Information Retrieval

Εργασία: Μηχανή αναζήτησης τραγουδιών

Φάση 2: Υλοποίηση της μηχανής αναζήτησης

Εικόνα 1: Πηγή https://suhashbollu.com/projects

Link of GitHub Repository:

https://github.com/NikoletaBerati/Information-Retrieval-Project

Μέλη της ομάδας:

Κωνσταντίνα Στεργίου, ΑΜ: 4804

Νικολέτα Μπεράτη, ΑΜ: 4884

ΕΙΣΑΓΩΓΗ: Στόχος και λειτουργικότητα

Στόχος του συστήματος που υλοποιήσαμε είναι η δημιουργία μιας μηχανής αναζήτησης, η οποία αφορά πληροφορία σχετική με τραγούδια, καλλιτέχνες, άλμπουμ και στίχους. Ειδικότερα, ο χρήστης έχει την δυνατότητα να θέτει ερωτήματα που αφορούν την αναζήτηση οποιουδήποτε χαρακτηριστικού ενός τραγουδιού, ενώ το σύστημα επιστρέφει τα έγγραφα που είναι σχετικά με το ερώτημα που έθεσε. Τα έγγραφα αυτά παρουσιάζονται διατεταγμένα με βάση την συνάφειά τους με το ερώτημα που τέθηκε από τον χρήστη. Για την υλοποίηση, η βιβλιοθήκη που χρησιμοποιήθηκε για τις παραπάνω λειτουργίες είναι η Lucene, η οποία εξειδικεύεται στην ανάκτηση πληροφορίας.

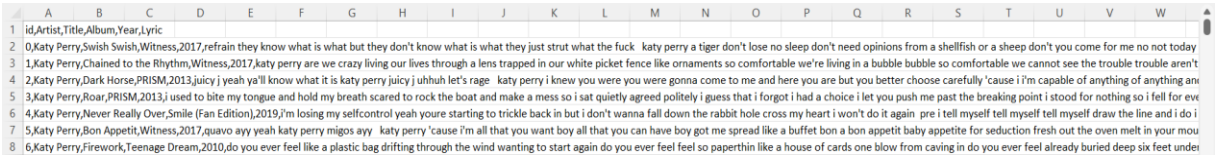
ΣΥΛΛΟΓΗ (corpus)

Η συλλογή των δεδομένων μας αποτελείται από ένα αρχείο σε μορφή csv, δηλαδή ένα αρχείο το οποίο περιέχει μια σειρά μεταβλητών με τιμές χωρισμένες με κόμμα. Η πληροφορία που διαθέτει η συλλογή μας είναι σχετική με τραγούδια. Περιέχει, δηλαδή, στοιχεία για τον καλλιτέχνη ενός τραγουδιού, τον τίτλο του, το άλμπουμ, το έτος κυκλοφορίας του τραγουδιού και τους στίχους του. **Πηγή συλλογής** των δεδομένων μας αποτελεί το ***Kaggle***, από το οποίο συλλέγουμε τις πληροφορίες που αναφέρθηκαν παραπάνω με σκοπό την δημιουργία της συλλογής μας.

Η συλλογή μας, λοιπόν, αποτελεί το αρχείο **data.csv** το οποίο περιέχει συνολικά 610 εγγραφές. Τα πεδία της συλλογής μας είναι τα εξής:

- id – το αναγνωριστικό του τραγουδιού
- Artist – ο καλλιτέχνης
- Title – ο τίτλος
- Album – το άλμπουμ στο οποίο ανήκει
- Year – η χρονολογία κυκλοφορίας
- Lyric – οι στίχοι του

Παραθέτουμε ένα στιγμιότυπο της συλλογής μας όπου αναπαρίστανται τα πεδία, καθώς και το link από το οποίο συλλέξαμε την πληροφορία:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	id,Artist,Title,Album,Year,Lyric																						
2	0,Katy Perry,Swish Swish,Witness,2017,refrain they know what is what but they don't know what is what they just strut what the fuck katy perry a tiger don't lose no sleep don't need opinions from a shellfish or a sheep don't you come for me no not today																						
3	1,Katy Perry,Chained to the Rhythm,Witness,2017,katy perry are we crazy living our lives through a lens trapped in our white picket fence like ornaments so comfortable we're living in a bubble bubble so comfortable we cannot see the trouble trouble aren't																						
4	2,Katy Perry,Dark Horse,PRISM,2013,juicy j yeah ya'll know what it is katy perry juicy j uhhuh let's rage katy perry i knew you were you were gonna come to me and here you are but you better choose carefully 'cause i i'm capable of anything of anything an																						
5	3,Katy Perry,Roar,PRISM,2013,i used to bite my tongue and hold my breath scared to rock the boat and make a mess so i sat quietly agreed politely i guess that i forgot i had a choice i let you push me past the breaking point i stood for nothing so i fell for eve																						
6	4,Katy Perry,Never Really Over,Smile (Fan Edition),2019,i'm losing my selfcontrol yeah youre starting to trickle back in but i don't wanna fall down the rabbit hole cross my heart i won't do it again pre i tell myself tell myself tell myself draw the line and i do i																						
7	5,Katy Perry,Bon Appetit,Witness,2017,quavo ayy yeah katy perry migos ayy katy perry 'cause i'm all that you want boy all that you can have boy got me spread like a buffet bon a bon appetit baby appetite for seduction fresh out the oven melt in your mou																						
8	6,Katy Perry,Firework,Teenage Dream,2010,do you ever feel like a plastic bag drifting through the wind wanting to start again do you ever feel feel so paperthin like a house of cards one blow from caving in do you ever feel already buried deep six feet under																						

Εικόνα 2: Ενδεικτικό στιγμιότυπο της συλλογής

Link για το Kaggle: https://www.kaggle.com/datasets/deepshah16/song-lyrics-dataset

ΑΝΑΛΥΣΗ ΚΕΙΜΕΝΟΥ ΚΑΙ ΚΑΤΑΣΚΕΥΗ ΕΥΡΕΤΗΡΙΟΥ

Στο σημείο αυτό, έχουμε δημιουργήσει την συλλογή μας η οποία αναλύθηκε παραπάνω, οπότε επόμενο βήμα είναι η κατασκευή του ευρετηρίου. Κάθε στήλη της συλλογής μας (id, Artist, Title, Album, Year, Lyric) αποτελεί ένα πεδίο (field), ενώ κάθε γραμμή της συλλογής αποτελεί ένα νέο έγγραφο (document). Η μονάδα εγγράφου, λοιπόν, αποτελείται από ολόκληρη την συλλογή μας, ενώ προκύπτει ότι τα έγγραφα που υπάρχουν μέσα στο ευρετήριο είναι σε πλήθος όσες είναι και οι καταχωρήσεις στην συλλογή μας.

Προκειμένου να υλοποιήσουμε προγραμματιστικά τα παραπάνω, χρησιμοποιούμε το API της Lucene ***org.apache.lucene.analysis*** για την επεξεργασία του κειμένου, το οποίο ορίζει έναν abstract αναλυτή για την μετατροπή κειμένου από έναν Reader σε ένα TokenStream. Ειδικότερα, επιλέγουμε τον *Standard Analyzer* ο οποίος μετατρέπει όλες τις λεκτικές μονάδες σε lowercase, αφαιρεί κοινές λέξεις-stop words και σημεία στίξης. Επιπλέον, χρησιμοποιούμε το API ***org.apache.lucene.document*** ώστε να έχουμε πρόσβαση στην κλάση Document, η οποία είναι απαραίτητη για την δημιουργία των εγγράφων και την ανάλυση σε πεδία (fields). Δημιουργείται, λοιπόν, μετά την εκτέλεση των παραπάνω ένα **δυναμικό και ανεστραμμένο ευρετήριο.**

Η κλάση η οποία είναι υπεύθυνη για την κατασκευή του ευρετηρίου και την δημιουργία των εγγράφων είναι η **Indexer.java,** η οποία περιέχει τις μεθόδους:

- **readData**: υπεύθυνη για την ανάγνωση του αρχείου που περιέχει τα δεδομένα (data.csv), η οποία χρησιμοποιεί ορισμένες βιβλιοθήκες ώστε να διευκολυνθεί αυτή η διαδικασία.
- **createIndex:** υπεύθυνη για την κατασκευή του ευρετηρίου και την προσθήκη των εγγράφων, σε περίπτωση που δεν έχει δημιουργηθεί ήδη.
- **createDocument**: υπεύθυνη για την δημιουργία των εγγράφων.

ΑΝΑΖΗΤΗΣΗ

Το σύστημά μας υποστηρίζει αναζήτηση εγγράφων:

- με **λέξεις κλειδιά** , δηλαδή ο χρήστης στο παράθυρο το οποίο εμφανίζεται πληκτρολογεί τις επιθυμητές λέξεις και το σύστημα θα πραγματοποιεί αναζήτηση σε όλα τα πεδία. Επιπλέον υποστηρίζει:
- **αναζήτηση πεδίου**, δηλαδή ο χρήστης θέτει ερωτήματα για τα οποία επιθυμεί αναζήτηση σε κάποιο συγκεκριμένο πεδίο. Η εναλλακτική αυτή αναζήτηση μπορεί να πραγματοποιηθεί σε οποιοδήποτε από τα πεδία, δηλαδή είτε στον καλλιτέχνη, το άλμπουμ, τον τίτλο του τραγουδιού, το έτος κυκλοφορίας και τους στίχους του με το πάτημα του αντίστοιχου κουμπιού (για παράδειγμα «Search by Title»).

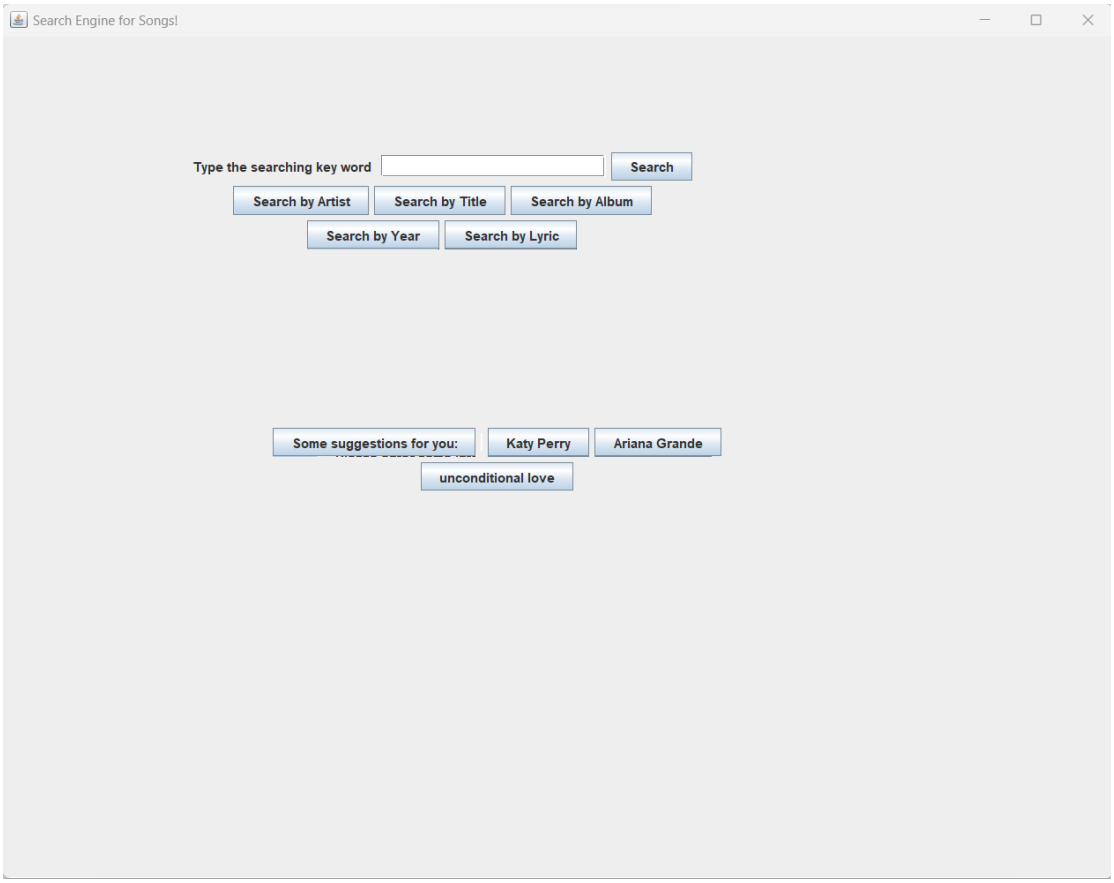
Ένα API που χρησιμοποιούμε για την λειτουργία της αναζήτησης είναι το ***org.apache.lucene.search*** το οποίο παρέχει δομές δεδομένων που αντιπροσωπεύουν τα ερωτήματα. Παρέχει τον IndexSearcher ο οποίος μετατρέπει τα ερωτήματα του χρήστη σε TopDocs και επιπλέον διαθέτει QueryParsers τα οποία χρησιμοποιούνται για την παραγωγή δομών ερωτημάτων από Strings.

Η κλάση η οποία είναι υπεύθυνη για την αναζήτηση των όρων είναι η **Searcher.java.** Η κλάση αυτή υλοποιεί τις μεθόδους searchBy..() οι οποίες αναζητούν το ερώτημα του χρήστη είτε σε κάποιο συγκεκριμένο πεδίο, όπως για παράδειγμα η μέθοδος searchByArtist η οποία αναζητά στο πεδίο «Artist», ή η searchByAll η οποία εκτελεί αναζήτηση σε όλα τα πεδία. Οι μέθοδοι αυτές επιστρέφουν τους δείκτες που δείχνουν στα έγγραφα τα οποία είναι συναφή με το ερώτημα που έθεσε ο χρήστης και

καλούνται από τις αντίστοιχες μεθόδους find..(), οι οποίες είναι υπεύθυνες για την επιστροφή λίστας που περιέχει όλα τα συναφή έγγραφα.

Παρακάτω στην *Εικόνα 2* παρουσιάζεται το παράθυρο το οποίο εμφανίζεται στον χρήστη προκειμένου να εκτελέσει την αναζήτηση που επιθυμεί. Πιο συγκεκριμένα, πληκτρολογεί τις λέξεις κλειδιά στο λευκό πλαίσιο και ανάλογα με το κουμπί που πατάει εκτελεί αναζήτηση σε όλα τα πεδία («Search») ή σε κάποιο συγκεκριμένο πεδίο (για παράδειγμα η αναζήτηση στον τίτλο γίνεται με το πάτημα του «Search By Title»).

Επιπρόσθετα, έχουμε υλοποιήσει την κλάση **History.java** η οποία είναι υπεύθυνη για την διατήρηση ενός αρχείου κειμένου στο οποίο αποθηκεύουμε όλα τα ερωτήματα που θέτει ο χρήστης, αποθηκεύοντας έτσι το ιστορικό αναζήτησης. Από αυτό το αρχείο κρατάμε τις λέξεις κλειδιά που εμφανίζονται τις περισσότερες φορές και τις παρουσιάζουμε στο παράθυρο αναζήτησης με την μορφή κουμπιών, ώστε ο χρήστης να μπορεί να αναζητήσει άμεσα τους προτεινόμενους όρους.



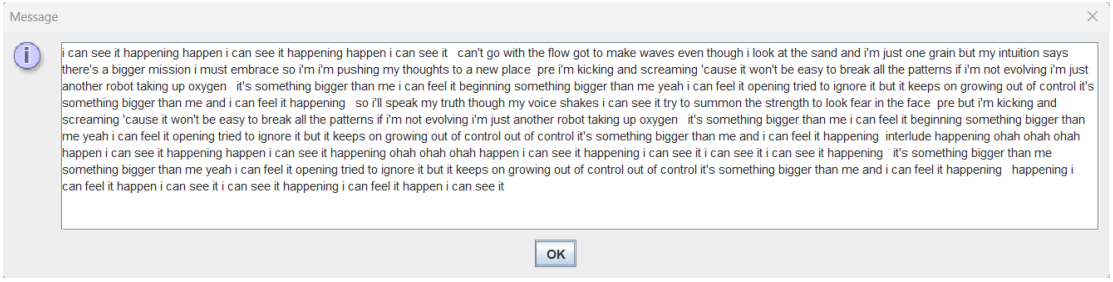
Εικόνα 3: Παράθυρο αναζήτησης

Η κλάση η οποία είναι υπεύθυνη για τη εμφάνιση του παραθύρου αναζήτησης είναι η **WindowSearchEngine.java ,** η οποία περιέχει την main συνάρτηση του προγράμματος. Για την προβολή του παραθύρου στο οποίο εμφανίζονται τα αποτελέσματα της αναζήτησης χρησιμοποιείται το interface της **Java Swing**, το οποίο είναι εύκολο στην χρήση και παρέχει πολλές βιβλιοθήκες.

ΠΑΡΟΥΣΙΑΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Το σύστημά μας παρουσιάζει τα αποτελέσματα διατεταγμένα με βάση την συνάφειά τους με το ερώτημα, χρησιμοποιώντας ως GUI το **Java Swing**. Ειδικότερα, παρουσιάζει τα αποτελέσματα σε ομάδες των 10, παρέχοντας την δυνατότητα στον χρήστη να προχωρήσει στα επόμενα 10 με το πάτημα του κατάλληλου κουμπιού («Next Page»), όπως και να πατήσει το αντίστοιχο κουμπί ώστε να γυρίσει στα προηγούμενα («Previous Page»). Μια επιπρόσθετη λειτουργία που εκτελεί η μηχανή αναζήτησης είναι η εμφάνιση των κελιών που περιέχουν τις λέξεις κλειδιά που αναζητά ο χρήστης με μπλε και τονισμένους χαρακτήρες. Όπως φαίνεται, λοιπόν, από την *Εικόνα 3* έχουμε επιλέξει το αποτέλεσμα της αναζήτησης να παρουσιάζεται σε ένα νέο παράθυρο με την μορφή ενός πίνακα όπου η πρώτη γραμμή αποτελεί τα πεδία στα οποία μπορεί να εκτελέσει αναζήτηση ο χρήστης. Δίνεται επίσης η δυνατότητα στον χρήστη να πατήσει με ένα κλικ σε οποιοδήποτε κελί επιθυμεί να δει πληροφορία η οποία δεν είναι ορατή στον πίνακα, όπως είναι οι στίχοι ενός τραγουδιού. Πατώντας, λοιπόν, πάνω στους στίχους ενός τραγουδιού, εμφανίζεται ένα νέο παράθυρο όπως αυτό στην *Εικόνα 4*:

Εικόνα 4: Παρουσίαση Αποτελεσμάτων



Εικόνα 5: Εμφάνιση όλων των στίχων με ένα κλικ

Μια επιπλέον λειτουργία που προσφέρει η μηχανή αναζήτησης που υλοποιήσαμε, είναι η ταξινόμηση των αποτελεσμάτων. Προσφέρεται η δυνατότητα να ταξινομηθεί ο πίνακας της *Εικόνας 3* με βάση τον καλλιτέχνη, πατώντας δηλαδή το κουμπί «**Sort by Artist**» το οποίο ομαδοποιεί τα αποτελέσματα της αναζήτησης των τραγουδιών με βάση τους καλλιτέχνες. Επίσης, πατώντας το κουμπί «**Sort by Title**» εμφανίζονται τα αποτελέσματα σε αλφαβητική σειρά με βάση τους τίτλους των τραγουδιών. Επιπρόσθετα, δίνεται η δυνατότητα ομαδοποίησης με βάση το άλμπουμ, πατώντας το κουμπί «**Sort by Album**». Τέλος, πατώντας το κουμπί «**Sort by Year**» τα αποτελέσματα παρουσιάζονται με βάση την χρονολογία κυκλοφορίας τους, ξεκινώντας από τα πιο πρόσφατα προς τα παλαιότερα.

Η κλάση η οποία είναι υπεύθυνη για την παρουσίαση των αποτελεσμάτων είναι η **WindowForResult.java.**

ΟΔΗΓΙΕΣ ΕΚΤΕΛΕΣΗΣ ΤΗΣ ΜΗΧΑΝΗΣ ΑΝΑΖΗΤΗΣΗΣ

Προκειμένου να χρησιμοποιήσει κανείς το σύστημά μας, είναι απαραίτητο να προστεθεί το Project στο Eclipse. Έπειτα, είναι απαραίτητο να προστεθούν στο build path του Project όλες οι βιβλιοθήκες οι οποίες βρίσκονται στον φάκελο με το όνομα jars. Ο φάκελος αυτός περιέχει βιβλιοθήκες της Lucene, βιβλιοθήκες αναγκαίες για την ανάγνωση του αρχείου .csv που περιέχει τα δεδομένα καθώς και βιβλιοθήκες που αφορούν το UI της Java Swing. Για να εκτελέσει κανείς την μηχανή αναζήτησης, τρέχει την **main** συνάρτηση του προγράμματος η οποία βρίσκεται στην κλάση **WindowSearchEngine.java**, με την προϋπόθεση ότι το ευρετήριο έχει ήδη δημιουργηθεί και υπάρχει στον φάκελο της εργασίας. Στην περίπτωση που δεν υπάρχει, είναι απαραίτητο να εκτελέσει ο χρήστης την κλάση **Indexer.java** η οποία είναι υπεύθυνη για την δημιουργία του ευρετηρίου και ύστερα να εκτελέσει την main της κλάσης **WindowSearchEngine.java.**