.
4. Τη δημιουργία του προτύπου.
5. Το ταίριασμα του προτύπου.

Κεφάλαιο 4ο

Software & Hardware για επαλήθευση μέσω δακτυλικού αποτυπώματος

4.1 Παράγοντες που επηρεάζουν την επιλογή Software

Μεταξύ των διαφορετικών μεθόδων για τον έλεγχο ταυτότητας χρήστη, η αναγνώριση μέσω δακτυλικών αποτυπωμάτων είναι μία από τις πιο ευρέως χρησιμοποιούμενες και αξιόπιστες. Η επιλογή του κατάλληλου λογισμικού είναι κρίσιμη για την επιτυχία οποιουδήποτε εγχειρήματος για βιομετρικό έλεγχο ταυτότητας. Παράγοντες όπως η συμβατότητα με τον σαρωτή δακτυλικών αποτυπωμάτων, η ακρίβεια και η ταχύτητα αντιστοίχισης δακτυλικών αποτυπωμάτων, η ευκολία ενσωμάτωσης, η ασφάλεια και το απόρρητο δεδομένων, το κόστος, η υποστήριξη και η εξυπηρέτηση πελατών θα πρέπει να λαμβάνονται υπόψη κατά την επιλογή λογισμικού για τη σάρωση των δακτυλικών αποτυπωμάτων.

Παρακάτω αναλύονται οι παράγοντες που πρέπει να λαμβάνονται υπόψη κατά την επιλογή του λογισμικού (Palma, 2023; Leonard & Ezeonyi, 2020).

4.1.1 Συμβατότητα με τον σαρωτή δακτυλικών αποτυπωμάτων

Ο πρώτος παράγοντας κατά την επιλογή λογισμικού αφορά τη συμβατότητα με τον ίδιο τον σαρωτή δακτυλικών αποτυπωμάτων. Δεν είναι όλο το λογισμικό σαρωτών δακτυλικών αποτυπωμάτων συμβατό με όλους τους τύπους σαρωτών δακτυλικών αποτυπωμάτων και πρέπει να διασφαλιστεί ότι το λογισμικό που επιλέγεται λειτουργεί άψογα με τον σαρωτή που είναι διαθέσιμος για τη λειτουργία του συστήματος. Επιπλέον, υπάρχουν προγράμματα τα οποία είναι παράλληλα συμβατά και με άλλους τύπους σαρωτές πχ για την ίριδα

4.1.2 Ακρίβεια και ταχύτητα αντιστοίχιση δακτυλικών αποτυπωμάτων

Η ακρίβεια και η ταχύτητα της αντιστοίχισης δακτυλικών αποτυπωμάτων είναι κρίσιμοι παράγοντες κατά την επιλογή λογισμικού. Το λογισμικό θα πρέπει να είναι σε θέση να αντιστοιχίζει γρήγορα και με ακρίβεια τα δακτυλικά αποτυπώματα για να εξασφαλίσει μια ομαλή εμπειρία χρήστη και να μειώσει τον κίνδυνο ψευδών θετικών

ή αρνητικών αποτελεσμάτων. Επίσης, για έργα βιομετρικής ταυτοποίησης μεγάλης κλίμακας, όπως συστήματα εθνικών ταυτοτήτων, εγγραφή ψηφοφόρων κ.α. το λογισμικό θα πρέπει να μπορεί να συγκρίνει εκατομμύρια πρότυπα δακτυλικών αποτυπωμάτων ISO[5] ανά δευτερόλεπτο για γρήγορη αναγνώριση και κατάργηση αντιγραφής.

4.1.3 Ευκολία ενσωμάτωσης

Ένας άλλος σημαντικός παράγοντας που πρέπει να λαμβάνεται υπόψη είναι ο βαθμός ευκολίας την ενσωμάτωσης του λογισμικού με τα υπάρχοντα συστήματα, εφαρμογές και ροές εργασίας της κάθε επιχείρησης ή οργανισμού. Το λογισμικό θα πρέπει να διαθέτει εργαλεία προγραμματιστών και APIs[6] που επιτρέπουν στους οργανισμούς να προσαρμόζουν γρήγορα και να ενσωματώνουν τον βιομετρικό έλεγχο ταυτότητας στις εφαρμογές τους. Επιπλέον, πρέπει να υποστηρίζει διάφορες γλώσσες προγραμματισμού, όπως C#, Java, PHP, C/C++, Delphi και VB, καθιστώντας εύκολη την ενσωμάτωση με υπάρχουσες εφαρμογές.

4.1.4 Ασφάλεια Δεδομένων και Απόρρητο

Η ασφάλεια και το απόρρητο δεδομένων θα πρέπει να αποτελούν κορυφαία προτεραιότητα κατά την επιλογή λογισμικού. Το λογισμικό θα πρέπει να χρησιμοποιεί πρωτόκολλα κρυπτογράφησης υψηλών προτύπων για να διασφαλίζει ότι τα βιομετρικά δεδομένα είναι ασφαλή και προστατευμένα από μη εξουσιοδοτημένη πρόσβαση.

4.1.5 Μοντέλο τιμολόγησης

Το μοντέλο τιμολόγησης είναι ένας άλλος κρίσιμος παράγοντας κατά την επιλογή λογισμικού. Ορισμένες λύσεις απαιτούν από τις επιχειρήσεις να επενδύσουν σε

---

[5] Το ISO/IEC 19794-2:2011 καθορίζει μια έννοια και μορφές δεδομένων για την αναπαράσταση των δακτυλικών αποτυπωμάτων χρησιμοποιώντας τη θεμελιώδη έννοια των μικροσκοπικών στοιχείων. Είναι γενικό, καθώς μπορεί να εφαρμοστεί και να χρησιμοποιηθεί σε ένα ευρύ φάσμα περιοχών εφαρμογών όπου εμπλέκεται η αυτοματοποιημένη αναγνώριση δακτυλικών αποτυπωμάτων.

[6]Το *API* (application programming interface) ή *διεπαφή προγραμματισμού εφαρμογών*, χρησιμοποιείται για τη μεταβίβαση δεδομένων μεταξύ εφαρμογών λογισμικού με έναν τυποποιημένο τρόπο.

μεγάλο βαθμό στο κόστος υποδομής και συντήρησης. Αντίθετα, άλλοι χρησιμοποιούν ένα μοντέλο λογισμικού ως υπηρεσία (SaaS-software-as-a-service) που επιτρέπει στις επιχειρήσεις να πληρώνουν μόνο για ότι χρησιμοποιούν. Έτσι, μπορεί να εξαλειφθεί το κόστος που σχετίζεται με την κατασκευή και τη διατήρηση ενός αξιόπιστου συστήματος βιομετρικής αναγνώρισης.

4.1.6 Υποστήριξη και Εξυπηρέτηση Πελατών

Ο πάροχος λογισμικού θα πρέπει να διαθέτει μια ομάδα υποστήριξης που να ανταποκρίνεται και να μπορεί να παρέχει βοήθεια όταν χρειάζεται. Ειδικότερα, πρέπει να παρέχει τεχνική βοήθεια και να μπορεί να βοηθήσει με οποιαδήποτε ζητήματα αντιμετωπίζουν οι πελάτες κατά τη διαδικασία ολοκλήρωσης και υλοποίησης.

4.2 Γλώσσες Προγραμματισμού

4.2.1 Python

Η Python είναι μια ευέλικτη και δυνατή γλώσσα προγραμματισμού που έχει γίνει δημοφιλής για την ανάπτυξη προηγμένων εφαρμογών, συμπεριλαμβανομένων των συστημάτων αναγνώρισης δακτυλικών αποτυπωμάτων. Έχει απλή και κατανοητή σύνταξη, που καθιστά ευκολότερη την κατανόηση και την ανάπτυξη του κώδικα. Επίσης, έχει μια πολύ ενεργή κοινότητα που συνεχώς επεκτείνει και βελτιώνει τη γλώσσα. Η Python διαθέτει πλούσια βιβλιοθήκη που περιλαμβάνει πακέτα όπως η OpenCV για την επεξεργασία εικόνων και η scikit-learn για την μηχανική μάθηση. Αυτές οι βιβλιοθήκες μπορούν να βοηθήσουν στην ανάλυση και την αναγνώριση δακτυλικών αποτυπωμάτων. Επιπλέον, η Python παρέχει μεγάλη ευελιξία και δυνατότητες για την προσαρμογή των εφαρμογών σύμφωνα με τις απαιτήσεις του προγράμματος (Python Software Foundation, 2023; Bukhari, 2022).

Τα βήματα για τον έλεγχο ταυτότητας μέσω δακτυλικών αποτυπωμάτων στην Python είναι τα εξής (Bukhari, 2022):

1. Εισαγωγή των απαραίτητων βιβλιοθηκών

2. Εισαγωγή του απαιτούμενου συνόλου δεδομένων
3. Διάβασμα των πραγματικών[7] εικόνων δακτυλικών αποτυπωμάτων
4. Δημιουργία αντικείμενου SIFT[8]
5. Εντοπισμός των βασικών σημείων και υπολογισμός των περιγραφικών παραγόντων
6. Εντοπισμός της καλύτερης αντιστοίχισης απόστασης μεταξύ σημείων-κλειδιών και περιγραφικών παραγόντων.
7. Αντιστοίχιση των σημείων κλειδιών για έλεγχο ταυτότητας με δακτυλικά αποτυπώματα σε python
8. Επιλογή του δακτυλικού αποτυπώματος με την καλύτερη αντιστοίχιση και βαθμολογία

4.2.2 Java

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού με υψηλή ασφάλεια και επεκτασιμότητα. Χρησιμοποιείται ευρέως για την ανάπτυξη εφαρμογών ασφαλείας και βιομετρίας, χάρη στην ικανότητά του να αντιμετωπίζει μεγάλη ποικιλία βιβλιοθηκών και εργαλείων. Η πλατφόρμα Android, η οποία βασίζεται στη Java, παρέχει προσβασιμότητα σε βιομετρικά δεδομένα όπως δακτυλικά αποτυπώματα. Παρόλα αυτά, η Java είναι λιγότερο ευέλικτη σε σύγκριση με την Python και η καμπύλη μάθησης είναι αυξημένη (Oracle, 2023).

4.2.3 JavaScript

Η JavaScript είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται κυρίως για τη δημιουργία διαδραστικών ιστοσελίδων και εφαρμογών. Μπορεί να χρησιμοποιηθεί για την ανάπτυξη διαδραστικών διαδικασιών σύνδεσης βάσεις δακτυλικών αποτυπωμάτων σε διαδικτυακές εφαρμογές. Ωστόσο, η δυνατότητά της να διαχειρίζεται βιομετρικά δεδομένα εξαρτάται από τις δυνατότητες του browser και

---

[7] Οι πραγματικές εικόνες είναι οι αμετάβλητες εικόνες δακτυλικών αποτυπωμάτων στο σύνολο δεδομένων. Έτσι, ο σκοπός είναι να συγκρίνουμε τελικά το δείγμα τροποποιημένης εικόνας με εικόνες πραγματικών δακτυλικών αποτυπωμάτων στη μεταβλητή εικόνας δακτυλικών αποτυπωμάτων.

[8] Είναι ένας αλγόριθμος χαρακτηριστικών που βρίσκει τα βασικά σημεία της εικόνας. Σε αυτήν την περίπτωση, το αντικείμενο SIFT ανιχνεύει τις ραβδώσεις στα δακτυλικά αποτυπώματα.

την υποστήριξη από το λειτουργικό σύστημα. Το JavaScript δεν είναι κατάλληλο για σύνθετες επεξεργασίες εικόνων και αναγνώρισης προτύπων, που απαιτούνται για την αναγνώριση δακτυλικών αποτυπωμάτων (developer.mozilla.org, 2023).

4.2.4 C++

Η C++ είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού με υψηλή απόδοση και εξαιρετικές δυνατότητες για χαμηλού επιπέδου προγραμματισμό. Είναι γνωστή για την υψηλή της απόδοση και μπορεί να χρησιμοποιηθεί για την ανάπτυξη εξελιγμένων συστημάτων αναγνώρισης δακτυλικών αποτυπωμάτων. Ωστόσο, η ανάπτυξη εφαρμογών σε C++ είναι συνήθως πιο πολύπλοκη και χρονοβόρα. Η C++ δεν διαθέτει τόσο εκτεταμένη τυποποίηση ή βιβλιοθήκες όσο η Python, γεγονός που μπορεί να καθιστά την προγραμματιστική διαδικασία πιο δύσκολη.

4.3 Σύστημα διαχείρισης βάσεων δεδομένων

4.3.1 PostgreSQL

Η PostgreSQL είναι ένα σύστημα διαχείρισης βάσεων δεδομένων ανοιχτού κώδικα, ιδιαίτερα προηγμένο και φιλικό προς στην κοινότητα των προγραμματιστών. Χαρακτηρίζεται από την ικανότητά της να υποστηρίξει τόσο μικρές όσο και μεγάλες εφαρμογές, παρέχοντας υψηλής ποιότητας διαχείριση δεδομένων και ευέλικτη δυνατότητα προσαρμογής. Η PostgreSQL υποστηρίζει πλήρως το πρότυπο SQL και περιλαμβάνει πολλές προηγμένες λειτουργίες, όπως τα Window Functions και Common Table Expressions. Ένας άλλος σημαντικός παράγοντας είναι η υψηλή απόδοση και η αξιοπιστία της PostgreSQL, που επιτρέπει την αντιμετώπιση μεγάλων όγκων δεδομένων και ταυτόχρονων ερωτήσεων. Επιπλέον, η PostgreSQL παρέχει μια ασφαλή και ισχυρή αρχιτεκτονική που προστατεύει τα δεδομένα από επιθέσεις ή απώλειες (The PostgreSQL Global Development Group, 2023).

4.3.2 MySQL

Το MySQL είναι ένα δημοφιλές ανοιχτού κώδικα σύστημα διαχείρισης βάσεων δεδομένων που χρησιμοποιείται ευρέως για την ανάπτυξη ιστοσελίδων και εφαρμογών. Είναι ευέλικτο και παρέχει υψηλή απόδοση για μικρά έως μεσαία σε μέγεθος σετ δεδομένων. Ωστόσο, η MySQL έχει περιορισμένες δυνατότητες όσον

αφορά την προηγμένη επεξεργασία δεδομένων και δεν υποστηρίζει πλήρως το πρότυπο SQL όπως η PostgreSQL (Oracle, 2023).

4.3.3 SQLite

Η SQLite είναι μια ελαφριά βάση δεδομένων που είναι ιδανική για μικρές εφαρμογές, όπως κινητά εφαρμογές ή εφαρμογές σε ενιαίο υπολογιστή. Είναι πολύ απλή στη χρήση και δεν χρειάζεται συγκεκριμένη εγκατάσταση ή ρύθμιση. Ωστόσο, δεν είναι κατάλληλη για πολύπλοκες εφαρμογές ή για την χειρισμό μεγάλων όγκων δεδομένων, καθώς δεν υποστηρίζει πολλούς χρήστες ταυτόχρονα, όπως η PostgreSQL (SQLite , 2023).

4.3.4 Oracle Database

Η Oracle Database είναι ένας ισχυρός εμπορικός διαχειριστής βάσεων δεδομένων που είναι γνωστός για την υψηλή απόδοση, την ευρεία λειτουργικότητα και την αξιοπιστία του. Παρέχει υψηλού επιπέδου ασφάλεια και δυνατότητες διαχείρισης για μεγάλες βάσεις δεδομένων. Ωστόσο, είναι ακριβή και η σύνθετη αρχιτεκτονική της κάνει την εκμάθηση και τη χρήση της πιο δύσκολη σε σχέση με την PostgreSQL (Oracle, 2023).

4.4 Βιβλιοθήκες για τον προγραμματισμό GPIO

4.4.1 RPi.GPIO

Η RPi.GPIO είναι μια βιβλιοθήκη Python που σχεδιάστηκε για τον προγραμματισμό GPIO στον Raspberry Pi. Παρέχει πλήρη έλεγχο των GPIO pins, δίνοντάς σας την ευελιξία για την υλοποίηση πολύπλοκων λειτουργιών. Επιπλέον, είναι ιδιαίτερα κατάλληλη για την εκμάθηση της Python και του προγραμματισμού GPIO στον Raspberry Pi. Είναι επίσης πολύ καλά τεκμηριωμένη και ευρέως υποστηρίζεται από την κοινότητα του Raspberry Pi (Tranter, 2019).

4.4.2 WiringPi

Η WiringPi είναι μια βιβλιοθήκη για τον Raspberry Pi που παρέχει πρόσβαση στις δυνατότητες GPIO της συσκευής. Αν και παρέχει ευκολία χρήσης με την ομοιότητά της με τον προγραμματισμό Arduino, η υποστήριξη της WiringPi έχει σταματήσει, το

οποίο σημαίνει ότι οι νέες ενημερώσεις και τα patches δεν είναι πλέον διαθέσιμα (Wiringpi.com, 2023).

4.4.3 GPIO Zero

Η GPIO Zero είναι μια άλλη βιβλιοθήκη Python για τον προγραμματισμό GPIO στον Raspberry Pi. Είναι πιο φιλική προς τον χρήστη σε σχέση με την RPi.GPIO, καθώς παρέχει έναν απλοποιημένο και πιο καθαρό συντακτικό. Ωστόσο, παρέχει λιγότερο έλεγχο και ευελιξία σε σχέση με την RPi.GPIO (Ben Nuttall Revision, 2021).

4.4.4 pigpio

Η pigpio είναι μια βιβλιοθήκη για τον Raspberry Pi που επιτρέπει τον έλεγχο των GPIO pins της συσκευής. Παρέχει λειτουργίες για τον έλεγχο της PWM (Pulse Width Modulation) και της servo control, αλλά είναι λιγότερο φιλική προς τον χρήστη σε σχέση με την RPi.GPIO (abyz.me.uk, 2023).

4.5 Πλαίσια για την ανάπτυξη ιστοσελίδων

4.5.1 Flask

Το Flask είναι ένα ελαφρύ πλαίσιο διακομιστή ιστού για Python που επικεντρώνεται στην απλότητα και την ευελιξία. Προσφέρει την ισχύ της Python με μια απλή και καθαρή διεπαφή προγραμματισμού. Η απλότητα του Flask το καθιστά ιδανικό για μικρά σε μέγεθος έργα και εφαρμογές όπου η υπερβολική πολυπλοκότητα θα μπορούσε να είναι εμπόδιο. Επίσης, το Flask είναι ευέλικτο και επεκτάσιμο, παρέχοντας έναν απλό τρόπο για την προσθήκη νέων λειτουργιών όπως χρειάζονται. Ωστόσο, το Flask δεν είναι μόνο για μικρά έργα. Με την κατάλληλη δομή και την επέκταση, το Flask μπορεί να χειριστεί ακόμη και μεγάλες εφαρμογές ιστού. Το Flask διαθέτει επίσης έναν από τους πιο ενεργούς και φιλικούς προς τον χρήστη κύκλους ανάπτυξης, κάτι που είναι σημαντικό για την επίλυση προβλημάτων και την εκμάθηση καλύτερων πρακτικών (Pallets, 2010).

4.5.2 Django

Το Django είναι ένα πλούσιο πλαίσιο για την ανάπτυξη ιστοσελίδων σε Python. Παρέχει πολλές ενσωματωμένες λειτουργίες, όπως την αυθεντικοποίηση χρηστών και τον έλεγχο πρόσβασης, αλλά μπορεί να είναι υπερβολικά περίπλοκο για απλές εφαρμογές (Django Software Foundation , 2023).

4.5.3 FastAPI

Το FastAPI είναι ένα σύγχρονο, γρήγορο (υψηλής απόδοσης), πλαίσιο βασισμένο στα πρότυπα για την κατασκευή API σε Python 3.6+ με τη χρήση των τύπων πεδίων Python. Είναι εύκολο στη χρήση και ισχυρό, αλλά δεν έχει τόσο μεγάλη κοινότητα υποστήριξης όσο το Flask (fastapi.tiangolo.com, 2023).

4.5.4 Bottle

Το Bottle είναι ένα ελαφρύ πλαίσιο για μικρές εφαρμογές ιστού σε Python. Είναι ακόμα πιο απλό και ελαφρύ από το Flask, αλλά δεν παρέχει την ίδια ευελιξία και τον έλεγχο όσο το Flask (bottlepy.org, 2023).

4.6 Λογισμικό ORM

4.6.1 SQLAlchemy

Το SQLAlchemy είναι ένα πλούσιο ORM που παρέχει ισχυρές, ευέλικτες και λεπτομερείς λειτουργίες διαχείρισης SQL databases. Αυτό το εργαλείο επιτρέπει την πραγματοποίηση πολύπλοκων ερωτημάτων και συναλλαγών στη βάση δεδομένων, ενώ παράλληλα παρέχει την ευκολία του υψηλού επιπέδου ORM. Το SQLAlchemy προσφέρει μια πολύ ισχυρή και ευέλικτη διεπαφή που επιτρέπει την εκτέλεση πολύπλοκων ερωτημάτων SQL με σχετική ευκολία, παρέχοντας παράλληλα μια σειρά επιπλέον λειτουργιών, όπως το mapping αντικειμένων και τον χειρισμό συναλλαγών. Το SQLAlchemy έχει επίσης μια μεγάλη και ενεργή κοινότητα που παρέχει συνεχή υποστήριξη και αναβαθμίσεις. Σε σύγκριση με τα Peewee, Django ORM και SQLObject, το SQLAlchemy παρέχει ένα πιο ισχυρό και ευέλικτο εργαλείο για τη διαχείριση βάσεων δεδομένων SQL, καθιστώντας το ιδανικό για περίπλοκες

εφαρμογές που απαιτούν λεπτομερή έλεγχο της βάσης δεδομένων (DataCamp, Inc., 2022).

4.6.2 Peewee

Το Peewee είναι ένα απλό ORM για Python που υποστηρίζει MySQL, PostgreSQL και SQLite. Είναι ελαφρύ και εύκολο στη χρήση, αλλά δεν παρέχει την ίδια πληθώρα λειτουργιών και ευελιξίας που προσφέρει το SQLAlchemy (GitHub., 2023).

4.6.3 Django ORM

Το Django ORM παρέχει έναν πολύ ισχυρό και ευέλικτο τρόπο για τη διαχείριση των βάσεων δεδομένων SQL στο Python. Ωστόσο, είναι στενά ενσωματωμένο στο Django και μπορεί να είναι υπερβολικό για μικρές εφαρμογές που δεν απαιτούν όλες τις λειτουργίες του Django (Django Software Foundation, 2023).

4.6.4 SQLObject

Το SQLObject είναι ένα ORM για Python που παρέχει μια απλή και ευέλικτη διεπαφή για τη διαχείριση SQL databases. Αν και είναι ευκολότερο στη χρήση από το SQLAlchemy, δεν έχει την ίδια ικανότητα ελέγχου και χειρισμού πολυπλοκότητας (sqlobject.org, 2023).

4.7 Παράγοντες που επηρεάζουν την επιλογή Hardware

Υπάρχει ένα ευρύ φάσμα παραγόντων που πρέπει να ληφθούν υπόψη κατά την επιλογή της κατάλληλης τεχνολογίας αισθητήρα δακτυλικών αποτυπωμάτων και φυσικού αισθητήρα για χρήση σε ένα συγκεκριμένο προϊόν, εφαρμογή ή διαδικασία. Η επιλογή της τεχνολογίας θα εξαρτηθεί από παραμέτρους απόδοσης όπως η ποιότητα εικόνας, η ταχύτητα και η κατανάλωση ενέργειας. Κατά το σχεδιασμό του «πραγματικού» προϊόντος πρέπει επίσης να συμπεριληφθούν στην εξίσωση περαιτέρω παράμετροι, όπως το μέγεθος του αισθητήρα, το κόστος και οι επιλογές συσκευασίας. Σε αυτήν την ενότητα θα αναπτύξουμε μερικούς από τους πιο σημαντικούς παράγοντες για την επιλογή της τεχνολογίας και του αισθητήρα δακτυλικών αποτυπωμάτων.

4.7.1 Ποιότητα και ανάλυση εικόνας

Η ποιότητα της εικόνας που δημιουργείται από τον αισθητήρα δακτυλικών αποτυπωμάτων είναι μια θεμελιώδης και σημαντική παράμετρος. Η υψηλή ποιότητα εικόνας επιτρέπει μικρότερους αισθητήρες και χαμηλότερο κόστος, καθώς περισσότερες λεπτομέρειες καταγράφονται ανά μονάδα επιφάνειας. Η ποιότητα της εικόνας εξαρτάται από την ικανότητα του αισθητήρα να ανιχνεύει αδύναμα σήματα και να φιλτράρει τον ανεπιθύμητο θόρυβο. Η ποιότητα της εικόνας μπορεί να μετρηθεί με διάφορους τρόπους, αλλά μια κοινή μέτρηση στα συστήματα αναγνώρισης δακτυλικών αποτυπωμάτων είναι η παράμετρος «αδυναμία εγγραφής»-Failure to Enroll (FTE). Ο λόγος FTE δίνει απλώς το ποσοστό των φορών που ο αισθητήρας αποτυγχάνει να διαβάσει επαρκώς το βιομετρικό αναγνωριστικό για τη συνεχή επεξεργασία και εγγραφή του χρήστη. Οι αστοχίες μπορεί π.χ. να προκαλούνται από βρεγμένο ή κατεστραμμένο δέρμα κατά τη σάρωση δακτυλικών αποτυπωμάτων. Μια άλλη μέτρηση που χρησιμοποιείται συχνά είναι η «κουκκίδα ανά ίντσα Dot Per Inch (DPI) που καθορίζει την ανάλυση του αισθητήρα. Με χαμηλή ανάλυση (χαμηλή τιμή dpi) δεν μπορούν να καταγραφούν λεπτές λεπτομέρειες, μειώνοντας έτσι την ποιότητα της εικόνας (Security Industry Association, 2019).

Επί του παρόντος, πολύ υψηλή ποιότητα εικόνας μπορεί να αποκτηθεί με αισθητήρες υπερήχων και ενεργούς χωρητικούς αισθητήρες που διαβάζουν το στρώμα του δέρματος που είναι πολύ πιο ευδιάκριτο και λιγότερο επιρρεπές σε παραμόρφωση από το εξωτερικό επιδερμικό στρώμα (Security Industry Association, 2019).

4.7.2 Ταχύτητα

Η ταχύτητα με την οποία λειτουργεί ένα σύστημα δακτυλικών αποτυπωμάτων έχει σημαντικό αντίκτυπο στο πόσο βολικό είναι στη χρήση του. Η μετατροπή αναλογικού σε ψηφιακό που πραγματοποιείται από το ηλεκτρονικό κύκλωμα στο ολοκληρωμένο κύκλωμα, καθώς και οι υπολογισμοί που εμπλέκονται στον έλεγχο ταυτότητας του δακτυλικού αποτυπώματος πρέπει να είναι αποτελεσματικοί και γρήγοροι. Η σύνταξη αλγορίθμων με μινιμαλιστικό τρόπο είναι σημαντική για την επίτευξη των επιθυμητών ιδιοτήτων. Χωρητικοί, θερμικοί αισθητήρες και αισθητήρες με βάση την πίεση μπορούν όλοι να λειτουργούν με πολύ υψηλή ταχύτητα. Επί του παρόντος, ο

χρόνος αφύπνισης και επαλήθευσης με τέτοιους αισθητήρες μπορεί να είναι κάτω από 500 ms (Security Industry Association, 2019).

4.7.3 Κατανάλωση ενέργειας

Η κατανάλωση ενέργειας ενός συστήματος αισθητήρων είναι ένας πολύ σημαντικός παράγοντας και κρίσιμος για εφαρμογές όπως κινητά τηλέφωνα, έξυπνες κάρτες και άλλα φορητά αντικείμενα. Η υψηλή κατανάλωση ενέργειας όχι μόνο αδειάζει την μπαταρία της συσκευής, αλλά δημιουργεί επίσης θερμότητα που μπορεί να βλάψει ή να ενοχλήσει άλλα στοιχεία της συσκευής (Security Industry Association, 2019).

Είναι σημαντικό να ληφθούν υπόψη οι απαιτήσεις ισχύος του αισθητήρα με την πάροδο του χρόνου, δηλαδή η πραγματική ποσότητα ενέργειας που διαχέεται από τον αισθητήρα σε μια δεδομένη εφαρμογή. Η κατανάλωση ενέργειας καθορίζεται τόσο από το υλικό όσο και από το λογισμικό του συστήματος αισθητήρων. Οι αισθητήρες CMOS έχουν γενικά χαμηλή κατανάλωση ενέργειας, καθώς καταναλώνουν σημαντική ισχύ μόνο όταν χρησιμοποιούνται για τη σάρωση ενός δακτύλου. Επιπλέον, το μέγεθος του αισθητήρα επηρεάζει επίσης την κατανάλωση ενέργειας. ένας μικρότερος αισθητήρας καταναλώνει λιγότερη ενέργεια. Οι χωρητικοί αισθητήρες έχουν αυτήν τη στιγμή τη χαμηλότερη κατανάλωση ενέργειας από τις τεχνολογίες αισθητήρων που διατίθενται στο εμπόριο στην αγορά. Οι οπτικοί, οι υπερήχοι και οι θερμικοί αισθητήρες απαιτούν πολύ περισσότερη ισχύ και επομένως είναι λιγότερο κατάλληλοι για κινητές εφαρμογές. Η τυπική κατανάλωση ενέργειας για έναν ενεργό χωρητικό αισθητήρα είναι 5 μA σε κατάσταση ηρεμίας και 20 mA σε χρήση (Security Industry Association, 2019).

4.7.4 Μέγεθος

Το μέγεθος είναι συχνά μια καθοριστική παράμετρος κατά την επιλογή ενός αισθητήρα δακτυλικών αποτυπωμάτων, ειδικά για κινητές εφαρμογές. Η χρήση ενός αισθητήρα μικρού μεγέθους επιτρέπει πρόσθετες επιλογές σχεδίασης κατά τη διαμόρφωση του επιθυμητού προϊόντος. Οι μικρότεροι αισθητήρες έχουν επίσης πλεονέκτημα κόστους έναντι των μεγαλύτερων αισθητήρων, απλώς και μόνο επειδή τα μικρότερα ολοκληρωμένα κυκλώματα χρησιμοποιούν λιγότερο πυρίτιο και μπορούν να γίνουν φθηνότεροι. Ωστόσο, υπάρχει μια αντιστάθμιση μεταξύ μεγέθους

και ποιότητας εικόνας στις εφαρμογές αναγνώρισης δακτυλικών αποτυπωμάτων. Η χρήση ενός πολύ μικρού αισθητήρα με πολύ χαμηλή ανάλυση θα έχει ως αποτέλεσμα κακή ποιότητα εικόνας, η οποία με τη σειρά της θα απαιτήσει πιο προηγμένους και ενεργοβόρους αλγόριθμους αντιστοίχισης, εναλλακτικά θα κάνει την ασφαλή εγγραφή και τον έλεγχο ταυτότητας δυσκίνητη ή ακόμα και αδύνατη. Οι ενεργοί χωρητικοί αισθητήρες συνήθως διατίθενται σε μεγέθη 84 x 84 mm έως 160 x 160 mm και μπορούν να είναι κυκλικοί, ορθογώνιοι ή διαμορφωμένοι σε οποιαδήποτε άλλη δισδιάστατη μορφή που απαιτείται από τον σχεδιαστή του προϊόντος.

4.7.5 Κόστος

Το κόστος είναι ένας σημαντικός παράγοντας κατά την επιλογή συστημάτων αισθητήρων. Το κόστος ενός ενεργού χωρητικού αισθητήρα δακτυλικών αποτυπωμάτων που βασίζεται σε πυρίτιο, συσχετίζεται σε μεγάλο βαθμό με το μέγεθος του αισθητήρα, καθώς το υλικό είναι ο κύριος παράγοντας κόστους. Το κόστος επηρεάζεται επίσης από τη διαδικασία παραγωγής, την ολοκλήρωση του συστήματος και την τεχνολογία που χρησιμοποιείται.

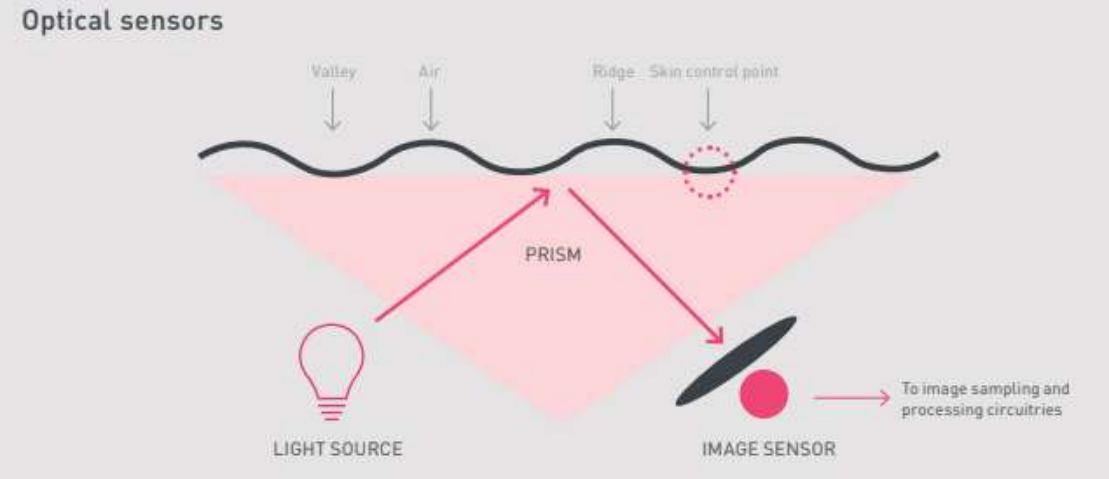
4.8 Τύποι σαρωτών δακτυλικών αποτυπωμάτων

Αυτοί περιλαμβάνουν (Leonard & Ezeonyi, 2020; Security Industry Association, 2019):

4.8.1 Οπτικός αισθητήρας

Οι οπτικοί αισθητήρες καταγράφουν μοτίβα δακτυλικών αποτυπωμάτων συλλαμβάνοντας ορατό φως και μετατρέποντάς το σε ηλεκτρικά σήματα που χρησιμοποιούνται για τη δημιουργία της εικόνας δακτυλικών αποτυπωμάτων. Οι αισθητήρες διαθέτουν συστοιχίες φωτοδιόδων ή ανιχνευτών φωτοτρανζίστορ που μετατρέπουν την ενέργεια του φωτός που χτυπά τον ανιχνευτή σε ηλεκτρικό φορτίο. Τα περισσότερα πακέτα οπτικών αισθητήρων περιλαμβάνουν επίσης ένα LED (Light Emitting Diode) ή μια συστοιχία LED, για να φωτίζει το άκρο του δακτύλου έτσι ώστε ο ανιχνευτής να μπορεί να καταγράψει την εικόνα δακτυλικών αποτυπωμάτων από το φως που ανακλάται στο δάχτυλο. Στη συνέχεια χρησιμοποιείται ένα πρίσμα με

προστατευτική επίστρωση για να αντανακλά το φως προς τον ανιχνευτή (Security Industry Association, 2019).

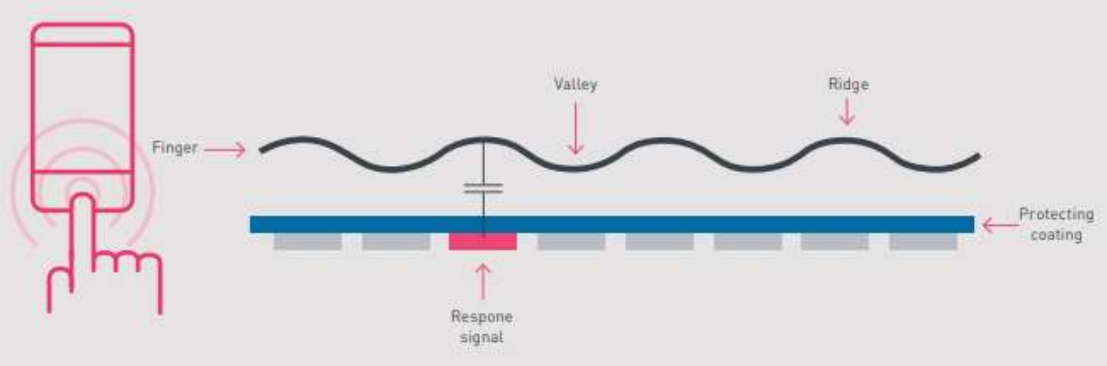


Εικόνα 7: Αρχή λειτουργίας ενός οπτικού αισθητήρα δακτυλικών αποτυπωμάτων, πηγή: (Security Industry Association, 2019)

Οι ανιχνευτές που χρησιμοποιούνται σήμερα στους οπτικούς αισθητήρες δακτυλικών αποτυπωμάτων είναι είτε CCD (Charge Coupled Devices) είτε CMOS (Complementary Metal Oxide Semiconductor) οπτικές συσκευές απεικόνισης. Οι ανιχνευτές CCD και CMOS είναι του ίδιου τύπου που μπορούν να βρεθούν στις ψηφιακές φωτογραφικές μηχανές. Οι ανιχνευτές CCD είναι ιδιαίτερα ευαίσθητοι σε χαμηλά επίπεδα φωτός και επομένως είναι καλοί για τη σύλληψη των κλίμακας του γκρι. Ιστορικά, οι ανιχνευτές CCD ήταν πολύ καλύτεροι από τους ανιχνευτές CMOS, αλλά καθώς η τεχνολογία CMOS έχει αναπτυχθεί πολύ τα τελευταία δέκα περίπου χρόνια, οι δυνατότητες της τεχνολογίας CMOS έχουν φτάσει στο ίδιο επίπεδο με το CCD. Η κατασκευή CCD είναι μάλλον δαπανηρή σε σύγκριση με την κατασκευή CMOS. Εκτός από το κόστος, οι οπτικές συσκευές απεικόνισης CMOS έχουν επίσης ένα σαφές πλεονέκτημα, καθώς μπορούν να κατασκευαστούν για να κρατούν μέρος της λογικής για την επεξεργασία εικόνας στο ίδιο τσιπ πυριτίου με τον ανιχνευτή. Αυτό οδήγησε στο ότι οι περισσότεροι οπτικοί αισθητήρες για ηλεκτρονικά είδη ευρείας κατανάλωσης, όπου το κόστος καθώς και η κατανάλωση ενέργειας είναι σημαντικά χαρακτηριστικά, χρησιμοποιούν ανιχνευτές CMOS. Η οπτική σύλληψη ήταν η πρώτη τεχνολογία ηλεκτρονικού αισθητήρα δακτυλικών αποτυπωμάτων που χρησιμοποιήθηκε και είναι πιθανώς η τεχνολογία που χρησιμοποιείται στον μεγαλύτερο αριθμό εφαρμογών (Security Industry Association, 2019).

4.8.2 Χωρητικός αισθητήρας

Αυτός είναι ένας τύπος σαρωτή που χρησιμοποιεί ηλεκτρισμό για να προσδιορίσει τα μοτίβα των δακτυλικών αποτυπωμάτων. Όταν ένα δάχτυλο ακουμπάει στην χωρητική επιφάνεια αφής, η συσκευή μετρά τη φόρτιση. Η χωρητικότητα είναι η ικανότητα μιας φυσικής οντότητας να συγκρατεί ηλεκτρικό φορτίο. Ένας χωρητικός αισθητήρας δακτυλικών αποτυπωμάτων δημιουργεί την εικόνα δακτυλικών αποτυπωμάτων χρησιμοποιώντας μια διάταξη που περιέχει πολλές χιλιάδες μικρές πλάκες πυκνωτών. Όταν το δάχτυλο τοποθετείται στον αισθητήρα, δημιουργούνται αμυδρά ηλεκτρικά φορτία, δημιουργώντας ένα μοτίβο ανάμεσα στις κορυφογραμμές ή τις κοιλάδες του δακτύλου και τις πλάκες του αισθητήρα. Χρησιμοποιώντας αυτά τα φορτία, ο αισθητήρας μετρά το μοτίβο χωρητικότητας σε όλη την ανιχνευόμενη επιφάνεια. Οι μετρούμενες τιμές ψηφιοποιούνται από τη λογική του αισθητήρα και στη συνέχεια αποστέλλονται στο μικροεπεξεργαστή για ανάλυση (Leonard & Ezeonyi, 2020).

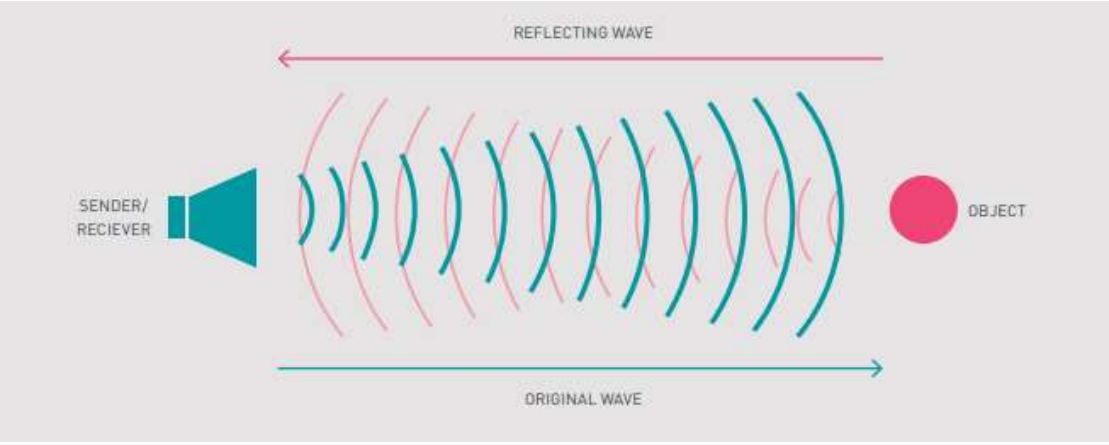


Εικόνα 8: Αρχή χωρητικής ανίχνευσης, πηγή: (Security Industry Association, 2019)

Αντί να φωτογραφίζει μια εικόνα των κορυφογραμμών και κοιλάδων σε ένα δακτυλικό αποτύπωμα όπως κάνει ένας οπτικός σαρωτής, ο αισθητήρας του χωρητικού σαρωτή δημιουργεί ένα περίπλοκο μοτίβο ηλεκτρικών σημάτων, το οποίο επεξεργάζεται για να σχηματίσει μια ψηφιακή εικόνα του δακτυλικού αποτυπώματος. Επειδή ο χωρητικός σαρωτής απαιτεί τη φυσική παρουσία ενός ανθρώπινου δακτύλου για τη δημιουργία της εικόνας, είναι επομένως πολύ πιο δύσκολο να εξαπατηθεί από μια οπτική συσκευή. Ένα άλλο πλεονέκτημα των συσκευών ανάγνωσης δακτυλικών αποτυπωμάτων με χωρητική ανίχνευση είναι ότι είναι πιο συμπαγείς και επομένως εύκολο να ενσωματωθούν σε φορητές συσκευές (Security Industry Association, 2019).

4.8.3 Αισθητήρας υπερήχων

Αυτός είναι ένας τύπος σαρωτή που χρησιμοποιεί ηχολογικό εντοπισμό για την εύρεση και την αναγνώριση αντικειμένων. Σε αντίθεση με την οπτική απεικόνιση, οι αισθητήρες υπερήχων χρησιμοποιούν ηχητικά κύματα πολύ υψηλής συχνότητας για να διεισδύσουν στο επιδερμικό στρώμα του δέρματος. Τα ηχητικά κύματα παράγονται με χρήση πιεζοηλεκτρικών μετατροπέων και η ανακλώμενη ενέργεια μετράται επίσης με χρήση πιεζοηλεκτρικών υλικών. Δεδομένου ότι το στρώμα του δέρματος παρουσιάζει το ίδιο χαρακτηριστικό σχέδιο του δακτυλικού αποτυπώματος, οι μετρήσεις ανακλώμενου κύματος μπορούν να χρησιμοποιηθούν για να σχηματίσουν μια εικόνα του δακτυλικού αποτυπώματος. Η χρήση του δερματικού στρώματος του δέρματος εξαλείφει την ανάγκη για καθαρό, άθικτο επιδερμικό δέρμα και μια καθαρή επιφάνεια αίσθησης. Αυτό καθιστά τους αισθητήρες υπερήχων καλούς στο να διαβάζουν τα βρεγμένα και κατεστραμμένα δάχτυλα. Ωστόσο, τα ξηρά δάχτυλα μπορεί συχνά να είναι πρόβλημα (πχ το απαραίτητο τζελ που βάζουν οι γιατροί στις κοιλιές πριν κάνουν ένα υπέρηχο για να κοιτάξουν τα μωρά). Οι αισθητήρες δακτυλικών αποτυπωμάτων υπερήχων έχουν το πλεονέκτημα ότι παρέχουν περισσότερες βιομετρικές πληροφορίες από τους περισσότερους άλλους αισθητήρες δακτυλικών αποτυπωμάτων. Τα προβλήματα με την τεχνολογία ήταν ότι είναι αργή, ακριβή, απαιτεί ενέργεια, είναι ογκώδης (μεγάλοι αισθητήρες) και απαιτεί μεγάλη επεξεργαστική ισχύ καθώς οι αλγόριθμοι είναι έντονοι για δεδομένα.



Εικόνα 9: Αρχή της ανίχνευσης δακτυλικών αποτυπωμάτων με υπερήχους, πηγή: (Security Industry Association, 2019)

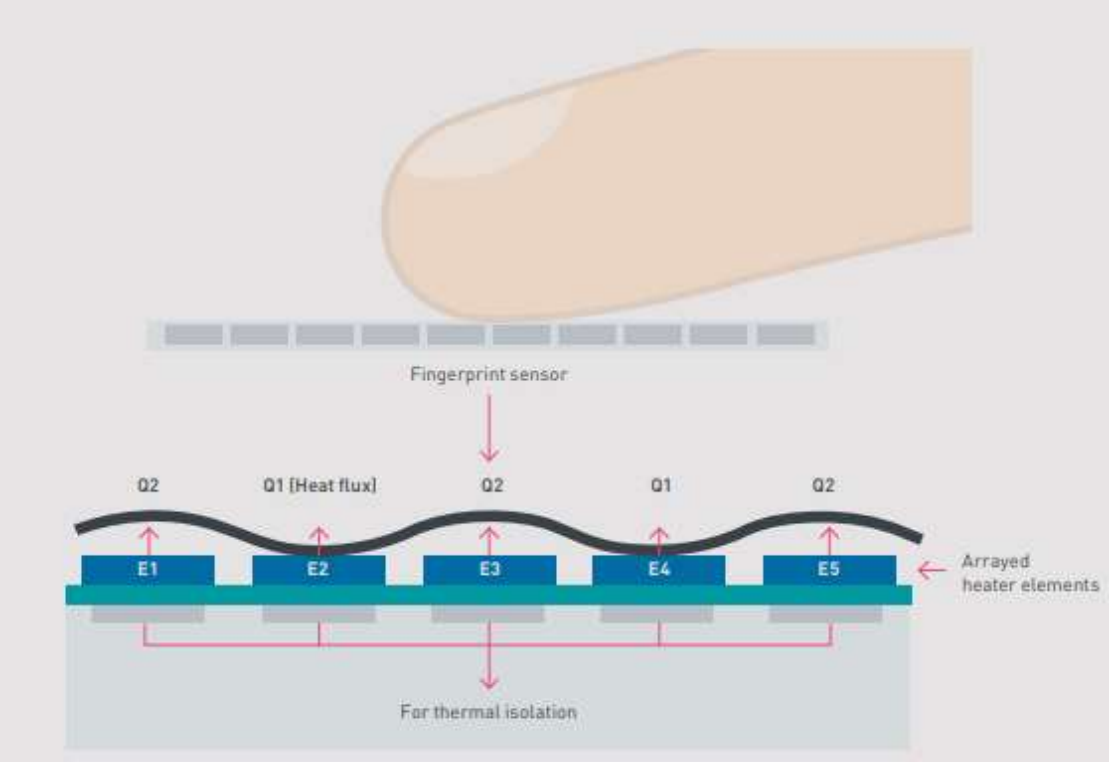
4.8.4 Θερμικός αισθητήρας

Οι θερμικοί αισθητήρες δακτυλικών αποτυπωμάτων δημιουργούν εικόνες δακτυλικών αποτυπωμάτων χρησιμοποιώντας μετρήσεις θερμοκρασίας. Οι αισθητήρες έχουν συστοιχίες πλακών σε πυροηλεκτρικό υλικό του ίδιου είδους που χρησιμοποιείται σε κάμερες υπερύθρων. Όταν το δάχτυλο αγγίζει τον αισθητήρα, οι ράχες του δακτύλου έρχονται σε επαφή με την επιφάνεια του αισθητήρα και μετράται η θερμοκρασία. Στη συνέχεια δημιουργείται η εικόνα δακτυλικών αποτυπωμάτων με βάση τη μέτρηση της θερμοκρασίας δέρματος από τις κορυφογραμμές και τη θερμοκρασία περιβάλλοντος στις κοιλάδες.

Υπάρχουν ορισμένα σημαντικά προβλήματα με τους θερμικούς αισθητήρες δακτυλικών αποτυπωμάτων (Leonard & Ezeonyi, 2020):

- Η αλλαγή θερμοκρασίας είναι δυναμική, επομένως, η εικόνα του δακτυλικού αποτυπώματος είναι παροδική και διαγράφεται μετά από περίπου ένα δέκατο του δευτερολέπτου όταν η επιφάνεια του αισθητήρα φτάσει στην ίδια θερμοκρασία με το δάχτυλο.
- Ευαισθησία στη φθορά καθώς και στη μόλυνση
- Όταν η θερμοκρασία περιβάλλοντος είναι κοντά στη θερμοκρασία της επιφάνειας του δακτύλου, ο αισθητήρας απαιτεί θέρμανση έτσι ώστε να υπάρχει διαφορά θερμοκρασίας τουλάχιστον ενός βαθμού Κελσίου – διαφορετικά η διαφορά θερμοκρασίας δεν μπορεί να μετρηθεί σωστά και δεν μπορεί να δημιουργηθεί εικόνα δακτυλικού αποτυπώματος.

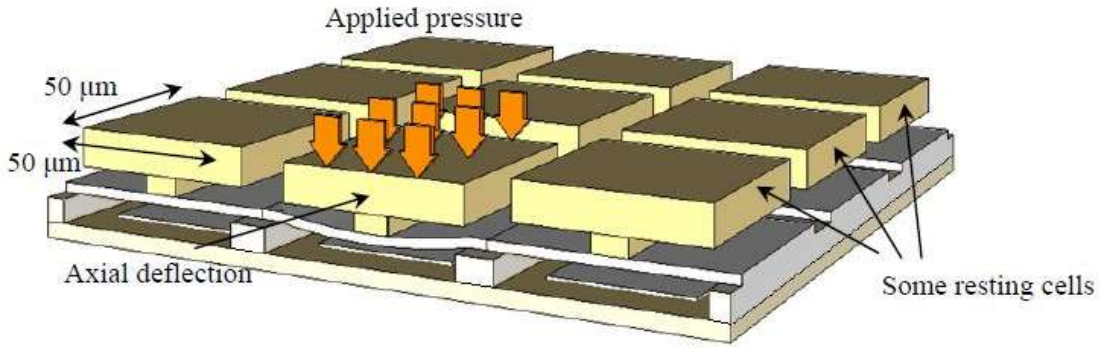
Μερικά από τα παραπάνω προβλήματα μπορούν να αντιμετωπιστούν από έναν ενεργό θερμικό αισθητήρα. Ένας ενεργός θερμικός αισθητήρας στέλνει έναν παλμό θερμότητας χαμηλής ισχύος σε κάθε εικονοστοιχείο αισθητήρα όταν το δάχτυλο τοποθετείται σταθερά στην επιφάνεια του αισθητήρα. Ο παλμός θερμότητας σπάει τη θερμική ισορροπία και έτσι επιτρέπει τη στατική λήψη της εικόνας δακτυλικών αποτυπωμάτων (Security Industry Association, 2019).



Εικόνα 10: Αρχή της ενεργού θερμικής ανίχνευσης δακτυλικών αποτυπωμάτων, πηγή: (Security Industry Association, 2019)

4.8.5 Αισθητήρας πίεσης

Μια αναδυόμενη κατηγορία αισθητήρων δακτυλικών αποτυπωμάτων βασίζεται σε υλικά λεπτής μεμβράνης ικανά να παράγουν ηλεκτρικό σήμα όταν τους ασκείται μηχανική καταπόνηση. Η επιφάνεια του αισθητήρα υλοποιείται ως ένα πολύ λεπτό και εύκαμπτο, μη αγώγιμο διηλεκτρικό υλικό. Όταν ένα δάχτυλο τοποθετείται στον αισθητήρα, οι κορυφογραμμές και οι κοιλάδες εφαρμόζουν διαφορετικά επίπεδα πίεσης στην επιφάνεια, με αποτέλεσμα ποικίλη ποσότητα ρεύματος που μπορεί να μετρηθεί και να χρησιμοποιηθεί για τη δημιουργία μιας εικόνας δακτυλικών αποτυπωμάτων (Security Industry Association, 2019).



Εικόνα 11: Αρχή του αισθητήρα πίεσης, πηγή: (Mainguet, 2023)

Οι αισθητήρες πίεσης μπορούν να γίνουν μικροί και είναι μια από τις λίγες κατηγορίες αισθητήρων εκτός από τους χωρητικούς αισθητήρες που μπορούν να ενσωματωθούν σε κινητές συσκευές όπως τηλέφωνα και tablet. Ωστόσο, οι υπάρχοντες αισθητήρες είναι ευαίσθητοι στη θερμοκρασία και λιγότερο κατάλληλοι για χρήση όπου οι περιβαλλοντικές συνθήκες είναι σκληρές ή μεταβάλλονται γρήγορα (Security Industry Association, 2019).

4.9 Τύποι μικροϋπολογιστών με ενιαία πλακέτα

4.9.1 Raspberry Pi

Ο Raspberry Pi είναι ένας πολύ ισχυρός μικροϋπολογιστής με ενιαία πλακέτα που είναι ιδανικός για πολλές εφαρμογές, συμπεριλαμβανομένων των έργων αυτοματισμού και των εφαρμογών Internet of Things (IoT). Διαθέτει μεγάλη επεξεργαστική δύναμη και είναι σε θέση να υποστηρίξει λειτουργικά συστήματα πλήρους λειτουργίας, όπως το Linux. Έχει επίσης μια πολύ δραστήρια κοινότητα που μπορεί να προσφέρει υποστήριξη και πόρους. Το Raspberry Pi έχει πλήρη πρόσβαση σε περιφερειακά GPIO, το οποίο είναι ιδανικό για έργα που απαιτούν άμεση αλληλεπίδραση με υλικό, όπως η ανάγνωση από αισθητήρες δακτυλικών αποτυπωμάτων (RaspberryTips, 2023).

4.9.2 Arduino

Το Arduino είναι μια ανοιχτού κώδικα πλατφόρμα που βασίζεται σε εύκολη χρήση υλικού και λογισμικού. Τα Arduino boards είναι ιδιαίτερα κατάλληλα για τον έλεγχο απλών ψηφιακών συσκευών, όπως αισθητήρες και διακόπτες, αλλά δεν έχουν την

επεξεργαστική δύναμη ή τη δυνατότητα πολυεργασίας του Raspberry Pi (Arduino, 2023).

4.9.3 BeagleBone Black

Το BeagleBone Black είναι ένας μικροϋπολογιστής με ενιαία πλακέτα που προσφέρει όμοια δυνατότητα όπως ο Raspberry Pi, αλλά με περισσότερες επιλογές σύνδεσης I/O και αναλογικής εισόδου. Ωστόσο, δεν έχει τόσο μεγάλη κοινότητα υποστήριξης όπως ο Raspberry Pi (beagleboard.org, 2023).

4.9.4 Odroid-XU4

Το Odroid-XU4 παρέχει μεγαλύτερη επεξεργαστική δύναμη και δυνατότητες πολυεργασίας σε σχέση με το Raspberry Pi, κάνοντάς το κατάλληλο για πιο απαιτητικές εφαρμογές. Ωστόσο, είναι πιο ακριβό και η υποστήριξη της κοινότητας και η διαθεσιμότητα των πόρων δεν είναι τόσο εκτεταμένη όσο για το Raspberry Pi (Hardkernel co., 2019).

Κεφάλαιο 5ο

Επεξήγηση κώδικα εφαρμογής

Σε αυτό το κεφάλαιο ακολουθεί η επεξήγηση του τρόπου με τον οποίο οι τεχνολογίες που χρησιμοποιούνται στην εφαρμογή αλληλεπιδρούν ώστε να παράξουν το τελικό σύστημα. Ο τρόπος με τον οποίο θα γίνεται η περιγραφή είναι παραθέτοντας αποσπάσματα κώδικα και έπειτα ακλουθεί η εξήγηση τους. Η ροή την οποία θα ακολουθούν οι επεξηγήσεις θα είναι όσον το δυνατό πιο κοντά στην πραγματική ροή με την οποία τρέχει το πρόγραμμα.

5.1 Αρχικοποίηση συνδέσεων μεταξύ συστημάτων

```
@dataclass
class Connection:
    username: str = os.environ["postgres_username"]
    password: str = os.environ["postgres_password"]
    host: str = "localhost"
    database_name: str = "fingerprint_system"
    port_number: str = "5432"
    connection_string: str = None
    engine: Engine = None

    def __post_init__(self):
        self.connection_string = f"postgresql://{self.username}:{self.password}@{self.host}:{self.port_number}/{self.database_name}"
        self.engine = create_engine(self.connection_string)


connection = Connection()

app = Flask(__name__, template_folder="templates")
app.config["SQLALCHEMY_DATABASE_URI"] = connection.connection_string
app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False
app.secret_key = 'secret'
db = SQLAlchemy(app)
```

Εικόνα 12:Αρχικοποίηση συνδέσεων και ρυθμίσεις κλάσεων βιβλιοθηκών

Πριν ξεκινήσει να τρέχει η εφαρμογή, πρέπει να ορίσουμε τον τρόπο με τον οποίο το σύστημα μας θα επικοινωνεί με τη βάση δεδομένων μας η οποία είναι η PostrgeSQL. Για αυτό ορίζουμε μία νέα κλάση η οποία θα περιέχει τα στοιχεία που είναι αναγκαία για τη σύνδεση αυτή. Χρησιμοποιώντας τον διακοσμητή @dataclass, η Python αυτοματοποιεί τη δημιουργία κάποιων βασικών μεθόδων για την κλάση. Τα χαρακτηριστικά της κλάσης περιλαμβάνουν το όνομα χρήστη, τον κωδικό πρόσβασης, τον υπολογιστή-οικοδεσπότη, το όνομα της βάσης δεδομένων και τον αριθμό θύρας. Αυτές οι πληροφορίες προέρχονται είτε από τις μεταβλητές περιβάλλοντος του συστήματος είτε από προκαθορισμένες τιμές. Η μέθοδος __post_init__ καλείται αυτόματα μετά τη δημιουργία ενός αντικειμένου της κλάσης και υπολογίζει τη συμβολοσειρά σύνδεσης (connection string) βάσει των παραπάνω

χαρακτηριστικών. Επίσης, δημιουργεί ένα "engine" για τη βάση δεδομένων, το οποίο μπορεί να χρησιμοποιηθεί για να πραγματοποιηθούν ενέργειες στη βάση δεδομένων.

Ύστερα, δημιουργούμε συνδέσεις μεταξύ του backend μας “Flask”, του ORM συστήματος SQLAlchemy που επιτρέπει την επικοινωνία μεταξύ της βάσης μας και του προγράμματος μας γραμμένο σε Python. Η δυνατότητα που έχουν αυτές οι βιβλιοθήκες να υποστηρίξουν άμεσα η μία την άλλη μετά από απλές ρυθμίσεις, τις κάνει ιδανικές για να υποστηρίξουν μία ολοκληρωμένη εφαρμογή σαν τη δική μας. Τα πιο βασικά σημεία του παραπάνω κώδικα είναι τα εξής:

1. app = Flask(__name__, template_folder="templates"): Δημιουργείται μια νέα εφαρμογή Flask με το όνομα app. Εδώ, ορίζεται επίσης ένας φάκελος με το όνομα "templates" που περιέχει τα HTML πρότυπα (templates) που θα χρησιμοποιηθούν από την εφαρμογή.
2. app.config["SQLALCHEMY_DATABASE_URI"] = connection.connection_string: Ρυθμίζεται η συμβολοσειρά σύνδεσης για τη βάση δεδομένων με τη βοήθεια του SQLALCHEMY, ένας ORM (Object-Relational Mapper) για Python. Η συμβολοσειρά σύνδεσης λαμβάνεται από το αντικείμενο connection που προηγουμένως έχει οριστεί.
3. app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False: Αυτή η ρύθμιση απενεργοποιεί την παρακολούθηση τροποποιήσεων για τα αντικείμενα SQLAlchemy για λόγους απόδοσης.
4. app.secret_key = 'secret': Ορίζεται ένα μυστικό κλειδί για την εφαρμογή Flask, το οποίο είναι απαραίτητο για διάφορες λειτουργίες, όπως η διαχείριση συνεδριών.
5. db = SQLAlchemy(app): Δημιουργείται ένα νέο αντικείμενο db που είναι μια ενσωμάτωση του SQLAlchemy με την εφαρμογή Flask. Με αυτό τον τρόπο, μπορούμε να διαχειριστούμε τη βάση δεδομένων μέσω της εφαρμογής Flask.

5.2 Δήλωση των πινάκων

Η SQLAlchemy μας δίνει τη δυνατότητα να περιγράψουμε τους πίνακες που θα αρχικοποιηθούν στην βάση μας σαν κλάσεις Python. Θα το εκμεταλλευτούμε αυτό καθώς μας δίνει απόλυτο έλεγχο στον ορισμό και τη σύνδεση αυτών των πινάκων στα πλαίσια της εφαρμογής μας. Θα αρχίσουμε την αναφορά στις κλάσεις αυτές μία προς μία, καθώς και από τι αποτελούνται.

```
class ClassStudent(db.Model):
    __tablename__ = "class_student"

    id = db.Column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    attendance = db.Column(JSONB)
    class_id = db.Column(UUID(as_uuid=True), db.ForeignKey("class.id"))
    student_id = db.Column(UUID(as_uuid=True), db.ForeignKey("student.id"))
```

Εικόνα 13: Δήλωση της δομής της κλάσης της συσχέτισης φοιτητών και τάξεων

1. __tablename__ = "class_student": Καθορίζει το όνομα του πίνακα στη βάση δεδομένων ως "class_student".
2. id = db.Column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4): Δημιουργεί μια στήλη με το όνομα "id" που είναι τύπου UUID, και είναι το πρωτεύον κλειδί του πίνακα. Η προεπιλεγμένη τιμή για αυτή τη στήλη δημιουργείται από τη συνάρτηση uuid.uuid4.
3. attendance = db.Column(JSONB): Δημιουργεί μια στήλη με το όνομα "attendance" που μπορεί να αποθηκεύσει δεδομένα τύπου JSONB.
4. class_id = db.Column(UUID(as_uuid=True), db.ForeignKey("class.id")): Δημιουργεί μια στήλη με το όνομα "class_id" τύπου UUID. Αυτή η στήλη είναι ένα ξένο κλειδί που συνδέεται με το πεδίο "id" του πίνακα "class".
5. student_id = db.Column(UUID(as_uuid=True), db.ForeignKey("student.id")): Όπως και με το "class_id", δημιουργεί μια στήλη με το όνομα "student_id" τύπου UUID. Αυτή η στήλη είναι επίσης ένα ξένο κλειδί που συνδέεται με το πεδίο "id" του πίνακα "student".

```
class TeacherStudent(db.Model):
    __tablename__ = "teacher_student"

    id = db.Column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    teacher_id = db.Column(UUID(as_uuid=True), db.ForeignKey("teacher.id"))
    student_id = db.Column(UUID(as_uuid=True), db.ForeignKey("student.id"))
```

Εικόνα 14: Δήλωση της δομής της κλάσης της συσχέτισης των καθηγητών και μαθητών

1. __tablename__ = "teacher_student": Ορίζει το όνομα του πίνακα στη βάση δεδομένων ως "teacher_student".
2. id = db.Column(...): Δημιουργεί μια στήλη με το όνομα "id" τύπου UUID, η οποία είναι το πρωτεύον κλειδί του πίνακα.
3. teacher_id = db.Column(...): Δημιουργεί μια στήλη με το όνομα "teacher_id" τύπου UUID. Αυτή η στήλη είναι ένα ξένο κλειδί που παραπέμπει στο πρωτεύον κλειδί του πίνακα "teacher".
4. student_id = db.Column(...): Δημιουργεί μια στήλη με το όνομα "student_id" τύπου UUID. Όπως και το "teacher_id", αυτή η στήλη είναι ένα ξένο κλειδί που παραπέμπει στο πρωτεύον κλειδί του πίνακα "student".

```
class Class(db.Model):
    __tablename__ = "class"

    id = db.Column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    name = db.Column(db.String(100), nullable=True)
    faculty = db.Column(db.String(100), nullable=True)
    subject = db.Column(db.String(100), nullable=True)
    room = db.Column(db.String(100), nullable=True)
    is_lab = db.Column(db.Boolean, nullable=True)
    period = db.Column(db.String(100), nullable=True)
    commenced = db.Column(JSONB)
    teacher_id = db.Column(UUID(as_uuid=True), db.ForeignKey("teacher.id"))
    student_id = db.relationship("ClassStudent", backref="class")
```

Εικόνα 15: Δήλωση της δομής της κλάσης της τάξης

1. __tablename__ = "class": Καθορίζει το όνομα του πίνακα στη βάση δεδομένων ως "class".

2. id = db.Column(...): Δημιουργεί μια στήλη "id" που είναι τύπου UUID και είναι το πρωτεύον κλειδί του πίνακα.
3. name, faculty, subject, room, is_lab, period: Δημιουργεί στήλες με τα αντίστοιχα ονόματα για να αποθηκεύσει πληροφορίες σχετικά με την τάξη, τη σχολή, το μάθημα, την αίθουσα, αν είναι εργαστήριο ή όχι, και την περίοδο διδασκαλίας.
4. commenced = db.Column(JSONB): Δημιουργεί μια στήλη για να αποθηκεύσει δεδομένα σε μορφή JSONB, που μπορεί να περιέχει πληροφορίες σχετικά με το πότε ξεκίνησε η τάξη.
5. teacher_id = db.Column(...): Δημιουργεί μια στήλη που είναι ξένο κλειδί και συνδέεται με το πεδίο "id" του πίνακα "teacher".
6. student_id = db.relationship(...): Αντί να είναι μια απλή στήλη, αυτή είναι μια σχέση μεταξύ των πινάκων "class" και "class_student". Παρέχει πρόσβαση σε όλους τους σπουδαστές που είναι συνδεδεμένοι με μια συγκεκριμένη τάξη. Το backref="class" επιτρέπει την αντίστροφη πρόσβαση από τον πίνακα "class_student" πίσω στον πίνακα "class".

```
class Student(db.Model):
    __tablename__ = "student"

    id = db.Column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    am = db.Column(db.String(12), nullable=True)
    firstname = db.Column(db.String(100), nullable=True)
    lastname = db.Column(db.String(100), nullable=True)
    email = db.Column(db.String(100), nullable=True)
    semester = db.Column(db.String(100), nullable=True)
    fingerprint_id = db.Column(db.String(100), nullable=True)
    teacher_id = db.relationship("TeacherStudent", backref="student")
    class_id = db.relationship("ClassStudent", backref="student")
```

Εικόνα 16: Δήλωση της δομής της κλάσης των μαθητών

1. __tablename__ = "student": Καθορίζει το όνομα του πίνακα στη βάση δεδομένων ως "student".
2. id = db.Column(...): Δημιουργεί μια στήλη με το όνομα "id" τύπου UUID, που είναι το πρωτεύων κλειδί του πίνακα.
3. am, firstname, lastname, email, semester, fingerprint_id: Δημιουργούνται στήλες με τα αντίστοιχα ονόματα για να αποθηκεύσουν πληροφορίες σχετικά

με τον μαθητή, όπως τον αριθμό μητρώου, το όνομα, το επώνυμο, το email, το εξάμηνο και το αναγνωριστικό δακτυλικού αποτυπώματος.

4. teacher_id = db.relationship(...): Δημιουργεί μια σχέση μεταξύ των πινάκων "student" και "TeacherStudent". Αυτή η σχέση παρέχει πρόσβαση σε όλους τους δασκάλους που είναι συνδεδεμένοι με έναν συγκεκριμένο μαθητή. Το backref="student" επιτρέπει την αντίστροφη πρόσβαση από τον πίνακα "TeacherStudent" πίσω στον πίνακα "student".
5. class_id = db.relationship(...): Όπως και με το teacher_id, δημιουργεί μια σχέση μεταξύ των πινάκων "student" και "ClassStudent". Αυτή η σχέση παρέχει πρόσβαση σε όλες τις τάξεις στις οποίες ένας συγκεκριμένος μαθητής είναι εγγεγραμμένος.

```
class Teacher(db.Model):
    __tablename__ = "teacher"

    id = db.Column(UUID(as_uuid=True), primary_key=True, default=uuid.uuid4)
    am = db.Column(db.String(12), nullable=True)
    firstname = db.Column(db.String(100), nullable=True)
    lastname = db.Column(db.String(100), nullable=True)
    email = db.Column(db.String(100), nullable=True)
    fingerprint_id = db.Column(db.String(4), nullable=True)
    student_id = db.relationship("TeacherStudent", backref="teacher")
```

Εικόνα 17: Δήλωση της δομής της κλάσης των δασκάλων

1. __tablename__ = "teacher": Καθορίζει το όνομα του πίνακα στη βάση δεδομένων ως "teacher".
2. id = db.Column(...): Δημιουργεί μια στήλη με το όνομα "id" τύπου UUID, που είναι το πρωτεύον κλειδί του πίνακα.
3. am, firstname, lastname, email, fingerprint_id: Δημιουργούνται στήλες για να αποθηκεύσουν πληροφορίες σχετικά με τον δάσκαλο, όπως τον αριθμό μητρώου, το όνομα, το επώνυμο, το email και το αναγνωριστικό δακτυλικού αποτυπώματος.
4. student_id = db.relationship(...): Δημιουργεί μια σχέση μεταξύ των πινάκων "teacher" και "TeacherStudent". Αυτή η σχέση παρέχει πρόσβαση σε όλους τους μαθητές που είναι συνδεδεμένοι με έναν συγκεκριμένο δάσκαλο. Το

backref="teacher" επιτρέπει την αντίστροφη πρόσβαση από τον πίνακα "TeacherStudent" πίσω στον πίνακα "teacher".

5.3 Αρχικοποίηση μέτρων ασφαλείας του συστήματος

Προτού ξεκινήσουμε την εφαρμογή πρέπει να είμαστε σίγουροι πως ο κάθε χρήστης εισέρχεται στο σύστημα χρησιμοποιώντας την αρχική σελίδα και δεν μπορεί να την παρακάμψει πληκτρολογώντας απευθείας την διεύθυνση μιας άλλης σελίδας. Όταν μπει επιτυχώς με το δακτυλικό του αποτύπωμα, τα στοιχεία του αποθηκεύονται σε μια συνεδρία ώστε να μπορεί να μπει στις υπόλοιπες σελίδες μέχρι να επιθυμήσει να βγει. Αυτό κάνει την εφαρμογή μας ασφαλή αφού ο μοναδικός τρόπος να αλληλεπιδράσει με το σύστημα μας είναι μέσω της αρχικής σελίδας μετά από ταυτοποίηση.

```
def login_required(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        if session.get('id') is None:
            return redirect('/', code=302)
        return f(*args, **kwargs)
    return decorated_function
```

Εικόνα 18: Δήλωση διακομιστή επιβεβαίωσης συνεδρίας χρήστη

1. @wraps(f): Είναι ένας διακομιστής που βοηθάει να διατηρηθεί η πληροφορία της αρχικής συνάρτησης ακόμα και αφού εφαρμοστεί ο διακομιστής.
2. decorated_function(*args, **kwargs): Ορίζει μια νέα συνάρτηση που θα αντικαταστήσει την αρχική συνάρτηση.
3. if session.get('id') is None: Ελέγχει αν υπάρχει μια τιμή για το κλειδί 'id' στη συνεδρία. Αν δεν υπάρχει, σημαίνει ότι ο χρήστης δεν είναι συνδεδεμένος.
4. return redirect('/', code=302): Εάν ο χρήστης δεν είναι συνδεδεμένος, επιστρέφει στην αρχική σελίδα με έναν κωδικό ανακατεύθυνσης 302.
5. return f(*args, **kwargs): Εάν ο χρήστης είναι συνδεδεμένος, η αρχική συνάρτηση εκτελείται ως συνήθως.

Συνοψίζοντας, ο διακομιστής login_required χρησιμοποιείται για να ελέγξει αν ένας χρήστης είναι συνδεδεμένος προτού επιτραπεί η πρόσβαση σε συγκεκριμένες συναρτήσεις ή διαδρομές της εφαρμογής

```
@app.route("/")
def index():
    session.clear()
    res = executeCommand(verifyUser)
    if res[0] == 0:
        teacher_data = Teacher.query.filter(Teacher.fingerprint_id == str(res[1])).first()
        if teacher_data:
            session["id"] = teacher_data.__dict__["id"]
            return redirect(f"dashboards/{res[1]}")
        else:
            return redirect(url_for("login_error"))
    elif res[0] == 8:
        return render_template("index.html"), {"Refresh": "1; url=/"}
    else:
        return redirect(url_for("login_error"))
```

Εικόνα 19:Λογική αρχικής σελίδας

Τώρα ορίζουμε την αρχική σελίδα που είναι η μοναδική που δεν ελέγχει την συνεδρία. Αυτό σημαίνει πως ο καθένας μπορεί να την ανοίξει. Όμως, όταν κάποιος μπαίνει σε αυτή τη σελίδα, η συνεδρία του καταστρέφεται και θα πρέπει να περάσει το μηχανισμό ταυτοποίησης για να μπορεί να ξαναχρησιμοποιήσει το σύστημα. Γενικά, κάθε σελίδα στην εφαρμογή έχει ένα κουμπί που οδηγεί σε αυτή τη σελίδα και επιτρέπει στο χρήστη να βγει από το σύστημα χωρίς να έχει το φόβο ότι κάποιος άλλος θα μπορεί μετά να μπει από τον υπολογιστή του χωρίς δακτυλικό αποτύπωμα.

1. @app.route("/"): Ορίζει μια διαδρομή προς τον αρχικό δικτυακό τόπο της εφαρμογής.
2. session.clear(): Καθαρίζει τις τρέχουσες πληροφορίες συνεδρίας του χρήστη.
3. res = executeCommand(verifyUser): Χρησιμοποιεί το πρόγραμμα του δακτυλικού αποτυπώματος και αποθηκεύει το αποτέλεσμα (επιτυχία ή απόρριψη) στη μεταβλητή res.

4. if res[0] == 0: Ελέγχει αν το πρώτο στοιχείο της απάντησης είναι 0, που πιθανώς υποδηλώνει επιτυχή επαλήθευση.
5. teacher_data = Teacher.query...: Προσπαθεί να βρει έναν δάσκαλο στη βάση δεδομένων με βάση το αναγνωριστικό δακτυλικού αποτυπώματος που επιστράφηκε από την executeCommand.
6. if teacher_data: Ελέγχει αν βρέθηκαν δεδομένα για τον δάσκαλο.
7. session["id"] = teacher_data.__dict__["id"]: Αποθηκεύει το αναγνωριστικό του δασκάλου στη συνεδρία.
8. return redirect(f"dashboards/{res[1]}"): Ανακατευθύνει τον χρήστη στον πίνακα ελέγχου του δασκάλου.
9. elif res[0] == 8: Ελέγχει αν το πρώτο στοιχείο της απάντησης είναι 8, ίσως υποδηλώνοντας μια συγκεκριμένη κατάσταση.
10. return render_template("index.html"), {"Refresh": "1; url=/"}:: Επιστρέφει τον αρχικό δικτυακό τόπο και προγραμματίζει μια ανακατεύθυνση για ανανέωση μετά από 1 δευτερόλεπτο.
11. else: Σε οποιαδήποτε άλλη περίπτωση ανακατευθύνει τον χρήστη σε μια σελίδα με μήνυμα λάθους σύνδεσης.

```
@app.route("/login_error")
def login_error():
    res = executeCommand(verifyUser)
    if res[0] == 0:
        teacher_data = Teacher.query.filter(Teacher.fingerprint_id == str(res[1])).first()
        if teacher_data:
            session["id"] = teacher_data.__dict__["id"]
            return redirect(f"dashboards/{res[1]}")
        else:
            return render_template("login_error.html"), {"Refresh": "1; url=/login_error"}
    else:
        return render_template("login_error.html"), {"Refresh": "1; url=/login_error"}
```

Εικόνα 20:Λογική στη σελίδα σφαλμάτων κατά την ταυτοποίηση

Εδώ, αυτό το κομμάτι κώδικα περιγράφει πως αντιμετωπίζουμε κάποιο λάθος κατά την αναγνώριση

1. @app.route("/login_error"): Ορίζει μια διαδρομή προς τη σελίδα "login_error" της εφαρμογής.

2. res = executeCommand(verifyUser): Εκτελεί μια εντολή ή συνάρτηση για την επαλήθευση του χρήστη και αποθηκεύει το αποτέλεσμα στη μεταβλητή res.
3. if res[0] == 0: Ελέγχει αν το πρώτο στοιχείο της απάντησης είναι 0, υποδεικνύοντας πιθανά επιτυχή επαλήθευση.
4. teacher_data = Teacher.query...: Προσπαθεί να βρει έναν δάσκαλο στη βάση δεδομένων με βάση το αναγνωριστικό δακτυλικού αποτυπώματος που επιστράφηκε από την executeCommand.
5. if teacher_data: Ελέγχει αν βρέθηκε κάποια εγγραφή για τον δάσκαλο.
6. session["id"] = teacher_data.__dict__["id"]: Αποθηκεύει το αναγνωριστικό του δασκάλου στη συνεδρία.
7. return redirect(f"dashboards/{res[1]}"): Ανακατευθύνει τον χρήστη στον πίνακα ελέγχου του δασκάλου.
8. else και else στο τέλος της συνάρτησης: Σε περίπτωση που δεν βρέθηκε κάποιος δάσκαλος ή για οποιαδήποτε άλλη κατάσταση που δεν επαληθεύτηκε επιτυχώς, επιστρέφει τη σελίδα "login_error.html" και προγραμματίζει μια ανακατεύθυνση για ανανέωση της σελίδας μετά από 1 δευτερόλεπτο.

5.4 Αρχικοποίηση διαδρομών στις σελίδες του συστήματος

Γενικά, εκτενής αναφορά στο πως αλληλεπιδρά ο χρήστης με τις σελίδες του συστήματος θα γίνει στο επόμενο κεφάλαιο όπου θα δούμε το σύστημα από την οπτική γωνία του δασκάλου αντί του προγραμματιστή. Εδώ θα δούμε κομμάτια κώδικα που αναφέρονται στις σελίδες και θα αναλύσουμε τη λειτουργικότητά τους από πλευράς API. Δηλαδή, λειτουργικότητα με την οποία το Frontend μπορεί να επικοινωνήσει με το Backend. Έπειτα από τις κλασσικές μεθόδους (GET, POST, UPDATE, DELETE), θα δούμε ότι μερικές φορές γίνονται αναφορές σε παραπάνω custom μεθόδους. Στην πραγματικότητα χρησιμοποιούμε μία τεχνική όπου μέσω της POST συνήθως, προωθούμε μία μεταβλητή που αντιπροσωπεύει μια έξτρα μέθοδο για να αυξήσουμε τους τρόπους με τους οποίους μπορούμε να αλληλεπιδράσουμε με μία σελίδα.

```
@app.route("/dashboards/<teacher_fingerprint_id>", methods=['GET', 'POST'])
@login_required
def dashboards(teacher_fingerprint_id):
```

Εικόνα 21:Λογική της σελίδας των τάξεων

Η συγκεκριμένη διαδρομή, @app.route("/dashboards/<teacher_fingerprint_id>", methods=['GET', 'POST']), χρησιμοποιεί δύο HTTP μεθόδους: GET και POST. Ας δούμε τι κάνει η εφαρμογή για κάθε μία από αυτές τις μεθόδους:

1. **GET:** Όταν ο χρήστης προσπελαύνει τη σελίδα για πρώτη φορά ή ανανεώνει τη σελίδα, εφαρμόζεται η μέθοδος GET. Αν η GET είναι η ενεργή μέθοδος, η συνάρτηση απλά επιστρέφει το template "dashboards.html", παρουσιάζοντας στοιχεία των τάξεων (classes) και άλλες πληροφορίες για τον εκπαιδευτικό.
2. **POST:** Αυτή η μέθοδος χρησιμοποιείται για να υποβάλλει δεδομένα στον διακομιστή. Στη συγκεκριμένη περίπτωση, υπάρχουν διάφορες δυνατότητες για τη διαχείριση των τάξεων:
    - **POST:** Όταν η υπο-μέθοδος είναι POST, δημιουργείται μια νέα τάξη στη βάση δεδομένων με τις πληροφορίες που παρέχει ο χρήστης.
    - **FINISH:** Ενημερώνει τα στοιχεία μιας ήδη υπάρχουσας τάξης.
    - **DELETE:** Διαγράφει τα στοιχεία ενός φοιτητή από μια τάξη και στη συνέχεια διαγράφει την ίδια την τάξη.
    - **TO_CLASS:** Μεταφέρει τον χρήστη σε μια συγκεκριμένη τάξη.
    - **BACK:** Μεταφέρει τον χρήστη πίσω στο πίνακα ελέγχου (dashboard).

Γενικότερα, η συνάρτηση dashboards διαχειρίζεται τις λειτουργίες που σχετίζονται με το πίνακα ελέγχου του εκπαιδευτικού, όπως η δημιουργία, ενημέρωση, και διαγραφή τάξεων.

```
@app.route("/dashboards/<teacher_fingerprint_id>/<class_id>", methods=['GET', 'POST'])
@login_required
def students_dashboard(teacher_fingerprint_id, class_id):
```

Εικόνα 22:Λογική της σελίδας των μαθητών για μία τάξη

Ας δούμε τι κάνει η συνάρτηση students_dashboard:

Αρχικά, αυτή η διαδρομή (@app.route) αντιπροσωπεύει τον πίνακα ελέγχου για τους μαθητές ενός συγκεκριμένου μαθήματος. Παρέχει διαχειριστικές λειτουργίες όπως η προσθήκη, επεξεργασία, διαγραφή και παρακολούθηση παρουσιών των μαθητών.

1. **GET:** Κατά την πρώτη πρόσβαση ή ανανέωση της σελίδας, η μέθοδος GET είναι ενεργή. Η σελίδα προβάλλει τον κατάλογο των μαθητών για το συγκεκριμένο μάθημα καθώς και την παρουσία τους, εφόσον το show_attendance είναι ενεργοποιημένο.
2. **POST:** Είναι ενεργή όταν υποβάλλονται φόρμες από το frontend:
    - **POST:** Προσθέτει έναν νέο μαθητή στο μάθημα. Αν ο μαθητής υπάρχει ήδη, εμφανίζει ένα μήνυμα λάθους. Διαφορετικά, δημιουργεί έναν νέο μαθητή και τον συνδέει με το συγκεκριμένο μάθημα και τον εκπαιδευτικό.
    - **FINISH:** Ενημερώνει τα στοιχεία ενός υπάρχοντος μαθητή.
    - **DELETE:** Διαγράφει έναν μαθητή από το μάθημα.
    - **BACK:** Μεταφέρει τον χρήστη πίσω στον πίνακα ελέγχου του εκπαιδευτικού.

Γενικά, η students_dashboard συνάρτηση παρέχει μια διεπαφή για τη διαχείριση των μαθητών ενός συγκεκριμένου μαθήματος, καθώς και για την παρακολούθηση της παρουσίας τους.

```
@app.route("/dashboards/<teacher_fingerprint_id>/<class_id>/attendance", methods=['GET', 'POST'])
@login_required
def attendance(teacher_fingerprint_id, class_id):
```

Εικόνα 23:Λογική της σελίδας λήψης παρουσιών για τους μαθητές

Εδώ έχει πιο πολλή αξία να δούμε πως αλληλεπιδρά με τη βάση σε σημεία κλειδιά όταν επιλέγουμε POST.

1. Φορτώνει τα δεδομένα του διδάσκοντα (teacher_data) και τα δεδομένα του μαθήματος (class_data) βάσει των παραμέτρων που δίνονται στη διαδρομή URL.
2. Έλεγχος για την έναρξη του μαθήματος:

- Αν το μάθημα έχει ξεκινήσει προηγουμένως (commenced), ελέγχει αν έχει καταγραφεί για τη σημερινή ημερομηνία. Αν όχι, προσθέτει τη σημερινή ημερομηνία στη λίστα.
- Εάν το μάθημα δεν έχει ξεκινήσει ποτέ, καταγράφει την έναρξη για τη σημερινή ημερομηνία.

3. Εκτελεί την εντολή verifyUser για να επαληθεύσει τον μαθητή που προσπαθεί να καταγράψει την παρουσία του.
4. Ελέγχει αν ο μαθητής ανήκει στο συγκεκριμένο μάθημα:
   - Αν ναι, ελέγχει την καταγραφή παρουσιών του μαθητή για τη σημερινή ημερομηνία.
   - Αν ο μαθητής έχει ήδη καταγραφεί, επιστρέφει μήνυμα ότι η παρουσία του έχει ήδη ληφθεί (attendance_already_taken.html).
   - Αν δεν έχει καταγραφεί, τότε το σύστημα καταγράφει την παρουσία του μαθητή για τη σημερινή ημερομηνία και επιστρέφει μήνυμα ότι η παρουσία έχει ληφθεί επιτυχώς (attendance_taken.html).
5. Επιστρέφει το αντίστοιχο πρότυπο (template) με τα δεδομένα του διδάσκοντα, του μαθήματος, και του μαθητή (εάν υπάρχει).

5.5 Σελίδες τύπου HTML που σερβίρονται στο χρήστη

Δεν θα γίνει εκτενής ανάλυση στον κώδικα HTML καθώς η λειτουργικότητα του έχει πιο πολύ ενδιαφέρον από πλευράς αλληλεπίδρασης με τον χρήστη. Έχει όμως ενδιαφέρον να αναλύσουμε λίγο παραπάνω τον μηχανισμό με τον οποίο το Flask αλληλεπιδρά με αυτό. Όταν αρχικοποιούσαμε το πρόγραμμα, δώσαμε και το μονοπάτι ενός φακέλου στο Flask που έκανε τη διαδικασία δήλωσης των ιστοσελίδων πολύ πιο απλή. Το μοναδικό που χρειαζόμασταν ήταν να εισάγουμε τα ονόματα των σελίδων που θέλαμε να εμφανιστούν μέσα σε ειδικές συναρτήσεις και να τις επιστρέψουμε στον τέλος των διαδρομών. Από εκείνο το σημείο, το Flask αναλαμβάνει όλες τις ενδιάμεσες λεπτομέρειες. Επιπροσθέτως, ο HTML κώδικας, άρχισε να υποστηρίζει κώδικα για τη δυναμική εισαγωγή δεδομένων από μεταβλητές Python. Επίσης, απέκτησε τη δυνατότητα εκτέλεσης ρουτινών επεξεργασίας μέσω ειδικής σύνταξης που ονομάζεται Jinja. Εδώ είναι ένα παράδειγμα στο οποίο φαίνονται οι δυνατότητες αυτές.

```
    {% for class in classes %}
      {% if class.id|string == class_id_for_edit|string %}
        <form method="post">
          <input type="hidden" name="method" value="FINISH"/>
          <input type="hidden" name="class_id_for_edit" value="{{ class.id }}">
          <tr style="border: 5px solid white;">
              <td style="border: 5px solid white; color:white; padding: 5px; color: black;"><input type="text" name="name" value="{{ class.name }}"/></td>
              <td style="border: 5px solid white; color:white; padding: 5px; color: black;"><input type="text" name="faculty" value="{{ class.faculty }}"/></td>
              <td style="border: 5px solid white; color:white; padding: 5px; color: black;"><input type="text" name="subject" value="{{ class.subject }}"/></td>
              <td style="border: 5px solid white; color:white; padding: 5px; color: black;"><input type="text" name="room" value="{{ class.room }}"/></td>
              <td style="border: 5px solid white; color:white; padding: 5px; color: black;"><p style="color:white;">is lab </p><input type="checkbox" id="is_lab" name="is_lab" {% if
class.is_lab %} checked {% endif %}></td>
              <td style="border: 5px solid white; color:white; padding: 5px; color: black;"><input type="text" name="period" value="{{ class.period }}"/></td>
              <td style="border: 5px solid white; color:white; padding: 5px 5px 5px 15px; color: black; text-align: center;"><input type="submit" value="Finish"/></td>
          </tr>
        </form>
      {% else %}
        <tr style="border: 5px solid white;">
          <td style="padding: 5px 5px 5px 10px;">
            <form method="POST">
              <input type="hidden" name="method" value="TO_CLASS" />
              <input type="hidden" name="class_id" value="{{ class.id }}">
              <input class="btn-link" type="submit" value="{{ class.name }}">
            </form>
          </td>
```

Εικόνα 24: Απόκομμα σελίδας HTML με δυνατότητες δυναμικής εκτέλεσης ρουτινών επεξεργασίας μέσω Jinja

Έχοντας τη δυνατότητα να καταναλώσουμε την πληροφορία ενός Python αντικειμένου, και να δηλώσουμε προγραμματιστικές δομές όπως την for loop ή την if else μπορέσαμε να σώσουμε πολύ χρόνο προγραμματισμού και να κάνουμε τον κώδικα του συστήματος μας πολύ πιο κατανοητό και συντηρήσιμο. Αυτός είναι ο λόγος που το Flask είναι μία τόσο δημοφιλής βιβλιοθήκη για την κατασκευή ιστοσελίδων.

5.6 Επικοινωνία με την βιβλιοθήκη του δακτυλικού

Στο πρόγραμμα μας, πρέπει πολλές φορές να επικοινωνήσουμε με το σύστημα του δακτυλικού. Αυτή η μέθοδος δρα ως πύλη με την οποία η Python μπορεί να στείλει εντολές στο σύστημα. Μέρος του προγράμματος δίνεται μαζί με το δακτυλικό hardware. Έπρεπε όμως να καταβληθεί πολλή προσπάθεια για να προσαρμοστεί σε ένα σύνολο από ρουτίνες που μπορεί να χρησιμοποιήσει απευθείας το σύστημα μου.

```
def executeCommand(fn):
    GPIO.output(Finger_RST_Pin, GPIO.LOW)
    time.sleep(0.25)
    GPIO.output(Finger_RST_Pin, GPIO.HIGH)
    time.sleep(0.25)

    return fn()
```

Εικόνα 25:Επικοινωνία της Python με τον αισθητήρα του δακτυλικού αποτυπώματος

1. Ενεργοποιεί το pin Finger_RST_Pin της βιβλιοθήκης GPIO σε χαμηλή κατάσταση (GPIO.LOW). Αυτό μπορεί να είναι μέρος μιας διαδικασίας

επανεκκίνησης ή αρχικοποίησης ενός συστήματος ανίχνευσης δακτυλικών αποτυπωμάτων.

2. Κοιμάται για 0.25 δευτερόλεπτα με την time.sleep(0.25), υποθετικά για να δώσει χρόνο στο σύστημα να ανταποκριθεί ή να ολοκληρώσει την προηγούμενη εντολή.
3. Ενεργοποιεί το pin Finger_RST_Pin πάλι σε υψηλή κατάσταση (GPIO.HIGH). Αυτό μπορεί να είναι το επόμενο βήμα στη διαδικασία επανεκκίνησης ή αρχικοποίησης.
4. Κοιμάται πάλι για 0.25 δευτερόλεπτα.
5. Τέλος, εκτελεί τη συνάρτηση fn() που περνάει ως όρισμα και επιστρέφει το αποτέλεσμά της.

Εν συντομία, η executeCommand είναι μια συνάρτηση που προετοιμάζει το σύστημα ανίχνευσης δακτυλικών αποτυπωμάτων πριν από την εκτέλεση μιας συγκεκριμένης εντολής ή λειτουργίας και επιστρέφει το αποτέλεσμά της.

Ακολουθεί αναφορά στις ρουτίνες

```
#**************************************************************************
# @brief    Register fingerprint
#**************************************************************************/
def addUser():
    global g_rx_buf
    r = GetUserCount()
    if r >= USER_MAX_CNT:
        return [ACK_FULL, -1]

    command_buf = [CMD_ADD_1, 0, r + 1, 3, 0]
    r = TxAndRxCmd(command_buf, 8, 6)

    i = 1
    while r == 0 and g_rx_buf[4] == 6:
        command_buf = [CMD_ADD_1, 0, r + i, 3, 0]
        r = TxAndRxCmd(command_buf, 8, 6)
        i += 1

    if r == ACK_TIMEOUT:
        return [ACK_TIMEOUT, -1]
    if r == ACK_SUCCESS and g_rx_buf[4] == ACK_SUCCESS:
        command_buf[0] = CMD_ADD_3
        r = TxAndRxCmd(command_buf, 8, 6)
        if r == ACK_TIMEOUT:
            return [ACK_TIMEOUT, -1]
        if r == ACK_SUCCESS and g_rx_buf[4] == ACK_SUCCESS:
            return [ACK_SUCCESS, command_buf[2]]
        else:
            return [ACK_FAIL, -1]
    else:
        return [ACK_FAIL, -1]
```

Εικόνα 26: Εισαγωγή καινούργιου δακτυλικού αποτυπώματος

1. Χρησιμοποιεί τη συνάρτηση GetUserCount για να λάβει τον αριθμό των ήδη εγγεγραμμένων δακτυλικών αποτυπωμάτων.
2. Ελέγχει αν έχει φτάσει στο μέγιστο αριθμό εγγραφών (USER_MAX_CNT). Εάν ναι, επιστρέφει ACK_FULL, που σημαίνει ότι ο αισθητήρας είναι πλήρης.
3. Αλλιώς, δημιουργεί ένα πακέτο εντολής (command_buf) με την εντολή CMD_ADD_1 και το ID του νέου χρήστη (r + 1).
4. Στη συνέχεια, χρησιμοποιεί την TxAndRxCmd για να στείλει την εντολή στον αισθητήρα δακτυλικών αποτυπωμάτων.
5. Εάν η απάντηση είναι επιτυχής (ACK_SUCCESS) και το g_rx_buf[4] είναι επίσης ACK_SUCCESS, τότε αλλάζει την εντολή σε CMD_ADD_3 και ξαναστέλνει την εντολή στον αισθητήρα.

6. Εάν η απάντηση είναι και πάλι επιτυχής, τότε επιστρέφει ACK_SUCCESS με το ID του νέου χρήστη. Σε περίπτωση αποτυχίας, επιστρέφει ACK_FAIL.

```
#*****************************************************************************
# @brief    Fingerprint matching
#*****************************************************************************/
def verifyUser():
    global g_rx_buf
    command_buf = [CMD_MATCH, 0, 0, 0, 0]
    r = TxAndRxCmd(command_buf, 8, 5);

    if r == ACK_TIMEOUT:
        return [ACK_TIMEOUT, -1]
    if r == ACK_SUCCESS and IsMasterUser(g_rx_buf[4]) == TRUE:
        return [ACK_SUCCESS, g_rx_buf[3]]
    elif g_rx_buf[4] == ACK_NO_USER:
        return [ACK_NO_USER, -1]
    elif g_rx_buf[4] == ACK_TIMEOUT:
        return [ACK_TIMEOUT, -1]
    else:
        return [ACK_GO_OUT, -1]   # The center of the fingerprint is out of alignment with sensor
```

Εικόνα 27: Ταυτοποίηση δακτυλικού αποτυπώματος

1. Στην αρχή, χρησιμοποιεί την TxAndRxCmd για να στείλει μια εντολή (CMD_MATCH) στον αισθητήρα δακτυλικών αποτυπωμάτων και να λάβει μια απάντηση.
2. Αν η απάντηση (r) είναι ACK_TIMEOUT, τότε σημαίνει ότι υπήρξε χρονική υπέρβαση και η συνάρτηση επιστρέφει αυτό το σφάλμα με τον κωδικό -1.
3. Αν η απάντηση (r) είναι ACK_SUCCESS και το δακτυλικό αποτύπωμα ανήκει σε έναν κύριο χρήστη (IsMasterUser(g_rx_buf[4]) επιστρέφει TRUE), τότε η συνάρτηση επιστρέφει με επιτυχία και δίνει τον κωδικό του χρήστη (g_rx_buf[3]).
4. Αν ο αισθητήρας επιστρέφει ότι δεν υπάρχει τέτοιος χρήστης (g_rx_buf[4] είναι ACK_NO_USER) ή αν υπάρχει ξανά χρονική υπέρβαση, τότε επιστρέφεται το αντίστοιχο σφάλμα με τον κωδικό -1.
5. Σε όλες τις άλλες περιπτώσεις, επιστρέφεται το σφάλμα ACK_GO_OUT που σημαίνει ότι το κέντρο του δακτυλικού αποτυπώματος δεν είναι σε ευθυγράμμιση με τον αισθητήρα.

```
#**************************************************************************
# @brief    Clear user
#**************************************************************************/
def clearUser(user_id):
    global g_rx_buf
    command_buf = [CMD_DEL, 0, user_id, 0, 0]
    r = TxAndRxCmd(command_buf, 8, 5)
    if r == ACK_TIMEOUT:
        return ACK_TIMEOUT
    if r == ACK_SUCCESS and g_rx_buf[4] == ACK_SUCCESS:
        return ACK_SUCCESS
    else:
        return ACK_FAIL
```

Εικόνα 28: Εκκαθάριση δακτυλικού αποτυπώματος

1. Ορίζει μια εντολή CMD_DEL που, όπως μπορούμε να υποθέσουμε, χρησιμοποιείται για να διαγράψει ένα δακτυλικό αποτύπωμα βάσει του δοθέντος user_id.
2. Στη συνέχεια, χρησιμοποιεί την TxAndRxCmd για να στείλει την εντολή στον αισθητήρα δακτυλικών αποτυπωμάτων και να λάβει μια απάντηση.
3. Αν η απάντηση (r) είναι ACK_TIMEOUT, τότε σημαίνει ότι υπήρξε χρονική υπέρβαση και η συνάρτηση επιστρέφει αυτό το σφάλμα.
4. Αν η απάντηση (r) είναι ACK_SUCCESS και η απάντηση από τον αισθητήρα είναι επίσης ACK_SUCCESS (καθορίζεται από το g_rx_buf[4]), τότε η συνάρτηση επιστρέφει επιτυχία.
5. Σε όλες τις άλλες περιπτώσεις, επιστρέφει ACK_FAIL, που σημαίνει ότι η διαδικασία αποτυγχάνει.

5.7 Εκκίνηση της εφαρμογής

```
if __name__ == "__main__":
    with app.app_context():
        db.create_all()
        app.run(debug=True)
```

Εικόνα 29: Εκκίνηση της εφαρμογής

Τώρα που οι συνδέσεις και οι αρχικοποιήσεις έχουν γίνει επιτυχώς, η εφαρμογή θα αρχίσει να τρέχει.

1. if __name__ == "__main__":: Ελέγχει αν το σενάριο εκτελείται ως κύριο πρόγραμμα. Εάν το σενάριο εισάγονταν ως μονάδα σε κάποιο άλλο πρόγραμμα, δεν θα εκτελούσε τον κώδικα που βρίσκεται μέσα σε αυτό το τμήμα.
2. with app.app_context():: Δημιουργεί ένα περιβάλλον (context) για την εφαρμογή Flask. Σε αυτό το περιβάλλον, μπορεί να εκτελέσει ενέργειες που χρειάζονται την εφαρμογή Flask να είναι παρούσα.
3. db.create_all(): Μέσα στο περιβάλλον της εφαρμογής, αυτή η εντολή δημιουργεί όλους τους πίνακες στη βάση δεδομένων που έχουν οριστεί αλλά δεν έχουν ακόμα δημιουργηθεί.
4. app.run(debug=True): Τέλος, εκκινεί την εφαρμογή Flask σε λειτουργία αποσφαλμάτωσης (debug mode). Αυτή η λειτουργία του επιτρέπει να βλέπει λεπτομερή μηνύματα σφαλμάτων και να κάνει αλλαγές στον κώδικα χωρίς να χρειάζεται να επανεκκινεί την εφαρμογή κάθε φορά.

5.8 Περιγραφή της ροής του προγράμματος

Τέλος, έχοντας αναλύσει όλα τα σημαντικά κομμάτια του συστήματος, ακολουθεί σύντομη περιγραφή του τρόπου με τον οποίο ρέει η πληροφορία στο σύστημα έπειτα από κάθε ενέργεια του χρήστη.

1. Ο χρήστης επιλέγει μία οποιαδήποτε ενέργεια στην ιστοσελίδα
2. Η ενέργεια αυτή στέλνει μία μοναδική διαδρομή και μέθοδο στο σύστημα
3. Το Flask παίρνει αυτήν την πληροφορία (και οποιαδήποτε άλλη πληροφορία έδωσε ο χρήστης αν χρησιμοποίησε φόρμα) και τη χρησιμοποιεί για να εκτελέσει την Python, ρουτίνα που αντιστοιχεί στη διαδρομή και τη μέθοδο που δόθηκε.
4. Αυτός ο κώδικας μπορεί να αλληλεπιδράσει με το σύστημα δακτυλικών ή τη βάση.
5. Αν αλληλεπιδράσει με το σύστημα δακτυλικών, θα χρησιμοποιήσει τον εξειδικευμένο κώδικα γραμμένο σε Python και την GPIO βιβλιοθήκη, για να στείλει ηλεκτρικό σήμα στο σύστημα. Θα πάρει πίσω καινούριο ηλεκτρικό σήμα που θα αποθηκευτεί σαν Python μεταβλητή και θα περιέχει στοιχεία για την αποτυχία ή την επιτυχία της πράξης

6. Αν αλληλεπιδράσει με τη βάση, θα χρησιμοποιήσει την SQLAlchemy.Αυτή η βιβλιοθήκη θα μετατρέψει τον Python κώδικα και την πληροφορία που θα της δοθεί σε SQL και θα το στείλει στην βάση PostgreSQL. Όταν η βάση τελειώσει την επεξεργασία, θα στείλει πίσω σήμα επιτυχίας ή αποτυχίας και κάποια παραπάνω δεδομένα που η SQLAlchemy θα μετατρέψει σε Python μεταβλητές.
7. Τέλος, όταν το πρόγραμμα τελειώσει με τα βήματα 5 ή 6 (ή και τα δύο), θα επιστρέψει πληροφορία ως Python μεταβλητές μαζί με μία καινούρια διεύθυνση και μια HTML σελίδα.
8. Πριν η σελίδα σερβιριστεί στον browser του χρήστη, το Flask θα χρησιμοποιήσει την Python πληροφορία που του δόθηκε και θα ψάξει για ειδικό συντακτικό ώστε να το εκτελέσει.
9. Ύστερα από την εκτέλεση του, η σελίδα θα περιέχει καθαρή HTML η οποία θα εμφανιστεί στο χρήστη

Έτσι τελειώνει ο κύκλος μιας ενέργειας του χρήστη.

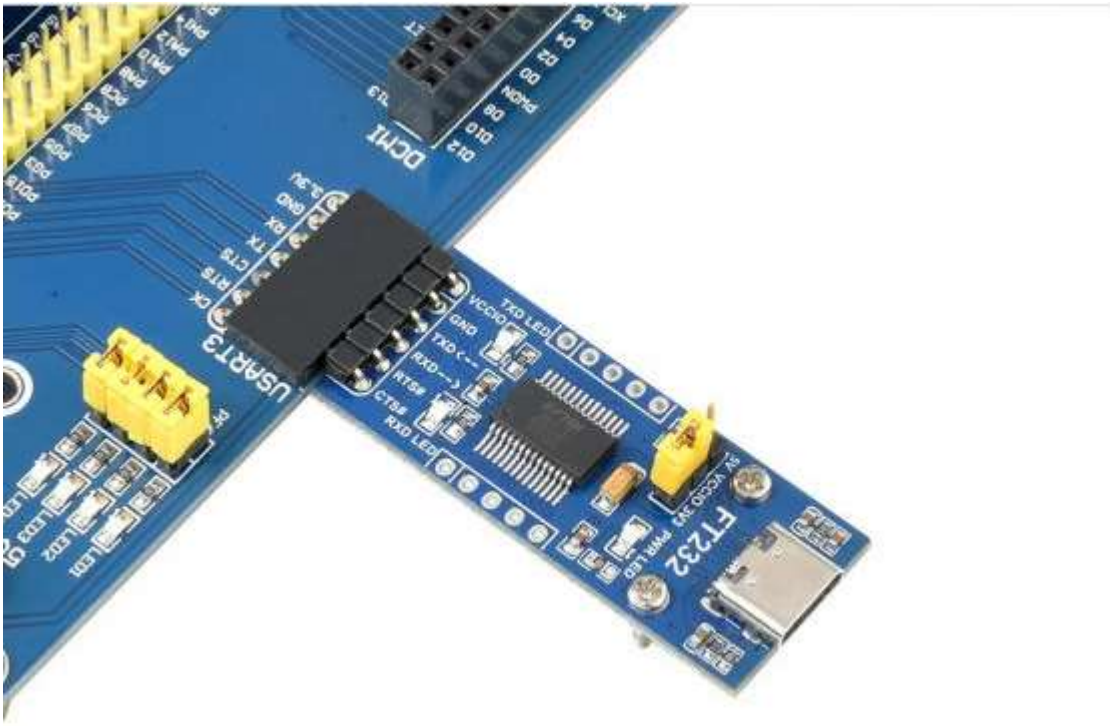
Κεφάλαιο 6ο

Hardware

Στο συγκεκριμένο κεφάλαιο θα γίνει αναφορά στο hardware που χρησιμοποιήσαμε για την δημιουργία της εφαρμογής.

6.1 USB to UART

Το FT232 UART UART Board είναι μια μονάδα UART που μετατρέπει τη διεπαφή USB σε επίπεδο TTL. Αυτή η μονάδα επιτρέπει την επικοινωνία δεδομένων συνδέοντας τη διεπαφή USB ενός υπολογιστή ή άλλης συσκευής με συσκευές λογικού επιπέδου TTL, όπως μικροελεγκτές, αισθητήρες κ.λπ.



Εικόνα 30: FT232 UART UART Board

6.2 Αισθητήρας Δακτυλικού Αποτυπώματος- waveshare

Πρόκειται για μια πλήρως ενσωματωμένη μονάδα αισθητήρα δακτυλικών αποτυπωμάτων. Η μονάδα ελέγχεται μέσω εντολών UART και είναι αρκετά εύκολη στη χρήση. Τα πλεονεκτήματά του περιλαμβάνουν παν-κατευθυντική επαλήθευση 360°, γρήγορη επαλήθευση, υψηλή σταθερότητα και χαμηλή κατανάλωση ενέργειας.

Βασισμένος σε έναν επεξεργαστή Cortex υψηλής απόδοσης, σε συνδυασμό με τον εμπορικό αλγόριθμο δακτυλικών αποτυπωμάτων υψηλής ασφάλειας, ο αισθητήρας δακτυλικών αποτυπωμάτων UART διαθέτει λειτουργίες όπως εγγραφή δακτυλικών αποτυπωμάτων, λήψη εικόνας, εύρεση λειτουργιών, δημιουργία και αποθήκευση προτύπων, αντιστοίχιση δακτυλικών αποτυπωμάτων κ.λπ..

Εικόνα 31: Αισθητήρας δακτυλικών αποτυπωμάτων – waveshare

6.3 Μικροϋπολογιστής- Raspberry Pi 4 Model B 8GB Elegant Kit

Το Raspberry Pi 4 Model B 8GB είναι ένας μικροϋπολογιστής που αναπτύχθηκε από τον Raspberry Pi Foundation. Πρόκειται για την τέταρτη γενιά της σειράς Raspberry Pi. Το Raspberry Pi 4 διαθέτει 8 GB RAM, ταχύτερη τετραπύρηνη CPU, υποστήριξη για διπλές οθόνες σε ανάλυση έως και 4K, Gigabit Ethernet, USB3.0, ασύρματο LAN, Bluetooth 5.0 και τροφοδοσία USB-C. Η ταχύτητα και η απόδοση του Raspberry Pi 4 είναι ένα βήμα παραπάνω από τα προηγούμενα μοντέλα. Πρόκειται για μια ολοκληρωμένη εμπειρία επιφάνειας εργασίας, ομαλή και πολύ αναγνωρίσιμη αλλά σε μικρότερο, πιο ενεργειακά αποδοτικό και πολύ πιο οικονομικό υπολογιστή.

Στο συγκεκριμένο κιτ περιλαμβάνεται και κάρτα μνήμης micro sd 32 GB.

Εικόνα 32: Raspberry Pi 4 Model B 8GB

Κεφάλαιο 7ο

Διεπαφή Χρήστη

Στο συγκεκριμένο κεφάλαιο παραθέτονται οι οδηγίες χρήσης του από την πλευρά του καθηγητή.

7.1 Εγκατάσταση Λογισμικού

Τα βήματα για την εγκατάσταση λειτουργικού συστήματος είναι :

α) λήψη και εγκατάσταση προγράμματος imager για λειτουργικό σύστημα windows (https://www.raspberrypi.org/software/)

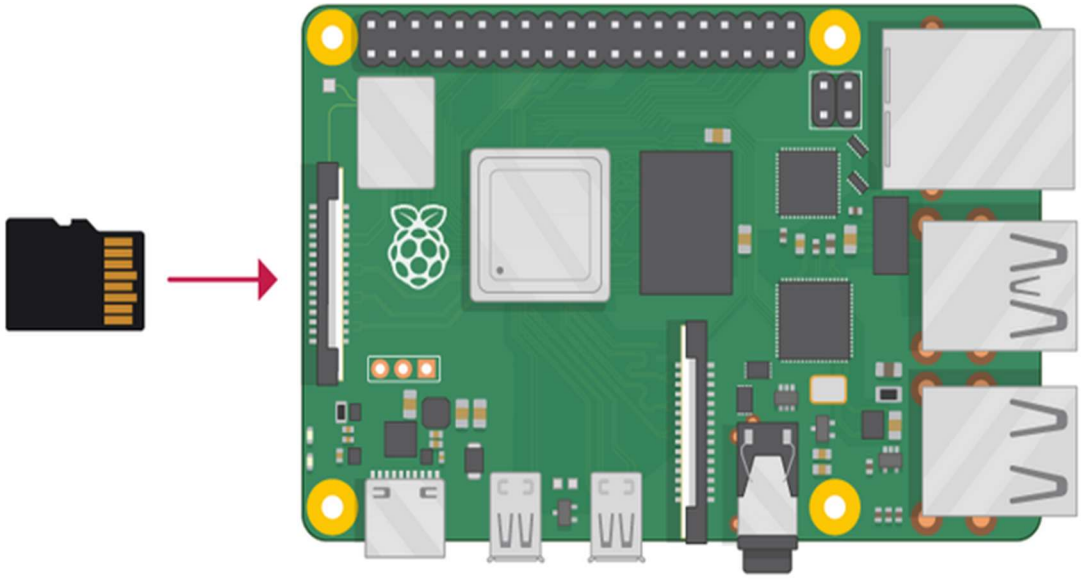
β) Τοποθέτηση της κάρτας SD στην κατάλληλη θύρα του υπολογιστή ή σε κατάλληλο usb αντάπτορα που θα τοποθετηθεί στην θύρα usb του υπολογιστή.

γ) Διαμόρφωση της κάρτας SD σε κωδικοποίηση NTFS ή FAT στην προκειμένη περίπτωση σε NTFS γινάτι χέει την καλύτερη αποδοχή στη διαχειριστή χώρου.

δ) Εκτελούμε το πρόγραμμα imager και από την καρτέλα Operating System διαλέγουμε το επιθυμητό λειτουργικό σύστημα και με το πλήκτρο write εκκινείται η διαδικασία εγγραφής του λειτουργικού στην κράτα SD.

ε) Τοποθέτηση του Raspberry στο ειδικά κατασκευασμένο κουτί προστασίας και στην συνέχεια γίνεται εισαγωγή της κάρτας SD στη θύρα του Raspberry που βρίσκεται στο κάτω μέρος της πλακέτας καθώς και την σύνδεση των υπολοίπων καλωδίων και εξαρτημάτων στις αντιστοιχίες θύρες (τροφοδοτικό, καλώδιο δικτύου, πληκτρολόγιο, ποντίκι, καλώδιο HDMI).

Εικόνα 33: Θήκη RaspberryPi



Εικόνα 34: Πλακέτα RaspberryPi

στ) Μόλις γίνει η ενεργοποίηση του Raspberry το σύστημα θα ζητήσει να γίνουν οι απαραίτητες ρυθμίσεις

B1 : Καλώς ήρθατε στο Raspberry Pi Desktop -> επιλέγουμε next

B2 : Set Country -> επιλέγουμε χώρα και γλώσσα

B3 : change Password -> επιλέγουμε κωδικό για το σύστημα (user -> pi ,pass ->1234)

B4 : set up Screen -> επιλέγουμε next

η) Για να γίνει εφικτή η σύνδεση μέσω Putty στο raspberry πρέπει να βρούμε την IP raspberry , αυτό μπορεί να γίνει με αρκετούς τρόπους θα δούμε μερικούς παρακάτω.

η1) Από ένα σταθμό εργασίας που είναι συνδεδεμένος στο ίδιο δίκτυο που είναι συνδεδεμένο και το raspberry ανοίγουμε ένα τερματικό (cmd) και πληκτρολογούμε την εντολή arp -a , Αυτή η εντολή θα εμφανίσει όλες της IP του τοπικού δικτύου.

η2) Τρέχουμε την εντολή ipconfig και θα εμφανίσει την ip του υπολογιστή που χρησιμοποιούμε , την μάσκα του δικτύου και την προεπιλεγμένη πύλη , αν έχουμε δυο συσκευές μόνο και γνωρίζουμε την ip του μηχανήματος που εργαζόμαστε η άλλη θα είναι του Raspberry

η3) Ένας άλλος τρόπος είναι να ανοίξουμε ένα φυλλομετρητή και να πληκτρολογήσουμε την προεπιλεγμένη ip του router (ZXHN H108N V2.5) και να συνδεθούμε στο διαχειριστικό του και από την καρτέλα Network -> Lan βλέπουμε το όνομα την IP και την MAC όλων των συσκευών που είναι συνδεδεμένες στο δίκτυο μας .

Η σύνδεση με το Raspberry γίνεται με τον παρακάτω τρόπο:

Ανοίγουμε το πρόγραμμα Putty και στο πεδίο host πληκτρολογούμε την ip διεύθυνση του raspberry και πιέζουμε το κουμπί open και θα ανοίξει ένα τερματικό.

ι1) Στο σημείο αυτό μας ζητείται από το σύστημα να εισάγουμε το username και το password της συσκευής που χρειάζεται να υπάρχει πρόσβαση.

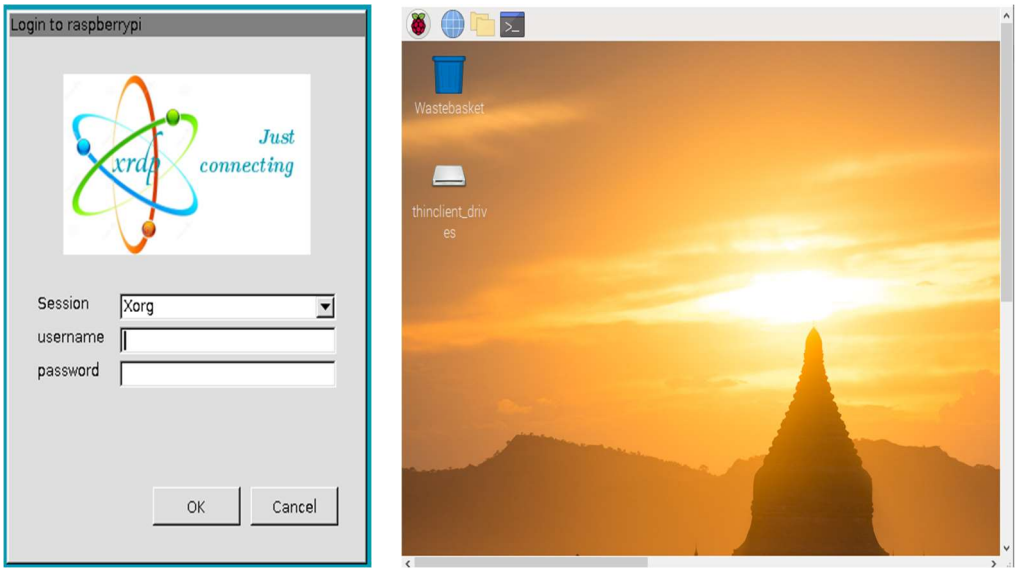
ι2) Αυτός ο τρόπος συνδέσεις είναι χρήσιμος για την εγκατάσταση πακέτων και προγραμμάτων μέσω τερματικού (cmd) , στην δίκια μας περίπτωση χρειάζεται να εγκαταστήσετε κάποια πακέτα βιβλιοθηκών ώστε να είναι δυνατή η απομακρυσμένη πρόσβαση με γραφικό περιβάλλον αυτό προκύπτει εκτελώντας την παρακάτω εντολές.

sudo apt-get install xrdp

sudo apt-get install tightvncserver

ι3) Μετά την ολοκλήρωση της εγκατάστασης επιτυχώς

επιχειρούμε την απομακρυσμένη πρόσβαση γράφοντας την IP του μηχανήματος και επιλέγουμε το κουμπί σύνδεση έπειτα παρατηρούμε ότι το πρωτόκολλο ανταπεξέρχεται και εμφανίστηκε ένα άλλο παράθυρο στο όποιο πρέπει να γράψουμε ip και password ώστε να ολοκληρωθεί η σύνδεση μέσω γραφικού περιβάλλοντος στο raspberry .



Εικόνα 35: Αναδυόμενο παράθυρο για απομακρυσμένη σύνδεση και επιφάνεια εργασίας

ι4) Αν δεν έχουν εγκατασταθεί η παραπάνω εντολές τότε δεν είναι δυνατή η απομακρυσμένη σύνδεση με γραφικό περιβάλλον.

7.2 Δημιουργία νέου χρήστη στο Raspberry

Για να δημιουργήσετε έναν νέο χρήστη μέσω του τερματικού (cmd) στο Raspberry Pi, ακολουθήστε αυτά τα βήματα:

1)Άνοιγμα του τερματικού: Συνδεθείτε με το Raspberry Pi είτε μέσω SSH εάν είναι συνδεδεμένο στο δίκτυο, είτε ανοίξτε ένα τερματικό κατευθείαν από την κονσόλα του.

2)Εκτέλεση με δικαιώματα διαχειριστή: Για να δημιουργήσετε ένα νέο χρήστη, θα πρέπει να είστε συνδεδεμένοι ως διαχειριστής ή να χρησιμοποιήσετε την εντολή sudo.

3)Εκτέλεση της εντολής προσθήκης χρήστη: Χρησιμοποιήστε την εντολή adduser για να δημιουργήσετε ένα νέο χρήστη.

Παράδειγμα:

sudo adduser newusername

Θα σας ζητηθεί να εισαγάγετε τον κωδικό πρόσβασης και άλλες πληροφορίες για τον νέο χρήστη.

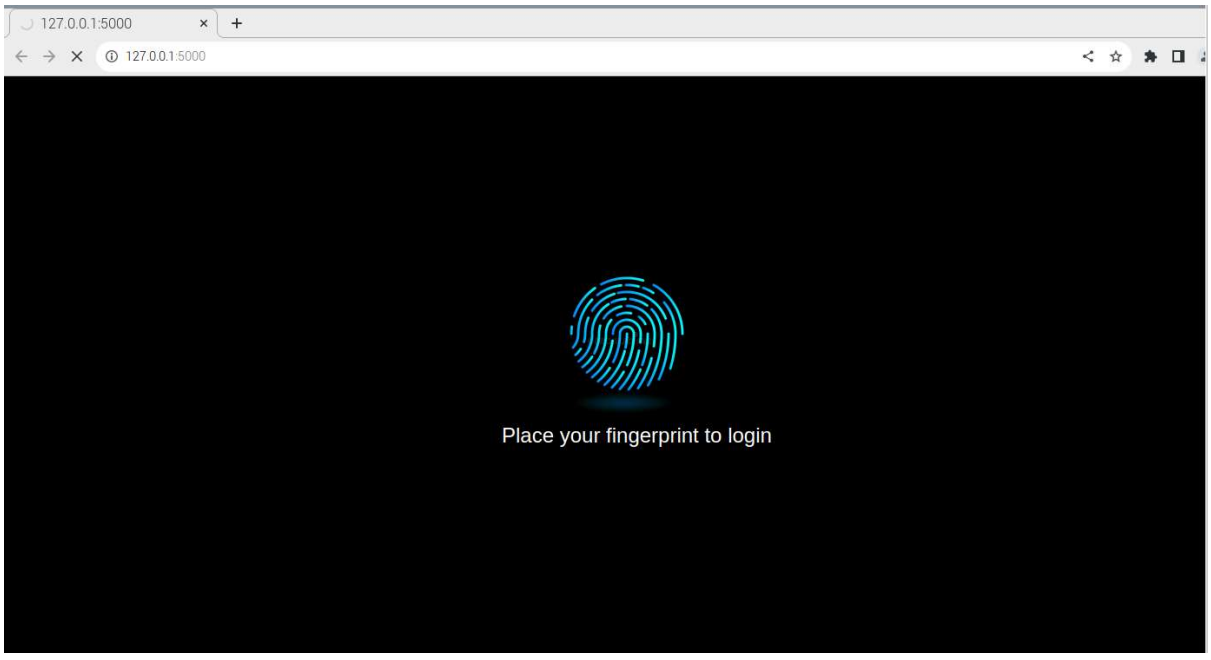
Εάν θέλετε να δώσετε στον νέο χρήστη δικαιώματα διαχειριστή, μπορείτε να τον προσθέσετε στην ομάδα sudo ώστε να μπορεί να εκτελεί εντολές με δικαιώματα διαχειριστή:

sudo usermod -aG sudo newusername

Με αυτά τα βήματα, θα έχετε δημιουργήσει ένα νέο χρήστη στο Raspberry Pi με το όνομα "newusername". Μην ξεχάσετε να αντικαταστήσετε το "newusername" με το επιθυμητό όνομα του χρήστη που θέλετε να δημιουργήσετε.

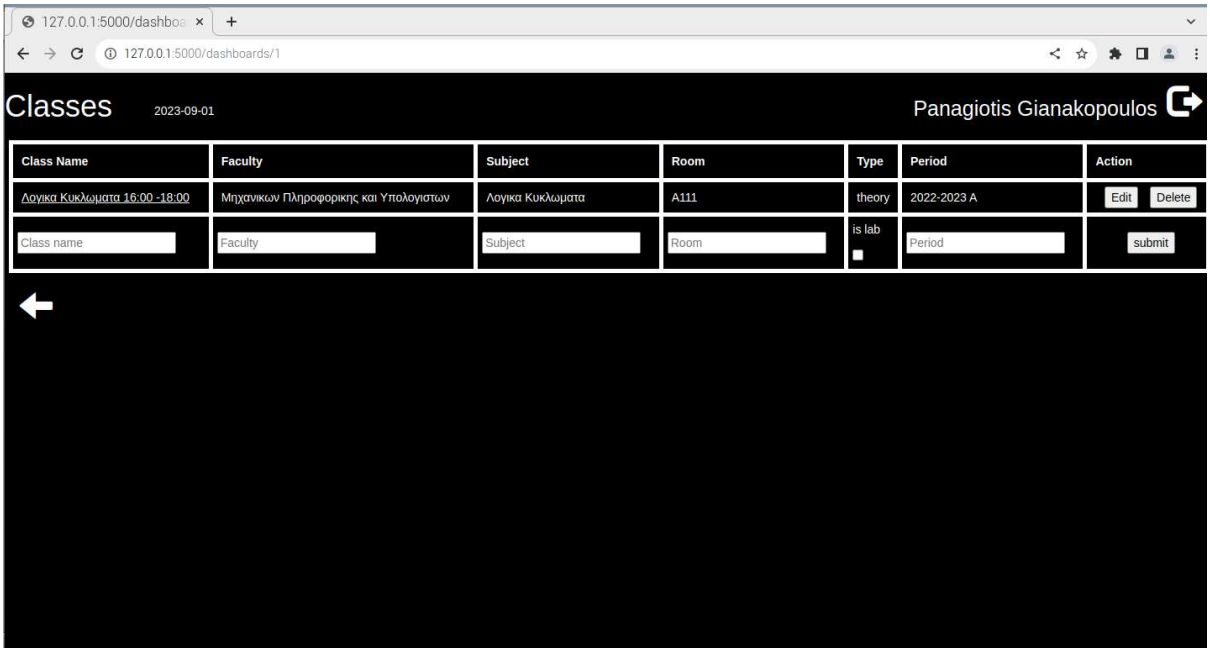
7.3 Παράδειγμα χρήσης διαχειριστικού ως καθηγητής

Αρχικά γίνεται η αυθεντικοποιηση του καθηγητή στο σύστημα.



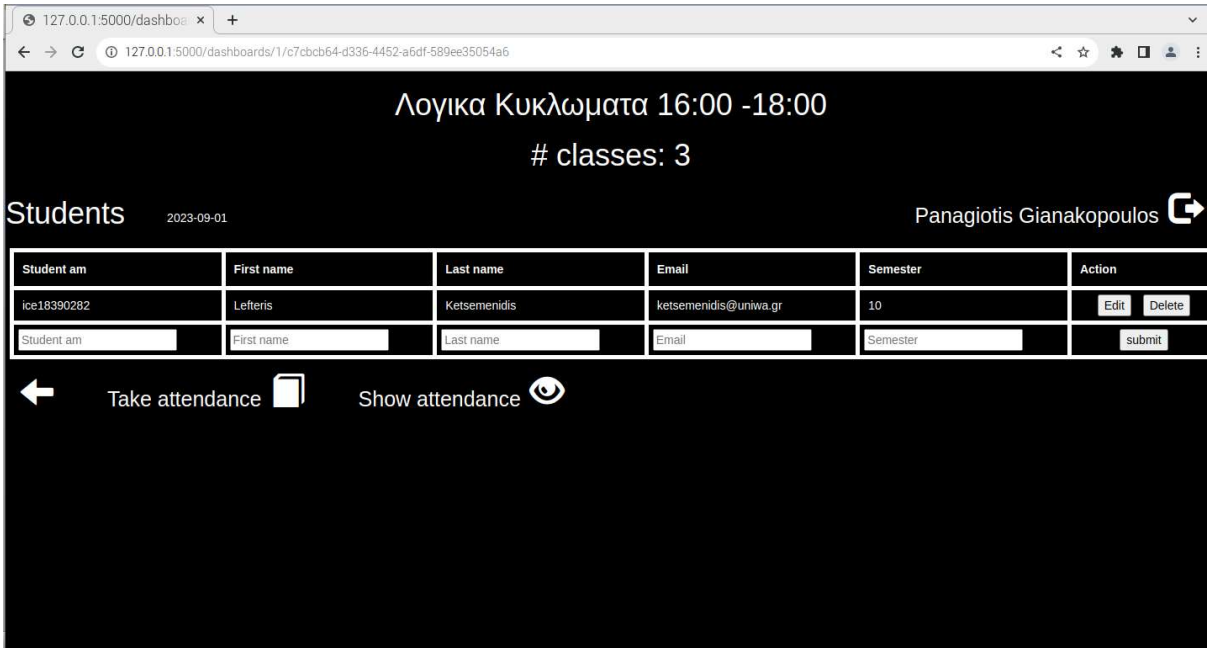
Εικόνα 36: Αρχική σελίδα εφαρμογής για Login του Καθηγητή

Στη συνεχεία γίνεται η δημιουργία του τμήματος με τίτλο «λογικά κυκλώματα» και εισάγουμε τιμές στα πεδία.



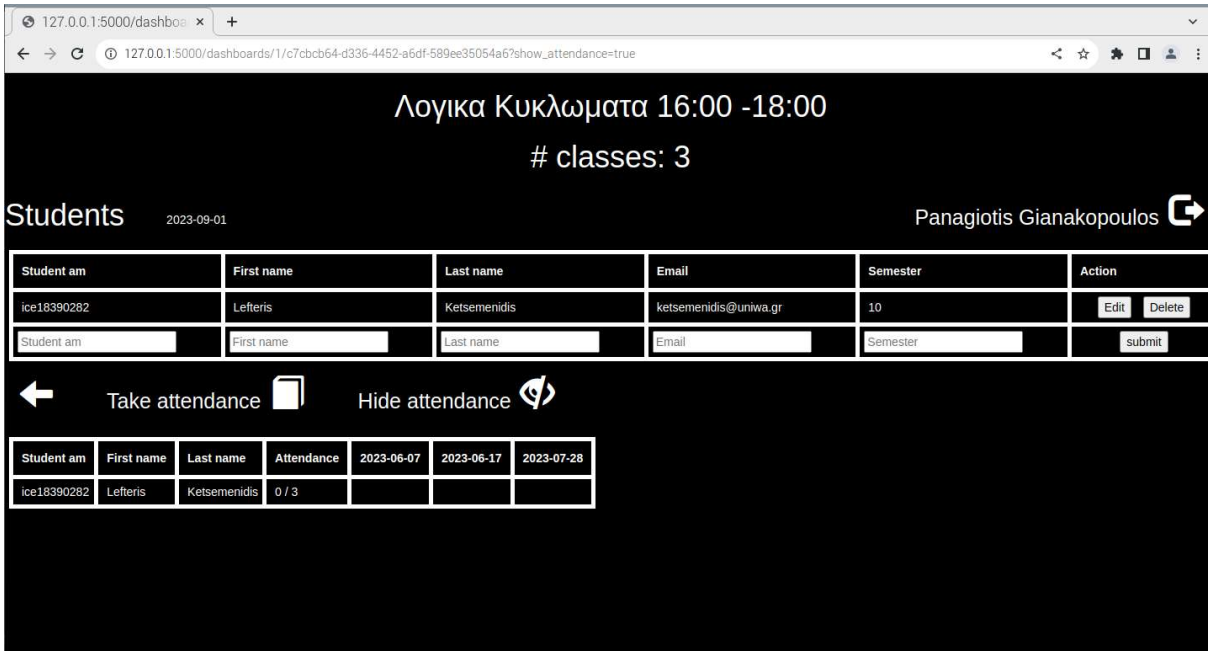
Εικόνα 37: Δημιουργία τμήματος

Έπειτα, προστίθεται ο πρώτος φοιτητής στο συγκεκριμένο τμήμα.



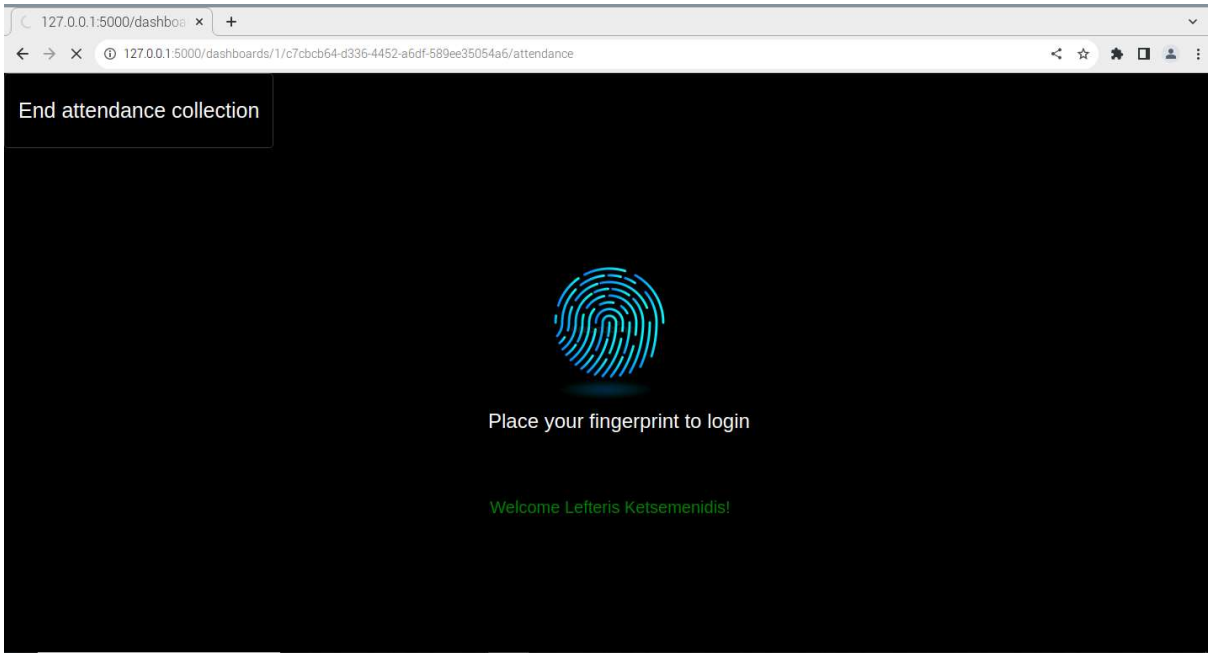
Εικόνα 38: Εισαγωγή στοιχείων φοιτητή

Πατώντας το κουμπί “show attendance” θα εμφανιστούν οι παρουσίες και το κουμπί μετατρέπεται σε “Hide attendance”.



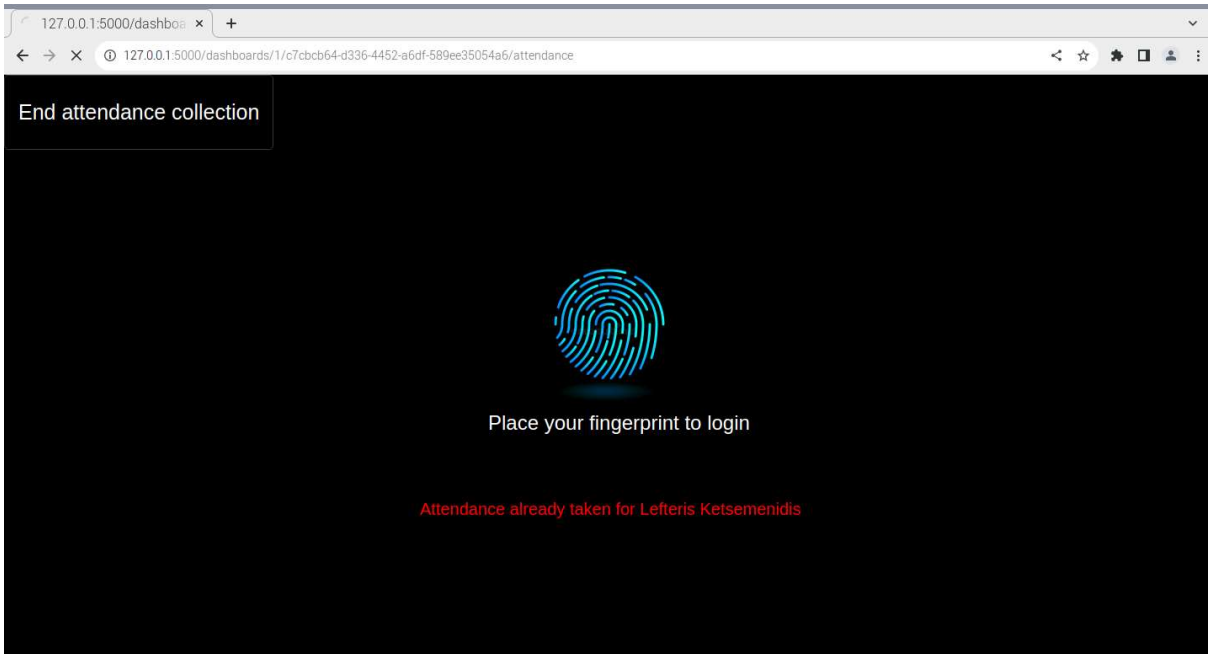
Εικόνα 39: Παρουσίες φοιτητή

Στη συνέχεια και αφού πατήσουμε το κουμπί “Take Attendance” ανοίγει η συνέδρια και οι φοιτητές μπορούν να περάσουν τα δακτυλικά τους αποτυπώματα για να παρθούν οι παρουσίες τους.



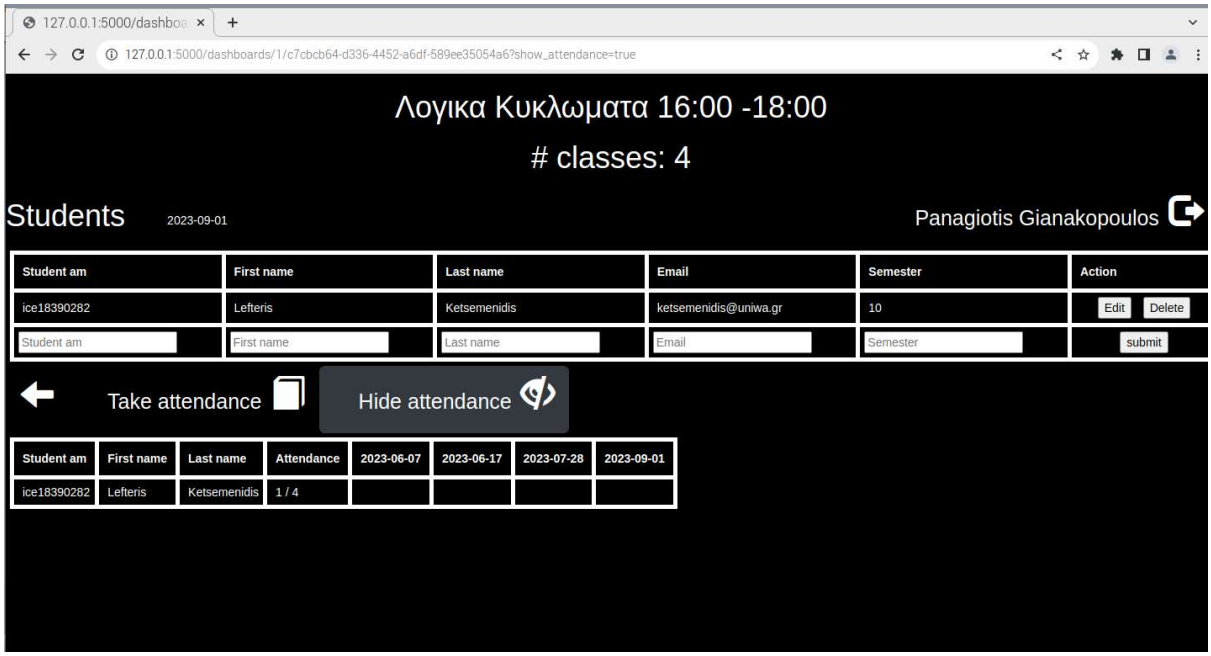
Εικόνα 40: Επιτυχής ταυτοποίηση φοιτητή

Σε περίπτωση που ο φοιτητής έχει ελεγχθεί βγαίνει το παρακάτω μήνυμα.



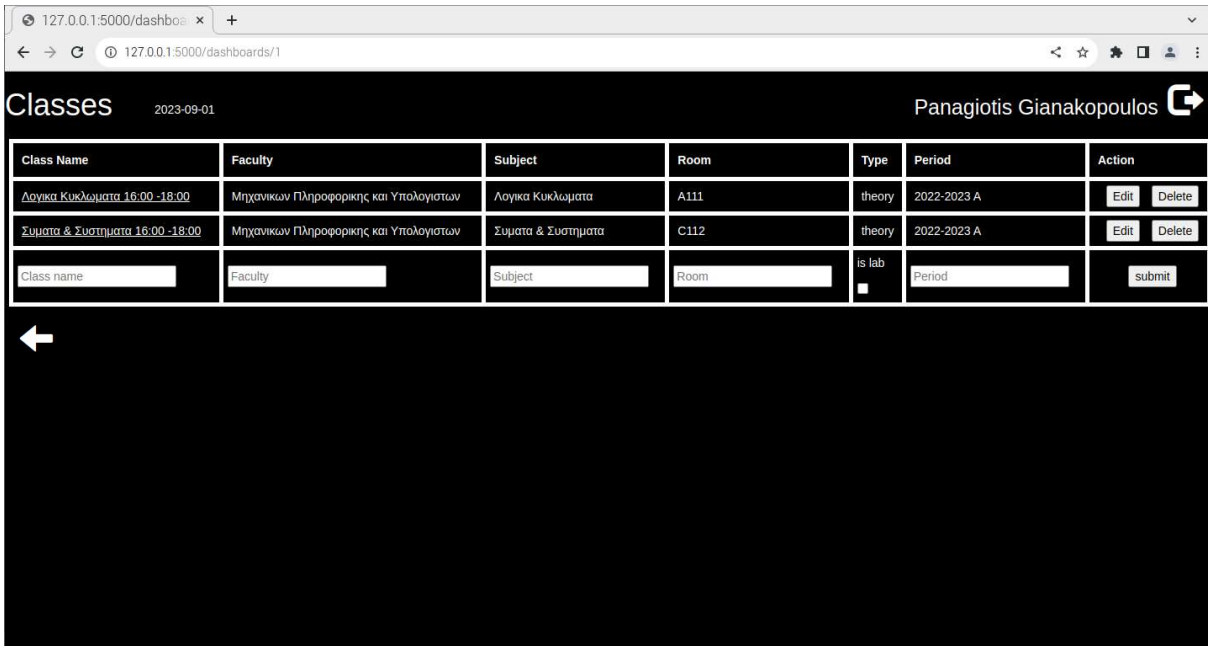
Εικόνα 41: Προειδοποίηση συστήματος για ήδη ελεγμένο φοιτητή

Στην παρακάτω εικόνα φαίνεται η επιτυχής παρουσία του ίδιου φοιτητή σε άλλη ημ/ναι στο ίδιο μάθημα.

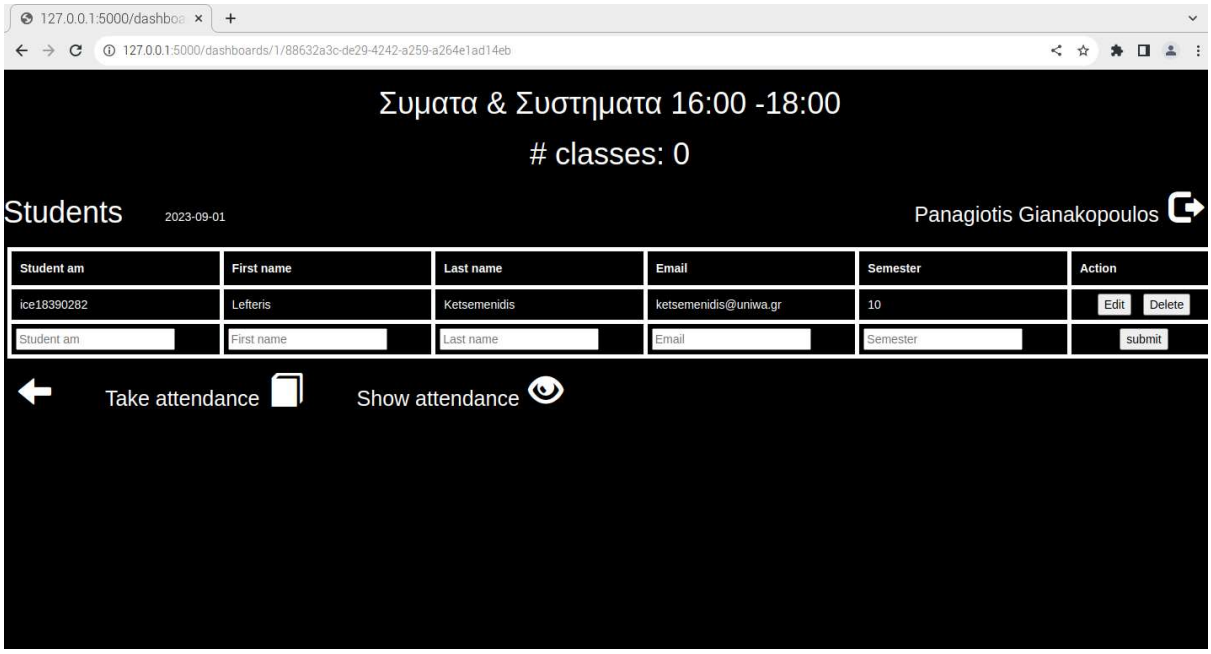


Εικόνα 42: Επιτυχής παρουσία ίδιου φοιτητή σε άλλη ημ/ναι στο ίδιο μάθημα

Έχουμε τη δυνατότητα να δημιουργήσουμε ένα νέο τμήμα εισάγοντας τα πεδία (class name, faculty, subject, room, type, period, action) που παρουσιάζονται στην παρακάτω εικόνα.

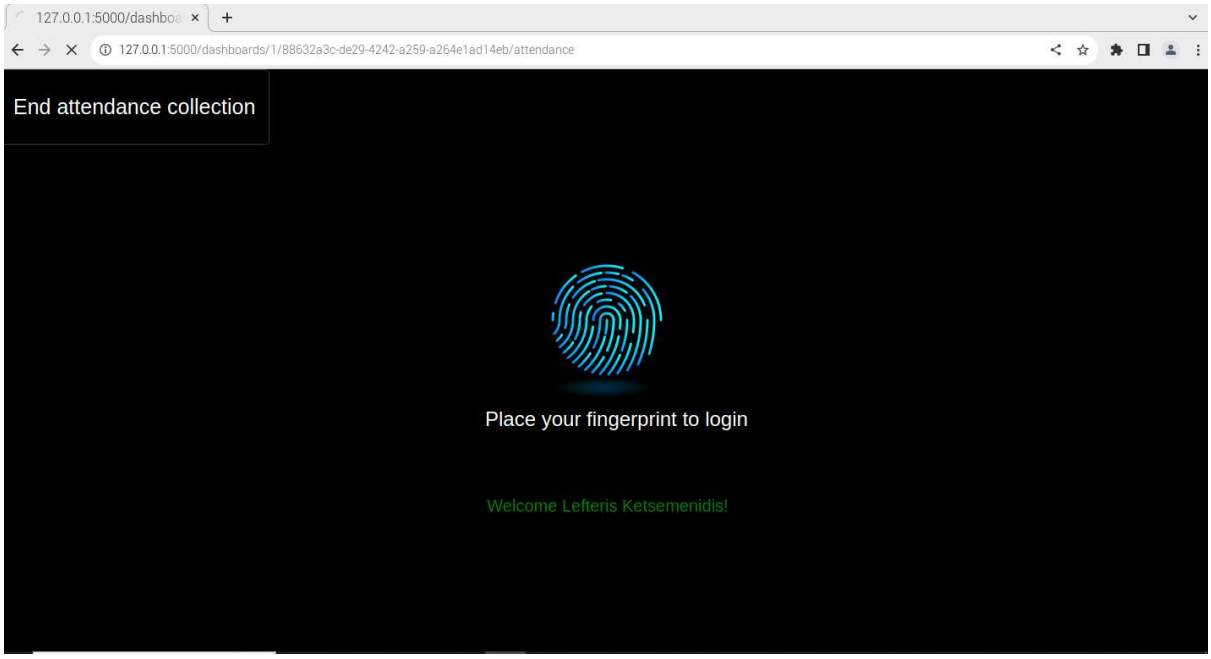


Εικόνα 43: Προσθήκη νέου τμήματος

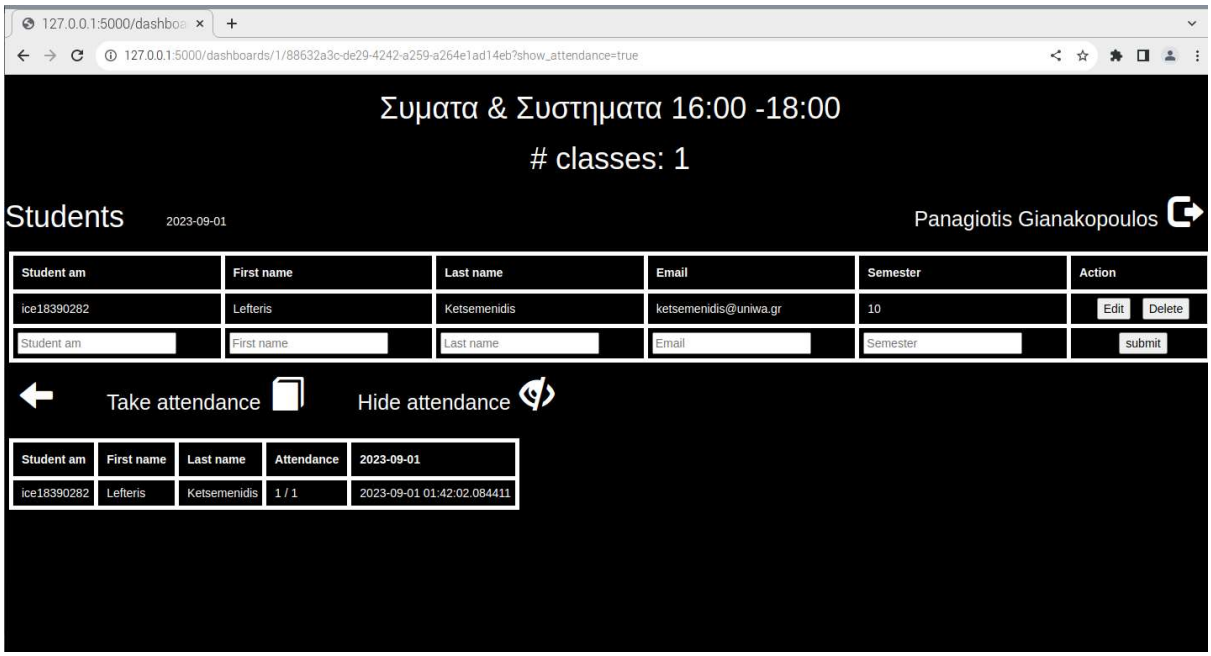


Εικόνα 44: Εισαγωγή νέου χρήστη

Στις παρακάτω δύο εικόνες φαίνεται ο τρόπος καταγραφής των παρουσιών του φοιτητή.

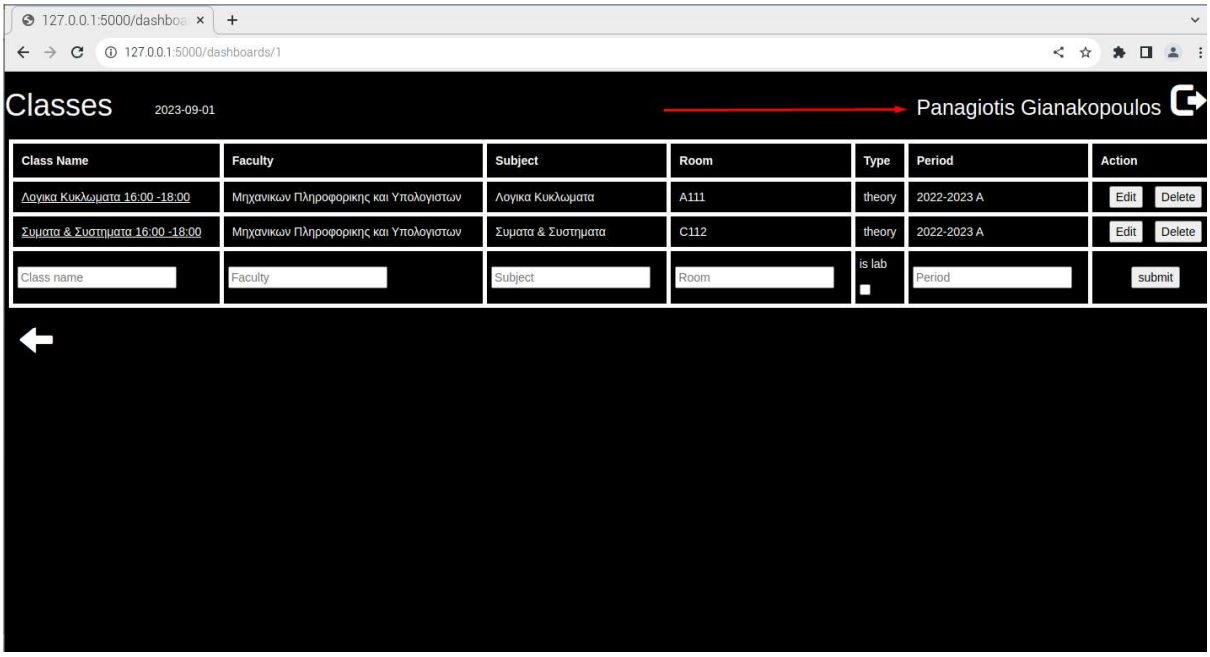


Εικόνα 45: Επιτυχής παρουσία χρήστη



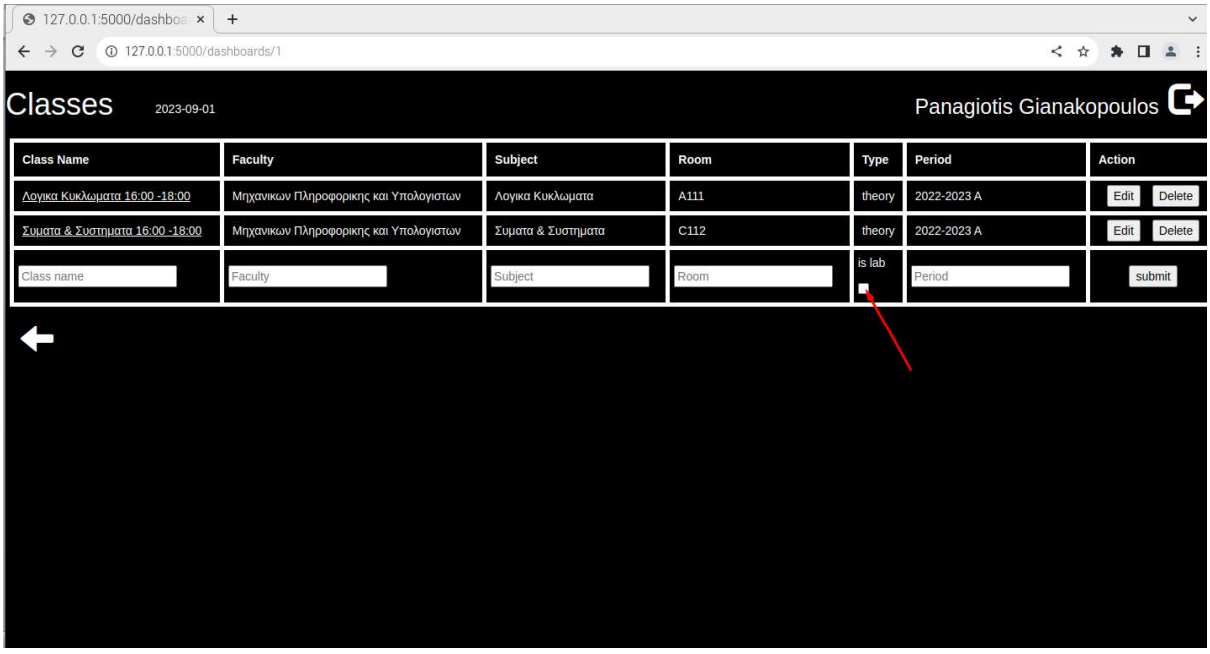
Εικόνα 46: Επιτυχής μέτρηση 1ης παρουσίας φοιτητή στο τμήμα.

Στην παρακάτω εικόνα στο πάνω δεξιά μέρος φαίνεται το όνομα του κάθε καθηγητή που είναι διαχειριστής στο τμήμα του.



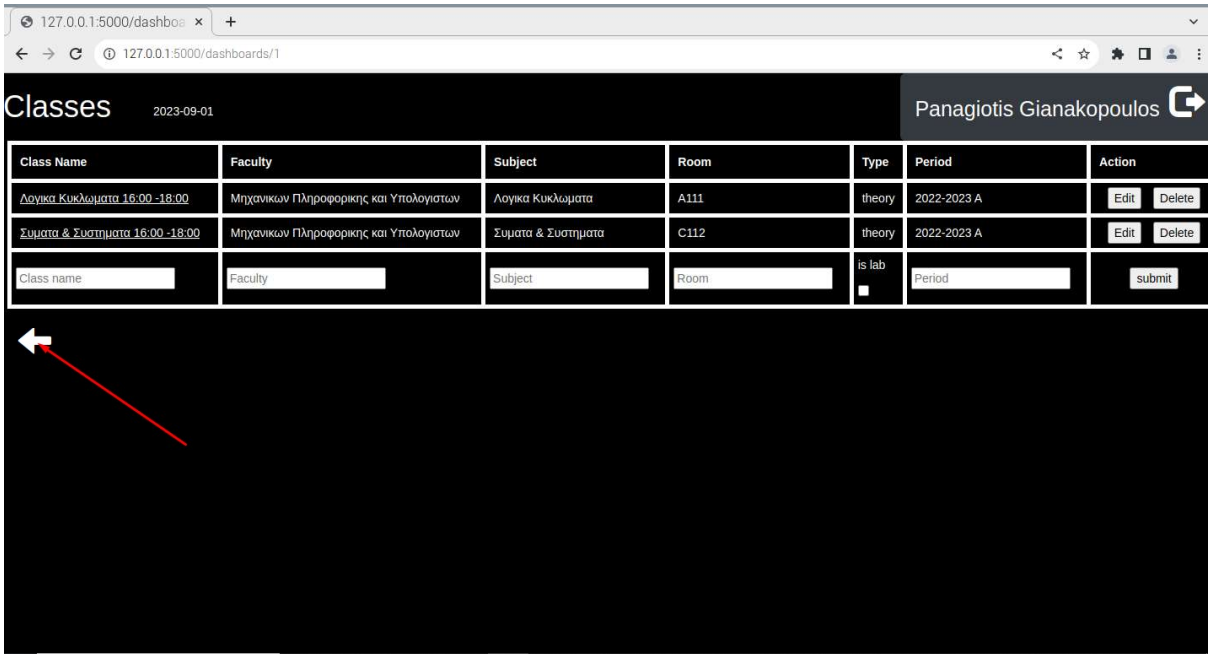
Εικόνα 47:Διαχειριστής τμήματος

Αν τικαρουμε το κουμπί «is lab» θα το πάρει ως εργαστήριο, διαφορετικά θα το πάρει ως θεωρία.



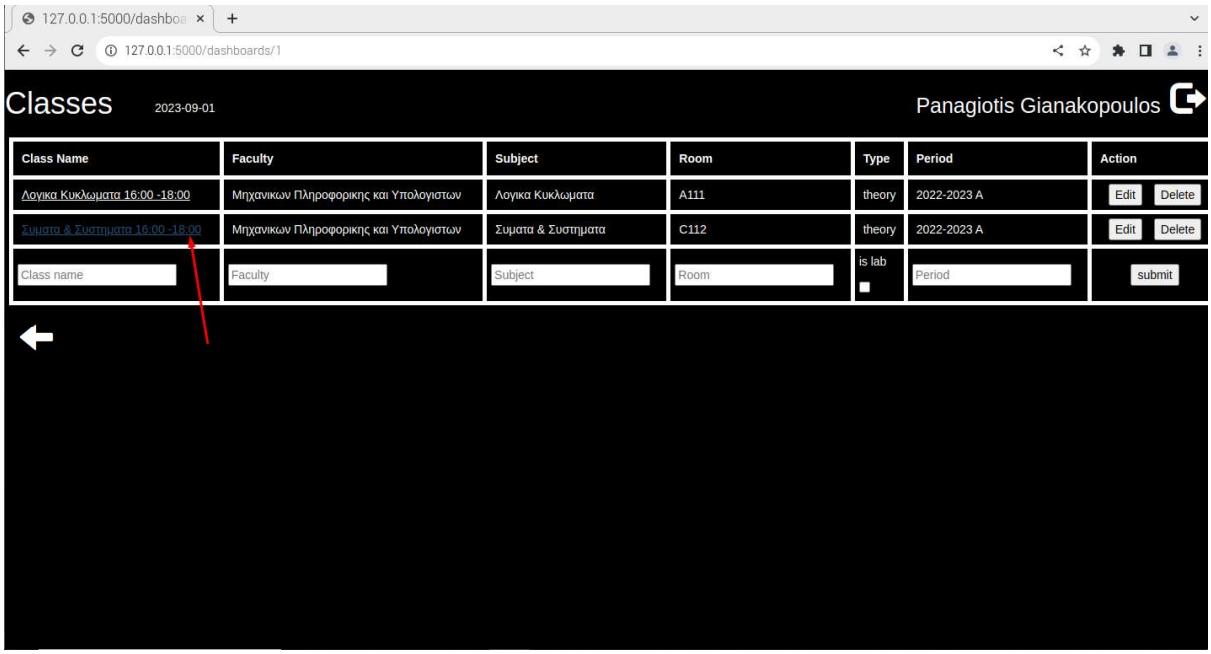
Εικόνα 48: Δήλωση εργαστηρίου

Μπορούμε να πάμε προς τα πίσω επιλέγοντας το παρακάτω βέλος.



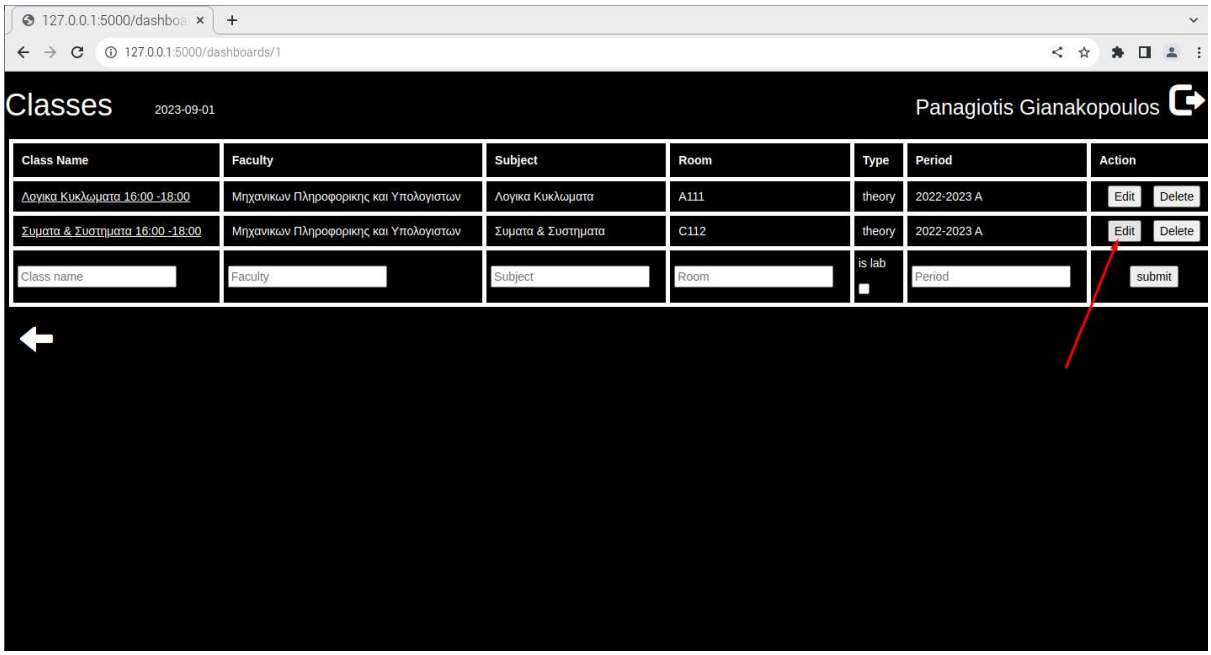
Εικόνα 49: Περιήγηση προς τα πίσω

Πατώντας στο παρακάτω λινκ, μπορείς να μεταβείς και να δεις τους εγγεγραμμένους φοιτητές.



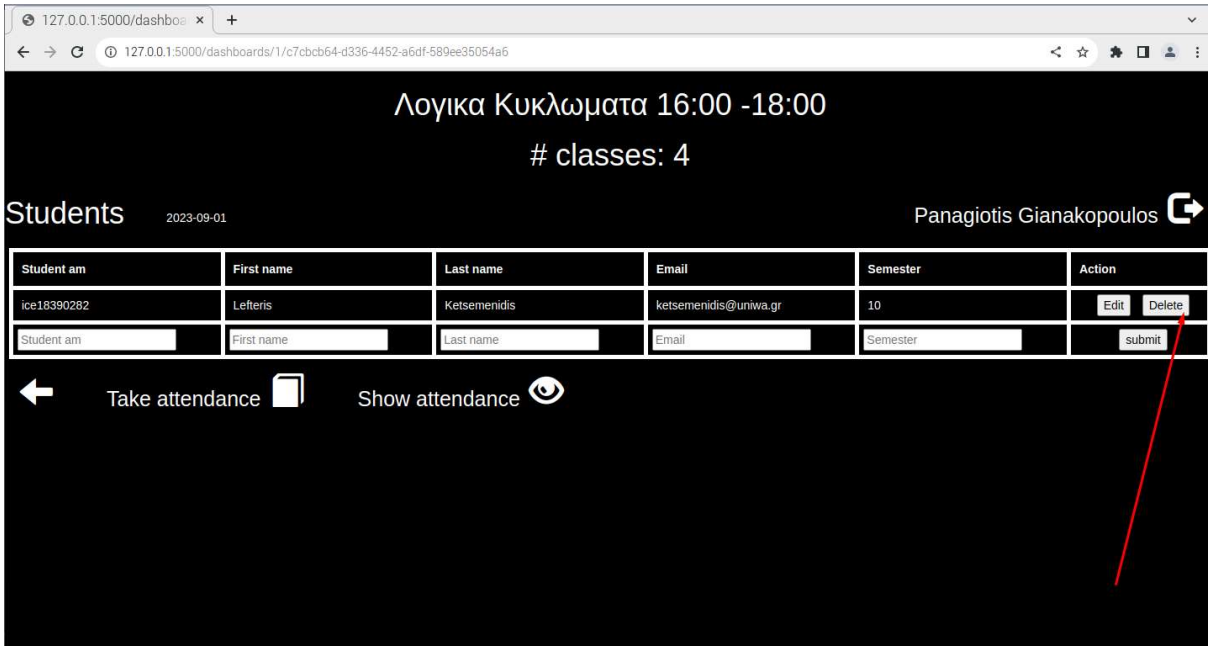
Εικόνα 50: Εγγεγραμμένοι φοιτητές

Δυνατότητα επεξεργασίας του μαθήματος πατώντας το κουμπί “edit”.



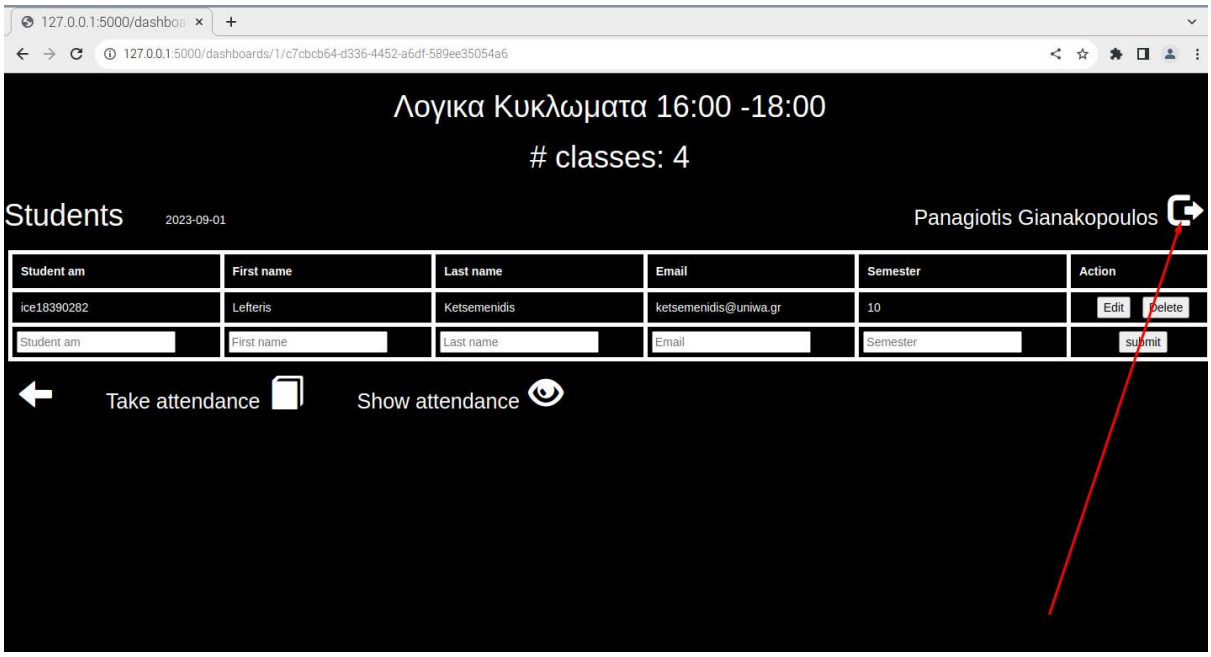
Εικόνα 51: Επεξεργασία του μαθήματος

Πατώντας το κουμπί «delete» μπορούμε να διαγράψουμε το χρήστη από το μάθημα.



Εικόνα 52: Διαγραφή χρήστη

Πατώντας το παρακάτω κουμπί γίνετε η το Logout από την εφαρμογή.



Εικόνα 53: Logout από την εφαρμογή

Βιβλιογραφία

Python Software Foundation. (2023). *www.python.org*. Ανάκτηση από Applications for Python: https://www.python.org/

abyz.me.uk. (2023). Ανάκτηση από The pigpio library: http://abyz.me.uk/rpi/pigpio/

Arduino. (2023). *www.arduino.cc*. Ανάκτηση από https://www.arduino.cc/

Aware, Inc. (2022). *www.aware.com*. Ανάκτηση από What You Need to Know About Voice Biometrics: https://www.aware.com/blog-what-you-need-to-know-about-voice-biometrics/

Aware, Inc. (2023). *www.aware.com*. Ανάκτηση από Mobile Biometric Authentication: Device-centric vs. server-centric architecture: https://www.aware.com/blog-mobile-biometric-authentication-pros-cons-server/

Barra, S., Choo, K.-K. R., Nappi, M., Castiglione, A., Narducci, F., & Ranjan, R. (2018). Biometrics-as-a-Service: Cloud-Based Technology, Systems, and Applications. *IEEE Cloud Computing. vol. 5*, σσ. 33-37.

beagleboard.org. (2023). *www.beagleboard.org*. Ανάκτηση από https://www.beagleboard.org/boards/beaglebone-black

Ben Nuttall Revision. (2021). *gpiozero.readthedocs.io*. Ανάκτηση από https://gpiozero.readthedocs.io/en/stable/

bottlepy.org. (2023). Ανάκτηση από Bottle: Python Web Framework: https://bottlepy.org/docs/dev/

Bukhari, S. U. (2022). *sesamedisk.com*. Ανάκτηση από Authentication in Python: Biometric Fingerprint Matching: https://sesamedisk.com/authentication-in-python-biometric-fingerprint-matching/

Conrad, E., Misenar, S., & Feldman, J. (2017). Chapter 5 - Domain 5: Identity and access management (controlling access and managing identity). Στο *Eleventh Hour CISSP: Study Guide, Third Edition.* Elsevier Inc.

DataCamp, Inc. (2022). *www.datacamp.com*. Ανάκτηση από SQLAlchemy Tutorial With Examples: https://www.datacamp.com/tutorial/sqlalchemy-tutorial-examples?utm_source=google&utm_medium=paid_search&utm_campaignid=19589720818&utm_adgroupid=143216588777&utm_device=c&utm_keyword=&utm_matchtype=&utm_network=g&utm_adpostion=&utm_creative=671350460558&u

developer.mozilla.org. (2023). *developer.mozilla.org*. Ανάκτηση από Web Authentication API: https://developer.mozilla.org/en-US/docs/Web/API/Web_Authentication_API?retiredLocale=el

Django Software Foundation . (2023). *www.djangoproject.com*. Ανάκτηση από https://www.djangoproject.com/

Django Software Foundation. (2023). *docs.djangoproject.com*. Ανάκτηση από https://docs.djangoproject.com/en/3.2/topics/db/models/

Farik, M., & Shawkat, A. (2015). Algorithm To Ensure And Enforce Brute-Force Attack-Resilient Password In Routers. *International Journal of Scientific & Technology Research 4(10)*, σσ. 184-188.

fastapi.tiangolo.com. (2023). Ανάκτηση από FastAPI: https://fastapi.tiangolo.com/

Frąckiewicz , M. (2023). *TS2 Space* . Ανάκτηση από The Challenges of Implementing Biometric Authentication: A Comprehensive Guide: https://ts2.space/en/the-challenges-of-implementing-biometric-authentication-a-comprehensive-guide/

GitHub. (2023). *peewee.readthedocs.io*. Ανάκτηση από https://peewee.readthedocs.io/en/latest/

Hardkernel co. (2019). *www.hardkernel.com*. Ανάκτηση από https://www.hardkernel.com/shop/odroid-xu4-special-price/

Kinzer, K. (2022). *JumpCloud Inc*. Ανάκτηση από Comparing Popular Types of Biometrics: https://jumpcloud.com/blog/comparing-types-of-biometrics

Kruglov, K. (2019). *Biometric data processing and storage system threats.* KASPERSKY LAB.

Lal, N. A., Prasad, S., & Farik, M. (2016). A Review Of Authentication Methods. *International Journal of Scientific & Technology Research 5(11)*, σσ. 246-249.

Leonard, A. T., & Ezeonyi, N. U. (2020). Overview of Technologies and Fingerprint Scanner Used for Biometric Capturing. *Innovation. Vol. 1, No. 1*, σσ. 1-5.

Mainguet, J.-F. (2023). *biometrics.mainguet.org*. Ανάκτηση από Mechanical (pressure) fingerprint sensing: https://biometrics.mainguet.org/types/fingerprint/fingerprint_sensors_physics_mechan.htm

miniOrange Security Software Pvt Ltd. (2023). *blog.miniorange.com*. Ανάκτηση από What is Authentication? Different Types of Authentication:

https://blog.miniorange.com/different-types-of-authentication-methods-for-security/

Molinaro, D. (2022). *www.avast.com*. Ανάκτηση από What Is Biometrics and How Secure Is Biometric Data?: https://www.avast.com/c-what-is-biometric-data

NAMIRIAL GmbH. (2019). *Biometric Signature Verification in Real-Time. Secure transactions through authentication signers with their handwritten signature.* Austria: NAMIRIAL GmbH.

NEC . (2022). *www.nec.co.nz*. Ανάκτηση από How is biometric data stored?: https://www.nec.co.nz/market-leadership/publications-media/how-is-biometric-data-stored/

Oracle. (2023). *www.mysql.com*. Ανάκτηση από https://www.mysql.com/

Oracle. (2023). *www.oracle.com*. Ανάκτηση από Java: https://www.oracle.com/java/

Pagnin , E., & Mitrokotsa, A. (2017). Privacy-Preserving Biometric Authentication: Challenges and Directions. *Hindawi. Security and Communication Networks*.

Pallets. (2010). *flask.palletsprojects.com*. Ανάκτηση από https://flask.palletsprojects.com/en/2.3.x/

Palma, S. (2023). *www.m2sys.com*. Ανάκτηση από How to Choose the Right Software For Fingerprint Scanner: https://www.m2sys.com/blog/biometric-hardware/how-to-choose-the-right-software-for-fingerprint-scanner/

Personal Data Protection Commission Singapore (PDPC). (2022). *Guide on responsible use of biometric data in security applications.* Security Association Singapore (SAS).

Rajarajeswari, S., & Stella, A. (2019). A Review of Authentication and Authorization Methods. *International Journal of Computer Science and Information Technology Research. Vol. 7, Issue 3*, σσ. 78-83.

RaspberryTips. (2023). *raspberrytips.com*. Ανάκτηση από What's the Difference Between a Raspberry Pi and a Computer?: https://raspberrytips.com/difference-raspberry-pi-computer/

Security Industry Association. (2019). *Biometric Technologies: Fingerprints White Paper.* www.securityindustry.org.

SQLite . (2023). *www.sqlite.org*. Ανάκτηση από https://www.sqlite.org/index.html

sqlobject.org. (2023). Ανάκτηση από SQLObject: https://sqlobject.org/

Thales. (2023). *A Comprehensive Guide to Authentication Technologies and Methods*. Ανάκτηση από cpl.thalesgroup.com: https://cpl.thalesgroup.com/resources/access-management/strong-authentication-technologies-survey-white-paper

The PostgreSQL Global Development Group. (2023). *www.postgresql.org*. Ανάκτηση από https://www.postgresql.org/

Tranter, J. (2019). *www.ics.com*. Ανάκτηση από https://www.ics.com/blog/control-raspberry-pi-gpio-pins-python

Wiringpi.com. (2023). *www.wiringpi.com*. Ανάκτηση από http://wiringpi.com/