**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**

Graphical user interface

Description automatically generated with medium confidence

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ 6Ο ΕΞΑΜΗΝΟ**

ΑΡΓΥΡΑ ΟΥΡΑΝΙΑ Π20023

ΑΡΜΕΝΗ ΑΘΗΝΑ Π20025

ΠΑΝΑΓΟΠΟΥΛΟΣ ΑΛΕΞΑΝΔΡΟΣ-ΒΑΣΙΛΕΙΟΣ Π20150

ΓΕΩΡΓΙΑΔΗΣ ΝΙΚΟΛΑΟΣ Π19032

ΕΡΓΑΣΙΑ 2022-2023

**ΠΕΙΡΑΙΑΣ 2022-2023**

Στην εργασία μας ζητείται να υλοποιήσουμε μια desktop εφαρμογή δημιουργίας διαιτολογίου.

Αρχικά, θα ορίσουμε τους στόχους της εργασίας, οι οποίοι κρίνονται απαραίτητοι, καθώς λειτουργούν ως μέτρα επίδοσης και αξιολόγησης της προόδου και της επίτευξης των αποτελεσμάτων. Θέτοντας τους στόχους, μπορούμε να οργανώσουμε και να διαχειριστούμε την εφαρμογή μας με πιο αποδοτικό τρόπο όπως επίσης και να κατανοήσουμε τις ανάγκες και τις προσδοκίες του χρήστη.

1. **Εισαγωγή**
   1. **Στόχοι της εργασίας**

Κύριος στόχος μας είναι να αναπτυχθεί λογισμικό, καθώς και να συνοδευτεί με την αντίστοιχη ανάπτυξή του σε μοντέλο κύκλου ζωής λογισμικού RUP(Rational Unified Process). Παρακάτω έχουμε αναλύσει μερικούς γενικούς στόχους που κρίνουμε απαραίτητους κατά την ανάπτυξη του λογισμικού:

1. **Αντικειμενικά χαρακτηριστικά :** Η εφαρμογή θα πρέπει να περιλαμβάνει την εισαγωγή κάποιων αντικειμενικών παραμέτρων, όπως για παράδειγμα τα προσωπικά στοιχεία του ασθενή και τις προτιμήσεις του σε φαγητά. Με βάση αυτά τα χαρακτηριστικά, ο διαιτολόγος, ως τελικός χρήστης της εφαρμογής, θα μπορεί να εκδώσει ένα εβδομαδιαίο πρόγραμμα με βάση τις ανάγκες του πελάτη.
2. **Ποιότητα και Απόδοση :**  Αποτελούν βασικές πτυχές του προγράμματος. Το πρόγραμμα μας, θα πρέπει να είναι αξιόπιστο για να μπορεί ο διαιτολόγος να βασιστεί πάνω σε αυτό, έγκυρο, εύκαμπτο, αποδοτικό και αποτελεσματικό. Άλλες θεμελιώδεις ιδιότητες που θέλουμε να πληροί το πρόγραμμά μας είναι η χρηστικότητα, η φορητότητα και η συντηρησιμότητα.
3. **Ασφάλεια :** Απαραίτητη ιδιότητα για την εφαρμογή μας, καθώς τα δεδομένα που θα εκχωρούνται θα πρέπει να προστατεύονται και να παραμένουν ιδιωτικά (ιδιωτικότητα). Το πρόγραμμα δε θα πρέπει να είναι ευπαθή και ευάλωτο, δηλαδή να έχει ελαττώματα ή αδυναμίες που θα μπορούσαν να χρησιμοποιηθούν από κάποιον κακόβουλο χρήστη που μπορεί να χειριστεί τα δεδομένα ή γενικότερα πληροφορίες της εφαρμογής.
4. **Επεκτασιμότητα :** Η εφαρμογή θα πρέπει να είναι εύκολα επεκτάσιμη και προσαρμόσιμη σε μελλοντικές απαιτήσεις και αλλαγές.
5. **Ικανοποίηση του χρήστη :** Κύριος στόχος μας είναι ο χρήστης να μείνει ικανοποιημένος από την εφαρμογή μας και να καλύπτονται οι ανάγκες του και οι απαιτήσεις του. Αυτό θα επιτευχθεί με την ευχρηστία της εφαρμογής, την επεξήγησή της, το εγχειρίδιο χρήσης, κτλ.
6. **Φιλικό προς το χρήστη :** Δε θα πρέπει να δυσχεραίνει το χρήστη και θα πρέπει να είναι κατανοητό προς αυτόν. Επίσης, θα υπάρχουν οδηγίες και πληροφορίες για την ευκολία εκμάθησης της εφαρμογής, έτσι ώστε να αναπτυχθεί μία «οικειότητα» με το χρήστη.
7. **Ανθεκτικότητα :** Η εφαρμογή θα λειτουργεί υπό κανονικές ή συνήθεις συνθήκες. Να υπάρχει δηλαδή λειτουργικότητα (υποστήριξη των ενεργειών που θα ήθελε να κάνει ο χρήστης) και να μπορεί να ανταπεξέλθει σε πιθανά λάθη που μπορεί να κάνει ο χρήστης (πχ εισαγωγή αρνητικού βάρους ή ύψους). Σε αυτό συμβάλει η προβλεψιμότητα των ενεργειών που θα κάνει ο χρήστης και η ανανηψιμότητα δηλαδή η δυνατότητα επανόρθωσης λάθους.
8. **Συνέπεια :** Χρησιμοποίηση ίδιων συμβάσεων καθ’ όλη τη διάρκεια της εφαρμογής . Για παράδειγμα, να μη γίνεται απότομη αλλαγή χρωματισμού, απότομη αλλαγή κουμπιών και γενικότερα διαφορετικές αναπαραστάσεις για το ίδιο πράγμα, καθώς αυτό θα δυσκολέψει το χρήστη κατά τη χρήση της εφαρμογής.
   1. **Ορισμός του προβλήματος προς επίλυση**

Καλούμαστε να δημιουργήσουμε μία desktop εφαρμογή, η οποία θα δημιουργεί διαιτολόγια, ανάλογα με τα στοιχεία του πελάτη που θα εκχωρούνται. Για να γίνουν όμως αυτές οι ενέργειες, θα πρέπει να υλοποιήσουμε στην αρχή μερικούς πίνακες, όπως έναν πίνακα τροφίμων που περιγράφει την περιεκτικότητά τους σε διάφορα στοιχεία όπως πρωτεΐνες, υδατάνθρακες, ενέργεια (θερμίδες-kcal), λίπη, κτλ. Θα πρέπει να αναπτυχθεί σύστημα δημιουργίας λογαριασμού/σύνδεσης, πατώντας εγγραφή/σύνδεση και συμπληρώνοντας τα στοιχεία μας όπως Όνομα, Επώνυμο, Κωδικό Πρόσβασης και E-mail. Επίσης, θα πρέπει να γίνεται διαφοροποίηση μεταξύ ασθενών και διαιτολόγων-διατροφολόγων κατά τη σύνδεση. Ο διατροφολόγος πρέπει να έχει τη δυνατότητα να εκδίδει πλήρη εβδομαδιαία προγράμματα και να βλέπει αυτά που έχει ήδη εκδώσει για τους πελάτες του, ενώ ο πελάτης θα πρέπει να βλέπει τα πλήρη εβδομαδιαία προγράμματα που έχουν εκδοθεί μόνο για εκείνον και να μην μπορεί να δημιουργήσει γενικά διαιτολόγια.Τέλος, για την έκδοση του εβδομαδιαίου προγράμματος θα πρέπει να ληφθούν υπόψη οι απαιτήσεις και ανάγκες του πελάτη, (δηλαδή εάν ο πελάτης θέλει να αυξήσει, να μειώσει ή να κρατήσει σταθερό το βάρος του) και να ελέγχεται αν αυτές είναι λογικές σε σχέση με τα στοιχεία που καταχώρησε.

1. **Σύντομη παρουσίαση της RUP**

Η μεθοδολογία Rational Unified Process Methodology ή αλλιώς RUP είναι μία ευέλικτη μέθοδος ανάπτυξης λογισμικού, στην οποία ο κύκλος ζωής ενός έργου, ή η ανάπτυξη λογισμικού, χωρίζεται σε 4 φάσεις :

* **Φάση Έναρξης**
* **Φάση Επεξεργασίας**
* **Φάση Κατασκευής**
* **Μεταβατική Φάση**

**Φάση Έναρξης (Inception)**

Κατά την πρώτη φάση καθορίζεται η βασική ιδέα και η δομή του έργου. Επίσης, καθορίζεται η αναγκαιότητα του έργου, η βιωσιμότητά του και η καταλληλότητά του και μας επιτρέπει να εκτιμήσουμε το κόστος και τον απαιτούμενο χρόνο.

**Φάση Επεξεργασίας (Elaboration)**

Σε αυτό το στάδιο γίνεται ο προγραμματισμός και η μοντελοποίηση. Επίσης, αξιολογούνται και αναλύονται οι απαιτήσεις της εφαρμογής. Εκτελείται λεπτομερές σχέδιο αξιολόγησης και ανάπτυξης και μειώνονται οι κίνδυνοι.

**Φάση Κατασκευής (Construction)**

Το έργο αναπτύσσεται και ολοκληρώνεται, δηλαδή το σύστημα λογισμικού κατασκευάζεται στο σύνολό του. Δημιουργείται σύστημα ή πηγαίος κώδικας και στη συνέχεια γίνεται δοκιμή. Επίσης, πραγματοποιείται κωδικοποίηση. Τέλος , πραγματοποιούνται και κάποια test cases, ενώ σε αυτή τη φάση περιλαμβάνεται και το εγχειρίδιο χρήσης.

**Μεταβατική Φάση (Transition)**

Το τελικό έργο κυκλοφορεί στο κοινό, δηλαδή είναι έτοιμο για χρήση. Μόλις ο χρήστης αρχίσει να χρησιμοποιεί το σύστημα, σχεδόν πάντα προκύπτουν προβλήματα που απαιτούν πραγματοποίηση αλλαγών στο σύστημα. Στόχος, ωστόσο, είναι να διασφαλιστεί μια θετική και ομαλή μετάβαση στον χρήστη. Πραγματοποιείται δοκιμή beta, και τα ελαττώματα αφαιρούνται από την εφαρμογή με βάση τα σχόλια από το κοινό.

Οι φάσεις αποτελούνται από τις εξής δράσεις:

* **Μοντελοποίηση**
* **Ανάλυση και Σχεδιασμός**
* **Υλοποίηση**
* **Δοκιμή και Εφαρμογή**

Όπου κάθε φάση μπορεί να περιέχει μερικές ή όλες από αυτές τις δράσεις.

Για τον προγραμματισμό των ενεργειών της εργασίας, δημιουργήσαμε ένα Gantt Chart βασισμένο στην RUP, χρησιμοποιώντας την εφαρμογή drawio.com , <https://app.diagrams.net/> .

Ο πίνακας περιλαμβάνει το όνομα της ενέργειας (Task Name) και τις ενέργειες με βάση τη σειρά που πρέπει να γίνουν. Για τη **φάση της έναρξης (Inception)** χρειάζεται να γίνουν οι εξής ενέργειες: πίνακας τροφίμων και ανάλυση των διατροφικών χαρακτηριστικών του. Για τη **φάση της επεξεργασίας (Elaboration)** αρχικά θα πρέπει να γίνεται εγγραφή στην εφαρμογή-δημιουργία λογαριασμού, επιλογή εγγραφής ως διαιτολόγος ή ασθενής, σύνδεση στην εφαρμογή, εισαγωγή στοιχείων ασθενή (που συλλέξαμε στη φάση της έναρξης), δημιουργία ενός Use Case διαγράμματος (διάγραμμα περιπτώσεων χρήσης που περιγράφει τις υψηλού επιπέδου λειτουργίες και το πεδίο εφαρμογής ενός συστήματος και προσδιορίζονται οι αλληλεπιδράσεις μεταξύ συστήματος και χρηστών), συλλογή στοιχείων του ασθενή και αξιολόγηση των διατροφικέςών ανάγκών του (για παράδειγμα αν θέλει να αυξήσει το βάρος του, χρειαζόμαστε μία διατροφή με περισσότερες θερμίδες από αυτές που προσλαμβάνει ημερησίως) όπως επίσης και έλεγχος για το αν τα στοιχεία που δόθηκαν από το χρήστη είναι έγκυρα, για παράδειγμα ο δείκτης μάζας σώματος του χρήστη μπορεί να είναι κάτω από 18,5, το οποίο σημαίνει ότι ανήκει στο εύρος λιποβαρών, άρα δε μπορούμε να δώσουμε στο χρήστη διαιτολόγιο που θα στοχεύει στη μείωση του βάρους του, διότι κρίνεται επικίνδυνο για την υγεία του. Στη συνέχεια, για τη **φάση κατασκευής (Construction)** θα γίνει η σχεδίαση συστήματος-λογισμικού (Code Design), η δημιουργία διαιτολογίου του ασθενή, έλεγχος για τυχόν απαιτούμενες αλλαγές στο διαιτολόγιο με βάση τις ανάγκες και απαιτήσεις του ασθενή, δοκιμές της εφαρμογής ή αλλιώς (Test Cases) και εγχειρίδιο χρήσης της εφαρμογής (User Manual). Τέλος, για τη **μεταβατική φάση (Transition)** θα γίνεται έλεγχος για τυχόν προβλήματα της εφαρμογής, πραγματοποίηση αλλαγών στο σύστημα ή αλλιώς επίλυση των προβλημάτων που βρέθηκαν, έλεγχος απ’ το διαιτολόγο για την καταλληλότητα του διαιτολογίου της εφαρμογής, παράδοση του διαιτολογίου στον ασθενή, έλεγχος διαιτολογίου απ’ τον ασθενή (Δοκιμή beta) το οποίο είναι προσαρμοσμένο στις απαιτήσεις και τις ανάγκες του, ανάλυση κριτικής του χρήστη και τέλος υποστήριξη και συντήρηση της εφαρμογής.

A screenshot of a computer

Description automatically generated with medium confidence

**3.Φάση : Έναρξη (Inception)**

Ο όρος κύκλος ζωής έκδοσης ενός λογισμικού (software release life cycle) αναφέρεται στις φάσεις ανάπτυξης και ύπαρξης ενός λογισμικού. Ξεκινά από τη σύλληψη της ιδέας και την μελέτη για τις απαιτήσεις και προδιαγραφές του. Ακολουθεί η υλοποίησή του, με τη δημιουργία του κώδικα και την αποσφαλμάτωση ή βελτίωσή του. Γίνεται διάθεση του τελικού προϊόντος στον καταναλωτή (η δημιουργία διαιτολογίου στην περίπτωσή μας). Επίσης, είναι απαραίτητη η υποστήριξη του λογισμικού με ενημερώσεις (updates), οι οποίες βελτιώνουν τη λειτουργικότητά του ή διορθώνουν σφάλματα (bugs). Στο στάδιο ανάπτυξης ακολουθούνται 4 φάσεις, ώστε το προϊόν να είναι αξιόπιστο όταν θα φτάσει στον καταναλωτή.

Η «φάση Έναρξης» χρησιμοποιείται στην αρχή ενός προγράμματος και επικεντρώνεται στη μάθηση και στην καλύτερη κατανόηση του πλαισίου εφαρμογής. Ο στόχος είναι να σχεδιαστεί ένα πρόγραμμα που να είναι πιο αποτελεσματικό. Οι ιδέες, οι σχέσεις και οι προκλήσεις τις εφαρμογής πρέπει να εφαρμοστούν και να «λυθούν» από την έναρξη του προγράμματος. Είναι σημαντική κυρίως για νέες αναπτυξιακές προσπάθειες, στις οποίες υπάρχουν σημαντικοί επιχειρηματικοί κίνδυνοι και κίνδυνοι απαιτήσεων που πρέπει να αντιμετωπιστούν πριν προχωρήσει το έργο. Επίσης, βοηθάει στο να χτιστεί η ομάδα με το σωστό τρόπο από την αρχή. Η αρχική φάση είναι μία ευκαιρία για την ομάδα να υλοποιήσει τις ενέργειες της εφαρμογής. Τέλος, ενισχύεται η συνεργασία, ο συλλογικός σχεδιασμός της εφαρμογής που βασίζεται στη μάθηση.

Οι πρωταρχικοί στόχοι της αρχικής φάσης περιλαμβάνουν :

* **Καθορισμό του πεδίου εφαρμογής του λογισμικού και του εύρους του έργου**, συμπεριλαμβανομένων ενός επιχειρησιακού οράματος, κριτηρίων αποδοχής και του τι πρόκειται να συμπεριληφθεί και τι όχι. Αυτό περιλαμβάνει την αποτύπωση του πλαισίου και των πιο σημαντικών απαιτήσεων και περιορισμών σε τέτοιο βαθμό ώστε να μπορούμε να αντλούμε κριτήρια αποδοχής για το τελικό προϊόν.
* **Σχεδιασμό και προετοιμασία επιχειρηματικής υπόθεσης.** Γίνεται αξιολόγηση των εναλλακτικών περιπτώσεων για τη διαχείριση κινδύνου, στελέχωση, επιχειρησιακό σχέδιο με βάση το λογισμικό που θα αναπτύξουμε καθώς και των παραμέτρων κόστους, κερδοφορίας και χρονοδιαγράμματος.
* **Σύνθεση μιας υποψήφιας αρχιτεκτονικής**, αξιολόγηση των συμβιβασμών στο σχεδιασμό και στην κατασκευή/αγορά/επαναχρησιμοποίηση, έτσι ώστε το κόστος, το χρονοδιάγραμμα και οι πόροι να μπορούν να εκτιμηθούν. Ο στόχος εδώ είναι να αποδειχθεί η σκοπιμότητα μέσω κάποιου είδους απόδειξης της ιδέας. Αυτό μπορεί να λάβει τη μορφή ενός μοντέλου που προσομοιώνει αυτό που απαιτείται ή ενός αρχικού πρωτοτύπου που διερευνά ποιες θεωρούνται περιοχές υψηλού κινδύνου. Η προσπάθεια δημιουργίας πρωτοτύπων κατά την έναρξη θα πρέπει να περιορίζεται στην απόκτηση εμπιστοσύνης ότι μία λύση είναι δυνατή (η λύση πραγματοποιείται κατά την επεξεργασία και την κατασκευή).
* **Προετοιμασία του περιβάλλοντος για το έργο,** αξιολόγηση του έργου και της οργάνωσης, επιλογή εργαλείων και απόφαση ποια μέρη της διαδικασίας θα βελτιωθούν.
  1. **Σύλληψη απαιτήσεων**

Για τη σύλληψη των απαιτήσεων της εφαρμογής για τη διμιουργία διαιτολογίου θα ακολουθήσουμε τα παρακάτω βήματα :

1. **Αναγνώριση των χρηστών** : Όπως προαναφέρθηκε και πρηγουμένως, τελικοί χρήστες της εφαρμογής θα είναι οι διαιτολόγοι/διατροφολόγοι, οι οποίοι θα εκδίδουν το πρόγραμμα με βάση τις ανάγκες και απαιτήσεις του ασθενή και οι ασθενείς οι οποίοι θα μπορούν να βλέπουν στην εφαρμογή το τελικό πρόγραμμα διατροφής τους.
2. **Ανάλυση των απαιτήσεων** : Η εφαρμογή θα πρέπει να ικανοποιεί τις απαιτήσεις, τις ανάγκες και τις προσδοκίες των χρηστών. Απαιτείται η σωστή ανάλυση των διατροφικών χαρακτηριστικών του πίνακα τροφίμων, η διευκόλυνση του χρήστη για τη σύνδεσή του στην εφαρμογή, η εισαγωγή σωστών στοιχείων από τον ασθενη, η αξιολόγηση των διατροφικών αναγκών του ασθενή, η δημιουργία διαιτολογίου από το διαιτολόγο.

Καταγραφή απαιτήσεων του διαιτολόγου

* Δημιουργία πίνακα τροφίμων και ανάλυση των διατροφικών χαρακτηριστικών του.
* Εισαγωγή στοιχείων ασθενή και διόρθωση σε περίπτωση λάθους (πχ αρνητικό βάρος).
* Αξιολόγηση των διατροφικών αναγκών του ασθενή.
* Υπολογισμός θερμιδικών αναγκών βάσει πληροφοριών του ασθενή.
* Δημιουργία διαιτολογίου με βάση τις ανάγκες και απαιτήσεις του ασθενή.

Καταγραφή απαιτήσεων του ασθενή

* Καταχώρηση των στοιχείων του, των αναγκών του και των απαιτήσεών του στο διαιτολόγο.
* Δυνατότητα προβολής διαιτολογίου με βάση τις ανάγκες και απαιτήσεις του.

Λειτουργικές απαιτήσεις

* Εγγραφή και σύνδεση στην εφαρμογή και επιλογή εγγραφής ως διαιτολόγος η ασθενής.
* Εύκολη εγγραφή και πρόσβαση στην εφαρμογή και σύνδεση σε αυτή.
* Αποθήκευση και επεξεργασία προσωπικών πληροφοριών (βάρος, ύψος, αλλεργίες κτλ.) για την εξατομίκευση του διαιτολογίου.
* Δημιουργία διαιτολογίου.
* Περιβάλλον εφαρμογής φιλικό και κατανοητό προς τους χρήστες.

Απαιτήσεις ελέγχου

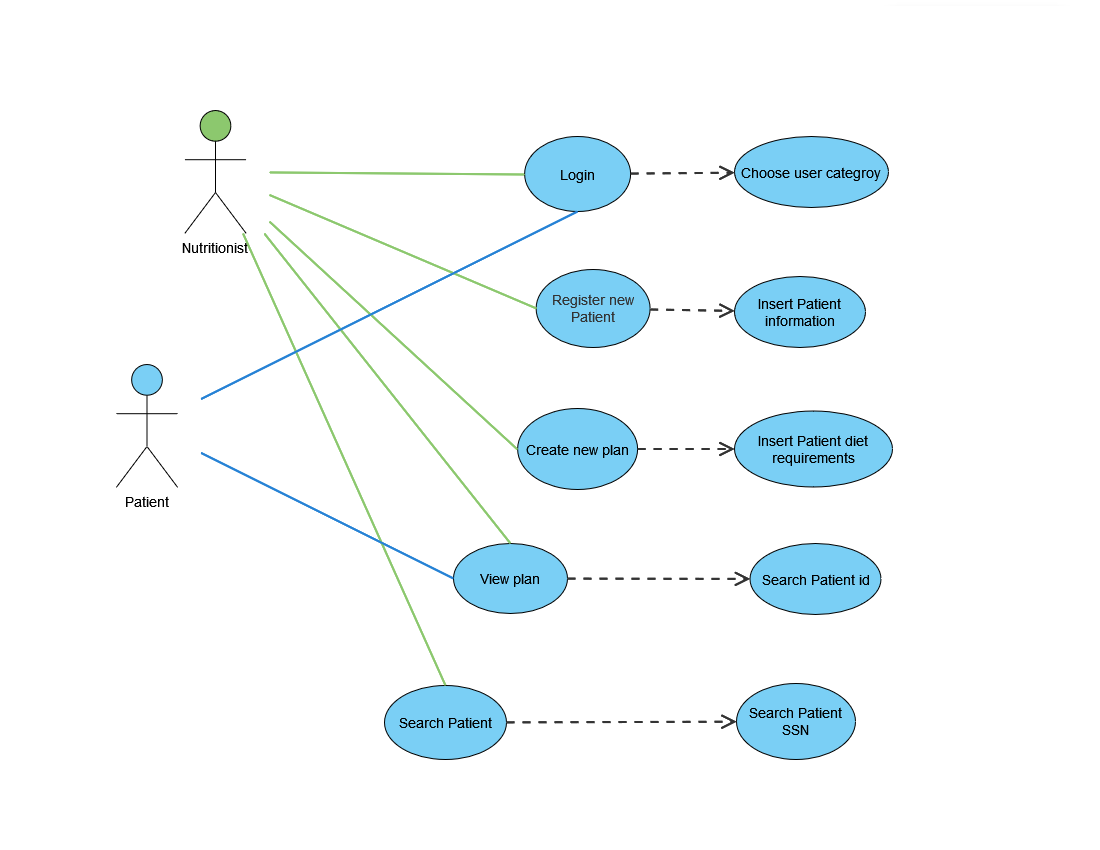
* Έλεγχος εισαγωγής έγκυρων στοιχείων του ασθενή.
* Έλεγχος για τυχόν απαιτούμενες αλλαγές στο διαιτολόγιο.
* Έλεγχος για πιθανά προβλήματα στην εφαρμογή.

Απαιτήσεις ασφάλειας

* Ασφάλεια δεδομένων κατά την εγγραφή και σύνδεση του χρήστη στην εφαρμογή.
* Παροχή προστασίας και ιδιωτικότητας των δεδομένων του ασθενή.
* Προστασία των δεδομένων και των πληροφοριών της εφαρμογής.
  1. **Ανάλυση-Σχεδιασμός**

Σε αυτή τη φάση παρουσιάζουμε 2 βασικά διαγράμματα στις αρχικές τους φάσεις. To ένα ονομάζεται Διάγραμμα Τάξεων (Class Diagram) το οποίο παρουσιάζει τις τάξεις από τις οποίες είναι δομημένη η εφαρμογή μας και τις σχέσεις που έχουν αυτές μεταξύ τους. Το δεύτερο, είναι το Διάγραμμα Περιπτώσεων Χρήσης (Use Case Diagram) το οποίο παρουσιάζει τις διαφορετικές λειτουργίες στις οποίες έχει/έχουν πρόσβαση ένας ή περισσότεροι ενεργοποιός/οι (actors). Οι λειτουργίες μπορούν να έχουν ως επεκτάσεις (<<extend>>) ή/και εμφολευμένες (<<uses>>) άλλες λειτουργίες (στη πρώτη φάση αυτές οι σχέσεις δεν παρουσιάζονται).

* + 1. **Διάγραμμα Περιπτώσεων Χρήσης**



1η έκδοση διαγράμματος περιπτώσεων χρήσης

* + 1. **Διάγραμμα Τάξεων**

**A picture containing text, diagram, line, screenshot

Description automatically generated**

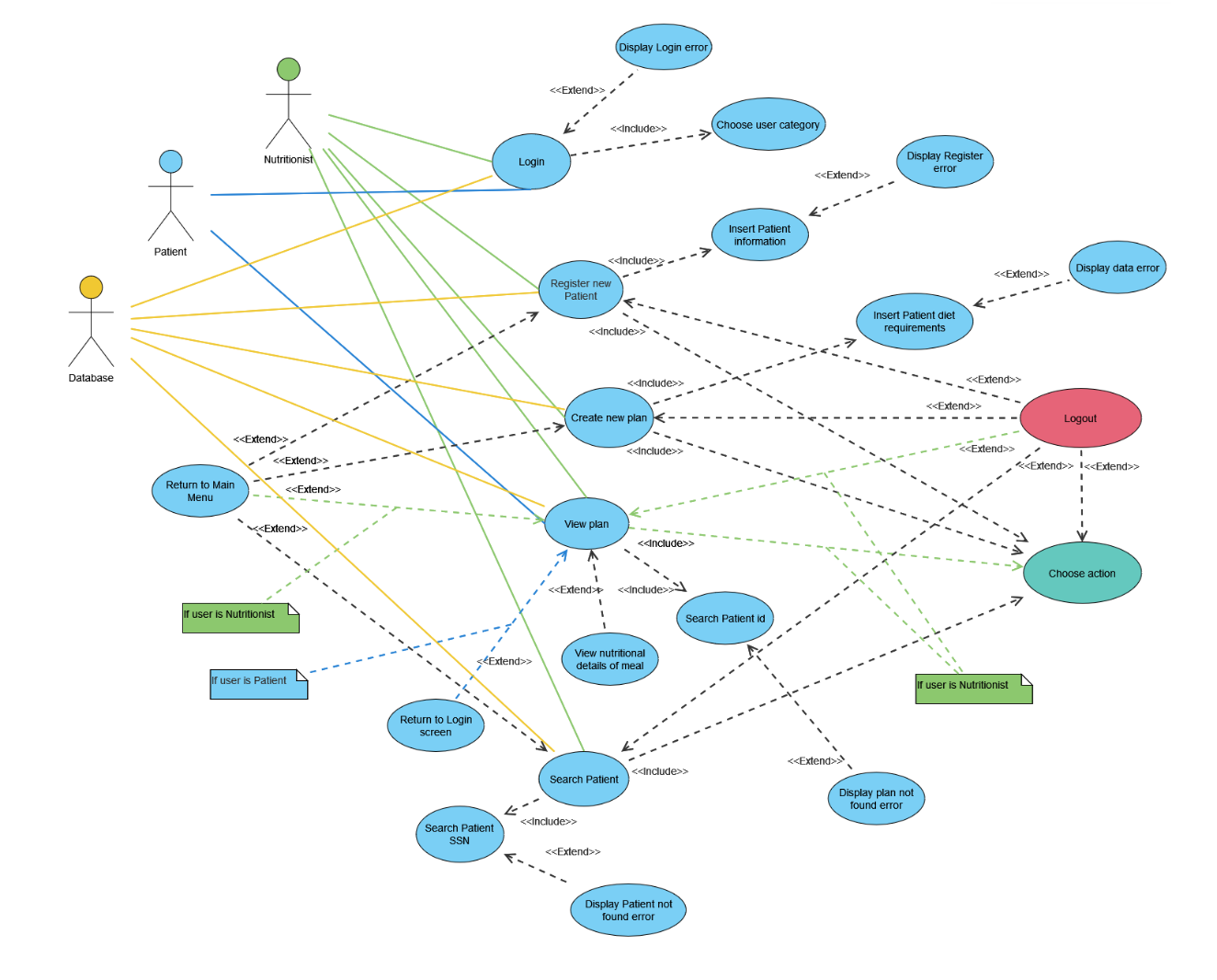
1η έκδοση διαγράμματος τάξεων

**4.Φάση : Εκπόνηση Μελέτης (Elaboration)**

* 1. **Ανάλυση-Σχεδιασμός**

Σε αυτή τη φάση παρουσιάζουμε τις δεύτερες εκδόσεις των παραπάνω διαγραμμάτων οι οποίες προέκυψαν κατά τη φάση εκπόνησης μελέτης του λογισμικού. Παρουσιάζονται επίσης οι αρχικές εκδόσεις των εξής διαγραμμάτων:

* Διάγραμμα Αντικειμένων
* Διάγραμμα Συνεργασίας
* Διάγραμμα Σειράς
* Διάγραμμα Δραστηριοτήτων
* Διάγραμμα Καταστάσεων
* Διάγραμμα Εξαρτημάτων
* Διάγραμμα Διανομής
  + 1. **Διάγραμμα Περιπτώσεων Χρήσης (2η έκδοση)**

****

2η έκδοση διαγράμματος περιπτώσεων χρήσης

* + 1. **Διάγραμμα Τάξεων (2η έκδοση)**

**A picture containing text, diagram, line, parallel

Description automatically generated**

2η έκδοση διαγράμματος τάξεων

* + 1. **Διάγραμμα Αντικειμένων (1η έκδοση)**

Τα διαγράμματα αντικειμένων είναι διαγράμματα δομής που χρησιμοποιούνται για το σχεδιασμό της στατικής κατάστασης του συστήματος κατά μια συγκεκριμένη χρονική στιγμή. Κάθε αντικείμενο σχεδιάζεται ως ένα ορθογώνιο. Το σύνολο των αντικειμένων σχεδιάζεται με βάση τους συνδέσμους που ορίζονται πάνω σε αυτό. Το αντικείμενο είναι μια παρουσία μιας κλάσης σε μια συγκεκριμένη στιγμή του χρόνου εκτέλεσης που μπορεί να έχει τις δικές της τιμές κατάστασης και δεδομένων. Ομοίως, ένα στατικό διάγραμμα αντικειμένου UML είναι μια εμφάνιση ενός διαγράμματος κλάσης. Δείχνει ένα στιγμιότυπο της λεπτομερούς κατάστασης ενός συστήματος σε μια χρονική στιγμή, επομένως ένα διάγραμμα αντικειμένων περιλαμβάνει αντικείμενα και τις σχέσεις τους που μπορεί να θεωρηθούν μια ειδική περίπτωση ενός διαγράμματος κλάσης ή ενός διαγράμματος επικοινωνίας.

* + 1. **Διάγραμμα Συνεργασίας (1η έκδοση)**

Τα διαγράμματα συνεργασίας (γνωστά ως Διάγραμμα επικοινωνίας στο UML 2.x) χρησιμοποιούνται για να δείξουν πώς αλληλεπιδρούν τα αντικείμενα για να εκτελέσουν τη συμπεριφορά μιας συγκεκριμένης περίπτωσης χρήσης ή ενός μέρους μιας περίπτωσης χρήσης. Μαζί με τα διαγράμματα ακολουθίας, η συνεργασία χρησιμοποιείται από τους σχεδιαστές για να καθορίσουν και να αποσαφηνίσουν τους ρόλους των αντικειμένων που εκτελούν μια συγκεκριμένη ροή γεγονότων μιας περίπτωσης χρήσης. Αποτελούν την κύρια πηγή πληροφοριών που χρησιμοποιούνται για τον προσδιορισμό των ευθυνών και των διεπαφών της τάξης.

* + 1. **Διάγραμμα Σειράς (1η έκδοση)**

Ένα διάγραμμα ακολουθίας/σειράς είναι ένας τύπος διαγράμματος αλληλεπίδρασης επειδή περιγράφει πώς —και με ποια σειρά— μια ομάδα αντικειμένων συνεργάζεται. Αυτά τα διαγράμματα χρησιμοποιούνται από προγραμματιστές λογισμικού και επαγγελματίες για την κατανόηση των απαιτήσεων για ένα νέο σύστημα ή για την τεκμηρίωση μιας υπάρχουσας διαδικασίας.

* + 1. **Διαγραμμα Δραστηριοτήτων (1η έκδοση)**

Τα Διαγράμματα δραστηριότητας (Activity diagrams) είναι γραφικές αναπαραστάσεις των ροών εργασίας (workflows) των σταδιακών δραστηριοτήτων και δράσεων με υποστήριξη για επιλογή, επανάληψη και συγχρονισμό. Στην UML (Unified Modeling Language), είναι διαγράμματα που προορίζονται να μοντελοποιήσουν τόσο τις υπολογιστικές όσο και τις οργανωτικές διαδικασίες (π.χ. ροές εργασίας). Τα διαγράμματα δραστηριότητας παρουσιάζουν τη συνολική ροή του ελέγχου.

* + 1. **Διάγραμμα Καταστάσεων (1η έκδοση)**

Τα διαγράμματα καταστάσεων (State diagrams) βοηθούν: στην ανάπτυξη ενός συστήματος, στην κατανόηση πολύπλοκων χαρακτηριστικών ή ροών εργασίας εξειδικευμένων περιοχών του συστήματος. Τα διαγράμματα καταστάσεων αναπαριστούν τη δυναμική συμπεριφορά ολόκληρου του συστήματος ή ενός μέρους του συστήματος.

* + 1. **Διάγραμμα Εξαρτημάτων (1η έκδοση)**

Το διάγραμμα εξαρτημάτων είναι ένα *διάγραμμα δομής υλοποίησης* που χρησιμοποιείται για να μοντελοποιήσει:

* Πηγαίο κώδικα
* Εκτελέσιμες εκδόσεις
* Βάσεις δεδομένων
* Δυναμικά προσαρμοζόμενα συστήματα

Περιλαμβάνει:

* Εξαρτήματα
* Διεπαφές (*interfaces*)
* Σχέσεις εξάρτησης, γενίκευσης, σύνδεσης και υλοποίησης

**4.1.9 Διάγραμμα Διανομής (1η έκδοση)**

Τα διαγράμματα διανομής μπορούν να αντιπροσωπεύουν έναν μεγάλο αριθμό διαφορετικών συσκευών υλικού και να επεκτείνουν τις γνώσεις μας από τον καλύτερο τρόπο από το να δούμε μερικές μελέτες περιπτώσεων.

* 1. **Υλοποίηση-Έλεγχος**
     1. **Υλοποίηση : 1η εκτελέσιμη έκδοση**
     2. **Αναφορά ελέγχου για την 1η εκτελέσιμη έκδοση**

**5.Φάση: Κατασκευή (Construction)**

* 1. **Ανάλυση-Σχεδιασμός**

Σε αυτή τη φάση παρουσιάζουμε τις τρίτες και τελευταίες εκδόσεις των διαγραμμάτων περιπτώσεων χρήσης και τάξεων οι οποίες προέκυψαν κατά τη φάση κατασκευής του λογισμικού. Παρουσιάζονται επίσης οι τελευταίες εκδόσεις των υπόλοιπων 7 διαγραμμάτων.

* + 1. **A picture containing diagram, text, circle, screenshot

       Description automatically generatedΔιάγραμμα Περιπτώσεων Χρήσης (3η έκδοση)**

3η έκδοση διαγράμματος περιπτώσεων χρήσης

* + 1. A picture containing text, diagram, line, screenshot

       Description automatically generated**Διάγραμμα Τάξεων (3η έκδοση)**

3η έκδοση διαγράμματος τάξεων

* + 1. **Διάγραμμα Αντικειμένων (2η έκδοση)**

A screenshot of a computer

Description automatically generated

2η έκδοση διαγράμματος αντικειμένων

* + 1. **Διάγραμμα Συνεργασίας (2η έκδοση)**

A screenshot of a computer

Description automatically generated

2η έκδοση διαγράμματος συνεργασίας

* + 1. **Διάγραμμα Σειράς (2η έκδοση)**

A screenshot of a computer

Description automatically generated

2η έκδοση διαγράμματος σειράς

* + 1. **Διάγραμμα Δραστηριοτήτητων (2η έκδοση)**
    2. **Διάγραμμα Καταστάσεων (2η έκδοση)**
    3. **Διάγραμμα Εξαρτημάτων (2η έκδοση)**
    4. **Διάγραμμα Διανομής (2η έκδοση)**
  1. **Υλοποίηση-Έλεγχος**
     1. **Υλοποίηση : Τελική εκτελέσιμη έκδοση**
     2. **Αναφορά ελέγχου για την τελική εκτελέσιμη έκδοση**

**6.Εγχειρίδια Χρήστη**

**6.1 Σύντομη Παρουσίαση του Προγράμματος**

**6.2 Παρουσίαση Σεναρίων Λειτουργίας**

Λεξικό :

**Beta Testing** : Στην ανάπτυξη λογισμικού, μία δοκιμή beta είναι η δεύτερη φάση της δοκιμής λογισμικού κατά την οποία μία δειγματοληψία του κοινού για το οποίο προορίζεται δοκιμάζει την εφαρμογή και την καταλληλότητά της.

Βιβλιογραφία : <https://www.geeksforgeeks.org/> , <https://www.toolshero.com/> , Σημειώσεις μαθήματος «Τεχνολογία Λογισμικού», Σημειώσεις μαθήματος «Αλληλεπίδραση Ανθρώπου-Υπολογιστή», <https://www.techtarget.com/> ,