ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής - Ανάπτυξη Λογισμικού και Τεχνητής Νοημοσύνης»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Χρήση τεχνολογίας NFC για τον έλεγχο φυσικής και λογικής πρόσβασης με χρήση Τεχνητής Νοημοσύνης. Near Field Communication Technology with AI-driven Authentication and Access Control
Ονοματεπώνυμο Φοιτητή	Γεωργαλάς Στυλιανός
Πατρώνυμο	Νικόλαος
Αριθμός Μητρώου	ΜΠΣΠ/ 2208
Επιβλέπων	Σακκόπουλος Ευάγγελος, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης Δεκέμβριος 2024

Τριμελής Εξεταστική Επιτροπή

Ευθύμιος Αλέπης	Δ. Σωτηρόπουλος	Ε. Σακκόπουλος
Καθηγητής	Επίκουρος Καθηγητής	Αναπληρωτής Καθηγητής

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον αναπληρωτή καθηγητή, κύριο Ευάγγελο Σακκόπουλο, που παρά όλες τις δυσκολίες που εμφανίστηκαν, συνέχισε να με καθοδηγεί στην εκπόνηση της παρούσας μεταπτυχιακής διατριβής.

Επιπλέον, θα ήθελα να ευχαριστήσω την οικογένεια μου που παραμένει δίπλα μου και με στηρίζει σε όλα τα βήματα μου.

Περίληψη

Στόχος της παρούσας μεταπτυχιακής διατριβής είναι η ανάπτυξη μιας web πλατφόρμας για τη διαχείριση εισόδου χρηστών σε φυσικές υποδομές, αξιοποιώντας καινοτόμες τεχνολογίες και μοντέλα μηχανικής μάθησης. Η πλατφόρμα θα χρησιμοποιεί NFC tags για την ασφαλή και αποτελεσματική είσοδο χρηστών σε εγκαταστάσεις με υψηλές απαιτήσεις ασφαλείας.

Η πλατφόρμα θα ενσωματώνει αλγορίθμους μηχανικής μάθησης για την ανίχνευση ύποπτων συμπεριφορών και την πρόληψη ανεπιθύμητων περιστατικών, με σκοπό τη βελτίωση της ασφάλειας και της εμπειρίας των χρηστών. Η υλοποίηση της πλατφόρμας αποσκοπεί στην παροχή υψηλής ακρίβειας και αξιοπιστίας, εξασφαλίζοντας ταυτόχρονα την ευκολία χρήσης για τους τελικούς χρήστες.

Προϋποτίθεται η ύπαρξη NFC tags τα οποία θα έχουν κατανεμηθεί σε ήδη εγγεγραμμένους χρήστες.

Abstract

The aim of this master's thesis is the development of a web platform for the management of user access to physical infrastructures, utilizing innovative technologies and machine learning models. The platform will use NFC tags for the safe and efficient entry of users into facilities with high security requirements.

The platform will incorporate machine learning algorithms to detect suspicious behavior and prevent unwanted incidents, with the aim of improving security and user experience. The implementation of the platform aims to provide high accuracy and reliability while ensuring ease of use for end users.

The existence of NFC tags is assumed, which will have been distributed to already registered users.

Περιεχόμενα

Κεφάλαιο 1ο: Αντικείμενο Διατριβής

Η διατριβή αυτή επικεντρώνεται στην ανάπτυξη μιας web πλατφόρμας διαχείρισης εισόδου χρηστών σε φυσικές υποδομές, αξιοποιώντας καινοτόμες τεχνολογίες και μοντέλα μηχανικής μάθησης.

Η πλατφόρμα συνδυάζει τη χρήση NFC tags για την ασφαλή και αποτελεσματική είσοδο χρηστών σε εγκαταστάσεις, όπως κτίρια γραφείων, πανεπιστημιακά ιδρύματα και άλλους χώρους με υψηλές απαιτήσεις ασφαλείας.

Επιπλέον, ενσωματώνει αλγορίθμους μηχανικής μάθησης για την ανίχνευση ύποπτων συμπεριφορών και την πρόληψη ανεπιθύμητων περιστατικών, βελτιώνοντας έτσι την ασφάλεια και την εμπειρία των χρηστών. Η πλατφόρμα σχεδιάζεται ώστε να προσφέρει υψηλή ακρίβεια και αξιοπιστία, ενώ παράλληλα διασφαλίζει την ευκολία χρήσης για τους τελικούς χρήστες, δημιουργώντας ένα ολοκληρωμένο και καινοτόμο σύστημα διαχείρισης εισόδου.

Κεφάλαιο 2ο: Ανάλυση Απαιτήσεων

Η Ανάλυση Απαιτήσεων είναι η διαδικασία καθορισμού, τεκμηρίωσης και διαχείρισης αναγκών και προσδοκιών για ένα σύστημα. Περιλαμβάνει την καταγραφή των λειτουργικών και μη λειτουργικών απαιτήσεων, τη διευκρίνιση των επιχειρησιακών στόχων και των περιορισμών, καθώς και τη διασφάλιση ότι οι απαιτήσεις είναι σαφείς, πλήρεις, συνεπείς και εφικτές.

Η ανάλυση απαιτήσεων χωρίζεται σε διάφορες κατηγορίες, οι οποίες βοηθούν στην οργάνωση και τη διαχείριση τους όπου τρεις απο τις κυριότερες είναι :

1. Απαιτήσεις Χρήστη
2. Απαιτήσεις Συστήματος
3. Απαιτήσεις Λογισμικού

Κεφάλαιο 2.1: Απαιτήσεις Χρήστη

Οι απαιτήσεις χρήστη αφορούν τους διαχειριστές (admins) που θα χρησιμοποιούν την πλατφόρμα για τη διαχείριση της εισόδου χρηστών σε φυσικές υποδομές:

- **Ασφαλής Πρόσβαση**: Η εφαρμογή θα προσφέρει μια σελίδα σύνδεσης (login page) και λειτουργία αποσύνδεσης (logout), διασφαλίζοντας την ασφαλή πρόσβαση των διαχειριστών στην πλατφόρμα.
- **Εγγραφή και Διαχείριση NFC Tags**: Οι διαχειριστές πρέπει να μπορούν να εγγράφουν και να διαχειρίζονται NFC tags για τους χρήστες μέσω ενός NFC API όπου θα είναι προσβάσιμο μόνο μέσω Android mobile.
- **Διαχείριση Προφίλ Χρηστών**: Οι διαχειριστές πρέπει να έχουν τη δυνατότητα να δημιουργούν, να επεξεργάζονται και να διαγράφουν προφίλ χρηστών.
- **Προβολή και Διαχείριση Δικαιωμάτων Εισόδου**: Οι διαχειριστές πρέπει να μπορούν να αναθέτουν και να διαχειρίζονται δικαιώματα εισόδου χρηστών σε διάφορες εγκαταστάσεις.
- **Παρακολούθηση και Αναφορές**: Οι διαχειριστές πρέπει να έχουν πρόσβαση σε αναφορές και δεδομένα σχετικά με τη χρήση των NFC tags και την είσοδο των χρηστών.
- **Dashboard με Γραφήματα**: Οι διαχειριστές πρέπει να έχουν στη διάθεσή τους διαδραστικά dashboards με γραφήματα που απεικονίζουν τη ροή των χρηστών και άλλα κρίσιμα δεδομένα.

Κεφάλαιο 2.2: Απαιτήσεις Συστήματος

Οι απαιτήσεις συστήματος επικεντρώνονται στην ασφάλεια, την αξιοπιστία και την απόδοση της πλατφόρμας:

- **Ασφάλεια Χρηστών**: Η πλατφόρμα πρέπει να ενσωματώνει μία τεχνολογία ασφαλείας για την προστασία των δεδομένων και την ασφάλεια των χρηστών αυτό επιτυγχάνεται με την βοήθεια του Spring Security. [2]
- **Ασφάλεια Δεδομένων**: Χρήση κρυπτογράφησης και ασφαλών πρωτοκόλλων επικοινωνίας για τη διασφάλιση της ακεραιότητας και της εμπιστευτικότητας των δεδομένων.
- **Αξιοπιστία και Διαθεσιμότητα**: Η πλατφόρμα πρέπει να εξασφαλίζει υψηλή διαθεσιμότητα και αξιοπιστία, ελαχιστοποιώντας τον χρόνο διακοπής λειτουργίας.
- **Υποστήριξη NFC** : Το σύστημα πρέπει να υποστηρίζει τη διαχείριση των NFC tags μέσω ενός NFC API, ευκόλως προσβάσιμο μέσω mobile, επιτρέποντας στους διαχειριστές να καταχωρούν και να διαχειρίζονται τα tags.
- **Επεκτασιμότητα**: Η πλατφόρμα πρέπει να είναι επεκτάσιμη για να υποστηρίζει αυξανόμενο αριθμό χρηστών και εγκαταστάσεων.
- **Διαδραστικά Dashboards**: Το σύστημα πρέπει να υποστηρίζει τη δημιουργία και προβολή διαδραστικών dashboards με γραφήματα για την ανάλυση της ροής των χρηστών και άλλων δεδομένων.

Κεφάλαιο 2.3: Απαιτήσεις Λογισμικού

Οι απαιτήσεις λογισμικού επικεντρώνονται στην ανάπτυξη και λειτουργικότητα της πλατφόρμας:

- **Ενσωμάτωση Spring Security**: Χρήση του Spring Security για την υλοποίηση μηχανισμών ταυτοποίησης και εξουσιοδότησης, καθώς και για την προστασία των δεδομένων.[2]
- **Διαχείριση Πρόσβασης**: Το σύστημα θα χρησιμοποιεί JWT (JSON Web Tokens), τα οποία θα κρυπτογραφούνται με αλγόριθμο Base64, για την ασφαλή ταυτοποίηση και διαχείριση των χρηστών. Τα tokens θα εξασφαλίζουν την προστασία των δεδομένων και θα επιτρέπουν την πρόσβαση μόνο σε εξουσιοδοτημένους χρήστες.
- **Ασφάλεια Δεδομένων**: Χρήση ασφαλών πρωτοκόλλων επικοινωνίας (https) που βασίζεται στο TLS (Transport Layer Security), για τη διασφάλιση της ακεραιότητας και της εμπιστευτικότητας των δεδομένων. Η ενσωμάτωση πιστοποιητικών SSL/TLS εξασφαλίζει κρυπτογράφηση της επικοινωνίας μεταξύ του διακομιστή και του χρήστη, αποτρέποντας πιθανές υποκλοπές δεδομένων ή μη εξουσιοδοτημένη πρόσβαση.
- **Χρήση Μηχανικής Μάθησης**: Ενσωμάτωση μοντέλου μηχανικής μάθησης Isolation Forest για την ανίχνευση ανωμαλιών και την πρόβλεψη πιθανών απειλών ασφαλείας.
- **Διασύνδεση Χρήστη (UI/UX)**: Ανάπτυξη ενός εύχρηστου και διαισθητικού περιβάλλοντος χρήστη για τους διαχειριστές με τη χρήση του Vue.js και του framework Quasar, το οποίο προσφέρει σαφείς λειτουργίες και εργαλεία διαχείρισης..
- **Διαχείριση Προφίλ και Δικαιωμάτων**: Υλοποίηση λειτουργιών για τη δημιουργία και τη διαχείριση προφίλ χρηστών και δικαιωμάτων εισόδου μέσω μιας κεντρικής διαχείρισης.
- **Ειδοποιήσεις σε Πραγματικό Χρόνο**: Παροχή ειδοποιήσεων σε πραγματικό χρόνο για ασυνήθιστες ή ύποπτες δραστηριότητες, χρησιμοποιώντας τα δεδομένα που συλλέγονται από τα NFC tags και τα μοντέλα μηχανικής μάθησης.
- **Διαδραστικά Dashboards**: Ανάπτυξη διαδραστικών dashboards με τη χρήση της βιβλιοθηκης ECharts για την απεικόνιση γραφημάτων που παρουσιάζουν τη ροή των χρηστών, τις εισόδους/εξόδους, και άλλες σημαντικές πληροφορίες. Αυτό επιτρέπει

στους διαχειριστές να έχουν μια ολοκληρωμένη και οπτικά κατανοητή εικόνα της κατάστασης.[5]

Η ανάλυση αυτών των απαιτήσεων θα συμβάλει στη δημιουργία μιας ολοκληρωμένης και αποτελεσματικής πλατφόρμας.

Κεφάλαιο 3ο: Τεχνολογικά Εργαλεία και Τεχνικές

Στον τομέα της σύγχρονης ανάπτυξης λογισμικού, η επιλογή των βασικών τεχνολογιών θέτει τα θεμέλια για την επιτυχία ενός έργου. Εδώ, εξετάζουμε βασικά εργαλεία και πλαίσια που διευκολύνουν την αποδοτική ανάπτυξη και διαχείριση της εφαρμογής.

Κεφάλαιο 3.1: Quasar Framework : Unleashing the Potential of Vue 3.js

Το Quasar Framework ξεχωρίζει ως ένα ισχυρό εργαλείο ανάπτυξης ιστοσελίδων, προσφέροντας στους προγραμματιστές μία πλήρη εργαλειοθήκη για τη δημιουργία υψηλής ποιότητας και αποκριτικότητας εφαρμογές.

Αξιοποιώντας τις δυνατότητες της Vue 3, το Quasar απλοποιεί τη διαδικασία ανάπτυξης, διευκολύνοντας την ταχεία προτυποποίηση και την άνετη ανάπτυξη σε διαφορετικές πλατφόρμες.

Ένα από τα κύρια πλεονεκτήματα του Quasar βρίσκεται στην εκτενή βιβλιοθήκη προκατασκευασμένων στοιχείων και εργαλείων, τα οποία διευκολύνουν την ανάπτυξη και εξασφαλίζουν την αποτελεσματικότητα στον σχεδιασμό και τη λειτουργικότητα. Επιπλέον,το Quasar προσφέρει ενσωματωμένη υποστήριξη για responsive design αλλά και SSR (Server-Side Rendering) βελτιώνοντας την εμπειρία του χρήστη παρέχοντας γρήγορη φόρτωση, δίνοντας την δυνατότητα δημιουργίας οπτικά ελκυστικών εφαρμογών που προσαρμόζονται σε διαφορετικές συσκευές και μεγέθη οθονών.

Με το Quasar, οι προγραμματιστές μπορούν να εκμεταλλευτούν τις σύγχρονες τεχνολογίες του web για τη δημιουργία δυναμικών, πλούσιων σε λειτουργίες εφαρμογών, ενώ απολαμβάνουν την ευελιξία που παρέχει το οικοσύστημα της Vue.js.[6]

Κεφάλαιο 3.2: FastAPI: Effortless ML Model Deployment

Με την εξαιρετική του ταχύτητα και τη συνοχή του, το FastAPI ανοίγει νέους ορίζοντες στον κόσμο της ανάπτυξης λογισμικού, επιτρέποντας στους προγραμματιστές να επικεντρωθούν στη δημιουργικότητά τους και την καινοτομία, αντί να ανησυχούν για τις λεπτομέρειες της υλοποίησης.

Από την αρχική ανάπτυξη έως την παραγωγική εκτέλεση, το FastAPI είναι η επιλογή που επιτρέπει στις ομάδες ανάπτυξης να προχωρήσουν γρήγορα και με βεβαιότητα.

Έτσι, το FastAPI αναδεικνύεται ως ένα επικρατέστερο εργαλείο για την επίτευξη ολοκληρωμένων λύσεων μηχανικής μάθησης που ενσωματώνουν ταχύτητα, αξιοπιστία και ευελιξία. [8]

Κεφάλαιο 3.3: Spring Boot: Streamlined approach to Java-Based Microservices

Το Spring Boot αποτελεί μία τεράστια αλλαγή στο οικοσύστημα της Java, προσφέροντας στους προγραμματιστές μια απλοποιημένη προσέγγιση για την κατασκευή Micro-Services. Ξεκινώντας από τον πυρήνα του, το Spring Boot υιοθετεί τη φιλοσοφία convention over configuration, μειώνοντας έτσι την ανάγκη για verbose boilerplate code και απλοποιώντας σημαντικά τις διεργασίες. Χρησιμοποιώντας embedded servers, το Spring Boot εξαλείφει την πολυπλοκότητα εγκατάστασης εξωτερικού server, επιτρέποντας στους προγραμματιστές να επικεντρωθούν στη λογική της εφαρμογής παρά στις ανησυχίες υποδομής.

Επιπλέον, η δυνατότητα auto-configuration του Spring Boot διαμορφώνει τα components της εφαρμογής με βάση το classpath, ελαχιστοποιώντας την χειροκίνητη παραμετροποίηση του project. Αυτή η προσέγγιση επιταχύνει σημαντικά τον ρυθμό ανάπτυξης, επιτρέποντας στις ομάδες να πρωτοτυπήσουν και να εφαρμόσουν γρήγορα τις αρχιτεκτονικές microservices. [I] [II] [1]

Κεφάλαιο 3.4: PostgreSQL: A Powerful Database

Η αποτελεσματική διαχείριση των βάσεων δεδομένων και των schema είναι θεμελιώδους σημασίας για τη διασφάλιση της ακεραιότητας, της συνέπειας και της απόδοσης των δεδομένων σε σύγχρονες εφαρμογές.

Η PostgreSQL αποτελεί μία ισχυρή και πλούσια σε δυνατότητες Relational Database Management System, εμπιστευμένη από προγραμματιστές και οργανισμούς παγκοσμίως. Γνωστή για τη συμμόρφωσή της με τα SQL standards και τις ιδιότητες ACID (Atomicity, Consistency, Isolation, Durability), η PostgreSQL είναι μια αξιόπιστη βάση για την αποθήκευση και την αναζήτηση δομημένων δεδομένων.

Ένα από τα σημαντικότερα πλεονεκτήματα της PostgreSQL βρίσκεται στην επεκτασιμότητα της, υποστηρίζοντας ένα ευρύ φάσμα data types, indexing mechanisms, και advanced SQL features. Η PostgreSQL προσφέρει βέλτιστη απόδοση και επεκτασιμότητα χειρίζοντας πολύπλοκα transactions, πραγματοποιώντας full-text search και διαχειρίζοντας geospatial data.

Επιπλέον, το ενεργό open-source community του εξασφαλίζει συνεχή ανάπτυξη και υποστήριξη, συμβάλλει στη συνεχιζόμενη ανάπτυξη σύγχρονων εφαρμογών. [III] [3]

Κεφάλαιο 3.5: Docker: Containerization for Seamless Deployment

Το Docker έφερε επανάσταση στην ανάπτυξη εφαρμογών μέσω του Containerization, προσφέροντας ένα προτυποποιημένο και αποτελεσματικό τρόπο διανομής και εκτέλεσης εφαρμογών σε διαφορετικά περιβάλλοντα.

Στον πυρήνα του, τα εκάστοτε Docker containers ενθυλακώνουν τον κώδικα της εφαρμογής, τα dependencies και το περιβάλλον εκτέλεσης, επιτρέποντας να υπάρχει consistent και reproducible ανάπτυξη κώδικα από το development έως την παραγωγή.

Ένα κύριο πλεονέκτημα του Docker είναι το πόσο ελαφρύ είναι, επιτρέποντας στους προγραμματιστές να δημιουργούν και να αναπτύσσουν γρήγορα και αποτελεσματικά containers, χρησιμοποιώντας container images και αρχεία Dockerfile, οι προγραμματιστές μπορούν να ορίσουν το περιβάλλον εκτέλεσης των εφαρμογών και τα dependencies δηλωτικά χωρίς να χρειάζεται καμία εγκατάσταση, εξασφαλίζοντας συνέπεια και αναπαραγωγιμότητα σε διαφορετικά περιβάλλοντα.

Επιπλέον, το οικοσύστημα εργαλείων και υπηρεσιών του Docker, συμπεριλαμβανομένου του Docker Compose για οργάνωση πολλών docker container απλοποιεί την ανάπτυξη και την μεταφορά των εφαρμογών σε containers. Είτε αν θέλουμε να αναπτύξουμε αρχιτεκτονικές micro

service είτε μονολιθικές εφαρμογές, το Docker παρέχει μια ευέλικτη και κλιμακούμενη λύση για την επιτάχυνση της διαδικασίας ανάπτυξης. **[**IV] [4]

Κεφάλαιο 3.6: Nginx: Reverse Proxy and Deployment Strategies Simplified

Το Nginx αποτελεί ένα από τα πιο δημοφιλή και ευέλικτα εργαλεία για τη διαχείριση και την ανάπτυξη web εφαρμογών. Με τη λειτουργία του ως reverse proxy, επιτρέπει την κατεύθυνση των αιτημάτων (request) στους κατάλληλους servers, προσφέροντας βελτιωμένη ταχύτητα, ασφάλεια και διαθεσιμότητα.
Παράλληλα, διευκολύνει την ανάπτυξη εφαρμογών με δυνατότητες όπως load balancing, SSL, και caching, κάνοντας τις εφαρμογές πιο αποδοτικές και ανθεκτικές. Εδώ θα εστιάσουμει στις βασικές στρατηγικές χρήσης του Nginx για την σύνδεση του frontend με το backend και την εξασφάλιση υψηλής απόδοσης σε περιβάλλοντα παραγωγής. [9]

Κεφάλαιο 4: Μοντέλο περιπτώσεων χρήσης (Use case model)

Το Μοντέλο Περιπτώσεων Χρήσης είναι μια παρουσίαση ανάλυσης επιχειρηματικών διαδικασιών που περιγράφει τα βήματα των αλληλεπιδράσεων μεταξύ ενός χρήστη (actor) και ενός συστήματος .
Το μοντέλο αναλύει αλληλεπιδράσεις και καθορίζει τις προσδοκίες για το πώς ο χρήστης θα αλληλεπιδράσει με το σύστημα. Αποτελείται από δύο βασικά στοιχεία: το διάγραμμα περιπτώσεων χρήσης (use case diagram), το οποίο είναι μια γραφική αναπαράσταση που δείχνει ποιοι χρήστες μπορούν να εκτελούν ποιες περιπτώσεις χρήσης, και την περιγραφη (use case narrative) , η οποία είναι μια αναλυτική περιγραφή των βημάτων και των διαλόγων που ακολουθούνται κατά την αλληλεπίδραση του actor με το σύστημα.

Κεφάλαιο 4.1: Διαγράμματα περιπτώσεων χρήσης (Use case Diagram)

Εικόνα 1: Use Case Diagram

Κεφάλαιο 4.2: Περιγραφή περιπτώσεων χρήσης (Use case Narrative)

Ακολουθεί ένα παράδειγμα Περιγραφής Περίπτωσης Χρήσης για λειτουργίες Admin user:
Login: Ο χρήστης (Admin) συνδέεται στο σύστημα για να αποκτήσει πρόσβαση στην κεντρική διαχείριση.
Βασικό Σενάριο (Main Flow):

- Ο διαχειριστής ανοίγει τη σελίδα σύνδεσης και εισάγει το όνομα χρήστη και τον κωδικό του. Το σύστημα ελέγχει την εγκυρότητα των στοιχείων αν τα στοιχεία είναι σωστά, ο διαχειριστής ανακατευθύνεται στο Πίνακα Ελέγχου (Dashboard). Ο διαχειριστής αποκτά πρόσβαση σε όλες τις διαθέσιμες λειτουργίες του συστήματος.

Εναλλακτικά Σενάρια (Alternative Flows):

- Αν τα στοιχεία είναι λανθασμένα το σύστημα εμφανίζει μήνυμα σφάλματος και ζητά από τον διαχειριστή να εισάγει ξανά τα σωστά στοιχεία ο διαχειριστής προσπαθεί ξανά να συνδεθεί.

View Dashboard: Ο πίνακας ελέγχου του συστήματος επιτρέπει στον διαχειριστή να παρακολουθήσει τις κύριες πληροφορίες όπως Γράφημα, Στατιστικά Χρηστών, κ.α.
Βασικό Σενάριο (Main Flow):

- Το σύστημα εμφανίζει τα στατιστικά και τα γραφήματα που σχετίζονται με τις ενέργειες των χρηστών ο διαχειριστής παρακολουθεί τις πληροφορίες που εμφανίζονται στην αρχική σελίδα του πίνακα ελέγχου.

Εναλλακτικά Σενάρια (Alternative Flows):

- Αν ο πίνακας ελέγχου περιέχει ελλιπή δεδομένα το σύστημα ειδοποιεί τον διαχειριστή ότι κάποια δεδομένα δεν είναι διαθέσιμα ή ότι υπάρχει κάποιο πρόβλημα στην επικοινωνία με τον server.

Manage Users: Ο διαχειριστής διαχειρίζεται τους χρήστες, προσθέτοντας, τροποποιώντας ή διαγράφοντας χρήστες.
Βασικό Σενάριο (Main Flow):

- Ο διαχειριστής εισέρχεται στην ενότητα διαχείρισης χρηστών μεσω του πίνακα ελέγχου (dashboard). Ο διαχειριστής προσθέτει, τροποποιεί ή διαγράφει χρήστες, όπως απαιτείται. Το σύστημα ενημερώνει την κατάσταση των χρηστών και τις αντίστοιχες πληροφορίες.

Εναλλακτικά Σενάρια (Alternative Flows):

- Αν ο χρήστης που προσπαθεί να καταργήσει είναι admin role . Το σύστημα εμφανίζει μήνυμα σφάλματος, ενημερώνοντας τον διαχειριστή ότι ο χρήστης είναι ρόλου διαχειριστή .
- Αν ο διαχειριστής προσπαθήσει να προσθέσει χρήστη με διπλό όνομα χρήστη ή email Το σύστημα εμφανίζει μήνυμα σφάλματος ότι το όνομα χρήστη είναι ήδη καταχωρημένο.Ο διαχειριστής εισάγει νέο όνομα χρήστη ή email και επανυποβάλλει τη διαδικασία.

Manage NFC Tags: Ο διαχειριστής διαχειρίζεται τα NFC tags, προσθέτοντας, επεξεργάζοντας ή διαγράφοντας αυτά από το σύστημα.
Βασικό Σενάριο (Main Flow):

- Ο διαχειριστής εισέρχεται στην ενότητα διαχείρισης NFC tags. Ο διαχειριστής επιλέγει να προσθέσει, τροποποιήσει ή διαγράψει NFC tags απο έναν χρήστη. Το σύστημα ενημερώνει τη βάση δεδομένων με τις νέες πληροφορίες για τα NFC tags.

Εναλλακτικά Σενάρια (Alternative Flows):

- Αν το το σύστημα δεν μπορεί να διαβάσει ή να αναγνωρίσει το NFC Tag ειδοποιεί τον διαχειριστή ότι το NFC tag δεν μπορεί να διαβαστεί. Ο διαχειριστής μπορεί να επιλέξει να προχωρήσει σε καταχώρηση μέσω άλλου browser ή άλλης συσκευής μέσω σάρωσης QR code.
- Αν ο χρήστης έχει NFC tag ήδη καταχωρημένο. Το σύστημα ειδοποιεί τον διαχειριστή ότι ο χρήστης χρησιμοποιεί ήδη tag. Ο διαχειριστής μπορεί να επιλέξει να αποδεσμεύσει το tag.

Manage Profile: Ο διαχειριστής διαχειρίζεται το προφίλ του τροποποιώντας πληροφορίες ή ρυθμίσεις.
Βασικό Σενάριο (Main Flow):

- Ο διαχειριστής εισέρχεται στην ενότητα διαχείρισης προφίλ και τροποποιεί τις πληροφορίες ή τις ρυθμίσεις του προφίλ του (π.χ. όνομα, e-mail). Το σύστημα ενημερώνει το προφίλ με τις νέες αλλαγές.

Εναλλακτικά Σενάρια (Alternative Flows):

- Αν ο διαχειριστής προσπαθήσει να καταχωρήσει αδύνατη τιμή για το πεδίο (π.χ., λάθος μορφή e-mail). Το σύστημα εμφανίζει μήνυμα σφάλματος και ζητά από τον διαχειριστή να εισάγει σωστή τιμή.Ο διαχειριστής διορθώνει την πληροφορία και υποβάλλει ξανά την αλλαγή.

Manage Infrastructure: Ο διαχειριστής διαχειρίζεται την υποδομή του συστήματος, όπως τα κτίρια και τις πόρτες, προσθέτοντας, τροποποιώντας ή διαγράφοντας τα.

Βασικό Σενάριο (Main Flow):

- Ο διαχειριστής εισέρχεται στην ενότητα διαχείρισης υποδομής.Ο διαχειριστής επιλέγει να προσθέσει, τροποποιήσει ή διαγράψει κτίρια ή πόρτες.Το σύστημα ενημερώνει τη βάση δεδομένων με τις νέες πληροφορίες για τα κτίρια και τις πόρτες

Εναλλακτικά Σενάρια (Alternative Flows):

- Αν ο διαχειριστής προσπαθήσει να διαγράψει κτίριο που έχει συσχετισμένες πόρτες. Το σύστημα εμφανίζει προειδοποίηση ότι δεν μπορεί να διαγραφεί το κτίριο εάν υπάρχουν πόρτες που σχετίζονται με αυτό. Ο διαχειριστής διορθώνει την πληροφορία και υποβάλλει ξανά την αλλαγή.

Train ML Model: Ο διαχειριστής έχει τη δυνατότητα να εκπαιδεύσει το μοντέλο μηχανικής μάθησης (ML) για να βελτιώσει τις προβλέψεις ή τις αναλύσεις του συστήματος, βασισμένο σε υπάρχοντα δεδομένα χρηστών.

Βασικό Σενάριο (Main Flow):

- Ο διαχειριστής επιλέγει την επιλογή "Εκπαίδευση Μοντέλου" από το μενού του συστήματος. Επιλέγει το σύνολο δεδομένων που θα χρησιμοποιηθεί για την εκπαίδευση με ημερολογιακό κριτήριο. Το σύστημα εκπαιδεύει το μοντέλο χρησιμοποιώντας τα δεδομένα εκπαίδευσης και αξιολογεί την απόδοση του μοντέλου με τα δεδομένα δοκιμής. Το εκπαιδευμένο μοντέλο αποθηκεύεται στο σύστημα και είναι διαθέσιμο για χρήση στις μελλοντικές προβλέψεις.

Εναλλακτικά Σενάρια (Alternative Flows):

- Αν η εκπαίδευση του μοντέλου αποτύχει λόγω υπερφόρτωσης ή παραμέτρων που δεν είναι σωστές. Το σύστημα εμφανίζει προειδοποίηση ή μήνυμα σφάλματος ότι η εκπαίδευση απέτυχε και προτείνει λύσεις για τη βελτίωση των παραμέτρων. Ο διαχειριστής μπορεί να τροποποιήσει τις παραμέτρους και να επαναλάβει την εκπαίδευση

Κεφάλαιο 5: Αρχιτεκτονική Εφαρμογής

Η εφαρμογή βασίζεται σε μια πολυεπίπεδη αρχιτεκτονική που αποτελείται από τα εξής βασικά στοιχεία:

- **Frontend (Vue Quasar Framework)**

  Το frontend έχει αναπτυχθεί με το Quasar Framework, το οποίο βασίζεται στο Vue.js. Είναι υπεύθυνο για τη διαχείριση του UI/UX και την επικοινωνία με τα backend services μέσω API κλήσεων.

- **Backend (Java Spring Boot)**

  Το backend υλοποιείται με Spring Boot, το οποίο παρέχει RESTful APIs για την επεξεργασία αιτημάτων από το frontend. Είναι υπεύθυνο για τη λογική της εφαρμογής και την αλληλεπίδραση με τη βάση δεδομένων και το μοντέλο μηχανικής μάθησης.

- **Μοντέλο Μηχανικής Μάθησης (Python FastAPI)**

  Το ML μοντέλο έχει αναπτυχθεί χρησιμοποιώντας FastAPI και παρέχει endpoints για την εκτέλεση προβλέψεων και εκμάθησης του .

- **Nginx (Reverse Proxy)**

  Το Nginx λειτουργεί ως reverse proxy, όπου διαχειρίζεται την ανακατεύθυνση των αιτημάτων μεταξύ frontend και backend. Εξασφαλίζει καλύτερη απόδοση μέσω τεχνικών SSL offloading και caching και ασφάλεια.

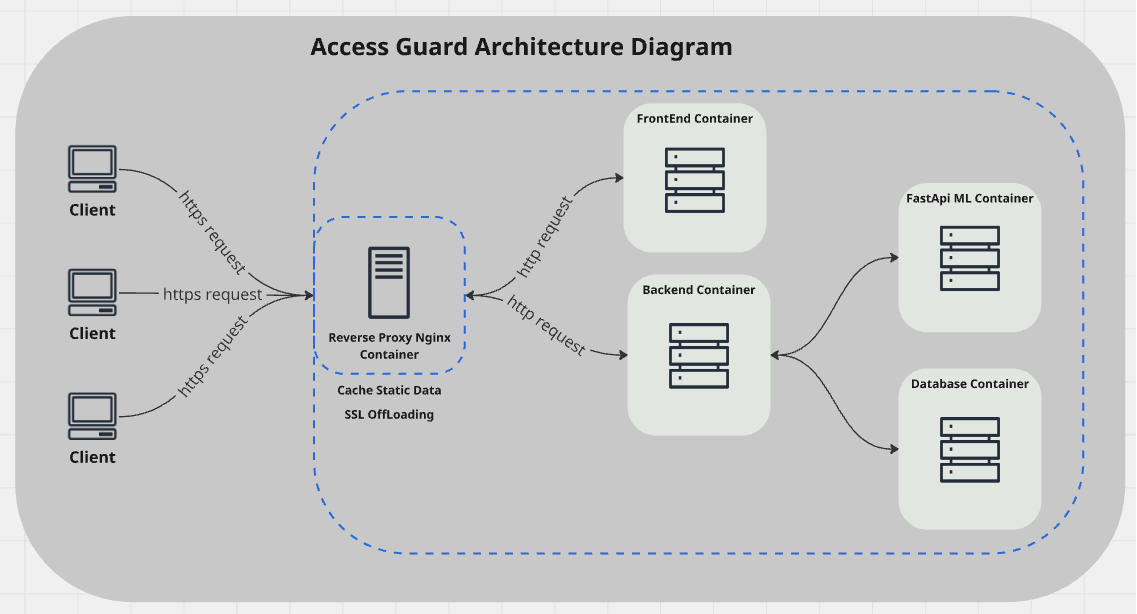
- **Βάση Δεδομένων(PostgreSQL)**

  Χρησιμοποιούμε μια σχεσιακή βάση δεδομένων που συνδέεται με το backend για αποθήκευση δεδομένων της εφαρμογής.

- **Επικοινωνία**

  Τα συστήματα επικοινωνούν μεταξύ τους μέσω RESTful APIs και HTTP πρωτοκόλλου, ενώ η ασφαλής επικοινωνία εξασφαλίζεται με τη χρήση SSL/TLS στο Nginx.

Κεφάλαιο 5.1: Architecture Diagram



Εικόνα 2: Architecture Diagram

Κεφάλαιο 6: Ανάπτυξη της Εφαρμογής

Η ανάπτυξη της πλατφόρμας περιλαμβάνει τη διαμόρφωση τόσο του backend όσο και του frontend, καθώς και την ενσωμάτωση ενός Machine Learning μοντέλου για την ανίχνευση ανωμαλιών. Η πλατφόρμα θα χρησιμοποιεί Docker containers για να εξασφαλίσει ένα σταθερό και αναπαραγώγιμο περιβάλλον ανάπτυξης και παραγωγής.

Κεφάλαιο 6.1: Ανάπτυξη και Παραμετροποίηση Backend (FastAPI και Spring Boot)

Η υλοποίηση του backend της πλατφόρμας μας βασίζεται στο FastAPI και Spring Boot, που επιλέχθηκαν για την ταχύτητα και την ευελιξία τους.[8]

Το FastAPI, στην έκδοση 0.109.1, χρησιμοποιείται κυρίως για την ενσωμάτωση του Isolation Forest ML μοντέλου, το οποίο χρησιμοποιείται για την ανίχνευση ανωμαλιών στα δεδομένα. Με τη βοήθεια της βιβλιοθήκης Pydantic, το FastAPI εξασφαλίζει την επαλήθευση των δεδομένων και την αυτόματη δημιουργία documentation για τα API endpoints.

Παράλληλα, χρησιμοποιούμε το Spring Boot για την υλοποίηση των βασικών υπηρεσιών της εφαρμογής. Επιλέξαμε Spring Boot 3.3 MVC αντί για WebFlux, δεδομένου ότι σε αυτή τη φάση της ανάπτυξης δεν μας απασχολεί ο παραλληλισμός. Το MVC παρέχει την απαραίτητη σταθερότητα και ευκολία για την υλοποίηση των βασικών λειτουργιών της εφαρμογής, χωρίς την πολυπλοκότητα της αντιδραστικής προγραμματιστικής προσέγγισης του WebFlux.

Επιπλέον, για τον έλεγχο της λειτουργικότητας και της σταθερότητας του κώδικα, χρησιμοποιήθηκε JUnit, όπου προσφέρει ένα ευέλικτο πλαίσιο για τη δημιουργία και testing.

Κεφάλαιο 6.1.2: API Endpoints

Για να εξασφαλίσουμε την ομαλή επικοινωνία μεταξύ του Frontend και του Backend της πλατφόρμας, δημιουργήθηκαν διάφορα API endpoints με το Spring Boot. Αυτά τα endpoints παρέχουν τις απαραίτητες λειτουργίες για την διαχείριση χρηστών και τη συλλογή δεδομένων πρόσβασης. Ενδεικτικά, περιλαμβάνονται τα εξής:

- **/auth/login:** Post Endpoint για τη σύνδεση χρηστών στην πλατφόρμα.
- **/auth/me:** Get Endpoint για τον ανάκτηση πληροφοριών συνδεδεμένου χρήστη.
- **/user:** Post Endpoint για τη δημιουργία νέων χρηστών.
- **/user/{id}:** Put Endpoint για την ενημέρωση πληροφοριών χρηστών.
- **/access:** Get Endpoint για την ανάκτηση δεδομένων πρόσβασης.
- **/nfc:** Post Endpoint για δημιουργία NFC Tag.
- **/nfc/checkValid:** Get Endpoint για την έλεγχο NFC Tag εαν είναι expired.

Επιπλέον, για την υλοποίηση της ανίχνευσης ανωμαλιών μέσω του Isolation Forest ML μοντέλου, χρησιμοποιούμε το FastAPI. Αυτό το μοντέλο προβλέπει ανωμαλίες με βάση τα δεδομένα από τη βάση δεδομένων PostgreSQL.

Τα σχετικά endpoints περιλαμβάνουν:

- **train:** Endpoint για την εκπαίδευση του μοντέλου με νέα δεδομένα.
- **evaluate:** Endpoint για το performance του μοντέλου.
- **predict:** Endpoint που δέχεται δεδομένα από τη βάση και επιστρέφει προβλέψεις ανωμαλιών.

Κεφάλαιο 6.1.3: Isolation Forest ML Model

Το Isolation Forest ML Model αποτελεί ένα κρίσιμο μέρος του συστήματος ασφαλείας της πλατφόρμας, καθώς παίρνει δεδομένα χρηστών ως είσοδο σε πραγματικό χρόνο και ανιχνεύει πιθανές παραβιάσεις ασφάλειας. Χρησιμοποιώντας ένα ή περισσότερα features, το μοντέλο προβλέπει εάν υπάρχει κάποια ανωμαλία. [13]

Η λειτουργία του μοντέλου βασίζεται στην απομόνωση των ανωμαλιών στα δεδομένα. Αντί να προσπαθεί να εντοπίσει τα κανονικά παραδείγματα, το μοντέλο απομονώνει τις ανωμαλίες, τις οποίες θεωρεί λιγότερο “κανονικές” και πιο εύκολες να απομονωθούν. Με τη χρήση των thresholds, το μοντέλο αξιολογεί την ανωμαλία σε σχέση με ένα κατώφλι, καθορίζοντας εάν πρόκειται για παραβίαση ή όχι.

Το μοντέλο **Isolation Forest** είναι ιδανικό για την ανίχνευση ανωμαλιών στην πλατφόρμα καθώς προσφέρει ορισμένα σημαντικά πλεονεκτήματα σε σχέση με άλλα μοντέλα και μπορεί να εντοπίσει γρήγορα ασυνήθιστες συμπεριφορές ή ενέργειες χρηστών, καθιστώντας το συχνά την καλύτερη επιλογή για πολλές περιπτώσεις χρήσης όπως:

- **Αποτελεσματικότητα σε μεγάλα δεδομένα**: Είναι ελαφρύ και έχει χαμηλές απαιτήσεις σε υπολογιστική ισχύ, γεγονός που το καθιστά κατάλληλο για πλατφόρμες με μεγάλο αριθμό χρηστών και υψηλό όγκο δεδομένων.
- **Ανεξαρτησία από την κατανομή των δεδομένων**: Δεν απαιτεί υποθέσεις για την κατανομή των δεδομένων, όπως π.χ. η κανονικότητα, κάτι που το κάνει ευέλικτο σε διαφορετικά σενάρια.
- **Εντοπισμός άγνωστων ανωμαλιών**: Σε αντίθεση με προβλεπόμενα μοντέλα, το Isolation Forest λειτουργεί χωρίς εκ των προτέρων κατηγοριοποίηση των ανωμαλιών, καθιστώντας το ικανό να ανιχνεύει νέες και απροσδόκητες ανωμαλίες.
- **Απλότητα και γρήγορη υλοποίηση**: Η εκπαίδευση και η χρήση του είναι απλές και γρήγορες, συγκριτικά με πιο σύνθετα μοντέλα όπως το Deep Learning, που απαιτούν μεγάλο όγκο δεδομένων και εκπαίδευσης.
- **Ακρίβεια και ερμηνευσιμότητα**: Παρέχει αξιόπιστα αποτελέσματα, ενώ η εσωτερική λειτουργία του είναι πιο εύκολα κατανοητή σε σχέση με άλλα μοντέλα όπως τα νευρωνικά δίκτυα.

Με βάση τα παραπάνω, το Isolation Forest αποτελεί μια ισορροπημένη λύση που συνδυάζει ταχύτητα, αξιοπιστία και ευελιξία, καθιστώντας το ιδιαίτερα χρήσιμο για ανίχνευση ανωμαλιών στην πλατφορμα.

Κεφάλαιο 6.2: Ανάπτυξη και Παραμετροποίηση Frontend (Quasar Vue3 Framework)

Το Quasar Framework χρησιμοποιήθηκε για την ανάπτυξη του frontend, παρέχοντας μια πλούσια και ευέλικτη διεπαφή χρήστη. Η επιλογή του βασισμένο στην Vue.js 3 προσφέρει σημαντικά πλεονεκτήματα, ειδικά αν θέλουμε να δημιουργήσουμε γρήγορα σύγχρονες, υψηλής ποιότητας web και mobile εφαρμογές. [6]

Μερικα βασικά πλεονεκτήματα έναντι άλλων frameworks είναι τα εξής:

- **Βασισμένο στο Vue.js 3:** Επωφελείται από όλα τα πλεονεκτήματα της Vue 3, όπως Composition API,reactive δυνατότητες και καλύτερη απόδοση.

  Παρέχει συμβατότητα με την κοινότητα και τα εργαλεία της Vue [7].
- **Έτοιμα Components και Layouts:** Το Quasar διαθέτει μία μεγάλη συλλογή από έτοιμα UI components (π.χ. κουμπιά, πίνακες) που είναι responsive, μοντέρνα και εύκολα προσαρμόσιμα.

  Προσφέρει έτοιμα layouts για ταχύτερη υλοποίηση.

- **Βελτιστοποίηση Απόδοσης:** Ενσωματωμένα εργαλεία για βελτιστοποίηση απόδοσης, όπως tree-shaking, lazy loading και δυνατότητα compact builds.

  Εξασφαλίζει γρήγορο χρόνο φόρτωσης και βέλτιστη εμπειρία χρήστη.
- **Απλοποιημένη Διαχείριση Στυλ:** Παρέχει υποστήριξη για CSS pre-processors (SASS, LESS), καθώς και δυνατότητες για εύκολη παραμετροποίηση θεμάτων με global variables.
- **PWA, SSR και BEX Υποστήριξη:** Εύκολη δημιουργία Progressive Web Apps (PWA), Server-Side Rendered (SSR) εφαρμογών και Browser Extensions (BEX) χωρίς πρόσθετη πολυπλοκότητα.
- **Εργαλεία Ανάπτυξης (DevTools):** Έχει ενσωματωμένα devtools που διευκολύνουν τη διαδικασία ανάπτυξης και δοκιμών.

  Παρέχει CLI για την απλοποίηση του setup και του deployment.
- **Ισχυρή Κοινότητα**: Πολύ καλά τεκμηριωμένο framework, το οποίο παρέχει οδηγούς και παραδείγματα για γρήγορη εκμάθηση.

  Υποστηρίζεται από μια ενεργή κοινότητα, που σημαίνει συνεχή ανάπτυξη και υποστήριξη.
- **Απλότητα και Ευελιξία**: Σε σχέση με άλλα frameworks (π.χ. Angular ή React), το Quasar διατηρεί την απλότητα της Vue, ενώ προσφέρει εργαλεία που καλύπτουν ένα ευρύ φάσμα αναγκών.

Εφόσον λοιπόν θέλαμε να συνδυάσουμε την ευκολία ανάπτυξης, τη συμβατότητα με πολλές πλατφόρμες και την υψηλή απόδοση, το Quasar Framework ηταν μια εξαιρετική επιλογή, ειδικά αν γνωρίζουμε Vue.

Κεφάλαιο 6.2.1: Components και Σελίδες

Η δημιουργία των components και σελίδων έγινε με στόχο την επίτευξη μιας φιλικής προς τον χρήστη εμπειρίας.

- **Components**: Δημιουργήθηκαν επαναχρησιμοποιήσιμα components για διάφορα τμήματα της διεπαφής χρήστη, όπως φόρμες, πίνακες δεδομένων, και κουμπιά.

  Η χρήση των components διευκολύνει τη συντήρηση και την επέκταση της εφαρμογής.
- **Σελίδες**: Αναπτύχθηκαν διάφορες σελίδες που περιλαμβάνουν λειτουργίες όπως το dashboard, οι ρυθμίσεις χρήστη και η προβολή των αποτελεσμάτων του μοντέλου μηχανικής μάθησης.

  Η πλοήγηση μεταξύ των σελίδων γίνεται μέσω του Vue Router, το οποίο ενσωματώνεται στο Quasar Framework.

Κεφάλαιο 6.2.2: Χρήση Web NFC API

Η χρήση του Web NFC API είναι μια σχετικά νέα τεχνολογία που επιτρέπει την αλληλεπίδραση με φυσικές κάρτες NFC απευθείας από τον browser αλλά μόνον μέσω μιας συσκευής mobile Android. Ο συγκεκριμένος περιορισμός προκυπτει απο την ίδια την τεχνολογία και όχι από την πλατφόρμα που υλοποιούμε. [10]

- **Web NFC API**: Το API αυτό επιτρέπει την ανάγνωση και εγγραφή δεδομένων σε NFC tags μέσω του browser. Υποστηρίζεται από σύγχρονους browsers Android και παρέχει έναν εύκολο τρόπο για την ενσωμάτωση NFC λειτουργιών στην εφαρμογή.
- **Ενσωμάτωση**: Δημιουργήθηκε ένα component (DashUserTable.vue) και μια σελίδα (RegisterNFC.vue) που που χρησιμοποιεί το Web NFC API για την αλληλεπίδραση με NFC tags. Οι διαχειριστές μπορούν να σαρώσουν την κάρτα των χρηστών για να εγγράψουν δεδομένα στην κάρτα.

Πλεονεκτήματα της Χρήσης του Web NFC API

- **Εύκολη Ενσωμάτωση**: Οι προγραμματιστές μπορούμε να χρησιμοποιήσουμε pure JavaScript για να αξιοποιήσουμε NFC δυνατότητες, χωρίς την ανάγκη επιπλέον εφαρμογών ή plugins.
- **Ασφαλής Επικοινωνία**: Το Web NFC API λειτουργεί μόνο μέσω HTTPS, διασφαλίζοντας την προστασία των δεδομένων που ανταλλάσσονται.
- **Περιορισμένη Εμβέλεια**: Το NFC λειτουργεί σε μικρή απόσταση (συνήθως έως 4 εκατοστά), καθιστώντας δύσκολη την υποκλοπή δεδομένων από μη εξουσιοδοτημένους χρήστες.

Πώς Λειτουργεί

- **Σάρωση (Scanning)**: Το API επιτρέπει τη σάρωση NFC tags για την ανάγνωση δεδομένων που περιέχονται σε αυτά, όπως URLs, κείμενο, ή προσαρμοσμένα δεδομένα.
- **Εγγραφή (Writing)**: Οι ιστοσελίδες μπορούν να εγγράψουν δεδομένα σε NFC tags, όπως για παράδειγμα να αποθηκεύσουν πληροφορίες σύνδεσης ή ρυθμίσεις.
- **Πρωτόκολλα**: Υποστηρίζει μορφές NFC Data Exchange Format (**NDEF**) για την ανταλλαγή δεδομένων.

Κεφάλαιο 6.3: Παραμετροποίηση Docker Containers

Η παραμετροποίηση των Docker containers είναι μια κρίσιμη διαδικασία για τη βελτιστοποίηση της λειτουργίας και της διαχείρισής τους. Περιλαμβάνει την προσαρμογή των ρυθμίσεων ενός container ώστε να ανταποκρίνεται στις απαιτήσεις της εφαρμογής και του περιβάλλοντος στο οποίο εκτελείται. Ακολουθούν κάποιες έννοιες και πρακτικές που σχετίζονται με την παραμετροποίηση:

- **Environment Variables:** Παράμετροι που μπορούν να χρησιμοποιηθούν για να περάσουν τιμές στις εφαρμογές μέσω των container. Αυτό γίνεται μέσω της επιλογής -e ή μέσω ενός αρχείου .env .
  ex. docker run -e ENV_VARIABLE=value access-guard
- **Resource Limits:** Για τον καθορισμό πόρων από ένα container, μπορουμε να ορίσουμε όρια για CPU και μνήμη.
  ex. docker run --memory="512m" --cpus="1.0" access-guard
- **Volumes:** Τα volumes χρησιμοποιούνται για την αποθήκευση δεδομένων που απαιτούνται από το container, εξασφαλίζοντας τη διατήρηση των δεδομένων ακόμα και αν το container διαγραφεί.
  ex. docker run -v /host/path:/container/path access-guard

Κεφάλαιο 6.3.1: Δημιουργία Dockerfile

Το Dockerfile είναι το αρχείο που περιγράφει τα βήματα για τη δημιουργία ενος Docker image. Μέσα σε αυτό το αρχείο καθορίζονται οι εντολές που πρέπει να εκτελέσει το Docker για να δημιουργήσει ένα container που περιέχει την εφαρμογή ή το περιβάλλον που θέλουμε να αναπτύξουμε. Κάποιες Βασικές Εντολές του Dockerfile στο BackEnd microservice που χρησιμοποιήσαμε:

- **FROM:** Η εντολή FROM καθορίζει την βάση πάνω στην οποία θα δημιουργηθεί η εικόνα. Αυτή μπορεί να είναι μια υπάρχουσα εικόνα του Docker Hub ή μια άλλη εικόνα που έχουμε δημιουργήσει.
  ex. FROM eclipse-temurin:21

- **ADD:** Η εντολή ADD καθορίζει την βάση πάνω στην οποία θα δημιουργηθεί η εικόνα. Αυτή μπορεί να είναι μια υπάρχουσα εικόνα του Docker Hub ή μια άλλη εικόνα που έχουμε δημιουργήσει.

  ex. ADD target/AccessGuard-1.1.2.jar guard.jar
- **EXPOSE:** Η εντολή EXPOSE δηλώνει τις θύρες που θα ακούει το container για επικοινωνία.

  ex. EXPOSE 8080
- **CMD:** Η εντολή CMD καθορίζει την εντολή που θα εκτελείται όταν ξεκινήσει το container. Στην περίπτωση αυτή, η εντολή είναι να ξεκινήσει μια εφαρμογή Java με συγκεκριμένες ρυθμίσεις μνήμης JVM .

  ex. CMD ["java", "-jar", "-Xms512M", "-Xmx1536M", "guard.jar"]

Κεφάλαιο 6.3.2: Συνθεση Docker Compose

Με το **Docker Compose** μας επιτρέπεται η εύκολη οργάνωση και εκκίνηση πολλών containers ταυτόχρονα, χρησιμοποιώντας ένα μόνο αρχείο YAML, το οποίο περιγράφει την σύνθεση των containers, τα δίκτυα, τα volumes και τις εξαρτήσεις μεταξύ τους.

Ακολουθεί ένα παράδειγμα του Spring Boot BackEnd Development mode για το πώς μπορεί να δομηθεί το αρχείο docker-compose.yml

```
services:
  postgres:
        image: postgres:14.2-alpine3.15
        container_name: access_guard_postgres
        ports:
        - "5432:5432"
        volumes:
        - access-guard-postgres-data:/var/lib/postgresql/data
        - ./conf/postgres-init.sql:/docker-entrypoint-initdb.d/postgres-init.sql:ro
        env_file:
        - ./env/postgres.env
        networks:
        - access-guard-network

volumes:
  access-guard-postgres-data:
        name: access_guard_postgres_data
        driver: local

networks:
  access-guard-network:
        name: access_guard_network
        driver: bridge
```

Εικόνα 3: Backend Docker Compose File

Εξήγηση του Παραδείγματος

- **services:** Η ενότητα που περιγράφει τα containers που θέλουμε να εκτελέσουμε. Στο παράδειγμα, έχουμε ένα service
- **networks:** Ορίζουμε το δίκτυο access-guard-network για να επιτρέψουμε στα containers να επικοινωνούν μεταξύ τους.
- **volumes:** Δημιουργούμε ένα volume named access_guard_postgres_data, το οποίο χρησιμοποιείται για να αποθηκεύει τα δεδομένα της βάσης δεδομένων, διασφαλίζοντας ότι τα δεδομένα παραμένουν ακόμα κι αν το container διαγραφεί**.**

Κεφάλαιο 6.4: Παραμετροποίηση Nginx

Η παρακάτω διαμόρφωση του Nginx έχει ως στόχο να λειτουργήσει ως reverse proxy για την προώθηση requests σε δύο διαφορετικές υπηρεσίες: μία frontend εφαρμογή και ένα backend API.
Παράλληλα, εξασφαλίζει ασφαλή επικοινωνία μέσω SSL, αξιοποιώντας self-signed πιστοποιητικά που προαπαιτείται απο το WEB NFC API.
Η χρήση του Nginx επιτυγχάνει επίσης SSL Offloading, μεταφέροντας την επεξεργασία της αποκρυπτογράφησης/κρυπτογράφησης του SSL traffic από τον web server στο Nginx. Αυτό μειώνει το φορτίο στον server και βελτιώνει την απόδοση.
Το αρχείο περιλαμβάνει βασικές ρυθμίσεις για τον καθορισμό των διαθέσιμων πόρων του server, τη δρομολόγηση των αιτημάτων με βάση τη διαδρομή (path), και τη διαχείριση κεφαλίδων για τη σωστή μεταφορά δεδομένων.
Στη συνέχεια, αναλύονται οι εντολές για την καλύτερη κατανόηση της λειτουργίας και της χρησιμότητάς τους.

```
worker_processes 1;

events {
        worker_connections 1024;
}

http {
        server {
        listen 443 ssl;

        # SSL certificates
        ssl_certificate /etc/nginx/certs/selfsigned.crt;
        ssl_certificate_key /etc/nginx/certs/selfsigned.key;

        # Route to the frontend
        location / {
        proxy_pass http://access-guard-ui:80; # Container name for the frontend
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        }

        # Route to the backend
        location /api/ {
        proxy_pass http://backend:8080; # Container name for the backend
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        }
        }
}
```

Εικόνα 4: Nginx Configuration File

worker_processes 1; -> Καθορίζει τον αριθμό των worker processes του Nginx. Εδώ ορίζεται σε 1, που σημαίνει ότι θα χρησιμοποιηθεί ένας worker process. Αυτό μπορεί να αυξηθεί ανάλογα με τον αριθμό των CPU cores του συστήματος για καλύτερη απόδοση.

worker_connections 1024; -> Ορίζει τον μέγιστο αριθμό συνδέσεων που μπορεί να διαχειριστεί κάθε worker process ταυτόχρονα. Εδώ επιτρέπονται μέχρι 1024 συνδέσεις.

http {} Block

server {} -> Ορίζει έναν server block που περιλαμβάνει ρυθμίσεις για έναν συγκεκριμένο virtual server (host).

SSL Settings

listen 443 ssl; -> Ορίζει ότι ο server θα ακούει αιτήματα στο port 443 (το προεπιλεγμένο για HTTPS) και θα χρησιμοποιεί SSL/TLS για ασφαλή επικοινωνία.

ssl_certificate /etc/nginx/certs/selfsigned.crt;

ssl_certificate_key /etc/nginx/certs/selfsigned.key; -> Καθορίζει τη διαδρομή για το αρχείο πιστοποιητικού SSL που θα χρησιμοποιηθεί για την κρυπτογράφηση των συνδέσεων και για το ιδιωτικό κλειδί (private key) που συνοδεύει το πιστοποιητικό SSL.

Frontend Route

l**ocation / {} ->**Δηλώνει ότι όλες οι αιτήσεις που καταλήγουν στη βασική διαδρομή (/) θα δρομολογούνται προς τον frontend container.

proxy_pass http://access-guard-ui:80; -> Ορίζει ότι τα αιτήματα θα προωθούνται στον container με όνομα access-guard-ui στη θύρα 80.

proxy_set_header Host $host; -> Ορίζει την προώθηση της κεφαλίδας Host (όνομα του server που ζητήθηκε) στον προορισμό.

proxy_set_header X-Real-IP $remote_addr; -> Στέλνει τη διεύθυνση IP του πελάτη (client) στον προορισμό μέσω της κεφαλίδας X-Real-IP.

proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for; -> Καταγράφει την αλυσίδα των διευθύνσεων IP που πέρασαν από τους proxies μέσω της κεφαλίδας X-Forwarded-For.

Backend Route

location /api/ {} -> Δηλώνει ότι όλες οι αιτήσεις που ξεκινούν με /api/ θα δρομολογούνται προς τον backend container.

proxy_pass http://backend:8080; -> Ορίζει ότι τα αιτήματα θα προωθούνται στον container με όνομα backend στη θύρα 8080.

proxy_set_header Host $host; -> Ορίζει την προώθηση της κεφαλίδας Host (όνομα του server που ζητήθηκε) στον προορισμό.

proxy_set_header X-Real-IP $remote_addr; -> Στέλνει τη διεύθυνση IP του πελάτη (client) στον προορισμό μέσω της κεφαλίδας X-Real-IP.

proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for; -> Καταγράφει την αλυσίδα των διευθύνσεων IP που πέρασαν από τους proxies μέσω της κεφαλίδας X-Forwarded-For.

Κεφάλαιο 7: Δοκιμή και Αντιμετώπιση Προβλημάτων

Κεφάλαιο 7.1: Στρατηγικές Δοκιμών

Οι στρατηγικές δοκιμών περιλαμβάνουν τη μεθοδολογία και τα είδη των δοκιμών που θα εφαρμόσουμε για να διασφαλίσουμε την ποιότητα και την αξιοπιστία της πλατφορμας.

- **Unit Testing**: Το Unit testing αποτελεί ένα κρίσιμο στάδιο υλοποίησης της πλατφορμας, επικεντρώνοντας την λειτουργικότητα του κάθε component ξεχωριστά. Κάθε module οπως user authentication, access permissions, ή οι μηχανισμοί logging, μπορούν να εξεταστούν ξεχωριστά για να διασφαλίσουμε την ορθή λειτουργία τους. Αυτό βοηθάει στο να αναγνωριστούν προβλήματα σε πρωιμο στάδιο υλοποίησης και να εξασφαλίσουμε ότι κάθε κομμάτι του συστήματος λειτουργεί σωστά πριν το τελικό integration. Ένα αποτελεσματικό unit testing φέρει πολύ καλά αποτελέσματα στην πλατφόρμα κάνοντας το πιθανό debugging πολύ ευκολότερο.
- **Integration Testing**: Σκοπός του Integration Testing είναι να διασφαλίσουμε ότι τα διάφορα υποσυστήματα, όπως η βάση δεδομένων, ο μηχανισμός αυθεντικοποίησης και οι μηχανισμοί ελέγχου πρόσβασης, αλληλεπιδρούν σωστά και ότι δεν υπάρχουν προβλήματα στην επικοινωνία τους.
- **System Testing**: Έγιναν δοκιμές του συστήματος ελέγχου πρόσβασης ως σύνολο. Εξασφαλίσαμε ότι όλα τα υποσυστήματα λειτουργούν σωστά και σε αρμονία μεταξυ τους καλύπτοντας τις απαιτήσεις που είχαμε ορίσει στην αρχή του έργου. Περιλαμβανομένων τόσο των λειτουργικων όσο και τις μη λειτουργικων απαιτήσεων του συστήματος, όπως η ασφάλεια.
- **Performance Testing**: Εξέτασαμε την απόδοση του συστήματος ελέγχου πρόσβασης υπό διαφορετικά σενάρια φόρτου. Εξετάσαμε την ταχύτητα του συστήματος, την ικανότητά του να διαχειρίζεται μεγάλο αριθμό χρηστών ταυτόχρονα και τη σταθερότητά του υπό έντονο φόρτο ή ακραίες συνθήκες. Ένα από τα βασικά οφέλη του Quasar είναι η εξαιρετική απόδοσή του, καθώς είναι βελτιστοποιημένο για υψηλές επιδόσεις και κλίμακες. Για να βελτιώσουμε περαιτέρω την απόδοση του συστήματος, ορίσαμε συγκεκριμένες παραμέτρους μνήμης στο JVM, εξασφαλίζοντας την αποδοτικότερη διαχείριση των πόρων και τη βελτίωση της συνολικής απόδοσης του συστήματος.
- **Security Testing**: Εξετάσαμε την ικανότητα του συστήματος να προστατεύει τα δεδομένα και να αποτρέπει κακόβουλες επιθέσεις. Αυτό περιλαμβάνει τη δοκιμή για αδυναμίες όπως μεσω SQL injection, παραβιάσεις αυθεντικοποίησης, και αδυναμίες στο επίπεδο της πρόσβασης, εξασφαλίζοντας ότι μόνο οι εξουσιοδοτημένοι χρήστες μπορούν να αποκτούν πρόσβαση σε ευαίσθητες πληροφορίες. Για να ενισχύσουμε την ασφάλεια, εφαρμόσαμε πρωτόκολλα HTTPS και SSL/TLS για ασφαλή επικοινωνία μεταξύ των χρηστών και του διακομιστή. Επιπλέον, εφαρμόσαμε κρυπτογράφηση για τους κωδικούς πρόσβασης των χρηστών χρησιμοποιώντας κωδικοποίηση Base64, καθώς και κρυπτογράφηση Base64 για τα tokens πρόσβασης των χρηστών, διασφαλίζοντας ότι μόνο οι εξουσιοδοτημένοι χρήστες έχουν πρόσβαση σε ευαίσθητες πληροφορίες και ενισχύσαμε τη συνολική ασφάλεια του συστήματος.

Κεφάλαιο 7.2: Ανάλυση και Διόρθωση Σφαλμάτων

Σε αυτό το κεφάλαιο, περιγράφουμε τη διαδικασία που ακολουθούμε για την ανάλυση και τη διόρθωση των σφαλμάτων που εντοπίσαμε κατά τη διάρκεια της ανάπτυξης και των δοκιμών του συστήματος. Η διαδικασία αυτή περιλαμβάνει διάφορα στάδια, από την καταγραφή των σφαλμάτων έως τη διάγνωση και διόρθωσή τους, εξασφαλίζοντας την καλή λειτουργία της πλατφορμας.

- **Αναφορά Σφαλμάτων :** Η διαδικασία ξεκινά με την καταγραφή κάθε σφάλματος που εντοπίσαμε κατά τη διάρκεια των δοκιμών. Αυτό περιλαμβάνει τη χρήση εργαλείων διαχείρισης έργων, όπως το Trello, για την παρακολούθηση και την καταγραφή των σφαλμάτων. Κάθε σφάλμα καταγράφεται αναφέροντας την περιγραφή του προβλήματος, τα βήματα αναπαραγωγής, και την κατάσταση του σφάλματος.
- **Διάγνωση Σφαλμάτων:** Αφού καταγραφούν τα σφάλματα, η διαδικασία διάγνωσης ακολουθεί την ανάλυση των logs του συστήματος για τον εντοπισμό της αιτίας του σφάλματος, την αναπαραγωγή του σφάλματος, ακολουθώντας τα βήματα που έχουν καταγραφεί στην αναφορά και την χρήση εργαλείων εντοπισμού σφαλμάτων (debugging tools) για να εντοπιστεί ακριβώς το σημείο του κώδικα που προκαλεί το πρόβλημα.
- **Διόρθωση Σφαλμάτων:** Μόλις διαγνωστεί η αιτία του σφάλματος, ακολουθειτε η τροποποίηση του Κώδικα δηλαδη διορθώνονται τα λάθη στον κώδικα, είτε πρόκειται για λάθη σύνταξης είτε για λογικά σφάλματα και αν θεωρηθεί απαραίτητο σε πιο σύνθετα σφάλματα, η αρχιτεκτονική του συστήματος μπορεί να επανασχεδιαστεί για να βελτιωθεί η απόδοση ή η αξιοπιστία της πλατφορμας.
- **Regression Testing:** Μετά από κάθε διόρθωση, εκτελούνται regression tests για να διασφαλιστεί ότι η διόρθωση δεν εισάγει νέα σφάλματα στο σύστημα. Αυτές οι δοκιμές περιλαμβάνουν αυτόματες δοκιμές και δοκιμές συστήματος, για να επαληθευτεί η σωστή λειτουργία όλων των λειτουργιών του συστήματος.

Αυτή η διαδικασία ανάλυσης και διόρθωσης σφαλμάτων δημιουργεί έναν κύκλο συνεχούς βελτίωσης, διασφαλίζοντας ότι το σύστημα παραμένει αξιόπιστο και αποδοτικό σε κάθε στάδιο της ανάπτυξής του.

Κεφάλαιο 8: Deployment σε Παραγωγικό Περιβάλλον

Κεφάλαιο 8.1: Project Setup

Για να μπορέσουμε να εκτελέσουμε την εφαρμογή σε παραγωγικό περιβάλλον, είτε στον προσωπικό υπολογιστή είτε σε έναν διακομιστή, ακολουθήστε τα παρακάτω βήματα:

- Κάντε clone το repo εκτελώντας την παρακάτω εντολή :
  git clone https://github.com/SteliosGeorgalas/access_guard

- Περιηγηθείτε στον φάκελο AccessGuard.
  cd AccessGuard

Κεφάλαιο 8.2: Αρχικοποίηση Βάσης Δεδομένων

Η αρχικοποίηση της βάσης δεδομένων στην Postgres πραγματοποιείται μόνο την πρώτη φορά και υλοποιείται μέσω ενός αρχικού script (init script) σε μορφή .sh. Αυτό το script εκτελείται αυτόματα, καθώς έχει οριστεί ως volume στο αρχείο docker-compose.yml.

Στην περίπτωση που το script εντοπίσει ότι η βάση δεδομένων υπάρχει ήδη, παρακάμπτει τη διαδικασία της αρχικοποίησης, εξασφαλίζοντας ότι η διαδικασία θα εκτελεστεί μόνο όταν είναι απαραίτητη. Αυτή η προσέγγιση βοηθά στη διατήρηση της συνέπειας και στην αποφυγή περιττών επανεκτελέσεων του script.

Κεφάλαιο 8.3: Ενεργοποίηση και Διαχείριση των Microservices

Εφόσον πλέον είμαστε μέσα στον φάκελο του project μπορούμε να εντοπίσουμε το .yml αρχείο όπου βρίσκονται παραμετροποιημένα τα micro service.

- Εκτελούμε την εντολή docker compose -f docker-compose-prod.yml up -d και τα services είναι παραμετροποιημένα να σηκωθούν με την σωστή σειρά.
- Εφόσον τα micro service έχουν κινηθεί θα πρέπει με την εντολή docker ps να βλέπουμε ότι είναι healthy και είναι up

Κεφάλαιο 8.4: Εγγραφή αρχικού admin Χρήστη

Δεδομένου ότι η εφαρμογή μας απευθύνεται μόνο σε admin χρήστες και δεν είναι προσβάσιμη από απλούς users δεν έχουμε δώσει την δυνατότητα να κάνει κάποιος μόνος του self register .

Οπότε για να γίνει η αρχική παραμετροποίηση του χρήστη έχουμε φτιάξει μία μέθοδο στην main κλάση της java όπου εγγράφει έναν admin user με default credentials encrypted .

Username: admin

Password : 123

Διαφορετικά μπορούμε μέσω του endpoint api/v1/createuser να δημιουργήσουμε έναν νέο χρήστη με την βοήθεια του εργαλείου Postman .

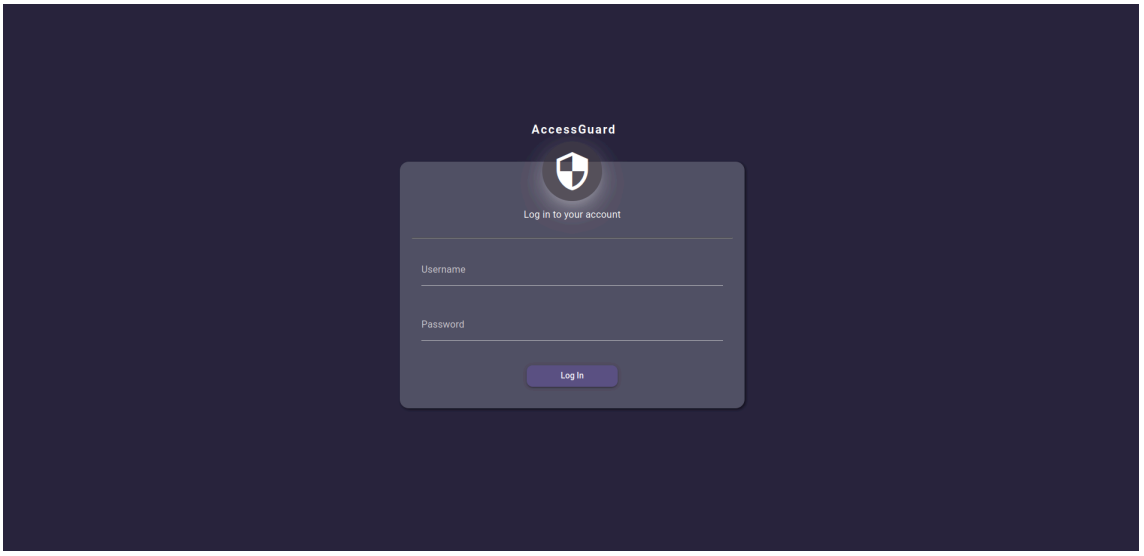
Κεφάλαιο 8.5: Πρόσβαση στην Εφαρμογή

Αφού ολοκληρωθεί η εγκατάσταση και η εκκίνηση των services , μπορούμε να αποκτήσουμε πρόσβαση μέσω ενός web browser χρησιμοποιώντας τη διεύθυνση URL https://localhost
Η εφαρμογή θα μας ζητήσει να κάνουμε login με τα default credentials του admin ή με τα credentials του χρήστη που έχουμε φτιάξει.

Κεφάλαιο 9: Γραφική Αναπαράσταση Πλατφόρμας

Κεφάλαιο 9.1: Σελίδα Σύνδεσης

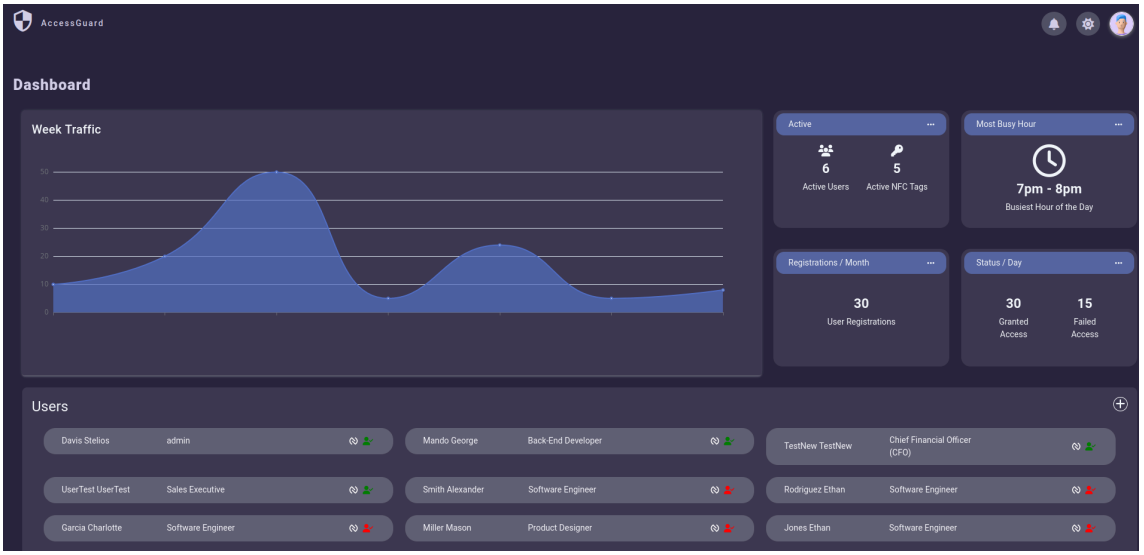
Ο διαχειριστής έχοντας μπει στην πλατφόρμα μπορεί να προχωρήσει σε σύνδεση είτε με τα default credentials που έχουν δημιουργηθεί από την εφαρμογή είτε με τα στοιχεία που έχει ορίσει ο ίδιος.



Εικόνα 5: Login Page

Κεφάλαιο 9.2: Dashboard

Εισάγοντας τα σωστά credentials θα γίνει redirected στο dashboard Βλέπε **Εικόνα 6: Landing Page/ Dashboard** που θα δει το ṁain γραφημα και τέσσερα widgets στην δεξιά μεριά με τις πληροφορίες που έχουμε πάρει από την βάση. Επίσης στην κάτω μεριά εμφανίζεται ένας πίνακας με όλους τους χρήστες που έχουν εγγραφεί όπου μπορεί να προχωρήσει σε παραμετροποιήσεις .

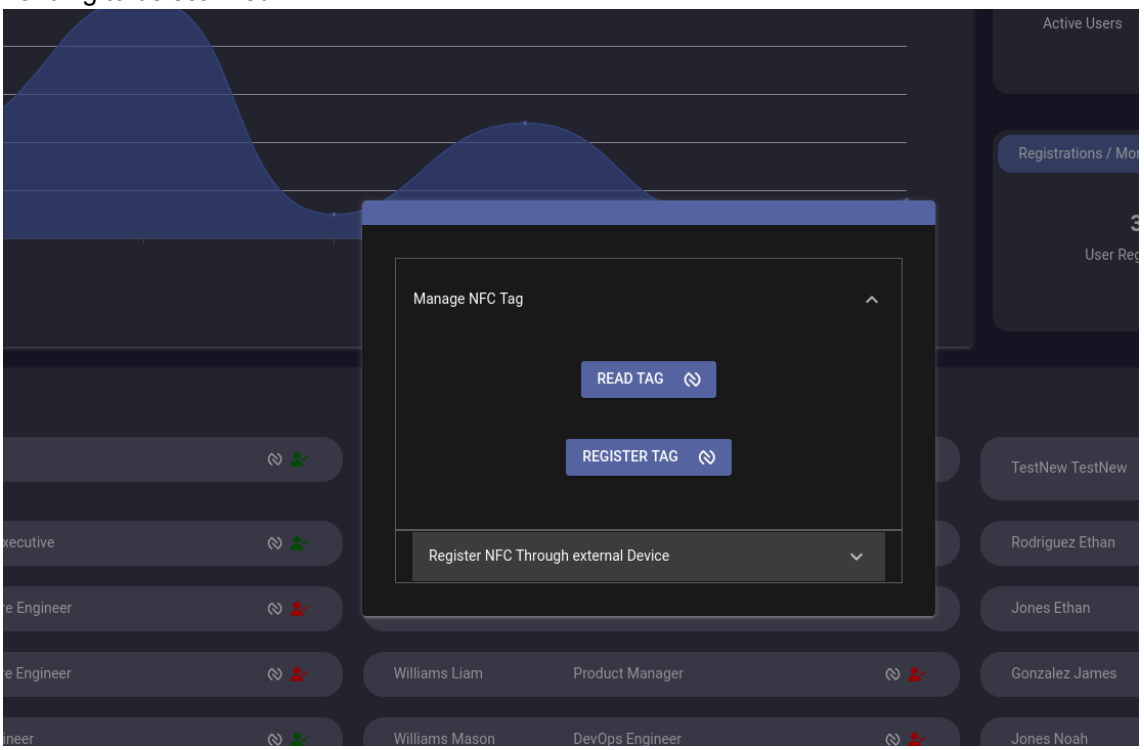


Εικόνα 6: Landing Page/ Dashboard

Κεφάλαιο 9.3: Χρήση NFC

Κεφάλαιο 9.3.1: Σάρωση NFC

Για την σάρωση ενός nfc Tag ο admin μπορεί να προχωρήσει πατώντας το κουμπί Read Tag Βλέπε **Εικόνα 7: NFC Scan** όπου θα εμφανιστεί ένα μήνυμα από την Πλατφόρμα “Nfc Tag Pending to be scanned”

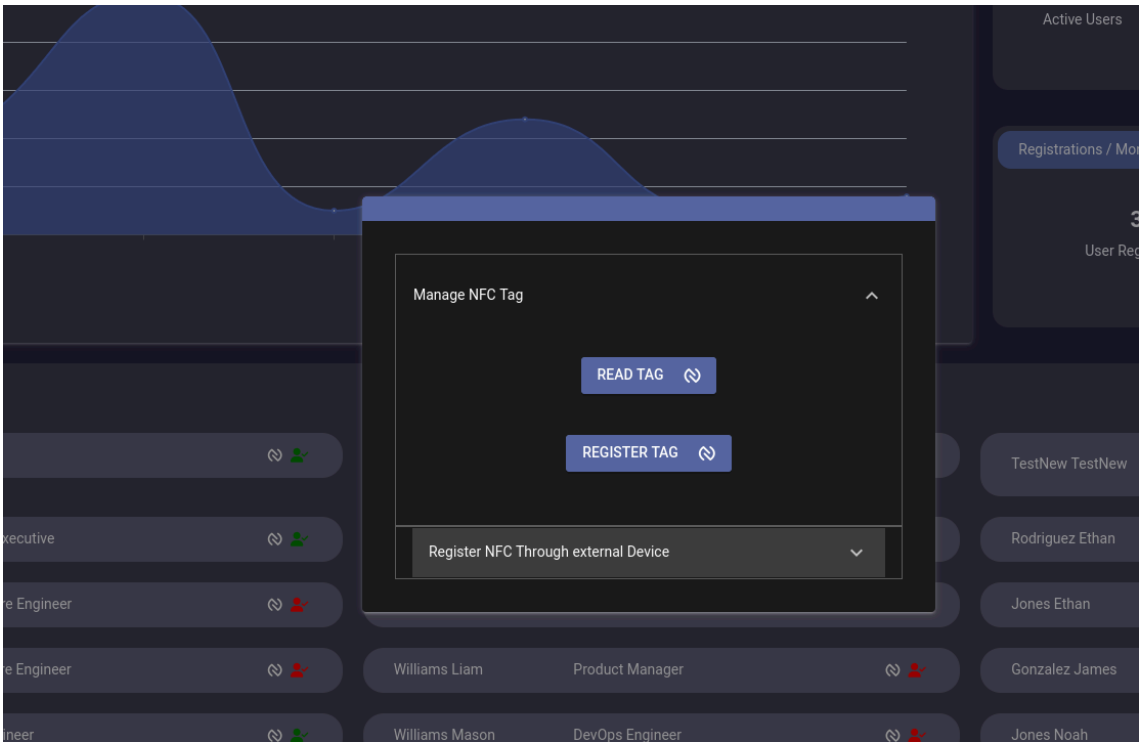


Εικόνα 7: NFC Scan

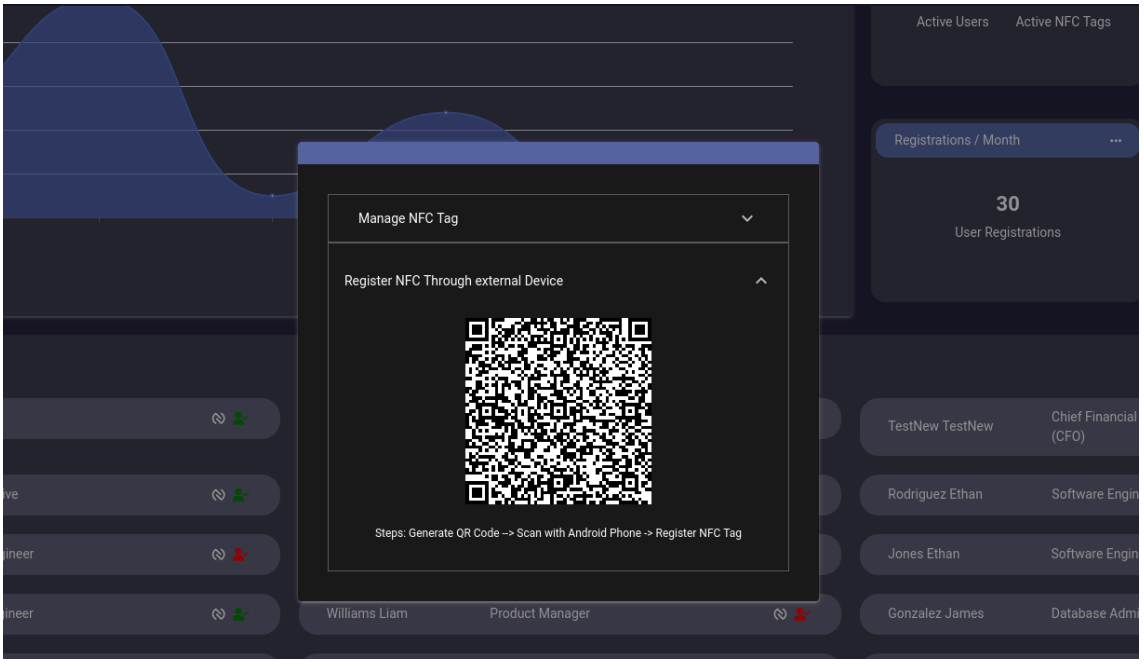
Κεφάλαιο 9.3.2: Εγγραφή NFC

Για την εγγραφή NFC η πλατφορμα υποστηρίζει δύο τρόπους:

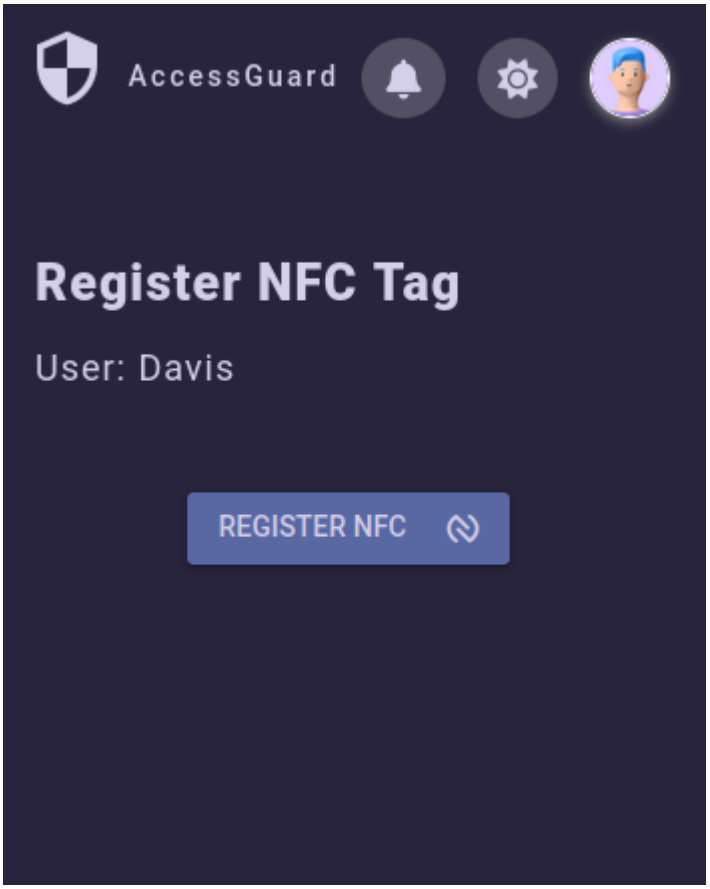
- Ο ένας είναι με απευθείας εγγραφή στο Tag εάν και εφόσον η πλατφορμα ελέγξει ότι ο browser υποστηρίζεται Βλέπε **Εικόνα 8: Register Tag** Κουμπί Register Tag
- Αν ο browser δεν υποστηρίζεται τότε η πλατφόρμα αξιοποιεί έναν δεύτερο τρόπο εγγραφής nfc όπου επιτυγχάνεται μέσω της σάρωσης ενός QR code Βλέπε **Εικόνα 9: Register Tag Qr Code Via External Device** όπου όταν ο admin τον σαρώσει θα γίνει redirected σε εξωτερική σελίδα Βλέπε **Εικόνα 10: Register Tag Page Via External Device**.
  Για το validity της σελίδας και για να γνωρίζει η πλατφόρμα σε ποιον χρήστη πάμε να κάνουμε assign ένα tag δημιουργούμε ένα token που μπαίνει σαν url query param και έχει συγκεκριμένο expiration time .
  Κατα το mount της σελίδα γίνεται validate το token και αν είναι valid ο admin μπορεί να προχωρήσει στο NFC Register μέσω του Android Mobile του .



Εικόνα 8: Register Tag



Εικόνα 9: Register Tag Qr Code Via External Device

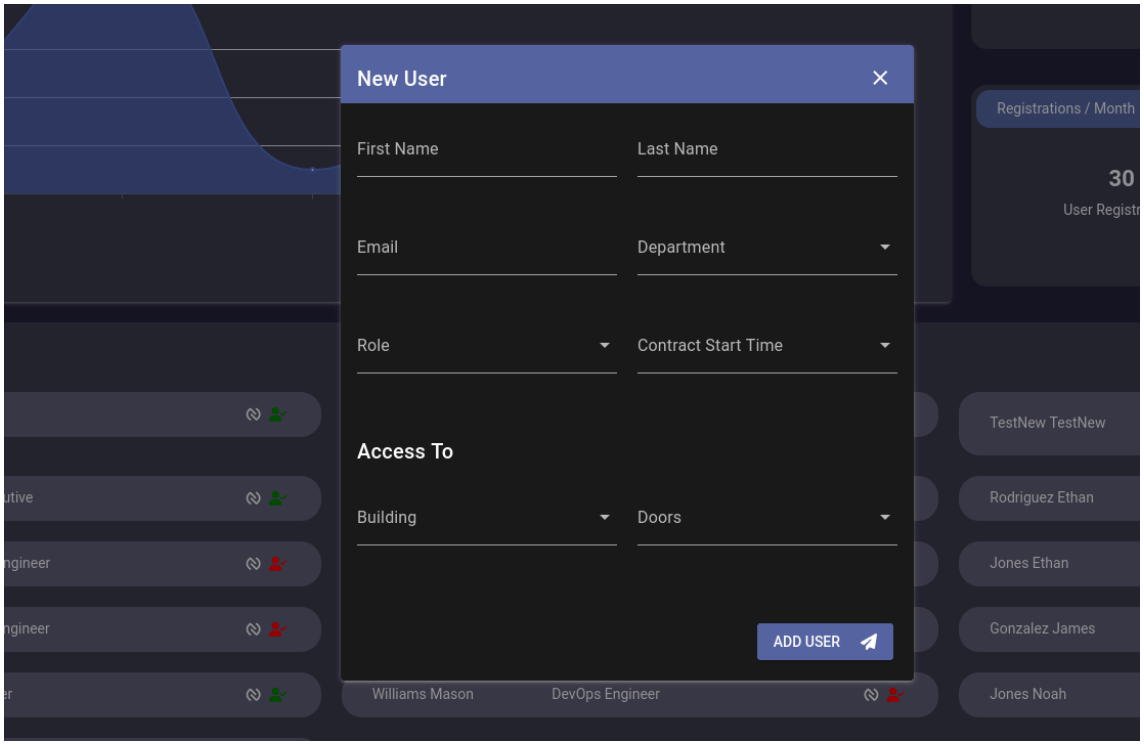


Εικόνα 10: Register Tag Page Via External Device

Κεφάλαιο 9.4: Διαχείριση Χρηστών

Κεφάλαιο 9.4.1: Προσθήκη νέου χρήστη

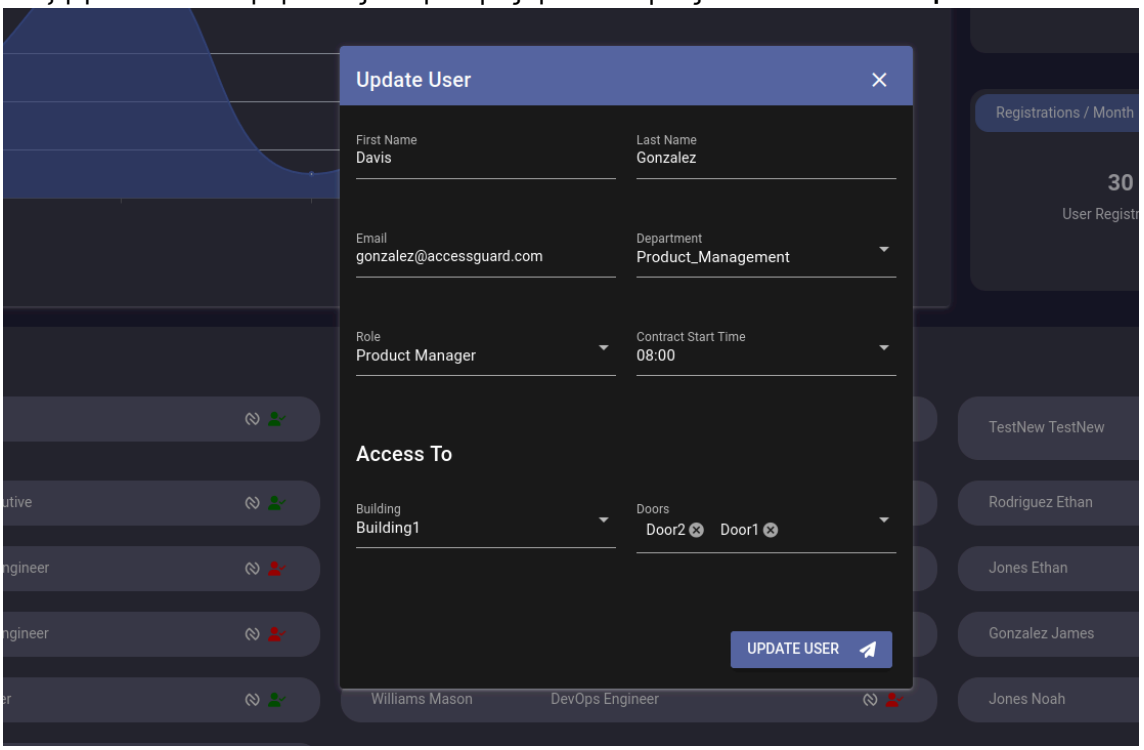
Ο διαχειριστής μπορεί να δημιουργήσει έναν νέο χρήστη ενημερώνοντας τα απαραίτητα στοιχεία και να πατήσει Add User Βλέπε **Εικόνα 11: Add new User**



Εικόνα 11: Add new User

Κεφάλαιο 9.4.2: Επεξεργασία χρήστη

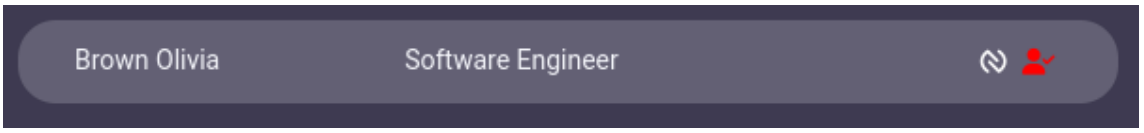
Ο διαχειριστής μπορεί να επεξεργαστεί τα στοιχεία ενός χρήστη πατώντας το κουμπί επεξεργασία και να προβεί στις απαραίτητες τροποποιήσεις Βλέπε **Εικόνα 12: Update User**



Εικόνα 12: Update User

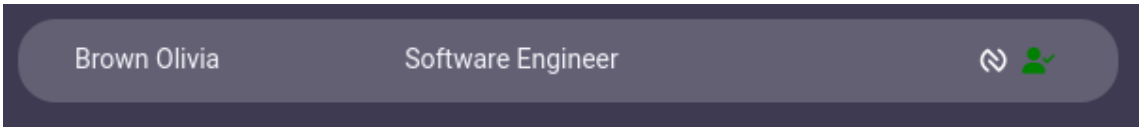
Κεφάλαιο 9.4.3: Απενεργοποίηση/ Ενεργοποίηση χρήστη

Πατώντας το κόκκινο User Icon Βλέπε **Εικόνα 13: Disabled User** του απενεργοποιημένου χρήστη πλέον θα ενεργοποιηθεί και θα εμφανιστεί το αντίστοιχο ενημερωτικό μήνυμα από την πλατφόρμα “User Successfully Enabled.”



Εικόνα 13: Disabled User

Πατώντας το πρασινο User Icon Βλέπε **Εικόνα 14: Enabled User** του ενεργοποιημένου χρήστη πλέον θα απενεργοποιηθεί και θα εμφανιστεί το αντίστοιχο ενημερωτικό μήνυμα από την πλατφόρμα “User Successfully Disabled.”



Εικόνα 14: Enabled User

Κεφάλαιο 10: Isolation Forest ML Model

Κεφάλαιο 10.1: Εκπαίδευση ML Μοντέλου

Για την εκπαίδευση του μοντέλου θα χρησιμοποιήσουμε ένα generated access dataset χρηστών 60000 εγγραφών όπου το 10% των εγγραφών είναι anomaly και θα ορίσουμε στο model contamination 0.1 και n_estimators = 50 .
Το dataset δημιουργήθηκε με την βοήθεια της python με το scipt (generate_dataset.py) και το παραγόμενο αρχείο είναι το synthetic_dataset.json.
To endpoint που χτυπάμε για την εκπαίδευση του μοντέλου είναι:
http://localhost:8000/train
Και το request που στέλνουμε της μορφής
ex.

```
{
        "user_old_access_logs": [
        {
        "access_type": 1,
        "failures": 0,
        "access_time": "2024-05-21T07:00:00",
        "contract_time": "08:00"
        },
        .
        .
        .
        .
        .
        .
        .
        .
```

```
        {
        "access_type": 1,
        "failures": 5,
        "access_time": "2024-05-21T19:45:00",
        "contract_time": "08:00"
        }
        ]
}
```


Στις εγγραφές διαπιστώνουμε ότι υπάρχουν ανωμαλίες όπου τις ξεχωρίζουμε απο το failures ή απο την ώρα εισόδου(access_time) όπου είναι και τα features που χρησιμοποιούμε στο μοντέλο.
Το προσδοκώμενο response είναι :


```
{
        "message": "Model trained successfully"
}
```


Κεφάλαιο 10.2: Πρόβλεψη Βάση ML Μοντέλου

Για να προχωρήσουμε σε ένα anomaly detection για testing θα στείλουμε δυο request στο endpoint
http://localhost:8000/predict
Το ένα request θα το στείλουμε με valid data για να δούμε αν θα πάρουμε το expected outcome και το δεύτερο request θα είναι με anomaly data .

- **Valid Access Request**


```
{
        "building_id": "32",
        "contract_time": "08:00",
        "access_time": "2024-05-21T14:30:00",
        "access_type": 1,
        "is_secure": 1,
        "failures": 0
}
```


- **Valid Access Response**


```
{
        "building_id": "32",
        "anomaly": false,
        "anomaly_score": 0.07467984276296619
}
```


- **Anomaly Access Request**


```
{
        "building_id": "32",
        "contract_time": "08:00",
        "access_time": "2024-05-21T19:30:00",
        "access_type": 1,
```

"is_secure": 1,
"failures": 5
}

- **Anomaly Access Response**

{
"building_id": "32",
"anomaly": true,
"anomaly_score": -0.08422519938258088
}

Όπως διαπιστώνουμε στην περίπτωση του anomaly request έχουμε ορίσει failure = 5 οπως επίσης και η ώρα εισόδου είναι μετά το πέρας του ωραρίου εργασίας του χρήστη και το μοντέλο σωστά επιστρέφει ότι έχει εντοπίσει ανωμαλία και επιστρέφει και το score.

Κεφάλαιο 10.3: Απόδοση Μοντέλου

Η αξιολόγηση ενός μοντέλου όπως το Isolation Forest, το οποίο ανήκει στην κατηγορία των αλγορίθμων unsupervised, είναι μια διαδικασία που μπορεί να διαφέρει ανάλογα με τα δεδομένα.
Σε γενικές γραμμές, το μοντέλο αξιολογείται με σκοπό να εκτιμηθεί η ικανότητά του να εντοπίζει σωστά τις ανωμαλίες, διατηρώντας ακρίβεια στα valid δεδομένα.
Εάν υπάρχουν labels, η διαδικασία αξιολόγησης μοιάζει με αυτή των supervised μοντέλων, ενώ εάν δεν υπάρχουν, η αξιολόγηση βασίζεται σε πιο ποιοτικές και ερευνητικές μεθόδους.
Σε κάθε περίπτωση, η αξιολόγηση πραγματοποιείται για testing purposes, καθώς ο αλγόριθμος λειτουργεί ανεξάρτητα από την ύπαρξη των labels.
Οπότε δημιουργούμε με την βοηθεια python script synthetic data όπως κάναμε και στο train ενα dataset για το evaluation του μοντέλου 4000 εγγραφών εκ των οποίων οι 0.08 είναι contaminated ωστε για να ελέγξουμε τις μετρικές που θα παραχθούν (generate_eval_dataset.py)
Το response που παίρνουμε είναι

ML Model Eval response

{
"precision": 0.7519582245430809,
"recall": 0.9,
"f1_score": 0.8193456614509246
}

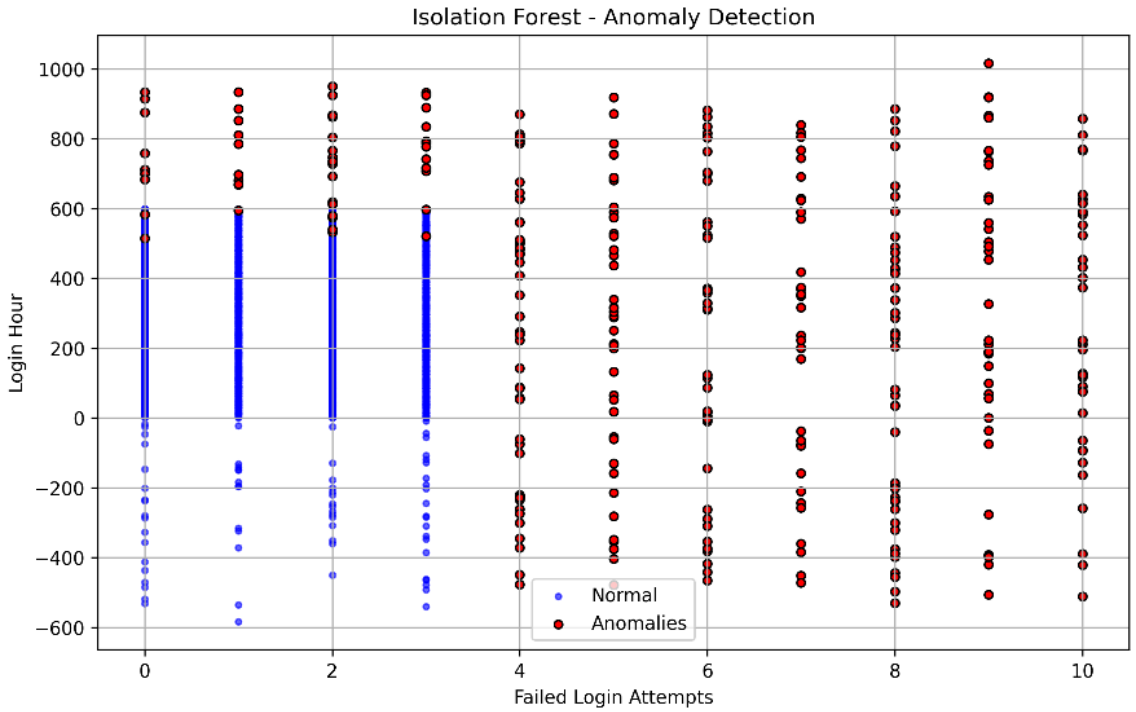
Εδώ βλέπουμε ότι το μοντέλο έχει υψηλή ακρίβεια (75%), πράγμα που σημαίνει ότι όταν εντοπίζει ανωμαλίες, έχει μεγάλη πιθανότητα να είναι σωστό.
Το recall είναι επίσης υψηλο (90%), πράγμα που σημαίνει ότι το μοντέλο εντοπίζει το μεγαλύτερο ποσοστό των πραγματικών ανωμαλιών.
Το F1-Score είναι ικανοποιητικό (0.81), που υποδεικνύει ότι το μοντέλο επιτυγχάνει καλή ισορροπία στην αναγνώριση των ανωμαλιών χωρίς να παράγει πάρα πολλά ψευδώς θετικά ή ψευδώς αρνητικά αποτελέσματα.
Επιπρόσθετα εκτυπώνουμε και το confusion matrix για να έχουμε μία σαφέστερη γενική εικόνα απόδοσης του μοντέλου μας και τα αποτελέσματα διαμορφώνονται ως εξής:

[[3585 95]
[32 288]]

- **True Positives (288)**: Ο αριθμός των πραγματικών ανωμαλιών που εντοπίστηκαν σωστά από το μοντέλο.
- **True Negatives (3585)**: Οι κανονικές παρατηρήσεις που ταξινομήθηκαν σωστά ως μη ανωμαλίες.
- **False Positives (95)**: Οι κανονικές παρατηρήσεις που λανθασμένα χαρακτηρίστηκαν ως ανωμαλίες.
- **False Negatives (32)**: Οι ανωμαλίες που το σύστημα απέτυχε να εντοπίσει, ταξινομώντας τις ως κανονικές.

Η σχετικά χαμηλή τιμή των false positives και false negatives υποδεικνύει ότι το μοντέλο έχει καλή συνολική απόδοση, με τη σωστή διάκριση των ανωμαλιών και των φυσιολογικών παρατηρήσεων.

Για την απεικόνιση των αποτελεσμάτων χρησιμοποιήσαμε ένα διάγραμμα χρησιμοποιώντας δύο features. Κάθε σημείο στο διάγραμμα αντιπροσωπεύει μια εγγραφή του συνόλου δεδομένων, με τους άξονες να δείχνουν τις τιμές των δύο χαρακτηριστικών. Κάθετα απεικονίζεται η διαφορά χρόνου σε λεπτά και οριζόντια οι αποτυχημένες προσπάθειες εισόδου ενός χρήστη. Οι valid εγγραφές εμφανίζονται με έναν διακριτό μπλε χρωματισμό, ενώ οι ανωμαλίες επισημαίνονται με κόκκινο χρώμα, υποδεικνύοντας αποκλίσεις από τη φυσιολογική συμπεριφορά. Η παρουσία αυτών των σημείων βοηθά στην οπτική κατανόηση των περιοχών του χαρακτηριστικού χώρου όπου εντοπίζονται ανωμαλίες.



Εικόνα 15: Isolation Forest Plot

Κεφάλαιο 11: Συμπεράσματα και Μελλοντικές Επεκτάσεις

Κεφάλαιο 11.1: Συμπεράσματα

Σε αυτή την πτυχιακή εργασία, αναπτύξαμε μια web εφαρμογή χρησιμοποιώντας FastAPI, Spring Boot, Vue js Quasar, Nginx , Docker και PostgreSQL.

Η εφαρμογή αυτή περιλαμβάνει τη χρήση ενός μοντέλου μηχανικής μάθησης Isolation Forest για ανίχνευση ανωμαλιών και τη χρήση του Web NFC API για την αλληλεπίδραση με φυσικές κάρτες μέσω NFC.

Μέσα από τη διαδικασία ανάπτυξης, καταφέραμε να δημιουργήσουμε μια λειτουργική και επεκτάσιμη εφαρμογή που μπορεί να καλύψει βασικές ανάγκες ανάλυσης και διαχείρισης δεδομένων.

Κεφάλαιο 11.2: Προτάσεις για Μελλοντική Έρευνα και Ανάπτυξη

Κεφάλαιο 11.2.1: Ενσωμάτωση Keycloak για Ασφάλεια

Keycloak είναι ένα εργαλείο ανοικτού κώδικα για την ταυτοποίηση και την εξουσιοδότηση. Μπορεί να παρέχει ενιαία είσοδο (Single Sign-On) με ενσωμάτωση για LDAP και Active Directory. [12]

Δυνατότητες Keycloak:

- **SSO (Single Sign-On)**: Επιτρέπει στους χρήστες να συνδέονται μία φορά και να έχουν πρόσβαση σε πολλές εφαρμογές χωρίς να επαναλαμβάνουν τη διαδικασία σύνδεσης.
- **LDAP/AD Integration**: Εύκολη ενσωμάτωση με υπάρχοντα συστήματα ταυτοποίησης όπως LDAP και Active Directory.
- **Κοινωνική Ταυτοποίηση**: Υποστήριξη για κοινωνικές ταυτότητες όπως Google, Facebook, και Twitter.
- **Επιβεβαίωση Δύο Παραγόντων (2FA)**: Προσθήκη επιπέδου ασφαλείας μέσω επιβεβαίωσης δύο παραγόντων.
- **Κεντρική Διαχείριση**: Κεντρική διαχείριση χρηστών και δικαιωμάτων μέσω μιας εύχρηστης διεπαφής.

Κεφάλαιο 11.2.2: Eureka ως Service Locator για Load Balancing και Failover

Eureka είναι ένα εργαλείο ανοιχτού κώδικα που αναπτύχθηκε από την Netflix και χρησιμοποιείται για service discovery σε κατανεμημένα συστήματα, ιδιαίτερα σε περιβάλλοντα microservice. Ανήκει στο οικοσύστημα του Spring Cloud, γεγονός που το καθιστά ιδιαίτερα δημοφιλές για εφαρμογές που αναπτύσσονται με το Spring Boot.[13]

Δυνατότητες του Eureka:

- **Service Registration & Discovery**: Ο κύριος σκοπός του Eureka είναι η αυτόματη καταγραφή και ανακάλυψη υπηρεσιών σε συστήματα μικροϋπηρεσιών. Κάθε υπηρεσία καταχωρεί τη διεύθυνση και άλλες παραμέτρους στον Eureka Server, και άλλες υπηρεσίες μπορούν να αναζητήσουν αυτές τις καταχωρήσεις για να επικοινωνήσουν μεταξύ τους. Το Eureka χρησιμοποιεί RESTful APIs για την καταγραφή και ανακάλυψη υπηρεσιών.

- **Health checking**: Το Eureka παρέχει δυνατότητες health checking μέσω του heartbeat. Αν μια υπηρεσία δεν ανανεώσει την καταχώρησή της (μέσω heartbeat) εντός του καθορισμένου χρονικού διαστήματος, θεωρείται ότι έχει αποτύχει και αφαιρείται από την καταχώρηση στο σύστημα. Αυτή η λειτουργία συμβάλλει στην αποφυγή της επικοινωνίας με υπηρεσίες που δεν είναι διαθέσιμες.
- **Client-Side Load Balancing**: Αντί να χρησιμοποιεί έναν κεντρικό load balancer, το Eureka επιτρέπει την εξισορρόπηση φορτίου στο επίπεδο του πελάτη. Κάθε service που επικοινωνεί με το Eureka γνωρίζει όλες τις υπηρεσίες και τα endpoints τους και μπορεί να επιλέξει ποια υπηρεσία να καλέσει, βασισμένο σε καταχωρημένες πληροφορίες και την κατάσταση των υπηρεσιών. Αυτό γίνεται συνήθως μέσω της ενσωμάτωσης με το **Ribbon**, το οποίο παρέχει τις δυνατότητες load balancer.
- **Integration with Spring Cloud**: Το Eureka ενσωματώνεται εύκολα με το Spring Cloud και παρέχει ενσωματωμένη υποστήριξη για client-side load balancing (με χρήση του Ribbon), circuit breaker (με χρήση του Hystrix), καθώς και άλλες δυνατότητες που απαιτούνται για την ανάπτυξη μικροϋπηρεσιών, όπως routing μέσω **Zuul** και άλλες.

Κεφάλαιο 11.2.3: Grafana: Rich Visualization for Monitoring Data

Grafana είναι ένα ανοιχτού κώδικα εργαλείο visualization που χρησιμοποιείται για την παρακολούθηση και την ανάλυση δεδομένων από διάφορες πηγές, όπως βάσεις δεδομένων, συστήματα καταγραφής, και συστήματα παρακολούθησης. Είναι δημοφιλές κυρίως για τη δημιουργία dashboards που παρέχουν οπτικές αναπαραστάσεις των δεδομένων σε πραγματικό χρόνο, επιτρέποντας στους χρήστες να παρακολουθούν την απόδοση, την υγεία και άλλες σημαντικές παραμέτρους των εφαρμογών και υποδομών. [14]

Δυνατότητες του Grafana:

- **Real-time Data Monitoring** : Με το Grafana, μπορείτε να παρακολουθείτε δεδομένα σε πραγματικό χρόνο. Ενημερώνει τα γραφήματα και τους πίνακες σε τακτά χρονικά διαστήματα, δίνοντάς σας τη δυνατότητα να παρακολουθείτε την απόδοση του συστήματός σας ή την κατάσταση των εφαρμογών σε πραγματικό χρόνο.
  Το Grafana υποστηρίζει την αυτόματη ανανέωση των dashboards, για να μπορείτε να βλέπετε τις τελευταίες πληροφορίες χωρίς να χρειάζεται να ανανεώνετε χειροκίνητα τη σελίδα.
- **Alerting (Ειδοποιήσεις)**: Εκτός από την οπτικοποίηση, το Grafana προσφέρει και ειδοποιήσεις (alerts). Ανάλογα με τις μετρικές που παρακολουθείτε, μπορείτε να ρυθμίσετε κανόνες ειδοποίησης για να σας ειδοποιούν όταν οι τιμές των δεδομένων ξεπερνούν ή πέφτουν κάτω από προκαθορισμένα όρια.
- **Δημιουργία και Διαχείριση Dashboards: Τ**ο Grafana επιτρέπει τη δημιουργία και διαχείριση dashboards με widgets που ανανεώνονται δυναμικά και μπορούν να περιέχουν πολλαπλές οπτικοποιήσεις (π.χ., γραφήματα και διαγράμματα).
  Τα dashboards μπορούν να προσαρμοστούν πλήρως με βάση τις ανάγκες του χρήστη και να περιλαμβάνουν φίλτρα, χρονικές περιοχές και άλλες ρυθμίσεις που επιτρέπουν την ακριβή προβολή των δεδομένων.
- **Υποστήριξη για Containerized Εφαρμογές:** Το Grafana υποστηρίζει τις ανάγκες των σύγχρονων cloud-native εφαρμογών, όπως αυτές που εκτελούνται σε Docker ή Kubernetes. Υπάρχουν έτοιμα plugins και dashboards για την παρακολούθηση και ανάλυση της απόδοσης containerized συστημάτων.

Κεφάλαιο 11.2.4: Prometheus: Flexible Monitoring Solution

Prometheus είναι ένα σύστημα παρακολούθησης και ειδοποίησης για κατανεμημένα συστήματα, το οποίο παρέχει μια ισχυρή και ευέλικτη λύση για τη συλλογή, αποθήκευση και ανάλυση μετρικών δεδομένων. Έχει σχεδιαστεί ειδικά για να παρακολουθεί την απόδοση εφαρμογών και υποδομών σε περιβάλλοντα μικροϋπηρεσιών, cloud-native αρχιτεκτονικών και containerized συστημάτων. Το Prometheus είναι πολύ δημοφιλές στον τομέα της παρακολούθησης, κυρίως λόγω της ευχρηστίας του, της κλίμακας του και των ενσωματωμένων δυνατοτήτων ειδοποιήσεων και ανάλυσης.[11]
Δυνατότητες:

- **Χρήση Time Series**: Το Prometheus αποθηκεύει τα δεδομένα του σε μια βάση δεδομένων τύπου time-series, που σημαίνει ότι κάθε μετρική καταγράφεται ως χρονική σειρά δεδομένων (με timestamp) και μπορεί να αναλυθεί βάσει του χρόνου. Αυτό το καθιστά ιδιαίτερα χρήσιμο για την παρακολούθηση της απόδοσης και της διαθεσιμότητας των συστημάτων σε πραγματικό χρόνο.
- **PromQL Query :** Το Prometheus διαθέτει μια ισχυρή γλώσσα query, γνωστή ως PromQL (Prometheus Query Language), η οποία επιτρέπει στους χρήστες να εκτελούν πολύπλοκες αναλύσεις πάνω σε μετρικές. Μπορεί να χρησιμοποιηθεί για την αναζήτηση, τον υπολογισμό και την ανάλυση δεδομένων σε πραγματικό χρόνο.
- **Ενσωμάτωση με Άλλα Εργαλεία**: Ενσωματώνεται άψογα με άλλες λύσεις παρακολούθησης, όπως το **Grafana** για οπτικοποίηση των μετρικών σε πίνακες ελέγχου (dashboards). Το Grafana μπορεί να συνδεθεί με το Prometheus ως πηγή δεδομένων και να δημιουργήσει γραφήματα, διαγράμματα και αναφορές με πραγματικά δεδομένα.
  Υποστηρίζει επίσης την ενσωμάτωση με άλλες υπηρεσίες συλλογής δεδομένων και αποθήκευσης, όπως το **Kubernetes**, **Docker**, **consul** και άλλες λύσεις cloud-native.

Κεφάλαιο 11.2.5: Επέκταση Εφαρμογής για Χρήση από Απλούς Χρήστες

Η εφαρμογή μπορεί να επεκταθεί για χρήση από απλούς χρήστες, παρέχοντας πρόσβαση σε πληροφορίες για τις εγκαταστάσεις που έχουν πρόσβαση και τα δεδομένα τους.
Δυνατότητες:

- **Πρόσβαση σε Εγκαταστάσεις**: Οι χρήστες θα μπορούν να δουν σε ποιες εγκαταστάσεις έχουν πρόσβαση και να διαχειριστούν τα στοιχεία τους.
- **Ad-hoc Αιτήματα Πρόσβασης**: Δυνατότητα για τους χρήστες να υποβάλλουν ad-hoc αιτήματα πρόσβασης σε χώρους, τα οποία θα εγκρίνονται από τον admin. Αυτό προσφέρει ευελιξία και άμεση διαχείριση δικαιωμάτων πρόσβασης σε έκτακτες ανάγκες.
- **Dashboard Εισόδων**: Παροχή ενός ειδικά παραμετροποιημένου dashboard όπου οι χρήστες θα μπορούν να δουν το ιστορικό των access data και άλλες σχετικές πληροφορίες.
- **Ειδοποιήσεις**: Υποστήριξη για ειδοποιήσεις σχετικά με σημαντικά γεγονότα ή αλλαγές στην πρόσβαση βάση ενός notification μηχανισμού.

Κεφάλαιο 11.2.6: Επέκταση ML Model

Για την περαιτέρω βελτίωση της ακρίβειας και της ευελιξίας του συστήματος ανίχνευσης ανωμαλιών, θα μπορούσε να γίνει επέκταση της λειτουργικότητας με τη χρήση περισσότερων features. Η χρήση περισσότερων features, όπως μετρικές σχετικές με τον ρόλο του χρήστη, την συμπεριφορά του ή την τοποθεσία που εισέρχεται μπορεί να αυξήσει τη διακριτική ικανότητα του μοντέλου και να βελτιώσει την ανίχνευση πιο σύνθετων ή κρυφών ανωμαλιών.
Επιπλέον, η μετάβαση από το τρέχον Isolation Forest σε ένα νευρωνικό δίκτυο π.χ. Autoencoder ή Variational Autoencoder θα μπορούσε να προσφέρει μεγαλύτερη ικανότητα αναγνώρισης μη γραμμικών σχέσεων στα δεδομένα. Τα νευρωνικά δίκτυα μπορούν να μάθουν πιο σύνθετα πρότυπα και να προσαρμόζονται καλύτερα σε πολυδιάστατα δεδομένα, καθιστώντας τα ιδανικά για περιβάλλοντα με υψηλή πολυπλοκότητα και ποικιλία.

Με αυτές τις βελτιώσεις και επεκτάσεις, η εφαρμογή θα γίνει πιο ασφαλής, ευέλικτη και φιλική προς τον χρήστη, ικανοποιώντας τις αυξημένες απαιτήσεις και ανάγκες των χρηστών ενισχύοντας τη συνολική απόδοση της πλατφορμας.

Βιβλιογραφία:

I. Malin, M., & Berglund, E. (2016). *Spring Boot in Action: Spring Boot in Action*. Manning Publications.

II. Turnquist, G. L. (2022). *Learning Spring Boot 3.0: Simplify the development of production-grade applications using Java and Spring* (3rd ed.). Packt Publishing.

III. Obe, R., & Hsu, L. (2017). *PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database*. O'Reilly Media.

IV. Poulton, N. (n.d.). *Docker Deep Dive: Zero to Docker in a single book*. Nigel Poulton.

V. Sakkopoulos, E., Ioannou, Z.-M., & Viennas, E. (2018). Mobile personal information exchange over BLE. In *Proceedings of the 9th International Conference on Information, Intelligence, Systems and Applications (IISA 2018)* (pp. 1–8).

VI. Zafeiropoulou, A., & Sakkopoulos, E. (2023). Harmonising digital identity documents. In *Proceedings of the 14th International Conference on Information, Intelligence, Systems and Applications (IISA 2023)* (pp. 1–8).

VII. Moreno, R. T., Bernabé, J. B., Rodríguez, J. G., Frederiksen, T. K., Stausholm, M., Martínez, N., Sakkopoulos, E., Ponte, N., & Skarmeta, A. F. (2020). The OLYMPUS architecture: Oblivious identity management for private user-friendly services. *Sensors, 20*(3), 945. https://doi.org/10.3390/s20030945

VIII. Papadopoulou, E., & Sakkopoulos, E. (2023). Semantic cataloging of public services using basic government vocabularies and the data catalog vocabulary for a unified European digital market. In *Proceedings of the 14th International Conference on Information, Intelligence, Systems and Applications (IISA 2023)* (pp. 1–4).

Online Resources:

1. *Spring Boot official documentation*. Retrieved from https://spring.io/projects/spring-boot
2. *Spring Security official documentation*. Retrieved from https://spring.io/projects/spring-security
3. *PostgreSQL documentation*. Retrieved from https://www.postgresql.org/docs/16/index.html
4. *Docker documentation*. Retrieved from https://docs.docker.com
5. *Echarts documentation*. Retrieved from https://echarts.apache.org/en/index.html
6. *Quasar Framework documentation*. Retrieved from https://quasar.dev/docs
7. *Vue.js documentation*. Retrieved from https://vuejs.org/guide/introduction.html
8. *FastAPI documentation*. Retrieved from https://devdocs.io/fastapi/
9. *Nginx documentation*. Retrieved from https://docs.nginx.com
10. *Web NFC API documentation*. Retrieved from https://developer.mozilla.org/
11. *Prometheus documentation*. Retrieved from https://prometheus.io/docs/introduction/overview/
12. *Keycloak documentation*. Retrieved from https://www.keycloak.org/documentation
13. *Eureka documentation*. Retrieved from https://docs.spring.io/spring-cloud-netflix/
14. *Grafana documentation*. Retrieved from https://grafana.com/docs/grafana/latest/
15. *Isolation Forest guide*. Retrieved from https://www.datacamp.com/tutorial/isolation-forest