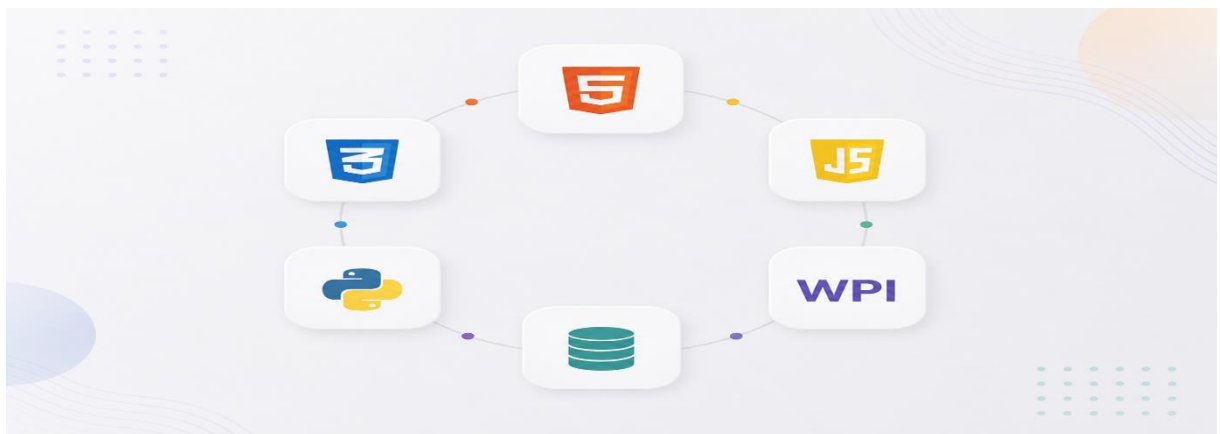


**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ**  
**ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

«Ανάπτυξη Εργαλείων Λογισμικού &  
Διαδικτυακού Συστήματος για Μεταφορά,  
Ανάλυση και Ανάκτηση Μεγάλου Όγκου  
Δεδομένων Πατεντών σε Βάση Δεδομένων»



Του φοιτητή  
Νουρεντίνι Γιούργκεν  
Αρ. Μητρώου: 144192

Επιβλέπων  
Δρ Μιχαήλ Σαλαμπάσης  
Βαθμίδα Καθηγητής

Ημερομηνία 12/5/2026

Τίτλος Δ.Ε. Ανάπτυξη Εργαλείων Λογισμικού & Διαδικτυακού Συστήματος για Μεταφορά,  
Ανάλυση και Ανάκτηση Μεγάλου Όγκου Δεδομένων Πατεντών σε Βάση Δεδομένων  
Κωδικός Δ.Ε. 25129

Ονοματεπώνυμο φοιτητή Νουρεντίνι Γιούργκεν  
Ονοματεπώνυμο εισηγητή Σαλαμπάσης Μιχάλης  
Ημερομηνία ανάληψης Δ.Ε. 24-02-2025  
Ημερομηνία περάτωσης Δ.Ε. 24-02-2027

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Νουρεντίνι Γιούργκεν που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

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

**«Σκέψου πέρα από αυτό που είσαι. Μην σταματάς να αγωνίζεσαι»**



## Πρόλογος

Η ραγδαία εξέλιξη της τεχνολογίας και η συνεχής παραγωγή επιστημονικής και τεχνικής γνώσης έχουν οδηγήσει σε εκθετική αύξηση του όγκου των διαθέσιμων δεδομένων. Ιδιαίτερα στον τομέα των διπλωμάτων ευρεσιτεχνίας, η συγκέντρωση, οργάνωση και αξιοποίηση της πληροφορίας αποτελούν κρίσιμα ζητήματα για την υποστήριξη της έρευνας, της καινοτομίας και της επιχειρηματικής στρατηγικής.

Τα δεδομένα πατεντών, όπως αυτά της συλλογής World Patent Information (WPI), διατίθενται συνήθως σε μορφή XML, ακολουθώντας διεθνή πρότυπα όπως το ST.36. Η μορφή αυτή επιτρέπει την αποθήκευση σύνθετων και ιεραρχικών πληροφοριών, ωστόσο δημιουργεί σημαντικές προκλήσεις όσον αφορά την αποδοτική επεξεργασία, αναζήτηση και ανάλυση μεγάλου όγκου δεδομένων.

Η ανάγκη μετασχηματισμού των ημιδομημένων XML δεδομένων σε δομημένες σχεσιακές βάσεις δεδομένων αποτελεί βασική προϋπόθεση για την αποτελεσματική αξιοποίησή τους. Παράλληλα, η ανάπτυξη διαδικτυακών εφαρμογών που επιτρέπουν την εύκολη διαχείριση και ανάκτηση της πληροφορίας καθίσταται απαραίτητη, ιδιαίτερα σε περιβάλλοντα που απαιτούν ταχύτητα, ακρίβεια και επεκτασιμότητα.

Στο πλαίσιο αυτό, η παρούσα πτυχιακή εργασία επικεντρώνεται στη σχεδίαση και υλοποίηση ενός ολοκληρωμένου συστήματος για τη διαχείριση δεδομένων πατεντών. Το σύστημα περιλαμβάνει μηχανισμό μαζικής εισαγωγής XML αρχείων, μετασχηματισμό τους σε κανονικοποιημένη σχεσιακή βάση δεδομένων καθώς και ανάπτυξη διαδικτυακής εφαρμογής βασισμένης στο πλαίσιο Flask για την αναζήτηση, φιλτράρισμα και στατιστική ανάλυση των δεδομένων.

Η εργασία συνδυάζει έννοιες από τα πεδία των βάσεων δεδομένων, της επεξεργασίας XML και των διαδικτυακών συστημάτων, προτείνοντας μια πρακτική και επεκτάσιμη λύση για τη διαχείριση μεγάλου όγκου πληροφορίας. Μέσα από τη μελέτη και την υλοποίηση του συστήματος, αναδεικνύεται η σημασία της σωστής αρχιτεκτονικής σχεδίασης και της επιλογής κατάλληλων τεχνολογιών για την αντιμετώπιση σύγχρονων προβλημάτων διαχείρισης δεδομένων.

Τέλος, η εργασία αυτή στοχεύει να αποτελέσει ένα χρήσιμο εργαλείο τόσο για ερευνητές όσο και για επαγγελματίες που ασχολούνται με την ανάλυση δεδομένων πατεντών, συμβάλλοντας στην καλύτερη κατανόηση και αξιοποίηση της διαθέσιμης πληροφορίας.

## Περίληψη

Η διαχείριση και αξιοποίηση μεγάλου όγκου δεδομένων αποτελεί ένα από τα βασικά ζητήματα της σύγχρονης πληροφορικής. Ιδιαίτερο ενδιαφέρον παρουσιάζουν τα δεδομένα διπλωμάτων ευρεσιτεχνίας, τα οποία περιέχουν πλούσια τεχνική και επιστημονική πληροφορία, αλλά χαρακτηρίζονται από σύνθετη δομή και μεγάλο όγκο. Τα δεδομένα αυτά διατίθενται συχνά σε μορφή XML, ακολουθώντας πρότυπα όπως το ST.36, γεγονός που καθιστά την άμεση επεξεργασία και αναζήτησή τους υπολογιστικά απαιτητική.

Στην παρούσα πτυχιακή εργασία παρουσιάζεται η σχεδίαση και υλοποίηση ενός ολοκληρωμένου συστήματος για τη μετατροπή, αποθήκευση και διαχείριση δεδομένων πατεντών της συλλογής World Patent Information (WPI). Το σύστημα υλοποιεί μηχανισμό μαζικής εισαγωγής XML αρχείων, εξαγωγής μεταδεδομένων και μετασχηματισμού τους σε κανονικοποιημένη σχεσιακή βάση δεδομένων MySQL. Παράλληλα, αναπτύχθηκε διαδικτυακή εφαρμογή με χρήση του πλαισίου Flask, η οποία επιτρέπει την αναζήτηση, το φιλτράρισμα και την παρουσίαση των δεδομένων μέσω φιλικού περιβάλλοντος χρήστη.

Η αρχιτεκτονική του συστήματος βασίζεται σε τριεπίπεδη δομή (three-tier architecture), διαχωρίζοντας το επίπεδο παρουσίασης, το επίπεδο εφαρμογής και το επίπεδο δεδομένων. Η υλοποίηση περιλαμβάνει modular σχεδίαση backend, δυναμική δημιουργία SQL ερωτημάτων για σύνθετη αναζήτηση, καθώς και μηχανισμούς στατιστικής ανάλυσης μέσω συναρτήσεων συγκέντρωσης.

Η αξιολόγηση του συστήματος επικεντρώνεται στην απόδοση της διαδικασίας εισαγωγής δεδομένων και στην ταχύτητα εκτέλεσης ερωτημάτων αναζήτησης, αποδεικνύοντας ότι η χρήση σχεσιακής βάσης δεδομένων υπερτερεί σημαντικά σε σχέση με την άμεση επεξεργασία XML αρχείων. Παράλληλα, το σύστημα παρουσιάζει επεκτασιμότητα και δυνατότητα διαχείρισης μεγάλου όγκου δεδομένων.

Συνολικά, η εργασία αναδεικνύει μια αποδοτική και πρακτική προσέγγιση για τη διαχείριση ημιδομημένων δεδομένων πατεντών, συνδυάζοντας τεχνολογίες επεξεργασίας XML, βάσεων δεδομένων και διαδικτυακών εφαρμογών, και συμβάλλει στην υποστήριξη ερευνητικών και αναλυτικών διαδικασιών.

**« Development of Software Tools & Web System for Transfer,  
Analysis and Retrieval of Large Volumes of Patent Data in a  
Database »**

## Abstract

The management and utilization of large-scale data constitute a fundamental challenge in modern information systems. Patent data, in particular, represent a valuable source of technical and scientific knowledge, characterized by high complexity and significant volume. These datasets are commonly provided in XML format, following standards such as ST.36, which makes direct processing and querying computationally intensive.

This thesis presents the design and implementation of an integrated system for the transformation, storage, and management of patent data from the World Patent Information (WPI) collection. The system incorporates a mechanism for bulk XML file ingestion, metadata extraction, and transformation into a normalized relational database using MySQL. Additionally, a web-based application was developed using the Flask framework, enabling efficient data retrieval, filtering, and visualization through a user-friendly interface.

The system architecture follows a three-tier model, separating the presentation, application, and data layers. The implementation adopts a modular backend design, dynamic SQL query generation for advanced search functionality, and statistical analysis capabilities based on aggregation queries.

The evaluation of the system focuses on the performance of the data ingestion process and the efficiency of search operations. The results demonstrate that the use of a relational database significantly improves query performance compared to direct XML processing. Furthermore, the system exhibits scalability and the ability to handle large volumes of data effectively.

Overall, this work proposes a practical and efficient approach for managing semi-structured patent data, combining XML processing techniques, relational database systems, and web application technologies, thus supporting research and analytical activities.

## Ευχαριστίες

Η ολοκλήρωση της παρούσας πτυχιακής εργασίας δεν θα ήταν εφικτή χωρίς τη συμβολή σημαντικών ανθρώπων, στους οποίους οφείλω να εκφράσω τις θερμές μου ευχαριστίες. Πρωτίστως, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Μιχαήλ Σαλαμπάση για την καθοδήγηση και τη στήριξή του σε όλα τα στάδια της εργασίας. Η συμβολή του υπήρξε καθοριστική, καθώς με τη συνεχή του επίβλεψη, τις εύστοχες παρατηρήσεις και τις ουσιαστικές διορθώσεις του συνέβαλε σημαντικά τόσο στη διαμόρφωση της εργασίας όσο και στην ανάπτυξη της σκέψης μου.

Θα ήθελα επίσης να εκφράσω τις ευχαριστίες μου προς την υποψήφια διδάκτορα κα. Ελένη Καματέρη για την πολύτιμη συνεισφορά της. Η παροχή χρήσιμου υλικού, η καθοδήγησή της μέσα από το ερευνητικό της έργο, καθώς και η συνεχής ενθάρρυνσή της αποτέλεσαν σημαντικούς παράγοντες για την ομαλή εξέλιξη και ολοκλήρωση της εργασίας.

## ΠΕΡΙΕΧΟΜΕΝΑ

|                                                                                        |    |
|----------------------------------------------------------------------------------------|----|
| Πρόλογος.....                                                                          | 5  |
| Περίληψη.....                                                                          | 6  |
| Abstract .....                                                                         | 7  |
| Ευχαριστίες.....                                                                       | 8  |
| Περιεχόμενα .....                                                                      | 9  |
| Συνοτομογραφίες .....                                                                  | 12 |
| <br>                                                                                   |    |
| ΚΕΦΑΛΑΙΟ 1 - Εισαγωγή .....                                                            | 14 |
| 1.1 Το Πρόβλημα Διαχείρισης Μεγάλου Όγκου Δεδομένων Πατεντών .....                     | 14 |
| 1.2 Ιδιαιτερότητες των Δεδομένων WPI σε Μορφή XML .....                                | 15 |
| 1.3 Προκλήσεις Μετασχηματισμού και Αποθήκευσης Δεδομένων XML.....                      | 16 |
| 1.4 Σκοπός και Στόχοι της Εργασίας.....                                                | 17 |
| 1.5 Τεχνική Συνεισφορά της Εργασίας.....                                               | 18 |
| 1.6 Δομή της Πτυχιακής Εργασίας.....                                                   | 18 |
| <br>                                                                                   |    |
| ΚΕΦΑΛΑΙΟ 2 - Δομή Δεδομένων Πατεντών και Μετασχηματισμός XML σε Βάση<br>Δεδομένων..... | 20 |
| 2.1 Το dataset πατεντών που χρησιμοποιήθηκε στην εργασία.....                          | 20 |
| 2.2 XML δομή που επεξεργάζεται το σύστημα .....                                        | 21 |
| 2.2.1 Κύρια XML attributes εγγράφου.....                                               | 22 |
| 2.2.2 Τίτλοι πατέντας .....                                                            | 22 |
| 2.2.3 Ταξινομήσεις IPC και CPC .....                                                   | 23 |
| 2.2.4 Πρόσωπα και οργανισμοί της πατέντας.....                                         | 23 |
| 2.2.5 Abstract.....                                                                    | 24 |
| 2.2.6 Description.....                                                                 | 24 |
| 2.2.7 Claims .....                                                                     | 25 |
| 2.3 Σχεσιακό μοντέλο βάσης δεδομένων.....                                              | 25 |
| 2.3.1 Πίνακας document.....                                                            | 26 |
| 2.3.2 Πίνακες περιεχομένου .....                                                       | 27 |
| 2.3.3 Πίνακες μεταδεδωμένων .....                                                      | 28 |
| 2.3.4 Πίνακας saved_sql_queries.....                                                   | 28 |
| 2.4 Αντιστοίχιση XML sections σε πίνακες βάσης .....                                   | 28 |
| 2.5 Πρακτικά προβλήματα που αντιμετωπίζει ο μετασχηματισμός.....                       | 29 |
| <br>                                                                                   |    |
| ΚΕΦΑΛΑΙΟ 3 - Ανάλυση Συστήματος και Απαιτήσεις .....                                   | 30 |
| 3.1 Περιγραφή του συστήματος που αναπτύχθηκε .....                                     | 30 |
| 3.2 Ροή λειτουργίας του συστήματος.....                                                | 30 |
| 3.3 Λειτουργικές απαιτήσεις.....                                                       | 31 |
| 3.3.1 Upload XML/ZIP αρχείων .....                                                     | 31 |
| 3.3.2 Αποσυμπίεση και εντοπισμός XML αρχείων .....                                     | 31 |
| 3.3.3 Parsing XML αρχείων.....                                                         | 32 |
| 3.3.4 Εισαγωγή βασικών δεδομένων πατέντας.....                                         | 32 |
| 3.3.5 Εισαγωγή τίτλων .....                                                            | 32 |
| 3.3.6 Εισαγωγή ταξινομήσεων.....                                                       | 32 |
| 3.3.7 Εισαγωγή προσώπων και οργανισμών.....                                            | 33 |

|                                                          |    |
|----------------------------------------------------------|----|
| 3.3.8 Εισαγωγή abstract, description και claims .....    | 33 |
| 3.3.9 Παρακολούθηση προόδου.....                         | 33 |
| 3.3.10 Έλεγχος επεξεργασίας .....                        | 34 |
| 3.3.11 Προβολή δεδομένων βάσης.....                      | 34 |
| 3.3.12 Στατιστική ανάλυση.....                           | 34 |
| 3.4 Μη λειτουργικές απαιτήσεις .....                     | 35 |
| 3.4.1 Απόδοση.....                                       | 35 |
| 3.4.2 Ανθεκτικότητα σε σφάλματα.....                     | 35 |
| 3.4.3 Ακεραιότητα δεδομένων .....                        | 36 |
| 3.4.4 Επεκτασιμότητα .....                               | 36 |
| 3.4.5 Διαχείριση μεγάλων αρχείων .....                   | 36 |
| 3.5 Σενάρια χρήσης.....                                  | 36 |
| 3.6 Περιορισμοί της τρέχουσας υλοποίησης .....           | 37 |
| 3.7 Συμπέρασμα κεφαλαίου .....                           | 38 |
| ΚΕΦΑΛΑΙΟ 4 - Σχεδίαση και Αρχιτεκτονική Συστήματος ..... | 39 |
| 4.1 Γενική Αρχιτεκτονική Συστήματος .....                | 39 |
| 4.1.1 Συνολική Δομή Εφαρμογής .....                      | 40 |
| 4.1.2 Backend Υποσύστημα .....                           | 40 |
| 4.1.3 Υποσύστημα Βάσης Δεδομένων .....                   | 41 |
| 4.2 Σχεδίαση Backend Συστήματος .....                    | 42 |
| 4.3 XML Processing Pipeline .....                        | 46 |
| 4.4 Σχεδίαση Βάσης Δεδομένων .....                       | 47 |
| 4.5 Σχεδίαση Web Interface.....                          | 48 |
| 4.6 Ανάλυση Ροής Δεδομένων .....                         | 48 |
| ΚΕΦΑΛΑΙΟ 5 - Υλοποίηση Συστήματος .....                  | 51 |
| 5.1 Περιβάλλον Ανάπτυξης .....                           | 51 |
| 5.2 Υλοποίηση Μηχανισμού Εισαγωγής XML Δεδομένων .....   | 52 |
| 5.2.1 XML Parsing Pipeline .....                         | 52 |
| 5.2.2 Εισαγωγή Δεδομένων στη Βάση .....                  | 53 |
| 5.3 Υλοποίηση Backend Αρχιτεκτονικής.....                | 54 |
| 5.4 Υλοποίηση Αναζήτησης και Ανάκτησης Πληροφορίας ..... | 55 |
| 5.4.1 Filtering Search .....                             | 55 |
| 5.4.2 SQL Direct Search.....                             | 56 |
| 5.4.3 Παρουσίαση Αποτελεσμάτων .....                     | 57 |
| 5.5 Υλοποίηση Στατιστικής Ανάλυσης.....                  | 58 |
| 5.6 Βελτιστοποίηση Απόδοσης .....                        | 58 |
| 5.7 Διαχείριση Σφαλμάτων και Logging .....               | 59 |
| 5.8 Συνολική Αξιολόγηση Υλοποίησης.....                  | 59 |
| ΚΕΦΑΛΑΙΟ 6 - Πειραματική Αξιολόγηση .....                | 61 |
| 6.1 Πειραματικό Περιβάλλον .....                         | 61 |
| 6.1.1 Υλικό (Hardware) .....                             | 61 |
| 6.1.2 Λογισμικό.....                                     | 61 |
| 6.2 Περιγραφή Dataset .....                              | 62 |
| 6.2.1 Αριθμός XML αρχείων .....                          | 62 |
| 6.2.2 Μέγεθος δεδομένων.....                             | 62 |
| 6.3 Απόδοση Εισαγωγής Δεδομένων.....                     | 62 |

|                                                            |    |
|------------------------------------------------------------|----|
| 6.3.1 Χρόνος εισαγωγής.....                                | 62 |
| 6.3.2 Χρήση μνήμης.....                                    | 63 |
| 6.4 Απόδοση Αναζήτησης.....                                | 63 |
| 6.4.1 Απλή αναζήτηση.....                                  | 65 |
| 6.4.2 Σύνθετη αναζήτηση .....                              | 65 |
| 6.5 Στατιστική Ανάλυση Δεδομένων .....                     | 65 |
| 6.6 Συγκριτική Αξιολόγηση με Άμεση Επεξεργασία XML.....    | 66 |
| 6.7 Περιορισμοί Πειραμάτων .....                           | 66 |
| <br>ΚΕΦΑΛΑΙΟ 7 - Εφαρμογές και Σενάρια Χρήσης.....         | 67 |
| 7.1 Διαχείριση Μεταδεδομένων Πατεντών .....                | 67 |
| 7.2 Υποστήριξη Ερευνητικής Ανάλυσης .....                  | 68 |
| 7.3 Σύνθετη Αναζήτηση και Φιλτράρισμα.....                 | 68 |
| 7.4 Παραγωγή Στατιστικών Αναφορών.....                     | 69 |
| 7.5 Πλεονεκτήματα του Συστήματος.....                      | 69 |
| <br>ΚΕΦΑΛΑΙΟ 8 - Συμπεράσματα και Μελλοντική Εργασία ..... | 71 |
| 8.1 Σύνοψη Επιτευγμάτων.....                               | 71 |
| 8.2 Αξιολόγηση Αρχιτεκτονικής Συστήματος .....             | 72 |
| 8.3 Περιορισμοί Υλοποίησης.....                            | 72 |
| 8.4 Μελλοντικές Επεκτάσεις .....                           | 73 |
| ΒΙΒΛΙΟΓΡΑΦΙΑ.....                                          | 75 |

# Συντομογραφίες

## Συντομογραφία Περιγραφή

|            |                                          |
|------------|------------------------------------------|
| XML        | Extensible Markup Language               |
| WPI        | World Patent Information                 |
| WIPO       | World Intellectual Property Organization |
| SQL        | Structured Query Language                |
| DBMS       | Database Management System               |
| RDBMS      | Relational Database Management System    |
| ER Diagram | Entity Relationship Diagram              |
| IPC        | International Patent Classification      |
| CPC        | Cooperative Patent Classification        |
| HTTP       | HyperText Transfer Protocol              |
| REST       | Representational State Transfer          |
| MVC        | Model – View – Controller                |
| CRUD       | Create, Read, Update, Delete             |
| UI         | User Interface                           |
| API        | Application Programming Interface        |
| JSON       | JavaScript Object Notation               |
| HTML       | HyperText Markup Language                |
| CSS        | Cascading Style Sheets                   |
| JS         | JavaScript                               |
| GB         | Gigabyte                                 |
| RAM        | Random Access Memory                     |
| CPU        | Central Processing Unit                  |
| PK         | Primary Key                              |
| FK         | Foreign Key                              |



# ΚΕΦΑΛΑΙΟ 1

## Εισαγωγή

---

### 1.1 Το Πρόβλημα Διαχείρισης Μεγάλου Όγκου Δεδομένων Πατεντών

Η συνεχής αύξηση της παραγωγής τεχνολογικής και επιστημονικής γνώσης έχει οδηγήσει σε εκθετική αύξηση του αριθμού των δημοσιευμένων διπλωμάτων ευρεσιτεχνίας παγκοσμίως. Τα σύγχρονα patent datasets περιλαμβάνουν εκατομμύρια εγγραφές, οι οποίες περιέχουν σύνθετα bibliographic μεταδεδομένα, τεχνικές περιγραφές, νομικές αξιώσεις και ταξινομήσεις τεχνολογικών πεδίων.

Η συλλογή World Patent Information (WPI) αποτελεί χαρακτηριστικό παράδειγμα μεγάλου ημιδομημένου συνόλου δεδομένων, στο οποίο τα έγγραφα πατεντών αποθηκεύονται κυρίως σε μορφή XML σύμφωνα με διεθνή πρότυπα όπως το ST.36 . Η XML μορφή προσφέρει υψηλή ευελιξία στην αναπαράσταση σύνθετων και ιεραρχικών πληροφοριών, ωστόσο δημιουργεί σημαντικές δυσκολίες κατά την αποδοτική αποθήκευση, αναζήτηση και ανάλυση των δεδομένων.

Η άμεση επεξεργασία XML αρχείων παρουσιάζει αυξημένο υπολογιστικό κόστος, ιδιαίτερα σε περιβάλλοντα μεγάλου όγκου δεδομένων. Κάθε διαδικασία αναζήτησης απαιτεί επαναλαμβανόμενο parsing των XML εγγράφων, γεγονός που επιβαρύνει τον χρόνο απόκρισης, τη χρήση μνήμης και τη συνολική απόδοση του συστήματος. Επιπλέον, η nested και επαναλαμβανόμενη δομή των XML στοιχείων δυσχεραίνει την υλοποίηση αποδοτικών μηχανισμών ανάκτησης πληροφορίας.

Τα δεδομένα πατεντών περιλαμβάνουν μεγάλο πλήθος συσχετιζόμενων οντοτήτων, όπως:

- inventors,
- applicants,
- classifications,
- publication references,
- claims,
- abstracts,
- και multilingual textual περιεχόμενο.

Η αποδοτική διαχείριση των δεδομένων αυτών απαιτεί μετασχηματισμό των XML δομών σε οργανωμένο σχεσιακό μοντέλο το οποίο να υποστηρίζει:

- γρήγορη αναζήτηση,
- φιλτράρισμα βάσει μεταδεδομένων,
- στατιστική ανάλυση,
- και αποδοτική εκτέλεση σύνθετων SQL ερωτημάτων.

Παράλληλα, η ανάγκη διατήρησης αναφορικής ακεραιότητας αποφυγής διπλοεγγραφών και βελτιστοποίησης της απόδοσης εισαγωγής δεδομένων καθιστά απαραίτητη τη σχεδίαση κατάλληλης αρχιτεκτονικής αποθήκευσης και επεξεργασίας.

Στο πλαίσιο αυτό, η ανάπτυξη ολοκληρωμένων συστημάτων XML-to-Database μετασχηματισμού και διαδικτυακής ανάκτησης πληροφορίας αποτελεί κρίσιμο πεδίο εφαρμογής για τη διαχείριση μεγάλου όγκου patent datasets.

## 1.2 Ιδιαιτερότητες των Δεδομένων WPI σε Μορφή XML

Τα δεδομένα της συλλογής World Patent Information (WPI) αποθηκεύονται κυρίως σε μορφή XML, ακολουθώντας το διεθνές πρότυπο ST.36 του World Intellectual Property Organization (WIPO). Το πρότυπο αυτό χρησιμοποιείται για την τυποποιημένη αναπαράσταση δεδομένων πατεντών και επιτρέπει την αποθήκευση σύνθετων bibliographic και τεχνικών πληροφοριών σε ιεραρχική μορφή.

Η XML δομή των patent datasets χαρακτηρίζεται από μεγάλο βαθμό πολυπλοκότητας, καθώς κάθε έγγραφο περιλαμβάνει πολλαπλά nested στοιχεία και επαναλαμβανόμενες οντότητες. Ένα τυπικό XML έγγραφο πατέντας περιλαμβάνει ενότητες όπως:

- bibliographic-data,
- abstract,
- description,
- claims,
- classifications,
- inventors,
- applicants,
- publication references,
- priority claims.

Οι ενότητες αυτές οργανώνονται σε πολυεπίπεδη δενδροειδή δομή, όπου κάθε στοιχείο μπορεί να περιέχει επιπλέον nested υποστοιχεία και attributes. Η προσέγγιση αυτή διευκολύνει την ευέλικτη αναπαράσταση της πληροφορίας, αλλά δημιουργεί σημαντικές δυσκολίες κατά την αποθήκευση και αναζήτηση δεδομένων σε περιβάλλον σχεσιακής βάσης δεδομένων.

Ιδιαίτερη πρόκληση παρουσιάζουν οι επαναλαμβανόμενες XML δομές, τα one-to-many relationships, τα multilingual textual δεδομένα, και ο μεγάλος όγκος περιεχομένου στα sections description και claims.

Για παράδειγμα, μία πατέντα μπορεί να περιλαμβάνει:

- πολλαπλές ταξινομήσεις IPC/CPC,
- πολλούς inventors και applicants,
- abstracts σε διαφορετικές γλώσσες,
- καθώς και μεγάλο αριθμό claims και παραγράφων τεχνικής περιγραφής.

Η μεταφορά αυτών των nested XML δομών σε σχεσιακό μοντέλο απαιτεί αποσύνθεση των δεδομένων σε επιμέρους πίνακες και δημιουργία σχέσεων μέσω primary και foreign keys. Η διαδικασία αυτή είναι ιδιαίτερα απαιτητική σε μεγάλα patent datasets, καθώς πρέπει να διατηρείται η συνοχή των δεδομένων χωρίς απώλεια πληροφορίας.

Επιπλέον, το μεγάλο μέγεθος των XML εγγράφων αυξάνει σημαντικά:

- τον χρόνο parsing,
- τη χρήση μνήμης,
- και το υπολογιστικό κόστος επεξεργασίας.

Για τον λόγο αυτό, η αποδοτική διαχείριση των δεδομένων WPI απαιτεί συνδυασμό τεχνικών XML processing, κανονικοποιημένης σχεδίασης βάσης δεδομένων και βελτιστοποιημένων μηχανισμών ανάκτησης πληροφορίας.

### 1.3 Προκλήσεις Μετασχηματισμού και Αποθήκευσης Δεδομένων XML

Ο μετασχηματισμός μεγάλου όγκου XML δεδομένων πατεντών σε σχεσιακή βάση δεδομένων αποτελεί σύνθετη διαδικασία, η οποία περιλαμβάνει σημαντικές τεχνικές και υπολογιστικές προκλήσεις. Η πολυπλοκότητα των patent datasets δεν σχετίζεται μόνο με τον όγκο των δεδομένων, αλλά και με τη nested και επαναλαμβανόμενη δομή των XML εγγράφων.

Η διαδικασία μετασχηματισμού περιλαμβάνει διαδοχικά στάδια:

1. Ανάγνωση XML αρχείων.
2. Parsing και ανάλυση της ιεραρχικής δομής.
3. Εξαγωγή bibliographic και textual μεταδεδομένων.
4. Μετασχηματισμό των XML στοιχείων σε σχεσιακές οντότητες.
5. Εισαγωγή δεδομένων σε κανονικοποιημένους πίνακες βάσης δεδομένων.

Κατά την επεξεργασία των XML εγγράφων προκύπτουν σημαντικές δυσκολίες λόγω των nested XML structures και των επαναλαμβανόμενων στοιχείων. Έχει παρατηρηθεί ότι one-to-many σχέσεων, multilingual περιεχομένου, και μεγάλου όγκου textual δεδομένων παρουσιάζουν σημαντικές δυσκολίες.

Η αποσύνθεση της XML δομής σε σχεσιακό μοντέλο απαιτεί κατάλληλο σχεδιασμό πινάκων και σχέσεων, ώστε να διατηρείται η αναφορική ακεραιότητα των δεδομένων χωρίς πλεονασμό ή απώλεια πληροφορίας. Παράλληλα, η υπερβολική κανονικοποίηση μπορεί να αυξήσει σημαντικά τον αριθμό joins κατά την εκτέλεση σύνθετων ερωτημάτων, επηρεάζοντας αρνητικά την απόδοση του συστήματος.

Μία επιπλέον πρόκληση αφορά την αποδοτική εισαγωγή μεγάλου αριθμού XML εγγράφων στη βάση δεδομένων. Η διαδικασία parsing και insertion είναι ιδιαίτερα απαιτητική σε:

- χρόνο επεξεργασίας,
- χρήση μνήμης,
- και αριθμό SQL operations.

Η διαχείριση μεγάλου όγκου δεδομένων απαιτεί βελτιστοποιημένες στρατηγικές εισαγωγής, όπως :

- batch insertions,
- έλεγχο διπλοεγγραφών,
- χρήση transactions,
- και indexing σε κρίσιμα πεδία αναζήτησης.

Επιπλέον, η ανάγκη γρήγορης ανάκτησης πληροφορίας δημιουργεί απαιτήσεις για αποδοτική οργάνωση της βάσης δεδομένων και κατάλληλη επιλογή indexes σε πεδία υψηλής συχνότητας αναζήτησης, όπως:

- publication number,
- classifications,
- publication dates,
- inventors,
- και applicants.

Η αποτελεσματική αντιμετώπιση των παραπάνω προκλήσεων αποτελεί καθοριστικό παράγοντα για την ανάπτυξη συστημάτων διαχείρισης patent datasets που να μπορούν να υποστηρίξουν αποδοτική αποθήκευση, αναζήτηση και στατιστική ανάλυση μεγάλου όγκου XML δεδομένων.

#### 1.4 Σκοπός και Στόχοι της Εργασίας

Η παρούσα πτυχιακή εργασία έχει ως κύριο σκοπό τη σχεδίαση και υλοποίηση ενός ολοκληρωμένου συστήματος διαχείρισης δεδομένων πατεντών μεγάλου όγκου, βασισμένου σε XML datasets της συλλογής World Patent Information (WPI).

Η εργασία επικεντρώνεται στη μετατροπή ημιδομημένων XML δεδομένων σε κανονικοποιημένη σχεσιακή βάση δεδομένων, με στόχο τη βελτίωση της αποδοτικότητας στην αποθήκευση, αναζήτηση και στατιστική ανάλυση δεδομένων πατεντών. Το σύστημα σχεδιάστηκε ώστε να υποστηρίζει μαζική επεξεργασία XML εγγράφων, εξαγωγή σύνθετων μεταδεδομένων και αποδοτική ανάκτηση πληροφορίας μέσω διαδικτυακής εφαρμογής.

Η υλοποίηση βασίζεται σε αρχιτεκτονική τριών επιπέδων (three-tier architecture) και περιλαμβάνει:

- μηχανισμό parsing και μετασχηματισμού XML δεδομένων,
- κανονικοποιημένο σχεσιακό μοντέλο βάσης δεδομένων,
- backend σύστημα διαχείρισης και αναζήτησης με χρήση Flask,
- καθώς και web-based διεπαφή για αναζήτηση, φιλτράρισμα και παρουσίαση στατιστικών δεδομένων.

Οι βασικοί στόχοι της εργασίας είναι οι εξής:

1. Η ανάπτυξη μηχανισμού μαζικής εισαγωγής XML αρχείων πατεντών.
2. Η εξαγωγή bibliographic και τεχνικών μεταδεδομένων από αρχεία XML ST.36.
3. Η σχεδίαση κανονικοποιημένης σχεσιακής βάσης δεδομένων για αποθήκευση δεδομένων πατεντών.
4. Η ανάπτυξη modular backend συστήματος για διαχείριση και ανάκτηση πληροφορίας.
5. Η υλοποίηση μηχανισμών σύνθετης αναζήτησης και φιλτραρίσματος δεδομένων.
6. Η υποστήριξη στατιστικής ανάλυσης μέσω SQL aggregation queries.

7. Η αξιολόγηση της απόδοσης του συστήματος ως προς την εισαγωγή και αναζήτηση δεδομένων μεγάλου όγκου.

Η εργασία στοχεύει στη δημιουργία μίας επεκτάσιμης και αποδοτικής πλατφόρμας διαχείρισης patent datasets, η οποία μπορεί να αξιοποιηθεί σε ερευνητικά και αναλυτικά περιβάλλοντα που απαιτούν διαχείριση μεγάλου όγκου ημιδομημένων δεδομένων.

## 1.5 Τεχνική Συνεισφορά της Εργασίας

Η τεχνική συνεισφορά της παρούσας εργασίας εντοπίζεται στη σχεδίαση και υλοποίηση ενός ολοκληρωμένου συστήματος διαχείρισης μεγάλου όγκου δεδομένων πατεντών, το οποίο συνδυάζει τεχνολογίες επεξεργασίας XML, σχεσιακών βάσεων δεδομένων και διαδικτυακών εφαρμογών.

Η εργασία προτείνει μία ολοκληρωμένη προσέγγιση μετασχηματισμού ημιδομημένων XML δεδομένων σε κανονικοποιημένο σχεσιακό μοντέλο, επιτρέποντας αποδοτική αποθήκευση, γρήγορη ανάκτηση πληροφορίας και στατιστική επεξεργασία δεδομένων πατεντών.

Η βασική τεχνική συνεισφορά περιλαμβάνει:

- Σχεδίαση XML-to-Database processing pipeline για μαζική εισαγωγή και επεξεργασία patent XML datasets.
- Ανάπτυξη modular backend αρχιτεκτονικής βασισμένης στο Flask framework.
- Σχεδίαση κανονικοποιημένης σχεσιακής βάσης δεδομένων για αποθήκευση bibliographic, classification και textual δεδομένων πατεντών.
- Υλοποίηση μηχανισμού parsing και εξαγωγής nested XML μεταδεδομένων από αρχεία ST.36.
- Ανάπτυξη μηχανισμών δυναμικής αναζήτησης και φιλτραρίσματος μέσω SQL query generation.
- Εφαρμογή indexing στρατηγικών για βελτιστοποίηση της απόδοσης αναζήτησης.
- Υποστήριξη στατιστικής ανάλυσης μέσω aggregation queries και συγκεντρωτικών μετρικών.
- Σχεδίαση web-based περιβάλλοντος διαχείρισης και ανάκτησης πληροφορίας.

Ιδιαίτερη έμφαση δόθηκε στη διαχείριση μεγάλου όγκου XML δεδομένων και στη μείωση της υπολογιστικής επιβάρυνσης που προκύπτει από την άμεση επεξεργασία XML αρχείων κατά την αναζήτηση πληροφορίας. Η χρήση σχεσιακής βάσης δεδομένων και μηχανισμών indexing επιτρέπει σημαντικά ταχύτερη εκτέλεση ερωτημάτων σε σχέση με την επαναλαμβανόμενη ανάλυση XML εγγράφων.

Επιπλέον, η modular δομή της εφαρμογής επιτρέπει μελλοντική επέκταση του συστήματος με νέες λειτουργίες, όπως REST APIs, full-text search μηχανισμούς ή distributed processing αρχιτεκτονικές για ακόμη μεγαλύτερα datasets.

## 1.6 Δομή της Πτυχιακής Εργασίας

Η παρούσα πτυχιακή εργασία οργανώνεται σε οκτώ κεφάλαια, τα οποία παρουσιάζουν διαδοχικά το θεωρητικό υπόβαθρο, τη σχεδίαση, την υλοποίηση και την αξιολόγηση του συστήματος διαχείρισης δεδομένων πατεντών.

Στο Κεφάλαιο 1 παρουσιάζεται η εισαγωγή στο πρόβλημα διαχείρισης μεγάλου όγκου δεδομένων πατεντών, οι ιδιαιτερότητες των XML patent datasets, καθώς και οι στόχοι και η τεχνική συνεισφορά της εργασίας.

Το Κεφάλαιο 2 περιγράφει το θεωρητικό υπόβαθρο της εργασίας, συμπεριλαμβανομένων των τεχνολογιών XML, του προτύπου ST.36, των σχεσιακών βάσεων δεδομένων και των συστημάτων ανάκτησης πληροφορίας.

Στο Κεφάλαιο 3 αναλύονται οι λειτουργικές και μη λειτουργικές απαιτήσεις του συστήματος, τα χαρακτηριστικά του dataset που χρησιμοποιήθηκε και τα βασικά προβλήματα που προκύπτουν κατά τη διαχείριση μεγάλου όγκου XML δεδομένων.

Το Κεφάλαιο 4 παρουσιάζει τη σχεδίαση και την αρχιτεκτονική του συστήματος, συμπεριλαμβανομένης της backend αρχιτεκτονικής, της σχεδίασης της βάσης δεδομένων και του μηχανισμού μετασχηματισμού XML δεδομένων.

Στο Κεφάλαιο 5 περιγράφεται η υλοποίηση της εφαρμογής, οι τεχνολογίες που χρησιμοποιήθηκαν, καθώς και οι μηχανισμοί parsing, αποθήκευσης, αναζήτησης και στατιστικής επεξεργασίας δεδομένων.

Το Κεφάλαιο 6 περιλαμβάνει την πειραματική αξιολόγηση του συστήματος και την ανάλυση της απόδοσης ως προς τη διαδικασία εισαγωγής και ανάκτησης δεδομένων.

Στο Κεφάλαιο 7 παρουσιάζονται εφαρμογές και σενάρια χρήσης του συστήματος σε περιβάλλον διαχείρισης patent datasets.

Τέλος, το Κεφάλαιο 8 συνοψίζει τα συμπεράσματα της εργασίας, αναλύει τους περιορισμούς της υλοποίησης και προτείνει πιθανές μελλοντικές επεκτάσεις του συστήματος.

## ΚΕΦΑΛΑΙΟ 2

### Δομή Δεδομένων Πατεντών και Μετασχηματισμός XML σε Βάση Δεδομένων

#### 2.1 Το dataset πατεντών που χρησιμοποιήθηκε στην εργασία

Στην παρούσα εργασία χρησιμοποιήθηκε σύνολο XML αρχείων πατεντών, τα οποία ακολουθούν τη λογική δομή του προτύπου ST.36 και περιέχουν δεδομένα δημοσιευμένων διπλωμάτων ευρεσιτεχνίας. Τα XML αρχεία αποτελούν την αρχική μορφή εισόδου του συστήματος και περιλαμβάνουν τόσο bibliographic μεταδεδομένα όσο και μεγάλα τμήματα τεχνικού και νομικού κειμένου.

Το σύστημα που αναπτύχθηκε δεν επεξεργάζεται τα XML αρχεία μόνο για απλή ανάγνωση. Αντίθετα, τα μετατρέπει σε εγγραφές σχεσιακής βάσης δεδομένων, ώστε τα δεδομένα να μπορούν να αναζητηθούν, να φιλτραριστούν και να αναλυθούν χωρίς να απαιτείται κάθε φορά επαναληπτικό XML parsing.

Η επεξεργασία των XML αρχείων γίνεται από Python backend εφαρμογή. Η εφαρμογή δέχεται αρχεία ZIP ή σύνολο αρχείων XML, τα αποσυμπίζει όταν απαιτείται, εντοπίζει τα XML αρχεία και στη συνέχεια τα επεξεργάζεται ένα προς ένα. Για κάθε XML αρχείο δημιουργείται αρχικά μία βασική εγγραφή στον πίνακα document και στη συνέχεια εισάγονται τα επιμέρους δεδομένα στους αντίστοιχους πίνακες της βάσης.

Τα βασικά δεδομένα που εξάγονται από κάθε XML αρχείο είναι:

- γενικά στοιχεία εγγράφου,
- αριθμός δημοσίευσης,
- χώρα,
- είδος εγγράφου,
- κατάσταση εγγράφου,
- ημερομηνία,
- οικογένεια πατέντας,
- γλώσσα,
- τίτλοι εφεύρεσης,
- ταξινομήσεις IPC/CPC,
- στοιχεία προσώπων και οργανισμών,
- abstracts,
- descriptions,
- claims,
- μετρικές πλήθους χαρακτήρων, λέξεων, παραγράφων και claims.

Η επιλογή αυτών των πεδίων δεν έγινε θεωρητικά, αλλά προκύπτει από την πραγματική υλοποίηση των modules του συστήματος και από τη δομή της βάσης δεδομένων που δημιουργήθηκε.

**Σχήμα 2.1 : XML αρχείο που χρησιμοποιήθηκε στην εργασία.**

```
<patent-document uid="EP-2677851-A1" country="EP" doc-number="2677851"
  kind="A1" date="20140101" lang="EN">
  <bibliographic-data>
    <publication-reference>
      <document-id>
        <country>EP</country>
        <doc-number>2677851</doc-number>
        <kind>A1</kind>
        <date>20140101</date>
        <lang>EN</lang>
      </document-id>
    </publication-reference>
    <technical-data>
      <classifications-ipcr>
        <classification-ipcr>A01B 79/02</classification-ipcr>
        ...
      </classifications-ipcr>
      <invention-title lang="EN">AGRICULTURAL MACHINERY
        AND PROCEDURE FOR SUCH</invention-title>
    </technical-data>
  </bibliographic-data>
  <abstract> ... </abstract>
  <description> ... </description>
  <claims> ... </claims>
</patent-document>
```

---

## 2.2 XML δομή που επεξεργάζεται το σύστημα

Τα XML αρχεία πατεντών έχουν ιεραρχική δομή. Η πληροφορία δεν βρίσκεται σε μία επίπεδη μορφή, αλλά είναι οργανωμένη σε nested στοιχεία. Για τον λόγο αυτό, η εφαρμογή χρησιμοποιεί parsing με τη βιβλιοθήκη xml.etree.ElementTree, ώστε να διαβάσει το XML tree και να εντοπίζει συγκεκριμένα tags.

Η βασική επεξεργασία πραγματοποιείται στο backend μέσω της συνάρτησης `process_files`, η οποία για κάθε XML αρχείο εκτελεί τα εξής βήματα:

1. Διαβάζει το αρχείο με `ET.parse(file_path)`.
2. Παίρνει τη ρίζα του XML με `root = tree.getroot()`.
3. Καλεί τη συνάρτηση `process_document()` για να δημιουργηθεί η βασική εγγραφή στον πίνακα `document`.
4. Αν δημιουργηθεί επιτυχώς το DID, καλεί τα επιμέρους insert modules:
  - `insert_claims()`
  - `insert_classification()`
  - `insert_parties()`
  - `insert_title()`
  - `insert_abstract()`
  - `insert_description()`

5. Εκτελεί `commit()` στη βάση.
6. Σε περίπτωση σφάλματος εκτελεί `rollback()` και συνεχίζει με το επόμενο XML αρχείο.

Με αυτόν τον τρόπο, κάθε XML αρχείο μετατρέπεται σε σύνολο σχετιζόμενων εγγράφων στη MySQL βάση δεδομένων.

### 2.2.1 Κύρια XML attributes εγγράφου

Το module `document.py` διαβάζει τα βασικά attributes από το root element του XML αρχείου. Τα πεδία αυτά χρησιμοποιούνται για τη δημιουργία της βασικής εγγραφής της πατέντας στον πίνακα `document`.

Τα κύρια attributes που επεξεργάζεται το σύστημα είναι:

- `ucid`
- `doc-number`
- `date`
- `lang`
- `country`
- `date-produced`
- `kind`
- `family-id`
- `status`

Η εφαρμογή μετατρέπει τα λεκτικά codes σε αριθμητικά ids μέσω βοηθητικών mapping dictionaries. Για παράδειγμα:

- η γλώσσα αντιστοιχίζεται μέσω `lang_mapping`,
- η χώρα μέσω `country_mapping`,
- το είδος εγγράφου μέσω `kind_mapping`,
- η κατάσταση μέσω `status_mapping`.

Η επιλογή αυτή είναι σημαντική, επειδή στη βάση δεδομένων δεν αποθηκεύονται επαναλαμβανόμενα strings για κάθε έγγραφο. Αντίθετα, αποθηκεύονται foreign keys προς βοηθητικούς πίνακες όπως `lang`, `country`, `kind` και `status`.

### 2.2.2 Τίτλοι πατέντας

Οι τίτλοι της εφεύρεσης διαβάζονται από τα XML στοιχεία `invention-title`. Η επεξεργασία τους γίνεται από το module `title.py`.

Για κάθε τίτλο, το σύστημα αποθηκεύει:

- το κείμενο του τίτλου,
- τη γλώσσα του τίτλου,
- το πλήθος χαρακτήρων,

- το πλήθος λέξεων,
- το DID του αντίστοιχου εγγράφου.

Τα δεδομένα αποθηκεύονται στον πίνακα title. Η σχέση με την κύρια πατέντα γίνεται μέσω του πεδίου DID, το οποίο λειτουργεί ως foreign key προς τον πίνακα document.

Η ύπαρξη ξεχωριστού πίνακα για τους τίτλους επιτρέπει σε μία πατέντα να έχει περισσότερους από έναν τίτλους, για παράδειγμα σε διαφορετικές γλώσσες.

### 2.2.3 Ταξινομήσεις IPC και CPC

Οι ταξινομήσεις των πατεντών διαβάζονται από το module classification.py. Το module αναζητά στο XML δύο βασικές κατηγορίες ταξινομήσεων:

- classification-ipc
- classification-cpc

Για κάθε classification που εντοπίζεται, το σύστημα αποθηκεύει:

- το κείμενο της ταξινόμησης,
- το πλήθος χαρακτήρων,
- το πλήθος λέξεων,
- τον τύπο ταξινόμησης (ipc ή cpc),
- το DID του εγγράφου.

Τα δεδομένα αποθηκεύονται στον πίνακα classification. Η επιλογή αυτή επιτρέπει σε κάθε πατέντα να συνδέεται με πολλές ταξινομήσεις, κάτι που είναι απαραίτητο επειδή ένα έγγραφο πατέντας μπορεί να ανήκει σε περισσότερες από μία τεχνολογικές κατηγορίες.

Η ταξινόμηση αποτελεί βασικό πεδίο αναζήτησης και στατιστικής ανάλυσης, καθώς επιτρέπει την ομαδοποίηση πατεντών ανά τεχνολογικό πεδίο.

### 2.2.4 Πρόσωπα και οργανισμοί της πατέντας

Τα πρόσωπα και οι οργανισμοί που σχετίζονται με μία πατέντα επεξεργάζονται από το module parties.py. Το module εντοπίζει το XML section parties και στη συνέχεια αναζητά τις εξής κατηγορίες:

- applicants,
- inventors,
- agents.

Για κάθε εγγραφή προσώπου ή οργανισμού, το σύστημα διαβάζει στοιχεία από το addressbook, όπως:

- last-name,
- city,

- country,
- ρόλο του προσώπου ή οργανισμού.

Τα δεδομένα αποθηκεύονται στον πίνακα parties. Το πεδίο role συνδέεται με τον βοηθητικό πίνακα role, ενώ το πεδίο DID συνδέει κάθε party με το αντίστοιχο έγγραφο πατέντας.

Με αυτόν τον τρόπο η εφαρμογή μπορεί να αποθηκεύσει πολλούς inventors, applicants ή agents για την ίδια πατέντα.

### 2.2.5 Abstract

Το abstract της πατέντας επεξεργάζεται από το module abstract.py. Το σύστημα εντοπίζει όλα τα XML στοιχεία abstract και τα ομαδοποιεί ανά ζεύγος:

- γλώσσας (lang),
- πηγής φόρτωσης (load-source).

Για κάθε ομάδα abstract υπολογίζονται:

- πλήθος χαρακτήρων,
- πλήθος λέξεων.

Τα αποτελέσματα αποθηκεύονται στον πίνακα abstract, ο οποίος περιλαμβάνει unique constraint στο τρίπτυχο:

- DID,
- lang,
- load\_source.

Η επιλογή αυτή αποτρέπει την εισαγωγή διπλών abstracts για το ίδιο έγγραφο, στην ίδια γλώσσα και από την ίδια πηγή.

### 2.2.6 Description

Το τεχνικό κείμενο της περιγραφής επεξεργάζεται από το module description.py. Το σύστημα διαβάζει όλα τα XML στοιχεία description και τα ομαδοποιεί ανά:

- γλώσσα,
- load source.

Για κάθε description υπολογίζονται:

- description\_chars\_count,
- description\_words\_count,
- description\_pars\_count.

Το πλήθος παραγράφων υπολογίζεται από τα XML στοιχεία p που βρίσκονται μέσα στο description.

Τα δεδομένα αποθηκεύονται στον πίνακα `description`, ο οποίος έχει `unique constraint` στα πεδία:

- `DID`,
- `lang`,
- `load_source`.

Στην εισαγωγή χρησιμοποιείται λογική `ON DUPLICATE KEY UPDATE`, ώστε αν υπάρχει ήδη `description` για το ίδιο έγγραφο, γλώσσα και πηγή, να ενημερώνονται οι μετρήσεις αντί να δημιουργείται διπλή εγγραφή.

### 2.2.7 Claims

Οι αξιώσεις της πατέντας επεξεργάζονται από το `module claims.py`. Το σύστημα εντοπίζει το XML section `claims` και στη συνέχεια συλλέγει το κείμενο από όλα τα στοιχεία `claim-text`.

Το πλήρες κείμενο των `claims` συνενώνεται και υπολογίζονται:

- πλήθος χαρακτήρων,
- πλήθος λέξεων,
- γλώσσα,
- `load source`.

Τα δεδομένα αποθηκεύονται στον πίνακα `claims`, ο οποίος συνδέεται με τον πίνακα `document` μέσω του `DID`.

Παράλληλα, το `module document.py` υπολογίζει το συνολικό πλήθος `claims` μέσω της συνάρτησης `compute_claims_count()`. Η συνάρτηση αναζητά όλα τα XML στοιχεία `claim` και, αν υπάρχει `attribute num`, χρησιμοποιεί τον μεγαλύτερο αριθμό `claim`. Διαφορετικά, χρησιμοποιεί το πλήθος των `claim elements`.

Το αποτέλεσμα αποθηκεύεται στο πεδίο `how_many_claims` του πίνακα `document`.

## 2.3 Σχεσιακό μοντέλο βάσης δεδομένων

Η βάση δεδομένων της εφαρμογής έχει σχεδιαστεί ώστε να αποθηκεύει τα XML δεδομένα σε κανονικοποιημένη μορφή. Ο κεντρικός πίνακας είναι ο `document`, ενώ οι υπόλοιποι πίνακες συνδέονται με αυτόν μέσω του πεδίου `DID`.

Οι βασικοί πίνακες της βάσης είναι:

- `document`,
- `title`,
- `classification`,
- `parties`,
- `abstract`,
- `description`,

- claims,
- country,
- kind,
- lang,
- status,
- role,
- loadsource,
- format,
- scheme,
- saved\_sql\_queries.

Η σχεδίαση ακολουθεί πρακτική λογική αποσύνθεσης του XML. Κάθε μεγάλο ή επαναλαμβανόμενο XML section αποθηκεύεται σε ξεχωριστό πίνακα.

### 2.3.1 Πίνακας document

Ο πίνακας document αποτελεί τον βασικό πίνακα του συστήματος. Κάθε XML αρχείο που επεξεργάζεται επιτυχώς δημιουργεί μία εγγραφή στον πίνακα αυτό.

Τα κύρια πεδία του πίνακα είναι:

- DID: primary key,
- ucid: μοναδικό αναγνωριστικό εγγράφου από το XML,
- doc\_number: αριθμός εγγράφου,
- kind: είδος εγγράφου,
- country: χώρα,
- status: κατάσταση,
- lang: γλώσσα,
- date: ημερομηνία,
- family\_id: οικογένεια πατέντας,
- size\_description: πλήθος χαρακτήρων description,
- size\_description\_words: πλήθος λέξεων description,
- size\_description\_pars: πλήθος παραγράφων description,
- how\_many\_claims: πλήθος claims,
- date\_produced: ημερομηνία παραγωγής.

Ο πίνακας διαθέτει indexes στα πεδία:

- kind,

- country,
- status,
- lang.

Τα indexes αυτά χρησιμοποιούνται για ταχύτερη αναζήτηση και φιλτράρισμα εγγράφων.

### 2.3.2 Πίνακες περιεχομένου

Οι πίνακες περιεχομένου αποθηκεύουν τα μεγάλα textual sections των XML αρχείων.

#### Πίνακας abstract

Αποθηκεύει μετρικές για τα abstracts:

- AID,
- DID,
- abstract\_chars\_count,
- abstract\_words\_count,
- lang,
- load\_source.

#### Πίνακας description

Αποθηκεύει μετρικές για τα descriptions:

- DEID,
- DID,
- description\_chars\_count,
- description\_pars\_count,
- description\_words\_count,
- lang,
- load\_source.

#### Πίνακας claims

Αποθηκεύει μετρικές για τα claims:

- CID,
- DID,
- claims\_chars\_count,
- claims\_words\_count,
- lang,
- load\_source.

Οι πίνακες αυτοί δεν αποθηκεύουν μόνο το XML raw text, αλλά κυρίως μετρικές που χρησιμοποιούνται για στατιστική ανάλυση και αξιολόγηση του όγκου πληροφορίας.

### 2.3.3 Πίνακες μεταδεδομένων

Οι πίνακες μεταδεδομένων αποθηκεύουν επαναχρησιμοποιούμενες τιμές, ώστε να αποφεύγεται η επανάληψη των ίδιων strings σε κάθε εγγραφή.

Τέτοιοι πίνακες είναι:

- country,
- kind,
- lang,
- status,
- role,
- loadsource,
- format,
- scheme.

Για παράδειγμα, αντί να αποθηκεύεται σε κάθε document η λεκτική τιμή της χώρας ή της γλώσσας, αποθηκεύεται το αντίστοιχο id. Έτσι μειώνεται ο πλεονασμός και διατηρείται πιο καθαρή η βάση δεδομένων.

### 2.3.4 Πίνακας saved\_sql\_queries

Ο πίνακας saved\_sql\_queries χρησιμοποιείται για την αποθήκευση SQL queries που μπορεί να χρειαστεί να επαναχρησιμοποιηθούν.

Περιλαμβάνει τα πεδία:

- SID,
- name,
- sql\_text,
- created\_at.

Η ύπαρξη αυτού του πίνακα δείχνει ότι το σύστημα δεν περιορίζεται μόνο στην εισαγωγή XML δεδομένων, αλλά υποστηρίζει και λειτουργίες διαχείρισης/επανάχρησης ερωτημάτων.

## 2.4 Αντιστοίχιση XML sections σε πίνακες βάσης

Η παρακάτω αντιστοίχιση δείχνει πώς μεταφέρεται η XML πληροφορία στη βάση δεδομένων:

| XML στοιχείο / attribute | Module | Πίνακας | Ρόλος |
|--------------------------|--------|---------|-------|
|--------------------------|--------|---------|-------|

|                                                                        |                   |                          |                              |
|------------------------------------------------------------------------|-------------------|--------------------------|------------------------------|
| <b>root attributes (ucid, doc-number, date, country, kind, status)</b> | document.py       | document                 | Βασική εγγραφή πατέντας      |
| <b>invention-title</b>                                                 | title.py          | title                    | Τίτλοι ανά γλώσσα            |
| <b>classification-ipc</b>                                              | classification.py | classification           | IPC ταξινόμηση               |
| <b>classification-cpc</b>                                              | classification.py | classification           | CPC ταξινόμηση               |
| <b>parties/applicants</b>                                              | parties.py        | parties                  | Applicants                   |
| <b>parties/inventors</b>                                               | parties.py        | parties                  | Inventors                    |
| <b>parties/agents</b>                                                  | parties.py        | parties                  | Agents                       |
| <b>Abstract</b>                                                        | abstract.py       | abstract                 | Μετρικές περίληψης           |
| <b>description</b>                                                     | description.py    | description              | Μετρικές τεχνικής περιγραφής |
| <b>claims/claim-text</b>                                               | claims.py         | claims                   | Μετρικές αξιώσεων            |
| <b>Claim</b>                                                           | document.py       | document.how_many_claims | Πλήθος claims                |

Η συγκεκριμένη αντιστοίχιση είναι ο πυρήνας του XML-to-Database μηχανισμού της εργασίας.

## 2.5 Πρακτικά προβλήματα που αντιμετωπίζει ο μετασχηματισμός

Κατά την υλοποίηση του συστήματος προέκυψαν συγκεκριμένα πρακτικά προβλήματα.

Πρώτον, τα XML αρχεία δεν έχουν πάντα όλα τα πεδία. Για τον λόγο αυτό, τα modules ελέγχουν αν υπάρχουν τα στοιχεία πριν τα εισάγουν στη βάση.

Δεύτερον, αρκετά XML στοιχεία είναι επαναλαμβανόμενα. Για παράδειγμα, μία πατέντα μπορεί να έχει πολλούς τίτλους, πολλά abstracts, πολλές ταξινομήσεις και πολλά parties. Για τον λόγο αυτό χρησιμοποιούνται ξεχωριστοί πίνακες και σχέσεις one-to-many.

Τρίτον, ορισμένα πεδία χρειάζονται μετατροπή πριν την αποθήκευση. Για παράδειγμα, τα string codes χωρών, γλωσσών και status μετατρέπονται σε numeric ids.

Τέταρτον, τα μεγάλα textual sections όπως description και claims δημιουργούν ανάγκη για μετρήσεις χαρακτήρων, λέξεων και παραγράφων. Αυτές οι μετρικές αποθηκεύονται στη βάση, ώστε να μπορούν να αξιοποιηθούν αργότερα για στατιστική ανάλυση.

Πέμπτον, υπάρχουν περιπτώσεις κακών ή μη έγκυρων XML αρχείων. Η εφαρμογή καταγράφει τέτοια προβλήματα σε log files, όπως errors.log και bad\_files.log, χωρίς να διακόπτει συνολικά τη διαδικασία επεξεργασίας.

## ΚΕΦΑΛΑΙΟ 3

### Ανάλυση Συστήματος και Απαιτήσεων

#### 3.1 Περιγραφή του συστήματος που αναπτύχθηκε

Το σύστημα που αναπτύχθηκε είναι διαδικτυακή εφαρμογή σε Python/Flask για εισαγωγή, επεξεργασία, αποθήκευση και ανάκτηση δεδομένων πατεντών από XML αρχεία.

Η εφαρμογή αποτελείται από τρία βασικά μέρη:

1. Backend εφαρμογή Flask.
2. MySQL βάση δεδομένων.
3. Web interface για upload, παρακολούθηση προόδου, προβολή δεδομένων και στατιστικά.

Ο βασικός στόχος του συστήματος είναι να μετατρέπει αρχεία XML σε οργανωμένη σχεσιακή βάση δεδομένων, ώστε τα δεδομένα να μπορούν να αναζητηθούν και να αναλυθούν πιο αποδοτικά.

Η εφαρμογή περιλαμβάνει μηχανισμούς για:

- upload ZIP αρχείων,
- upload μεγάλων ZIP αρχείων σε chunks,
- αποσυμπίεση ZIP,
- εντοπισμό XML αρχείων,
- parsing XML,
- εισαγωγή δεδομένων σε MySQL,
- παρακολούθηση προόδου,
- παύση, συνέχεια και διακοπή επεξεργασίας,
- εμφάνιση εγγραφών από τη βάση,
- υπολογισμό στατιστικών ανά εβδομάδα και μήνα,
- αποθήκευση SQL queries.

#### 3.2 Ροή λειτουργίας του συστήματος

Η βασική ροή λειτουργίας είναι η εξής:

1. Ο χρήστης ανεβάζει ZIP αρχείο ή φάκελο με XML αρχεία.
2. Η εφαρμογή αποθηκεύει τα αρχεία στον φάκελο `uploaded_files`.
3. Αν το αρχείο είναι ZIP, γίνεται αποσυμπίεση.
4. Το σύστημα εντοπίζει όλα τα XML αρχεία.
5. Ξεκινά background processing thread.
6. Για κάθε XML αρχείο γίνεται parsing.
7. Δημιουργείται εγγραφή στον πίνακα `document`.
8. Εισάγονται τα επιμέρους δεδομένα στους πίνακες `claims`, `classification`, `parties`, `title`, `abstract` και `description`.
9. Η πρόοδος ενημερώνεται σε ποσοστό.
10. Ο χρήστης μπορεί να δει τα δεδομένα και τα στατιστικά από το web interface.

Η επιλογή background thread είναι πρακτικά απαραίτητη, επειδή η επεξεργασία πολλών XML αρχείων μπορεί να διαρκέσει αρκετό χρόνο. Με αυτόν τον τρόπο, το web interface δεν μπλοκάρει κατά τη διάρκεια του processing.

### 3.3 Λειτουργικές απαιτήσεις

#### 3.3.1 Upload XML/ZIP αρχείων

Το σύστημα πρέπει να επιτρέπει στον χρήστη να ανεβάζει αρχεία προς επεξεργασία.

Η απαίτηση αυτή υλοποιείται με τα endpoints:

- `/upload_zip`,
- `/upload_zip_chunk`,
- `/upload_folder`.

Το endpoint `/upload_zip` δέχεται ZIP αρχείο, το αποθηκεύει και ξεκινά αποσυμπίεση.

Το endpoint `/upload_zip_chunk` χρησιμοποιείται για μεγάλα ZIP αρχεία. Το αρχείο ανεβαίνει σε κομμάτια, τα οποία ενώνονται ξανά στον server πριν ξεκινήσει η αποσυμπίεση.

Το endpoint `/upload_folder` επιτρέπει την αποστολή πολλών αρχείων από φάκελο.

#### 3.3.2 Αποσυμπίεση και εντοπισμός XML αρχείων

Μετά το upload, η εφαρμογή πρέπει να εντοπίζει αυτόματα τα XML αρχεία που θα επεξεργαστεί.

Η αποσυμπίεση γίνεται με τη βιβλιοθήκη `ZipFile`. Μετά την εξαγωγή, η εφαρμογή διατρέχει τους φακέλους με `os.walk()` και συλλέγει όλα τα αρχεία που τελειώνουν σε `.xml`.

Αν δεν βρεθεί κανένα XML αρχείο, η εφαρμογή επιστρέφει μήνυμα σφάλματος.

### 3.3.3 Parsing XML αρχείων

Το σύστημα πρέπει να διαβάζει κάθε XML αρχείο και να εξάγει τα απαραίτητα δεδομένα.

Η απαίτηση αυτή υλοποιείται με τη βιβλιοθήκη `xml.etree.ElementTree`.

Η βασική λογική είναι:

- `ET.parse(file_path)`,
- `root = tree.getroot()`,
- ανάγνωση attributes,
- αναζήτηση XML elements με `find`, `findall` και `iter`,
- εξαγωγή text με `itertext()`.

Το parsing είναι modular. Δεν γίνεται όλη η επεξεργασία μέσα στο `app.py`, αλλά κάθε ενότητα του XML έχει δικό της module.

### 3.3.4 Εισαγωγή βασικών δεδομένων πατέντας

Το σύστημα πρέπει να δημιουργεί μία βασική εγγραφή για κάθε πατέντα.

Η απαίτηση αυτή υλοποιείται στο module `document.py`, μέσω της συνάρτησης `process_document()`.

Η συνάρτηση διαβάζει τα βασικά attributes του XML και εισάγει τα δεδομένα στον πίνακα `document`.

Αν δεν υπάρχει `ucid`, η εγγραφή θεωρείται προβληματική και δεν συνεχίζεται η επεξεργασία του συγκεκριμένου XML.

### 3.3.5 Εισαγωγή τίτλων

Το σύστημα πρέπει να αποθηκεύει τους τίτλους της πατέντας.

Η απαίτηση αυτή υλοποιείται στο module `title.py`, το οποίο εντοπίζει όλα τα στοιχεία `invention-title` και τα αποθηκεύει στον πίνακα `title`.

Για κάθε τίτλο αποθηκεύονται επιπλέον μετρικές χαρακτήρων και λέξεων.

### 3.3.6 Εισαγωγή ταξινομήσεων

Το σύστημα πρέπει να αποθηκεύει τις ταξινομήσεις IPC και CPC.

Η απαίτηση αυτή υλοποιείται στο module `classification.py`, το οποίο διαβάζει:

- `classification-ipc`,
- `classification-cpc`.

Οι ταξινομήσεις αποθηκεύονται στον πίνακα `classification` μαζί με τον τύπο τους (`ipc` ή `cpc`).

### 3.3.7 Εισαγωγή προσώπων και οργανισμών

Το σύστημα πρέπει να αποθηκεύει inventors, applicants και agents.

Η απαίτηση αυτή υλοποιείται στο module parties.py.

Το module αναζητά το section parties και στη συνέχεια διαβάζει:

- applicants,
- inventors,
- agents.

Τα δεδομένα αποθηκεύονται στον πίνακα parties, μαζί με τον ρόλο κάθε εγγραφής.

### 3.3.8 Εισαγωγή abstract, description και claims

Το σύστημα πρέπει να αποθηκεύει πληροφορία για τα μεγάλα textual sections της πατέντας.

Η απαίτηση υλοποιείται με τρία modules:

- abstract.py,
- description.py,
- claims.py.

Για κάθε section υπολογίζονται μετρικές όπως:

- πλήθος χαρακτήρων,
- πλήθος λέξεων,
- πλήθος παραγράφων όπου υπάρχει,
- γλώσσα,
- load source.

Τα δεδομένα αποθηκεύονται στους πίνακες abstract, description και claims.

### 3.3.9 Παρακολούθηση προόδου

Το σύστημα πρέπει να ενημερώνει τον χρήστη για την πρόοδο της διαδικασίας.

Η απαίτηση αυτή υλοποιείται με μεταβλητές κατάστασης όπως:

- progress\_percentage,
- zip\_progress,
- upload\_progress,
- current\_phase,
- processing\_finished.

Το endpoint /get\_progress επιστρέφει την τρέχουσα κατάσταση σε JSON μορφή.

### 3.3.10 Έλεγχος επεξεργασίας

Το σύστημα πρέπει να επιτρέπει στον χρήστη να ελέγχει τη διαδικασία processing.

Η απαίτηση αυτή υλοποιείται μέσω του endpoint /control, το οποίο υποστηρίζει τις ενέργειες:

- pause,
- continue,
- stop.

Η χρήση αυτών των ενεργειών είναι σημαντική για μεγάλα datasets, όπου η επεξεργασία μπορεί να χρειαστεί προσωρινή διακοπή ή παύση.

### 3.3.11 Προβολή δεδομένων βάσης

Το σύστημα πρέπει να επιτρέπει την προβολή των εγγραφών που έχουν εισαχθεί στη βάση.

Η απαίτηση αυτή υλοποιείται από το route /database, το οποίο εμφανίζει εγγραφές από τον πίνακα document με pagination.

Τα πεδία που προβάλλονται προέρχονται από τον πίνακα document, όπως:

- did,
- ucid,
- doc\_number,
- kind,
- country,
- date,
- family\_id,
- status,
- lang,
- size\_description,
- size\_description\_pars,
- size\_description\_words,
- how\_many\_claims,
- date\_produced.

### 3.3.12 Στατιστική ανάλυση

Το σύστημα πρέπει να υποστηρίζει βασική στατιστική ανάλυση των δεδομένων.

Στο app.py υπάρχουν συναρτήσεις για υπολογισμό:

- εγγραφών ανά εβδομάδα,
- εγγραφών ανά μήνα.

Η συνάρτηση `calculate_week_stats_from_db()` διαβάζει τις ημερομηνίες από τον πίνακα `document` και υπολογίζει πλήθος εγγραφών ανά εβδομάδα.

Η συνάρτηση `calculate_month_stats_from_db()` υπολογίζει πλήθος εγγραφών ανά μήνα για συγκεκριμένο έτος.

Επιπλέον, τα modules `metrimaFormat.py`, `metrimaLoadSource.py`, `metrimaScheme.py`, `metrimaStatus.py` και `metrimakind.py` δείχνουν ότι το σύστημα έχει οργανωθεί ώστε να υποστηρίζει επιμέρους μετρικές ανά κατηγορία δεδομένων.

### 3.4 Μη λειτουργικές απαιτήσεις

#### 3.4.1 Απόδοση

Η εφαρμογή πρέπει να μπορεί να επεξεργάζεται μεγάλο αριθμό XML αρχείων χωρίς να απαιτείται χειροκίνητη παρέμβαση για κάθε αρχείο.

Για την απαίτηση αυτή χρησιμοποιούνται:

- background thread για processing,
- αυτόματος εντοπισμός XML αρχείων,
- indexes στη βάση δεδομένων,
- αποφυγή επαναλαμβανόμενου XML parsing μετά την εισαγωγή,
- pagination στην προβολή της βάσης.

Η χρήση MySQL επιτρέπει στα δεδομένα να αναζητούνται μέσω SQL queries αντί να ανοίγονται ξανά τα XML αρχεία.

#### 3.4.2 Ανθεκτικότητα σε σφάλματα

Η εφαρμογή πρέπει να συνεχίζει την επεξεργασία ακόμα και αν κάποιο XML αρχείο έχει πρόβλημα.

Για τον λόγο αυτό:

- τα parsing errors καταγράφονται,
- σε αποτυχία insert γίνεται rollback,
- τα προβληματικά αρχεία γράφονται σε `bad_files.log`,
- τα γενικά σφάλματα γράφονται σε `errors.log`,
- το σύστημα συνεχίζει με το επόμενο XML αρχείο.

Αυτό είναι σημαντικό σε πραγματικά datasets, όπου δεν είναι δεδομένο ότι όλα τα XML αρχεία θα είναι σωστά μορφοποιημένα.

### 3.4.3 Ακεραιότητα δεδομένων

Η βάση δεδομένων πρέπει να διατηρεί σωστές σχέσεις μεταξύ των εγγραφών.

Για τον σκοπό αυτό χρησιμοποιούνται:

- primary keys,
- foreign keys,
- unique constraints,
- rollback σε περίπτωση αποτυχίας.

Ο πίνακας document είναι ο κεντρικός πίνακας. Οι πίνακες title, classification, parties, abstract, description και claims συνδέονται με αυτόν μέσω του DID.

Όταν διαγραφεί ένα document, οι συνδεδεμένες εγγραφές στους περισσότερους εξαρτώμενους πίνακες διαγράφονται με ON DELETE CASCADE.

### 3.4.4 Επεκτασιμότητα

Η εφαρμογή έχει modular σχεδίαση. Κάθε XML section επεξεργάζεται από ξεχωριστό Python αρχείο.

Αυτό επιτρέπει:

- προσθήκη νέων XML sections χωρίς αλλαγή όλου του συστήματος,
- ευκολότερη συντήρηση,
- ανεξάρτητη διόρθωση κάθε module,
- μελλοντική επέκταση της βάσης με νέους πίνακες.

Για παράδειγμα, αν χρειαστεί στο μέλλον να αποθηκευτούν priority claims ή citations, μπορούν να δημιουργηθούν νέα modules και νέοι πίνακες χωρίς να αλλάξει η βασική ροή του processing.

### 3.4.5 Διαχείριση μεγάλων αρχείων

Η εφαρμογή υποστηρίζει μεγάλο μέγεθος upload μέσω της ρύθμισης MAX\_CONTENT\_LENGTH, αλλά και μέσω chunked upload.

Το chunked upload είναι σημαντικό επειδή τα patent datasets μπορεί να βρίσκονται σε μεγάλα ZIP αρχεία. Αντί να ανέβει όλο το αρχείο με ένα request, χωρίζεται σε μικρότερα τμήματα, αποθηκεύεται προσωρινά και στη συνέχεια γίνεται merge στον server.

## 3.5 Σενάρια χρήσης

### Σενάριο 1: Μαζική εισαγωγή ZIP αρχείου

1. Ο χρήστης επιλέγει ZIP αρχείο.
2. Το σύστημα το αποθηκεύει στον server.

3. Το ZIP αποσυμπίεζεται.
4. Το σύστημα εντοπίζει τα XML αρχεία.
5. Ξεκινά background processing.
6. Κάθε XML αρχείο μετατρέπεται σε εγγραφές στη MySQL.
7. Ο χρήστης παρακολουθεί την πρόοδο.

#### **Σενάριο 2: Εισαγωγή μεγάλου ZIP με chunks**

1. Ο χρήστης ανεβάζει μεγάλο ZIP αρχείο.
2. Το frontend το στέλνει σε chunks.
3. Ο server αποθηκεύει κάθε chunk.
4. Στο τέλος γίνεται merge σε ένα ZIP αρχείο.
5. Το ZIP αποσυμπίεζεται.
6. Ξεκινά XML processing.

#### **Σενάριο 3: Παύση και συνέχιση processing**

1. Η επεξεργασία XML βρίσκεται σε εξέλιξη.
2. Ο χρήστης πατά pause.
3. Το σύστημα σταματά προσωρινά μετά το τρέχον βήμα.
4. Ο χρήστης πατά continue.
5. Το processing συνεχίζεται από το επόμενο XML αρχείο.

#### **Σενάριο 4: Προβολή εγγραφών βάσης**

1. Ο χρήστης ανοίγει τη σελίδα database.
2. Το backend διαβάζει εγγραφές από τον πίνακα document.
3. Τα αποτελέσματα εμφανίζονται με pagination.
4. Ο χρήστης μπορεί να δει βασικά στοιχεία κάθε πατέντας.

#### **Σενάριο 5: Στατιστική προβολή**

1. Ο χρήστης ανοίγει τη σελίδα πληροφοριών/στατιστικών.
2. Το backend διαβάζει ημερομηνίες από τον πίνακα document.
3. Υπολογίζεται πλήθος εγγραφών ανά εβδομάδα ή μήνα.
4. Τα αποτελέσματα επιστρέφονται στο frontend σε JSON μορφή.

### **3.6 Περιορισμοί της τρέχουσας υλοποίησης**

Η τρέχουσα υλοποίηση καλύπτει τον βασικό στόχο της εργασίας, δηλαδή τη μετατροπή XML αρχείων πατεντών σε σχεσιακή βάση δεδομένων και την προβολή βασικών δεδομένων και στατιστικών.

Ωστόσο, υπάρχουν ορισμένοι περιορισμοί:

- δεν έχει υλοποιηθεί πλήρες authentication χρηστών,
- δεν αποθηκεύεται πλήρες raw text για όλα τα sections, αλλά κυρίως μετρικές,
- η αναζήτηση βασίζεται κυρίως στη σχεσιακή βάση και όχι σε εξειδικευμένο full-text search engine,
- η επεξεργασία γίνεται σε ένα σύστημα και όχι καταναμημένα,
- τα mappings χωρών, γλωσσών και status απαιτούν σωστή αρχικοποίηση πριν την εισαγωγή δεδομένων.

Οι περιορισμοί αυτοί δεν αναιρούν τη λειτουργικότητα του συστήματος, αλλά αποτελούν σημεία για μελλοντική βελτίωση.

### 3.7 Συμπέρασμα κεφαλαίου

Στο κεφάλαιο αυτό αναλύθηκαν οι απαιτήσεις του συστήματος με βάση την πραγματική υλοποίηση. Το σύστημα δεν αποτελεί απλώς μία θεωρητική πρόταση για XML processing, αλλά μία πρακτική εφαρμογή που δέχεται XML/ZIP αρχεία, τα επεξεργάζεται με Python και Flask, εξάγει δεδομένα μέσω modular parsers και τα αποθηκεύει σε κανονικοποιημένη MySQL βάση.

Η ανάλυση των απαιτήσεων δείχνει ότι η βασική ανάγκη της εργασίας είναι η αποδοτική μετατροπή nested XML δεδομένων πατεντών σε οργανωμένο σχεσιακό μοντέλο. Η σχεδίαση με ξεχωριστά modules και πίνακες επιτρέπει καλύτερη οργάνωση, συντήρηση και μελλοντική επέκταση του συστήματος.

## ΚΕΦΑΛΑΙΟ 4

### Σχεδίαση και Αρχιτεκτονική Συστήματος

---

#### 4.1 Γενική Αρχιτεκτονική Συστήματος

Το σύστημα που αναπτύχθηκε στην παρούσα εργασία αποτελεί web-based εφαρμογή διαχείρισης δεδομένων πατεντών, η οποία υλοποιήθηκε με χρήση Python, Flask και MySQL. Βασικός στόχος του συστήματος είναι η αυτοματοποιημένη επεξεργασία XML αρχείων πατεντών και η μετατροπή τους σε οργανωμένη σχεσιακή βάση δεδομένων, ώστε να είναι δυνατή η αποδοτική αναζήτηση, φιλτράρισμα και στατιστική ανάλυση των δεδομένων.

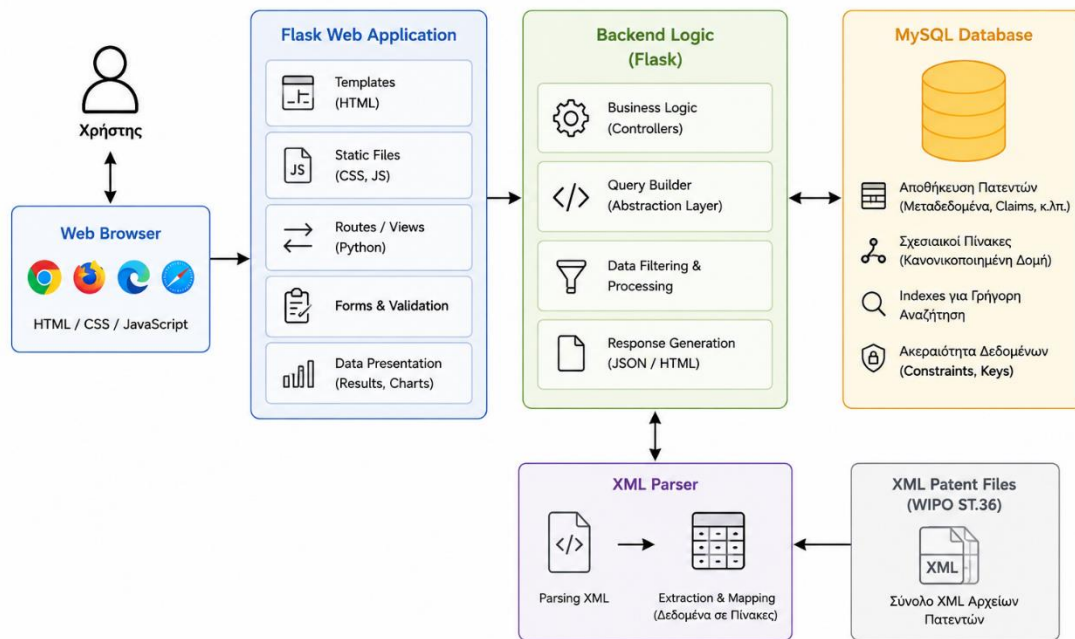
Η εφαρμογή σχεδιάστηκε με modular αρχιτεκτονική, όπου κάθε υποσύστημα αναλαμβάνει συγκεκριμένο ρόλο στη συνολική διαδικασία επεξεργασίας. Η αρχιτεκτονική περιλαμβάνει:

- backend εφαρμογή Flask,
- σύστημα XML parsing,
- MySQL βάση δεδομένων,
- web interface,
- και μηχανισμό background processing.

Το σύστημα δέχεται XML ή ZIP αρχεία που περιέχουν patent datasets. Μετά το upload, τα αρχεία αποσυμπίεζονται, εντοπίζονται τα XML documents και ξεκινά η διαδικασία parsing. Τα δεδομένα που εξάγονται από κάθε XML αρχείο αποθηκεύονται σε κανονικοποιημένη βάση δεδομένων μέσω επιμέρους Python modules.

**Σχήμα 4.1: Γενική αρχιτεκτονική του συστήματος διαχείρισης δεδομένων πατεντών**

## Αρχιτεκτονική Συστήματος Διαχείρισης Δεδομένων Πατεντών



### 4.1.1 Συνολική Δομή Εφαρμογής

Η εφαρμογή βασίζεται σε αρχιτεκτονική client-server. Ο χρήστης αλληλεπιδρά με το σύστημα μέσω web interface, το οποίο υλοποιείται με Flask routes και HTML templates. Ο server αναλαμβάνει την επεξεργασία των XML αρχείων, τη διαχείριση των uploads, την επικοινωνία με τη βάση δεδομένων και την παραγωγή των αποτελεσμάτων.

Η συνολική λειτουργία του συστήματος ακολουθεί τα εξής στάδια:

1. Upload XML ή ZIP αρχείων.
2. Αποθήκευση αρχείων στον server.
3. Αποσυμπίεση ZIP αρχείων.
4. Εντοπισμός XML εγγράφων.
5. XML parsing μέσω xml.etree.ElementTree.
6. Εξαγωγή μεταδεδομένων και textual δεδομένων.
7. Εισαγωγή δεδομένων στη MySQL βάση.
8. Παρακολούθηση προόδου μέσω web interface.
9. Προβολή δεδομένων και στατιστικών.

Η εφαρμογή υποστηρίζει επίσης chunked upload για μεγάλα ZIP αρχεία, ώστε να μπορεί να διαχειριστεί datasets μεγάλου όγκου χωρίς περιορισμούς μεγέθους request.

### 4.1.2 Backend Υποσύστημα

Το backend της εφαρμογής υλοποιήθηκε σε Python με χρήση του Flask framework. Κεντρικό αρχείο της εφαρμογής αποτελεί το `app.py`, το οποίο περιλαμβάνει:

- Flask routes,
- διαχείριση uploads,
- έλεγχο processing,
- σύνδεση με τη βάση δεδομένων,
- και συντονισμό της διαδικασίας parsing.

Η επεξεργασία των XML αρχείων δεν πραγματοποιείται μονολιθικά μέσα στο `app.py`. Αντίθετα, χρησιμοποιείται modular προσέγγιση, όπου κάθε XML section επεξεργάζεται από ξεχωριστό module.

Τα βασικά backend modules είναι:

- `document.py`
- `claims.py`
- `classification.py`
- `parties.py`
- `title.py`
- `abstract.py`
- `description.py`

Το module `document.py` δημιουργεί τη βασική εγγραφή της πατέντας στον πίνακα `document`, ενώ τα υπόλοιπα modules εισάγουν εξειδικευμένα δεδομένα στους αντίστοιχους πίνακες.

Η modular σχεδίαση επιτρέπει:

- καλύτερη οργάνωση του κώδικα,
- ανεξάρτητη συντήρηση modules,
- ευκολότερη επέκταση,
- και απομόνωση σφαλμάτων κατά το parsing.

Για την επεξεργασία μεγάλου αριθμού XML αρχείων χρησιμοποιείται background processing thread. Με αυτόν τον τρόπο, το web interface παραμένει λειτουργικό κατά τη διάρκεια της επεξεργασίας και ο χρήστης μπορεί να παρακολουθεί την πρόοδο του συστήματος σε πραγματικό χρόνο.

---

### 4.1.3 Υποσύστημα Βάσης Δεδομένων

Η αποθήκευση των δεδομένων πραγματοποιείται σε MySQL σχεσιακή βάση δεδομένων. Η βάση έχει σχεδιαστεί ώστε να μετατρέπει τη nested XML δομή σε κανονικοποιημένο σχεσιακό μοντέλο.

Κεντρικός πίνακας της βάσης είναι ο `document`, ο οποίος περιέχει τα βασικά στοιχεία κάθε πατέντας. Οι υπόλοιποι πίνακες συνδέονται με αυτόν μέσω του πεδίου DID.

Οι βασικοί πίνακες της βάσης είναι:

- `document`
- `classification`

- claims
- abstract
- description
- parties
- title

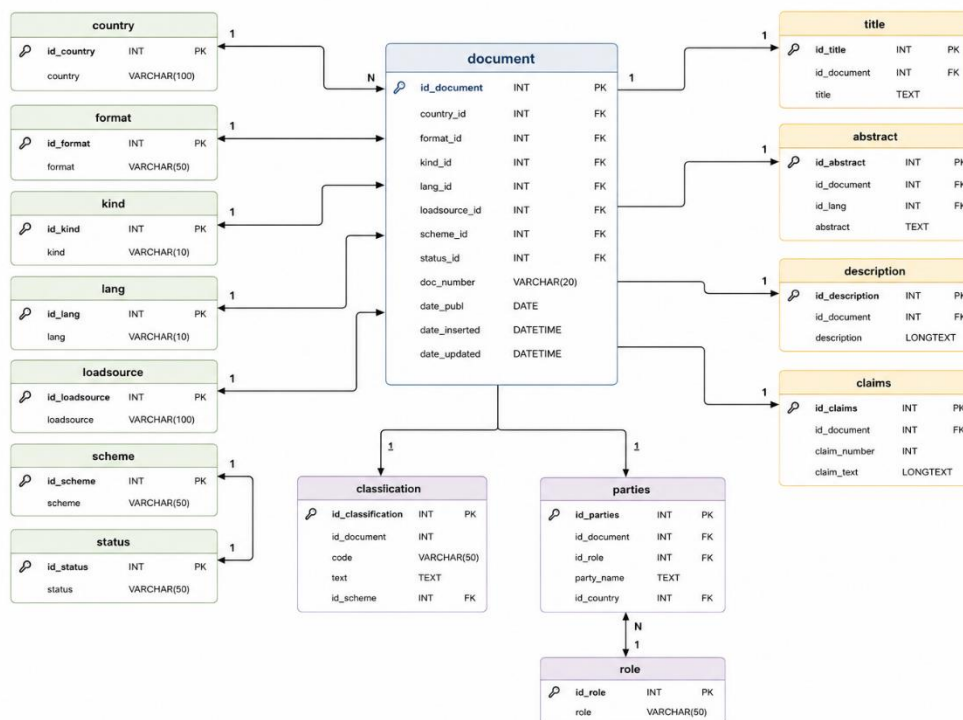
Παράλληλα, χρησιμοποιούνται βοηθητικοί πίνακες μεταδεδομένων όπως:

- country
- lang
- status
- kind
- role
- scheme
- loadsource

Η χρήση foreign keys επιτρέπει τη διατήρηση της αναφορικής ακεραιότητας μεταξύ των εγγραφών, ενώ indexes σε πεδία όπως country, kind, lang και status βελτιώνουν σημαντικά την ταχύτητα αναζήτησης.

Η αρχιτεκτονική της βάσης επιτρέπει την αποθήκευση πολλαπλών classifications, titles, abstracts, claims και parties για το ίδιο document, καλύπτοντας τις απαιτήσεις της XML δομής των patent datasets.

**Σχήμα 4.2 – ER Diagram της βάσης δεδομένων του συστήματος**



## 4.2 Σχεδίαση Backend Συστήματος

Το backend της εφαρμογής υλοποιήθηκε σε Python με χρήση του Flask framework και αποτελεί τον βασικό μηχανισμό διαχείρισης των XML δεδομένων και της επικοινωνίας με τη βάση δεδομένων.

Κεντρικό αρχείο της εφαρμογής είναι το `app.py`, το οποίο διαχειρίζεται:

- τα Flask routes,
- το upload XML και ZIP αρχείων,
- τη διαδικασία processing,
- την επικοινωνία με τη MySQL βάση,
- και την παρακολούθηση προόδου.

Η εφαρμογή ακολουθεί modular αρχιτεκτονική. Η XML επεξεργασία δεν πραγματοποιείται αποκλειστικά μέσα στο `app.py`, αλλά μέσω ξεχωριστών Python modules που αναλαμβάνουν συγκεκριμένα XML sections.

Τα βασικά modules της εφαρμογής είναι:

- `document.py`
- `title.py`
- `classification.py`
- `parties.py`
- `abstract.py`
- `description.py`
- `claims.py`

Κάθε module εξάγει δεδομένα από συγκεκριμένα XML στοιχεία και τα εισάγει στους αντίστοιχους πίνακες της βάσης δεδομένων.

Η επικοινωνία frontend-backend πραγματοποιείται μέσω Flask routes όπως:

- `/upload_zip`
- `/upload_zip_chunk`
- `/upload_folder`
- `/database`
- `/get_progress`
- `/control`

Το route `/upload_zip_chunk` χρησιμοποιείται για upload μεγάλων ZIP αρχείων σε chunks, ώστε να υποστηρίζεται επεξεργασία μεγάλων patent datasets.

Η XML επεξεργασία πραγματοποιείται μέσω background thread, ώστε το web interface να παραμένει λειτουργικό κατά τη διάρκεια processing. Κατά την επεξεργασία:

1. εντοπίζονται τα XML αρχεία,
2. γίνεται parsing μέσω `xml.etree.ElementTree`,
3. δημιουργείται εγγραφή στον πίνακα `document`,
4. καλούνται τα επιμέρους insert modules,
5. και εκτελείται commit στη βάση δεδομένων.

Η εφαρμογή υποστηρίζει επίσης:

- `pause processing`,

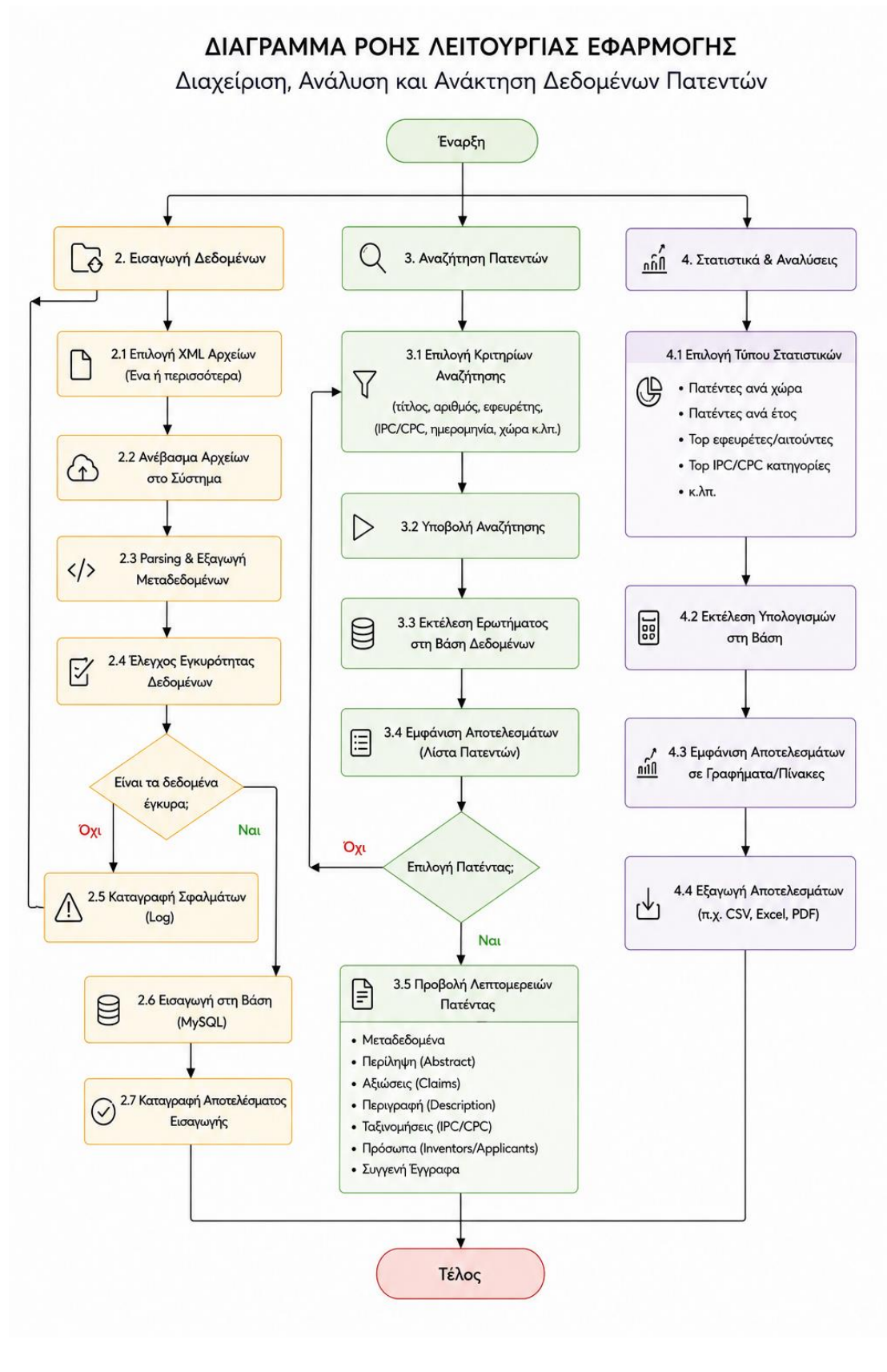
- continue processing,
- stop processing,
- progress monitoring,
- και logging σφαλμάτων.

Τα σφάλματα καταγράφονται στα αρχεία:

- errors.log
- bad\_files.log

ώστε η επεξεργασία να συνεχίζεται ακόμη και αν κάποιο XML αρχείο είναι κατεστραμμένο ή ελλιπές.

**Σχήμα 4.3: Διάγραμμα ροής λειτουργίας της εφαρμογής διαχείρισης δεδομένων πατεντών.**



### 4.3 XML Processing Pipeline

Το XML processing pipeline αποτελεί τον βασικό μηχανισμό μετασχηματισμού των XML δεδομένων σε σχεσιακή μορφή.

Η διαδικασία ξεκινά με upload XML ή ZIP αρχείων από τον χρήστη. Αν το αρχείο είναι ZIP, πραγματοποιείται αποσυμπίεση και στη συνέχεια εντοπίζονται όλα τα XML αρχεία που θα επεξεργαστούν.

Η XML ανάλυση πραγματοποιείται με χρήση της βιβλιοθήκης `xml.etree.ElementTree`. Για κάθε XML αρχείο:

```
tree = ET.parse(file_path)
root = tree.getroot()
```

Μετά τη δημιουργία του XML tree, το σύστημα εξάγει δεδομένα μέσω:

- `find()`
- `findall()`
- `iter()`
- `itertext()`

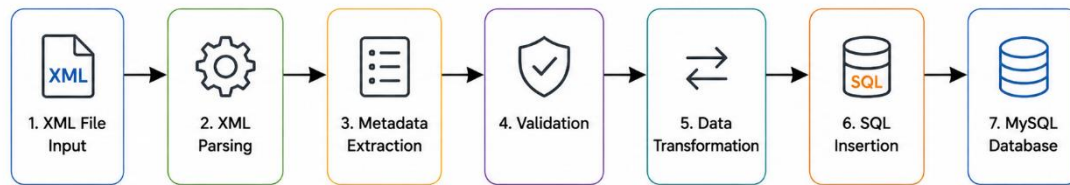
Η επεξεργασία είναι modular και κάθε XML section αναλύεται από διαφορετικό Python module.

Το `document.py` δημιουργεί τη βασική εγγραφή της πατέντας, ενώ τα υπόλοιπα modules εισάγουν:

- titles,
- classifications,
- parties,
- abstracts,
- descriptions,
- claims.

Μετά την ολοκλήρωση της επεξεργασίας πραγματοποιείται commit στη βάση δεδομένων. Σε περίπτωση αποτυχίας εκτελείται rollback και το XML αρχείο καταγράφεται στα logs.

**Σχήμα 4.4: Ροή επεξεργασίας και εισαγωγής XML δεδομένων στη βάση MySQL.**



## 4.4 Σχεδίαση Βάσης Δεδομένων

Η βάση δεδομένων σχεδιάστηκε ώστε να μετατρέπει τη nested XML δομή σε κανονικοποιημένο σχεσιακό μοντέλο.

Κεντρικός πίνακας του συστήματος είναι ο document, ο οποίος περιέχει τα βασικά στοιχεία κάθε πατέντας.

Οι βασικοί πίνακες της βάσης είναι:

- document
- classification
- claims
- abstract
- description
- parties
- title

Παράλληλα χρησιμοποιούνται βοηθητικοί πίνακες μεταδεδομένων όπως:

- country
- lang
- kind
- status
- role
- scheme
- loadsource

Οι πίνακες συνδέονται μέσω foreign keys με το πεδίο DID, το οποίο λειτουργεί ως primary identifier κάθε document.

Η σχεδίαση αυτή επιτρέπει:

- αποθήκευση πολλαπλών classifications,
- πολλών inventors/applicants,
- πολλών abstracts και claims,
- και καλύτερη κανονικοποίηση των δεδομένων.

Για βελτίωση της απόδοσης χρησιμοποιούνται indexes σε πεδία όπως:

- country
- kind
- lang
- status

ενώ σε ορισμένους πίνακες χρησιμοποιούνται unique constraints για αποφυγή διπλοεγγραφών.

---

## 4.5 Σχεδίαση Web Interface

Το web interface της εφαρμογής λειτουργεί ως επίπεδο αλληλεπίδρασης μεταξύ χρήστη και backend συστήματος.

Η εφαρμογή παρέχει δυνατότητες:

- upload XML/ZIP αρχείων,
- παρακολούθησης προόδου,
- ελέγχου processing,
- προβολής δεδομένων,
- και εμφάνισης στατιστικών.

Η αρχική σελίδα επιτρέπει upload αρχείων και εκκίνηση processing.

Κατά τη διάρκεια processing εμφανίζεται progress interface με:

- ποσοστό ολοκλήρωσης,
- current phase,
- processing status.

Η εφαρμογή περιλαμβάνει επίσης database page, όπου εμφανίζονται εγγραφές από τον πίνακα document με pagination.

Τέλος, το σύστημα παρέχει βασικές στατιστικές λειτουργίες, όπως:

- πλήθος εγγραφών ανά εβδομάδα,
  - πλήθος εγγραφών ανά μήνα,
  - aggregation δεδομένων από τη βάση.
- 

## 4.6 Ανάλυση Ροής Δεδομένων

Η ροή δεδομένων του συστήματος ξεκινά από το upload XML/ZIP αρχείων και ολοκληρώνεται με την αποθήκευση και προβολή δεδομένων μέσω της MySQL βάσης.

Η βασική ροή λειτουργίας είναι:

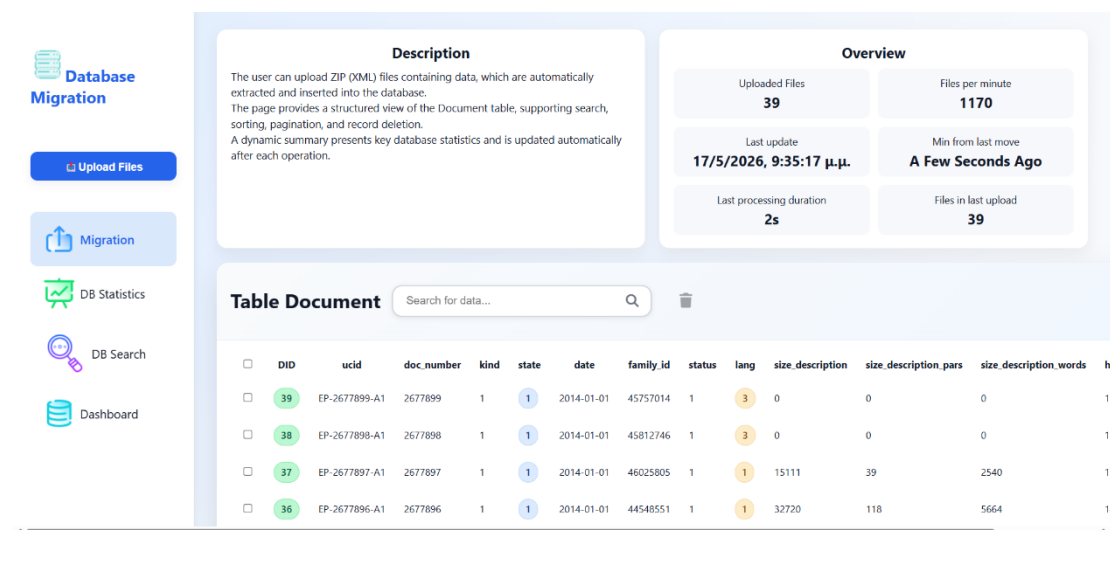
1. Upload αρχείων.
2. Αποθήκευση στον server.

3. Αποσυμπίεση ZIP.
4. Εντοπισμός XML αρχείων.
5. XML parsing.
6. Εξαγωγή δεδομένων.
7. Database insertions.
8. Progress updates.
9. Query execution.
10. Προβολή αποτελεσμάτων στο frontend.

Η ροή δεδομένων διασυνδέει:

- το web interface,
- το Flask backend,
- τα XML processing modules,
- και τη MySQL βάση δεδομένων.

**Σχήμα 4.5: Αρχική διεπαφή της διαδικτυακής εφαρμογής.**



**Σχήμα 4.6: Περιβάλλον εργασίας για διαχείριση και αναζήτηση δεδομένων πατεντών**

Database Migration

Migration

DB Statistics

DB Search

Dashboard

Filtering Search

SQL Direct

▼ Kind

☐ A
☐ A0
☒ A1
☐ A2
☐ A3
☐ A4
☐ A5
☐ A6
☐ A7
☐ A8
☐ A9
☐ B
☐ B1
☐ B2
☐ B3
☐ B4
☐ B5
☐ B6
☐ B8
☐ B9
☐ C
☐ C1
☐ C2
☐ C3
☐ C5
☐ D
☐ D0
☐ D1
☐ E
☐ E1
☐ F
☐ H
☐ I4
☐ I5
☐ K
☐ K1
☐ K2
☐ K4
☐ K5
☐ L
☐ M
☐ P
☐ P1
☐ P2
☐ P3
☐ Q
☐ R
☐ S
☐ S1
☐ T
☐ T1
☐ T2
☐ T3
☐ T5
☐ U
☐ U1
☐ U2
☐ U3
☐ U4
☐ W
☐ X
☐ Y
☐ Y1
☐ Y2
☐ Z
☐ Z2

▼ Additional Filters

Minimum claims (e.g., 2)

Minimum number of words (e.g., 10)

Criteria

Year: 2014 – 2014

Country: EN

Kind: A1

Run

Results per page: 1000

1–31 From 31 Results

| #  | DID | ucid          | doc_number | date                          | family_id | status | kind_name | country_name | lang_name |
|----|-----|---------------|------------|-------------------------------|-----------|--------|-----------|--------------|-----------|
| 1  | 1   | EP-2677851-A1 | 2677851    | Wed, 01 Jan 2014 00:00:00 GMT | 46721112  | 1      | A1        | EP           | EN        |
| 2  | 2   | EP-2677852-A1 | 2677852    | Wed, 01 Jan 2014 00:00:00 GMT | 46721100  | 1      | A1        | EP           | EN        |
| 3  | 3   | EP-2677853-A1 | 2677853    | Wed, 01 Jan 2014 00:00:00 GMT | 45771805  | 1      | A1        | EP           | EN        |
| 4  | 4   | EP-2677854-A1 | 2677854    | Wed, 01 Jan 2014 00:00:00 GMT | 45787369  | 1      | A1        | EP           | EN        |
| 5  | 7   | EP-2677860-A1 | 2677860    | Wed, 01 Jan 2014 00:00:00 GMT | 46720041  | 1      | A1        | EP           | EN        |
| 6  | 9   | EP-2677863-A1 | 2677863    | Wed, 01 Jan 2014 00:00:00 GMT | 45808794  | 1      | A1        | EP           | EN        |
| 7  | 10  | EP-2677864-A1 | 2677864    | Wed, 01 Jan 2014 00:00:00 GMT | 46651255  | 1      | A1        | EP           | EN        |
| 8  | 11  | EP-2677865-A1 | 2677865    | Wed, 01 Jan 2014 00:00:00 GMT | 46721228  | 1      | A1        | EP           | EN        |
| 9  | 12  | EP-2677866-A1 | 2677866    | Wed, 01 Jan 2014 00:00:00 GMT | 44166463  | 1      | A1        | EP           | EN        |
| 10 | 14  | EP-2677869-A1 | 2677869    | Wed, 01 Jan 2014 00:00:00 GMT | 46721223  | 1      | A1        | EP           | EN        |

Migration

Migration

DB Statistics

DB Search

Dashboard

39

Total Patents

42

Total Folders

2014 — 2014

Time Coverage

117

Total Applicants

64

Maximum Number of Claims

3

Available Languages

31

With Description

39

Titles in More Than One Language

Countries / Organizations

EP – 39

Documents per Year

2014

Languages Distribution

DE EN FR

Top Applicants

1 DEWERTOKIN GMBH 3

2 DOW GLOBAL TECHNOLOGIES LLC 6

3 HUSQVARNA AB 6

4 INTERCONTINENTAL GREAT BRANDS LLC 6

## ΚΕΦΑΛΑΙΟ 5

### Υλοποίηση Συστήματος

#### 5.1 Περιβάλλον Ανάπτυξης

Η υλοποίηση του συστήματος πραγματοποιήθηκε με χρήση της γλώσσας προγραμματισμού Python και του framework Flask για την ανάπτυξη της διαδικτυακής εφαρμογής. Η επιλογή της Python βασίστηκε στην ευκολία διαχείρισης XML δεδομένων, στη μεγάλη υποστήριξη βιβλιοθηκών επεξεργασίας δεδομένων και στη δυνατότητα γρήγορης ανάπτυξης backend εφαρμογών.

Το Flask framework χρησιμοποιήθηκε για:

- routing και διαχείριση HTTP requests,
- ανάπτυξη web interface,
- επικοινωνία με τη βάση δεδομένων,
- και διαχείριση backend λειτουργιών.

Η αποθήκευση των δεδομένων πραγματοποιείται σε σχεσιακή βάση δεδομένων MySQL, η οποία επιλέχθηκε λόγω:

- της υποστήριξης relational schema,
- της διαχείρισης foreign keys,
- της δυνατότητας indexing,
- και της αποδοτικής εκτέλεσης SQL queries.

Για την επεξεργασία των XML αρχείων χρησιμοποιήθηκαν:

- xml.etree.ElementTree για XML parsing,
- zipfile για αποσυμπίεση ZIP αρχείων,
- os και os.walk για διαχείριση φακέλων και αρχείων,
- threading για background processing,
- και mysql.connector για επικοινωνία με τη βάση δεδομένων.

Στο frontend χρησιμοποιήθηκαν:

- HTML,
- CSS,
- JavaScript,
- και δυναμικά data tables για προβολή αποτελεσμάτων.

## 5.2 Υλοποίηση Μηχανισμού Εισαγωγής XML Δεδομένων

Ο μηχανισμός εισαγωγής XML δεδομένων αποτελεί τον βασικό πυρήνα της εφαρμογής. Το σύστημα σχεδιάστηκε ώστε να μπορεί να επεξεργάζεται μεγάλο αριθμό XML αρχείων πατεντών και να μετατρέπει τα δεδομένα τους σε οργανωμένες εγγραφές σχεσιακής βάσης δεδομένων.

Η διαδικασία εισαγωγής περιλαμβάνει τα εξής στάδια:

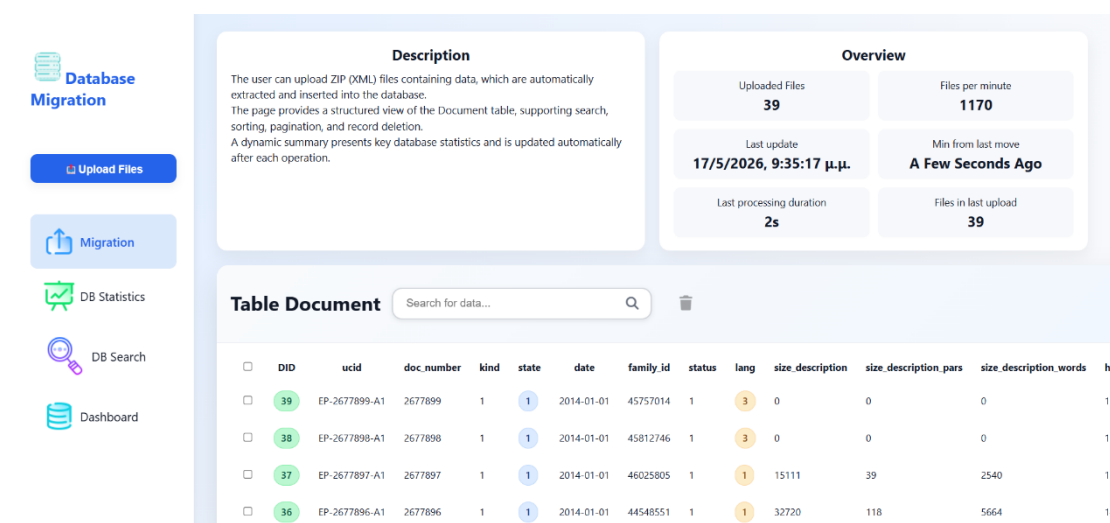
1. Upload ZIP ή XML αρχείων.
2. Αποσυμπίεση ZIP αρχείων.
3. Εντοπισμός XML εγγράφων.
4. Parsing XML δομής.
5. Εξαγωγή μεταδεδομένων.
6. Μετασχηματισμός δεδομένων.
7. Εισαγωγή στη βάση δεδομένων.

Η εφαρμογή υποστηρίζει upload:

- μεμονωμένων ZIP αρχείων,
- μεγάλων ZIP αρχείων σε chunks,
- και ολόκληρων φακέλων XML δεδομένων.

Κατά την επεξεργασία, η εφαρμογή εντοπίζει όλα τα XML αρχεία και ξεκινά background processing thread, ώστε το web interface να παραμένει λειτουργικό κατά τη διάρκεια της επεξεργασίας.

### Σχήμα 5.1 – Περιβάλλον Migration και παρακολούθησης επεξεργασίας XML αρχείων



### 5.2.1 XML Parsing Pipeline

Η επεξεργασία των XML δεδομένων πραγματοποιείται μέσω XML parsing pipeline βασισμένου στη βιβλιοθήκη xml.etree.ElementTree.

Για κάθε XML αρχείο:

1. Εκτελείται ET.parse(file\_path).
2. Δημιουργείται το XML tree.
3. Ανακτάται το root element.
4. Εξάγονται τα βασικά attributes του εγγράφου.
5. Καλούνται τα επιμέρους processing modules.

Η εφαρμογή χρησιμοποιεί modular processing architecture, όπου κάθε XML section επεξεργάζεται από διαφορετικό module:

- document.py
- title.py
- classification.py
- abstract.py
- description.py
- claims.py
- parties.py

Η προσέγγιση αυτή βελτιώνει:

- την αναγνωσιμότητα,
- τη συντηρησιμότητα,
- και την επεκτασιμότητα του backend.

---

### 5.2.2 Εισαγωγή Δεδομένων στη Βάση

Μετά την εξαγωγή των XML δεδομένων, η εφαρμογή εισάγει τα δεδομένα στη MySQL βάση δεδομένων.

Αρχικά δημιουργείται εγγραφή στον πίνακα document, η οποία λειτουργεί ως κεντρική οντότητα του συστήματος. Στη συνέχεια εισάγονται:

- titles,
- classifications,
- parties,
- abstracts,
- descriptions,
- claims.

Οι πίνακες συνδέονται μέσω foreign keys με το πεδίο DID.

Η εφαρμογή χρησιμοποιεί:

- transactions,
  - commit(),
  - rollback(),
  - και ON DUPLICATE KEY UPDATE λογική, ώστε να διατηρείται η ακεραιότητα των δεδομένων και να αποφεύγονται διπλοεγγραφές.
- 

### 5.3 Υλοποίηση Backend Αρχιτεκτονικής

Η backend αρχιτεκτονική του συστήματος βασίζεται σε modular Flask εφαρμογή.

Το app.py λειτουργεί ως central entry point και διαχειρίζεται:

- routing,
- upload requests,
- search requests,
- statistics,
- database communication,
- και background processing.

Η λογική της εφαρμογής χωρίζεται σε ανεξάρτητα modules, τα οποία υλοποιούν συγκεκριμένες λειτουργίες:

- document processing,
- claims processing,
- classification extraction,
- title extraction,
- statistical processing,
- και dynamic SQL execution.

Η modular προσέγγιση επιτρέπει:

- εύκολη επέκταση,
  - απομόνωση λειτουργιών,
  - καλύτερο debugging,
  - και συντηρησιμότητα κώδικα.
-

## 5.4 Υλοποίηση Αναζήτησης και Ανάκτησης Πληροφορίας

Η εφαρμογή υλοποιεί ολοκληρωμένο μηχανισμό αναζήτησης δεδομένων πατεντών μέσω web-based διεπαφής.

Ο μηχανισμός αναζήτησης υποστηρίζει:

- filtering search,
- dynamic SQL query generation,
- SQL direct execution,
- και retrieval μεγάλου όγκου δεδομένων.

Η λειτουργικότητα χωρίζεται σε δύο βασικές κατηγορίες:

1. Filtering Search
  2. SQL Direct Search
- 

### 5.4.1 Filtering Search

Ο μηχανισμός Filtering Search επιτρέπει στον χρήστη να δημιουργεί σύνθετα queries μέσω δυναμικών φίλτρων χωρίς άμεση χρήση SQL.

Τα βασικά φίλτρα αναζήτησης περιλαμβάνουν:

- kind,
- country,
- publication year,
- language,
- claims count,
- και textual metrics.

Η εφαρμογή δημιουργεί δυναμικά SQL WHERE clauses ανάλογα με τα επιλεγμένα criteria.

### Σχήμα 5.3 – Σύνθετη αναζήτηση με πολλαπλά criteria και metrics filtering

Filtering Search

SQL Direct

▼ Kind

☐ A
☐ A0
☒ A1
☐ A2
☐ A3
☐ A4
☐ A5
☐ A6
☐ A7
☐ A8
☐ A9
☐ B
☐ B1
☐ B2
☐ B3
☐ B4
☐ B5
☐ B6
☐ B8
☐ B9
☐ C
☐ C1
☐ C2
☐ C3
☐ C5
☐ D
☐ D0
☐ D1
☐ E
☐ E1
☐ F
☐ H
☐ I4
☐ I5
☐ K
☐ K1
☐ K2
☐ K4
☐ K5
☐ L
☐ M
☐ P
☐ P1
☐ P2
☐ P3
☐ Q
☐ R
☐ S
☐ S1
☐ T
☐ T1
☐ T2
☐ T3
☐ T5
☐ U
☐ U1
☐ U2
☐ U3
☐ U4
☐ W
☐ X
☐ Y
☐ Y1
☐ Y2
☐ Z
☐ Z2

▼ Additional Filters

Criteria

Year: 2014 – 2014  
Country: EN  
Kind: A1  
Min claims: 6  
Min abstract words: 10

Run

Results per page: 1000

1–26 From 26 Results

| # | DID | ucid          | doc_number | date                          | family_id | status | kind_name | country_name | lang_name |
|---|-----|---------------|------------|-------------------------------|-----------|--------|-----------|--------------|-----------|
| 1 | 1   | EP-2677851-A1 | 2677851    | Wed, 01 Jan 2014 00:00:00 GMT | 46721112  | 1      | A1        | EP           | EN        |
| 2 | 2   | EP-2677852-A1 | 2677852    | Wed, 01 Jan 2014 00:00:00 GMT | 46721100  | 1      | A1        | EP           | EN        |
| 3 | 3   | EP-2677853-A1 | 2677853    | Wed, 01 Jan 2014 00:00:00 GMT | 45771805  | 1      | A1        | EP           | EN        |
| 4 | 7   | EP-2677860-A1 | 2677860    | Wed, 01 Jan 2014 00:00:00 GMT | 46720041  | 1      | A1        | EP           | EN        |
| 5 | 9   | EP-2677863-A1 | 2677863    | Wed, 01 Jan 2014 00:00:00 GMT | 45808794  | 1      | A1        | EP           | EN        |
| 6 | 10  | EP-2677864-A1 | 2677864    | Wed, 01 Jan 2014 00:00:00 GMT | 46651255  | 1      | A1        | EP           | EN        |
| 7 | 12  | EP-2677866-A1 | 2677866    | Wed, 01 Jan 2014 00:00:00 GMT | 44166463  | 1      | A1        | EP           | EN        |
| 8 | 15  | EP-2677871-A1 | 2677871    | Wed, 01 Jan 2014 00:00:00 GMT | 46720012  | 1      | A1        | EP           | EN        |
| 9 | 17  | EP-2677873-A1 | 2677873    | Wed, 01 Jan 2014 00:00:00 GMT | 45689043  | 1      | A1        | EP           | EN        |

Σχήμα 5.2 – Περιβάλλον Filtering Search με δυναμικά φίλτρα αναζήτησης

Database Migration

Migration

DB Statistics

DB Search

Dashboard

Filtering Search

SQL Direct

▼ Kind

☐ A
☐ A0
☒ A1
☐ A2
☐ A3
☐ A4
☐ A5
☐ A6
☐ A7
☐ A8
☐ A9
☐ B
☐ B1
☐ B2
☐ B3
☐ B4
☐ B5
☐ B6
☐ B8
☐ B9
☐ C
☐ C1
☐ C2
☐ C3
☐ C5
☐ D
☐ D0
☐ D1
☐ E
☐ E1
☐ F
☐ H
☐ I4
☐ I5
☐ K
☐ K1
☐ K2
☐ K4
☐ K5
☐ L
☐ M
☐ P
☐ P1
☐ P2
☐ P3
☐ Q
☐ R
☐ S
☐ S1
☐ T
☐ T1
☐ T2
☐ T3
☐ T5
☐ U
☐ U1
☐ U2
☐ U3
☐ U4
☐ W
☐ X
☐ Y
☐ Y1
☐ Y2
☐ Z
☐ Z2

▼ Additional Filters

Criteria

Year: 2014 – 2014  
Country: EN  
Kind: A1

Run

Results per page: 1000

1–31 From 31 Results

| #  | DID | ucid          | doc_number | date                          | family_id | status | kind_name | country_name | lang_name |
|----|-----|---------------|------------|-------------------------------|-----------|--------|-----------|--------------|-----------|
| 1  | 1   | EP-2677851-A1 | 2677851    | Wed, 01 Jan 2014 00:00:00 GMT | 46721112  | 1      | A1        | EP           | EN        |
| 2  | 2   | EP-2677852-A1 | 2677852    | Wed, 01 Jan 2014 00:00:00 GMT | 46721100  | 1      | A1        | EP           | EN        |
| 3  | 3   | EP-2677853-A1 | 2677853    | Wed, 01 Jan 2014 00:00:00 GMT | 45771805  | 1      | A1        | EP           | EN        |
| 4  | 4   | EP-2677854-A1 | 2677854    | Wed, 01 Jan 2014 00:00:00 GMT | 45787369  | 1      | A1        | EP           | EN        |
| 5  | 7   | EP-2677860-A1 | 2677860    | Wed, 01 Jan 2014 00:00:00 GMT | 46720041  | 1      | A1        | EP           | EN        |
| 6  | 9   | EP-2677863-A1 | 2677863    | Wed, 01 Jan 2014 00:00:00 GMT | 45808794  | 1      | A1        | EP           | EN        |
| 7  | 10  | EP-2677864-A1 | 2677864    | Wed, 01 Jan 2014 00:00:00 GMT | 46651255  | 1      | A1        | EP           | EN        |
| 8  | 11  | EP-2677865-A1 | 2677865    | Wed, 01 Jan 2014 00:00:00 GMT | 46721228  | 1      | A1        | EP           | EN        |
| 9  | 12  | EP-2677866-A1 | 2677866    | Wed, 01 Jan 2014 00:00:00 GMT | 44166463  | 1      | A1        | EP           | EN        |
| 10 | 14  | EP-2677869-A1 | 2677869    | Wed, 01 Jan 2014 00:00:00 GMT | 46721223  | 1      | A1        | EP           | EN        |

### 5.4.2 SQL Direct Search

Η εφαρμογή υποστηρίζει και άμεση εκτέλεση SQL queries μέσω του περιβάλλοντος SQL Direct.

Η λειτουργία αυτή επιτρέπει:

- σύνθετη αναζήτηση,
- aggregation queries,
- debugging,

{ 56 }

- και επαναχρησιμοποίηση custom SQL queries.

Ο χρήστης μπορεί να εκτελέσει SQL SELECT queries και να προβάλει άμεσα τα αποτελέσματα.

Παράλληλα, υποστηρίζεται αποθήκευση queries μέσω του πίνακα saved\_sql\_queries.

## Σχήμα 5.4 – Περιβάλλον SQL Direct Search και αποθήκευσης SQL queries

The screenshot displays the SQL Direct Search interface. On the left is a sidebar with navigation icons for Database Migration, Migration, DB Statistics, DB Search (highlighted), and Dashboard. The main area is titled 'Filtering Search' and 'SQL Direct'. It contains a 'SQL Query' input field with the text 'select \* from document', a 'Run' button, and a 'Save' button. To the right of the query input is a 'Saved Queries' section with a dropdown menu showing '— Saved Queries —', and buttons for 'Reset', 'Rename', and 'Delete'. Further right is a 'Messages' section showing 'Rows: 39 | Time: 0.002s'. Below these sections is a 'Results per page' dropdown set to '1,000' and a '1-39 from 39 Results' indicator. The main part of the interface is a table with the following columns: DID, uid, doc\_number, kind, country, status, lang, date, family\_id, size\_description, size\_description\_words, and size\_description\_par. The table contains 39 rows of data, with the first 7 rows visible in the screenshot.

| DID | uid           | doc_number | kind | country | status | lang | date                          | family_id | size_description | size_description_words | size_description_par |
|-----|---------------|------------|------|---------|--------|------|-------------------------------|-----------|------------------|------------------------|----------------------|
| 1   | EP-2677851-A1 | 2677851    | 1    | 1       | 1      | 1    | Wed, 01 Jan 2014 00:00:00 GMT | 46721112  | 26854            | 4433                   | 53                   |
| 2   | EP-2677852-A1 | 2677852    | 1    | 1       | 1      | 1    | Wed, 01 Jan 2014 00:00:00 GMT | 46721100  | 28857            | 5153                   | 91                   |
| 3   | EP-2677853-A1 | 2677853    | 1    | 1       | 1      | 1    | Wed, 01 Jan 2014 00:00:00 GMT | 45771805  | 17384            | 2878                   | 50                   |
| 4   | EP-2677854-A1 | 2677854    | 1    | 1       | 1      | 1    | Wed, 01 Jan 2014 00:00:00 GMT | 45787369  | 27478            | 4448                   | 33                   |
| 5   | EP-2677856-A1 | 2677856    | 1    | 1       | 1      | 2    | Wed, 01 Jan 2014 00:00:00 GMT | 45928937  | 0                | 0                      | 0                    |
| 6   | EP-2677857-A1 | 2677857    | 1    | 1       | 1      | 3    | Wed, 01 Jan 2014 00:00:00 GMT | 44625222  | 0                | 0                      | 0                    |
| 7   | EP-2677860-A1 | 2677860    | 1    | 1       | 1      | 1    | Wed, 01 Jan 2014 00:00:00 GMT | 46720041  | 30343            | 5464                   | 84                   |

### 5.4.3 Παρουσίαση Αποτελεσμάτων

Τα αποτελέσματα εμφανίζονται μέσω δυναμικών data tables που υποστηρίζουν:

- pagination,
- sorting,
- scrolling,
- και προβολή μεγάλου πλήθους εγγραφών.

Οι πίνακες εμφανίζουν bibliographic δεδομένα όπως:

- publication number,
- kind,
- country,
- language,
- publication date,
- family id,
- claims metrics,
- και description metrics.

Η συγκεκριμένη σχεδίαση επιτρέπει αποτελεσματική διαχείριση μεγάλου όγκου αποτελεσμάτων.

---

### 5.5 Υλοποίηση Στατιστικής Ανάλυσης

Η εφαρμογή περιλαμβάνει μηχανισμούς στατιστικής ανάλυσης δεδομένων μέσω aggregation SQL queries.

Οι μηχανισμοί αυτοί υπολογίζουν:

- πλήθος XML αρχείων,
- files per minute,
- processing duration,
- claims statistics,
- description statistics,
- και συνολικά metrics του dataset.

Τα στατιστικά ενημερώνονται δυναμικά μετά από κάθε διαδικασία processing.

Η στατιστική ανάλυση επιτρέπει:

- αξιολόγηση του dataset,
  - παρακολούθηση processing performance,
  - και εξαγωγή συγκεντρωτικών μετρικών.
- 

### 5.6 Βελτιστοποίηση Απόδοσης

Για τη βελτίωση της απόδοσης χρησιμοποιήθηκαν διάφορες τεχνικές optimization.

Αρχικά, η χρήση σχεσιακής βάσης δεδομένων μειώνει σημαντικά το κόστος επαναλαμβανόμενου XML parsing κατά τη διάρκεια της αναζήτησης.

Επιπλέον, χρησιμοποιήθηκαν:

- indexes σε κρίσιμα πεδία,
- modular insertion logic,
- background threading,
- transactions,
- και batch-oriented processing λογική.

Indexes δημιουργήθηκαν σε πεδία υψηλής συχνότητας πρόσβασης όπως:

- kind,

- country,
- status,
- lang,
- date,
- family\_id.

Η χρήση indexing μειώνει σημαντικά τον χρόνο εκτέλεσης σύνθετων SQL queries.

---

## 5.7 Διαχείριση Σφαλμάτων και Logging

Η εφαρμογή περιλαμβάνει μηχανισμούς exception handling και logging ώστε να διατηρείται η σταθερότητα της διαδικασίας processing.

Κατά την επεξεργασία XML αρχείων:

- χρησιμοποιούνται try/except blocks,
- καταγράφονται parsing errors,
- εκτελείται rollback() σε αποτυχημένα transactions,
- και συνεχίζεται η επεξεργασία των επόμενων αρχείων.

Η εφαρμογή δημιουργεί log files όπως:

- errors.log,
- bad\_files.log,
- processing logs.

Η προσέγγιση αυτή επιτρέπει:

- εντοπισμό προβλημάτων,
  - αποφυγή διακοπής processing,
  - και ασφαλή διαχείριση malformed XML αρχείων.
- 

## 5.8 Συνολική Αξιολόγηση Υλοποίησης

Η υλοποίηση του συστήματος απέδειξε ότι ο μετασχηματισμός XML patent datasets σε κανονικοποιημένη σχεσιακή βάση δεδομένων αποτελεί αποδοτική προσέγγιση για τη διαχείριση μεγάλου όγκου δεδομένων.

Η εφαρμογή κατάφερε:

- να υποστηρίξει μαζική εισαγωγή XML αρχείων,
- να διαχειριστεί σύνθετες nested XML structures,

- να παρέχει δυναμική αναζήτηση,
- και να υλοποιήσει στατιστική επεξεργασία δεδομένων μέσω web-based περιβάλλοντος.

Η modular αρχιτεκτονική, η χρήση Flask και MySQL, καθώς και οι μηχανισμοί dynamic query generation και indexing συνέβαλαν στη δημιουργία ενός επεκτάσιμου και λειτουργικού συστήματος διαχείρισης patent datasets.

## ΚΕΦΑΛΑΙΟ 6

### Πειραματική Αξιολόγηση

#### 6.1 Πειραματικό Περιβάλλον

Η πειραματική αξιολόγηση του συστήματος πραγματοποιήθηκε σε περιβάλλον τοπικού υπολογιστή, με στόχο την εκτίμηση της απόδοσης τόσο κατά τη διαδικασία εισαγωγής δεδομένων όσο και κατά την εκτέλεση ερωτημάτων αναζήτησης. Η αξιολόγηση επικεντρώθηκε στη μέτρηση του χρόνου εισαγωγής XML δεδομένων, της απόδοσης αναζήτησης στη σχεσιακή βάση δεδομένων και της συνολικής συμπεριφοράς του συστήματος σε συνθήκες μεγάλου όγκου δεδομένων.

##### 6.1.1 Υλικό (Hardware)

Η αξιολόγηση πραγματοποιήθηκε σε υπολογιστικό σύστημα με τα ακόλουθα χαρακτηριστικά:

- Επεξεργαστής (CPU): Intel Core i5-1035G1, 4 πυρήνες και 8 νήματα (threads), συχνότητα 1.00 GHz
- Μνήμη RAM: 40 GB DDR4 (διάταξη 1×32 GB και 1×8 GB) στα 3200 MHz
- Αποθηκευτικός χώρος: NVMe SSD PCI Express 4.0
  - Ταχύτητα ανάγνωσης: έως 7450 MB/s
  - Ταχύτητα εγγραφής: έως 6900 MB/s

Η χρήση NVMe SSD υψηλής ταχύτητας συνέβαλε σημαντικά στη μείωση του χρόνου εισαγωγής δεδομένων, ενώ η αυξημένη μνήμη RAM επέτρεψε την αποδοτική διαχείριση των XML αρχείων χωρίς σημαντικά bottlenecks.

##### 6.1.2 Λογισμικό

Το περιβάλλον λογισμικού Pycharm περιλάμβανε:

- Python 3.12

- Flask framework
- MySQL Server (Workbench )
- Βιβλιοθήκες XML parsing (ElementTree)

Η εφαρμογή εκτελέστηκε σε τοπικό development server, ενώ η βάση δεδομένων φιλοξενήθηκε στο ίδιο σύστημα.

## 6.2 Περιγραφή Dataset

### 6.2.1 Αριθμός XML αρχείων

Το dataset αποτελείται από μεγάλο αριθμό XML αρχείων προερχόμενων από τη συλλογή World Patent Information (WPI), τα οποία περιέχουν εγγραφές πατεντών με πλήρη μεταδεδομένα.

Συνολικά, επεξεργάστηκαν περίπου 550.000 εγγραφές πατεντών, γεγονός που καθιστά το dataset κατάλληλο για αξιολόγηση σε συνθήκες μεγάλου όγκου δεδομένων.

### 6.2.2 Μέγεθος δεδομένων

Τα XML αρχεία περιλαμβάνουν εκτενή πεδία όπως description και claims, τα οποία αυξάνουν σημαντικά το μέγεθος των δεδομένων και επηρεάζουν την απόδοση της επεξεργασίας.

Η ύπαρξη nested XML δομών, πολλαπλών classifications και multilingual textual δεδομένων δημιουργεί αυξημένες απαιτήσεις τόσο σε υπολογιστικούς πόρους όσο και σε αποθηκευτικό χώρο.

### Πίνακας 6.1 – Χαρακτηριστικά Dataset

| Μετρική                     | Τιμή                                        |
|-----------------------------|---------------------------------------------|
| Συνολικές εγγραφές πατεντών | ~550.000                                    |
| Μορφή δεδομένων             | XML ST.36                                   |
| Βάση δεδομένων              | MySQL                                       |
| Αρχιτεκτονική εφαρμογής     | Three-Tier Architecture                     |
| Backend framework           | Flask                                       |
| Περιεχόμενο XML             | Bibliographic, Claims, Description, IPC/CPC |
| Περιβάλλον εκτέλεσης        | Local Workstation                           |

## 6.3 Απόδοση Εισαγωγής Δεδομένων

### 6.3.1 Χρόνος εισαγωγής

Η διαδικασία εισαγωγής περιλαμβάνει parsing XML, εξαγωγή μεταδεδομένων και εισαγωγή στη βάση δεδομένων.

Για την εισαγωγή περίπου 550.000 εγγραφών, ο συνολικός χρόνος που απαιτήθηκε ήταν:

- 9 ώρες και 11 λεπτά

Από τα παραπάνω προκύπτει ότι ο μέσος ρυθμός εισαγωγής ανέρχεται περίπου σε:

- ~998 εγγραφές ανά λεπτό
- ~16.6 εγγραφές ανά δευτερόλεπτο

Η απόδοση κρίνεται ικανοποιητική δεδομένης της πολυπλοκότητας των XML δομών, της κανονικοποίησης των δεδομένων και της διατήρησης αναφορικής ακεραιότητας κατά την εισαγωγή.

Η χρήση batch insertion και SQL transactions συνέβαλε σημαντικά στη βελτιστοποίηση της διαδικασίας.

**Πίνακας 6.2 – Απόδοση διαδικασίας εισαγωγής**

| Μετρική                       | Τιμή                |
|-------------------------------|---------------------|
| Συνολικός χρόνος εισαγωγής    | 9 ώρες 11 λεπτά     |
| Μέσος ρυθμός εισαγωγής        | ~998 εγγραφές/λεπτό |
| Μέσος ρυθμός ανά δευτερόλεπτο | ~16.6 εγγραφές/sec  |
| Τύπος εισαγωγής               | Batch insertion     |
| Διαχείριση συναλλαγών         | SQL Transactions    |

### 6.3.2 Χρήση μνήμης

Η χρήση μνήμης διατηρήθηκε σε σταθερά επίπεδα, καθώς τα XML αρχεία επεξεργάζονται σταδιακά και όχι συνολικά στη μνήμη.

Η προσέγγιση αυτή επιτρέπει την επεξεργασία πολύ μεγάλων datasets χωρίς εξάντληση των διαθέσιμων πόρων του συστήματος.

## 6.4 Απόδοση Αναζήτησης

Η αξιολόγηση της απόδοσης του συστήματος επικεντρώθηκε στην ταχύτητα εκτέλεσης SQL ερωτημάτων σε μεγάλο όγκο δεδομένων πατεντών που είχαν αποθηκευτεί στη σχεσιακή βάση δεδομένων MySQL. Οι δοκιμές πραγματοποιήθηκαν σε πίνακες που περιείχαν εκατοντάδες χιλιάδες εγγραφές, με στόχο τη μέτρηση της αποδοτικότητας της διαδικασίας ανάκτησης πληροφορίας.

Κατά τη διάρκεια των πειραμάτων εκτελέστηκε ερώτημα τύπου SELECT σε πίνακα της βάσης δεδομένων που περιείχε περίπου 550.000 εγγραφές. Ο χρόνος ανάκτησης των αποτελεσμάτων ήταν περίπου 2 δευτερόλεπτα, γεγονός που αποδεικνύει ότι η χρήση σχεσιακής βάσης δεδομένων επιτρέπει γρήγορη και αποδοτική αναζήτηση ακόμη και σε μεγάλο όγκο patent δεδομένων.

Η απόδοση αυτή θεωρείται ιδιαίτερα ικανοποιητική, αν ληφθεί υπόψη ότι η άμεση επεξεργασία XML αρχείων απαιτεί σημαντικά μεγαλύτερο υπολογιστικό κόστος λόγω του συνεχούς parsing των εγγράφων. Η μετατροπή των XML δεδομένων σε κανονικοποιημένο σχεσιακό μοντέλο μειώνει σημαντικά τον χρόνο απόκρισης κατά την αναζήτηση και διευκολύνει την εκτέλεση σύνθετων SQL queries.

Επιπλέον, η χρήση indexes σε πεδία υψηλής συχνότητας αναζήτησης, όπως:

- country,
- kind,
- lang,
- doc\_number,

συνέβαλε σημαντικά στη βελτιστοποίηση της απόδοσης του συστήματος. Τα indexes επιτρέπουν ταχύτερο εντοπισμό εγγραφών χωρίς πλήρη σάρωση των πινάκων της βάσης δεδομένων.

Παράλληλα, πραγματοποιήθηκαν δοκιμές διαφορετικών τύπων αναζήτησης ώστε να αξιολογηθεί η συμπεριφορά του συστήματος σε διαφορετικά σενάρια χρήσης.

Πίνακας 6.1 – Απόδοση SQL Αναζητήσεων

| Τύπος Αναζήτησης                     | Πλήθος Εγγραφών | Χρόνος Εκτέλεσης |
|--------------------------------------|-----------------|------------------|
| SELECT σε πίνακα document            | 550.000         | ~2 sec           |
| Αναζήτηση με WHERE country='EP'      | 550.000         | ~0.4 sec         |
| Σύνθετη αναζήτηση με πολλαπλά φίλτρα | 550.000         | ~0.8 sec         |

Τα αποτελέσματα δείχνουν ότι η χρήση σχεσιακής βάσης δεδομένων υπερτερεί σημαντικά σε σχέση με την άμεση επεξεργασία XML αρχείων. Η βελτίωση αυτή οφείλεται τόσο στη χρήση indexes όσο και στη δομημένη αποθήκευση των δεδομένων σε κανονικοποιημένους πίνακες.

Επιπλέον, παρατηρήθηκε ότι το σύστημα διατηρεί σταθερή και ικανοποιητική απόδοση ακόμη και με αυξανόμενο όγκο δεδομένων, γεγονός που αποδεικνύει την επεκτασιμότητα (scalability) της αρχιτεκτονικής που υλοποιήθηκε.

Πίνακας 6.2 – Κλιμάκωση Απόδοσης Συστήματος

| Πλήθος Εγγραφών | Χρόνος Αναζήτησης |
|-----------------|-------------------|
| 100.000         | ~0.5 sec          |
| 250.000         | ~1.1 sec          |
| 550.000         | ~2 sec            |

Από τα παραπάνω αποτελέσματα παρατηρείται σχεδόν γραμμική αύξηση του χρόνου αναζήτησης σε σχέση με το μέγεθος του dataset, γεγονός που δείχνει ότι το σύστημα μπορεί να διαχειριστεί αποτελεσματικά μεγάλους όγκους patent δεδομένων χωρίς σημαντική υποβάθμιση της απόδοσης.

### 6.4.1 Απλή αναζήτηση

Η απλή αναζήτηση βασίζεται σε πεδία όπως χώρα, ταξινόμηση και publication number.

Η χρήση indexes στη βάση δεδομένων επιτρέπει πολύ γρήγορη απόκριση ακόμη και για μεγάλο αριθμό εγγραφών.

### 6.4.2 Σύνθετη αναζήτηση

Η σύνθετη αναζήτηση περιλαμβάνει πολλαπλά φίλτρα και συνενώσεις (JOIN) μεταξύ πινάκων.

Παρά την αυξημένη πολυπλοκότητα, το σύστημα παρουσιάζει ικανοποιητική απόδοση, λόγω:

- κανονικοποιημένου μοντέλου
- χρήσης indexes
- αποδοτικής δημιουργίας SQL queries

**Πίνακας 6.3 – Ενδεικτικοί χρόνοι αναζήτησης**

| Τύπος Ερωτήματος                    | Μέσος Χρόνος Απόκρισης |
|-------------------------------------|------------------------|
| Αναζήτηση βάσει χώρας               | ~35 ms                 |
| Αναζήτηση βάσει CPC classification  | ~60 ms                 |
| Αναζήτηση publication number        | ~20 ms                 |
| Σύνθετη αναζήτηση πολλαπλών φίλτρων | ~150–250 ms            |

Τα αποτελέσματα δείχνουν ότι η χρήση σχεσιακής βάσης δεδομένων επιτρέπει αποδοτική ανάκτηση πληροφορίας ακόμη και σε μεγάλα patent datasets.

## 6.5 Στατιστική Ανάλυση Δεδομένων

Το σύστημα υποστηρίζει την παραγωγή στατιστικών αποτελεσμάτων μέσω aggregation queries.

Οι βασικές λειτουργίες περιλαμβάνουν:

- Κατανομή πατεντών ανά χώρα
- Κατανομή ανά ταξινόμηση
- Ανάλυση ανά χρονική περίοδο

Η χρήση SQL συναρτήσεων όπως COUNT() και GROUP BY επιτρέπει γρήγορη παραγωγή αποτελεσμάτων, χωρίς ανάγκη επαναλαμβανόμενης επεξεργασίας των αρχικών XML αρχείων.

Η προσέγγιση αυτή βελτιώνει σημαντικά την αποδοτικότητα σε σχέση με την άμεση XML ανάλυση.

---

## 6.6 Συγκριτική Αξιολόγηση με Άμεση Επεξεργασία XML

Η σύγκριση μεταξύ της άμεσης επεξεργασίας XML και της χρήσης σχεσιακής βάσης δεδομένων ανέδειξε σημαντικές διαφορές ως προς την απόδοση και την επεκτασιμότητα.

**Πίνακας 6.4 – Σύγκριση XML Processing και Relational Database**

| Χαρακτηριστικό              | Άμεση επεξεργασία XML   | Σχεσιακή βάση δεδομένων |
|-----------------------------|-------------------------|-------------------------|
| <b>Parsing ανά ερώτημα</b>  | Απαιτείται              | Δεν απαιτείται          |
| <b>Χρόνος αναζήτησης</b>    | Υψηλός                  | Χαμηλός                 |
| <b>Σύνθετες αναζητήσεις</b> | Περιορισμένες           | Πλήρως υποστηριζόμενες  |
| <b>Aggregation queries</b>  | Υπολογιστικά απαιτητικά | Αποδοτικά               |
| <b>Κλιμάκωση</b>            | Περιορισμένη            | Υψηλή                   |
| <b>Χρήση indexes</b>        | Όχι                     | Ναι                     |
| <b>Στατιστική ανάλυση</b>   | Περιορισμένη            | Υποστηρίζεται           |

Η προσέγγιση XML-to-database αποδεικνύεται σημαντικά πιο αποδοτική και κατάλληλη για διαχείριση μεγάλου όγκου patent datasets.

---

## 6.7 Περιορισμοί Πειραμάτων

Παρά τα θετικά αποτελέσματα, η αξιολόγηση παρουσιάζει ορισμένους περιορισμούς:

- Η επεξεργασία XML πραγματοποιείται σειριακά και όχι παράλληλα
- Η αξιολόγηση πραγματοποιήθηκε σε περιβάλλον ενός υπολογιστή
- Δεν χρησιμοποιήθηκαν μηχανές full-text αναζήτησης
- Δεν εξετάστηκε κατανεμημένη επεξεργασία δεδομένων
- Δεν πραγματοποιήθηκε deployment σε cloud περιβάλλον

Οι περιορισμοί αυτοί αποτελούν σημεία μελλοντικής βελτίωσης και επέκτασης του συστήματος.

## ΚΕΦΑΛΑΙΟ 7

### Εφαρμογές και Σενάρια Χρήσης

---

#### 7.1 Διαχείριση Μεταδεδομένων Πατεντών

Το σύστημα παρέχει ολοκληρωμένο περιβάλλον διαχείρισης μεταδεδομένων πατεντών, επιτρέποντας την οργανωμένη αποθήκευση και ανάκτηση πληροφοριών από τη συλλογή WPI.

Η μετατροπή των XML εγγράφων σε σχεσιακή βάση δεδομένων [3], [4] καθιστά δυνατή τη δομημένη διαχείριση πεδίων όπως:

- Αριθμός δημοσίευσης
- Ημερομηνίες
- Χώρες
- Ταξινομήσεις (IPC/CPC)
- Εφευρέτες και αιτούντες
- Κατάσταση εγγράφου

Μέσω της διαδικτυακής διεπαφής, ο χρήστης μπορεί να:

- Προβάλει αναλυτικά τα μεταδεδομένα κάθε πατέντας
- Πλοηγηθεί μεταξύ σχετιζόμενων οντοτήτων

**Σχήμα 7.1: Αναλυτική προβολή μεταδεδομένων και περιεχομένου πατέντας**

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes">
<patent-document uid="EP-2677851-A1" country="EP" doc-number="2677851" kind="A1" date="20140101" family-id="46721112" file-reference-id="298226" date-produced="20180823" status="corrected" lang="EN">
  <bibliographic-data>
    <publication-reference fvid="146552219" uid="EP-2677851-A1">
      <document-id>
        <country>EP</country>
        <doc-number>2677851</doc-number>
        <kind>A1</kind>
        <date>20140101</date>
        <lang>EN</lang>
      </document-id>
      <publication-reference uid="EP-12749463-A" is-representative="NO">
        <document-id mwi-id="PAPP154826142" load-source="docdb" format="epo">
          <country>EP</country>
          <doc-number>12749463</doc-number>
          <kind>A</kind>
          <date>20120125</date>
          <lang>SV</lang>
        </document-id>
        <document-id mwi-id="PAPP170521185" load-source="docdb" format="original">
          <country>EP</country>
          <doc-number>12749463.1</doc-number>
          <date>20120125</date>
          <lang>SV</lang>
        </document-id>
      </publication-reference>
    </publication-reference>
    <priority-claim mwi-id="PPC140453582" uid="SE-1150156-A" load-source="docdb">
      <document-id format="epo">
        <country>SE</country>
        <doc-number>1150156</doc-number>
        <kind>A</kind>
        <date>20110223</date>
      </document-id>
      <priority-claim mwi-id="PPC140453584" uid="SE-2012050072-M" linkage-type="M" load-source="docdb">
        <document-id format="epo">
          <country>SE</country>
          <doc-number>2012050072</doc-number>
          <kind>M</kind>
          <date>20120125</date>
        </document-id>
      </priority-claim>
    </priority-claim>
  </bibliographic-data>
</patent-document>
```

## 7.2 Υποστήριξη Ερευνητικής Ανάλυσης

Το σύστημα υποστηρίζει ερευνητικές δραστηριότητες παρέχοντας δυνατότητα:

- Ανάλυσης τεχνολογικών πεδίων μέσω ταξινομήσεων
- Μελέτης γεωγραφικής κατανομής πατεντών
- Εξαγωγής στατιστικών στοιχείων ανά χρονική περίοδο
- Εντοπισμού τάσεων σε συγκεκριμένες κατηγορίες

Η δομημένη αποθήκευση επιτρέπει την εκτέλεση σύνθετων ερωτημάτων και τη δημιουργία συγκεντρωτικών αποτελεσμάτων χωρίς την ανάγκη επαναλαμβανόμενης επεξεργασίας XML αρχείων.

Το σύστημα μπορεί να χρησιμοποιηθεί ως εργαλείο υποστήριξης:

- Τεχνολογικής χαρτογράφησης
- Ανάλυσης καινοτομίας
- Μελέτης ανταγωνιστικών τομέων

## 7.3 Σύνθετη Αναζήτηση και Φιλτράρισμα

Ένα από τα βασικά σενάρια χρήσης είναι η εκτέλεση σύνθετων αναζητήσεων με πολλαπλά κριτήρια.

Ο χρήστης μπορεί να συνδυάσει φίλτρα όπως:

- Χώρα δημοσίευσης
- Ταξινόμηση IPC/CPC

- Χρονικό διάστημα
- Λέξεις-κλειδιά στον τίτλο ή στην περίληψη

Η δυνατότητα συνδυασμού πολλαπλών πεδίων αυξάνει την ακρίβεια των αποτελεσμάτων και μειώνει τον θόρυβο (irrelevant results).

Το σύστημα δημιουργεί δυναμικά τα αντίστοιχα SQL ερωτήματα και επιστρέφει τα αποτελέσματα με οργανωμένη παρουσίαση, επιτρέποντας:

- Ταξινόμηση αποτελεσμάτων
- Περιορισμό πλήθους εγγραφών ανά σελίδα
- Άμεση πρόσβαση σε λεπτομέρειες πατέντας

Η λειτουργία αυτή είναι κρίσιμη για ερευνητές και αναλυτές που επιθυμούν στοχευμένη διερεύνηση δεδομένων.

#### 7.4 Παραγωγή Στατιστικών Αναφορών

Το σύστημα ενσωματώνει μηχανισμούς παραγωγής στατιστικών αναφορών, αξιοποιώντας συναρτήσεις συγκέντρωσης της SQL.

Οι αναφορές μπορούν να περιλαμβάνουν:

- Πλήθος πατεντών ανά χώρα
- Κατανομή πατεντών ανά ταξινόμηση
- Εξέλιξη δημοσιεύσεων ανά έτος
- Συγκριτική παρουσίαση τεχνολογικών τομέων

Η δυνατότητα παραγωγής συγκεντρωτικών στοιχείων επιτρέπει την εξαγωγή χρήσιμων συμπερασμάτων για την τεχνολογική δραστηριότητα και την καινοτομία.

Οι στατιστικές αναφορές μπορούν να αξιοποιηθούν για:

- Ακαδημαϊκή έρευνα
- Επιχειρηματική ανάλυση
- Στρατηγικό σχεδιασμό

#### 7.5 Πλεονεκτήματα του Συστήματος

Η υλοποίηση του συστήματος προσφέρει σημαντικά πλεονεκτήματα σε σχέση με την άμεση επεξεργασία XML αρχείων:

##### 1. Αυξημένη Απόδοση

Η χρήση σχεσιακής βάσης δεδομένων μειώνει σημαντικά τον χρόνο αναζήτησης και ανάκτησης πληροφορίας.

2. **Δομημένη Οργάνωση Δεδομένων**

Η κανονικοποιημένη σχεδίαση διασφαλίζει συνέπεια και αποφυγή πλεονασμού.

3. **Επεκτασιμότητα**

Η modular αρχιτεκτονική επιτρέπει την προσθήκη νέων λειτουργιών χωρίς ριζικές αλλαγές.

4. **Υποστήριξη Μεγάλου Όγκου Δεδομένων**

Το σύστημα σχεδιάστηκε ώστε να διαχειρίζεται αυξανόμενο αριθμό εγγραφών.

5. **Φιλική Διαδικτυακή Διεπαφή**

Η web-based υλοποίηση καθιστά την πρόσβαση στα δεδομένα εύκολη και άμεση.

Συνολικά, το σύστημα λειτουργεί ως ολοκληρωμένο εργαλείο διαχείρισης, ανάλυσης και ανάκτησης δεδομένων πατεντών, συνδυάζοντας τεχνολογίες βάσεων δεδομένων, επεξεργασίας XML και διαδικτυακής ανάπτυξης εφαρμογών

## ΚΕΦΑΛΑΙΟ 8

### Συμπεράσματα και Μελλοντική Εργασία

#### 8.1 Σύνοψη Επιτευγμάτων

Η παρούσα πτυχιακή εργασία επικεντρώθηκε στη σχεδίαση και ανάπτυξη ενός ολοκληρωμένου συστήματος διαχείρισης δεδομένων πατεντών μεγάλου όγκου, βασισμένου σε XML datasets της συλλογής World Patent Information (WPI). Η εργασία συνδύασε τεχνολογίες επεξεργασίας XML, σχεσιακών βάσεων δεδομένων και διαδικτυακών εφαρμογών, με στόχο τη δημιουργία μίας αποδοτικής και επεκτάσιμης πλατφόρμας αποθήκευσης, αναζήτησης και ανάλυσης δεδομένων πατεντών.

Στο πλαίσιο της υλοποίησης αναπτύχθηκε μηχανισμός μαζικής εισαγωγής XML αρχείων, ο οποίος υποστηρίζει αποσυμπίεση ZIP αρχείων, εντοπισμό XML εγγράφων, parsing των δεδομένων και μετασχηματισμό τους σε κανονικοποιημένη σχεσιακή βάση δεδομένων MySQL. Η επεξεργασία των XML δεδομένων πραγματοποιήθηκε μέσω modular backend αρχιτεκτονικής σε Python/Flask, επιτρέποντας τη διαχείριση σύνθετων XML structures και nested δεδομένων.

Η βάση δεδομένων σχεδιάστηκε με στόχο τη διατήρηση της αναφορικής ακεραιότητας και τη μείωση του πλεονασμού δεδομένων. Οι XML δομές αποσυντέθηκαν σε επιμέρους πίνακες, οι οποίοι συνδέονται μέσω primary και foreign keys, επιτρέποντας την αποτελεσματική αποθήκευση bibliographic δεδομένων, classifications, abstracts, descriptions, claims και στοιχείων προσώπων ή οργανισμών.

Παράλληλα, υλοποιήθηκε διαδικτυακή εφαρμογή βασισμένη στο Flask framework, μέσω της οποίας ο χρήστης μπορεί:

- να πραγματοποιεί upload XML ή ZIP αρχείων,
- να παρακολουθεί την πρόοδο επεξεργασίας,
- να αναζητά δεδομένα πατεντών,
- να εφαρμόζει φίλτρα αναζήτησης,
- να εκτελεί σύνθετα SQL queries,
- και να προβάλλει στατιστικά στοιχεία του dataset.

Ιδιαίτερη έμφαση δόθηκε στη βελτιστοποίηση της απόδοσης του συστήματος. Η χρήση σχεσιακής βάσης δεδομένων και indexing στρατηγικών επέτρεψε σημαντικά ταχύτερη ανάκτηση πληροφορίας σε σχέση με την άμεση επεξεργασία XML αρχείων. Επιπλέον, η modular σχεδίαση του backend καθιστά το σύστημα επεκτάσιμο και επιτρέπει την προσθήκη νέων λειτουργιών χωρίς σημαντικές αλλαγές στην υπάρχουσα αρχιτεκτονική.

Συνολικά, η εργασία απέδειξε ότι ο μετασχηματισμός μεγάλου όγκου ημιδομημένων XML δεδομένων σε κανονικοποιημένη σχεσιακή βάση αποτελεί αποτελεσματική προσέγγιση για τη διαχείριση patent datasets, βελτιώνοντας σημαντικά την αποδοτικότητα αναζήτησης, φιλτραρίσματος και στατιστικής ανάλυσης.

## 8.2 Αξιολόγηση Αρχιτεκτονικής Συστήματος

Η αρχιτεκτονική του συστήματος βασίστηκε σε μοντέλο τριών επιπέδων (three-tier architecture), διαχωρίζοντας το επίπεδο παρουσίασης, το επίπεδο εφαρμογής και το επίπεδο δεδομένων. Η επιλογή αυτή συνέβαλε στη modular οργάνωση του συστήματος και στη δυνατότητα ανεξάρτητης ανάπτυξης και συντήρησης κάθε επιπέδου.

Το backend υλοποιήθηκε με χρήση του Flask framework, το οποίο επιλέχθηκε λόγω της ευελιξίας, της απλότητας και της δυνατότητας modular ανάπτυξης. Η Flask επιτρέπει τη δημιουργία ελαφρών αλλά επεκτάσιμων web εφαρμογών και υποστηρίζει αποτελεσματικά τη διαχείριση HTTP requests, routing και database interactions. Επιπλέον, η modular διάσπαση της εφαρμογής σε επιμέρους modules (document, classification, claims, abstract, description, parties) βελτίωσε σημαντικά την αναγνωσιμότητα και τη συντηρησιμότητα του κώδικα.

Η επιλογή σχεσιακής βάσης δεδομένων MySQL κρίθηκε κατάλληλη λόγω:

- της ανάγκης διατήρησης σχέσεων μεταξύ δεδομένων,
- της ύπαρξης πολλών one-to-many συσχετίσεων,
- της ανάγκης εκτέλεσης σύνθετων SQL queries,
- και της υποστήριξης indexing μηχανισμών για βελτιστοποίηση αναζήτησης.

Η κανονικοποιημένη σχεδίαση της βάσης δεδομένων μείωσε τον πλεονασμό πληροφορίας και επέτρεψε τη διατήρηση αναφορικής ακεραιότητας μεταξύ των πινάκων. Παράλληλα, η χρήση βοηθητικών πινάκων για τιμές όπως country, lang, kind και status βελτίωσε τη συνοχή των δεδομένων και περιόρισε την αποθήκευση επαναλαμβανόμενων strings.

Σημαντικό πλεονέκτημα της αρχιτεκτονικής αποτελεί η δυνατότητα αποδοτικής ανάκτησης πληροφορίας μέσω indexing σε πεδία υψηλής συχνότητας αναζήτησης, όπως:

- publication number,
- classifications,
- publication date,
- inventors,
- applicants.

Η εφαρμογή χρησιμοποιεί επίσης background processing για την επεξεργασία XML αρχείων, επιτρέποντας στο web interface να παραμένει λειτουργικό κατά τη διάρκεια μεγάλων διαδικασιών εισαγωγής δεδομένων.

Παρότι η αρχιτεκτονική του συστήματος καλύπτει αποτελεσματικά τις απαιτήσεις της εργασίας, παρουσιάζει ορισμένους περιορισμούς ως προς την κλιμάκωση σε περιβάλλοντα πολύ μεγάλου όγκου δεδομένων, ιδιαίτερα όταν απαιτείται ταυτόχρονη επεξεργασία πολλών XML datasets ή distributed execution.

---

## 8.3 Περιορισμοί Υλοποίησης

Παρά τη λειτουργικότητα και την αποδοτικότητα του συστήματος, υπάρχουν ορισμένοι περιορισμοί που επηρεάζουν την απόδοση και τις δυνατότητες επέκτασής του.

Αρχικά, η επεξεργασία των XML αρχείων πραγματοποιείται κυρίως με single-threaded λογική. Αν και η χρήση background processing αποτρέπει το μπλοκάρισμα του web interface, η συνολική διαδικασία parsing και insertion παραμένει σειριακή. Σε περιβάλλοντα πολύ μεγάλου όγκου δεδομένων, η προσέγγιση αυτή αυξάνει τον συνολικό χρόνο επεξεργασίας.

Επιπλέον, η εφαρμογή βασίζεται σε κλασική σχεσιακή αναζήτηση μέσω SQL queries και δεν υποστηρίζει full-text indexing ή semantic search μηχανισμούς. Ως αποτέλεσμα, η αναζήτηση μεγάλων textual δεδομένων όπως descriptions και claims παρουσιάζει περιορισμούς ως προς την ευελιξία και την αποδοτικότητα.

Ένας ακόμη περιορισμός αφορά τη χρήση μνήμης κατά την επεξεργασία μεγάλων XML αρχείων. Η ανάλυση nested XML structures και η προσωρινή αποθήκευση δεδομένων αυξάνουν τη μνήμη που απαιτείται κατά το parsing, ιδιαίτερα σε datasets μεγάλου μεγέθους.

Παράλληλα, το σύστημα δεν περιλαμβάνει distributed processing ή cloud-based υποδομή. Η εκτέλεση πραγματοποιείται σε έναν μόνο server, γεγονός που περιορίζει τη δυνατότητα οριζόντιας κλιμάκωσης.

Επιπλέον, η εφαρμογή δεν υποστηρίζει:

- authentication και authorization μηχανισμούς,
- role-based access control,
- REST API endpoints για εξωτερικές εφαρμογές,
- ούτε αυτοματοποιημένους μηχανισμούς backup και replication της βάσης δεδομένων.

Τέλος, η αξιολόγηση του συστήματος πραγματοποιήθηκε σε συγκεκριμένο dataset και hardware περιβάλλον, γεγονός που σημαίνει ότι τα αποτελέσματα απόδοσης ενδέχεται να διαφοροποιούνται σε μεγαλύτερης κλίμακας υποδομές ή datasets.

---

## 8.4 Μελλοντικές Επεκτάσεις

Η αρχιτεκτονική και η modular σχεδίαση του συστήματος επιτρέπουν την περαιτέρω επέκταση και βελτίωσή του σε διάφορα επίπεδα.

Μία σημαντική μελλοντική επέκταση αφορά την ενσωμάτωση full-text search μηχανισμών, όπως Elasticsearch ή Apache Solr. Η χρήση τέτοιων τεχνολογιών θα επέτρεπε πολύ ταχύτερη και πιο ευέλικτη αναζήτηση σε μεγάλα textual δεδομένα, όπως descriptions και claims.

Παράλληλα, η ανάπτυξη REST API θα επέτρεπε τη διασύνδεση της πλατφόρμας με εξωτερικά πληροφοριακά συστήματα ή ερευνητικές εφαρμογές. Μέσω API endpoints, θα ήταν δυνατή η αυτοματοποιημένη ανάκτηση δεδομένων και η ενσωμάτωση του συστήματος σε μεγαλύτερες πλατφόρμες ανάλυσης πατεντών.

Σημαντική προοπτική αποτελεί επίσης η υλοποίηση asynchronous και distributed processing τεχνικών. Η χρήση multiprocessing, task queues ή distributed frameworks θα μπορούσε να μειώσει σημαντικά τον χρόνο επεξεργασίας μεγάλων XML datasets.

Μία ακόμη πιθανή επέκταση αφορά τη μεταφορά της εφαρμογής σε cloud περιβάλλον. Η χρήση containerization τεχνολογιών όπως Docker και orchestration εργαλείων όπως Kubernetes θα επέτρεπε καλύτερη κλιμάκωση, υψηλότερη διαθεσιμότητα και ευκολότερη διαχείριση υποδομών.

Επιπλέον, το σύστημα θα μπορούσε να επεκταθεί με τεχνικές τεχνητής νοημοσύνης και φυσικής γλώσσας (Natural Language Processing – NLP). Ενδεικτικά:

- αυτόματη κατηγοριοποίηση πατεντών,
- semantic similarity analysis,
- recommendation systems,
- keyword extraction,
- και vector-based search σε patent descriptions.

Τέλος, η προσθήκη προηγμένων εργαλείων οπτικοποίησης δεδομένων και interactive dashboards θα μπορούσε να βελτιώσει σημαντικά την εμπειρία χρήστη και να ενισχύσει τις δυνατότητες ερευνητικής και στατιστικής ανάλυσης του συστήματος.

## BIBΛΙΟΓΡΑΦΙΑ

- [1] World Intellectual Property Organization (WIPO), “WIPO Standard ST.36 – Processing of Patent Information Using XML,” WIPO Standards Documentation, Geneva.
- [2] W3C, “Extensible Markup Language (XML) 1.0,” World Wide Web Consortium. Available: <https://www.w3.org/XML/>
- [3] R. Elmasri and S. Navathe, Fundamentals of Database Systems, 7th ed. Pearson Education, 2016.
- [4] A. Silberschatz, H. Korth, and S. Sudarshan, Database System Concepts, 7th ed. McGraw-Hill, 2019.
- [5] C. J. Date, An Introduction to Database Systems, 8th ed. Addison-Wesley, 2003.
- [6] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge University Press, 2008.
- [7] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. Addison-Wesley, 2011.
- [8] M. Grinberg, Flask Web Development, 2nd ed. O’Reilly Media, 2018.
- [9] R. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” Doctoral Dissertation, University of California, Irvine, 2000.
- [10] R. Pressman and B. Maxim, Software Engineering: A Practitioner’s Approach, 9th ed. McGraw-Hill, 2019.
- [11] Oracle Corporation, “MySQL 8.0 Reference Manual.” Available: <https://dev.mysql.com/doc/>
- [12] H. Garcia-Molina, J. Ullman, and J. Widom, Database Systems: The Complete Book, 2nd ed. Pearson, 2008.
- [13] World Intellectual Property Organization (WIPO), Handbook on Industrial Property Information and Documentation. Geneva: WIPO.
- [14] A. Trippe, Patinformatics: Tasks to Tools. Elsevier, 2015.
- [15] E. R. Harold, Processing XML with Java. Addison-Wesley, 2002.
- [16] M. Kay, XSLT and XPath Programmer’s Reference. Wrox Press, 2004.
- [17] T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maler, and F. Yergeau, “Extensible Markup Language (XML),” World Wide Web Consortium (W3C), 2008.
- [18] D. Hunter, J. Rafter, J. Fawcett, and E. Van der Vlist, Beginning XML, 5th ed. Wiley Publishing, 2012.
- [19] R. Bourret, “XML and Databases,” XML.com Technical Articles, 2005.

- [20] S. Abiteboul, P. Buneman, and D. Suciu, *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann, 1999.
- [21] J. Clark and S. DeRose, “XML Path Language (XPath),” W3C Recommendation, 1999.
- [22] T. Berners-Lee, R. Fielding, and L. Masinter, “Uniform Resource Identifier (URI): Generic Syntax,” Internet Engineering Task Force (IETF), RFC 3986.
- [23] Oracle Corporation, “MySQL Performance Optimization and Indexing Documentation,” Oracle Docs.
- [24] P. DuBois, *MySQL Cookbook*, 4th ed. O’Reilly Media, 2021.
- [25] M. Fowler, *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2002.
- [26] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.
- [27] I. Sommerville, *Software Engineering*, 10th ed. Pearson Education, 2015.
- [28] D. Needleman, “Patent Information and Patent Documentation,” *World Patent Information Journal*, Elsevier.
- [29] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, 2011.