



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ
ΠΛΗΡΟΦΟΡΙΚΗΣ
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»**

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Εφαρμογή Ανάλυσης Συναισθήματος με Spring Emotion Analysis App with Spring
Ονοματεπώνυμο Φοιτητή	Στέργιος Τουλιούδας
Πατρώνυμο	Κωνσταντίνος
Αριθμός Μητρώου	ΜΠΠΛ2245
Επιβλέπων	Ευθύμιος Αλέπης Καθηγητής

Μάρτιος 2025

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Ευθύμιος Αλέπης
Καθηγητής

Μαρία Βίρβου
Καθηγήτρια

Κωνσταντίνα Χρυσafiάδη
Επίκουρος Καθηγήτρια

Περιεχόμενα

1 Περίληψη	4
1.1 Περίληψη	4
2 Εισαγωγή	4-6
2.1 Σκοπός	4-5
2.2 Προαπαιτούμενα	5
2.3 Τεχνολογίες	5
3 Related Work	5-6
4 Ανάλυση Απαιτήσεων	6-8
4.1 Λειτουργικές Απαιτήσεις	6-7
4.2 Μη Λειτουργικές Απαιτήσεις	7-8
4.3 Χρησιμότητα	8-9
4.4 Έρευνα αγοράς για παρόμοια ΠΣ	9
5 UML – Διαγράμματα	9-10
5.1 Περιγραφή – Δημιουργία	9
5.2 Διάγραμμα περίπτωσης χρήσης	9-10
6 Βάση Δεδομένων – MySQL	10-19
6.1 Περιγραφή	10-16
6.2 Διάγραμμα Οντοτήτων Συσχετίσεων (EER Diagram)	16-19
6.3 Συμπεράσματα	19
7 Προγραμματιστικό περιβάλλον	19-21
7.1 Java	19-120
7.2 Framework	20
7.3 Apache Maven	20-21
8 Spring Boot αρχιτεκτονική	21-27
8.1 Εισαγωγή	21
8.2 Βασικά Χαρακτηριστικά	21-22
8.3 Λειτουργία	22
8.4 Χαρακτηριστικά Εφαρμογής	22-24
8.5 Μοντέλο MVC (Model-View-Controller)	24-27
8.6 Επίπεδο DAO - Μοντέλο SOA	27
9 NLP (Natural Language Processor)	27-30
9.1 Περιγραφή	27-28
9.2 Χρησιμότητα	28-30
10 Εκτέλεση Εφαρμογής	30-46
10.1 Εκτέλεση ως απλός χρήστης – Στιγμιότυπα	32-37
10.2 Εκτέλεση ως διαχειριστής – Στιγμιότυπα	37-46
11 Συμπεράσματα	46-47
12 Βιβλιογραφία	47

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στους καθηγητές της τριμελούς εξεταστικής επιτροπής, την κ. Μαρία Βίρβου, την κ. Κωνσταντίνα Χρυσafiάδη και, φυσικά, τον επιβλέποντα καθηγητή της εργασίας αυτής, κ. Ευθύμιο Αλέπη, για την πολύτιμη καθοδήγησή τους καθ' όλη τη διάρκεια των μεταπτυχιακών μου σπουδών. Η συμβολή τους, τόσο μέσα από τη διδασκαλία των μαθημάτων τους όσο και μέσω του επιστημονικού τους έργου, υπήρξε καθοριστική για την ολοκλήρωση της παρούσας διπλωματικής εργασίας.

Ευχαριστώ ιδιαίτερα τον κ. Ευθύμιο Αλέπη για τη διδασκαλία του στον προγραμματισμό σε Java και την ανάπτυξη web εφαρμογών με Spring Boot, οι οποίες αποτέλεσαν θεμέλιο γνώσης για την παρούσα εργασία. Επίσης, εκφράζω την ευγνωμοσύνη μου προς την κ. Μαρία Βίρβου, της οποίας η διδασκαλία στα μαθήματα *Τεχνολογίες Λογισμικού* και *Αλληλεπίδραση Ανθρώπου-Υπολογιστή* με εφοδίασε με πολύτιμες γνώσεις για τη σχεδίαση και ανάπτυξη φιλικών προς τον χρήστη εφαρμογών. Ιδιαίτερες ευχαριστίες οφείλω και στην κ. Κωνσταντίνα Χρυσafiάδη, η οποία, μέσα από το μάθημα «*Γλώσσες Προγραμματισμού και Μεταγλωττιστές*», προσέφερε σημαντικές θεωρητικές και πρακτικές βάσεις, τις οποίες αξιοποίησα σε μεγάλο βαθμό στη μελέτη μου.

1 Περίληψη

1.1 Περίληψη

Η παρούσα μεταπτυχιακή εργασία παρουσιάζει μια διαδικτυακή εφαρμογή ανάλυσης συναισθημάτων, σχεδιασμένη να αξιολογεί την ψυχική κατάσταση του χρήστη. Η ανάλυση των συναισθημάτων αποτελεί ένα σύγχρονο πεδίο έρευνας στην Τεχνητή Νοημοσύνη (Artificial Intelligence – AI) και τη μηχανική μάθηση (Machine Learning – ML), με εφαρμογές στην ψυχική υγεία, τα μέσα κοινωνικής δικτύωσης και τη βελτίωση της εμπειρίας χρήστη (User Experience – UX) [2, 6, 8].

Η ανάλυση συναισθήματος έχει επίσης σημαντικό ρόλο στην εκπαίδευση, όπου χρησιμοποιείται σε ευφυή εκπαιδευτικά συστήματα για την προσαρμογή της εκπαιδευτικής εμπειρίας ανάλογα με τη συναισθηματική κατάσταση των μαθητών [1, 3]. Επίσης, οι τεχνολογίες AI και ML αξιοποιούνται σε εφαρμογές υγείας (m-health) και ανάλυσης δεδομένων ηλεκτρονικών ιατρικών φακέλων, υποστηρίζοντας αυτοματοποιημένες διαγνώσεις και παροχή εξατομικευμένων συμβουλών [4, 5].

Η ανάπτυξη της εφαρμογής στηρίζεται σε σύγχρονες τεχνολογίες διαδικτύου και διαδικτυακών εφαρμογών, αξιοποιώντας τεχνολογίες backend και frontend για τη διαχείριση και παρουσίαση των δεδομένων. Ειδικότερα, η χρήση τεχνολογιών Spring Boot [10] για το backend και σύγχρονων τεχνικών sentiment analysis [7, 9] βελτιώνει την απόδοση και την ακρίβεια του συστήματος. Η ανάλυση συναισθήματος αποτελεί ένα από τα πιο καινοτόμα πεδία εφαρμογής της Τεχνητής Νοημοσύνης, με εφαρμογές στην εκπαίδευση, την υγεία, αλλά και την επιχειρηματική ανάλυση [3, 5].

Αυτή η μεταπτυχιακή διατριβή παρουσιάζει μια διαδικτυακή εφαρμογή ανάλυσης συναισθημάτων, σχεδιασμένη να αξιολογεί την ψυχική κατάσταση του χρήστη. Μετά την ανάλυση της συναισθηματικής του κατάστασης, η εφαρμογή παρέχει εξατομικευμένες προτάσεις για τη βελτίωσή της και προτείνει ειδικούς ψυχικής υγείας που βρίσκονται στην περιοχή που δήλωσε ο χρήστης.

Επιπλέον, η εφαρμογή προσφέρει διάφορες λειτουργίες, όπως πρόσβαση σε πληροφορίες για εξειδικευμένους επαγγελματίες ψυχικής υγείας σε διαφορετικές περιοχές της Ελλάδας, τη δυνατότητα προγραμματισμού ραντεβού με τον ειδικό που έχει δώσει πρόσβαση στην πλατφόρμα, καθώς και την επιλογή αποστολής μηνύματος προς τον πάροχο υπηρεσιών.

Από την πλευρά της διαχείρισης, η πλατφόρμα επιτρέπει στον διαχειριστή να παρακολουθεί στατιστικά δεδομένα σχετικά με την ψυχική υγεία των εγγεγραμμένων χρηστών, να επιβεβαιώνει, να απορρίπτει ή να προγραμματίζει ραντεβού χειροκίνητα. Επιπλέον, η εφαρμογή περιλαμβάνει ολοκληρωμένα εργαλεία διαχείρισης για την αποδοτική διαχείριση των χρηστών και των δεδομένων τους.

2 Εισαγωγή

2.1 Σκοπός

Ο σκοπός της παρούσας εργασίας είναι η ανάπτυξη μίας διαδικτυακής εφαρμογής που θα παρέχει σε πραγματικό χρόνο, ανάλυση της ψυχικής κατάστασης ενός χρήστη με βάση την περιγραφή που ο ίδιος εισάγει. Η εφαρμογή δίνει τη δυνατότητα σε οποιονδήποτε να αποκτήσει γρήγορα και εύκολα μια αρχική εικόνα της ψυχικής του κατάστασης.

Παράλληλα, μπορεί να λειτουργήσει ως μια επιπλέον «γέφυρα» επικοινωνίας μεταξύ πελάτη και ειδικού, καθώς επιτρέπει στον χρήστη να λαμβάνει γνώση της συναισθηματικής του κατάστασης οποιαδήποτε στιγμή, να προτείνει ένα ραντεβού με τον ειδικό και να καταγράφει σκέψεις ή σχόλια που τον απασχολούν.

Επιπλέον, ο ειδικός μπορεί, μέσω της εφαρμογής, να παρακολουθεί την πορεία και τη συναισθηματική εξέλιξη των πελατών του, αποκτώντας έτσι μια πιο ολοκληρωμένη και σαφή εικόνα της ψυχικής τους κατάστασης.

2.2 Προαπαιτούμενα

Η εφαρμογή μπορεί να χρησιμοποιηθεί από οποιονδήποτε χρήστη μέσω υπολογιστή, με την προϋπόθεση ότι διαθέτει πρόσβαση στο διαδίκτυο. Για την πρόσβαση στην εφαρμογή, ο χρήστης θα πρέπει είτε να την έχει εγκατεστημένη στη συσκευή του είτε να γνωρίζει τη διεύθυνση του διακομιστή (server) όπου αυτή φιλοξενείται.

Θεωρείται δεδομένο ότι ο πάροχος της εφαρμογής, κατά την εκχώρηση δικαιωμάτων πρόσβασης στον χρήστη, έχει πραγματοποιήσει τις απαραίτητες ρυθμίσεις ώστε να του επιτρέπεται η σύνδεση τόσο από τον σταθερό του υπολογιστή όσο και από άλλες συσκευές.

Στο μέλλον, υπάρχει η δυνατότητα ανάπτυξης μιας αντίστοιχης εφαρμογής για κινητές συσκευές Android, ώστε να επιτρέπεται η χρήση της από smartphones και tablets, παρέχοντας μεγαλύτερη ευελιξία στους χρήστες.

2.3 Τεχνολογίες

Η ανάπτυξη της εφαρμογής πραγματοποιήθηκε χρησιμοποιώντας τη γλώσσα προγραμματισμού **Java** (έκδοση *OpenJDK 11.0.16.1*) και το προγραμματιστικό περιβάλλον **IntelliJ IDEA 2024.1.4 Ultimate Edition**.

Για τα frameworks, επιλέχθηκε το **Spring Boot**, αξιοποιώντας τα ακόλουθα εργαλεία: **Thymeleaf, DevTools, Spring Web, JPA** και **Stanford CoreNLP**.

Η βάση δεδομένων που χρησιμοποιήθηκε είναι η **MySQL**, με διαχειριστικό εργαλείο το **MySQL Workbench**.

3 Related Work

Η ανάλυση συναισθημάτων αποτελεί ένα από τα πλέον ενεργά ερευνητικά πεδία στην Τεχνητή Νοημοσύνη (AI) και τη Μηχανική Μάθηση (ML), με εφαρμογές που εκτείνονται από την ψυχική υγεία έως τη βελτίωση της εμπειρίας χρήστη (User Experience – UX). Η χρήση προηγμένων αλγορίθμων και τεχνικών μηχανικής μάθησης επιτρέπει την ακριβή κατηγοριοποίηση συναισθημάτων μέσω δεδομένων που προέρχονται από κείμενα, μέσα κοινωνικής δικτύωσης, ακόμη και φυσικές εκφράσεις των χρηστών [2, 6, 8].

Σύμφωνα με τη Virvou [2], η αλληλεπίδραση ανθρώπου-μηχανής μπορεί να ενισχυθεί μέσω της Τεχνητής Νοημοσύνης, επιτρέποντας στις εφαρμογές να προσαρμόζονται στις ανάγκες των χρηστών και να βελτιώνουν τη συνολική εμπειρία τους. Στον τομέα της εκπαίδευσης, έχουν αναπτυχθεί ευφυή συστήματα εκμάθησης που προσαρμόζουν το περιεχόμενο ανάλογα με τη συναισθηματική κατάσταση του χρήστη, γεγονός που έχει αποδειχθεί ότι βελτιώνει την αποτελεσματικότητα της μάθησης [3].

Η ανάλυση συναισθημάτων εφαρμόζεται επίσης στον τομέα της υγείας, όπου μπορεί να αξιοποιηθεί για την υποστήριξη διαγνώσεων και τη βελτίωση της ψυχικής υγείας των χρηστών μέσω προσωποποιημένων συστάσεων [1, 5]. Η μελέτη των Alerpis & Virvou [1] επικεντρώνεται στη χρήση συναισθηματικών πρακτόρων σε ευφυή συστήματα εκπαίδευσης, προτείνοντας τρόπους για την

αυτόματη αναγνώριση και απόκριση σε συναισθηματικές καταστάσεις των χρηστών. Επιπλέον, η ασφάλεια και η ιδιωτικότητα των δεδομένων στις εφαρμογές που επεξεργάζονται ευαίσθητες πληροφορίες χρήστη αποτελεί κρίσιμο ζήτημα, όπως αναφέρεται στη μελέτη των Papageorgiou et al. [5], όπου εξετάζονται τα θέματα προστασίας προσωπικών δεδομένων στις εφαρμογές υγείας.

Τέλος, η ανάπτυξη διαδικτυακών εφαρμογών ανάλυσης συναισθημάτων μπορεί να ωφεληθεί από σύγχρονες τεχνικές cloud computing και ανάλυσης δεδομένων, όπως αναφέρουν οι Riadsolh et al. [7] και Khan et al. [9]. Οι τεχνικές αυτές επιτρέπουν τη συλλογή και ανάλυση μεγάλων όγκων δεδομένων, βελτιώνοντας την ακρίβεια και την απόδοση των συστημάτων που βασίζονται στην AI.

Με βάση τα παραπάνω, η παρούσα εργασία αξιοποιεί τις εξελίξεις στον τομέα της ανάλυσης συναισθημάτων για τη δημιουργία μιας διαδικτυακής εφαρμογής που υποστηρίζει τη βελτίωση της ψυχικής υγείας των χρηστών, παρέχοντας παράλληλα χρήσιμες λειτουργίες, όπως η αναζήτηση ειδικών ψυχικής υγείας και ο προγραμματισμός συνεδριών.

Πώς λειτουργεί η ανάλυση συναισθήματος:

1) Συλλογή δεδομένων:

- Οι αξιολογήσεις πελατών λαμβάνονται από πλατφόρμες όπως ηλεκτρονικά καταστήματα, μέσα κοινωνικής δικτύωσης, αξιολογήσεις σε γνωστές πλατφόρμες (Google) κ.λπ.

2) Προκαταρκτική επεξεργασία κειμένου

- Αφαίρεση ειδικών χαρακτήρων, emojis, stopwords («και», «το», «να» κ.λπ.).

3) Ανάλυση συναισθήματος

- Σύγκριση των λέξεων με λεξικά συναισθήματος (πχ. AFINN, SentiWordNet).
- Μοντέλα «Machine Learning» με χρήση αλγορίθμων για ταξινόμηση της κριτικής ως θετική, αρνητική ή ουδέτερη.
- Υπολογισμός ενός σκορ συναισθήματος για κάθε αξιολόγηση.

4) Εξαγωγή Συμπερασμάτων

- Οι επιχειρήσεις αναλύουν το συνολικό ποσοστό θετικών/αρνητικών κριτικών.
- Μπορούν να εντοπιστούν προβλήματα σε προϊόντα ή υπηρεσίες.
- Υπολογισμός ενός σκορ συναισθήματος για κάθε αξιολόγηση.

Παράδειγμα:

Αξιολόγηση: «Το κινητό είναι απίστευτα γρήγορο, αλλά η μπαταρία του τελειώνει γρήγορα.»

Ανάλυση:

«απίστευτα γρήγορο» → Θετικό

«μπαταρία τελειώνει γρήγορα» → Αρνητικό

Τελικό σκορ: πχ 0.3 → ουδέτερο προς θετικό.

Παρότι η ανάλυση συναισθήματος παραμένει η ίδια, η παρούσα εργασία διαφοροποιείται σημαντικά από τις προαναφερθείσες προσεγγίσεις. Αναλυτικά, η παρούσα εφαρμογή χρησιμοποιεί **NLP** για συγκεκριμένες λειτουργίες (όπως ληματοποίηση, τμηματοποίηση λέξεων (tokenization), διαχωρισμός κειμένου σε προτάσεις, εξαγωγή σημείων στίξης) και βασίζεται σε εξειδικευμένες βιβλιοθήκες που έχουν φτιαχτεί για αναγνώριση ποικίλων συναισθημάτων. Αυτό έχει ως πλεονέκτημα ότι η **εφαρμογή εκτείνεται πέραν του εύρους των τριών μόνο συναισθημάτων (positive, negative, neutral)** και η συναισθηματική ανάλυση περιλαμβάνει τα συναισθήματα: **joy, sadness, anxiety, anger, fear, neutral** αλλά και όλους τους δυνατούς συνδυασμούς μεταξύ αυτών πχ **«anger_anxiety»** ή **«anger_anxiety_sadness»**. Αυτό οδηγεί σε μεγαλύτερη ακρίβεια όσον αφορά το εκτιμώμενο συναίσθημα και μπορεί να εξάγει σε **πραγματικό χρόνο** σαφώς καλύτερες διαπιστώσεις.

Επιπλέον, αντί να επικεντρώνεται σε γενικές ψυχολογικές τάσεις, η εφαρμογή έχει σχεδιαστεί για **προσωπική χρήση από ειδικούς ψυχικής υγείας**, προσφέροντας ένα εργαλείο που μπορεί να βοηθήσει στην εξαγωγή συμπερασμάτων για την ψυχολογική κατάσταση του ατόμου, ενώ παράλληλα επιτρέπει στον χρήστη να έχει μια άμεση και προσωποποιημένη ανατροφοδότηση σχετικά με τη συναισθηματική του διάθεση.

4 Ανάλυση Απαιτήσεων

4.1 Λειτουργικές Απαιτήσεις

Οι λειτουργικές απαιτήσεις (functional requirements) ενός πληροφοριακού συστήματος περιγράφουν τι πρέπει να κάνει το σύστημα. Πρόκειται για συγκεκριμένες λειτουργίες, χαρακτηριστικά και συμπεριφορές που πρέπει να υποστηρίζει η εφαρμογή για να ικανοποιήσει τις ανάγκες των χρηστών της. Οι λειτουργικές απαιτήσεις τις παρούσας εφαρμογής είναι οι εξής:

Ως χρήστης:

1) Εγγραφή Χρήστη

- Ο χρήστης ξεκινώντας η εφαρμογή έχει τη δυνατότητα να δημιουργήσει ένα λογαριασμό χρήστη(εάν φυσικά δεν έχει) που θα του δώσει τη δυνατότητα να εισέλθει στο Πληροφοριακό Σύστημα.

2) Ενημέρωση σχετικά με το σκοπό της εφαρμογής

- Πλοηγώντας στο σημείο «**About**» μπορεί να ενημερωθεί για τα χαρακτηριστικά της εφαρμογής και το τι ακριβώς προσφέρει, ώστε να αξιολογήσει εάν του είναι χρήσιμη ή όχι.

3) Ανάλυση Συναισθήματος σε πραγματικό Χρόνο

- Ο χρήστης εισάγει κείμενο και το σύστημα αναλύει τη συναισθηματική του κατάσταση.
- Εμφάνιση αποτελέσματος – προτάσεις για να βελτιώσει την υπάρχουσα κατάσταση
- Πληροφορίες για ειδικούς ψυχικής Υγείας στην περιοχή του χρήστη.

4) Πρόταση για ραντεβού

- Ο χρήστης έχει τη δυνατότητα να «**προτείνει**» ένα ραντεβού στον γιατρό ο οποίος διαχειρίζεται την εφαρμογή καταχωρώντας την ακριβή ημερομηνία και ώρα.

5) Σχόλια - Αξιολόγηση

- Η εφαρμογή δίνει τη δυνατότητα στο χρήστη να γράψει κάποια σχόλια τύπου αξιολόγησης με σκοπό τη βελτίωση της εφαρμογής.

6) Πληροφορίες για ειδικούς ψυχικής Υγείας σε διάφορες πόλεις της Ελλάδας

- Ο χρήστης μπορεί να αντλήσει πληροφορίες για όλους τους καταχωρημένους ειδικούς ψυχικής υγείας όπως το όνομα, την ειδικότητα, το τηλέφωνο, το email και την περιοχή.

7) Αποσύνδεση χρήστη

- Διακοπή σύνδεσης με δυνατότητα επανασύνδεσης.

Ως διαχειριστής:

1) Εγγραφή Χρήστη

2) Επικύρωση – Ακύρωση – Δημιουργία Ραντεβού με πελάτη

- Ο διαχειριστής της εφαρμογής δύναται να διαγράψει ένα προτεινόμενο ραντεβού πελάτη
- Ακόμη, μπορεί να δημιουργήσει εκ νέου ένα ραντεβού σε οποιαδήποτε ημερομηνία επιθυμεί.
- Σε περίπτωση που δε διαγράψει ένα προτεινόμενο ραντεβού πελάτη, αυτό παραμένει καταχωρημένο στη βάση.

3) Εξαγωγή Στατιστικών

- Ο διαχειριστής έχει τη δυνατότητα να δει κάποια στατιστικά που προκύπτουν από τις εγγραφές των χρηστών, όπως, «**το πιο κυρίαρχο συναίσθημα**», «το ποσοστό κυρίαρχου συναισθήματος ανά γένος» κ.ά.

4) Διαχείριση κατηγοριών

- Δυνατότητα προσθήκης – τροποποίησης – διαγραφής πελάτη/ασθενή.
- Δυνατότητα προσθήκης – τροποποίησης – διαγραφής λογαριασμού χρηστών.
- Δυνατότητα προσθήκης – τροποποίησης – διαγραφής καταχωρημένων γιατρών.

5) Αποσύνδεση

4.2 Μη Λειτουργικές Απαιτήσεις

Οι μη λειτουργικές απαιτήσεις της εφαρμογής περιλαμβάνουν πτυχές που δεν σχετίζονται άμεσα με τις λειτουργίες του συστήματος, αλλά επηρεάζουν την απόδοση, την ασφάλεια και την εμπειρία του χρήστη. Αναλυτικά αυτές είναι:

1) Απόδοση και Ταχύτητα

- Η ανάλυση του συναισθήματος πρέπει να γίνεται σε σχετικά μικρό χρονικό διάστημα ώστε να μη κουράζει και αποθαρρύνει το χρήστη.

- Η φόρτωση των στατιστικών επίσης.

2) Ασφάλεια και Προστασία Δεδομένων

- Πρόσβαση μόνο από εξουσιοδοτημένους χρήστες.

- Καταχώρηση στοιχείων σε ασφαλές σύστημα βάσεων δεδομένων(MySql) το οποίο απαιτεί τα δικά του διαπιστευτήρια.

3) Διαθεσιμότητα και Αξιοπιστία

- Η εφαρμογή πρέπει να είναι διαθέσιμη ανά πάσα ώρα και στιγμή(24/7).

- Ανθεκτικότητα σε υψηλό φόρτο χρηστών.

4) Επεκτασιμότητα

- Να υποστηρίζει μελλοντική προσθήκη βιβλιοθηκών για καλύτερη ανάλυση συναισθήματος

- Δυνατότητα προσθήκης περισσότερων στατιστικών για αποδοτικότερη αξιοποίηση των αποτελεσμάτων των χρηστών.

- Δυνατότητα υποστήριξης μεγαλύτερου αριθμού χρηστών χωρίς πτώση απόδοσης.

5) Φιλικότητα προς το Χρήστη

- Η διεπαφή πρέπει να είναι εύχρηστη και κατανοητή.

- Να παρέχει σαφείς οδηγίες χρήσης.

6) Συμβατότητα

- Να λειτουργεί σε διάφορους browsers και κινητές συσκευές.

4.3 Χρησιμότητα

Η εφαρμογή ανάλυσης συναισθήματος που αναπτύχθηκε έχει **πολλαπλές χρήσεις και οφέλη**, τόσο για τους **χρήστες** όσο και για τους **ειδικούς ψυχικής υγείας**. Ο συνδυασμός **ανάλυσης φυσικής γλώσσας (NLP)** και η εξαγωγή σε πραγματικό χρόνο εκτιμώμενου συναισθήματος παρέχει και στις δύο μεριές ένα πολύτιμο εργαλείο για τη βελτίωση της ψυχολογικής κατάστασης των χρηστών.

1. Χρησιμότητα για τους χρήστες

Η εφαρμογή μπορεί να λειτουργήσει ως ένα **προσωπικό εργαλείο αυτογνωσίας**, βοηθώντας τους χρήστες να κατανοήσουν καλύτερα τα συναισθήματά τους. Συγκεκριμένα, η εφαρμογή παρέχει:

- Επίγνωση της ψυχικής διάθεσης σε πραγματικό χρόνο.

- Παρακολούθηση της Συναισθηματικής Κατάστασης έτσι ώστε οι χρήστες να μπορούν να δουν πώς διαμορφώνεται η ψυχολογική τους κατάσταση μέσα στον χρόνο.

- Προτάσεις Βελτίωσης με τεχνικές χαλάρωσης τις οποίες προτείνει ο ειδικός ψυχικής υγείας.

- Πρόληψη Ψυχικών Διαταραχών. Αν ένας χρήστης εμφανίζει συχνά αρνητικές συναισθηματικές καταστάσεις, η εφαρμογή δίνει τη δυνατότητα μέσω των στατιστικών να μπορεί να τον προτρέψει να αναζητήσει επαγγελματική βοήθεια.

- Γρήγορη και εύκολη πρόσβαση στα ραντεβού. Οι πελάτες έχουν τη δυνατότητα ανά πάσα ώρα και στιγμή να προτείνουν ένα ραντεβού με τον ειδικό που τους παρακολουθεί, χωρίς να χρειάζεται να πάρουν τηλέφωνο στο ιατρείο ή ακόμη και να μεταβούν από εκεί.

2. Χρησιμότητα για τους ειδικούς

Οι ειδικοί ψυχικής υγείας (ψυχολόγοι, ψυχίατροι, σύμβουλοι) μπορούν να χρησιμοποιήσουν την εφαρμογή ως ένα **επιπλέον εργαλείο** στην πρακτική τους. Συγκεκριμένα η εφαρμογή τους παρέχει:

- **Συμπληρωματική Πληροφορία για τη Συναισθηματική Κατάσταση των Πελατών:**

Ο ειδικός μπορεί να βλέπει στατιστικά και τάσεις στη συναισθηματική κατάσταση των ασθενών του, αποκτώντας μια πιο ολοκληρωμένη εικόνα.

- **Εξατομίκευση Θεραπευτικής Προσέγγισης:** Μέσα από την ανάλυση των δεδομένων, μπορεί να προσαρμόσει καλύτερα τη θεραπευτική του προσέγγιση.

- **Διαχείριση Ραντεβού και Αλληλεπίδραση με τους Πελάτες:** Οι ειδικοί μπορούν να δέχονται αιτήματα για ραντεβού και να επικοινωνούν άμεσα μέσω email.

- **Διαχείριση πελατών και ειδικών:** Ο ειδικός που διαχειρίζεται την εφαρμογή μπορεί ανά πάσα ώρα και στιγμή να προσθέσει, να διαγράψει ή να ενημερώσει τα στοιχεία ενός πελάτη ή ακόμα και ενός καταχωρημένου ειδικού.

3. Μελλοντική Χρησιμότητα

Χρησιμοποιώντας κατάλληλα εργαλεία και βιβλιοθήκες με σκοπό τη βελτιστοποίηση του αλγορίθμου ανάλυσης συναισθήματος η παρούσα εφαρμογή θα μπορούσε να διευρύνει τη χρησιμότητά της και σε άλλους τομείς, όπως:

- **Εταιρικό Περιβάλλον:** Ανάλυση συναισθηματικής κατάστασης εργαζομένων για βελτίωση του εργασιακού περιβάλλοντος.
- **Εκπαίδευση:** Ανάλυση της συναισθηματικής κατάστασης των μαθητών για προσαρμογή της διδασκαλίας.
- **Εξυπηρέτηση Πελατών:** Χρήση της τεχνολογίας για ανάλυση συναισθημάτων σε αξιολογήσεις προϊόντων ή υποστήριξη πελατών(ήδη χρησιμοποιείται).

Μία συνηθισμένη προσέγγιση για τη δημιουργία ενός αλγορίθμου ανάλυσης συναισθήματος είναι με «**Μηχανική Μάθηση**». Η μηχανική μάθηση χρησιμοποιεί **εκπαιδευμένα μοντέλα** που μπορούν να προβλέψουν το συναίσθημα ενός κειμένου βασισμένα σε παραδείγματα από μεγάλα σύνολα δεδομένων. Φτιάχνοντας λοιπόν, ένα καλύτερα εκπαιδευμένο μοντέλο το οποίο θα εκτιμούσε με μεγαλύτερη ακρίβεια το συναίσθημα του χρήστη, η παρούσα εφαρμογή θα μπορούσε να αξιοποιηθεί και στους τομείς που αναφέρθηκαν. Διαθέσιμες βάσεις δεδομένων που χρησιμοποιούνται ευρέως στο διαδίκτυο είναι:

- **IMDB Dataset**, για κριτικές ταινιών
- **Kaggle Datasets**, διάφορες συλλογές δεδομένων για NLP
- **Amazon Reviews**, κριτικές προϊόντων από το Amazon

κ.ά.

4.4 Έρευνα αγοράς για παρόμοια ΠΣ

Η ανάλυση συναισθήματος έχει βρει εφαρμογές σε διάφορους τομείς, όπως το μάρκετινγκ, την εξυπηρέτηση πελατών και την κλινική ιατρική. Ωστόσο, η χρήση της για ιατρικούς σκοπούς, όπως η ανάλυση συναισθημάτων ασθενών μέσω κειμένου, είναι ακόμα σε πρώιμο στάδιο. Παρόλο που υπάρχουν εφαρμογές που επιτρέπουν στους χρήστες να καταγράφουν τα συναισθήματά τους, δεν είναι ευρέως διαδεδομένες εφαρμογές που να αναλύουν το κείμενο του χρήστη σε πραγματικό χρόνο για ιατρικούς σκοπούς. Η ανάπτυξη τέτοιων εφαρμογών θα μπορούσε να προσφέρει πολύτιμα εργαλεία για επαγγελματίες υγείας, βοηθώντας στην παρακολούθηση και κατανόηση της συναισθηματικής κατάστασης των ασθενών τους.

5 UML – Διαγράμματα

5.1 Περιγραφή

Η Ενοποιημένη Γλώσσα Μοντελοποίησης (Unified Modeling Language) UML είναι μια γραφική γλώσσα γενικού σκοπού, η οποία χρησιμοποιείται για τον προσδιορισμό, οπτικοποίηση, ανάπτυξη και τεκμηρίωση των κατασκευασμάτων ενός πληροφοριακού συστήματος. Στο πλαίσιο της παρούσας εργασίας, θα παρουσιαστούν και θα αναλυθούν τέσσερα βασικά UML διαγράμματα, τα οποία συμβάλλουν στην καλύτερη κατανόηση της λειτουργίας της εφαρμογής:

- **Διάγραμμα Περιπτώσεων Χρήσης(Use Case Diagram)**, το οποίο περιγράφει τις βασικές αλληλεπιδράσεις των χρηστών με το σύστημα, καταγράφοντας τα κύρια σενάρια χρήσης.
- **Διάγραμμα Κλάσεων(Class Diagram)**, το οποίο αποτυπώνει τη δομή της εφαρμογής, τις κλάσεις, τις ιδιότητές τους και τις μεταξύ τους σχέσεις.
- **Διάγραμμα Ακολουθίας (Sequence Diagram)**, το οποίο αναπαριστά τη ροή επικοινωνίας μεταξύ των διάφορων αντικειμένων του συστήματος για συγκεκριμένες λειτουργίες.
- **Διάγραμμα Δραστηριοτήτων (Activity Diagram)**, το οποίο δείχνει τη ροή εργασιών και τα διαφορετικά μονοπάτια εκτέλεσης μιας διαδικασίας μέσα στην εφαρμογή.

5.2 Διάγραμμα Περιπτώσεων Χρήσης



Εικόνα 1: Use Case Diagram

6 Βάση Δεδομένων – MySQL

6.1 Περιγραφή

Στην εφαρμογή (και γενικότερα στις εφαρμογές **Spring Boot**), η σύνδεση με τη βάση δεδομένων επιτυγχάνεται μέσω του **JPA (Jakarta Persistence API)** και της υλοποίησής του, όπως το **Hibernate**. Το **JPA** είναι μια προδιαγραφή (specification) που ορίζει το πώς πρέπει να γίνεται η **αντικειμενοστρεφής χαρτογράφηση (Object-Relational Mapping - ORM)**, ενώ το **Hibernate** είναι μια **υλοποίηση** αυτής της προδιαγραφής που εκτελεί τις λειτουργίες και συνδέεται με τη βάση.

Πως γίνεται η σύνδεση

i) Ορισμός των Εξαρτήσεων (Dependencies)

Για να λειτουργήσει η σύνδεση, πρέπει να υπάρχουν οι κατάλληλες βιβλιοθήκες στο **pom.xml (Maven)**.

```

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>

  <dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <scope>runtime</scope>
  </dependency>

```

Εικόνα 2: Dependencies

- **spring-boot-starter-data-jpa**: Περιλαμβάνει όλη τη λειτουργικότητα για **JPA** και **Hibernate**.
- **mysql-connector-j**: Οδηγός (JDBC driver) για τη σύνδεση με **MySQL**.

ii) Ρύθμιση της Σύνδεσης στη Βάση Δεδομένων

Η σύνδεση διαμορφώνεται στο “**application.properties**” που βρίσκεται στο φάκελο «resources».

```

# Στοιχεία σύνδεσης με τη MySQL
spring.datasource.url=jdbc:mysql://localhost:3306/emotion_analysis
spring.datasource.username=springstudent
spring.datasource.password=springstudent
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

# Χρήση του Hibernate μέσω JPA
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect

```

Εικόνα 3: application.properties

iii) Δημιουργία Οντοτήτων

Παράδειγμα → Κλάση «Patient»:

```

@Entity
@Table(name = "patient")
public class Patient {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name = "first_name") 3 usages
    private String firstName;

    @Column(name = "last_name") 3 usages
    private String lastName;

    @Column(name = "age") 3 usages
    private int age;

    @Column(name = "gender") 3 usages
    private String gender;

    @Column(name = "comment", columnDefinition = "TEXT") 3 usages
    private String comment;

    @Column(name = "registration_day") 5 usages
    private LocalDateTime registration_day;

    // Change to ManyToOne relationship
    @ManyToOne(cascade = CascadeType.MERGE) 3 usages
    @JoinColumn(name = "location_id") // This column in Patient table references Location
    private Location location;

    @ManyToMany(cascade = CascadeType.MERGE) 3 usages
    @JoinTable(
        name = "patient_sentiment", // Join table name
        joinColumns = @JoinColumn(name = "patient_id"), // Foreign key for Patient
        inverseJoinColumns = @JoinColumn(name = "sentiment_id") // Foreign key for Sentiment
    )
    private Set<Sentiment> sentiments = new HashSet<>(); // Initialize with an empty Set

    // New OneToOne relationship with User
    @ManyToOne(cascade = CascadeType.MERGE) 4 usages
    @JoinColumn(name = "user_id", referencedColumnName = "id") // Foreign key reference to User
    private User user;

    public Patient() {}
}

```

Εικόνα 4: class Patient

Χαρακτηριστικά κλάσης και των **'annotations(@)'**:

Ορισμός της οντότητας (Entity) και του πίνακα στη βάση δεδομένων

- **@Entity**: Δηλώνει ότι η κλάση Patient είναι μία οντότητα που θα αντιστοιχιστεί σε ένα πίνακα στη βάση δεδομένων.
- **@Table(name = "patient")**: Ορίζει το όνομα του πίνακα που θα δημιουργηθεί στη βάση δεδομένων

Όταν η εφαρμογή εκκινήσει, η **Spring Boot** και το **Hibernate** θα αναγνωρίσουν αυτή την κλάση ως μία οντότητα και θα δημιουργήσουν αυτόματα πίνακα "patient" στη βάση MySQL.

Πρωτεύον κλειδί (Primary key) και αυτόματη αύξηση ID:

```

@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private int id;

```

Εικόνα 5: Patient Id

- **@id**: Ορίζει το πεδίο id ως πρωτεύον κλειδί της οντότητας.
- **@GeneratedValue(strategy = GenerationType.IDENTITY)**: Δηλώνει ότι το πεδίο **id** θα δημιουργείται **αυτόματα** από τη βάση δεδομένων και θα αυξάνεται κατά 1 σε κάθε νέα εγγραφή

Ορισμός των στηλών (Columns)

```
@Column(name = "first_name") 3 usages
private String firstName;
```

Εικόνα 6

- **@Column(name = "first_name")**: Ορίζει ότι η μεταβλητή **firstName** αντιστοιχεί στη στήλη **first_name** του πίνακα **patient**.
- Ομοίως άλλες μεταβλητές όπως **lastName**, **gender**, **age** κ.λπ., ακολουθούν την ίδια λογική.

Διαχείριση μεγάλων κειμένων με “Text”

```
@Column(name = "comment", columnDefinition = "TEXT") 3 usages
private String comment;
```

Εικόνα 7

- **columnDefinition = "TEXT"**: Δηλώνει ότι η στήλη **comment** θα αποθηκεύει μεγάλα κείμενα ως τύπο **TEXT** στη βάση MySQL.
- Αυτό γίνεται γιατί ο τύπος **VARCHAR** έχει περιορισμό στο μήκος των χαρακτήρων.

Συσχέτιση ManyToOne με τον πίνακα (Location)

```
@ManyToOne(cascade = CascadeType.MERGE) 3 usages
@JoinColumn(name = "location_id")
private Location location;
```

Εικόνα 8: Σύνδεση “Ένα προς Πολλά” με τον πίνακα «Location» μέσω του «location_id»

- Ο κάθε ασθενής ανήκει σε μία τοποθεσία (Location).
- Η στήλη **location_id** είναι το **ξένο κλειδί (foreign key)** που συνδέει τον πίνακα **patient** με τον πίνακα **location**
- Το **CascadeType.MERGE** επιτρέπει τη συγχώνευση των αλλαγών μεταξύ των σχετιζόμενων αντικειμένων.

Συσχέτιση «ManyToMany» με τον πίνακα Sentiment

```

@ManyToOne(cascade = CascadeType.MERGE) 3 usages
@JoinTable(
    name = "patient_sentiment", // Join table name
    joinColumns = @JoinColumn(name = "patient_id"), // Foreign key for Patient
    inverseJoinColumns = @JoinColumn(name = "sentiment_id") // Foreign key for Sentiment
)
private Set<Sentiment> sentiments = new HashSet<>(); // Initialize with an empty Set

```

Εικόνα 9: Πολλά προς πολλά συσχέτιση με τον πίνακα «Sentiment»

- Οι ασθενείς μπορούν να έχουν **πολλαπλά συναισθήματα** και κάθε συναίσθημα μπορεί να αφορά **πολλούς ασθενείς** (σχέση **πολλοί-προς-πολλούς**).
- Ορίζεται ένας **ενδιάμεσος πίνακας patient_sentiment** για τη διαχείριση αυτής της σχέσης.
- Η στήλη **patient_id** συνδέει τον ασθενή με το συναίσθημα.
- Η στήλη **sentiment_id** συνδέει το συναίσθημα με τους ασθενείς που το έχουν βιώσει.

Συσχέτιση ManyToOne με τον πίνακα User

```

@ManyToOne(cascade = CascadeType.MERGE) 4 usages
@JoinColumn(name = "user_id", referencedColumnName = "id")
private User user;

```

Εικόνα 10: Συσχέτιση ένα προς πολλά με τον πίνακα «User» μέσω του «user_id»

- Ορίζει τη σχέση μεταξύ **Patient** και **User** (κάθε ασθενής συνδέεται με έναν χρήστη).
- Η στήλη «user_id» συνδέεται με τη στήλη «id» του πίνακα «user».

Όταν εκκινήσει η εφαρμογή, η **Spring Boot** και το **Hibernate** θα μετατρέψουν αυτόματα την κλάση Patient στον ακόλουθο πίνακα στη βάση MySQL:

```

CREATE TABLE patient (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(255),
    last_name VARCHAR(255),
    age INT,
    gender VARCHAR(255),
    comment TEXT,
    registration_day DATETIME,
    location_id INT,
    user_id INT,
    FOREIGN KEY (location_id) REFERENCES location(id),
    FOREIGN KEY (user_id) REFERENCES user(id)
);

```

Εικόνα 11: Δημιουργία πίνακα

iv) Δημιουργία Repository για CRUD Λειτουργίες

```
@Repository 3 usages ⚡ Stergios9 *
public interface PatientRepository extends JpaRepository<Patient, Integer> {

    List<Patient> findAll(); ⚡ Stergios9

    Patient save(Patient patient); new *

    void deleteById(Integer integer); 2 usages new *

    Patient findById(int id); new *
```

Εικόνα 12: CRUD λειτουργίες στο Repository της οντότητας «Patient»

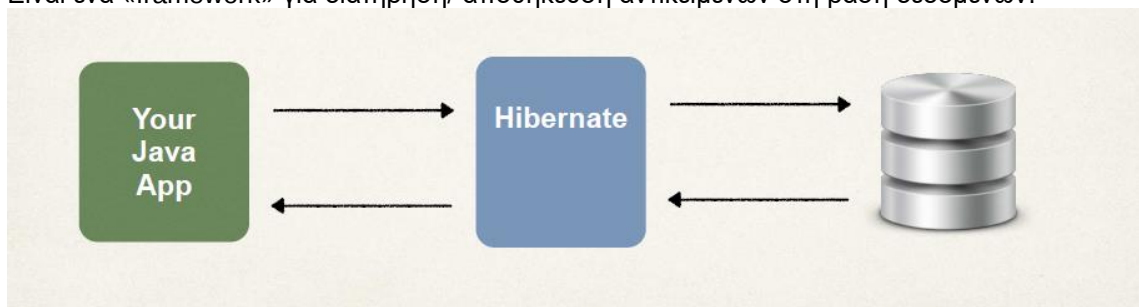
- Το **JpaRepository**: Παρέχει έτοιμες μεθόδους όπως save(), findById(), deleteById(), findAll() κ.ά.
- Για εξειδικευμένα ερωτήματα, όπως για παράδειγμα, εάν θέλουμε να βρούμε τη λίστα των ασθενών με βάση το γένος(gender) η δομή είναι αυτή:

```
@Query("SELECT p FROM Patient p WHERE p.gender = :gender") 1 usage ⚡ Stergios9
List<Patient> findByGender(@Param("gender") String gender);
```

Εικόνα 13: Costume query για στατιστικό αποτέλεσμα

Τι είναι το «Hibernate»

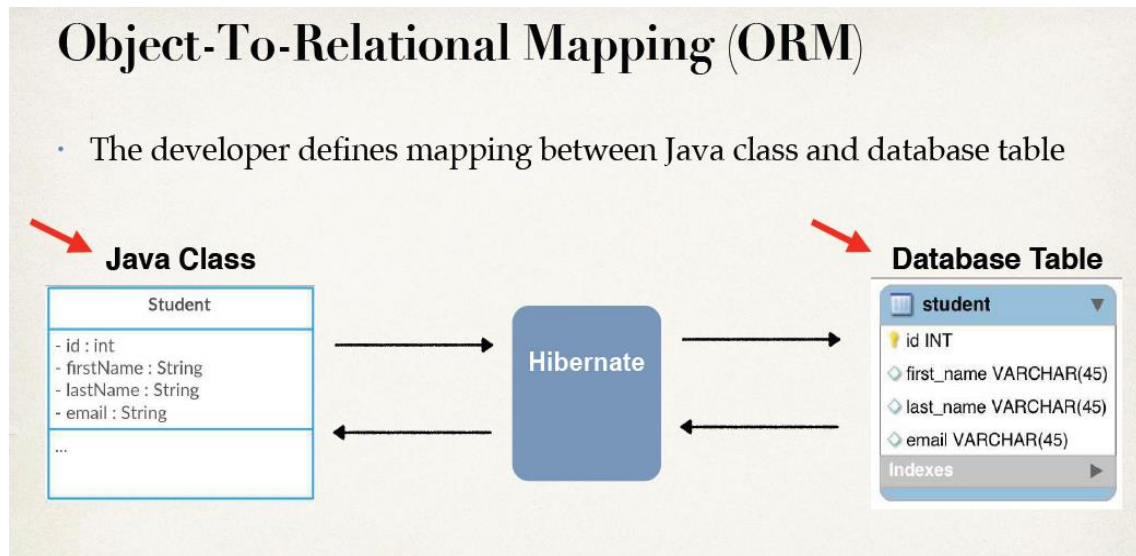
- ✓ Είναι ένα «framework» για διατήρηση/ αποθήκευση αντικειμένων στη βάση δεδομένων.



Εικόνα 14

Πλεονεκτήματα του **Hibernate**

- ✓ Το **Hibernate** μειώνει το σύνολο του κώδικα που πρέπει να γράψεις.
- ✓ Το **Hibernate** παρέχει το **Object -to-Relational Mapping(ORM)**.



Εικόνα 15: Λειτουργία του Hibernate

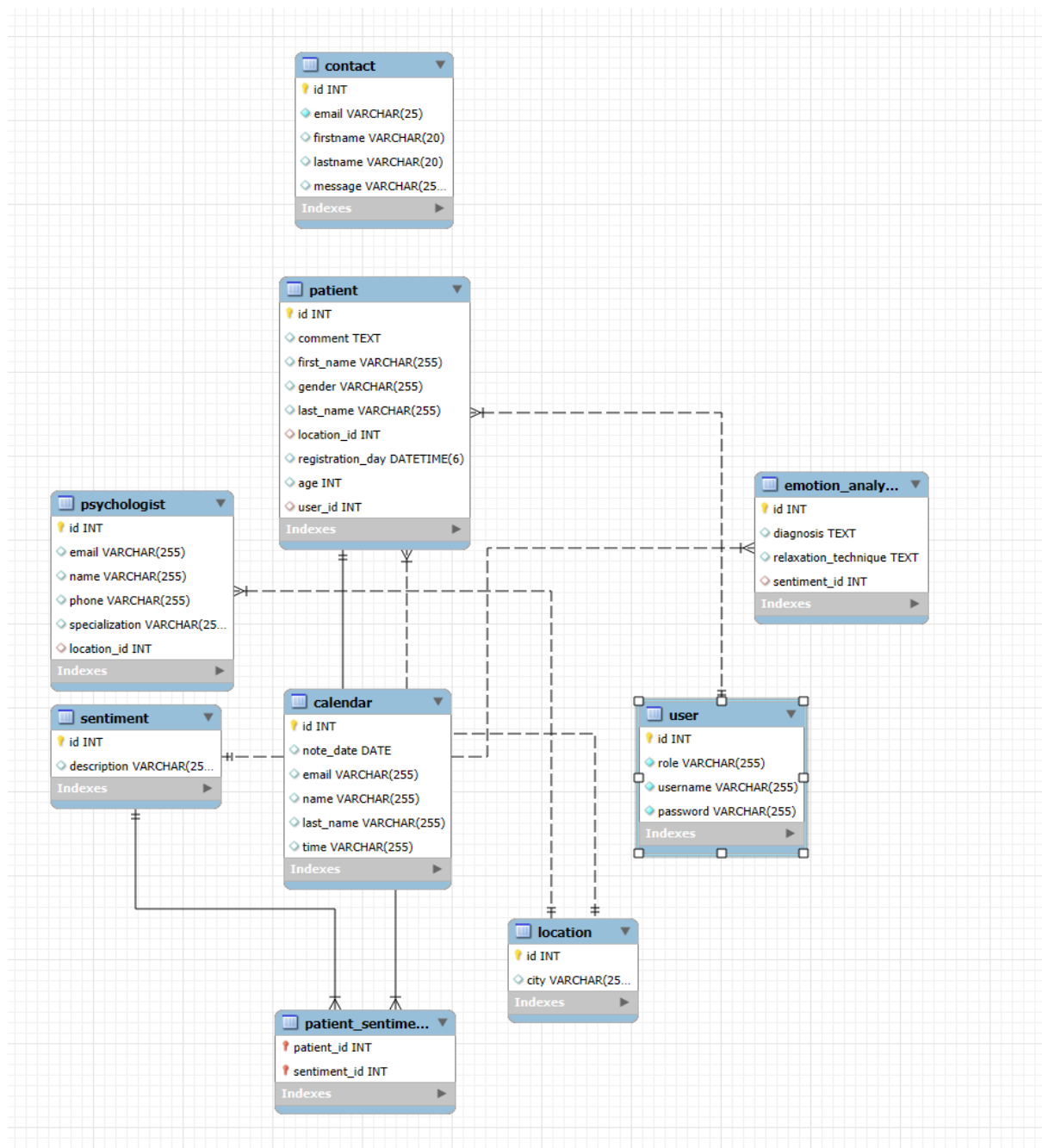
Παράδειγμα:

Στην εφαρμογή, η κλάση **Patient** αντιπροσωπεύει έναν ασθενή/χρήστη του συστήματος και είναι μια οντότητα (Entity) που αποθηκεύεται στη βάση δεδομένων. Η **Spring Boot**, σε συνδυασμό με το **Spring Data JPA**, επιτρέπει την αυτόματη δημιουργία του αντίστοιχου πίνακα στη βάση δεδομένων

Τι είναι το «JPA»

- ✓ Το **Jakarta Persistence API (JPA)**, παλαιότερα γνωστό ως **Java Persistence API**, είναι ένα πρότυπο για τη διαχείριση δεδομένων σε εφαρμογές Java.
- ✓ Αποτελεί το πρότυπο API για τη χαρτογράφηση αντικειμένων σε σχεσιακές βάσεις δεδομένων (Object-Relational Mapping - ORM).
- ✓ Το JPA **δεν είναι μία υλοποίηση**, αλλά **μόνο μια προδιαγραφή** που καθορίζει τον τρόπο διαχείρισης της αποθήκευσης δεδομένων.
- ✓ Ορίζει **ένα σύνολο από διεπαφές (interfaces)** που πρέπει να ακολουθούν οι υλοποιήσεις του.
- ✓ Για να χρησιμοποιηθεί, απαιτείται μια **συγκεκριμένη υλοποίηση**, όπως το **Hibernate**, το **EclipseLink** ή το **OpenJPA**

6.2 Διάγραμμα Οντοτήτων Συσχετίσεων (EER Diagram)



Εικόνα 16: EER diagram

Στην εικόνα 16 αναπαριστάται γραφικά η δομή της βάσης δεδομένων της εφαρμογής, η οποία περιγράφει τις οντότητες(entities) που περιλαμβάνονται στο σύστημα, τις σχέσεις μεταξύ τους και τα χαρακτηριστικά(attributes) που τις συνοδεύουν.

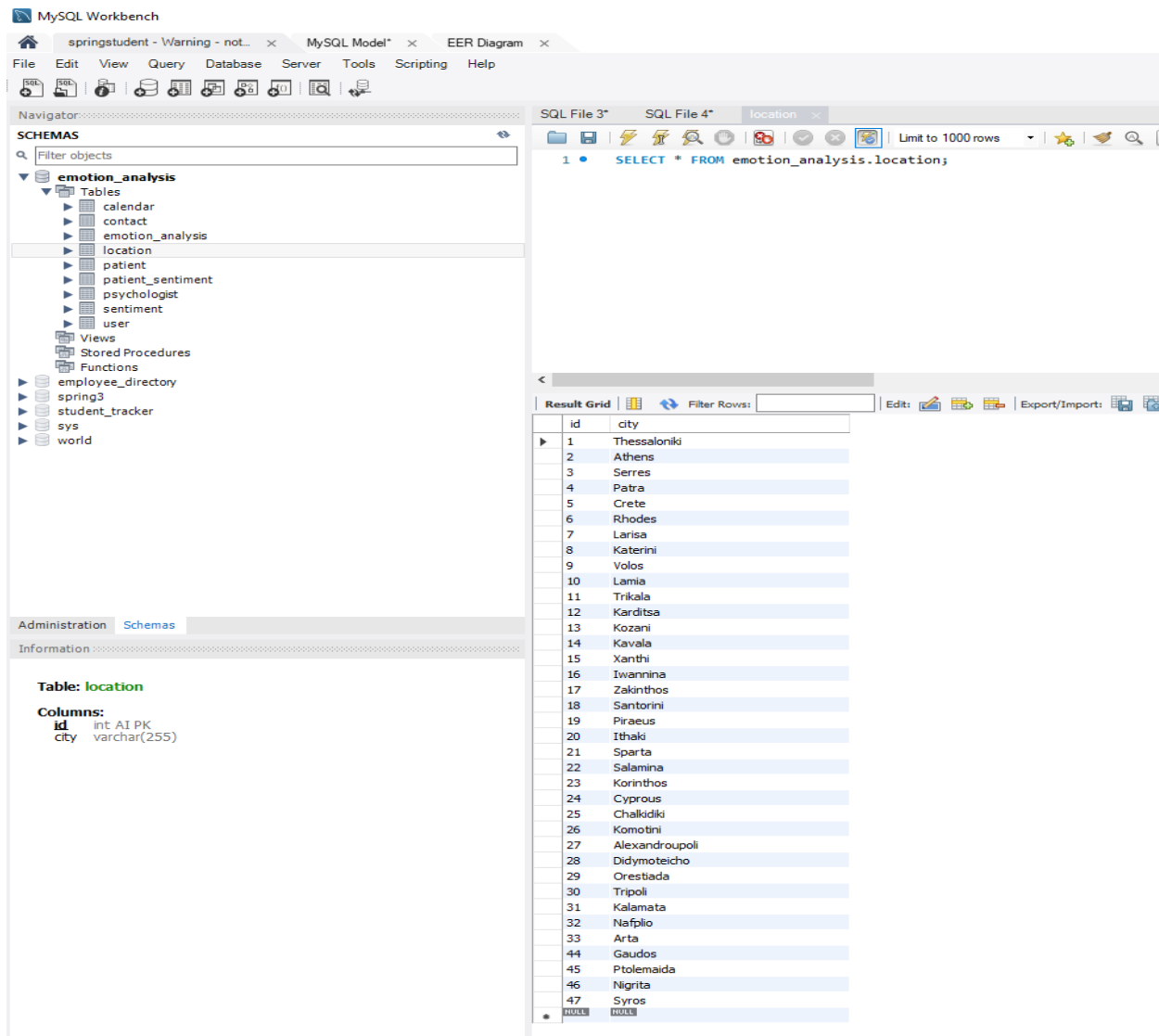
Βασικά Στοιχεία του EER Διαγράμματος

1. **Οντότητες (Entities)** → Αντιπροσωπεύουν αντικείμενα ή έννοιες με μοναδική υπόσταση. Στην παρούσα εργασία οι οντότητες είναι οι εξής:

- ✓ **patient** : αντιπροσωπεύει τον χρήστη ο οποίος είναι ο ασθενής
- ✓ **psychologist** : αντιπροσωπεύει τον ειδικό ψυχικής υγείας
- ✓ **user**: αντιπροσωπεύει ένα χρήστη με το δικούς του κωδικούς και ρόλο
- ✓ **calendar**: αντιπροσωπεύει τα ραντεβού

- ✓ **contact:** αντιπροσωπεύει τις αξιολογήσεις που γίνονται από τους χρήστες
- ✓ **sentiment:** αντιπροσωπεύει τα συναισθήματα που υπάρχουν καταχωρημένα στη βάση
- ✓ **location:** αντιπροσωπεύει όλες τις καταχωρημένες περιοχές στην Ελλάδα
- ✓ **Emotion_analysis:** αντιπροσωπεύει τις διαγνώσεις και τεχνικές που είναι καταχωρημένες ανά συναισθημα

Ακολουθεί στιγμιότυπο της βάσης δεδομένων όπου αριστερά απεικονίζονται όλοι οι πίνακες που έχουμε δημιουργήσει:



Εικόνα 17: Βάση δεδομένων εφαρμογής

Εάν παρατηρήσουμε προσεκτικά τους πίνακες, βλέπουμε ότι υπάρχει ο πίνακας «**patient_sentiment**». Ο πίνακας αυτός δεν αποτελεί μία ξεχωριστή οντότητα όπως όλοι οι υπόλοιποι, αλλά δημιουργείται αυτόματα από τη σχέση «ManyToMany» των οντοτήτων «patient» και «sentiment». Για να το υλοποιήσουμε αυτό προσδιορίσαμε τη σχέση και στις δύο οντότητες ως εξής:

i) Οντότητα Patient:

```
@ManyToMany(cascade = CascadeType.MERGE) 3 usages
@JoinTable(
    name = "patient_sentiment", // Join table name
    joinColumns = @JoinColumn(name = "patient_id"), // Foreign key for Patient
    inverseJoinColumns = @JoinColumn(name = "sentiment_id") // Foreign key for Sentiment
)
private Set<Sentiment> sentiments = new HashSet<>(); // Initialize with an empty Set
```

Εικόνα 18: Πολλά προς πολλά συσχέτιση στην κλάση (Model) Patient

ii) Οντότητα Sentiment:

```
@ManyToMany(mappedBy = "sentiments", cascade = CascadeType.ALL)
private Set<Patient> patients = new HashSet<>();
```

Εικόνα 19: Πολλά προς πολλά συσχέτιση στην κλάση «Sentiment»

2. **Συσχετίσεις (Relationships)** → Δείχνουν πώς οι οντότητες σχετίζονται μεταξύ τους (π.χ. "Ένας χρήστης κλείνει ένα ραντεβού").
3. **Χαρακτηριστικά (Attributes)** → Οι ιδιότητες μιας οντότητας (π.χ. "Όνομα", "Ηλικία", "Ημερομηνία Ραντεβού").
4. **Πρωτεύον κλειδί (Primary key)** → Ένα μοναδικό χαρακτηριστικό που προσδιορίζει κάθε οντότητα (π.χ. "ID Χρήστη").
5. **Συσχετίσεις πληθικότητας (cardinality)** → Δηλώνουν αν μια σχέση είναι 1:1, 1:N ή N:M (π.χ. "Ένας γιατρός μπορεί να έχει πολλά ραντεβού, αλλά ένα ραντεβού ανήκει σε έναν μόνο γιατρό").

6.3 Συμπέρασμα

Με τη χρήση του **Spring Boot** και του **Hibernate**, ο ορισμός των οντοτήτων (Entities) και των σχέσεων τους μετατρέπεται αυτόματα σε πίνακες MySQL χωρίς να χρειάζεται να γράψουμε χειροκίνητα SQL ερωτήματα. Τα σχόλια **@Entity**, **@Table**, **@Column**, **@ManyToOne**, **@ManyToMany** διασφαλίζουν ότι η βάση δεδομένων σχεδιάζεται σωστά και λειτουργεί αποδοτικά, διατηρώντας τις σχέσεις των δεδομένων οργανωμένες.

Επιπλέον, η διαχείριση της βάσης γίνεται με πολύ εύκολο και γρήγορο τρόπο μέσω των «Repositories» αφού το «JpaRepository» παρέχει έτοιμες μεθόδους για τις οποίες δε χρειάζεται να γράψουμε ερωτήματα SQL.

7 Προγραμματιστικό Περιβάλλον**7.1 JAVA**

Για την ανάπτυξη της εφαρμογής επιλέχθηκε η "Java", καθώς αποτελεί μία από τις πιο διαδεδομένες και ισχυρές γλώσσες προγραμματισμού, ειδικά στον χώρο των διαδικτυακών εφαρμογών και των επιχειρησιακών συστημάτων. Η Java είναι μία αντικειμενοστραφής, ουδέτερη ως προς την πλατφόρμα, ασφαλής γλώσσα που έχει σχεδιαστεί για να είναι πιο εύκολη στην εκμάθηση και δυσκολότερη στην κατάχρηση από ότι η "C++".

Πλεονεκτήματα της Java στην ανάπτυξη διαδικτυακών εφαρμογών

1. **Αρχή "Write Once, Run Anywhere" (WORA)**
Η Java είναι μια πλατφορμοανεξάρτητη γλώσσα, δηλαδή ο κώδικας μεταγλωττίζεται σε bytecode, που μπορεί να εκτελεστεί από οποιαδήποτε συσκευή, λόγω του Java Virtual Machine (JVM). Αυτό προσφέρει ευκολία στη μετεγκατάσταση εφαρμογών μεταξύ διαφορετικών συστημάτων και διακομιστών.
2. **Αντικειμενοστραφής Προγραμματισμός (OOP)**
Η Java βασίζεται στον αντικειμενοστραφή προγραμματισμό, επιτρέποντας επαναχρησιμοποίηση κώδικα, καλύτερη οργάνωση και ευκολία στη συντήρηση του λογισμικού.

3. Υποστήριξη για Web Development με Spring Boot

Η Java διαθέτει ισχυρά frameworks για την ανάπτυξη διαδικτυακών εφαρμογών, με πιο δημοφιλές το Spring Boot. Το Spring Boot επιτρέπει τη γρήγορη ανάπτυξη RESTful APIs, τη διαχείριση δεδομένων με JPA/Hibernate και τη διασύνδεση με βάσεις δεδομένων, όπως η MySQL.

4. Ασφάλεια & Αξιοπιστία

Η Java παρέχει ισχυρούς μηχανισμούς ασφαλείας, όπως διαχείριση μνήμης, garbage collection, ενσωματωμένα API κρυπτογράφησης και προστασία από επιθέσεις τύπου buffer overflow, καθιστώντας την κατάλληλη για εφαρμογές υψηλών απαιτήσεων.

5. Υποστήριξη Πολυνηματικότητας (Multithreading)

Η Java υποστηρίζει ταυτόχρονη εκτέλεση διεργασιών (multithreading), βελτιώνοντας τις επιδόσεις και την απόκριση των εφαρμογών. Αυτό είναι κρίσιμο για διαδικτυακές εφαρμογές που εξυπηρετούν πολλαπλούς χρήστες ταυτόχρονα.

6. Μεγάλη Κοινότητα & Βιβλιοθήκες

Ως μία από τις δημοφιλέστερες γλώσσες προγραμματισμού, η Java διαθέτει τεράστια κοινότητα προγραμματιστών, εκτενή τεκμηρίωση, καθώς και πολλές έτοιμες βιβλιοθήκες και εργαλεία που διευκολύνουν την ανάπτυξη εφαρμογών.

7.2 Framework

Το framework (πλαίσιο εργασίας) αποτελεί μια τυποποιημένη ομάδα αρχών, πρακτικών και κριτηρίων για την αντιμετώπιση προβλημάτων. Στην πράξη, είναι Δομημένα (σε φακέλους) πακέτα κλάσεων – αρχείων κώδικα τα οποία μπορούν να χρησιμοποιηθούν για να φτιάξουμε εφαρμογές και εξυπηρετούν στην παροχή κοινής δομής κλάσεων-αρχείων κώδικα στη λογική της επαναχρησιμοποίησης.

Frameworks που χρησιμοποιούνται στην εφαρμογή

Τα κύρια frameworks που αξιοποιούνται είναι:

1. Spring Boot

Το Spring Boot είναι ένα **backend framework** που βασίζεται στο **Spring Framework** και χρησιμοποιείται για τη δημιουργία αποδοτικών, επεκτάσιμων και εύκολα διαχειρίσιμων web εφαρμογών.

2. Spring Data JPA & Hibernate

Το **Spring Data JPA** είναι ένα εργαλείο που επιτρέπει την εύκολη αλληλεπίδραση με βάσεις δεδομένων, κάνοντας χρήση του **Java Persistence API (JPA)**. Το **Hibernate** είναι η default υλοποίηση του **JPA** στο Spring Boot.

Πλεονεκτήματα:

- ✓ Αυτόματη δημιουργία SQL ερωτημάτων, μειώνοντας την ανάγκη για χειροκίνητη SQL.
- ✓ Mapping αντικειμένων Java σε πίνακες βάσης δεδομένων (ORM), κάνοντας πιο εύκολη τη διαχείριση δεδομένων.
- ✓ Υποστήριξη caching για βελτίωση της απόδοσης.
- ✓ Σχέσεις μεταξύ οντοτήτων (OneToMany, ManyToOne, ManyToMany, OneToOne) για αναπαράσταση πολύπλοκων δεδομένων.

3. Thymeleaf

Το Thymeleaf είναι ένα template engine που χρησιμοποιείται για τη δημιουργία δυναμικών HTML σελίδων στο backend.

Πλεονεκτήματα:

- ✓ Εύκολη ενσωμάτωση με Spring Boot.
- ✓ Υποστήριξη dynamic content χωρίς ανάγκη JavaScript.
- ✓ Γρήγορη απόδοση και χαμηλή κατανάλωση πόρων.

7.3 Apache Maven

Το **Apache Maven** είναι ένα **εργαλείο διαχείρισης** έργων λογισμικού το οποίο βασίζει την λειτουργία του στην χρήση ενός **Project Object Model (POM)**. Το POM είναι ένα XML αρχείο από το οποίο το Maven μπορεί κεντρικώς να οργανώνει και να διαχειρίζεται την δομή όλων των συστατικών μίας εφαρμογής. Στο αρχείο POM δηλώνονται οι βιβλιοθήκες (π.χ. jars) και εκδόσεις τους που πρέπει να

συνοδεύουν την εφαρμογή, καθορίζονται ποια συστατικά της εφαρμογής θα γίνουν build και ποια όχι, το είδος του αρχείου που θα παραχθεί (π.χ. jar, war), η έκδοση του λογισμικού και γενικότερα μία πληθώρα διαφόρων ενεργειών διαχείρισεως της εφαρμογής. Ένα σημαντικό πλεονέκτημα του Maven είναι ότι μπορεί να χρησιμοποιεί μία κεντρική αποθήκη βιβλιοθηκών (Repository) και κάθε βιβλιοθήκη που υπάρχει εκεί να μπορεί αυτομάτως να προσφέρεται σε πολλές διαφορετικές εφαρμογές όταν η συγκεκριμένη βιβλιοθήκη απαιτείται, χωρίς να υπάρχει η ανάγκη από τον προγραμματιστή εκ νέου να την αναζητήσει.

Στιγμιότυπο από το POM.xml

```
xml

<dependencies>
  <!-- Spring Boot Web -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>

  <!-- Spring Data JPA & Hibernate -->
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>

  <!-- MySQL Driver -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
  </dependency>
</dependencies>
```

Εικόνα 20:

Με αυτό τον τρόπο, το Maven:

- Κατεβάζει αυτόματα τις βιβλιοθήκες που απαιτούνται για την εφαρμογή.
- Διαχειρίζεται τις εκδόσεις των εξαρτήσεων.
- Διασφαλίζει ότι όλες οι βιβλιοθήκες είναι συμβατές μεταξύ τους.

8 Spring Boot Αρχιτεκτονική

8.1 Εισαγωγή

Το Spring Boot είναι ένα framework ανάπτυξης εφαρμογών Java που βασίζεται στο Spring Framework και προσφέρει έναν πιο απλό, αυτοματοποιημένο και γρήγορο τρόπο ανάπτυξης εφαρμογών. Διευκολύνει την ανάπτυξη εφαρμογών Java προσφέροντας αυτοματοποιημένη διαμόρφωση, ενσωματωμένο server, εύκολη διαχείριση εξαρτήσεων και υποστήριξη του MVC μοντέλου. Με την αρχιτεκτονική του, η εφαρμογή είναι καθαρά δομημένη, εύκολα επεκτάσιμη και συμβατή με μοντέρνες τεχνολογίες.

8.2 Βασικά Χαρακτηριστικά

1) Αυτόματη Διαμόρφωση (Auto Configuration)

- ✓ Το Spring Boot διαμορφώνει αυτόματα την εφαρμογή με βάση τις εξαρτήσεις (dependencies) που περιλαμβάνονται στο pom.xml (Maven).
- ✓ Δεν απαιτείται χειροκίνητη ρύθμιση των βασικών υπηρεσιών, όπως σύνδεση σε βάση δεδομένων ή ρύθμιση του server.

2) Embedded Servers (Ενσωματωμένοι Servers)

- ✓ Δεν χρειάζεται ξεχωριστή εγκατάσταση Tomcat, Jetty ή Undertow. Ο server ξεκινά αυτόματα όταν εκτελείται η εφαρμογή.
- ✓ Αυτόματη Διαμόρφωση (Auto Configuration) – Ανιχνεύει τις εξαρτήσεις και προσαρμόζει αυτόματα τις ρυθμίσεις.

3) Spring Boot Starters (Πακέτα Εξαρτήσεων)

- ✓ Παρέχει έτοιμα starter dependencies για διάφορες λειτουργίες (π.χ. spring-boot-starter-web, spring-boot-starter-data-jpa).
- ✓ Εξοικονομεί χρόνο από το να προσθέτουμε ξεχωριστές βιβλιοθήκες χειροκίνητα.

4) Spring Boot DevTools

- ✓ Ενεργοποιεί αυτόματη επανεκκίνηση (live reload) της εφαρμογής κατά την αλλαγή του κώδικα.
- ✓ Βελτιώνει την ταχύτητα ανάπτυξης μειώνοντας τον χρόνο εκκίνησης της εφαρμογής.

5) Εύκολη Σύνδεση με Βάσεις Δεδομένων

- ✓ Υποστηρίζει MySQL, PostgreSQL, H2 και πολλές άλλες βάσεις δεδομένων.
- ✓ Χρησιμοποιεί Spring Data JPA για αλληλεπίδραση με τη βάση, χωρίς να απαιτείται SQL χειροκίνητα.

6) Υποστήριξη για RESTful APIs

- ✓ Εύκολη δημιουργία REST API με @RestController και @RequestMapping.

7) Υποστήριξη για Templates (Thymeleaf)

- ✓ Το Thymeleaf χρησιμοποιείται για τη δημιουργία δυναμικών HTML σελίδων σε συνδυασμό με Spring MVC.

8.3 Λειτουργία

Το **Spring Boot** βασίζεται στο **Spring Framework** και χρησιμοποιεί την αρχιτεκτονική Model-View-Controller (MVC) για την ανάπτυξη εφαρμογών web.

Βασικά Στοιχεία Αρχιτεκτονικής

Controller Layer (Επίπεδο Ελέγχου)

- ✓ Αναλαμβάνει την επικοινωνία μεταξύ του χρήστη και της εφαρμογής.
- ✓ Περιέχει τις μεθόδους που διαχειρίζονται τα αιτήματα (@RestController).

Service Layer (Επιχειρησιακή Λογική)

- ✓ Επεξεργάζεται τα δεδομένα και περιέχει τους **κανόνες επιχειρησιακής λογικής**.
- ✓ Ανακτά δεδομένα από το Repository και τα επιστρέφει στον Controller.

Repository Layer (Επίπεδο Δεδομένων)

- ✓ Χρησιμοποιεί **Spring Data JPA** για την αλληλεπίδραση με τη βάση δεδομένων.
- ✓ Υλοποιείται μέσω των **Repositories** (@Repository).

Database Layer (Βάση Δεδομένων)

- ✓ Χρησιμοποιεί **Hibernate (JPA)** για τον χειρισμό των δεδομένων.
- ✓ Η σύνδεση ρυθμίζεται μέσω του application.properties.

8.4 Χαρακτηριστικά Εφαρμογής

1) spring-boot-starter-data-jpa (Spring Data JPA)

- Επιτρέπει αλληλεπίδραση με βάσεις δεδομένων χρησιμοποιώντας **Repositories** αντί για SQL queries.
- Χρησιμοποιείται σε συνδυασμό με **Hibernate**, που είναι η πιο διαδεδομένη υλοποίηση του JPA.

Πως χρησιμοποιείται στην εφαρμογή:

- Έχουν δημιουργηθεί οντότητες (@Entity), όπως η κλάση Patient
- Έχουν δημιουργηθεί **Repositories**, όπως PatientRepository, για ανάκτηση, αποθήκευση, διαγραφή δεδομένων

```
@Repository
public interface PatientRepository extends JpaRepository<Patient, Integer> {
    List<Patient> findByLastNameAndFirstName(String lastName, String firstName);
}
```

Εικόνα 21: Patient Repository

2) spring-boot-starter-thymeleaf (Thymeleaf)

- Είναι ένα **template engine** που επιτρέπει την εμφάνιση δυναμικών δεδομένων σε HTML σελίδες.
- Ενσωματώνεται εύκολα με το **Spring MVC** και χρησιμοποιεί δεδομένα από τα Controllers.

Πως χρησιμοποιείται στην εφαρμογή:

- Υπάρχουν HTML templates σελίδες
- Τα δεδομένα προβάλλονται στη σελίδα χρησιμοποιώντας **Thymeleaf syntax** (th:text, th:each).

```
<table>
  <tr th:each="patient : ${patients}">
    <td th:text="${patient.firstName}"></td>
    <td th:text="${patient.lastName}"></td>
  </tr>
</table>
```

Εικόνα 22

3) spring-boot-starter-web, το οποίο:

- Επιτρέπει την ανάπτυξη **RESTful APIs** και web εφαρμογών.
- Παρέχει υποστήριξη για **Spring MVC**, ώστε να διαχειρίζεσαι HTTP αιτήματα.

Πως χρησιμοποιείται στην εφαρμογή:

- Μέσω των Controllers (@RestController) για να τρέχουν τα “API endpoints”

```
@RestController
@RequestMapping("/patients")
public class PatientController {
    @Autowired
    private PatientService patientService;

    @GetMapping("/{id}")
    public ResponseEntity<Patient> getPatientById(@PathVariable int id) {
        return ResponseEntity.ok(patientService.findById(id));
    }
}
```

Εικόνα 23

4) spring-boot-devtools

- Βελτιώνει την ανάπτυξη παρέχοντας **αυτόματη επανεκκίνηση** της εφαρμογής όταν αλλάζει ο κώδικας.
- Περιλαμβάνει **Live Reload**, ώστε να βλέπεις άμεσα τις αλλαγές στον browser.

Πως χρησιμοποιείται στην εφαρμογή:

- Όταν γίνεται αλλαγή στον κώδικα, η εφαρμογή **επανεκκινείται αυτόματα**, επιταχύνοντας την ανάπτυξη.

5) mysql-connector-j

- Επιτρέπει τη **σύνδεση της εφαρμογής με τη MySQL βάση δεδομένων**.
- 7 Χρησιμοποιείται από το **Spring Boot Data JPA** για να αποθηκεύει και να ανακτά δεδομένα.

Πως χρησιμοποιείται στην εφαρμογή:

- Καθορίζεται η σύνδεση στη βάση δεδομένων μέσω application.properties:

8.5 Μοντέλο MVC (Model-View-Controller)

Το **MVC (Model-View-Controller)** είναι ένα δημοφιλές σχεδιαστικό **μοτίβο** (design pattern) που χωρίζει μια εφαρμογή σε τρία επίπεδα:

- **Model (Μοντέλο)** → Αναπαριστά τα δεδομένα της εφαρμογής και τους κανόνες επιχειρησιακής λογικής.
- **View (Προβολή)** → Εμφανίζει τα δεδομένα στον χρήστη (π.χ., HTML, JSON, XML).
- **Controller (Ελεγκτής)** → Διαχειρίζεται τα αιτήματα των χρηστών και αλληλεπιδρά με το Model.

Το Spring Boot υποστηρίζει **MVC μέσω του Spring Web**, όπου κάθε αίτημα HTTP περνάει από έναν **Controller** και επιστρέφει τα δεδομένα στον χρήστη.

Παράδειγμα MVC

Model: Το Model αναπαριστά τα δεδομένα της εφαρμογής και τη δομή τους. Είναι υπεύθυνο για την αλληλεπίδραση με τη βάση δεδομένων και την επιχειρησιακή λογική. Στο **Spring Boot**, το το Model υλοποιείται μέσω **κλάσεων οντοτήτων (Entities)** που χρησιμοποιούν **JPA (Java Persistence API)**.

Παράδειγμα Model (Οντότητα Patient)

```

@Entity
@Table(name = "patient")
public class Patient {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name = "first_name") 3 usages
    private String firstName;

    @Column(name = "last_name") 3 usages
    private String lastName;

    @Column(name = "age") 3 usages
    private int age;

    @Column(name = "gender") 3 usages
    private String gender;

    @Column(name = "comment", columnDefinition = "TEXT") 3 usages
    private String comment;

    @Column(name = "registration_day") 5 usages
    private LocalDateTime registration_day;

    // Change to ManyToOne relationship
    @ManyToOne(cascade = CascadeType.MERGE) 3 usages
    @JoinColumn(name = "location_id") // This column in Patient table references Location
    private Location location;

    @ManyToMany(cascade = CascadeType.MERGE) 3 usages
    @JoinTable(
        name = "patient_sentiment", // Join table name
        joinColumns = @JoinColumn(name = "patient_id"), // Foreign Key for Patient
        inverseJoinColumns = @JoinColumn(name = "sentiment_id") // Foreign key for Sentiment
    )
    private Set<Sentiment> sentiments = new HashSet<>(); // Initialize with an empty Set

    // New OneToOne relationship with User
    @ManyToOne(cascade = CascadeType.MERGE) 4 usages
    @JoinColumn(name = "user_id", referencedColumnName = "id") // Foreign key reference to User
    private User user;

    public Patient() {}
}

```

Εικόνα 24: Patient Entity

Τι κάνει το Model;

- Χρησιμοποιεί την ανατολή **@Entity** για να δημιουργήσει τον αντίστοιχο πίνακα στη βάση δεδομένων.
- Ορίζει τις στήλες με **@Column(name = "...")**.
- Περιλαμβάνει **getters** και **setters** για πρόσβαση στα δεδομένα.

Controller: Το **Controller** λαμβάνει αιτήματα από τους χρήστες, αλληλεπιδρά με το Model μέσω του Service Layer και επιστρέφει τα αποτελέσματα.

```

@RestController
@RequestMapping("/patients")
public class PatientController {

    @Autowired
    private PatientService patientService;

    @GetMapping("/{id}")
    public ResponseEntity<Patient> getPatientById(@PathVariable int id) {
        return ResponseEntity.ok(patientService.findById(id));
    }

    @PostMapping("/")
    public ResponseEntity<Patient> createPatient(@RequestBody Patient patient) {
        return ResponseEntity.ok(patientService.savePatient(patient));
    }
}

```

Εικόνα 25: PatientController

Τι κάνει ο Controller;

- Διαχειρίζεται τα HTTP αιτήματα (GET, POST, DELETE, κ.λπ.).
- Χρησιμοποιεί το `@RestController` για να υποδηλώσει ότι είναι Controller.
- Καλεί το **PatientService** για να εκτελέσει την επιχειρησιακή λογική.
- Χρησιμοποιεί το `@RequestMapping("/patients")` για να καθορίσει το βασικό URL του Controller.

View: Το **View** είναι το μέρος της εφαρμογής που εμφανίζει τα δεδομένα στον χρήστη. Στις **RESTful APIs** του Spring Boot, το View μπορεί να επιστρέφει **JSON ή XML**, αντί για HTML.

Παράδειγμα JSON Response

Αν καλέσουμε το API GET /patients/1, θα λάβουμε:

```
{
  "id": 1,
  "firstName": "John",
  "lastName": "Doe",
  "age": 30,
  "gender": "Male",
  "comment": "Feeling good today!"
}
```

Εικόνα 26

Παράδειγμα View σε HTML (Thymeleaf)

Αν θέλουμε να εμφανίσουμε τα δεδομένα σε **HTML μέσω Thymeleaf**, μπορούμε να έχουμε το αρχείο patients.html:

```
<table>
  <thead>
    <tr>
      <th>ID</th>
      <th>First Name</th>
      <th>Last Name</th>
      <th>Gender</th>
      <th>Location</th>
      <th>Comment</th>
      <th>Registration Day</th>
    </tr>
  </thead>
  <tbody>
    <!-- Show "No patients found" if the list is empty -->
    <tr th:if="#lists.isEmpty(patients)">
      <td colspan="7" class="no-patients">No patients found</td>
    </tr>
    <!-- Loop through each patient -->
    <tr th:each="patient : ${patients}">
      <td th:text="${patient.id}"></td>
      <td th:text="${patient.firstName}"></td>
      <td th:text="${patient.lastName}"></td>
      <td th:text="${patient.gender}"></td>
      <td th:text="${patient.location.city}"></td>
      <td th:text="${patient.comment}"></td>
      <td th:text="${patient.getRegistrationDay()}"></td>
    </tr>
  </tbody>
</table>
```

Εικόνα 27

Τι κάνει το View

- Αν χρησιμοποιούμε REST API, επιστρέφει JSON/XML δεδομένα.
- Αν χρησιμοποιούμε web εφαρμογή με Thymeleaf, δημιουργεί δυναμικές HTML σελίδες.

Συμπέρασμα

Το Spring Boot **MVC** χωρίζει τη λογική της εφαρμογής σε **Model, View, Controller**, επιτρέποντας καθαρό και δομημένο κώδικα.

- **To Model** αποθηκεύει και διαχειρίζεται τα δεδομένα.
- **To Controller** λαμβάνει αιτήματα και αλληλοεπιδρά με το Model.
- **To View** εμφανίζει τα δεδομένα στον χρήστη.

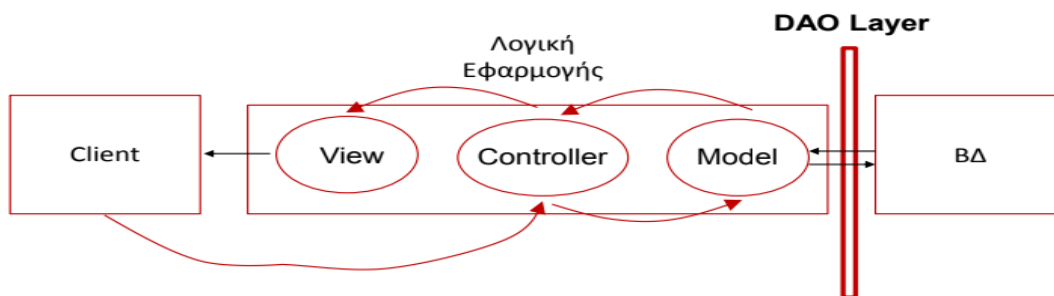
Έτσι, η αρχιτεκτονική **MVC του Spring Boot** βοηθά στην εύκολη **ανάπτυξη, συντήρηση και επέκταση** της εφαρμογής.

8.6 Επίπεδο DAO - Μοντέλο SOA

Για την καλύτερη και αποδοτικότερη υλοποίηση η εφαρμογή έχει ενσωματώσει τα επιπλέον επίπεδα **DAO Layer Service Layer**.

Επίπεδο DAO

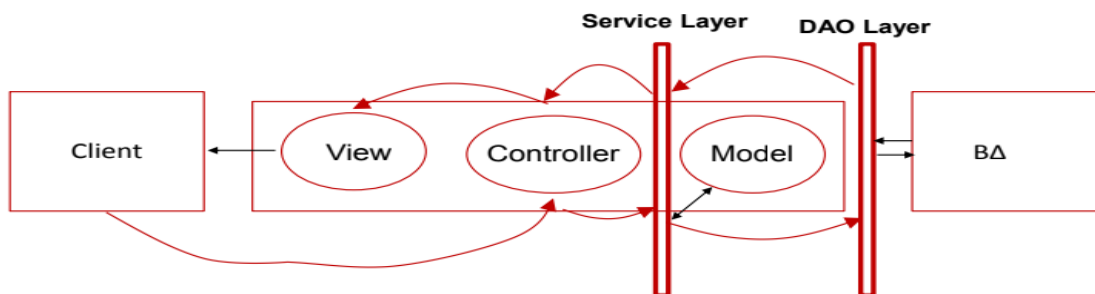
DAO Layer: Παρέχει υπηρεσίες μόνιμης αποθήκευσης των κλάσεων στη ΒΔ. Βρίσκεται ανάμεσα στη ΒΔ και το **Model** και παρέχει persistence services. Σε αυτό το επίπεδο έχουμε τα Repositories τα οποία επικοινωνούν με τη βάση δεδομένων και κάνουν όλες τις ενέργειες.



Εικόνα 28

Μοντέλο SOA (Service–Oriented-Architecture)

Service Layer: Το service layer είναι ανεξάρτητο και αποτελείται από υπηρεσίες που παρέχει η εφαρμογή στους clients. Συγκεκριμένα, όλες οι υπηρεσίες έχουν οριστεί σε ένα interface και έπεται σε ξεχωριστή κλάση, αφού γίνεται “implementation” αυτού, γίνεται η υλοποίηση.



Εικόνα 29

9 NLP (Natural Language Processing)

9.1 Εισαγωγή

Το Natural Language Processing (NLP) είναι ένα πεδίο της Τεχνητής Νοημοσύνης (AI) που επιτρέπει στους υπολογιστές να κατανοούν, να επεξεργάζονται και να παράγουν ανθρώπινη γλώσσα.

Χρησιμοποιείται ευρέως σε εφαρμογές που σχετίζονται με ανάλυση συναισθήματος, chatbot, αναγνώριση ομιλίας, μετάφραση γλώσσας και πολλά άλλα.

9.2 Χρησιμότητα

Τι προσφέρει το NLP

1. **Ανάλυση Συναισθήματος (Sentiment Analysis)**
 - Αναγνωρίζει αν ένα κείμενο έχει **θετικό, αρνητικό ή ουδέτερο** συναίσθημα
2. **Κατανόηση & Ανάλυση Κειμένου**
 - Ανάλυση δομής προτάσεων (**syntactic parsing**).
 - Αναγνώριση οντοτήτων (**Named Entity Recognition - NER**, π.χ., ονόματα, τοποθεσίες, ημερομηνίες).
 - Κατηγοριοποίηση κειμένου (**text classification**).
3. **Μετατροπή Κειμένου σε Ομιλία (Text-to-Speech - TTS) & Ομιλίας σε Κείμενο**
 - Χρήσιμο για chatbot, βοηθούς φωνής (π.χ. Alexa, Google Assistant).
4. **Αυτόματη Μετάφραση (Machine Translation)**
 - Μετατροπή κειμένου από μία γλώσσα σε άλλη (Google Translate).
5. **Σύνοψη Κειμένου (Text Summarization)**
 - Περίληψη μεγάλων κειμένων σε μικρότερες εκδοχές.
6. **Αυτόματη Διόρθωση & Αναγνώριση Γλωσσικών Λαθών**
 - Διόρθωση γραμματικών και συντακτικών λαθών.

Πως χρησιμοποιείται στην εφαρμογή

Η παρούσα εφαρμογή χρησιμοποιεί το **Stanford CoreNLP** και υλοποιεί μία προσέγγιση για **custom sentiment analysis**. Συγκεκριμένα, έχουμε δημιουργήσει κάποια λεξικά λέξεων αλλά και εκφράσεων που εκφράζουν τα συναισθήματα (χαρά, λύπη, θυμό, άγχος, φόβο) και προσπαθούμε να εντοπίσουμε μέσα στο κείμενο είτε τις λέξεις αυτές είτε λήμματα των λέξεων αυτών.

Λειτουργία κώδικα

1. **Προετοιμασία NLP Pipeline**
 - Αρχικά χωρίζεται το κείμενο σε προτάσεις, λέξεις(tokens), εντοπίζει τα σημεία ή αριθμό
 - Το κείμενο εισάγεται στο pipeline για να διαχωριστεί σε προτάσεις και λέξεις.
2. **Επεξεργασία προτάσεων & ανάλυση λέξεων**
 - Για κάθε πρόταση ελέγχει εάν το λήμμα κάθε λέξης ανήκει σε κάποιο από τα συναισθηματικά λεξικά (**SentimentWords**).
3. **Διαχείριση άρνησης**
 - Αν βρεθεί λέξη που δηλώνει άρνηση (π.χ., *not, never*), τότε το συναίσθημα αναστρέφεται. Πχ το “i feel happy” δηλώνει συναίσθημα χαράς (joy) αφού το “happy” ανήκει στο λεξικό με τις λέξεις που εκφράζουν χαρά, άρα το “i don’t feel happy” αναστρέφεται και εκφράζει συναίσθημα λύπης(sadness).
4. **Ανίχνευση Phrasal Verbs**
 - Υπάρχουν επίσης λεξικά με καταχωρημένες εκφράσεις που εκφράζουν τα παραπάνω συναισθήματα. Η διαδικασία είναι η ίδια με τις λέξεις
5. **Τελική καταμέτρηση συναισθημάτων**
 - Κάθε φορά που εντοπίζεται και καταγράφεται ένα συναίσθημα αυξάνεται ένας δείκτης για το συναίσθημα αυτό. Πχ εάν βρούμε 2 συναισθηματικές λέξεις χαράς και μία άγχους τότε ο δείκτης “**joyfulCount**” αυξάνει κατά δύο μονάδες ενώ ο “**anxiousCount**” κατά μία.
 - Υπολογίζεται το ποσοστό των συναισθηματικών λέξεων που έχει βρεθεί (joyfulCount & anxiousCount) σε σχέση με το σύνολο των λέξεων που έχει όλο το κείμενο. Εάν το ποσοστό ξεπερνάει ένα συγκεκριμένο όριο που έχουμε θέσει, τότε προκύπτει το συναίσθημα αυτό.

Παράδειγμα 1

“The last few days i feel really anxious and stressed while my program is full of duties. From the other side, i have a good feeling of getting better and accomplish all of them quite soon.”

Στην παράγραφο αυτή το σύνολο των λέξεων είναι 36 και αριθμός των συναισθηματικών λέξεων 4 (anxious, stressed, good, good). Οι λέξεις anxious και stressed εκφράζουν άγχος (anxiety) ενώ η λέξη good εκφράζει χαρά (joy). Επομένως, τα ποσοστά που προκύπτουν σε σχέση με τον συνολικό αριθμό του κειμένου είναι : **anxiousScore = 0.0555** και **joyScore= 0.0555**. Επειδή το σκορ αυτό, που τυχαίνει να είναι ίδιο και για τα δύο συναισθήματα, είναι μεγαλύτερο από το ποσοστό που έχουμε ορίσει αυθαίρετα και είναι το “0.01”, προκύπτει από το μεν πρώτο σκορ το συναισθήμα anxiety και από το μεν δεύτερο το συναισθήμα joy .Επομένως, το τελικό συναισθήμα που προκύπτει είναι το **“anxiety_joy”**.

Παράδειγμα 2

Τροποποιούμε λίγο το Παράδειγμα 1 προσθέτοντας μία άρνηση πριν από το anxious και αφαιρούμε το επίθετο “stressed”.

*“The last few days i **don't** feel really **anxious** while my program is full of duties. From the other side, i have a **good** feeling of getting **better** and accomplish all of them quite soon. ”*

Το συναισθήμα που προκύπτει είναι το “joy” αφού οι συναισθηματικές λέξεις που προκύπτουν είναι 3 (not anxious, good, good) οι οποίες εκφράζουν όλες χαρά με αποτέλεσμα -> **joyScore= 0.0857..**

Παρατηρούμε ότι ενώ η άρνηση στο Παράδειγμα 2 εκφράζεται με το **“don't”** και παρόλο που ενδιαμέσως με το επίθετο **“anxious”** μεσολαβούν 2 λέξεις (feel, really) το αρνητικό πρόσημο προσμετράται.

Τι προσφέρει αυτή η προσέγγιση

Η προσέγγιση που χρησιμοποιεί ο συγκεκριμένος αλγόριθμος παρέχει κάποια πλεονεκτήματα στην εκτίμηση συναισθήματος σε σχέση με αυτόν που παρέχει η βιβλιοθήκη stanford nlp, όπως:

- **Ευελιξία:** Η βιβλιοθήκη της stanford nlp, όσον αφορά την εκτίμηση συναισθήματος, περιορίζεται μόνο στα συναισθήματα: positive, negative, neutral. Αντίθετα, ο αλγόριθμος της εφαρμογής περιέχει και επεξεργάζεται τα συναισθήματα: joy, sadness, fear anger, anxiety, neutral αλλά και όλους τους συνδυασμούς μεταξύ αυτών πχ. “sadness_fear_anger”. Επιπλέον, μπορούμε να προσθέσουμε δικά μας λεξικά με περισσότερα συναισθήματα ή και περισσότερες λέξεις.
- **Υποστήριξη Άρνησης:** Διαχειρίζεται φράσεις όπως “I'm not happy”, αναστρέφοντας το συναισθήμα.
- **Εντοπισμός Σύνθετων Εκφράσεων (Phrasal Verbs).**

Κώδικας

Λεξικά για λέξεις που δηλώνουν άγχος (anxious)

```
public static final Set<String> ANXIOUS_WORDS = new HashSet<>(Arrays.asList( 3 usages
    "anxious", "stress", "worried", "nervous", "tense", "uneasy",
    "restless", "jittery", "agitated", "fretful", "worry", "anxiety",
    "panicky", "fraught", "distracted", "alarmed", "distressed", "exhaust", "exhausted", "exhausting", "crippling"
));
```

Εικόνα 30

Λέξεις που δηλώνουν άρνηση

```
public class NegationWords { 2 usages  ⤴ Stergios9
    public static final Set<String> NEGATION_WORDS = new HashSet<>(Arrays.asList( 1 usage
        "not", "never", "no", "none", "hardly", "barely", "rarely", "seldom", "without", "neither"
    ));
}
```

Εικόνα 31

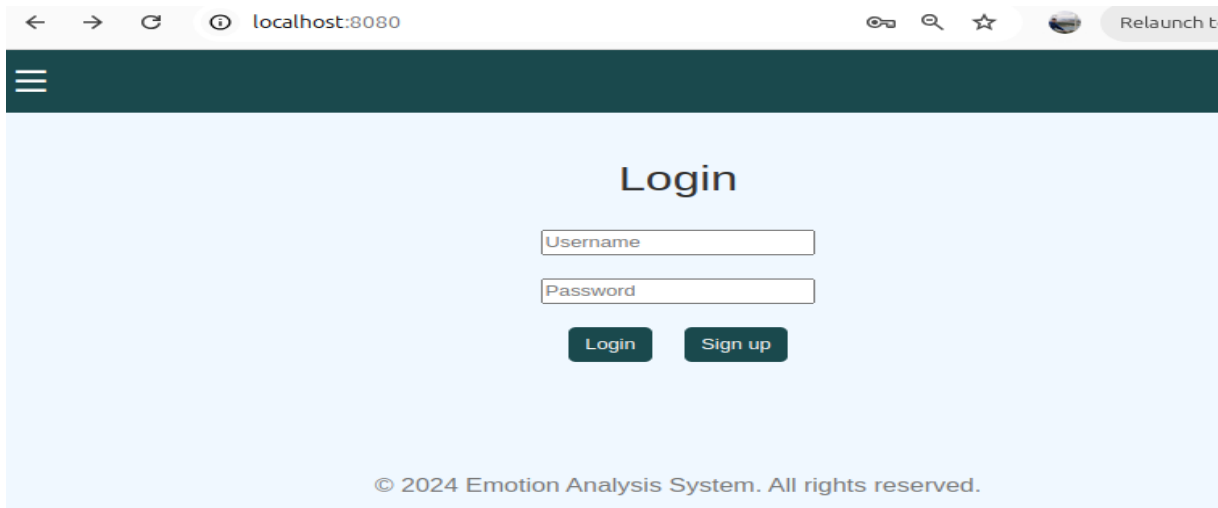
Φράσεις που δηλώνουν χαρά

```
public List<String> getJoyPhrasalVerbs() { 1 usage  ⚙ Stergios9
    return Arrays.asList("light up", "cheer up", "liven up", "crack up", "perk up", "bask in",
        "glow with", "kick up", "beam at", "laugh off", "revel in");
}
```

Εικόνα 32

10 Εκτέλεση Εφαρμογής

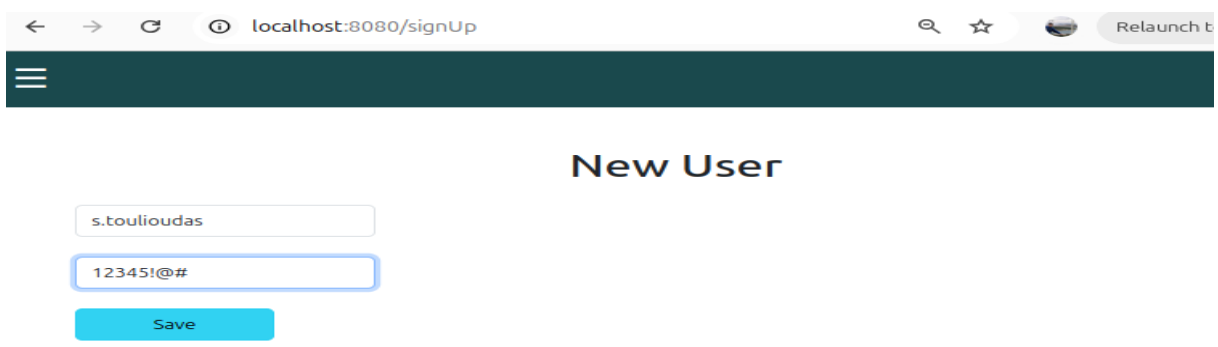
Αρχικά, όπως έχουμε ήδη αναφέρει, η τεχνολογία Spring Boot διαθέτει ενσωματωμένο διακομιστή που στην περίπτωση μας είναι ο Tomcat. Αφού έχουμε ορίσει στο Application.properties την πόρτα (8080) εκκινούμε την εφαρμογή:



Εικόνα 33

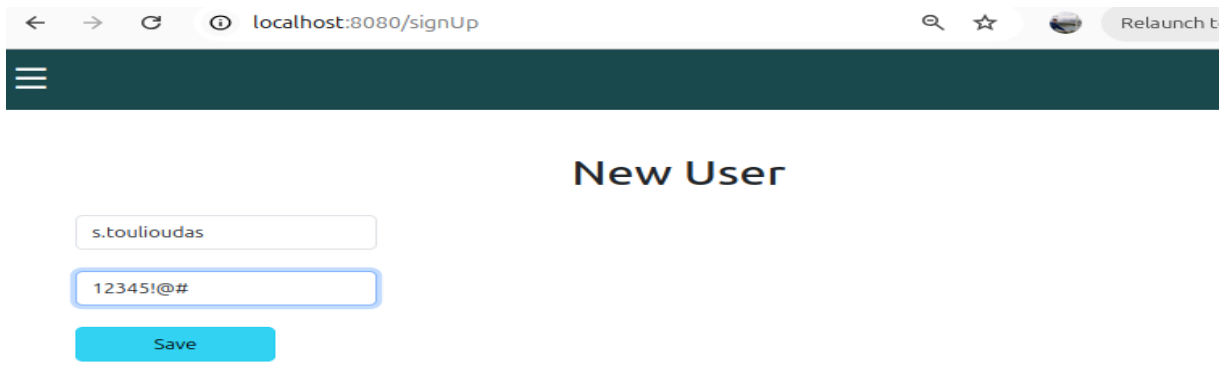
Οι επιλογές που έχουμε είναι είτε να συνδεθούμε με ένα λογαριασμό είτε να δημιουργήσουμε ένα καινούργιο. Έστω ότι δημιουργούμε ένα καινούργιο:

Επιλέγω «Sign up»



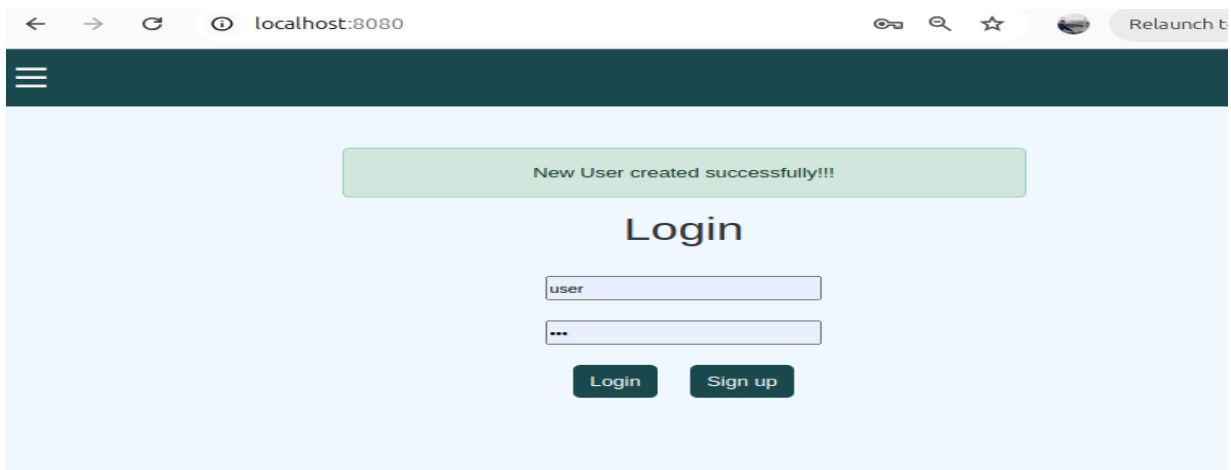
Εικόνα 34

Από εδώ έχουμε τη δυνατότητα να ορίσουμε ένα «username» και «password» και να το καταχωρήσουμε πατώντας το κουμπί «Save». Μόλις το κάνουμε αυτό η φόρμα μας επιστρέφει στην αρχική σελίδα σύνδεσης επιστρέφοντας το μήνυμα επιτυχίας.



A screenshot of a web browser showing a registration page. The address bar displays 'localhost:8080/signUp'. The page has a dark green header with a hamburger menu icon. The main content area is white and titled 'New User'. It contains two input fields: the first contains 's.toulioudas' and the second contains '12345!@#'. Below the fields is a blue 'Save' button.

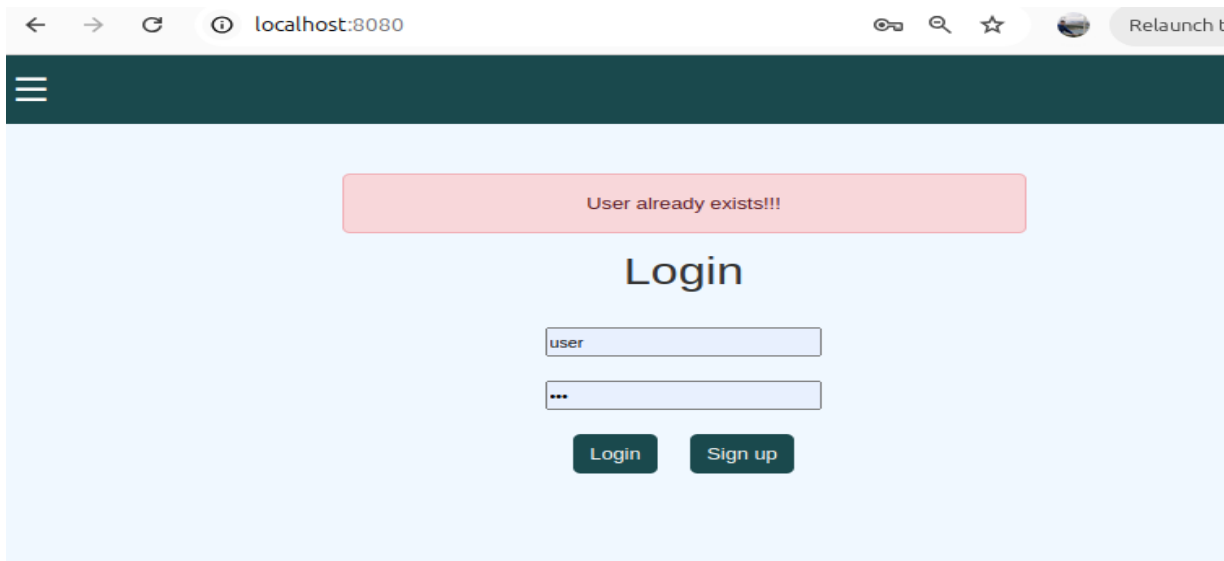
Εικόνα 35



A screenshot of a web browser showing a login page. The address bar displays 'localhost:8080'. The page has a dark green header with a hamburger menu icon. The main content area is light blue. At the top, a green message box says 'New User created successfully!!!'. Below it is the title 'Login'. There are two input fields: the first contains 'user' and the second contains '...'. Below the fields are two buttons: 'Login' and 'Sign up'.

Εικόνα 36

Σε περίπτωση που το "username" του χρήστη υπάρχει ήδη καταχωρημένο μας επιστρέφει στην αρχική σελίδα επιστρέφοντας μήνυμα σφάλματος. Αυτό σημαίνει ότι ο χρήστης δεν έχει δημιουργήσει λογαριασμό.



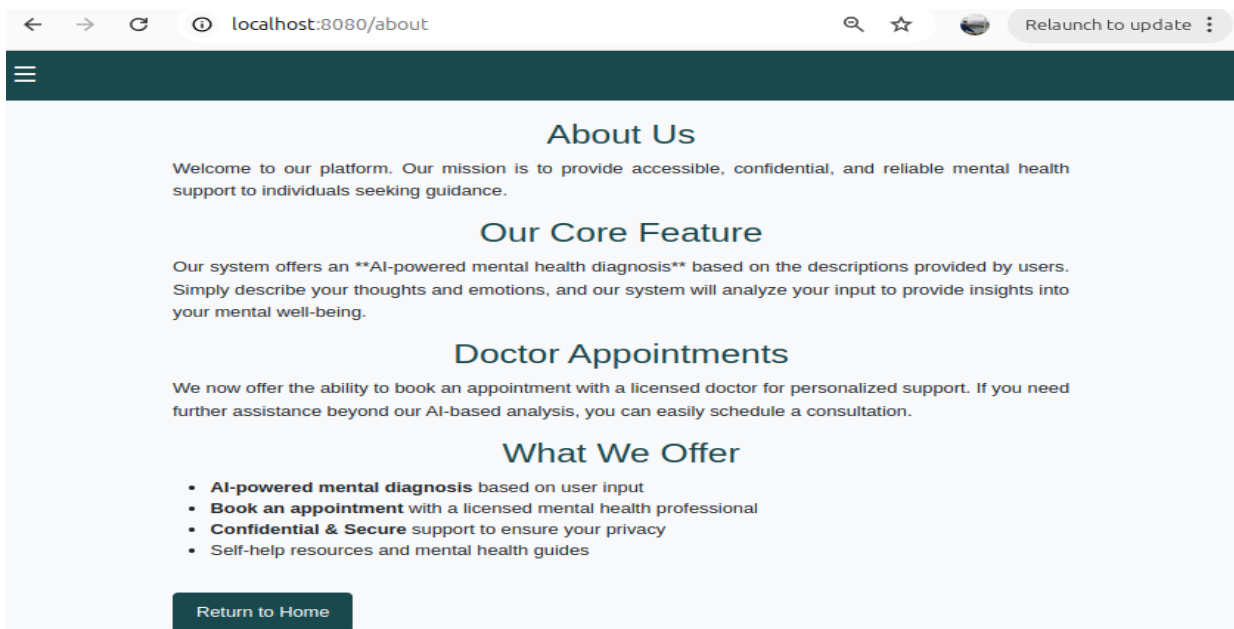
Εικόνα 37

Έστω λοιπόν, ότι έχουμε συνδεθεί με τα σωστά διαπιστευτήρια είτε έχοντας δημιουργήσει ένα καινούργιο λογαριασμό χρήστη είτε επειδή είχαμε από πριν. Τα δικαιώματα που έχει ένας χρήστης στην εφαρμογή είναι είτε να είναι ένας απλός χρήστης(**user**) είτε διαχειριστής της εφαρμογής (**administrator**).

10.1 Εκτέλεση ως απλός χρήστης

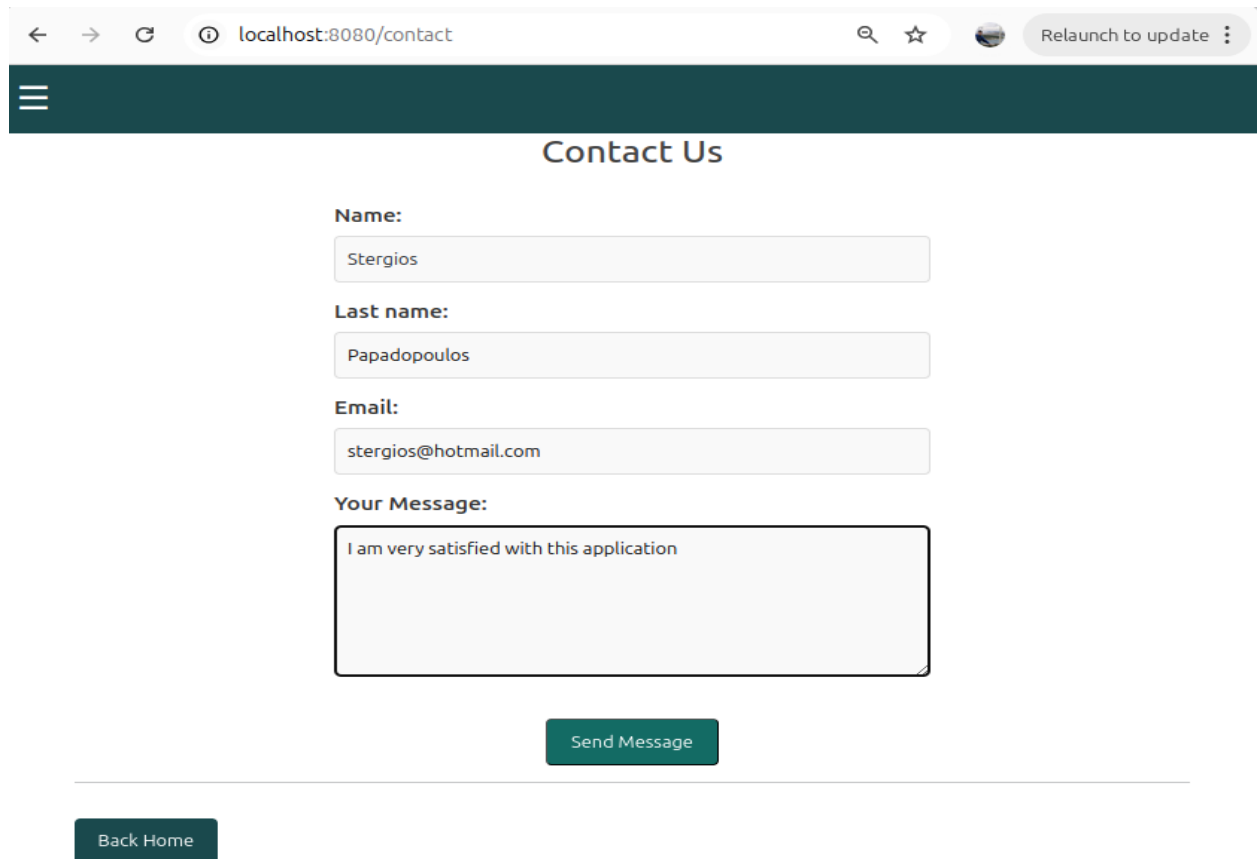
Μόλις ο χρήστης συνδεθεί η εφαρμογή πλοηγείται στην κεντρική σελίδα (welcome.html). Αριστερά και πάνω στη σελίδα υπάρχει ένα μενού επιλογών. Οι επιλογές αναλύονται παρακάτω:

ABOUT: Παρέχει πληροφορίες σχετικά με το σκοπό της εφαρμογής, κάποια χαρακτηριστικά αυτής και ποιες υπηρεσίες προσφέρει.



Εικόνα 38

CONTACT: Εδώ υπάρχει η δυνατότητα να στείλει ένας χρήστης ένα μήνυμα ή κάποια αξιολόγηση σχετικά με κάποια υπηρεσία της εφαρμογής ή και γενικότερα οτιδήποτε θα μπορούσε να βελτιώσει τις υπηρεσίες που αυτή παρέχει ή ακόμη να θέσει και κάποια ερώτηση. Σε περίπτωση κάποιου ερωτήματος, ο διαχειριστής της εφαρμογής ο οποίος θα είναι ο γιατρός ή κάποιος /α από τη γραμματεία του ιατρείου, μπορεί να απαντήσει στο ηλεκτρονικό ταχυδρομείο του χρήστη.



← → ↻ ⓘ localhost:8080/contact 🔍 ☆ 🌐 Relaunch to update ⋮

☰

Contact Us

Name:

Last name:

Email:

Your Message:

I am very satisfied with this application

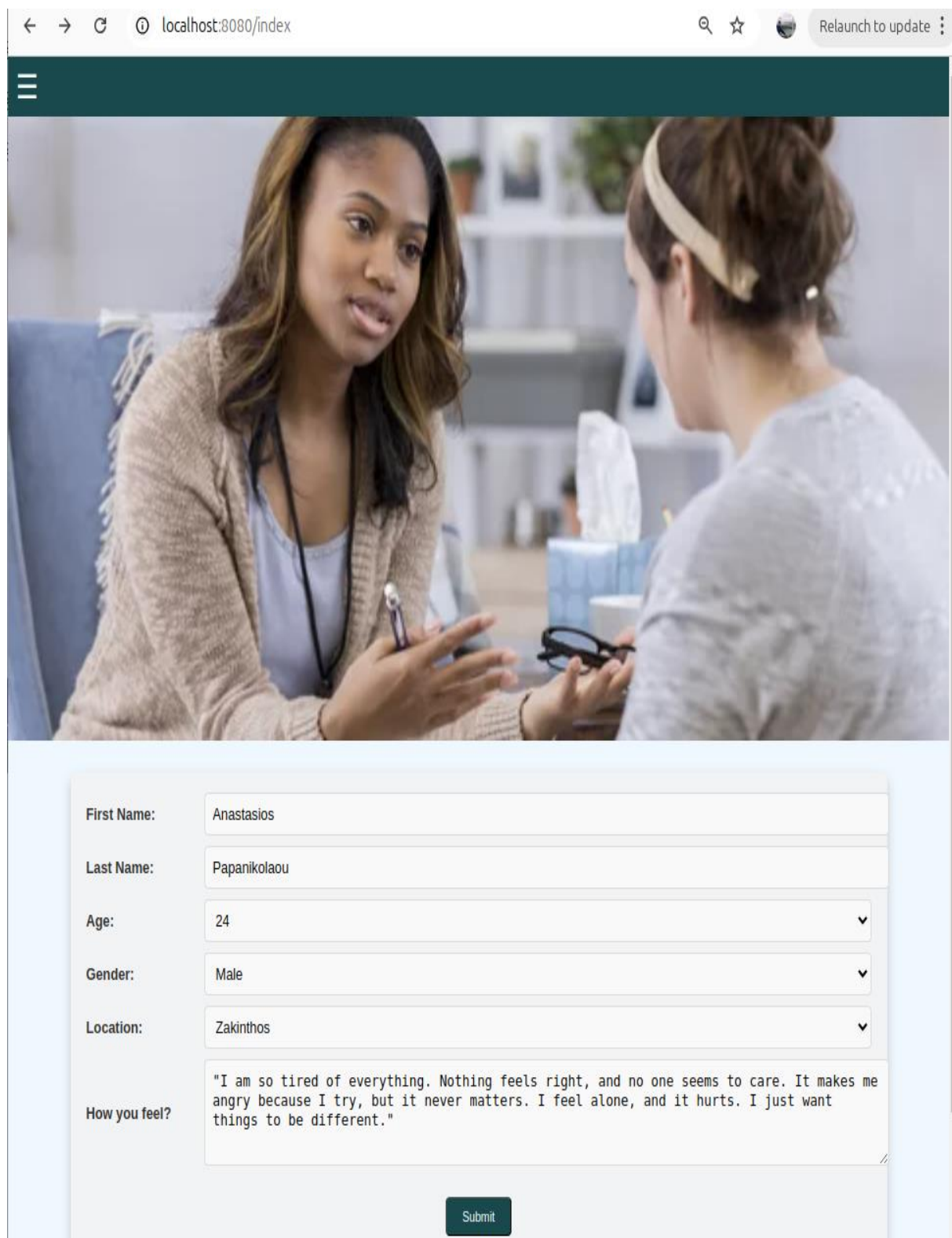
Send Message

Back Home

Εικόνα 39

Σε περίπτωση επιτυχούς καταχώρησης εμφανίζεται μήνυμα.

MEDICAL DIAGNOSIS: Εδώ ο χρήστης μπορεί να λάβει μία συναισθηματική διάγνωση απλά και μόνο συμπληρώνοντας τα στοιχεία του και κάνοντας μία περιγραφή της ψυχικής του κατάστασης.



localhost:8080/index

Relaunch to update

First Name: Anastasios

Last Name: Papanikolaou

Age: 24

Gender: Male

Location: Zakynthos

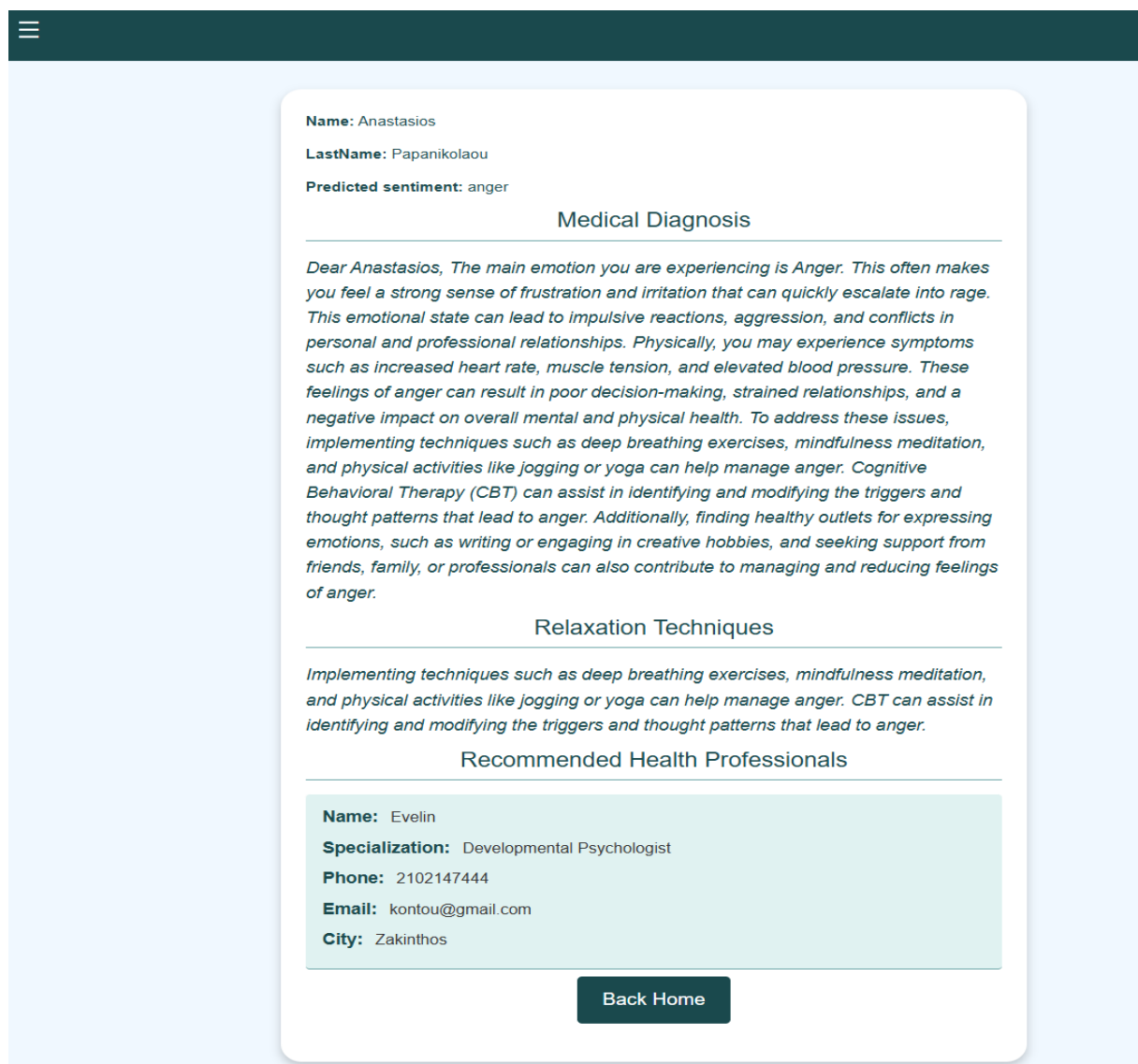
How you feel?

"I am so tired of everything. Nothing feels right, and no one seems to care. It makes me angry because I try, but it never matters. I feel alone, and it hurts. I just want things to be different."

Submit

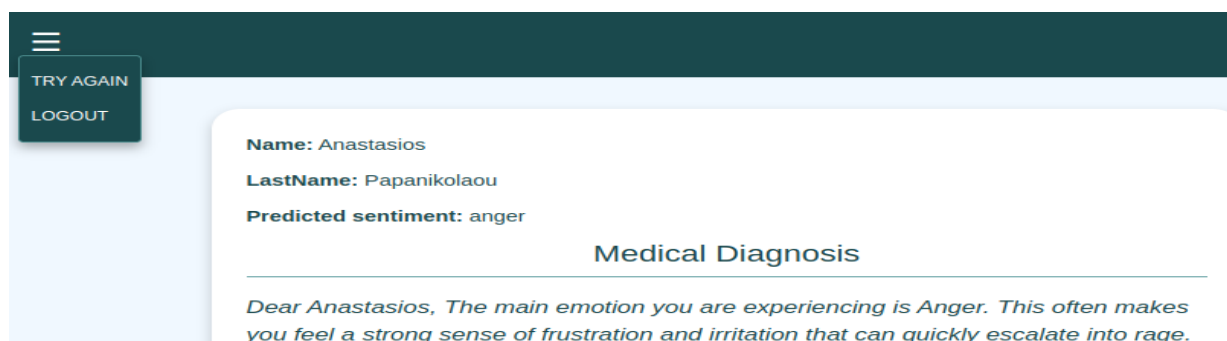
Εικόνα 40

Πατώντας το κουμπί "Submit" εμφανίζονται τα αποτελέσματα



Εικόνα 41

Εάν ο χρήστης επιθυμεί, μπορεί να ξαναχρησιμοποιήσει αυτή την υπηρεσία πατώντας την επιλογή “TRY AGAIN” από το μενού



Εικόνα 42

BOOK APPOINTMENT: Αυτή η υπηρεσία δίνει τη δυνατότητα στον χρήστη/πελάτη της εφαρμογής να προτείνει ένα ραντεβού με τον ειδικό ψυχικής υγείας του ιατρείου το οποίο του παρείχε πρόσβαση

στην εφαρμογή. Ο χρήστης μπορεί να είναι ήδη πελάτης, δηλαδή μπορεί να έχει ήδη εξεταστεί από τον συγκεκριμένο ειδικό ή απλά να έχει αποκτήσει δικαιώματα στην εφαρμογή.

Ο χρήστης πρέπει να επιλέξει ημερομηνία, ώρα και στη συνέχεια να συμπληρώσει το ονοματεπώνυμό του και το email. Μόλις τα συμπληρώσει αυτά, μπορεί να πατήσει το κουμπί “Confirm”. Πατώντας αυτό το κουμπί, το ραντεβού έχει καταχωρηθεί στη βάση δεδομένων και έπεται είναι δουλειά του ειδικού για το εάν το απορρίψει, επικυρώσει ή κάνει κάποια αλλαγή. Για παράδειγμα, εάν δύο ραντεβού συμπίπτουν την ίδια ημέρα και ώρα, τότε ο ειδικός μπορεί να κάνει κάποιες αλλαγές και φυσικά να ενημερώσει με email τον πελάτη.

Η υπηρεσία αυτή απαγορεύει στον χρήστη να επιλέξει ημερομηνία προηγούμενη της σημερινής, κάτι το οποίο εκτός από αδύνατο να γίνει, θα μπορούσε να δημιουργήσει λανθασμένες προσπάθειες ή ακόμη και παρεξηγήσεις.

Καταχώρηση ραντεβού →

The screenshot shows a web interface for booking an appointment. At the top, there is a navigation bar with a hamburger menu icon on the left and a title bar in the center. Below the title bar, there is a calendar for February 2025. The calendar is a table with columns for the days of the week (Sun, Mon, Tue, Wed, Thu, Fri, Sat) and rows for the days of the month. The dates 1 through 28 are visible. Below the calendar, there is a section titled "Apply Appointment for:". This section contains four input fields: a date field (pre-filled with "--:-- --"), a "Name:" field, a "Last Name:" field, and an "Email:" field. Below these fields is a "Confirm" button. At the bottom of the form, there is a "Return to Home" button.

Εικόνα 43

This screenshot shows the same appointment booking interface as the previous one, but with a confirmation message displayed. The message is a white box with a green border and a green "OK" button. The text inside the box reads: "Ο ιστότοπος localhost:8080 λέει the appointment was set for 2025-02-19". Below the message, the calendar and the "Apply Appointment for:" form are visible. The date field in the form is now pre-filled with "02:22 μμ". The "Name:" field is pre-filled with "John", the "Last Name:" field is pre-filled with "Vairamis", and the "Email:" field is pre-filled with "vair@gmail.com". The "Confirm" button is still present at the bottom of the form, along with the "Return to Home" button.

Εικόνα 44

Το μήνυμα που εμφανίζεται μας επιβεβαιώνει ότι το ραντεβού έχει καταχωρηθεί.

PSYCHOLOGISTS PER LOCATION: Εδώ ο χρήστης μπορεί να δει όλους τους καταχωρημένους ειδικούς που υπάρχουν σε διάφορες περιοχές τις Ελλάδας. Αυτό είναι εξαιρετικά χρήσιμο σε περίπτωση που ο πελάτης θέλει να επισκεφτεί κάποιον άλλον ειδικό ή εάν ο ίδιος βρίσκεται σε άλλη περιοχή από τη δική του.

Psychologists

ID	First name	Specialization	Phone	Email	Location	Action
63	Alexia	2100009222	Neuropsychologist	alexia@gmail.com	Thessaloniki	
9	Alexia	2100009222	Neuropsychologist	alexia@gmail.com	Patra	
64	Alexiasou	2100009222	Neuropsychologist	alexia@hotmail.com	Thessaloniki	
19	Anastasios	2102299223	Health Psychologist	tasos@gmail.com	Kavala	
17	Charalampos	2102753422	Neuropsychologist	char@hotmail.com	Karditsa	
12	Christos	2369447556	Developmental Psychologist	chris@gmai.com	Larisa	
6	Debora	2321747412	Emergency Psychiatrist	deb@hotmail.com	Serres	
18	Dimitris	2369777788	Clinical Psychologist	Dim@gmail.com	Kozani	
23	Eirini	2102245970	General Psychiatrist	vougiou@gmail.com	Santorini	
58	Eulabios	6977928033	Community Psychiatrist	vasilis@gmail.com	Zakinthos	
22	Evelin	2102147444	Developmental Psychologist	kontou@gmail.com	Zakinthos	
24	Fanouria	2102293355	Child and Adolescent Psychiatrist	fan@gmail.com	Piraeus	
38	Fatseas	3302299222	Consultation-Liaison Psychiatrist	fats@gmail.com	Nafplio	
41	Giannis	8756765765	Psyciatrist	john@gmail.com	Serres	
7	Helena	2102299776	Forensic Psychologist	papa@gmail.com	Crete	
5	Hercules	2102238122	Clinical Psychologist	herc@gmail.com	Thessaloniki	
21	Iakwvos	2102299444	Sports Psychologist	iak@gmail.com	Iwannina	
31	Jane	2100479222	Emergency Psychiatrist	jane@gmail.com	Komotini	
55	Jianna	2310334455	Community Psychiatrist	vasilis@gmail.com	Ithaki	
25	Kostas	2102147477	Geriatric Psychiatrist	tsakonas@gmail.com	Ithaki	
.....						
32	Varvara	2102147455	Forensic Psychiatrist	varara@gmail.com	Komotini	
20	Vasiliki	2102299999	Counseling Psychologist	vasiliki@gmail.com	Xanthi	
56	Vasilis	6977928033	Community Psychiatrist	vasilis@gmail.com	Thessaloniki	
37	Xara	2144622422	Psychiatrist	xara@gmail.com	Kalamata	
48	Zaxarias	2448888990	Forensic Psychiatrist	zaxos@gmairiana	Ptolemaida	

[Back Home](#)

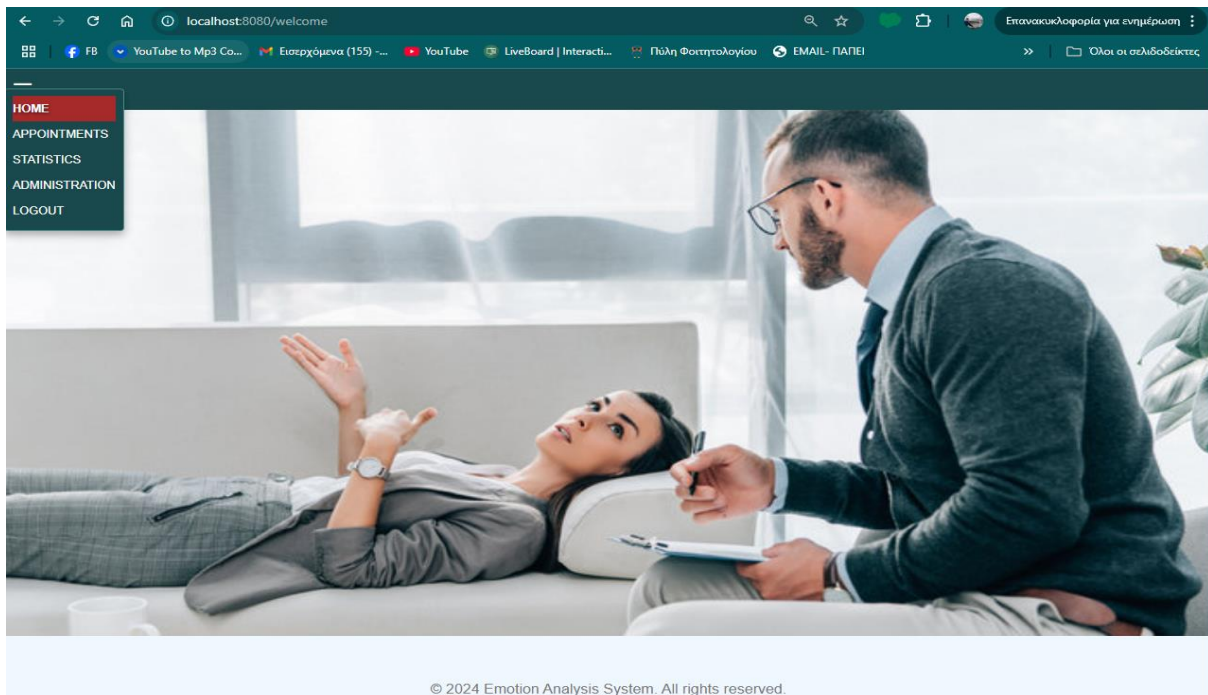
Εικόνα 45

Παρατηρούμε ότι λίστα περιλαμβάνει όλα τα απαραίτητα στοιχεία για τους ειδικούς. Επίσης, είναι πολύ χρήσιμο το γεγονός ότι στα στοιχεία περιλαμβάνεται και η ειδικότητα του ειδικού.

LOGOUT: Σε όλα τα σημεία της πλοήγησης υπάρχει η δυνατότητα της ασφαλούς αποσύνδεσης του χρήστη που μας οδηγεί στην αρχική σελίδα σύνδεσης.

10.2 Εκτέλεση ως διαχειριστής

Όπως και στην περίπτωση του απλού χρήστη, μόλις επιβεβαιωθούν τα στοιχεία του διαχειριστή η εφαρμογή οδηγείται στην κεντρική σελίδα. Οι επιλογές αναφέρονται παρακάτω αναλυτικά.



Εικόνα 46

APPOINTMENTS: Δίνει τη δυνατότητα στο γιατρό να διαγράψει ή να αφήσει ένα ραντεβού καταχωρημένο στη βάση. Επίσης, έχει τη δυνατότητα να δημιουργήσει και ίδιος ένα νέο ραντεβού. Έστω επιλέγουμε την 19 Φεβρουαρίου 2025 στην οποία καταχωρήσαμε στην εικόνα 44 τον χρήστη "John Vairamis".

Appoitments

Add Appointment						
ID	LAST NAME	FIRST NAME	DATE	EMAIL	TIME	ACTION
38	Ploutarxos	Aristofanis	2024-12-02	gmail.com	11:11	Update Delete
39	22222	22222	2024-12-02	22222	14:22	Update Delete
40	333	333	2024-12-09	333	20:19	Update Delete
41	Vogiatzis	Asterios	2024-12-31	asterias@gmail.com	15:15	Update Delete
42	Korais	Athasios	2024-12-09	korahs@hotmail.com	23:21	Update Delete
44	Paparas	Stergios	2024-12-02	@hmail.com	17:55	Update Delete
45	Papadopoulos	Stergios	2024-12-03	papadopoulos@gmail.com	15:06	Update Delete
46	Agelares	Vasilhs	2025-01-06	agelares@gmail.com	17:24	Update Delete
48	TOULIOUDAS	Stergios	2025-02-19	Aa@gmail.com	11:44	Update Delete
52	TOULIOUDAS	STERGIOS	2025-03-08	tselios1990@gmail.com	10:49	Update Delete
53	Vairamis	John	2025-02-19	vair@gmail.com	11:44	Update Delete

[Back](#)

Εικόνα 47

Σε αυτό το σημείο ο διαχειριστής μπορεί να προσθέσει-αλλάξει ή διαγράψει κάποιο ραντεβού πατώντας απλά τα ανάλογα κουμπιά που υπάρχουν. Εάν πατήσει διαγραφή(«DELETE»), πρώτα του

εμφανίζεται ένα μήνυμα επιβεβαίωσης και στη συνέχεια, εφόσον επιβεβαιώσει την πρόθεσή του, το μήνυμα επιτυχίας.

Έστω ότι θέλουμε να διαγράψουμε το ραντεβού με αναγνωριστικό κωδικό(«ID») 53:

Ο ιστότοπος localhost:8080 λέει
Are you sure you want to delete this appointment?

OK Ακύρωση

Add Appointment

ID	LAST NAME	FIRST NAME	DATE	EMAIL	TIME	ACTION
38	Ploutarxos	Aristofanis	2024-12-02	gmail.com	11:11	Update Delete
39	22222	22222	2024-12-02	22222	14:22	Update Delete
40	333	333	2024-12-09	333	20:19	Update Delete
41	Vogiatzis	Asterios	2024-12-31	asterias@gmail.com	15:15	Update Delete
42	Korais	Athasios	2024-12-09	korahs@hotmail.com	23:21	Update Delete
44	Paparas	Stergios	2024-12-02	@hmail.com	17:55	Update Delete
45	Papadopoulos	Stergios	2024-12-03	papadopoulos@gmail.com	15:06	Update Delete
46	Agelares	Vasilhs	2025-01-06	agelares@gmail.com	17:24	Update Delete
48	TOULIOUDAS	Stergios	2025-02-19	Aa@gmail.com	11:44	Update Delete
52	TOULIOUDAS	STERGIOS	2025-03-08	tselios1990@gmail.com	10:49	Update Delete
53	Vairamis	John	2025-02-19	vair@gmail.com	11:44	Update Delete

[Back](#)

Appoitments

Appointment deleted successfully!!

Add Appointment

ID	LAST NAME	FIRST NAME	DATE	EMAIL	TIME	ACTION
38	Ploutarxos	Aristofanis	2024-12-02	gmail.com	11:11	Update Delete
39	22222	22222	2024-12-02	22222	14:22	Update Delete
40	333	333	2024-12-09	333	20:19	Update Delete
41	Vogiatzis	Asterios	2024-12-31	asterias@gmail.com	15:15	Update Delete
42	Korais	Athasios	2024-12-09	korahs@hotmail.com	23:21	Update Delete
44	Paparas	Stergios	2024-12-02	@hmail.com	17:55	Update Delete
45	Papadopoulos	Stergios	2024-12-03	papadopoulos@gmail.com	15:06	Update Delete
46	Agelares	Vasilhs	2025-01-06	agelares@gmail.com	17:24	Update Delete

[Back](#)

Στην περίπτωση που ο ειδικός/γιατρός θέλει να προσθέσει ένα ραντεβού, επιλέγει το κουμπί «Add Approitment» που βρίσκεται πάνω αριστερά, και έπειτα ανοίγει μία νέα φόρμα στην οποία εισάγει τα πεδία που απαιτούνται:



New Appointment

[Back to Appointment](#)

Εάν ο ειδικός επιθυμεί να τροποποιήσει κάποιο ραντεβού, τότε επιλέγει το κουμπί «Update» και ανοίγει η ίδια φόρμα με την προηγούμενη, με τη διαφορά ότι τα στοιχεία του ασθενή είναι ήδη συμπληρωμένα:



Update Appointments

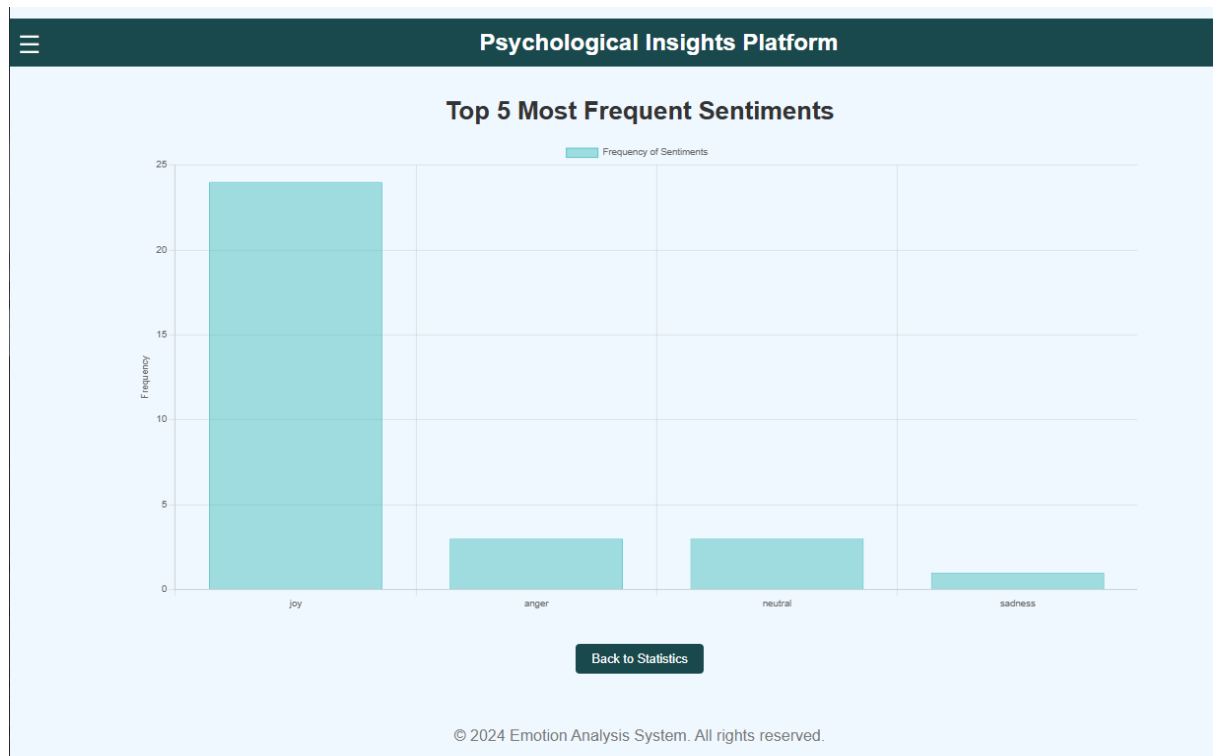
[Back to Appointments](#)

Σε κάθε περίπτωση αλλαγής, θεωρείται σωστό ο ειδικός να ενημερώνει τον πελάτη του είτε με τηλέφωνο είτε με ηλεκτρονικό ταχυδρομείο για οποιαδήποτε αλλαγή προκύψει.

STATISTICS: Η εφαρμογή έχει τη δυνατότητα υπολογισμού στατιστικών δεδομένων τα οποία προκύπτουν από τις διαγνώσεις των χρηστών και τα συναισθήματα που προκύπτουν από αυτά. Έτσι λοιπόν, μπορούμε να λάβουμε αποτελέσματα σχετικά με το ποιο είναι το κυρίαρχο συναίσθημα των πελατών, πόσοι από αυτούς είναι άνδρες και πόσες γυναίκες κ.ά.

Στην κεντρική σελίδα των στατιστικών έχουμε ενδεικτικά τις επιλογές «MOST DOMINANT SENTIMENT» ΚΑΙ «PERCENTAGE BY GENDER»

«MOST DOMINANT SENTIMENT»



Εικόνα 48

«PERCENTAGE BY GENDER»

The figure is a table titled "Sentiment Percentage by Gender" from the "Psychological Insights Platform". The table has three columns: Gender, Sentiment, and Percentage (%). The data is as follows:

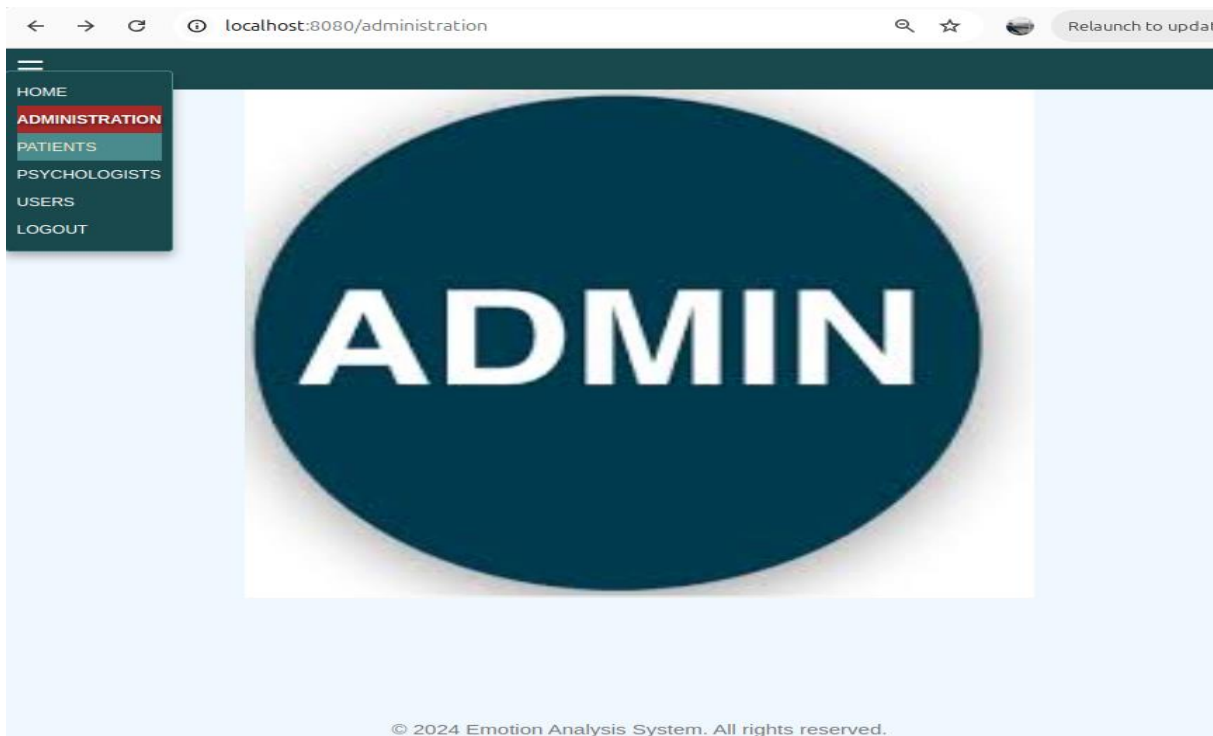
Gender	Sentiment	Percentage (%)
Male	joy	77
Male	anger	10
Male	neutral	10
Male	sadness	3

A "Back to Statistics" button is at the bottom, and the copyright notice "© 2024 Emotion Analysis System. All rights reserved." is at the bottom right.

Εικόνα 49

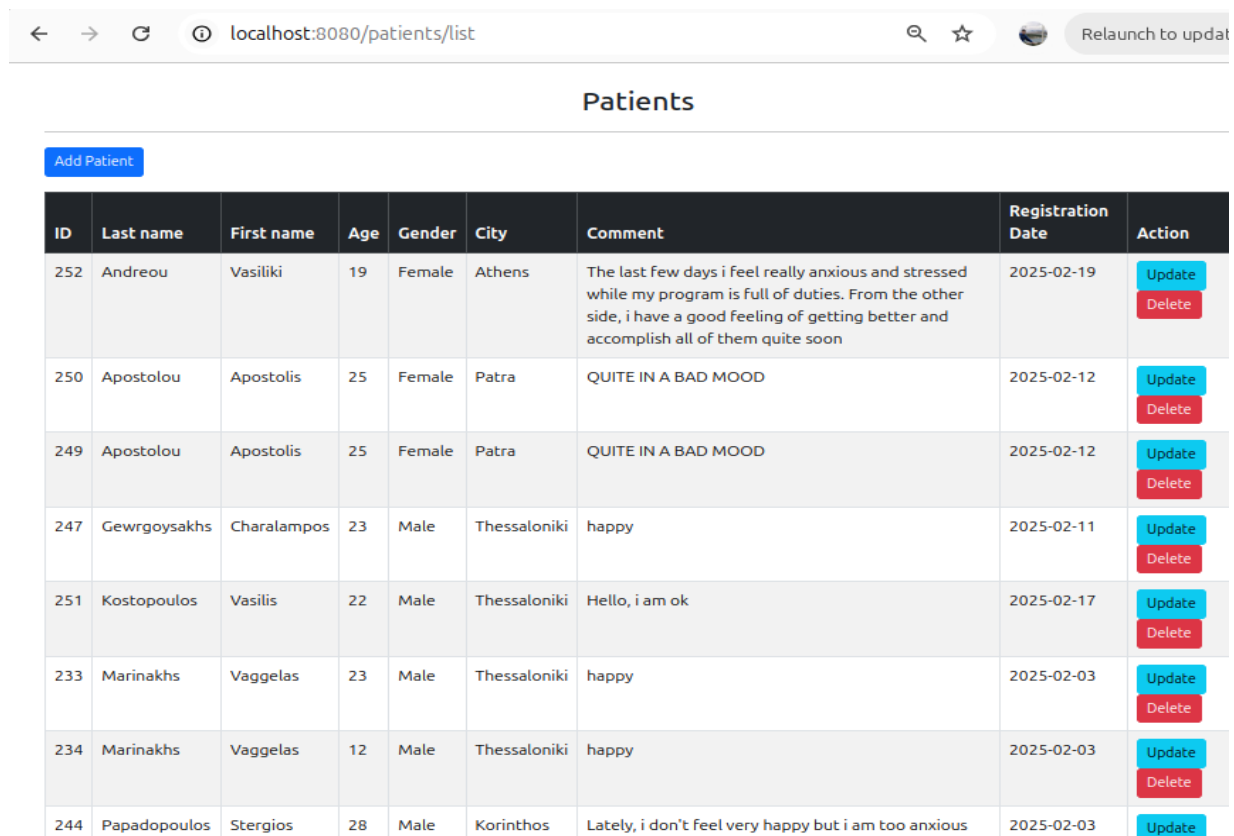
ADMINISTRATION: Η εφαρμογή παραχωρεί το δικαίωμα στον διαχειριστή για επεξεργασία των ασθενών(Patient), των ειδικών ψυχικής υγείας(Psychologist) αλλά και των χρηστών(User) που είναι καταχωρημένοι στη βάση της εφαρμογής. Να διευκρινήσουμε σε αυτό το σημείο ότι οι ασθενείς είναι ξεχωριστή οντότητα, επομένως αποτελεί ξεχωριστό πίνακα στη βάση, από τους χρήστες. Ένας ασθενής καταχωρείται μόνο όταν λαμβάνει μία διάγνωση. Οι κωδικοί user που έχει του επιτρέπουν απλά την είσοδο στην εφαρμογή.

Βρισκόμαστε στη κεντρική σελίδα διαχείρισης. Από εδώ υπάρχει η δυνατότητα επιλογής της μονάδας που θέλουμε να επεξεργαστούμε. Έστω επιλέγουμε να επεξεργαστούμε έναν "Patient":



Εικόνα 50

Η εφαρμογή μας πλοηγεί σε μία νέα σελίδα και μας εμφανίζει όλους τους ασθενείς που είναι καταχωρημένοι.

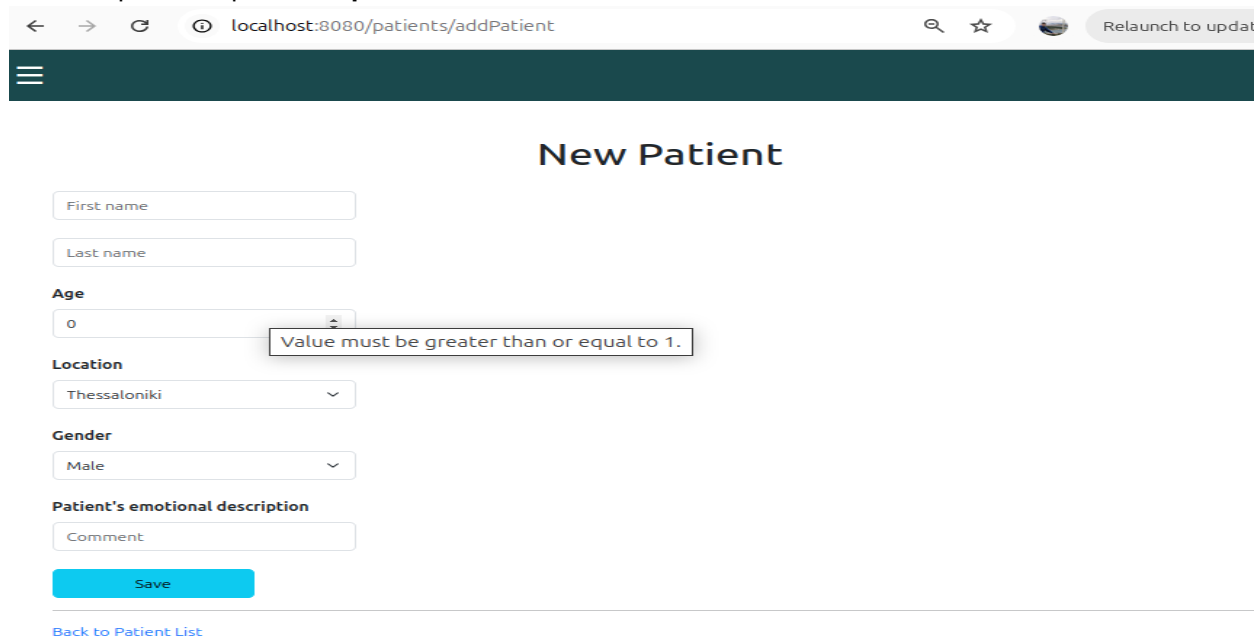


ID	Last name	First name	Age	Gender	City	Comment	Registration Date	Action
252	Andreou	Vasiliki	19	Female	Athens	The last few days i feel really anxious and stressed while my program is full of duties. From the other side, i have a good feeling of getting better and accomplish all of them quite soon	2025-02-19	Update Delete
250	Apostolou	Apostolis	25	Female	Patra	QUITE IN A BAD MOOD	2025-02-12	Update Delete
249	Apostolou	Apostolis	25	Female	Patra	QUITE IN A BAD MOOD	2025-02-12	Update Delete
247	Gewrgoyasakhs	Charalampos	23	Male	Thessaloniki	happy	2025-02-11	Update Delete
251	Kostopoulos	Vasilis	22	Male	Thessaloniki	Hello, i am ok	2025-02-17	Update Delete
233	Marinakhs	Vaggelas	23	Male	Thessaloniki	happy	2025-02-03	Update Delete
234	Marinakhs	Vaggelas	12	Male	Thessaloniki	happy	2025-02-03	Update Delete
244	Papadopoulos	Stergios	28	Male	Korinthos	Lately, i don't feel very happy but i am too anxious	2025-02-03	Update

Εικόνα 51

Παρατηρούμε ότι πάνω αριστερά υπάρχει το κουμπί “Add patient” και δεξιά από κάθε εγγραφή τα κουμπιά “Update” και “Delete”.

✓ Επιλέγω το κουμπί “Add patient”:



First name

Last name

Age

0

Value must be greater than or equal to 1.

Location

Thessaloniki

Gender

Male

Patient's emotional description

Comment

Save

[Back to Patient List](#)

Εικόνα 52

Εάν συμπληρώσω τα πεδία και πατήσω το κουμπί “Save”, ένας νέος χρήστης αποθηκεύεται.

✓ Επιλέγω το κουμπί “Update”

Update Patient

Vasiliki

Andreou

Age

19

Location

Thessaloniki

Gender

Female

Patient's emotional description

The last few days i feel really anxioi

Save

[Back to Patient List](#)

Εικόνα 53

Εμφανίζεται η φόρμα με συμπληρωμένα τα στοιχεία του συγκεκριμένου ασθενή. Εάν αλλάξουμε το περιεχόμενο κάποιου πεδίου και το αποθηκεύσουμε τότε γίνεται ενημέρωση του συγκεκριμένου ασθενή στη βάση.

Έστω λοιπόν, ότι αλλάζουμε το όνομα και την ηλικία του συγκεκριμένου ασθενή από “Vasiliki” σε “Helena” και από “19” σε “35”.

Patients

Patient updated successfully!!

Add Patient

ID	Last name	First name	Age	Gender	City	Comment	Registration Date	Action
252	Andreou	Helena	35	Female	Thessaloniki	The last few days i feel really anxious and stressed while my program is full of duties. From the other side, i have a good feeling of getting better and accomplish all of them quite soon	2025-02-19	Update Delete
250	Apostolou	Apostolis	25	Female	Patra	QUITE IN A BAD MOOD	2025-02-12	Update Delete
249	Apostolou	Apostolis	25	Female	Patra	QUITE IN A BAD MOOD	2025-02-12	Update Delete

Εικόνα 54

Εάν παρατηρήσουμε, η ενημέρωση έχει γίνει και εμφανίστηκε το αντίστοιχο μήνυμα.

✓ Επιλέγω το κουμπί “Delete”

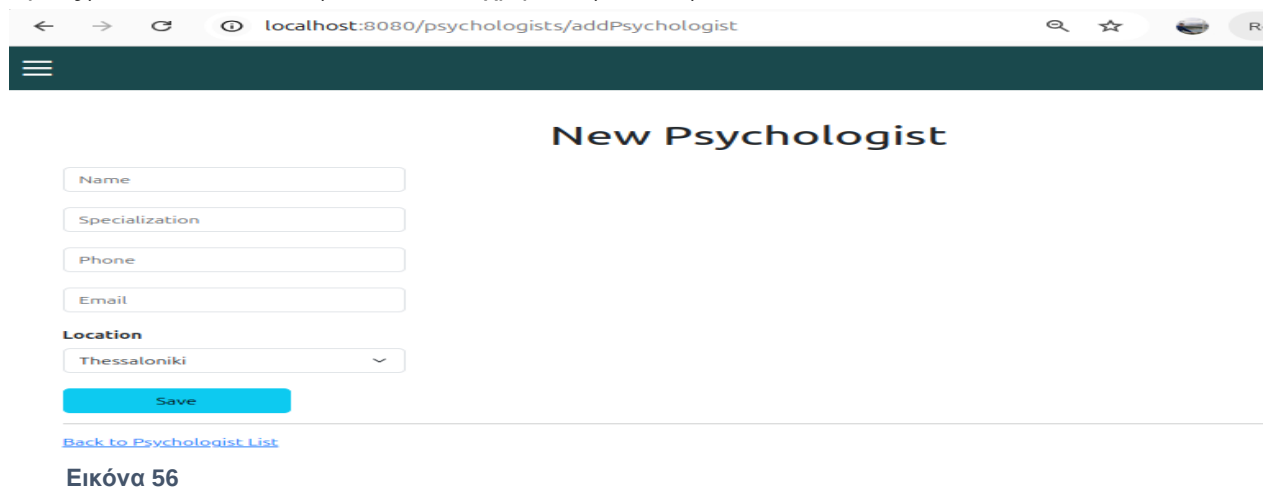
Έστω ότι επιλέγω να σβήσω το χρήστη με κωδικό “250”(σειρά, όνομα: Apostolou Apostolis)



ID	Last name	First name	Age	Gender	City	Comment	Registration Date	Action
252	Andreou	Helena	35	Female	Thessaloniki	The last few days i feel really anxious and stressed while my program is full of duties. From the other side, i have a good feeling of getting better and accomplish all of them quite soon	2025-02-19	Update Delete
249	Apostolou	Apostolis	25	Female	Patra	QUITE IN A BAD MOOD	2025-02-12	Update Delete
247	Gewrgoysakhs	Charalampos	23	Male	Thessaloniki	happy	2025-02-11	Update Delete

Εικόνα 55

Αντίστοιχη διαδικασία είναι και για την επεξεργασία τόσο των ειδικών ψυχικής υγείας(PSYCHOLOGISTS) όσο και των χρηστών(USERS).



New Psychologist

Name

Specialization

Phone

Email

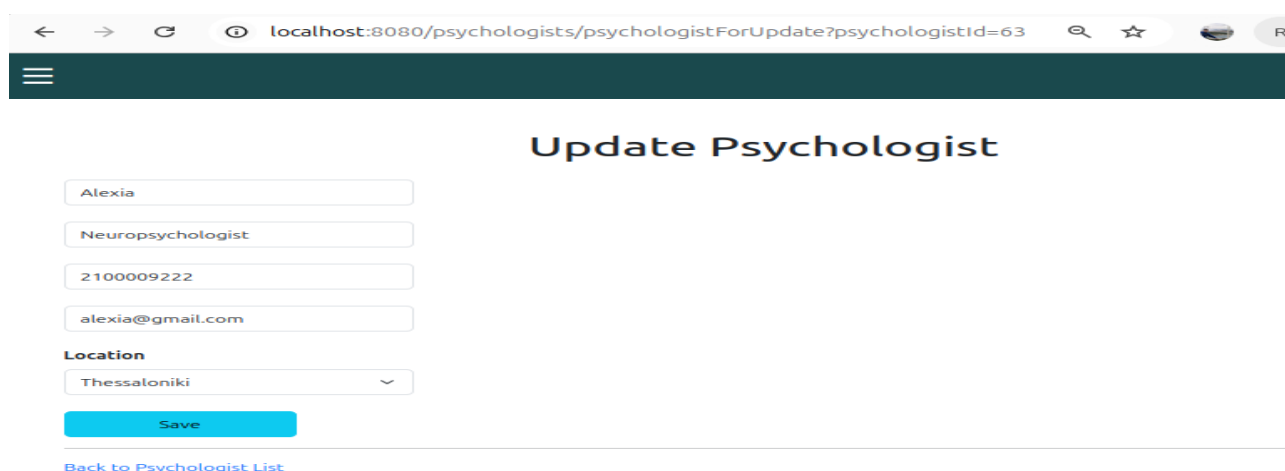
Location

Thessaloniki

[Save](#)

[Back to Psychologist List](#)

Εικόνα 56



Update Psychologist

Alexia

Neuropsychologist

2100009222

alexia@gmail.com

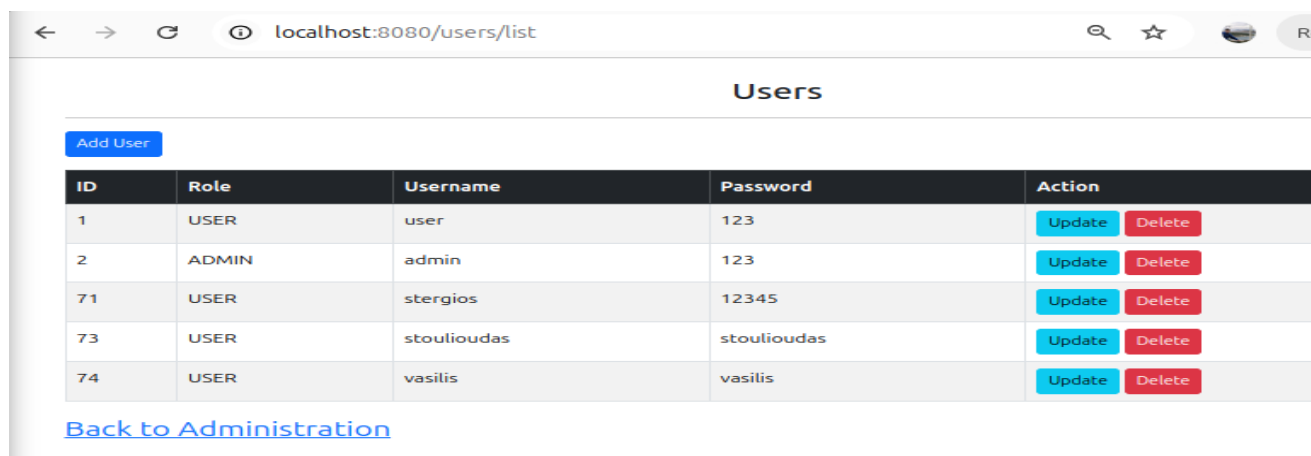
Location

Thessaloniki

[Save](#)

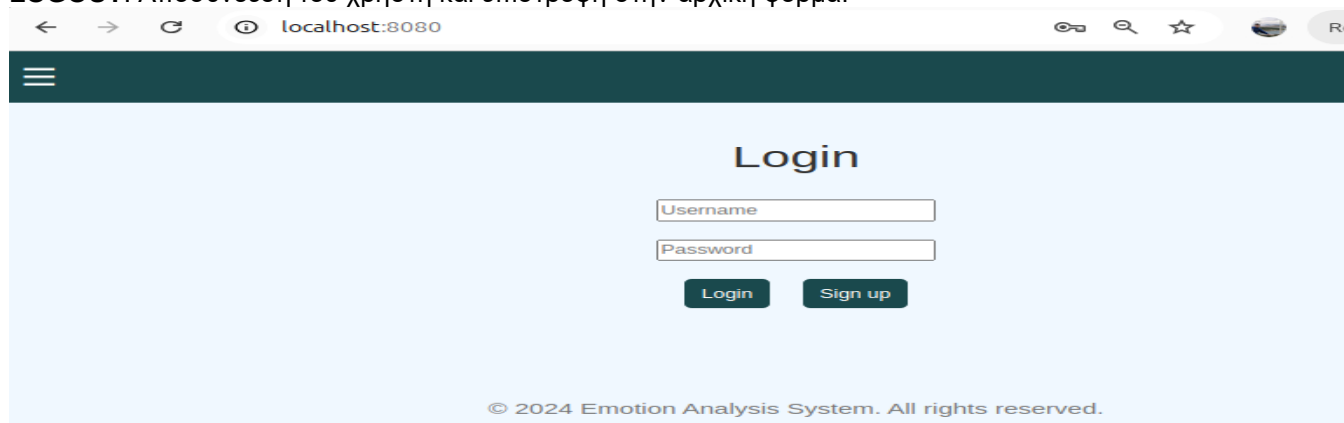
[Back to Psychologist List](#)

Εικόνα 57



Εικόνα 58

LOGOUT: Αποσύνδεση του χρήστη και επιστροφή στην αρχική φόρμα.



Εικόνα 59

11 Συμπεράσματα

Η παρούσα εργασία ανέδειξε τη σημασία της ανάλυσης συναισθήματος ως ένα ισχυρό εργαλείο στον τομέα της ψυχικής υγείας. Σήμερα, δεν υπάρχουν πολλές εφαρμογές που να προσφέρουν μια ολοκληρωμένη προσέγγιση στην παρακολούθηση των συναισθημάτων των χρηστών, ενώ παράλληλα να παρέχουν τη δυνατότητα αξιολόγησης, προγραμματισμού ραντεβού και ανάλυσης δεδομένων. Η εφαρμογή που αναπτύχθηκε καλύπτει αυτό το κενό, επιτρέποντας στους χρήστες να λαμβάνουν μια πρώτη εκτίμηση για την ψυχική τους υγεία, με γρήγορο, εύκολο και οικονομικό τρόπο, καθώς απαιτεί μόνο ένα μικρό ποσό εγγραφής αντί για το κόστος ενός παραδοσιακού ραντεβού με ειδικό.

Από την πλευρά των γιατρών, η εφαρμογή προσφέρει έναν πρωτοποριακό τρόπο παρακολούθησης των ασθενών, καθώς μπορούν να αναλύουν τα συναισθηματικά δεδομένα που συλλέγονται μέσω της πλατφόρμας. Με αυτόν τον τρόπο, αποκτούν μια πιο ολοκληρωμένη εικόνα για την ψυχική κατάσταση των ασθενών τους, βασισμένη σε στατιστικά δεδομένα και διαχρονικές αναλύσεις. Επιπλέον, διευκολύνεται η διαχείριση των ραντεβού, βελτιώνοντας τη συνολική εμπειρία τόσο για τους ασθενείς όσο και για τους ειδικούς.

Η υλοποίηση της εφαρμογής βασίστηκε στο Spring Boot, το οποίο προσέφερε μια ευέλικτη, αξιόπιστη και επεκτάσιμη αρχιτεκτονική. Με τη χρήση του Thymeleaf δημιουργήθηκε μια διαδραστική διεπαφή, ενώ το Spring Data JPA βελτιστοποίησε τη διαχείριση της βάσης δεδομένων. Η Service-Oriented Architecture (SOA) επέτρεψε τη διάσπαση της εφαρμογής σε ανεξάρτητες υπηρεσίες, αυξάνοντας την επαναχρησιμοποίηση του κώδικα και διευκολύνοντας τη μελλοντική επέκταση του

συστήματος.

Συνολικά, η εφαρμογή αποτελεί ένα καινοτόμο εργαλείο που αξιοποιεί σύγχρονες τεχνολογίες για την προαγωγή της ψυχικής υγείας, προσφέροντας αποτελεσματικές, προσβάσιμες και οικονομικά προσιτές λύσεις τόσο για τους επαγγελματίες όσο και για τους χρήστες.

12 Βιβλιογραφία

- [1] Alepis, E., & Virvou, M. (2011). Automatic generation of emotions in tutoring agents for affective e-learning in medical education. *Expert Systems with Applications*, 38(8), 9840-9847.
- [2] Virvou, Maria. "Artificial Intelligence and User Experience in reciprocity: Contributions and state of the art." *Intelligent Decision Technologies* 17.1 (2023): 73-125.
- [3] Chrysafiadi, K., Virvou, M., Tsihrintzis, G. A., & Hatzilygeroudis, I. (2023). An adaptive learning environment for programming based on fuzzy logic and machine learning. *International Journal on Artificial Intelligence Tools*, 32(05), 2360011.
- [4] Alepis, E., & Lambrinidis, C. (2013). M-health: supporting automated diagnosis and electronic health records. *SpringerPlus*, 2, 1-9.
- [5] Papageorgiou, A., Strigkos, M., Politou, E., Alepis, E., Solanas, A., & Patsakis, C. (2018). Security and privacy analysis of mobile health applications: the alarming state of practice. *Ieee Access*, 6, 9390-9403.
- [6] Troussas, C., Virvou, M., Espinosa, K. J., Llaguno, K., & Caro, J. (2013, July). Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning. In *IISA 2013* (pp. 1-6). IEEE.
- [7] Riadsolh, A., Lasri, I., & ElBelkacemi, M. (2020). Cloud-Based Sentiment Analysis for Measuring Customer Satisfaction in the Moroccan Banking Sector Using Na? ve Bayes and Stanford NLP. *Journal of Automation, Mobile Robotics and Intelligent Systems*, 64-71.
- [8] Agüero-Torales, M. M., Cobo, M. J., Herrera-Viedma, E., & López-Herrera, A. G. (2019). A cloud-based tool for sentiment analysis in reviews about restaurants on TripAdvisor. *Procedia Computer Science*, 162, 392-399.
- [9] Khan, M. T., Durrani, M., Ali, A., Inayat, I., Khalid, S., & Khan, K. H. (2016). Sentiment analysis and the complex natural language. *Complex Adaptive Systems Modeling*, 4, 1-19.
- [10] Luan, X. (2021). Implementation and analysis of software development in spring boot. *Doctoral dissertation*.
- [11] Spring Boot , Διαδικτυακός τόπος : <https://www.baeldung.com/spring-boot>.
- [12] Java 12, 8th edition, Rogers CadenHead, εκδόσεις Γκιούρδας
- [13] [NEW] Spring Boot 3, Spring 6 & Hibernate for Begginers. UDEMY, [Online]. Available: <https://www.udemy.com/course/spring-hibernate-tutorial/>
- [14] "Maven – Introduction" Apache.org. [Online]. Available: <https://maven.apache.org/what-is-maven.html>.
- [15] "MVC Design Pattern", Geeksforgeeks.org, 18 Aug 2017. [Online] Available: <https://www.geeksforgeeks.org/mvc-design-pattern/>.