

Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.



ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Θεμιστοκλής Έξαρχος,

Αναπληρωτής Καθηγητής, Ιόνιο Πανεπιστήμιο

ΦΟΙΤΗΤΗΣ: Γεώργιος Παπακωνσταντίνου

A/M: Π2013137

ΚΕΡΚΥΡΑ 2024

Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.

Τριμελής Επιτροπή

Θεμιστοκλής Έξαρχος
Αναπληρωτής Καθηγητής, Ιόνιο Πανεπιστήμιο

Παναγιώτης Βλάμος
Καθηγητής, Ιόνιο Πανεπιστήμιο

Αριστείδης Βραχάτης
Επίκουρος Καθηγητής, Ιόνιο Πανεπιστήμιο

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	4
ΚΕΦΑΛΑΙΟ 1	5
1.1 ΠΛΗΡΟΦΟΡΙΑΚΟ ΣΥΣΤΗΜΑ	5
1.2 ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΣΤΟΝ ΤΟΜΕΑ ΤΗΣ ΥΓΕΙΑΣ.....	8
1.3 ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ	24
1.4 ΧΡΗΣΙΜΟΤΗΤΑ ΕΡΩΤΗΜΑΤΟΛΟΓΙΩΝ ΣΤΟ ΥΓΕΙΟΝΟΜΙΚΟ ΣΥΣΤΗΜΑ	25
ΚΕΦΑΛΑΙΟ 2	28
2.1 DJANGO, WEB ΠΛΑΙΣΙΟ	28
2.2 ΡΥΘΜΟΝ, ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ	28
2.3 HYPERTEXT MARKUP LANGUAGE	29
2.4 CASCADING STYLE SHEETS	30
ΚΕΦΑΛΑΙΟ 3	31
3.1 ΕΦΑΡΜΟΓΗ	31
3.2 ΒΑΣΙΚΑ ΒΗΜΑΤΑ ΓΙΑ ΤΗ ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΝΕΟΥ DJANGO PROJECT ΚΑΙ ΕΝΟΣ DJANGO APP.	32
3.3 MODELS.PY	38
3.4 FORMS.PY	40
3.5 VIEWS.PY	40
3.6 ΣΥΝΑΡΤΗΣΕΙΣ ΔΙΑΧΕΙΡΙΣΗΣ ΧΡΗΣΤΩΝ	41
3.7 ΣΥΝΑΡΤΗΣΕΙΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΡΩΤΗΣΕΩΝ	46
3.8 URLS.PY	57
3.9 ADMIN.PY	58
ΚΕΦΑΛΑΙΟ 4	61
4.1 ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ : ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΑΣΘΕΝΩΝ	61
4.1.1 ΔΗΜΙΟΥΡΓΙΑ ΛΟΓΑΡΙΑΣΜΟΥ	61
4.1.2 ΣΥΝΔΕΣΗ ΣΤΟ ΣΥΣΤΗΜΑ	62
4.1.3 ΔΗΜΙΟΥΡΓΙΑ ΝΕΟΥ ΑΣΘΕΝΟΥΣ ΚΑΙ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ.....	63
4.1.4 ΠΡΟΒΟΛΗ ΛΙΣΤΑΣ ΑΣΘΕΝΩΝ.....	66
4.1.5 ΕΠΕΞΕΡΓΑΣΙΑ ΣΤΟΙΧΕΙΩΝ ΚΑΙ ΕΡΩΤΗΣΕΩΝ ΑΣΘΕΝΗ.....	68
4.2 ΔΟΚΙΜΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	70
ΣΥΜΠΕΡΑΣΜΑΤΑ ΑΠΟ ΤΗΝ ΧΡΗΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	76
ΒΙΒΛΙΟΓΡΑΦΙΑ	77

ΠΕΡΙΛΗΨΗ

Ανέπτυξα ένα online σύστημα πληροφοριών για γιατρούς, το οποίο αντιμετωπίζει τη διαχείριση της φροντίδας των ασθενών. Το σύστημα αυτό σχεδιάστηκε με σκοπό τη διευκόλυνση της άνετης καταγραφής και διαχείρισης των πληροφοριών των ασθενών, προσφέροντας ένα ολοκληρωμένο εργαλείο για τους ιατρούς να παρακολουθούν και να διαχειρίζονται τις περιπτώσεις των ασθενών τους.

Ένα από τα βασικά χαρακτηριστικά του συστήματος είναι η περιλαμβανόμενη δυνατότητα για τους ασθενείς να συμπληρώνουν ένα ενσωματωμένο ερωτηματολόγιο μέσω του γιατρού τους. Αυτό το ερωτηματολόγιο επιτρέπει στους ασθενείς να απαντήσουν σε σημαντικές ερωτήσεις που αφορούν την υγεία και την ευημερία τους, παρέχοντας στους γιατρούς πολύτιμες πληροφορίες για την κατάστασή τους.

Η συστηματική προσέγγιση που ακολουθείται με τη χρήση αυτού του συστήματος συμβάλλει στην εξατομίκευση των υπηρεσιών υγείας. Οι γιατροί έχουν τη δυνατότητα να προσφέρουν βελτιωμένη φροντίδα μέσω της παρακολούθησης και της ανάλυσης των δεδομένων των ασθενών τους, λαμβάνοντας υπόψη τις ατομικές τους ανάγκες και επιδόσεις.

Για την υλοποίηση αυτού του συστήματος, ανέλυσα προσεκτικά τις δυνατότητες και τα χαρακτηριστικά των εργαλείων που χρησιμοποίησα. Χρησιμοποίησα το Linux Ubuntu 22.04 ως λειτουργικό σύστημα για το περιβάλλον ανάπτυξης και τη διαχείριση πακέτων της Conda στην έκδοση 22.9.0. Επίσης, επέλεξα την έκδοση 3.10.6 της Python ως γλώσσα προγραμματισμού για την υλοποίηση της λογικής της εφαρμογής.

Για το web πλαίσιο, επέλεξα το Django στην έκδοση 4.1, το οποίο μου παρείχε ένα ισχυρό πλαίσιο εργασίας για τη διαχείριση της βάσης δεδομένων, τη δρομολόγηση URL και την ανάπτυξη εφαρμογών με στοχευμένες λειτουργίες.

Για τη διαμόρφωση και την εμφάνιση των σελίδων, χρησιμοποίησα τις γλώσσες HTML για τη διάταξη της σελίδας, CSS για τον σχεδιασμό και τη μορφοποίηση, και JavaScript για διαδραστικές λειτουργίες. Επιπλέον, ενσωμάτωσα το Bootstrap στην έκδοση 4.6 για τον σχεδιασμό του frontend της εφαρμογής, προσφέροντας ανταπόκριση και στυλ στη διεπαφή χρήστη.

Μέσω αυτών των εργαλείων και τεχνολογιών, κατάφερα να αναπτύξω ένα πλήρες και λειτουργικό σύστημα, το οποίο ενσωματώνει αποτελεσματικά την ευκολία χρήσης, την αξιοπιστία και τον σύγχρονο σχεδιασμό.

ΚΕΦΑΛΑΙΟ 1

1.1 Πληροφοριακό σύστημα

Στην επιστήμη των υπολογιστών, ένα πληροφοριακό σύστημα (information system) αναφέρεται σε ένα σύνολο συνδεδεμένων στοιχείων ή συνιστωσών που λειτουργούν από κοινού για τη συλλογή, αποθήκευση, επεξεργασία, διανομή και ανάκτηση πληροφοριών. Τα πληροφοριακά συστήματα περιλαμβάνουν συνήθως τη χρήση υπολογιστικών τεχνολογιών και λογισμικού για την αυτοματοποίηση διαφόρων διαδικασιών.

Ένα τυπικό πληροφοριακό σύστημα μπορεί να αποτελείται από **τέσσερα βασικά στοιχεία**:

- **Υπολογιστικά Συστήματα:** Τα υπολογιστικά συστήματα είναι αναπόσπαστο μέρος της σύγχρονης τεχνολογικής κοινωνίας και αφορούν ένα ευρύ φάσμα συσκευών και λογισμικού που χρησιμοποιούνται για την εκτέλεση εργασιών και την επίλυση προβλημάτων.

Αυτά τα συστήματα καλύπτουν διάφορες ανάγκες και περιλαμβάνουν υπολογιστές προσωπικού υπολογιστή, διακομιστές, φορητές συσκευές, ενσωματωμένα συστήματα και πολλά άλλα. Κάθε κατηγορία συστημάτων έχει τις δικές της ειδικές απαιτήσεις και χαρακτηριστικά, που καθορίζουν τον τρόπο λειτουργίας και τη χρήση τους.

Επιπλέον, το λογισμικό που χρησιμοποιείται σε αυτά τα συστήματα είναι εξίσου σημαντικό. Αυτό μπορεί να περιλαμβάνει λειτουργικά συστήματα, εφαρμογές γραφείου, εφαρμογές βάσεων δεδομένων, λογισμικό ανάλυσης δεδομένων, προγράμματα επεξεργασίας εικόνας και ήχου, και πολλά άλλα προγράμματα που απαιτούνται για τη διαχείριση και την εκτέλεση διαφόρων εργασιών.

Τα υπολογιστικά συστήματα συνεχώς εξελίσσονται και προσαρμόζονται για να ανταποκρίνονται στις ανάγκες των χρηστών και των εφαρμογών τους. Η συνεχής ανάπτυξη τόσο στον τομέα του υλικού όσο και του λογισμικού εξασφαλίζει ότι τα υπολογιστικά συστήματα παραμένουν ικανά να υποστηρίξουν τις αναγκαίες λειτουργίες και εφαρμογές σε διάφορους τομείς, όπως η υγεία, η επιχειρηματικότητα, η εκπαίδευση κ.α.

- **Δίκτυα:** Τα δίκτυα αποτελούν ένα ουσιαστικό μέρος του πληροφοριακού τοπίου, παρέχοντας την υποδομή για τη μεταφορά δεδομένων μεταξύ διάφορων συσκευών. Αυτά τα δίκτυα μπορούν να είναι ενσύρματα, όπως τα δίκτυα Ethernet, όπου η μετάδοση γίνεται μέσω καλωδίων, ή ασύρματα, όπως τα δίκτυα Wi-Fi, όπου η επικοινωνία γίνεται μέσω ασύρματων συνδέσεων.

Η κύρια λειτουργία των δικτύων είναι η ανταλλαγή δεδομένων μεταξύ διαφορετικών συσκευών, όπως υπολογιστές, σέρβερ, εκτυπωτές και άλλες συσκευές. Αυτό επιτρέπει τη συνεργασία και την κοινή χρήση πόρων και πληροφοριών σε διαφορετικά μέρη του πληροφοριακού συστήματος. Μέσω των δικτύων, οι χρήστες μπορούν να ανταλλάσσουν

δεδομένα, να μοιράζονται αρχεία, να προσπελάζουν κοινόχρηστους πόρους και να επικοινωνούν μεταξύ τους.

Επιπλέον, τα δίκτυα επιτρέπουν την πρόσβαση σε διαδίκτυο και σε άλλους εξωτερικούς πόρους, παρέχοντας έτσι στους χρήστες πλήρη συνδεσιμότητα και πρόσβαση σε πληροφορίες και υπηρεσίες που βρίσκονται στο διαδίκτυο.

Τα δίκτυα διαδραματίζουν ζωτικό ρόλο σε κλάδους, όπως η υγεία, η επιχειρηματικότητα, η εκπαίδευση, η ψυχαγωγία και η έρευνα, επιτρέποντας την αποτελεσματική διαχείριση και ανταλλαγή πληροφοριών σε διευρυμένες κοινότητες και οργανισμούς. Η διαρκής ανάπτυξη και εξέλιξη των τεχνολογιών δικτύων συνεχίζεται για την εξασφάλιση υψηλής απόδοσης, ασφάλειας και ευελιξίας στην επικοινωνία και τη μεταφορά δεδομένων.

- **Δεδομένα:** Τα δεδομένα αποτελούν τον θεμέλιο λίθο κάθε πληροφοριακού συστήματος και αντιπροσωπεύουν τις πληροφορίες που συλλέγονται, αποθηκεύονται και επεξεργάζονται από το σύστημα. Μπορούν να περιλαμβάνουν δομημένα δεδομένα, όπως πίνακες βάσεων δεδομένων, όπου η πληροφορία οργανώνεται σε συγκεκριμένες δομές και σχέσεις, ή μη δομημένα δεδομένα, όπως αρχεία κειμένου, εικόνες, ήχοι και βίντεο, όπου η πληροφορία δεν ακολουθεί συγκεκριμένη δομή.

Η σωστή οργάνωση, αποθήκευση και διαχείριση των δεδομένων είναι κρίσιμη για την ομαλή λειτουργία και την αποτελεσματική ανάλυσή τους από το σύστημα. Ένα αποτελεσματικό σύστημα διαχείρισης δεδομένων (ΣΔΔ) επιτρέπει την οργάνωση, την αποθήκευση, την ανάκτηση και την ενημέρωση των δεδομένων με αξιοπιστία και αποτελεσματικότητα.

Επιπλέον, η ανάλυση των δεδομένων είναι σημαντική για την ανάκτηση πληροφοριών και την ανάδειξη προτύπων, τάσεων και συνδέσμων που μπορούν να εξαγάγουν αξία για το σύστημα. Η αποτελεσματική ανάλυση δεδομένων μπορεί να οδηγήσει σε βελτιωμένες αποφάσεις και στρατηγικές για το σύστημα ή τον οργανισμό που τα χρησιμοποιεί.

Τέλος, η ασφάλεια των δεδομένων είναι ζωτικής σημασίας για την προστασία της εμπιστευτικότητας, της ακεραιότητας και της διαθεσιμότητας της πληροφορίας. Η υιοθέτηση ασφαλών πρακτικών αποθήκευσης, μεταφοράς και επεξεργασίας των δεδομένων βοηθά στην αντιμετώπιση των απειλών και των επιθέσεων που μπορεί να αντιμετωπίσει ένα πληροφοριακό σύστημα.

- **Άνθρωποι/Χρήστες:** Οι άνθρωποι αποτελούν τον πιο σημαντικό παράγοντα στην επιτυχία ενός πληροφοριακού συστήματος. Είναι οι τελικοί χρήστες που αλληλεπιδρούν με το σύστημα, χρησιμοποιώντας τις δυνατότητές του για να πραγματοποιούν εργασίες και να λαμβάνουν αποτελέσματα. Η σχεδίαση του πληροφοριακού συστήματος πρέπει να λαμβάνει υπόψη τις ανάγκες, τις δεξιότητες και τις προτιμήσεις των χρηστών, προκειμένου να παρέχει μια εμπειρία που θα είναι εύχρηστη και αποδοτική.

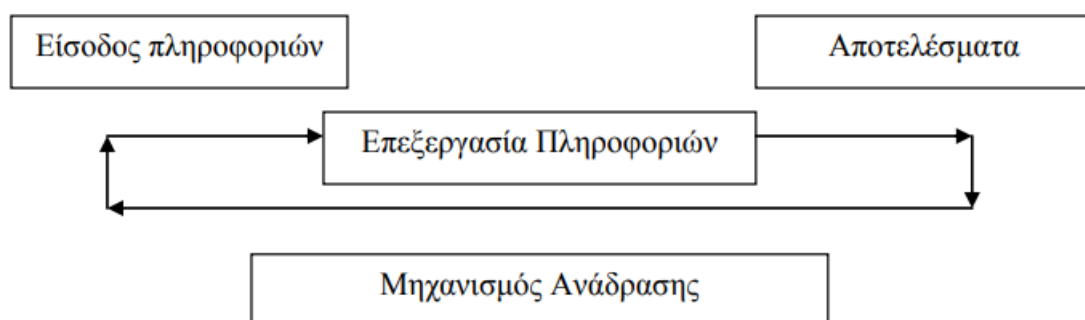
Η ενσωμάτωση των αναγκών και των προτιμήσεων των χρηστών στη σχεδίαση του συστήματος μπορεί να επιτευχθεί μέσω διαφόρων τεχνικών, όπως η χρήση διαδραστικών περιβαλλόντων χρήστη, η διεπαφή με τον χρήστη (UI) και η εμφάνιση (UX) σχεδιασμός, και η εφαρμογή αρχών ανθρωποκεντρικού σχεδιασμού.

Επιπλέον, οι χρήστες πρέπει να έχουν πρόσβαση σε κατάλληλη εκπαίδευση και υποστήριξη για τη χρήση του πληροφοριακού συστήματος, προκειμένου να αξιοποιήσουν πλήρως τις δυνατότητές του και να αποφύγουν πιθανά προβλήματα ή σφάλματα.

Τέλος, η διατήρηση μιας στενής σχέσης με τους χρήστες και η συνεχής ανάδραση είναι ουσιώδης για τη βελτίωση του πληροφοριακού συστήματος με τον χρόνο, ανταποκρινόμενος στις ανάγκες και τις αλλαγές στο περιβάλλον και τις απαιτήσεις των χρηστών.

Τα πληροφοριακά συστήματα έχουν ενσωματωθεί σε κάθε πτυχή της ζωής μας, επιτρέποντας τη διαχείριση και την αξιοποίηση πληροφοριών για τη λήψη αποφάσεων και την εκτέλεση διαδικασιών. Από τη διαχείριση επαγγελματικών διαδικασιών μέχρι την παροχή υγειονομικής περίθαλψης, τη διδασκαλία και την εξερεύνηση νέων γνώσεων, αποτελούν αναπόσπαστο μέρος της καθημερινότητάς μας (Stair, 2022).

Σχηματικά ο βασικός μηχανισμός λειτουργίας ενός πληροφοριακού συστήματος μπορεί να απεικονιστεί ως εξής (διάγραμμα 1) :



ΔΙΑΓΡΑΜΜΑ 1 Ο ΒΑΣΙΚΟΣ ΜΗΧΑΝΙΣΜΟΣ ΕΝΟΣ ΠΛΗΡΟΦΟΡΙΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ (ΠΗΓΗ: ΑΝΑΣΤΑΣΙΑΔΗΣ 2001)

1.2 Πληροφοριακά Συστήματα στον τομέα της υγείας

Τα νοσοκομεία και οι οργανισμοί υγειονομικής περίθαλψης είναι πολύπλοκα συστήματα που περιλαμβάνουν αναρίθμητες περίπλοκες λειτουργίες και διαδικασίες. Συνυπολογίζοντας τις εξελίξεις στην τεχνολογία και την ιατρική γνώση, η πολυπλοκότητα αυτή επιτείνεται περαιτέρω. Με αυτή την πολυπλοκότητα, υπάρχει παραγωγή τεράστιου όγκου πληροφοριών. Τα Συστήματα Πληροφοριών υγείας (ΣΠΥ) είναι υπολογιστικά συστήματα που συλλαμβάνουν, αποθηκεύουν, διαχειρίζονται ή μεταδίδουν αυτόν τον τεράστιο όγκο πληροφοριών που αφορούν την υγεία των ατόμων, την κλινική περίθαλψη ή τις δραστηριότητες των οργανισμών που σχετίζονται με την υγεία. Μπορούμε να τα διαχωρίσουμε σε **τέσσερις κατηγορίες**:

- (1) θεμελιώδη συστήματα,
- (2) οικονομικά συστήματα,
- (3) τμηματικά συστήματα.
- (4) ηλεκτρονικοί ιατρικοί φάκελοι (EMR).

Τα **θεμελιώδη συστήματα** χειρίζονται τις διαχειριστικές πτυχές των οργανισμών υγειονομικής περίθαλψης και περιλαμβάνουν το κύριο ευρετήριο ασθενών και τα υπολογιστικά συστήματα, τα οποία ενημερώνουν άλλα ΣΠΥ σχετικά με τις δραστηριότητες εισαγωγής, εξόδου και μεταφοράς. Το μεταδιδόμενο μήνυμα από ένα σύστημα περιλαμβάνει δημογραφικές πληροφορίες όπως όνομα, ημερομηνία γέννησης και φύλο. Τα θεμελιώδη συστήματα χρησιμεύουν για την εύρεση αυτών των πληροφοριών όπως το όνομα, η ημερομηνία γέννησης, το φύλο, η φυλή και ο αριθμός κοινωνικής ασφάλισης, διασφαλίζοντας ότι όλοι οι εγγεγραμμένοι ασθενείς αντιπροσωπεύονται μία φορά χωρίς διπλές ταυτότητες. Επίσης διασφαλίζουν συνεπή δημογραφικά στοιχεία σε όλα τα ΣΠΥ εντός μιας υγειονομικής περίθαλψης οργανισμού.

Τα **χρηματοοικονομικά συστήματα** χειρίζονται τις λογιστικές πτυχές ενός οργανισμού που σχετίζεται με την υγεία και περιλαμβάνουν συστήματα χρέωσης, τα οποία διαχειρίζονται τις χρεώσεις των νοσοκομείων. Το γενικό λογιστικό βιβλίο είναι ένα άλλο λογιστικό σύστημα που χρησιμεύει ως σπονδυλική στήλη για τα οικονομικά και μη οικονομικά δεδομένα. Υπάρχουν

επίσης χρηματοοικονομικά συστήματα που χειρίζονται οικονομικές και στρατηγικές αποφάσεις υποστήριξης.

Τα **τμηματικά συστήματα** είναι υπολογιστικά συστήματα που είναι ειδικά για τις ανάγκες και λειτουργίες. Το πιο ορατό σύστημα για τους χειρουργικούς παθολόγους είναι το σύστημα πληροφοριών εργαστηρίου ανατομικής παθολογίας (AP-LIS). Το AP-LIS υποστηρίζει ένα ευρύ φάσμα λειτουργιών για το εργαστήριο ανατομικής παθολογίας. Όπως αντίστοιχο, το κλινικό εργαστήριο διαθέτει το σύστημα πληροφοριών κλινικού εργαστηρίου, το οποίο συχνά δεν είναι διαλειτουργικό με το AP-LIS.

Το ακτινολογικό σύστημα διαθέτει το formation system (RIS), το οποίο ασχολείται με καταλόγους ασθενών, παρακολούθηση ασθενών, παραγγελίες, εργασίες, ροές, και καταχώρηση αποτελεσμάτων με αναφορά. Σε συνδυασμό με το RIS στην ακτινολογία είναι το σύστημα αρχειοθέτησης και επικοινωνίας εικόνων, το οποίο διαχειρίζεται τα μεγάλα αποθετήρια των ακτινολογικών εικόνων. Η καρδιολογία διαθέτει επίσης τη δική της έκδοση, ξεχωριστή από εκείνη της ακτινολογίας, η οποία διαχειρίζεται τις καρδιολογικές εικόνες, όπως τα υπερηχοκαρδιογράφημα. Υπάρχουν τμηματικά συστήματα για τα τμήματα που εκτελούν εξειδικευμένες εξετάσεις (π.χ. ECGs στην καρδιολογία, εξετάσεις πνευμονικής λειτουργίας για πνευμονολογικά και ηλεκτρομυογράφημα για τη νευρολογία). Ορισμένα ιδρύματα μπορεί να φέρουν διάφορες μεθόδους απεικόνισης σε ένα ξεχωριστό σύστημα, το οποίο ονομάζεται ουδέτερο αρχείο προμηθευτή (vendor neutral archive). Υπάρχουν τμηματικά συστήματα για τα τμήματα φαρμακείων και διαιτολογίας για τον εξορθολογισμό ροής εργασίας με το χειρισμό παραγγελιών φαρμάκων και διαιτολογικών παραγγελιών, τη διαχείριση συντονισμού, και τη διανομή.

Τα κλινικά εργαστήρια και τα τμήματα οικονομικής διαχείρισης ήταν τα πρώτα που υιοθέτησαν HIS. Άλλοι τύποι ΣΠΥ αναπτύχθηκαν για να αντικατοπτρίζουν τις επιχειρησιακές και λειτουργικές ανάγκες των αντίστοιχων τμημάτων τους. Ως συνέπεια αυτής της υποεξειδίκευσης, τα ΣΠΥ αναπτύχθηκαν ανεξάρτητα σε απομονωμένα συστήματα και με τη δική τους ατομική υποδομή βάσης δεδομένων. Αυτό συνέβη με το AP-LIS και το σύστημα πληροφοριών κλινικών εργαστηρίων και η συνέπεια ήταν η έλλειψη διαλειτουργικών ΣΠΥ που δεν διασυνδέονται ή αλληλεπιδρούν απρόσκοπτα μεταξύ τους. Υπάρχουν και άλλοι λόγοι για αυτή την έλλειψη διαλειτουργικότητας, όπως η έλλειψη τυποποίησης για τη διαλειτουργικότητα. Οι προμηθευτές των τμηματικών HIS είχαν κίνητρο να δημιουργήσουν εξάρτηση των οργανισμών υγειονομικής περίθαλψης και όχι να ενωθούν για τη δημιουργία διαλειτουργικών προτύπων με άλλους ανταγωνιστικούς προμηθευτές. Αυτό το ήθος συνεχίστηκε μέχρι πρόσφατα, όπου οι δυνάμεις που υποστηρίζουν τη διαλειτουργικότητα έγιναν αρκετά ισχυρές ώστε να επηρεάσουν την πολιτική.

Η ύπαρξη ενός πιο διαλειτουργικού πληροφοριακού συστήματος ήταν το ζητούμενο. Δεν ήταν πρόσφατη ιδέα, αλλά ξεκίνησε ήδη από το 1991. Τότε, το Ινστιτούτο Ιατρικής (Institute of Medicine IOM) έθεσε ένα όραμα και απηύθυνε έντονη έκκληση για εθνική εφαρμογή των ηλεκτρονικών ιατρικών φακέλων. Το IOM αναγνώρισε ότι οι ομάδες ιατρών, τα νοσοκομεία, και

άλλοι οργανισμοί υγειονομικής περίθαλψης παρείχαν συχνά περίθαλψη χωρίς το όφελος της πλήρους πληροφόρησης σχετικά με την κατάσταση του ασθενούς, το ιατρικό ιστορικό του, την υπηρεσία που παρέχονται σε άλλα περιβάλλοντα, ή φάρμακα που συνταγογραφούνται από άλλους κλινικούς ιατρούς. Το ΙΟΜ κάλεσε για διαλειτουργικότητα μέσω της αυτοματοποίησης και της σύνδεσης των πληροφοριών σχετικά με τις υπηρεσίες που παρέχονται σε ασθενείς σε περιπατητικές και ιδρυματικές ρυθμίσεις (π.χ., συναντήσεις, διαδικασίες και βοηθητικές εξετάσεις). Αυτή η διαλειτουργικότητα θα παρείχε με τη σειρά της μια πλούσια πηγή πληροφοριών για σκοπούς μέτρησης και βελτίωσης της ποιότητας, ώστε να καταστεί δυνατή η ποιότητα της ολοκληρωμένης κλινικής περίθαλψης. Εξαιτίας εκθέσεων όπως αυτές που εκδόθηκαν από το ΙΟΜ, υπήρξε πρόσφατα ευρεία διάδοση. Το πιο σημαντικό είναι ότι τα ομοσπονδιακά προγράμματα κινήτρων έχουν στοχεύσει στα EMR για τον επανασχεδιασμό του συστήματος υγειονομικής περίθαλψης με στόχο τη δημιουργία ποιοτικής ολοκληρωμένης κλινικής περίθαλψης.

Η διαλειτουργικότητα από τα ΣΠΥ πραγματοποιείται με δύο αρχετυπικές αρχιτεκτονικές:

(1) ολοκληρωμένα συστήματα και

(2) διασυνδεδεμένα συστήματα.

Ολοκληρωμένα συστήματα είναι εκείνα στα οποία τα δεδομένα στεγάζονται στην ίδια βάση δεδομένων και χρησιμοποιούνται από διάφορα ΣΠΥ (δηλ., υπηρεσιακά, οικονομικά και χρηματιστηριακά). Τα διασυνδεδεμένα συστήματα είναι ξεχωριστές εφαρμογές και τα δεδομένα στεγάζονται στις δικές τους βάσεις δεδομένων. Τα δεδομένα επικοινωνούν μεταξύ κάθε ξεχωριστού συστήματος μέσω μιας μηχανής διασύνδεσης που συνήθως χρησιμοποιεί ένα διεθνές σύστημα Health Level 7 (HL7).

Οι περισσότεροι οργανισμοί υγειονομικής περίθαλψης των ΗΠΑ βρίσκονται σε κάποια επανάληψη του Έκδοσης 2 για το HL7- αλλιώς προσδιορίζεται ως HL7 Version 2.x (V2). Η σημασία του HL7Version 2.x (V2) θα εξεταστεί αργότερα. Υπάρχουν βάσιμα επιχειρήματα και για τις δύο αρχέτυπες αρχιτεκτονικές όσον αφορά τα πλεονεκτήματα και τα μειονεκτήματα.

Με τα ολοκληρωμένα συστήματα, η ροή πληροφοριών είναι θεωρητικά πιο απρόσκοπτη λόγω

της βάσης μιας βάσης δεδομένων. Φανταστείτε ένα ολοκληρωμένο σύστημα για την ανατομική παθολογία, κλινικό εργαστήριο και μοριακή παθολογία για την αναφορά ενός υποθετικού δείγματος μυελού των οστών. Επί του παρόντος δεν υπάρχει ένα τέτοιο βελτιστοποιημένο ολοκληρωμένο σύστημα, ωστόσο, μια παθολογική έκθεση για έναν πυρήνα μυελού των οστών μπορεί να παραχθεί από το AP-LIS με απρόσκοπτη ενσωμάτωση ενός πλήρους αίματος (CBC)

και/ή του αποτελέσματος αναρρόφησης από το σύστημα πληροφοριών κλινικού εργαστηρίου. Μόλις επιστραφούν τα αποτελέσματα των βοηθητικών μοριακών δοκιμών, ένα προσάρτημα μπορεί να επισημανθεί στην έκθεση, όλες οι συναλλαγές κάτω από μία εφαρμογή χωρίς ανάγκη για διαφορετικές συνδέσεις, ενέργειες αντιγραφής-επικόλλησης ή διόρθωσης ζητημάτων μορφοποίησης. Ωστόσο, υπάρχει η γενική αίσθηση ότι τα ολοκληρωμένα συστήματα δεν είναι βελτιστοποιημένα για επιχειρησιακές και λειτουργικές ανάγκες των τμημάτων. Οι μεγάλοι πωλητές είναι σε καλύτερη θέση να δημιουργήσουν ολοκληρωμένα συστήματα, αλλά λόγω του μεγέθους τους πιστεύεται ότι είναι λιγότερο εστιασμένοι ή δεκτικοί σε στις επιχειρησιακές και λειτουργικές ανάγκες κάθε επιμέρους τμήματος. Μεγάλοι προμηθευτές θεωρούνται επίσης λιγότερο ευέλικτοι για να χειριστούν μικρές αλλαγές που βελτιώνουν τις εφαρμογές. Επιπλέον, επειδή αυτά τα ολοκληρωμένα συστήματα βασίζονται σε μία βάση δεδομένων, οι μικρές αλλαγές σε μια εφαρμογή μπορεί να έχουν απρόβλεπτες συνέπειες σε άλλες ολοκληρωμένες εφαρμογές. Στην ίδια κατεύθυνση, τα ολοκληρωμένα συστήματα είναι πιο ευάλωτα σε διαταραχές. Οι αλλοιώσεις στη βάση δεδομένων, οι διακοπές λειτουργίας ή οι αναβαθμίσεις επηρεάζουν όλες τις εφαρμογές επειδή δεν είναι σε θέση να λειτουργήσουν ανεξάρτητα. Η θεωρητική απλούστευση που προκύπτει επειδή τα διαφορετικά συστήματα τρέχουν στην ίδια βάση δεδομένων είναι ένα σημείο πώλησης των ολοκληρωμένων συστημάτων.

Με τα **συστήματα διασύνδεσης**, τα τμήματα μπορούν να διατηρούν το δικό τους τμηματικό HIS, το οποίο συχνά ανταποκρίνεται περισσότερο στις επιχειρησιακές και λειτουργικές ανάγκες του αντίστοιχου τμήματος. Ο όρος, best of breed, προέρχεται από την περιγραφή τέτοιων τμηματικών συστημάτων. Οι περισσότερες από τις αποχρώσεις και προσαρμογές που απαιτούν τα τμήματα ίσως αντιμετωπίζονται αποτελεσματικότερα με ένα σύστημα best of breed. Συνήθως τα καλύτερα συστήματα προέρχονται από μικρότερους και πιο ευέλικτους προμηθευτές με κίνητρα αυτοσυντήρησης για να κάνουν το πιο βελτιστοποιημένο σύστημα στον τομέα τους. Συχνά οι εν λόγω προμηθευτές είναι πιο δεκτικοί και πιο ικανοί να χειριστούν αλλαγές στα προϊόντα τους.

Τα διασυνδεδεμένα συστήματα εξαρτώνται σε μεγάλο βαθμό από καλά υλοποιημένες επικοινωνίες και δρομολόγηση. Αυτός είναι ο λόγος για τον οποίο απαιτούνται περισσότερες δοκιμές σε τέτοιου είδους αρχετυπικές αρχαιοθήκες. Στην πράξη, ακόμη και στο καλύτερο διασυνδεδεμένο σύστημα, η διαλειτουργικότητα ποτέ δεν επιτυγχάνεται πλήρως. Με το ίδιο σενάριο κάτω από ένα διασυνδεδεμένο σύστημα, η ύπαρξη μιας ολοκληρωμένης έκθεσης για τον ίδιο πυρήνα μυελού των οστών γίνεται πιο δύσκολη. Μια παθολογική έκθεση για έναν πυρήνα μυελού των οστών παράγεται από ένα AP-LIS, αλλά δεν υπάρχει απρόσκοπτη ενσωμάτωση της CBC ή/και του αποτελέσματος της αναρρόφησης από τις πληροφορίες του κλινικού εργαστηρίου. Η επισημάνση ενός μοριακού προσαρτήματος αποτελεί παρόμοια πρόκληση. Η διαδικασία υπογραφής επιβάλλει ξεχωριστές συνδέσεις και, μόλις ληφθούν τα αποτελέσματα, ο μηχανισμός για τη μεταφορά π.χ. των αποτελεσμάτων στο AP-LIS είναι συνήθως copy-paste. Χωρίς απρόσκοπτη αυτοματοποίηση για τη μεταφορά των

αποτελεσμάτων, μπορεί να προκύψουν σφάλματα. Η συνολική διαδικασία αποσύνδεσης γίνεται αποδιοργανωτική λόγω των διαφορετικών συνδέσεων, των ενεργειών αντιγραφής-επικόλλησης και της διόρθωσης της μορφοποίησης.

Στην πράξη, τα περισσότερα οργανωτικά συστήματα κλινικής πληροφορικής είναι συνδυασμοί διασυνδεδεμένων συστημάτων, με ποικίλους βαθμούς του καθενός. Σε έναν ιδανικό κόσμο, ένα ολοκληρωμένο σύστημα θα πρέπει να είναι βελτιστοποιημένο για τις επιχειρησιακές και λειτουργικές ανάγκες των τμημάτων. Ομοίως, ένα διασυνδεδεμένο σύστημα θα πρέπει να έχει τους απρόσκοπτους τρόπους διασύνδεσης μεταξύ των διαφόρων εφαρμογών.

Η Επιτροπή για την Ποιότητα της Φροντίδας Υγείας στην Αμερική το 2001 κάλεσε τους οργανισμούς υγειονομικής περίθαλψης να στοχεύουν σε υγειονομική περίθαλψη που είναι ασφαλής, αποτελεσματική, έγκαιρη, αποδοτική, δίκαιη και με επίκεντρο τον ασθενή. Αυτό θα απαιτούσε μεγαλύτερη πρόσβαση σε κοινές πληροφορίες, επέκταση των διαύλων επικοινωνίας και την αντιμετώπιση των κοινωνικοτεχνικών προκλήσεων. Αυτό επίσης σηματοδοτούσε ένα βασικό ρόλο για την τεχνολογία των πληροφοριών. Μόνο μέσω της υποστήριξης από προσεκτικά και συνειδητά σχεδιασμένων πληροφοριακών συστημάτων θα μπορούσε η υγειονομική περίθαλψη να επιτύχει αυτούς τους στόχους. Η λύση ήταν το **EMR**, το οποίο λειτουργεί ως θεμελιώδες εργαλείο βοηθώντας τους χρήστες να ενσωματώνουν, να διανέμουν, να οργανώνουν, να ερμηνεύουν και να αντιδρούν καλύτερα στις πληροφορίες και τις γνώσεις για την υγεία.

Θα συζητηθεί πολύ η προστιθέμενη αξία των EMR- ωστόσο, μερικά από τα μειονεκτήματα θα πρέπει να επισημανθούν. Τα EMR μπορεί να είναι ακριβά. Συχνά φαίνεται να υπάρχει ένα επίπεδο τεχνικής εξοικείωσης ως προϋπόθεση για τη χρήση των EMR. Υπάρχουν κόστη σε εκπαίδευση, επειδή τα EMR δεν είναι απαραίτητα σχεδιασμένα ώστε να είναι διαισθητικά για τους λιγότερο εξοικειωμένους με τους υπολογιστές. Οι διαταραχές της ροής εργασίας είναι επίσης αξιοσημείωτες, ιδίως με ηλεκτρονικές σημειώσεις, οι οποίες είχαν πολλά αναφερόμενα μειονεκτήματα. Εισαγωγή δεδομένων για την τεκμηρίωση μέσω κλικ, η πληκτρολόγηση μπορεί να πάρει σε ορισμένους χρήστες περισσότερο χρόνο από το να γράφουν σε χαρτί. Τα πλεονεκτήματα ενός EMR όσον αφορά τις λειτουργικότητές του φαίνεται να υπερκαλύπτουν τα εμπόδια. Ένα καλά σχεδιασμένο και λειτουργικό EMR δίνει περαιτέρω τη δυνατότητα στους χρήστες να χρησιμοποιούν από αυτό το σύστημα υγείας πληροφορίες και γνώσεις για να επηρεάσουν την πολιτική και τη λήψη αποφάσεων, τη δράση, την ατομική προσπάθεια, τα αποτελέσματα της δημόσιας υγείας και την έρευνα.

ΔΙΟΙΚΗΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ

Ανάλογα με τον οργανισμό υγειονομικής περίθαλψης, τα ιδρυτικά και οικονομικά συστήματα μπορούν να εμφανίζονται ως ξεχωριστά ΣΠΥ. Με τέτοιους τύπους αρχιτεκτονικής διασύνδεσης, απαιτούνται εκτεταμένες δοκιμές για να διασφαλιστεί ότι η φροντίδα που συνέβη μαζί με την τεκμηρίωση που σημειώθηκε στο EMR συμπίπτει και ταιριάζει σωστά με την τιμολόγηση, τους κωδικούς χρέωσης και τον κατάλογο ασθενών και παρόχων στα θεμελιώδη και οικονομικά συστήματα. Τα ΣΠΥ έχουν εξελιχθεί ώστε να ενσωματώνουν τις λειτουργίες των θεμελιωδών και των οικονομικών συστημάτων με ολοκληρωμένο τρόπο. Αυτή η ενσωμάτωση των θεμελιωδών και οικονομικών ΣΔΠ για να τη διαχείριση της τεκμηρίωσης για κανονιστικούς σκοπούς, συμμόρφωση, διασφάλιση ποιότητας και τιμολόγηση καταλήγει να είναι το back end των σύγχρονων EMR.

ΣΥΓΚΕΝΤΡΩΣΗ ΤΩΝ ΠΛΗΡΟΦΟΡΙΩΝ ΚΑΙ ΤΩΝ ΓΝΩΣΕΩΝ ΣΤΟΝ ΤΟΜΕΑ ΤΗΣ ΥΓΕΙΑΣ

Το μπροστινό μέρος του EMR είναι το πιο ορατό σύστημα για τους ιατρούς και τους ασθενείς. Είναι αυτό το μπροστινό μέρος που είναι πανταχού παρόν σε πολλές κλινικές ρυθμίσεις.

Το ΣΠΔ χρησιμεύει ως ένα ενοποιητικό περιβάλλον εργασίας, συγκεντρώνοντας τις πληροφορίες για την υγεία για προβολή και παρουσίαση. Όπως και στο χάρτινο διάγραμμα, οι πληροφορίες υγείας αποκτώνται, συλλέγονται, ταξινομούνται και αποθηκεύονται από άλλα συστήματα πληροφοριών, όπως το AP-LIS. Αλλά πέρα από ένα χάρτινο διάγραμμα, το ΣΠΔ είναι προσβάσιμο, φορητό και διαθέσιμο -παράγοντες που αποτέλεσαν την πηγή ικανοποίησης των ιατρών από ένα τέτοιο σύστημα πληροφοριών.

Το EMR έχει βελτιώσει την πρακτική της χειρουργικής παθολογίας. Τα EMR ξεπερνούν τα προβλήματα της προσβασιμότητας και της φορητότητας των χάρτινων διαγραμμάτων, όπως η φυσική αρχειοθέτηση και ανάκτηση των πληροφοριών και η λανθασμένη τοποθέτηση. Δεν υπάρχει πλέον η ανάγκη να ζητείται ένα χάρτινο διάγραμμα ή να καλέσετε ή να στείλετε e-mail για κλινική συσχέτιση. Οι πληροφορίες υγείας είναι διαθέσιμες και κατά τη διάρκεια του 24ώρου και πολλαπλοί χρήστες μπορούν να έχουν πρόσβαση σε έναν ενιαίο φάκελο ασθενούς κατά παραγγελία, κάτι που είναι δύσκολο έργο με ένα χάρτινο διάγραμμα.

ΔΙΑΧΕΙΡΙΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Για τη χειρουργική παθολογία, η διαχείριση των αποτελεσμάτων μεταφράζεται στον τρόπο παρουσίασης των αναφορών χειρουργικής παθολογίας στο EMR. Οι εκθέσεις χειρουργικής παθολογίας ανακατασκευάζονται οπτικά στο EMR μέσω της ανταλλαγής μηνυμάτων HL7 έκδοση 2.x (V2) και αποδίδονται σε μη μορφοποιημένο κείμενο. Αυτή η μετάφραση έχει ως αποτέλεσμα την απώλεια στυλ και μορφοποίησης στην αποδιδόμενη έκθεση EMR. Τα EMR δεν είναι σε θέση να χειρίζονται έντονα, πλάγια, χρώματα, πίνακες, σχήματα ή εικόνες.

Το HL7 έκδοση 2.x (V2) πρότυπο ανταλλαγής μηνυμάτων που αποδίδει τις αναφορές σε ένα EMR λειτουργεί καλύτερα για εργαστηριακές εκθέσεις, οι οποίες είναι μικρότερες και λιγότερο εκτενείς από τις εκθέσεις χειρουργικής παθολογίας.

Το πρότυπο ανταλλαγής μηνυμάτων HL7 Έκδοση 2.x (V2) δεν προοριζόταν για την απεικόνιση χειρουργικών παθολογικών αναφορών σε διαισθητική διάταξη. Οι εκθέσεις χειρουργικής παθολογίας είναι σημαντικά μεγαλύτερες- σκεφτείτε τις εκ τομές όγκων με μεγάλες συνοπτικές λίστες ελέγχου. Οι αναφορές που παράγονται από τα ουρογεννητικά και μοριακά διαγνωστικά εργαστήρια δεν ανακατασκευάζονται εύκολα με τον ίδιο τρόπο στα ιδρυματικά EMR. Παθολόγοι, οι οποίοι ευθυγράμμισαν τα δεδομένα τους μέσω μιας μορφής ευθυγραμμισμένων στηλών στο Microsoft Word, εκπλήσσονται συχνά όταν διαπιστώνουν ότι οι πληροφορίες που εμφανίζονται στο EMR δεν διατηρούν καλά τις ευθυγραμμίσεις.

Ένα ακόμη πιο επικίνδυνο παράδειγμα είναι ορισμένα σύμβολα όπως το "~", όπου οι λέξεις που ακολουθούν το σύμβολο μπορούν να διαγραφούν. Έτσι, δηλώσεις όπως "~no carcinoma identified" γίνονται "carcinoma identified". Οι παθολόγοι θα πρέπει να γνωρίζουν την αναπαράσταση των εκθέσεων τους στο EMR και όχι μόνο από το AP-LIS. Επιπλέον, είναι ευθύνη ενός εργαστηρίου, σύμφωνα με τις τροποποιήσεις για τη βελτίωση των κλινικών εργαστηρίων (CLIA), να γνωρίζει πώς εμφανίζονται τα αποτελέσματα στο EMR και να διασφαλίζει ότι, εάν υπάρχουν πολλαπλά ιδρυματικά EMR, τα αποτελέσματα είναι συνεπή. Οι παθολόγοι θα πρέπει να γνωρίζουν πού βρίσκεται το αποτέλεσμα των εκθέσεων που δημιουργούν αντιστοιχίζονται στο EMR και αν υπάρχουν δυνατότητες για κρυφά αποτελέσματα υπό οποιαδήποτε μετατροπή. Η διασφάλιση ότι οι χειρουργικές παθολογικές εκθέσεις παραμένουν ευανάγνωστες και εύκολα κατανοητές αποδεικνύεται όλο και πιο δύσκολη με την ταχεία ανάπτυξη προηγμένων μοριακών δοκιμών, όπως η αλληλούχιση επόμενης γενιάς (NGS). Η NGS είναι εγγενώς περίπλοκη και οι διαισθητικές διατάξεις απαιτούν κατά κανόνα σχήματα, γραφήματα, πίνακες, διαγράμματα κ.ο.κ., αντί για αδόμητες παραγράφους κειμένου. Ακόμη χειρότερα, σε αντίθεση με τις λίστες ελέγχου για τον καρκίνο, δεν υπάρχουν καθιερωμένες κατευθυντήριες γραμμές για την υποβολή πληροφοριών από το NGS. Για να ενισχυθούν δυνατότητες αναφοράς στο EMR, ορισμένα ιδρύματα έχουν εφαρμόσει μια διεπαφή PDF για να ξεπεραστούν οι περιορισμοί των ανακατασκευασμένων αναφορών της έκδοσης 2.x (V2) του

HL7. Η έκθεση σε μορφή PDF είναι καλή για σύνθετες αναφορές επειδή επιτρέπει διαισθητικές διατάξεις με δυνατότητες συμπερίληψης έντονων γραμμμάτων και χρωμάτων και δυνατότητα αναδημιουργίας σχημάτων, γραφημάτων, πινάκων και ούτω καθεξής. Αυτό όχι μόνο αποτρέπει την παρερμηνεία μέσω της μορφοποίησης και της λάθος μετάδοσης, αλλά γίνεται και μια μορφή παθητικής υποστήριξης κλινικών αποφάσεων.

Η υλοποίηση μιας διεπαφής PDF από ένα AP-LIS απαιτεί συνεργασίες με ομάδες τεχνολογίας πληροφοριών από άλλα HIS εντός του ιδρύματος, ιδίως τη μηχανή διεπαφής και το EMR. Πολλά EMR δεν διαθέτουν τη λειτουργικότητα για την εμφάνιση αρχείων PDF ή δεν έχουν τη λειτουργικότητα αυτή άμεσα διαθέσιμη. Επιπλέον, ιδρύματα με αποθήκες δεδομένων μπορεί να αντιστέκονται στην ιδέα, επειδή τα PDF εμφανίζουν και αποδίδουν μόνο πληροφορίες που δεν μπορούν εύκολα να εξορύσσονται από τα δεδομένα. Για να παρακαμφθούν αυτές οι ανησυχίες, θα πρέπει να διασφαλιστούν οι διαδικασίες σε όλα τα μέρη ότι οι ροές δεδομένων διατηρούνται εάν οι τρέχουσες διαδικασίες εξόρυξης πληροφοριών είναι σε ισχύ. Όταν το έγγραφο Microsoft Word δημιουργείται κατά την υπογραφή στο AP-LIS, ενεργοποιείται η δημιουργία μιας έκδοσης PDF που αποθηκεύεται σε έναν προσβάσιμο θεσμικό διακομιστή. Το μήνυμα HL7 έκδοσης 2.x (V2), που δημιουργείται μετά την ολοκλήρωση της υπογραφής της έκθεσης, περιέχει ένα πεδίο δείκτη το οποίο επιτρέπει στο EMR να ανακτήσει και να εμφανίσει το έγγραφο PDF. Η ιδρυματική αποθήκη δεδομένων, σε αυτό το παράδειγμα, συλλαμβάνει πληροφορίες μέσω της ανάλυσης του μηνύματος HL7 και έτσι διατηρείται η ροή δεδομένων τροφοδοσίας. Ως προστιθέμενη αξία για την κλινική περίθαλψη, μια καλοδιαμορφωμένη έκθεση σε μορφή PDF μπορεί να εμφανιστεί στο EMR. Η διεπαφή PDF αντιπροσωπεύει το πρώτο στάδιο της ενισχυμένης αναφοράς και αναμφισβήτητα την παθητική υποστήριξη αποφάσεων. Μπορεί να λειτουργήσει ως λύση ενισχυμένης αναφοράς εντός των ορίων της έκδοσης 2.x (V2) του HL7. Το επόμενο στάδιο της αναφοράς μπορεί να έρθει μέσω της έννοιας της αρχιτεκτονικής κλινικών εγγράφων (CDA). Οι τρέχουσες επαναλήψεις της έκδοσης HL7 2.x (V2) δεν περιλαμβάνουν την CDA. Η CDA αποτελεί συστατικό στοιχείο της έκδοσης 3 (V3) του HL7 η οποία είναι η επόμενη γενιά της ανταλλαγής μηνυμάτων HL7. Το CDA προοριζόταν για τη μετάδοση προτύπων να λαμβάνουν υπόψη τα δομικά στοιχεία των κλινικών εγγράφων. Οι σημειώσεις κλινικής επίσκεψης δεν είναι απλά συσσωρευμένες γραμμές κειμένου, αλλά και έγγραφα με δομικά στοιχεία, όπως η φυσική εξέταση, η ανασκόπηση των συστημάτων και η ενότητα αξιολόγησης και σχεδίου.

Οι εκθέσεις χειρουργικής παθολογίας, επίσης υψηλής περιεκτικότητας σε κείμενο, μπορούν να θεωρηθούν ανάλογες με τις κλινικές σημειώσεις επίσκεψης με δομικά στοιχεία, όπως το τμήμα της τελικής διάγνωσης, το τμήμα της αδρής περιγραφής, το τμήμα του προσαρτήματος κ.ο.κ.

Οι εκθέσεις δομούνται και τοποθετούνται με τέτοιο τρόπο ώστε να τυποποιούν την εμφάνιση των εγγράφων στο EMR. Το πλεονέκτημα της χρήσης CDA σε σχέση με μια διεπαφή PDF είναι ότι η ροή δεδομένων ή η έκθεση που αποδίδεται στο EMR γίνεται πιο δομημένη και ευκολότερη στην επεξεργασία και την εξόρυξη δεδομένων, σε αντίθεση με το αδόμητο PDF.

Ευτυχώς, η λειτουργία CDA αναμένεται να ενσωματωθεί σε όλα τα εμπορικά EMR των ΗΠΑ, καθώς περιλαμβάνεται στις απαιτήσεις για ουσιαστική χρήση σύμφωνα με τον νόμο HITECH. Ωστόσο, παραμένει ασαφές αν αυτή η εφαρμογή του CDA θα γίνει μέσω της υιοθέτησης του HL7 Version 3 (V3), κυρίως λόγω της διστακτικότητας των προμηθευτών EMR να υιοθετήσουν τα πρότυπα HL7 Version 3. Οι λόγοι για τους οποίους οι προμηθευτές EMR δεν υιοθετούν αυτά τα πρότυπα είναι πολυπαραγοντικοί. Επειδή οι σημερινοί προμηθευτές EMR είναι μεγάλοι και ευρέως διαδεδομένοι, το κίνητρο για την εξέλιξη προς καλύτερα πρότυπα ανταλλαγής μηνυμάτων είναι περιορισμένο. Τα συστήματα αυτά δημιουργούν επίσης μεγαλύτερη εξάρτηση και αλληλεξάρτηση. Οι εξελίξεις και οι βελτιώσεις έχουν ακούσιες συνέπειες σε όλο το σύστημα, καθιστώντας ακόμη και τις σταδιακές αλλαγές πιο δύσκολες. Εν ολίγοις, τα συστήματα γίνονται τόσο μεγάλα που οι ανατρεπτικές ή επαναστατικές αλλαγές είναι δύσκολες στην ενσωμάτωσή τους.

Επιπλέον, το κόστος για αλλαγές υποδομής στα ΣΠΔ για την υιοθέτηση του προτύπου HL7 έκδοσης 3 (V3) είναι απαγορευτικό, καθώς η V3 δεν είναι συμβατή με την έκδοση 2.x (V2). Αυτή η ασυμβατότητα σχεδιάστηκε σκόπιμα, ώστε η V3 να είναι επαναστατική και να μην περιορίζεται από τους κληρονομικούς περιορισμούς της V2. Επιπλέον, η πολυπλοκότητα της V3 και η ανωριμότητα της εφαρμογής της επιδεινώνουν την κατάσταση.

Ως αποτέλεσμα, οι προμηθευτές EMR είναι απρόθυμοι να αναλάβουν τους πόνους υιοθέτησης της V3, εκτός αν πιεστούν σημαντικά. Ωστόσο, η εφαρμογή του CDA μπορεί να γίνει και μέσω του Fast Healthcare Interoperability Resources (FHIR). Αν και η συζήτηση του FHIR ξεφεύγει από το πεδίο αυτού του άρθρου, αξίζει να σημειωθεί ότι το FHIR είναι μια υβριδική λύση ανταλλαγής μηνυμάτων, που μπορεί να λειτουργήσει αυτόνομα και να ενσωματώσει πολλές ευεργετικές λειτουργίες της HL7 V3, όπως το CDA. Το πλεονέκτημα του FHIR είναι ότι μπορεί να χρησιμοποιηθεί σε συνεργασία με τα τρέχοντα πρότυπα HL7 V2.x (V2).

Οι εκθέσεις χειρουργικής παθολογίας απαιτούν πιο δυναμικές δυνατότητες για τη διαχείριση των ενημερώσεων. Αυτές οι ενημερώσεις συχνά περιλαμβάνουν προσθήκες, καθώς ένα παθολογικό δείγμα υπογράφεται ως αρχική έκθεση και στη συνέχεια εκτελούνται συμπληρωματικές εξετάσεις, τα αποτελέσματα των οποίων επισημαίνονται συνήθως στο τέλος της αρχικής έκθεσης. Αυτό μπορεί να είναι επαρκές για μία μόνο εξέταση, όπως μια ανοσοϊστοχημική προγνωστική ομάδα σε ένα δείγμα μαστού. Ωστόσο, όταν εκτελούνται πολλαπλές εξετάσεις σε διάφορα μέρη ενός δείγματος, η τοποθέτηση επιπλέον στοιχείων στο τέλος της έκθεσης καθιστά τη διάταξη λιγότερο διαισθητική. Το NGS προσθέτει στην πολυπλοκότητα αυτού του ζητήματος, καθώς μπορεί να εκτελείται σε πολλαπλές περιοχές όγκου που αναφέρονται ως διαφορετικά μέρη ενός δείγματος. Ιδανικά, η επισήμανση ενός προσαρτήματος στο σχετικό τμήμα είναι η πιο διαισθητική προσέγγιση. Ωστόσο, αυτό απαιτεί τα τμήματα να διαχωρίζονται ως συστατικά της τελικής διάγνωσης, κάτι που δεν υποστηρίζεται από τις περισσότερες AP-LIS.

Με τη γονιδιωματική αναφορά, τα AP-LIS δεν είναι έτοιμα, καθώς υπάρχουν πολλά άλυστα ζητήματα. Πληροφορίες όπως τα χαρακτηριστικά απόδοσης μιας ανάλυσης ή αν αυτή είναι μονής ανάλυσης ή πολυπαραμετρική είναι δύσκολο να μεταδοθούν. Επιπλέον, προκύπτουν ερωτήματα σχετικά με την αναφορά μεταλλάξεων ή αρνητικών αποτελεσμάτων, καθώς και αν θα πρέπει να αποθηκεύονται πληροφορίες σχετικά με την αλληλουχία, όπως η θέση, το βάθος αλληλούχισης, η ποιότητα αλληλούχισης και οι περιοχές που εξετάστηκαν.

Τα τρέχοντα πρότυπα επικοινωνίας HL7 έκδοσης 2.x (V2) δεν υποστηρίζουν τη μορφοποίηση δεδομένων και μεταδεδομένων για γονιδιωματικές πληροφορίες, γεγονός που περιορίζει την αναφορά σε απλοποιημένα γονιδιωματικά αποτελέσματα. Προς το παρόν, η διεπαφή PDF αποτελεί την καλύτερη λύση για την εκτέλεση βελτιωμένων παθολογικών αναφορών, με την προοπτική της χρήσης CDA στο μέλλον. Αυτό παρέχει κάποια λύση στους περιορισμούς που επιβάλλουν τα τρέχοντα πρότυπα επικοινωνίας, επιτρέποντας τη μετάδοση πληροφοριών με γραφικό ή διαδραστικό τρόπο.

Μία πιθανή βέλτιστη λύση για τη βελτιωμένη αναφορά θα μπορούσε να προέλθει από μια διεπαφή προγράμματος εφαρμογής (API), η οποία θα χρησιμοποιεί υπερσυνδέσμους με βάση τον Παγκόσμιο Ιστό HTML για τη διανομή πληροφοριών από και προς ένα EMR. Το πλεονέκτημα αυτής της προσέγγισης είναι ότι οι πληροφορίες και οι αναφορές μπορούν να ενημερώνονται δυναμικά, δημιουργώντας πραγματικά διαδραστικές αναφορές.

Θα υπάρχει επίσης η δυνατότητα παροχής μεταδεδομένων της έκθεσης, καθώς και πληροφοριών σχετικά με τους περιορισμούς και το επίπεδο των αποδείξεων για την ερμηνεία, μέσω του ιστορικού και των παραπομπών κατά παραγγελία.

ΜΗΝΥΜΑΤΑ (ΗΛΕΚΤΡΟΝΙΚΗ ΕΠΙΚΟΙΝΩΝΙΑ ΚΑΙ ΣΥΝΔΕΣΙΜΟΤΗΤΑ)

Δεν αρκεί πλέον η απλή απόδοση μιας παθολογικής έκθεσης σε ένα EMR και η υπόθεση ότι όλες οι κλινικές ομάδες είναι ενήμερες και λαμβάνουν τις απαραίτητες πληροφορίες, ιδιαίτερα όταν υπάρχουν ενημερώσεις. Η κλινική περίθαλψη, καθώς γίνεται πιο ολοκληρωμένη, επιβάλλει σε διάφορες ομάδες και προσωπικό να λαμβάνουν ενημερωμένες πληροφορίες για πολλούς ασθενείς. Η διαχείριση του πότε, ποιος και πού δρομολογούνται οι αναφορές με τυχόν ενημερώσεις είναι υψίστης σημασίας.

Αυτό δημιουργεί προκλήσεις στον τρόπο που τα EMR διαχειρίζονται τις πληροφορίες αναφορών με ενημερώσεις, κυρίως τις προσθήκες. Συχνά, ο πάροχος που υποβάλλει ένα παθολογικό δείγμα δεν είναι το άτομο για το οποίο προορίζονται οι ενημερωμένες πληροφορίες. Ορισμένα EMR δεν διαθέτουν τη λειτουργικότητα ειδοποίησης των κατάλληλων παρόχων για τέτοιες ενημερώσεις. Για παράδειγμα, εάν ένα δείγμα οσφυονωτιαίας μαστοτομής υπογραφεί, ο χειρουργός που το υπέβαλε είναι συνήθως ενήμερος για την τελική

παθολογία και την κατάσταση του περιθωρίου εκτομής. Ωστόσο, εάν ζητηθεί NGS από έναν ογκολόγο για το ίδιο δείγμα και στη συνέχεια υπάρξει μεταβίβαση της περίθαλψης σε άλλο ογκολόγο, η δρομολόγηση των αποτελεσμάτων NGS στον νέο πάροχο είναι σημαντικά δύσκολη.

Καταστροφικά σφάλματα μπορούν να προκύψουν αν τα αποτελέσματα δεν προωθούνται κατάλληλα, με κάποιες κρίσιμες πληροφορίες να μην γνωστοποιούνται σε μέλη της κλινικής ομάδας. Αν και τα EMR βελτιώνονται στη διαχείριση των μεταβαλλόμενων παρόχων και στη διανομή των πληροφοριών, οι μηχανισμοί ειδοποίησης για ενημερωμένες παθολογικές αναφορές εξακολουθούν να είναι σε εξέλιξη.

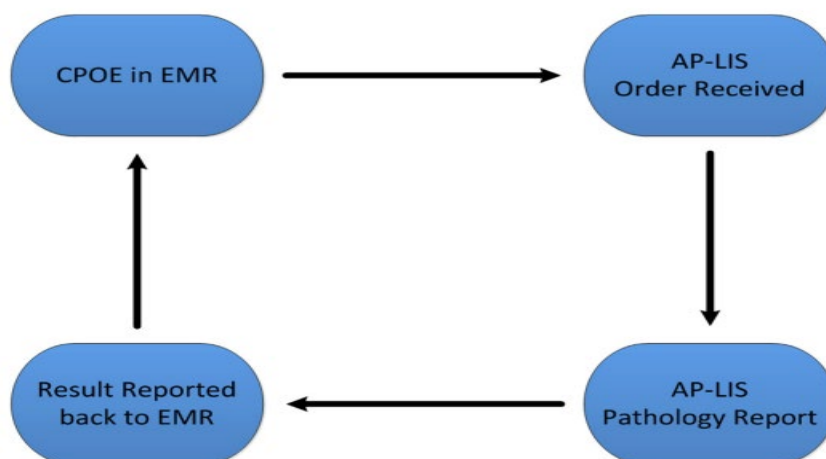
Ορισμένα EMR έχουν τη δυνατότητα να στέλνουν ομαδικές ειδοποιήσεις στους παρόχους που εμπλέκονται στην περίθαλψη, ενισχύοντας την ηλεκτρονική επικοινωνία με την κλινική ομάδα. Τα ιδανικά EMR θα παρακολουθούν την αλυσίδα επιμέλειας, ενώ εναλλακτικοί μηχανισμοί περιλαμβάνουν δυνατότητες ανταλλαγής μηνυμάτων, εντολές προς έγκριση και προώθηση μέσω φαξ, εκτύπωσης και άλλων συστημάτων. Δυστυχώς, ορισμένα από τα EMR που χρησιμοποιούνται σήμερα δεν διαθέτουν τέτοιες ισχυρές δυνατότητες προώθησης και, ακόμη και αν υπάρχουν, οι μηχανισμοί τους δεν είναι αρκετά καθορισμένοι για τον προσδιορισμό του κατάλληλου παρόχου για την προώθηση των αποτελεσμάτων.

ΜΗΧΑΝΟΓΡΑΦΗΜΕΝΗ ΚΑΤΑΧΩΡΗΣΗ ΕΝΤΟΛΩΝ ΠΑΡΟΧΟΥ

Η ηλεκτρονική εισαγωγή εντολών παρόχου (CPOE) είναι μια διαδικασία που επιτρέπει στους παρόχους να χρησιμοποιούν υπολογιστές για την άμεση εισαγωγή φαρμάκων, διαδικασιών, εντολών, διαβουλεύσεων και εξετάσεων, όπως ακτινολογικές και εργαστηριακές. Οι παραγγελίες εισάγονται στο EMR και το σύστημα διαβιβάζει την εντολή μέσω του δικτύου με τη βοήθεια του HL7. Από πλευράς ποιότητας και ασφάλειας, το CPOE έχει αποδείξει ότι μειώνει τα ιατρικά σφάλματα. Σε μελέτη των Bates και συνεργατών, τα λάθη στις ιατρικές παραγγελίες μειώθηκαν κατά περισσότερο από το ήμισυ. Το CPOE μειώνει τα σφάλματα μέσω της εξάλειψης της παρερμηνείας λόγω δυσανάγνωστου γραφικού χαρακτήρα ή μεταγραφής, της εύκολης ταυτοποίησης του παρόχου, της ελαχιστοποίησης περιττών εξετάσεων και της διασφάλισης ότι περιλαμβάνονται επαρκείς πληροφορίες στις παραγγελίες. Εκτός από την εξάλειψη παρερμηνειών, ο πάροχος παραγγελίας είναι εύκολα αναγνωρίσιμος και το σύστημα αποτρέπει διπλές παραγγελίες. Επιπλέον, οι παραγγελίες δεν μπορούν να ολοκληρωθούν αν δεν παρέχονται επαρκείς πληροφορίες σε ορισμένα πεδία του μενού. Δυστυχώς, τα σημερινά AP-LIS δεν υποστηρίζουν τη διασύνδεση για CPOE με τα EMR. Επίσης, τα νοσοκομειακά συστήματα δεν έχουν σχεδιαστεί για να διαχειρίζονται τα στοιχεία ενός καλά εφαρμοσμένου CPOE στη χειρουργική παθολογία, όπως η εκτύπωση γραμμωτού κώδικα, η σάρωση και η διανομή στο σημείο προσκόμισης των ιστών. Αντί για ηλεκτρονικές παραγγελίες, οι σημερινές ροές εργασίας συχνά ξεκινούν με δείγματα και χάρτινες αιτήσεις, οι οποίες καταλήγουν σε

αναφορές που εμφανίζονται σε ένα EMR. Αυτή η διαδικασία περιγράφει έναν μη κλειστό βρόχο, καθώς η παθολογική έκθεση δεν ξεκινά με ηλεκτρονική παραγγελία μέσω CPOE στο EMR. Έτσι, τα προ-αναλυτικά δεδομένα δεν καταγράφονται ούτε αξιολογούνται.

Το παρακάτω σχήμα δείχνει μια εξιδανικευμένη διεπαφή CPOE με ένα AP-LIS, η οποία έχει δημιουργήσει έναν κλειστό βρόχο με την έκθεση, ξεκινώντας από μια παραγγελία στο CPOE. Με το κλείσιμο του βρόχου, το CPOE επιτρέπει μια ροή εργασίας χωρίς χαρτί, με γραμμωτούς κωδικούς που εκτυπώνονται στο σημείο απόκτησης των ιστών και μεταφέρονται στα σημεία καταχώρησης της παθολογίας. Το CPOE για δείγματα χειρουργικής παθολογίας ανοίγει την πόρτα για την εφαρμογή επιχειρηματικής ανάλυσης, καταγράφοντας προ-αναλυτικά δεδομένα, όπως ο χρόνος από την ηλεκτρονική παραγγελία μέχρι την παραλαβή του δείγματος στο εργαστήριο, καθώς και τα δεδομένα παρακολούθησης από τη σάρωση των γραμμωτών κωδικών πριν από το σημείο προσκόμισης των παθολογικών.



ΥΠΟΣΤΗΡΙΞΗ ΑΠΟΦΑΣΕΩΝ

Η υποστήριξη αποφάσεων (DS) παρέχει στους παρόχους τη δυνατότητα να διαθέτουν γνώσεις και πληροφορίες για τον ασθενή. Η εξιδανικευμένη DS φιλτράρεται έξυπνα και παρουσιάζεται εγκαίρως. Περιλαμβάνει μια ποικιλία εργαλείων για την ενίσχυση της λήψης αποφάσεων, συμπεριλαμβανομένων των ηλεκτρονικών ειδοποιήσεων και υπενθυμίσεων προς τους παρόχους, κλινικών κατευθυντήριων οδηγιών, συνόλων εντολών για συγκεκριμένες καταστάσεις, υποδείγματα τεκμηρίωσης, διαγνωστική υποστήριξη και σχετικές με το πλαίσιο πληροφορίες αναφοράς. Όπως συζητήθηκε προηγουμένως με την αναφορά των αποτελεσμάτων NGS, μια εστιασμένη PDF έκθεση και σύνοψη μπορεί να αποτελέσει μια παθητική έκδοση του DS. Πολλά ιδρύματα διαθέτουν ήδη μια διεπαφή PDF ως συστατικό στοιχείο της κλινικής αναφοράς των NGS για την ενσωμάτωση στα EMR.

Η DS καθίσταται αναγκαία για την αντιμετώπιση της αυξανόμενης υπερφόρτωσης με πληροφορίες που οι κλινικοί γιατροί παρέχουν μια πλατφόρμα για την ενσωμάτωση της τεκμηριωμένης γνώσης στην παροχή φροντίδας. Η πλειονότητα των εφαρμογών DS λειτουργεί ως συνιστώσα ολοκληρωμένων EMR.

Το DS γίνεται επίσης ο λόγος για την εφαρμογή των EMR για την εκπλήρωση κυβερνητικών εντολών για ουσιαστική χρήση. Το DS δεν είναι χωρίς εμπόδια. Ιδανικά, οι εφαρμογές DS διασυνδέονται με το AP-LIS- ωστόσο, οι περισσότεροι πωλητές AP-LIS δεν έχουν ακόμη αναπτύξει τέτοιες λειτουργίες διασύνδεσης. Το DS επίσης, εάν είναι κακώς σχεδιασμένο, δημιουργεί αναποτελεσματικότητα με διακοπές, όπως πολλά αναδυόμενα παράθυρα, προειδοποιήσεις, εμπόδια και ειδοποιήσεις για την έναρξη μιας παραγγελίας.

Το DS θεωρείται απαραίτητο συστατικό των δοκιμών και των εκθέσεων NGS, διότι τα συστήματα DS μπορούν να χρησιμοποιήσουν πανομικά δεδομένα για να καθοδηγήσουν την επιλογή θεραπείας και την ιατρική ακριβείας. Επί του παρόντος υπάρχουν αρκετές απαραίτητες ολοκληρωμένες λειτουργίες CPOE και DS που δεν υποστηρίζονται από τα τρέχοντα EMR για να καταστεί δυνατή η διεξαγωγή δοκιμών NGS.

Δεν υπάρχει απρόσκοπτος μηχανισμός για τα αρχεία, όσον αφορά τη συγκατάθεση και τη συμβουλευτική, να μεταφέρονται σε ένα εργαστήριο NGS. Για τη γεννητική γραμμή, δεν υπάρχει μηχανισμός που να διασφαλίζει ότι δεν θα υπάρξουν περιττές δαπανηρές επαναλήψεις όπως η παραπομπή σε προϋπάρχον αποτέλεσμα εξέτασης ή η διενέργεια διπλής εξέτασης σε όλη τη διάρκεια της ζωής.

Οι πολύπλοκες εξετάσεις, όπως η NGS, απαιτούν βοήθεια. Οι κλινικοί ιατροί πρέπει να είναι σε ένα επίπεδο πολυπλοκότητας που να γνωρίζουν πότε η παραγγελία υβριδισμού φθορισμού in situ (FISH), καρυότυπου ή NGS είναι κατάλληλη σε όλα τα είδη των σεναρίων. Ένα άλλο μεγάλο εμπόδιο είναι η έλλειψη προτύπων που να επιτρέπουν ισχυρά εργαλεία βιοπληροφορικής, όπως η παραλλαγή βάσεις δεδομένων να διασυνδέονται με εφαρμογές DS στο EMR.

ΥΠΟΣΤΗΡΙΞΗ ΑΣΘΕΝΩΝ

Κατά τη συζήτηση για την υποστήριξη των ασθενών, εισάγεται η έννοια της διαδικτυακής πύλης ασθενών(portal). Μια διαδικτυακή πύλη ασθενών είναι ένας ασφαλής διαδικτυακός ιστότοπος που παρέχει στους ασθενείς εύκολη 24ωρη πρόσβαση σε προσωπικές πληροφορίες υγείας από οπουδήποτε υπάρχει σύνδεση στο Διαδίκτυο. Τα EMRs του σήμερα εφαρμόζουν portal ασθενών για να βελτιώσουν την επικοινωνία μεταξύ των παρόχων, ενδυναμώνουν τους ασθενείς, υποστηρίζουν τη φροντίδα μεταξύ των επισκέψεων και, κυρίως, βελτιώνουν τα αποτελέσματα τους. Οι πύλες ασθενών θεωρούνται ως μια λύση για την

ικανοποίηση των προσδοκιών του σταδίου 1 της ουσιαστικής χρήσης για την εμπλοκή των ασθενών στη δική τους υγειονομική περίθαλψη.

Πολλές πύλες ασθενών εμφανίζουν εργαστηριακά αποτελέσματα, αλλά οι εκθέσεις χειρουργικής παθολογίας εγείρουν άλλα ζητήματα. Από τη σκοπιά του χειρουργικού παθολόγου, το πότε και πώς εμφανίζονται οι παθολογικές εκθέσεις σε ένα portal ασθενών δημιουργεί ζητήματα. Εάν ένας ασθενής έχει κάνει εκτομή για έναν όγκο, ποιος είναι ο κατάλληλος χρόνος για να εμφανιστεί η έκθεση στο portal; Εάν η έκθεση εμφανιστεί πολύ σύντομα, ο ασθενής μπορεί να μην δει εγκαίρως τον πάροχο για μια συζήτηση σχετικά με θέματα πρόγνωσης. Ο τρόπος εμφάνισης της έκθεσης είναι επίσης σημαντικός. Θα μπορούσε ένας ασθενής να είναι σε θέση να ερμηνεύσει μια παθολογική έκθεση γραμμένη για τους κλινικούς ιατρούς, ή αυτό θα οδηγήσει σε περισσότερες ερωτήσεις από κλινικούς ιατρούς ή ενδεχομένως ακόμη και από παθολόγους;

Καθώς η εφαρμογή των πυλών ασθενών γίνεται όλο και πιο διαδεδομένη, θα πρέπει να υπάρξει μια πιο εστιασμένη στον ασθενή έκθεση με κατάλληλη γλώσσα και ορολογία χωρίς να δημιουργεί περισσότερα προβλήματα από ό,τι επιλύει, διότι μπορεί ενδεχομένως να οδηγήσει σε ερμηνευτική ασυμφωνία μεταξύ της επίσημης παθολογικής έκθεσης και της εστιασμένης στον ασθενή. Ακόμα δεν έχει εισαχθεί μια βελτιστοποιημένη λύση ή μοντέλο.

ΚΑΤΑΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ, ΥΠΟΒΟΛΗ ΕΚΘΕΣΕΩΝ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΤΗΣ ΥΓΕΙΑΣ ΤΟΥ ΠΛΗΘΥΣΜΟΥ

Τα πρώιμα EMR θα μπορούσαν να θεωρηθούν ως μια βελτιωμένη ηλεκτρονική έκδοση του χάρτινου διαγράμματος επειδή είναι σε θέση να παρέχουν ασφαλή, σε πραγματικό χρόνο, διαδραστικά και φορητά στοιχεία για ένα άτομο. Λόγω αυτής της εστίασης στο άτομο, το EMR θεωρείται ασθενοκεντρικό, επειδή λειτουργεί εντός 1 οργανισμού υγειονομικής περίθαλψης και επικεντρώνεται περισσότερο στα επεισόδια/συναντήσεις και την άμεση υγειονομική περίθαλψη, όπως οι κλινικές πληροφορίες που συλλέγονται από το γραφείο ενός παρόχου.

Ένας άλλος όρος που χρησιμοποιείται εναλλακτικά, αν και λανθασμένα, είναι ο ηλεκτρονικός φάκελος υγείας(ΗΦΥ). Οι EMR και οι EHR έχουν παρόμοια λειτουργικότητα επειδή και οι δύο λειτουργούν ως βελτιωμένες ηλεκτρονικές εκδόσεις των χάρτινων διαγραμμάτων. Οι ΗΦΥ είναι πιο διαχρονικοί, καλύπτοντας το σύνολο των επεισοδίων/συναντήσεων ενός ασθενούς και σε όλους τους οργανισμούς υγειονομικής περίθαλψης. Σε αντίθεση με τα EMR, οι ΗΦΥ μπορούν να περιλαμβάνουν περισσότερους από έναν μόνο φορέα υγειονομικής περίθαλψης που αρχικά συνέλεξε και συνέταξε τις πληροφορίες. Κατασκευάζονται για να μοιράζονται πληροφορίες με άλλους παρόχους υγειονομικής περίθαλψης, όπως εργαστήρια και ειδικούς, ώστε να περιέχουν πληροφορίες από όλους τους κλινικούς ιατρούς που εμπλέκονται στη φροντίδα του ασθενούς. Στη θεωρία, ο ΗΦΥ δίνει τη

δυνατότητα εύκολης ανταλλαγής ιατρικών πληροφοριών μεταξύ των ενδιαφερομένων και οι πληροφορίες ενός ασθενούς ακολουθούν το άτομο αυτό μέσω των διαφόρων τρόπων περίθαλψης που χρησιμοποιεί. Οι πληροφορίες μπορούν να μετακινηθούν μαζί με τον ασθενή - σε έναν ειδικό, σε ένα νοσοκομείο, σε ένα γηροκομείο, στην επόμενη πολιτεία ή ακόμη και σε ολόκληρη τη χώρα. Αυτό επιτρέπει τη σύγκριση των διαφόρων αρχείων ενός ατόμου και επιτρέπει την εξαγωγή δεδομένων από αρχεία διαφορετικών ατόμων. Αυτή η κινητικότητα των πληροφοριών επιτρέπει στον ΗΦΥ να προχωρήσει πέρα από την υγεία και την άμεση ασθενοκεντρική υγειονομική περίθαλψη, προς την κατεύθυνση με στόχο τη βελτίωση της υγείας του πληθυσμού.

Ένας τεράστιος πλούτος δεδομένων περιέχεται στις βάσεις δεδομένων των EMRs/ΗΦΥ. Κλινικές σημειώσεις και αναφορές από το εργαστήριο, την παθολογία και την ακτινολογία. Αμέτρητοι άλλοι τύποι στοιχείων δεδομένων καταγράφονται από το EMR/ΗΦΥ, όπως ημερομηνίες κλινικών επισκέψεων, αίμα και εμβολιασμοί. Τέτοια δεδομένα μπορούν να παρακολουθούνται διαχρονικά και να χρησιμοποιούνται για να ελέγξετε πώς τα πάνε οι ασθενείς και να εντοπίσετε ποιοι ασθενείς οφείλουν να υποβληθούν σε προληπτικές εξετάσεις. Μέσω αυτής της παρακολούθησης των συλλεχθέντων δεδομένων, τα EMRs/ΗΦΥ, εννοιολογικά, βελτιώνουν τη συνολική ποιότητα της περίθαλψης εντός του οργανισμού υγείας. Με ακόμη περισσότερα προηγμένα queries και παρακολούθηση της βάσης δεδομένων EMR/EHR και την κινητικότητα των εξαγόμενων δεδομένων από διαφορετικά αρχεία, οι πληροφορίες μπορούν να χρησιμοποιηθούν για ελέγχους, έρευνα, αξιολόγηση αποτελεσμάτων και επιτήρηση. Η δυνατότητα χρήσης αυτών των πληροφοριών με ουσιαστικότερους τρόπους για τη διαλεύκανση της διορατικότητας και της γνώσης είναι ένας βασικός στόχος των ΗΦΥ.

Με αυτή την υπόσχεση για καλύτερη συνολική υγειονομική περίθαλψη για τα άτομα και την υγεία του πληθυσμού, υπάρχουν εμπόδια που απαιτούν λύσεις προτού επιτευχθεί το πλήρες δυναμικό των ΗΦΥ στο πλαίσιο της βελτίωσης της υγειονομικής περίθαλψης των ατόμων και της υγείας του πληθυσμού. Τα EMRs, παρά το γεγονός ότι περιέχουν τεράστιο πλούτο δεδομένων, τα περισσότερα από τα δεδομένα που καταγράφονται είναι αδόμητα και δεν επιδέχονται εύκολα επεξεργασία από υπολογιστή. Αυτό δεν αποτελεί συνήθως ζήτημα με στοιχεία δεδομένων, όπως ημερομηνίες κλινικών επισκέψεων, αρτηριακές πιέσεις και εργαστηριακά δεδομένα, τα οποία είναι συχνά σύντομα, αριθμητικά και λιγότερο κειμενικά. Για τα στοιχεία δεδομένων που περιέχονται σε έγγραφα κειμένου, όπως οι κλινικές σημειώσεις και οι παθολογικές εκθέσεις, ωστόσο, οι συσσωρευμένες γραμμές κειμένου σε ορισμένα EMR παρέχουν ελάχιστη δομή ή πλαίσιο για τη δημιουργία ανάλυσης αλγορίθμων για την εξόρυξη και την εξαγωγή των δεδομένων. Ένα παράδειγμα είναι η καταγραφή δεδομένων από ένα οικογενειακό ιστορικό ενός κλινικού σημειώματος για οικογενειακές συσχετίσεις διαδικασιών νόσου. Το οικογενειακό ιστορικό αναφέρεται μερικές φορές ως συσσωρευμένες γραμμές κειμένου που δεν προσφέρουν καμία δομή ή πλαίσιο σε έναν υπολογιστή. Η γραφική αναπαράσταση με οικογενειακά δέντρα υποθετικά είναι ένας καλύτερος μηχανισμός για την καταγραφή δεδομένων- ωστόσο, αυτή η λειτουργία δεν είναι καθολική.

Η αφηγηματική μορφή με βάση το κείμενο τμημάτων όπως το οικογενειακό ιστορικό μπορεί να αλλάξει στο μέλλον με την ουσιαστική χρήση με την εντολή ότι τα EMR θα καταγράφουν καλύτερα τα δεδομένα μέσω περισσότερων δομημένων μηχανισμών. Η CDA, όπως συζητήθηκε προηγουμένως, μπορεί να αποτελέσει ένα βήμα προς τα εμπρός και υπόσχεται μια τέτοια δομημένη καταγραφή. Προς το παρόν, ωστόσο, τα πρότυπα μπορούν να κλειδώσουν τη δομή ορισμένων στοιχείων δεδομένων ενός εγγράφου. Η συνοπτική αναφορά δειγμάτων καρκίνου έχει εφαρμοστεί ευρέως και χρησιμοποιείται και εφαρμόζεται εύκολα. Λόγω της συνέπειας και της δομής των πληροφοριών, αυτά τα συνοπτικά πρότυπα καρκίνου έχουν δομημένες γραμμές κειμένου με τρόπο που να επιδέχεται επεξεργασία από υπολογιστή. Τα πρότυπα για πολλές κλινικές σημειώσεις EMR έχουν λάβει μικτές αντιδράσεις, με μειονέκτημα τον περιορισμό της ευελιξίας στην εφαρμογή του προτύπου σε διαφοροποιημένες καταστάσεις. Αυτό βασικά εξαναγκάζει την τεκμηρίωση σε πλαίσια χωρίς δυνατότητα προσαρμογής για κλινικές εξαιρέσεις.

1.3 Ερωτηματολόγιο

Ερωτηματολόγιο είναι μια ομάδα ή μια ακολουθία ερωτήσεων, οι οποίες έχουν σχεδιαστεί για να συλλέξουν πληροφορίες για κάποιο θέμα από έναν ερωτώμενο. Τα ερωτηματολόγια μπορούν να είναι τόσο τα κλασικά που χρησιμοποιούνται σε δειγματοληπτικές ή απογραφικές έρευνες όσο και φόρμες συλλογής διοικητικών δεδομένων. Τα ερωτηματολόγια παίζουν κεντρικό ρόλο στη διαδικασία συλλογής δεδομένων. Έχουν σημαντικό αντίκτυπο στην ποιότητα των δεδομένων γιατί είναι το μέσο με το οποίο αυτά συλλέγονται.

Οι ερωτήσεις που θα τεθούν πρέπει να συμμορφώνονται με τη διατύπωση στόχων της έρευνας και να συλλέγουν πληροφορίες που είναι χρήσιμες για την ανάλυση. Όλες οι ανάγκες για πληροφορίες πρέπει να ικανοποιηθούν. Από την άλλη πλευρά, κάθε ερώτηση πρέπει να έχει ξεκάθαρο λόγο για να βρίσκεται στο ερωτηματολόγιο. Πρέπει να είναι γνωστό το γιατί γίνεται η κάθε ερώτηση όπως και το πώς θα χρησιμοποιηθούν οι πληροφορίες που συλλέγονται. Η διατύπωση των ερωτήσεων πρέπει να είναι ξεκάθαρη. Οι ερωτήσεις πρέπει να γίνονται με σειρά που είναι λογική για τους ερωτώμενους. Πρέπει να είναι ευκολονόητες και να μπορούν να απαντηθούν με ακρίβεια. Τέλος, το ερωτηματολόγιο πρέπει να δοκιμάζεται πριν από τη χρήση του με γνωστικές δοκιμές.

Η αποτελεσματική συλλογή δεδομένων απαιτεί προσεκτική σχεδίαση και υλοποίηση. Κατά τη σύνταξη ενός ερωτηματολογίου, είναι ουσιώδες να είναι οι ερωτήσεις σαφείς, κατανοητές και ακριβείς. Η σαφήνεια στη διατύπωση εξασφαλίζει ότι οι ερωτώμενοι κατανοούν τι ζητείται από αυτούς και παρέχουν ακριβείς απαντήσεις.

Επιπλέον, η επιλογή των κατάλληλων τύπων ερωτήσεων επηρεάζει την ποιότητα των δεδομένων. Οι ανοικτές ερωτήσεις επιτρέπουν στους ερωτώμενους να εκφράσουν τις απόψεις τους ελεύθερα, ενώ οι κλειστές ερωτήσεις προσφέρουν σαφείς επιλογές και διευκολύνουν τη συγκέντρωση συγκρίσιμων δεδομένων. Ένα καλό ερωτηματολόγιο πρέπει να είναι φιλικό προς τους ερωτώμενους, προσφέροντας κατανοητές ερωτήσεις και ενθαρρύνοντας τη συμμετοχή.

Ένα άλλο κρίσιμο σημείο είναι οι έλεγχοι ποιότητας πριν από την εφαρμογή του ερωτηματολογίου. Η δοκιμή του σε μικρό δείγμα ατόμων βοηθά στην αναγνώριση πιθανών προβλημάτων, όπως ασάφειες ή αντιφάσεις στις ερωτήσεις.

Τέλος, η κατάρτιση του προσωπικού που θα διαχειριστεί το ερωτηματολόγιο είναι κρίσιμη. Τα άτομα που συλλέγουν τα δεδομένα πρέπει να γνωρίζουν τον σκοπό της έρευνας και τον τρόπο συμπλήρωσης του ερωτηματολογίου, προκειμένου να εξασφαλιστεί ομοιομορφία και αξιοπιστία στα δεδομένα.

Επομένως, ένα καλά σχεδιασμένο ερωτηματολόγιο εξασφαλίζει την αποτελεσματική συλλογή δεδομένων με όσο το δυνατόν λιγότερα σφάλματα και αντιφάσεις, παρέχοντας έτσι αξιόπιστα και αντιπροσωπευτικά αποτελέσματα με μειωμένο κόστος και μικρή διάρκεια.

(Πετράκος, 2021)

1.4 Χρησιμότητα Ερωτηματολογίων στο Υγειονομικό Σύστημα

Τα ερωτηματολόγια αποτελούν ισχυρό εργαλείο για τη συλλογή πληροφοριών από τους ασθενείς, προσφέροντας οφέλη σε πολλούς τομείς της υγειονομικής περίθαλψης.

Συλλογή Αντικειμενικών Δεδομένων από τους Ασθενείς:

Τα ερωτηματολόγια προσφέρουν έναν τρόπο για τους ασθενείς να μοιραστούν την πραγματική τους εμπειρία και να παρέχουν αντικειμενικά δεδομένα σχετικά με την υγεία τους, τα συμπτώματά τους και την ποιότητα ζωής τους. Η συλλογή αντικειμενικών δεδομένων από μεγάλο αριθμό ασθενών μπορεί να επιτρέψει την ανάλυση των τάσεων και τη δημιουργία προγραμμάτων παρέμβασης για τη βελτίωση της ποιότητας της φροντίδας. Με βάση τα δεδομένα που συλλέγονται, οι ιατροί και οι νοσηλευτές μπορούν να προσαρμόσουν τη θεραπεία και τη φροντίδα για να ανταποκριθούν καλύτερα στις ανάγκες κάθε ασθενούς. Τέλος, τα δεδομένα από τα ερωτηματολόγια μπορούν να χρησιμοποιηθούν για την πραγματοποίηση επιστημονικών ερευνών, την αξιολόγηση της αποτελεσματικότητας της θεραπείας και την εκπαίδευση του υγειονομικού προσωπικού. Με αυτόν τον τρόπο, η συλλογή αντικειμενικών δεδομένων από ερωτηματολόγια συμβάλλει σημαντικά στη βελτίωση της ποιότητας και της αποτελεσματικότητας της φροντίδας της υγείας.

Προσαρμοστική και Εξατομικευμένη Φροντίδα:

Η προσαρμοστική και εξατομικευμένη φροντίδα αποτελεί κρίσιμο παράγοντα στη βελτίωση της ποιότητας της υγειονομικής περίθαλψης. Μέσω της συλλογής ατομικών πληροφοριών μέσω ερωτηματολογίων, οι υγειονομικοί επαγγελματίες αποκτούν μια πιο πλήρη εικόνα των αναγκών και των προτιμήσεων κάθε ασθενούς. Κατανοώντας καλύτερα το ιστορικό υγείας, τις προτιμήσεις τους σε σχέση με τη θεραπεία και άλλους παράγοντες που επηρεάζουν τη φροντίδα τους, οι υγειονομικοί επαγγελματίες είναι σε θέση να παρέχουν μια φροντίδα που προσαρμόζεται ακριβώς στις ανάγκες και τις προσδοκίες τους.

Η εξατομικευμένη φροντίδα επιτρέπει τη δημιουργία ενός προσωποποιημένου πλάνου θεραπείας που λαμβάνει υπόψη τις μοναδικές ανάγκες και τις προτιμήσεις κάθε ατόμου. Αυτό

μπορεί να περιλαμβάνει την προσαρμογή των φαρμακευτικών αγωγών, την προτεραιοποίηση συγκεκριμένων θεραπειών ή μεθόδων πρόληψης, και την παροχή προσωποποιημένων συμβουλών για τον τρόπο ζωής. Με την εξατομικευμένη φροντίδα, οι ασθενείς αισθάνονται όχι μόνο ότι λαμβάνουν πιο προσεκτική και στοχευμένη περίθαλψη, αλλά επίσης ότι οι ανησυχίες και οι προτιμήσεις τους λαμβάνονται σοβαρά υπόψη από το ιατρικό προσωπικό. Με αυτόν τον τρόπο, η προσαρμοστική και εξατομικευμένη φροντίδα συμβάλλει στη βελτίωση της ικανοποίησης των ασθενών και στην ενίσχυση της αποτελεσματικότητας της θεραπείας.

Παρακολούθηση Εξελίξεων των Ασθενών:

Η παρακολούθηση των εξελίξεων των ασθενών μέσω της συνεχούς χρήσης ερωτηματολογίων είναι κρίσιμη για την παρακολούθηση της υγείας τους με τον πάροδο του χρόνου. Αυτή η πρακτική παρέχει πολύτιμες πληροφορίες σχετικά με την αποτελεσματικότητα των θεραπειών και την πρόοδο της ασθένειας. Μέσω των ερωτηματολογίων, οι υγειονομικοί επαγγελματίες μπορούν να παρακολουθήσουν την αλλαγή των συμπτωμάτων, την ανταπόκριση στη θεραπεία και τη γενική εξέλιξη της κατάστασης υγείας του ασθενούς.

Μέσω αυτής της παρακολούθησης, οι γιατροί μπορούν να αξιολογήσουν την απόκριση του ασθενούς στη θεραπεία και να προσαρμόσουν το πλάνο θεραπείας ανάλογα. Επιπλέον, η παρακολούθηση της προόδου μπορεί να ανιχνεύσει προβλήματα ή επιπλοκές που μπορεί να προκύψουν κατά τη διάρκεια της θεραπείας, επιτρέποντας την άμεση αντίδραση και την προσαρμογή της προσέγγισης.

Τέλος, η παρακολούθηση των εξελίξεων των ασθενών μπορεί να παρέχει στους ίδιους τους ασθενείς μια καλύτερη κατανόηση της προόδου τους και της κατάστασης της υγείας τους. Αυτό μπορεί να συμβάλει στην ενίσχυση της συμμετοχής τους στη θεραπευτική διαδικασία και στη διαχείριση της υγείας τους με πιο αποτελεσματικό τρόπο.

Πρόληψη Προβλημάτων Υγείας:

Τα ερωτηματολόγια μπορούν να χρησιμοποιηθούν για τον προσδιορισμό πιθανών προβλημάτων υγείας πριν αυτά γίνουν σοβαρά, επιτρέποντας έγκαιρη παρέμβαση και πρόληψη. Αποτελεί κρίσιμο κομμάτι της υγειονομικής περίθαλψης και η χρήση ερωτηματολογίων μπορεί να συμβάλει σημαντικά σε αυτόν τον τομέα. Μέσω των ερωτηματολογίων, οι υγειονομικοί επαγγελματίες μπορούν να προσδιορίσουν πιθανά προβλήματα υγείας που ενδέχεται να εξελιχθούν σε σοβαρά προβλήματα στο μέλλον. Μπορεί να γίνει έγκαιρη αναγνώριση των πιθανών ανωμαλιών ή συμπτωμάτων που μπορεί να υποδεικνύουν προβλήματα υγείας, επιτρέποντας στους επαγγελματίες της υγείας να παρέμβουν άμεσα και να λάβουν τα αναγκαία μέτρα πρόληψης.

Επιπλέον, η χρήση ερωτηματολογίων μπορεί να βοηθήσει στην παρακολούθηση πιθανών παραγόντων κινδύνου και συμπτωμάτων που σχετίζονται με συγκεκριμένες ασθένειες ή παθήσεις. Αυτό επιτρέπει την εφαρμογή προληπτικών μέτρων για τη μείωση του κινδύνου εμφάνισης αυτών των προβλημάτων υγείας ή την πρόληψη της περαιτέρω εξέλιξής τους σε σοβαρότερα προβλήματα.

Τέλος, η πρόληψη προβλημάτων υγείας μέσω ερωτηματολογίων μπορεί να συμβάλει στην προαγωγή της ενημέρωσης και της ευαισθητοποίησης των ατόμων για την υγεία τους. Μέσω της συνεχούς εκπαίδευσης και ενημέρωσης μέσω των ερωτηματολογίων, οι ασθενείς μπορούν να είναι πιο ενήμεροι για τις προσωπικές τους υγειονομικές ανάγκες και να λαμβάνουν προληπτικά μέτρα για την προστασία της υγείας τους.

Ερευνητικές Εφαρμογές:

Η συλλογή δεδομένων από ερωτηματολόγια μπορεί να χρησιμοποιηθεί σε ερευνητικά πλαίσια για την κατανόηση των παραγόντων που επηρεάζουν την υγεία και την απόδοση της θεραπείας. Μέσω των ερωτηματολογίων, οι ερευνητές μπορούν να συγκεντρώσουν δεδομένα σχετικά με διάφορες πτυχές της υγείας και της νόσου από διαφορετικές ομάδες ανθρώπων.

Τα ερωτηματολόγια μπορούν να συμπεριλάβουν ερωτήσεις που αφορούν στις συνήθειες διατροφής, τις ασκήσεις, τις ιατρικές ιστορίες, τα γονίδια, τη χρήση φαρμάκων και άλλους παράγοντες που μπορεί να επηρεάσουν την υγεία. Η συλλογή αυτών των δεδομένων μπορεί να βοηθήσει στην ανάπτυξη πιο αποτελεσματικών προγραμμάτων πρόληψης και θεραπείας.

Επιπλέον, η χρήση ερωτηματολογίων στην έρευνα επιτρέπει τη σύγκριση δεδομένων μεταξύ διαφορετικών ομάδων ανθρώπων, όπως η σύγκριση μεταξύ ατόμων που πάσχουν από μια συγκεκριμένη ασθένεια και ατόμων που δεν την πάσχουν. Αυτό μπορεί να οδηγήσει στην ανάδειξη παραγόντων που σχετίζονται με την εμφάνιση ή την πρόοδο της ασθένειας, καθώς και στην ανίχνευση πιθανών παραγόντων κινδύνου ή προστασίας.

Τέλος, η ανάλυση των δεδομένων που συλλέγονται μέσω ερωτηματολογίων μπορεί να προσφέρει ενδιαφέρουσες ενδείξεις και συστάσεις για τη βελτίωση της υγείας και της απόδοσης της θεραπείας, καθώς και για την ανάπτυξη μελλοντικών ερευνητικών προγραμμάτων. Έτσι, η συλλογή δεδομένων μέσω ερωτηματολογίων αποτελεί σημαντικό εργαλείο για την προώθηση της επιστημονικής έρευνας και την προαγωγή της υγείας.

(Γαλάνης, 2022)

ΚΕΦΑΛΑΙΟ 2

2.1 DJANGO, Web Πλαίσιο

Το Django είναι ένα υψηλού επιπέδου framework ανάπτυξης ιστοσελίδων στην Python, που επιτρέπει την γρήγορη ανάπτυξη ασφαλών και συντηρήσιμων ιστότοπων. Δημιουργημένο από έμπειρους προγραμματιστές, το Django αναλαμβάνει πολλές από τις δυσκολίες της ανάπτυξης ιστοσελίδων. Είναι ελεύθερο και ανοιχτού κώδικα, έχει μια ενεργή και ευεργετική κοινότητα, εξαιρετική τεκμηρίωση και πολλές επιλογές για δωρεάν και πληρωμένη υποστήριξη. Ακολουθεί το μοντέλο αρχιτεκτονικής Model-View-Controller (MVC), αλλά χρησιμοποιεί το δικό του σύστημα που ονομάζεται Model-View-Template (MVT). Παρέχει ένα πλήρες σύνολο λειτουργιών για τη διαχείριση των δεδομένων, τον χειρισμό των URL, τη διαχείριση των συνεδριών, την αυθεντικοποίηση χρηστών και πολλά άλλα.

Μπορεί να χρησιμοποιηθεί για τη δημιουργία σχεδόν οποιουδήποτε είδους ιστοσελίδας, από συστήματα διαχείρισης περιεχομένου, μέχρι κοινωνικά δίκτυα και ιστοτόπους ειδήσεων. Μπορεί να συνεργαστεί με οποιοδήποτε framework στην πλευρά του πελάτη και μπορεί να παρέχει περιεχόμενο σε σχεδόν οποιοδήποτε μορφότυπο (συμπεριλαμβανομένων HTML, RSS, JSON και XML).

Εσωτερικά, ενώ παρέχει επιλογές για σχεδόν κάθε λειτουργικότητα που θα μπορούσαμε να θέλουμε μπορεί επίσης να επεκταθεί για να χρησιμοποιεί άλλα στοιχεία αν απαιτείται.

Βοηθά τους προγραμματιστές να αποφύγουν πολλά συνηθισμένα λάθη ασφαλείας παρέχοντας ένα πλαίσιο που έχει σχεδιαστεί για να προστατεύει αυτόματα τον ιστότοπο και να παρέχει έναν ασφαλή τρόπο διαχείρισης των λογαριασμών χρηστών και των κωδικών πρόσβασης¹.

2.2 Python, Γλώσσα Προγραμματισμού

Η Python είναι διερμηνευόμενη, γενικού σκοπού (general-purpose) και υψηλού επιπέδου, γλώσσα προγραμματισμού. Ανήκει στις γλώσσες προστακτικού προγραμματισμού (Imperative programming) και υποστηρίζει τόσο το διαδικαστικό (procedural programming) όσο και το αντικειμενοστρεφές (object-oriented programming) προγραμματιστικό υπόδειγμα. Δημιουργήθηκε από τον Ολλανδό Γκίντο βαν Ρόσσουμ (Guido van Rossum) στο ερευνητικό κέντρο Centrum Wiskunde & Informatica (CWI) το 1989 και κυκλοφόρησε για πρώτη φορά το 1991.

¹ Jaiswal, 2015

Ο κύριος στόχος της είναι η αναγνωσιμότητα του κώδικά της και η ευκολία χρήσης της. Το συντακτικό της επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα από ό,τι θα ήταν δυνατόν σε γλώσσες όπως η C++ ή η Java. Διακρίνεται λόγω του ότι έχει πολλές βιβλιοθήκες που διευκολύνουν ιδιαίτερα αρκετές συνηθισμένες εργασίες και για την ταχύτητα εκμάθησής της. Μειονεκτεί στο ότι επειδή είναι διερμηνευόμενη είναι πιο αργή από τις μεταγλωττιζόμενες (compiled) γλώσσες όπως η C και η C++. Για αυτόν τον λόγο δεν είναι κατάλληλη για γραφή λειτουργικών συστημάτων.

Η Python αναπτύσσεται ως ανοιχτό λογισμικό (open source) και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Ο κώδικας διανέμεται με την άδεια Python Software Foundation License η οποία είναι συμβατή με την GPL. Το όνομα της γλώσσας προέρχεται από την ομάδα των Άγγλων κωμικών Μόντυ Πάιθον και δεν έχει καμιά σχέση με το φίδι πύθωνα, παρότι το λογότυπό της παραπέμπει σε κάτι τέτοιο².

2.3 Hypertext Markup Language

Η HTML είναι ένα σύνολο κανόνων για την διαμόρφωση της εμφάνισης και του περιεχομένου μιας ιστοσελίδας. Ουσιαστικά, δεν είναι γλώσσα προγραμματισμού, αλλά γλώσσα περιγραφής ιδιοτήτων των στοιχείων που αποτελούν μία ιστοσελίδα. Χρησιμοποιώντας ειδικές ετικέτες (tags) δίνονται εντολές, για το πως να διαχειριστεί ένα έγγραφο html το περιεχόμενό του και πως να το εμφανίσει στον χρήστη ένας web browser, το πρόγραμμα δηλαδή που χρησιμοποιούμε για να προβάλουμε τις διάφορες σελίδες στο διαδίκτυο.

Ενδεικτικά, οι εντολές / tags της HTML, μπορούν:

- Να εισάγουν σε μία ιστοσελίδα links (συνδέσμους)
- Να εισάγουν σε μία ιστοσελίδα εικόνες
- Να διαμορφώσουν το κείμενο με έντονα ή πλάγια γράμματα κλπ

Για να δημιουργηθεί ένα αρχείο html, αρκεί ένα αρχείο απλού κειμένου, το οποίο θα έχει κατάληξη .html ή .htm και το αρχείο αυτό να περιέχει τις επιθυμητές εντολές με τις ανάλογες παραμέτρους τους.

Η HTML μπορεί να γραφτεί απευθείας ως κώδικας (πηγαίος κώδικας) ή μπορεί να παραχθεί αυτόματα από κάποιο πρόγραμμα κατασκευής ιστοσελίδων (υπάρχουν πολλά διαθέσιμα open source προγράμματα που μπορεί να κατεβάσει κανείς και να εγκαταστήσει στο

² Guttag, 2022

μηχάνημά του) την στιγμή που ο δημιουργός, σε γραφικό περιβάλλον, απλά χρησιμοποιεί τα διάφορα εργαλεία που του παρέχει το πρόγραμμα για να το σχεδιασμό της σελίδας του³.

2.4 Cascading Style Sheets

Η CSS είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων ύφους που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html⁴.

³ Duckett, 2011

⁴ Portilla, 2019

ΚΕΦΑΛΑΙΟ 3

3.1 Εφαρμογή

Ανέπτυξα ένα online πληροφοριακό σύστημα σχεδιασμένο για γιατρούς που διαχειρίζονται τη φροντίδα των ασθενών. Το σύστημα διευκολύνει την άνετη καταγραφή και διαχείριση των πληροφοριών των ασθενών. Επιπλέον, περιλαμβάνει ένα ενσωματωμένο ερωτηματολόγιο που επιτρέπει στους ασθενείς να απαντήσουν μέσω του γιατρού σε ερωτήσεις που είναι κρίσιμες για την ευημερία τους. Αυτή η συστηματική προσέγγιση συμβάλλει στην εξατομίκευση των υπηρεσιών υγείας, εξουσιοδοτώντας τους γιατρούς να παρέχουν βελτιωμένη φροντίδα.

Για την υλοποίηση χρησιμοποίησα το Linux Ubuntu 22.04 ως λειτουργικό σύστημα για το περιβάλλον ανάπτυξης. Αξιοποίησα τη διαχείριση πακέτων της Conda στην έκδοση 22.9.0 για να διαχειριστώ τις εξαρτήσεις του έργου μου. Επέλεξα να χρησιμοποιήσω την έκδοση 3.10.6 της Python ως γλώσσα προγραμματισμού για την υλοποίηση της λογικής της εφαρμογής.

Για την ανάπτυξη του web πλαισίου, επέλεξα το Django στην έκδοση 4.1. Αυτό μου προσέφερε ένα ισχυρό πλαίσιο εργασίας που επιτρέπει εύκολη διαχείριση της βάσης δεδομένων, δρομολόγηση URL, και ανάπτυξη εφαρμογών με στοχευμένες λειτουργίες.

Για τον επεξεργαστή κειμένου, επέλεξα το Atom, έναν ελαφρύ και επεκτάσιμο προγραμματιστικό editor που προσφέρει ευελιξία στον κώδικα.

Για τη διαμόρφωση και την εμφάνιση των σελίδων, χρησιμοποίησα γλώσσες όπως το HTML για τη διάταξη της σελίδας, το CSS για τον σχεδιασμό και τη μορφοποίηση, και το JavaScript για διαδραστικές λειτουργίες. Χρησιμοποίησα το Bootstrap στην έκδοση 4.6 για τον σχεδιασμό του frontend της εφαρμογής μου, προσφέροντας ανταπόκριση και στυλ στη διεπαφή χρήστη.

Με αυτά τα εργαλεία και τεχνολογίες, κατάφερα να αναπτύξω μια πλήρη και λειτουργική εφαρμογή, συνδυάζοντας αποτελεσματικά την ευκολία χρήσης, την αξιοπιστία και τον σύγχρονο σχεδιασμό.

3.2 Βασικά βήματα για τη δημιουργία ενός νέου django project και ενός django app.

Ανοίγω το command line του υπολογιστή μου και εκτελώ την παρακάτω εντολή για την εγκατάσταση του django:

- `pip install django`

Δημιουργία και ενεργοποίηση ενός νέου περιβάλλοντος Django:

- `conda create –MyEnv django`
- `conda activate MyEnv`

Δημιουργία ενός νέου φακέλου git και μετάβαση σε αυτόν:

- `mkdir git`
- `cd git`

Δημιουργία του django project και μεταβαση σε αυτο:

- `django-admin startproject surveyformproject`
- `cd surveyformproject`

Εφαρμογή μετατροπών βάσης δεδομένων:

- `python manage.py migrate`

Δημιουργία Django App:

- `python manage.py startapp poll`

Εκκίνηση του development server:

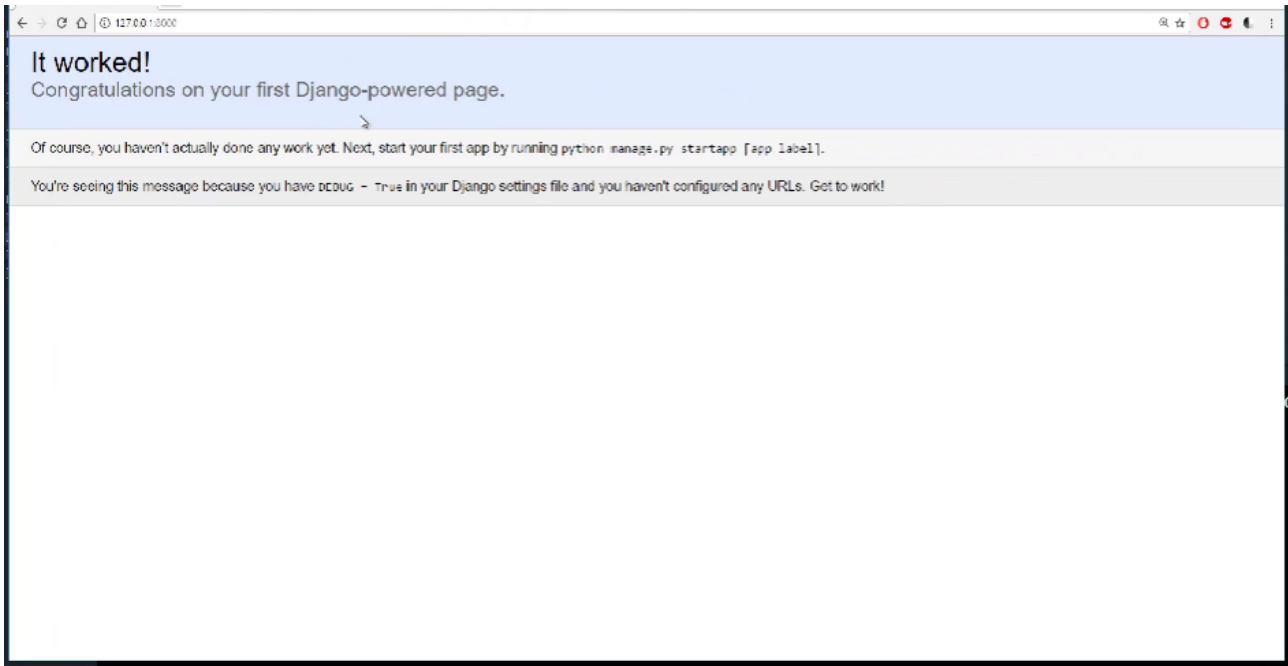
- `python manage.py runserver`

```
(MyEnv) p13papa1@p13papa1-VirtualBox:~/git/surveyformproject$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

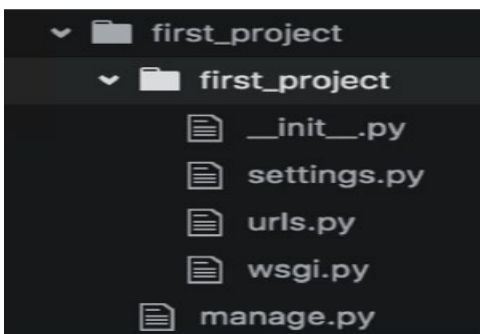
System check identified no issues (0 silenced).
January 16, 2024 - 16:46:02
Django version 4.1, using settings 'PollProject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Μόλις εκτελέσουμε την παραπάνω εντολή μας δίνετε η διεύθυνση <http://127.0.0.1:8000/> για να δούμε την αρχική σελίδα του django η οποία όμως θα έχει μια απλοϊκή μορφή επειδή δεν την έχουμε αναπτύξει ακόμα.

Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.



Μετά την εκτέλεση της εντολής `django-admin startproject firstproject` θα δημιουργηθεί ένας νέος κατάλογος με το όνομα `firstproject` που θα περιέχει τη δομή ενός βασικού django-project και θα έχει την παρακάτω μορφή:



Ας δούμε έναν έναν τους υποφακέλους που δημιουργήθηκαν:

- `__init__.py`

Αυτό είναι ένα κενό Python script που, λόγω του ειδικού του ονόματος, επιτρέπει στο Python να γνωρίζει ότι αυτός ο κατάλογος μπορεί να θεωρηθεί ως ένα πακέτο (package). Ο σκοπός του `__init__.py` είναι να εκτελείται κατά την εισαγωγή του πακέτου ή της εφαρμογής και να προβαίνει σε απαραίτητες αρχικοποιήσεις. Σε περιπτώσεις καταλόγων που θεωρούνται πακέτα, η παρουσία του `__init__.py` είναι απαραίτητη για να ενημερωθεί η Python ότι πρόκειται για πακέτο. Συνεπώς, μπορούμε να πούμε ότι είναι ένα κλειδί για τον τρόπο με τον οποίο οργανώνονται και αναγνωρίζονται πακέτα στην Python.

- `Settings.py`

Εδώ αποθηκεύουμε όλες τις ρυθμίσεις του project μας. Στην πορεία ανάπτυξης της εφαρμογής μας θα χρειαστεί να μετάβουμε πολλές φορές στο αρχείο `settings.py` για να αποθηκεύσουμε νέες ρυθμίσεις.

- `Urls.py`

Αυτό είναι ένα Python script που θα αποθηκεύει όλα τα μοτίβα URL για το project μας. Πρόκειται για τις διάφορες σελίδες της εφαρμογής μας στον ιστό.

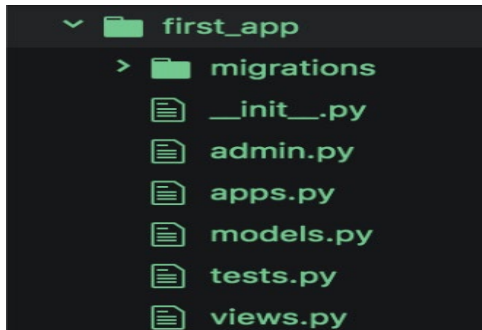
- `Wsgi.py`

Αυτό είναι ένα Python script που λειτουργεί ως διεπαφή (interface) του διακομιστή (Web Server Gateway Interface - WSGI). Αργότερα, θα μας βοηθήσει στο να αναπτύξουμε την εφαρμογή μας σε παραγωγικό περιβάλλον.

- `Manage.py`

Αυτό είναι ένα Python script που θα χρησιμοποιούμε συχνά. Θα συσχετίζεται με πολλές εντολές καθώς χτίζουμε την εφαρμογή μας στον ιστό.

Αφού έχουμε μεταβεί στον κατάλογο εκτελούμε την εντολή `python manage.py startapp first_app` για να δημιουργηθεί μια αρχική δομή της εφαρμογής μας μέσα στο project η οποία έχει παρόμοια μορφή με λίγες διαφορές με τον κατάλογο του project μας.



- `__init__.py`

Αυτό είναι ένα κενό Python script που, λόγω του ειδικού του ονόματος, επιτρέπει στο Python να γνωρίζει ότι αυτός ο κατάλογος μπορεί να θεωρηθεί ως ένα πακέτο (package).

Ο σκοπός του `__init__.py` είναι να εκτελείται κατά την εισαγωγή του πακέτου ή της εφαρμογής και να προβαίνει σε απαραίτητες αρχικοποιήσεις.

Σε περιπτώσεις καταλόγων που θεωρούνται πακέτα, η παρουσία του `__init__.py` είναι απαραίτητη για να ενημερωθεί η Python ότι πρόκειται για πακέτο. Συνεπώς, μπορούμε να πούμε ότι είναι ένα κλειδί για τον τρόπο με τον οποίο οργανώνονται και αναγνωρίζονται πακέτα στην Python.

- `Admin.py`

Μπορείτε να εγγράψετε τα μοντέλα σας εδώ, τα οποία θα χρησιμοποιηθούν από το Django στη διαχειριστική διεπαφή (admin interface) του Django.

- `Apps.py`

Εδώ μπορείτε να τοποθετήσετε τις συγκεκριμένες ρυθμίσεις της εφαρμογής.

- `Models.py`

Εδώ αποθηκεύετε τα μοντέλα δεδομένων της εφαρμογής.

- Tests.py

Εδώ μπορείτε να αποθηκεύσετε συναρτήσεις δοκιμής για να ελέγξετε τον κώδικά σας.

- Views.py

Εδώ βρίσκονται οι συναρτήσεις που χειρίζονται αιτήσεις και επιστρέφουν απαντήσεις.

- Migration folder

Αυτός ο κατάλογος αποθηκεύει πληροφορίες που αφορούν τη βάση δεδομένων, σε σχέση με τα μοντέλα.

- Forms.py

Ένα αρχείο που θα δημιουργήσουμε εμείς, η χρήση του προσφέρει ορισμένα πλεονεκτήματα και διευκολύνσεις στη διαχείριση της λειτουργικότητας που σχετίζεται με την είσοδο δεδομένων από τον χρήστη.

Πριν αρχίσουμε να προγραμματίζουμε την εφαρμογή μας χρειάζεται να την δηλώσουμε στις αποθηκευμένες εφαρμογές που έχει ήδη το project μας και βρίσκονται στο αρχείο settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'first_app'  
]
```

3.3 MODELS.PY

Το αρχείο models.py αποτελεί τον χώρο όπου καθορίζονται τα μοντέλα δεδομένων, που αντιπροσωπεύουν τη δομή της βάσης δεδομένων της εφαρμογής. Τα μοντέλα αυτά παρέχουν μια αφηγηματική αναπαράσταση των δεδομένων που αποθηκεύονται και ανακτώνται από τη βάση δεδομένων. Κάθε μοντέλο αντιστοιχεί σε έναν πίνακα της βάσης δεδομένων, και κάθε πεδίο του μοντέλου αντιστοιχεί σε μια στήλη στον πίνακα. Μέσω των μοντέλων, καθορίζονται οι τύποι δεδομένων, οι σχέσεις με άλλα μοντέλα, και οι κανόνες διαχείρισης των δεδομένων. Το παρακάτω αρχείο models.py ασχολείται με πληροφορίες χρήστη, δεδομένα ασθενών, ερωτήσεις και απαντήσεις.

```
from django.db import models
from django.contrib.auth.models import User

# Create your models here.
class UserProfileInfo(models.Model):

    user = models.OneToOneField(User, on_delete=models.CASCADE)

    def __str__(self):
        return self.user.username

class Patient(models.Model):

    IDENTITY = models.CharField(max_length=10, null=True)
    EDUCCATEGORY = models.CharField(max_length=100, null=True)
    EDUCPATIENT = models.CharField(max_length=100, null=True)
    SEXPATIENT = models.CharField(max_length=100, null=True)
    AGEPATIENT = models.CharField(max_length=10, null=True)
    ECUCCARERCAT = models.CharField(max_length=100, null=True)
    EDUCCARER = models.CharField(max_length=100, null=True)
    SEXCARER = models.CharField(max_length=100, null=True)
    AGECARER = models.CharField(max_length=100, null=True)
    RELATIONSHIPCARER = models.CharField(max_length=100, null=True)
    LIVESIN = models.CharField(max_length=100, null=True)
    kentro = models.CharField(max_length=100, null=True)
    date = models.CharField(max_length=100, null=True)
    NPI = models.CharField(max_length=100, null=True)
    STAGE = models.CharField(max_length=100, null=True)
    IADL = models.CharField(max_length=100, null=True)
    MMSE = models.CharField(max_length=100, null=True)
    DOCTOR = models.ForeignKey(User, on_delete=models.CASCADE, null=True)

    def __str__(self):
        return f"{self.IDENTITY}"

class Question(models.Model):
    question_text = models.CharField(max_length=750)

    def __str__(self):
        return self.question_text

class Choice(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    patient = models.ForeignKey(Patient, on_delete=models.CASCADE, null=True)
    notes = models.CharField(max_length=200, null=True)
    first_answer = models.CharField(max_length=10)
    second_answer = models.CharField(max_length=10, null=True)

    def __str__(self):
        return f"{self.patient} answered question {self.question.id} : {self.first_answer}"
```

Ας αναλύσουμε τι κάνει κάθε μια από τις κλάσεις

```
# Create your models here.
class UserProfileInfo(models.Model):

    user = models.OneToOneField(User, on_delete=models.CASCADE)

    def __str__(self):
        return self.user.username
```

Η κλάση UserProfileInfo συσχετίζεται με το ενσωματωμένο μοντέλο χρήστη (User) του django χρησιμοποιώντας έναν δείκτη OneToOneField διασφαλίζοντας ότι κάθε χρήστης έχει ένα μοναδικό προφίλ.

```
class Patient(models.Model):

    IDENTITY = models.CharField(max_length=10, null=True)
    EDUCATEGORY = models.CharField(max_length=100, null=True)
    EDUCPATIENT = models.CharField(max_length=100, null=True)
    SEXPATIENT = models.CharField(max_length=100, null=True)
    AGPATIENT = models.CharField(max_length=10, null=True)
    ECUCARERCAT = models.CharField(max_length=100, null=True)
    EDUCARER = models.CharField(max_length=100, null=True)
    SEXCARER = models.CharField(max_length=100, null=True)
    AGECARER = models.CharField(max_length=100, null=True)
    RELATIONSHIPCARER = models.CharField(max_length=100, null=True)
    LIVESIN = models.CharField(max_length=100, null=True)
    kentro = models.CharField(max_length=100, null=True)
    date = models.CharField(max_length=100, null=True)
    NPI = models.CharField(max_length=100, null=True)
    STAGE = models.CharField(max_length=100, null=True)
    IADL = models.CharField(max_length=100, null=True)
    MMSE = models.CharField(max_length=100, null=True)
    DOCTOR = models.ForeignKey(User, on_delete=models.CASCADE, null=True)

    def __str__(self):
        return f"{self.IDENTITY}"
```

Η κλάση Patient περιγράφει χαρακτηριστικά και πληροφορίες που σχετίζονται με τον ασθενή όπως το όνομα, την εκπαίδευση, την ηλικία, το φύλο του ασθενή κ.ά.

Το πεδίο doctor είναι ένα πεδίο εξωτερικού κλειδιού (ForeignKey) που συσχετίζει το μοντέλο Patient με το μοντέλο User. Η παράμετρος on_delete=models.CASCADE σημαίνει ότι όταν ο χρήστης διαγραφεί, οι ασθενείς που του ανήκουν θα διαγραφούν επίσης. Η μέθοδος __str__ καθορίζει πώς θα εμφανίζεται ένα αντικείμενο της κλάσης όταν εκτυπώνεται. Στη συγκεκριμένη περίπτωση, επιστρέφει την τιμή του πεδίου IDENTITY ως αναπαράσταση του αντικειμένου.

3.4 FORMS.PY

```
from django import forms
from django.contrib.auth.models import User

class UserForm(forms.ModelForm):
    password = forms.CharField(widget=forms.PasswordInput())

    class Meta():
        model = User
        fields = ('username', 'email', 'password')
```

Το παραπάνω τμήμα κώδικα δημιουργεί μια κλάση με την ονομασία `UserForm`, η οποία σχετίζεται με το ενσωματωμένο μοντέλο χρήστη (`User`) από το `django.contrib.auth.models`. Το form περιλαμβάνει πεδία για το όνομα χρήστη (`username`), το email και τον κωδικό πρόσβασης (`password`). Το πεδίο του κωδικού πρόσβασης χρησιμοποιεί ένα widget που αποκρύπτει τους χαρακτήρες κατά την εισαγωγή. Επίσης, με τη χρήση της κλάσης `Meta`, ορίζονται οι ρυθμίσεις για τη σύνδεση με το μοντέλο και τα πεδία που περιλαμβάνονται στη φόρμα.

3.5 VIEWS.PY

Το αρχείο `views.py` περιέχει τις συναρτήσεις προβολής που αντιστοιχούν σε διάφορα URL patterns. Οι προβολές είναι υπεύθυνες για την ανταπόκριση σε αιτήματα HTTP, επεξεργασία δεδομένων και επιστροφή απαντήσεων. Κάθε συνάρτηση προβολής αντιστοιχεί σε ένα συγκεκριμένο URL και όταν ένα αίτημα φτάσει στην εφαρμογή, το σύστημα δρομολόγησης καλεί την κατάλληλη προβολή για να εκτελέσει την επιθυμητή λειτουργία. Το `views.py` περιέχει τη λογική της εφαρμογής μας, όπως την αποθήκευση και ανάκτηση δεδομένων από τη βάση δεδομένων, ελέγχει την πρόσβαση των χρηστών, διαχειρίζεται τις φόρμες υποβολής δεδομένων, την παραγωγή html ή άλλων τύπων και πολλά άλλα. Θα χρησιμοποιήσουμε αρκετές διαφορετικές προβολές για να καλύψουμε διάφορες λειτουργίες της εφαρμογής.

```
from django.shortcuts import render, redirect, get_object_or_404
from poll.forms import UserForm, UserProfileInfoForm

from django.contrib.auth import authenticate, login, logout
from django.http import HttpResponseRedirect, HttpResponse
from django.urls import reverse
from django.contrib.auth.decorators import login_required
from .models import Question, Choice, Patient
import csv
from datetime import datetime
```

Στην αρχή εισάγουμε τα απαραίτητα πακέτα και εξαρτήσεις. Στη δεύτερη σειρά έχουμε δημιουργήσει ένα φάκελο forms με τα αρχεία UserForm και UserProfileInfoForm που αποτελούν μοναδική χρήση του αρχείου forms.py για την δημιουργία του χρήστη όπως θα εξηγήσουμε παρακάτω.

3.6 Συναρτήσεις διαχείρισης χρηστών

```
def index(request):
    return render(request, 'poll/index.html')
```

```
{% extends "poll/base.html" %}
{% block body_block %}

<div class="jumbotron alert alert-info">
    <h1>Κέντρο Ημέρας</h1>

</div>

{% endblock %}
```

```
{% load static %}
```

Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.

```
<body class="bg-info">
  <nav class="navbar navbar-light" style="background-color: #e3f2fd;">
    <div class="container-fluid">
      <ul class="nav navbar-nav navbar-expand-lg navbar-light">
        <li><a class='navbar-brand' href="{% url 'index' %}">Κέντρο Ημέρας</a></li>
        <li><a class="navbar-link" href="{% url 'admin:index' %}">Διαχειριστής</a></li>
        <li><a class="navbar-link" href="{% url 'poll:register' %}">Εγγραφή</a></li>
        <li><a class="navbar-link" href="{% url 'poll:questions' %}">Εγγραφή νέου ασθενούς και συμπλήρωση ερωτηματολογίου</a></li>
        <li><a class="navbar-link" href="{% url 'poll:saved_patients' %}">Εγγεγραμμένοι Ασθενείς</a></li>

        {% if user.is_authenticated %}
        <li><a class="navbar-link" href="{% url 'logout' %}">Έξοδος από το σύστημα</a></li>

        {% else %}
        <li><a class="navbar-link" href="{% url 'poll:user_login' %}">Είσοδος στο σύστημα</a></li>

        {% endif %}

      </ul>
    </div>

  </nav>
  <div class="container">

    {% block body_block %}
    {% endblock %}

  </div>

</body>
```

Η συνάρτηση index απλώς επιστρέφει το html αρχείο της αρχικής σελίδας κάνοντας χρήση των template tagging μπορούμε να επεκτείνουμε τα στοιχεία ενός html αρχείου σε όσα θέλουμε. Στην παραπάνω περίπτωση έχω ένα αρχείο base.html όπου έχω διαμορφώσει το navbar με τις επιλογές που έχει να περιληφθεί ο χρήστης. Χρησιμοποιώντας το tag {% load static %} στην αρχή του κώδικα και κλείνοντας στο τέλος με το {% block body_block %} και {% endblock %} μου δίνεται η δυνατότητα να επεκτείνω τα στοιχεία του base.html και σε άλλα html αρχεία.

Στο αρχείο index.html έχω βάλει πάλι αγκύλες όπου κάνω επέκταση του base.html.

```
{% extends "poll/base.html" %}
{% block body_block %}

<div class="jumbotron alert alert-info">
  <h1>Κέντρο Ημέρας</h1>

</div>

{% endblock %}
```

Η συνάρτηση `special` παρέχει ένα απλό μήνυμα στους συνδεδεμένους χρήστες. Το `@login_required` το πρόσθεσα ακριβώς πάνω από πολλές συναρτήσεις ώστε να ειδοποιεί ότι πρέπει να συνδεθεί ο χρήστης πριν αρχίσει να περιηγείται στην ιστοσελίδα της εφαρμογής.

```
@login_required
def special(request):
    return HttpResponseRedirect("you are logged in , nice!")
```

Η συνάρτηση `user_logout` αποσυνδέει τον χρήστη και τον κατευθύνει στην αρχική σελίδα.

```
@login_required
def user_logout(request):
    logout(request)
    return HttpResponseRedirect(reverse('index'))
```

Η κλάση `UserForm` στο αρχείο `forms.py` περιλαμβάνει πεδία όπως `username`, `email`, `password` και αντιστοιχεί στα στοιχεία σύνδεσης του χρήστη.

```
from django import forms
from django.contrib.auth.models import User
from poll.models import UserProfileInfo, Question, Choice

class UserForm(forms.ModelForm):
    password = forms.CharField(widget=forms.PasswordInput())

    class Meta():
        model = User
        fields = ('username', 'email', 'password')
```

Στη συνάρτηση register στο αρχείο views.py, το αντικείμενο user_form δημιουργείται για τη συλλογή, επεξεργασία και αποθήκευση των δεδομένων εισόδου που αφορούν τη σύνδεση του χρήστη.

```
def register(request):  
  
    registered = False  
  
    if request.method == "POST":  
  
        user_form = UserForm(data=request.POST)  
        profile_form = UserProfileInfoForm(data=request.POST)  
  
        if user_form.is_valid():  
  
            user = user_form.save()  
            user.set_password(user.password)  
            user.save()  
  
            registered = True  
  
        else:  
            print(user_form.errors)  
  
    else:  
        user_form = UserForm()  
  
    return render(request, 'poll/registration.html',  
                  {'user_form':user_form,  
                   'registered':registered})
```

Η συνάρτηση User_login διαχειρίζεται τη διαδικασία σύνδεσης του χρήστη.

```
def user_login(request):  
  
    if request.method == 'POST':  
        username = request.POST.get('username')  
        password = request.POST.get('password')  
  
        user = authenticate(username=username,password=password)  
  
        if user:  
            if user.is_active:  
                login(request,user)  
                return HttpResponseRedirect(reverse('index'))  
  
            else:  
                return HttpResponse("ACCOUNT NOT ACTIVE")  
  
        else:  
            print("Someone tried to login and failed!")  
            print("Username: {} and password {}".format(username,password))  
            return HttpResponse("invalid login details supplied")  
  
    else:  
        return render(request,'poll/login.html',{})
```

Αφού λάβει ένα αίτημα POST, εξάγει τα στοιχεία username και password από το αίτημα και τα ελέγχει χρησιμοποιώντας την συνάρτηση authenticate. Αν η ταυτοποίηση είναι επιτυχής και ο λογαριασμός είναι ενεργός, ο χρήστης συνδέεται και κατευθύνεται στην αρχική σελίδα. Σε περίπτωση που ο λογαριασμός δεν είναι ενεργός ή τα στοιχεία σύνδεσης δεν είναι έγκυρα επιστρέφονται αντίστοιχα μηνύματα λάθους.

3.7 Συναρτήσεις διαχείρισης ερωτήσεων

```
@login_required
def questions(request):

    all_questions = Question.objects.all()
    context = {
        'questions': all_questions
    }

    return render(request, 'poll/questions.html', context)
```

Η συνάρτηση questions επιστρέφει όλες τις ερωτήσεις στη σελίδα των ερωτήσεων.

Δημιουργεί ένα αντικείμενο all_questions με όλες τις ερωτήσεις από τη βάση δεδομένων, τις αποθηκεύει σε ένα λεξικό και επιστρέφει αυτό το λεξικό μαζί με την σελίδα html.

```
{% csrf_token %}
<ul class="text-justify">
    {% for question in questions %}
        <li>
            <div class="jumbotron" style="background-color: #ADD8E6; color: #414a7e;">
                {{ question.question_text|safe }}
            <div class="form-check">
                <label class="form-check-label" for="yes">Ναι</label>
                <input onclick="enfunction({{ question.id }})" id="yes" type="radio" name="first_answer_{{ question.id }}" value="Ναι">
                <label class="form-check-label" for="NO">Όχι</label>
                <input onclick="myfunction({{ question.id }})" id="NO" type="radio" name="first_answer_{{ question.id }}" value="Όχι">
            </div>

            <h3>Αν ναι, καλύπτεται η ανάγκη;</h3>

            <div class="form-check">
                <label class="form-check-label" for="No">Όχι</label>
                <input id="No_{{ question.id }}" type="radio" name="second_answer_{{ question.id }}" value="Όχι">
                <label class="form-check-label" for="Medium">Μερικώς</label>
                <input id="Medium_{{ question.id }}" type="radio" name="second_answer_{{ question.id }}" value="Μερικώς">
                <label class="form-check-label" for="Full">Πλήρως</label>
                <input id="Full_{{ question.id }}" type="radio" name="second_answer_{{ question.id }}" value="Πλήρως"><br>
                <label for="textbox">Σημειώσεις:</label>
                <input type="text" id="textbox" name="notes_{{ question.id }}" style="width:80%; height:200px;">
            </div>
        </li>
    <br>
    {% endfor %}
</ul>
<button type="submit"><a class="navbar-link">Submit</a></button>
</form>

{% endblock %}
```

Με τη χρήση template tags μπορούμε να ενσωματώσουμε τα δεδομένα από τη βάση δεδομένων και να τα εμφανίσουμε στη σελίδα html που επιθυμούμε. Η παρακάτω συνάρτηση submit_answers εκτός από τις ερωτήσεις αποθηκεύει και τα στοιχεία του ασθενή.

```
def submit_answers(request):  
  
    if request.method == 'POST':  
        IDENTITY = request.POST.get(f'IDENTITY')  
        EDUCATEGORY = request.POST.get(f'EDUCCATEGORY')  
        EDUCPATIENT = request.POST.get(f'EDUCPATIENT')  
        SEXPATIENT = request.POST.get(f'SEXPATIENT')  
        AGEPATIENT = request.POST.get(f'AGEPATIENT')  
        ECUCARERCAT = request.POST.get(f'ECUCCARERCAT')  
        EDUCARER = request.POST.get(f'EDUCCARER')  
        SEXCARER = request.POST.get(f'SEXCARER')  
        AGECARER = request.POST.get(f'AGECARER')  
        RELATIONSHIPCARER = request.POST.get(f'RELATIONSHIPCARER')  
        LIVESIN = request.POST.get(f'LIVESIN')  
        kentro = request.POST.get(f'kentro')  
        date = request.POST.get(f'date')  
        NPI = request.POST.get(f'NPI')  
        STAGE = request.POST.get(f'STAGE')  
        IADL = request.POST.get(f'IADL')  
        MMSE = request.POST.get(f'MMSE')  
  
        patient = Patient(  
            DOCTOR = request.user,  
            IDENTITY = IDENTITY,  
            EDUCATEGORY = EDUCCATEGORY,  
            EDUCPATIENT = EDUCPATIENT,  
            SEXPATIENT = SEXPATIENT,  
            AGEPATIENT = AGEPATIENT,  
            ECUCARERCAT = ECUCARERCAT,  
            EDUCARER = EDUCARER,  
            SEXCARER = SEXCARER,  
            AGECARER = AGECARER,  
            RELATIONSHIPCARER = RELATIONSHIPCARER,  
            LIVESIN = LIVESIN,  
            kentro = kentro,  
            date = date,  
            NPI = NPI,  
            STAGE = STAGE,  
            IADL = IADL,  
            MMSE = MMSE,  
        )  
        patient.save()
```

```
questions = Question.objects.all()

for question in questions:
    first_answer = request.POST.get(f'first_answer_{question.id}')
    second_answer = request.POST.get(f'second_answer_{question.id}')
    notes = request.POST.get(f'notes_{question.id}')

    if first_answer:
        choice = Choice(
            question=question,
            user=request.user,
            patient=patient,
            first_answer=first_answer,
            second_answer=second_answer,
            notes=notes
        )
        choice.save()
        print('saved')

return render(request, 'poll/submit_answers.html')
```

Η συνάρτηση submit_answers ανακτά τα στοιχεία του ασθενούς και του φροντιστή, καθώς και απαντήσεις σε διάφορες ερωτήσεις. Στη συνέχεια δημιουργεί ένα αντικείμενο Patient με τα δεδομένα του ασθενούς και τέλος αποθηκεύει τις απαντήσεις του χρήστη στις ερωτήσεις του ερωτηματολογίου.

Η HTML με τη φόρμα για τη δημιουργία του ασθενούς.

```
<div class="jumbotron border border-success" style="background-color: #ADD8E6; color: #414a7e;">
  <h1>Συμπληρώστε τα στοιχεία του ασθενή</h1>
  <div class="form-group">

    <label for="IDENTITY">Identity:</label>
    <input type="text" class="form-control" id="IDENTITY" name="IDENTITY"><br>

    <label for="EDUCCATEGORY">Category of Education:</label>
    <select class="form-control" type="text" id="EDUCCATEGORY" name="EDUCCATEGORY">
      <option> </option>
      <option>0-6</option>
      <option>7-12</option>
      <option>13+</option>
    </select><br>

    <label for="EDUCPATIENT">Education of Patient:</label>
    <input type="text" class="form-control" id="EDUCPATIENT" name="EDUCPATIENT"><br>

    <label for="SEXPATIENT">Sex of Patient:</label>
    <select class="form-control" type="text" id="SEXPATIENT" name="SEXPATIENT">
      <option> </option>
      <option>Male</option>
      <option>Female</option>
    </select><br>

    <label for="AGEPATIENT">Age of Patient:</label>
    <input type="text" class="form-control" id="AGEPATIENT" name="AGEPATIENT"><br>
```

Η HTML για την επιτυχημένη υποβολή ερωτήσεων και την δημιουργία του ασθενούς.

```
{% extends "poll/base.html" %}
{% block body_block %}

  <h1>Submit Answers Success</h1>

{% endblock %}
```

Η συνάρτηση `saved_patients` εμφανίζει όλους τους ασθενείς που έχει αποθηκεύσει ο χρήστης με την ίδια μεθοδολογία που εμφανίζονται και οι ερωτήσεις.

```
@login_required
def saved_patients(request):
    all_patients = Patient.objects.filter(DOCTOR=request.user)
    context = {
        'patients': all_patients
    }

    return render(request, 'poll/saved_patients.html', context)
```

Η HTML για την απεικόνιση των ασθενών με χρήση template tags.

```
<body class="">

<h1 class="display-4" style="color:#2083CE;">Αποθηκευμένοι ασθενείς</h1>

{% csrf_token %}

<div class="mb-3">
    <form class="" action="{% url 'poll:search_patients' %}" method="get">

        <label for="searchInput" class="form-label">Αναζήτηση Ασθενούς:</label>
        <input type="text" name="IDENTITY" class="form-control" id="searchInput" placeholder="Εισάγετε το όνομα...">
        <button type="submit" class="btn btn-primary">Αναζήτηση</button>
    </form>
</div>
<div class="">
    {% for patient in patients %}

    <ol class="list-group">
        <a class="text-dark" href="{% url 'poll:patient_choices' %}?id={{ patient.id }}">
            <li class="list-group-item list-group-item-action badge rounded-pill">

                {{ patient }}

            </li>
        </a>

    </ol>

    </div>

{% endfor %}
<br>
<a href="{% url 'poll:download_all' %}" class="btn btn-primary"> download>
Download CSV For All
</a>

<h2><a href="{% url 'index' %}">Πίσω στο Κέντρο Ημέρας</a></h2>
```

Η συνάρτηση `search_patients` επιστρέφει τον ασθενή που ταιριάζει με το `IDENTITY` που αναζήτησε ο χρήστης.

```
def search_patients(request):
    search_query = request.GET.get('IDENTITY', '')
    patients = Patient.objects.filter(IDENTITY=search_query)

    return render(request, 'poll/saved_patients.html',{'patients':patients})
```

Η συνάρτηση `download_all` δημιουργεί ένα αρχείο `csv` με όλες τις απαντήσεις των ασθενών του χρήστη.

```
def download_all(request):
    patients = Patient.objects.filter(DOCTOR=request.user)

    csv_filename = 'all_patients'

    response = HttpResponse(content_type='text/csv')
    response['Content-Disposition'] = f'attachment; filename="{csv_filename}"'

    csv_writer = csv.writer(response)
    header_row = ['Identity',
                  'Category of Education',
                  'Education of Patient',
                  'Sex of Patient',
                  'Age of Patient',
                  'Carer category of education',
                  'Education of Carer',
                  'Sex of Carer',
                  'Age of Carer',
                  'Relationship between patient and caregiver',
                  'LIVES IN',
                  'kentro',
                  'date',
                  'NPI',
                  'STAGE',
                  'IADL',
                  'MMSE']

    questions = Question.objects.all()
    for q in questions:
        header_row.extend([f'{q.question_text[:3]} - First Answer', f'- Second Answer', f'- Notes'])
    csv_writer.writerow(header_row)

    for patient in patients:
        patient_choices = Choice.objects.filter(patient=patient)

        row = [patient.IDENTITY,
               patient.EDUCCATEGORY,
               patient.EDUCPATIENT,
               patient.SEXPATIENT,
               patient.AGEPATIENT,
               patient.EDUCCARERCAT,
               patient.EDUCCARER,
               patient.SEXCARER,
               patient.AGECARER,
               patient.RELATIONSHIPCARE,
               patient.LIVESIN,
               patient.kentro,
               patient.date,
               patient.NPI,
               patient.STAGE,
               patient.IADL,
               patient.MMSE]

        for q in questions:
            choice = patient_choices.filter(question=q).first()
            if choice:
                row.extend([choice.first_answer, choice.second_answer, choice.notes])
            else:
                row.extend(['', '', ''])
        csv_writer.writerow(row)

    return response
```

Δημιουργεί δυναμικά ένα αρχείο csv, περιλαμβάνοντας πληροφορίες για κάθε ασθενή που ανήκει σε έναν συγκεκριμένο χρήστη. Ανακτά όλους τους ασθενείς του συγκεκριμένου χρήστη από τη βάση δεδομένων και αποθηκεύει τα στοιχεία τους καθώς και τις απαντήσεις τους στο ερωτηματολόγιο.

Η συνάρτηση `patient_choices` εμφανίζει όλες τις απαντήσεις που έχει δώσει ένας χρήστης για ένα συγκεκριμένο ασθενή.

```
def patient_choices(request):
    patient_id = request.GET.get('id', None)

    if patient_id is not None:
        try:
            patient = Patient.objects.get(id=patient_id)
            patient_choices = Choice.objects.filter(patient=patient)

            context = {
                'patient': patient,
                'patient_choices': patient_choices
            }

            return render(request, 'poll/patient_choices.html', context)
        except Patient.DoesNotExist:
            return render(request, 'poll/saved_patients.html')

    return render(request, 'poll/patient_choices.html')
```

Αν το αίτημα περιέχει την παράμετρο `id` η συνάρτηση προσπαθεί να βρει τον ασθενή στη βάση δεδομένων. Αν βρεθεί ο ασθενής ανακτά όλες τις απαντήσεις που σχετίζονται με αυτόν. Στη συνέχεια αποθηκεύει τα δεδομένα του ασθενούς και τις απαντήσεις του σε ένα λεξικό και τα επιστρέφει για να εμφανιστούν στη σελίδα `patient_choices.html`. Αν ο ασθενής δε βρεθεί τότε επιστρέφει στη σελίδα `saved_patients.html`.

Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.

Η HTML για την απεικόνιση των απαντήσεων του ασθενούς.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Patient Choices</title>
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css" integrity="sha384-738d6QlthSxJ9xj346Pv9DbbdCv0/D3rJEIN+AOsQLIOe2J+ljVxiMifpPsd
```

```
  {% load static %}

  <link rel="stylesheet" type="text/css" href="{% static 'poll/css/style.css' %}">
</head>

<body class="container bg-info">
  {% csrf_token %}
  <h1 class="text-white">Απαντήσεις του Ασθενούς {{ patient.IDENTITY }}</h1>

  {% if patient.choices %}
    <table class="table table-striped text-white" border="1">
      <thead>
        <tr>
          <th>Ερώτηση</th>
          <th>Πρώτη Απάντηση</th>
          <th>Δεύτερη Απάντηση</th>
          <th>Σημειώσεις</th>
        </tr>
      </thead>
      <tbody>
        {% for choice in patient.choices %}
          <tr>
            <td>{{ choice.question.question_text|safe }}</td>
            <td>{{ choice.first_answer }}</td>
            <td>{{ choice.second_answer }}</td>
            <td>{{ choice.notes }}</td>
          </tr>
        {% endfor %}
      </tbody>
    </table>
    <a href="{% url 'poll:download_patient_choices_csv' patient.id %}" class="btn btn-primary" download>
      Download CSV
    </a>

    <a class="btn btn-primary" href="{% url 'poll:del_patient' %}?id={{ patient.id }}">διαγραφή ασθενή</a>
  {% else %}
    <p>0 ασθενής δεν έχει δώσει ακόμη απαντήσεις.</p>
  {% endif %}
</body>
</html>
```

Η συνάρτηση `download_patient_choices_csv` δημιουργεί ένα αρχείο `csv` με τις απαντήσεις ενός συγκεκριμένου ασθενούς.

```
def download_patient_choices_csv(request, patient_id):
    patient = get_object_or_404(Patient, id=patient_id)
    patient_choices = Choice.objects.filter(patient=patient)

    csv_filename = f'patient_{patient.id}_choices.csv'

    response = HttpResponse(content_type='text/csv')
    response['Content-Disposition'] = f'attachment; filename="{csv_filename}"'

    csv_writer = csv.writer(response)
    csv_writer.writerow(['Question', 'First Answer', 'Second Answer', 'Notes'])

    for choice in patient_choices:
        csv_writer.writerow([
            choice.question.question_text[:3],
            choice.first_answer,
            choice.second_answer,
            choice.notes,
        ])

    return response
```

Ανακτά τον ασθενή με βάση το `patient_id` από τη βάση δεδομένων. Στη συνέχεια ανακτά όλες τις απαντήσεις που έχει δώσει ο χρήστης για τον συγκεκριμένο ασθενή και δημιουργεί ένα αρχείο `csv` όπου τις αποθηκεύει.

Η συνάρτηση `del_patient` διαχειρίζεται τη διαδικασία διαγραφής του ασθενούς.

```
def del_patient(request):  
    patient_id = request.GET.get('id', None)  
  
    if patient_id is not None:  
        patient = Patient.objects.get(id=patient_id)  
        patient_choices = Choice.objects.filter(patient=patient)  
  
        patient.delete()  
        patient_choices.delete()  
  
        all_patients = Patient.objects.filter(DOCTOR=request.user)  
        context = {  
            'patients': all_patients  
        }  
  
        return render(request, 'poll/saved_patients.html', context)  
  
    return render(request, 'poll/saved_patients.html')
```

Λαμβάνει το αναγνωριστικό `id` του ασθενούς, ελέγχει αν έχει περαστεί και αν ναι προχωρά στη διαγραφή του ασθενούς και των σχετικών επιλόγων του. Κατά τη διαγραφή, ανακτά τον ασθενή από τη βάση δεδομένων και όλες τις απαντήσεις που έχει δώσει και τις διαγράφει. Μετά τη διαγραφή ενημερώνει τη λίστα όλων των ασθενών που ανήκουν στον ίδιο χρήστη και επιστρέφει μια ανανεωμένη σελίδα με την με την λίστα των ασθενών. Σε περίπτωση που δεν περάσει το `id` του ασθενή, απλώς επιστρέφει τη σελίδα με τη λίστα των ασθενών.

Η συνάρτηση `update_answers` διαχειρίζεται την ενημέρωση των στοιχείων του ασθενή και των απαντήσεων.

```
def update_answers(request):  
  
    if request.method == 'POST':  
        patient_id = request.POST.get(f'patient_id')  
        patient = get_object_or_404(Patient, pk=patient_id)  
        patient.IDENTITY = request.POST.get(f'IDENTITY')  
        patient.EDUCCATEGORY = request.POST.get(f'EDUCCATEGORY')  
        patient.EDUCPATIENT = request.POST.get(f'EDUCPATIENT')  
        patient.SEXPATIENT = request.POST.get(f'SEXPATIENT')  
        patient.AGEPATIENT = request.POST.get(f'AGEPATIENT')  
        patient.ECUCCARERCAT = request.POST.get(f'ECUCCARERCAT')  
        patient.EDUCCARER = request.POST.get(f'EDUCCARER')  
        patient.SEXCARER = request.POST.get(f'SEXCARER')  
        patient.AGECARER = request.POST.get(f'AGECARER')  
        patient.RELATIONSHIPCARER = request.POST.get(f'RELATIONSHIPCARER')  
        patient.LIVESIN = request.POST.get(f'LIVESIN')  
        patient.kentro = request.POST.get(f'kentro')  
        patient.date = request.POST.get('date')  
        patient.NPI = request.POST.get(f'NPI')  
        patient.STAGE = request.POST.get(f'STAGE')  
        patient.IADL = request.POST.get(f'IADL')  
        patient.MMSE = request.POST.get(f'MMSE')
```

```
for question in Question.objects.all():  
    first_answer = request.POST.get(f'first_answer_{question.id}')  
    second_answer = request.POST.get(f'second_answer_{question.id}')  
    notes = request.POST.get(f'notes_{question.id}')  
  
    if first_answer:  
        choices = Choice.objects.filter(patient=patient, user=request.user, question=question).order_by('-id')  
  
        if choices.exists():  
            choice = choices.first()  
  
            choice.first_answer = first_answer  
            choice.second_answer = second_answer  
            choice.notes = notes  
            choice.save()  
            print('Αντικαταστάθηκαν οι παλιές απαντήσεις')  
  
            continue  
  
        new_choice = Choice(  
            question=question,  
            user=request.user,  
            patient=patient,  
            first_answer=first_answer,  
            second_answer=second_answer,  
            notes=notes  
        )  
        new_choice.save()  
        print('Αποθηκεύτηκε νέα εγγραφή')  
  
return redirect('poll:submit answers')
```

Αρχικά λαμβάνει τα στοιχεία του ασθενούς, χρησιμοποιώντας το `request.POST.get()` για κάθε πεδίο και ενημερώνει τα νέα δεδομένα στη βάση με τη μέθοδο `save()` αντικείμενο του ασθενούς. Στη συνέχεια ελέγχει για κάθε ερώτηση αν υπάρχουν ήδη απαντήσεις στο μοντέλο `choice` για τον συγκεκριμένο ασθενή. Αν υπάρχουν, επιλέγεται η τελευταία απάντηση και ενημερώνονται οι απαντήσεις. Διαφορετικά, καταχωρείται μια νέα απάντηση στο μοντέλο `Choice`. Τέλος η συνάρτηση επιστρέφει τον χρήστη στη σελίδα `submit answers` όπου λαμβάνει μήνυμα επιτυχής ολοκλήρωσης της διαδικασίας.

3.8 URLS.PY

Το αρχείο urls.py χρησιμοποιείται για τον ορισμό των (URL patterns) της εφαρμογής μας.

Η διαχείριση των URL είναι σημαντική για τον προσδιορισμό του τρόπου με τον οποίο οι χρήστες θα προσπελάσουν τις διάφορες σελίδες της εφαρμογής. Ο ρόλος του είναι να συσχετίζει συγκεκριμένα URLs με view functions που θα εκτελεστούν όταν ο χρήστης προσπαθεί να ανοίξει μια συγκεκριμένη διαδρομή. Περιέχει τοπικούς κανόνες δρομολόγησης για την εφαρμογή μας και αυτοί οι κανόνες καθορίζουν ποια προβολή πρέπει να κληθεί για κάθε συγκεκριμένο URL. Παρακάτω έχουμε ορίσει τα διάφορα URL patterns που χρησιμοποιούμε για την εφαρμογή μας

```
from django.contrib import admin
from django.urls import path
from poll import views

#template urls
app_name= 'poll'

urlpatterns=[
    path('admin/', admin.site.urls),
    path('register/',views.register,name='register'),
    path('user_login/',views.user_login,name='user_login'),
    path('special/',views.special,name='special'),
    path('logout/',views.user_logout,name='logout'),
    path('questions/',views.questions,name='questions'),
    path('submit_answers/',views.submit_answers,name='submit_answers'),
    path('user_answers/', views.user_answers, name='user_answers'),
    path('create_patient/',views.create_patient, name='create_patient'),
    path('saved_patients/',views.saved_patients, name='saved_patients'),
    path('patient_choices',views.patient_choices, name='patient_choices'),
    path('demo',views.demo,name='demo'),
    path('search_patients',views.search_patients, name='search_patients'),
    path('download_patient_choices_csv/<int:patient_id>', views.download_patient_choices_csv, name='download_patient_choices_csv'),
    path('download_all', views.download_all, name='download_all'),
    path('del_patient', views.del_patient, name='del_patient')
```

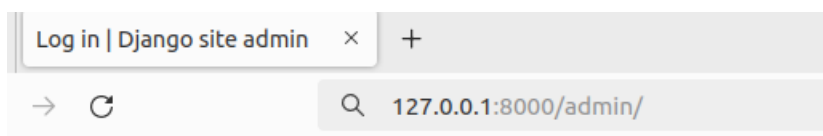
3.9 ADMIN.PY

Το αρχείο `admin.py` χρησιμεύει για τον καθορισμό των διαχειριστικών ρυθμίσεων(administrative settings) για τα μοντέλα της εφαρμογής. Μας επιτρέπει να προσαρμόσουμε τον τρόπο με τον οποίο θα παρουσιάζονται και θα επεξεργάζονται τα δεδομένα από τα μοντέλα στον διαχειριστικό πίνακα(admin interface). Το αρχείο `admin.py` μας δίνει την δυνατότητα να καθορίσουμε διάφορα χαρακτηριστικά, ωστόσο εμείς το χρησιμοποιούμε μόνο για την εγγραφή μοντέλων.

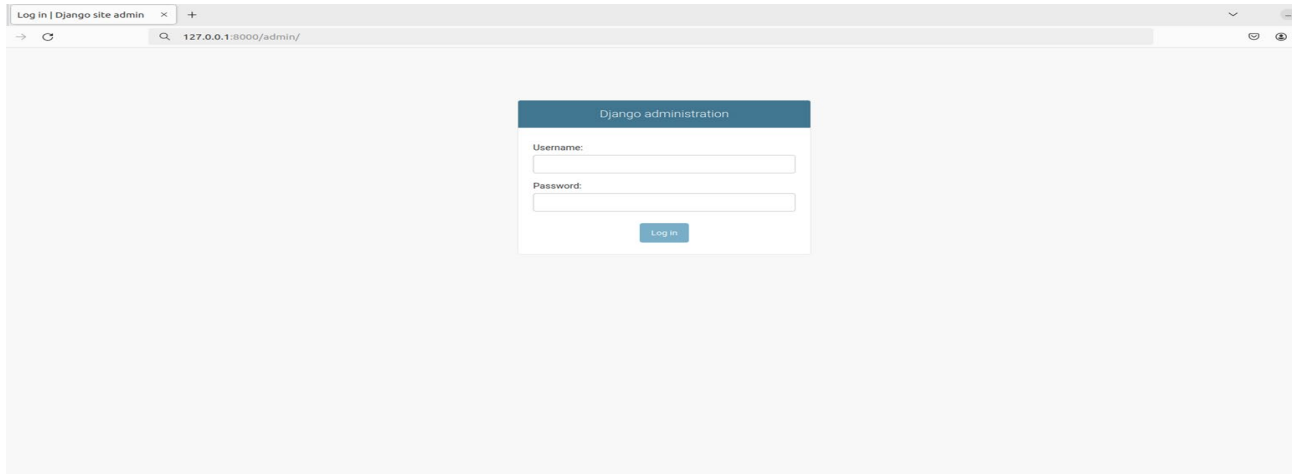
```
from django.contrib import admin
from poll.models import UserProfileInfo, Question, Choice, Patient
# Register your models here.
admin.site.register(UserProfileInfo)
admin.site.register(Question)
admin.site.register(Choice)
admin.site.register(Patient)
```

Με αυτόν το τρόπο εισάγουμε τα μοντέλα μας στο django admin interface. Με τη μέθοδο `register` του αντικείμενου `admin.site` καθορίζουμε ποια μοντέλα θα εμφανίζονται και θα διαχειρίζονται στο django admin interface. Αυτό σημαίνει ότι μπορούμε να προσθέσουμε, να επεξεργαστούμε ή και να διαγράψουμε αντικείμενα των μοντέλων μας.

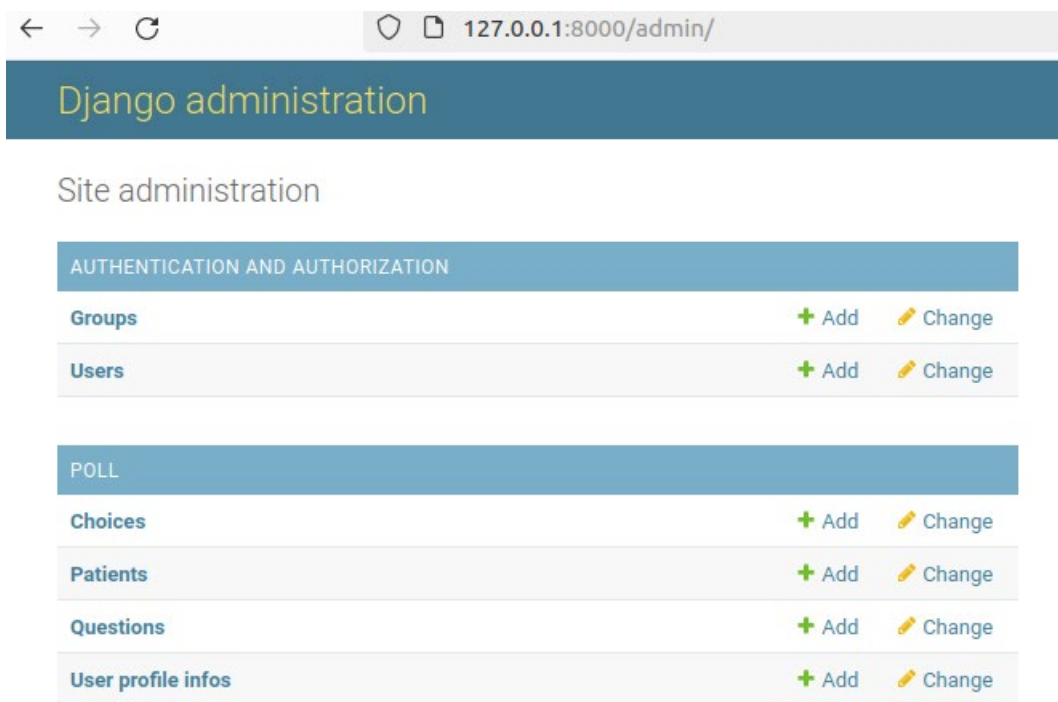
Αφού έχουμε δηλώσει τα μοντέλα μας είμαστε έτοιμοι να δούμε πως εμφανίζονται στον διαχειριστικό πίνακα αλλά πριν από αυτό πρέπει να δημιουργήσουμε έναν `superuser` εκτελώντας την εντολή `python manage.py createsuperuser` στο κελί εντολών (bash shell). Αφού εκτελέσουμε την εντολή αυτή θα μας ζητηθεί να δώσουμε `username`, `email` και `password` ώστε να μπορούμε να μπούμε στον διαχειριστικό πίνακα. Πληκτρολογούμε την εντολή `python manage.py runserver` και στο url που μας δίνετε βάζουμε `/admin/` για να μας πάει στη σελίδα του διαχειριστή.



Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.



Έπειτα βάζουμε το username και τον κωδικό που μας ζητήθηκε και μπορούμε να δούμε τα μοντέλα μας.



Με αυτόν το τρόπο μπορούμε να εισάγουμε τις ερωτήσεις μας, τους ασθενείς, τους χρήστες καθώς και να επεξεργαστούμε ή να διαγράψουμε κάποια στοιχεία. Παρακάτω φαίνονται οι ερωτήσεις που έχουν αποθηκευτεί στη βάση.

Ανάπτυξη πλατφόρμας διαχείρισης ασθενών σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.

Start typing to filter...	
AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add
Users	+ Add
POLL	
Choices	+ Add
Patients	+ Add
Questions	+ Add
User profile infos	+ Add

61 questions

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups

Users

POLL

Choices

Patients

Questions

User profile infos

Change question

A-1 Γνωστικά ελλείμματα μη διαγνωσμένα
• Έχει λάβει διάγνωση για άνοια ο/η εξεταζόμενος/η
ΚΑΙ
• Παρουσιάζει μία ή περισσότερες αλλαγές στις ακόλουθες γνωστικές λειτουργικότητά του/της
ο Απώλεια μνήμης
ο Σύγχυση
ο Αλλαγές στην καθημερινή του/της λειτουργικότητα

Question text:

A-1 Γνωστικά ελλείμματα μη διαγνωσμένα

DELETE

Save and add another

Save and continue editing

SAVE

Πατώντας πάνω στην πρώτη ερώτηση βλέπουμε ότι μπορούμε να την επεξεργαστούμε και να αποθηκεύσουμε τις αλλαγές που κάναμε ή μπορούμε να την διαγράψουμε. Ίδιες δυνατότητες μας δίνονται και με τα υπόλοιπα αντικείμενα των μοντέλων μας οπως να μπορούμε να δούμε τους διάφορους χρήστες μας, τους ασθενείς και τα στοιχεία τους, καθώς και τις απαντήσεις που έχουν δοθεί στο ερωτηματολόγιο.

60

ΚΕΦΑΛΑΙΟ 4

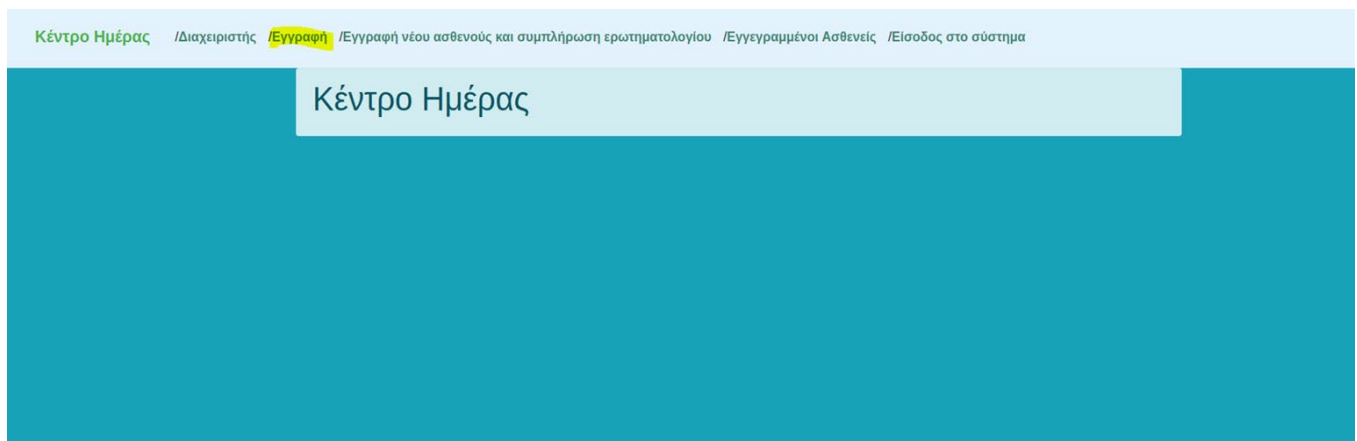
4.1 Παρουσίαση Εφαρμογής : Σύστημα Διαχείρισης Ασθενών

Η ανάπτυξη της εφαρμογής έχει φτάσει σε ένα σημείο όπου παρέχει μια εξαιρετικά ικανοποιητική λύση για τη διαχείριση και τη φροντίδα των ασθενών. Η προσέγγιση μας στη δημιουργία ενός πληροφοριακού συστήματος επικεντρώνεται στην ολοκληρωμένη διαχείριση των ιατρικών δεδομένων, επιτρέποντας στους γιατρούς να παρέχουν βελτιωμένη και εξατομικευμένη φροντίδα.

Παρακάτω υπάρχει ένα απλό παράδειγμα λειτουργίας της εφαρμογής για την καλύτερη κατανόηση από μελλοντικούς χρήστες.

4.1.1 Δημιουργία λογαριασμού

Για να δημιουργήσουμε έναν λογαριασμό, επιλέγουμε την επιλογή 'Εγγραφή' στην αρχική σελίδα της εφαρμογής. Συμπληρώνουμε τα απαραίτητα πεδία όπως όνομα, email και επιθυμητός κωδικός πρόσβασης. Κατόπιν, πατάμε 'Εγγραφή' για να ολοκληρώσουμε τη διαδικασία.



REGISTER HERE

Fill out the form:

Username: Required.

Email address:

Password:

[Εγγραφή](#)

4.1.2 Σύνδεση στο σύστημα

Για να συνδεθούμε στο σύστημα, στην αρχική σελίδα της εφαρμογής επιλέγουμε 'είσοδος στο σύστημα' και εισάγουμε το username και τον κωδικό πρόσβασης στη φόρμα σύνδεσης. Πατάμε 'login' και αν είμαστε εγγεγραμμένοι θα έχουμε πρόσβαση στον προσωπικό μας λογαριασμό.

Κέντρο Ημέρας / Διαχειριστής / Εγγραφή / Εγγραφή νέου ασθενούς και συμπλήρωση ερωτηματολογίου / Εγγεγραμμένοι Ασθενείς / **Είσοδος στο σύστημα**

Κέντρο Ημέρας

Username: Password: [Login](#)

4.1.3 Δημιουργία νέου ασθενούς και ερωτηματολόγιο

Για να προσθέσουμε έναν νέο ασθενή επιλέγουμε 'Εγγραφή νέου ασθενούς και συμπλήρωση ερωτηματολογίου'. Συμπληρώνουμε τα απαραίτητα πεδία, όπως το όνομα και άλλες σχετικές πληροφορίες και το ερωτηματολόγιο για αυτόν τον ασθενή, προκειμένου να συλλέξουμε συγκεκριμένες πληροφορίες.

Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.

A-1 Γνωστικά ελλείμματα μη διαγνωσμένα

- Έχει λάβει διάγνωση για άνοια ο/η εξεταζόμενος/η;

ΚΑΙ

- Παρουσιάζει μία ή περισσότερες αλλαγές στις ακόλουθες γνωστικές λειτουργίες ή στη λειτουργικότητά του/της;

ο Απώλεια μνήμης

ο Σύγχυση

ο Αλλαγές στην καθημερινή του/της λειτουργικότητα

Ναι ☒ Όχι ☐

Αν ναι, καλύπτεται η ανάγκη;

Όχι ☒ Μερικώς ☐ Πλήρως ☐

Σημειώσεις:

Education of Carer:

A-2 Σύνθετη παρουσίαση των γνωστικών συμπτωμάτων

Ο/Η ασθενής με άνοια:

- Έχει μια ξαφνική έναρξη ή αλλαγή στις γνωστικές λειτουργίες ή σε άλλες πτυχές της νοητικής του/της κατάστασης;

- Έχει νέα συμπτώματα που δεν έχουν αξιολογηθεί;

Ναι ☐ Όχι ☒

Αν ναι, καλύπτεται η ανάγκη;

Όχι ☐ Μερικώς ☐ Πλήρως ☐

Σημειώσεις: Δεν υπήρξε καμία αλλαγή στις γνωστικές λειτουργίες ή σε άλλες πτυχές της νοητικής του κατάστασης

07/02/2024



NPI:

STAGE:

MILD



IADL:

MMSE:

4



Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.

A-3 Προσέγγιση γνωστικών συμπτωμάτων

- Έχει ο/η ασθενής διαγνωστεί πρόσφατα με άνοια;
 - Ανήκει στα πρώιμα προς τα μεσαία στάδια της άνοιας;
- Ναι ☒ Όχι ☐

Αν ναι, καλύπτεται η ανάγκη;

Όχι ☐ Μερικώς ☒ Πλήρως ☐

Σημειώσεις: Δεν υπάρχει επαρκής κάλυψη της ανάγκης

IG-1 Έλλειψη εγγράφων σχετικών με τη δικαστική συμπαράσταση για τη λήψη αποφάσεων (π.χ. πλήρης δικαστική συμπαράσταση, υπηρεσία υγειονομικής περίθαλψης)

- Είναι εξουσιοδοτημένος ο φροντιστής μέσω πλήρους δικαστικής συμπαράστασης;
 - Διαθέτει μία διαρκή δικαστική συμπαράσταση για την υγειονομική περίθαλψη;
 - Έχει δώσει ο φροντιστής απώτερες οδηγίες (π.χ. Διαθήκη) που εκφράζουν τις προτιμήσεις του/της για την φροντίδα τελικού σταδίου;
- Ναι ☒ Όχι ☐

Αν ναι, καλύπτεται η ανάγκη;

Όχι ☒ Μερικώς ☐ Πλήρως ☐

Σημειώσεις:

Submit

Αφού έχουμε συμπληρώσει τη φόρμα και έχουμε απαντήσει στις ερωτήσεις επιλέγουμε Submit για να αποθηκευτεί ο ασθενής μας.

4.1.4 Προβολή λίστας ασθενών

Για να δούμε τη λίστα των ασθενών μας, πηγαίνουμε στην ενότητα 'Εγγεγραμμένοι Ασθενείς'. Σε περίπτωση που χρειαστεί να αναζητήσουμε έναν ασθενή, πληκτρολογούμε στην 'Αναζήτηση Ασθενούς' το όνομα. Το σύστημα θα παρουσιάσει τον ασθενή που πληροί τα κριτήρια αναζήτησής μας.

Αποθηκευμένοι ασθενείς

Αναζήτηση Ασθενούς:

Εισάγετε το όνομα...

Αναζήτηση

Φίλιππος

Download CSV For All

[Πίσω στο Κέντρο Ημέρας](#)

Απαντήσεις του Ασθενούς Φίλιππος

Ερώτηση	Πρώτη Απάντηση	Δεύτερη Απάντηση	Σημειώσεις
A-1 Γνωστικά ελλείμματα μη διαγνωσμένα • Έχει λάβει διάγνωση για άνοια ο/η εξεταζόμενος/η; ΚΑΙ • Παρουσιάζει μία ή περισσότερες αλλαγές στις ακόλουθες γνωστικές λειτουργίες ή στη λειτουργικότητά του/της; ο Απώλεια μνήμης ο Σύγχυση ο Αλλαγές στην καθημερινή του/της λειτουργικότητα	Ναι	Όχι	
A-2 Σύνθετη παρουσίαση των γνωστικών συμπτωμάτων Ο/Η ασθενής με άνοια: • Έχει μια ξαφνική έναρξη ή αλλαγή στις γνωστικές λειτουργίες ή σε άλλες πτυχές της νοητικής του/της κατάστασης; • Έχει νέα συμπτώματα που δεν έχουν αξιολογηθεί;	Όχι	None	Δεν υπήρξε καμία αλλαγή στις γνωστικές λειτουργίες ή σε άλλες πτυχές της νοητικής του κατάστασης
A-3 Προσέγγιση γνωστικών συμπτωμάτων • Έχει ο/η ασθενής διαγνωστεί πρόσφατα με άνοια; • Ανήκει στα πρώιμα προς τα μεσαία στάδια της άνοιας;	Ναι	Μερικώς	Δεν υπάρχει επαρκής κάλυψη της ανάγκης
B-1 Ψυχωτικά συμπτώματα Ο/Η ασθενής με άνοια: • Ανησυχεί ή αναστατώνεται λόγω καχυποψίας, παράνοιας ή παρανοήσεων (π.χ. παραισθήσεις, ψευδαισθήσεις, παρανοϊκές ιδέες); • Αυτές οι παρανοήσεις τον/την κάνουν να υποφέρει ή προκαλούν μη	Ναι	Μερικώς	

Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.

Γ-4 Διαχείριση φαρμακευτικής αγωγής • Λαμβάνει ο/η ασθενής με άνοια πολλά φάρμακα (συνταγογραφούμενα και μη) που δεν έχουν επανελεγχθεί τους τελευταίους 6 μήνες; • Λαμβάνει πολλά φάρμακα (συνταγογραφούμενα και μη) που δεν έχουν επανελεγχθεί έπειτα από νοσηλεία; • Υπάρχει ένδειξη ανάγκης προσαρμογής της δόσης ή του δοσολογικού σχήματος των φαρμάκων; • Λαμβάνει φάρμακα συνταγογραφημένα από πολλαπλούς ιατρούς ή προτεινόμενα από φαρμακοποιούς που πρέπει να ελεγχθούν για αλληλεπιδράσεις ή πιθανές αλληλεπικαλύψεις;	Ναι	Πλήρως	
Γ-5 Οδοντιατρική εξέταση/φροντίδα • Έχει φτωχή στοματική υγιεινή, όπως φαίνεται από κακοσμία, σπασμένα ή με τερηδόνα δόντια ή παθήσεις του στόματος/ούλων; • Έχει χαλαρή ή ακατάλληλα προσαρμοσμένη οδοντοστοιχία;	Ναι	Μερικώς	
ΙΓ-1 Έλλειψη εγγράφων σχετικών με τη δικαστική συμπαράσταση για τη λήψη αποφάσεων (π.χ. πλήρης δικαστική συμπαράσταση, υπηρεσία υγειονομικής περίθαλψης) • Είναι εξουσιοδοτημένος ο φροντιστής μέσω πλήρους δικαστικής συμπαράστασης; • Διαθέτει μία διαρκή δικαστική συμπαράσταση για την υγειονομική περίθαλψη; • Έχει δώσει ο φροντιστής απώτερες οδηγίες (π.χ. Διαθήκη) που εκφράζουν τις προτιμήσεις του/της για την φροντίδα τελικού σταδίου;	Ναι	Όχι	

Download CSV

διαγραφή ασθενή

Το σύστημα έχει αποθηκεύσει τον ασθενή που δημιουργήσαμε με το όνομα Φίλιππος και έχει συλλέξει τις απαντήσεις σε ένα αρχείο csv που μπορούμε να κατεβάσουμε.

patient_103_choices.csv - LibreOffice Calc

Αρχείο Επεξεργασία Προβολή Εισαγωγή Μορφή Τεχνοπλοπείες Φύλλο Δεδομένα Εργαλεία Παράθυρο Βοήθεια

Liberation Sans 10 pt B I U A Z Z Ω Σ % 0.0 0.00 0.00 < > < >

	A	B	C	D	E	F
1	Question	First Answer	Second Answer	Notes		
2	A-1	Nai	Όχι			
3	A-2	Όχι		Δεν υπήρξε καμία αλλαγή στις γνωστικές λειτουργίες ή σε άλλες πτυχές της νοητικής του κατάστασης		
4	A-3	Nai	Μερικώς	Δεν υπάρχει επαρκής κάλυψη της ανάγκης		
5	B-1	Nai	Μερικώς			
6	B-2	Nai	Όχι			
7	B-3	Nai	Μερικώς			
8	B-4	Nai	Πλήρως			
9	B-5	Όχι				
10	G-1	Nai	Μερικώς			
11	G-2	Nai	Όχι			
12	G-3	Όχι				
13	G-4	Nai	Πλήρως			
14	G-5	Nai	Μερικώς			
15	H-	Nai	Όχι			
16						
17						
18						
19						
20						
21						
22						

< patient_103_choices >

Φύλλο 1 από 1 Προεπιλογή Ελληνικά Μέσος όρος ; Άθροισμα: 0

Πατώντας την επιλογή Διαγραφή Ασθενή το σύστημα διαγράφει τον ασθενή μαζί με τα στοιχεία και τις απαντήσεις που έχουν δοθεί γι'αυτόν και επιστρέφει στην σελίδα με τους αποθηκευμένους ασθενείς.

4.1.5 Επεξεργασία στοιχείων και ερωτήσεων ασθενή

Η τελευταία λειτουργία που υλοποιήθηκε στην εφαρμογή μας δίνει την δυνατότητα να επεξεργαστούμε τα στοιχεία και τις ερωτήσεις για τον ασθενή. Στη σελίδα της προβολής των στοιχείων του ασθενή προσθέσαμε μια ακόμα επιλογή όπου μας κατευθύνει στη σελίδα υποβολής των απαντήσεων ώστε να καταχωρίσουμε νέες και να τις αποθηκεύσουμε στη βάση.

Απαντήσεις του Ασθενούς stelios			
Ερώτηση	Πρώτη Απάντηση	Δεύτερη Απάντηση	Σημειώσεις
A-1 Γνωστικά ελλείμματα μη διαγνωσμένα • Έχει λάβει διάγνωση για άνοια ο/η εξεταζόμενος/η; ΚΑΙ • Παρουσιάζει μία ή περισσότερες αλλαγές στις ακόλουθες γνωστικές λειτουργίες ή στη λειτουργικότητά του/της: ο Απώλεια μνήμης ο Σύγχυση ο Αλλαγές στην καθημερινή του/της λειτουργικότητα	Όχι	None	
A-2 Σύνθετη παρουσίαση των γνωστικών συμπτωμάτων Ο/Η ασθενής με άνοια: • Έχει μια ξαφνική έναρξη ή αλλαγή στις γνωστικές λειτουργίες ή σε άλλες πτυχές της νοητικής του/της κατάστασης; • Έχει νέα συμπτώματα που δεν έχουν αξιολογηθεί;	Όχι	None	

[Download CSV](#) [διαγραφή ασθενή](#) [επεξεργασία στοιχείων](#)

Επιλέγοντας ‘επεξεργασία στοιχείων’ μπορούμε να δούμε τις ήδη καταχωρημένες απαντήσεις του ασθενή και να συμπληρώσουμε τα κενά πεδία ή να αντικαταστήσουμε τα παλιά στοιχεία με καινούργια

Συμπληρώστε τα στοιχεία του ασθενή

Identity:

stelios

Category of Education:

0-6

Education of Patient:

Sex of Patient:

Male

Age of Patient:

Carer's category of education:

A-1 Γνωστικά ελλείμματα μη διαγνωσμένα

- Έχει λάβει διάγνωση για άνοια ο/η εξεταζόμενος/η;

ΚΑΙ

- Παρουσιάζει μία ή περισσότερες αλλαγές στις ακόλουθες γνωστικές λειτουργίες ή στη λειτουργικότητά του/της:

ο Απώλεια μνήμης

ο Σύγχυση

ο Αλλαγές στην καθημερινή του/της λειτουργικότητα

Ναι ☒ Όχι ☐

Αν ναι, καλύπτεται η ανάγκη;

Όχι ☐ Μερικώς ☐ Πλήρως ☒

Σημειώσεις:

Επιλέγοντας υποβολή στοιχείων ανανεώνουμε αυτόματα και την σελίδα προβολής του ασθενή

Απαντήσεις του Ασθενούς stelios

Ερώτηση	Πρώτη Απάντηση	Δεύτερη Απάντηση	Σημειώσεις
A-1 Γνωστικά ελλείμματα μη διαγνωσμένα • Έχει λάβει διάγνωση για άνοια ο/η εξεταζόμενος/η; ΚΑΙ • Παρουσιάζει μία ή περισσότερες αλλαγές στις ακόλουθες γνωστικές λειτουργίες ή στη λειτουργικότητά του/της: ο Απώλεια μνήμης ο Σύγχυση ο Αλλαγές στην καθημερινή του/της λειτουργικότητα	Ναι	Πλήρως	
A-2 Σύνθετη παρουσίαση των γνωστικών συμπτωμάτων Ο/Η ασθενής με άνοια: • Έχει μια ξαφνική έναρξη ή αλλαγή στις γνωστικές λειτουργίες ή σε άλλες πτυχές της νοητικής του/της κατάστασης; • Έχει νέα συμπτώματα που δεν έχουν αξιολογηθεί;	Όχι	None	

[Download CSV](#) [διαγραφή ασθενή](#) [επεξεργασία στοιχείων](#)

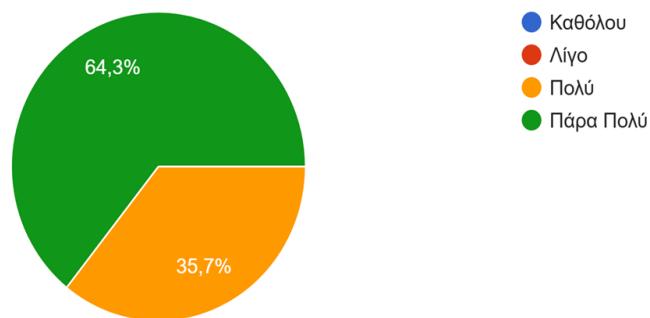
4.2 Δοκιμή και αξιολόγηση της εφαρμογής

Η προσεκτική αξιολόγηση και ανάλυση αναδεικνύει την ποιότητα και την αποτελεσματικότητα της εφαρμογής που δημιουργήσαμε. Εξετάζει κρίσιμες πτυχές της λειτουργικότητας και της απόδοσης, προσαρμόζοντας και βελτιώνοντας συνεχώς την εμπειρία των χρηστών. Μέσω αυτού του ερωτηματολογίου, επιδιώκουμε να αναδείξουμε τα θετικά σημεία, προκειμένου να διασφαλίσουμε ότι η εφαρμογή μας παρέχει την υψηλότερη δυνατή ποιότητα υγειονομικής φροντίδας. Τα αποτελέσματα θα μας κατευθύνουν προς τη σταδιακή βελτίωση της πλατφόρμας, προσφέροντας μια εξελιγμένη εμπειρία για τους χρήστες μας. Στη δοκιμή και αξιολόγηση συμμετείχαν δέκα τρία φυσικά πρόσωπα τα οποία έβαλαν σε λειτουργία την εφαρμογή και αλληλοεπίδρασαν με τις λειτουργίες της. Παρακάτω φαίνονται οι απαντήσεις/γραφήματα που δώσανε μετά την δοκιμή της εφαρμογής.

Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.

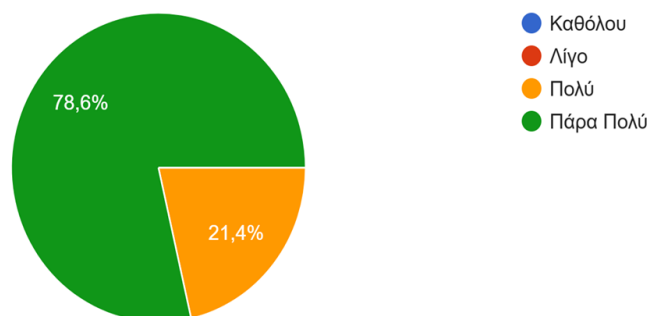
Πόσο εύκολο βρήκατε τον εαυτό σας στη χρήση της εφαρμογής;

14 απαντήσεις



Πόσο γρήγορα μπορέσατε να εκτελέσετε βασικές λειτουργίες;

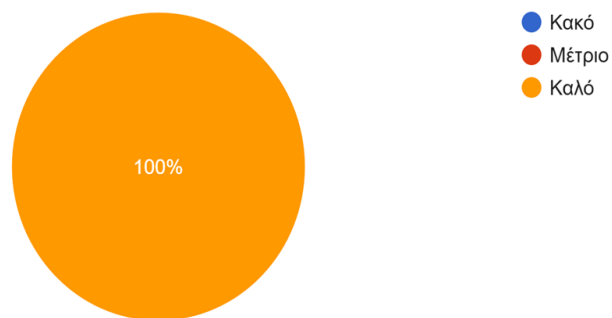
14 απαντήσεις



Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.

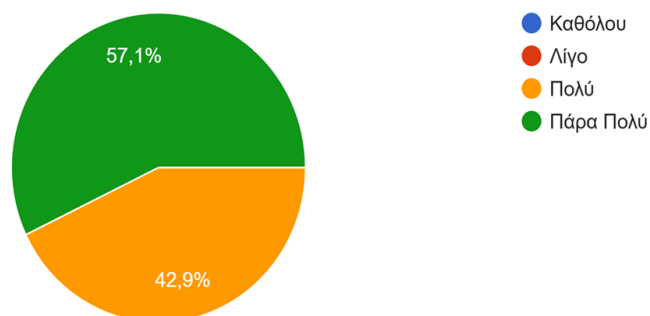
Πώς αξιολογείτε τον γενικό σχεδιασμό της εφαρμογής;

14 απαντήσεις



Είναι εύκολο να πλοηγηθείτε μέσα στις διάφορες ενότητες της εφαρμογής;

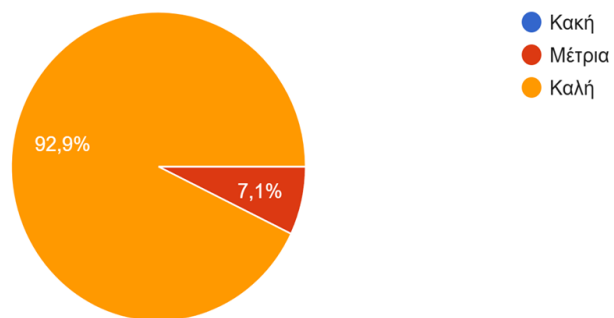
14 απαντήσεις



Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.

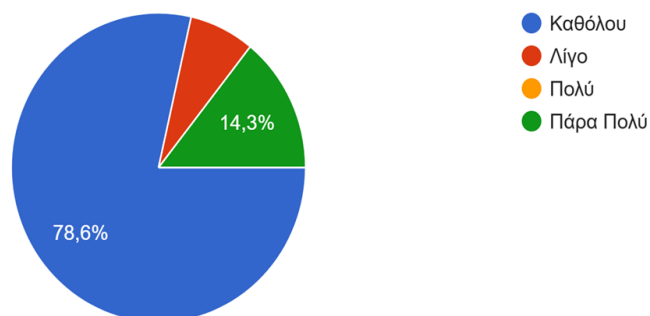
Πώς αξιολογείτε την αποτελεσματικότητα των λειτουργιών της εφαρμογής;

14 απαντήσεις



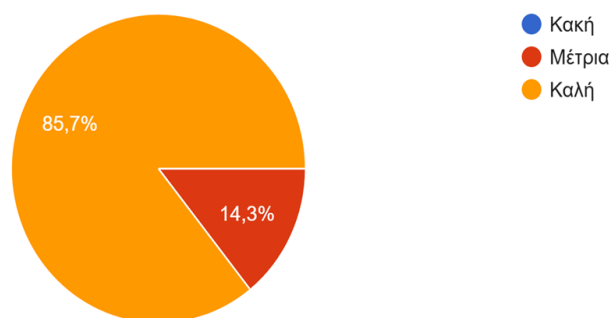
Υπήρξαν προβλήματα ή δυσκολίες κατά τη χρήση των λειτουργιών;

14 απαντήσεις

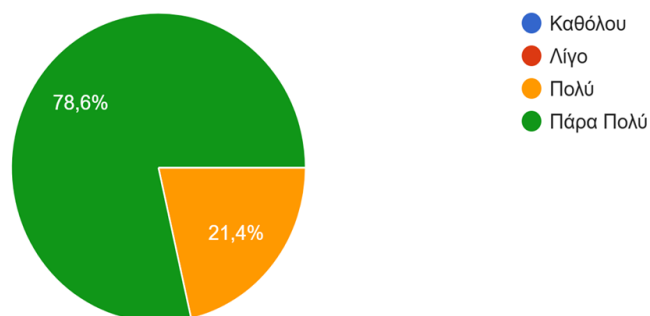


Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.

Πώς βιώσατε τη διαχείριση και την καταγραφή των πληροφοριών των ασθενών;
14 απαντήσεις



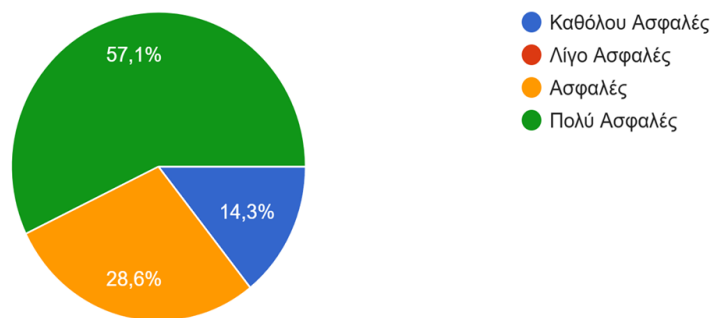
Βοηθά το ερωτηματολόγιο στην καλύτερη παρακολούθηση της κατάστασης των ασθενών;
14 απαντήσεις



Ανάπτυξη πλατφόρμας διαχείρισης ασθενών
σε ιατρεία μνήμης και κέντρα ημέρας – Παπακωνσταντίνου Γ.

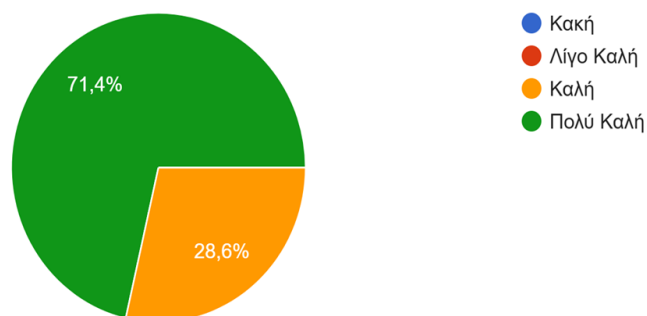
Πώς αισθάνεστε όσον αφορά την ασφάλεια των πληροφοριών σας;

14 απαντήσεις



Συνολικά, πώς αξιολογείτε την εμπειρία σας χρησιμοποιώντας την εφαρμογή;

14 απαντήσεις



Συμπεράσματα από την χρήση και αξιολόγηση της εφαρμογής

Η αξιολόγηση της εφαρμογής αποκάλυψε αρκετούς τομείς βελτίωσης. Αρχικά, η ενίσχυση της χρηστικότητας και της ευκολίας χρήσης είναι ουσιώδης για τη βελτίωση της εμπειρίας των χρηστών. Επιπλέον, η βελτίωση της απόδοσης και της απόκρισης της εφαρμογής θα συμβάλει στην ομαλή λειτουργία της. Η διασφάλιση της ασφάλειας των προσωπικών δεδομένων των χρηστών είναι άλλος ένας σημαντικός παράγοντας για την εμπιστοσύνη και την αποδοχή της εφαρμογής. Επιπλέον, η αισθητική βελτίωση της διεπαφής χρήστη θα κάνει την εφαρμογή πιο ελκυστική και φιλική προς τον χρήστη. Η συλλογή και αξιολόγηση ανατροφοδοτήσεων από τους χρήστες είναι επίσης ουσιώδης για τη συνεχή βελτίωση και προσαρμογή της εφαρμογής στις ανάγκες τους. Τέλος, η διερεύνηση επιπλέον επεκτάσεων για τη μελλοντική ανάπτυξη της εφαρμογής θα επιτρέψει την ενσωμάτωση νέων λειτουργιών και δυνατοτήτων που θα εξυπηρετούν ακόμα καλύτερα τις ανάγκες των χρηστών. Με αυτές τις βελτιώσεις, η εφαρμογή θα μπορεί να παρέχει ακόμα πιο ολοκληρωμένη και αποτελεσματική υποστήριξη στους γιατρούς και τους ασθενείς της.

Βιβλιογραφία

Ελληνική βιβλιογραφία

1. Βουτσίδου-Μιχαηλίδου. (2021). Ηλεκτρονικά προγράμματα και σύγχρονο νοσοκομειακό μάνατζμεντ. Πεδίο.
2. Μέθοδοι και πρακτικές στατιστικών ερευνών. (2021). . σοφία.
3. Μεθοδολογία της έρευνας στις επιστήμες υγείας. (2022). . Κριτική.

Ξένη βιβλιογραφία

1. Bates DW, Leape LL, Cullen DJ, et al. Effect of computerized physician order entry and a team intervention on prevention of serious medication errors. JAMA 1998;280(15):1311–6.
2. Dick RS, Steen EB, Detmer DE, editors. The computer-based patient record: an essential technology for health care, revised edition. Washington, DC: The National Academies Press; 1997.
3. Dolin RH, Alschuler L, Beebe C, et al. The HL7 clinical document architecture. J Am Med Inform Assoc 2001;8(6):552–69.
4. Duckett, J. (2011). HTML and CSS Design and Build Websites. John Wiley & Sons Inc.
5. Guttag, J. V. (2022). Εισαγωγή στον Υπολογισμό και τον Προγραμματισμό με την Python, 3η έκδοση. Παπασωτηρίου.
6. HealthIT.gov. Clinical Decision Support (CDS). Available at: <http://www.healthit.gov/policy-researchers-implementers/clinical-decision-support-cds>. Accessed July 8, 2014.
7. HealthIT.gov. Health Information Exchange (HIE). Available at: <http://www.healthit.gov/providers-professionals/health-information-exchange/what-hie>. Accessed July 8, 2014.
8. HealthIT.gov. What is a patient portal? Available at: <http://www.healthit.gov/providers-professionals/faqs/what-patient-portal>. Accessed July 8, 2014.
9. Institute of Medicine (IOM) Committee on Data Standards for Patient Safety. Key capabilities of an electronic health record system: letter report. Washington, DC: The National Academies Press; 2003.
10. Institute of Medicine (IOM). Crossing the quality chasm: a new health system for the 21st century. Washington, DC: The National Academies Press; 2001.
11. Jaiswal, S. (2015). Learning django web development: From idea to prototype, a learner's Guide for Web Development with the django application framework. Packt.
12. Mamykina L, Vawdrey DK, Stetson PD, et al. Clinical documentation: composition or synthesis? J Am Med Inform Assoc 2012;19(6):1025–31.

13. Pantanowitz L, Tuthill JM, Balis UG. Pathology informatics: theory & practice. Chicago, IL: American Society of Clinical Pathologists Press; 2012.
14. Payne TH, Hoey PJ, Nichol P, et al. Preparation and use of preconstructed orders, order sets, and order menus in a computerized provider order entry system. J Am Med Inform Assoc 2003;10(4):322–9.
15. Portilla, J. (2019). Python and Django Full Stack Web Developer Bootcamp. Udemy. <https://www.udemy.com/>
16. Stair, R. M. (2022). Αρχές πληροφοριακών συστημάτων. Τζιόλα.
17. Sirintrapun & Artz, 152.
18. Tarczy-Hornoch P, Amendola L, Aronson SJ, et al. A survey of informatics approaches to whole-exome and whole-genome clinical reporting in the electronic health record. Genet Med 2013;15(10):824–32.
19. Yu P, Artz D, Warner J. Electronic health records (EHRs): supporting ASCO’s vision of cancer care. Am Soc Clin Oncol Educ Book 2014;225–31.