



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Ανάπτυξη πλατφόρμας διαχείρισης νοσηλευτικού
προσωπικού με αλγορίθμους βελτιστοποίησης»

UltiNurse

Των φοιτητών
Στέφανου Παπανάτσιου it174902
Ορφέα-Ιωάννη Παντελιά-Γκιώνη it174951

Επιβλέπων
Παναγιώτης Τζέκης

Μάιος 2025

Τίτλος Π.Ε.

Ανάπτυξη πλατφόρμας διαχείρισης νοσηλευτικού προσωπικού με αλγορίθμους βελτιστοποίησης

Κωδικός Π.Ε.

24278

Όνοματεπώνυμο φοιτητών

**Στέφανος Παπανάτσιος
Ορφέας-Ιωάννης Παντελιάς-Γκιώνης**

Όνοματεπώνυμο εισηγητή

Τζέκης Παναγιώτης

Ημερομηνία ανάληψης Π.Ε.

Οκτώβριος 2024

Ημερομηνία περάτωσης Π.Ε.

Μάιος 2025

Βεβαιώνουμε ότι είμαστε οι συγγραφείς αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχαμε για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχουμε καταγράψει τις όποιες πηγές από τις οποίες κάναμε χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνουμε ότι αυτή η εργασία προετοιμάστηκε από εμάς προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία των φοιτητών **Στέφανου Παπανάτσιου** και **Ορφέα-Ιωάννη Παντελιά-Γκιώνη** που την εκπόνησαν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, οι συγγραφείς/δημιουργοί εκχωρούν στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση των συγγραφέων/δημιουργών.*

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Η επιλογή του συγκεκριμένου ερευνητικού πεδίου για την πτυχιακή μας εργασία πηγάζει από το έντονο ενδιαφέρον μας για την αξιοποίηση της τεχνολογίας προς όφελος του τομέα της υγείας και, ειδικότερα, για την αντιμετώπιση των σύγχρονων προκλήσεων που επηρεάζουν την παροχή φροντίδας. Η προοπτική ανάπτυξης μιας καινοτόμου λύσης που θα μπορούσε να συμβάλει στην αποτελεσματικότερη οργάνωση, στη βελτίωση της ποιότητας των παρεχόμενων υπηρεσιών και στην υποστήριξη των επαγγελματιών υγείας, αποτέλεσε ισχυρό κίνητρο.

Κατά την εκπόνηση της εργασίας, αποκομίσαμε πολύτιμες γνώσεις στον σχεδιασμό και την εφαρμογή αλγορίθμων βελτιστοποίησης σε πραγματικά προβλήματα, καθώς και στην ανάλυση δεδομένων στον ευαίσθητο χώρο της υγείας. Εμβαθύναμε, επίσης, στην κατανόηση των σύνθετων αναγκών που διέπουν τη στελέχωση των νοσηλευτικών μονάδων. Το σημαντικότερο όφελος, ωστόσο, ήταν η ευκαιρία να διερευνήσουμε και να προτείνουμε μια λύση με δυνητικά θετικό αντίκτυπο στην αποδοτικότητα του συστήματος υγείας και στην καθημερινότητα τόσο των ασθενών όσο και του νοσηλευτικού προσωπικού.

Περίληψη

Η παρούσα πτυχιακή εργασία εστιάζει στη δημιουργία μιας καινοτόμου πλατφόρμας που στοχεύει στη βελτιστοποίηση της διαχείρισης του νοσηλευτικού προσωπικού. Η εφαρμογή αξιοποιεί αλγορίθμους βελτιστοποίησης για να αναλύει δεδομένα σχετικά με τις ανάγκες των ασθενών και τη διαθεσιμότητα των νοσηλευτών. Με αυτόν τον τρόπο, επιτυγχάνεται η αποτελεσματική κατανομή του προσωπικού, εξασφαλίζοντας ότι οι κατάλληλοι επαγγελματίες είναι διαθέσιμοι τη σωστή στιγμή. Η πλατφόρμα στοχεύει στο να συμβάλει στη βελτίωση της ποιότητας των παρεχόμενων ιατρικών υπηρεσιών, μειώνοντας ταυτόχρονα την πίεση που ασκείται στο προσωπικό και ενισχύοντας τη συνολική αποδοτικότητα του υγειονομικού συστήματος.

«Development of a nursing staff management platform
with optimization algorithms»

Stefanos Papanatsios

Orfeas-Ioannis Pantelias-Gkionis

Abstract

This thesis focuses on the creation of an innovative platform that aims to optimize the management of nursing staff. The application will utilize optimization algorithms to analyze data on patient needs and nurse availability. In this way, efficient staff allocation will be achieved, ensuring that the right professionals are available at the right time. The platform will contribute to improving the quality of medical services provided, while reducing pressure on staff and enhancing the overall efficiency of the healthcare system.

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τον καθηγητή και επιβλέπων της πτυχιακής εργασίας κ. Παναγιώτη Τζέκη, που μας επέτρεψε να αναλάβουμε και να ασχοληθούμε με αυτό το αντικείμενο, καθώς και για την καθοδήγηση και βοήθεια του καθ' όλη τη διάρκεια της εκπόνησης.

Θα θέλαμε επίσης να ευχαριστήσουμε τις οικογένειες μας και τους φίλους μας για την οικονομική, ψυχολογική, καθώς και κάθε άλλη μορφής βοήθεια και στήριξη που μας έχουν προσφέρει.

Περιεχόμενα

Πρόλογος.....	3
Περίληψη.....	4
Abstract.....	5
Ευχαριστίες.....	6
Περιεχόμενα.....	7
Κατάλογος Σχημάτων.....	10
Κατάλογος Πινάκων.....	11
Συντομογραφίες.....	12
Κεφάλαιο 1ο: Εισαγωγή.....	13
1.1 Εισαγωγή.....	13
1.2 Πρόλογος.....	13
1.3 Πρόβλημα.....	13
1.4 Σκοποί.....	14
1.5 Στόχοι.....	14
1.6 Δομή.....	15
1.7 Επίλογος.....	16
Κεφάλαιο 2ο: Τεχνολογίες.....	17
2.1 Εισαγωγή.....	17
2.2 Βάση δεδομένων.....	17
2.2.1 PostgreSQL.....	17
2.2.2 PostGIS.....	17
2.2.3 Neon.....	18
2.3 Γλώσσα προγραμματισμού.....	18
2.3.1 TypeScript.....	18
2.4 Backend.....	19
2.4.1 Node.js.....	19
2.4.2 Drizzle ORM.....	20
2.4.3 tRPC.....	20
2.5 Frontend.....	21
2.5.1 React.js.....	21
2.5.2 Next.js.....	22
2.5.3 Tailwind CSS.....	22
2.6 Επίλογος.....	23

Κεφάλαιο 3ο: Αλγόριθμοι.....	24
3.1 Εισαγωγή.....	24
3.2 Άπληστος.....	24
3.2.1 Εισαγωγή.....	24
3.2.2 Ιστορικό πλαίσιο.....	24
3.2.3 Πλαίσιο εφαρμογής.....	24
3.2.4 Μαθηματική διατύπωση.....	25
3.2.5 Θεωρητικές εκτιμήσεις.....	26
3.2.6 Υλοποίηση στον κώδικα.....	26
3.3 WSM.....	27
3.3.1 Εισαγωγή.....	27
3.3.2 Ιστορικό πλαίσιο.....	28
3.3.3 Πλαίσιο εφαρμογής.....	28
3.3.4 Μαθηματική διατύπωση.....	28
3.3.5 Θεωρητικές εκτιμήσεις.....	29
3.3.6 Υλοποίηση στον κώδικα.....	29
3.4 Simulated Annealing.....	31
3.4.1 Εισαγωγή.....	31
3.4.2 Ιστορικό πλαίσιο.....	31
3.4.3 Πλαίσιο εφαρμογής.....	31
3.4.4 Μαθηματική διατύπωση.....	31
3.4.5 Εκτίμηση και προοπτικές.....	32
3.4.6 Υλοποίηση στον κώδικα.....	32
3.5 Tabu Search.....	34
3.5.1 Εισαγωγή.....	34
3.5.2 Ιστορικό πλαίσιο.....	34
3.5.3 Πλαίσιο εφαρμογής.....	34
3.5.4 Μαθηματική διατύπωση.....	34
3.5.5 Εκτίμηση και προοπτικές.....	35
3.5.6 Υλοποίηση στον κώδικα.....	35
3.6 TOPSIS.....	37
3.6.1 Εισαγωγή.....	37
3.6.2 Ιστορικό πλαίσιο.....	37
3.6.3 Πλαίσιο εφαρμογής.....	38
3.6.4 Πλαίσιο εφαρμογής.....	38

3.6.5	Υλοποίηση στον Κώδικα.....	39
3.6.6	Σύνοψη αλγοριθμικής συμπεριφοράς.....	40
3.7	Επίλογος.....	41
Κεφάλαιο 4ο: Πλατφόρμα.....		42
4.1	Εισαγωγή.....	42
4.2	Σχεδιασμός βάσης δεδομένων.....	42
4.2.1	Κύριοι πίνακες.....	42
4.2.2	Αυτόματα παραχθέντες πίνακες.....	44
4.3	Frontend.....	46
4.3.1	Αρχική οθόνη.....	46
4.3.2	Σύνδεση.....	46
4.3.3	Σελίδα ασθενών.....	48
4.3.4	Δημιουργία ασθενούς.....	49
4.3.5	Σελίδα βαρδιών.....	51
4.3.6	Δημιουργία βάρδιας.....	52
4.3.7	Ανάθεση νοσηλευτικού προσωπικού σε βάρδια.....	54
4.3.8	Μαζική ανάθεση νοσηλευτικού προσωπικού σε βάρδιες.....	56
4.4	Επίλογος.....	59
Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης.....		60
5.1	Συμπεράσματα.....	60
5.2	Προτάσεις βελτίωσης.....	63
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		64

Κατάλογος Σχημάτων

Σχήμα 2.1: Λογότυπο PostgreSQL.....	17
Σχήμα 2.2: Λογότυπο PostGIS.....	18
Σχήμα 2.3: Λογότυπο Neon.....	18
Σχήμα 2.4: Λογότυπο TypeScript.....	19
Σχήμα 2.5: Λογότυπο Node.js.....	19
Σχήμα 2.6: Λογότυπο Drizzle.....	20
Σχήμα 2.7: Λογότυπο tRPC.....	21
Σχήμα 2.8: Λογότυπο React.....	21
Σχήμα 2.9: Λογότυπο Next.js.....	22
Σχήμα 2.10: Λογότυπο Tailwind CSS.....	22
Σχήμα 4.1: Διάγραμμα Οντοτήτων-Συσχετίσεων (ΟΣΔ) κύριων πινάκων.....	44
Σχήμα 4.2: Διάγραμμα Οντοτήτων-Συσχετίσεων (ΟΣΔ) πινάκων της βιβλιοθήκης Auth.js.....	45
Σχήμα 4.3: Διάγραμμα Οντοτήτων-Συσχετίσεων (ΟΣΔ) πινάκων της επέκτασης PostGIS.....	45
Σχήμα 4.4: Αρχική οθόνη εφαρμογής σε πλήρες μέγεθος οθόνης.....	46
Σχήμα 4.5: Κουμπί σύνδεσης με GitHub.....	46
Σχήμα 4.6: Σελίδα ασθενών σε κινητό τηλέφωνο.....	48
Σχήμα 4.7: Παράθυρο διαλόγου δημιουργίας νέου ασθενούς.....	49
Σχήμα 4.8: Ροή δημιουργίας νέου ασθενούς.....	50
Σχήμα 4.9: Σελίδα βαρδιών σε τάμπλετ.....	51
Σχήμα 4.10: Παράθυρο διαλόγου δημιουργίας νέας βάρδιας.....	52
Σχήμα 4.11: Ροή δημιουργίας νέας βάρδιας.....	53
Σχήμα 4.12: Παράθυρο διαλόγου ανάθεσης νοσηλευτικού προσωπικού σε βάρδια.....	54
Σχήμα 4.13: Ροή ανάθεσης νοσηλευτικού προσωπικού σε βάρδια.....	55
Σχήμα 4.14: Παράθυρο διαλόγου μαζικής ανάθεσης νοσηλευτικού προσωπικού σε βάρδιες.....	56
Σχήμα 4.15: Ροή μαζικής ανάθεσης νοσηλευτικού προσωπικού σε βάρδιες.....	58

Κατάλογος Πινάκων

Πίνακας 3.1: Κριτήρια άπληστου αλγορίθμου και η υλοποίησή τους στην Π.Ε.....	27
Πίνακας 3.2: Πίνακας συμμόρφωσης με αρχές Simulated Annealing.....	33
Πίνακας 3.3: Κριτήρια αλγορίθμου Tabu και η υλοποίηση τους στην Π.Ε.....	37
Πίνακας 3.4: Κριτήρια αλγορίθμου TOPSIS και η υλοποίησή τους στην Π.Ε.....	41
Πίνακας 5.1: Συγκριτική αξιολόγηση αλγορίθμων με σειριακό αλγόριθμο εκτέλεσης.....	60
Πίνακας 5.2: Συγκριτική αξιολόγηση αλγορίθμων με αλγόριθμο εκτέλεσης Tabu.....	61
Πίνακας 5.3: Συγκριτική αξιολόγηση αλγορίθμων με αλγόριθμο εκτέλεσης Simulated Annealing.....	61
Πίνακας 5.4: Συγκριτική αξιολόγηση βαρών με WSM και σειριακό αλγόριθμο εκτέλεσης.....	62

Συντομογραφίες

ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
ΟΣΔ	Διάγραμμα Οντοτήτων-Συσχετίσεων
Π.Ε.	Πτυχιακή Εργασία
ΠΟΕΔΗΝ	Πανελλήνια Ομοσπονδία Εργαζομένων Δημοσίων Νοσοκομείων
ΠΟΥ	Παγκόσμιος Οργανισμός Υγείας
AHP	Analytic Hierarchy Process
ICN	International Council of Nurses
MCDM	Multiple-Criteria Decision-Making
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
WSM	Weighted Sum Model

Κεφάλαιο 1ο: Εισαγωγή

1.1 Εισαγωγή

Σε αυτό το κεφάλαιο γίνεται μια εισαγωγική αναφορά στο θέμα της πτυχιακής εργασίας, τους σκοπούς πραγματοποίησής της, τους στόχους στους οποίους αποσκοπεί η περάτωσή της, καθώς και τη δομή που θα ακολουθηθεί.

1.2 Πρόλογος

Από το 2023, ο Παγκόσμιος Οργανισμός Υγείας (ΠΟΥ) εκτιμά ότι θα υπάρξει έλλειψη περίπου 4,5 εκατομμυρίων νοσηλευτών παγκοσμίως [1], ενώ το Διεθνές Συμβούλιο Νοσηλευτών (ICN) προβλέπει ότι θα χρειαστούν πάνω από 10 εκατομμύρια νοσηλευτές μέχρι το 2030 για να αντιμετωπιστεί η υπάρχουσα έλλειψη και να ληφθούν υπόψη οι αναμενόμενες συνταξιοδοτήσεις [2].

Καθώς οι δημογραφικές μεταβολές και τα εξελισσόμενα μοντέλα υγειονομικής περίθαλψης δίνουν όλο και μεγαλύτερη έμφαση στη φροντίδα με επίκεντρο τον ασθενή, ο ρόλος των κατ' οίκον νοσηλευτριών έχει γίνει τόσο πιο σημαντικός, όσο και πιο σύνθετος. Η κατ' οίκον φροντίδα προσφέρει μια κρίσιμη εναλλακτική λύση στη θεραπεία σε ιδρύματα, επιτρέποντας στους ασθενείς να λαμβάνουν υποστήριξη σε οικεία περιβάλλοντα, ενώ παράλληλα μειώνει την επιβάρυνση των κεντρικών εγκαταστάσεων υγειονομικής περίθαλψης. Ωστόσο, ο συντονισμός και η διαχείριση του προσωπικού φροντίδας σε αυτά τα αποκεντρωμένα περιβάλλοντα παρουσιάζουν σημαντικές υλικοτεχνικές και οργανωτικές προκλήσεις.

Σε αντίθεση με τα παραδοσιακά κλινικά περιβάλλοντα, η κατ' οίκον φροντίδα απαιτεί ευέλικτο προγραμματισμό και ανταπόκριση στις ατομικές ανάγκες και προτιμήσεις. Η αντιμετώπιση αυτών των προκλήσεων απαιτεί προηγμένα συστήματα διαχείρισης ικανά να βελτιστοποιήσουν την κατανομή των πόρων χωρίς να διακυβεύεται η ποιότητα της φροντίδας ή η επαγγελματική ικανοποίηση των νοσηλευτριών. Οι υπολογιστικές προσεγγίσεις προσφέρουν πολλά υποσχόμενες λύσεις σε αυτά τα πολυπαραγοντικά προβλήματα.

Η παρούσα εργασία διερευνά τον σχεδιασμό και την υλοποίηση μιας πλατφόρμας διαχείρισης νοσηλευτικού προσωπικού προσαρμοσμένης ειδικά στο πλαίσιο της κατ' οίκον φροντίδας. Με την εφαρμογή αλγορίθμων βελτιστοποίησης στον προγραμματισμό και τον συντονισμό των υπηρεσιών φροντίδας, το προτεινόμενο σύστημα επιδιώκει να ενισχύσει τη λειτουργική αποτελεσματικότητα, να βελτιώσει τις συνθήκες εργασίας για το προσωπικό και τελικά να συμβάλει σε πιο βιώσιμα μοντέλα παροχής υγειονομικής περίθαλψης.

1.3 Πρόβλημα

Μια κρίσιμη παθογένεια είναι η ανεπάρκεια νοσοκομειακών κλινών, συμπεριλαμβανομένων των κλινών εντατικής θεραπείας, η οποία συχνά οδηγεί σε πληρότητες που δυσχεραίνουν την εξυπηρέτηση των ασθενών. Ειδικότερα στην Ελλάδα, όπως καταγγέλει η Π.Ο.Ε.ΔΗ.Ν. (Πανελλήνια Ομοσπονδία Εργαζομένων Δημοσίων Νοσοκομείων), «Στη χώρα μας λειτουργούν 3,5 κλίνες ανά 1.000 κατοίκους τη στιγμή που ο μέσος όρος στις χώρες της Ευρωπαϊκής Ένωσης είναι 5,3 κλίνες ανά 1.000 κατοίκους» [3]. Αυτό αναγκάζει πολλές φορές τους ασθενείς να αναζητήσουν «αποκλειστικές» νοσηλεύτριες, ώστε να αποφύγουν την αναμονή για κλίνη και να νοσηλευτούν μερικώς ή εξ ολοκλήρου κατ' οίκον.

Σε πρόσφατη έρευνα που διεξήχθη στην Αυστραλία, οι περισσότεροι από τους μισούς ερωτηθέντες δήλωσαν ότι είχαν λάβει προσωπική άδεια κατά τους προηγούμενους έξι μήνες λόγω κόπωσης ως

αποτέλεσμα του προγράμματός τους. Παρόμοιο ποσοστό δήλωσαν επίσης ότι θα ήθελαν να έχουν τη δυνατότητα να επιλέγουν πότε θα εργάζονται σε νυχτερινή βάρδια, καθώς αυτό θα βελτιώνει την ικανοποίησή τους από το πρόγραμμα [4]. Σε παρόμοια έρευνα το 2020 είχε διαπιστωθεί ότι η χρήση λογισμικού συμμετοχικού χρονοπρογραμματισμού σε νοσοκομειακούς θαλάμους είχε ως αποτέλεσμα τη μείωση τόσο των μακροχρόνιων (>3), όσο και των σύντομων (1-3) ημερών απουσίας, λόγω ασθένειας των νοσηλευτών [5].

Η παγκόσμια έλλειψη νοσηλευτών αποτελεί κρίσιμη απειλή για την πρόσβαση και τη συνέχεια της υγειονομικής περίθαλψης. Η πίεση αυτή δεν περιορίζεται στους θαλάμους των νοσοκομείων. Η κατ' οίκον φροντίδα, στην οποία βασίζεται όλο και περισσότερο η διαχείριση χρόνιων ασθενειών, η φροντίδα των ηλικιωμένων και η αποκατάσταση μετά την ανάκαμψη, επηρεάζεται εξίσου, αν όχι περισσότερο. Με περιορισμένους πόρους και αυξανόμενη βάση ασθενών, η πρόκληση δεν έγκειται μόνο στην αύξηση του εργατικού δυναμικού, αλλά και στην αποτελεσματικότερη και δικαιότερη χρήση του υφιστάμενου νοσηλευτικού προσωπικού.

Η κατ' οίκον φροντίδα παρουσιάζει μοναδικές υλικοτεχνικές πολυπλοκότητες. Σε αντίθεση με τα νοσοκομειακά περιβάλλοντα όπου το προσωπικό εργάζεται σε μια ενιαία εγκατάσταση, οι νοσηλεύτές/τριες που προσφέρουν υπηρεσίες στο σπίτι, πρέπει να αντιμετωπίσουν μεταβλητούς χρόνους ταξιδιού, μεταβαλλόμενες ανάγκες των ασθενών και ιδιαίτερα εξατομικευμένα προγράμματα φροντίδας. Τα χειροκίνητα ή ημι-αυτοματοποιημένα συστήματα προγραμματισμού συχνά δεν ανταποκρίνονται σε αυτή την πολυπλοκότητα, με αποτέλεσμα αναποτελεσματική κάλυψη, διπλή προσπάθεια ή κενά στη φροντίδα - όλα αυτά επιδεινώνουν περαιτέρω την κόπωση του νοσηλευτικού προσωπικού και προκαλούν τη δυσαρέσκεια των ασθενών.

1.4 Σκοποί

Αυτή η πλατφόρμα δημιουργείται με σκοπό την παροχή πρακτικής, βάσει δεδομένων, υποστήριξης για τον συντονισμό των υπηρεσιών κατ' οίκον φροντίδας. Αντί να αντιμετωπίζει τις ελλείψεις εργατικού δυναμικού ως καθαρά αριθμητικές ελλείψεις, η πλατφόρμα υιοθετεί μια προσέγγιση σε επίπεδο συστήματος, με στόχο την αποτελεσματικότερη και πιο βιώσιμη χρήση των υφιστάμενων ανθρώπινων πόρων. Οι σκοποί της περιγράφονται κατωτέρω:

- **Μετριασμός του αντίκτυπου των ελλείψεων προσωπικού:** Η πλατφόρμα, βελτιστοποιώντας την κατανομή του διαθέσιμου νοσηλευτικού προσωπικού μέσω προγραμματισμού βάσει αλγορίθμων, επιδιώκει να μειώσει τον χρόνο αδράνειας, τα επικαλυπτόμενα ραντεβού και την υποαπασχόληση. Με τον τρόπο αυτό διασφαλίζεται ότι ένα περιορισμένο εργατικό δυναμικό μπορεί να παρέχει φροντίδα σε όσο το δυνατόν περισσότερους ασθενείς χωρίς να υποβαθμίζεται η ποιότητα.
- **Υποστήριξη της ευημερίας και της διατήρησης του προσωπικού:** Η πλατφόρμα ενσωματώνει τις προσαρμόσιμες προτιμήσεις βάρδιας για τη μείωση της κόπωσης και της εξουθένωσης - σημαντικοί παράγοντες που ευθύνονται για την απουσία νοσοκόμων και τη φθορά του ηθικού τους. Βελτιώνοντας τις καθημερινές συνθήκες εργασίας, το σύστημα συμβάλλει στη μακροπρόθεσμη σταθερότητά του εργατικού δυναμικού.

1.5 Στόχοι

Με βάση το πρόβλημα που περιγράφηκε και την ανάγκη για βελτιωμένη διαχείριση του νοσηλευτικού προσωπικού στην κατ' οίκον φροντίδα, η παρούσα πτυχιακή εργασία θέτει τους ακόλουθους στόχους:

1. **Σχεδιασμός και ανάπτυξη βάσης δεδομένων:** Να σχεδιαστεί και να υλοποιηθεί μια κατάλληλη δομή βάσης δεδομένων ικανή να αποθηκεύει και να διαχειρίζεται αποτελεσματικά πληροφορίες σχετικά με:
 - Το νοσηλευτικό προσωπικό (π.χ. προσόντα, διαθεσιμότητα, προτιμήσεις βαρδιών, διεύθυνση).
 - Τους ασθενείς (π.χ. απαιτούμενες ώρες/ημέρες φροντίδας, διεύθυνση).
 - Τις βάρδιες (π.χ. ημερομηνία, διάρκεια, ανάγκες φροντίδας ασθενή, ανάθεση σε νοσηλεύτη/τρια).
2. **Υλοποίηση αλγορίθμων βελτιστοποίησης προγραμματισμού:** Να αναπτυχθούν και να ενσωματωθούν στην πλατφόρμα αλγόριθμοι βελτιστοποίησης οι οποίοι θα αυτοματοποιούν τη διαδικασία δημιουργίας προγραμμάτων εργασίας (βαρδιών) για το νοσηλευτικό προσωπικό, λαμβάνοντας υπόψη πολλαπλούς παράγοντες όπως:
 - Τις ανάγκες κάλυψης των ασθενών.
 - Τη διαθεσιμότητα και τα προσόντα του προσωπικού.
 - Τη γεωγραφική εγγύτητα για την ελαχιστοποίηση του χρόνου μετακίνησης.
 - Τις δηλωμένες προτιμήσεις του προσωπικού (π.χ. επιλογή νυχτερινής βάρδιας ή βάρδιας το σαββατοκύριακο).
3. **Ανάπτυξη περιβάλλοντος διεπαφής χρήστη (User Interface):** Να δημιουργηθεί ένα λειτουργικό και φιλικό προς τον χρήστη περιβάλλον διεπαφής που θα επιτρέπει:
 - Στους διαχειριστές να εισάγουν/επεξεργάζονται δεδομένα βαρδιών και ασθενών.
 - Να επιλέγουν την κατάλληλη νοσηλεύτρια για μια βάρδια, σύμφωνα με τις προτάσεις του αλγορίθμου βελτιστοποίησης.
 - Να εκκινούν τη διαδικασία βελτιστοποίησης ενός συνόλου βαρδιών.
 - Να μεταβάλλουν το κατά πόσο συγκεκριμένοι παράγοντες επηρεάζουν την τελική βαθμολογία του αλγορίθμου.
 - Να διαλέγουν διαφορετικούς αλγορίθμους βελτιστοποίησης, ώστε να βλέπουν τη διαφορά ανάμεσα σε έναν απλοϊκό αλγόριθμο και σε έναν πιο πολύπλοκο και εξελιγμένο.

1.6 Δομή

Στο **1ο κεφάλαιο** έγινε μια εισαγωγή στην πτυχιακή εργασία, όπου αναφέρθηκαν τα προβλήματα που αντιμετωπίζουν οι νοσηλεύτριες ανά τον κόσμο αλλά και ειδικότερα στην Ελλάδα, καθώς και τον τρόπο με τον οποίο οι συγγραφείς της παρούσας εργασίας προσπαθούν να βάλουν το λιθαράκι τους στην επίλυσή τους.

Στο **2ο κεφάλαιο** παρουσιάζεται και αναλύεται η επιλογή των κατάλληλων τεχνολογικών υποδομών, οι οποίες είναι καθοριστικές για την επιτυχία ενός έργου, καθώς επηρεάζουν την ανάπτυξη, την απόδοση και την ποιότητα της εφαρμογής. Ένα σωστά επιλεγμένο τεχνολογικό υπόβαθρο διασφαλίζει την αποτελεσματικότητα, την επεκτασιμότητα και την αξιοπιστία του τελικού λογισμικού, συμβάλλοντας επίσης και στην ευκολότερη επίτευξη των στόχων της πτυχιακής.

Στο **3ο κεφάλαιο**, η ανάλυση εμβαθύνει στην τεχνική υλοποίηση της εφαρμογής, παρουσιάζοντας τους δύο θεμελιώδεις πυλώνες της: την αρχιτεκτονική των δεδομένων στο backend και το περιβάλλον χρήστη στο frontend. Περιγράφεται λεπτομερώς η δομή της βάσης δεδομένων, συμπεριλαμβανομένων των πινάκων που σχεδιάστηκαν ειδικά για τις ανάγκες της εφαρμογής και εκείνων που παρήχθησαν αυτόματα από τις χρησιμοποιούμενες τεχνολογίες, καθώς και οι βασικές οθόνες, οι ροές εργασίας και οι λειτουργίες του frontend, με ιδιαίτερη έμφαση στον αλγόριθμο ανάθεσης προσωπικού που αποτελεί κεντρικό στοιχείο της πτυχιακής εργασίας, με σκοπό την επίτευξη της βέλτιστης διαχείρισης νοσηλευτικού προσωπικού.

Στο **4ο κεφάλαιο** αναλύεται ένα ευρύ φάσμα αλγοριθμικών προσεγγίσεων για τον χρονο-προγραμματισμό του νοσηλευτικού προσωπικού. Παρουσιάζονται λεπτομερώς ο Άπληστος αλγόριθμος, η μέθοδος του Μοντέλου Σταθμισμένου Αθροίσματος (WSM), η Προσομοιωμένη Ανόπτηση (Simulated Annealing), η Αναζήτηση Tabu (Tabu Search) και η τεχνική TOPSIS. Για κάθε προσέγγιση, εξετάζονται το θεωρητικό πλαίσιο, η μαθηματική διατύπωση, η εφαρμογή στο πρόβλημα, οι εκτιμήσεις απόδοσης και οι λεπτομέρειες υλοποίησης στον κώδικα. Η συγκριτική αυτή παρουσίαση αναδεικνύει τις δυνατότητες και τους περιορισμούς κάθε μεθόδου, βοηθώντας στην επιλογή της καταλληλότερης τεχνικής ανάλογα με τις συγκεκριμένες ανάγκες και τους περιορισμούς του προβλήματος.

Στο **5ο κεφάλαιο** εξάγονται τα συμπεράσματα της εργασίας και διατυπώνονται προτάσεις βελτίωσης. Αξιολογούνται συγκριτικά οι μέθοδοι ανάθεσης και η επίδραση τεχνικών βελτιστοποίησης στην απόδοσή τους. Αναλύεται επίσης η σχέση προτιμήσεων προσωπικού και λειτουργικών αναγκών. Τέλος, προτείνονται μελλοντικές επεκτάσεις και βελτιώσεις της πλατφόρμας.

1.7 Επίλογος

Στο κεφάλαιο αυτό παρατέθηκε το πρόβλημα που αντιμετωπίζουν τα συστήματα υγείας και το νοσηλευτικό προσωπικό παγκοσμίως, οι σκοποί επίλυσης του προβλήματος, οι στόχοι της πτυχιακής εργασίας, καθώς και η δομή την οποία θα ακολουθήσουν τα επόμενα κεφάλαια.

Κεφάλαιο 2ο: Τεχνολογίες

2.1 Εισαγωγή

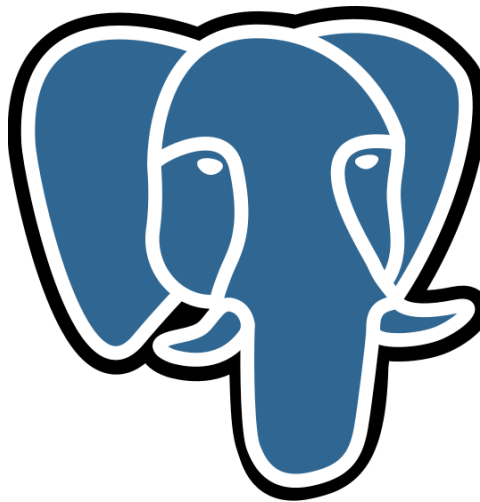
Η τεχνολογική υποδομή μιας εφαρμογής λογισμικού είναι καθοριστική, καθώς επηρεάζει την διαδικασία και το χρόνο ανάπτυξης της, την απόδοσή της, καθώς και την εμπειρία του τελικού χρήστη. Το παρόν κεφάλαιο εστιάζει στο τεχνολογικό υπόβαθρο της συγκεκριμένης εφαρμογής, αναλύοντας τα εργαλεία και τις βιβλιοθήκες που επιλέχθηκαν. Κάθε στοιχείο ενσωματώθηκε στρατηγικά για να ανταποκριθεί σε διακριτές απαιτήσεις και να αντιμετωπίσει συγκεκριμένες προκλήσεις.

Κεντρικός σκοπός ήταν η δημιουργία μιας σύγχρονης, αποδοτικής και εύχρηστης πλατφόρμας, σχεδιασμένης για την αποτελεσματική ανάκτηση, επεξεργασία και παρουσίαση δεδομένων. Εξετάζεται η συμβολή κάθε τεχνολογίας στην επίτευξη αυτών των στόχων, αναδεικνύοντας τον τρόπο με τον οποίο τα χαρακτηριστικά τους συνέβαλαν στη διαμόρφωση μιας επεκτάσιμης, συντηρήσιμης και αξιόπιστης πλατφόρμας λογισμικού.

2.2 Βάση δεδομένων

2.2.1 PostgreSQL

Η PostgreSQL είναι μια ισχυρή, ανοικτού κώδικα αντικειμενο-σχεσιακή βάση δεδομένων με πάνω από 35 χρόνια ενεργής ανάπτυξης που της έχει χαρίσει ισχυρή φήμη για την αξιοπιστία, την ανθεκτικότητα των χαρακτηριστικών και τις επιδόσεις της [6]. Η αξιοποίησή της έγινε λόγω προηγούμενης γνώσης των συγγραφέων, ευρείας χρήσης της από το πανεπιστημιακό ίδρυμα, καθώς και λόγω της υποστήριξής της από την πλατφόρμα στην οποία φιλοξενείται η βάση δεδομένων, όπως αναφέρεται στην παράγραφο 2.2.3. Στην πτυχιακή εργασία χρησιμοποιήθηκε η πιο πρόσφατη αυτή τη στιγμή έκδοσή της, η PostgreSQL 17.



Σχήμα 2.1: Λογότυπο PostgreSQL

2.2.2 PostGIS

Το PostGIS επεκτείνει τις δυνατότητες της σχεσιακής βάσης δεδομένων PostgreSQL προσθέτοντας υποστήριξη για αποθήκευση, ευρετηρίαση και αναζήτηση γεωχωρικών δεδομένων. Χρησιμοποιήθηκε στην τρέχουσα πτυχιακή εργασία ώστε να είναι δυνατή η αποθήκευση των τοποθεσιών τόσο των νοσηλευτριών, όσο και των ασθενών, καθώς και ο υπολογισμός των μεταξύ τους αποστάσεων [7].



Σχήμα 2.2: Λογότυπο PostGIS

2.2.3 Neon

Η Neon είναι μια πλατφόρμα βασισμένη στην PostgreSQL που βοηθά στη δημιουργία πιο γρήγορων, αξιόπιστων, και επεκτάσιμων εφαρμογών. Χάρη στη *serverless* αρχιτεκτονική της, δεν υπάρχει χρέωση όσο δε γίνεται ενεργή χρήση της βάσης από τους χρήστες, με αποτέλεσμα να μειώνονται τα λειτουργικά κόστη της υπηρεσίας (για τις ανάγκες τις πτυχιακής εργασίας, το δωρεάν πλάνο υπήρξε υπερ-αρκετό). Στις πρωτοποριακές της δυνατότητες περιλαμβάνεται επίσης το “branching”, με το οποίο, χρησιμοποιώντας την τεχνική της αντιγραφής-κατά-την-εγγραφή (copy-on-write), μπορούν να δημιουργηθούν άμεσα αντίγραφα μιας βάσης ώστε η ανάπτυξη και η δοκιμή ενός λογισμικού να κατασταθεί πολύ πιο εύκολη και κοντινή στις πραγματικές συνθήκες της εφαρμογής, για παράδειγμα με την δημιουργία ενός νέου “branch” όταν ο προγραμματιστής θέλει να δοκιμάσει μια καινούρια δυνατότητα χρησιμοποιώντας αληθινά δεδομένα χρηστών από τη βάση παραγωγής (production database) [8].



Σχήμα 2.3: Λογότυπο Neon

2.3 Γλώσσα προγραμματισμού

2.3.1 TypeScript

Η TypeScript, η οποία δημιουργήθηκε από την Microsoft ως ένα υπερέκδο της JavaScript, προσθέτει στατικούς τύπους (static types) στη γλώσσα, βοηθώντας στον εντοπισμό και τη διόρθωση σφαλμάτων νωρίτερα στη φάση ανάπτυξης μιας εφαρμογής, και βελτιώνοντας σημαντικά την αναγνωσιμότητα και τη συντηρησιμότητα του κώδικα, ειδικά σε μεγάλα και πολύπλοκα έργα όπου συνεργάζονται πολλοί προγραμματιστές. Επιπλέον, ενισχύει την παραγωγικότητα των προγραμματιστών, παρέχοντας καλύτερη υποστήριξη από τα ολοκληρωμένα περιβάλλοντα ανάπτυξης (IDEs), όπως η ακριβέστερη αυτόματη συμπλήρωση κώδικα (auto-completion) και η ευκολότερη αναδιάρθρωση (refactoring). Παρόλο που προσθέτει αυτά τα χαρακτηριστικά, η TypeScript μεταγλωττίζεται (compiles) σε απλή

JavaScript, διασφαλίζοντας έτσι πλήρη συμβατότητα με όλα τα περιβάλλοντα στα οποία εκτελείται η τελευταία [9].

Στην πτυχιακή εργασία χρησιμοποιήθηκε η πιο πρόσφατη αυτή τη στιγμή έκδοσή της, η TypeScript 5.7, για την ανάπτυξη τόσο του backend, όσο και του frontend. Με αυτό τον τρόπο, δίνεται η δυνατότητα στον προγραμματιστή να επαναχρησιμοποιεί τον κώδικα με τη χρήση interfaces, τύπων δεδομένων (types), και βοηθητικών συναρτήσεων (utility functions). Απαιτείται πλέον η γνώση μόνο μίας γλώσσα προγραμματισμού αντί για δύο, τα δεδομένα δεν χρειάζονται μετατροπή ανάμεσα σε γλώσσες, καθώς έχουν ήδη τη σωστή δομή και μπορούν να χρησιμοποιηθούν άμεσα, και τα εργαλεία (tooling), η διαδικασία build, και η διαχείριση εξαρτήσεων (dependency management) για ολόκληρη την πτυχιακή εργασία απλοποιούνται.



Σχήμα 2.4: Λογότυπο TypeScript

2.4 Backend

2.4.1 Node.js

Το Node.js είναι ένα περιβάλλον που επιτρέπει την εκτέλεση κώδικα JavaScript στην πλευρά του διακομιστή (server), αντί για την πλευρά του φυλλομετρητή (browser). Είναι ανοιχτού κώδικα και βασίζεται στη μηχανή V8 της Google (την ίδια που χρησιμοποιεί ο Chrome) για γρήγορη εκτέλεση κώδικα. Η κύρια αξία του βρίσκεται στο ασύγχρονο (asynchronous) και non-blocking I/O μοντέλο του, το οποίο το καθιστά εξαιρετικά αποδοτικό για εφαρμογές που διαχειρίζονται πολλές ταυτόχρονες συνδέσεις, όπως διακομιστές, APIs, εφαρμογές συνομιλίας πραγματικού χρόνου και μικροπηρεσίες (microservices), χωρίς να χρειάζεται ο διακομιστής να σταματάει την εκτέλεση του κώδικα περιμένοντας μια λειτουργία να ολοκληρωθεί. Έρχεται μαζί με το npm (Node Package Manager), το οποίο δίνει πρόσβαση σε μια τεράστια συλλογή από έτοιμες βιβλιοθήκες και εργαλεία (πακέτα), επιταχύνοντας σημαντικά την ανάπτυξη μιας εφαρμογής. Στην πτυχιακή εργασία χρησιμοποιήθηκε η έκδοση Node 20 [10].



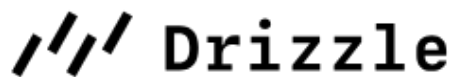
Σχήμα 2.5: Λογότυπο Node.js

2.4.2 Drizzle ORM

Το Drizzle ORM είναι ένα σύγχρονο, ελαφρύ και υψηλής απόδοσης TypeScript ORM (Object-Relational Mapper), σχεδιασμένο με έμφαση στην ευκολία για τον προγραμματιστή και την ταχύτητα. Ξεχωρίζει για την απουσία εξωτερικών εξαρτήσεων (external dependencies), γεγονός που το καθιστά ιδανικό για serverless περιβάλλοντα.

Ένα από τα κύρια χαρακτηριστικά του Drizzle είναι η σύνταξή του, η οποία είναι σχεδιασμένη να μοιάζει πολύ με την SQL. Αυτή η προσέγγιση, με την οποία οι συναρτήσεις μοιάζουν σαν SQL (SQL-like) σημαίνει ότι οι προγραμματιστές που είναι ήδη εξοικειωμένοι με αυτήν, μπορούν να υιοθετήσουν το Drizzle με λίγη προσπάθεια και σε ελάχιστο χρόνο. Αντί να κρύβει την SQL πίσω από πολύπλοκες αφαιρέσεις (abstractions), το Drizzle την αγκαλιάζει, επιτρέποντας στους προγραμματιστές να γράφουν queries χρησιμοποιώντας μεθόδους που αντιστοιχούν άμεσα σε εντολές SQL (π.χ., `db.select(...).from(table).where(...)`). Αυτό παρέχει τον πλήρη έλεγχο και τη δύναμη της SQL, ενώ ταυτόχρονα προσφέρει τα οφέλη της ασφάλειας τύπων (type-safety) στον προγραμματιστή, χάρη στην ενσωμάτωση της με την TypeScript. Η διασφάλιση της ασφάλειας τύπων αρχίζει από τον ορισμό του σχήματος (schema), φτάνοντας μέχρι και την εκτέλεση των ερωτημάτων (queries), μειώνοντας την πιθανότητα για σφάλματα κατά την εκτέλεση του προγράμματος.

Το Drizzle χρησιμοποιείται στην εργασία μας μαζί με το Drizzle Kit, το οποίο μας επιτρέπει να ορίζουμε το σχήμα της βάσης μας σε TypeScript και στη συνέχεια να παράγουμε αυτόματα αρχεία SQL migration, με βάση τις αλλαγές που κάναμε στο σχήμα. Μπορεί επίσης να εκτελεί τα αρχεία migration που παράγονται στη συνδεδεμένη βάση, καθώς και να γνωρίζει ποια migration έχουν ήδη εκτελεστεί, ώστε να αποτρέψει την διπλή εκτέλεσή τους και πιθανή απώλεια δεδομένων [11].



Σχήμα 2.6: Λογότυπο Drizzle

2.4.3 tRPC

Το tRPC είναι ένα framework για την TypeScript που επιτρέπει την ασφαλή ως προς τους τύπους (type-safe) κλήση συναρτήσεων του backend απευθείας από το frontend, σαν να ήταν τοπικές συναρτήσεις. Αυτό εξαλείφει την ανάγκη για χειροκίνητη δημιουργία schemas ή παραγωγή κώδικα, προσφέροντας end-to-end type safety και μια εξαιρετική εμπειρία προγραμματιστή με αυτόματη συμπλήρωση (autocomplete) και έλεγχο τύπων (type-checking) κατά την ανάπτυξη της εφαρμογής [12].



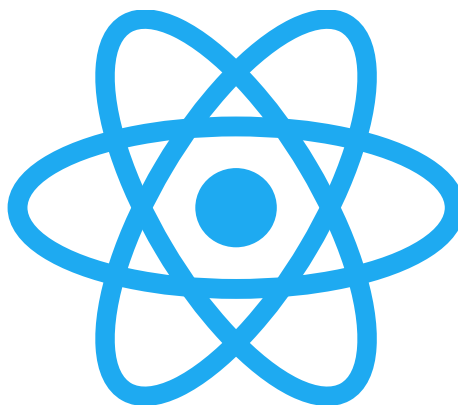
Σχήμα 2.7: Λογότυπο tRPC

2.5 Frontend

2.5.1 React.js

Η React.js, γνωστή και ως “React”, είναι η πιο δημοφιλής ανοιχτού κώδικα βιβλιοθήκη-πλαίσιο για τη δημιουργία διεπαφών χρήστη (UI framework). Γραμμένη σε JavaScript, αναπτύχθηκε από τη Meta (πρώην Facebook) και χρησιμοποιείται κυρίως για εφαρμογές μιας σελίδας (single-page applications, SPAs). Επιτρέπει στους προγραμματιστές να χτίζουν πολύπλοκες και διαδραστικές διεπαφές χρήστη με ευέλικτο και αποτελεσματικό τρόπο, δημιουργώντας επαναχρησιμοποιήσιμα δομικά στοιχεία που ονομάζονται components. Η React χρησιμοποιεί ένα Virtual DOM, μια εικονική αναπαράσταση του πραγματικού DOM του φυλλομετρητή για να βελτιστοποιεί τις ενημερώσεις στην οθόνη, αλλάζοντας μόνο τα τμήματα της σελίδας που έχουν τροποποιηθεί, γεγονός που την καθιστά εξαιρετικά γρήγορη και αποδοτική.

Η React ακολουθεί μια δηλωτική (declarative) προσέγγιση, όπου περιγράφεις τι θέλεις να εμφανιστεί στο UI για μια δεδομένη κατάσταση (state), και η React αναλαμβάνει να ενημερώσει το DOM αποτελεσματικά για να φτάσει σε αυτή την κατάσταση, αντί να γράφεις βήμα-βήμα τις εντολές για την τροποποίηση του DOM (imperative προσέγγιση). Ακόμη, χρησιμοποιεί την τεχνική της μονόδρομης ροής δεδομένων (one-way data binding), όπου τα δεδομένα ρέουν από τα γονικά (parent) προς τα θυγατρικά (child) components μέσω arguments που ονομάζονται “props”, κάνοντας τη ροή των δεδομένων πιο προβλέψιμη και διευκολύνοντας την αποσφαλμάτωση [13].



Σχήμα 2.8: Λογότυπο React

2.5.2 Next.js

Το Next.js είναι ένα από τα πιο δημοφιλή (αν όχι το πιο δημοφιλέσ αυτή τη στιγμή) ανοιχτού κώδικα JavaScript framework, το οποίο χρησιμοποιείται ευρέως για την κατασκευή σύγχρονων διαδικτυακών εφαρμογών, επεκτείνοντας τις δυνατότητες της React για να διευκολύνει τη δημιουργία εφαρμογών, ιδιαίτερα αυτών που αναπτύσσονται ως full-stack λύσεις. Διαθέτει υποστήριξη για TypeScript και ενσωματώνει αυτόματες βελτιστοποιήσεις για εικόνες, γραμματοσειρές και scripts, συμβάλλοντας σε μια καλύτερη εμπειρία για τον τελικό χρήστη. Η πλοήγηση στην εφαρμογή απλοποιείται χάρη στο σύστημα routing που βασίζεται στη δομή των αρχείων και των φακέλων του project.

Μια σημαντική εξέλιξη που υιοθετεί και ενσωματώνει το Next.js, ειδικά με το App Router, είναι τα React Server Components (RSC). Αυτά τα components εκτελούνται αποκλειστικά στον διακομιστή και δε στέλνουν JavaScript στον χρήστη. Αυτό έχει πολλαπλά οφέλη: μειώνει δραστικά το μέγεθος του αρχείου JavaScript που κατεβάζει ο χρήστης, επιτρέποντας ταχύτερους αρχικούς χρόνους φόρτωσης της σελίδας και καλύτερη απόδοση, ειδικά σε πιο αργές συνδέσεις ή λιγότερο ισχυρές συσκευές. Επίσης, τα Server Components μπορούν να έχουν απευθείας πρόσβαση σε πόρους του διακομιστή, όπως βάσεις δεδομένων ή το σύστημα αρχείων, χωρίς την ανάγκη δημιουργίας ξεχωριστών API endpoints, απλοποιώντας την αρχιτεκτονική για την ανάκτηση δεδομένων. Συνυπάρχουν με τα παραδοσιακά Client Components, τα οποία εκτελούνται στον φυλλομετρητή και διαχειρίζονται την διαδραστικότητα και την κατάσταση (state) της διεπαφής χρήστη [14].



Σχήμα 2.9: Λογότυπο Next.js

2.5.3 Tailwind CSS

Η Tailwind CSS είναι ένα σύγχρονο CSS framework που ακολουθεί την προσέγγιση “utility-first”. Αντί να σου δίνει έτοιμα, στυλιζαρισμένα components (όπως κουμπιά ή κάρτες), σου παρέχει ένα εκτενές σύνολο από μικρές, μονοσήμαντες κλάσεις CSS, τα λεγόμενα “utility classes”. Κάθε τέτοια κλάση αντιστοιχεί συνήθως σε έναν μόνο κανόνα CSS, όπως π.χ. “pt-4” για “padding-top” ή “text-red-500” για να γίνει το κείμενο κόκκινο. Αυτές οι κλάσεις εφαρμόζονται απευθείας στην HTML, κάτι το οποίο επιτρέπει την ταχύτερη ανάπτυξη κώδικα, καθώς δεν υπάρχει πια το επίπεδο αφαίρεσης (abstraction layer) που υπήρχε με το συνδυασμό HTML και CSS μέσω των παραδοσιακών κλάσεων CSS. Στην τελική ιστοσελίδα, ο κώδικας περνάει από ένα στάδιο μεταγλώττισης στο οποίο οι κανόνες που χρησιμοποιήθηκαν, μαζί με τις βασικές κλάσεις της Tailwind, καθώς και τυχόν CSS που έχει γραφτεί, μετατρέπονται σε ένα και μοναδικό αρχείο CSS. Χάρη στο μικρό μέγεθός του, αυτό το αρχείο μπορεί να κατεβεί και να αναλυθεί (parsing) ταχύτερα από το φυλλομετρητή του χρήστη [15].



Σχήμα 2.10: Λογότυπο Tailwind CSS

2.6 Επίλογος

Για την υλοποίηση της παρούσας εργασίας, επιλέχθηκε μια σύγχρονη τεχνολογική στοίβα με κριτήρια την επεκτασιμότητα, τη συντηρησιμότητα και την παραγωγικότητα. Στον πυρήνα της αρχιτεκτονικής βρίσκεται η βάση δεδομένων PostgreSQL, αναγνωρισμένη για τη σταθερότητά της, η οποία ενισχύεται με την επέκταση PostGIS για προηγμένες γεωχωρικές δυνατότητες. Η υποδομή φιλοξενείται στην serverless πλατφόρμα της Neon, απλοποιώντας σημαντικά τη διαχείριση της υποδομής και την αυτόματη κλιμάκωση πόρων (auto-scaling). Η καθολική υιοθέτηση της TypeScript, τόσο στο backend (Node.js) όσο και στο frontend, προάγει την ευρωστία του κώδικα και μειώνει τα σφάλματα κατά την εκτέλεση χάρη στη στατική τυποποίηση (static typing), βελτιστοποιώντας τη διαδικασία ανάπτυξης. Το Drizzle ORM παρέχει μια αποτελεσματική διεπαφή για την αλληλεπίδραση με τη βάση δεδομένων, ενώ το tRPC εγγυάται τη δημιουργία πλήρως τυποασφαλών (type-safe) APIs μεταξύ χρήστη και διακομιστή. Στο frontend, ο συνδυασμός React.js και Next.js προσφέρει ένα ευέλικτο περιβάλλον χρήστη, και το Tailwind CSS διευκολύνει την ταχεία ανάπτυξη ενός συνεκτικού οπτικού στυλ.

Κεφάλαιο 3ο: Αλγόριθμοι

3.1 Εισαγωγή

Η κατανομή νοσηλευτικού προσωπικού αποτελεί ένα πολυπαραμετρικό πρόβλημα, το οποίο εμπλέκει επιχειρησιακούς περιορισμούς, ποιοτικά κριτήρια και ανθρώπινες προτιμήσεις. Η ανάγκη για εύρυθμο χρονοπρογραμματισμό σε πραγματικό χρόνο, με βάση δυναμικά δεδομένα, καθιστά αναγκαία την υιοθέτηση εργαλείων που υπερβαίνουν τις δυνατότητες των παραδοσιακών μεθόδων. Η διαχείριση αυτής της πολυπλοκότητας προϋποθέτει όχι μόνο λειτουργική αυτοματοποίηση, αλλά και την ικανότητα ενσωμάτωσης ευέλικτων λογικών αποφάσεων.

Η εργασία αυτή επιδιώκει να προσεγγίσει το πρόβλημα χρησιμοποιώντας ένα ενιαίο πλαίσιο από διαφορετικά πεδία γνώσης, ενοποιώντας αρχές από τον χώρο της πληροφορικής, της επιχειρησιακής έρευνας και της μοντελοποίησης συστημάτων. Μέσα από την ανάλυση και επεξεργασία δεδομένων από ρεαλιστικά περιβάλλοντα, επιδιώκεται η ανάδειξη τρόπων με τους οποίους η τεχνολογία μπορεί να ενισχύσει τη διαφάνεια, την αποδοτικότητα και τη δικαιοσύνη στη διαδικασία ανάθεσης βαρδιών.

3.2 Άπληστος

3.2.1 Εισαγωγή

Ο άπληστος αλγόριθμος είναι ένα ευρέως γνωστό αλγοριθμικό πρότυπο που χτίζει μια λύση κομμάτι-κομμάτι, επιλέγοντας πάντα το αμέσως επόμενο κομμάτι που προσφέρει το πιο άμεσο όφελος ή «άπληστη» επιλογή. Χρησιμοποιείται ευρέως σε προβλήματα βελτιστοποίησης όπου τοπικές βέλτιστες επιλογές αναμένεται ότι θα οδηγήσουν σε μια συνολικά βέλτιστη λύση [16]. Παραδείγματα κλασικών προβλημάτων που επιλύονται μέσω άπληστων μεθόδων περιλαμβάνουν την κωδικοποίηση Huffman [17], τους αλγορίθμους Kruskal και Prim για την κατασκευή ελαχίστων δέντρων εξάπλωσης [18], καθώς και προβλήματα επιλογής δραστηριοτήτων.

3.2.2 Ιστορικό πλαίσιο

Τα εννοιολογικά θεμέλια του άπληστου αλγορίθμου χρονολογούνται από τα μέσα του 20ου αιώνα, αποκτώντας μεγάλη αναγνωρισιμότητα μέσω της χρήσης του στη θεωρία γραφημάτων και τη βελτιστοποίηση δικτύων. Η επίσημη μελέτη των άπληστων μεθόδων επεκτάθηκε στις δεκαετίες του 1960 και 1970 με την εμφάνιση της επιστήμης των υπολογιστών σε θεωρητικό επίπεδο. Οι άπληστοι αλγόριθμοι χαρακτηρίζονται από την απλότητα και την ταχύτητα εκτέλεσής τους, αν και δεν εγγυώνται την εύρεση της βέλτιστης λύσης σε κάθε περίπτωση. Η χρήση τους συνήθως δικαιολογείται όταν το πρόβλημα δεν απαιτεί την εύρεση της βέλτιστης λύσης.

3.2.3 Πλαίσιο εφαρμογής

Στο πλαίσιο του χρονοπρογραμματισμού του νοσηλευτικού προσωπικού, ο άπληστος αλγόριθμος χρησιμοποιείται για την ανάθεση του νοσηλευτικού προσωπικού σε βάρδιες βάσει ενός ευρετικού συστήματος βαθμολόγησης. Ο στόχος είναι να ανατεθεί σε κάθε βάρδια ένας κατάλληλο μέλος του νοσηλευτικού προσωπικού, έτσι ώστε η αντιστοίχιση να ικανοποιεί τους λειτουργικούς περιορισμούς και να αντικατοπτρίζει την αντιστοίχιση των προτιμήσεων και τη συμβατότητα δεξιοτήτων.

Ο αλγόριθμος λειτουργεί διαδοχικά στο σύνολο των βαρδιών. Για κάθε βάρδια, αξιολογεί όλα τα κατάλληλα μέλη του νοσηλευτικού προσωπικού χρησιμοποιώντας μια συνάρτηση βαθμολόγησης πολλαπλών κριτηρίων, και αναθέτει το μέλος με την υψηλότερη βαθμολογία. Ο αλγόριθμος δεν

αναθεωρεί προηγούμενες αναθέσεις ούτε λαμβάνει υπόψη του μελλοντικές συγκρούσεις, καθιστώντας τον υπολογιστικά αποτελεσματικό αλλά δυνητικά μη βέλτιστο όσον αφορά τους συνολικούς στόχους.

3.2.4 Μαθηματική διατύπωση

Έστω $S = \{s_1, s_2, \dots, s_n\}$ το σύνολο των βαρδιών και $C = \{c_1, c_2, \dots, c_m\}$ το σύνολο των διαθέσιμων ατόμων του νοσηλευτικού προσωπικού. Ο στόχος είναι να προσδιοριστεί μια αντιστοίχιση:

$$S = \{s_1, s_2, \dots, s_n\}$$

$$C = \{c_1, c_2, \dots, c_m\}$$

Ωστε να οριστεί μια συνάρτηση ανάθεσης:

$$f: S \rightarrow C \quad (4.1)$$

ώστε για κάθε $s \in S$ να αντιστοιχεί ένα μέλος του νοσηλευτικού προσωπικού $c \in C$, ο οποίος μεγιστοποιεί μια συνάρτηση καταλληλότητας $S(c, s)$, υπό συγκεκριμένους περιορισμούς.

Ο άπληστος κανόνας επιλογής διατυπώνεται ως:

$$f(s) = \underset{c \in C_s}{\operatorname{argmax}} S(c, s) \quad (4.2)$$

όπου $C_s \subseteq C$ είναι το υποσύνολο του επιλέξιμου νοσηλευτικού προσωπικού για τη βάρδια s , με βάση σκληρούς περιορισμούς όπως:

- δεξιότητες,
- διαθεσιμότητα,
- απόσταση από το σημείο εργασίας.

Η συνάρτηση καταλληλότητας $S(c, s)$ ορίζεται ως γραμμικός σταθμισμένος συνδυασμός:

$$S(c, s) = w_D \cdot D(c, s) + w_n \cdot N(c, s) + w_w \cdot W(c, s) \cdot SkillMatch(c, s) \quad (4.3)$$

όπου:

- $D(c, s) \in [0, 1]$: δείκτης απόστασης (1 αν εντός ορίου, 0 αλλιώς),
- $N(c, s) \in [0, 1]$: δείκτης συμφωνίας με νυχτερινές βάρδιες,
- $W(c, s) \in [0, 1]$: δείκτης συμφωνίας με βάρδιες Σαββατοκύριακου,
- $SkillMatch(c, s) \in \mathcal{N}_0$: ο αριθμός των αντιστοιχισμένων δεξιοτήτων που απαιτούνται από τη βάρδια και διαθέτει το μέλος,

- $w_d, w_n, w_w, w_s \in \mathbb{R}^+$: συντελεστές βαρύτητας.

Ο αλγόριθμος Greedy αναθέτει επαναληπτικά τα μέλη του νοσηλευτικού προσωπικού στις βάρδιες υπολογίζοντας το $S(c, s)$ για κάθε επιλέξιμο c , επιλέγοντας το μέλος με την υψηλότερη βαθμολογία και χαρακτηρίζοντας τη βάρδια ως γεμάτη. Η διαδικασία αυτή συνεχίζεται μέχρι να ανατεθούν όλες οι βάρδιες ή να μην απομένουν επιλέξιμα μέλη του νοσηλευτικού προσωπικού.

3.2.5 Θεωρητικές εκτιμήσεις

Ενώ ο αλγόριθμος Greedy προσφέρει μια λύση χαμηλής πολυπλοκότητας -ιδιαίτερα επιθυμητή σε περιβάλλοντα υψηλής απόδοσης- η μυωπική φύση του σημαίνει ότι δεν εγγυάται μια συνολικά βέλτιστη λύση. Ειδικότερα, δεν επιλύει συγκρούσεις ανάθεσης ή βελτιστοποιεί τον φόρτο εργασίας του νοσηλευτικού προσωπικού σε πολλαπλές βάρδιες. Αυτοί οι περιορισμοί μπορούν να αντιμετωπιστούν μέσω εναλλακτικών προσεγγίσεων, όπως ο δυναμικός προγραμματισμός, ο ακέραιος γραμμικός προγραμματισμός ή οι τεχνικές συνδυαστικής βελτιστοποίησης (π.χ. διατυπώσεις με βάση το σακίδιο), οι οποίες μοντελοποιούν το πρόβλημα συνολικά, αλλά συχνά με κόστος την υπολογιστική απόδοση.

3.2.6 Υλοποίηση στον κώδικα

Η υλοποίηση του άπληστου αλγορίθμου στον παρόντα πηγαίο κώδικα εντοπίζεται κυρίως εντός της συνάρτησης `rankNursesGreedy`, η οποία είναι υπεύθυνη για την αξιολόγηση και κατάταξη των διαθέσιμων μελών του νοσηλευτικού προσωπικού (caregivers) ως προς την καταλληλότητά τους για μια συγκεκριμένη βάρδια (shift). Η συνάρτηση δέχεται ως ορίσματα τον πίνακα με τα μέλη του νοσηλευτικού προσωπικού, τη βάρδια προς κάλυψη και ένα αντικείμενο βαρών (weights) που ρυθμίζει τη σχετική σημασία των επιμέρους παραμέτρων.

Φιλτράρισμα Επιλέξιμων Υποψηφίων

Η διαδικασία ξεκινά με τον αποκλεισμό του νοσηλευτικού προσωπικού που δεν καλύπτει καμία από τις απαιτούμενες δεξιότητες της συγκεκριμένης βάρδιας. Το φιλτράρισμα αυτό υλοποιείται μέσω της μεθόδου `filter`, όπως φαίνεται στο παρακάτω απόσπασμα:

```

caregivers.filter((caregiver) =>
  shift.skills.some((skill) => caregiver.skills.includes(skill)),
);

```

Αντιστοίχως, το σύνολο C_s ορίζεται ως:

$$C_s = \{c \in C \mid \text{skills}(c) \cap \text{needs}(s) \neq \emptyset\}$$

Στη συνέχεια, για κάθε $C \in C_s$, υπολογίζεται:

$$S(c, s) = w_d \cdot D(c, s) + w_n \cdot N(c, s) \cdot w_w + W(c, s) + w_s \cdot M(c, s)$$

Η υλοποίηση περιλαμβάνει επιμέρους συνιστώσες που αποδίδονται ανάλογα με τη συμφωνία του μέλους του νοσηλευτικού προσωπικού με τις απαιτήσεις της βάρδιας, π.χ.:

```

if (caregiver.distance < SOFT_DISTANCE) {
    score += 10;
} else if (caregiver.distance <= HARD_DISTANCE) {
    score += 5;
} else {
    score -= 10;
}

```

Η τελική επιλογή πραγματοποιείται ως εξής:

```

const sorted = rankNursesGreedy(...).sort((a, b) => b.score - a.score);
const bestMatch = sorted[0];

```

Η παραπάνω γραμμή αντικατοπτρίζει τη βασική αρχή του άπληστου αλγορίθμου: επιλογή της τοπικά βέλτιστης λύσης χωρίς πρόβλεψη μελλοντικών επιπτώσεων.

Πίνακας 3.1: Κριτήρια άπληστου αλγορίθμου και η υλοποίησή τους στην Π.Ε.

Κριτήριο	Υλοποίηση
Τοπική βελτιστοποίηση	Επιλέγεται το μέλος με μέγιστη $S(c, s)$
Χαμηλή πολυπλοκότητα	Υλοποιείται με γραμμική διέλευση και ταξινόμηση
Μη εγγύηση ολικής βελτιστοποίησης	Δεν λαμβάνει υπόψη παρελθούσες ή μελλοντικές αναθέσεις

Η συγκεκριμένη προσέγγιση είναι κατάλληλη για περιπτώσεις όπου ζητείται ταχύτητα και ευκολία στην υλοποίηση, ενώ η ποιότητα της λύσης είναι επαρκής ακόμη και χωρίς παγκόσμια βελτιστοποίηση.

3.3 WSM

3.3.1 Εισαγωγή

Η μέθοδος του Μοντέλου Σταθμισμένου Αθροίσματος (Weighted Sum Model - WSM) αναφέρεται σε μια σειρά μεθοδολογιών υποστήριξης αποφάσεων που χρησιμοποιούνται για την αξιολόγηση και την κατάταξη εναλλακτικών λύσεων με βάση πολλαπλά, συχνά αντικρουόμενα, κριτήρια. Σε αντίθεση με τις μεθόδους βελτιστοποίησης ενός στόχου, η WSM αναγνωρίζει ρητά τους συμβιβασμούς που είναι εγγενείς στα προβλήματα αποφάσεων του πραγματικού κόσμου, όπου πρέπει να λαμβάνονται ταυτόχρονα υπόψη στόχοι όπως το κόστος, η αποδοτικότητα και η ικανοποίηση του χρήστη. Η μέθοδος WSM είναι ιδιαίτερα πολύτιμη σε πολύπλοκα περιβάλλοντα όπως η υγειονομική περίθαλψη, η εφοδιαστική και η κατανομή πόρων, όπου οι αποφάσεις πρέπει να ικανοποιούν μια ποικιλία ποιοτικών και ποσοτικών παραγόντων.

3.3.2 Ιστορικό πλαίσιο

Η μέθοδος του Μοντέλου Σταθμισμένου Αθροίσματος αποτελεί μία από τις παλαιότερες και απλούστερες τεχνικές Πολυκριτηριακής Λήψης Αποφάσεων (MCDM). Η βασική της αρχή έγκειται στον υπολογισμό της συνολικής αξίας κάθε εναλλακτικής λύσης ως το άθροισμα των επιδόσεών της σε κάθε κριτήριο, πολλαπλασιασμένων με τα αντίστοιχα βάρη των κριτηρίων.

Οι πρώτες εφαρμογές της μεθόδου εμφανίστηκαν τη δεκαετία του 1950, με χαρακτηριστικό παράδειγμα τη μελέτη των Churchman, Ackoff και Smith το 1954, οι οποίοι χρησιμοποίησαν την προσθετική προσέγγιση για την αξιολόγηση εναλλακτικών σε προβλήματα επιλογής επενδύσεων [19]. Η θεωρητική θεμελίωση της μεθόδου ενισχύθηκε από τον Fishburn το 1967, ο οποίος ανέπτυξε το πλαίσιο της προσθετικής χρησιμότητας για την αντιμετώπιση προβλημάτων με πολλαπλά κριτήρια.

Η απλότητα και η ευκολία εφαρμογής της μεθόδου WSM συνέβαλαν στην ευρεία διάδοσή της σε διάφορους τομείς, όπως η διαχείριση έργων, ο σχεδιασμός μεταφορών, η περιβαλλοντική αξιολόγηση και η επιλογή τεχνολογιών. Ωστόσο, η μέθοδος παρουσιάζει περιορισμούς, ιδιαίτερα όταν τα κριτήρια έχουν διαφορετικές μονάδες μέτρησης ή όταν απαιτείται η αντιμετώπιση ποιοτικών κριτηρίων.

3.3.3 Πλαίσιο εφαρμογής

Στο πλαίσιο του προγραμματισμού βάρδιας του νοσηλευτικού προσωπικού, το WSM χρησιμοποιείται για την αξιολόγηση της καταλληλότητας κάθε μέλους του νοσηλευτικού προσωπικού για μια δεδομένη βάρδια με βάση πολλαπλά κριτήρια. Σε αυτά περιλαμβάνονται η απόσταση μεταξύ μέλους και βάρδιας, η αντιστοιχία δεξιοτήτων, η ευθυγράμμιση προτιμήσεων (π.χ. για νύχτες ή σαββατοκύριακα) και άλλοι ειδικοί παράγοντες. Κάθε κριτήριο συμβάλλει σε μια συνολική βαθμολογία που αντικατοπτρίζει την ποιότητα της αντιστοίχισης μέλους και βάρδιας.

Με τη χρήση WSM, η διαδικασία λήψης αποφάσεων γίνεται σαφής και συστηματική, επιτρέποντας στους φορείς να σταθμίζουν τα κριτήρια ανάλογα με τη στρατηγική τους σημασία και τις οργανωτικές τους αξίες.

3.3.4 Μαθηματική διατύπωση

Έστω $A = \{a_1, a_2, \dots, a_n\}$ ένα σύνολο εναλλακτικών λύσεων απόφασης (στην προκειμένη περίπτωση, επιλέξιμα μέλη του νοσηλευτικού προσωπικού για μια δεδομένη βάρδια) και $C = \{c_1, c_2, \dots, c_m\}$ ένα σύνολο κριτηρίων αξιολόγησης (π.χ. απόσταση, αντιστοιχία δεξιοτήτων, προτίμηση). Κάθε εναλλακτική λύση a_i αξιολογείται σύμφωνα με κάθε κριτήριο c_j για να παραχθεί ένας πίνακας αποφάσεων:

$$X = [x_{ij}], \text{ όπου } x_{ij} \in \mathbb{R} \quad (4.4)$$

Για να συγκεντρωθούν οι αξιολογήσεις πολλαπλών κριτηρίων σε μια ενιαία σύνθετη βαθμολογία, χρησιμοποιείται ένα μοντέλο σταθμισμένου αθροίσματος (Weighted Sum Model – WSM):

$$S(a, i) = \sum_{j=1}^m w_j \cdot x_{ij} \quad (4.5)$$

όπου:

- $S(a, i)$ είναι η συνολική βαθμολογία της εναλλακτικής a_{ji}
- w_j είναι το βάρος που σχετίζεται με το κριτήριο c_{ji} έτσι ώστε $\sum_{j=1}^m w_j = 1$
- x_{ij} είναι η κανονικοποιημένη βαθμολογία της μεταβλητής a_i σε σχέση με το κριτήριο c_j

Η κανονικοποίηση είναι συχνά απαραίτητη για να εξασφαλιστεί η συγκρισιμότητα μεταξύ κριτηρίων διαφορετικής φύσης. Για παράδειγμα:

- Για κριτήρια οφέλους: $x'_{ij} = \frac{x_{ij}}{\max_i x_{ij}}$
- Για κριτήρια κόστους: $x'_{ij} = \frac{\min_i x_{ij}}{x_{ij}}$

3.3.5 Θεωρητικές εκτιμήσεις

Το WSM προσδίδει μια δομημένη, κατανοητή προσέγγιση σε πολύπλοκα καθήκοντα λήψης αποφάσεων, καθιστώντας τις αντισταθμίσεις σαφείς και ποσοτικοποιήσιμες. Ωστόσο, είναι ευαίσθητη στην επιλογή και τη στάθμιση των κριτηρίων και μπορεί να απαιτεί εξειδίκευση στον τομέα για την κατάλληλη παραμετροποίηση. Ενώ απλές μέθοδοι όπως το WSM είναι εύχρηστες και αποδοτικές, πιο εξελιγμένες τεχνικές (π.χ. AHP, TOPSIS ή ELECTRE) μπορεί να είναι καταλληλότερες σε περιβάλλοντα υψηλής αβεβαιότητας ή όταν απαιτείται χειρισμός ασαφών δεδομένων.

Στο πλαίσιο του χρονοπρογραμματισμού, το WSM παρέχει ένα σταθερό θεμέλιο πάνω στο οποίο ευρετικοί ή βελτιστοποιητικοί αλγόριθμοι (όπως άπληστοι ή knapsack) μπορούν να λάβουν τεκμηριωμένες, συνεπείς επιλογές.

3.3.6 Υλοποίηση στον κώδικα

Η υλοποίηση του WSM πραγματοποιείται στη συνάρτηση `calculateFitScoreWSM`, η οποία αξιολογεί κάθε μέλος του νοσηλευτικού προσωπικού ως προς μια δεδομένη βάρδια. Η συνάρτηση αυτή ακολουθεί τα εξής βασικά βήματα:

Αντιστοίχιση Δεξιοτήτων: Υπολογίζεται ο αριθμός κοινών δεξιοτήτων μεταξύ του μέλους και των αναγκών της βάρδιας. Κάθε ταιριαστή δεξιότητα ενισχύει τη βαθμολογία κατά 20 μονάδες:

```
const matchingCompetencies =
  shift.skills.filter((skill) =>
    caregiver.skills.includes(skill),
  ).length;
let score = matchingCompetencies * 20;
```

Απόσταση: Αν το μέλος βρίσκεται εντός αποδεκτής απόστασης (SOFT_DISTANCE), του προστίθενται 10 μονάδες. Για αποστάσεις μεταξύ SOFT_DISTANCE και HARD_DISTANCE, αφαιρείται σταθμισμένη ποινή. Για αποστάσεις πέραν του HARD_DISTANCE, αφαιρούνται 100 μονάδες:

```
if (caregiver.distance < SOFT_DISTANCE) {
    score += 10;
} else if (caregiver.distance <= HARD_DISTANCE) {
    score -= (caregiver.distance - SOFT_DISTANCE) * weights.distanceWeight;
} else {
    score -= 100;
}
```

Προτιμήσεις για βάρδιες νύχτας / Σαββατοκύριακου: Αν οι προτιμήσεις του μέλους ευθυγραμμίζονται με τον τύπο βάρδιας (νύχτα ή Σαββατοκύριακο), προστίθενται μονάδες σύμφωνα με τα αντίστοιχα βάρη:

```
if (isNightShift(shift) === caregiver.prefersNights) {
    score += weights.nightWeight * 5;
}

if (isWeekendShift(shift) === caregiver.prefersWeekends) {
    score += weights.weekendWeight * 5;
}
```

Κανονικοποίηση: Οι τελικές βαθμολογίες όλων των μελών κανονικοποιούνται ώστε να κυμαίνονται από 0 έως 100. Η μέγιστη βαθμολογία μεταφράζεται σε 100%, και οι υπόλοιπες αναλογικά χαμηλότερες:

```
const maxScore = Math.max(...caregivers.map((caregiver) => caregiver.score));
return caregivers.map((caregiver) => ({
    ...caregiver,
    percentage:
        caregiver.score > 0 && maxScore > 0
            ? Math.round((caregiver.score / maxScore) * 1000) / 10
            : 0,
}));
```

Η προσέγγιση αυτή ενσωματώνει τις βασικές αρχές του WSM, χρησιμοποιώντας όμως ρητή και προσαρμοσμένη βαθμολόγηση αντί για πίνακες αποφάσεων, κάνοντας την υλοποίηση περισσότερο ευέλικτη και κατανοητή. Η αρθρωτή δομή επιτρέπει εύκολη προσαρμογή των βαρών και των κανόνων βαθμολόγησης με βάση το εκάστοτε επιχειρησιακό πλαίσιο.

3.4 Simulated Annealing

3.4.1 Εισαγωγή

Η Προσομοιωμένη Ανόπτηση (Simulated Annealing - SA) είναι ένας στοχαστικός αλγόριθμος βελτιστοποίησης εμπνευσμένος από τη φυσική διαδικασία ανόπτησης μετάλλων. Σε περιβάλλοντα όπου οι αποφάσεις περιλαμβάνουν συνδυαστικούς περιορισμούς και πολλαπλά τοπικά άριστα, η SA προσφέρει έναν αποτελεσματικό τρόπο εξεύρεσης καλών λύσεων χωρίς να «παγιδεύεται» σε μη βέλτιστες καταστάσεις. Η μέθοδος ενσωματώνει μια ελεγχόμενη ποσότητα τυχαιότητας ώστε να εξερευνά τον χώρο των λύσεων προοδευτικά.

3.4.2 Ιστορικό πλαίσιο

Η τεχνική αναπτύχθηκε τη δεκαετία του 1980 από τους Kirkpatrick et al., οι οποίοι εισήγαγαν τη μέθοδο της προσομοιωμένης ανόπτησης ως μια ισχυρή στρατηγική βελτιστοποίησης για πολύπλοκα προβλήματα [20]. Η μέθοδος SA έχει χρησιμοποιηθεί ευρέως σε ποικιλία τομέων, όπως προβλήματα θεωρίας γράφων (π.χ. εύρεση ελάχιστου δένδρου κάλυψης), προγραμματισμού παραγωγής (π.χ. βελτιστοποίηση χρονοδιαγραμμάτων) και κατανομής πόρων σε έργα [21]. Σε αντίθεση με άλλες κλασικές μεθόδους αναζήτησης, η SA ξεχωρίζει γιατί επιτρέπει ελεγχόμενες υποδεέστερες κινήσεις, που εξομοιώνουν τη διαδικασία ανόπτησης μετάλλων, ώστε να μειώνεται ο κίνδυνος εγκλωβισμού σε τοπικά μέγιστα και να εξασφαλίζεται μια πιο ολοκληρωμένη εξερεύνηση του χώρου λύσεων. Η τεχνική αυτή έχει αποδειχθεί ιδιαίτερα ευέλικτη, επιτρέποντας την προσαρμογή της σε διαφορετικά είδη προβλημάτων, ενώ η χρήση της έχει επεκταθεί σημαντικά σε σύγχρονα βιομηχανικά συστήματα και συστήματα πληροφορικής, λόγω της ικανότητάς της να χειρίζεται μεγάλες και πολύπλοκες αναζητήσεις.

3.4.3 Πλαίσιο εφαρμογής

Στην παρούσα εφαρμογή, η SA χρησιμοποιείται για τη βελτιστοποιημένη ανάθεση μελών του νοσηλευτικού προσωπικού σε βάρδιες. Η προσέγγιση ξεκινά με μία αρχική (άπληστη) λύση και εφαρμόζει στοχαστικές αλλαγές (neighbor moves) ώστε να διερευνήσει εναλλακτικές κατανομές. Ο στόχος είναι να μεγιστοποιηθεί η συνολική καταλληλότητα των αναθέσεων, όπως αυτή ορίζεται από μια συνάρτηση κόστους που βασίζεται σε πολλαπλά κριτήρια.

3.4.4 Μαθηματική διατύπωση

Η SA εφαρμόζει την εξής στρατηγική:

- Ξεκινά από μια αρχική λύση με βάση αλγόριθμο επιλογής του χρήστη (π.χ. άπληστος): S_0
- Σε κάθε βήμα παράγει μια γειτονική λύση S'
- Αν S' είναι καλύτερη: αποδέχεται S'
- Αν είναι χειρότερη: αποδέχεται S' με πιθανότητα: $P = e^{-\Delta E/T}$

όπου:

- $\Delta E = \text{Score}(S) - \text{Score}(S')$
- T : θερμοκρασία που μειώνεται προοδευτικά

Η διαδικασία συνεχίζεται έως ότου:

- Η θερμοκρασία μειωθεί αρκετά
- Δεν παρατηρηθεί βελτίωση για πολλαπλές επαναλήψεις

3.4.5 Εκτίμηση και προοπτικές

Η SA είναι κατάλληλη για προβλήματα μεγάλης πολυπλοκότητας, όπου πλήρης εξερεύνηση όλων των συνδυασμών είναι απαγορευτική. Επιτρέπει υπολογιστικά ελαφριά βελτιστοποίηση και προσαρμογή σε επιχειρησιακές απαιτήσεις. Ωστόσο, η απόδοσή της εξαρτάται από παραμέτρους όπως:

- Αρχική θερμοκρασία
- Ρυθμός ψύξης (cooling rate)
- Μέγεθος γειτονιάς (neighbor functions)
- Κριτήρια αποδοχής

3.4.6 Υλοποίηση στον κώδικα

Η συνάρτηση `assignCaregiversWithSimulatedAnnealing` υλοποιεί την παραπάνω στρατηγική βελτιστοποίησης με τα ακόλουθα στάδια:

1. Αρχικοποίηση με Άπληστη Λύση

Η αρχική λύση δημιουργείται με άπληστη επιλογή μελών για κάθε βάρδια:

```
for (const shift of result) {  
  const rankedCaregivers = getNursesSortedByFit(...);  
  const selectedCaregiver = pickFromTop(rankedCaregivers);  
  shift.assignedCaregiver = selectedCaregiver;  
}
```

Αυτό δημιουργεί ένα ισχυρό σημείο εκκίνησης για περαιτέρω βελτιστοποίηση.

2. Ορισμός παραμέτρων ανόπτησης

Ρυθμίζονται οι βασικές παράμετροι:

```
initialTemperature = 100;  
coolingRate = 0.97;  
minTemperature = 0.01;  
maxIterationsPerTemperature = max(100, shifts.length * 2);
```

Οι τιμές έχουν επιλεγθεί ώστε να επιτρέπουν σταδιακή εξερεύνηση με έλεγχο τυχαιότητας.

3. Γειτονικές Κινήσεις (Neighbor Moves)

Η SA παράγει νέες λύσεις με δύο τύπους μετακινήσεων:

- i. Ανταλλαγή μελών μεταξύ δύο βάρδιών

```
swap(shift1.assignedCaregiver, shift2.assignedCaregiver)
```

- ii. Επανα-ανάθεση μέλους σε τυχαία βάρδια

```
reassign(shift, getNursesSortedByFit(...))
```

Οι κινήσεις επιλέγονται με πιθανότητα 50% - 50%.

4. Κριτήριο αποδοχής

Κάθε νέα λύση αξιολογείται και γίνεται δεκτή με βάση:

- Αν είναι καλύτερη → αποδοχή
- Αν είναι χειρότερη → αποδοχή με πιθανότητα:

```
const acceptanceProbability = Math.exp(scoreDiff / temperature);
if (Math.random() < acceptanceProbability) accept();
```

5. Ψύξη (Cooling)

Η θερμοκρασία μειώνεται σταδιακά:

```
temperature *= coolingRate;
```

6. Τελική διόρθωση αναθέσεων

Για οποιαδήποτε βάρδια παραμένει χωρίς μέλος νοσηλευτικού προσωπικού στο τέλος της διαδικασίας, γίνεται προσπάθεια να ανατεθεί στα εναπομείναντα διαθέσιμα μέλη:

```
if (!assignedCaregiver) {
  assignFromAvailable(rankedCaregivers);
}
```

Πίνακας 3.2: Πίνακας συμμόρφωσης με αρχές Simulated Annealing

Αρχή SA	Υλοποίηση
Τυχαία αποδοχή χειρότερων λύσεων	Metropolis criterion με εξαρτώμενη πιθανότητα
Ψύξη θερμοκρασίας	Εκθετική με σταθερό ρυθμό (coolingRate = 0.97)
Πολλαπλές κινήσεις γειτονιάς	Swap & Reassign μετακινήσεις
Αρχικοποίηση με εύλογη αρχική λύση	Greedy κατανομή
Περιορισμός επαναλήψεων χωρίς πρόοδο	Μετρητής iterationsWithoutImprovement
Εξέταση εναλλακτικών λύσεων σε βάθος	maxIterationsPerTemperature * 3

3.5 Tabu Search

3.5.1 Εισαγωγή

Ο αλγόριθμος Tabu Search αποτελεί μια προηγμένη μεταερευνητική τεχνική βελτιστοποίησης που προτείνει τη χρήση μηχανισμών μνήμης (tabu lists) ώστε να καθοδηγεί την αναζήτηση μέσα σε δύσκολους χώρους λύσεων, αποφεύγοντας την επανεπίσκεψη σε ήδη δοκιμασμένες ή ανεπιθύμητες λύσεις. Σε αντίθεση με τον άπληστο αλγόριθμο που εστιάζει σε τοπικές βελτιώσεις, ο Tabu Search εξετάζει και «χειρότερες» γειτονικές λύσεις με σκοπό τη διαφυγή από τοπικά άκρα (minima ή maxima) και την εξεύρεση καλύτερων παγκόσμιων λύσεων.

Η θεμελιώδης ιδέα είναι η εξερεύνηση του χώρου των λύσεων με ελεγχόμενη ανατροφοδότηση, χρησιμοποιώντας μια λίστα απαγορευμένων μετακινήσεων (tabu moves), η οποία εμποδίζει την επιστροφή σε πρόσφατες καταστάσεις, εκτός αν η μετακίνηση προσφέρει σημαντική βελτίωση.

3.5.2 Ιστορικό πλαίσιο

Ο Tabu Search προτάθηκε από τον Fred W. Glover στα τέλη της δεκαετίας του 1980, αρχικά ως μέθοδος επίλυσης συνδυαστικών προβλημάτων (π.χ. πρόβλημα πωλητή, βελτιστοποίηση δρομολογίων, ανάθεση πόρων) [22]. Η συμβολή του Glover έθεσε τα θεμέλια για έναν νέο τύπο αλγορίθμων που βασίζονται σε ευφυή αναζήτηση και ιστορική μνήμη [23]. Η τεχνική γνώρισε ευρεία αποδοχή τη δεκαετία του 1990 και εφαρμόστηκε σε τομείς όπως τα logistics, η επιχειρησιακή έρευνα και η τεχνητή νοημοσύνη, καθώς ενσωμάτωσε την έννοια της «αναζήτησης με στρατηγική μνήμη», ξεπερνώντας την περιορισμένη αντίληψη των τοπικών αναζητήσεων.

3.5.3 Πλαίσιο εφαρμογής

Στο συγκεκριμένο πλαίσιο χρονοπρογραμματισμού του νοσηλευτικού προσωπικού, ο αλγόριθμος Tabu Search εφαρμόζεται με σκοπό την παγκόσμια βελτιστοποίηση της αντιστοίχισης βαρδιών και μελών του προσωπικού. Αντί να εξετάζεται κάθε βάρδια ανεξάρτητα, η προσέγγιση στοχεύει σε ολιστική εκχώρηση του προσωπικού σε όλες τις βάρδιες ταυτόχρονα, επιδιώκοντας τη μεγιστοποίηση του συνολικού σκορ καταλληλότητας του συστήματος.

Ο αλγόριθμος ξεκινά από μια τυχαία αρχικοποιημένη λύση και, μέσω διαδοχικών επαναλήψεων, προσπαθεί να την βελτιώσει, ερευνώντας τοπικά εναλλακτικά σενάρια (neighboring solutions). Κάθε εναλλαγή νοσηλευτή σε μια βάρδια αποτελεί μια πιθανή κίνηση (move). Αν η κίνηση αυτή δεν περιλαμβάνεται στη λίστα Tabu, τότε μπορεί να επιλεγεί. Σε κάθε επανάληψη, αποθηκεύεται η καλύτερη λύση που έχει βρεθεί έως τότε, ακόμη κι αν το βήμα της παρούσας επανάληψης δεν βελτιώνει άμεσα το συνολικό σκορ.

3.5.4 Μαθηματική διατύπωση

Έστω:

- $S = \{s_1, s_2, \dots, s_n\}$ το σύνολο των βαρδιών,
- $C = \{c_1, c_2, \dots, c_m\}$ το σύνολο των νοσηλευτών,
- $A: S \rightarrow C \cup \{\emptyset\}$ μια εκχώρηση βαρδιών σε προσωπικό,
- $f(A) \in \mathbb{R}$ η συνολική βαθμολογία της λύσης A , η οποία υπολογίζεται ως το άθροισμα των επιμέρους τιμών καταλληλότητας $S(c, s)$ για κάθε εκχώρηση $A(s) = c$.

Ο αλγόριθμος επιδιώκει να υπολογίσει:

$$A = \underset{A \in A}{\operatorname{argmax}} f(A) \quad (4.6)$$

όπου A το σύνολο των επιτρεπτών εκχωρήσεων, και η αναζήτηση εκτελείται μέσω μετακινήσεων της μορφής:

$$\text{move} : A \rightarrow A' \quad (4.7)$$

όπου A' είναι το αποτέλεσμα της αντικατάστασης του ανατεθειμένου νοσηλευτή σε μία βάρδια. Η κίνηση (s_i, c_j) τίθεται στη λίστα Tabu για προκαθορισμένο αριθμό επαναλήψεων, ώστε να αποτραπεί η επιστροφή σε προηγούμενες επιλογές.

3.5.5 Εκτίμηση και προοπτικές

Ο Tabu Search υπερτερεί έναντι απλούστερων αλγορίθμων (όπως ο άπληστος) στο ότι προβλέπει και διορθώνει κακές αναθέσεις, αποφεύγει κύκλους και παρέχει καλύτερα αποτελέσματα σε προβλήματα με πολλές διαστάσεις και αντιφάσεις. Χάρη στην ευελιξία του, μπορεί να ενσωματώσει κριτήρια προτίμησης, απόσταση, δεξιότητες, και τύπο βάρδιας, και να παράγει ποιοτικές λύσεις ακόμη και όταν η αρχική κατάσταση είναι κακή.

Η χρήση ταμπού λίστας μειώνει τον κίνδυνο να παγιδευτεί ο αλγόριθμος σε τοπικά μέγιστα, ενώ η επιλογή χειρότερων μετακινήσεων σε συγκεκριμένες στιγμές ενισχύει τη στρατηγική διαφοροποίηση.

3.5.6 Υλοποίηση στον κώδικα

Η υλοποίηση του αλγορίθμου Tabu Search εντοπίζεται κυρίως στη συνάρτηση κώδικα `assignCaregiversWithTABUSearch`, η οποία εφαρμόζει τον αλγόριθμο σε ένα σύνολο βαρδιών και μελών του νοσηλευτικού προσωπικού.

3.5.6.1 Αρχικοποίηση

Η αρχική λύση προκύπτει μέσω του αρχικού αλγορίθμου που επιλέγει ο χρήστης και πραγματοποιεί ανάθεση κάθε βάρδιας σε διαθέσιμο μέλος:

```
const randomCaregiver = availableCaregivers[randomIndex];
shift.assignedCaregiver = randomCaregiver;
```

3.5.6.2 Υπολογισμός καταλληλότητας

Για κάθε νέα υποψήφια εκχώρηση, υπολογίζεται η συνολική βαθμολογία:

```
const newScore: number = calculateSolutionScore(  
  newSolution,  
  caregivers,  
  weights,  
  algorithmType,  
);
```

Η συνάρτηση `calculateSolutionScore` προσθέτει τις επιμέρους τιμές `score` από το `getNursesSortedByFit`, λαμβάνοντας υπόψη:

- Δεξιότητες (skill match)
- Απόσταση
- Προτιμήσεις για νυχτερινές και βάρδιες σαββατοκύριακο
- Τον αλγόριθμο σύγκρισης που επιθυμεί ο χρήστης

3.5.6.3 Ταμπού λίστα

Κάθε αλλαγή σε εκχώρηση προστίθεται στη λίστα:

```
tabuList.push({  
  shiftId: bestMove.shiftId,  
  caregiverId: bestMove.oldCaregiverId,  
});
```

Η λίστα διατηρείται σε σταθερό μέγεθος:

```
if (tabuList.length > tabuListSize) {  
  tabuList.shift();  
}
```

3.5.6.4 Βέλτιστη λύση

Η καλύτερη λύση διατηρείται και ενημερώνεται όταν εντοπίζεται βελτίωση:

```
if (currentScore > bestScore) {  
  bestSolution = [...currentSolution];  
  bestScore = currentScore;  
}
```

Πίνακας 3.3: Κριτήρια αλγορίθμου Tabu και η υλοποίησή τους στην Π.Ε.

Κριτήριο	Υλοποίηση
Εξερεύνηση χώρου λύσεων	Με εναλλαγές νοσηλευτών στις βάρδιες
Μη τοπική βελτιστοποίηση	Επιτρέπεται η επιλογή χειρότερων λύσεων προσωρινά
Χρήση ιστορικής μνήμης	Μέσω της λίστας Tabu
Προσαρμοστικότητα	Ενσωμάτωση διαφορετικών παραμέτρων & βαρών

Η συγκεκριμένη μεταερευνητική μέθοδος είναι ιδανική όταν η πολυπλοκότητα του προβλήματος καθιστά τις άπληστες προσεγγίσεις ανεπαρκείς και απαιτείται συνολική βελτιστοποίηση με ελεγχόμενες συμβιβαστικές λύσεις.

3.6 TOPSIS

3.6.1 Εισαγωγή

Η μέθοδος TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) αποτελεί μια ευρέως χρησιμοποιούμενη προσέγγιση στην πολυκριτηριακή λήψη αποφάσεων. Στόχος της είναι η κατάταξη εναλλακτικών λύσεων βάσει της εγγύτητάς τους προς την «ιδανική» λύση και της απομάκρυνσής τους από τη «χειρότερη» δυνατή. Σε αντίθεση με άπληστους αλγορίθμους, η μέθοδος TOPSIS λαμβάνει υπόψη το σύνολο των εναλλακτικών και των κριτηρίων, προσφέροντας μια πιο σφαιρική και μαθηματικά συνεπή εκτίμηση της καταλληλότητας κάθε επιλογής. Η χρήση της είναι ιδιαίτερα διαδεδομένη σε περιβάλλοντα όπου συνυπάρχουν πολλαπλά, ενδεχομένως αντικρουόμενα, κριτήρια.

3.6.2 Ιστορικό πλαίσιο

Η μέθοδος TOPSIS προτάθηκε αρχικά από τους Hwang και Yoon το 1981 ως μία από τις πρώτες πρακτικές και μαθηματικά συνεπείς τεχνικές πολυκριτηριακής ανάλυσης αποφάσεων. Βασίζεται στην αρχή ότι η βέλτιστη επιλογή είναι αυτή που βρίσκεται πλησιέστερα στην ιδανική λύση και ταυτόχρονα πιο απομακρυσμένη από την αντι-ιδανική. Η μέθοδος απέκτησε σημαντική απήχηση λόγω της απλότητας, της υπολογιστικής της αποδοτικότητας και της δυνατότητάς της να εφαρμόζεται σε προβλήματα με πολλαπλά, ενδεχομένως αντικρουόμενα, κριτήρια. Έχει εφαρμοστεί ευρέως σε τομείς όπως η διαχείριση έργων, η αξιολόγηση κινδύνου, η επιλογή προσωπικού, η ανάλυση προμηθευτών και, πιο πρόσφατα, με αυξανόμενο ενδιαφέρον, στον τομέα της υγειονομικής περίθαλψης και τον χρονοπρογραμματισμό προσωπικού [24], [25]. Η TOPSIS πλεονεκτεί έναντι απλούστερων τεχνικών καθώς επιτρέπει την ολοκληρωμένη ενσωμάτωση τόσο ποσοτικών όσο και ποιοτικών δεδομένων, μέσω κατάλληλης κανονικοποίησης και στάθμισης κριτηρίων, καθιστώντας την κατάλληλη για σύνθετες αποφάσεις σε πολυδιάστατα περιβάλλοντα. Η χρήση της σε αποφάσεις προσωπικού έχει επιβεβαιωθεί σε περιπτώσεις αξιολόγησης εργαζομένων με βάση πολλαπλά κριτήρια (όπως ψυχομετρικά τεστ, δεξιότητες και συνεντεύξεις), βελτιώνοντας την ακρίβεια και τη διαφάνεια της διαδικασίας επιλογής [26].

3.6.3 Πλαίσιο εφαρμογής

Στο πλαίσιο ανάθεσης νοσηλευτικού προσωπικού, η TOPSIS εφαρμόζεται για την κατάταξη υποψήφιων μελών του νοσηλευτικού προσωπικού, ως προς την καταλληλότητά τους να καλύψουν μια βάρδια. Η αξιολόγηση βασίζεται σε τέσσερα βασικά κριτήρια:

- **SkillMatch:** ποσοστό δεξιοτήτων που καλύπτονται
- **Distance:** κανονικοποιημένη εγγύτητα στην τοποθεσία του ασθενούς
- **NightShiftPreference:** συμφωνία με νυχτερινή βάρδια
- **WeekendShiftPreference:** συμφωνία με βάρδια Σαββατοκύριακου

Τα κριτήρια σταθμίζονται ανάλογα με τη σημασία τους και συγκεντρώνονται μέσω της διαδικασίας TOPSIS σε μια τελική βαθμολογία για κάθε μέλος του νοσηλευτικού προσωπικού.

3.6.4 Πλαίσιο εφαρμογής

Η μέθοδος TOPSIS μετατρέπει τις επιδόσεις κάθε εναλλακτικής (π.χ. caregiver) σε μια τελική βαθμολογία που αντανακλά την εγγύτητα στην ιδανική λύση. Ξεκινά από έναν πίνακα αποφάσεων $X = [x_{ij}]$, όπου:

- a_i : εναλλακτική (caregiver)
- c_j : κριτήριο (π.χ. δεξιότητες, απόσταση)
- x_{ij} : επίδοση του a_i στο c_j

Ακολουθούν τα βασικά βήματα:

1. Κανονικοποίηση

Οι τιμές κανονικοποιούνται για να βρίσκονται στην ίδια κλίμακα:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^n x_{ij}^2}} \quad (4.8)$$

2. Στάθμιση

Οι κανονικοποιημένες τιμές πολλαπλασιάζονται με βάρη w_j που καθορίζουν τη σημασία κάθε κριτηρίου:

$$v_{ij} = w_j \cdot r_{ij} \quad (4.9)$$

3. Ιδανικές λύσεις

Ορίζονται με βάση τον τύπο:

$$v_j^+ = \max_i v_{ij}, \quad v_j^- = \min_i v_{ij} \quad (4.10)$$

Η ιδανική λύση είναι αυτή με τις καλύτερες τιμές σε όλα τα κριτήρια· η αρνητική, η χειρότερη.

4. Απόσταση από τις λύσεις

Υπολογίζονται οι Ευκλείδειες αποστάσεις:

$$D_i^+ = \sqrt{\sum_j (v_{ij} - v_j^+)^2}, \quad D_i^- = \sqrt{\sum_j (v_{ij} - v_j^-)^2} \quad (4.11)$$

5. Σχετική εγγύτητα

Η τελική βαθμολογία βασίζεται στο πόσο κοντά βρίσκεται κάθε caregiver στην ιδανική λύση:

$$RC_i = \frac{D_i^-}{D_i^- + D_i^+} \Rightarrow Score = RC_i \cdot 100 \quad (4.12)$$

Όσο μεγαλύτερη η τιμή RC_i , τόσο καλύτερη η επιλογή.

3.6.5 Υλοποίηση στον Κώδικα

Η μέθοδος TOPSIS υλοποιείται στη συνάρτηση `calculateFitScoreTOPSIS`, η οποία εφαρμόζεται για την κατάταξη των μελών του νοσηλευτικού προσωπικού ως προς μια συγκεκριμένη βάρδια. Τα στάδια έχουν ως εξής:

1. Κανονικοποίηση

Για κάθε caregiver υπολογίζονται τέσσερα βασικά κριτήρια: ποσοστό δεξιοτήτων που καλύπτονται, κανονικοποιημένη απόσταση από τον ασθενή, και δυαδική αντιστοίχιση προτιμήσεων (νύχτες / σαββατοκύριακα). Κάθε caregiver επιστρέφει ένα αντικείμενο:

```
{
  skills: matchedSkills / totalSkills,
  distance: normalizedDistance,
  nightShift: matchNight ? 1 : 0,
  weekendShift: matchWeekend ? 1 : 0
}
```

Αυτό το βήμα αντιστοιχεί στην κατασκευή του πίνακα $X = [x_{ij}]$.

2. Κανονικοποίηση κριτηρίων

Οι τιμές κάθε κριτηρίου κανονικοποιούνται μέσω διανυσματικής κανονικοποίησης ώστε να είναι μεταξύ 0 και 1. Αυτό διασφαλίζει συγκρισιμότητα μεταξύ διαφορετικών μεγεθών:

```
normalized[criterion] = value / sqrt(sumOfSquares)
```

3. Στάθμιση με βάση τα βάρη

Τα normalized scores πολλαπλασιάζονται με βάρη που δίνονται μέσω του αντικειμένου weights, π.χ.:

```
weighted[criterion] = normalized[criterion] * weights.criterionWeight
```

Έτσι διαμορφώνεται ο πίνακας $V = [v_{ij}]$.

4. Εύρεση ιδανικών λύσεων

Για κάθε κριτήριο υπολογίζονται οι μέγιστες (ιδανικές) και ελάχιστες (αντι-ιδανικές) τιμές:

```
ideal[criterion] = Math.max(...)
negativeIdeal[criterion] = Math.min(...)
```

Αυτό ορίζει το εύρος των βέλτιστων και χειρότερων επιδόσεων.

5. Υπολογισμός αποστάσεων

Υπολογίζεται η Ευκλείδεια απόσταση κάθε caregiver από την ιδανική και αντι-ιδανική λύση:

```
distanceToIdeal = sqrt(sum((value - ideal)^2))
distanceToNegativeIdeal = sqrt(sum((value - negative)^2))
```

6. Σχετική εγγύτητα και τελική κατάταξη

Η σχετική εγγύτητα (relative closeness) υπολογίζεται με βάση τις δύο αποστάσεις:

```
relativeCloseness =
    distanceToNegative / (distanceToNegative + distanceToIdeal)
score = relativeCloseness * 100
```

Κατόπιν, τα μέλη ταξινομούνται φθίνοντα με βάση τη βαθμολογία:

```
rankedCaregivers.sort((a, b) => b.score - a.score)
```

3.6.6 Σύνοψη αλγοριθμικής συμπεριφοράς

Η υλοποίηση του αλγορίθμου πληρεί όλα τα χαρακτηριστικά ενός αλγορίθμου TOPSIS:

Πίνακας 3.4: Κριτήρια αλγορίθμου TOPSIS και η υλοποίησή τους στην Π.Ε.

Κριτήριο	Υλοποίηση
Κανονικοποίηση πίνακα αποφάσεων	Vector normalization
Εφαρμογή σταθμισμένων βαρών	Μέσω αντικειμένου weights
Υπολογισμός ιδανικής και αντι-ιδανικής τιμής	Μέσω min/max σε κάθε κριτήριο
Απόσταση από λύσεις	Ευκλείδεια απόσταση από κάθε στόχο
Σχετική εγγύτητα & τελική κατάταξη	Relative Closeness score με κλίμακα 0-100
Πολυκριτηριακή κατάταξη	Μέσω <code>rankedCaregivers.sort((a, b) => b.score - a.score)</code>

3.7 Επίλογος

Σε αυτό το κεφάλαιο παρουσιάστηκε ένα ευρύ φάσμα αλγοριθμικών προσεγγίσεων για τον χρονοπρογραμματισμό νοσηλευτικού προσωπικού, αναδεικνύοντας τις δυνατότητες και τους περιορισμούς κάθε αλγορίθμου. Ο άπληστος αλγόριθμος διακρίνεται για την υπολογιστική του απλότητα και την ταχύτητα εκτέλεσης, αν και συχνά στερείται στρατηγικής βελτιστοποίησης. Οι πολυκριτηριακές μέθοδοι παρέχουν συστηματικά εργαλεία αξιολόγησης εναλλακτικών λύσεων υπό πολλαπλά κριτήρια, υποστηρίζοντας πιο συνεπείς και τεκμηριωμένες αποφάσεις. Το Simulated Annealing προσθέτει ένα στοχαστικό στοιχείο, προσφέροντας τη δυνατότητα διαφυγής από τοπικά βέλτιστα μέσω ελεγχόμενης αποδοχής υποδεέστερων λύσεων, ενώ ο Tabu Search εκμεταλλεύεται μηχανισμούς μνήμης και στρατηγική εξερεύνηση του χώρου λύσεων για παγκόσμια βελτιστοποίηση.

Η συγκριτική αξιολόγηση των αλγορίθμων αυτών καταδεικνύει πως η επιλογή της κατάλληλης τεχνικής εξαρτάται από τη φύση του προβλήματος, τους επιχειρησιακούς περιορισμούς και το ζητούμενο επίπεδο ποιότητας της λύσης. Ο συνδυασμός ευρετικών, πολυκριτηριακών και μεταευρετικών στρατηγικών επιτρέπει τη δημιουργία ευέλικτων και αποδοτικών συστημάτων λήψης αποφάσεων για τη διαχείριση ανθρώπινου δυναμικού σε περιβάλλοντα υγειονομικής φροντίδας.

Κεφάλαιο 4ο: Πλατφόρμα

4.1 Εισαγωγή

Η επιτυχής υλοποίηση οποιασδήποτε σύγχρονης εφαρμογής λογισμικού βασίζεται σε δύο θεμελιώδεις πυλώνες: μια στιβαρή και αποδοτική αρχιτεκτονική δεδομένων στο backend και ένα διαισθητικό, λειτουργικό περιβάλλον χρήστη στο frontend. Το παρόν κεφάλαιο αφιερώνεται στην αναλυτική παρουσίαση αυτών των δύο κρίσιμων πτυχών της αναπτυχθείσας εφαρμογής. Αρχικά, θα εμβαθύνουμε στη δομή της βάσης δεδομένων, εξετάζοντας τους πίνακες που σχεδιάστηκαν ειδικά για την υποστήριξη των μοναδικών λειτουργιών της εφαρμογής, καθώς και εκείνους που ενσωματώθηκαν αυτόματα μέσω της αξιοποίησης εξειδικευμένων βιβλιοθηκών και τεχνολογιών, αναδεικνύοντας τη συνέργειά τους. Στη συνέχεια, η ανάλυση θα μετατοπιστεί στο περιβάλλον με το οποίο αλληλεπιδρά ο τελικός χρήστης. Θα περιγραφούν όλες οι θρόνες και υπο-θρόνες στις οποίες έχει πρόσβαση ο χρήστης, καθώς και οι ροές εργασίας και οι λειτουργίες των διεπαφών του frontend, δίνοντας έμφαση στον τρόπο με τον οποίο διευκολύνεται η διαχείριση των πληροφοριών και η εκτέλεση των κεντρικών εργασιών της πλατφόρμας, από την εισαγωγή δεδομένων έως την αυθεντικοποίηση και την πλοήγηση.

4.2 Σχεδιασμός βάσης δεδομένων

4.2.1 Κύριοι πίνακες

Για τις ανάγκες της εφαρμογής, δημιουργήθηκαν 3 πίνακες, οι οποίοι σε συνδυασμό με τους 5 που δημιουργήθηκαν για τις ανάγκες των βιβλιοθηκών και των τεχνολογιών που χρησιμοποιήθηκαν, φέρνουν το συνολικό αριθμό πινάκων σε 8. Οι παρακάτω αποτελούν στήλες που περιλαμβάνονται, εν μέρει ή πλήρως, στους πίνακες που αναλύονται μετέπειτα:

- **id:** Ένας μοναδικός ακέραιος αριθμός που δημιουργείται αυτόματα από τη βάση δεδομένων για κάθε νέα εγγραφή. Λειτουργεί ως το πρωτεύον κλειδί του πίνακα και δεν μπορεί να είναι κενός.
- **name:** Μία αλφαριθμητική μεταβλητή που μπορεί να έχει μήκος έως 255 χαρακτήρες. Αυτό το πεδίο είναι υποχρεωτικό (δεν μπορεί να είναι κενό).
- **created_at:** Μια χρονοσφραγίδα που περιλαμβάνει και τη ζώνη ώρας, η οποία καταγράφει αυτόματα την ημερομηνία και την ώρα δημιουργίας της εγγραφής. Η προεπιλεγμένη τιμή είναι η τρέχουσα ημερομηνία και ώρα και το πεδίο είναι υποχρεωτικό.
- **updated_at:** Μια χρονοσφραγίδα που περιλαμβάνει και τη ζώνη ώρας, η οποία καταγράφει την ημερομηνία και την ώρα της τελευταίας ενημέρωσης της εγγραφής. Η προεπιλεγμένη τιμή κατά τη δημιουργία είναι η τρέχουσα ημερομηνία και ώρα και το πεδίο είναι υποχρεωτικό. (Η τιμή αυτή ενημερώνεται αυτόματα κάθε φορά που τροποποιείται η εγγραφή).
- **location:** Ένα γεωμετρικό πεδίο τύπου Σημείου (Point geometry), που αποθηκεύει γεωγραφικές συντεταγμένες. Χρησιμοποιεί το σύστημα αναφοράς συντεταγμένων WGS (World Geodetic System) 84, γνωστό και ως SRID (Spatial Reference System Identifier) 4326. Το πεδίο είναι υποχρεωτικό.
- **skills:** Ένας πίνακας (array) που περιέχει ακέραιους αριθμούς. Κάθε ακέραιος αντιστοιχεί σε ένα μοναδικό αναγνωριστικό μιας προκαθορισμένης δεξιότητας από μια λίστα ενσωματωμένη στον κώδικα της εφαρμογής (π.χ., 1 για 'Δεξιότητα_1', 2 για 'Δεξιότητα_2' κ.ο.κ.). Η προεπιλεγμένη τιμή είναι ένας κενός πίνακας ([]) και το πεδίο είναι υποχρεωτικό (δεν μπορεί να είναι null, αλλά μπορεί να είναι κενός πίνακας).

Είναι τυπικό στις διαλέκτους SQL τα ονόματα των πινάκων, καθώς και τα ονόματα των στηλών ενός πίνακα, να ονομάζονται με τη μέθοδο `lower_snake_case`, δηλαδή με τη χρήση πεζών χαρακτήρων και τον διαχωρισμό των λέξεων με τον χαρακτήρα της κάτω παύλας “_”. Η σύμβαση αυτή ακολουθήθηκε και στην παρούσα υλοποίηση για τη διασφάλιση της συνέπειας και της βέλτιστης αναγνωσιμότητας του σχήματος της βάσης δεδομένων.

Οι πίνακες που δημιουργήθηκαν είναι οι εξής:

1. **caregiver (Νοσηλευτικό προσωπικό)**

- **id**
- **name**
- **created_at**
- **updated_at**
- **location**
- **skills**
- **prefers_nights**: Μια λογική τιμή (boolean - true ή false) που υποδεικνύει εάν υπάρχει προτίμηση για νυχτερινή εργασία. Η προεπιλεγμένη τιμή είναι false και το πεδίο είναι υποχρεωτικό.
- **prefers_weekends**: Μια λογική τιμή (boolean - true ή false) που υποδεικνύει εάν υπάρχει προτίμηση για εργασία τα σαββατοκύριακα. Η προεπιλεγμένη τιμή είναι false και το πεδίο είναι υποχρεωτικό.

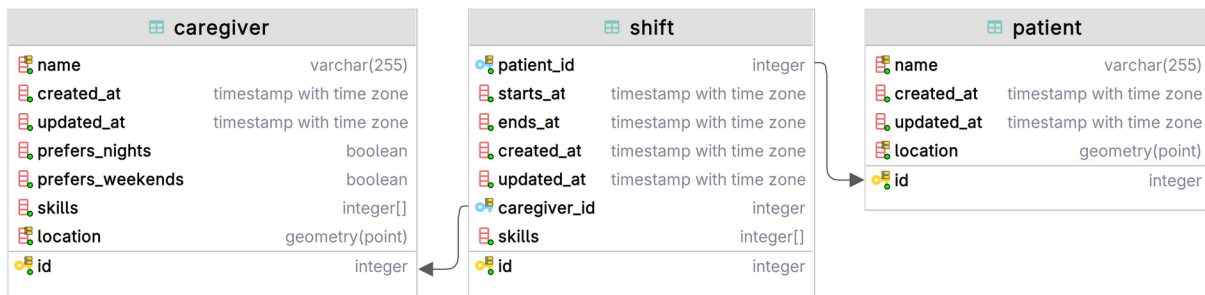
2. **patient (Ασθενείς)**

- **id**
- **name**
- **created_at**
- **updated_at**
- **location**

3. **shift (Βάρδιες)**

- **id**
- **created_at**
- **updated_at**
- **skills**
- **starts_at**: Μια χρονοσφραγίδα που περιλαμβάνει και τη ζώνη ώρας, η οποία δηλώνει την ακριβή ημερομηνία και ώρα έναρξης (π.χ. μιας βάρδιας). Το πεδίο είναι υποχρεωτικό (δεν μπορεί να είναι κενό).
- **ends_at**: Μια χρονοσφραγίδα που περιλαμβάνει και τη ζώνη ώρας, η οποία δηλώνει την ακριβή ημερομηνία και ώρα λήξης (π.χ. της ίδιας βάρδιας). Το πεδίο είναι υποχρεωτικό (δεν μπορεί να είναι κενό).

- **patient_id:** Ένας ακέραιος αριθμός - αναγνωριστικό. Αυτό το πεδίο είναι υποχρεωτικό και πρέπει να αντιστοιχεί σε μια έγκυρη εγγραφή στον πίνακα patient (λειτουργεί ως ξένο κλειδί - foreign key - με όνομα περιορισμού shift_patient_id_patient_id_fk). Ένα patient μπορεί να έχει πολλά shift (σχέση ένα-προς-πολλά).
- **caregiver_id:** Ένας ακέραιος αριθμός - αναγνωριστικό. Αυτό το πεδίο δεν είναι υποχρεωτικό (μπορεί να είναι κενό - null), αλλά αν περιέχει τιμή, αν δηλαδή έχει γίνει ανάθεση της βάρδιας, αυτή πρέπει να αντιστοιχεί σε μια έγκυρη εγγραφή στον πίνακα caregiver (λειτουργεί ως ξένο κλειδί - foreign key - με όνομα περιορισμού shift_caregiver_id_caregiver_id_fk). Ένα caregiver μπορεί να ανατεθεί σε πολλά shift (σχέση ένα-προς-πολλά).



Σχήμα 4.1: Διάγραμμα Οντοτήτων-Συσχετίσεων (ΟΣΔ) κύριων πινάκων

Η δομή αυτή επιλέχθηκε για να επιτρέπει την ευέλικτη καταγραφή των προτιμήσεων του νοσηλευτικού προσωπικού και την αποτελεσματική αντιστοίχιση με τις ανάγκες των ασθενών μέσω των βαρδιών.

4.2.2 Αυτόματα παραχθέντες πίνακες

Για τη λειτουργία της βιβλιοθήκης Auth.js, καθώς και για τη χρήση γεωχωρικών δεδομένων από την εφαρμογή, παρήχθησαν ακόμα οι πίνακες που περιγράφονται κατωτέρω.

Η βιβλιοθήκη Auth.js δημιούργησε αυτόματα στη βάση τους παρακάτω 4 πίνακες, με σκοπό την αποθήκευση όλων των απαραίτητων πληροφοριών σχετικά με την ταυτοποίηση χρηστών στην πλατφόρμα.

4. user (Χρήστης):

- Περιέχει πεδία όπως το όνομα του χρήστη, το email του και την εικόνα προφίλ του σε μορφή υπερσυνδέσμου.
- Αποθηκεύει τις βασικές, μοναδικές πληροφορίες για κάθε χρήστη της εφαρμογής (π.χ. id, όνομα, email, εικόνα προφίλ).
- Αποτελεί τον κεντρικό πίνακα αναφοράς για την ταυτοποίηση και τις πληροφορίες που σχετίζονται με τον χρήστη.

5. account (Λογαριασμός):

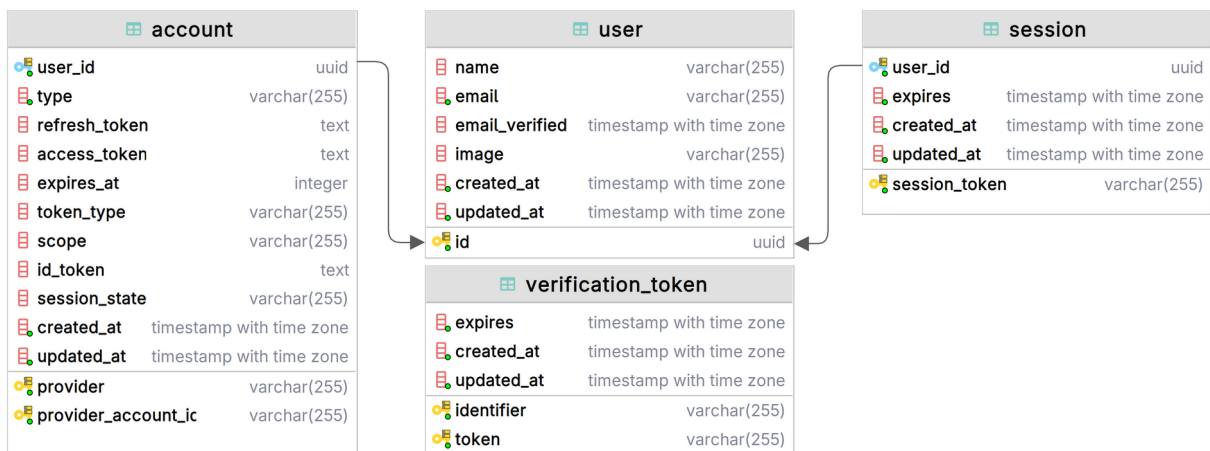
- Διασυνδέει τους λογαριασμούς χρηστών της εφαρμογής (του πίνακα user) με εξωτερικούς παρόχους ταυτότητας (π.χ. Google, GitHub, κλπ) για σύνδεση μέσω OAuth. Έτσι, ένας χρήστης μπορεί να έχει πάνω από έναν τρόπο να συνδεθεί στην εφαρμογή, εάν το επιθυμεί.
- Περιέχει τα απαραίτητα διακριτικά (tokens) και αναγνωριστικά (identifiers) για τη διαχείριση αυτών των εξωτερικών συνδέσεων.

6. session (Συνεδρία):

- Αποθηκεύει πληροφορίες για τις ενεργές συνεδρίες των συνδεδεμένων χρηστών, επιτρέποντάς τους να παραμένουν συνδεδεμένοι κατά την περιήγηση.
- Χρησιμοποιεί ένα μοναδικό διακριτικό συνεδρίας (session token) που συνδέεται με έναν χρήστη (μέσω του user id) και έχει ημερομηνία λήξης για λόγους ασφαλείας.

7. verification_token (Διακριτικό Επαλήθευσης):

- Αποθηκεύει προσωρινά, ασφαλή διακριτικά που χρησιμοποιούνται για την επαλήθευση ενεργειών όπως η επιβεβαίωση email ή η επαναφορά κωδικού πρόσβασης.
- Κάθε διακριτικό συνδέεται με ένα αναγνωριστικό (π.χ. email) και έχει συγκεκριμένη ημερομηνία λήξης για λόγους ασφαλείας.



Σχήμα 4.2: Διάγραμμα Οντοτήτων-Συσχετίσεων (ΟΣΔ) πινάκων της βιβλιοθήκης Auth.js

Τέλος, η επέκταση PostGIS της PostgreSQL, δημιούργησε έναν πίνακα στον οποίο βρίσκονται δεδομένα απαραίτητα ώστε η βάση να εκτελεί πράξεις και υπολογισμούς πάνω σε γεωχωρικά δεδομένα, καθώς και να τα μετατρέπει σε διαφορετικά μορφώτυπα (formats). Συγκεκριμένα:

8. spatial_ref_sys (Spatial Reference System - Χωρικό Σύστημα Αναφοράς):

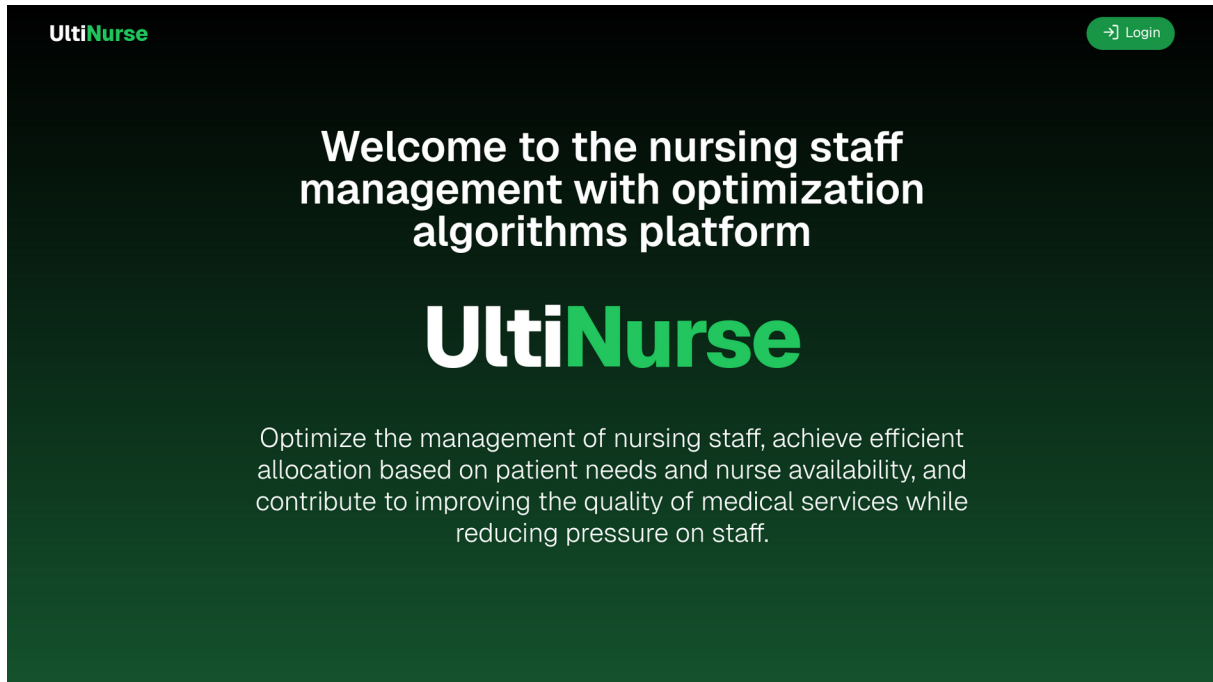
- Ορίζει διάφορα χωρικά συστήματα αναφοράς, όπως το σύστημα αναφοράς συντεταγμένων WGS 84 / SRID 4326, που χρησιμοποιείται από τη στήλη "location".
- Κάθε σύστημα αναφοράς περιγράφει πώς οι γεωγραφικές συντεταγμένες (π.χ. γεωγραφικό πλάτος/μήκος ή προβολικές συντεταγμένες) αντιστοιχούν σε θέσεις στην επιφάνεια της Γης.

spatial_ref_sys	
auth_name	varchar(256)
auth_srid	integer
srttext	varchar(2048)
proj4text	varchar(2048)
srid	integer

Σχήμα 4.3: Διάγραμμα Οντοτήτων-Συσχετίσεων (ΟΣΔ) πινάκων της επέκτασης PostGIS

4.3 Frontend

4.3.1 Αρχική οθόνη



Σχήμα 4.4: Αρχική οθόνη εφαρμογής σε πλήρες μέγεθος οθόνης

Στην αρχική οθόνη της εφαρμογής υπάρχει μια σύντομη περιγραφή στα αγγλικά:

«Βελτιστοποιήστε τη διαχείριση του νοσηλευτικού προσωπικού, επιτύχετε αποτελεσματική κατανομή με βάση τις ανάγκες των ασθενών και τη διαθεσιμότητα των νοσηλευτών και συμβάλλετε στη βελτίωση της ποιότητας των ιατρικών υπηρεσιών με ταυτόχρονη μείωση της πίεσης στο προσωπικό».

4.3.2 Σύνδεση

Για τη σύνδεση των χρηστών και τη διαχείριση συνεδριών (sessions), χρησιμοποιήθηκε η βιβλιοθήκη Auth.js (πρώην Next Auth), και πιο συγκεκριμένα η λειτουργία παρόχου αυθεντικοποίησης (authentication provider), με τον πάροχο της επιλογής μας να είναι η πλατφόρμα φιλοξενίας κώδικα βασισμένη στο σύστημα ελέγχου εκδόσεων “git”, το GitHub.



Σχήμα 4.5: Κουμπί σύνδεσης με GitHub

Η χρήση της «Σύνδεσης με GitHub» (GitHub login) στην ιστοσελίδα προσφέρει σημαντικά πλεονεκτήματα σε σχέση με την παραδοσιακή μέθοδο διατήρησης ενός πίνακα με κωδικούς στη βάση δεδομένων. Μερικά από τα κυριότερα θετικά είναι:

1. Ενισχυμένη Ασφάλεια:

- **Μεταφορά ευθύνης:** Η ευθύνη για την ασφαλή αποθήκευση και διαχείριση των κωδικών μεταφέρεται στο GitHub, μια πλατφόρμα με ισχυρούς μηχανισμούς ασφαλείας και εξειδικευμένες ομάδες για την προστασία δεδομένων. Αυτό μειώνει τον κίνδυνο παραβίασης των κωδικών στη δική μας βάση δεδομένων.
- **Πρότυπα OAuth 2.0:** Η σύνδεση μέσω GitHub συνήθως υλοποιείται με το πρωτόκολλο OAuth 2.0, ένα καθιερωμένο πρότυπο για ασφαλή εξουσιοδότηση. Αυτό διασφαλίζει ότι η εφαρμογή δεν αποκτά ποτέ απευθείας πρόσβαση στον κωδικό πρόσβασης του χρήστη στο GitHub.
- **Μείωση επιθέσεων:** Εξαλείφεται ο κίνδυνος επιθέσεων που στοχεύουν στην κλοπή κωδικών από τη βάση δεδομένων μας, όπως SQL injection ή άλλες ευπάθειες, καθώς και ο κίνδυνος διαρροής των κωδικών αυτών.
- **Passkeys:** Το GitHub υποστηρίζει πλέον passkeys, προσφέροντας μια εμπειρία σύνδεσης χωρίς κωδικό, ενισχύοντας περαιτέρω την ασφάλεια της όλης διαδικασίας.

2. Βελτιωμένη Εμπειρία Χρήστη (UX):

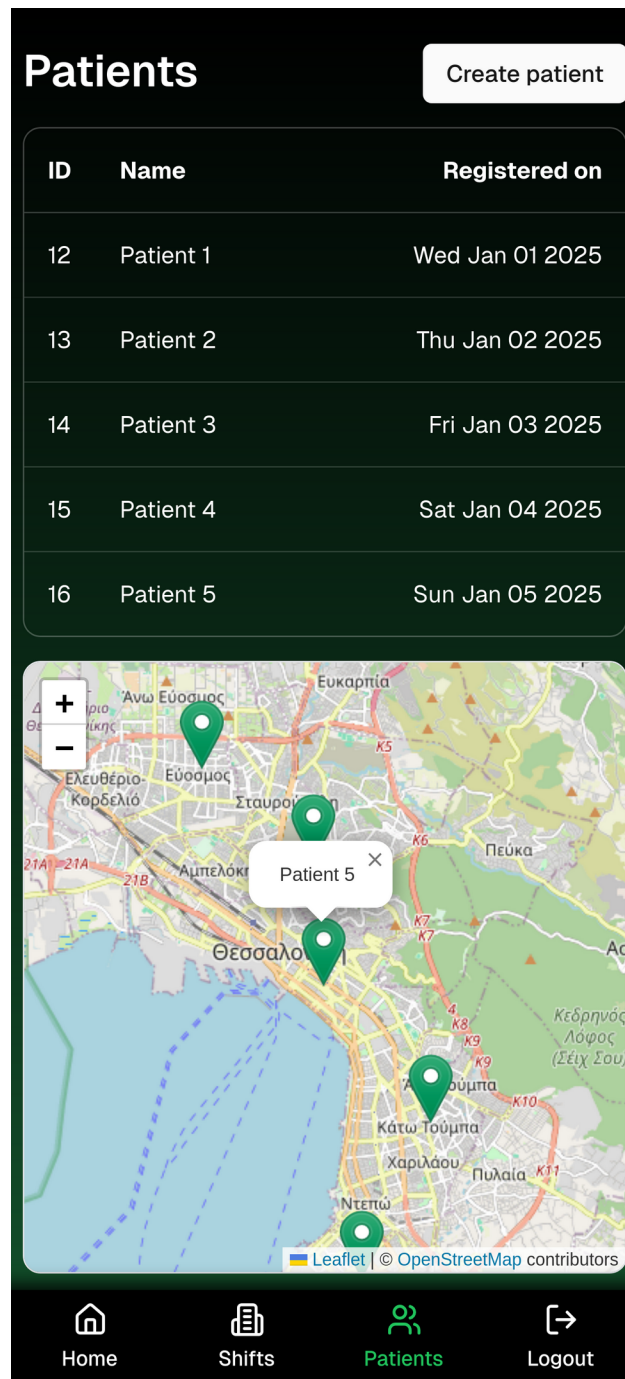
- **Ευκολία και ταχύτητα:** Οι χρήστες που διαθέτουν ήδη λογαριασμό στους επιλεγμένους παρόχους μπορούν να συνδεθούν στην ιστοσελίδα με ένα κλικ, χωρίς να χρειάζεται να δημιουργήσουν καινούριους λογαριασμούς και να θυμούνται νέους κωδικούς (ή ακόμα χειρότερα, να τους επαναχρησιμοποιούν μεταξύ ιστοσελίδων).
- **Λιγότερη τριβή:** Η διαδικασία εγγραφής και σύνδεσης γίνεται πιο απλή, μειώνοντας την πιθανότητα οι χρήστες να εγκαταλείψουν τη διαδικασία.
- **Εμπιστοσύνη:** Οι χρήστες συχνά αισθάνονται μεγαλύτερη εμπιστοσύνη όταν συνδέονται μέσω μιας γνωστής και αξιόπιστης πλατφόρμας, όπως το GitHub.

3. Μειωμένο Κόστος και Πολυπλοκότητα Ανάπτυξης:

- **Λιγότερος και απλούστερος κώδικας:** Δεν χρειάζεται να αναπτυχθεί και να συντηρηθεί πολύπλοκη λογική για τη διαχείριση κωδικών, την επαλήθευση email, την ανάκτηση κωδικού κ.λπ.
- **Εστίαση στον πυρήνα της εφαρμογής:** Οι προγραμματιστές μπορούν να επικεντρωθούν στην ανάπτυξη των βασικών λειτουργιών της ιστοσελίδας, αντί να ασχολούνται με θέματα αυθεντικοποίησης.

Παρ'όλο που η επιλογή παρόχου για την παρούσα πτυχιακή εργασία ήταν το GitHub, το Auth.js προσφέρει δεκάδες γνωστούς και έμπιστους παρόχους αυθεντικοποίησης, όπως Google, Apple, Facebook, κλπ.

4.3.3 Σελίδα ασθενών



Σχήμα 4.6: Σελίδα ασθενών σε κινητό τηλέφωνο

Στο επάνω μέρος της σελίδας “Patients” (Ασθενείς) υπάρχει ένας πίνακας που παραθέτει διάφορες πληροφορίες σχετικά με τους ασθενείς που είναι καταχωρημένοι στην πλατφόρμα. Οι στήλες του πίνακα είναι “ID” (μοναδικό αναγνωριστικό ενός ασθενή στη βάση δεδομένων), “Name” (Όνομα) και “Registered on” (Εγγράφηκε στις).

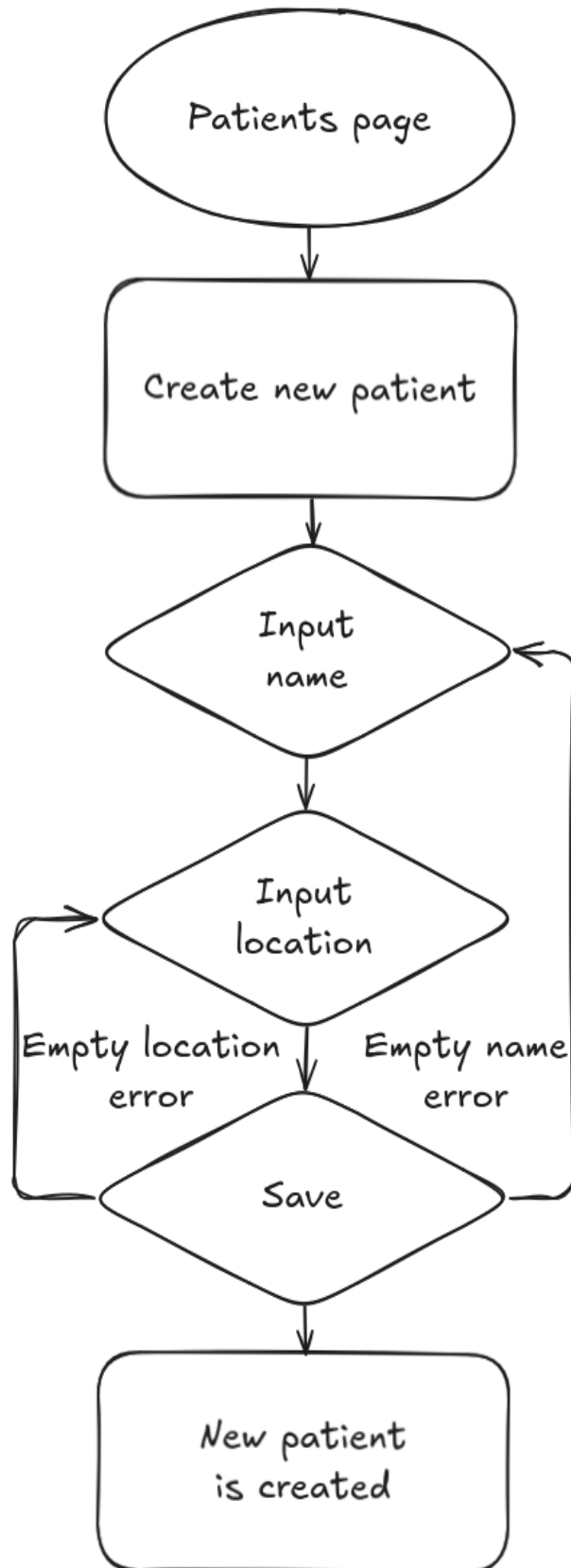
Πάνω από τον πίνακα, υπάρχει ένα κουμπί με το οποίο ο χρήστης μπορεί να δημιουργήσει έναν καινούριο ασθενή.

Κάτω από τον πίνακα των ασθενών, υπάρχει ένας γεωγραφικός χάρτης, βασισμένος στο OpenStreetMap, μια δωρεάν, ανοικτού τύπου βάση δεδομένων χαρτών, που ενημερώνεται και συντηρείται από εθελοντές, πάνω στον οποίο ο χρήστης μπορεί να δει την πραγματική τοποθεσία όλων των ασθενών της πλατφόρμας. Κάνοντας κλικ πάνω στους δείκτες του χάρτη (map markers) εμφανίζεται ένα αναδυόμενο παράθυρο (popup) με το όνομα του επιλεγμένου ασθενούς.

4.3.4 Δημιουργία ασθενούς

Σχήμα 4.7: Παράθυρο διαλόγου δημιουργίας νέου ασθενούς

Πατώντας το κουμπί δημιουργίας ασθενούς, εμφανίζεται ένα παράθυρο διαλόγου. Εδώ ο χρήστης μπορεί να συμπληρώσει το όνομα του ασθενούς, να επιλέξει στον χάρτη την τοποθεσία του, και να πατήσει Save (Αποθήκευση) για να καταχωριθεί ο νέος ασθενής στη βάση. Σε περίπτωση που κάποιο πεδίο παραμένει κενό, ένα κόκκινο κείμενο σφάλματος εμφανίζεται και ενημερώνει το χρήστη να το συμπληρώσει.



Σχήμα 4.8: Ροή δημιουργίας νέου ασθενούς

4.3.5 Σελίδα βαρδιών

ID	Patient	Caregiver	Night	Weekend	Starts At	Ends At
11	Patient 1	Dimitrios Papadopoulos ✕	✕	✕	2025-05-01 13:36	2025-05-01 18:36
12	Patient 2	Giorgos Konstantinidis ✕	✓	✓	2025-05-02 23:42	2025-05-03 00:42
35	Patient 3	+	✓	✕	2025-05-26 14:42	2025-05-26 19:42
36	Patient 3	+	✕	✓	2025-05-04 11:53	2025-05-04 15:53
37	Patient 4	+	✕	✓	2025-05-17 12:30	2025-05-17 14:30
38	Patient 5	+	✕	✕	2025-05-13 12:58	2025-05-13 17:58
39	Patient 3	+	✕	✕	2025-05-20 12:55	2025-05-20 17:55

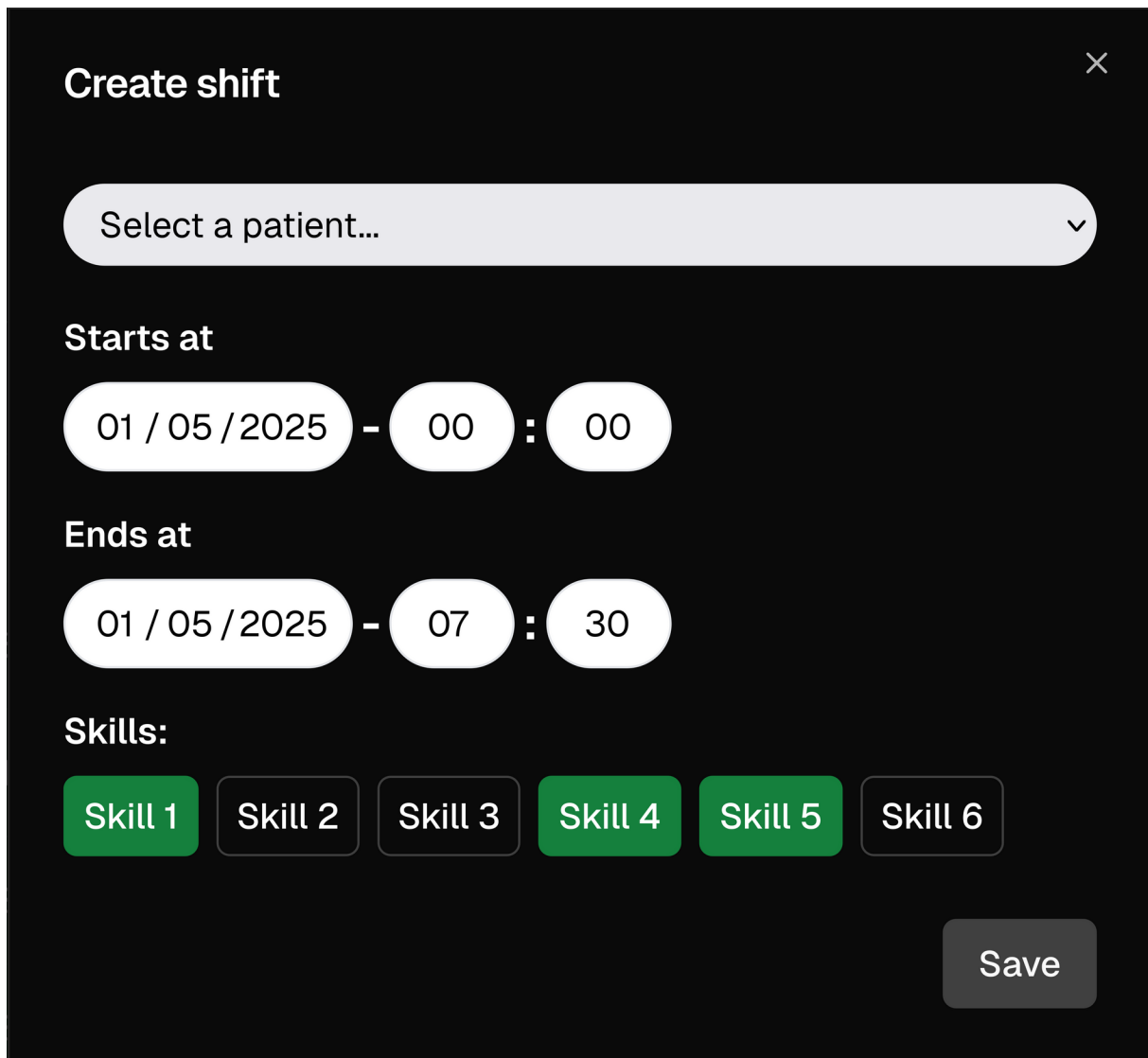
Σχήμα 4.9: Σελίδα βαρδιών σε τάμπλετ

Στο επάνω μέρος της σελίδας “Shifts” (Βάρδιες) υπάρχει ένας πίνακας που παραθέτει διάφορες πληροφορίες σχετικά με τις βάρδιες που είναι καταχωρημένες στην πλατφόρμα. Οι στήλες του πίνακα είναι:

- **ID:** Μοναδικό αναγνωριστικό μιας βάρδιας στη βάση δεδομένων.
- **Patient:** Ο ασθενής τον οποίο αφορά η βάρδια.
- **Caregiver:** Το μέλος του νοσηλευτικού προσωπικού που έχει ανατεθεί στη βάρδια. Μια ανάθεση μπορεί να προστεθεί με το πάτημα του εικονιδίου “+” και να αφαιρεθεί πατώντας το εικονίδιο “✕”.
- **Night:** Υποδεικνύει αν πρόκειται για νυχτερινή βάρδια, με εικονίδιο “✕” για όχι και τικ “✓” για ναι.
- **Weekend:** Υποδεικνύει αν πρόκειται για βάρδια Σαββατοκύριακου, με εικονίδιο “✕” για όχι και τικ “✓” για ναι.
- **Starts At:** Η ημερομηνία και ώρα έναρξης της βάρδιας.
- **Ends At:** Η ημερομηνία και ώρα λήξης της βάρδιας.

Πάνω από τον πίνακα, υπάρχουν κουμπιά με τα οποία ο χρήστης μπορεί να δημιουργήσει μια καινούρια βάρδια, να αναθέσει μαζικά νοσηλευτικό προσωπικό σε βάρδιες, καθώς και να αφαιρέσει όλες τις αναθέσεις από τις υπάρχουσες βάρδιες.

4.3.6 Δημιουργία βάρδιας



Create shift ×

Select a patient... ▾

Starts at

01 / 05 / 2025 - 00 : 00

Ends at

01 / 05 / 2025 - 07 : 30

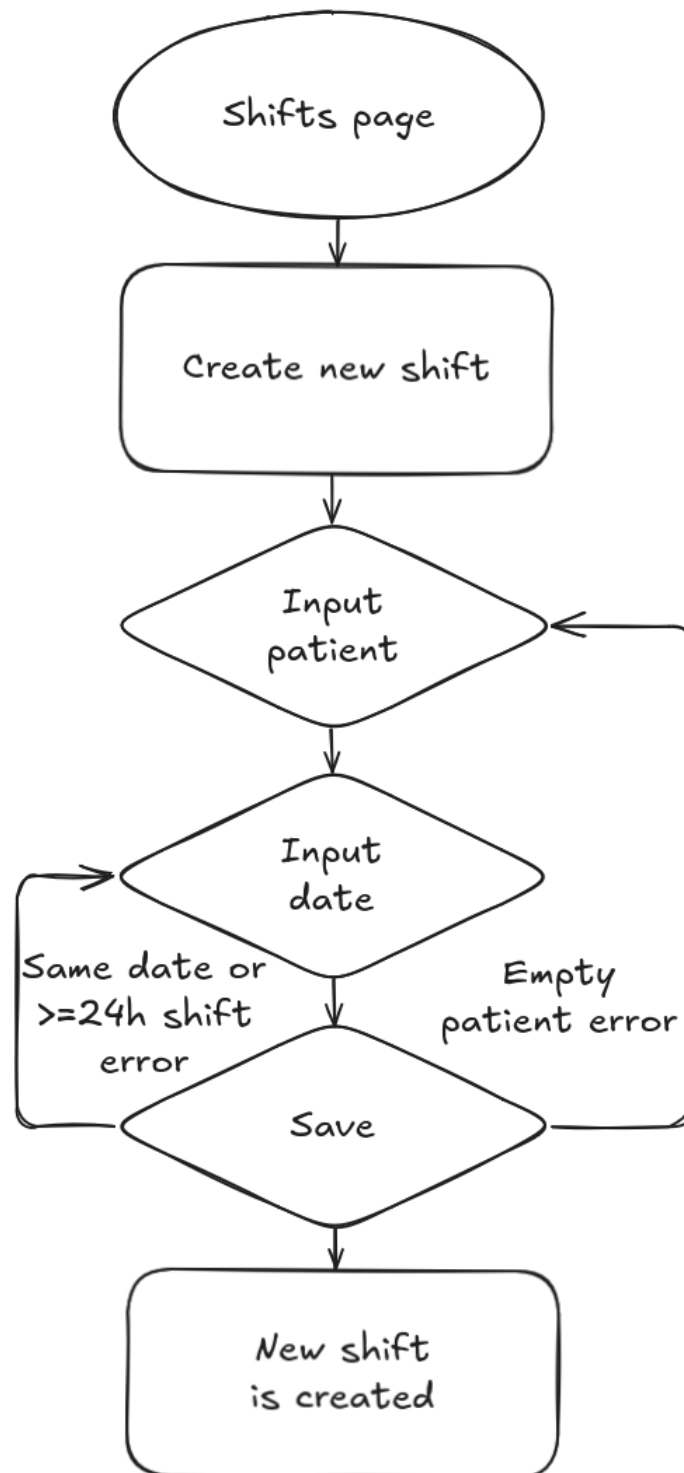
Skills:

Skill 1 Skill 2 Skill 3 Skill 4 Skill 5 Skill 6

Save

Σχήμα 4.10: Παράθυρο διαλόγου δημιουργίας νέας βάρδιας

Πατώντας το κουμπί δημιουργίας βάρδιας, εμφανίζεται ένα παράθυρο διαλόγου. Εδώ ο χρήστης μπορεί να επιλέξει τον ασθενή τον οποίο αφορά η βάρδια, να επιλέξει την ημερομηνία και ώρα που ξεκινά και τελειώνει η βάρδια, και να πατήσει Save (Αποθήκευση) ώστε να γίνει καταχώριση στη βάση. Στο πλαίσιο της πτυχιακής εργασίας, οι βάρδιες πρέπει να διαρκούν λιγότερο από 24 ώρες. Σε περίπτωση που δεν επιλεγεί ασθενής, η βάρδια έχει ίδια ημερομηνία και ώρα αρχής και τέλους ή βάρδια διαρκεί περισσότερο από το επιτρεπτό, ένα κόκκινο κείμενο σφάλματος εμφανίζεται και ενημερώνει το χρήστη να το συμπληρώσει.



Σχήμα 4.11: Ροή δημιουργίας νέας βάρδιας

4.3.7 Ανάθεση νοσηλευτικού προσωπικού σε βάρδια

Assign caregiver ×

Choose a caregiver to work this shift.

Algorithm:

WSM
TOPSIS
Greedy

Night Shift Weight: 1

Weekend Shift Weight: 1

Distance Weight: 1

Calculation took: 46μs

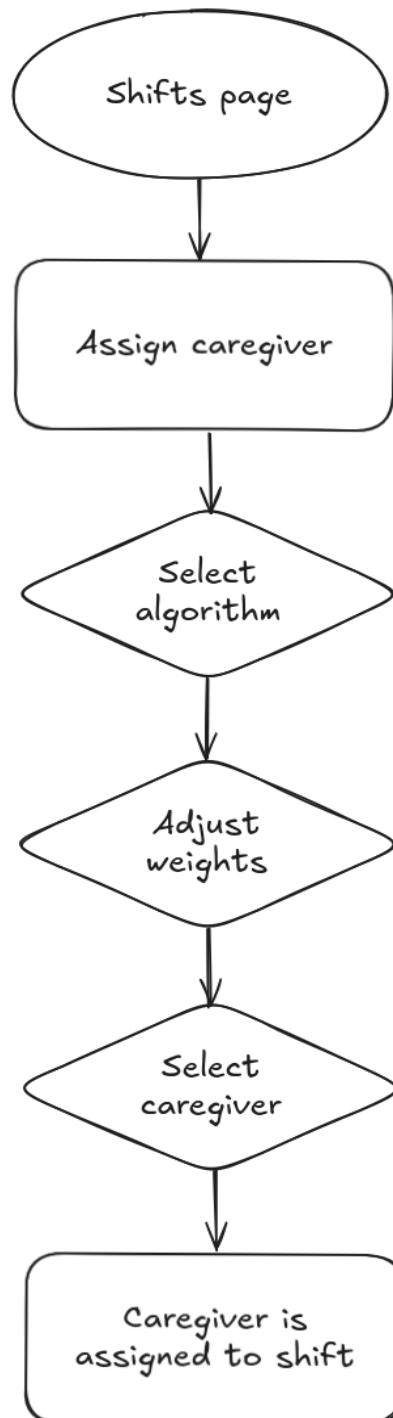
1. Maria Vasileiadou	100.0%
Distance: 2.6 Km	<input checked="" type="checkbox"/> Meets all needs <input checked="" type="checkbox"/> Distance <input checked="" type="checkbox"/> Night <input checked="" type="checkbox"/> Weekend
2. Eleni Karadimitriou	87.5%
Distance: 3.4 Km	<input checked="" type="checkbox"/> Meets all needs <input checked="" type="checkbox"/> Distance <input checked="" type="checkbox"/> Night <input checked="" type="checkbox"/> Weekend
3. Anastasia Ioannidou	61.3%
Distance: 5.5 Km	<input checked="" type="checkbox"/> Meets all needs <input checked="" type="checkbox"/> Distance <input checked="" type="checkbox"/> Night <input checked="" type="checkbox"/> Weekend
4. Katerina Alexiou	0.0%
Distance: 11.7 Km	<input checked="" type="checkbox"/> Meets all needs

Σχήμα 4.12: Παράθυρο διαλόγου ανάθεσης νοσηλευτικού προσωπικού σε βάρδια

Πατώντας το “+” σε οποιαδήποτε βάρδια, εμφανίζεται ένα παράθυρο διαλόγου. Εδώ ο χρήστης μπορεί να επιλέξει τον αλγόριθμο που θέλει να χρησιμοποιηθεί για τον υπολογισμό του ποσοστού καταλληλότητας, καθώς και το βάρος που θα δώσει ο αλγόριθμος σε κάθε παράμετρο. Αν για παράδειγμα επιλέξει να δώσει μεγαλύτερο βάρος στις νυκτερινές βάρδιες (Night Shifts) και η παρούσα βάρδια είναι νυκτερινή, τότε σε όσα μέλη του νοσηλευτικού προσωπικού προτιμούν να εργάζονται νυκτερινές βάρδιες θα αποδοθεί καλύτερη βαθμολογία (μεγαλύτερο δηλαδή ποσοστό) από τον

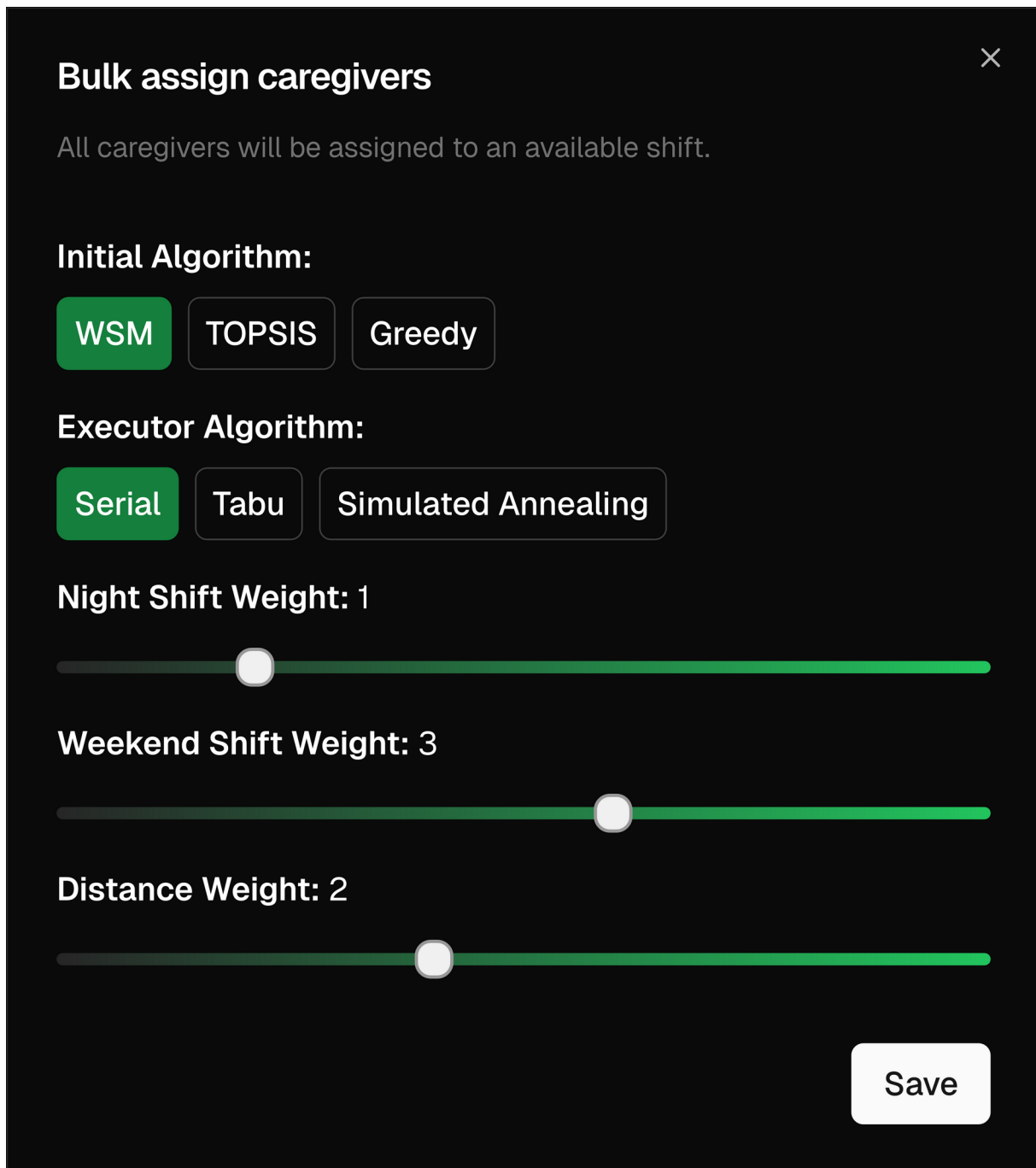
αλγόριθμο. Αντίστοιχα, εάν η βάρδια δεν είναι νυκτερινή, ένα μέλος του νοσηλευτικού προσωπικού θα έχει καλύτερη βαθμολογία αν έχει ως προτίμηση να μην εργάζεται νύκτες. Τέλος, μπορούμε να δούμε τον χρόνο που χρειάστηκε ο εκάστοτε αλγόριθμος για να βαθμολογήσει το νοσηλευτικό προσωπικό. Μετά από κάθε αλλαγή, γίνεται αυτόματα ανανέωση των διαθέσιμων επιλογών προσωπικού για τη βάρδια, χωρίς ο χρήστης να χρειάζεται να εκτελέσει κάποια άλλη ενέργεια.

Πατώντας πάνω σε μία από τις διαθέσιμες επιλογές προσωπικού, γίνεται ανάθεση του επιλεγμένου ατόμου στη βάρδια, το παράθυρο διαλόγου κλείνει, και ο πίνακας των βαρδιών ανανεώνεται αυτόματα.



Σχήμα 4.13: Ροή ανάθεσης νοσηλευτικού προσωπικού σε βάρδια

4.3.8 Μαζική ανάθεση νοσηλευτικού προσωπικού σε βάρδιες



Bulk assign caregivers ×

All caregivers will be assigned to an available shift.

Initial Algorithm:

WSM TOPSIS Greedy

Executor Algorithm:

Serial Tabu Simulated Annealing

Night Shift Weight: 1

Weekend Shift Weight: 3

Distance Weight: 2

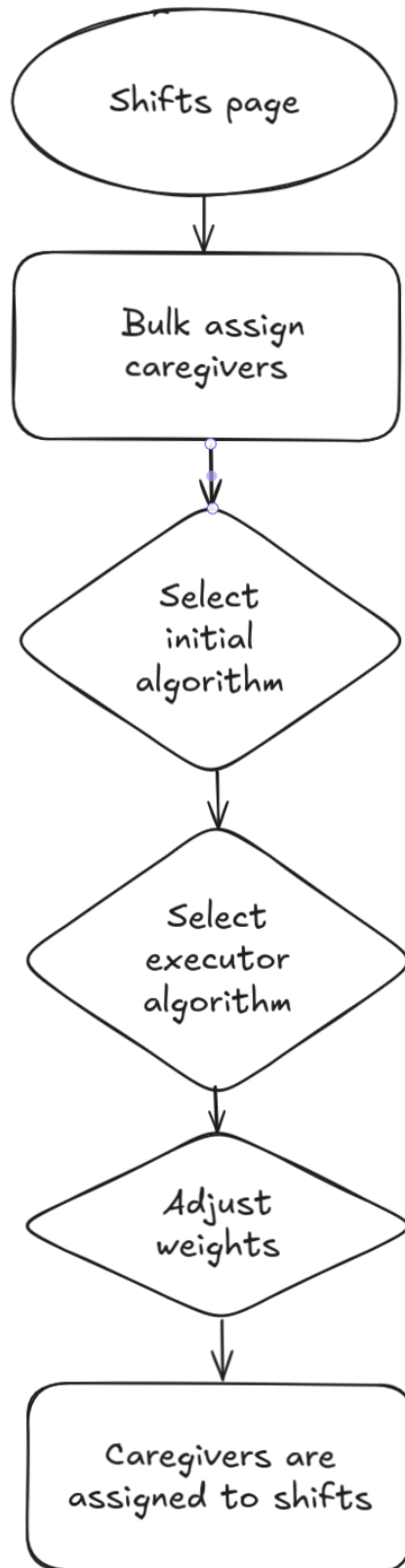
Save

Σχήμα 4.14: Παράθυρο διαλόγου μαζικής ανάθεσης νοσηλευτικού προσωπικού σε βάρδιες

Πατώντας το κουμπί “Bulk assign”, εμφανίζεται ένα παράθυρο διαλόγου για μαζική ανάθεση νοσηλευτικού προσωπικού σε βάρδιες. Εδώ ο χρήστης μπορεί να επιλέξει τον αλγόριθμο που θέλει να χρησιμοποιηθεί για τον υπολογισμό του ποσοστού καταλληλότητας, καθώς και το βάρος που θα δώσει ο αλγόριθμος σε κάθε παράμετρο. Η διαδικασία λειτουργεί με ακριβώς τον ίδιο τρόπο με το απλό παράθυρο διαλόγου ανάθεσης, με τη διαφορά ότι υπάρχει μια επιπρόσθετη επιλογή, αυτή του αλγορίθμου εκτέλεσης. Ο αλγόριθμος εκτέλεσης καθορίζει τον τρόπο με τον οποίο ο επιλεγμένος αλγόριθμος θα εκτελεστεί πάνω στις βάρδιες, με τον εξής τρόπο:

- Αν είναι επιλεγμένος ο Σειριακός (Serial) αλγόριθμος, θα ανακτηθούν οι διαθέσιμες βάρδιες και τα υπάρχοντα μέλη του νοσηλευτικού προσωπικού, και για κάθε βάρδια ξεχωριστά θα υπολογιστεί το πλέον κατάλληλο μέλος βάσει προκαθορισμένων κριτηρίων καταλληλότητας. Το μέλος αυτό θα ανατεθεί άμεσα στην αντίστοιχη βάρδια και στη συνέχεια θα αφαιρεθεί από τη λίστα των διαθέσιμων, ώστε να μη συμμετέχει στις επόμενες αναθέσεις. Η διαδικασία αυτή επαναλαμβάνεται διαδοχικά μέχρι να εξαντληθούν είτε οι βάρδιες είτε το προσωπικό, χωρίς αναθεώρηση προηγούμενων αναθέσεων.
- Αν είναι επιλεγμένος ο αλγόριθμος Ταμπού (Tabu), η διαδικασία ξεκινά με μια αρχική (τυχαία ή ευρετική) ανάθεση νοσηλευτικού προσωπικού σε βάρδιες, και στη συνέχεια εφαρμόζεται επαναληπτική βελτιστοποίηση. Σε κάθε επανάληψη, δοκιμάζονται εναλλακτικές μετακινήσεις (αναθέσεις) και αξιολογείται η νέα λύση βάσει συνολικού σκορ. Μετακινήσεις που έχουν ήδη εφαρμοστεί πρόσφατα καταγράφονται σε μια ταμπού λίστα και αποκλείονται προσωρινά από την επαναχρησιμοποίηση, ώστε να αποφευχθούν κύκλοι. Η διαδικασία συνεχίζεται για προκαθορισμένο αριθμό επαναλήψεων ή έως ότου δεν παρατηρείται βελτίωση, με στόχο τη σταδιακή σύγκλιση προς μία λύση υψηλής ποιότητας που υπερβαίνει τα όρια της τοπικής βελτιστοποίησης.
- Αν είναι επιλεγμένος ο αλγόριθμος της Προσομοιωμένης Ανόπτωσης (Simulated Annealing), το σύστημα ξεκινά με μια αρχική, ανάθεση νοσηλευτικού προσωπικού. Στη συνέχεια, διερευνά επαναληπτικά γειτονικές καταστάσεις ανάθεσης, πραγματοποιώντας μικρές τροποποιήσεις στην τρέχουσα λύση (π.χ., ανταλλάσσοντας βάρδιες μεταξύ δύο νοσηλευτών ή αναθέτοντας μια βάρδια σε διαφορετικό νοσηλευτή). Κάθε νέα διαμόρφωση αξιολογείται βάσει ενός συνολικού σκορ καταλληλότητας, το οποίο ο αλγόριθμος προσπαθεί να μεγιστοποιήσει. Οι βελτιωμένες λύσεις γίνονται πάντα αποδεκτές. Ωστόσο, για να αποφευχθεί η παγίδευση σε τοπικά βέλτιστα, ο αλγόριθμος επιτρέπει και την αποδοχή χειρότερων λύσεων με μια ορισμένη πιθανότητα, η οποία μειώνεται σταδιακά καθώς η «θερμοκρασία» του συστήματος «ψύχεται». Αυτή η διαδικασία συνεχίζεται, λαμβάνοντας πάντα υπόψη τους περιορισμούς αποκλειστικότητας (κάθε μέλος μπορεί να ανατεθεί σε μία μόνο βάρδια), μέχρι να επιτευχθεί μια ικανοποιητική λύση ή να ολοκληρωθεί ο προκαθορισμένος αριθμός επαναλήψεων, με στόχο την πλήρη κάλυψη των βαρδιών με τις καλύτερες δυνατές αντιστοιχίσεις.

Πατώντας το κουμπί αποθήκευσης, η διαδικασία μαζικής ανάθεσης εκτελείται σύμφωνα με τις επιλεγμένες ρυθμίσεις, το παράθυρο διαλόγου κλείνει, και ο πίνακας των βαρδιών ανανεώνεται αυτόματα.



Σχήμα 4.15: Ροή μαζικής ανάθεσης νοσηλευτικού προσωπικού σε βάρδιες

4.4 Επίλογος

Κλείνοντας την αναλυτική παρουσίαση των τεχνικών πτυχών της εφαρμογής, όπως αυτές αναπτύχθηκαν στο παρόν κεφάλαιο, διαμορφώνεται μια σαφής εικόνα της αρχιτεκτονικής που υποστηρίζει τη λειτουργικότητά της. Εξετάστηκε διεξοδικά η δομή της βάσης δεδομένων, όπου οι τρεις κεντρικοί πίνακες – caregiver, patient και shift – σχεδιάστηκαν με γνώμονα την ακριβή αποθήκευση και συσχέτιση των κρίσιμων πληροφοριών που αφορούν το νοσηλευτικό προσωπικό, τους ασθενείς και τις αντίστοιχες βάρδιες. Η επιλογή των συγκεκριμένων πεδίων και των τύπων δεδομένων τους έγινε με στόχο την κάλυψη των λειτουργικών απαιτήσεων της εφαρμογής. Παράλληλα, αναλύθηκε ο σημαντικός ρόλος των πέντε επιπλέον πινάκων, οι οποίοι παρήχθησαν αυτόματα από την ενσωμάτωση εξωτερικών βιβλιοθηκών. Αυτοί οι πίνακες είναι θεμελιώδεις για την υλοποίηση μηχανισμών ασφαλούς αυθεντικοποίησης των χρηστών και για την υποστήριξη της αποθήκευσης και επεξεργασίας γεωχωρικών δεδομένων, εμπλουτίζοντας σημαντικά τις δυνατότητες της βάσης.

Στη συνέχεια, η ανάλυση επικεντρώθηκε στο περιβάλλον χρήστη, όπου περιγράφηκαν αναλυτικά οι οθόνες, οι υπο-οθόνες και οι ροές εργασίας που συνθέτουν την εμπειρία πλοήγησης και αλληλεπίδρασης. Από την αρχική σελίδα και την ασφαλή διαδικασία σύνδεσης μέσω GitHub, έως τις εξειδικευμένες διεπαφές για τη διαχείριση των ασθενών και των βαρδιών, δόθηκε έμφαση στον τρόπο με τον οποίο ο χρήστης μπορεί να εκτελεί τις διάφορες εργασίες που είναι διαθέσιμες στην πλατφόρμα. Ειδικότερα, εξετάστηκαν οι μηχανισμοί για τη δημιουργία νέων εγγραφών ασθενών και βαρδιών, καθώς και η λειτουργία ανάθεσης νοσηλευτικού προσωπικού, όπου ο χρήστης μπορεί να επηρεάσει τη διαδικασία επιλογής μέσω παραμετροποίησης του επιλεγμένου αλγορίθμου.

Συμπερασματικά, η στιβαρή αρχιτεκτονική των δεδομένων στο backend παρέχει την απαραίτητη υποδομή για την αξιόπιστη λειτουργία της εφαρμογής, ενώ το διαισθητικό και λειτουργικό περιβάλλον του frontend καθιστά τις δυνατότητες του συστήματος προσβάσιμες και εύχρηστες. Η συνέργεια αυτών των δύο πυλώνων, όπως αναλύθηκε εκτενώς, είναι καθοριστική για την επίτευξη των στόχων της εφαρμογής. Ειδικότερα, η δυνατότητα παραμετροποίησης και εφαρμογής του αλγορίθμου αντιστοίχισης προσωπικού σε βάρδιες αποτελεί τον πυρήνα της βελτιστοποίησης στη διαχείριση του νοσηλευτικού δυναμικού, καθώς επιτρέπει την πιο αποτελεσματική κατανομή εργασίας λαμβάνοντας υπόψη πολλαπλές παραμέτρους όπως οι προτιμήσεις και η απόσταση. Αυτή η στοχευμένη προσέγγιση συμβάλλει καθοριστικά στην επίτευξη της βέλτιστης διαχείρισης του προσωπικού και, τελικά, στη βελτίωση της ποιότητας των παρεχόμενων υπηρεσιών.

Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης

5.1 Συμπεράσματα

Ο πίνακας 5.1 συγκρίνει τρεις αλγόριθμους από τους αλγόριθμους που αναλύθηκαν στο κεφάλαιο 4 -Greedy, TOPSIS και WSM- ως προς την ικανότητά τους να καλύπτουν δεξιότητες και τον χρόνο εκτέλεσης τους, σε ένα σενάριο που κάθε βάρδια αξιολογήθηκε σειριακά, για συνολικά 100 μέλη νοσηλευτικού προσωπικού (caregivers) και 100 βάρδιες (shifts).

Ο αλγόριθμος Greedy παρουσιάζει περιορισμένη απόδοση, με ποσοστά πλήρους κάλυψης δεξιοτήτων περίπου ~15% και κάλυψης τουλάχιστον μίας δεξιότητας ~33%. Ωστόσο, ο χρόνος εκτέλεσης του είναι εξαιρετικά χαμηλός, γεγονός που τον καθιστά κατάλληλο για περιπτώσεις όπου η ταχύτητα υπερισχύει της ακρίβειας.

Αντίθετα, οι αλγόριθμοι TOPSIS και WSM προσφέρουν σημαντικά υψηλότερα ποσοστά κάλυψης. Ο TOPSIS επιτυγχάνει πλήρη κάλυψη δεξιοτήτων ~60% και κάλυψη τουλάχιστον μίας δεξιότητας ~85%, με χρόνο εκτέλεσης 80 ms (millisecond - χιλιοστά του δευτερολέπτου). Ο WSM υπερτερεί ελαφρώς, με ~68% πλήρη κάλυψη και ~92% κάλυψη τουλάχιστον μίας δεξιότητας, ενώ διατηρεί πολύ χαμηλό χρόνο εκτέλεσης (15 ms), γεγονός που αντανακλά την αποδοτικότητα της μεθόδου στο συγκεκριμένο πλαίσιο.

Συνολικά, τα αποτελέσματα υποδεικνύουν ότι ο απλούστερος αλγόριθμος (Greedy) παρέχει ταχύτητα με περιορισμένη ακρίβεια, ενώ οι πιο σύνθετοι αλγόριθμοι (TOPSIS και WSM) προσφέρουν ανώτερη κάλυψη δεξιοτήτων. Η υπεροχή του WSM έναντι του TOPSIS φαίνεται να οφείλεται στη συγκεκριμένη φύση της εφαρμογής, όπου οι γραμμικά συνδυασμένες βαθμολογίες επαρκούν για να επιτύχουν υψηλή αποδοτικότητα, χωρίς την ανάγκη για πολύπλοκους υπολογισμούς αποστάσεων όπως στον TOPSIS.

Πίνακας 5.1: Συγκριτική αξιολόγηση αλγορίθμων με σειριακό αλγόριθμο εκτέλεσης

Αλγόριθμος αρχικοποίησης	Ποσοστό που πληρεί όλες τις δεξιότητες	Ποσοστό που πληρεί τουλάχιστον μία δεξιότητα	Χρόνος εκτέλεσης αλγορίθμου
Τυχαία Ανάθεση	~15%	~33%	1 ms
Greedy	~15%	~33%	11 ms
TOPSIS	~60%	~85%	80 ms
WSM	~68%	~92%	15 ms

Ο πίνακας 5.2 παρουσιάζει τα τελικά αποτελέσματα που προέκυψαν μετά την εφαρμογή του αλγορίθμου βελτιστοποίησης Tabu Search πάνω στην αρχική λύση που δημιουργήθηκε από τους αλγορίθμους που αναφερθηκαν παραπάνω, για συνολικά 100 μέλη νοσηλευτικού προσωπικού (caregivers) και 100 βάρδιες (shifts). Παρά τις διαφορές στις στρατηγικές αρχικής κατανομής, η βελτιστοποίηση μέσω του Tabu Search οδήγησε όλες τις μεθόδους σε παρόμοια επίπεδα απόδοσης. Συγκεκριμένα, όλοι οι αλγόριθμοι πέτυχαν ποσοστό κάλυψης ~97% για όλες τις απαιτούμενες δεξιότητες και ~100% για τουλάχιστον μία δεξιότητα ανά άτομο.

Πίνακας 5.2: Συγκριτική αξιολόγηση αλγορίθμων με αλγόριθμο εκτέλεσης Tabu

Αλγόριθμος αρχικοποίησης	Ποσοστό που πληρεί όλες τις δεξιότητες	Ποσοστό που πληρεί τουλάχιστον μία δεξιότητα	Χρόνος εκτέλεσης αλγορίθμου
Τυχαία Ανάθεση	~97%	~100%	33000 ms
Greedy	~97%	~100%	33000 ms
TOPSIS	~97%	~100%	35600 ms
WSM	~97%	~100%	34000 ms

Ο χρόνος εκτέλεσης παρουσίασε μικρές διαφοροποιήσεις, με τον αλγόριθμο TOPSIS να εμφανίζει τον υψηλότερο χρόνο στα 35.600 ms, ο αλγόριθμος WSM 34.000 ms, ενώ η Τυχαία Ανάθεση και ο Greedy ήταν οι ταχύτεροι με 33.000 ms. Τα αποτελέσματα αυτά υποδεικνύουν ότι, ανεξαρτήτως της αρχικής μεθόδου που χρησιμοποιείται για τη δημιουργία της αρχικής λύσης, ο Tabu Search είναι ιδιαίτερα αποτελεσματικός στη βελτιστοποίηση της ποιότητας της λύσης, οδηγώντας σε υψηλά και συγκλινόμενα επίπεδα απόδοσης. Αυτό ενισχύει την αξία της χρήσης μεθόδων μετα-ευρετικής βελτιστοποίησης, καθώς περιορίζει την εξάρτηση από την ποιότητα της αρχικής λύσης.

Ο πίνακας 5.3 παρουσιάζει τα τελικά αποτελέσματα που προέκυψαν από την εφαρμογή του αλγορίθμου Simulated Annealing σε αρχικές λύσεις που δημιουργήθηκαν μέσω των τριών αλγορίθμων που αναφέρθηκαν παραπάνω, για συνολικά 100 μέλη νοσηλευτικού προσωπικού (caregivers) και 100 βάρδιες (shifts). Σε αντίθεση με τα αποτελέσματα που παρατηρήθηκαν με τη χρήση του Tabu Search, η εφαρμογή του Simulated Annealing οδήγησε σε χαμηλότερα ποσοστά κάλυψης δεξιοτήτων.

Συγκεκριμένα, όλοι οι αλγόριθμοι κατέληξαν σε λύσεις με ~82% κάλυψη όλων των απαιτούμενων δεξιοτήτων και ~97% κάλυψη τουλάχιστον μίας δεξιότητας ανά άτομο. Οι χρόνοι εκτέλεσης κυμαίνονταν από 3.600 ms (Τυχαία Ανάθεση) έως 5.600 ms (TOPSIS), με τους Greedy και WSM να εντοπίζονται ενδιάμεσα με 4.600 ms και 5.000 ms αντίστοιχα.

Πίνακας 5.3: Συγκριτική αξιολόγηση αλγορίθμων με αλγόριθμο εκτέλεσης Simulated Annealing

Αλγόριθμος αρχικοποίησης	Ποσοστό που πληρεί όλες τις δεξιότητες	Ποσοστό που πληρεί τουλάχιστον μία δεξιότητα	Χρόνος εκτέλεσης αλγορίθμου
Τυχαία Ανάθεση	~82%	~97%	3600 ms
Greedy	~82%	~97%	4600 ms
TOPSIS	~82%	~97%	5600 ms
WSM	~82%	~97%	5000 ms

Τα αποτελέσματα αυτά υποδηλώνουν ότι, ενώ ο Simulated Annealing βελτιώνει την ποιότητα των αρχικών λύσεων, η συνολική του ικανότητα να προσεγγίσει υψηλά επίπεδα βελτιστοποίησης είναι περιορισμένη σε σχέση με τον Tabu Search. Παρά την ομοιογένεια στα τελικά ποσοστά απόδοσης μεταξύ των μεθόδων αρχικοποίησης, το γενικό επίπεδο ποιότητας παραμένει χαμηλότερο, γεγονός που αναδεικνύει τη σημασία της επιλογής κατάλληλης μετα-ευρετικής στρατηγικής στη διαδικασία βελτιστοποίησης.

Στον πίνακα 5.4 παρουσιάζεται η μεταβολή τριών βασικών μετρικών απόδοσης σε σχέση με την αύξηση του βάρους που δίνεται στις προτιμήσεις των εργαζομένων, στο πλαίσιο κατανομής βαρδιών μέσω WSM και σειριακού αλγορίθμου εκτέλεσης. Συγκεκριμένα, παρατηρείται ότι όσο αυξάνεται το βάρος (από 0 έως 5), αυξάνονται σημαντικά τα ποσοστά ικανοποίησης στις προτιμήσεις για βάρδιες νύχτας ή σαββατοκύριακα (~60% → ~83%) και, ακόμη πιο έντονα, στις περιπτώσεις όπου και οι δύο προτιμήσεις συνυπάρχουν (~31% → ~78%).

Ωστόσο, η βελτίωση αυτή συνοδεύεται από μία σταδιακή μείωση στο ποσοστό κάλυψης των αναγκών του συστήματος (~71% → ~62%). Το αποτέλεσμα αυτό αναδεικνύει τον αναμενόμενο συμβιβασμό μεταξύ ικανοποίησης των προτιμήσεων των εργαζομένων και λειτουργικής επάρκειας του συστήματος. Όσο περισσότερο ενισχύονται οι ατομικές προτιμήσεις μέσω των βαρών, τόσο περισσότερο περιορίζεται η δυνατότητα του συστήματος να καλύψει πλήρως τις επιχειρησιακές του ανάγκες.

Πίνακας 5.4: Συγκριτική αξιολόγηση βαρών με WSM και σειριακό αλγόριθμο εκτέλεσης

Βάρος	Ποσοστό ικανοποίησης στην προτίμηση νύχτας / σαββατοκύριακου	Ποσοστό ικανοποίησης στην προτίμηση και των δύο ταυτόχρονα	Ποσοστό που πληρεί όλες τις δεξιότητες
0	~60%	~31%	~71%
1	~66%	~37%	~68%
2	~75%	~59%	~67%
3	~78%	~67%	~65%
4	~80%	~74%	~63%
5	~83%	~78%	~62%

5.2 Προτάσεις βελτίωσης

- **Ακριβέστερες αποστάσεις:** Χρήση κάποιας υπηρεσίας υπολογισμού αποστάσεων μέσω πλοήγησης, αντί για υπολογισμού της απόστασης σε ευθεία γραμμή λαμβάνοντας υπόψη την καμπυλότητα της γης.
- **Περισσότεροι αλγόριθμοι:** Λόγω του σχεδιασμού της πλατφόρμας, μπορούν εύκολα να προστεθούν περισσότεροι αλγόριθμοι, που να βελτιστοποιούν τις αναθέσεις με περαιτέρω κριτήρια.
- **Περισσότερες επιλογές παραμετροποίησης:** Με την προσθήκη περισσότερων βαρών, όπως πχ. οικειότητα με ασθενή (προτίμηση νοσηλευτικού προσωπικού που έχει ήδη προηγούμενες βάρδιες με έναν ασθενή), οι οργανισμοί υγείας θα μπορούσαν να παρέχουν μια ακόμα πιο υψηλής ποιότητας υπηρεσία στους ασθενείς.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] World Health Organization: WHO, “Nursing and midwifery,” May 03, 2024. Accessed: March 01, 2025. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/nursing-and-midwifery>
- [2] “Nursing and Midwifery in Crisis | JBI.” Accessed: March 01, 2025. [Online]. Available: <https://jbi.global/news/article/nursing-and-midwifery-crisis>
- [3] Π.Ο.Ε.ΔΗ.Ν. (Πανελλήνια Ομοσπονδία Εργαζομένων Δημοσίων Νοσοκομείων): «Απόφαση Γενικού Συμβουλίου 17 Ιανουαρίου 2025,» Accessed: March 01, 2025. [Online]. Available: <https://www.poedhn.gr/apofaseis/item/8535-apofasi-genikoy-symvoulou-17-ianouariou-2025>
- [4] S. Holton *et al.*, “Developing nurse and midwife centred rostering principles using co-design: a mixed-methods study,” *BMC Nursing*, vol. 23, no. 1, Dec. 2024, doi: 10.1186/s12912-024-02522-7.
- [5] J. Turunen *et al.*, “The effects of using participatory working time scheduling software on sickness absence: A difference-in-differences study,” *International Journal of Nursing Studies*, vol. 112, p. 103716, Jul. 2020, doi: 10.1016/j.ijnurstu.2020.103716.
- [6] PostgreSQL. Accessed: Mar. 01, 2025. [Online]. Available: <https://www.postgresql.org>
- [7] PostGIS. Accessed: Mar. 01, 2025. [Online]. Available: <https://postgis.net>
- [8] Neon. Accessed: Mar. 01, 2025. [Online]. Available: <https://neon.tech>
- [9] TypeScript. Accessed: Mar. 01, 2025. [Online]. Available: <https://www.typescriptlang.org>
- [10] Node.js. Accessed: Mar. 01, 2025. [Online]. Available: <https://nodejs.org>
- [11] Drizzle ORM. Accessed: Mar. 01, 2025. [Online]. Available: <https://orm.drizzle.team>
- [12] tRPC. Accessed: Mar. 01, 2025. [Online]. Available: <https://trpc.io>
- [13] React. Accessed: Mar. 01, 2025. [Online]. Available: <https://react.dev>
- [14] Next.js. Accessed: Mar. 01, 2025. [Online]. Available: <https://nextjs.org>
- [15] Tailwind CSS. Accessed: Mar. 01, 2025. [Online]. Available: <https://tailwindcss.com>
- [16] A. Sarvade, “Analysis between Prim’s and Kruskal’s Algorithm,” *International Journal For Science Technology And Engineering*, vol. 12, no. 11, pp. 2464–2467, Nov. 2024, doi: 10.22214/ijraset.2024.65593.
- [17] H. Chen, “Greedy Methods in Plume Detection, Localization and Tracking,” *Greedy Algorithms*, W. Bednorz, Ed. Vienna, Austria: IntechOpen, 2008, pp. 305–324. doi: 10.5772/6094.
- [18] A. Malik, N. Goyat, and V. Saroha, “Greedy Algorithm: Huffman Algorithm,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 7, pp. 296–303, Jul. 2013.
- [19] G.-H. Tzeng and J.-J. Huang, *Multiple Attribute Decision Making: Methods and Applications*. Boca Raton, FL: CRC Press, 2011.
- [20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, May 1983, doi: 10.1126/science.220.4598.671.

- [21] J. Cho and Y. Kim, “A simulated annealing algorithm for resource constrained project scheduling problems,” *Journal of the Operational Research Society*, vol. 48, no. 7, pp. 736–744, Jul. 1997, doi: 10.1057/palgrave.jors.2600416.
- [22] F. Glover, “Tabu Search—Part i,” *INFORMS Journal on Computing*, vol. 1, no. 3, pp. 190–206, Aug. 1989, doi: 10.1287/ijoc.1.3.190.
- [23] F. Glover, “Tabu search: a tutorial,” *INFORMS Journal on Applied Analytics*, vol. 20, no. 4, pp. 74–94, Aug. 1990, doi: 10.1287/inte.20.4.74.
- [24] D. R. Ramdania, K. Manaf, F. R. Junaedi, A. Fathonih, and A. Hadiana, “TOPSIS Method on Selection of New Employees' Acceptance,” *Proc. 2020 6th International Conference on Wireless and Telematics (ICWT)*, Bali, Indonesia, Sep. 2020, pp. 1–5. doi: 10.1109/ICWT50448.2020.9243658.
- [25] M. Audina, “Design of a Decision Support System for Selecting Exemplary Health Officers with the TOPSIS Method,” *UPI YPTK Journal of Computer Science and Information Technology*, vol. 8, no. 4, pp. 119–124, Oct. 2022, doi: 10.35134/jcsitech.v8i4.49.
- [26] H. Gökkaya and T. Kellegöz, “Personel Tayin İşlemleri İçin AHP, TOPSIS ve Macar Algoritması Tabanlı Karar Destek Modeli,” *ODÜ Mühendislik Bilimleri Dergisi*, vol. 8, no. 1, pp. 2–18, May 2017. doi: 10.37093/ordumuh.300624.