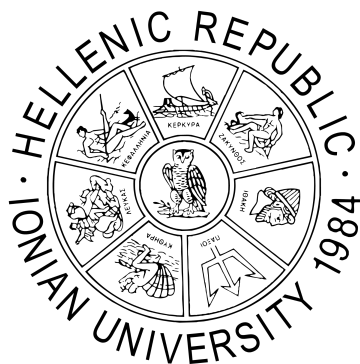


ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΕΡΚΥΡΑΣ
ΣΧΟΛΗ ΕΠΙΣΤΗΜΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ:

«Ανάπτυξη συστήματος Ηλεκτρονικού Επιχειρείν με χρήση της
τεχνολογίας Blockchain»

Προβόπουλος Βασίλειος	Π2018017	p18prov@ionio.gr
-----------------------	----------	--

Μέλη Τριμελούς Επιτροπής ΠΕ:

1. Δρ. Θεμιστοκλής Έξαρχος, Αναπληρωτής Καθηγητής
2. Δρ. Βασίλειος Καρυώτης, Αναπληρωτής Καθηγητής
3. Δρ. Παναγιώτης Κουρουθανάσης, Επιβλέπων Καθηγητής (Δ.Ε.Π.)

Περιεχόμενα

Abstract.....	4
Κεφάλαιο 1° - Η Ιχνηλασιμότητα στην Εφοδιαστική Αλυσίδα.....	5
1.1. Ορολογία της Εφοδιαστικής Αλυσίδας και η διαχείριση της.....	5
1.2. Ιχνηλασιμότητα στην Αλυσίδα Εφοδιασμού.....	7
1.3. Η Εφοδιαστική Αλυσίδα στον τομέα της Υγείας.....	9
1.4. Τα Στάδια στη Φαρμακευτική Αλυσίδα Εφοδιασμού.....	11
Κεφάλαιο 2° - Οι Αναδυόμενες Τεχνολογίες στη Φαρμακευτική Εφοδιαστική Αλυσίδα.....	14
2.1. Pharma 4.0.....	14
2.2. Η τεχνολογία «Blockchain».....	17
2.3. Ιδιότητες της τεχνολογίας «Blockchain».....	19
2.4. Περιορισμοί της τεχνολογίας «Blockchain».....	21
Κεφάλαιο 3° - Υλοποίηση Συστήματος Ιχνηλασιμότητας.....	23
3.1. Προγραμματιστικές Επιλογές.....	23
3.1.1. Python (ΓΠ).....	23
3.1.2. JSON (ΒΔ).....	23
3.1.3. Flask (Web Framework).....	24
3.1.4. Frontend (HTML, CSS & JS).....	24
3.2. Κώδικας Εφαρμογής.....	24
3.2.1. Data.py.....	25
3.2.1.1. Class Data.....	25
3.2.2. Blockchain.py.....	27
3.2.2.1. Class Block.....	27
3.2.2.2. Class Blockchain (1/2).....	28
3.2.2.3. Class Blockchain (2/2).....	31
3.2.3. Blockchain.json.....	35
3.2.4. Main.py.....	37
3.2.4.1. Balances.....	41
3.2.4.2. Distributor's Transaction History.....	42
3.3. Κώδικας Διεπαφής.....	45
3.3.1. App.py.....	45
3.3.1.1. '/blockchain', '/blockchain_json'.....	46
3.3.1.2. '/add_tx', '/submit_form'.....	47
3.3.1.3. '/trace', '/search_bn'.....	50
3.3.2. Viewer.js.....	51
3.3.3. Add_TX.js.....	54

3.3.4. Trace.js.....	57
3.4. Εκτέλεση Συστήματος Ιχνηλασιμότητας.....	59
Κεφάλαιο 4^ο - Παρουσίαση Συστήματος Ιχνηλασιμότητας.....	62
4.1. Αρχική Σελίδα – Περιηγητής Blockchain.....	62
4.2. Φόρμα Συναλλαγής (Προσθήκη).....	64
4.3. Ιχνηλάτηση Συναλλαγής (Φαρμάκου).....	71
Κεφάλαιο 5^ο - Συμπεράσματα.....	74
5.1. Συμπεράσματα.....	74
5.2. Μελλοντικές Προεκτάσεις (Συστήματος Ιχνηλασιμότητας).....	75
5.2.1. Υλοποίηση Κρυπτογραφημένης Βάσης Δεδομένων.....	76
5.2.2. Αυθεντικοποίηση και Επίπεδα Πρόσβασης Χρηστών.....	76
5.2.3. «Έξυπνα» Συμβόλαια και Αυτοματισμός.....	77
5.2.4. Βελτιστοποίηση της Γραφικής Διεπαφής.....	77
Βιβλιογραφία.....	79

Abstract

The authenticity and security of pharmaceuticals have never been more critical than in today's complicated global supply chains. The importance of traceability in the pharmaceutical supply chain is prevalent throughout this thesis, discussing demand and applications, feasible options, and improvements via groundbreaking technologies.

The first chapter defines the case for supply chain management and explores the complexities and demands for traceability from the pharmaceutical supply chain.

Subsequently, the second chapter explains how innovative technologies are rising to the traceability challenges and introduces Blockchain technology, which presents a potential answer for improving drug traceability by explaining its underlying theory, and potential applications.

The remaining chapters describe the development and deployment of an application written in Python programming language, which demonstrates the use of Blockchain in a traceability scenario along with a front-end web application using Flask to enhance its use-case.

Through experimentation, this research demonstrates the capabilities of Blockchain technology in bolstering traceability within the pharmaceutical domain.

Finally, the thesis concludes with insightful reflections on the key findings along with areas for improvement in the developed application.

Κεφάλαιο 1^ο - Η Ιχνηλασιμότητα στην Εφοδιαστική Αλυσίδα

1.1. Ορολογία της Εφοδιαστικής Αλυσίδας και η διαχείριση της

Με τον όρο εφοδιαστική αλυσίδα χαρακτηρίζεται η διαδικασία διάθεσης μιας υπηρεσίας ή ενός προϊόντος από τον προμηθευτή μέχρι τον καταναλωτή με τον πιο αποτελεσματικό, οικονομικά αποδοτικό και έγκαιρο τρόπο ώστε να ικανοποιούνται οι ανάγκες της αγοράς αλλά και οι απαιτήσεις των πελατών. Στη διαδικασία αυτή εμπλέκονται ανεξάρτητες οντότητες, με τις οποίες συντίθεται η λεγόμενη αλυσίδα εφοδιασμού που αποτελείται μεταξύ άλλων από τους προμηθευτές, τα εργοστάσια, τις αποθήκες, τα κέντρα διανομής, τους εμπόρους λιανικής και τους καταναλωτές.⁽¹⁾ Σε μια άλλη ορολογία, η αλυσίδα εφοδιασμού ορίζεται ως ένα μεγάλο δίκτυο μεταξύ των οντοτήτων όπου εμπλέκονται μέσω διασυνδέσεων στις διάφορες διαδικασίες και δραστηριότητες από τις οποίες παράγεται αξία με τη μορφή προϊόντων και υπηρεσιών με τελικό αποδέκτη τον καταναλωτή. Οι δραστηριότητες αυτές διακρίνονται στα μέρη της εφοδιαστικής αλυσίδας που σχετίζονται με την απόκτηση των απαραίτητων πρώτων υλών και πόρων (διαδικασίες upstream) αλλά και σε εκείνα τα μέρη της εφοδιαστικής αλυσίδας που εμπλέκονται στην παραγωγή, τη διανομή και την πώληση των τελικών προϊόντων ή υπηρεσιών (διαδικασίες downstream) σε πελάτες και σε καταναλωτές.⁽²⁾ Σύμφωνα με κάποιους ερευνητές, μια αλυσίδα εφοδιασμού ορίζεται ως «το σύνολο τριών ή περισσότερων οντοτήτων που εμπλέκονται άμεσα στη διάθεση προϊόντων, υπηρεσιών, οικονομικών στοιχείων και πληροφοριών από ένα σημείο προέλευσης σε έναν καταναλωτή».⁽¹⁾

Στη βιβλιογραφία των τελευταίων δεκαετιών, με την έννοια εφοδιαστική αλυσίδα περιγράφεται επί της ουσίας το κανάλι διανομής των παρεχόμενων προϊόντων και υπηρεσιών, ενώ πραγματοποιείται σαφής διάκριση στη διαχείριση της εφοδιαστικής αλυσίδας (Supply Chain Management, SCM) καθώς για αυτή απαιτούνται σημαντικές διαχειριστικές προσπάθειες από την κάθε οντότητα ξεχωριστά εντός της εφοδιαστικής αλυσίδας. Σύμφωνα με ορισμένους ερευνητές, στον όρο διαχείριση της εφοδιαστικής αλυσίδας αποδίδεται «ο συστημικός, στρατηγικός συντονισμός των επιχειρηματικών λειτουργιών μιας και κατ' επέκταση όλων των οντοτήτων εντός της αλυσίδας εφοδιασμού με σκοπό τη βελτίωση της μακροπρόθεσμης απόδοσης της κάθε οντότητας αλλά και της εφοδιαστικής αλυσίδας στο σύνολό της».⁽¹⁾ Αντίστοιχα, σύμφωνα με διαφορετικές βιβλιογραφικές ανασκοπήσεις, η διαχείριση της εφοδιαστικής αλυσίδας πρόκειται για «μια ενσωματωτική διαδικαστική φιλοσοφία σχετικά με τη διαχείριση της συνολικής ροής του καναλιού διανομής από τον προμηθευτή έως και τον τελικό καταναλωτή».⁽³⁾

Ιστορικά, η έννοια της διαχείρισης της εφοδιαστικής αλυσίδα (SCM) επινοήθηκε για πρώτη φορά από έναν Αμερικανό σύμβουλο βιομηχανίας στις αρχές της δεκαετίας του 1980. Ωστόσο, στην πράξη, τέτοιου είδους πρακτικές διαχείρισης της εφοδιαστικής

αλυσίδας παρατηρήθηκαν αρκετά νωρίτερα, ιδίως με την έλευση της πρώτης γραμμής παραγωγής. Στη συνέχεια, ακολούθησε μια περίοδος ενσωμάτωσης των πρακτικών διαχείρισης της εφοδιαστικής αλυσίδας με τη δημιουργία των συστημάτων ηλεκτρονικής ανταλλαγής δεδομένων (EDI) και την αφομοίωση από τις επιχειρήσεις των συστημάτων προγραμματισμού εταιρικών πόρων (ERP). Ο αντίκτυπος της παγκοσμιοποίησης με τη διασυνοριακή διακίνηση προϊόντων, υπηρεσιών, τεχνολογίας και πληροφοριών είχε ως αποτέλεσμα τη στρατηγική εστίαση της διαχείρισης της εφοδιαστικής αλυσίδας στην αύξηση του ανταγωνιστικού πλεονεκτήματος, δημιουργώντας προστιθέμενη αξία και μειώνοντας το κόστος παραγωγής μέσω της παγκόσμιας προμήθειας. Κατόπιν, η περίοδος εξειδίκευσης που μεσολάβησε τη δεκαετία του 1990, βρήκε τις βιομηχανίες να εγκαταλείπουν τις πρακτικές της καθετοποίησης των εφοδιαστικών τους διαδικασιών αναθέτοντας τις μη-βασικές τους λειτουργίες για την κατασκευή και τη διανομή των προϊόντων σε εξωτερικές εταιρείες. Η περίοδος της εξειδίκευσης ολοκληρώνεται με την εισαγωγή των πρακτικών σχεδιασμού για τον εφοδιασμό και τη συγκριτική αξιολόγηση των επιδόσεων της διαχείρισης της εφοδιαστικής αλυσίδας της κάθε βιομηχανίας. Τέλος, με την έλευση του Παγκόσμιου Ιστού (WWW, 2.0), στη διαχείριση της εφοδιαστικής αλυσίδας περιλαμβάνεται η μόχλευση προηγμένων τεχνολογιών και συνεργατικών πλατφορμών για τη βελτίωση της αποτελεσματικότητας, της ορατότητας αλλά και της ευελιξίας των διαδικασιών της εφοδιαστικής αλυσίδας.⁽⁴⁾

Η διανομή των εμπορευμάτων στην αλυσίδα εφοδιασμού δεν περιορίζεται αποκλειστικά στην κατεύθυνση από τον προμηθευτή προς τον τελικό καταναλωτή. Υπάρχουν αρκετές περιπτώσεις για τις οποίες τα αγαθά κινούνται προς την αντίθετη κατεύθυνση στην εφοδιαστική αλυσίδα, όπως είναι η ανακύκλωση, η ανακατασκευή, η ανάκληση ή η συντήρηση και η επισκευή τους. Η αντίστροφη εφοδιαστική αναφέρεται σε μια σειρά από δυνατότητες για τη διευκόλυνση της μετακίνησης των προϊόντων από τον τελικό καταναλωτή προς τον προμηθευτή. Σε αυτές τις δραστηριότητες περιλαμβάνεται η διαχείριση επιστροφών των προϊόντων, η ανακύκλωση, η ανακατασκευή, η ανάκληση και η επαναχρησιμοποίηση των πρώτων υλών και πόρων.⁽⁵⁾

Στη σημερινή παγκοσμιοποιημένη αγορά προϊόντων και υπηρεσιών, υπάρχουν αμέτρητες εναλλακτικές διαμορφώσεις της τελικής αλυσίδας εφοδιασμού. Συνεπώς, ένας οποιοσδήποτε οργανισμός δύναται να είναι μέλος πολλών αλυσίδων εφοδιασμού. Για παράδειγμα, ο αμερικανικός πάροχος τηλεπικοινωνιών AT&T δύναται να έχει τον κατασκευαστή ηλεκτρονικών ειδών ευρείας κατανάλωσης Motorola ως πελάτη σε μια αλυσίδα εφοδιασμού, ως συνεργάτη σε μια άλλη, ως προμηθευτή σε μια τρίτη και ως ανταγωνιστή σε μία ακόμη τέταρτη αλυσίδα εφοδιασμού.⁽¹⁾ Στις σύγχρονες μελέτες σχετικά με τη συγκριτική αξιολόγηση των οργανισμών, αναδεικνύονται σημαντικές διαφορές κόστους μεταξύ των εταιρειών στις οποίες παρουσιάζονται οι καλύτερες επιδόσεις για τον τομέα τους συγκριτικά με τις μέτριες επιδόσεις των υπολοίπων. Αυτό σημαίνει πως οι εταιρείες των οποίων πρωταρχικό μέλημα είναι η βελτιστοποίηση της διαχείρισης της εφοδιαστικής τους αλυσίδας, επρόκειτο να αποτελέσουν τις νέες

ιστορίες επιτυχίας στην παγκόσμια αγορά των προϊόντων, των υπηρεσιών και των πληροφοριών.⁽⁴⁾

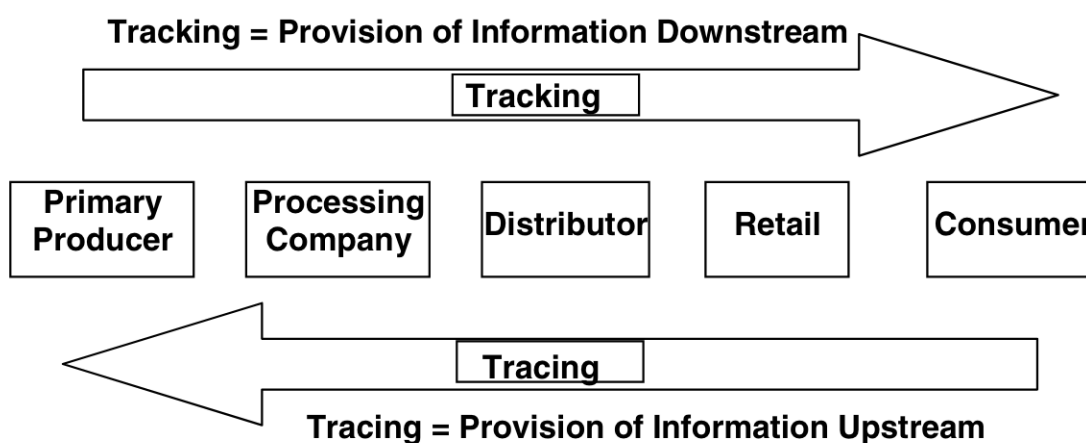
1.2. Ιχνηλασιμότητα στην Αλυσίδα Εφοδιασμού

Με τον όρο ιχνηλασιμότητα χαρακτηρίζεται η δυνατότητα παρακολούθησης μιας παρτίδας προϊόντος και του ιστορικού της στο σύνολο, σε μια αλυσίδα παραγωγής από τη συγκομιδή έως τη μεταφορά, την αποθήκευση, τη μεταποίηση, τη διανομή και την πώληση ή μεμονωμένα σε οποιοδήποτε στάδιο της εφοδιαστικής αλυσίδας.⁽⁶⁾ Σε μια άλλη ορολογία, η ιχνηλασιμότητα ορίζεται ως «η δυνατότητα παρακολούθησης της διακίνησης προς τα εμπρός μέσω καθορισμένων σταδίων της εκτεταμένης εφοδιαστικής αλυσίδας και η ανίχνευση προς τα πίσω του ιστορικού, της χρήσης ή της γεωγραφικής θέσης του αντικειμένου που βρίσκεται υπό εξέταση».⁽⁷⁾ Οποιαδήποτε ενέργεια πραγματοποιείται σε ένα προϊόν πριν την κατασκευή, κατά τη διάρκεια της κατασκευής, μετά την κατασκευή, τη συσκευασία και τη διανομή, θεωρείται ως ιχνηλασιμότητα. Σε αυτή περιλαμβάνονται, τα συστατικά πρώτων υλών, τα αποτελέσματα κλινικών δοκιμών, τα χαρακτηριστικά περιβάλλοντος όπως θερμοκρασία και υγρασία, οι χρησιμοποιούμενοι πόροι όπως ο ανθρώπινος παράγοντας ή οι μηχανές, οι τρόποι μεταφοράς και τα χρονοδιαγράμματα. Στην πραγματικότητα όμως, η ιχνηλάτηση πρόκειται για μια διαδικασία τήρησης αρχείων, που είναι υπεύθυνη για την εμφάνιση της διαδρομής μιας ορισμένης μονάδας ή παρτίδας προϊόντος ή συστατικού από τους προμηθευτές προς όλα τα στάδια που πραγματοποιείται η επεξεργασία με οριστική κατάληξη τους πελάτες και τους τελικούς καταναλωτές. Σε αυτά τα συστήματα ιχνηλασιμότητας, η καταγραφή των πληροφοριών γίνεται με μεγάλη ακρίβεια, καθώς παρουσιάζεται μεγάλος βαθμός εμπιστοσύνης αλλά και ευθύνης για τον κάθε χειριστή του συστήματος ιχνηλασιμότητας. Συνεπώς, σε ένα αποτελεσματικό σύστημα παρακολούθησης και ανίχνευσης, παρέχονται οι δυνατότητες της μεταφόρτωσης, της διατήρησης και της ανάγνωσης αυτών των πληροφοριών με τυποποιημένο τρόπο. Εντούτοις, εξαιτίας του μη-συνεκτικού προσδιορισμού των εννοιών της παρακολούθησης και της ιχνηλάτησης, δύναται να ανακύψουν συστήματα ιχνηλασιμότητας με διαφορετικές απαιτήσεις λειτουργίας μεταξύ τους, ενισχύοντας την ασυμβατότητα μεταξύ των συστημάτων αυτών.⁽⁸⁾

Η σύγχρονη αγορά προϊόντων και υπηρεσιών παγκοσμιοποιείται ολοένα και περισσότερο με αποτέλεσμα οι αλυσίδες εφοδιασμού να γίνονται μεγαλύτερες και πιο σύνθετες στην λειτουργικότητά τους. Δεν θεωρείται πια ασυνήθιστο για να ένα προϊόν να παράγεται στην Κίνα, να συσκευάζεται στις ΗΠΑ και να πωλείται στη Δανία. Εξαιτίας αυτής της κατάστασης, σημειώνονται διαρκώς νέες προκλήσεις στην αλυσίδα εφοδιασμού, για παράδειγμα όταν τα προϊόντα δεν ανταποκρίνονται στις κανονιστικές ρυθμίσεις ή όταν αποτυγχάνουν στους ποιοτικούς ελέγχους. Σε αυτή την περίπτωση, για

την ανάκληση ενός προϊόντος απαιτείται τόσο η εύρεση της αιτίας του προβλήματος, όσο και ο εντοπισμός των επηρεαζόμενων φορέων στην εφοδιαστική αλυσίδα.

Παράλληλα, παρουσιάζεται αυξημένη ζήτηση από τους ίδιους τους καταναλωτές για την παροχή πληροφοριών για τα προϊόντα σχετικά με την προέλευσή τους, τη γνησιότητα, ακόμη και τις κοινωνικές τους επιπτώσεις όπως είναι οι περιβαλλοντικές συνθήκες παραγωγής αλλά και οι ηθικές συνθήκες εργασίας των ανθρώπων κατά την παραγωγή τους.⁽⁹⁾ Συνεπώς, ασκείται πίεση στις βιομηχανίες από τους καταναλωτές με στόχο την εξασφάλιση της διαφάνειας και της ακεραιότητας σε ολόκληρη την αλυσίδα εφοδιασμού. Για την διατήρηση της εμπιστοσύνης των καταναλωτών και των ενδιαφερομένων μελών της εφοδιαστικής αλυσίδας, οι οργανισμοί πρέπει να συμμορφώνονται με τις νέες αυτές απαιτήσεις.⁽¹⁰⁾



Εικόνα 1.1.: Οι ροές πληροφοριών που σχετίζονται με τις διαδικασίες της παρακολούθησης και της ιχνηλάτησης στις αλυσίδες εφοδιασμού, *Schwägele (2005)*

Για να ανιχνεύονται τα προϊόντα στην αλυσίδα εφοδιασμού, συλλέγονται οι σχετικές πληροφορίες κατά τη διάρκεια όλων των σταδίων παραγωγής, από το στάδιο της πρωτογενούς παραγωγής μέχρι την επεξεργασία, τη διανομή, την λιανική πώληση έως και τον τελικό καταναλωτή. Σύμφωνα με το πλαίσιο της εικόνας 1.1., η ανιχνευσιμότητα διαχωρίζεται σε δύο βασικές λειτουργίες, την παρακολούθηση και την ιχνηλάτηση. Η παρακολούθηση ορίζεται ως η δυνατότητα ακολούθησης της πορείας ενός αντικειμένου στις διαδικασίες downstream της αλυσίδας εφοδιασμού από την αρχή μέχρι το τέλος. Αντίστοιχα, η ιχνηλάτηση ορίζεται ως η δυνατότητα αναγνώρισης της προέλευσης ενός αντικειμένου ή ομάδας αντικειμένων μέσω των καταγραφών στις διαδικασίες upstream της αλυσίδας εφοδιασμού. Με την ιχνηλασιμότητα, εξασφαλίζεται η διαφάνεια στις διαδικασίες των βιομηχανιών καθώς καθίσταται εφικτή η εύρεση της πηγής όλων των συστατικών ενός προϊόντος κατά την επεξεργασία του, με αποτέλεσμα να πραγματοποιούνται όλες οι απαιτούμενες διαρθρωτικές ενέργειες στα διάφορα στάδια της επεξεργασίας του και να διασφαλίζεται τελικά η βιωσιμότητά του στην εφοδιαστική αλυσίδα. Τέλος, η πρόσβαση στις πληροφορίες ιχνηλασιμότητας είναι

απαραίτητη και για τις κανονιστικές αρχές, που είναι επιφορτισμένες με τους τακτικούς ελέγχους συμμόρφωσης των προϊόντων σύμφωνα με τους υγειονομικούς κανονισμούς.⁽¹¹⁾

1.3. Η Εφοδιαστική Αλυσίδα στον τομέα της Υγείας

Η αλυσίδα εφοδιασμού των φαρμακευτικών σκευασμάτων γνωστή και ως PSC (Pharmaceutical SC), είναι όλες οι διαδικασίες με τις οποίες τα συνταγογραφημένα φάρμακα παράγονται, αποθηκεύονται και διανέμονται στους ανθρώπους που τα χρειάζονται. Σε αυτή την αλυσίδα περιλαμβάνονται όλα τα διαφορετικά μέρη και οι δραστηριότητες που απαιτούνται για τη δημιουργία ενός φαρμακευτικού σκευάσματος, ξεκινώντας από την αρχική κατάσταση της ανάπτυξης του και το στάδιο των κλινικών δοκιμών. Η σύνθεση της φαρμακευτικής αλυσίδας εφοδιασμού είναι αρκετά πολύπλοκη καθώς εμπλέκονται πολλές οντότητες και αρμόδιοι φορείς, ο καθένας με διαφορετικές υποχρεώσεις με σκοπό τη διασφάλιση της διαθεσιμότητας φαρμάκων τα οποία είναι επαρκώς δοκιμασμένα και ασφαλή για μαζική χρήση. Το διακύβευμα στις φαρμακευτικές αλυσίδες εφοδιασμού είναι μεγάλο καθώς η απόδοση αυτών των δικτύων είναι άμεσα συνδεδεμένη με το αγαθό της υγείας των ανθρώπων και συνεπώς όλοι οι εμπλεκόμενοι φορείς είναι υπεύθυνοι για την εξασφάλιση τόσο της καλής λειτουργίας των φαρμάκων, όσο και για τη συμμόρφωση με τις νομικές υποχρεώσεις. Συγχρόνως, στη φαρμακευτική αλυσίδα εφοδιασμού, παρατηρείται το υψηλό κόστος έρευνας και ανάπτυξης, παράλληλα με τις ραγδαίες εναλλαγές στις τεχνολογικές υποδομές της βιομηχανίας με αποτέλεσμα να επηρεάζεται η μέθοδος με την οποία δημιουργούνται τα φάρμακα, το κόστος παραγωγής τους αλλά και ο ρυθμός άφιξής τους στους ανθρώπους που τα χρειάζονται.⁽¹²⁾

Η φαρμακευτική βιομηχανία κινείται κατά μεγάλο βαθμό εξαιτίας της επείγουσας ανάγκης για την κατασκευή νέων φαρμάκων και τη γρήγορη διάθεσή τους στην αγορά. Η αρχική επιτυχία ενός νεοεισαχθέντος φαρμάκου είναι υπεύθυνη για τη δημιουργία σημαντικών κερδών για την κατασκευαστική εταιρεία ελλείψει του ανταγωνισμού. Εντούτοις, οι χρονικές προθεσμίες των βιομηχανιών δίχως ανταγωνισμό ελαττώνονται σταδιακά από τα πέντε έτη σε μόλις δύο ή ένα. Εξαιτίας των πιθανών υγειονομικών κινδύνων που εμπλέκονται στην κατασκευή νέων φαρμάκων, η αγορά αυτή είναι ρυθμισμένη με αυστηρούς κανονισμούς από την Αμερικανική Υπηρεσία Τροφίμων και Φαρμάκων (FDA) με σκοπό την εξασφάλιση τόσο της αποτελεσματικότητας όσο και της ασφάλειας των φαρμακευτικών σκευασμάτων. Αρμόδιοι φορείς στην φαρμακευτική αλυσίδα εφοδιασμού ισχυρίζονται πως οι διαδικασίες συμμόρφωσης με τις κανονιστικές ρυθμίσεις είναι τόσο χρονοβόρες και δαπανηρές για τις φαρμακευτικές εταιρείες ώστε να περιορίζεται ουσιαστικά το περιθώριο καινοτομίας με αποτέλεσμα να παρατηρούνται καθυστερήσεις στις διαδικασίες της έρευνας και της ανάπτυξης, του σχεδιασμού νέων εγκαταστάσεων και στις εγκρίσεις νέων πιστοποιήσεων.⁽¹³⁾

Η κατανόηση του κύκλου ζωής ενός φαρμακευτικού σκευάσματος είναι ζωτικής σημασίας για την παρακολούθηση της πορείας του στη φαρμακευτική εφοδιαστική αλυσίδα. Αρχικά, στο στάδιο της έρευνας, εξετάζονται χιλιάδες εθελοντές στην χημική ουσία του φαρμακευτικού σκευάσματος υπό ανάπτυξη, μιας διαδικασίας με χρονική διάρκεια έως και μιας δεκαετίας πριν από την πιθανή κυκλοφορία του σκευάσματος στην αγορά. Παράλληλα με την κυκλοφορία του, εξασφαλίζεται και η μοναδικότητα του φαρμάκου με την προστασία της ευρεσιτεχνίας του. Ακολούθως, στο στάδιο δοκιμών, πραγματοποιούνται αυστηρές κλινικές δοκιμές για την αξιολόγηση της τοξικότητας του φαρμάκου, αλλά και την ικανότητά του στην καταπολέμηση της επικείμενης ασθένειας. Στο τελευταίο στάδιο της ανάπτυξης, επιλέγεται η βιωσιμότερη βιοχημική οδός μαζί με τη σχετική διαδικασία κατασκευής για την παραγωγή του φαρμάκου, μιας διαδικασίας με χρονική διάρκεια από έξι έως και οκτώ χρόνια.⁽¹³⁾

Σε μια τυπική φαρμακευτική αλυσίδα εφοδιασμού, περιλαμβάνονται συνήθως η πρωτογενής παραγωγή αξιοποιώντας και τον εργοστασιακό εξοπλισμό, η δευτερογενής παραγωγή, οι αποθήκες, τα κέντρα διανομής, οι χονδρέμποροι, οι λιανοπωλητές και τα νοσοκομεία.⁽¹³⁾

Η πρωτογενής μονάδα παραγωγής είναι υπεύθυνη για την παραγωγή της δραστικής ουσίας του φαρμακευτικού σκευάσματος, μια διαδικασία που χαρακτηρίζεται από μεγάλες διάρκειες εργασιών που συνήθως οργανώνονται σε βάρδιες. Τα ενδιάμεσα προϊόντα της παραγωγής, υποβάλλονται σε ποιοτικούς ελέγχους με καθορισμένες και επικυρωμένες διαδικασίες για την πρόληψη της μόλυνσης των προϊόντων. Ωστόσο, οι παρατεταμένες διακοπές για τους ελέγχους μεταξύ των παραγωγών είναι υπεύθυνες για την μείωση της απόκρισης και στη συμβολή της χαμηλότερης απόδοσης της αλυσίδας εφοδιασμού στη φαρμακευτική βιομηχανία. Το πρόβλημα αυτό της απόδοσης εντίνεται όταν χρησιμοποιούνται εξωτερικοί συνεργάτες για την παραγωγή ενός τμήματος ή ολόκληρων των δραστικών ουσιών. Συνεπώς, θεωρείται πως η αλυσίδα εφοδιασμού των φαρμάκων είναι μορφολογικά εκτεταμένη, με όσα προβλήματα συντονισμού συνεπάγονται σε αυτές τις κατηγορίες.⁽¹³⁾

Στη δευτερογενή παραγωγή, λαμβάνεται η δραστική ουσία της πρωτογενούς παραγωγής και σε συνδυασμό με αδρανή υλικά, πραγματοποιούνται πρόσθετα στάδια επεξεργασίας όπως για παράδειγμα η κοκκοποίηση, η συμπίεση, η επικάλυψη, οι έλεγχοι ποιότητας και η διαδικασία της συσκευασίας, καταλήγοντας στην τελική εκδοχή του προϊόντος, σε μορφή SKU, όπως είναι τα χάπια. Οι εγκαταστάσεις δευτερογενούς παραγωγής βρίσκονται συχνά σε διαφορετικές περιοχές συγκριτικά με τις τοποθεσίες της πρωτογενούς παραγωγής, πρακτική που βασίζεται σε στρατηγικές βελτιστοποίησης της φορολογίας αλλά και του κόστους μεταφοράς εντός της φαρμακευτικής αλυσίδας εφοδιασμού. Τέλος, χαρακτηριστικός είναι και ο ρόλος των χονδρεμπόρων καθώς αν λίγοι αριθμητικά, μέσω αυτών καλύπτεται το μεγαλύτερο μέρος της ζήτησης φαρμάκων ώστε το υπολειπόμενο ποσοστό να ισομοιράζεται στα νοσοκομεία.⁽¹³⁾

1.4. Τα Στάδια στη Φαρμακευτική Αλυσίδα Εφοδιασμού

Η πολυπλοκότητα της φαρμακευτικής εφοδιαστικής αλυσίδας εξαρτάται τόσο από τον αριθμό των εμπλεκόμενων αρμόδιων φορέων, όσο κι από την ποικιλομορφία των διαδικασιών παραγωγής φαρμάκων που εμπιέρονται σε αυτή. Σύμφωνα με συγκεκριμένες έρευνες, η αλυσίδα εφοδιασμού ταξινομείται σε τρία επίπεδα ανάλογα με την πολυπλοκότητά της. Στο πρώτο επίπεδο, βρίσκεται η άμεση αλυσίδα στην οποία περιλαμβάνονται οι πελάτες, οι προμηθευτές κι ένας κεντρικός οργανισμός. Ύστερα, στο δεύτερο επίπεδο, βρίσκεται η εκτεταμένη αλυσίδα στην οποία περιλαμβάνονται οι προμηθευτές του άμεσου προμηθευτή, οι πελάτες των άμεσων πελατών και ο κεντρικός οργανισμός. Στο στο τρίτο και τελευταίο επίπεδο βρίσκεται η απόλυτη αλυσίδα στην οποία συμπεριλαμβάνονται οι οντότητες με ρόλο μεσάζοντα στις οργανωσιακές σχέσεις όπως είναι μεταξύ άλλων οι εταιρείες για την έρευνα αγοράς, τη διαφήμιση, και τη διοικητική υποστήριξη.⁽¹⁾ Συνήθως, εξαιτίας της πολυπλοκότητας του κύκλου ζωής ενός φαρμακευτικού σκευάσματος, οι αλυσίδες εφοδιασμού των φαρμάκων θεωρούνται κατ' ελάχιστον ως μορφολογικά εκτεταμένες ή απόλυτες.⁽¹⁴⁾



FIGURE 1a - DIRECT SUPPLY CHAIN



FIGURE 1b - EXTENDED SUPPLY CHAIN

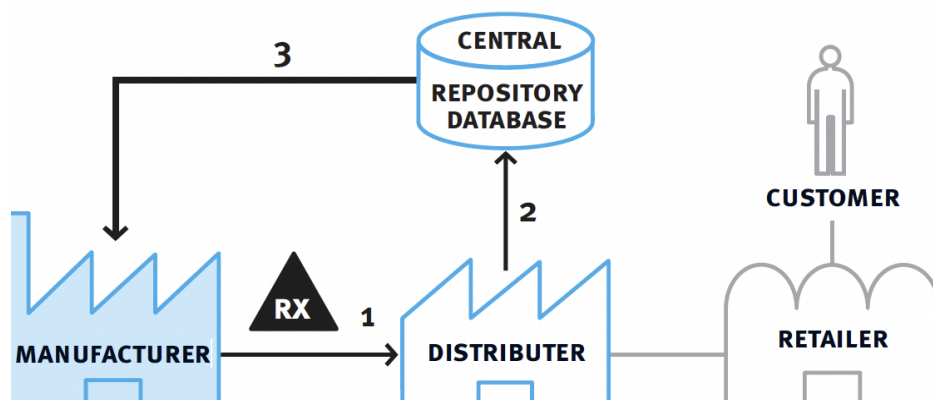


FIGURE 1c - ULTIMATE SUPPLY CHAIN

Εικόνα 1.2.: Τα επίπεδα της αλυσίδας εφοδιασμού, *Mentzer et al. (2001)*

Ανεξαρτήτως του σχήματος μιας τυπικής αλυσίδας εφοδιασμού, στον κλάδο των φαρμάκων παρατηρείται μια ακόμη διακριτή διαίρεση σε τρεις βαθμίδες. Η πρώτη βαθμίδα αντιστοιχίζεται με τις διαδικασίες "upstream" της προμήθειας. Έπειτα, στη δεύτερη βαθμίδα καλύπτονται οι διαδικασίες της και μεταφοράς και της διανομής.

Η τρίτη και η τελευταία βαθμίδα αντιστοιχίζεται με τις διαδικασίες "downstream" της κατανάλωσης.⁽¹⁴⁾



Εικόνα 1.3.: Μια τυπική φαρμακευτική εφοδιαστική αλυσίδα με τις βασικές διαδικασίες της, *Koh et al. (2003)*

Η φυσική μορφή ενός φαρμακευτικού προϊόντος δύναται να μεταβάλλεται εντός της πορείας του στην αλυσίδα εφοδιασμού. Μετά το πέρας ενός συγκεκριμένου σταδίου, το προϊόν θεωρείται ως τελικό για το στάδιο αυτό και στη συνέχεια εισέρχεται ως έχει στο επόμενο στάδιο της αλυσίδας. Σε ένα καθολικό στιγμιότυπο μιας φαρμακευτικής αλυσίδας, οι πληροφορίες κατηγοριοποιούνται σε αμετάβλητες ή αλλιώς στατικές (π.χ. το μέγεθος και το βάρος ενός προϊόντος), ημιδυναμικές (μεταβαλλόμενες σε μεγάλα διαστήματα, π.χ. αριθμοί παρτίδων ανά παραγωγή) και δυναμικές (μεταβαλλόμενες σε σύντομο χρονικό διάστημα, π.χ. ενδείξεις θέσης, θερμοκρασίας, πίεσης και υγρασίας.). Ως απάντηση στην αυξανόμενη ευαισθητοποίηση σχετικά με τα πλαστά φάρμακα, η Αμερικανική Υπηρεσία Τροφίμων και Φαρμάκων (FDA), έχει προωθήσει νομοθεσία για την αντιμετώπιση αυτού του ζητήματος. Σκοπός του νομοσχεδίου είναι η επαλήθευση της γνησιότητας των φαρμακευτικών σκευασμάτων και η ελαχιστοποίηση του κινδύνου παραποίησης. Σε αυτό ορίζονται οι πληροφορίες που πρέπει να αναγράφονται σε κάθε φάρμακο, στο κάθε στάδιο της εφοδιαστικής αλυσίδας. Ακολούθως, παρουσιάζονται οι πληροφορίες αυτές.⁽¹⁵⁾

ΠΛΗΡΟΦΟΡΙΑ	ΠΕΡΙΓΡΑΦΗ
1. Πληροφορίες Προϊόντος (Product Information): — Όνομα Προϊόντος (Product Name) — Κωδικός Προϊόντος (Product Code/SKU) — Αριθμός Παρτίδας (Batch/Lot Number)	— Το όνομα του φαρμακευτικού προϊόντος. — Ένα μοναδικό αναγνωριστικό που εκχωρείται στη συγκεκριμένη παραλλαγή προϊόντος ή συσκευασία. — Ο αριθμός παρτίδας που σχετίζεται

— Ημερομηνία Λήξης (Expiry Date)	με το διανεμόμενο προϊόν. — Η ημερομηνία πέραν της οποίας το προϊόν δεν πρέπει να χρησιμοποιηθεί.
2. Τεκμήρια Κατασκευής και Ποιοτικού Ελέγχου (Manufacturing and Quality Control Records) — Ημερομηνία Κατασκευής (Manufacturing Date) — Μονάδα Παραγωγής (Manufacturing Facility) — Αποτελέσματα Δοκιμών Ποιοτικού Ελέγχου (Quality Control Test Results) — Πρότυπες Διαδικασίες Λειτουργίας (Standard Operating Procedures/SOPs)	— Η ημερομηνία κατασκευής του προϊόντος. — Πληροφορίες σχετικά με την εγκατάσταση όπου κατασκευάστηκε το προϊόν, συμπεριλαμβανομένου του ονόματος και της τοποθεσίας του. — Αποτελέσματα διαφόρων δοκιμών ποιοτικού ελέγχου που πραγματοποιήθηκαν κατά τη διαδικασία κατασκευής, διασφαλίζοντας ότι το προϊόν πληροί προκαθορισμένες προδιαγραφές. — Λεπτομερής τεκμηρίωση των τυπικών διαδικασιών λειτουργίας που ακολουθούνται κατά τις διαδικασίες κατασκευής, συσκευασίας και ποιοτικού ελέγχου.
3. Αρχεία Διανομής (Distribution Records): — Ημερομηνίες Διανομής (Distribution Dates) — Τοποθεσίες Διανομής (Distribution Locations) — Ποσότητες Διανομής (Distribution Quantities)	— Ημερομηνίες κατά τις οποίες το προϊόν διανεμήθηκε σε διαφορετικές τοποθεσίες ή πελάτες. — Πληροφορίες σχετικά με τις τοποθεσίες όπου διανεμήθηκε το προϊόν, όπως αποθήκες, διανομείς ή εγκαταστάσεις υγειονομικής περίθαλψης. — Οι ποσότητες του προϊόντος που διανέμονται σε κάθε τοποθεσία ή πελάτη.
4. Τεκμηρίωση Κανονιστικής Συμμόρφωσης (Regulatory Compliance Documentation): — Κανονιστικές Εγκρίσεις (Regulatory Approvals) — Έγγραφα Συμμόρφωσης (Compliance Documents)	— Αντίγραφα ρυθμιστικών εγκρίσεων που έχουν ληφθεί για το προϊόν, συμπεριλαμβανομένων των επιστολών έγκρισης, πιστοποιητικών ή αδειών. — Τεκμηρίωση που αποδεικνύει τη συμμόρφωση με ρυθμιστικές απαιτήσεις, όπως καλές πρακτικές παραγωγής (GMP), συστήματα διαχείρισης ποιότητας και ρυθμιστικούς ελέγχους.

Πίνακας 1.1.: Τεκμηρίωση και Τήρηση Αρχείου Διαδικασιών σε μια τυπική φαρμακευτική αλυσίδα εφοδιασμού, Koh et al. (2003)

Κεφάλαιο 2ο - Οι Αναδυόμενες Τεχνολογίες στη Φαρμακευτική Εφοδιαστική Αλυσίδα

2.1. Pharma 4.0

Με τον όρο «Industry 4.0» προσδιορίζεται ως η μέχρι σήμερα τελευταία φάση της βιομηχανικής επανάστασης, που χαρακτηρίζεται από την ενσωμάτωση κυβερνοφυσικών συστημάτων (CPS) στη διαδικασία κατασκευής. Με αυτή την προσέγγιση, ενισχύεται η «νοημοσύνη» των διαδικασιών παραγωγής μέσω των διασυνδεδεμένων συστημάτων, του αυτοματισμού, της μηχανικής μάθησης και της παρακολούθησης σε πραγματικό χρόνο. Αν και «Industry 4.0» σηματοδοτείται για πρώτη φορά στον μεταποιητικό τομέα, είναι σημαντικές οι επιδράσεις του που παρατηρούνται και στις αλυσίδες εφοδιασμού. Στο «Pharma 4.0» εκδηλώνονται οι αρχές του «Industry 4.0» για τη διαχείριση της, καθώς η προσοχή του φαρμακευτικού τομέα επικεντρώνεται κυρίως στη χρήση ιατρικών κυβερνοφυσικών συστημάτων σε κάθε στάδιο της ανάπτυξης φαρμάκων. Γνωστά και ως MCPS, πρόκειται για «έξυπνα» συστήματα που αποτελούνται από ιατρικά όργανα και συσκευές, διασυνδεδεμένες μεταξύ τους για την αποτελεσματικότερη θεραπεία των ασθενών. Με την αξιοποίηση αυτής της προσέγγισης για την ανάπτυξη των φαρμάκων, επιτυγχάνεται η εξοικονόμηση χρημάτων, η επιτάχυνση της διαδικασίας παραγωγής και η βελτίωση της ποιότητας των φαρμακευτικώνσκευασμάτων. Αυτό πραγματώνεται, με τη συμπερίληψη όλων των συντελεστών (άνθρωποι, ιατρικές συσκευές, συστήματα) σε ενοποιημένες διαδικασίες, ώστε να παρέχονται ενημερώσεις σε πραγματικό χρόνο, τόσο στις ενδιαφερόμενες οντότητες, όσο και στους κανονιστικούς ρυθμιστικούς φορείς. Στο ιατρικό MCPS, χρησιμοποιείται η τεχνολογία κυβερνοφυσικών συστημάτων (CPS) για την εξασφάλιση της μέγιστης ιατροφαρμακευτικής περίθαλψης για τους ασθενείς, ιδίως σε δύσκολες ιατρικές καταστάσεις, όπως για παράδειγμα οι κλινικές δοκιμές.^(16, 17)

Μέχρι πρότινος, η διαδικασία προμήθειας των φαρμακευτικώνσκευασμάτων περιοριζόταν στην επιλογή και την αξιολόγηση των προμηθευτών για την απόκτηση των πρώτων υλών και του εξοπλισμού που απαιτούνταν για την παρασκευή των φαρμάκων, με κυρίαρχους παράγοντες το κόστος, την ποιότητα και την εξυπηρέτηση. Την ημέραν, όμως, για τη σύγχρονη προμήθεια απαιτούνται πιο προηγμένες υπηρεσίες όπως μεταξύ άλλων, οι συνεργατικές τεχνολογίες, οι ψηφιακές σχεσιακές συμβάσεις και τα ηλεκτρονικά συστήματα παραγγελιών και προμηθειών. Εξαιτίας των νέων, διαρκώς μεταβαλλόμενων απαιτήσεων και δεδομένων στη φαρμακευτική βιομηχανία, η συμβολή των αναδυόμενων τεχνολογιών (ET) θεωρείται καθοριστική σε μια σειρά από ενέργειες όπως μεταξύ άλλων, η εξασφάλιση της διαφάνειας των προμηθευτών, η ενίσχυση της λήψης αποφάσεων με τη χρήση της τεχνητής νοημοσύνης, οι «έξυπνες» προμήθειες, τα εικονικά δωμάτια προμηθευτών, και η διαχείριση ρίσκων και κινδύνων στο υπολογιστικό νέφος.⁽¹⁶⁾

Στην περίπτωση της διαδικασίας παραγωγής, αντί της παραδοσιακής μεθόδου παραγωγής παρτίδων, με τις αναδυόμενες τεχνολογίες παρέχεται η στροφή προς την «έξυπνη», αδιάκοπη παραγωγή, διασυνδέοντας όλες τις λειτουργικές διαδικασίες σε μια κεντρική πλατφόρμα. Με αυτή την ενοποίηση διευκολύνεται κυρίως η αυτοματοποίηση, η συλλογή δεδομένων σε πραγματικό χρόνο, η αξιοποίηση εργαλείων του διαδικτύου των πραγμάτων (IoT) και η ανάλυση δεδομένων με απώτερο σκοπό την αύξηση της αποδοτικότητας της παραγωγής, την ενίσχυση της ποιότητας και την βελτιστοποίηση της κατανομής των διαθέσιμων πόρων. Έπειτα, με τη χρήση του υπολογιστικού νέφους και της τεχνολογίας Blockchain, πραγματοποιείται ο ασφαλής διαμοιρασμός των ιατρικών δεδομένων, ενώ οι αλγόριθμοι της τεχνητής νοημοσύνης και της μηχανικής μάθησης αξιοποιούνται για την πρόληψη των μηχανικών αστοχιών, ενισχύοντας έτσι τις καλές πρακτικές συντήρησης. Τέλος, με τα κυβερνοφυσικά συστήματα και την τρισδιάστατη (3D) εκτύπωση, υποστηρίζεται στην πραγματικότητα η λιτή παραγωγή, η αποδοτική διαχείριση των αποθεμάτων αλλά και η ταχεία παραγωγή εξατομικευμένων φαρμάκων, ειδικά σε καταστάσεις έκτακτης ανάγκης. Συνεπώς, αποδεικνύεται καταλυτική η συμβολή των αναδυόμενων τεχνολογιών και στις διαδικασίες παραγωγής, καθώς με τη χρήση τους βελτιώνεται σημαντικά τόσο η παραγωγικότητα, όσο και η αποτελεσματικότητα των διαδικασιών παραγωγής.⁽¹⁶⁾

Σχετικά με τη διαδικασία της διανομής, χρησιμοποιώντας τις νεοεισαχθείσες τεχνολογίες όπως τα RFIDs, τα Barcodes και το Blockchain, παρέχεται συντονισμός, διαφάνεια και ιχνηλασιμότητα στη διανομή, ώστε να διατηρούνται τα ακριβή στοιχεία των φαρμακευτικών σκευασμάτων και οι πληροφορίες να διαμοιράζονται με ασφάλεια στη φαρμακευτική αλυσίδα εφοδιασμού μέσω του υπολογιστικού νέφους. Στη συνέχεια, με τη χρήση εργαλείων του IoT, πραγματώνεται η παρακολούθηση των αποστολών σε πραγματικό χρόνο, αποτρέπεται η κλοπή του εμπορεύματος και εξασφαλίζονται οι έγκαιρες παραδόσεις. Παράλληλα, εξαιτίας της τεχνητής νοημοσύνης και της μηχανικής μάθησης, οι διαδρομές των διανομών ελαχιστοποιούνται για τη μέγιστη βελτιστοποίηση του κόστους. Ύστερα, με τα κυβερνοφυσικά συστήματα, παρακολουθούνται διαρκώς οι συνθήκες για την αποφυγή αλλοιώσεων των φαρμακευτικών σκευασμάτων, ενώ με την «έξυπνη» επιμελητεία, μειώνεται το κόστος και ο χρόνος παράδοσης. Συνολικά, και αυτές οι αναδυόμενες τεχνολογίες προσμετρώνται θετικά, ενισχύοντας σημαντικά τη ζωτική ροή της διαδικασίας της διανομής.⁽¹⁶⁾

Τα τελευταία χρόνια, η φαρμακοβιομηχανία στηρίζεται ολοένα και περισσότερο σε προηγμένες στρατηγικές ενοποίησης των ιατρικών δεδομένων με σκοπό τη βελτίωση της έρευνας και της ανάπτυξης. Με την αξιοποίηση δομημένων και μη, δεδομένων μέσω της τεχνητής νοημοσύνης (AI), της μηχανικής εκμάθησης (ML) και της επεξεργασίας φυσικής γλώσσας (NPL), λαμβάνονται από τα συστήματα υποστήριξης (DSS), οι εμπειριστατωμένες αποφάσεις για την αποδοτικότερη ανάπτυξη των φαρμάκων. Έτσι, η ενοποίηση αυτών των ιατρικών δεδομένων από διάφορες πηγές, πρόκειται για ζωτικής σημασίας διαδικασία, απαραίτητη για την καλύτερη κατανόηση των μηχανισμών της

νόσου, τη βελτιστοποίηση των κλινικών δοκιμών και τη διασφάλιση της ασφάλειας και της αποτελεσματικότητας των φαρμακευτικών σκευασμάτων. Εντούτοις, στη διαδικασία της ενοποίησης τίθενται προκλήσεις που σχετίζονται με τον ασφαλή διαμοιρασμό των ευαίσθητων ιατρικών δεδομένων, την κανονιστική συμμόρφωση των φαρμάκων και τη διασύνδεση όλων των ενδιαφερόμενων φορέων σε ένα κοινό διαπλατφορμικό σύστημα. Για την αντιμετώπιση των προαναφερθέντων προκλήσεων, υιοθετούνται τα εργαλεία του υπολογιστικού νέφους και οι τεχνικές ορισμένων ψηφιακών τεχνολογιών, επιτρέποντας έτσι την ταχύτερη και την οικονομικότερη ανάπτυξη των φαρμάκων. Κατά συνέπεια, με τη χρήση των τεχνολογιών αυτών, επιταχύνεται η καινοτομία στην ανάπτυξη φαρμάκων, ενισχύεται η συνεργασία μεταξύ των ενδιαφερόμενων αρμόδιων φορέων, αποτρέπονται οι επικαλύψεις των ιατρικών δεδομένων και τελικώς βελτιώνονται τα αποτελέσματα των ασθενών.⁽¹⁷⁾

Με τα εργαλεία και τις τεχνικές του υπολογιστικού νέφους, επαναπροσδιορίζονται αρκετές από τις παραδοσιακές διαδικασίες της φαρμακοβιομηχανίας, ενισχύοντας έτσι τις δυνατότητες της παραγωγής, της αποθήκευσης και του διαμοιρασμού των ιατρικών δεδομένων. Μέσω των εργαλείων αυτών, υποστηρίζονται η διαδικασία της τηλεϊατρικής, οι εφαρμογές υγείας για κινητές συσκευές αλλά και η απομακρυσμένη παρακολούθηση, παρέχοντας έτσι υπηρεσίες υγειονομικής περίθαλψης υψηλότερης ποιότητας. Ακόμη, διευκολύνεται η πρόσβαση και η ενσωμάτωση των ιατρικών δεδομένων σε πραγματικό χρόνο, τα οποία κρίνονται απαραίτητα για την εξατομικευμένη ιατρική, την κλινική έρευνα και τις εφαρμογές που βασίζονται στην τεχνητή νοημοσύνη (AI). Επομένως, μειώνεται η εξάρτηση από την εσωτερική υποδομή ενός οργανισμού και βελτιώνεται η αποτελεσματικότητα της ροής εργασιών του οργανισμού αυτού. Ανεξάρτητα με τους προβληματισμούς για τα ζητήματα της ασφάλειας και της διαφύλαξης του απορρήτου των ιατρικών δεδομένων, με το υπολογιστικό νέφος, προσφέρονται στην ιατρική σημαντικά οφέλη για την ανάπτυξη των φαρμακευτικών σκευασμάτων, από τη διαχείριση των προκλινικών δεδομένων έως και τις κλινικές δοκιμές, έτσι ώστε να επιτυγχάνεται η ορθολογικότερη αξιοποίηση των δεδομένων αλλά και να προάγεται, τόσο η διαφάνεια, όσο και η επικοινωνία στις απομακρυσμένες ροές εργασιών των οργανισμών.⁽¹⁷⁾

Με την τεχνητή νοημοσύνη (AI) μετεξελίσσονται γρήγορα οι φαρμακευτικές επιστήμες και η υγειονομική περίθαλψη αξιοποιώντας πολύπλοκους αλγορίθμους και λογισμικό για την προσομοίωση της ανθρώπινης νοημοσύνης. Έτσι, υποβοηθούνται οι διαγνώσεις, οι θεραπείες και οι προβλέψεις ιατρικών καταστάσεων καθώς ενισχύεται η διαδικασία ανάπτυξης νέων φαρμάκων με τον εντοπισμό νέων μορίων, την ανάλυση πολύπλοκων δεδομένων και τη βελτίωση της λειτουργικής αποτελεσματικότητας. Ακόμη, η μηχανική μάθηση (ML), ένα υποσύνολο της τεχνητής νοημοσύνης, χρησιμοποιείται για την αναγνώριση προτύπων με στόχο την πρόβλεψη των ιδιοτήτων των νέων ενώσεων, μιας ζωτικής σημασίας διαδικασία για την ανακάλυψη νέων φαρμάκων. Για το λόγο αυτό, χρησιμοποιούνται τεχνικές όπως είναι οι ποσοτικές σχέσεις δομής - δραστηριότητας

(QSAR) στη μοντελοποίηση των ιδιοτήτων των φαρμάκων με στόχο τον προσδιορισμό του μεταβολισμού και της τοξικότητάς τους. Κατά συνέπεια, μέσω του συνδυασμού της τεχνητής νοημοσύνης με το υπολογιστικό νέφος, επιταχύνεται η διαδικασία ανάπτυξης νέων φαρμάκων, μειώνεται το κόστος παραγωγής και βελτιώνεται η αποτελεσματικότητα των φαρμάκων καθιστώντας τις αναδυόμενες αυτές τεχνολογίες αναπόσπαστο μέρος της καινοτομίας στη φαρμακοβιομηχανία του μέλλοντος.⁽¹⁷⁾

2.2. Η τεχνολογία «Blockchain»

Με τον όρο «Blockchain», περιγράφεται γενικά, ένα υπόδειγμα διαχείρισης δεδομένων σε ένα αποκεντρωμένο δίκτυο που διακρίνεται από συγκεκριμένες ιδιότητες. Ελλείπει της καθορισμένης προτυποποίησης και της επίσημης προδιαγραφής του όρου, χρησιμοποιούνται διαφορετικές προσεγγίσεις αυτού του γενικού ορισμού στις αναφορές της βιβλιογραφίας και στα υπάρχοντα δημοφιλή συστήματα που βασίζονται σε αυτή την τεχνολογία.⁽¹⁸⁾ Έτσι, σύμφωνα με τις ιδιότητές της, η τεχνολογία Blockchain δύναται να ορίζεται τεχνικά ως «μια κατανεμημένη βάση δεδομένων από συναλλαγές στην οποία διαφορετικοί υπολογιστές, οι λεγόμενοι κόμβοι, συνεργάζονται μεταξύ τους ως ενιαίο σύστημα για την αποθήκευση ακολουθιών bits, κρυπτογραφημένα ως ενιαία μονάδα ή μπλοκ και ύστερα διασυνδέονται μεταξύ τους σχηματίζοντας μια αλυσίδα».⁽¹⁹⁾

Το υπόδειγμα της τεχνολογίας «Blockchain» αναδεικνύεται για πρώτη φορά στο προσκήνιο των τεχνολογικών εξελίξεων με τον ερχομό του πρώτου κρυπτονομίσματος, το λεγόμενο Bitcoin. Όμως, οι τεχνολογικές δυνατότητες του υποδείγματος εκτείνονται και σε πολλές διαφορετικές χρήσεις, όπως είναι μεταξύ άλλων οι τραπεζικές υπηρεσίες, τα εναέρια δίκτυα, η ηλεκτρονική ψηφοφορία, οι υπηρεσίες υγειονομικής περίθαλψης, η διαχείριση της εφοδιαστικής αλυσίδας (SCM) αλλά και στις εφαρμογές του διαδικτύου των πραγμάτων (IoT).⁽²⁰⁾ Συγκεκριμένα, μέσω της τεχνολογίας Blockchain παρέχονται μεταξύ άλλων οι δυνατότητες της αποκέντρωσης, της αυτονομίας, της ακεραιότητας, της επαλήθευσης, της αμεταβλητότητας, της ανωνυμίας, της διαφάνειας και της παροχής ελέγχου (auditability). Τοιουτοτρόπως, καθίσταται η τεχνολογία Blockchain, εξαιρετικά αποτελεσματική στην ιχνηλάτηση και την παρακολούθηση των περιουσιακών στοιχείων. Στα οφέλη της παρακολούθησης και της ιχνηλάτησης συμπεριλαμβάνεται ο ασφαλής διαμοιρασμός των πληροφοριών, η συλλογή των δεδομένων σε πραγματικό χρόνο, η βελτιωμένη διαφάνεια αλλά και η ορατότητα στις διενέργειες των εφαρμογών αυτής της τεχνολογίας. Κατά συνέπεια, σύμφωνα με αρκετούς ερευνητές, όλα τα προαναφερθέντα χαρακτηριστικά είναι καλά ευθυγραμμισμένα με τις αρχές της διαχείρισης της αλυσίδας εφοδιασμού (SCM), όπως τη διασφάλιση της ποιότητας των προϊόντων, την αξιοπιστία και την οικονομική απόδοση.⁽²¹⁾

Με τις αναθεωρήσεις του υποδείγματος της τεχνολογίας Blockchain στα χρόνια, αναπτύσσονται τρεις βασικοί τύποι δικτύων Blockchain, οι δημόσιες (public) αλυσίδες, οι

αλυσίδες κοινοπραξιών (consortium) και οι ιδιωτικές (private) αλυσίδες. Σε ένα δημόσιο δίκτυο, οποιαδήποτε οντότητα ή οργανισμός εντάσσεται στο δίκτυο και γίνεται χρήστης, γνωστός και ως κόμβος. Οι κόμβοι με σημαντικούς υπολογιστικούς πόρους, γνωστοί και ως miners, είναι υπεύθυνοι για την επικύρωση των συναλλαγών. Στη συνέχεια, σε ένα δίκτυο κοινοπραξιών, περιορίζεται ουσιαστικά η συμμετοχή των κόμβων σε επιλεγμένες οντότητες και συγκεκριμένους οργανισμούς μόνο μέσω πρόσκλησης. Ακολούθως, σε ένα ιδιωτικό δίκτυο, η πρόσβαση περιορίζεται αποκλειστικά στα εσωτερικά τμήματα ενός οργανισμού που διαχειρίζεται το ιδιωτικό δίκτυο αυτό.⁽²²⁾

	Δημόσιο	Κοινοπραξία	Ιδιωτικό
Προσβασιμότητα	— Οποιαδήποτε οντότητα έχει πρόσβαση και δικαιώματα ανάγνωσης/εγγραφής στο λογιστικό βιβλίο (ledger)	— Μόνο επιλεγμένες οντότητες έχουν πρόσβαση (μέσω πρόσκλησης) και δικαιώματα ανάγνωσης/εγγραφής στο λογιστικό βιβλίο (ledger)	— Μόνο εξουσιοδοτημένες οντότητες ενός οργανισμού έχουν πρόσβαση
Δικαιώματα	— Ανευ άδειας (Permissionless)	— Με άδεια (Permissioned)	— Με άδεια (Permissioned)
Κυριότητα	—	— Πολλαπλές οντότητες	— Μεμονωμένη οντότητα / οργανισμός
Συγκεντρωτισμός	— Πλήρως αποκεντρωμένο	— Μερικώς συγκεντρωτικό	— Συγκεντρωτικό
Πρωτόκολλο Συναίνεσης	— Proof of Work (PoW), Proof of Stake (PoS)	— Ψηφοφορία ή άλλοι πολυκομματικοί αλγόριθμοι συναίνεσης	— Ψηφοφορία ή άλλοι πολυκομματικοί αλγόριθμοι συναίνεσης
Κρυπτογράφηση	— Ζεύγος ιδιωτικού - δημόσιου κλειδιού	— Ζεύγος ιδιωτικού - δημόσιου κλειδιού	— Ζεύγος ιδιωτικού - δημόσιου κλειδιού
Μηχανισμός Εμπιστοσύνης	—	— Έμπιστες οντότητες	— Έμπιστες οντότητες
Αμεταβλητότητα	— Μόνο προσάρτηση, αμετάβλητο λογιστικό βιβλίο (ledger), ανθεκτικό στις παραβιάσεις	— Το λογιστικό βιβλίο (ledger) τροποποιείται από τους συντονιστές της κοινοπραξίας, ευάλωτο στις παραβιάσεις	— Το λογιστικό βιβλίο (ledger) τροποποιείται από τον οργανισμό, ευάλωτο στις παραβιάσεις

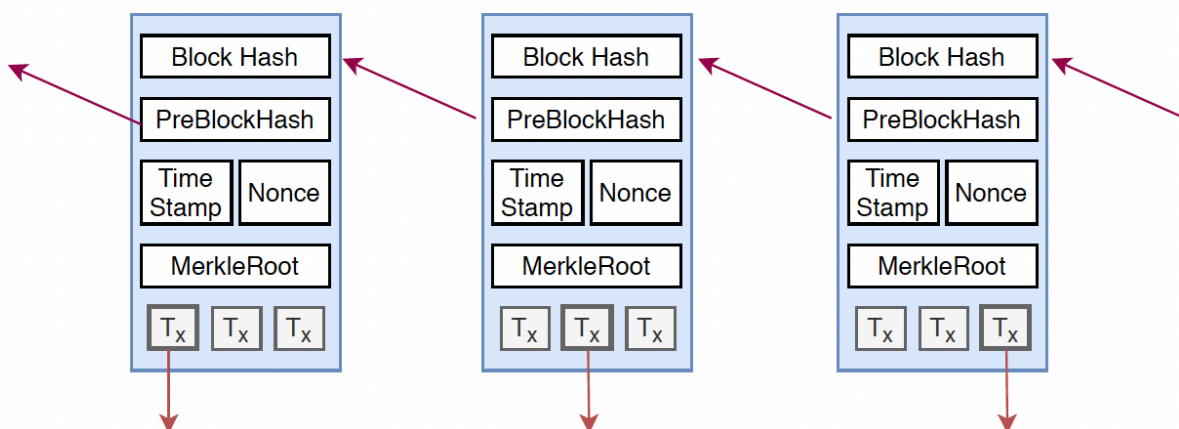
Πίνακας 2.1.: Σύγκριση των τριών βασικών τύπων δικτύου Blockchain, [Liu et al. \(2022\)](#)

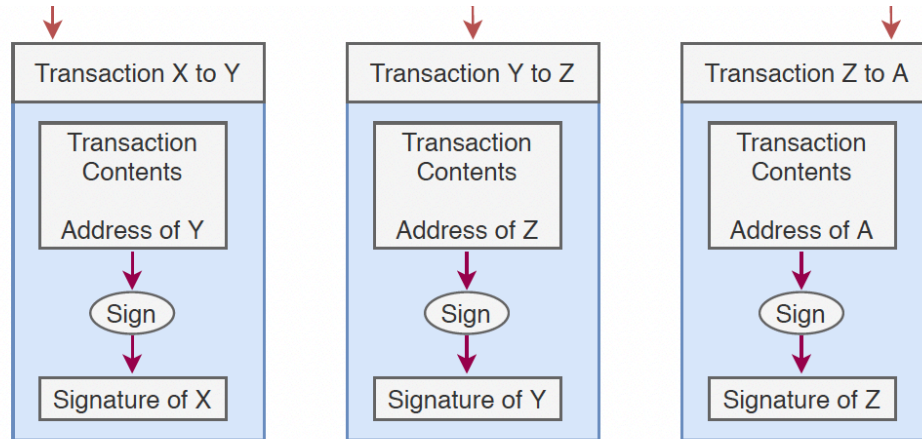
Τα τελευταία χρόνια, τα φλέγοντα ζητήματα στις αλυσίδες εφοδιασμού πρόκειται για την ιχνηλασιμότητα και την προέλευση των προϊόντων με σκοπό τη διασφάλιση της

βιωσιμότητας του κύκλου ζωής του προϊόντος. Παρόλο που η παρακολούθηση κατά τη μεταφορά των προϊόντων είναι εφικτή, συνήθως το επίπεδο αυτό λεπτομέρειας συχνά χάνεται ή καθίσταται ανέφικτο σε ολόκληρη την αλυσίδα εφοδιασμού, της οποίας ολοένα και αυξάνεται η πολυπλοκότητα εξαιτίας της παγκοσμιοποίησης. Επομένως, η ανάγκη αυτή για βελτιωμένη ιχνηλασιμότητα αποτέλεσε το έναυσμα για τον πειραματισμό των οργανισμών με την προσαρμογή και τη χρήση του δικτύου Blockchain στη διαχείριση της εφοδιαστικής τους αλυσίδας.⁽²¹⁾

2.3. Ιδιότητες της τεχνολογίας «Blockchain»

Το δίκτυο Blockchain αναφέρεται σε μια διαρκώς αναπτυσσόμενη αλυσίδα εγγραφών, που ονομάζονται μπλοκ και είναι διασυνδεδεμένα μεταξύ τους μέσω κρυπτογραφικών τεχνικών. Στο κάθε μπλοκ περιέχεται μια αναφορά του αμέσως προηγούμενου μπλοκ με τη χρήση ενός κρυπτογραφικού κατακερματισμού του προηγούμενου μπλοκ, γνωστό και ως γονικό (parent) μπλοκ του τρέχοντος μπλοκ. Το πρώτο μπλοκ στην αλυσίδα των μπλοκ ονομάζεται μπλοκ γένεσης (genesis) και ως πρωτεύων, στερείται γονικού μπλοκ. Η δομή ενός μπλοκ απαρτίζεται από την κεφαλίδα του μπλοκ και τον κορμό του μπλοκ. Στον κορμό του μπλοκ περιλαμβάνεται η λίστα με τις συναλλαγές, ενώ στην κεφαλίδα τα πεδία όπως ο αριθμός αναθεώρησης, ο κατακερματισμός του προηγούμενου μπλοκ, η ρίζα Merkle, η χρονοσφραγίδα και ο στόχος δυσκολίας. Με τη συμπλήρωση ενός μπλοκ από τα δεδομένα, καθίσταται το ίδιο ως αμετάβλητο, έτσι οποιαδήποτε αλλαγή θα απαιτούσε τη συμφωνία της πλειοψηφίας (51%) του δικτύου. Με αυτόν τον τρόπο διασφαλίζεται η ανθεκτικότητα των δικτύων Blockchain στις παραβιάσεις, παρέχοντας τη δυνατότητα επαλήθευσης του κάθε μπλοκ από όλους τους κόμβους του δικτύου.⁽²³⁾





Εικόνα 2.1.: Η δομή των μπλοκ με τις πληροφορίες της κεφαλίδας και του κορμού, *Singh et al. (2020)*

Ένα από τα βασικά χαρακτηριστικά που υποστηρίζεται από το δίκτυο Blockchain είναι τα «έξυπνα» συμβόλαια, πρόκειται για υπολογιστικά συναλλακτικά πρωτόκολλα που ενεργοποιούνται αυτόματα υπό προκαθορισμένους όρους. Με τα συμβόλαια αυτά, καθορίζονται οι κανόνες και οι υποχρεώσεις για τις διαδικασίες της συναλλαγής και της παράδοσης, οι οποίες χαρακτηρίζονται ως αυτοεκτελούμενες και αυτοεπιβαλλόμενες. Μέσω των μηχανισμών αυτών, ενισχύεται η καταλληλότητα των δικτύων Blockchain για τη διαχείριση της εφοδιαστικής αλυσίδας (SCM) καθώς έτσι ιχνηλατείται αποτελεσματικά κάθε πληροφορία στο κάθε στάδιο της αλυσίδας εφοδιασμού για την άμεση αναγνώριση και την επαλήθευση των προϊόντων.⁽²⁵⁾

Algorithm 1: Smart contract for monitoring drug supply chain

Step 1	“Input: Name, Country, City, Wallet, joinedDate, Caller, OwnerID
Step 2	Output: An event declaring that the drug has been supplied. An event declaring that the detail of the drug has been uploaded
Step 3	Data:
Step 4	Name: is the name of the drug and supplier
Step 5	Country: is the country of the drug and supplier
Step 6	City: is the city of the drug and supplier
Step 7	Wallet: is the price of the drug
Step 8	joinedDate: is the date of the drug
Step 9	ownerID: is the Ethereum address of the owner of the drug
Step 10	initializations
Step 11	if Caller == ownerID then
Step 12	Update Name, Update Country, Update City, Add Ethereum Address
Step 13	Emit an event declaring that the drug has been supplied Emit an event declaring that the detail of the drug has been uploaded
	else
Step 14	Revert the contract state and show an error.”

Εικόνα 2.2.: Έξυπνο συμβόλαιο με αυτοματισμούς για την παρακολούθηση ενός φαρμακευτικού σκευάσματος, *Aslam et al. (2023)*

2.4. Περιορισμοί της τεχνολογίας «Blockchain»

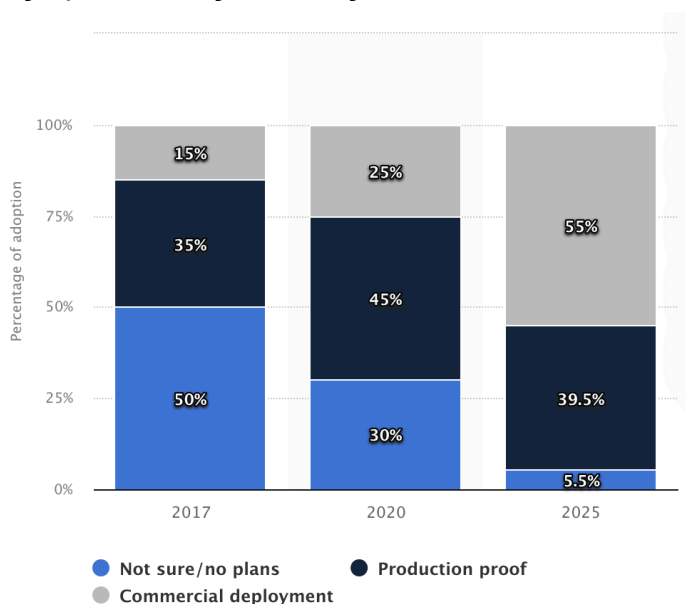
Στις πρακτικές εφαρμογές της τεχνολογίας Blockchain, όπως τα δημόσια δίκτυα, παρατηρούνται σημαντικές προκλήσεις επεκτασιμότητας. Έτσι, η επικύρωση των νέων συναλλαγών είναι υπεύθυνη για την πρόκληση συμφόρησης εξαιτίας του χρόνου και της υπολογιστικής ισχύος που απαιτούνται. Με αυτόν τον τρόπο, σε μια μεγάλη εφοδιαστική αλυσίδα φαρμάκων, δύναται να εμποδίζεται η αποδοτικότητά της, καθώς αυξάνονται οι συναλλαγές, προκαλώντας έτσι καθυστερήσεις στο δίκτυο αλλά και αυξημένο κόστος συντήρησης. Καθώς οι φαρμακευτικές αλυσίδες εφοδιασμού διαχειρίζονται ευαίσθητες πληροφορίες, όπως τα δεδομένα των ασθενών και η σύνθεση ιδιόκτητων σκευασμάτων, εγκυμονούνται για το απόρρητο των πληροφοριών αυτών. Στα δημόσια Blockchain, οι πληροφορίες των συναλλαγών του κάθε μπλοκ καθίστανται ορατές σε όλες τις οντότητες του δικτύου, με αποτέλεσμα να εκτίθενται ενδεχομένως εμπιστευτικές πληροφορίες. Πέρα από αυτό, στα ιδιωτικά δίκτυα απαιτούνται ισχυροί έλεγχοι για τη διασφάλιση του απορρήτου αλλά και κανονιστική συμμόρφωση με διεθνείς νομοθεσίες όπως ο Γενικός Κανονισμός για την Προστασία των Δεδομένων (GDPR). Αυτές οι διαδικασίες είναι πολύπλοκες εκ φύσεως τους αλλά και δαπανηρές για τους επιχειρηματικούς πόρους των οργανισμών υγείας. Αποδεικνύεται ότι η φαρμακοβιομηχανία υπόκειται σε αυστηρές ρυθμίσεις, με αυστηρές απαιτήσεις για την ακεραιότητα και την ιχνηλασιμότητα των φαρμακευτικών σκευασμάτων. Πράγματι, με τα δίκτυα Blockchain οι προαναφερθείσες απαιτήσεις ικανοποιούνται, ωστόσο οι κανονιστικές ρυθμίσεις στον υγειονομικό τομέα βρίσκονται σε αναντιστοιχία με το συγκεκριμένο υπόδειγμα. Έτσι, η κανονιστική αυτή ασυμφωνία καθιστά την τεχνολογία Blockchain αποτρεπτική για τους φαρμακευτικούς οργανισμούς εξαιτίας των πιθανών νομικών κινδύνων.^(20, 27)

Αρκετοί φαρμακευτικοί οργανισμοί εξαρτώνται από καθιερωμένα πληροφοριακά συστήματα, τα οποία όμως είναι παρωχημένα. Έτσι, η υπάρχουσα υποδομή καθίσταται τεχνολογικά ασύμβατη με το δίκτυο Blockchain, καθώς για οποιαδήποτε μετάβαση θα απαιτούνται σημαντικές αλλαγές στην υλικοτεχνική υποδομή, διαταράσσοντας την ίδια στιγμή και την υπάρχουσα λειτουργικότητα. Προς επίρρωση αυτού, με την έλλειψη των τυποποιημένων πρωτοκόλλων μετάβασης, καθίστανται αναγκαίες οι προσαρμοσμένες λύσεις τεχνολογικής υποστήριξης για την ενσωμάτωση της υποδομής Blockchain στην εφοδιαστική αλυσίδα του εκάστοτε οργανισμού, καθοριστικού δηλαδή παράγοντα για την αύξηση του κόστους αλλά και της πολυπλοκότητας της συγκεκριμένης επένδυσης. Με δεδομένα τα προαναφερθέντα, για την υλοποίηση ενός δικτύου Blockchain απαιτούνται σημαντικές επενδύσεις σε υλικό, λογισμικό και τεχνογνωσία. Ιδιαίτερα για τους μικρούς οργανισμούς στη φαρμακοβιομηχανία, το κόστος της αρχική εγκατάστασης όπως και η συνεχής συντήρηση του δικτύου Blockchain, δύναται να είναι αρκετά απαγορευτικό. Κατά συνέπεια, το κόστος αυτό αυξάνεται περισσότερο σε συνδυασμό με τις υψηλές υπολογιστικές και ενεργειακές απαιτήσεις του δικτύου Blockchain, περιορίζοντας έτσι

ουσιαστικά την ελκυστικότητά του μεταξύ των μικρότερων φαρμακευτικών οργανισμών εντός της αλυσίδας εφοδιασμού.⁽²⁷⁾

Στις φαρμακευτικές αλυσίδες εφοδιασμού (PSC), συμπεριλαμβάνονται πολλαπλοί παγκόσμιοι αρμόδιοι φορείς, ο καθένας χρησιμοποιώντας διαφορετική υλικοτεχνική υποδομή και προτυποποίηση διασύνδεσης. Η επίτευξη της διαλειτουργικότητας μεταξύ των υφιστάμενων υλικοτεχνικών υποδομών και του ανερχόμενου δικτύου Blockchain, πρόκειται για μια πραγματική πρόκληση, η οποία παρίσταται τα τελευταία χρόνια στη φαρμακοβιομηχανία. Δίχως τυποποιημένα πρωτόκολλα, η ενσωμάτωση των διαφόρων υλικοτεχνικών υποδομών σε μια ενοποιημένη λύση ενός δικτύου Blockchain καθίσταται δύσκολη με αποτέλεσμα να παρουσιάζεται τεχνολογικός κορεσμός στην ανάπτυξη της φαρμακευτικής αλυσίδας εφοδιασμού. Εξάλλου, η αποτελεσματικότητα της τεχνολογίας Blockchain, στηρίζεται στην ευρεία ενσωμάτωσή της από πολλούς οργανισμούς για την τελεσφόρηση των σημαντικότερων οφειλών της, δηλαδή της ιχνηλασιμότητας αλλά και του έγκαιρου εντοπισμού των πλαστών φαρμακευτικών σκευασμάτων. Συνεπώς, για τη βέλτιστη αξιοποίηση των χαρακτηριστικών ενός δικτύου Blockchain, κρίνεται απαραίτητη συμμετοχή της πλειοψηφίας των οργανισμών της φαρμακοβιομηχανίας.⁽²⁰⁾

Συμπερασματικά, αν και μέσω του δικτύου Blockchain προσφέρονται αναρίθμητα πιθανά οφέλη, παρουσιάζονται υπαρκτοί περιορισμοί για την ευρεία ενσωμάτωσή της από το σύνολο της φαρμακοβιομηχανίας. Τα ζητήματα όπως η νομική συμμόρφωση, το απόρρητο των πληροφοριών, η επεκτασιμότητα, το υψηλό κόστος, η ενοποίηση με την υπάρχουσα υλικοτεχνική υποδομή και η διαλειτουργικότητα, πρόκειται για ουσιαστικά εμπόδια προς επίλυση με σκοπό την καλύτερη απορρόφηση του δικτύου Blockchain από τις φαρμακευτικές εφοδιαστικές αλυσίδες.^(20, 27)



Εικόνα 2.3.: Προβλεπόμενη ενσωμάτωση της τεχνολογίας Blockchain σε εφαρμογές υγειονομικής περίθαλψης παγκοσμίως για τα έτη 2017, 2020 και 2025; [Statista \(2017\)](#)

Κεφάλαιο 3^ο - Υλοποίηση Συστήματος Ιχνηλασιμότητας

Σε αυτή την ενότητα παρουσιάζεται η υλοποίηση του Συστήματος Ιχνηλασιμότητας σε προγραμματιστικό επίπεδο σύμφωνα με τις τεχνολογίες που χρησιμοποιήθηκαν. Οι προκείμενες τεχνολογίες είναι τα κατασκευαστικά θεμέλια του προγράμματος και αποτέλεσαν τη βάση για την ανάπτυξη του γραφικού περιβάλλοντος. Στο περιβάλλον αυτό επιτυγχάνεται η ουσία του εγχειρήματος της ιχνηλασιμότητας των φαρμάκων με την προσπέλαση των καταγεγραμμένων συναλλαγών, την αναζήτηση των υπάρχοντων συναλλαγών και την προσθήκη νέων συναλλαγών σε μια διασυνδεδεμένη διεπαφή που πληροί τις ιδιότητες της κατανεμημένης τεχνολογίας Blockchain.

3.1. Προγραμματιστικές Επιλογές

3.1.1. Python (ΓΠ)

Η υλοποίηση του Συστήματος Ιχνηλασιμότητας πραγματοποιήθηκε με την υψηλού επιπέδου γλώσσα προγραμματισμού Python. Η επιλογή αυτή στηρίχθηκε τόσο στις ανάγκες του Συστήματος για αντικειμενοστρέφεια (δημιουργία Blockchain) και διαδικαστικών λειτουργιών (προσθήκη συναλλαγής κλπ.) όσο και στην υποστήριξη που παρέχει η ίδια η γλώσσα σε μια πληθώρα βιβλιοθηκών, οι οποίες χρησιμοποιήθηκαν για την συνεκτική κατασκευή της γραφικής διεπαφής χρήστη. Ο συνδυασμός των δυνατοτήτων αυτών επέτρεψαν τη δημιουργία μιας εφαρμογής με τη λιγότερη δυνατή επαναχρησιμοποίηση του πηγαίου κώδικα, με σκοπό την καλύτερη ευαναγνωσιμότητά του και την ευκολότερη επεκτασιμότητά του σε νέες συμπληρωματικές απαιτήσεις. Στα πλαίσια κατασκευής της εφαρμογής χρησιμοποιήθηκε η έκδοση της Python 3.9.6.

3.1.2. JSON (ΒΔ)

Μια από τις βασικές απαιτήσεις του Συστήματος Ιχνηλασιμότητας είναι η αποθήκευση των καταγεγραμμένων συναλλαγών (μητρώο δεδομένων) στο Blockchain μετά το πέρας της εκτέλεσης του προγράμματος. Η απαίτηση αυτή ικανοποιήθηκε με τη χρήση του ανοιχτού μορφότυπου JSON ως εναλλακτική μορφή NoSQL βάσης δεδομένων στην οποία χρησιμοποιείται απλό κείμενο και αποτελείται από ζεύγη χαρακτηριστικών-τιμών και δεδομένα τύπου συστοιχιών. Η επιλογή αυτή στηρίχθηκε στη διαδεδομένη μορφή του JSON που την καθιστά αρκετά συμβατή στην ασύγχρονη επικοινωνία περιηγητή-διακομιστή, εν προκειμένω στη λήψη-αποστολή δεδομένων (συναλλαγών) μεταξύ της εφαρμογής και της γραφικής διεπαφής χρήστη. Καθώς το μορφότυπο JSON δεν υποστηρίζει εγγενώς τον έλεγχο αμεταβλητότητας των δεδομένων, ο έλεγχος αυτός πραγματοποιείται στο προγραμματιστικό επίπεδο έτσι ώστε να

ικανοποιείται η ιδιότητα της «αμεταβλητότητας και μη-αναστρεψιμότητας» που απαιτεί η τεχνολογία Blockchain.

3.1.3. Flask (Web Framework)

Στα πλαίσια δημιουργίας μιας διασυνδεδεμένης γραφικής διεπαφής χρήστη για το Σύστημα Ιχνηλασιμότητας, απαιτείται η αμφίδρομη αλληλεπίδραση της διεπαφής με την εφαρμογή. Η απαίτηση αυτή ικανοποιήθηκε με τη χρήση της βιβλιοθήκης Flask, η οποία ουσιαστικά λειτουργεί ως ενδιάμεσος συνδετικός κρίκος για τις αιτήσεις HTTP που στέλνονται από τον ιστότοπο στην εφαρμογή και αντίστροφα. Όταν ο χρήστης αλληλεπιδρά με τον ιστότοπο, οι αιτήσεις προωθούνται στο πρόσθετο Flask, το οποίο επεξεργάζεται αυτές τις αιτήσεις και στέλνει τα απαραίτητα δεδομένα προς ή από την εφαρμογή. Η επιλογή αυτή στηρίχθηκε στον ευέλικτο τρόπο με τον οποίο διαχειρίζεται η βιβλιοθήκη τα αιτήματα επικοινωνίας, επιτρέποντας την εμφάνιση διαδραστικού περιεχομένου στη γραφική διεπαφή με βάση τις απαντήσεις που λαμβάνονται από την εφαρμογή.

3.1.4. Frontend (HTML, CSS & JS)

Η κατασκευή μιας γραφικής διεπαφής χρήστη για το Σύστημα Ιχνηλασιμότητας πρόκειται επί της ουσίας για τη δημιουργία μιας ιστοσελίδας με υποστήριξη δυναμικού περιεχομένου. Οι ετικέτες HTML χρησιμοποιούνται για τη δημιουργία της δομής της ιστοσελίδας (π.χ. Δημιουργία Φόρμας Συναλλαγής), ορίζοντας τα διάφορα στοιχεία και την οργάνωσή τους. Η CSS χρησιμοποιείται για να προσδιορίσει την εμφάνιση και τη διάταξη αυτών των στοιχείων, παρέχοντας στυλιστικές οδηγίες όπως χρώματα, γραμματοσειρές και διατάξεις. Τέλος, η JavaScript χρησιμοποιείται για τη διαχείριση της αλληλεπίδρασης με τον χρήστη και την εκτέλεση λειτουργιών βάσει των δράσεών του, όπως είναι ο έλεγχος εγκυρότητας των στοιχείων της Φόρμας Συναλλαγής. Με τη χρήση των τριών αυτών τεχνολογιών σε συνδυασμό και με τη βιβλιοθήκη Flask, δημιουργήθηκε μια πλήρης και δυναμική γραφική διεπαφή χρήστη για το Σύστημα της Ιχνηλασιμότητας Φαρμάκων.

3.2. Κώδικας Εφαρμογής

Η κατασκευή του Συστήματος Ιχνηλασιμότητας σύμφωνα με τις αναγκαίες απαιτήσεις του αρχικού σχεδιασμού διαφοροποιήθηκε σε δυο μέρη. Το πρώτο μέρος αφορά την υλοποίηση της εφαρμογής σε προγραμματιστικό επίπεδο, ενώ το δεύτερο μέρος αφορά τη δημιουργία της γραφικής διεπαφής χρήστη, αξιοποιώντας τις ήδη υπάρχουσες διαδικαστικές λειτουργίες της εφαρμογής που αναπτύχθηκε. Ουσιαστικά, αυτό σημαίνει πως οι παρεχόμενες λειτουργίες της γραφικής διεπαφής χρήστη είναι

εφικτές και σε επίπεδο εφαρμογής-τερματικού, προάγοντας την σημασία της διαμερισματοποίησης του πηγαίου κώδικα και ενισχύοντας τοιούτοτρόπως τη διαλειτουργικότητα του Συστήματος Ιχνηλασιμότητας.

3.2.1. Data.py

Στο αρχείο data.py ορίζονται οι κλάσεις που σχετίζονται με τα διακριτά στάδια της εφοδιαστικής αλυσίδας των φαρμάκων. Κάθε κλάση έχει ιδιότητες που μοντελοποιούν σχετικές πληροφορίες για το εκάστοτε στάδιο που περιγράφουν και όλες οι κλάσεις κληρονομούν τις ιδιότητες και τις μεθόδους της γονικής κλάσης 'Data'. Η γονική κλάση 'Data' παρέχει μεθόδους για τον υπολογισμό του κατακερματισμού (hashing) των αντικειμένων, τη σειριοποίηση των αντικειμένων στο μορφότυπο JSON και την αποσειριοποίησή τους στην αρχική τους μορφή.

3.2.1.1. Class Data

```
class Data:
    def __init__(self, product_name="", batch_number=""):
        self.product_name = product_name
        self.batch_number = batch_number
        self.tx_hash = self.calculate_hash()

    def calculate_hash(self):
        fields_dict = self.serialize()
        if fields_dict.__contains__("tx_hash"):
            fields_dict.pop("tx_hash")
        data_string = "@" + str(fields_dict) + "@"
        return hashlib.sha256(data_string.encode()).hexdigest()

    def print(self):
        print("\tTransaction Hash:", self.tx_hash)

    def serialize(self):
        data = {"type": self.__class__.__name__}
        attributes = vars(self)
        for var_name in attributes:
            data[var_name] = attributes[var_name]
        return data

    @classmethod
    def deserialize(cls, data):
        args = [data[var_name] for var_name in data.keys()]
        return cls(*args)
```

```

class Product(Data):
    def __init__(self, initiator, product_name, batch_number,
universal_product_code, packaging_type, product_strength, expiration_date):
        self.initiator = initiator
        self.product_name = product_name
        self.batch_number = batch_number
        self.universal_product_code = universal_product_code
        self.packaging_type = packaging_type
        self.product_strength = product_strength
        self.expiration_date = expiration_date
        super().__init__(product_name, batch_number)

```

The other classes included in the file are omitted from this sample!

- Η μέθοδος σειριοποίησης (**def serialize**) είναι υπεύθυνη για την μετατροπή των παραγόμενων αντικειμένων από τις κλάσεις σε μορφή κειμένου από ζεύγη χαρακτηριστικών-τιμών. Η κλήση της μεθόδου χρησιμοποιείται κυρίως για την αποθήκευση των δεδομένων στο μορφότυπο JSON. Τέλος, η μέθοδος είναι ευέλικτα κατασκευασμένη ώστε να λειτουργεί σε μελλοντικές τροποποιήσεις των κλάσεων ή και στην προσθήκη νέων.
- Η μέθοδος αποσειριοποίησης (**def deserialize**) είναι υπεύθυνη για την μετατροπή των κειμένων από ζεύγη χαρακτηριστικών-τιμών στη μορφή αντικειμένων κλάσεων. Η μέθοδος αυτή χρησιμοποιείται στην ανακατασκευή της δομής του Blockchain από τις πληροφορίες που είναι αποθηκευμένες στη βάση δεδομένων. Η κλήση της πραγματοποιείται κατά την εκκίνηση της εφαρμογής, όπου και επιχειρείται η φόρτωση των πληροφοριών από τη βάση δεδομένων.
- Οι μέθοδοι σειριοποίησης και αποσειριοποίησης (**def serialize**, **def deserialize**) είναι αλληλένδετοι καθώς μέσω αυτών εξασφαλίζεται η ιδιότητα της τεχνολογίας Blockchain για τη «συνεχιζόμενη ύπαρξη των δεδομένων» μετά το πέρας της εκτέλεσης της εφαρμογής.
- Η μέθοδος κατακερματισμού (**def calculate_hash**) είναι υπεύθυνη για την παραγωγή ενός μοναδικού αναγνωριστικού για τις σειριοποιημένες πληροφορίες των αντικειμένων. Με τη μέθοδο αυτή εξασφαλίζεται η ακεραιότητα των δεδομένων σύμφωνα με τις απαιτήσεις της τεχνολογίας Blockchain και αποφεύγεται η χρήση διπλότυπων πληροφοριών.

3.2.2. Blockchain.py

Στο αρχείο blockchain.py ορίζονται οι απαραίτητες κλάσεις που σχετίζονται με τη δομή της τεχνολογίας Blockchain. Πιο συγκεκριμένα, η κλάση Block ορίζει τα περιεχόμενα του κάθε μπλοκ, τα οποία εξαρτώνται από το πρωτόκολλο υλοποίησης του Συστήματος Ιχνηλασιμότητας. Έπειτα, η κλάση Blockchain, καθορίζει τις ιδιότητες και τις μεθόδους μιας προσομοίωσης της τεχνολογίας Blockchain, η οποία είναι προσαρμοσμένη για την ιχνηλασιμότητα φαρμάκων.

3.2.2.1. Class Block

```
class Block:
    def __init__(self, index, previous_hash, timestamp, transactions_list:
List[Data]):
        self.index = index
        self.previous_hash = previous_hash
        self.timestamp = timestamp
        self.transactions_list = transactions_list
        self.hash = self.calculate_hash()

    def calculate_hash(self):
        data_string = str(self.index) + self.previous_hash
        for tx in self.transactions_list:
            data_string += "#"
            data_string += tx.tx_hash
        return hashlib.sha256(data_string.encode()).hexdigest()

    def print(self):
        print("Block", self.index)
        print("Timestamp:", self.timestamp)
        print("Previous Hash:", self.previous_hash)
        print("Transactions: {")
        for tx in self.transactions_list:
            tx.print()
            print("\t<----->")
        print("}")
        print("Hash:", self.hash)

    def convert_to_json(self):
        block_data = {
            "index": self.index,
```

```

        "previous_hash": self.previous_hash,
        "timestamp": self.timestamp,
        "transactions_list": [
            tx.serialize() for tx in self.transactions_list
        ],
        "hash": self.hash
    }
    return block_data

```

- Η μέθοδος αρχικοποίησης (`def __init__`) είναι υπεύθυνη για τον προσδιορισμό των βασικών ιδιοτήτων του κάθε μπλοκ. Δηλαδή τον δείκτη (index) του μπλοκ, το αναγνωριστικό hash του προηγούμενου μπλοκ (previous hash) στο τρέχον, τη χρονική δημιουργία (timestamp) του μπλοκ, τη λίστα (transactions_list) με τις πληροφορίες του μπλοκ και τέλος τη μέθοδο κατακερματισμού (calculate_hash) για την παραγωγή ενός μοναδικού αναγνωριστικού για το τρέχον μπλοκ.
- Η μέθοδος κατακερματισμού (`def calculate_hash`) είναι υπεύθυνη για την παραγωγή ενός μοναδικού αναγνωριστικού για το τρέχον μπλοκ. Ο συνδυασμός των συμβολοσειρών για την κατασκευή του αναγνωριστικού πρόκειται για προγραμματιστικό μυστικό. Με αυτόν τον τρόπο, η κακόβουλη παραποίηση των πληροφοριών του μπλοκ καθίσταται ανέφικτη δίχως τη γνώση του μοτίβου για την παραγωγή του νέου σωστού αναγνωριστικού για το αλλοιωμένο μπλοκ.
- Η μέθοδος μετατροπής στο μορφότυπο JSON (`def convert_to_json`) είναι υπεύθυνη για τη μετατροπή των αντικειμένων της κλάσης Block σε μορφή κειμένου που αποτελείται από ζεύγη χαρακτηριστικών-τιμών ώστε να αποθηκεύονται οι πληροφορίες του μπλοκ μετά το πέρας της εκτέλεσης της εφαρμογής. Τα σταθερά πεδία (π.χ. Δείκτης, χρονοσφραγίδα, κτλ.) ενός μπλοκ μετατρέπονται αυτούσια σε ζεύγη χαρακτηριστικών-τιμών, ενώ για τα μεταβαλλόμενα πεδία όπως είναι η λίστα των συναλλαγών (transactions_list) γίνεται χρήση της μεθόδου σειριοποίησης (`def serialize`) του κάθε αντικειμένου συναλλαγής από τη γονική κλάση 'Data' του αρχείου data.py.

3.2.2.2. Class Blockchain (1/2)

```

class Blockchain:
    TRANSACTIONS_PER_BLOCK = 5

    def __init__(self):
        self.chain = [self.create_genesis_block()]
    def create_genesis_block(self):
        return Block(0, "", str(datetime.datetime.now().strftime("%d-%m-%Y
%H:%M:%S:%f"))), [])

```

```

def add_block(self, transactions_list):
    previous_block = self.chain[-1]
    index = len(self.chain)
    timestamp = str(datetime.datetime.now().strftime("%d-%m-%Y
%H:%M:%S:%f"))
    new_block = Block(index, previous_block.hash, timestamp,
transactions_list)
    self.chain.append(new_block)
    return new_block

def add_transaction(self, transaction_object: Data):
    current_block = self.chain[-1]
    if len(self.chain) == 1:
        self.add_block([transaction_object])
    elif len(current_block.transactions_list) <
Blockchain.TRANSACTIONS_PER_BLOCK :
        current_block.transactions_list.append(transaction_object)
        self.timestamp = str(datetime.datetime.now().strftime("%d-%m-%Y
%H:%M:%S:%f"))
        current_block.hash = current_block.calculate_hash()
    else:
        self.add_block([transaction_object])

def print_blockchain(self):
    for block in self.chain:
        block.print()

def validate_chain_continuity(self):
    for b in range(len(self.chain) - 1):
        if self.chain[b].hash != self.chain[b+1].previous_hash:
            raise Exception("Chain is broken. Block {} HASH = {} but
Block {} PREVIOUS_HASH = {}."
                            .format(self.chain[b].index,
self.chain[b].hash, self.chain[b+1].index, self.chain[b+1].hash))
    print("\nBlocks are valid; Chain is immutable.")

```

- Ο κατασκευαστής `def __init__()` είναι υπεύθυνος για την αρχικοποίηση του Blockchain, δηλαδή μιας νέας αλυσίδας από μπλοκ, δημιουργώντας το πρώτο μπλοκ στην αλυσίδα με την κλήση της μεθόδου `def create_genesis_block()`.
- Η μέθοδος `def create_genesis_block()` είναι υπεύθυνη για τη δημιουργία του πρώτου μπλοκ στην αλυσίδα. Πρόκειται για ένα ειδικό μπλοκ χωρίς προηγούμενο

κατακερματισμό ως το πρώτο μπλοκ της αλυσίδας. Περιλαμβάνει τη χρονική δημιουργία του (timestamp) στην αλυσίδα και μια κενή λίστα (transactions_list) από συναλλαγές.

- Η μέθοδος `def add_block()` είναι υπεύθυνη για την προσθήκη ενός νέου μπλοκ στην αλυσίδα. Η διαδικασία αυτή πραγματοποιείται με τον προσδιορισμό του τρέχοντος δείκτη (index), την ανάκτηση του προηγούμενου κατακερματισμού (previous_hash) από την αλυσίδα, την καταγραφή της χρονικής δημιουργίας (timestamp) για το νέο μπλοκ και την προσάρτηση της λίστας των τρεχόντων συναλλαγών (transactions_list) σε αυτό.
- Η μέθοδος `def add_transaction()` είναι υπεύθυνη για την προσθήκη μιας νέας συναλλαγής στο τρέχον μπλοκ της αλυσίδας. Αυτή η διαδικασία πραγματοποιείται με την εισαγωγή και την προσάρτηση ενός αντικειμένου συναλλαγής στη λίστα συναλλαγών (transactions_list) του τρέχοντος μπλοκ. Εάν το τρέχον μπλοκ είναι πλήρες, τότε δημιουργείται ένα νέο μπλοκ το οποίο και προστίθεται στην αλυσίδα των μπλοκ με αυτή τη μοναδική νέα συναλλαγή. Η συνθήκη πληρότητας ενός μπλοκ ελέγχεται από τη μεταβλητή 'TRANSACTIONS_PER_BLOCK' με το μέγιστο αριθμό συναλλαγών να ανέρχεται στις 5 συναλλαγές ανά μπλοκ. Η επιλογή αυτή αποτελεί μια παραδοχή με σκοπό την ανάδειξη της μοναδικότητας του κάθε μπλοκ στο Σύστημα Ιχνηλασιμότητας. Σε πραγματικές περιπτώσεις, η σταθερή απόδοση συναλλαγών ανά μπλοκ πρόκειται για σχεδιαστική επιλογή της εκάστοτε υλοποίησης της τεχνολογίας Blockchain με στόχο την εξασφάλιση συνεπών και προβλέψιμων χρόνων επεξεργασίας για τις συναλλαγές στο δίκτυό της.
- Η μέθοδος `def print_blockchain()` είναι υπεύθυνη για την εκτύπωση των πληροφοριών του κάθε μπλοκ στην αλυσίδα χρησιμοποιώντας τη μέθοδο `def print()` της κλάσης Block.
- Η μέθοδος `def validate_chain_continuity()` είναι υπεύθυνη για την επικύρωση της «αμεταβλητότητας» της τεχνολογίας Blockchain. Το τρέχον μπλοκ (b+1) της αλυσίδας συμπεριλαμβάνει στις πληροφορίες του το μοναδικό αναγνωριστικό του αμέσως προηγούμενου μπλοκ (b), συνεπώς με τον έλεγχο:

```
if self.chain[b].hash != self.chain[b+1].previous_hash
```

Εξασφαλίζεται η έγκυρη αλληλουχία της αλυσίδας, καθώς η παραποίηση ενός μπλοκ θα προκαλούσε την αλλαγή του μοναδικού του αναγνωριστικού, προκαλώντας έτσι την αναντιστοιχία του με το αποθηκευμένο αναγνωριστικό του επόμενου μπλοκ.

3.2.2.3. Class Blockchain (2/2)

```
class Blockchain:
    @classmethod
    def load_from_file(cls):
        print("IMPORTED DATABASE:", BLOCKCHAINDB_FILENAME, "\n")
        blockchain = cls()
        blockchain.chain = [] # Reset Chain
        index = 0
        with open(BLOCKCHAINDB_FILENAME, 'r') as file:
            blockchain_data = json.load(file)
            for block_data in blockchain_data:
                transactions_list = []
                for tx_data in block_data["transactions_list"]:
                    tx_class = globals()[tx_data["type"]]
                    tx_data.pop("type")
                    tx_data.pop("tx_hash")
                    tx = tx_class.deserialize(tx_data)
                    transactions_list.append(tx)
                block = Block(
                    index=block_data["index"],
                    previous_hash=block_data["previous_hash"],
                    timestamp=block_data["timestamp"],
                    transactions_list=transactions_list
                )
                saved_hash = block_data["hash"]
                if (block.hash != saved_hash):
                    raise Exception("Invalid HASH detected in Block [{}]:
{}.\n It should be: {}".format(block.index, saved_hash, block.hash))
                if (block.index != index):
                    raise Exception("Block [{}] INDEX is invalid. It should
be [{}].".format(block.index, index))
                print("Block [{}] is valid".format(block.index))
                blockchain.chain.append(block)
                index += 1
            blockchain.validate_chain_continuity()
            return blockchain

    def convert_to_json(self):
        blockchain_data = []
        for block in self.chain:
            blockchain_data.append(block.convert_to_json())
        return blockchain_data
```

```

def save_to_file(self):
    blockchain_data = self.convert_to_json()
    with open(BLOCKCHAINDB_FILENAME, 'w') as file:
        json.dump(blockchain_data, file, indent=4)

def search_by_batch_number(self, product_name, batch_number) ->
list[Data]:
    search_result = []
    for block in reversed(self.chain):
        for tx in reversed(block.transactions_list):
            if tx.batch_number == batch_number and tx.product_name ==
product_name:
                search_result.append(tx)
    if search_result == []:
        print("No TRACE was found for the specified Product and Batch
Number")
        return []
    trace = []
    for transaction in reversed(search_result): # Reversed to correct
the order of transactions from first to last
        trace.append(transaction)
    trace = Blockchain.sort_trace(trace)
    return trace

@staticmethod
def sort_trace(tx_list: list[Data]):
    sorted_trace = []
    product_txs = []
    manufact_txs = []
    distribution_txs = []
    regulatory_txs = []
    for tx in tx_list:
        if tx.__class__ == Product:
            product_txs.append(tx)
        elif tx.__class__ == ManufacturingQCR:
            manufact_txs.append(tx)
        elif tx.__class__ == DistributionRecords:
            distribution_txs.append(tx)
        elif tx.__class__ == RegulatoryCompliance:
            regulatory_txs.append(tx)

```

```

        distribution_txs = sorted(distribution_txs, key=lambda x:
x.distribution_date)
        sorted_trace += distribution_txs
        sorted_trace += manufact_txs
        sorted_trace += product_txs
        sorted_trace += regulatory_txs
        return sorted_trace

def search_block_from_hash(self, hash):
    for block in reversed(self.chain):
        if block.hash == hash:
            return block
    print("No Block was found for the specified HASH")

def search_transaction_from_tx_hash(self, hash):
    for block in reversed(self.chain):
        for tx in block.transactions_list:
            if tx.tx_hash == hash:
                return tx
    print("No Transaction was found for the specified HASH")

```

- Η μέθοδος φόρτωσης `def load_from_file()` είναι υπεύθυνη για την ανάγνωση ενός μορφότυπου JSON και την προσπάθεια ανακατασκευής ενός στιγμιότυπου Blockchain από τις πληροφορίες που εμπεριέχονται σε αυτό. Αρχικά, φορτώνεται το μορφότυπο στη μνήμη και διαβάζονται σεριακά τα δεδομένα του μορφότυπου ώστε να ανακατασκευαστούν τα αντίστοιχα 'Block' αντικείμενα. Για κάθε συμπληρωμένο μπλοκ από δεδομένα υπολογίζεται το μοναδικό αναγνωριστικό κατακερματισμού του και συγκρίνεται με αυτό που είναι αποθηκευμένο στο μορφότυπο. Αυτό το στάδιο είναι αρκετά σημαντικό καθώς σε περίπτωση κακόβουλης αλλοίωσης του μορφότυπου, αναγνωρίζεται εγκαίρως η παραβίαση της ακεραιότητας του Blockchain και η υπό εξέλιξη διαδικασία της ανακατασκευής τερματίζεται. Έπειτα, αν το συμπληρωμένο μπλοκ είναι έγκυρο, προστίθεται στην αλυσίδα του υπό κατασκευή αντικειμένου Blockchain. Στο τέλος, γίνεται κλήση της μεθόδου `def validate_chain_continuity()` και το κατασκευασμένο 'Blockchain' αντικείμενο φορτώνεται στη μνήμη της εφαρμογής.
- Η μέθοδος μετατροπής στο μορφότυπο JSON `def convert_to_json()` είναι υπεύθυνη για τη μετατροπή των αντικειμένων της κλάσης Blockchain σε μορφή κειμένου που αποτελείται από ζεύγη χαρακτηριστικών-τιμών. Συνολικά, με αυτή τη μέθοδο διατηρείται το τρέχον στιγμιότυπο του Blockchain σε μια μορφή που μπορεί εύκολα να αποθηκευτεί και να μεταδοθεί σύμφωνα με τις ανάγκες και τις απαιτήσεις της εφαρμογής.

- Η μέθοδος `def save_to_file()` είναι υπεύθυνη για την αποθήκευση του τρέχοντος στιγμιότυπου του Blockchain στο μορφότυπο JSON. Η διαδικασία αυτή είναι απαραίτητη για ώστε να διατηρείται η συνέχεια του Blockchain μετά το πέρας της εκτέλεσης της εφαρμογής.
- Η μέθοδος αναζήτησης `def search_by_batch_number()` είναι υπεύθυνη για την εύρεση συγκεκριμένων συναλλαγών στην αλυσίδα των μπλοκ, οι οποίες ταυτίζονται με ένα μοναδικό συνδυασμό ονόματος προϊόντος (`product_name`) και αριθμού παρτίδας (`batch_number`). Αρχικά, φορτώνονται τα κριτήρια αναζήτησης και αρχικοποιείται μια κενή λίστα αποτελεσμάτων (`search_result`). Ύστερα, για κάθε μπλοκ στην αλυσίδα, διαβάζονται σειριακά οι συναλλαγές του με αντίστροφη σειρά (από το τελευταίο μπλοκ στο πρώτο, από την τελευταία συναλλαγή στην πρώτη). Εάν τα κριτήρια αναζήτησης ταυτίζονται με τα πεδία των αναζητούμενων συναλλαγών, τότε οι συναλλαγές αυτές προστίθενται στη λίστα αποτελεσμάτων. Έπειτα, στη λίστα αποτελεσμάτων πραγματοποιείται η κλήση της μεθόδου `def sort_trace()`, η οποία είναι υπεύθυνη για την τελική διάταξη των συναλλαγών ανάλογα με τον τύπο (`type -> αντίστοιχο στάδιο στην εφοδιαστική αλυσίδα`) τους και σύμφωνα με τις απαιτήσεις του Συστήματος Ιχνηλασιμότητας. Σε αντίθετη περίπτωση, επιστρέφεται κενή η λίστα των αποτελεσμάτων, τερματίζεται η αναζήτηση και εκτυπώνεται μήνυμα λάθους για αδυναμία εύρεσης συναλλαγής με τα καθορισμένα κριτήρια αναζήτησης.
Ανάλογα με την υλοποίηση της εκάστοτε δομής μιας αλυσίδας, ενδέχεται να είναι πιο αποτελεσματική η αναζήτηση από τα τελευταία μπλοκ προς τα παλαιότερα:

```
for block in reversed(self.chain):
    for tx in reversed(block.transactions_list):
```

Αυτό συμβαίνει επειδή στα πιο πρόσφατα μπλοκ είναι πιο πιθανό να εμπεριέχονται οι συναλλαγές που σχετίζονται με τα κριτήρια εύρεσης, επομένως με την αντίστροφη αναζήτηση ενδέχεται να μειώνεται ο αριθμός των απαιτούμενων επαναλήψεων, βελτιώνοντας έτσι και το χρόνο εκτέλεσης της μεθόδου.

- Η μέθοδος αναζήτησης `def search_block_from_hash()` είναι υπεύθυνη για την εύρεση ενός συγκεκριμένου μπλοκ στην αλυσίδα χρησιμοποιώντας το μοναδικό αναγνωριστικό κατακερματισμού του. Για κάθε μπλοκ της αλυσίδας, πραγματοποιείται αντίστροφη αναζήτηση με το δοθέν μοναδικό αναγνωριστικό. Εάν βρεθεί το ζητούμενο μπλοκ, επιστρέφεται στην εφαρμογή για περεταίρω επεξεργασία, αλλιώς τερματίζεται η αναζήτηση και επιστρέφεται μήνυμα λάθους. Με την εύρεση ενός μπλοκ δημιουργείται μια διαδρομή ελέγχου, επιτρέποντας τον εντοπισμό ενός συνόλου από μπλοκ στα οποία εμπεριέχονται συγκεκριμένες πληροφορίες που είναι πολύτιμες για σκοπούς κανονιστικής συμμόρφωσης αλλά και μελλοντικής ανάλυσης.

- Η μέθοδος αναζήτησης `def search_transaction_from_tx_hash()` είναι υπεύθυνη για την εύρεση μιας συγκεκριμένης συναλλαγής ενός μπλοκ στην αλυσίδα χρησιμοποιώντας το μοναδικό αναγνωριστικό κατακερματισμού της. Για κάθε συναλλαγή του κάθε μπλοκ της αλυσίδας, πραγματοποιείται αντίστροφη αναζήτηση με το δοθέν μοναδικό αναγνωριστικό. Εάν βρεθεί η ζητούμενη συναλλαγή, επιστρέφεται στην εφαρμογή για περεταίρω επεξεργασία, αλλιώς τερματίζεται η αναζήτηση και επιστρέφεται μήνυμα λάθους. Με την εύρεση μιας συναλλαγής δημιουργείται μια διαδρομή ελέγχου, επιτρέποντας τον εντοπισμό ενός συνόλου από συναλλαγές, διαδικασίας απαραίτητης για τις διάφορες διεργασίες και ανάγκες του Συστήματος Ιχνηλασιμότητας.

3.2.3. Blockchain.json

Στο αρχείο blockchain.json παρουσιάζεται η δομή ενός στιγμιότυπου του Blockchain, το οποίο παράγεται κατά την διάρκεια της εκτέλεσης της εφαρμογής και αποθηκεύεται στο μορφότυπο JSON μετά το πέρας της εκτέλεσης της εφαρμογής. Η δομή του συγκεκριμένου μορφότυπου αποτελείται από έναν πίνακα με εμφωλευμένα Block αντικείμενα. Τα αντικείμενα αυτά αποτελούνται από ζεύγη χαρακτηριστικών-τιμών και από τα γνωρίσματα 'transactions_list', τα οποία είναι πίνακες. Στους πίνακες αυτούς, εμπεριέχονται πληροφορίες για τις συναλλαγές όπως ο τύπος των συναλλαγών μεταξύ και άλλων χαρακτηριστικών.

```
[
  {
    "index": 0,
    "previous_hash": "",
    "timestamp": "20-04-2024 19:41:52:652278",
    "transactions_list": [],
    "hash":
"5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9"
  },
  {
    "index": 1,
    "previous_hash":
"5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9",
    "timestamp": "20-04-2024 19:41:52:652670",
    "transactions_list": [
      {
        "type": "Product",
        "initiator": "MediCare Labs",
        "product_name": "Aspirin Capsules",
        "batch_number": "LOT003",
```

```

        "universal_product_code": "UPC345678901",
        "packaging_type": "Bottle",
        "product_strength": "100mg",
        "expiration_date": "2024-12-31",
        "tx_hash":
"dae36907c7be3cb7ebf965e40ddaaac2cd5ab58f62c8b290b29b96542dafd036"
    },
    {
        "type": "DistributionRecords",
        "initiator": "HealthDispatch",
        "product_name": "Digestive Enzyme Capsules",
        "batch_number": "LOT012",
        "distribution_date": "2024-04-20",
        "distribution_location": "Warehouse12",
        "distribution_quantities": 900,
        "receiver": "PharmacyL",
        "destination_address": "2727 Cedar St",
        "tx_hash":
"5cc9df28b175df488a2b6cd493fe7afa482b3f80e02af09d730ba323b18eabe2"
    },
    {
        "type": "DistributionRecords",
        "initiator": "MediFleet",
        "product_name": "Hair Growth Formula",
        "batch_number": "LOT017",
        "distribution_date": "2024-04-20",
        "distribution_location": "Warehouse17",
        "distribution_quantities": 1200,
        "receiver": "PharmacyQ",
        "destination_address": "3737 Walnut St",
        "tx_hash":
"ac15435d0fcf5e1986ca7be4d7bfa24ad220bc0452f4034aac250745e06a12df"
    },
    {
        "type": "DistributionRecords",
        "initiator": "PharmaFleet",
        "product_name": "Joint Support Tablets",
        "batch_number": "LOT013",
        "distribution_date": "2024-04-20",
        "distribution_location": "Warehouse13",
        "distribution_quantities": 1000,
        "receiver": "PharmacyM",
        "destination_address": "2929 Elm St",

```

```

        "tx_hash":
"3550e8170396065604b5965a02fc744a4c512c1741f2b552f108e31215fc4501"
    },
    {
        "type": "ManufacturingQCR",
        "initiator": "Quality Control Solutions Ltd.",
        "product_name": "Probiotic Capsules",
        "batch_number": "LOT014",
        "datetime": "2024-04-15 10:30:00",
        "manufacturing_facility": "Facility14",
        "quality_control_test_results": "Pass",
        "standard_operating_procedures": "SOP456",
        "tx_hash":
"44f7c350ab2d5a32774275571fbfeb29d0ccf8e24c911646140c9d7cd73786ed"
    }
],
    "hash":
"26747f0ae0290ac76a1e85f6319b70db6a78396534d96e3fe490048367603af7"
}
]

```

3.2.4. Main.py

Στο αρχείο main.py ορίζεται μια διεπαφή γραμμής εντολών (CLI), η οποία αλληλεπιδρά με την τεχνολογία Blockchain της εφαρμογής για το Σύστημα Ιχνηλασιμότητας. Χρησιμοποιείται ένας ατέρμων βρόχος για να διατηρηθεί η διεπαφή ενεργή μέχρις ότου να επιλεγεί η επιθυμητή ενέργεια. Η διεπαφή CLI παρέχει ένα βολικό τρόπο για τους χειριστές της εφαρμογής να αλληλεπιδρούν με την εφοδιαστική αλυσίδα του Blockchain, επιτρέποντας τη διαφάνεια, την ιχνηλασιμότητα και την αποτελεσματική διαχείριση των δεδομένων του Συστήματος Ιχνηλασιμότητας.

```

print("Traceability CLI:\n")
while True:
    print("1) Print Blockchain\n2) Trace Batch Number \n3) Search Block\n4)
Search Transaction\n5) Add Transaction\n6) Amend Blockchain\n7) Show
Current Distribution Balances\n8) Show Transaction History For
Distributor")
    inp = input("> Input: ")
    if inp == "1":
        supply_chain.print_blockchain()
    elif inp == "2":

```

```

        product_name = input("> Provide Product Name: ")
        batch_number = input("> Provide Batch Number: ")
        trace_result = supply_chain.search_by_batch_number(product_name,
batch_number)
        for tx in trace_result:
            if tx:
                tx.print()
                print("\t<----->")
elif inp == "3":
    block_hash = input("> Provide the HASH of a Block: ")
    block = supply_chain.search_block_from_hash(block_hash)
    if block:
        block.print()
elif inp == "4":
    transaction_hash = input("> Provide the HASH of a Transaction: ")
    tx = supply_chain.search_transaction_from_tx_hash(transaction_hash)
    if tx:
        tx.print()
elif inp == "5":
    tx_type = input("> TRANSACTION_TYPE: ")
    if not globals().__contains__(tx_type):
        print("This 'TRANSACTION_TYPE' is not supported.")
    else:
        tx_class = globals()[tx_type]
        variable_list = tx_class.__init__.__code__.co_varnames[1:]
        value_list = []
        for var in variable_list:
            value_list.append(input("> Provide {}: ".format(var)))
        new_tx = tx_class(*value_list)
        supply_chain.add_transaction(new_tx)
        block_number = supply_chain.chain[-1].index
        print("New transaction was added in Block [{}] with HASH:
{}".format(block_number, new_tx.tx_hash))
elif inp == "6":
    supply_chain.save_to_file()
    print("Changes have been made; Blockchain is now updated:",
BLOCKCHAINDB_FILENAME)
elif inp == "7":
    product_name = input("> Provide Product Name: ")
    batch_number = input("> Provide Batch Number: ")
    trace_result = supply_chain.search_by_batch_number(product_name,
batch_number)
    distribution_txs = [tx for tx in trace_result if

```

```

tx.__class__.__name__ == DistributionRecords.__name__]
    balances = calculate_current_balances(distribution_txs)
    print(balances)
    elif inp == "8":
        product_name = input("> Provide Product Name: ")
        batch_number = input("> Provide Batch Number: ")
        trace_result = supply_chain.search_by_batch_number(product_name,
batch_number)
        distribution_txs = [tx for tx in trace_result if
tx.__class__.__name__ == DistributionRecords.__name__]
        distributor = input("> Provide Distributor Name: ")
        specific_distribution_txs = [tx for tx in distribution_txs if
(tx.initiator == distributor or tx.receiver == distributor)]
        [current_balance, transactions] =
trace_transaction_history(specific_distribution_txs, distributor)
        print(current_balance)
        for tx in transactions:
            print(tx)
    else :
        print("INVALID_INPUT.")
        continue
    input("PRESS 'ENTER' TO CONTINUE: ")

```

Στη διεπαφή γραμμής εντολών (CLI) ενσωματώνονται όλες οι μέθοδοι και οι διαδικασίες του Συστήματος Ιχνηλασιμότητας που αναπτύχθηκαν προηγουμένως, παρέχοντας τη δυνατότητα ελέγχου της ορθής λειτουργίας της εφαρμογής δίχως να είναι απαραίτητη η πρόσβαση στο γραφικό περιβάλλον του Συστήματος Ιχνηλασιμότητας.

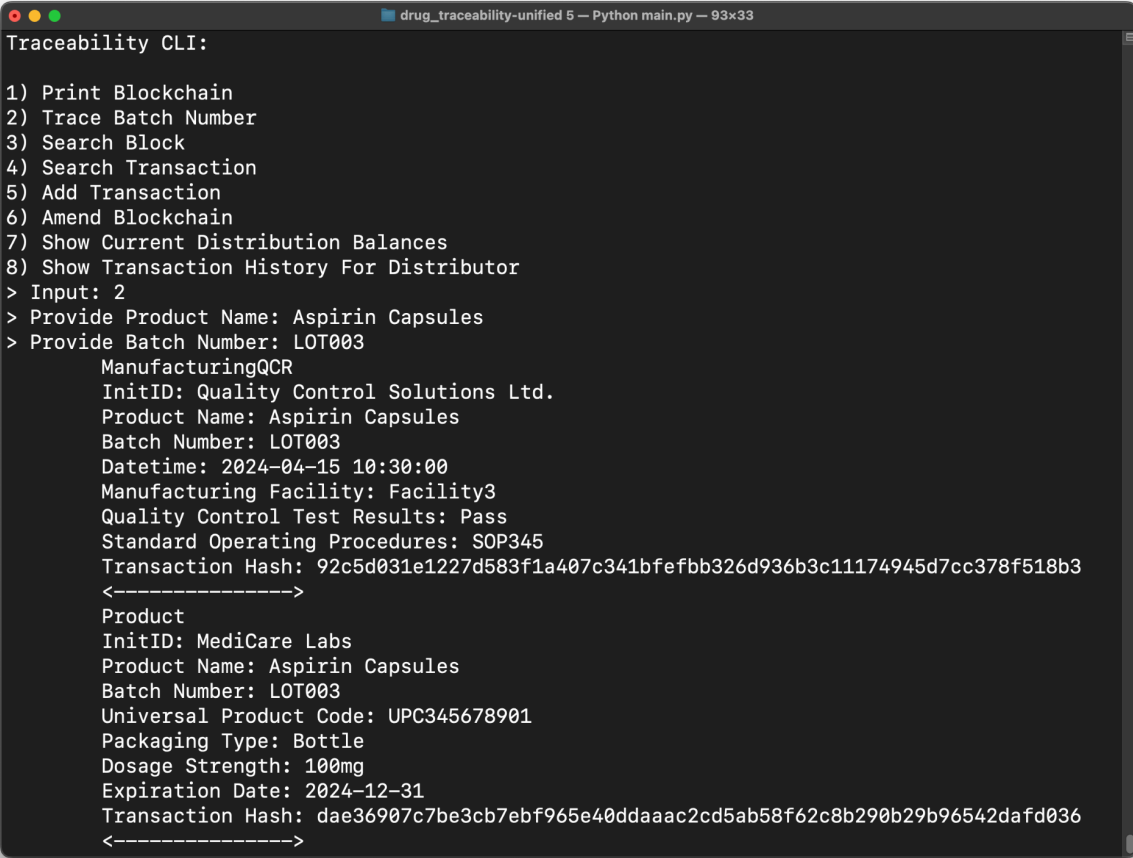
```

Traceability CLI:
1) Print Blockchain
2) Trace Batch Number
3) Search Block
4) Search Transaction
5) Add Transaction
6) Amend Blockchain
7) Show Current Distribution Balances
8) Show Transaction History For Distributor
> Input: 5
> TRANSACTION_TYPE: RegulatoryCompliance
> Provide initiator: FDA
> Provide product_name: Vitamin B1 Tablets
> Provide batch_number: LOT047
> Provide regulatory_approvals: FDA Approved
> Provide compliance_docs: Doc66, Doc47
New transaction was added in Block [17] with HASH: b3ffceb7768870477d02945be20cc1023c0b1a4083daf4b5ce82c25fe1f0fc4d

```

Εικόνα 3.1.: Προσθήκη μιας νέας συναλλαγής (φαρμάκου) στην αλυσίδα των μπλοκ

Στην [εικόνα 3.1.](#), προστίθεται μια νέα συναλλαγή στην αλυσίδα των μπλοκ από τον διαχειριστή του συστήματος. Αρχικά, πληκτρολογείται ο τύπος (TRANSACTION_TYPE) της συναλλαγής. Έπειτα, συμπληρώνονται σειριακά ένα προς ένα τα απαιτούμενα πεδία της κανονιστικής συμμόρφωσης του φαρμάκου. Στο τέλος, παράγεται το αναγνωριστικό κατακερματισμού της τρέχουσας συναλλαγής (0fc4d), η οποία και προστίθεται στο 17^ο μπλοκ της αλυσίδας.



```
Traceability CLI:
1) Print Blockchain
2) Trace Batch Number
3) Search Block
4) Search Transaction
5) Add Transaction
6) Amend Blockchain
7) Show Current Distribution Balances
8) Show Transaction History For Distributor
> Input: 2
> Provide Product Name: Aspirin Capsules
> Provide Batch Number: LOT003
ManufacturingQCR
InitID: Quality Control Solutions Ltd.
Product Name: Aspirin Capsules
Batch Number: LOT003
Datetime: 2024-04-15 10:30:00
Manufacturing Facility: Facility3
Quality Control Test Results: Pass
Standard Operating Procedures: SOP345
Transaction Hash: 92c5d031e1227d583f1a407c341bfefbb326d936b3c11174945d7cc378f518b3
<----->
Product
InitID: MediCare Labs
Product Name: Aspirin Capsules
Batch Number: LOT003
Universal Product Code: UPC345678901
Packaging Type: Bottle
Dosage Strength: 100mg
Expiration Date: 2024-12-31
Transaction Hash: dae36907c7be3cb7ebf965e40ddaaac2cd5ab58f62c8b290b29b96542dafd036
<----->
```

Εικόνα 3.2.: Ιχνηλάτηση ενός φαρμάκου στην αλυσίδα των μπλοκ

Στην [εικόνα 3.2.](#), ιχνηλατείται ένα φάρμακο στην αλυσίδα των μπλοκ από τον διαχειριστή του συστήματος. Αρχικά, πληκτρολογείται το όνομα προϊόντος και ο αριθμός παρτίδας. Στη συνέχεια, αναζητούνται οι συναλλαγές της αλυσίδας που ταυτίζονται με τα παρεχόμενα κριτήρια. Στο τέλος, εκτυπώνεται το αποτέλεσμα της αναζήτησης στο τερματικό του διαχειριστή.

3.2.4.1. Balances

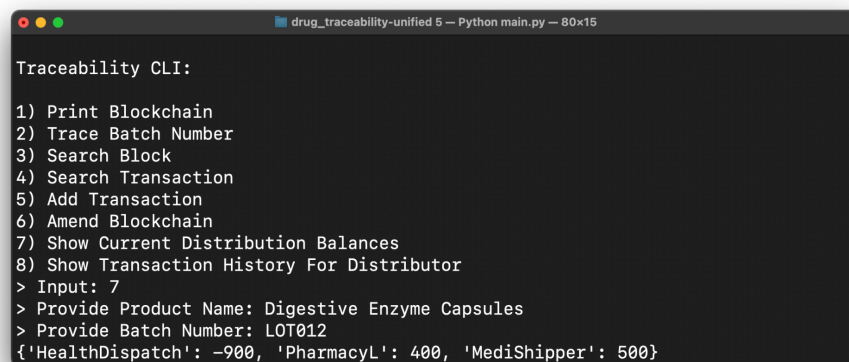
```
def calculate_current_balances(distribution_txs: List[DistributionRecords])
-> dict:
    accounts = {}
    for tx in distribution_txs:
        initiator = tx.initiator
        receiver = tx.receiver
        quantities = int(tx.distribution_quantities)

        if initiator not in accounts.keys():
            accounts[initiator] = 0
        accounts[initiator] -= quantities

        if receiver not in accounts.keys():
            accounts[receiver] = 0
        accounts[receiver] += quantities

    return accounts
```

Η μέθοδος ισολογισμού `def calculate_current_balances()` είναι υπεύθυνη για την καταγραφή της διακίνησης των φαρμάκων μεταξύ των οντοτήτων του Συστήματος Ιχνηλασιμότητας. Αρχικά, εισάγεται μια λίστα συναλλαγών του σταδίου διανομής (Distribution Stage: List[DistributionRecords]), όπου στην κάθε συναλλαγή περιλαμβάνεται η διανομή μιας ποσότητας ενός φαρμάκου μεταξύ δύο οντοτήτων. Έπειτα, για κάθε συναλλαγή ενημερώνονται τα λογιστικά υπόλοιπα των εμπλεκόμενων φορέων. Έτσι, η μεταφερόμενη ποσότητα αφαιρείται από το υπόλοιπο του αποστολέα (Expediter_ID) και προστίθεται στο υπόλοιπο του παραλήπτη (Addressee). Στο τέλος, επιστρέφεται στην εφαρμογή ενά λεξικό με τα τρέχοντα υπόλοιπα των φορέων που εμπλέκονται στις συναλλαγές.



Εικόνα 3.3.: Ισολογισμός διανομής φαρμάκων μεταξύ τριών οντοτήτων

Στην [εικόνα 3.3.](#), καταγράφεται η διανομή μιας ποσότητας φαρμάκου από τον αρχικό προμηθευτή/κατασκευαστή (vendor) στους εμπόρους λιανικής (retailers). Στην περίπτωση του παραδείγματος, ο προμηθευτής 'HealthDispatch' αποστέλλει 900 κάψουλες πεπτικών ενζύμων σε εμπόρους λιανικής, επομένως παρουσιάζει ελλειμματικό ισοζύγιο (-900). Οι έμποροι λιανικής εισπράττουν το εμπόρευμα, συνεπώς παρουσιάζουν πλεονασματικό ισοζύγιο ('PharmacyL': +400, 'MediShipper': +500).

3.2.4.2. Distributor's Transaction History

```
def trace_transaction_history(distribution_txs: List[DistributionRecords],
                              distributor: str):
    tx_history = []
    balance = 0
    for distribution_tx in distribution_txs:
        initiator = distribution_tx.initiator
        receiver = distribution_tx.receiver
        quantities = int(distribution_tx.distribution_quantities)

        if initiator == distributor:
            balance -= quantities
            tx = {
                "tx_type": "send",
                "from": initiator,
                "to": receiver,
                "quantities": str(quantities)
            }
            tx_history.append(tx)

        elif receiver == distributor:
            balance += quantities
            tx = {
                "tx_type": "receive",
                "from": initiator,
                "to": receiver,
                "quantities": str(quantities)
            }
            tx_history.append(tx)

    return [balance, tx_history]
```

Η μέθοδος `def trace_transaction_history()` είναι υπεύθυνη για την καταγραφή του ιστορικού συναλλαγών ενός διανομέα στο Σύστημα Ιχνηλασιμότητας. Αρχικά, εισάγεται μια λίστα συναλλαγών του σταδίου διανομής και το όνομα του διανομέα. Έπειτα, για κάθε συναλλαγή καθορίζεται εάν ο διανομέας πρόκειται για τον αποστολέα ή τον παραλήπτη μιας ποσότητας ενός φαρμάκου, ενημερώνεται το ιστορικό του αναλόγως και καταγράφεται η συναλλαγή σε μια λίστα ιστορικού. Στο τέλος, επιστρέφεται το τρέχον υπόλοιπο του διανομέα και μια λίστα με όλες τις συναλλαγές με τις οποίες συσχετίζεται.

Στο προηγούμενο παράδειγμα της συναλλαγής με όνομα προϊόντος `'Digestive Enzyme Capsules'` και αριθμό παρτίδας `'LOT012'`, επιστρέφεται το ιστορικό διακίνησης ανάλογα με τον παρεχόμενο διανομέα.

```
> Provide Product Name: Digestive Enzyme Capsules
> Provide Batch Number: LOT012
> Provide Distributor Name: HealthDispatch
-900
{'tx_type': 'send', 'from': 'HealthDispatch', 'to': 'PharmacyL',
'quantities': '900'}

> Provide Product Name: Digestive Enzyme Capsules
> Provide Batch Number: LOT012
> Provide Distributor Name: PharmacyL
400
{'tx_type': 'receive', 'from': 'HealthDispatch', 'to': 'PharmacyL',
'quantities': '900'}
{'tx_type': 'send', 'from': 'PharmacyL', 'to': 'MediShipper',
'quantities': '500'}

> Provide Product Name: Digestive Enzyme Capsules
> Provide Batch Number: LOT012
> Provide Distributor Name: MediShipper
500
{'tx_type': 'receive', 'from': 'PharmacyL', 'to': 'MediShipper',
'quantities': '500'}
```

```
drug_traceability-unified 5 — Python main.py — 85x17

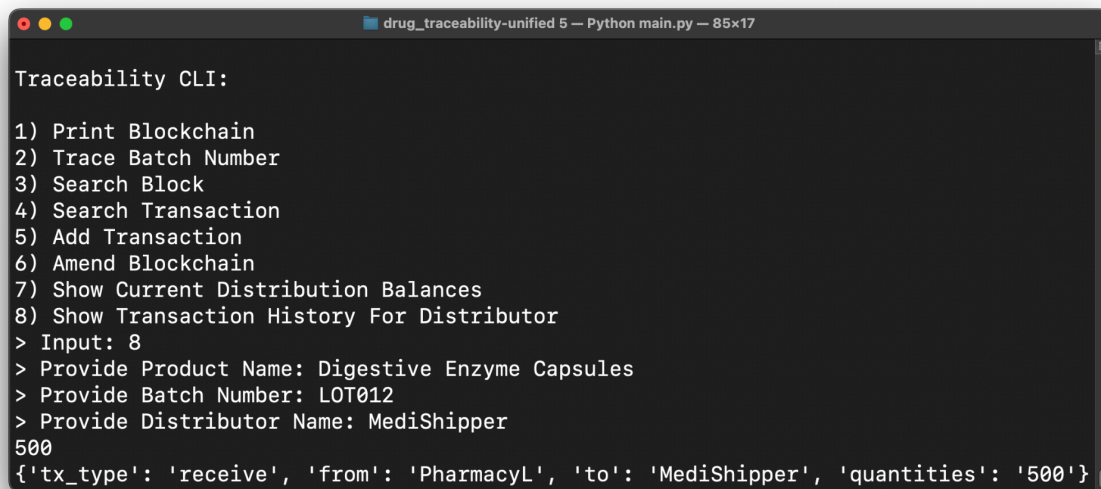
Traceability CLI:
1) Print Blockchain
2) Trace Batch Number
3) Search Block
4) Search Transaction
5) Add Transaction
6) Amend Blockchain
7) Show Current Distribution Balances
8) Show Transaction History For Distributor
> Input: 8
> Provide Product Name: Digestive Enzyme Capsules
> Provide Batch Number: LOT012
> Provide Distributor Name: HealthDispatch
-900
{'tx_type': 'send', 'from': 'HealthDispatch', 'to': 'PharmacyL', 'quantities': '900'}
```

Εικόνα 3.4.: Ιστορικό διακίνησης φαρμάκων της οντότητας 'HealthDispatch'

```
drug_traceability-unified 5 — Python main.py — 90x18

Traceability CLI:
1) Print Blockchain
2) Trace Batch Number
3) Search Block
4) Search Transaction
5) Add Transaction
6) Amend Blockchain
7) Show Current Distribution Balances
8) Show Transaction History For Distributor
> Input: 8
> Provide Product Name: Digestive Enzyme Capsules
> Provide Batch Number: LOT012
> Provide Distributor Name: PharmacyL
400
{'tx_type': 'receive', 'from': 'HealthDispatch', 'to': 'PharmacyL', 'quantities': '900'}
{'tx_type': 'send', 'from': 'PharmacyL', 'to': 'MediShipper', 'quantities': '500'}
```

Εικόνα 3.5.: Ιστορικό διακίνησης φαρμάκων της οντότητας 'PharmacyL'



```
Traceability CLI:
1) Print Blockchain
2) Trace Batch Number
3) Search Block
4) Search Transaction
5) Add Transaction
6) Amend Blockchain
7) Show Current Distribution Balances
8) Show Transaction History For Distributor
> Input: 8
> Provide Product Name: Digestive Enzyme Capsules
> Provide Batch Number: LOT012
> Provide Distributor Name: MediShipper
500
{'tx_type': 'receive', 'from': 'PharmacyL', 'to': 'MediShipper', 'quantities': '500'}
```

Εικόνα 3.6.: Ιστορικό διακίνησης φαρμάκων της οντότητας 'MediShipper'

3.3. Κώδικας Διεπαφής

Στο προγραμματιστικό επίπεδο της εφαρμογής υλοποιήθηκαν οι βασικές λειτουργίες της τεχνολογίας Blockchain για τις πρωτεύουσες ανάγκες της ιχνηλασιμότητας φαρμάκων. Ωστόσο, εξαιτίας της πολυπλοκότητας χειρισμού τέτοιων συστημάτων από τους τελικούς χρήστες, είναι απαραίτητη η δημιουργία ενός γραφικού περιβάλλοντος ως ένα σημείο αναφοράς για τους φορείς που είναι υπεύθυνοι για την παραγωγή, τη διακίνηση και την πώληση φαρμάκων. Με τη διεπαφή αυτή οπτικοποιείται η ιχνηλάτηση σε πραγματικό χρόνο, ελέγχεται η αυθεντικότητα των φαρμακευτικών σκευασμάτων, εξασφαλίζεται η κανονιστική συμμόρφωση και παρέχονται σημαντικές πληροφορίες για τη διακίνηση μεταξύ των αρμόδιων φορέων.

3.3.1. App.py

Στο αρχείο app.py ορίζονται οι προγραμματιστικές διαδρομές (routes) για τη δημιουργία μιας διαδικτυακής εφαρμογής χρησιμοποιώντας τη βιβλιοθήκη Flask για τη διαχείριση της εφοδιαστικής αλυσίδας, στηριζόμενη στην τεχνολογία Blockchain που αναπτύχθηκε προηγουμένως. Οι διαδρομές είναι στην προκειμένη περίπτωση διευθύνσεις URL, στις οποίες έχουν πρόσβαση οι τελικοί χρήστες για την αλληλεπίδραση με τη διαδικτυακή εφαρμογή. Με αυτόν τον τρόπο, οι διαδρομές χειρίζονται τα εισερχόμενα αιτήματα HTTP από τους χρήστες και καθορίζονται τελικώς οι

ενέργειες που πρέπει να γίνονται όταν ένας χρήστης επισκέπτεται μια συγκεκριμένη διεύθυνση URL.

3.3.1.1. '/blockchain', '/blockchain_json'

```
@app.route('/blockchain', methods=['GET'])
@app.route('/blockchain_json', methods=['GET'])
def blockchain_viewer():
    if request.path == '/blockchain_json':
        return jsonify(supply_chain.convert_to_json())
    return render_template('viewer.html')
```

Οι παραπάνω διαδρομές λειτουργούν ως τελικά σημεία (endpoints) για την προβολή του περιεχομένου όλων των μπλοκ της εφοδιαστικής αλυσίδας στο Blockchain. Η διαδρομή `'/blockchain'` είναι υπεύθυνη για την ανακατεύθυνση του τελικού χρήστη στη σελίδα περιήγησης της εφοδιαστικής αλυσίδας. Έπειτα, με τη διαδρομή `'/blockchain_json'` επιστρέφονται τα δεδομένα των μπλοκ της αλυσίδας στο μορφότυπο JSON.

```
const jsonData = fetch('/blockchain_json')
    .then(response => response.json())
    .then(data => {
        return data;
    })
    .then(arrayData => {
        console.log(arrayData);
        createAllBlocks(arrayData);
    })
    .catch(error => {
        console.error('Error:', error);
    });

function createAllBlocks(blocksData) {
    const blockchainContainer =
document.getElementById('blockchain-container');
    blocksData.forEach(blockData => {
        const blockElement = createBlock(blockData);
        blockchainContainer.appendChild(blockElement);
    });
}
// createBlock(blockData) and createTransactionTable(transaction, index)
are omitted from this sample!
```

Σε προγραμματιστικό επίπεδο, αποστέλλεται ένα αίτημα 'GET' από την πλευρά της διαδικτυακής εφαρμογής (client-side) στην πλευρά του διακομιστή (server-side). Πιο συγκεκριμένα, με την εντολή `fetch('/blockchain_json')` αποστέλλεται το αίτημα 'GET' στο τελικό σημείο `@app.route('/blockchain_json')` του διακομιστή. Στη διαδρομή αυτή, μετατρέπονται τα ζητούμενα δεδομένα στο μορφότυπο JSON και επιστρέφονται πίσω ως απάντηση στη διαδικτυακή εφαρμογή. Ύστερα, την απάντηση χειρίζονται ασύγχρονα τα αντικείμενα υπόσχεσης, με τα σφάλματα των αιτημάτων να καταγράφονται από την κονσόλα. Στο τέλος, πραγματοποιείται κλήση της συνάρτησης `createAllBlocks()` για την οπτική δημιουργία των μπλοκ από τα δοθέντα δεδομένα.

3.3.1.2. '/add_tx', '/submit_form'

```
@app.route('/add_tx')
def add_tx():
    return render_template('add_tx.html')
@app.route('/submit_form', methods=['POST'])
def submit_form():

    tx_type = str(request.form['transactionType'])
    print(tx_type)

    if tx_type == 'Product':
        p_init = str(request.form['p_init'])
        p_pn = str(request.form['p_productName'])
        p_bn = str(request.form['p_batchNumber'])
        upc = str(request.form['upc'])
        pack_type = str(request.form['packagingType'])
        if pack_type == 'Other':
            other_packaging_type = str(request.form['otherPackagingInput'])
            pack_type = other_packaging_type
        psval = str(request.form['productStrengthValue'])
        psunit = str(request.form['productStrengthUnit'])
        dosage_strength = psval + ' ' + psunit
        expdate = str(request.form['expirationDate'])
        expdtobj = libdt.fromisoformat(expdate)
        fexpdate = str(expdtobj.strftime("%m/%Y"))

        pd = Product(initiator=p_init, batch_number=p_bn,
product_name=p_pn, universal_product_code=upc, packaging_type=pack_type,
product_strength=dosage_strength, expiration_date=fexpdate)
        sfields = {
```

```

        'InitiatorID': p_init,
        'Product Name': p_pn,
        'Batch Number': p_bn,
        'Universal Product Code': upc,
        'Packaging Type': pack_type,
        'Dosage Strength': dosage_strength,
        'Expiration Date': fexpdate,
    }

    supply_chain.add_transaction(pd)
    supply_chain.save_to_file()
    block_number = supply_chain.chain[-1].index

    return render_template('sf_success.html',
block_number=block_number, fields=sfields, tx_hash=pd.tx_hash)

    elif tx_type == 'ManufacturingQCR':
        # Omitted code
        mfg = ManufacturingQCR(initiator=m_init, product_name=m_pn,
batch_number=m_bn, datetime=fdt, manufacturing_facility=mf,
quality_control_test_results=qcr, standard_operating_procedures=sop)
        sfields = {
            'InitiatorID': m_init,
            'Product Name': m_pn,
            'Batch Number': m_bn,
            'Manufacturing Date': fdt,
            'Manufacturing Facility': mf,
            'Quality Control Test Results': qcr,
            'Standard Operating Procedures': sop,
        }
        supply_chain.add_transaction(mfg)
        supply_chain.save_to_file()
        block_number = supply_chain.chain[-1].index

        return render_template('sf_success.html',
block_number=block_number, fields=sfields, tx_hash=mfg.tx_hash)

    elif tx_type == 'DistributionRecords':
        # Omitted code
        distr = DistributionRecords(initiator=d_init, product_name=d_pn,
batch_number=d_bn, distribution_date=distrdate,
distribution_location=distrloc, distribution_quantities=distrqnt,
receiver=d_addressee, destination_address=destaddr)

```

```

sfields = {
    'ExpediterID': d_init,
    'Product Name': d_pn,
    'Batch Number': d_bn,
    'Distribution Date': fdistrdate,
    'Distribution Location': distrloc,
    'Distribution Quantities': distrqnt,
    'Addressee': d_addressee,
    'Destination Address': destaddr,
}
supply_chain.add_transaction(distr)
supply_chain.save_to_file()
block_number = supply_chain.chain[-1].index

return render_template('sf_success.html',
block_number=block_number, fields=sfields, tx_hash=distr.tx_hash)

elif tx_type == 'RegulatoryCompliance':
    # Omitted code
    regcomp = RegulatoryCompliance(initiator=r_init, product_name=r_pn,
batch_number=r_bn, regulatory_approvals=regappr, compliance_docs=compdocs)
    sfields = {
        'InitiatorID': r_init,
        'Product Name': r_pn,
        'Batch Number': r_bn,
        'Regulatory Approvals': regappr,
        'Compliance Documents': compdocs,
    }
    supply_chain.add_transaction(regcomp)
    supply_chain.save_to_file()
    block_number = supply_chain.chain[-1].index

    return render_template('sf_success.html',
block_number=block_number, fields=sfields, tx_hash=regcomp.tx_hash)

```

Οι παραπάνω διαδρομές λειτουργούν ως τελικά σημεία (endpoints) για την προσθήκη μιας νέας συναλλαγής στην αλυσίδα των μπλοκ. Η διαδρομή `’/add_tx’` είναι υπεύθυνη για την ανακατεύθυνση του χρήστη στη σελίδα με τη φόρμα συναλλαγής. Έπειτα, με τη διαδρομή `’/submit_form’` δημιουργείται το αντικείμενο της συναλλαγής και ακολούθως ενημερώνονται παράλληλα η αλυσίδα των μπλοκ με τη νέα προσθήκη αλλά και η βάση δεδομένων του Blockchain.

```
<form action="/submit_form" method="post" id="transactionForm">
</form>
```

Σε προγραμματιστικό επίπεδο, αποστέλλεται ένα αίτημα 'POST' κατά την υποβολή της φόρμας νέας συναλλαγής από το χρήστη (client-side) στο διακομιστή (server-side). Ειδικότερα, αποστέλλεται το 'POST' στο τελικό σημείο `@app.route('/submit_form')` του διακομιστή. Η μέθοδος `submit_form()` εκτελείται για τη διαχείριση αυτού του αιτήματος. Αρχικά, αναζητείται ο τύπος συναλλαγής ('tx_type') από τα δεδομένα της φόρμας που υποβλήθηκαν. Ανάλογα με τον τύπο της συναλλαγής, ανακτώνται τα ανάλογα πεδία της φόρμας και δημιουργούνται αντικείμενα διαφορετικών τύπων συναλλαγών (π.χ. 'Product', 'ManufacturingQCR', κλπ.). Στη συνέχεια, προστίθεται η συναλλαγή στην αλυσίδα των μπλοκ και ενημερώνεται η βάση δεδομένων του Blockchain με την προσθήκη. Στο τέλος, επιστρέφεται μια σελίδα ανασκόπησης σχετικά με την προστιθέμενη συναλλαγή, συμπεριλαμβανομένου του αριθμού μπλοκ, των πεδίων συναλλαγής και του μοναδικού κατακερματισμού της.

3.3.1.3. '/trace', '/search_batch_number/

```
@app.route('/trace', methods=['GET'])
def trace_drug():
    return render_template('trace.html')
@app.route('/search_batch_number/<product_name>/<batch_number>',
methods=['GET'])
def search_batch_number_endpoint(product_name, batch_number):
    trace_result = supply_chain.search_by_batch_number(product_name,
batch_number)
    if len(trace_result) == 0:
        return []
    json_data = []
    for tx in trace_result:
        if tx:
            json_data.append(tx.serialize())
    return jsonify(json_data)
```

Οι παραπάνω διαδρομές λειτουργούν ως τελικά σημεία (endpoints) για την αναζήτηση (ιχνηλάτηση) συναλλαγών στην αλυσίδα των μπλοκ του Blockchain. Η διαδρομή `'/trace'` είναι υπεύθυνη για την ανακατεύθυνση του χρήστη στη σελίδα εύρεσης συναλλαγών. Στη συνέχεια, με τη διαδρομή `'/search_batch_number/'` αναζητούνται οι επιθυμητές συναλλαγές και επιστρέφονται οπτικά τα αποτελέσματα στο χρήστη.

```

async function fetchData(productName, batchNumber) {
  try {
    const response = await
fetch(`/search_batch_number/${productName}/${batchNumber}`);
    const data = await response.json();
    console.log(data);
    displayResult(data);
    return data;
  } catch (error) {
    console.error('Error:', error);
    return Promise.resolve();
  }
}

```

Σε προγραμματιστικό επίπεδο, αποστέλλεται ένα αίτημα 'GET' από την πλευρά της διαδικτυακής εφαρμογής (client-side) στην πλευρά του διακομιστή (server-side). Με την εντολή `fetch(`/search_batch_number/${productName}/${batchNumber}`)` αποστέλλεται το αίτημα 'GET' με τις μεταβλητές όνομα προϊόντος και αριθμός παρτίδας στο τελικό σημείο `@app.route('/search_batch_number/<product_name>/<batch_number>')` του διακομιστή. Στη διαδρομή αυτή, αναζητούνται στην αλυσίδα των μπλοκ οι συναλλαγές με τα προκαθορισμένα κριτήρια. Σε περίπτωση μη εύρεσης ταυτόσημων πληροφοριών, επιστρέφεται μια κενή λίστα συναλλαγών. Διαφορετικά, τα αποτελέσματα αναζήτησης σειριοποιούνται στο μορφότυπο JSON και επιστρέφονται πίσω ως απάντηση στη διαδικτυακή εφαρμογή. Ύστερα, την απάντηση χειρίζονται ασύγχρονα τα αντικείμενα υπόσχεσης, με τα σφάλματα των αιτημάτων να καταγράφονται από την κονσόλα. Στο τέλος, πραγματοποιείται κλήση της συνάρτησης `displayResult()`; για την οπτική παρουσίαση των αποτελεσμάτων αναζήτησης στο χρήστη.

3.3.2. Viewer.js

```

document.addEventListener("DOMContentLoaded", function() {

  const jsonData = fetch("/blockchain_json")
    .then(response => response.json())
    .then(data => {
      return data;
    })
    .then(arrayData => {
      console.log(arrayData);
      createAllBlocks(arrayData);
    })

```

```

        .catch(error => {
            console.error('Error:', error);
        });

function createBlock(blockData) {
    const blockElement = document.createElement("div");
    blockElement.className = "block";

    const blockHTML = `
    <h2>Block #${blockData.index}</h2>
    <table class="block-info">
        <tr>
            <th>Previous Hash</th>
            <td>${blockData.previous_hash}</td>
        </tr>
        <tr>
            <th>Timestamp</th>
            <td>${blockData.timestamp}</td>
        </tr>
        <tr>
            <th>Hash</th>
            <td>${blockData.hash}</td>
        </tr>
    </table>
    `;

    blockElement.innerHTML = blockHTML;
    const toggleButton = document.createElement("button");
    toggleButton.textContent = "Show Transactions";
    blockElement.appendChild(toggleButton);
    blockData.transactions_list.forEach((transaction, index) => {
        const transactionTable =
createTransactionTable(transaction, index + 1);
        transactionTable.style.display = "none";
        blockElement.appendChild(transactionTable);
    });
    toggleButton.addEventListener("click", () => {
        const transactionTables =
blockElement.querySelectorAll(".transaction-table");
        transactionTables.forEach(table => {
            if (table.style.display === "none") {
                table.style.display = "table";
            } else {

```

```

        table.style.display = "none";
    }
    });
});
return blockElement;
}
function createTransactionTable(transaction, index) {
    const transactionTable = document.createElement("table");
    transactionTable.className = "transaction-table";
    transactionTable.innerHTML = `<tr><th>Transaction
#${index}</th></tr>`;
    const typeConfig = {
        "Product": {
            propertiesToShow: ["type", "initiator",
"product_name", "batch_number", "universal_product_code", "packaging_type",
"product_strength", "expiration_date", "tx_hash"],
            displayNameMapping: {
                "type": "Type",
                "initiator": "InitID",
                "product_name": "Prod. Name",
                "batch_number": "Batch No.",
                "universal_product_code": "UPC",
                "packaging_type": "Packaging",
                "product_strength": "Dosage",
                "expiration_date": "Expiration",
                "tx_hash": "TXN_HASH"
            }
        }
    }
    // The remaining typeConfigs are omitted from this
sample!
    };
    const typeConfigEntry = typeConfig[transaction["type"]];
    if (typeConfigEntry) {
        const {
            propertiesToShow,
            displayNameMapping
        } = typeConfigEntry;

        propertiesToShow.forEach(property => {
            const row = document.createElement("tr");
            const displayName = displayNameMapping[property] ||
property;

            row.innerHTML =

```

```

`<th>${displayName}</th><td>${transaction[property]}</td>`;
        transactionTable.appendChild(row);
    });
}
return transactionTable;
}

function createAllBlocks(blocksData) {
    const blockchainContainer =
document.getElementById("blockchain-container");
    blocksData.forEach(blockData => {
        const blockElement = createBlock(blockData);
        blockchainContainer.appendChild(blockElement);
    });
}

});

```

3.3.3. Add_TX.js

```

function showFields() {
    document.getElementById('productFields').classList.add('hidden');

    document.getElementById('manufacturingQCRFields').classList.add('hidden');

    document.getElementById('distributionRecordsFields').classList.add('hidden'
);

    document.getElementById('regulatoryComplianceFields').classList.add('hidden
');
    document.querySelectorAll('#transactionForm input, #transactionForm
select, #transactionForm textarea').forEach(el => el.required = false);
    [...document.getElementById('transactionForm').elements].forEach(el
=> el.id !== 'transactionType' && el.id !== 'submitButton' && (el.value =
''));

    var selectedType = document.getElementById('transactionType').value;

    switch (selectedType) {
        case 'Product':

```

```

document.getElementById('productFields').classList.remove('hidden');
    p_init.required = true;
    p_productName.required = true;
    p_batchNumber.required = true;
    upc.required = true;
    packagingType.required = true;
    productStrengthValue.required = true;
    productStrengthUnit.required = true;
    expirationDate.required = true;
    break;
case 'ManufacturingQCR':
    // Omitted code
    break;
case 'DistributionRecords':
    // Omitted code
    break;
case 'RegulatoryCompliance':
    // Omitted code
    break;
default:
    break;
}

document.getElementById('submitButton').classList.remove('hidden');
}

function handlePackagingType() {
    var packagingTypeDropdown = document.getElementById('packagingType');
    var otherPackagingTypeDiv =
document.getElementById('otherPackagingType');
    var otherPackagingInput =
document.getElementById('otherPackagingInput');

    if (packagingTypeDropdown.value === 'Other') {
        otherPackagingTypeDiv.classList.remove('hidden');
        otherPackagingInput.required = true;
    } else {
        otherPackagingTypeDiv.classList.add('hidden');
        otherPackagingInput.required = false;
    }
}

const receiverOptions = {

```

```

    'Company A': ["ClientA_1", "ClientA_2", "ClientA_3"],
    'Company B': ["ClientB_1", "ClientB_2", "ClientB_3"]
    // Add more mappings as needed
};

function updateReceiverOptions() {
    const senderInput = document.getElementById("d_init");
    const receiverContainer =
document.getElementById("receiverContainer");

    const sender = senderInput.value;
    const isKnownSender =
Array.from(document.getElementById("knownSenders").options)
    .map(option => option.value)
    .includes(sender);

    receiverContainer.innerHTML = "";

    if (isKnownSender) {
        const receiverDropdown = document.createElement("select");
        const defaultReceiverDropdown =
document.createElement("option");
        defaultReceiverDropdown.value = "";
        defaultReceiverDropdown.text = "Select Partner";
        defaultReceiverDropdown.selected = true;
        defaultReceiverDropdown.disabled = true;
        receiverDropdown.id = "receiver";
        receiverDropdown.name = "receiver";
        receiverDropdown.required = true;
        receiverDropdown.add(defaultReceiverDropdown);

        receiverOptions[sender].forEach(receiver => {
            const option = document.createElement("option");
            option.value = receiver;
            option.text = receiver;
            receiverDropdown.add(option);
        });

        const customOption = document.createElement("option");
        customOption.value = "Custom";
        customOption.text = "Custom Partner";
        receiverDropdown.add(customOption);
    }
}

```

```

        receiverDropdown.onchange = function() {
            if (receiverDropdown.value === "Custom") {
                const customReceiverInput =
document.createElement("input");
                customReceiverInput.type = "text";
                customReceiverInput.id = "receiver";
                customReceiverInput.name = "receiver";
                customReceiverInput.pattern = "[a-zA-Z0-9 ]+";
                customReceiverInput.title = "INVALID INIT FORMAT:
SPECIAL_CHARS DISALLOWED";
                customReceiverInput.required = true;
                receiverContainer.innerHTML = "";
                receiverContainer.appendChild(customReceiverInput);
            }
        };

        receiverContainer.appendChild(receiverDropdown);

    } else {
        const receiverText = document.createElement("input");
        receiverText.type = "text";
        receiverText.id = "receiver";
        receiverText.name = "receiver";
        receiverText.pattern = "[a-zA-Z0-9 ]+";
        receiverText.title = "INVALID INIT FORMAT: SPECIAL_CHARS
DISALLOWED";
        receiverText.required = true;
        receiverContainer.appendChild(receiverText);
    }
}

updateReceiverOptions();

```

3.3.4. Trace.js

```

function displayResult(transactions) {
    const resultDiv = document.getElementById('result');
    resultDiv.innerHTML = '';
    var cnt = 1;
    if (transactions.length === 0) {
        resultDiv.innerHTML = '<p>No Transactions were found for the

```

```

specified combination.</p>';
    } else {
        transactions.forEach(transaction => {
            const transactionTable =
createTransactionTable(transaction, cnt);
            cnt = cnt + 1;
            resultDiv.appendChild(transactionTable);
        })
    }
}

async function fetchData(productName, batchNumber) {
    try {
        const response = await
fetch(`/search_batch_number/${productName}/${batchNumber}`);
        const data = await response.json();
        console.log(data);
        displayResult(data);
        return data;
    } catch (error) {
        console.error('Error:', error);
        return Promise.resolve();
    }
}

document.getElementById('searchForm').addEventListener('submit',
function(event) {
    event.preventDefault();

    var productName = document.getElementById('productName').value;
    var batchNumber = document.getElementById('batchNumber').value;
    fetchData(productName, batchNumber);
});

function createTransactionTable(transaction, cnt) {
    const transactionTable = document.createElement("table");
    transactionTable.className = "transaction-table";
    transactionTable.innerHTML = `<tr><th>TXN #${cnt}</th></tr>`;

    const typeConfig = {
        "Product": {
            propertiesToShow: ["type", "initiator", "product_name",
"batch_number", "universal_product_code", "packaging_type",

```

```

"product_strength", "expiration_date", "tx_hash"],
    displayNameMapping: {
        "type": "Type",
        "initiator": "InitID",
        "product_name": "Prod. Name",
        "batch_number": "Batch No.",
        "universal_product_code": "UPC",
        "packaging_type": "Packaging",
        "product_strength": "Dosage",
        "expiration_date": "Expiration",
        "tx_hash": "TXN_HASH"
    }
}
// The remaining typeConfigs are omitted from this sample!
};

const typeConfigEntry = typeConfig[transaction["type"]];

if (typeConfigEntry) {
    const {
        propertiesToShow,
        displayNameMapping
    } = typeConfigEntry;

    propertiesToShow.forEach(property => {
        const row = document.createElement("tr");
        const displayName = displayNameMapping[property] ||
property;
        row.innerHTML =
`<th>${displayName}</th><td>${transaction[property]}</td>`;
        transactionTable.appendChild(row);
    });
}

return transactionTable;
}

```

3.4. Εκτέλεση Συστήματος Ιχνηλασιμότητας

Για την εγκατάσταση της εφαρμογής του Συστήματος Ιχνηλασιμότητας σε τοπικό επίπεδο για σκοπούς δοκιμής και εντοπισμού σφαλμάτων, απαιτείται η χρήση ενός λειτουργικού συστήματος που να υποστηρίζει την πρόσφατη έκδοση της γλώσσας

προγραμματισμού Python3. Ενδείκνυται η έκδοση 3.9.6 ή μεταγενέστερη. Ακόμη, για τις διαδικασίες του διακομιστή απαιτείται η εγκατάσταση της βιβλιοθήκης Flask στην Python3, χρησιμοποιώντας το πρόγραμμα εγκατάστασης πακέτων της (PIP3).

- ☐ Python3: <https://www.python.org/downloads/>
- ☐ PIP3: <https://packaging.python.org/en/latest/tutorials/installing-packages/>
- ☐ Flask 3.0.3: <https://flask.palletsprojects.com/en/3.0.x/installation/>

Η κατασκευή της εφαρμογής του Συστήματος Ιχνηλασιμότητας πραγματοποιήθηκε στο λειτουργικό σύστημα MacOS 13.6.2 (22G320), στο οποίο είναι προεγκατεστημένη η Python 3.9.6. Παρατίθενται ακολούθως οι ενδεικτικές ρυθμίσεις παραμέτρων για την ορθή εκτέλεση της εφαρμογής στο προαναφερόμενο λειτουργικό σύστημα.

```
.
├── app.py
├── blockchain.json
├── blockchain.py
├── data.py
├── main.py
├── static
│   ├── add_tx.css
│   ├── add_tx.js
│   ├── trace.css
│   ├── trace.js
│   ├── viewer.css
│   └── viewer.js
└── templates
    ├── add_tx.html
    ├── base.html
    ├── sf_success.html
    ├── trace.html
    └── viewer.html
```

3 directories, 16 files

Σχήμα 3.1.: Φάκελος τρέχουσας δομής εγκατάστασης

Στο [σχήμα 3.1.](#), παρουσιάζεται η λειτουργική δομή του τρέχοντος φακέλου εφαρμογής για την ορθή λειτουργία του Συστήματος Ιχνηλασιμότητας. Στη συνέχεια, δημιουργούμε ένα νέο εικονικό περιβάλλον το οποίο είναι αυτόνομο και απαρτίζεται από τις δικές του βιβλιοθήκες χωρίς να παρεμβάλλονται τα καθολικά εγκατεστημένα πακέτα

του κεντρικού συστήματος. Ενεργοποιούμε το νέο εικονικό περιβάλλον που μόλις δημιουργήσαμε και εγκαθιστούμε μέσα σε αυτό τη βιβλιοθήκη Flask.

```
p18prov@2018017 Pharma_SCM % python3 -m venv .venv
p18prov@2018017 Pharma_SCM % . .venv/bin/activate
(.venv) p18prov@2018017 Pharma_SCM % pip install Flask
Collecting Flask
  Downloading flask-3.0.3-py3-none-any.whl (101 kB)
    |████████████████████████████████████████| 101 kB 1.7 MB/s
Collecting Jinja2>=3.1.2
  Using cached Jinja2-3.1.3-py3-none-any.whl (133 kB)
Collecting click>=8.1.3
  Using cached click-8.1.7-py3-none-any.whl (97 kB)
Collecting itsdangerous>=2.1.2
  Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Collecting importlib-metadata>=3.6.0
  Downloading importlib_metadata-7.1.0-py3-none-any.whl (24 kB)
Collecting blinker>=1.6.2
  Downloading blinker-1.8.0-py3-none-any.whl (9.3 kB)
Collecting Werkzeug>=3.0.0
  Downloading werkzeug-3.0.2-py3-none-any.whl (226 kB)
    |████████████████████████████████████████| 226 kB 6.6 MB/s
Collecting zipp>=0.5
  Downloading zipp-3.18.1-py3-none-any.whl (8.2 kB)
Collecting MarkupSafe>=2.0
  Using cached MarkupSafe-2.1.5-cp39-cp39-macosx_10_9_universal2.whl (18 kB)
Installing collected packages: zipp, MarkupSafe, Werkzeug, Jinja2, itsdangerous,
importlib-metadata, click, blinker, Flask
Successfully installed Flask-3.0.3 Jinja2-3.1.3 MarkupSafe-2.1.5 Werkzeug-3.0.2
blinker-1.8.0 click-8.1.7 importlib-metadata-7.1.0 itsdangerous-2.2.0 zipp-3.18.1
```

Τέλος, ανάλογα με τις ανάγκες εκκίνησης του Συστήματος Ιχνηλασιμότητας, εκτελούμε είτε τη διεπαφή γραμμής εντολών (CLI, 'main.py'), είτε τη διαδικτυακή εφαρμογή (Web App, 'app.py') χρησιμοποιώντας τις παρακάτω εντολές.

```
(.venv) p18prov@2018017 Pharma_SCM % python3 main.py
Traceability CLI:
1) Print Blockchain
2) Trace Transaction(s)
5) Add Transaction
> Input:
(.venv) p18prov@2018017 Pharma_SCM % python3 app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production
deployment. Use a production WSGI server instead.
* Running on http://localhost:6969
```

Κεφάλαιο 4° - Παρουσίαση Συστήματος Ιχνηλασιμότητας

Στο επίπεδο ανάπτυξης λογισμικού, η διαδικτυακή εφαρμογή βοηθά στη βελτιστοποίηση της διαχείρισης της εφοδιαστικής αλυσίδας για τα φαρμακευτικά προϊόντα. Αυτό επιτυγχάνεται με την οπτικοποίηση των πολύπλοκων χαρακτηριστικών της νεότερης τεχνολογίας Blockchain. Κάθε λειτουργία έχει τη δική της αυτόνομη σελίδα μέσα στην εφαρμογή. Ωστόσο, όλες οι σελίδες συνδέονται μεταξύ τους με μια μπάρα πλοήγησης (navigation bar) ώστε να εναλλάσσονται οι χρήστες ευκολότερα μεταξύ των διαφορετικών λειτουργιών. Με αυτόν τον τρόπο, ενισχύεται η διαλειτουργικότητα της εφαρμογής και καθίσταται η χρήση της πιο ομαλή και ευέλικτη.

4.1. Αρχική Σελίδα – Περιηγητής Blockchain

[Blockchain](#) [Add Transaction](#) [Trace](#)

Block #0

Previous Hash	
Timestamp	20-04-2024 20:46:07:780837
Hash	5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9

Show Transactions

Block #1

Previous Hash	5feceb66ffc86f38d952786c6d696c79c2dbc239dd4e91b46729d73a27fb57e9
Timestamp	20-04-2024 20:46:07:781194
Hash	c81ac78d374d053aaeb066fda735678116758bf1d01e15eb3d26e8f6dab75ed9

Show Transactions

Εικόνα 4.1.: Η αρχική σελίδα της διαδικτυακής εφαρμογής

Στην [εικόνα 4.1.](#), παρουσιάζεται η λειτουργία του περιγητή της τεχνολογίας Blockchain. Σε αυτή τη σελίδα, προβάλλεται όλο το περιεχόμενο της εφοδιαστικής αλυσίδας των φαρμακευτικών σκευασμάτων. Εμπεριέχονται πληροφορίες όπως ο δείκτης (index) του κάθε μπλοκ, το αναγνωριστικό κατακερματισμού του προηγούμενου (previous hash) μπλοκ, ο κατακερματισμός του τρέχοντος μπλοκ, η χρονική δημιουργία του και μια λίστα με τις συναλλαγές του.

Αρχικά, οι συναλλαγές του κάθε μπλοκ αποκρύπτονται από τους χρήστες. Εξαιτίας αυτής της συνθήκης, παρέχεται στο χρήστη η δυνατότητα γρήγορης εναλλαγής μεταξύ των μπλοκ. Κατά την εύρεση ενός επιλεγμένου μπλοκ, ο χρήστης δύναται να προβάλει τις λεπτομέρειες των συναλλαγών χρησιμοποιώντας το πράσινο κουμπί με την ένδειξη 'Show Transactions'.

Block #17

Previous Hash	37087e7f15beebc9ee6ac224fe6fb2d324e20436fc1ef0a1515ea69826b7a2f5
Timestamp	20-04-2024 21:25:38:369830
Hash	ab4ef5d19906847ea306a03cee7a1886358c9801b53d9cc1e4bd436b87b236fa

Show Transactions

Transaction #1

Type	RegulatoryCompliance
InitID	FDA
Prod. Name	Vitamin D Tablets
Batch No.	LOT036
Regulatory	FDA Approved
Compliance	Doc55, Doc86
TXN_HASH	61b1ac9937151ebd81822c8d278898820c6a850e33b4b1c12fd7339b0de2cc0a

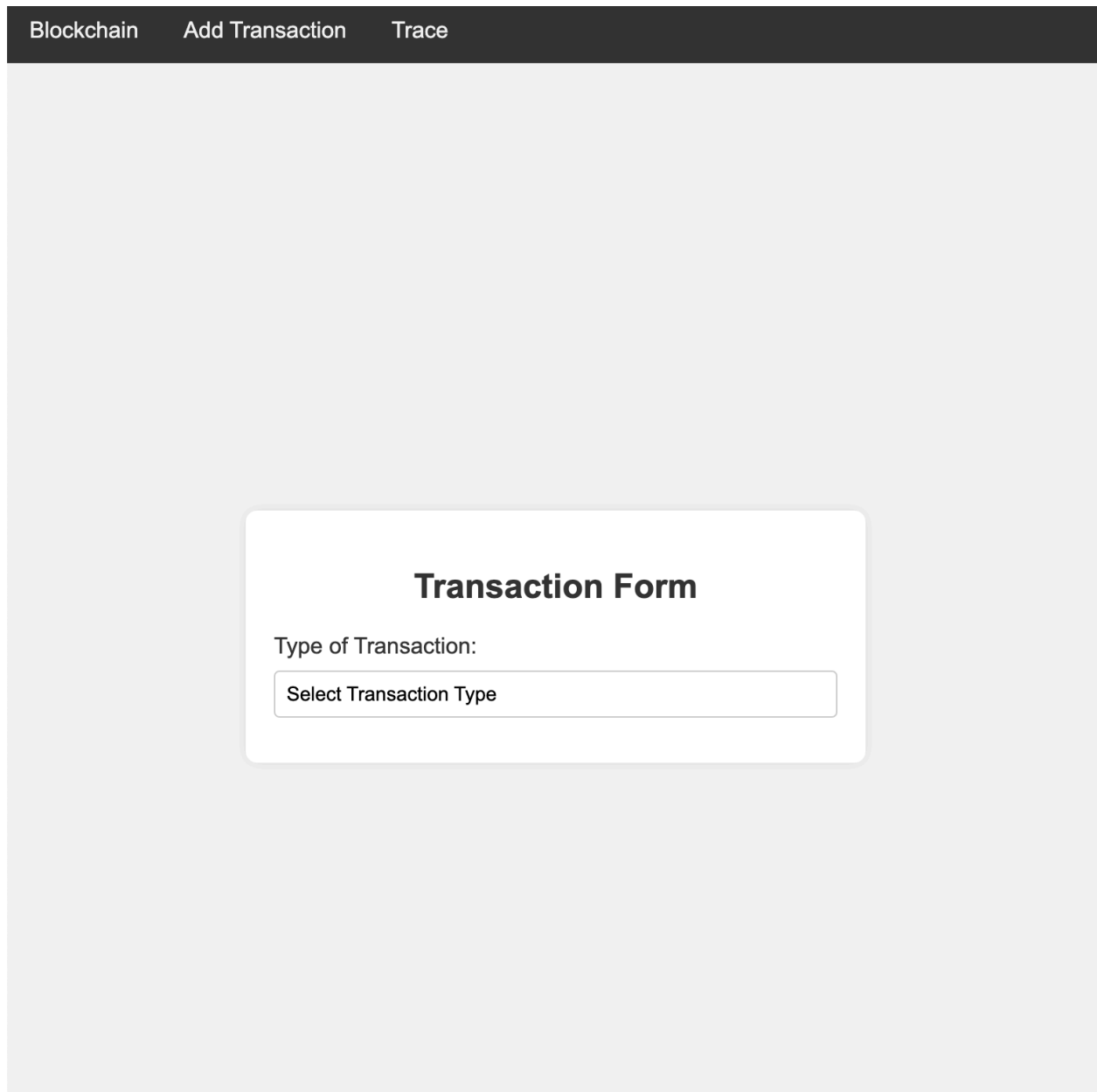
Transaction #2

Type	RegulatoryCompliance
InitID	FDA
Prod. Name	Vitamin B1 Tablets
Batch No.	LOT047
Regulatory	FDA Approved
Compliance	Doc66, Doc47
TXN_HASH	b3ffceb7768870477d02945be20cc1023c0b1a4083daf4b5ce82c25fe1f0fc4d

Εικόνα 4.2.: Οι συναλλαγές του 17^{ου} μπλοκ της εφοδιαστικής αλυσίδας

Στην [εικόνα 4.2.](#), το 17^ο μπλοκ της εφοδιαστικής αλυσίδας επεκτείνεται για την προβολή όλων των συναλλαγών του. Συνολικά, προσαρτώνται δύο συναλλαγές στο μπλοκ με αναγνωριστικά κατακερματισμού '2cc0a' και '0fc4d' αντίστοιχα.

4.2. Φόρμα Συναλλαγής (Προσθήκη)



Blockchain Add Transaction Trace

Transaction Form

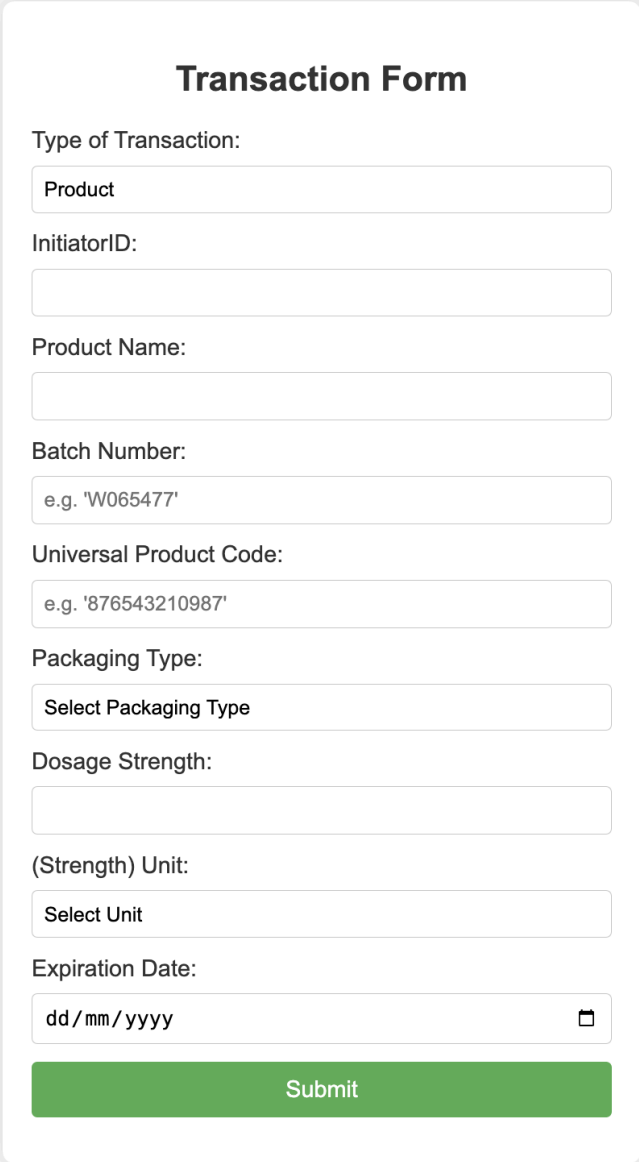
Type of Transaction:

Select Transaction Type

Εικόνα 4.3.: Η αρχική κατάσταση της φόρμας συναλλαγής

Στην [εικόνα 4.3.](#), παρουσιάζεται η λειτουργία της προσθήκης μιας νέας συναλλαγής στην εφοδιαστική αλυσίδα χρησιμοποιώντας μια φόρμα ως μέθοδο εισαγωγής.

Τα πεδία της φόρμας μεταβάλλονται ανάλογα με τον τύπο της συναλλαγής. Ο χρήστης δύναται να επιλέξει μεταξύ των ακόλουθων τύπων: ['Product', 'Manufacturing', 'Distribution', 'Regulatory'].



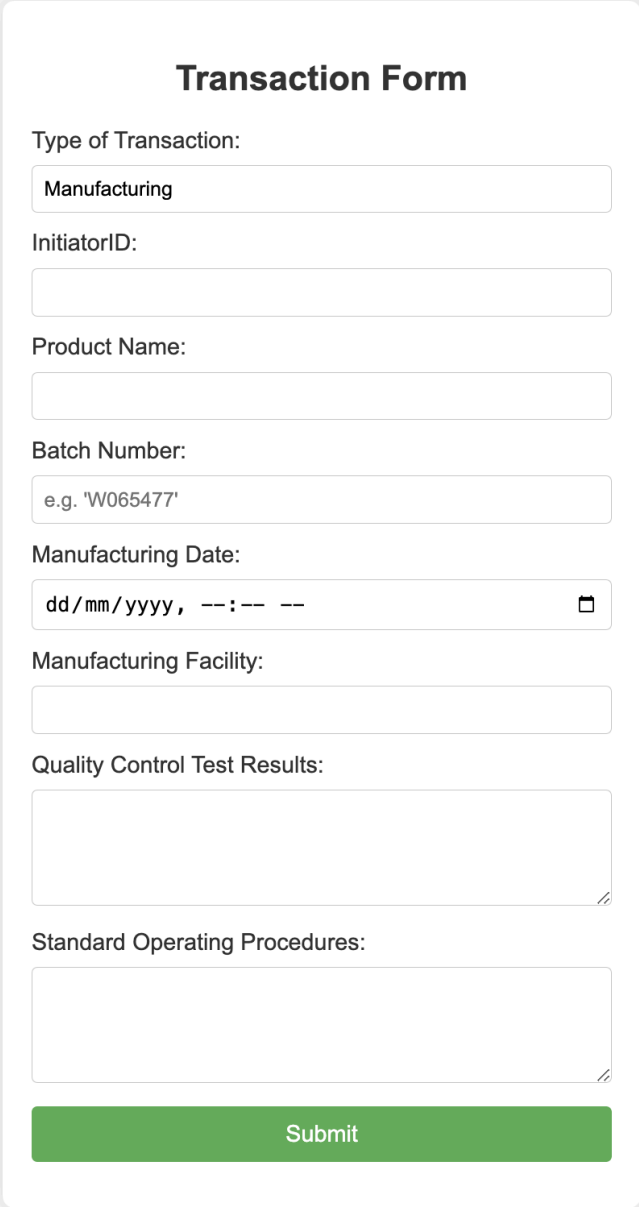
The image shows a 'Transaction Form' for the product stage in the supply chain. The form is titled 'Transaction Form' and contains several input fields. The 'Type of Transaction' field is set to 'Product'. The 'InitiatorID' field is empty. The 'Product Name' field is empty. The 'Batch Number' field has a placeholder 'e.g. 'W065477''. The 'Universal Product Code' field has a placeholder 'e.g. '876543210987''. The 'Packaging Type' field has a placeholder 'Select Packaging Type'. The 'Dosage Strength' field is empty. The '(Strength) Unit' field has a placeholder 'Select Unit'. The 'Expiration Date' field has a placeholder 'dd/mm/yyyy' and a calendar icon. A green 'Submit' button is at the bottom.

Field	Value
Type of Transaction	Product
InitiatorID	
Product Name	
Batch Number	e.g. 'W065477'
Universal Product Code	e.g. '876543210987'
Packaging Type	Select Packaging Type
Dosage Strength	
(Strength) Unit	Select Unit
Expiration Date	dd/mm/yyyy
Submit	Submit

Εικόνα 4.4.: Η φόρμα συναλλαγής για το στάδιο προϊόντος στην εφοδιαστική αλυσίδα

Στην [εικόνα 4.4.](#), παρουσιάζονται τα πεδία της φόρμας για ένα φαρμακευτικό σκεύασμα στο στάδιο προϊόντος στην εφοδιαστική αλυσίδα. Τα πεδία είναι προσαρμοσμένα ανάλογα με τον τύπο της μεταβλητής που δέχονται ως είσοδο. Το όνομα προϊόντος έχει για είσοδο ένα πεδίο εισαγωγής απλού κειμένου ενώ η ημερομηνία λήξης δέχεται ως είσοδο ένα αντικείμενο ημερολογίου. Αυτές οι επιλογές εξασφαλίζουν την εγκυρότητα

των εισαγόμενων πληροφοριών. Τέλος, όπου είναι εφικτό, παρέχονται επίσης και υποδείξεις στα πεδία εισαγωγής.



The image shows a 'Transaction Form' with the following fields and labels:

- Type of Transaction:** A dropdown menu with 'Manufacturing' selected.
- InitiatorID:** An empty text input field.
- Product Name:** An empty text input field.
- Batch Number:** A text input field containing the placeholder text 'e.g. 'W065477''.
- Manufacturing Date:** A date picker field showing the format 'dd/mm/yyyy, --:-- --' and a calendar icon.
- Manufacturing Facility:** An empty text input field.
- Quality Control Test Results:** A large empty text area with a small icon in the bottom right corner.
- Standard Operating Procedures:** A large empty text area with a small icon in the bottom right corner.
- Submit:** A green button at the bottom of the form.

Εικόνα 4.5.: Η φόρμα συναλλαγής για το στάδιο κατασκευής στην εφοδιαστική αλυσίδα

Στην [εικόνα 4.5.](#), παρουσιάζονται τα πεδία της φόρμας για ένα φαρμακευτικό σκεύασμα στο στάδιο κατασκευής στην εφοδιαστική αλυσίδα.

Transaction Form

Type of Transaction:

Distribution

InitiatorID (Expediter):

Product Name:

Batch Number:

e.g. 'W065477'

Distribution Date:

dd/mm/yyyy

Distribution Location:

Distribution Quantities:

Addressee:

Destination Address:

Submit

Εικόνα 4.6.: Η φόρμα συναλλαγής για το στάδιο διανομής στην εφοδιαστική αλυσίδα

Στην [εικόνα 4.6.](#), παρουσιάζονται τα πεδία της φόρμας για ένα φαρμακευτικό σκεύασμα στο στάδιο διανομής στην εφοδιαστική αλυσίδα. Σε αυτό το στάδιο, ο επικυρωτής (Initiator) μιας συναλλαγής είναι και ο διανομέας των φαρμάκων. Συγχρόνως, ο παραλήπτης αυτής της συναλλαγής δύναται να είναι και διανομέας σε μια περίπτωση μελλοντικής αναδιανομής της ποσότητας που εισέπραξε από τον αρχικό διανομέα. Στην [εικόνα 3.5](#) του ιστορικού διανομής παρουσιάζεται ακριβώς αυτή η περίπτωση του διπλού ρόλου παραλήπτη - αποστολέα στην εφοδιαστική αλυσίδα των φαρμάκων.

Transaction Form

Type of Transaction:

Regulatory

InitiatorID:

KOUROUTHANASSIS

Product Name:

Vitamin K2 Tablets

Batch Number:

LOT2024

Regulatory Approvals:

National Organization for Medicines (EOF)

Compliance Documents:

E0F-KOUR2024, E0F-CONSENSUS_2018

Submit

Εικόνα 4.7.: Η φόρμα συναλλαγής για το στάδιο κανονιστικής συμμόρφωσης στην εφοδιαστική αλυσίδα

Στην [εικόνα 4.7.](#), παρουσιάζονται ενδεικτικά τα συμπληρωμένα πεδία της φόρμας για ένα φαρμακευτικό σκεύασμα στο στάδιο κανονιστικής συμμόρφωσης στην εφοδιαστική αλυσίδα.

Transaction Form

Type of Transaction:

Regulatory

InitiatorID:

KOUROUTHANASSIS

Product Name:

Vitamin K2 Tablets

Batch Number:

e.g. 'W065477'

Regulatory App

Please fill in this field.

National Organization for Medicines (EOF)

Compliance Documents:

E0F-KOUR2024, E0F-CONSENSUS_2018

Submit

Εικόνα 4.8.: Αδυναμία υποβολής της φόρμας λόγω ελλιπούς συμπλήρωσης

Στην [εικόνα 4.8.](#), πραγματοποιείται ο έλεγχος εγκυρότητας των πληροφοριών μιας συναλλαγής και παρουσιάζεται σφάλμα υποβολής εξαιτίας της ελλιπούς συμπλήρωσης του αριθμού παρτίδας του φαρμακευτικού σκευάσματος. Επισημαίνεται, πως το κάθε πεδίο μιας συναλλαγής διέπεται από δικούς του ξεχωριστούς κανόνες, ανάλογα με το είδος της πληροφορίας που περιγράφεται.

Block Number #17

Transaction Details:

InitiatorID	KOUROUTHANASSIS
Product Name	Vitamin K2 Tablets
Batch Number	LOT2024
Regulatory Approvals	National Organization for Medicines (EOF)
Compliance Documents	EOF-KOUR2024, EOF-CONSENSUS_2018

Transaction Hash: 7847e36aba517733f09188a9dad5abb46a71acc8464e8e2d80cf9368ea105f90

Εικόνα 4.9.: Επισκόπηση επιτυχούς προσθήκης μιας συναλλαγής στην εφοδιαστική αλυσίδα

Στην [εικόνα 4.9.](#), παρουσιάζεται η σελίδα ανασκόπησης μιας νέας συναλλαγής ύστερα από την επιτυχημένη υποβολή της στην εφοδιαστική αλυσίδα, συνοψίζοντας έτσι τις πληροφορίες που μόλις προστέθηκαν.

4.3. Ιχνηλάτηση Συναλλαγής (Φαρμάκου)

Blockchain Add Transaction Trace

Medicine Info

Product Name:

Batch Number:

Search

Εικόνα 4.10.: Η αρχική κατάσταση της ιχνηλάτησης μιας συναλλαγής

Στην [εικόνα 4.10.](#), παρουσιάζεται η λειτουργία της ιχνηλάτησης των συναλλαγών στην εφοδιαστική αλυσίδα για ένα φάρμακο, χρησιμοποιώντας ως πρωτεύοντα κλειδιά το όνομα του σκευάσματος ('product_name') και τον αριθμό παρτίδας ('batch_number') του.

Medicine Info

Product Name:

Vitamin K2 Tablets

Batch Number:

LOT2023

Search

There are no results matching the specified combination. Ensure that the information provided is accurate.

Εικόνα 4.11.: Μήνυμα λάθους στην περίπτωση ανέφικτης ιχνηλάτησης

Στην [εικόνα 4.11.](#), πραγματοποιείται ο έλεγχος εγκυρότητας των πληροφοριών κατά την διάρκεια ιχνηλάτησης ενός ενδεικτικού φαρμακευτικού σκευάσματος και παρουσιάζεται σφάλμα ιχνηλάτησης, καθώς δεν υπάρχουν συναλλαγές στην εφοδιαστική αλυσίδα που να ταυτίζονται με τα παρεχόμενα κριτήρια ιχνηλάτησης. Σύμφωνα με το παράδειγμα 4.7., που αναπτύχθηκε προηγουμένως, υπάρχουν συναλλαγές στην αλυσίδα που να ταυτίζονται με το παρεχόμενο όνομα σκευάσματος. Ωστόσο, το σφάλμα της τρέχουσας ιχνηλάτησης, έγκειται στον παρεχόμενο αριθμό παρτίδας 'LOT2023' αντί για 'LOT2024' του αρχικού παραδείγματος.

Medicine Info

Product Name:

Vitamin K2 Tablets

Batch Number:

LOT2024

Search

TXN #1

Type Product
InitID KOUROUTHANASSIS
Prod. Name Vitamin K2 Tablets
Batch No. LOT2024
UPC 123456789012
Packaging Blister Pack
Dosage 1 mg
Expiration 05/2026
TXN_HASH 8747de9a14b1532147b841d3899579f7d49d7d9514c3b39f5382c29d3984ee42

TXN #2

Type RegulatoryCompliance
InitID KOUROUTHANASSIS
Prod. Name Vitamin K2 Tablets
Batch No. LOT2024
Regulatory National Organization for Medicines (EOF)
Compliance EOF-KOUR2024, EOF-CONSENSUS_2018
TXN_HASH 7847e36aba517733f09188a9dad5abb46a71acc8464e8e2d80cf9368ea105f90

Εικόνα 4.12.: Αποτελέσματα ιχνηλάτησης ενός φαρμακευτικού σκευάσματος στην εφοδιαστική αλυσίδα

Στην [εικόνα 4.12.](#), παρουσιάζονται τα αποτελέσματα ιχνηλάτησης για τον συνδυασμό ονόματος σκευάσματος 'Vitamin K2 Tablets' και αριθμού παρτίδας 'LOT2024'. Σε αυτό το παράδειγμα, επιστρέφονται μόνο τα στάδια της κανονιστικής συμμόρφωσης και του προϊόντος στην εφοδιαστική αλυσίδα. Συνεπώς, αυτό σημαίνει πως τα υπολειπόμενα στάδια βρίσκονται υπό προσθήκη ή θα προστεθούν μελλοντικά, ανάλογα με το χρονοδιάγραμμα κατασκευής και διανομής τους.

Κεφάλαιο 5° - Συμπεράσματα

5.1. Συμπεράσματα

Η ανερχόμενη τεχνολογία «Blockchain» υπόσχεται στην πράξη τον ουσιαστικό μετασχηματισμό της φαρμακευτικής αλυσίδας εφοδιασμού, ενισχύοντας σημαντικά την ακεραιότητα και τη διαφάνεια των ιατρικών δεδομένων. Πράγματι, στην τελευταία φάση των εξελίξεων για τη φαρμακοβιομηχανία, αναφερόμενη και ως «Pharma 4.0» όπου η εστίαση επικεντρώνεται στην αφομοίωση ψηφιακών τεχνολογιών για τις διαδικασίες της παραγωγής και της διανομής των φαρμάκων, η τεχνολογία «Blockchain» αναδεικνύεται ως μια κομβική καινοτομία. Με την αξιοποίηση της ιδιότητας της αμεταβλητότητας του λογιστικού της βιβλίου (βάση δεδομένων) αλλά και της αποκεντρωμένης φύσης αυτής της τεχνολογίας, διασφαλίζεται πως κάθε συναλλαγή εντός της εφοδιαστικής αλυσίδας καταγράφεται με απόλυτη ακρίβεια, δίχως τη δυνατότητα παραβίασης, αλλοίωσης ή περαιτέρω παραμετροποίησης. Συνεπώς, με την εγγενή ιδιότητα της αμεταβλητότητας, ενισχύεται η ακεραιότητα των ιατρικών δεδομένων, μιας κρίσιμης δηλαδή απαίτησης σε έναν τομέα όπου η αυθεντικότητα και η ιχνηλασιμότητα των φαρμακευτικών προϊόντων αποτελούν ζητήματα πρωταρχικής σημασίας.

Μια από τις πιο πιεστικές προκλήσεις στη βιομηχανία των φαρμάκων πρόκειται για την καταπολέμηση των πλαστών φαρμακευτικών σκευασμάτων, στα οποία ενέχονται σοβαροί κίνδυνοι για την ασφάλεια της δημόσιας υγείας. Εξαιτίας της ικανότητας της προκείμενης τεχνολογίας στο να παρέχεται μια διαφανή και ανιχνεύσιμη διαδρομή του κάθε φαρμάκου στην αλυσίδα εφοδιασμού, από την προμήθεια των πρώτων υλών, την παραγωγή, έως τη διανομή και τη λιανική πώληση των φαρμακευτικών σκευασμάτων, προσφέρεται έτσι μια αξιόπιστη λύση στο πρόβλημα των πλαστών φαρμάκων. Έπειτα, με τη συμπερίληψη όλων των ενδιαφερόμενων φορέων σε μια ενιαία διαπλατφορμική υποδομή, ενισχύονται τόσο τα επίπεδα εμπιστοσύνης μεταξύ των φορέων, όσο και η λογοδοσία των ενεργειών τους. Με αυτή η διαφάνεια στις διαδικασίες της τεχνολογίας, αποτρέπονται τυχόν αποκλίσεις, καθώς οι μη εξουσιοδοτημένες τροποποιήσεις στην αλυσίδα εφοδιασμού ανιχνεύονται και αναστρέφονται εγκαίρως, ώστε να μειώνονται σημαντικά οι πιθανότητες έλευσης πλαστών φαρμακευτικών σκευασμάτων στην αγορά.

Η ενσωμάτωση της τεχνολογίας «Blockchain» βρίσκεται σε πλήρη ευθυγράμμιση με τις αρχές του «Pharma 4.0» όπου υποστηρίζονται οι «έξυπνες» τεχνολογίες για τη βελτίωση της ποιότητας, της αποτελεσματικότητας και της κανονιστικής συμμόρφωσης στη φαρμακοβιομηχανία. Κατά συνέπεια, με την ορθή χρήση της προαναφερόμενης τεχνολογίας, δύνανται να εξορθολογίζονται οι εργασίες ροής, να μειώνονται τα διοικητικά κόστη των οργανισμών όπως επίσης να ενισχύεται η ευρύτερη ανταποκρισιμότητά τους στην αλυσίδα εφοδιασμού. Οι αναφερόμενες βελτιστοποιήσεις πραγματώνονται μέσω αυτοματισμών όπως τα «έξυπνα» συμβόλαια, για τη διασφάλιση της έγκαιρης και της

ακριβούς εκτέλεσης του συνόλου των λειτουργιών της φαρμακευτικής εφοδιαστικής αλυσίδας.

Ανεξάρτητα με τις πιθανές θετικές επιδράσεις της τεχνολογίας «Blockchain» στην αλυσίδα εφοδιασμού των φαρμάκων, παρουσιάζονται προκλήσεις στην ουσιαστική υλοποίησή της από τους αρμόδιους οργανισμούς. Ως αναδυόμενη τεχνολογία, τίθενται αρκετά τεχνικά, ρυθμιστικά και οργανωτικά εμπόδια προς αντιμετώπιση. Ζητήματα όπως η επεκτασιμότητα, η διαλειτουργικότητα με τα υπάρχοντα συστήματα, το απόρρητο των ιατρικών δεδομένων και το αυξημένο κόστος της αρχικής επένδυσης, πρόκειται για καταστάσεις τροχοπέδια στην ευρεία ενσωμάτωσής της από το σύνολο της φαρμακοβιομηχανίας. Έπειτα, η βιομηχανία αυτή υπόκειται σε αυστηρές κανονιστικές ρυθμίσεις ώστε να διασφαλίζεται η συμμόρφωσης με τους διάφορους εθνικούς και διεθνούς οργανισμούς, με αποτέλεσμα να αυξάνεται η πολυπλοκότητα ενσωμάτωσης της αναδυόμενης αυτής τεχνολογίας.

Πέρα από τις προκλήσεις της νέας τεχνολογίας, τα οφέλη της στη φαρμακευτική αλυσίδα εφοδιασμού είναι ποιοτικά αναρίθμητα. Η «ωρίμανση» της τεχνολογίας στις μελλοντικές αναθεωρήσεις της και η εξοικείωση των ενδιαφερόμενων φορέων με αυτή, είναι αναγκαίοι και ικανοί παράγοντες για την αντιμετώπιση των αρχικών δυσκολιών στην κατεύθυνση της ολικής ενσωμάτωσής της από τη βιομηχανία φαρμάκων. Κατά συνέπεια, ο συνεχής πειραματισμός με την προκείμενη τεχνολογία πρόκειται για ένα πολλά υποσχόμενο βήμα προς τα εμπρός στη συνεχιζόμενη εξέλιξη της φαρμακευτικής αλυσίδας εφοδιασμού στην ψηφιακή εποχή, καθώς ο εστιασμός του κλάδου τη σήμερον ημέρα βρίσκεται κυρίως στην ασφάλεια, τη διαφάνεια και την αποτελεσματικότητα της παροχής των φαρμακευτικών σκευασμάτων.

5.2. Μελλοντικές Προεκτάσεις (Συστήματος Ιχνηλασιμότητας)

Στα πλαίσια πειραματισμού με το υπόδειγμα της τεχνολογίας Blockchain αναπτύχθηκε μια τοπική εφαρμογή εκτός ροής για την προσομοίωση μιας φαρμακευτικής αλυσίδας εφοδιασμού αξιοποιώντας τα βασικά χαρακτηριστικά και τις ιδιότητες της τεχνολογίας Blockchain. Στο αποκαλούμενο Σύστημα Ιχνηλασιμότητας, πραγματοποιούνται πιλοτικά συγκεκριμένες λειτουργίες όπως η προβολή όλων των συναλλαγών ταξινομημένες ανά μπλοκ, η προσθήκη νέας συναλλαγής για το κάθε στάδιο στην φαρμακευτική αλυσίδα εφοδιασμού (PSC) και η εύρεση μιας συναλλαγής με βάση το συνδυασμό του ονόματος προϊόντος και του αριθμού παρτίδας. Για τους σκοπούς της υλοποίησης μιας εφαρμογής όπου προσομοιώνεται η λειτουργικότητα της εφοδιαστικής αλυσίδας, πραγματοποιήθηκαν συγκεκριμένες παραδοχές στην υλοποίηση της τεχνολογίας Blockchain, οι οποίες επιδέχονται βελτίωσης για μια πιθανή εμπορική εκμετάλλευση της εφαρμογής στο μέλλον.

5.2.1. Υλοποίηση Κρυπτογραφημένης Βάσης Δεδομένων

Στην τρέχουσα υλοποίηση του Συστήματος Ιχνηλασιμότητας, χρησιμοποιείται σε μορφή απλού κειμένου το μορφότυπο JSON για την αποθήκευση των φαρμακευτικών πληροφοριών. Αυτή η προσέγγιση, αν και είναι λειτουργική για την προσομοίωση της εφοδιαστικής αλυσίδας, είναι προβληματική για υλοποιήσεις μεγαλύτερης κλίμακας. Τα αρχεία απλού κειμένου στο μορφότυπο JSON δεν είναι κρυπτογραφημένα, καθιστώντας τα εξαιρετικά ευάλωτα σε μη-εξουσιοδοτημένη πρόσβαση. Έτσι, εκτίθενται ευαίσθητες πληροφορίες σχετικά με τα φάρμακα, τους προμηθευτές και τους ασθενείς, θέτοντας σε κίνδυνο συνολικά την ακεραιότητα της εφοδιαστικής αλυσίδας. Επίσης, παραβιάζονται οι αρχές της ίδιας της τεχνολογίας Blockchain για τη διασφάλιση της ακεραιότητας και της αμεταβλητότητας των πληροφοριών, απαξιώνοντας με αυτόν τον τρόπο το νόημα της αξιοποίησης του υποδείγματος για την ενισχυμένη ασφάλεια των πληροφοριών.

Για την αντιμετώπιση των ζητημάτων της ιδιωτικότητας και της ακεραιότητας των πληροφοριών του Συστήματος Ιχνηλασιμότητας, απαιτείται η μετάβαση της υπάρχουσας μεθόδου αποθήκευσης των πληροφοριών σε μια πραγματική βάση δεδομένων. Με αυτή την ενέργεια, αυξάνεται η απόδοση της αλυσίδας εφοδιασμού καθώς ένα ολοκληρωμένο σύστημα διαχείρισης βάσης δεδομένων διαχειρίζεται καλύτερα τις σύνθετες αναζητήσεις (queries) ή τα μεγάλα σύνολα δεδομένων. Επίσης, στα ολοκληρωμένα συστήματα αυτά, παρέχεται σημαντικές δυνατότητες όπως είναι μεταξύ άλλων η ανάκαμψη της τρέχουσας κατάστασης σε περίπτωση σφάλματος καταγραφής νέας συναλλαγής, η ευρετηρίαση τεκμηρίων και η αποτροπή της επικάλυψης δεδομένων.

Προτείνονται οι ολοκληρωμένες λύσεις των λογισμικών MongoDB ή SQLAlchemy με κριτήριο τη συμβατότητά τους με την υπάρχουσα υλοποίηση της βιβλιοθήκης Flask στο προγραμματιστικό επίπεδο.

- MongoDB: <https://www.mongodb.com/>
- SQLAlchemy: <https://www.sqlalchemy.org/>

5.2.2. Αυθεντικοποίηση και Επίπεδα Πρόσβασης Χρηστών

Στην τρέχουσα υλοποίηση του Συστήματος Ιχνηλασιμότητας τα δικαιώματα αλλά και τα επίπεδα πρόσβασης των εμπλεκόμενων φορέων δεν είναι σαφώς καθορισμένα. Αυτό έχει ως αποτέλεσμα να παρέχεται πρόσβαση στη γραφική διεπαφή του δικτύου Blockchain από όλους τους χειριστές της εφαρμογής. Η αυθεντικοποίηση κι ο έλεγχος πρόσβασης στο σύστημα είναι σημαντικές για τη διασφάλιση της εξουσιοδοτημένης συμμετοχής των οντοτήτων στις ενέργειες του δικτύου Blockchain.

Με την αποδοτική αυθεντικοποίηση και τον έλεγχο πρόσβασης σε μια εφαρμογή Blockchain για τη διαχείριση της φαρμακευτικής εφοδιαστικής αλυσίδας, επαληθεύεται η ταυτότητα των αρμόδιων συμμετεχόντων και καθορίζεται η προσβασιμότητα και ο ρόλος

τους στις λειτουργίες της εφαρμογής. Έτσι, διασφαλίζεται η ασφάλεια, η ακεραιότητα και η ιχνηλασιμότητα εντός της αλυσίδας εφοδιασμού, αντιστοιχίζοντας τις δυνατές ενέργειες των οντοτήτων ανάλογα με την καταλληλότητα του ρόλου τους, προστατεύοντας έτσι το σύστημα από τη μη-εξουσιοδοτημένη πρόσβαση.

5.2.3. «Έξυπνα» Συμβόλαια και Αυτοματισμός

Στην τρέχουσα υλοποίηση του Συστήματος Ιχνηλασιμότητας, όλες οι διαδικασίες του δικτύου εκτελούνται χειροκίνητα. Όμως, οι χειρωνακτικές ενέργειες είναι συχνά πιο αργές εξαιτίας της ανάγκης για ανθρώπινη παρέμβαση, προκαλώντας συμφόρηση, ιδίως σε φαρμακευτικές εφοδιαστικές αλυσίδες μεγάλης κλίμακας. Αντίθετα, με τα «έξυπνα» συμβόλαια, παρέχεται ένα θεμελιωμένο πλαίσιο για την αυτοματοποίηση και την επαλήθευση των συναλλαγών εντός της φαρμακευτικής αλυσίδας εφοδιασμού.

Με τα «έξυπνα» συμβόλαια παρέχονται αυτοματισμοί ενισχύοντας την απόδοση της εφοδιαστικής αλυσίδας. Στο παράδειγμα του Συστήματος Ιχνηλασιμότητας, με ένα συμβόλαιο ελέγχεται ένα προκαθορισμένο όριο αποθέματος ενός φαρμακείου. Εάν το απόθεμα του φαρμακείου είναι μικρότερο από το προκαθορισμένο όριο, ενεργοποιείται αυτόματα μια παραγγελία στον προμηθευτή του φαρμακευτικού σκευάσματος για την αναπλήρωση του αποθέματος. Σε μια άλλη περίπτωση, με ένα διαφορετικό συμβόλαιο ελέγχεται η υποβολή των νέων συναλλαγών για την αποφυγή διπλότυπων δεδομένων. Κατά συνέπεια, τα «έξυπνα» συμβόλαια είναι ιδανικά σε σενάρια όπου οι διαδικασίες της αλυσίδας εφοδιασμού είναι επαναλαμβανόμενες και σαφώς καθορισμένες.

5.2.4. Βελτιστοποίηση της Γραφικής Διεπαφής

Στην τρέχουσα υλοποίηση του Συστήματος Ιχνηλασιμότητας, μέσω της γραφικής διεπαφής, υποστηρίζονται οι δυνατότητες της προβολής όλων των συναλλαγών των μπλοκ της αλυσίδας, της προσθήκης νέων συναλλαγών αλλά και της ιχνηλάτησης ενός φαρμάκου. Ο υπάρχων σχεδιασμός της διεπαφής είναι περιορισμένης λειτουργικότητας, παρέχοντας μόνο τη βασικές λειτουργίες στους χειριστές της εφαρμογής. Με την οπτική αναβάθμιση του γραφικού περιβάλλοντος της εφαρμογής, θα παρέχεται στους χειριστές ένα δυναμικό (σε αντίθεση με το υπάρχον στατικό) περιβάλλον με εμπλουτισμένες νέες λειτουργίες όπως η προσθήκη της [λειτουργίας ιχνηλάτησης του ιστορικού συναλλαγών του διανομέα από τη διεπαφή γραμμής εντολών](#), η δυνατότητα προτεραιοποίησης των σημαντικότερων συναλλαγών όπως επίσης και η ενημέρωση σε πραγματικό χρόνο των τρεχόντων επιπέδων αποθέματος του κάθε διανομέα.

Με την αναδιαμόρφωση της γραφικού περιβάλλοντος της εφαρμογής, δύναται να ενσωματώνονται και εργαλεία οπτικοποίησης δεδομένων όπως το D3.js ή το Chart.js για την παρουσίαση της πορείας των φαρμακευτικών σκευασμάτων εντός της εφοδιαστικής αλυσίδας με πιο διαισθητικό και ενημερωτικό τρόπο, παρέχοντας ακόμη και στατιστικά

στοιχεία σε πραγματικό χρόνο. Παράλληλα, με την παρουσία πινάκων παρουσιάσεων και λεπτομερών προβολών των συναλλαγών, επιτρέπεται στους χειριστές ο ευέλικτος έλεγχος της τρέχουσας κατάστασης των φαρμακευτικών σκευασμάτων και η προβολή του ιστορικού τους εντός της φαρμακευτικής αλυσίδας εφοδιασμού.

Βιβλιογραφία

1. Mentzer JT, DeWitt W, Keebler JS, Min S, Nix NW, Smith CD, Zacharia ZG. Defining Supply Chain Management. *Journal of Business Logistics*. 2001 Sep;22(2):1-25.
2. Christopher M. *Logistics and Supply Chain Management: Strategies for Reducing Costs and Improving Services*. 1998.
3. Cooper MC, Ellram LM. Characteristics of Supply Chain Management and the Implications for Purchasing and Logistics Strategy. *The International Journal of Logistics Management*. 1993 Jul 1;4(2):13-24.
4. Jain J, Dangayach GS, Agarwal G. Soumya Banerjee (2010). Supply Chain Management: Literature Review and Some Issues, *Journal of Studies on Manufacturing*;1:11-25.
5. Moise M. The Importance of Reverse Logistics for Retail Activity. *The Amfiteatru Economic Journal*. 2008;10(24):192-209.
6. Moe T. Perspectives on Traceability in Food Manufacture. *Trends in Food Science & Technology*. 1998 May 1;9(5):211-4.
7. GS1 Global Traceability Standard. (2012, November). GS1. Retrieved May 19, 2024, from https://www.gs1.org/sites/default/files/docs/traceability/Global_Traceability_Standard.pdf
8. Schwegele F, Andree S. Tracking and Tracing in the Meat Area. *Scientific journal" Meat Technology"*. 2009 Jun 25;50(1-2):11-20.
9. New S. The Transparent Supply Chain. *Harvard Business Review*. 2010 Oct 1;88(10).
10. Parmigiani A, Klassen RD, Russo MV. Efficiency Meets Accountability: Performance Implications of Supply Chain Configuration, Control, and Capabilities. *Journal of Operations Management*. 2011 Mar 1;29(3):212-23.
11. Schwägele F. Traceability from a European Perspective. *Meat Science*. 2005 Sep 1;71(1):164-73.
12. Mendoza RL. Continuity and Change in the Drug Supply Chain: Actors, Actions, and Aversions. *Journal of Medical Economics*. 2021 Jan 1;24(1):689-97.
13. Shah N. Pharmaceutical Supply Chains: Key Issues and Strategies for Optimisation. *Computers & Chemical Engineering*. 2004 Jun 15;28(6-7):929-41.

14. Wong WP, Saw PS, Jomthanachai S, Wang LS, Ong HF, Lim CP. Digitalization Enhancement in the Pharmaceutical Supply Network using a Supply Chain Risk Management Approach. *Scientific Reports*. 2023 Dec 15;13(1):22287.
15. Koh R, Schuster EW, Chackrabarti I, Bellman A. Securing the Pharmaceutical Supply Chain. White Paper, Auto-ID Labs, Massachusetts Institute of Technology. 2003 Sep;1:19.
16. Saha E, Rathore P, Parida R, Rana NP. The Interplay of Emerging Technologies in Pharmaceutical Supply Chain Performance: An Empirical Investigation for the Rise of Pharma 4.0. *Technological Forecasting and Social Change*. 2022 Aug 1;181:121768.
17. Hariry RE, Barenji RV, Paradkar A. From Industry 4.0 to Pharma 4.0. *Handbook of Smart Materials, Technologies, and Devices: Applications of Industry 4.0*. 2020:1-22.
18. Tabatabaei MH, Vitenberg R, Veeraragavan NR. Understanding Blockchain: Definitions, Architecture, Design, and System Comparison. *Computer Science Review*. 2023 Nov 1;50:100575.
19. Lemieux VL. Trusting Records: Is Blockchain Technology The Answer?. *Records Management Journal*. 2016 Jul 18;26(2):110-39.
20. Ghiro L, Restuccia F, D'Oro S, Basagni S, Melodia T, Maccari L, Cigno RL. A Blockchain Definition to Clarify its Role for the Internet of Things. In 2021 19th Mediterranean Communication and Computer Networking Conference (MedComNet) 2021 (pp. 1-8). IEEE.
21. Azevedo P, Gomes J, Romão M. Supply Chain Traceability Using Blockchain. *Operations Management Research*. 2023 Sep;16(3):1359-81.
22. Liu M, Robin A, Wu K, Xu J. Blockchain's Impact on Accounting and Auditing: A Use Case on Supply Chain Traceability. *Journal of Emerging Technologies in Accounting*. 2022 Sep 1;19(2):105-19.
23. Le TV, Hsu CL. A Systematic Literature Review of Blockchain Technology: Security Properties, Applications and Challenges. *Journal of Internet Technology*. 2021 Jul 1;22(4):789-802.
24. Singh R, Dwivedi AD, Srivastava G. Internet of Things based Blockchain for Temperature Monitoring and Counterfeit Pharmaceutical Prevention. *Sensors*. 2020 Jul 16;20(14):3951.
25. Kostyuchenko Y, Jiang Q. Blockchain Applications to Combat the Global Trade of Falsified Drugs. In 2020 International Conference on Data Mining Workshops (ICDMW) 2020 Nov 17 (pp. 890-894). IEEE.

26. Aslam M, Jabbar S, Abbas Q, Albathan M, Hussain A, Raza U. Leveraging Ethereum Platform for Development of Efficient Tractability System in Pharmaceutical Supply Chain. *Systems*. 2023 Apr 17;11(4):202.
27. Tapscott D, Tapscott A. *Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World*. Penguin; 2016 May 10.
28. Statista. Blockchain Adoption Rate in Health Apps Worldwide 2017 [Internet]. Statista; 2017 [cited 2024 Jun 3]. Available from: <https://www.statista.com/statistics/759208/healthcare-blockchain-adoption-rate-in-health-apps-worldwide/>