



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ
ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟ-
ΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗ-
ΡΟΦΟΡΙΚΗΣ

Εξυπηρετητής ανεύρεσης Video

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Βασιλική Δ. Μπιλιλή

Επιβλέπων : Γεώργιος Στασινόπουλος

Καθηγητής Ε.Μ.Π.



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΜΗΧΑΝΙΚΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Εξυπηρετητής ανεύρεσης Video

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Βασιλική Δ. Μπιλιλή

Επιβλέπων : Γεώργιος Στασινόπουλος
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

.....
Γ. Στασινόπουλος
Καθηγητής Ε.Μ.Π.

.....
Μ. Θεολόγου
Αν. καθηγητής Ε.Μ.Π.

.....
Β. Λούμος
Καθηγητής Ε.Μ.Π.

Αθήνα,

Ιούλιος

2008

.....

Βασιλική Δ. Μπιλιλή

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Βασιλική Δ. Μπιλιλή 2008

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Ο σκοπός αυτής της διπλωματικής εργασίας ήταν η δημιουργία μιας εφαρμογής internet για τη διαχείριση αρχείων βίντεο. Η εφαρμογή αυτή χωρίζεται σε τρεις επιμέρους ενότητες. Η πρώτη παρέχει τη δυνατότητα αποστολής βίντεο προς το σύστημα, η δεύτερη δίνει τη δυνατότητα στο χρήστη να δει πληροφορίες και εικόνες από τα υπάρχοντα βίντεο, ενώ στην τρίτη επιμέρους ενότητα της εφαρμογής ο χρήστης μπορεί να αναζητήσει κάποιο βίντεο σε όλη τη βάση του συστήματος. Πέρα από την εφαρμογή internet έχει υλοποιηθεί και μία εφαρμογή offline η οποία χρησιμοποιείται για την αποστολή μεγάλου μεγέθους βίντεο προς το σύστημα. Ο σχεδιασμός της εφαρμογής έγινε με τέτοιο τρόπο ώστε να είναι κατανοητή από όλες τις ομάδες πιθανών χρηστών.

Η καινοτομία της εφαρμογής αυτής έγκειται στο νέο τρόπο αναζήτησης βίντεο που χρησιμοποιείται. Αυτός βασίζεται στο αυτούσιο οπτικοακουστικό υλικό και πιο συγκεκριμένα στο υλικό των εικόνων από τις οποίες αποτελείται το βίντεο, αντικαθιστώντας τον παλιό τρόπο αναζήτησης με χρήση λέξεων κλειδιά που χρησιμοποιείται ευρέως από τις μεγαλύτερες μηχανές αναζήτησης βίντεο στο διαδίκτυο. Ο νέος αυτός τρόπος είναι εξίσου αποτελεσματικός τόσο στο χρόνο υλοποίησης της αναζήτησης όσο και στην ακρίβεια των αποτελεσμάτων.

Περνώντας στο κομμάτι της υλοποίησης, χρησιμοποιήθηκαν κυρίως εργαλεία της Microsoft. Για τη συγγραφή του κώδικα χρησιμοποιήθηκαν οι βιβλιοθήκες του asp.net Framework v.2 και η γλώσσα C#. Πιο συγκεκριμένα σχεδιάστηκαν επτά web-pages ενώ επιπλέον κώδικας έχει γραφεί για την υλοποίηση της offline εφαρμογής. Ακόμα, έχει σχεδιασθεί μία βάση δεδομένων για την οργάνωση των βίντεο που στέλνει ο χρήστης και την αποθήκευση των απαραίτητων για αυτά πληροφοριών. Η υλοποίηση της βάσης δεδομένων έγινε με τον SqlServer 2005.

Στην παρούσα εργασία περιγράφεται λεπτομερώς κάθε πτυχή της εφαρμογής ενώ δίνεται και πληθώρα εικόνων ώστε ο αναγνώστης να αποκτήσει μία πλήρη εικόνα για αυτήν. Ακόμα, γίνεται περιγραφή όλων των πινάκων και των πεδίων της βάσης δεδομένων που χρησιμοποιείται. Τέλος, το μεγαλύτερο κομμάτι της εργασίας αποτελείται από τον πηγαίο κώδικα της εφαρμογής ο οποίος έχει εμπλουτιστεί με λεπτομερή σχόλια που επεξηγούν όλες τις διαδικασίες που επιτελούνται.

Λέξεις Κλειδιά

Αλγόριθμος αναζήτησης video, video uploading, web-based application, asp.net, fingerprint

Abstract

The aim of this thesis was the creation of a web-based application for video processing. This application is actually divided into three different parts, upload – view – search. Upload application provides the chance of a fast and easy uploading of videos with size smaller than 16MB. View application provides a total view of the system's database where the user is able to see images of each stored video and even more download the video content. Search application provides a fast and accurate search of a small query clip into the whole videos' database. Except of that web-based application, it has also been created a windows application in offline mode where the user is able to upload a video larger than 16MB. The planning of this procedure became in that way in order to be comprehensible of every group of possible users.

The innovation of this application is the algorithm of searching procedure. This algorithm is a content-based video search method. It actually replaces the widely spread algorithm of searching based on keywords which is used from the most famous videos search engines in the web. It should be mentioned that this new way of searching provides equally accurate and fast results.

Asp.net libraries and the programming language C# are used for the creation of the application. In addition, it has been created a database system for the organization of the videos uploaded by the users. For this scope, Microsoft SqlServer 2005 it has been used.

In this thesis, we make a fully detailed analysis of the application and we provide a variety of images in order to a give a whole description of our system. In addition, we describe all the tables and the fields of the videos' database. Finally, it should be mentioned that a long part of this thesis is covered by the code which has been enriched with multiple comments so that all the procedures are explained.

Key words

Video search, video uploading, web-based application, asp.net, fingerprint

Ευχαριστίες

Με αφορμή τη συγκεκριμένη εργασία η οποία αποτελεί το επιστέγασμα των σπουδών μου στο τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών του Εθνικού Μετσόβιο Πολυτεχνείου θα ήθελα να ευχαριστήσω όλους εκείνους που με στήριξαν για να φτάσω ως το τέλος και να ολοκληρώσω έναν απαιτητικό κύκλο σπουδών.

Κατ' αρχάς, θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής εργασίας μου καθηγητή Γεώργιο Στασινόπουλο για την πολύτιμη βοήθειά του και την εμπιστοσύνη που μου επέδειξε καθ' όλη τη διάρκεια της εργασίας μου. Επίσης, είμαι ευγνώμων στον υποψήφιο διδάκτορα Σεραφείμ Παπαστέφανο για την πολύτιμη καθοδήγησή του και τη βοήθειά του σε κάθε φάση της δημιουργίας της εφαρμογής. Ωφείλω ακόμα ευχαριστίες στον ερευνητικό συνεργάτη του εργαστηρίου Φώτη Ανδρουτσόπουλο για τις πολύτιμες υποδείξεις και παρατηρήσεις του. Επίσης, ευχαριστώ θερμά τη φίλη και συμφοιτήτριά μου Κωσταντίνα Ν. Μαυροδημητράκη για την ηθική της συμπαράσταση τόσο κατά τη διάρκεια εκπόνησης της εργασίας μου όσο και κατά των τελευταίων μηνών των φοιτητικών μας χρόνων . Πάνω από όλα, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην οικογένεια μου και στις αδελφικές μου φίλες Διονυσία και Βασιλική Τσουκαλά-Τζίκα για την διαρκή υποστήριξή τους που μου επέτρεψε την επιτυχή διεκπεραίωση των σπουδών μου. Τέλος, θα ήθελα να ευχαριστήσω τους φίλους και τους συναδέλφους μου για τα όμορφα φοιτητικά χρόνια που περάσαμε μαζί.

Περιεχόμενα

1	ΕΙΣΑΓΩΓΗ	13
1.1	ΓΕΝΙΚΑ	13
1.2	ΤΙ ΥΛΟΠΟΙΗΘΗΚΕ ΣΤΗ ΔΙΠΛΩΜΑΤΙΚΗ	13
1.3	ΠΕΡΙΕΧΟΜΕΝΑ ΤΟΥ ΚΕΙΜΕΝΟΥ	14
2	ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ ΓΙΑ ΤΑ ΒΙΝΤΕΟ	17
2.1	ΚΩΔΙΚΟΠΟΙΗΣΗ ΒΙΝΤΕΟ	17
2.2	Η ΕΝΝΟΙΑ ΤΟΥ FINGERPRINT	17
3	Η ΕΦΑΡΜΟΓΗ ΔΙΑΧΕΙΡΙΣΗΣ ΒΙΝΤΕΟ	19
3.1	ΕΙΣΑΓΩΓΗ ΒΙΝΤΕΟ	19
3.1.1	<i>Εισαγωγή βίντεο μέσω web – εφαρμογής</i>	19
3.1.2	<i>Εισαγωγή βίντεο μέσω off-line tool</i>	26
3.2	ΠΑΡΟΥΣΙΑΣΗ ΒΙΝΤΕΟ	28
3.3	ΑΝΑΖΗΤΗΣΗ ΒΙΝΤΕΟ	32
3.3.1	<i>Εισαγωγή</i>	32
3.3.2	<i>Λίγα λόγια για τον τρόπο αναζήτησης</i>	32
3.3.3	<i>Περιγραφή</i>	33
3.3.4	<i>Προγραμματιστικό κομμάτι</i>	36
4	ΕΞΩΤΕΡΙΚΑ ΠΡΟΓΡΑΜΜΑΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	39
4.1	ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ	39
4.1.1	<i>Πίνακες</i>	39
4.1.2	<i>Σχεσιακά μοντέλα</i>	42
4.2	ΧΡΗΣΗ CSS	43
4.3	ΕΞΑΓΩΓΗ ΕΙΚΟΝΩΝ	44
5	Ο ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	47
5.1	ΒΟΗΘΗΤΙΚΕΣ ΚΛΑΣΕΙΣ	47
5.2	ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΤΩΝ WEB - PAGES	59
5.2.1	<i>Πηγαίος κώδικας της σελίδας «Default»</i>	59
5.2.2	<i>Πηγαίος κώδικας της σελίδας «Upload»</i>	61
5.2.3	<i>Πηγαίος κώδικας της σελίδας «Large videos»</i>	69
5.2.4	<i>Πηγαίος κώδικας της σελίδας «View»</i>	77
5.2.5	<i>Πηγαίος κώδικας για τη σελίδα «Details»</i>	81
5.2.6	<i>Πηγαίος κώδικας της σελίδας «Search»</i>	88
5.2.7	<i>Πηγαίος κώδικας της σελίδας «Results»</i>	100
5.3	ΠΡΟΔΙΑΓΡΑΦΕΣ ΤΟΥ SINORCA.CSS	104
5.4	ΠΗΓΑΙΟΣ ΚΩΔΙΚΑΣ ΓΙΑ WINDOWS APPLICATION UPLOADER	113
5.4.1	<i>Program.cs</i>	113
5.4.2	<i>Form1.cs</i>	113

1 Εισαγωγή

1.1 Γενικά

Η εξάπλωση του διαδικτύου στις μέρες μας είναι ραγδαία. Οι χρήστες του πολλαπλασιάζονται με θεαματικούς ρυθμούς και προέρχονται από όλο το φάσμα των κοινωνικών επιπέδων. Μετά την έκρηξη και του γρήγορου internet οι υπηρεσίες που αυτό προσφέρει είναι εντυπωσιακές. Οι χρήστες έχουν πρόσβαση σε πολύτιμες πηγές πληροφοριών για όλους τους τομείς της ανθρώπινης δραστηριότητας που εκπροσωπούνται στο διαδίκτυο από ένα τεράστιο πλήθος ιστοσελίδων. Αυτό έχει σαν αποτέλεσμα την αλματώδη αύξηση του διακινούμενου ψηφιακού υλικού. Ένα μεγάλο κομμάτι του υλικού αυτού είναι ο τομέας των video. Στις μέρες μας το βίντεο είναι ένα κομμάτι της καθημερινότητάς μας. Ενώ παλαιότερα η χρήση αλλά και η εύρεση οπτικοακουστικού υλικού δεν ήταν δημοφιλής, σήμερα μπορεί κανείς να γράψει ένα βίντεο ακόμα και με τη χρήση ενός κινητού τηλεφώνου και να το ανεβάσει στο διαδίκτυο. Έτσι, το διαδίκτυο έχει κατακλυστεί από πλούσιο οπτικοακουστικό υλικό στο οποίο οι χρήστες πρέπει να έχουν εύκολη και γρήγορη πρόσβαση. Ο μεγάλος όγκος των ψηφιακών δεδομένων video που είναι διαθέσιμα στο internet έχει οδηγήσει στην οργάνωσή τους σε βάσεις δεδομένων. Οστόσο οι διάφοροι τρόποι ανάκτησής τους αποτελούν ένα ιδιαίτερα ελκυστικό αντικείμενο προς έρευνα και έναν από τους στόχους της παρούσας διπλωματικής εργασίας.

1.2 Τι υλοποιήθηκε στη διπλωματική

Σε αυτή την εργασία έχει υλοποιηθεί μία εφαρμογή internet όπου οι χρήστες έχουν τη δυνατότητα να ανεβάσουν video σε κάποιον server, να δουν το σύνολο των video που ήδη υπάρχουν καθώς και να ψάξουν video μέσα σε μία μεγάλη βάση οπτικοακουστικού υλικού. Η πρωτοτυπία αυτής ακριβώς της εφαρμογής έγκειται στον τρόπο αναζήτησης του βίντεο μέσα στη βάση. Ο τυπικός τρόπος που ως σήμερα χρησιμοποιείται ευρέως από γνωστές μηχανές αναζήτησης βίντεο είναι η αναζήτηση με χρήση λέξεων-κλειδιά. Δηλαδή ο χρήστης καλείται να περιγράψει με λέξεις το βίντεο που αναζητά και το αποτέλεσμα ανακαλείται από τη βάση σύμφωνα με αυτές τις λέξεις. Στη δικιά μας εφαρμογή, ο τρόπος αναζήτησης είναι διαφορετικός. Η χρήση λέξεων ως μέσο αναζήτησης καταργείται και τη θέση της παίρνει η αναζήτηση με χρήση του ίδιου του βίντεο. Τα βίντεο χωρίζονται σε δύο κατηγορίες, μικρής και μεγάλης διάρκειας. Ο χρήστης καλείται να δώσει ένα βίντεο μικρής

διάρκειας και ο αλγόριθμος αναζήτησης βασιζόμενος στο αυτούσιο οπτικοακουστικό υλικό το εντοπίζει στα βίντεο μεγάλης διάρκειας που ο χρήστης έχει επιλέξει. Πρέπει να σημειωθεί ότι τα video τα οποία επεξεργαζόμαστε είναι ολιγόλεπτες διαφημίσεις, μικρά video clip ή και μικρές συνεντεύξεις ενώ στη βάση υπάρχει ένα πλούσιο απτικοακουστικό υλικό δέκα τεσσάρων περίπου ωρών με προγράμματα καναλιών, ταινίες και συνεντεύξεις. Έτσι, εξασφαλίζεται η απαραίτητη ποικιλομορφία του υλικού της εφαρμογής.

Για την υλοποίηση της εφαρμογής έχουν σχεδιαστεί επτά σελίδες aspx επικοινωνίας του χρήστη με το σύστημα. Μέσω αυτών ο χρήστης μπορεί να αποστείλει ολιγόλεπτα βίντεο (upload), να δει το υπάρχον στη βάση οπτικοακουστικό υλικό καθώς και κάποια χαρακτηριστικά του (view), να αναζητήσει μικρά βίντεο μέσα σε ολόκληρη τη βάση (search). Αξίζει να σημειωθεί ότι μετά από ρύθμιση της εφαρμογής, ο χρήστης μπορεί να ανεβάσει στο σύστημα, μέσω της εφαρμογής internet, βίντεο με μέγεθος μικρότερο των 16MB. Για το λόγο αυτό έχει υλοποιηθεί μία επιπλέον offline εφαρμογή με την οποία γίνεται εισαγωγή βίντεο μεγαλύτερων των 16MB. Ακόμα, ο χρήστης έχει τη δυνατότητα να δει εικόνες του υλικού που υπάρχει στη βάση. Αυτές δημιουργούνται από την εφαρμογή κατά τη διαδικασία του upload. Τέλος, δίνεται η δυνατότητα στο χρήστη να κατεβάσει το video που επιθυμεί και να το αποθηκεύσει στο δίσκο του.

Για τις ανάγκες της εφαρμογής, κρίθηκε απαραίτητη και η δημιουργία μίας βάσης δεδομένων. Εκεί αποθηκεύονται τα βίντεο που οι χρήστες αποστέλουν προς το σύστημα αλλά και όλες οι απαραίτητες για αυτά πληροφορίες. Η επικοινωνία με τη βάση επιτυγχάνεται με τις κατάλληλες εντολές μέσω του πηγαίου κώδικα.

Η υλοποίηση της εφαρμογής πραγματοποιήθηκε με χρήση τεχνολογιών της Microsoft. Το πρόγραμμα έχει γραφεί σε .NET Framework 2.0 χρησιμοποιώντας τη γλώσσα C# και το Microsoft Visual Studio 2005. Τέλος, για τη δημιουργία της βάσης χρησιμοποιήθηκε ο Microsoft Sql Server 2005.

1.3 Περιεχόμενα του κειμένου

Στο ανά χείρας κείμενο, περιέχονται εκτός από την εισαγωγή μία λεπτομερή περιγραφή και ανάλυση της εφαρμογής διαχείρισης video καθώς και μία λεπτομερής ανάλυση της υλοποίησης της εφαρμογής.

Πιο συγκεκριμένα, στο κεφάλαιο 2 περιγράφονται κάποιες βασικές έννοιες για τα βίντεο έτσι ώστε ο αναγνώστης να αποκτήσει την απαιτούμενη εξοικείωση με αυτές και να έχει μία πιο σφαιρική αντίληψη της εφαρμογής. Στο κεφάλαιο 3 περιγράφονται όλες οι δυνατό-

τητες της εφαρμογής αρχίζοντας από την επιλογή της εισαγωγής video, συνεχίζοντας με την παρουσίαση βίντεο και τελειώνοντας με την δυνατότητα της εύρεσης βίντεο μέσα στη βάση του συστήματος. Σε αυτό το κεφάλαιο περιγράφεται λεπτομερώς τόσο η διαδικασία που πρέπει να ακολουθήσει ο χρήστης της εφαρμογής στην εκάστοτε περίπτωση όσο και οι διεργασίες που εκτελούνται με βάση τον πηγαίο κώδικα της εφαρμογής ανάλογα κάθε φορά με την επιλογή του χρήστη. Τέλος, το κεφάλαιο αυτό έχει εμπλουτιστεί από μία ποικιλία αντιπροσωπευτικών εικόνων όπου φαίνονται όλα τα στάδια της εφαρμογής. Στο κεφάλαιο 4 περιγράφονται τα εξωτερικά προγράμματα που χρησιμοποιούνται από την εφαρμογή. Συγκεκριμένα, ένα μεγάλο κομμάτι του κεφαλαίου έχει αφιερωθεί στην ανάλυση των πινάκων και των πεδίων της βάσης δεδομένων του συστήματος ενώ δίνονται και τα σχεσιακά μοντέλα που υπάρχουν μεταξύ αυτών των πινάκων. Ακόμα, αναλύεται και η χρήση css για την καλύτερα διαμόρφωση των σελίδων html, ενώ γίνεται και λόγος για τον τρόπο εξαγωγής των εικόνων. Τέλος, στο κεφάλαιο 5 δίνεται ο πηγαίος κώδικας της εφαρμογής, εμπλουτισμένος με σχόλια ώστε να γίνεται πιο κατανοητός από τον αναγνώστη.

2 Εισαγωγικές έννοιες για τα βίντεο

Ακριβώς πριν την λεπτομερή περιγραφή της εφαρμογής κρίνεται απαραίτητη η επεξήγηση κάποιων βασικών εννοιών της επεξεργασίας βίντεο ούτως ώστε ο αναγνώστης να μπορέσει να καταλάβει πλήρως όλες τις διεργασίες που επιτελούνται κατά την εκτέλεσή της.

2.1 Κωδικοποίηση βίντεο

Στο παρελθόν, τα βίντεο ήταν σε αναλογική μορφή και βρίσκονταν αποθηκευμένα σε μαγνητικές κασέτες. Με το πέρασμα των χρόνων, η αναλογική μορφή καταργήθηκε και τη θέση της πήρε η ψηφιακή. Η ψηφιοποίηση όμως των βίντεο απαιτεί την κωδικοποίησή τους σύμφωνα με κάποιους κανόνες. Η κωδικοποίηση αυτή αφορά τόσο το κομμάτι της εικόνας όσο και το κομμάτι του ήχου και επιτελείται από προγράμματα γνωστά ως video και audio codecs αντίστοιχα. Κάποια γνωστά video codecs είναι τα mpeg1, mpeg2, h264, ενώ audio codecs τα mp3, aac, wav. Σε ένα απλό αρχείο βίντεο λοιπόν διακρίνουμε τρία πράγματα: τον video codec, τον audio codec και τον container. Ο container είναι αυτός που περιέχει ουσιαστικά τον video και τον audio codec. Γνωστοί containers είναι ο avi και ο mov.

2.2 Η έννοια του fingerprint

Τα βίντεο ουσιαστικά αποτελούνται από έναν αριθμό εικόνων ή αλλιώς frames οι οποίες και εμφανίζονται διαδοχικά κατά τη διάρκεια εκτέλεσής τους. Αυτές ουσιαστικά αποτελούν το αυτούσιο οπτικοακουστικό υλικό. Οι διαδοχικές εικόνες άλλοτε μοιάζουν μεταξύ τους και άλλοτε πάλι είναι τελείως διαφορετικές. Οι τελείως διαφορετικές διαδοχικές εικόνες παρατηρούνται όταν στο βίντεο υπάρχει αλλαγή σκηνής. Κάνοντας τώρα λόγο για το fingerprint ενός βίντεο αυτό είναι ότι δηλώνει και η λέξη, δηλαδή ένα αποτύπωμα του περιεχομένου του βίντεο. Πιο συγκεκριμένα, πρόκειται για ένα txt αρχείο που περιέχει αριθμούς αντιπροσωπευτικούς για το αυτούσιο οπτικοακουστικό υλικό. Οι αριθμοί αυτοί κυμαίνονται από 1 έως 100 και είναι τόσοι όσες και οι εικόνες του βίντεο, ένας αριθμός για κάθε εικόνα. Αυτοί ουσιαστικά δηλώνουν το ποσοστό της εικόνας, στην οποία αντιστοιχούν, που δεν είναι ίδιο με την προηγούμενη εικόνα. Όταν είναι αρκετά μεγάλοι πλησιάζοντας το 100 σημαίνει ότι μεγάλο ποσοστό της εικόνας είναι διαφορετικό από την προηγούμενη κάτι που σηματοδοτεί αλλαγή σκηνής. Από την άλλη μεριά, ένας μικρός αριθμός δηλώνει ότι μικρό ποσοστό της εικόνας στην οποία αντιστοιχεί έχει αλλάξει σε σχέση με την προηγούμενη αποκλείοντας

και το ενδεχόμενο αλλαγής σκηνής. Έτσι, γίνεται κατανοητό ότι ουσιαστικά το fingerprint αποτελεί μία «ταυτότητα» για το βίντεο αφού μας δίνει πληροφορίες για το περιεχόμενο του (video content).

Στο fingerprint στήριζεται και ο νέος τρόπος εύρεσης βίντεο σε μία βάση που παρουσιάζεται από την εφαρμογή μας. Για αυτό το σκοπό τα βίντεο κατηγοριοποιούνται σε δύο μεγάλες κατηγορίες τα βίντεο μικρής και μεγάλης διάρκειας. Ο αλγόριθμος αναζήτησης ουσιαστικά αναζητά το fingerprint του μικρού βίντεο μέσα σε αυτά των μεγάλων βίντεο που υπάρχουν στη βάση.

3 Η εφαρμογή διαχείρισης βίντεο

Η εφαρμογή αυτή όπως περιγράφηκε και προηγουμένως αποτελείται κυρίως από web-pages οι οποίες καθοδηγούν τον χρήστη ανάλογα με τη διαχείριση που αυτός επιθυμεί να πραγματοποιήσει. Αξίζει να σημειωθεί ότι οι σελίδες αυτές είναι προσεκτικά σχεδιασμένες ώστε να γίνονται πλήρως κατανοητές από όλες τις πιθανές ομάδες χρηστών. Ο χρήστης σε μία πρώτη επαφή βλέπει μια σελίδα καλωσορίσματος όπου του δίνεται η δυνατότητα να επιλέξει μία από τις προτεινόμενες ενέργειες upload-view-search ώστε να αρχίσει η πλοήγησή του. Παρακάτω δίνεται λεπτομερής περιγραφή για την εκάστοτε δυνατότητα της εφαρμογής.

3.1 Εισαγωγή βίντεο

Σε αυτό το κομμάτι της εργασίας περιγράφεται η διαδικασία αποστολής βίντεο προς το σύστημα. Η αποστολή αυτή μπορεί να γίνει είτε μέσω της εφαρμογής internet είτε μέσω της offline εφαρμογής που είναι απαραίτητη για την αποστολή μεγάλων video(>16MB). Ακόμα εάν ο χρήστης θέλει να ανεβάσει ένα μεγάλο βίντεο και διαθέτει έτοιμο το fingerprint, τότε του δίνεται η δυνατότητα να ανεβάσει μόνο αυτό αποφεύγοντας τη χρονοβόρα διαδικασία της δημιουργίας του μέσω της offline εφαρμογής.

3.1.1 Εισαγωγή βίντεο μέσω web – εφαρμογής

3.1.1.1 Εισαγωγή:

Η αποστολή βίντεο προς το σύστημα είναι εξαιρετικά απλή για τον χρήστη. Αυτός αρκεί να πατήσει το link upload στην αρχική σελίδα για να ξεκινήσει η διαδικασία.

3.1.1.2 Περιγραφή:

Πατώντας το link upload στην αρχική σελίδα ο χρήστης έρχεται σε επαφή με τη σελίδα εισαγωγής βίντεο. Εκεί καλείται μέσω του browse να επιλέξει το βίντεο που επιθυμεί να αποστείλει στο σύστημα. Κατόπιν, καλείται να δώσει κάποια στοιχεία για αυτό όπως ένα όνομα με το οποίο αυτό θα αποθηκευθεί στη βάση, κάποιες γενικές πληροφορίες που αυτός κρίνει απαραίτητο (τα ονόματα των ηθοποιών αν πρόκειται για ταινία, το προϊόν αν πρόκει-

ται για διαφήμιση, κ.τ.λ) τα οποία και αυτά αποθηκεύονται στη βάση. Ακόμα, πρέπει να επιλέξει μέσω μιας λίστας τον τύπο του βίντεο που θέλει να αποστείλει (ταινία, διαφήμιση, μουσικό video clip, default). Τέλος, ο χρήστης πρέπει να επιλέξει σε ποια κατηγορία ανήκει το βίντεο που ανεβάζει, δηλαδή εάν είναι βίντεο μεγάλης ή μικρής διάρκειας (query clip or database video) ή αν ανήκει και στις δύο παραπάνω κατηγορίες. Αξίζει να σημειωθεί ότι είναι απαραίτητο ο χρήστης να συμπληρώσει όλα τα πεδία εκτός από αυτό των γενικών πληροφοριών το οποίο επιτρέπεται να μείνει κενό. Παρακάτω φαίνεται και η ακριβής φόρμα της εφαρμογής με την οποία έρχεται σε επαφή ο χρήστης.

Upload a video

Search | View our database |

Insert a name for your clip: C:\Documents and Settings Browse... KeryKapnisto

Insert general information: kapnisto tyri

What kind of clip is it? advertisement

video to be used as a query clip

video to be entered into the database

If your clip is larger than 16MB... [Click Here](#)

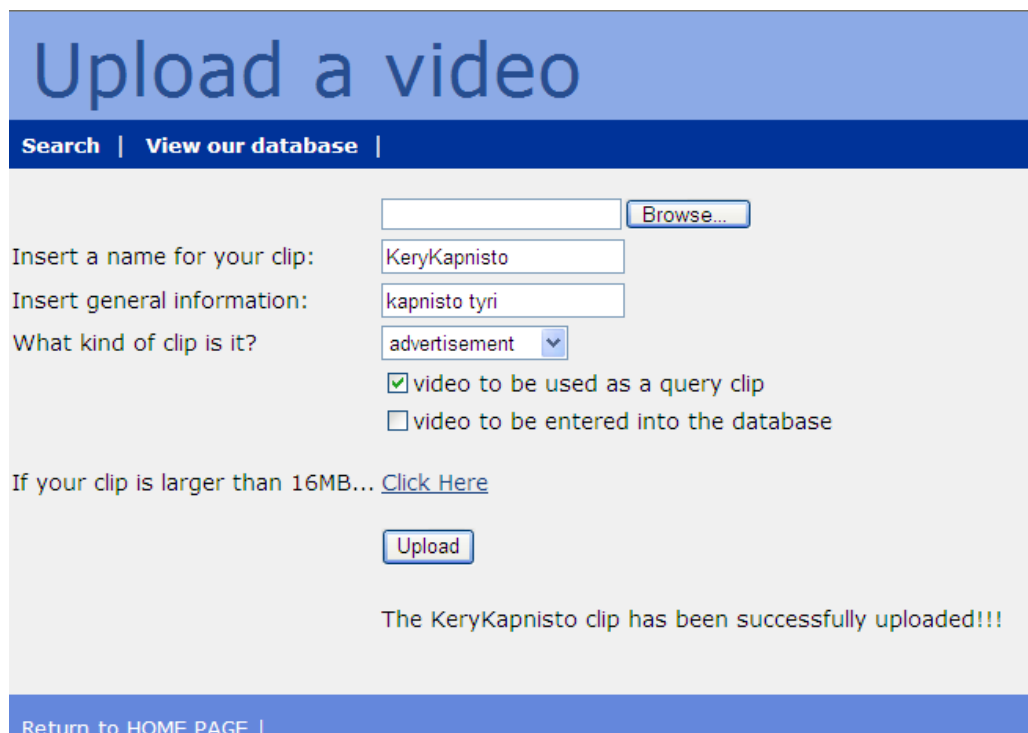
Upload

[Return to HOME PAGE |](#)

Εικόνα 1 : Video uploading

Πατώντας το button upload τα δεδομένα που έδωσε ο χρήστης αποθηκεύονται στη βάση ενώ το βίντεο υπόκειται μία διεργασία. Αρχικά, από αυτό εξάγουμε το txt αρχείο που περιγράψαμε και προηγουμένως και είναι γνωστό ως fingerprint του βίντεο. Επιπλέον, από το οπτικοακουστικό υλικό που ανεβάζει ο χρήστης εξάγονται κάποιες φωτογραφίες. Αυτές οι φωτογραφίες αποτελούν ουσιαστικά τα frames του βίντεο όπου υπάρχει αλλαγή σκηνής και εξάγονται σύμφωνα και με τα δεδομένα του fingerprint. Ο χρήστης μπορεί να δει τις φωτογραφίες αυτές στο κομμάτι της εφαρμογής –view- όπου ο ίδιος πλοηγείται στη βάση

και μπορεί να δει πληροφορίες για κάθε βίντεο που υπάρχει σε αυτή. Μόλις ολοκληρωθεί η διαδικασία του uploading στο χρήστη εμφανίζεται το μήνυμα που φαίνεται παρακάτω:



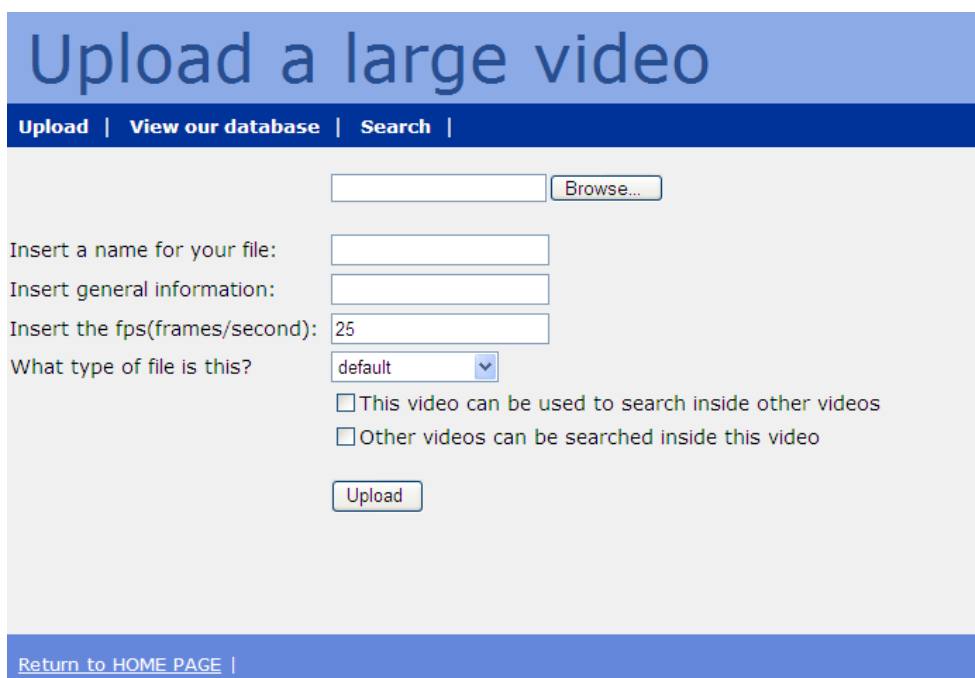
The screenshot shows a web interface for uploading a video. At the top, there is a blue header with the text "Upload a video". Below the header, there is a dark blue navigation bar with the text "Search | View our database |". The main content area is light gray and contains the following elements:

- A text input field with the value "KeryKapnisto" and a "Browse..." button to its right.
- The label "Insert a name for your clip:" followed by the input field.
- The label "Insert general information:" followed by an input field with the value "kapnisto tyri".
- The label "What kind of clip is it?" followed by a dropdown menu with "advertisement" selected.
- Two checkboxes: one checked and labeled "video to be used as a query clip", and one unchecked and labeled "video to be entered into the database".
- The text "If your clip is larger than 16MB... [Click Here](#)".
- An "Upload" button.
- A success message: "The KeryKapnisto clip has been successfully uploaded!!!".
- A footer bar with the text "Return to HOME PAGE |".

Εικόνα 2 : Video uploading-success message

Στο κάτω μέρος της σελίδας υπάρχει ένα link που καλεί τον χρήστη να το επισκεφθεί εάν θέλει να ανεβάσει μέσω της web-εφαρμογής ένα βίντεο μεγαλύτερο των 16MB. Πρόκειται ουσιαστικά για τη διαδικασία εισαγωγής του fingerprint μεγάλων βίντεο όταν ο χρήστης διαθέτει αυτό το αρχείο έτοιμο. Η εικόνα της εφαρμογής είναι παρόμοια με την προηγού-

μενη. Ο χρήστης καλείται να δώσει ακριβώς τα ίδια στοιχεία για το βίντεο που θέλει να ανεβάσει και κατόπιν να πατήσει το button upload.



The screenshot shows a web interface for uploading a large video. At the top, there is a blue header with the text "Upload a large video". Below the header is a navigation bar with links: "Upload", "View our database", and "Search". The main form area contains several input fields and a "Browse..." button for file selection. The form includes fields for "Insert a name for your file:", "Insert general information:", and "Insert the fps(frames/second):" (with the value "25" entered). There is also a dropdown menu for "What type of file is this?" set to "default". Two checkboxes are present: "This video can be used to search inside other videos" and "Other videos can be searched inside this video". An "Upload" button is located at the bottom of the form. At the very bottom of the page, there is a link: "Return to HOME PAGE |".

Εικόνα 3 : Video uploading of large videos

Διαφορετική οστόσο είναι η διαδικασία που ακολουθείται στη συνέχεια. Μετά το πάτημα του upload αφού το fingerprint είναι έτοιμο ουσιαστικά το μόνο που γίνεται είναι να αποθηκεύονται όλα τα δεδομένα που έδωσε ο χρήστης συμπεριλαμβανομένου και του fingerprint στη βάση δεδομένων. Πρέπει οστόσο να σημειωθεί ότι σε αυτή την περίπτωση δεν είναι δυνατόν να δημιουργηθούν φωτογραφίες, αφού δεν υπάρχει το περιεχόμενο του βίντεο παρά μόνο η «ταυτότητά» του, ούτε βέβαια μπορεί ο χρήστης να κατεβάσει το βίντεο στο δίσκο του. Το μόνο που μπορεί να αποθηκεύσει στο δίσκο του είναι το txt αρχείο που αποτελεί το fingerprint. Η χρησιμότητα αυτής της επιπλέον δυνατότητας online-upload γίνεται κυρίως αντιληπτή στο κομμάτι του search όπου ο χρήστης μπορεί να ψάξει μέσα σε ένα μεγάλο βίντεο ακόμα και αν αυτό δεν υπάρχει αυτόουσιο στη βάση του συστήματος. Αρχικά να υπάρχει μόνο το fingerprint του. Η φόρμα επικοινωνίας με το χρήστη φαίνεται παραπάνω.

3.1.1.3 Προγραμματιστικό κομμάτι:

Η διαδικασία του uploading έχει υλοποιηθεί με τις σελίδες upload.aspx, large_videos.aspx. Η πρώτη αναφέρεται στο online-uploading βίντεο με μέγεθος μικρότερο των 16MB και η δεύτερη στο uploading των fingerprint βίντεο μεγαλύτερων των 16MB.

❖ Upload.aspx

Σχεδιασμός της φόρμας:

Για το σχεδιασμό της φόρμας επικοινωνίας με τον χρήστη χρησιμοποιήθηκαν τα components του asp.net. Συγκεκριμένα, για την εισαγωγή των απαραίτητων πληροφοριών από το χρήστη χρησιμοποιήθηκαν textboxes και checkboxes που είναι ιδιαίτερα ευδιάκριτα στις παραπάνω εικόνες. Ακόμα, για την λίστα επιλογής του τύπου του βίντεο χρησιμοποιήθηκε η DropDownList η οποία πήρε τα στοιχεία της μετά από επικοινωνία με τη βάση μέσω της SqlDataSource. Επιπλέον, για τη δημιουργία του button Upload και των links χρησιμοποιήθηκαν τα components Button, Hyperlink, ενώ για τη δημιουργία του browsing χρησιμοποιήθηκε τα component File. Τέλος, πρέπει να σημειωθεί ότι με τη βοήθεια του component RequiredFieldValidator η σελίδα εμφανίζει μήνυμα λάθους όταν το πεδίο του ονόματος δεν έχει συμπληρωθεί όπως φαίνεται παρακάτω:

The screenshot shows a web form titled "Upload a video". At the top, there is a navigation bar with "Search" and "View our database" links. The form contains several input fields: a file path "C:\clips_saved\tellas-cov" with a "Browse..." button, a text box for "Insert a name for your clip:", another text box for "Insert general information:", and a dropdown menu for "What kind of clip is it?" set to "default". Below the dropdown are two checkboxes: "video to be used as a query clip" and "video to be entered into the database". A link "Click Here" is provided for clips larger than 16MB. An "Upload" button is present, and below it, a red error message reads "name is needed". At the bottom, there is a "Return to HOME PAGE" link.

Εικόνα 4 : Video uploading-error message

Επεξήγηση του κώδικα:

Με το πάτημα του button upload το βίντεο αποθηκεύεται και καλούνται οι κλάσεις CDoCrop, CInsert, CExtractImage οι οποίες περιέχουν τις απαιτούμενες συναρτήσεις.

CDoCrop:

Η CDoCrop περιέχει τις συναρτήσεις doMencoder(), doFfmpeg() οι οποίες ουσιαστικά καλούν τις αντίστοιχες external processes. Αρχικά η doMencoder μετατρέπει τον video codec του βίντεο που εισάγεται σε video codec H264. Από το output αυτής της εντολής παίρνουμε το fps που είναι μία χρήσιμη παράμετρος και δηλώνει τα frames/second που έχει το συγκεκριμένο βίντεο. Εάν αυτό δεν υπάρχει τότε το fps παίρνει την τιμή 25. Στη συνέχεια εκτελείται η doFfmpeg η οποία επιστρέφει ένα string που αποτελεί το fingerprint του βίντεο. Το output string αυτό διαμορφώνεται κατάλληλα ώστε οι αριθμοί να κυμαίνονται από το 1-100 και να αντιστοιχεί ένας σε κάθενα εικόνα. Αυτοί σώζονται σε ένα txt αρχείο. Ακόμα κάθε αριθμός του αρχείου μετατρέπεται σε byte και αποθηκεύεται σε έναν πίνακα ο οποίος και στέλνεται στη βάση δεδομένων.

CInsert:

Η CInsert περιέχει τη συνάρτηση Insert η οποία παίρνει μία πληθώρα παραμέτρων για κάθε βίντεο και τις στέλνει στη βάση δεδομένων. Οι παράμετροι περιγράφονται παρακάτω:

- το όνομα του αρχείου που έδωσε ο χρήστης
- το είδος του βίντεο, το είδος δηλαδή του βίντεο container(avi, mov κτλ)
- το fingerprint
- το fps
- η διάρκεια του βίντεο
- μία boolean μεταβλητή που δηλώνει εάν έχουν δημιουργηθεί ή όχι εικόνες για το βίντεο
- ένας αριθμός που δηλώνει τον τύπο του βίντεο, εάν πρόκειται δηλαδή για ταινία(2), διαφήμιση(3), music video clip(4) ή άλλο(1)
- οι πληροφορίες που έδωσε ο χρήστης
- ένας αριθμός που δηλώνει εάν το βίντεο ανήκει στην κατηγορία database video(1) ή στην κατηγορία query clip(2) ή και στις δύο κατηγορίες(3)
- ο αριθμός των εικόνων(frames) που διαθέτει το βίντεο.

Αυτές οι παράμετροι μέσω της συνάρτησης Insert στέλνονται στη βάση, μέσω της SqlConnection και της εντολής SqlCommand Insert η οποία και επιστρέφει το id όπου τις αποθήκευσε.

CExtractImage:

Η CExtractImage περιέχει τη συνάρτηση ExtractImage() η οποία σχηματίζει τις εικόνες των frames όπου υπάρχει αλλαγή σκηνής. Τα frames αυτά εντοπίζονται μέσα από την πληθώρα των frames που διαθέτει το βίντεο με βάση το fingerprint. Στην εφαρμογή έχει θεωρηθεί ότι κάποια εικόνα αποτελεί αλλαγή σκηνής όταν ο αντίστοιχος αριθμός της στο fingerprint είναι πάνω από 90. Αυτό σημαίνει ότι πάνω από το 90% της εικόνας είναι διαφορετικό από τις διπλάνες της κάτι που δηλώνει με βεβαιότητα ότι αυτή απεικονίζει μία αλλαγή σκηνής. Τέλος, πρέπει να σημειωθεί ότι πάνω στις εικόνες που εξάγονται αναγράφεται και ο ακριβής χρόνος του βίντεο όπου υπάρχει το συγκεκριμένο frame. Αυτός ο χρόνος υπολογίζεται από τη διαίρεση του αριθμού του frame με το fps(frames/second). Ο χρήστης έχει πρόσβαση στις εικόνες μέσω της διαδικασίας view όπου και του δίνεται η δυνατότητα να δει τα χαρακτηριστικά του εκάστοτε βίντεο που είναι αποθηκευμένο στη βάση.

Η CExtractImage περιέχει και τη συνάρτηση Rename_Image() η οποία μετονομάζει τις εικόνες ούτως ώστε αυτές να αποθηκεύονται στο σύστημά μας με μοναδικό όνομα που περιέχει και το id του βίντεο στο οποίο ανήκουν.

❖ **Large_videos.aspx**

Σχεδιασμός της φόρμας:

Η φόρμα που χρησιμοποιήθηκε για αυτή την δυνατότητα της εφαρμογής είναι ακριβώς η ίδια με την προηγούμενη οπότε η επεξήγηση του σχεδιασμού της θεωρείται πλεονάζων αφού έχει γίνει παραπάνω.

Επεξήγηση κώδικα:

Ο κώδικας που χρησιμοποιήθηκε εδώ είναι παρόμοιος με αυτόν της εφαρμογής Upload αλλά πολύ πιο απλός. Ουσιαστικά οι μόνες διεργασίες που γίνονται είναι η διαμόρφωση του fingerprint ώστε οι αριθμοί να κυμαίνονται από 1-100 και η μετατροπή και αποθήκευσή του σε πίνακα από bytes. Κατόπιν, καλείται η CInsert με τις παραμέτρους που γράφθηκαν παραπάνω οι οποίες και αποθηκεύονται στη βάση. Τέλος, πρέπει εδώ να σημειωθεί

ότι στην παράμετρο είδος του βίντεο εδώ αποθηκεύεται το ctxt ή txt που ουσιαστικά δείχνει ότι το παρόν βίντεο είναι διαθέσιμο μόνο ως fingerprint. Ευνόητο είναι ότι εδώ δεν καλείται η CExtractImage αφού δεν είναι δυνατόν να δημιουργηθούν φωτογραφίες τη στιγμή που δεν υπάρχει αυτούσιο το βίντεο.

3.1.2 Εισαγωγή βίντεο μέσω off-line tool

3.1.2.1 Εισαγωγή:

Η δυνατότητα της εισαγωγής βίντεο σε offline mode υλοποιήθηκε με μία windows εφαρμογή που ουσιαστικά αποτελεί μία επέκταση της εφαρμογής internet. Η δημιουργία της κρίθηκε απαραίτητη για την εισαγωγή βίντεο με απαγορευτικά μεγάλο μέγεθος (>16MB) για την εφαρμογή internet.

3.1.2.2 Περιγραφή:

Η φόρμα της εφαρμογής με την οποία έρχεται σε επαφή ο χρήστης είναι πολύ παρόμοια με αυτή της web-based εφαρμογής. Σε αυτήν υπάρχει το πεδίο όπου ο χρήστης επιλέγει το βίντεο μέσω browse από το σύστημά του και τα πεδία εισαγωγής των απαραίτητων πληροφοριών όπως το όνομα του βίντεο, οι επιπλέον πληροφορίες και το είδος του βίντεο (ταινία, διαφήμιση, music video clip, άλλο). Τέλος, ο χρήστης πρέπει να επιλέξει αν το βίντεο που εισάγει επιθυμεί να ανήκει στην κατηγορία query clip ή database video ή και στις δύο παραπάνω κατηγορίες. Πατώντας το button upload τα δεδομένα που εισήγαγε ο χρήστης αποθηκεύονται στη βάση ενώ το βίντεο υπόκειται ακριβώς την ίδια διεργασία που περιγράφηκε στην εφαρμογή online upload. Δηλαδή, με βάση το περιεχόμενο του βίντεο γίνεται η εξαγωγή του txt αρχείου(fingerprint). Μετέπειτα, με βάση τα δεδομένα του fingerprint γίνεται η εξαγωγή των φωτογραφιών που απεικονίζουν τις αλλαγές σκηνής και όλες οι πληροφορίες στέλνονται στη βάση δεδομένων. Στην παρακάτω εικόνα φαίνεται η φόρμα επικοινωνίας με τον χρήστη.

The screenshot shows a Windows application window titled "Form1". The window contains a form for video uploading. At the top right, there is a "Browse" button. Below it, there are three text input fields. The first is labeled "Insert a name:", the second "Insert general information:", and the third "What kind of clip is it?". The third field has a dropdown menu open, showing the following options: "default", "movie", "advertisement", and "music video clip". Below the dropdown, there are two checkboxes: "video to be used as a query clip" and "video to be entered into the database". At the bottom of the form, there is an "Upload" button.

Εικόνα 5 : Video uploading in offline mode

3.1.2.3 Προγραμματιστικό κομμάτι

Σχεδιασμός της φόρμας

Όπως παρατηρούμε η φόρμα αυτή είναι παρόμοια με αυτή που χρησιμοποιείται στο online uploading αφού ουσιαστικά εδώ υλοποιείται ακριβώς η ίδια διαδικασία. Τα components που χρησιμοποιήθηκαν σε αυτή τη windows εφαρμογή είναι textboxes, checkboxes, labels, buttons και μία listbox που συνδέεται μέσω της typeBindingSource με τη βάση δεδομένων.

Επεξήγηση κώδικα:

Ο κώδικας που χρησιμοποιήθηκε για την υλοποίηση αυτής της εφαρμογής είναι όμοιος με αυτόν που χρησιμοποιήθηκε στην εφαρμογή του online uploading οπότε ο αναγνώστης καλείται να κοιτάξει παραπάνω για επιπλέον λεπτομέρειες.

3.2 Παρουσίαση βίντεο

3.2.1.1 Εισαγωγή:

Η διαδικασία της παρουσίασης αποτελεί ένα πολύ ενδιαφέρον κομμάτι της εφαρμογής αφού στον χρήστη δίνεται η δυνατότητα να πλοηγηθεί μέσα στη βάση και να πάρει χρήσιμες πληροφορίες για τα βίντεο που αυτή διαθέτει. Μπορεί ακόμα να δει εικόνες από το κάθε βίντεο αν αυτές είναι διαθέσιμες. Τέλος, μπορεί να κατεβάσει το βίντεο στο δίσκο του.

3.2.1.2 Περιγραφή:

Πατώντας το button view στην αρχική σελίδα εμφανίζεται μια λίστα με το οπτικοακουστικό υλικό που διαθέτει η βάση του συστήματος. Τα βίντεο είναι χωρισμένα σε δύο λίστες, τα βίντεο μικρής διάρκειας(query clips) και αυτά μεγάλης διάρκειας(database videos). Για καθένα από αυτά είναι διαθέσιμα το id και το όνομά του. Δίπλα από κάθε καταχώρηση υπάρχει ένα link “details” που πατώντας το γίνεται μεταφορά σε μία καινούρια σελίδα με επιπλέον πληροφορίες για το συγκεκριμένο βίντεο. Ακόμα, στην κορυφή της δεύτερης λίστας αναγράφεται και η διάρκεια των μεγάλων βίντεο(database video) που υπάρχουν στη βάση. Παρακάτω διατίθεται και η εικόνα αυτή της εφαρμογής όπου φαίνονται οι παραπάνω λεπτομέρειες.

This is our database

[Upload](#) | [Search](#) |

Query Clips

id	name	
61	TellasCowboy	Details
64	KeryKapnisto	Details
65	CitroenC4	Details
66	AllStarSxoleio	Details
67	KinderBueno	Details
68	KitKat	Details
70	Kuprou	Details
72	VodafoneSpiti	Details
74	cineplus	Details

Database Clips (~5 hours and 37 minutes)

id	name	
47	MEGA.ctxt	Details
48	STAR.ctxt	Details
49	ANT1.ctxt	Details

[Return to HOME PAGE](#) |

Εικόνα 6 : View of videos' database

Πατώντας το link details σε ένα από τα βίντεο που υπάρχει αυτούσιο στο σύστημα, ο χρήστης μπορεί να δει επιπλέον πληροφορίες για το συγκεκριμένο βίντεο που είναι αποθηκευμένες στη βάση όπως το id, το όνομα με το οποίο είναι αποθηκευμένο στη βάση, την ημερομηνία όπου το βίντεο έγινε uploaded από κάποιον χρήστη, κάποιες επιπλέον πληροφορίες που έχει εισάγει ο χρήστης και έχουν αποθηκευθεί στη βάση, τον αριθμό των φωτογραφιών που είναι διαθέσιμες και τον τύπο του βίντεο. Επιπλέον, υπάρχουν άλλες δύο δυνατότητες που απεικονίζονται στο button Remove και στο link Download. Πατώντας το Remove ο χρήστης μπορεί να σβήσει το video από τη βάση ενώ πατώντας το Download μπορεί να κατεβάσει το βίντεο και να το αποθηκεύσει στο σύστημά του. Ακόμα, σε αυτό το κομμάτι της εφαρμογής φαίνονται και οι φωτογραφίες που έχουν δημιουργηθεί κατά τη διαδικασία του uploading και απεικονίζουν τις αλλαγές σκηνών. Παρακάτω, δίνεται και η εικόνα της εφαρμογής.

View Details....!!!

[Upload](#) | [View our database](#) | [Search](#) |

id	name	date	duration(sec)	clip_info	clip_picture_number	video_type	
65	small_clip	4/4/2008 4:03:08	μμ 3,88		2	avi	<input type="button" value="Remove"/> Download



[Go Back](#) |


Εικόνα 7 : View details of the selected clip

Αν τώρα ο χρήστης πατήσει details σε ένα από τα βίντεο που δεν υπάρχει αυτούσιο στο σύστημα αλλά κάποιος χρήστης έχει ανεβάσει μέσω της σελίδας large_videos.aspx το fingerprint αυτού, τότε θα δει μια εικόνα όπως η παρακάτω:

View Details....!!!

[Upload](#) | [View our database](#) | [Search](#) |

id	name	date	duration(sec)	clip_info	clip_picture_number	video_type	
11	MEGA.ts.ctxt	4/4/2008 3:44:53	μμ 2297,16		0	ctxt	<input type="button" value="Remove"/>



[Go Back](#) |

Εικόνα 8 : View details of the selected clip-fingerprint

Σε αυτή μπορούμε να διακρίνουμε σχεδόν τα ίδια πεδία με την προηγούμενη περίπτωση. Η μόνη διαφορά είναι ότι εδώ δεν υπάρχει η δυνατότητα download αφού στο σύστημα δεν υπάρχει το αυτούσιο οπτικοακουστικό υλικό. Ακόμα, σε αυτήν την περίπτωση δεν έχουν δημιουργηθεί φωτογραφίες ενώ στη θέση τους φαίνεται μια δεδομένη φωτογραφία όπως διακρίνεται και παραπάνω. Τέλος, το γεγονός ότι το βίντεο δεν υπάρχει αυτούσιο στη βάση δηλώνεται και στο πεδίο του video_type όπου εμφανίζεται η συντομογραφία cxtx.

3.2.1.3 Προγραμματιστικό κομμάτι

Σχεδιασμός της φόρμας

Το κομμάτι της παρουσίασης του οπτικοακουστικού υλικού έχει ουσιαστικά υλοποιηθεί με δύο σελίδες.aspx την View.aspx και την Details.aspx. Στην σελίδα View.aspx για την παρουσίαση των βίντεο της εκάστοτε κατηγορίας έχει χρησιμοποιηθεί το component του asp.net gridview. Τα περιεχόμενα αυτού (id, name) διατείνονται έπειτα από επικοινωνία με τη βάση δεδομένων ενώ δίπλα από κάθε τέτοιο ζευγάρι έχει χρησιμοποιηθεί και το πεδίο hyperlink με τίτλο details. Στη σελίδα Details.aspx χρησιμοποιήθηκε επίσης το component του asp.net gridview με περιεχόμενα το id του βίντεο, το όνομα του βίντεο, την ημερομηνία εισαγωγής του στο σύστημα, τη διάρκεια του, τον αριθμό των φωτογραφιών που διαθέτει και τον τύπο του καθώς και κάποιες επιπλέον πληροφορίες που έχει εισάγει ο χρήστης.

Επεξήγηση κώδικα

Ο κώδικας της σελίδας View είναι πολύ απλός αφού το μόνο που πραγματοποιείται είναι μία επικοινωνία με τη βάση, μέσω της SqlConnection, από όπου ανασύρεται η συνολική διάρκεια των database video. Κατόπιν, ο χρόνος μετατρέπεται σε ώρες και λεπτά, ώστε να γίνεται πιο αντιληπτός από το χρήστη, και αναγράφεται στην αντίστοιχη label ούτως ώστε να εμφανιστεί στην οθόνη.

Ο κώδικας της σελίδας Details είναι λίγο πιο περίπλοκος από τον προηγούμενο. Εδώ, αρχικά διαβάζονται τα περιεχόμενα του gridview για το βίντεο που ο χρήστης έχει επιλέξει στη σελίδα View. Στη συνέχεια, δημιουργείται δυναμικά το hyperlink Download και μέσω του NavigateUrl υποδεικνύεται το μονοπάτι(path) όπου το βίντεο είναι αποθηκευμένο στο σύστημα. Ακόμα, δημιουργούνται δυναμικά τόσα components images όσες και οι εικόνες που έχουν δημιουργηθεί για το αντίστοιχο βίντεο. Σε αυτά τα components φορτώνονται οι εικόνες που έχουν δημιουργηθεί κατά τη διάρκεια του upload. Εάν, το βίντεο που έχει επι-

λέξει ο χρήστης δεν υπάρχει αυτούσιο αλλά μόνο το txt αρχείο του, τότε στο component φορτώνεται μία standard εικόνα που υπάρχει στο σύστημα. Στη συνέχεια, στον πηγαίο κώδικα υπάρχει η συνάρτηση GridView1_RowCommand() που εκτελείται όταν ο χρήστης πατήσει το button Remove. Αρχικά, πραγματοποιείται επικοινωνία με τη βάση, μέσω της SqlConnection, και με τη χρήση της SqlCommand DELETE σβήνεται από τη βάση τόσο το επιλεγμένο βίντεο όσο και το μονοπάτι(path) όπου είναι αποθηκευμένη η εκάστοτε εικόνα του. Στη συνέχεια, σβήνονται τα αντίστοιχα αρχεία από το σύστημα. Τέλος, μέσω της Response.Redirect() ο χρήστης μεταφέρεται στην προηγούμενη σελίδα View.aspx όπου βλέπει ότι δεν υπάρχει πια καταχώρηση για το βίντεο που επέλεξε να σβήσει από τη βάση.

3.3 Αναζήτηση βίντεο

3.3.1 Εισαγωγή

Η τελευταία αλλά και πιο ενδιαφέρουσα δυνατότητα της εφαρμογής είναι η δυνατότητα αναζήτησης ενός ή και περισσότερων clip μέσα σε μία μεγάλη βάση από βίντεο. Η αναζήτηση όπως περιγράφηκε παραπάνω γίνεται με βάση το αυτούσιο περιεχόμενο του video και όχι με χρήση λέξεων κλειδιά κάτι που αποτελεί και την καινοτομία της εφαρμογής. Στη συνέχεια, στο χρήστη εμφανίζονται τα αποτελέσματα όπου για κάθε clip αναζήτησης αναφέρονται τα πιθανά βίντεο όπου βρέθηκε και η χρονική στιγμή όπου βρέθηκε μέσα σε αυτά. Για κάθε αποτέλεσμα δίνεται και ένα ποσοστό λάθους αλλά και ο συνολικός χρόνος αναζήτησης. Παρακάτω γίνεται λεπτομερής περιγραφή της τρίτης και τελευταίας δυνατότητας της εφαρμογής τόσο στο επίπεδο χρήσης όσο και στο επίπεδο υλοποίησης.

3.3.2 Λίγα λόγια για τον τρόπο αναζήτησης

Στις γνωστές μηχανές αναζήτησης τόσο οι αλγόριθμοι αναζήτησης όσο και οι βάσεις δεδομένων είναι οργανωμένες με βάση λέξεις «κλειδιά» οι οποίες περιγράφουν το εκάστοτε βίντεο. Ο χρήστης καλείται να περιγράψει με λέξεις το βίντεο που αναζητά. Οι αλγόριθμοι αναζήτησης χρησιμοποιούν αυτές τις λέξεις για να επιλέξουν τα πιθανά βίντεο προσπαθώντας να τις «ταιριάξουν» με τις λέξεις «κλειδιά» που περιγράφουν τα βίντεο στη βάση.

Σε αυτή την εφαρμογή χρησιμοποιείται ένα διαφορετικό μοντέλο αναζήτησης. Τα βίντεο χωρίζονται σε δύο μεγάλες κατηγορίες, τα μικρής διάρκειας βίντεο και τα μεγάλης διάρκειας βίντεο. Ο χρήστης έχει τη δυνατότητα να επιλέξει ένα ή περισσότερα μικρής διάρκειας βίντεο και να βρει σε ποιο από τα μεγάλης διάρκειας βίντεο αυτά βρίσκονται. Ουσιαστικά, ξεφεύγουμε από τον τρόπο της λεκτικής περιγραφής και δοκιμάζουμε τον τρόπο αναζήτησης ενός βίντεο μέσα σε πολλά άλλα, δίνοντας αυτούσιο το περιεχόμενό του. Όλος ο αλγόριθμος αναζήτησης στηρίζεται στο txt αρχείο(fingerprint) που υπάρχει στη βάση για κάθε βίντεο είτε μικρής είτε μεγάλης διάρκειας και έχει δημιουργηθεί κατά τη διάρκεια του uploading. Ουσιαστικά, γίνεται ένα ταιριασμα των fingerprints των βίντεο προς αναζήτηση με τα fingerprints των μεγάλης διάρκειας βίντεο. Περισσότερες λεπτομέρειες για τον αλγόριθμο θα δοθούν στην περιγραφή του προγραμματιστικού μέρους αυτής της δυνατότητας της εφαρμογής.

3.3.3 Περιγραφή

Ο χρήστης πατώντας το button search στην αρχική σελίδα της εφαρμογής έρχεται σε επαφή με την εικόνα που δίνεται παρακάτω. Εδώ, καλείται να επιλέξει κάποια από τα βίντεο της πρώτης στήλης, που αποτελούν τα μικρής διάρκειας βίντεο, που επιθυμεί να εντοπίσει σε ποια από τα βίντεο της άλλης στήλης(μεγάλης διάρκειας βίντεο) βρίσκονται. Για παράδειγμα, στην παρακάτω εικόνα φαίνεται ότι ο χρήστης έχει επιλέξει να ψάξει εάν τα video Tel-lasCowboy, KeryKapnisto, KitKat βρίσκονται μέσα στο περιεχόμενο των βίντεο MEGA, STAR, ANT1. Έτσι, θα καταφέρει γρήγορα και εύκολα να μάθει εάν κάποιο από τα επιλεγμένα query clips βρίσκεται μέσα στα βίντεο των τριών μεγάλων καναλιών. Πρέπει να σημειωθεί εδώ ότι τα βίντεο που βρίσκονται στις στήλες είναι βίντεο που υπάρχουν ήδη στη βάση του συστήματος. Οστόσο, δίνεται η δυνατότητα στον χρήστη να αναζητήσει ένα βίντεο, που δεν υπάρχει ήδη στο σύστημα, μέσω της επιλογής Browse που φαίνεται στο κάτω μέρος της εικόνας.

Search...

[Upload](#) | [View our database](#) |

Select query clips to be used for search

Select	id	name
<input checked="" type="checkbox"/>	61	TellasCowboy
<input checked="" type="checkbox"/>	64	KeryKapnisto
<input type="checkbox"/>	65	CitroenC4
<input type="checkbox"/>	66	AllStarSxoleio
<input type="checkbox"/>	67	KinderBueno
<input checked="" type="checkbox"/>	68	KitKat
<input type="checkbox"/>	70	Kuprou
<input type="checkbox"/>	72	VodafonSpiti

Select database videos to be searched

Select	id	name
<input checked="" type="checkbox"/>	47	MEGA.ctxt
<input checked="" type="checkbox"/>	48	STAR.ctxt
<input checked="" type="checkbox"/>	49	ANT1.ctxt

Do you want to search for another clip?

C:\Documents and Setting

[Return to HOME PAGE](#) |

Εικόνα 9 : Video search

Πατώντας το button Search εμφανίζεται μια άλλη σελίδα που περιέχει τα αποτελέσματα της αναζήτησης.

JavaScript Switch Statement
http://www.w3schools.com/js/js_switch.asp

The results

[Upload](#) | [Search](#) | [View our database](#) |

Query clip: TellasCowboy

Found at: MEGA.ctxt at : 00:04:38 with loss: 4
Found at: STAR.ctxt at : 01:07:35 with loss: 7
Search time: 843,75 milliseconds

Query clip: KeryKapnisto

Not found
Search time: 343,75 milliseconds

Query clip: KitKat

Found at: MEGA.ctxt at : 00:30:53 with loss: 3
Search time: 328,125 milliseconds

Query clip: xrusos-odhgos.ts.avi

Not found
Search time: 328,125 milliseconds

Indexing time = 6875 milliseconds.

[Go back](#) |

Εικόνα 10 : Results of videos' search procedure

Παρατηρούμε ότι για κάθε query clip που επέλεξε ο χρήστης εμφανίζονται τόσο τα ονόματα των βίντεο όσο και ο ακριβής χρόνος όπου αυτό βρέθηκε. Επίσης, στα αποτελέσματα φαίνεται και η συνολική διάρκεια της αναζήτησης για κάθε query clip αλλά και ένας αριθμός πιθανού σφάλματος που ουσιαστικά είναι ένα ποσοτικό μέγεθος που δείχνει την απόκλιση με την οποία βρέθηκε το query clip στο αντίστοιχο database video. Φυσικά, είναι πιθανό να μην βρεθεί το query clip σε κανένα βίντεο οπότε και αναφέρεται το μήνυμα “Not Found”. Το τελευταίο query clip xrusos-odhgos.ts.avi δεν υπήρχε στη βάση του συστήματος απλά ο χρήστης το επέλεξε προς αναζήτηση μέσω του Browse, κάτι που φαίνεται και στην προηγούμενη εικόνα. Έτσι, η παράμετρος «indexing time» κάτω από αυτό αναφέρεται στην ώρα που χρειάστηκε ώστε να δημιουργηθεί το fingerprint του βίντεο, ενώ η παράμετρος Search time αναφέρεται κατά τα γνωστά στο χρόνο αναζήτησής του ανάμεσα σε αυτά των μεγάλων βίντεο.

3.3.4 Προγραμματιστικό κομμάτι

Σχεδιασμός της φόρμας

Για την υλοποίηση της διαδικασίας του Search σχεδιάστηκαν δύο σελίδες η Search.aspx και η Results.aspx. Για τον σχεδιασμό τους χρησιμοποιήθηκαν τα components που διαθέτει ο asp.net.

Πιο συγκεκριμένα, για την δημιουργία των λιστών της πρώτης σελίδας χρησιμοποιήθηκαν δύο components gridView τα οποία παίρνουν τα στοιχεία τους μέσω των SqlDataSource που επικοινωνούν με τη βάση και ανασύρουν από αυτή τα απαραίτητα κάθε φορά πεδία. Για παράδειγμα, η SqlDataSource που είναι συνδεδεμένη με την πρώτη λίστα ανασύρει από τη βάση το id και το name των βίντεο που είναι καταχωρημένα ως query clips ενώ η δεύτερη αυτά των βίντεο που είναι καταχωρημένα ως database videos. Επιπλέον, έχουν χρησιμοποιηθεί CheckBoxes ώστε ο χρήστης να πραγματοποιεί τις επιλογές του ενώ έχει χρησιμοποιηθεί και το component File ούτως ώστε ο χρήστης να μπορέσει να επιλέξει και από το σύστημά του βίντεο προς αναζήτηση χωρίς αυτό να ήδη καταχωρημένο στη βάση. Τέλος, για τη δημιουργία των επικεφαλίδων και του κουμπιού χρησιμοποιήθηκαν Labels και Button αντίστοιχα.

Όσο αναφορά τη δημιουργία της δεύτερης σελίδας, εδώ έχει χρησιμοποιηθεί μόνο μια Label πάνω στην οποία γράφονται τα αποτελέσματα και έτσι εμφανίζονται στην οθόνη του χρήστη.

Επεξήγηση του κώδικα

Ο χρήστης αφού επιλέξει τα query clips και τα database videos που θέλει να έχει η αναζήτησή του πατά το button Search. Με το πάτημα αυτό ενεργοποιούνται διεργασίες προκειμένου να υλοποιήσουν το αίτημά του. Αρχικά, τα δεδομένα των επιλογών του χρήστη αποθηκεύονται σε λίστες ούτως ώστε σταδιακά να επεξεργαστούν. Για την ακρίβεια, σε λίστες αποθηκεύονται τα id των query clips και των database videos, ενώ για καθένα από αυτά μετά από επικοινωνία με τη βάση ανασύρεται το όνομα του και το αρχείο που αποτελεί το fingerprint του. Αυτά αποθηκεύονται επίσης σε λίστες για να χρησιμοποιηθούν στην αναζήτηση. Σε περίπτωση που ο χρήστης έχει επιλέξει να αναζητήσει βίντεο το οποίο δεν είναι ήδη καταχωρημένο στη βάση, είναι απαραίτητο να γίνει πρώτα η επεξεργασία του. Το βίντεο αρχικά μετατρέπεται σε H264 με την συνάρτηση doMencoder() της κλάσης CDoCrop ενώ στα συνέχεια καλείται η συνάρτηση doFfmpeg() της ίδιας κλάσης και το αποτέλεσμα αυτής διαμορφώνεται κατάλληλα ούτως ώστε όλοι οι αριθμοί να κυμαίνονται από το 1 έως

το 100 και κάθε αριθμός να αντιστοιχεί σε κάθε frame του βίντεο. Αυτό αποθηκεύεται σε ένα αρχείο και αποτελεί το fingerprint του μη καταχωρημένου βίντεο που θα χρησιμοποιηθεί και στην αναζήτηση. Η παραπάνω διεργασία είναι γνωστή αφού η ίδια πραγματοποιείται και κατά τη διαδικασία του upload. Η διαφορά είναι ότι εδώ το βίντεο δεν καταχωρείται στη βάση αλλά από αυτό εξάγεται το fingerprint που αποτελεί και το κλειδί στον αλγόριθμο της αναζήτησης. Υπάρχει περίπτωση ο χρήστης να έχει εισάγει ως επιπλέον αρχείο προς αναζήτηση το fingerprint ενός βίντεο και όχι το αυτούσιο βίντεο. Τότε η μόνη διεργασία που γίνεται είναι η κατάλληλη διαμόρφωση του αρχείου ώστε κάθε αριθμός να αντιστοιχεί σε κάθε frame και αυτοί να κυμαίνονται από 1 ως 100 όπως και παραπάνω.

Οι λίστες με τα id, τα ονόματα και τα fingerprints των επιλεγμένων database videos και query clips στέλνονται με Sessions στην σελίδα Results.aspx όπου γίνεται και η επεξεργασία τους. Πιο συγκεκριμένα, για την διαδικασία της αναζήτησης καλείται η συνάρτηση Search() της κλάσης CSearch. Αυτή παίρνει σαν παραμέτρους τα fingerprint κάθε query clip και κάθε database video και έναν αριθμό που είναι ορισμένος 50 και αποτελεί ένα «κατώφλι» για τον αλγόριθμο της αναζήτησης. Έτσι, το fingerprint κάθε query clip που έχει επιλέξει ο χρήστης αναζητάται στα fingerprint κάθε database video. Πιο συγκεκριμένα, τα δεδομένα για τα frames στο fingerprint του query clip «παιρνούν» σταδιακά και συγκρίνονται με αυτά του fingerprint κάθε μεγάλου βίντεο. Ως μέτρο σύγκρισης χρησιμοποιείται μία μεταβλητή “weight” που απεικονίζει τη διαφορά που παρατηρείται καθώς τα δεδομένα του μικρού βίντεο συγκρίνονται σταδιακά με αυτά του μεγάλου. Αν το μέτρο σύγκρισης αυτό σε κάποιο κομμάτι του database video γίνει μικρότερο από το «κατώφλι» (threshold) τότε θεωρείται ότι σε αυτό το σημείο του database video βρέθηκε το query clip με απόκλιση όση δηλώνει η μεταβλητή weight. Ως αποτέλεσμα ο αλγόριθμος επιστρέφει ένα αντικείμενο CResults που έχει σαν μεταβλητές το ακριβές frame όπου έγινε ο εντοπισμός, την απόκλιση η οποία παρατηρήθηκε και τον αριθμό των αποτελεσμάτων για κάθε query clip.

Η εμφάνιση των αποτελεσμάτων γίνεται στην σελίδα Results.aspx. Εκεί, για κάθε query clip ουσιαστικά εμφανίζονται τα δεδομένα των αντικειμένων CResults. Τα query clips και τα database videos εμφανίζονται με τα ονόματά τους, ενώ δεν εμφανίζεται ο αριθμός του frame του database video όπου εντοπίστηκε το query clip αλλά η ώρα που εντοπίστηκε ούτως ώστε να γίνεται το αποτέλεσμα πιο κατανοητό στον χρήστη. Ο υπολογισμός της ώρας έγινε με διαίρεση του αύξοντα αριθμού του frame με το fps(frames/second) του database video. Υπάρχει βέβαια περίπτωση η απόκλιση κατά τη σύγκριση των δεδομένων του μικρού βίντεο με αυτά των μεγάλων να μην γίνει ποτέ μικρότερη από το κατώφλι (threshold) οπότε και δεν έχουμε κανένα αποτέλεσμα και σημειώνεται ότι το query clip δεν βρέθηκε

στο συγκεκριμένο database video. Τέλος, για κάθε query clip μετράται ο ακριβής χρόνος που χρειάστηκε για την αναζήτησή του μέσα στα προεπιλεγμένα από το χρήστη database video. Όλα αυτά αποθηκεύονται σε ένα string με τη βοήθεια της κλάσης StringBuilder και αυτό στη συνέχεια γράφεται στο component Label ώστε να εμφανιστεί στη σελίδα της εφαρμογής.

4 Εξωτερικά προγράμματα της εφαρμογής

4.1 Βάση δεδομένων

Όπως αναφέρθηκε και παραπάνω, τα δεδομένα είναι οργανωμένα σε μία βάση δεδομένων. Από εκεί όποτε είναι απαραίτητο ανακαλούνται αλλά και στέλνονται πληροφορίες. Για τη δημιουργία της βάσης χρησιμοποιήθηκε ο Microsoft SQL Server 2005.

4.1.1 Πίνακες

Η βάση είναι οργανωμένη σε τρεις πίνακες. Αυτοί δεν είναι ανεξάρτητοι αντίθετα συνδέονται με σχέσεις που εκφράζονται στα σχεσιακά μοντέλα που θα περιγραφούν στη συνέχεια.

4.1.1.1 Περιγραφή πρώτου πίνακα:

Ο πρώτος πίνακας ονομάζεται clips και σε αυτόν καταχωρείται ο κύριος όγκος των πληροφοριών για εκάστοτε βίντεο. Τα πεδία που περιέχει είναι id, name, duration, date, video_type, txt, fps, clip_type, clip_info, clip_search, clip_picture_number. Παρακάτω δίνονται αναλυτικές πληροφορίες για κάθε πεδίο.

id: decimal(18, 0) και δηλώνει το id του κάθε video. Επίσης, έχει οριστεί ως το **πρωτεύον κλειδί** του πίνακα και αυξάνεται από το σύστημα κατά μία μονάδα κάθε φορά που εισάγεται ένα επιπλέον βίντεο.

name: nvarchar(255), εδώ αποθηκεύεται το όνομα που εισάγει ο χρήστης

duration: float και δηλώνει τη διάρκεια του βίντεο

date: datetime και δηλώνει την ημερομηνία που το βίντεο αποθηκεύτηκε στο σύστημα.

video_type: nvarchar(255), και δηλώνει τον τύπο του βίντεο(.avi,.wmv)

txt: varbinary και εδώ αποθηκεύεται το fingerprint του εκάστοτε βίντεο σε δυαδική μορφή.

fps: float, δηλώνει ένα χαρακτηριστικό του βίντεο (frames/second)

clip_type: nvarchar(max) και δηλώνει το είδος του βίντεο αν είναι δηλαδή ταινία, διαφήμιση, μουσικό video clip

clip_info: nvarchar(255), εδώ αποθηκεύονται οι γενικές πληροφορίες που έχει εισάγει ο χρήστης κατά τη διαδικασία του upload του εκάστοτε βίντεο.

clip_search: int, δηλώνει αν το βίντεο είναι μικρό query clip κάτι που σηματοδοτείται με τον αριθμό 2 ή αν είναι database video που σηματοδοτείται με τον αριθμό 1. Πρέπει να σημειωθεί ότι το βίντεο μπορεί να ανήκει και στις δύο κατηγορίες και τότε στη βάση γράφεται ο αριθμός 3.

clip_picture_number: int, δηλώνει τον αριθμό των φωτογραφιών που έχουν δημιουργηθεί για κάθε βίντεο.

Table - dbo.clips		Summary
Column Name	Data Type	Allow Nulls
id	decimal(18, 0)	<input type="checkbox"/>
name	nvarchar(255)	<input type="checkbox"/>
duration	float	<input checked="" type="checkbox"/>
date	datetime	<input checked="" type="checkbox"/>
video_type	nvarchar(255)	<input checked="" type="checkbox"/>
txt	varbinary(MAX)	<input checked="" type="checkbox"/>
fps	float	<input checked="" type="checkbox"/>
clip_type	int	<input type="checkbox"/>
clip_info	nvarchar(255)	<input checked="" type="checkbox"/>
clip_search	int	<input type="checkbox"/>
clip_picture_number	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Εικόνα 11 : Database table “clips”

4.1.1.2 Περιγραφή δευτέρου πίνακα:

Κάθε βίντεο έχει μία ή περισσότερες αντιπροσωπευτικές εικόνες οι οποίες αντιστοιχούν στις αλλαγές σκηνών που συμβαίνουν στα βίντεο. Ο δεύτερος λοιπόν πίνακας κρατά πληροφορίες για το που βρίσκονται αποθηκευμένες αυτές οι χαρακτηριστικές εικόνες του εκάστοτε βίντεο και αποτελείται από τα εξής πεδία id, path

id : int , εδώ ουσιαστικά αποθηκεύεται το id του clip στο οποίο ανήκουν οι αντιπροσωπευτικές εικόνες που ουσιαστικά αποτελεί και δευτερεύον κλειδί στον πίνακα “clips”

path : nvarchar(255) , δηλώνει το όνομα του αρχείου όπου θα αποθηκεύονται αυτές οι εικόνες πχ:C:\images

Table - dbo.image_url		Summary
Column Name	Data Type	Allow Nulls
id	decimal(18, 0)	<input type="checkbox"/>
path	nvarchar(MAX)	<input type="checkbox"/>
		<input type="checkbox"/>

Εικόνα 12 : Database table “image_url”

4.1.1.3 Περιγραφή τρίτου πίνακα:

Κάθε βίντεο κατηγοριοποιείται ανάλογα και με την επιλογή του χρήστη σε τέσσερις μεγάλες κατηγορίες : default, movie, advertisement, music video clip. Η κατηγοριοποίηση αυτή είναι δεδομένη και έτσι τα ονόματα των κατηγοριών αυτών αποθηκεύονται στον τρίτο πίνακα της εφαρμογής μας με όνομα **type**. Αυτός αποτελείται από δύο πεδία: id , type.

id : int , δηλώνει του id κάθε τύπου. Δηλαδή ο τύπος default έχει id ίσο με 1, ο τύπος movie έχει id ίσο με 2, ο τύπος advertisement έχει id ίσο με 3 ενώ ο τύπος music video clip έχει id ίσο με 4

type : nvarchar(255) , εδώ αποθηκεύεται το όνομα του τύπου (default, movie, advertisement, music video clip)

Table - dbo.type		Summary
Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
type	nvarchar(255)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

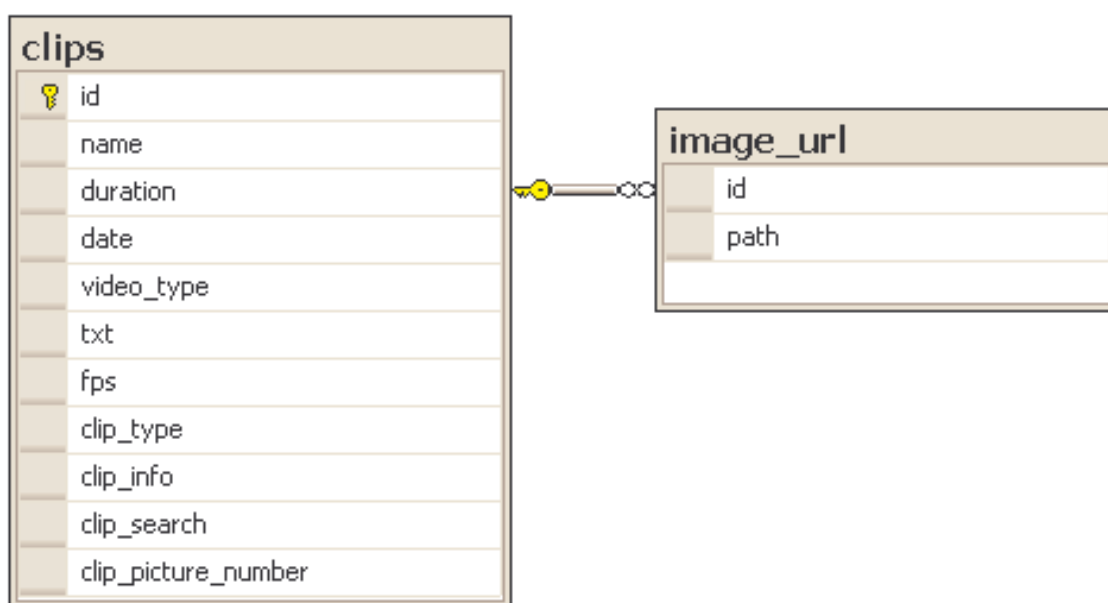
Εικόνα 13 : Database table “type”

Table - dbo.type		Summary
id	type	
1	default	
2	movie	
3	advertisement	
4	music video clip	

Εικόνα 14 : Content of the database table “type”

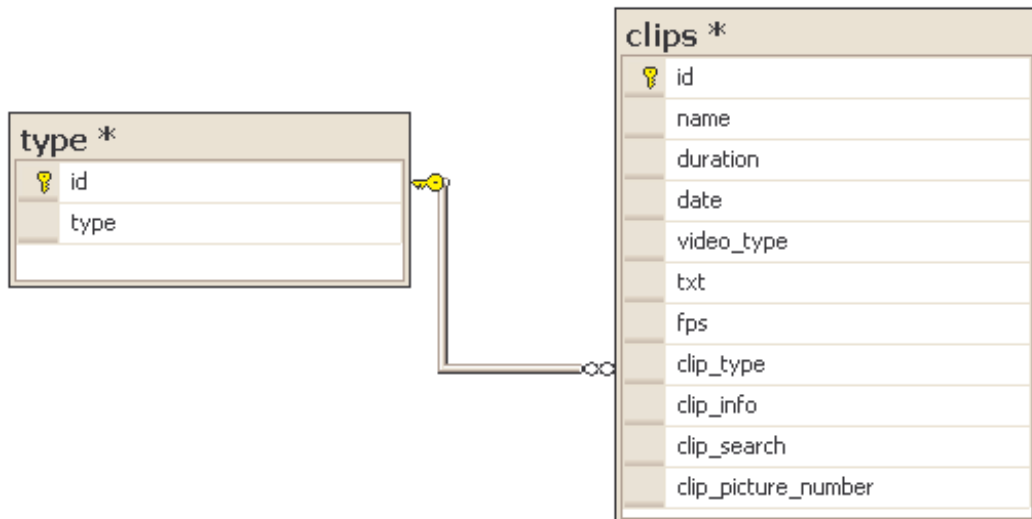
4.1.2 Σχισιακά μοντέλα

Όπως αναφέρθηκε και παραπάνω οι πίνακες της βάσης δεδομένων δεν είναι ανεξάρτητοι μεταξύ τους, αντίθετα συνδέονται με σχέσεις που φαίνονται στα παρακάτω σχισιακά μοντέλα. Πιο συγκεκριμένα, το πεδίο `id` του πίνακα `clips` συνδέεται με ξένο κλειδί με το πεδίο `id` του πίνακα `image_url`. Έτσι, για κάθε βίντεο που είναι καταχωρημένο στον πρώτο πίνακα υπάρχει και μία καταχώρηση στον δεύτερο πίνακα που δείχνει το μονοπάτι (`path`) του συστήματος όπου είναι αποθηκευμένες οι εικόνες του βίντεο με το συγκεκριμένο `id`. Αυτή η σχέση απεικονίζεται στο παρακάτω διάγραμμα.



Εικόνα 15 : Database diagram “type-image_url”

Παρατηρώντας τώρα το επόμενο διάγραμμα είναι εμφανές ότι το πεδίο `clip_type` του πίνακα `clips` συνδέεται με ξένο κλειδί με το πεδίο `id` του πίνακα `type`. Έτσι, όταν ο χρήστης επιλέξει τον τύπο του βίντεο στη φόρμα εισαγωγής που περιγράφηκε παραπάνω, αυτός αντιστοιχίζεται σε έναν αριθμό από το 1 έως το 4 όπως δηλώνουν τα πεδία του πίνακα `type`.



Εικόνα 16 : Database diagram “clips-type”

4.2 Χρήση CSS

Ως γνωστών, οι σελίδες που βλέπουμε στον ιστό είναι αρχεία τα οποία γράφονται στην markup γλώσσα html. Για την καλύτερη μορφοποίηση των στοιχείων, των αντικειμένων και γενικά των κειμένων των markup σελίδων (HTML documents) χρησιμοποιούνται τα CSS = Cascading Style Sheets. Πρόκειται δηλαδή για στυλ (styles) που εφαρμόζονται για να καλλωπίσουν τον κώδικα HTML.

Στην εφαρμογή χρησιμοποιήθηκε το sinorca-screen.css. Αυτό περιέχει οδηγίες για το πως θα μορφοποιηθούν τα στοιχεία της aspx σελίδας. Αρχικά, το αρχείο των προδιαγραφών του sinorca.css αποθηκεύθηκε στο σύστημα. Κατόπιν ενσωματώθηκε ένας σύνδεσμος προς το .css σε κάθε σελίδα aspx. Δηλαδή, στο tag <head> </head> κάθε σελίδας aspx ενσωματώθηκε το tag :

```
<link rel="stylesheet" type="text/css" href="~/sinorca-screen.css" media="screen" title="Sinorca (screen)" />
```

Έτσι, συνδέθηκε το αρχείο sinorca-screen.css με κάθε τέτοια σελίδα και το style που προτείνει το sinorca-screen.css εφαρμόστηκε σε όλα τα στοιχεία της σελίδας. Πλήθος τέτοιων αρχείων .css είναι διαθέσιμα στο διαδίκτυο από όπου και εμείς το κατεβάσαμε.

Παρακάτω δίνονται ενδεικτικά δύο εικόνες που δείχνουν τη μορφή της εφαρμογής πριν και μετά τη χρήση του sinorca-screen.css.

Insert a name for your clip:

Insert general information:

What kind of clip is it? ▼

video to be used as a query clip

video to be entered into the database

If your clip is larger than 16MB... [Click Here](#)

Εικόνα 17 : Video uploading without css

Upload a video

[Search](#) | [View our database](#) |

Insert a name for your clip:

Insert general information:

What kind of clip is it? ▼

video to be used as a query clip

video to be entered into the database

If your clip is larger than 16MB... [Click Here](#)

[Return to HOME PAGE](#) |

Εικόνα 18 : Video uploading with css

4.3 Εξαγωγή εικόνων

Όπως αναφέρθηκε και παραπάνω, όταν κάποιος χρήστης ανεβάζει ένα βίντεο η εφαρμογή δημιουργεί έναν αριθμό αντιπροσωπευτικών εικόνων. Αυτές οι εικόνες σημειώνεται ότι είναι αντιπροσωπευτικές διότι απεικονίζουν τις διάφορες σκηνές του βίντεο, για την ακρίβεια εξάγονται την στιγμή που συμβαίνει η αλλαγή σκηνής.

Για την εξαγωγή των εικόνων χρησιμοποιήθηκε η βιβλιοθήκη Interop.DexterLib.dll. Η διαδικασία για την δημιουργία των εικόνων αρχίζει με το διάβασμα του αποτυπώματος κάθε βίντεο από όπου γίνεται αντιληπτό πότε έχει συμβεί αλλαγή σκηνής. Τότε δημιουργείται η εικόνα η οποία αναπαριστά το αντίστοιχο frame. Πρέπει ακόμα να σημειωθεί ότι στο πάνω αριστερά κομμάτι της εικόνας αναγράφεται ο χρόνος του βίντεο στον οποίο αντιστοιχεί το συγκεκριμένο frame.

5 Ο πηγαίος κώδικας της εφαρμογής

Παρακάτω δίνεται ο κώδικας της εφαρμογής. Αρχικά δίνονται οι κλάσεις που χρησιμοποιούνται και μετά ακολουθεί ο κώδικας για κάθε σελίδα καθώς και ο κώδικας σχεδιασμού της κάθε σελίδας όπου μπορεί κανείς να διακρίνει όλα τα components από τα οποία αποτελείται.

5.1 Βοηθητικές κλάσεις

Κλάση **IConstants** : Χρησιμοποιείται για τον ορισμό των τιμών των σταθερών μεταβλητών

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public class IConstants
{
    public const string CONNECTION_STRING = "Data
Source=ISOLAB234\\SQLEXPRESS;Initial Catalog=videos;Integrated
Security=True;Pooling=true";
    public const String IMAGE_PATH = "C:\\Inetpub\\wwwroot\\images";
    public const String SAVED_VIDEO_PATH =
"C:\\Inetpub\\wwwroot\\clips_saved\\";
}
```

Κλάση **CDoCrop**: Η κλάση αυτή περιέχει τις συναρτήσεις doMencoder() και doFfmpeg() που υλοποιούν τις αντίστοιχες εξωτερικές διαδικασίες με τις κατάλληλες παραμέτρους.

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.IO;

public class CDoCrop
{
    public static String doMencoder(String str_clip_url)
    {
        System.Diagnostics.Process mencoder = new System.Diagnostics.Process();
        mencoder.StartInfo.FileName = @"C:\util\mencoder.exe";
        mencoder.StartInfo.Arguments = str_clip_url + " -o " +
IConstants.SAVED_VIDEO_PATH + "H264c -nosound -quiet -of rawvideo -oac copy
-vf lavcdeint,scale=320:240,crop=240:176 -ovc x264 -x264encopts
qp=26:nocabac:nochroma-me:analyse=p8x8,i4x4:threads=auto:keyint=9999999:no-
psnr:no-ssim:scenecut=-1";
        mencoder.StartInfo.UseShellExecute = false;
        mencoder.StartInfo.RedirectStandardOutput = true;
        mencoder.StartInfo.RedirectStandardError = true;
        mencoder.Start();

        String m = mencoder.StandardOutput.ReadToEnd();
        mencoder.WaitForExit();

        return m;
    }
}
```



```

    }

    public static String doFfmpeg()
    {
        System.Diagnostics.Process ffmpeg = new System.Diagnostics.Process();
        ffmpeg.StartInfo.FileName = @"C:\util\ffmpeg-v2.exe";
        ffmpeg.StartInfo.Arguments = "-f h264 -i " +
IConstants.SAVED_VIDEO_PATH + "H264c -f null a.avi";
        ffmpeg.StartInfo.UseShellExecute = false;
        ffmpeg.StartInfo.RedirectStandardOutput = false;
        ffmpeg.Start();
        ffmpeg.WaitForExit();

        StreamReader re = File.OpenText("out.txt");
        String s = re.ReadToEnd();
        re.Close();
        return s;
    }
}

```

Κλάση **CExtractImage**: Η κλάση αυτή περιέχει τη συνάρτηση που δημιουργεί τις εικόνες Save_Image() και τη συνάρτηση που μετονομάζει τις εικόνες RenameImage().

```

using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Collections.Generic;

```

```

using System.Text;
using System.Drawing;
using System.Drawing.Imaging;
using System.Collections;
using System.Runtime.InteropServices;
using System.IO;
using System.Collections.Generic;
using DexterLib;

public class CExtractImage : IConstants
{
    public static Boolean Save_Image(String str_clip_url, String clip_name,
List<int> FramesList, Double fps)
    {
        string storagePath; //used for the path where we save files
        MediaDetClass md; //needed to extract pictures

        List<String> fBitmapNameList=new List<String> ();
        storagePath = IMAGE_PATH + "\\\" + clip_name + "\\\";

        if( !Directory.Exists( storagePath ) )
            Directory.CreateDirectory( storagePath );

        md = new MediaDetClass();
        md.Filename = str_clip_url;
        md.CurrentStream = 0;
        int len = (int)md.StreamLength;
        Boolean not_img_created = false;
        try
        {
            //creation of a bitmap-name for each frame
            for (int i = 0; i < FramesList.Count; i++)
            {

```

```

    String fBitmapName = storagePath + i;
    fBitmapNameList.Add(fBitmapName);
}
for (int j = 0; j < FramesList.Count; j++)
{
    //calculation of the time of each frame
    Double d_second = (double)FramesList[j]/ fps;
    int hours = (int)d_second / 3600;
    int mins = ((int)d_second - hours * 3600) / 60;
    int secs = (int)d_second - hours * 3600 - mins*60;
    String timetime = hours.ToString("00") + ":" +
mins.ToString("00") + ":"+secs.ToString("00");

    //creation of th bitmap
    md.WriteBitmapBits(FramesList[j] / fps, 320, 240,
fBitmapNameList[j]);

    //addition of the time on the image corner
    Graphics g;
    Font drawFont = new Font("Arial", 16);
    SolidBrush drawBrush = new SolidBrush(Color.Red);

    System.Drawing.Image img =
System.Drawing.Image.FromFile(fBitmapNameList[j]);
    g = Graphics.FromImage(img);
    g.DrawString(timetime, drawFont, drawBrush, 11,23);
    g.Save();
    g.Dispose();
    img.Save(fBitmapNameList[j] + ".jpg",
ImageFormat.Jpeg);
    img.Dispose();
    System.IO.File.Delete(fBitmapNameList[j] + ".bmp");
}

```

```

    }
    catch (Exception ex)
    {
        not_img_created = true;
    }
    return not_img_created;
} //end Save_Image

```

```

    public static void Rename_Image( String clip_name , Decimal id, String
katalixi)
    {
        Directory.Move(IMAGE_PATH + "\\\" + clip_name, IMAGE_PATH +
"\\\" + "video_id" + id + "." + katalixi);
        return;
    }
}

```

Κλάση **CInsert**: Περιέχει τη συνάρτηση Insert() που επικοινωνεί με τη βάση και στέλνει τα δεδομένα

```

using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

```

```

public class CInsert : IConstants
{

    public static Decimal Insert(String clip_name, String katalixi, Byte[] barr, double
fps, double duration, Boolean not_img_created, int clip_type_index, String clip_info, int
search13, int pictures_num)
    {

        DateTime time_upload = DateTime.Now;
        SqlConnection conn = new SqlConnection(CONNECTION_STRING);
        conn.Open();

        //insert data into table clips
        SqlCommand cmd = new SqlCommand("INSERT INTO
clips(name,date,video_type,txt,fps,duration,clip_type,clip_info,clip_search,clip_picture_n
umber) values(@P1,@P2,@P3,@P4,@P5,@P6,@P11,@P12,@P13,@P14); SELECT
@@IDENTITY", conn);
        cmd.Parameters.Add("@P1", SqlDbType.NVarChar, 255).Value =
clip_name;

        cmd.Parameters.Add("@P2", SqlDbType.DateTime, 255).Value =
time_upload;
        cmd.Parameters.Add("@P3", SqlDbType.NVarChar, 255).Value = katalixi;
        cmd.Parameters.Add("@P4", SqlDbType.VarBinary, barr.Length).Value =
barr;

        cmd.Parameters.Add("@P5", SqlDbType.Float, 255).Value = fps;
        cmd.Parameters.Add("@P6", SqlDbType.Float, 255).Value = duration;
        cmd.Parameters.Add("@P11", SqlDbType.Int, 255).Value = clip_type_index;
        cmd.Parameters.Add("@P12", SqlDbType.NVarChar, 255).Value =
clip_info;

```

```

cmd.Parameters.Add("@P13", SqlDbType.Int, 255).Value = search13;
cmd.Parameters.Add("@P14", SqlDbType.Int, 255).Value = pictures_num;
Decimal id =(Decimal)cmd.ExecuteScalar();

//insert data into table image_url
if (not_img_created == false)
{
    SqlCommand cmd1 = new SqlCommand("INSERT INTO
image_url(id,path) values(@P7,@P8)", conn);
    cmd1.Parameters.Add("@P7", SqlDbType.Decimal, 18).Value = id;
    cmd1.Parameters.Add("@P8", SqlDbType.NVarChar, 255).Value =
IMAGE_PATH + "\\\" + clip_name + "\\\";
    cmd1.ExecuteNonQuery();
    conn.Close();
}
return id;
}
}

```

Κλάση **CSearch()**: Περιέχει τη συνάρτηση Search() που πραγματοποιεί την αναζήτηση του μικρής διάρκειας query clip μέσα στο μεγάλης διάρκειας database video. Η συνάρτηση επιστρέφει ένα αντικείμενο τύπου CResults που περιγράφεται παρακάτω

```

using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;

```

```

using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public class CSearch : CResults
{

public static CResults Search(Byte[] small_data, Byte[] big_data, double threshold)
{
    //check for error in video data
    if (big_data.Length == 0 || small_data.Length == 0 || big_data.Length <
small_data.Length)
    {
        return null;
    }

    int i, j;
    long sum = 0;
    CResults results = new CResults();

    //extreme case when both clips have equal lengths
    if (big_data.Length == small_data.Length)
    {
        for (i = 1; i < big_data.Length; i++)
        {
            //calculation of difference between the big database video and the small
            //query clip
            int diff_curr = (big_data[i] - small_data[i]);
            sum += diff_curr * diff_curr;
        }

        //calculation of an amount of difference

```

```

double weight = (double)sum / (double)(big_data.Length-1);
if(weight < threshold)
{
    results.AddResult(0, weight);
}

return results;
} //end if(big_data.Length == small_data.Length)

//when both clips have not equal lengths we follow the same search
//algorithm
for(i=1;i<big_data.Length-small_data.Length-1;i++)
{
    sum = 0;
    for(j=5;j<small_data.Length-1;j++)
    {
        int diff_curr=(small_data[j] - big_data[i+j]);
        sum+= diff_curr * diff_curr;

        if (j % 20 == 0)
        {
            double weightTillNow = (double)sum / (double)(j-5+1 );

            //check for very different data
            if (weightTillNow > 3*threshold)
            {
                goto conti;
            }
        }
    }
} //end for (j=5;j<small_data.Length-1;j++)

```



```

        double weight = (double)sum / (double)(small_data.Length - 5);

//if difference is smaller than threshold the query clip is found and we add the
//result
        if (weight < threshold)
        {
            results.AddResult(i, weight);
        }
        conti:
        ;
    } //end for(i=1;i<big_data.Length-small_data.Length-1;i++)

    return results;

}
}

```

Κλάση **CResults**: Ορίζει το αντικείμενο που αντιπροσωπεύει το αποτέλεσμα της αναζήτησης.

```

using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

```

```

public class CResults {
    public int FoundNum
    {
        get {
            return frames.Count;
        }
    }

    //used for the frames where the query clip matches with the database video
    System.Collections.Generic.List<int> frames;

    //used for the difference between the original frame of the query clip and the
    //frame of the database video where the match found
    System.Collections.Generic.List<Double> weights;

    public void AddResult(int frame, double weight) {
        frames.Add(frame);
        weights.Add(weight);
    }

    public int GetFrameResult(int i)
    {
        return frames[i];
    }

    public double GetWeightResult(int i)
    {
        return weights[i];
    }

    //constructor
    public CResults()
    {

```

```

frames = new System.Collections.Generic.List<int>();
weights = new System.Collections.Generic.List<Double>();
}
}

```

5.2 Πηγαίος κώδικας των web - pages

Παρακάτω δίνεται ο κώδικας που υλοποιείται καθώς τρέχει η πρώτη σελίδα καλωσορίσματος καθώς και ο κώδικας που περιέχει τα components της σελίδας.

5.2.1 Πηγαίος κώδικας της σελίδας «Default»

5.2.1.1 Κώδικας μορφοποίησης της σελίδας «Default»

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>Search Demo</title>
<link rel="stylesheet" type="text/css" href="~/sinorca-screen.css"
media="screen" title="Sinorca (screen)" />
</head>
<body>

<!-- ##### Header ##### -->
<div id="header">
<div class="superHeader">
<div class="midHeader">
<h1 class="headerTitle">Videos</h1>
</div>
</div>

```

```

</div>

<form id="form1" runat="server">
  <div style="text-align: center">
    <asp:Image ID="Image1" runat="server"
ImageUrl="~/images/video_picture.jpg" /><br />
    <br />
    <br />
    <br />
    <br />
    <asp:Label ID="Label1" runat="server" Text="Upload a
video....."></asp:Label>
    <asp:HyperLink ID="HyperLink1" runat="server"
NavigateUrl="~/Upload.aspx">
    <font color="#153E7E">here</font>
    </asp:HyperLink><br />
    <br />
    <asp:Label ID="Label2" runat="server" Text="View our database.....">
</asp:Label>
    <asp:HyperLink ID="HyperLink2" runat="server"
NavigateUrl="~/View_clips.aspx"> <font color="#153E7E">here</font>
    </asp:HyperLink>
    <br /><br />
    <asp:Label ID="Label3" runat="server" Text="Search for a
video....."></asp:Label>
    <asp:HyperLink ID="HyperLink3" runat="server"
NavigateUrl="~/Search_Clip.aspx"><font
color="#153E7E">here</font></asp:HyperLink>
  </div>
</form>
  <div id="footer"></div>
</body>
</html>

```

5.2.1.2 Default.aspx.cs

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
}
```

5.2.2 Πηγαίος κώδικας της σελίδας «Upload»

5.2.2.1 Κώδικας μορφοποίησης της σελίδας «Upload»

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Upload.aspx.cs" Inherits="Upload" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Search Demo</title>
    <link rel="stylesheet" type="text/css" href="~/sinorca-screen.css"
media="screen" title="Sinorca (screen)" />
</head>
<body>
```

```

<!-- ##### Header ##### -->
<div id="header">
  <div class="superHeader">
    <div class="midHeader">
      <h1 class="headerTitle">Upload a video</h1>
    </div>

    <div class="subHeader">
      <span class="doNotDisplay">Navigation:</span>
      <a href="Search_Clip.aspx">Search</a> |
      <span class="doNotDisplay">Navigation:</span>
      <a href="View_clips.aspx">View our database</a> |
    </div>
  </div>
</div>
</div>
<br /><br /><br />
<form id="form1" runat="server">
  <div>
    <br /><br />
    <table id="table_upload">
      <tr>
        <td/>
        <td><input id="File1" type="file" runat="server"/><br /></td>
      </tr>
      <tr>
        <td><asp:Label ID="Label2" runat="server" Text="Insert a name for
your clip:"></asp:Label></td>
        <td><asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox><br /></td>
      </tr>
      <tr>
        <td><asp:Label ID="Label7" runat="server" Text="Insert general
information:"></asp:Label></td>

```

```

        <td><asp:TextBox ID="TB_clip_info"
runat="server"></asp:TextBox><br /></td>
    </tr>
    <tr>
        <td><asp:Label ID="Label8" runat="server" Text="What kind of clip
is it?"></asp:Label></td>
        <td><asp:DropDownList ID="DropDownList1" runat="server"
DataSourceID="SqlDataSource1" DataTextField="type" DataValueField="id">
        </asp:DropDownList><asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$ ConnectionStrings:videosConnectionString
%>"
        SelectCommand="SELECT [id], [type] FROM
[type]"></asp:SqlDataSource>
        <br />
    </td>
    </tr>
    <tr>
        <td></td>
        <td><asp:CheckBox ID="CB_small" runat="server" Text="video to
be used as a query clip" /><br /></td>
    </tr>
    <tr>
        <td></td>
        <td><asp:CheckBox ID="CB_big" runat="server" Text="video to be
entered into the database" />
        <br /><br />
    </td>
    </tr>
    <tr>
        <td></td>
        <td><asp:Label ID="Label15" runat="server" Text="If your clip is
larger than 16MB..."></asp:Label><br /><br /></td>
        <td><asp:HyperLink ID="HyperLink2" runat="server"
NavigateUrl="~/large_videos.aspx" ForeColor="#153E7E">Click
Here</asp:HyperLink><br /><br /></td>
    </tr>

```

```

        </tr>
        <tr>
            <td></td>
            <td><asp:Button ID="Button1" runat="server"
OnClick="Upload_Click" Text="Upload" /><br /></td>
        </tr>
        <tr>
            <td></td>
            <td><asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server" ControlToValidate="TextBox1"
ErrorMessage="name is
needed"></asp:RequiredFieldValidator></td>
        </tr>
        <tr>
            <td></td>
            <td><asp:Label ID="Label1" runat="server"></asp:Label></td>
        </tr>
        <tr>
            <td></td>
            <td><asp:Label ID="Label4" runat="server"
ForeColor="Red"></asp:Label></td>
        </tr>
    </table>
    <br /><br />
</div>
<div id="footer">
    <a href="Default.aspx">Return to HOME PAGE</a> |
</div>
</form>
</body>
</html>

```

5.2.2.2 Upload.aspx.cs

```
using System;
```



```

using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.ComponentModel;
using System.Diagnostics;
using System.IO;

public partial class Upload : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    //function for the event "click of the upload button"
    protected void Upload_Click(object sender, EventArgs e)
    {

        if (!Page.IsValid)
            return;

        //name of the selected clip
        String str_file_name =
System.IO.Path.GetFileName(File1.PostedFile.FileName);

        //check if none of the checkboxes is checked
        if (CB_big.Checked == false && CB_small.Checked == false)
        {

```

```

        Label1.Text = "Sorry try again...you must check at least one of the
checkboxes above";
    }
else
{
    String clip_name = TextBox1.Text; //name inserted by the user
    clip_name = clip_name.Replace(' ', '_');
    String clip_info = TB_clip_info.Text; //info inserted by the user

    //video container
    String katalixi = str_file_name.Substring(str_file_name.LastIndexOf(".") + 1);

    //initial path where the video is saved
    String str_clip_url = IConstants.SAVED_VIDEO_PATH + str_file_name;

    try
    {
        File1.PostedFile.SaveAs(str_clip_url);
    }

    catch (Exception ex)
    {
        Label1.Text = ex.Message;
        return;
    }

    //Process mencoder
    String m = CDoCrop.doMencoder(str_clip_url);

    //fps from the output of mencoder
    String[] words_mencoder = m.Split(' ');
    String str_fps = null;
    double fps;

    for (int i = 0; i < words_mencoder.Length; i++)

```

```

    {
        if (words_mencoder[i].Trim().Equals("fps"))
            str_fps = words_mencoder[i - 1];
    }
    IFormatProvider ifp = new System.Globalization.CultureInfo("en-US");
try
    {
        fps = Double.Parse(str_fps, ifp);
    }
catch (Exception ex)
    {
        fps = 25;
    }

    //Process ffmpeg-v2
    String s = CDoCrop.doFfmpeg();
    System.Collections.Generic.List<int> intraList = new
System.Collections.Generic.List<int>();
    System.Collections.Generic.List<int> FramesList = new
System.Collections.Generic.List<int>();

    //suitable formation of the ffmpeg output and creation of binary file for the database
    String[] words = s.Split('\n');
    for (int i = 0; i < words.Length; i++)
    {
        if (!(words[i].IndexOf(';') > 0)) break;
        String word = words[i].Substring(words[i].IndexOf(';') + 1);
        IFormatProvider ifp1 = new System.Globalization.CultureInfo("en-
US");

        double d = Double.Parse(word, ifp1);
        int num = (int)(100.0 * d + 0.5);
        System.Diagnostics.Debug.Assert(num >= 0 && num <= 100);
        intraList.Add(num);
    }

```

```

//frames that represent scene changes and are used for image extraction
if (num > 90)
{
    FramesList.Add(i);
}
}
Byte[] barr = new Byte[intraList.Count];
for (int i = 0; i < intraList.Count; i++)
{
    barr[i] = (Byte)intraList[i];
}

//used for the number of the video frames
int num_frames = words.Length - 1;
double duration = num_frames / fps; //used for clip's duration

// used for the clip's type
String str_clip_type_index = DropDownList1.SelectedValue;
int clip_type_index = Int32.Parse(str_clip_type_index);
int search13 = 0;

//check which one of the checkboxes is checked
if (CB_big.Checked == true) search13 += 1;
if (CB_small.Checked == true)
    search13 += 2;

//creation of images
Boolean not_img_created = false;
not_img_created = CExtractImage.Save_Image(str_clip_url, clip_name,
FramesList, fps);

//insert in the database

```

```

        Decimal id = CInsert.Insert(clip_name, katalixi, barr, fps, duration,
not_img_created, clip_type_index, clip_info, search13, FramesList.Count);

        //rename video_file and images according the id
        String str_id_url = IConstants.SAVED_VIDEO_PATH + "video_id" + id + "."
+ katalixi;

        File.Move(str_clip_url, str_id_url);
        CExtractImage.Rename_Image(clip_name, id, katalixi);

        Label1.Text = "The " + clip_name + " clip has been successfully
uploaded!!!";
        } //end else(CB_big.Checked == false && CB_small.Checked == false)
return;

    } //end Upload_Click
} //end Upload

```

5.2.3 Πηγαίος κώδικας της σελίδας «Large videos»

5.2.3.1 Κώδικας μορφοποίησης της σελίδας «Large videos»

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="large_videos.aspx.cs" Inherits="large_videos" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>Search Demo</title>
<link rel="stylesheet" type="text/css" href="~/sinorca-screen.css"
media="screen" title="Sinorca (screen)" />
</head>

```

```

<body>

<!-- ##### Header ##### -->
<div id="header">
  <div class="superHeader">
    <div class="midHeader">
      <h1 class="headerTitle">Upload a large video</h1>
    </div>

    <div class="subHeader">
      <span class="doNotDisplay">Navigation:</span>
      <a href="Upload.aspx">Upload</a> |
      <span class="doNotDisplay">Navigation:</span>
      <a href="View_clips.aspx">View our database</a> |
      <span class="doNotDisplay">Navigation:</span>
      <a href="Search_Clip.aspx">Search</a> |
    </div>
  </div>
</div>
</div>
<br /><br /><br />

<form id="form1" runat="server">
  <div>
    <br /><br />
    <table>
      <tr>
        <td></td>
        <td><input id="File1" type="file" runat="server" /><br />
          <br />
        </td>
      </tr>
      <tr>
        <td><asp:Label ID="Label3" runat="server" Text="Insert a name for
your file:"></asp:Label></td>

```

```

        <td><asp:TextBox ID="TextBox1"
runat="server"></asp:TextBox><br /></td>
    </tr>
    <tr>
        <td><asp:Label ID="Label5" runat="server" Text="Insert general
information:"></asp:Label></td>
        <td><asp:TextBox ID="TB_file_info"
runat="server"></asp:TextBox><br /></td>
    </tr>
    <tr>
        <td><asp:Label ID="Label6" runat="server" Text="Insert the
fps(frames/second):"></asp:Label>&nbsp;</td>
        <td><asp:TextBox ID="TextBox2"
runat="server">25</asp:TextBox><br /></td>
    </tr>
    <tr>
        <td><asp:Label ID="Label2" runat="server" Text="What type of file is
this?"></asp:Label>&nbsp;</td>
        <td><asp:DropDownList ID="DropDownList1" runat="server"
DataSourceID="SqlDataSource1" DataTextField="type" DataValueField="id">
            <asp:ListItem Value="0">movie</asp:ListItem>
            <asp:ListItem Value="1">advertisement</asp:ListItem>
            <asp:ListItem Value="2">music_video_clip</asp:ListItem>
        </asp:DropDownList>
        <asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:videosConnectionString %>"
SelectCommand="SELECT [id], [type] FROM
[type]"></asp:SqlDataSource><br />
    </td>
    </tr>
    <tr>
        <td></td>
        <td><asp:CheckBox ID="CB_small" runat="server" Text="This video
can be used to search inside other videos" /><br /></td>
    </tr>

```

```

</tr>
<tr>
  <td></td>
  <td><asp:CheckBox ID="CB_big" runat="server" Text="Other videos
can be searched inside this video" /><br /> &nbsp;   <br /></td>
</tr>
<tr>
  <td></td>
  <td><asp:Button ID="Button1" runat="server"
OnClick="Upload_FingerPrint_Click" Text="Upload " /><br /></td>
</tr>
<tr>
  <td></td>
  <td><br /><asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server" ControlToValidate="TextBox1"
  ErrorMessage="name is needed"></asp:RequiredFieldValidator>
<br />
  <asp:CustomValidator ID="CustomValidator1" runat="server"
ErrorMessage="You must select at least one of the
checkboxes"></asp:CustomValidator></td>
</tr>
<tr>
  <td></td>
  <td><asp:Label ID="Label4" runat="server"
Text=""></asp:Label></td>
</tr>
<tr>
  <td></td>
  <td><asp:Label ID="LB_NotChecked" runat="server"
Text=""></asp:Label></td>
</tr>
<tr>
  <td></td>

```



```

        <td><asp:Label ID="Label1" runat="server" ForeColor="Red"
Text=""></asp:Label></td>
    </tr>
</table>
</div>
</form>
<div id="footer">
    <a href="Default.aspx">Return to HOME PAGE</a> |
</div>
</body>
</html>

```

5.2.3.2 Large videos.aspx.cs

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.IO;
using System.Data.SqlClient;

public partial class large_videos : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void Upload_FingerPrint_Click(object sender, EventArgs e)

```

```

{

    if (!Page.IsValid)
        return;

    //name of the selected ctxt file
    String str_file_name =
System.IO.Path.GetFileName(File1.PostedFile.FileName);

    //check if none of the checkboxes is checked
    if (CB_big.Checked == false && CB_small.Checked == false)
    {
        LB_NotChecked.Text = "Sorry try again...you must check at least one of
the checkboxes above";
    }
    else
    {
        String clip_name = TextBox1.Text; //name inserted by the user
        clip_name = clip_name.Replace(' ', '_');
        String clip_info = TB_file_info.Text; //info inserted by the user

        //video container
        String katalixi = str_file_name.Substring(str_file_name.LastIndexOf(".") + 1);

        //initial path where the video is saved
        String str_clip_url = IConstants.SAVED_VIDEO_PATH +
str_file_name;

        try
        {
            File1.PostedFile.SaveAs(str_clip_url);
        }
        catch (Exception ex)
        {
            Label1.Text = ex.Message;
        }
    }
}

```

```

        return;
    }

    //suitable formation of the ffmpeg output and creation of binary file for the
    //database
    StreamReader re = File.OpenText(str_clip_url);
    String input = re.ReadToEnd();

    System.Collections.Generic.List<int> largeList = new
System.Collections.Generic.List<int>();
    String[] words = input.Split('\n');
    for (int i = 0; i < words.Length; i++)
    {
        if (!(words[i].IndexOf(';') > 0)) break;
        String word = words[i].Substring(words[i].IndexOf(';') + 1);
        IFormatProvider ifp1 = new System.Globalization.CultureInfo("en-
US");

        double d = Double.Parse(word, ifp1);
        int num = (int)(100.0 * d + 0.5);
        System.Diagnostics.Debug.Assert(num >= 0 && num <= 100);
        largeList.Add(num);
    }
    Byte[] barr = new Byte[largeList.Count];
    for (int i = 0; i < largeList.Count; i++)
    {
        barr[i] = (Byte)largeList[i];
    }

    //used for the number of the video frames
    int num_frames = words.Length - 1;

    // used for the clip's type

```

```

String str_clip_type_index = DropDownList1.SelectedValue;
int clip_type_index = Int32.Parse(str_clip_type_index);

Double fps = Double.Parse(TextBox2.Text);
double duration = num_frames / fps; //used for clip's duration

//check which one of the checkboxes is checked
int search13 = 0;
if (CB_big.Checked == true) search13 += 1;
if (CB_small.Checked == true) search13 += 2;

//we have no images here
Boolean not_img_created = true;

//insert the database
Decimal id = CInsert.Insert(clip_name, katalixi, barr, fps, duration,
not_img_created, clip_type_index, clip_info, search13, 0);

//rename of the video_file according the id
String str_id_url = IConstants.SAVED_VIDEO_PATH + "video_id" +
id + "." + katalixi;
File.Copy(str_clip_url, str_id_url);
LB_NotChecked.Text = "The " + clip_name + " fingerprint has been
successfully uploaded!!!";
} //end else(CB_big.Checked == false && CB_small.Checked == false)

return;
} //end Upload_Fingerprint_Click
} //end large_videos

```

5.2.4 Πηγαίος κώδικας της σελίδας «View»

5.2.4.1 Κώδικας μορφοποίησης της σελίδας View clips.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="View_clips.aspx.cs" Inherits="DetailsCourse" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
<title>Search Demo</title>
<link rel="stylesheet" type="text/css" href="~/sinorca-screen.css"
media="screen" title="Sinorca (screen)" />
</head>
<body>

<!-- ##### Header ##### -->
<div id="header">
<div class="superHeader">
<div class="midHeader">
<h1 class="headerTitle">This is our database</h1>
</div>
<div class="subHeader">
<span class="doNotDisplay">Navigation:</span>
<a href="Upload.aspx">Upload</a> |
<span class="doNotDisplay">Navigation:</span>
<a href="Search_Clip.aspx">Search</a> |
</div>
</div>
</div>
<br />

<form id="form1" runat="server">
<div>
<br />
```

```

<br /><br /><br /><br />
<table>
  <tr>
    <td>
      Query Clips</td>
    <td>Database Clips
      <asp:Label ID="LB_TotalBigDuration" runat="server" Text=""
ForeColor="#153E7E"></asp:Label></td>
  </tr>
  <tr>
    <td style="width: 274px" valign="top">
      <asp:GridView ID="GV_small" runat="server"
AutoGenerateColumns="False"
      DataSourceID="SqlDataSource1" CellPadding="4" ForeColor="Blue"
GridLines="None" CellSpacing="2" HorizontalAlign="Left" >
        <Columns>
          <asp:BoundField DataField="id" HeaderText="id"
InsertVisible="False" ReadOnly="True"
SortExpression="id" />
          <asp:BoundField DataField="name" HeaderText="name"
SortExpression="name" />
          <asp:HyperLinkField DataNavigateUrlFields="id"
DataNavigateUrlFormatString="Details_Clips.aspx?id={0} "
Text="Details" ControlStyle-ForeColor="#153E7E"/>
        </Columns>
        <FooterStyle BackColor="#990000" Font-Bold="True"
ForeColor="White" />
        <RowStyle BackColor="DarkGray" ForeColor="White" />
        <SelectedRowStyle BackColor="#FFCC66" Font-Bold="True"
ForeColor="Navy" />
        <PagerStyle BackColor="#FFCC66" ForeColor="#333333"
HorizontalAlign="Center" />
        <HeaderStyle BackColor="#153E7E" Font-Bold="True"
ForeColor="White" />

```

```

        <AlternatingRowStyle BackColor="DarkGray" />
    </asp:GridView>
</td>
<td rowspan="3" valign="top">
    <asp:GridView ID="GV_big" runat="server"
AutoGenerateColumns="False"
    DataSourceID="SqlDataSource2" CellPadding="4" ForeColor="Blue"
GridLines="None" CellSpacing="2" Height="10px" HorizontalAlign="Left"
Width="10px" >
        <Columns>
            <asp:BoundField DataField="id" HeaderText="id"
InsertVisible="False" ReadOnly="True"
SortExpression="id" />
            <asp:BoundField DataField="name" HeaderText="name"
SortExpression="name" />
            <asp:HyperLinkField DataNavigateUrlFields="id"
DataNavigateUrlFormatString="Details_Clips.aspx?id={0} "
Text="Details" ControlStyle-ForeColor="#153E7E"/>
        </Columns>
        <FooterStyle BackColor="#990000" Font-Bold="True"
ForeColor="White" />
        <RowStyle BackColor="DarkGray" ForeColor="White" />
        <SelectedRowStyle BackColor="#FFCC66" Font-Bold="True"
ForeColor="Navy" />
        <PagerStyle BackColor="#FFCC66" ForeColor="#333333"
HorizontalAlign="Center" />
        <HeaderStyle BackColor="#153E7E" Font-Bold="True"
ForeColor="White" />
        <AlternatingRowStyle BackColor="DarkGray" />
    </asp:GridView>
</td>
</tr>
<tr>
<td></td>

```

```

        <td></td>
    </tr>

</table>
<br /><br /><br />
<asp:SqlDataSource ID="SqlDataSource2" runat="server"
ConnectionString="<%$ ConnectionStrings:videosConnectionString %>"
    SelectCommand="SELECT [id], [name] FROM [clips] WHERE
(([clip_search] =1) OR([clip_search] = 3))">
    </asp:SqlDataSource>
    <asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:videosConnectionString %>"
    SelectCommand="SELECT [id], [name] FROM [clips] WHERE
(([clip_search] =2) OR([clip_search] = 3))">
    </asp:SqlDataSource>
    <br />
    <br />
</div>
</form>
<div id="footer">
    <a href="Default.aspx">Return to HOME PAGE</a> |
</div>
</body>
</html>

```

5.2.4.2 View clips.aspx.cs

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;

```



```

using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;

public partial class DetailsCourse : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        SqlConnection conn = new
SqlConnection(Constants.CONNECTION_STRING);
        conn.Open();

        //the total duration of database videos
        SqlCommand cmd = new SqlCommand("SELECT sum (duration) FROM
clips where clip_search= 1 or clip_search= 3 ;", conn);
        Double clip_duration = (Double) cmd.ExecuteScalar();
        int hours = (int)(clip_duration) / 3600;
        int mins = (int)(clip_duration - hours * 3600) / 60;
        LB_TotalBigDuration.Text = "(~" + hours.ToString() + " hours and "+
mins.ToString() + " minutes)";
        conn.Close();
    }
}

```

5.2.5 Πηγαίος κώδικας για τη σελίδα «Details»

5.2.5.1 Κώδικας μορφοποίησης της σελίδας Details.aspx

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Details_Clips.aspx.cs" Inherits="DetailsCourse" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">

```

```

<title>Search Demo</title>
  <link rel="stylesheet" type="text/css" href="~/sinorca-screen.css"
media="screen" title="Sinorca (screen)" />
</head>
<body>

<!-- ##### Header ##### -->
  <div id="header">
    <div class="superHeader">
      <div class="midHeader">
        <h1 class="headerTitle">View Details....!!!</h1>
      </div>

      <div class="subHeader">
        <span class="doNotDisplay">Navigation:</span>
        <a href="Upload.aspx">Upload</a> |
        <span class="doNotDisplay">Navigation:</span>
        <a href="View_clips.aspx">View our database</a> |
        <span class="doNotDisplay">Navigation:</span>
        <a href="Search_Clip.aspx">Search</a> |
      </div>
    </div>
  </div>
  </div>
  <br /><br /><br /><br /><br />

  <form id="form1" runat="server">
    <div>
      <br />
      <asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False"
      DataSourceID="SqlDataSource1"
onrowcommand="GridView1_RowCommand">
        <Columns>

```

```

        <asp:BoundField DataField="id" HeaderText="id"
InsertVisible="False" ReadOnly="True"
        SortExpression="id" />
        <asp:BoundField DataField="name" HeaderText="name"
SortExpression="name" />
        <asp:BoundField DataField="date" HeaderText="date"
SortExpression="date" />
        <asp:BoundField DataField="clip_type" HeaderText="clip_type"
SortExpression="clip_type" />
        <asp:BoundField DataField="duration" HeaderText="duration(sec)"
SortExpression="duration" />
        <asp:BoundField DataField="clip_info" HeaderText="clip_info"
SortExpression="clip_info" />
        <asp:BoundField DataField="clip_picture_number"
HeaderText="clip_picture_number"
        SortExpression="clip_picture_number" />
        <asp:BoundField DataField="video_type" HeaderText="video_type"
SortExpression="video_type" />
        <asp:ButtonField Text="Remove" ButtonType="Button"
CommandName="OnRemoveClick" />
    </Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%"$ ConnectionStrings:videosConnectionString %>"
SelectCommand="SELECT [id], [name], [duration], [date], [clip_type],
[clip_info], [clip_picture_number], [video_type] FROM [clips] WHERE ([id] = @id)">

    <SelectParameters>
        <asp:QueryStringParameter Name="id" QueryStringField="id"
Type="Decimal" />
    </SelectParameters>
</asp:SqlDataSource>
<br />

```

```

        &nbsp;<asp:Label ID="LB_error" runat="server"
ForeColor="Red"></asp:Label><br />
    </div>
    <div id="koko" runat="server" />
</form>
<div id="footer">
    <a href="View_clips.aspx">Go Back</a> |
</div>
</body>
</html>

```

5.2.5.2 Details.aspx.cs

```

using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.IO;

public partial class DetailsCourse : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        //used for the id of the selected video
        String id = GridView1.Rows[0].Cells[0].Text;

```

```

//used for the container of the selected video
String katalixi = GridView1.Rows[0].Cells[7].Text;

//used for the path of the selected video
String clip_url = IConstants.SAVED_VIDEO_PATH + "video_id" + id +
"." + katalixi;

//used for the name of the selected video
String clip_name = GridView1.Rows[0].Cells[1].Text;

//creation of hyperlink download
System.Web.UI.WebControls.HyperLinkField download = new
System.Web.UI.WebControls.HyperLinkField();
download.Text = "Download";
download.NavigateUrl = "clips_saved/video_id" + id + "." + katalixi;
GridView1.Columns.Add(download);

//creation of images
String str_picture_num = GridView1.Rows[0].Cells[6].Text;
int pictures_num = Int32.Parse(str_picture_num);
if (pictures_num != 0)
{
    for (int i = 0; i < pictures_num; i++)
    {
        System.Web.UI.WebControls.Image myImage = new
System.Web.UI.WebControls.Image();
myImage.ImageUrl = "images/video_id" + id + "." + katalixi + "/" + i
+ ".jpg";
        koko.Controls.Add(myImage);
    }
}
else
{

```

```

        System.Web.UI.WebControls.Image myImage = new
System.Web.UI.WebControls.Image();
        myImage.ImageUrl = "images/video_picture.jpg";
        koko.Controls.Add(myImage);
    }
} //end Page_Load

protected void GridView1_RowCommand(Object sender,
GridViewCommandEventArgs e)
{
    if (e.CommandName == "OnRemoveClick")
    {
        //delete from database the selected video
        int index = Convert.ToInt32(e.CommandArgument);
        GridViewRow selectedRow = GridView1.Rows[index];
        String selectedId = selectedRow.Cells[0].Text;
        int deleteId = (int)Int32.Parse(selectedId);
        SqlConnection conn = new
SqlConnection(Iconstants.CONNECTION_STRING);
        conn.Open();
        SqlCommand cmd = new SqlCommand("DELETE FROM image_url
where id= " + deleteId + ";", conn);
        SqlCommand cmd1 = new SqlCommand("DELETE FROM clips where
id= " + deleteId + ";", conn);
        cmd.ExecuteNonQuery();
        cmd1.ExecuteNonQuery();
        conn.Close();

        //delete the files of the selected video
        String katalixi = GridView1.Rows[0].Cells[7].Text;
        String id= GridView1.Rows[0].Cells[0].Text;
        String clip_name = GridView1.Rows[0].Cells[1].Text;
        String clip_url = Iconstants.SAVED_VIDEO_PATH + "video_id" + id
+ "." + katalixi;

```

```

try
{
    File.Delete(clip_url);
    File.Delete(IConstants.SAVED_VIDEO_PATH + "H264c");
}
catch (Exception ex)
{
    LB_error.Text = ex.Message;
    return;
}

//delete the images of the selected video
String str_picture_num = GridView1.Rows[0].Cells[6].Text;
int pictures_num = Int32.Parse(str_picture_num);
if (pictures_num != 0)
{
    try
    {
        for (int i = 0; i < pictures_num; i++)
        {
            File.Delete(IConstants.IMAGE_PATH + "\\video_id" + id + "."
+ katalixi + "\\ " + i + ".jpg");
            File.Delete(IConstants.IMAGE_PATH + "\\video_id" + id + "."
+ katalixi + "\\ " + i);
        }
        Directory.Delete(IConstants.IMAGE_PATH + "\\video_id" + id +
"." + katalixi);
    }
    catch (Exception ex)
    {
        LB_error.Text = ex.Message;
        return;
    }
}

```

```

    } //end if(pictures_num != 0)
    Response.Redirect("View_clips.aspx");
  }
}
}

```

5.2.6 Πηγαίος κώδικας της σελίδας «Search»

5.2.6.1 Κώδικας μορφοποίησης της σελίδας Search.aspx

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Search_Clip.aspx.cs" Inherits="Search_Clip" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Search Demo</title>
  <link rel="stylesheet" type="text/css" href="~/sinorca-screen.css"
media="screen" title="Sinorca (screen)" />
</head>
<body>

<!-- ##### Header ##### -->
<div id="header">
  <div class="superHeader">
  <div class="midHeader">
    <h1 class="headerTitle">Search...</h1>
  </div>

  <div class="subHeader">
    <span class="doNotDisplay">Navigation:</span>
    <a href="Upload.aspx">Upload</a> |
    <span class="doNotDisplay">Navigation:</span>

```



```

        <a href="View_clips.aspx">View our database</a> |
    </div>
</div>
</div>
<br /><br /><br />

<form id="form1" runat="server">
<div>

    <asp:Label ID="Label1" runat="server" Text="Select the video that you
want to search..."></asp:Label><br />
    <br />
    <table id="TSearch" runat="server">
        <tr>
            <td style="width: 274px" valign="top"> Select query clips to be used
for search</td>
            <td valign="top"> Select database videos to be searched</td>
        </tr>
        <tr>
            <td style="width: 274px" valign="top">
                <asp:GridView ID="GV_small" runat="server"
AutoGenerateColumns="False"
                DataSourceID="SqlDataSource1" CellPadding="4" ForeColor="Blue"
GridLines="None" CellSpacing="2" HorizontalAlign="Left" >
                    <Columns>

                        <asp:TemplateField HeaderText="Select">
                            <ItemTemplate>
                                <asp:CheckBox ID="CheckBox1" runat="server"/>
                                <asp:Label ID="check1" runat="server" Text="<%#
((System.Data.DataRowView)Container.DataItem)["id"] %>"
                                    Visible="false"></asp:Label>
                            </ItemTemplate>
                        <HeaderStyle HorizontalAlign="Left" />

```

```

        <FooterStyle HorizontalAlign="Left" />
    </asp:TemplateField>

    <asp:BoundField DataField="id" HeaderText="id"
InsertVisible="False" ReadOnly="True"
SortExpression="id" />
    <asp:BoundField DataField="name" HeaderText="name"
SortExpression="name" />
</Columns>
<FooterStyle BackColor="#990000" Font-Bold="True"
ForeColor="White" />
<RowStyle BackColor="DarkGray" ForeColor="white" />
<SelectedRowStyle BackColor="#FFCC66" Font-Bold="True"
ForeColor="Navy" />
<PagerStyle BackColor="#FFCC66" ForeColor="#333333"
HorizontalAlign="Center" />
<HeaderStyle BackColor="#153E7E" Font-Bold="True"
ForeColor="White" />
<AlternatingRowStyle BackColor="DarkGray" />
</asp:GridView>
</td>
<td rowspan="3" valign="top">
    <asp:GridView ID="GV_big" runat="server"
AutoGenerateColumns="False"
DataSourceID="SqlDataSource2" CellPadding="4" ForeColor="Blue"
GridLines="None" CellSpacing="2" Height="10px" HorizontalAlign="Left"
Width="10px" >
        <Columns>
            <asp:TemplateField HeaderText="Select">
                <ItemTemplate>
                    <asp:CheckBox ID="CheckBox2" Checked="True"
runat="server"/>
                    <asp:Label ID="check2" runat="server" Text="<%#
((System.Data.DataRowView)Container.DataItem)["id"] %>"

```

```

        Visible="false"></asp:Label>
    </ItemTemplate>

    <HeaderStyle HorizontalAlign="Left" />
    <FooterStyle HorizontalAlign="Left" />
</asp:TemplateField>

    <asp:BoundField DataField="id" HeaderText="id"
InsertVisible="False" ReadOnly="True"
    SortExpression="id" />
    <asp:BoundField DataField="name" HeaderText="name"
SortExpression="name" />
</Columns>
    <FooterStyle BackColor="#990000" Font-Bold="True"
ForeColor="White" />
    <RowStyle BackColor="DarkGray" ForeColor="white" />
    <SelectedRowStyle BackColor="#FFCC66" Font-Bold="True"
ForeColor="Navy" />
    <PagerStyle BackColor="#FFCC66" ForeColor="#333333"
HorizontalAlign="Center" />
    <HeaderStyle BackColor="#153E7E" Font-Bold="True"
ForeColor="White" />
    <AlternatingRowStyle BackColor="DarkGray" />
</asp:GridView>
</td>
</tr>
<tr>
    <td><asp:Label ID="Label2" runat="server" Text="Do you want to
search for another clip?"></asp:Label></td>
    <td></td>
</tr>
<tr>
    <td><input id="File1" type="file" runat="server" /></td>
    <td></td>

```

```

        </tr>
        <tr>
            <td><asp:Button ID="B_Search" runat="server"
OnClick="B_Search_Click" Text="Search" /></td>
        </tr>
    </table>
    <asp:SqlDataSource ID="SqlDataSource2" runat="server"
ConnectionString="<%"$ ConnectionStrings:videosConnectionString %>"
SelectCommand="SELECT [id], [name] FROM [clips] WHERE
(([clip_search] =1) OR([clip_search] = 3))">
    </asp:SqlDataSource>
    <asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%"$ ConnectionStrings:videosConnectionString %>"
SelectCommand="SELECT [id], [name] FROM [clips] WHERE
(([clip_search] =2) OR([clip_search] = 3))">
    </asp:SqlDataSource>
    <br />
</div>
</form>
<div id="footer">
    <a href="Default.aspx">Return to HOME PAGE</a> |
</div>
</body>
</html>

```

5.2.6.2 Search.aspx.cs

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;

```

```

using System.Web.UI.HtmlControls;
using System.Data.SqlClient;
using System.IO;

public partial class Search_Clip : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void B_Search_Click(object sender, EventArgs e)
    {
        System.Collections.Generic.List<int> SmallList = new
System.Collections.Generic.List<int>(); //used for the ids of the selected query clips
        //used for the ids of the selected database videos
        System.Collections.Generic.List<int> BigList = new
System.Collections.Generic.List<int>();
        //used for the binary files of the selected query clips
        System.Collections.Generic.List<Byte[]> big_data = new
System.Collections.Generic.List<Byte[]>();
        //used for the binary files of the selected database videos
        System.Collections.Generic.List<Byte[]> small_data = new
System.Collections.Generic.List<Byte[]>();
        //used for the names of the selected query clips
        System.Collections.Generic.List<String> SmallListNames = new
System.Collections.Generic.List<String>();
        //used for the names of the selected database videos
        System.Collections.Generic.List<String> BigListNames = new
System.Collections.Generic.List<String>();
        //used for the fps parametre of the selected database videos
        System.Collections.Generic.List<Double> FpsList = new
System.Collections.Generic.List<Double>();
        Byte[] barr;

```

```

String str_file_name =
System.IO.Path.GetFileName(File1.PostedFile.FileName);
//find the checked query clips and add their id to the "small" videos' ids list
for (int i = 0; i < GV_small.Rows.Count; i++)
{
    CheckBox ck = (CheckBox)GV_small.Rows[i].FindControl("CheckBox1");
    Label lb = (Label)GV_small.Rows[i].FindControl("check1");
    int id_small = Int32.Parse(lb.Text);
    bool isSelected = ck.Checked;

    if (isSelected == true)
    {
        SmallList.Add(id_small);
    }
}

//find the checked database videos and add their ids to the "big" videos' ids list
for (int j = 0; j < GV_big.Rows.Count; j++)
{
    CheckBox ck = (CheckBox)GV_big.Rows[j].FindControl("CheckBox2");
    Label lb = (Label)GV_big.Rows[j].FindControl("check2");
    int id_big = Int32.Parse(lb.Text);
    bool isSelected = ck.Checked;

    if (isSelected == true)
    {
        BigList.Add(id_big);
    }
}

//retrieve from the database the name and the binary file of the selected
//query clips

```

```

        SqlConnection conn = new
SqlConnection(IConstants.CONNECTION_STRING);
        conn.Open();

        for (int k = 0; k < SmallList.Count; k++)
        {
            SqlCommand cmd = new SqlCommand("SELECT txt,name FROM clips
where id= " + SmallList[k] + ";", conn);
            SqlDataReader rdr = cmd.ExecuteReader();
            while (rdr.Read())
            {
                Byte[] txt_small = (Byte[])rdr[0];
                String name_checked = (String)rdr[1];
                small_data.Add(txt_small);
                SmallListNames.Add(name_checked);
            }
            rdr.Close();
        }

        TimeSpan idxTime = new TimeSpan(0);
        //check if there is a file in the browse textbox
        if (str_file_name != "")
        {
            DateTime t1 = DateTime.Now;

            try
            {
                File1.PostedFile.SaveAs(IConstants.SAVED_VIDEO_PATH +
str_file_name);
            }
            catch (Exception ex)
            {
                Label1.Text = ex.Message;
                return;
            }
        }
    }
}

```

```

    }
    String katalixi = str_file_name.Substring(str_file_name.LastIndexOf(".") + 1);

    //if the file is not a fingerprint do the indexing process(mencoder-ffmpeg)
    if (katalixi != "ctxt")
    {
        String str_clip_url = IConstants.SAVED_VIDEO_PATH +
str_file_name;

        System.Diagnostics.Process mencoder = new
System.Diagnostics.Process();
        mencoder.StartInfo.FileName = @"C:\util\mencoder.exe";
        mencoder.StartInfo.Arguments = IConstants.SAVED_VIDEO_PATH
+ str_file_name + " -o " + IConstants.SAVED_VIDEO_PATH + "H264c -nosound -
quiet -of rawvideo -oac copy -vf lavcdeint,scale=320:240,crop=240:176 -ovc x264 -
x264encopts qp=26:nocabac:nochroma-
me:analyse=p8x8,i4x4:threads=auto:keyint=9999999:no-psnr:no-ssim:scenecut=-1";
        mencoder.StartInfo.UseShellExecute = false;
        mencoder.StartInfo.RedirectStandardOutput = true;
        mencoder.Start();
        String m = mencoder.StandardOutput.ReadToEnd();
        mencoder.WaitForExit();
        String[] words_mencoder = m.Split(' ');
        String str_fps = null;

        for (int i = 0; i < words_mencoder.Length; i++)
        {
            if (words_mencoder[i].Trim().Equals("fps"))
                str_fps = words_mencoder[i - 1];
        }
        IFormatProvider ifp = new System.Globalization.CultureInfo("en-US");
        double fps;
        try
        {

```



```

        fps = Double.Parse(str_fps, ifp);
    }
    catch (Exception ex)
    {
        fps = 25;
    }
    System.Diagnostics.Process ffmpeg = new System.Diagnostics.Process();
    ffmpeg.StartInfo.FileName = @"C:\util\ffmpeg-v2.exe";
    ffmpeg.StartInfo.Arguments = "-f h264 -i " +
IConstants.SAVED_VIDEO_PATH + "H264c -f null a.avi";
    ffmpeg.StartInfo.UseShellExecute = false;
    ffmpeg.StartInfo.RedirectStandardOutput = false;
    ffmpeg.Start();
    ffmpeg.WaitForExit();
    StreamReader re = File.OpenText("out.ctxt");
    String s = re.ReadToEnd();
    re.Close();

    System.Collections.Generic.List<int> intraList = new
System.Collections.Generic.List<int>();
    String[] words = s.Split('\n');
    for (int i = 0; i < words.Length; i++)
    {
        if (!(words[i].IndexOf(';') > 0)) break;
        String word = words[i].Substring(words[i].IndexOf(';') + 1);
        IFormatProvider ifp1 = new System.Globalization.CultureInfo("en-
US");

        double d = Double.Parse(word, ifp1);
        int num = (int)(100.0 * d + 0.5);
        System.Diagnostics.Debug.Assert(num >= 0 && num <= 100);
        intraList.Add(num);
    }
    int num_frames = words.Length - 1;
    double duration = num_frames / fps;

```

```

barr = new Byte[intraList.Count];

for (int i = 0; i < intraList.Count; i++)
{
    barr[i] = (Byte)intraList[i];
}
} //end if (katalixi != "ctxt")

//if the file is a ctxt document (fingerprint)
else
{
    String str_clip_url = IConstants.SAVED_VIDEO_PATH +
str_file_name;

    StreamReader re = File.OpenText(str_clip_url);
    String input = re.ReadToEnd();
    System.Collections.Generic.List<int> largeList = new
System.Collections.Generic.List<int>();
    String[] words = input.Split('\n');
    for (int i = 0; i < words.Length; i++)
    {
        if (!(words[i].IndexOf(';') > 0)) break;
        String word = words[i].Substring(words[i].IndexOf(';') + 1);
        IFormatProvider ifp1 = new System.Globalization.CultureInfo("en-
US");

        double d = Double.Parse(word, ifp1);
        int num = (int)(100.0 * d + 0.5);
        System.Diagnostics.Debug.Assert(num >= 0 && num <= 100);
        largeList.Add(num);
    }
    int num_frames = words.Length - 1;
    barr = new Byte[largeList.Count];
    for (int i = 0; i < largeList.Count; i++)
    {
        barr[i] = (Byte)largeList[i];
    }
}

```

```

    }
}

//addition of the name and the binary file at the appropriate lists
small_data.Add(barr);
SmallListNames.Add(str_file_name);
DateTime t2 = DateTime.Now;
idxTime = t2 - t1; //used for calculation of needed indexing time
} //telos if (str_file_name != null)

//retrieve from the database the name and the binary files of the selected
//database videos
for (int l = 0; l < BigList.Count; l++)
{
    SqlCommand cmd = new SqlCommand("SELECT txt,name,fps FROM
clips where id= " + BigList[l] + ";", conn);
    SqlDataReader rdr = cmd.ExecuteReader();

    while (rdr.Read())
    {
        Byte[] txt_big = (Byte[])rdr[0];
        String name_checked = (String)rdr[1];
        Double clip_fps = (Double)rdr[2];
        big_data.Add(txt_big);
        BigListNames.Add(name_checked);
        FpsList.Add(clip_fps);
    }
    rdr.Close();
}

//send to the results.aspx the appropriate data
Session.Add("IndexingTime", idxTime);
Session.Add("SmallData", small_data); //binary data
Session.Add("BigData", big_data);

```

```

        Session.Add("ClipFound", SmallListNames); //names of clips
        Session.Add("WhereFound", BigListNames);
        Session.Add("FpsBig", FpsList); //fps
        Response.Redirect("Results.aspx");
    }
}

```

5.2.7 Πηγαίος κώδικας της σελίδας «Results»

5.2.7.1 Κώδικας μορφοποίησης της σελίδας Results.aspx

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Results.aspx.cs" Inherits="Results" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>Search Demo</title>
<link rel="stylesheet" type="text/css" href="~/sinorca-screen.css"
media="screen" title="Sinorca (screen)" />
</head>
<body>

<!-- ##### Header ##### -->
<div id="header">
<div class="superHeader">
<div class="midHeader">
<h1 class="headerTitle">The results</h1>
</div>

<div class="subHeader">
<span class="doNotDisplay">Navigation:</span>

```

```

        <a href="Upload.aspx">Upload</a> |
        <span class="doNotDisplay">Navigation:</span>
        <a href="Search_Clip.aspx">Search</a> |
        <span class="doNotDisplay">Navigation:</span>
        <a href="View_clips.aspx">View our database</a> |
    </div>
</div>
</div>
<br /><br /><br /><br /><br /><br />

    <asp:Label ID="ResultsLabel" runat="server" Text=""
ForeColor="#153E7E"></asp:Label>&nbsp;
    <br /><br />
    <form id="form1" runat="server">
    <div>
    <br /><br /><br />
    </div>
    </form>
    <div id="footer">
        <a href="Search_Clip.aspx">Go back</a> |
    </div>
</body>
</html>

```

5.2.7.2 Results.aspx.cs

```

using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;

```

```

using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Web.SessionState;
using System.Collections.Generic;

public partial class Results : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        //names of selected query clips
        List<String> ClipsFound = (List<String>)Session["ClipFound"];
        //names of selected database videos
        List<String> WhereFound = (List<String>)Session["WhereFound"];
        //binary data of selected query clips
        List<Byte[]> SmallData = (List<Byte[]>)Session["SmallData"];
        //binary data of selected database videos
        List<Byte[]> BigData = (List<Byte[]>)Session["BigData"];
        //fps of big videos
        List<Double> FpsList = (List<Double>)Session["FpsBig"];
        //time needed for the indexing process
        TimeSpan idxTime = (TimeSpan)Session["IndexingTime"];

        CResults search_results;
        System.Text.StringBuilder sb = new System.Text.StringBuilder();

        for (int m = 0; m < SmallData.Count; m++)
        {
            sb.AppendFormat("<h3>Query clip: {0}</h3>\r\n", ClipsFound[m]);
            bool found = false;
            DateTime t1 = DateTime.Now;
            for (int n = 0; n < BigData.Count; n++)
            {
                //process of searching...
                search_results = CSearch.Search(SmallData[m], BigData[n], 20);
            }
        }
    }
}

```

```

        if (search_results == null)
        {
            sb.AppendFormat("Error:This video cannot be searched into the
video {0}",WhereFound[n]);
            sb.Append("<br />");
        }
        else
        {
            for (int idx = 0; idx < search_results.FoundNum; idx++)
            {

                //calculation of time of the exact match
                Double d_second = search_results.GetFrameResult(idx) /
FpsList[n];

                int second = Convert.ToInt32(d_second);
                int hours = second / 3600;
                int mins = (second - hours * 3600) / 60;
                int secs = second - hours * 3600 - mins * 60;
                String timetime = hours.ToString("00") + ":" +
mins.ToString("00") + ":" + secs.ToString("00");
                int weight =
Convert.ToInt32(search_results.GetWeightResult(idx));
                String curr_weight = weight.ToString();
                sb.AppendFormat("Found at: {0}", WhereFound[n]);

                sb.AppendFormat(" at : {0}", timetime);
                sb.AppendFormat(" with loss: {0}", curr_weight);
                sb.Append("<br />");
                found = true;
            }

        } //end else
    } //end for (int n = 0; n < BigData.Count; n++)

```

```

        if (found == false)
        {
            sb.AppendFormat("Not found");
            sb.Append("<br />");
        }

        DateTime t2 = DateTime.Now;
        TimeSpan diff = t2 - t1;

        Double diafora = diff.TotalMilliseconds;
        String str_diafora = diafora.ToString();
        sb.AppendFormat("Search time: {0} milliseconds", str_diafora);
        sb.Append("<hr />\r\n");

    } //end for (int m = 0; m < SmallData.Count; m++)

    sb.AppendFormat("Indexing time = {0} milliseconds.",
idxTime.TotalMilliseconds);
    ResultsLabel.Text = sb.ToString();
}

}

```

5.3 Προδιαγραφές του sinorca.css

Sinorca.css

```

/*****
* TITLE: Sinorca Screen Stylesheet *
* URI : sinorca/sinorca-screen.css *
* MODIF: 2003-Apr-30 19:31 +0800 *

```



```

*****/

/* ##### Common Styles ##### */
body {
    color: black;
    background-color: rgb(240,240,240);
    font-family: verdana, helvetica, arial, sans-serif;
    font-size: 71%; /* Enables font size scaling in MSIE */
    margin: 0;
    padding: 0;
}

html > body {
    font-size: 10.5pt;
}

acronym, .titleTip {
    border-bottom: 1px dotted rgb(153,153,153);
    cursor: help;
    margin: 0;
    padding: 0 0 0.4px 0;
}

.doNotDisplay {
    display: none;
}

.smallCaps {
    font-size: 110%;
    font-variant: small-caps;
}

/* ##### Header ##### */
.superHeader {

```

```
color: white;
background-color: rgb(100,135,220);
height: 2em;
}
```

```
.superHeader a {
color: white;
background-color: transparent;
text-decoration: none;
font-size: 91%;
margin: 0;
padding: 0 0.5ex 0 0.25ex;
}
```

```
.superHeader a:hover {
text-decoration: underline;
}
```

```
.superHeader .left {
position: absolute;
left: 1.5mm;
top: 0.75ex;
}
```

```
.superHeader .right {
position: absolute;
right: 1.5mm;
top: 0.75ex;
}
```

```
.midHeader {
color: rgb(39,78,144);
background-color: rgb(140,170,230);
}
```

```

.headerTitle {
  font-size: 337%;
  font-weight: normal;
  margin: 0 0 0 4mm;
  padding: 0.25ex 0;
}

.subHeader {
  color: white;
  background-color: rgb(0,51,153);
  margin: 0;
  padding: 1ex 1ex 1ex 1.5mm;
}

.subHeader a {
  color: white;
  background-color: transparent;
  text-decoration: none;
  font-weight: bold;
  margin: 0;
  padding: 0 0.75ex 0 0.5ex;
}

.subHeader a:hover {
  text-decoration: underline;
}

.superHeader .highlight, .subHeader .highlight {
  color: rgb(253,160,91);
  background-color: transparent;
}

/* ##### Side Bar ##### */
#side-bar {
  width: 15em;
}

```

```
float: left;
clear: left;
border-right: 1px solid rgb(153,153,153);
}
```

```
#side-bar div {
border-bottom: 1px solid rgb(153,153,153);
}
```

```
.sideBarTitle {
font-weight: bold;
margin: 0 0 0.5em 2.5mm;
padding: 1em 0 0 0;
}
```

```
#side-bar ul {
list-style-type: none;
list-style-position: outside;
margin: 0;
padding: 0 0 1.1em 0;
}
```

```
#side-bar li {
margin: 0;
padding: 0.1ex 0; /* Circumvents a rendering bug (?) in MSIE 6.0 */
}
```

```
#side-bar a, .thisPage {
color: rgb(0,102,204);
background-color: transparent;
text-decoration: none;
margin: 0;
padding: 0.75em 1ex 0.75em 5mm;
display: block;
```

```
}
```

```
.thisPage {  
  color: black;  
  background-color: white;  
  padding-left: 4mm;  
  border-top: 1px solid rgb(153,153,153);  
  border-bottom: 1px solid rgb(153,153,153);  
}
```

```
#side-bar a:hover {  
  color: white;  
  background-color: rgb(100,135,220);  
  text-decoration: none;  
}
```

```
.sideBarText {  
  line-height: 1.5em;  
  margin: 0 0 1em 0;  
  padding: 0 1.5ex 0 2.5mm;  
  display: block;  
}
```

```
#side-bar .sideBarText a {  
  text-decoration: underline;  
  margin: 0;  
  padding: 0;  
  display: inline;  
}
```

```
#side-bar .sideBarText a:hover {  
  color: rgb(0,102,204);  
  background-color: transparent;  
  text-decoration: none;
```

```

}

.lighterBackground {
  color: inherit;
  background-color: white;
}

/* ##### Main Copy ##### */

#main-copy {
  color: black;
  background-color: white;
  text-align: justify;
  line-height: 1.5em;
  margin: 0 0 0 15em;
  padding: 0.5mm 5mm 5mm 5mm;
  border-left: 1px solid rgb(153,153,153);
}

#main-copy p {
  margin: 1em 1ex 2em 1ex;
  padding: 0;
}

#main-copy a {
  color: rgb(0,102,204);
  background-color: transparent;
  text-decoration: underline;
}

#main-copy a:hover {
  text-decoration: none;
}

```

```
}
```

```
#main-copy h1 {  
  color: white;  
  background-color: rgb(100,135,220);  
  font-size: 100%;  
  font-weight: bold;  
  margin: 3em 0 0 0;  
  padding: 0.5ex 0 0.5ex 1ex;  
}
```

```
#main-copy .topOfPage {  
  color: white;  
  background-color: transparent;  
  font-size: 91%;  
  font-weight: bold;  
  text-decoration: none;  
  margin: 2.5ex 1ex 0 0; /* For MSIE */  
  padding: 0;  
  float: right;  
}
```

```
#main-copy > .topOfPage {  
  margin: 2.75ex 1ex 0 0; /* For fully standards-compliant user agents */  
}
```

```
dl {  
  margin: 1em 1ex 2em 1ex;  
  padding: 0;  
}
```

```
dt {  
  font-weight: bold;  
  margin: 0 0 0 0;
```

```

padding: 0;
}

dd {
margin: 0 0 2em 2em;
padding: 0;
}

/* ##### Footer ##### */
#footer {
color: white;
background-color: rgb(100,135,220);
font-size: 91%;
margin: 0;
padding: 1em 2.5mm 2.5ex 2.5mm;
clear: both;
}

#footer .left {
line-height: 1.45em;
float: left;
clear: left;
}

#footer .right {
text-align: right;
line-height: 1.45em;
}

#footer a {
color: white;
background-color: transparent;
text-decoration: underline;
}

```



```
#footer a:hover {  
    text-decoration: none;  
}
```

5.4 Πηγαίος κώδικας για windows application uploader

5.4.1 Program.cs

```
using System;  
using System.Collections.Generic;  
using System.Windows.Forms;  
  
namespace WindowsApplicationUploader  
{  
    static class Program  
    {  
        static void Main()  
        {  
            Application.EnableVisualStyles();  
            Application.SetCompatibleTextRenderingDefault(false);  
            Application.Run(new Form1());  
        }  
    }  
}
```

5.4.2 Form1.cs

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;
```

```

using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Data.SqlClient;
using System.Configuration;
using System.Collections;
using System.Data.Odbc;
using System.Diagnostics;
using System.Runtime.InteropServices; // DllImport

namespace WindowsApplicationUploader
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'videosDataSet1.type' table. You
            can move, or remove it, as needed.
            this.typeTableAdapter1.Fill(this.videosDataSet1.type);
            // TODO: This line of code loads data into the 'videosDataSet.type' table. You
            can move, or remove it, as needed.
            this.typeTableAdapter.Fill(this.videosDataSet.type);
        }

        private void Browse_Click(object sender, EventArgs e)
        {
            //creation of browse textbox and button
            OpenFileDialog fdlg = new OpenFileDialog();
            fdlg.Title = "Open File";
            fdlg.InitialDirectory = @"c:\";
            fdlg.Filter = "All files (*.*)|*.*|All files (*.*)|*.*";
        }
    }
}

```

```

fdlg.FilterIndex = 2;
fdlg.RestoreDirectory = true;
if (fdlg.ShowDialog() == DialogResult.OK)
{
    TB_browse.Text = fdlg.FileName;
}
}

private void Upload_Click(object sender, EventArgs e)
{
    String str_browse_name = TB_browse.Text;
    String str_file_name =
str_browse_name.Substring(str_browse_name.LastIndexOf("\\") + 1);
    //check if none of the ckeckboxes is checked
    if (CB_big.Checked == false && CB_small.Checked == false)
    {
        label2.Text = "Sorry try again...you must check at least one of the checkboxes
above";
    }
    else
    {
        String clip_name = TB_name.Text;
        clip_name = clip_name.Replace(' ', '_');
        String clip_info = TB_info.Text;
        String katalixi = str_file_name.Substring(str_file_name.LastIndexOf(".") + 1);

        //initial clip's url
        String str_clip_url = IConstants.SAVED_VIDEO_PATH + str_file_name;
        FileInfo fi = new FileInfo(TB_browse.Text);
        File.Copy(TB_browse.Text, str_clip_url,true);
        //External process doMencoder
        String m = CDoCrop.doMencoder(str_clip_url);
        String[] words_mencoder = m.Split(' ');

```

```

String str_fps = null;
double fps;
//fps from the output of mencoder
for (int i = 0; i < words_mencoder.Length; i++)
{
    if (words_mencoder[i].Trim().Equals("fps"))
        str_fps = words_mencoder[i - 1];
}
IFormatProvider ifp = new System.Globalization.CultureInfo("en-US");
try
{
    fps = Double.Parse(str_fps, ifp);
}
catch (Exception ex)
{
    fps = 25;
}
//external process doffmpeg
String s = CDoCrop.doFfmpeg();
System.Collections.Generic.List<int> intraList = new Sys-
tem.Collections.Generic.List<int>();
System.Collections.Generic.List<int> FramesList = new Sys-
tem.Collections.Generic.List<int>();

//suitable of the ffmpeg output and creation of binary file for the database
String[] words = s.Split('\n');

for (int i = 0; i < words.Length; i++)
{
    if (!(words[i].IndexOf(';') > 0)) break;
    String word = words[i].Substring(words[i].IndexOf(';') + 1);
    IFormatProvider ifp1 = new System.Globalization.CultureInfo("en-US");
    double d = Double.Parse(word, ifp1);
    int num = (int)(100.0 * d + 0.5);
}

```

```

System.Diagnostics.Debug.Assert(num >= 0 && num <= 100);
intraList.Add(num);
//frames that represent scene changes and are used for image extraction
if (num > 90)
{
    FramesList.Add(i);
}
}
Byte[] barr = new Byte[intraList.Count];
TextWriter tw = new StreamWriter("a.ctxt");
for (int i = 0; i < intraList.Count; i++)
{
    tw.Write(intraList[i] + "\r\n");
    barr[i] = (Byte)intraList[i];
}
tw.Close();
File.Copy("a.ctxt", IConstants.OUTPUT_PATH + clip_name + ".ctxt", true);

//used for the number of the video frames
int num_frames = words.Length - 1;
double duration = num_frames / fps; //used for clip's duration
int clip_type_index = listBox1.SelectedIndex;

//check which one of the checkboxes is checked
int search13 = 0;
if (CB_big.Checked == true)
search13 += 1;

if (CB_small.Checked == true)
search13 += 2;

//creation of images
Boolean not_img_created = false;

```

```

        not_img_created = CExtractImage.Save_Image(str_clip_url, clip_name,
FramesList, fps);

        //insert in the databse
        Decimal id = CInsert.Insert(clip_name, katalixi, barr, fps, duration,
not_img_created, clip_type_index, clip_info, search13, FramesList.Count);

        //rename video_file and images according the id
        String str_id_url = IConstants.SAVED_VIDEO_PATH + clip_name + "_id"
+ id + "." + katalixi;

        File.Move(str_clip_url, str_id_url);
        CExtractImage.Rename_Image(clip_name, id, katalixi);
        label1.Text = "The " + clip_name + " clip has been successfully uploaded!!!!";
    }
    return;

}

} //end partial class Form1

} // end namespace

```