

Πανεπιστήμιο Κρήτης  
Σχολή Θετικών Επιστημών  
Τμήμα Επιστήμης Υπολογιστών

---

## **Διπλωματική Εργασία**

---

Μελέτη και ανάπτυξη αλγορίθμων υλοποίησης εξατομικευμένων συστάσεων προς χρήστες με βάση τις προτιμήσεις τους και χρησιμοποίηση τους για σχηματισμό εικονικών κοινοτήτων στο Διαδίκτυο  
&  
Ανάπτυξη του συστήματος MRS (Movie Recommendation System) για εφαρμογή των αλγορίθμων στην περιοχή ενδιαφέροντος «ταινίες κινηματογράφου»

---

**Παπαγγελής Εμμανουήλ [Α.Μ. 1055]**

Επιβλέπων Καθηγητής  
**Πλεξουσάκης Δημήτρης**

Ηράκλειο, Οκτώβριος 2002



## Πίνακας Περιεχομένων

---

<b>1. Εισαγωγικά</b>	<b>6</b>
<b>2. Περιγραφή</b>	<b>7</b>
<b>3. Τεχνολογίες που χρησιμοποιήθηκαν</b>	<b>9</b>
3.1 Apache Web Server	9
3.2 PHP Scripting Language	9
3.3 MSSQL Server 2000 (Personal Edition)	12
<b>4. Αρχιτεκτονική συστήματος</b>	<b>13</b>
4.1 Μοντέλο Πελάτη / Εξυπηρετητή (Client/Server Μοντέλο)	13
4.2 Τεχνική Λεξικογραφικής Ανάλυσης (Parsing technique)	15
4.3 Περιορισμοί Διαδικτύου	15
<b>5. Ανάλυση τεχνικών χαρακτηριστικών Συστήματος</b>	<b>17</b>
5.1 Εισαγωγικά	17
5.2 Sessions	17
5.3 Database Class	17
5.4 Graph Class	18
5.5 Function Library	18
5.6 Template File	18
5.7 Cascade Style Sheet File	18
5.8 Configuration File	18
5.9 Διαφορετικοί Ρόλοι Χρηστών (Εγγεγραμμένος, Μη-Εγγεγραμμένος, Διαχειριστής)	19
<b>6. Σχεδίαση Βάσης</b>	<b>20</b>
6.1 Περιγραφή	20
6.2 Σχήμα Οντοτήτων-Σχέσεων Βάσης Δεδομένων	20
6.3 Σχεσιακό Μοντέλο της βάσης δεδομένων	21
6.4 Στιγμιότυπο της υλοποιημένης βάσης δεδομένων	23
<b>7. Λειτουργίες Συστήματος</b>	<b>25</b>
7.1 Login	25
7.2 My Profile	25
7.3 Browse Movies	27
7.4 Rate Movies	28
7.5 Rating Prediction	29
7.6 Recommendations	30
7.7 Find Buddies	31

7.8 Communities	33
7.9 Imdb	34
7.10 Statistics	35
7.11 Query Tool	35
7.12 Logout	36
<b>8. Αλγόριθμοι Υλοποίησης Συστάσεων</b>	<b>37</b>
8.1 Αλγόριθμος Υλοποίησης Συστάσεων 1	37
8.2 Αλγόριθμος Υλοποίησης Συστάσεων 2 (Collaborative Filtering)	38
8.3 Αλγόριθμος Υλοποίησης Συστάσεων 3	40
8.4 Ανάλυση και συμπεράσματα από την εφαρμογή των αλγορίθμων συστάσεων	42
8.5 Θέματα απόδοσης και αξιοπιστίας αλγορίθμων	43
<b>9. Άλλοι αλγόριθμοι</b>	<b>46</b>
9.1 Αλγόριθμος Πρόβλεψης Ψήφου	46
9.2 Αλγόριθμος εύρεσης «φίλων» με βάση τις προτιμήσεις σε κατηγορίες ταινιών	47
9.3 Αλγόριθμος εύρεσης χρηστών που ικανοποιούν συγκεκριμένα κριτήρια (φίλτρα) όσον αφορά τις προτιμήσεις τους σε συγκεκριμένες κατηγορίες ταινιών	48
9.4 Κοινότητες χρηστών και βαθμός συνοχής κοινότητας	49
<b>10. Μελλοντικές Επεκτάσεις</b>	<b>51</b>
<b>Παράρτημα Α (Ενδεικτικός Κώδικας)</b>	<b>52</b>
A1. Η κλάση που υλοποιεί τις διάφορες συναλλαγές με τη βάση	52
A2. Το αρχείο διαμόρφωσης (configuration file)	55
A3. Συνάρτηση συντακτικής ανάλυσης σελίδων του IMDB (IMDB parser)	56
A4. Υπολογισμός της βαθμολογίας MRS	58
A5. Αλγόριθμος υλοποίησης πρώτης μεθόδου συστάσεων	59
A6. Υπολογισμός συντελεστή συσχέτισης Pearson (Pearson Correlation Coefficient)	60
A7. Αλγόριθμος Υπολογισμού Συσχέτισης δύο Προφίλ με βάση τις βαθμολογήσεις τους	61
A8. Αλγόριθμος Υπολογισμού Συσχέτισης δύο Προφίλ με βάση τις προτιμήσεις τους σε συγκεκριμένες κατηγορίες	62
A9. Αλγόριθμος εύρεσης φίλων με βάση τις προτιμήσεις τους σε κατηγορίες ταινιών	62
A10. Αλγόριθμος υπολογισμού δεύτερης μεθόδου συστάσεων (Collaborative Filtering)	63
A11. Αλγόριθμος υλοποίησης της τρίτης μεθόδου συστάσεων	65
A12. Αλγόριθμος που επιστρέφει το ποσοστό της θετικής προτίμησης ενός χρήστη σε μία κατηγορία ταινίας	68
A13. Αλγόριθμος που υπολογίζει εάν ένας χρήστης ικανοποιεί τα κριτήρια του φιλτραρίσματος ή όχι	68

A14. Αλγόριθμος Υπολογισμού Πρόβλεψης ψήφου	69
A15. Υπολογισμός της δραστηριότητας του χρήστη (κατανομή των ψήφων)	71
<b>Πίνακας Διαγραμμάτων</b>	<b>73</b>
<b>Βιβλιογραφία</b>	<b>74</b>
Βιβλία	74
Δημοσιεύσεις	74
Διαδικτυακές Πηγές	76

## 1. Εισαγωγικά

---

Τα τελευταία χρόνια γίνεται πολύ συχνά λόγος για συστήματα συστάσεων (recommendation systems). Τα συστήματα αυτά απασχόλησαν στο παρελθόν και απασχολούν ακόμη ερευνητές καθώς και εμπορικές δραστηριότητες. Σήμερα με την ολοένα και μεγαλύτερη εισροή δεδομένων στον παγκόσμιο ιστό και με την εκθετική αύξηση του όγκου πληροφορίας την οποία δέχεται ο χρήστης τα συστήματα αυτά αποκτούν ακόμη μεγαλύτερη σημασία.

Ο σκοπός της ανάπτυξης τέτοιων συστημάτων είναι η επιλεκτική προώθηση πληροφορίας στο χρήστη με κύριο γνώμονα την προσωπικότητα του. Μέσα από τις επιλογές του χρήστη το σύστημα αποκτάει γνώση για την ιδιαιτερότητα του και εκμεταλλευόμενο τη γνώση αυτή μπορεί να του κάνει συστάσεις.

Διάφορες τεχνικές που έχουν αναπτυχθεί για συστήματα συστάσεων κάνουν χρήση αλγορίθμων οι οποίοι υπολογίζουν το βαθμό συσχέτισης των προφίλ χρηστών. Με τον τρόπο αυτό όμως υπονοείται η δημιουργία εικονικών κοινοτήτων, δηλαδή ομάδων χρηστών με παρόμοιο προφίλ.

Η εργασία αυτή έχει σκοπό την ανάπτυξη αλγορίθμων και τεχνικών για δημιουργία συστάσεων, για ταινίες του κινηματογράφου, στον χρήστη καθώς επίσης την δημιουργία δυναμικών κοινοτήτων χρηστών.

## 2. Περιγραφή

---

Στα πλαίσια της εργασίας αυτής αναπτύχθηκε ένα σύστημα συστάσεων για ταινίες κινηματογράφου. Ο τομέας αυτός πέρα από το ότι αποτελεί από μόνος του πόλο έλξης χρηστών καθώς είναι γνωστό ότι οι φίλοι του κινηματογράφου είναι από τις πιο δραστήριες κοινότητες στον πραγματικό κόσμο, αλλά και στο Διαδίκτυο, μπορεί και συμπεριλαμβάνει χαρακτηριστικά κατάλληλα για ερευνητική δουλειά (κατηγοριοποιήσεις, έννοια της βαθμολόγησης, έννοια των προφίλ).

Έτσι λοιπόν, αναπτύχθηκε ένας δικτυακός τόπος μέσα από τον οποίο κάποιος μπορεί να περιηγηθεί ανάμεσα στις ταινίες και να διαβάσει περιγραφές για αυτές, μπορεί να ψηφίσει ταινίες συμβάλλοντας ενεργά στην διαμόρφωση των βαθμολογιών, μπορεί να πάρει συστάσεις για το ποιες ταινίες αξίζει να δει.

Ένας από τους βασικούς διαχωρισμούς τους οποίους κάνει το σύστημα είναι μεταξύ μελών και μη μελών. Όταν ένας χρήστης γίνεται μέλος του συστήματος τότε άμεσα ο ρόλος του γίνεται πιο ουσιαστικός. Η αλληλεπίδρασή του με το σύστημα είναι πιο ουσιαστική γιατί πλέον είναι αναγνωρίσιμος και αυτό δίνει τη δυνατότητα στο σύστημα να εκμεταλλευτεί την οποιαδήποτε πληροφορία προκύπτει από τη δραστηριότητα του ώστε να κατανοήσει στον μεγαλύτερο δυνατό βαθμό τις ιδιαιτερότητές του. Με τον τρόπο αυτό το σύστημα βελτιώνει τις συστάσεις που κάνει προς τον χρήστη με άμεσο όφελος για αυτόν. Όμως το ότι ένας χρήστης επιλέγει να γίνει μέλος σημαίνει επιπλέον ότι η άποψη του ακριβώς επειδή δεν είναι ανώνυμη έχει μεγαλύτερο βάρος. Ο χρήστης μέλος πλέον μπορεί να καθορίζει ουσιαστικά την βαθμολογία των ταινιών.

Αποτέλεσμα του να είναι κάποιος μέλος της κοινότητας είναι να μπορεί να βρει φίλους οι οποίοι έχουν παρόμοιες προτιμήσεις στις ταινίες του κινηματογράφου και να γίνεται δέκτης διαφόρων τρόπων συστάσεων, οι οποίοι τελικά καταλήγουν στο να προτείνουν ταινίες στον χρήστη που πραγματικά αυτός θα ήθελε να δει. Ακόμη μπορεί γραφικά να δει τη δραστηριότητα του, όσον αφορά την κατανομή των ψήφων που έχει δώσει, καθώς και το ιστορικό της βαθμολογίας.

Επιτακτική ήταν επίσης η ανάπτυξη κάποιων λειτουργιών για την εύκολη διαχείριση του συστήματος αλλά και για την διευκόλυνση της παρακολούθησης της κίνησης στο δικτυακό τόπο η οποία πραγματοποιήθηκε με την ανάπτυξη ενός λογαριασμού

διαχείρισης του συστήματος την οποία μπορεί να ελέγχει μόνο όποιος έχει τα κατάλληλα προνόμια (privileges).

Η εργασία αυτή επιδιώκει να έχει διπλό ρόλο. Πρώτα να αποτελέσει πεδίο ερευνητικής δουλειάς για τον τρόπο με τον οποίο από τη δραστηριότητα ενός χρήστη μπορεί να αναπτυχθεί το προφίλ του και αυτό να αποτελέσει αντικείμενο ανάλυσης για τη δημιουργία εξατομικευμένων συστάσεων, δεύτερον να αποτελέσει έναν ευχάριστο τόπο στον οποίο κάποιος μπορεί έγκυρα να ενημερωθεί για τις ταινίες που τον ενδιαφέρουν, αλλά και να αναπτύξει φιλίες με ανθρώπους οι οποίοι έχουν κοινά ενδιαφέροντα.



### 3. Τεχνολογίες που χρησιμοποιήθηκαν

---

Το Movie Recommendation System (MRS) είναι ένα online-σύστημα. Η δημιουργία του στηρίχτηκε σε σύγχρονες τεχνολογίες για δημιουργία συστημάτων για το Διαδίκτυο. Αναλυτικά Χρησιμοποιήθηκαν:

- Apache Web Server
- PHP Scripting Language
- MSSQL Server 2000 (Personal Edition)

#### 3.1 Apache Web Server

Ο Apache HTTP Server είναι ο δημοφιλέστερος web server στο Διαδίκτυο από τον Απρίλιο του 1996, συγκεντρώνοντας σήμερα πάνω από το 60% του συνόλου. Ο Apache HTTP Server αποτελεί μία προσπάθεια ανάπτυξης και συντήρησης ενός ανοικτού-κώδικα HTTP server για σύγχρονα λειτουργικά συστήματα όπως, Unix και Windows NT. Ο σκοπός της προσπάθειας είναι να αναπτυχθεί ένας ασφαλής, αποδοτικός και επεκτάσιμος server ο οποίος να μπορεί να παρέχει σύγχρονες υπηρεσίες με βάση τα υπάρχοντα HTTP standards.

*Παρακάτω φαίνεται η δημοτικότητα του Apache Web Server έναντι των υπολοίπων web servers.*

Server	July 2002 Count	July 2002 %	June 2002 Count	June 2002 %	Growth %
Apache 1.3	5193016	61.67%	5128994	64.19%	-3.92%
IIS 4.0	619342	7.36%	614902	7.69%	-4.41%
IIS 5.0	1258196	14.94%	1189187	14.88%	0.41%
Apache 1.2	56078	0.67%	66624	0.83%	-20.12%
Netscape-Ent 3	53024	0.63%	53893	0.67%	-6.63%

πηγή: [www.securityspace.com](http://www.securityspace.com)

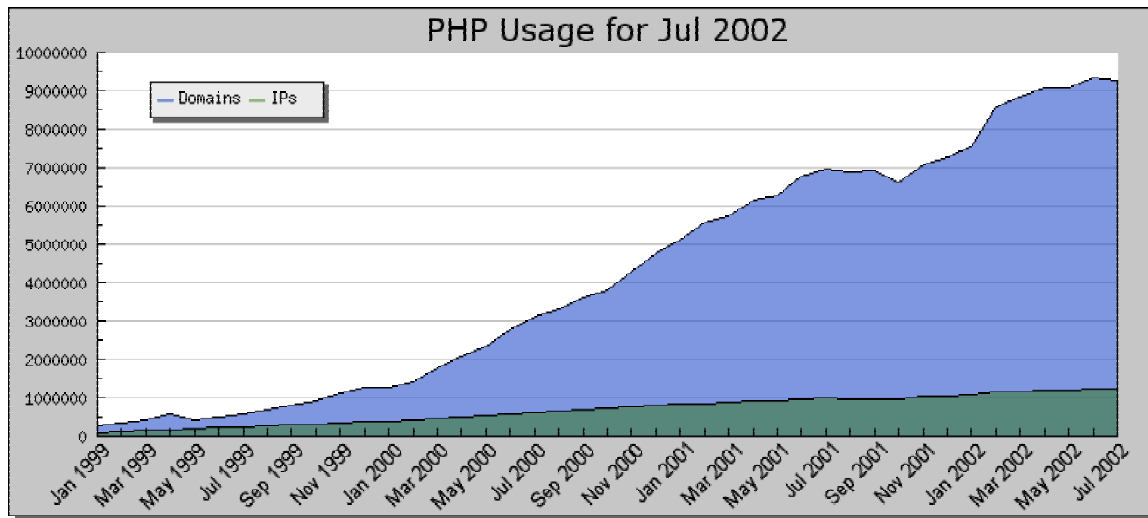
#### 3.2 PHP Scripting Language

Η PHP η οποία είναι συντομογραφία του "PHP: Hypertext Preprocessor" είναι μία ευρέως διαδεδομένη γενικού σκοπού γλώσσα προγραμματισμού η οποία είναι κατάλληλη για ανάπτυξη δικτυακών τόπων αφού μπορεί να ενσωματωθεί σε σελίδες HTML. Το συντακτικό της ενσωματώνει χαρακτηριστικά από τις γλώσσες C, Java και Perl. Ο βασικός σκοπός της γλώσσας είναι να επιτρέπει στους προγραμματιστές να

γράφουν γρήγορα δυναμικές σελίδες για το Διαδίκτυο, αλλά είναι κατάλληλη και για άλλα πράγματα.

Παρακάτω παραθέτονται κάποια στοιχεία σχετικά με το πόσο διαδεδομένη είναι η php σύμφωνα με έγκριτους φορείς

*Χρήση της php μέχρι τον Ιούλιο του 2002*



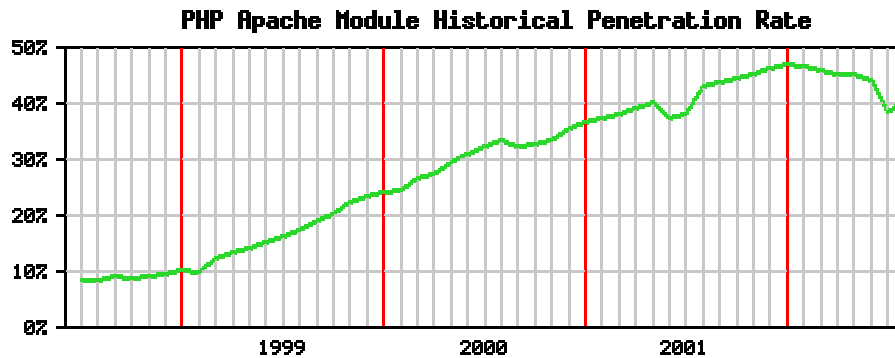
πηγή: [www.php.net](http://www.php.net)

*Ο παρακάτω πίνακας δείχνει τα πιο δημοφιλή modules για τον Apache:*

Module	July 2002 Count	July 2002 %	June 2002 Count	June 2002 %	Growth %
PHP	2305572	40.03	2128669	38.19	4.83
Perl	1773063	30.79	1683286	30.20	1.95
OpenSSL	1400994	24.33	1240173	22.25	9.34
mod_ssl	1355715	23.54	1200038	21.53	9.34
FrontPage	1038819	18.04	993803	17.83	1.17

πηγή: [www.securityspace.com](http://www.securityspace.com)

*Η ανάπτυξη της php ως module του apache*



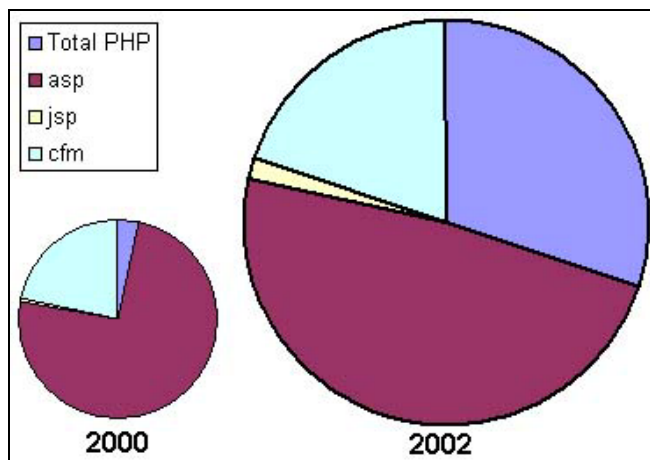
πηγή: [www.securityspace.com](http://www.securityspace.com)

Παρακάτω φαίνεται η δημοτικότητα της php σε σχέση με τις υπόλοιπες διαδεδομένες γλώσσες προγραμματισμού, καθώς και ο ρυθμός ανάπτυξης σε βάθος χρόνου ένα έτος.

Position	Delta 12 months	Programming Language	Ratings	Delta 12 months
1	=	Java	40.4	+63.6%
2	=	C	31.7	+54.6%
3	=	C++	24.6	+54.7%
4	↑↑	Perl	15.3	+128.7%
5	=	Basic	11.8	+42.5%
6	↓↓	HTML	11.0	+10.0%
7	↑↑↑↑	PHP	10.7	+686.8%
8	=	SQL	4.3	+60.4%
9	↓↓	Delphi/Pascal	2.4	-17.8%
10	↑↑↑↑↑↑↑↑	XML	2.3	+274.0%
11	↑↑↑↑↑↑↑↑	C#	2.2	+429.6%
12	↓↓↓	JavaScript	1.9	+11.8%
13	=	Python	1.8	+58.9%
14	↓↓	Fortran	1.7	+34.1%
15	↓↓↓↓	COBOL	1.7	+12.5%
16	↓↓	Lisp	1.3	+54.4%
17	↑↑	Prolog	1.1	+63.0%
18	↓↓	Awk	1.1	+36.1%
19	↓↓	Scheme	1.1	+31.7%
20	↓↓↓↓	Ada	1.1	+27.1%

πηγή: [www.tiobe.com](http://www.tiobe.com)

*Η δημοτικότητα και ο ρυθμός ανάπτυξης των γλωσσών ανάπτυξης διαδικτυακών εφαρμογών*



πηγή: [www.weblogs.com](http://www.weblogs.com)

Language	October 2000	March 2002	Growth	Share
PHP	157,470	7,549,230	4694%	30%
asp	3,166,710	11,958,185	278%	48%
jsp	24,435	413,827	1594%	2%
cfm	936,223	4,950,133	429%	20%

πηγή: [www.weblogs.com](http://www.weblogs.com)

### 3.3 MSSQL Server 2000 (Personal Edition)

Ο Microsoft SQL Server 2000 (MSSQLServer2000) είναι ένα από τα καλύτερα συστήματα διαχείρισης βάσεων δεδομένων (DBMS) τα οποία κυκλοφορούν στην αγορά. Τα συστήματα αυτά είναι σημαντικά στην ανάπτυξη συστημάτων που απαιτούν την ύπαρξη βάσης δεδομένων. Βασικές λειτουργίες που υποστηρίζονται από τα συστήματα αυτά είναι:

- Ορισμός δεδομένων
- Χειρισμός δεδομένων
- Ασφάλεια και ακεραιότητα δεδομένων
- Ανάκαμψη και ταυτοχρονισμός
- Λεξικό δεδομένων
- Απόδοση

## 4. Αρχιτεκτονική συστήματος

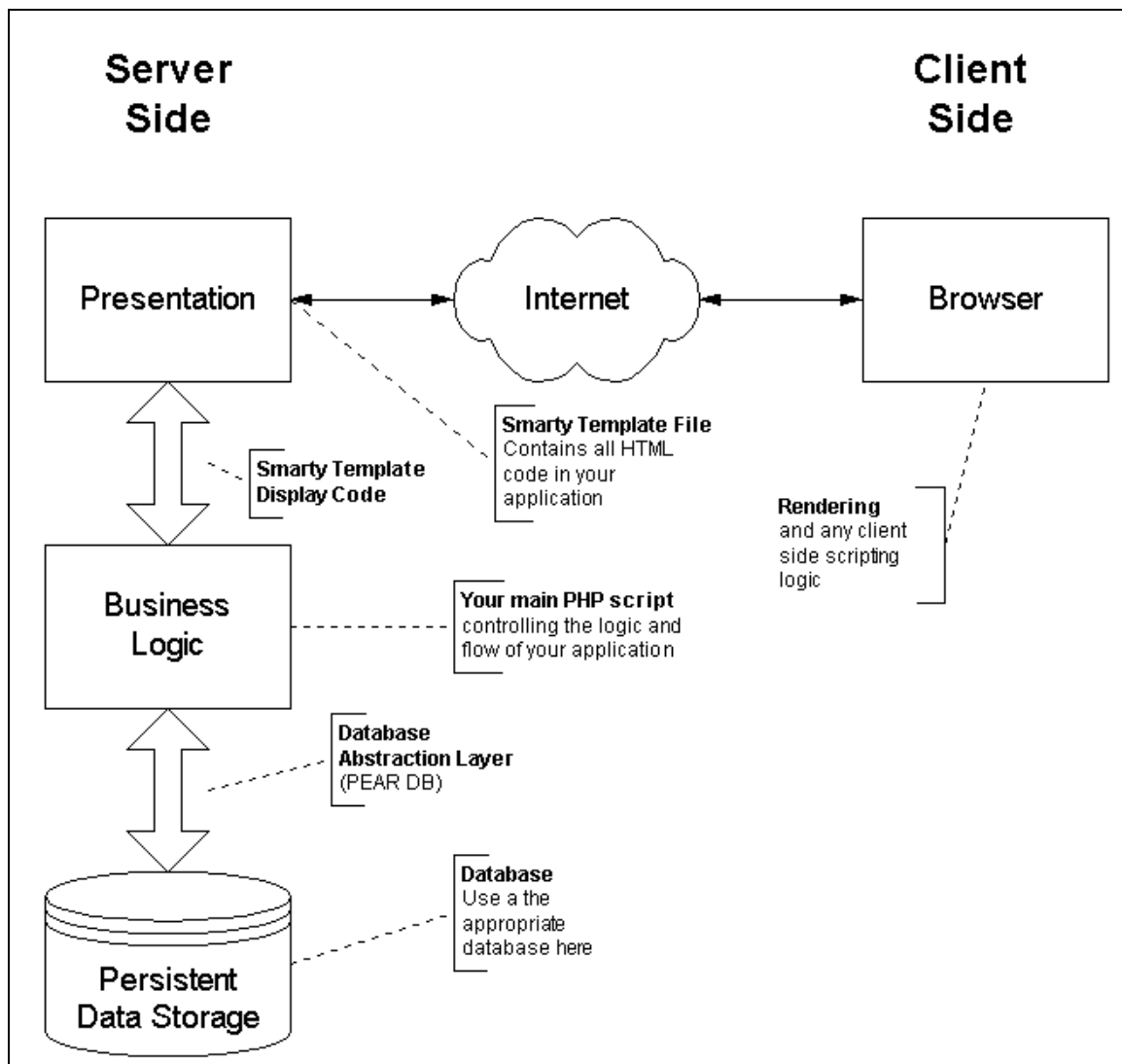
---

### 4.1 Μοντέλο Πελάτη / Εξυπηρετητή (Client/Server Μοντέλο)

Το Client/Server μοντέλο είναι ένα από τα πιο συχνά χρησιμοποιούμενα μοντέλα στο Διαδίκτυο. Σύμφωνα με αυτό πολλοί πελάτες (clients) εξυπηρετούνται από έναν εξυπηρετητή (server). Το μοντέλο αυτό ταιριάζει στις online εφαρμογές όπου πολλοί χρήστες με τη βοήθεια του browser ζητάνε πληροφορίες από ένα συγκεκριμένο δικτυακό τόπο. Στην προκειμένη περίπτωση οι browsers είναι οι clients και ο τόπος είναι ο server.

Παρακάτω φαίνεται η αρχιτεκτονική του μοντέλου Client/Server. Στη μεριά του server υπάρχουν τρία επίπεδα οργάνωσης. Το επίπεδο της βάσης δεδομένων (Database Layer), το επίπεδο των εφαρμογών (Application Layer), το επίπεδο παρουσίασης (Presentation Layer). Στο μοντέλο αυτό όλη η επεξεργασία της πληροφορίας γίνεται από τον server.

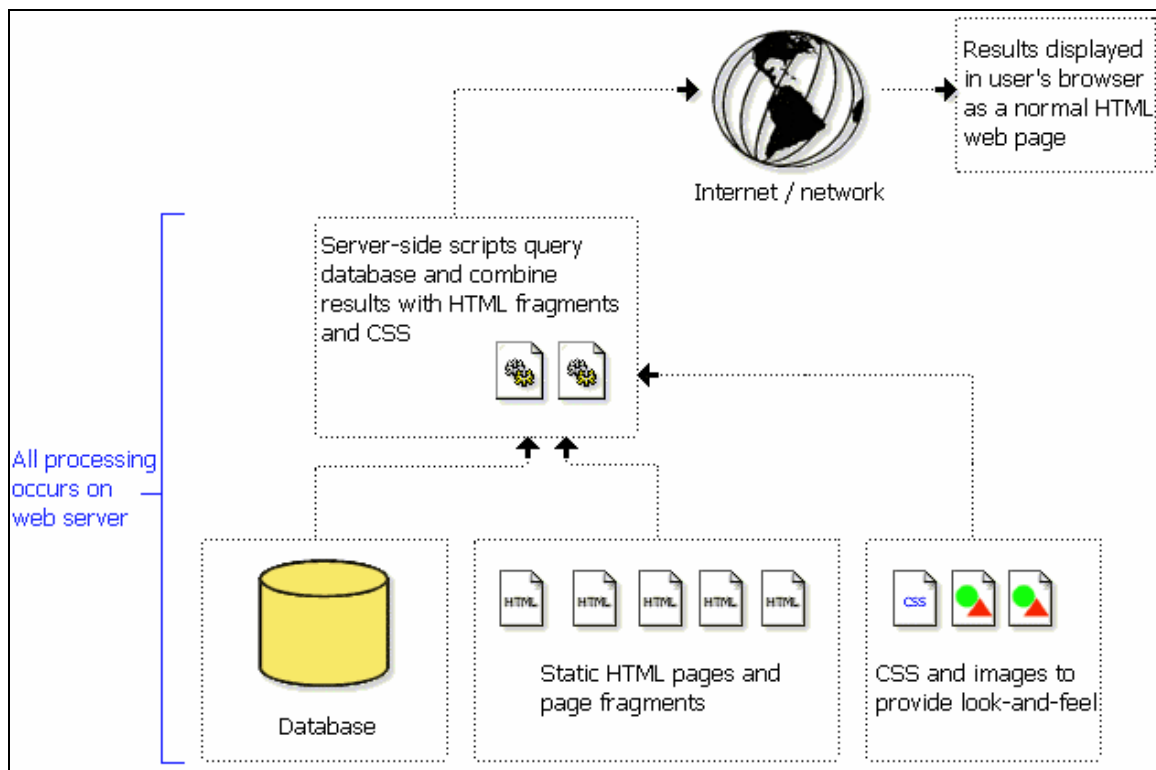
*Αρχιτεκτονική του Client/Server Μοντέλου*



Το παρακάτω διάγραμμα δείχνει πως μπορεί μία βάση δεδομένων να ενσωματωθεί σε ένα client/server μοντέλο για τη δημιουργία ενός δικτυακού τόπου.

Στο μοντέλο αυτό όλα τα αρχεία του δικτυακού τόπου βρίσκονται στον server. Τα αρχεία αυτά μπορεί να είναι HTML σελίδες, κώδικας, cascading style sheets (css) αρχεία, φωτογραφίες, βάση δεδομένων. Όταν στον server έρθει μία αίτηση για μία PHP σελίδα, εκτελείται ο κώδικας και τα αποτελέσματά του εμφανίζονται στον client μέσω του browser του server. Ο PHP κώδικας χρησιμοποιείται για να παράγει δυναμικές σελίδες HTML. Για να παραχθεί μία δυναμική σελίδα πολλές φορές ο PHP κώδικας εκτελεί επερωτήσεις σε βάση δεδομένων και συνδυάζει τα αποτελέσματα της επερώτησης με φωτογραφίες και πληροφορία από τα css (cascading style sheets) αρχεία για να δημιουργήσει την τελική σελίδα HTML την οποία θα δείξει στον client.

*Ενσωμάτωση Βάσης Δεδομένων στο Client/Server Μοντέλο*



πηγή: [www.digitext.com](http://www.digitext.com)

**4.2 Τεχνική Λεξικογραφικής Ανάλυσης (Parsing technique)**

Το MRS (Movie Recommendation System) χρησιμοποιεί πληροφορία η οποία παρέχεται από τον δικτυακό τόπο του IMDB (Internet Movie Data Base, [www.imdb.com](http://www.imdb.com)). Θα ήταν πολύ δύσκολο η πληροφορία αυτή να εισάγεται στη βάση δεδομένων του MRS χειρονακτικά. Για το λόγο αυτό προέκυψε η ανάγκη αυτόματης ανανέωσης της πληροφορίας. Για να πραγματοποιηθεί αυτό χρησιμοποιήθηκε η τεχνική της λεξικογραφικής ανάγνωσης / ανάλυσης (parsing). Σύμφωνα με τη τεχνική αυτή οι σελίδες του IMDB αναλύονται λεξικογραφικά και η χρήσιμη πληροφορία αποθηκεύεται στην τοπική βάση δεδομένων του MRS. Με τον τρόπο αυτό λύθηκε ένα από τα μεγαλύτερα προβλήματα του MRS, αυτό της ανανέωσης και συντήρησης της πληροφορίας. [A3]

**4.3 Περιορισμοί Διαδικτύου**

Μία από τις μεγαλύτερες προκλήσεις τις οποίες έχουν να αντιμετωπίσουν όλα τα online συστήματα είναι να ικανοποιούν τις απαιτήσεις των χρηστών για γρήγορα αποτελέσματα στις αιτήσεις που κάνουν. Αν και πολλές φορές υπεύθυνο για τέτοια προβλήματα είναι το ίδιο το εύρος ζώνης (bandwidth) του δικτύου, ο υπεύθυνος ανάπτυξης συστημάτων θα πρέπει να προσπαθεί να κάνει όσο το δυνατό πιο αποδοτικό το σύστημά του. Τα χαρακτηριστικά που επηρεάζουν την απόδοση ενός online συστήματος είναι τα εξής:

- Μέγεθος HTML σελίδας
- Μέγεθος εικόνων που χρησιμοποιούνται στο layout
- Χρόνος απόκτησης των δεδομένων
- Χρόνος επεξεργασίας των δεδομένων

Στο MRS γίνεται χρήση πολύπλοκων αλγορίθμων για να μπορέσει το σύστημα να προτείνει ταινίες στον χρήστη. Η απόδοση των αλγορίθμων αυτών εξαρτάται άμεσα από το πλήθος των χρηστών και από το πλήθος των ταινιών που υπάρχουν στη βάση. Απαιτηση του συστήματος είναι οι αλγόριθμοι αυτοί να είναι ικανοποιητικά γρήγοροι, και το πλήθος των απαντήσεων το οποίο δίνουν να είναι περιορισμένο έτσι ώστε ο χρήστης να λαμβάνει γρήγορα τις συστάσεις του συστήματος.

Όσον αφορά τα υπόλοιπα χαρακτηριστικά έγινε μεγάλη προσπάθεια έτσι ώστε το MRS να συνδυάζει με τον καλύτερο τρόπο γρήγορο κατέβασμα των σελίδων και καλή σχεδίαση. Σε μεγάλο βαθμό το MRS σύστημα είναι συναφές με τις αρχές που ορίζει ο τομέας της επικοινωνίας ανθρώπου μηχανής για την ανάπτυξη διεπαφών.



## **5. Ανάλυση τεχνικών χαρακτηριστικών Συστήματος**

---

### **5.1 Εισαγωγικά**

Βασικής σημασίας είναι η προσπάθεια που έγινε το MRS (Movie Recommendation System) να δομηθεί έτσι ώστε να είναι:

- εύκολα επεκτάσιμο
- εύκολα συντηρήσιμο
- ασφαλές
- αποδοτικό
- εύκολα μεταφέρσιμο σε άλλη βάση δεδομένων (αντί MSSQL)
- εύκολα μεταφέρσιμο σε άλλη περιοχή ενδιαφέροντος (εκτός ταινίες κινηματογράφου)

Παρακάτω αναλύονται διάφορα τεχνικά χαρακτηριστικά και αποφάσεις που πάρθηκαν κατά την ανάπτυξη του, ούτως ώστε το σύστημα να εξυπηρετεί τις παραπάνω απαιτήσεις.

### **5.2 Sessions**

Τα sessions είναι μία γνωστή τεχνική η οποία χρησιμοποιείται κατά την ανάπτυξη εφαρμογών στο Διαδίκτυο. Επιτρέπει την εύκολη αποθήκευση πληροφορίας στον browser του χρήστη με αποτέλεσμα να είναι εύκολη η εξατομίκευση ορισμένων διεργασιών. Για την αύξηση της απόδοσης και της ασφάλειας καθώς και για την ευκολία ανάπτυξης κρίθηκε απαραίτητο να χρησιμοποιηθεί η τεχνική αυτή στο MRS σύστημα.

### **5.3 Database Class**

Το MRS είναι ένα ολοκληρωμένο πληροφοριακό σύστημα. Για λόγους ασφαλείας αλλά και επεκτασιμότητας κρίθηκε αναγκαίο όλες οι συναλλαγές με τη βάση να γίνονται με τη βοήθεια μίας κλάσης. Με τον τρόπο αυτό κρίσιμη πληροφορία, όπως κωδικοί πρόσβασης στην βάση δεδομένων δεν βρίσκονται εκτεθειμένοι. Το σημαντικότερο όμως είναι πως με τη χρήση της κλάσης αυτής το σύστημα παραμένει ανεξάρτητο από τη βάση δεδομένων την οποία χρησιμοποιεί, δηλαδή μπορεί εύκολα το σύστημα να μεταφερθεί σε άλλη βάση δεδομένων. [A1]

## 5.4 Graph Class

Στο σύστημα ουσιαστικό ρόλο παίζουν οι γραφικές αναπαραστάσεις οι οποίες δημιουργούνται δυναμικά και αφορούν την κατανομή των ψήφων [A15]. Τα γραφήματα αυτά δημιουργούνται με τη βοήθεια μίας κλάσης γραφημάτων.

## 5.5 Function Library

Όλες οι συναρτήσεις οι οποίες γράφηκαν κατά την ανάπτυξη του συστήματος βρίσκονται συγκεντρωμένες σε ένα αρχείο. Η δημιουργία βιβλιοθήκης συναρτήσεων έλυσε πολλά προβλήματα κατά τη διάρκεια της ανάπτυξης και ήταν ένα από τα χαρακτηριστικά καλής δόμησης του τόπου.

## 5.6 Template File

Ένα από τα βασικότερα χαρακτηριστικά σχεδίασης του τόπου αποτελεί η ύπαρξη αρχείου template. Με τη βοήθεια του αρχείου αυτού κάθε σελίδα δεν χρειάζεται να φτιαχτεί από την αρχή. Το περίγραμμα του τόπου είναι ένα και χρησιμοποιείται σε όλες τις σελίδες. Με τον τρόπο αυτό οποιαδήποτε αλλαγή χρειαστεί να γίνει στον δικτυακό τόπο και αφορά όλες τις σελίδες αρκεί να γίνει μόνο σε αυτό το αρχείο και η αλλαγή θα εφαρμοστεί σε όλες τις σελίδες. Με τον τρόπο αυτό θα μπορούσε εύκολα αναπτυχθεί μία υπηρεσία στην οποία ο χρήστης επιτρέπει τον τρόπο εμφάνισης της σελίδας. (personalization)

## 5.7 Cascade Style Sheet File

Τα css αρχεία (cascade style sheet) είναι μία τεχνική με την οποία ο προγραμματιστής μπορεί να ελέγχει τα γνωρίσματα διαφόρων αντικειμένων της σελίδας μέσα από ένα αρχείο το οποίο ενσωματώνεται σε όλες τις σελίδες. Έτσι η εμφάνιση των σελίδων μπορεί εύκολα να αλλάξει, αλλάζοντας μόνο αυτό το αρχείο.

## 5.8 Configuration File

Κατά την ανάπτυξη του συστήματος προέκυψε η ανάγκη της δημιουργίας ενός αρχείου διαμόρφωσης (configuration file) μέσα από το οποίο ο διαχειριστής έχει τη δυνατότητα να αλλάζει κάποιες μεταβλητές του συστήματος (global variables) επηρεάζοντας την εμφάνιση του τόπου (layout), αλλά και τις παραμέτρους των αλγορίθμων(π.χ. τα όρια ποιότητας). [A2]

### **5.9 Διαφορετικοί Ρόλοι Χρηστών (Εγγεγραμμένος, Μη-Εγγεγραμμένος, Διαχειριστής)**

Το σύστημα MRS υποστηρίζει τριών ειδών χρήστες:

- Μη-εγγεγραμμένους
- Εγγεγραμμένους
- Διαχειριστές

Ανάλογα με το ρόλο του καθενός το σύστημα του δίνει διαφορετικές δυνατότητες. Ο μη-εγγεγραμμένος χρήστης δεν είναι αναγνωρίσιμος από το σύστημα με αποτέλεσμα να μη μπορεί να γίνει δέκτης εξατομικευμένων υπηρεσιών. Καθώς δεν είναι αναγνωρίσιμος το σύστημα υποτιμάει το ρόλο του στον καθορισμό βαθμολογιών δίνοντας μεγαλύτερο βάρος στις ψήφους εγγεγραμμένων χρηστών και αυτό για λόγους αξιοπιστίας.

Επίσης υποστηρίζεται και ο ρόλος του διαχειριστή. Ο διαχειριστής έχει στη διάθεσή του ορισμένα εργαλεία παρακολούθησης του δικτυακού τόπου, καθώς και επίβλεψης για το αν όλα λειτουργούν κανονικά.

## 6. Σχεδίαση Βάσης

---

### 6.1 Περιγραφή

Ο μεγάλος όγκος πληροφορίας την οποία χρησιμοποιεί το σύστημα MRS βρίσκεται αποθηκευμένος σε μία βάση δεδομένων. Η βάση είναι σε κανονική μορφή Boyce-Codd. Μεγάλη προσοχή δόθηκε στο να εξυπηρετεί η βάση με τον καλύτερο τρόπο τις λειτουργίες που αναπτύχθηκαν, αλλά και να είναι απλή και εύκολα επεκτάσιμη.

Παρακάτω φαίνεται το διάγραμμα οντοτήτων-σχέσεων της βάσης με τις πληθικότητες, το σχεσιακό μοντέλο που προκύπτει καθώς και στιγμιότυπο της βάσης όπως αυτό τελικά είναι υλοποιημένο.

Τα στοιχεία που κρατούνται στη βάση είναι οι χρήστες, οι ταινίες, το προφίλ των χρηστών, οι ψήφοι που έχει δώσει κάποιος χρήστης, οι ψήφοι που έχει πάρει μία ταινία. Οι οντότητες που προκύπτουν είναι:

- Οι χρήστες
- Οι ταινίες
- Τα προφίλ των χρηστών

Οι μεταξύ τους σχέσεις είναι:

- Ένας χρήστης έχει προφίλ
- Ένας χρήστης ψηφίζει μία ταινία

Επίσης από τα πλειότιμα γνωρίσματα της οντότητας «movie» προκύπτουν οι σχέσεις:

- Μία ταινία ανήκει σε κατηγορίες
- Μία ταινία παίρνει ψήφους από μη εγγεγραμμένους χρήστες

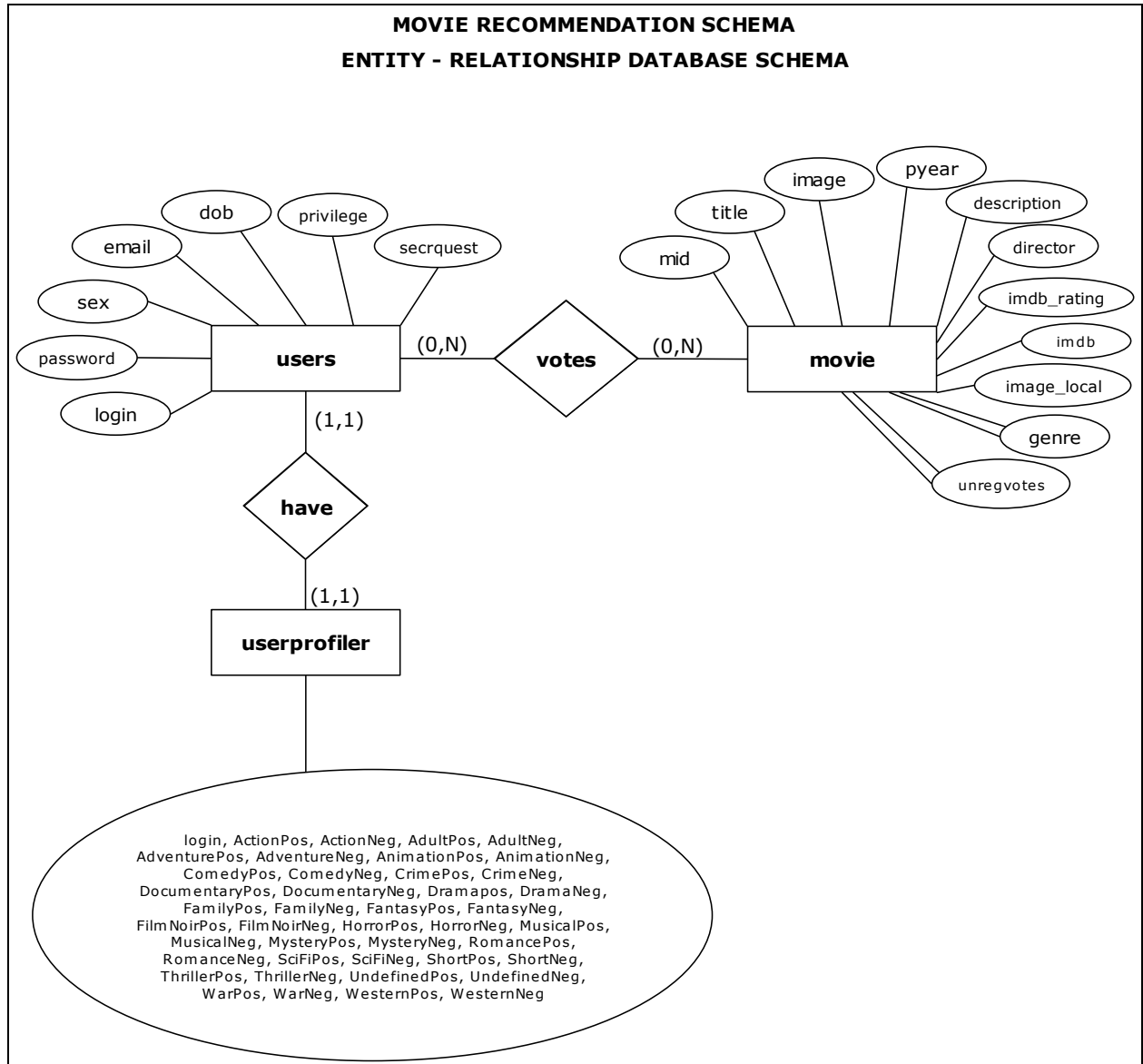
Παρακάτω αναλύεται το σχεσιακό μοντέλο το οποίο προκύπτει από το διάγραμμα οντοτήτων-σχέσεων της βάσης.

### 6.2 Σχήμα Οντοτήτων-Σχέσεων Βάσης Δεδομένων

Το σχήμα Οντοτήτων-Σχέσεων (ER) αποτελεί έναν εύχρηστο τρόπο περιγραφής μίας βάσης δεδομένων. Το ER διευκολύνει τη διαδικασία δημιουργίας του σχεσιακού μοντέλου της βάσης δεδομένων και επιτρέπει τη γρήγορη κατανόηση από τον

προγραμματιστή της οργάνωσης του τόπου. Παρακάτω φαίνεται το σχεσιακό μοντέλο το οποίο προκύπτει από το διάγραμμα οντοτήτων-σχέσεων της βάσης του MRS.

Διάγραμμα Οντοτήτων-Σχέσεων του MRS



### 6.3 Σχεσιακό Μοντέλο της βάσης δεδομένων

Το σχεσιακό μοντέλο μιας βάσης δεδομένων προκύπτει από το διάγραμμα οντοτήτων-σχέσεων της βάσης και αποτελεί το ενδιάμεσο βήμα μεταξύ σχεδίασης και υλοποίησης. Στο βήμα αυτό καθορίζονται οι πίνακες που θα έχει η βάση καθώς και όλα τα γνωρίσματα του κάθε πίνακα, όπως αυτά προκύπτουν από τις πληθικότητες των

σχέσεων. Το σχεσιακό μοντέλο αναπαριστά τη πληροφορία με τον ίδιο ακριβώς τρόπο με τον οποίο αυτή αποθηκεύεται στη βάση δεδομένων και για το λόγο αυτό αποτελεί σημαντικό εργαλείο για τον προγραμματιστή. Παρακάτω φαίνεται το σχεσιακό μοντέλο για το MRS.

*Σχεσιακό Μοντέλο της Βάσης Δεδομένων του MRS*

*users*

login	password	sex	email	dob	privilege	secrequest
-------	----------	-----	-------	-----	-----------	------------

*movie*

mid	title	image	pyear	description	director	imdb_rating	imdb	image_local
-----	-------	-------	-------	-------------	----------	-------------	------	-------------

*userprofiler*

login	ActionPos	ActionNeg	AdultPos	AdultNeg	AdventurePos	AdventureNeg
-------	-----------	-----------	----------	----------	--------------	--------------

AnimationPos	AnimationNeg	ComedyPos	ComedyNeg	CrimePos	CrimeNeg
--------------	--------------	-----------	-----------	----------	----------

DocumentaryPos	DocumentaryNeg	DramaPos	DramaNeg	FamilyPos	FamilyNeg
----------------	----------------	----------	----------	-----------	-----------

FantasyPos	FantasyNeg	FilmNoirPos	FilmNoirNeg	HorrorPos	HorrorNeg
------------	------------	-------------	-------------	-----------	-----------

MusicalPos	MusicalNeg	MysteryPos	MysteryNeg	RomancePos	RomanceNeg
------------	------------	------------	------------	------------	------------

SciFiPos	SciFiNeg	ShortPos	ShortNeg	ThrillerPos	ThrillerNeg
----------	----------	----------	----------	-------------	-------------

UndefinedPos	UndefinedNeg	WarPos	WarNeg	WesternPos	WesternNeg
--------------	--------------	--------	--------	------------	------------

*unregvotes*

mid	vote
-----	------

*votes*

login	mid	vote
-------	-----	------

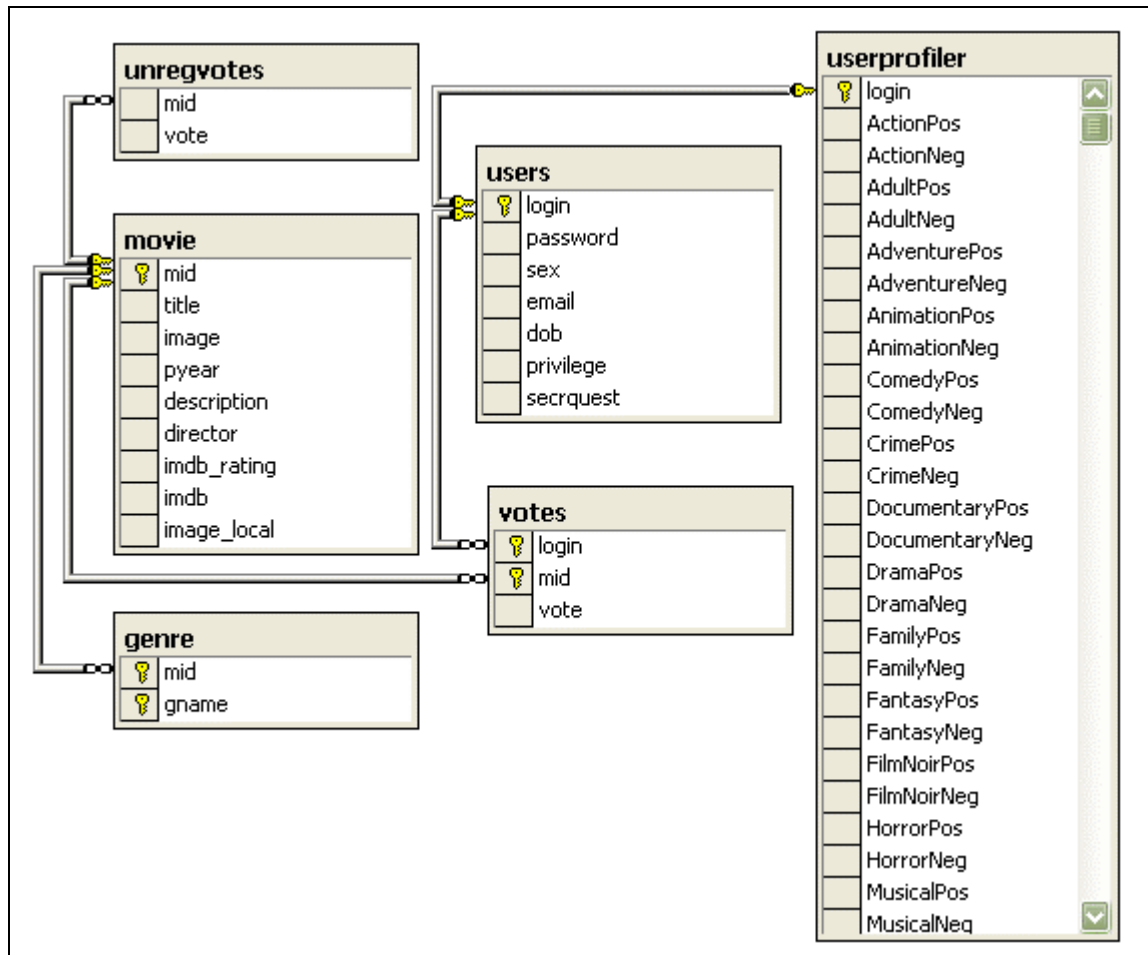
*genre*

mid	gname
-----	-------

## 6.4 Στιγμιότυπο της υλοποιημένης βάσης δεδομένων

Παρακάτω φαίνεται το σχήμα της βάσης δεδομένων του συστήματος, το οποίο έχει προκύψει από το σχεσιακό μοντέλο της βάσης δεδομένων που μελετήσαμε προηγουμένως.

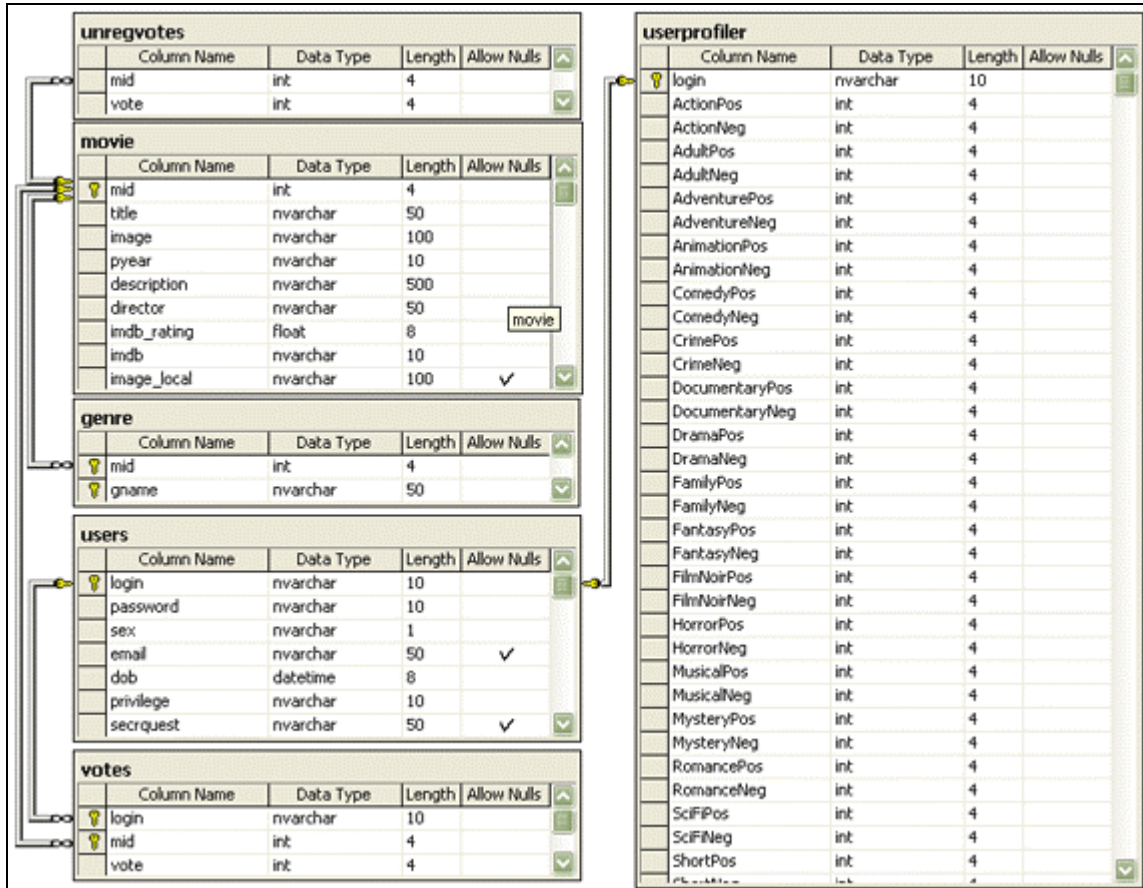
*Το σχήμα της βάσης δεδομένων του MRS συστήματος*



Πηγή: MSSQLServer2000

Παρακάτω φαίνεται το σχήμα της βάσης δεδομένων του συστήματος MRS αλλά αυτή τη φορά φαίνονται και οι τύποι του κάθε γνωρίσματος, καθώς και αν επιτρέπεται να παίρνει το γνώρισμα τιμές NULL.

*Το σχήμα της βάσης δεδομένων του MRS συστήματος και οι τύποι των γνωρισμάτων*



Πηγή: MSSQLServer2000



## 7. Λειτουργίες Συστήματος

### 7.1 Login

Το σύστημα υποστηρίζει τη δημιουργία λογαριασμών από τον χρήστη. Η διαδικασία εισόδου είναι η διαδικασία που επιτρέπει στον χρήστη να γίνει αναγνωρισμός από το σύστημα (Σχεδιάγραμμα 1). Με τον τρόπο αυτό ο χρήστης μπορεί να έχει πρόσβαση σε προσωπικά δεδομένα. Εάν ο χρήστης δεν είναι ήδη εγγεγραμμένος μέσω της διαδικασίας αυτής μπορεί να εγγραφεί στο σύστημα. Επίσης εάν είναι ήδη εγγεγραμμένος αλλά έχει ξεχάσει το μυστικό του κωδικό μπορεί εύκολα να ανακτήσει το παλιό του κωδικό.



(Σχεδιάγραμμα 1)

### 7.2 My Profile

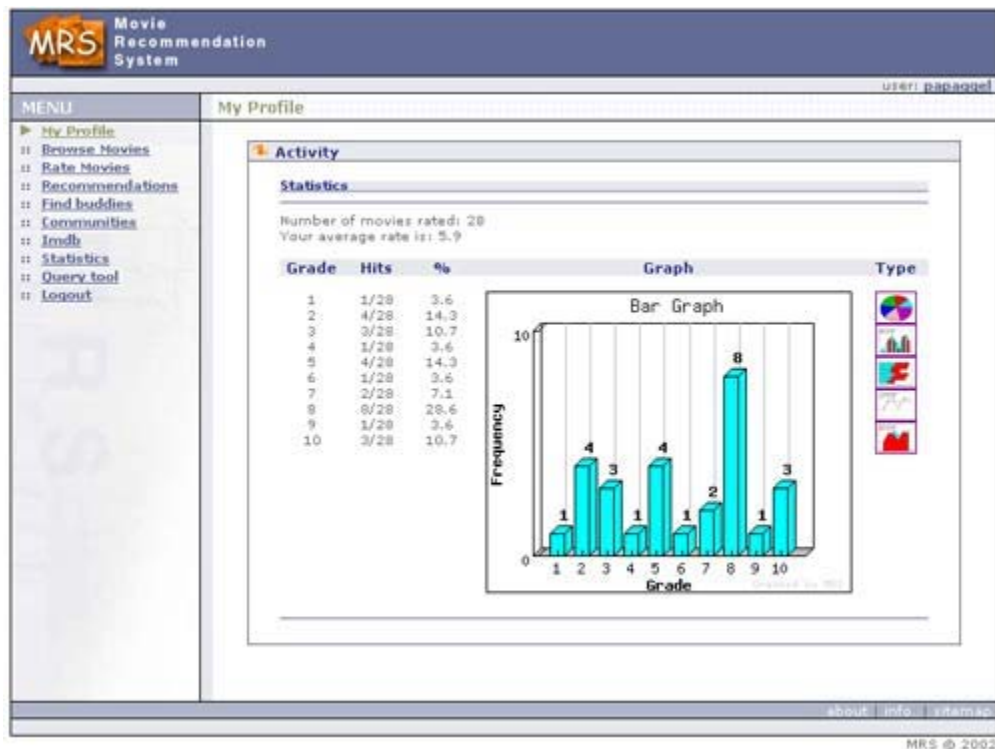
Μέσα από τη διαδικασία αυτή ο χρήστης έχει πρόσβαση στην προσωπική του σελίδα. Από τη σελίδα αυτή έχει τη δυνατότητα:

- να αλλάξει τις αρχικές πληροφορίες που είχε δώσει στο σύστημα για το προφίλ του
- να δει ένα ιστορικό με τις ταινίες που έχει ψηφίσει και εάν επιθυμεί να αλλάξει κάποια παλιότερη βαθμολογία του (Σχεδιάγραμμα 2)



(Σχεδιάγραμμα 2)

- να δει γραφικά την δραστηριότητά του στο σύστημα, δηλαδή να δει την κατανομή των ψήφων του σε απόλυτες και σχετικές τιμές, τον συνολικό αριθμό των ψήφων του, καθώς και το μέσο όρο της βαθμολογίας του (Σχεδιάγραμμα 3)

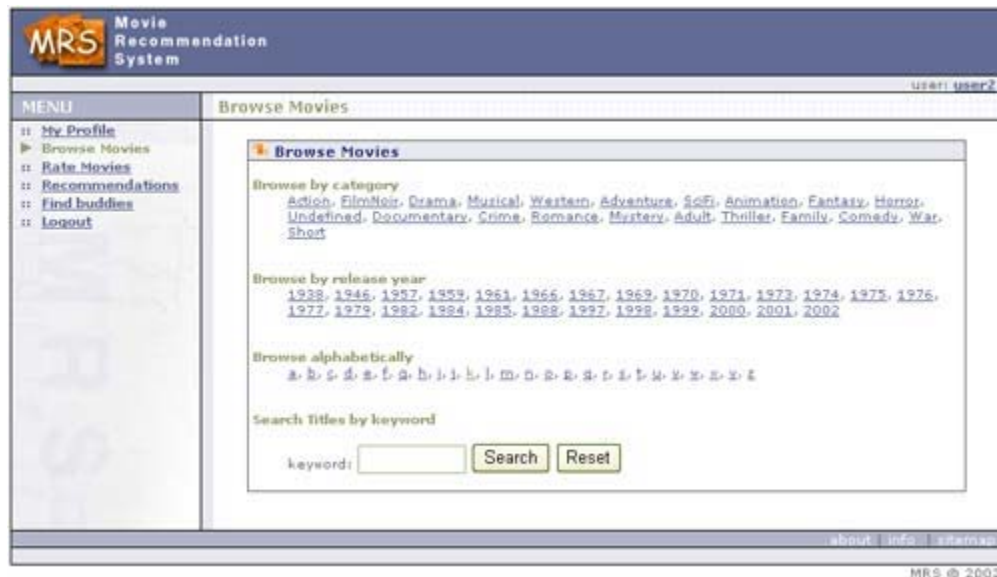


(Σχεδιάγραμμα 3)

### 7.3 Browse Movies

Η διαδικασία περιήγησης επιτρέπει στον χρήστη να περιηγηθεί ανάμεσα στις ταινίες ώστε να ενημερωθεί για τις ταινίες που τον ενδιαφέρουν. Η περιήγηση είναι δυνατή με διάφορους τρόπους ούτως ώστε να είναι εύκολη και γρήγορη η εύρεση κάποιας συγκεκριμένης ταινίας από έναν χρήστη, αλλά συγχρόνως να είναι λειτουργική για τον χρήστη που θέλει να δει ένα σύνολο από ταινίες (Σχεδιάγραμμα 4). Η περιήγηση μπορεί να γίνει με τους εξής τρόπους:

- Με βάση την κατηγορία
- Με βάση το έτος έκδοσης
- Με βάση το όνομα – Αλφαβητικά
- Με βάση το όνομα με βοήθεια μηχανής αναζήτησης

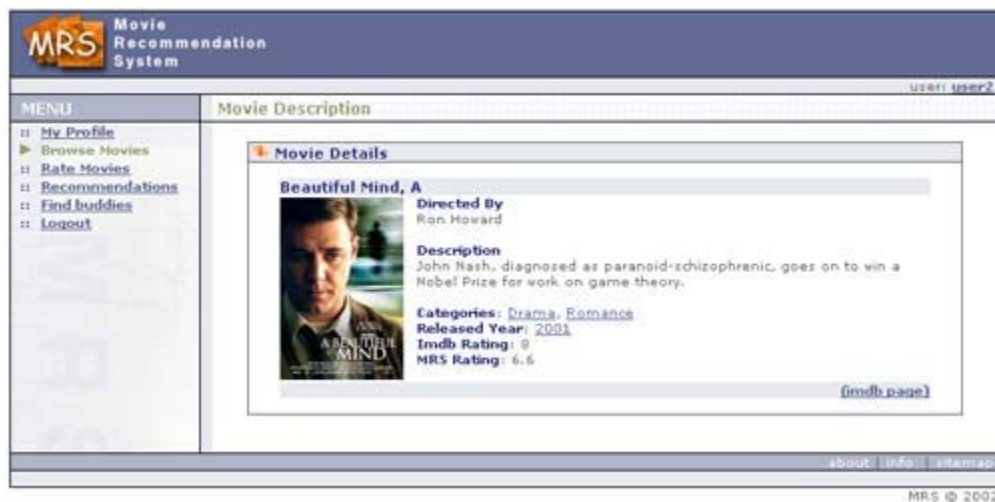


(Σχεδιάγραμμα 4)

Καθώς ο χρήστης επιλέγει κάποιον από τους παραπάνω τρόπους, μία σειρά από ταινίες εμφανίζονται οι οποίες ικανοποιούν τα κριτήρια της αρχικής επιλογής του. Έπειτα ο χρήστης μπορεί να επιλέξει για ποια συγκεκριμένη ταινία θέλει να δει λεπτομέρειες. Τελικά η περιήγηση καταλήγει σε σελίδα αφιερωμένη σε μία συγκεκριμένη ταινία (Σχεδιάγραμμα 5). Στη σελίδα αυτή δίνεται πληροφορία σχετικά με:

- Το όνομα της ταινίας
- Το poster της ταινίας
- Τον σκηνοθέτη της ταινίας

- Περιγραφή της ταινίας
- Κατηγορίες στις οποίες ανήκει αυτή η ταινία
- Έτος έκδοσης της ταινίας
- Βαθμολογία που δίνει το IMDB για αυτήν την ταινία
- Βαθμολογία που δίνει το MRS για αυτήν την ταινία
- Αναφορά-Σύνδεσμος στην επίσημη σελίδα του IMDB για αυτή την ταινία



(Σχεδιάγραμμα 5)

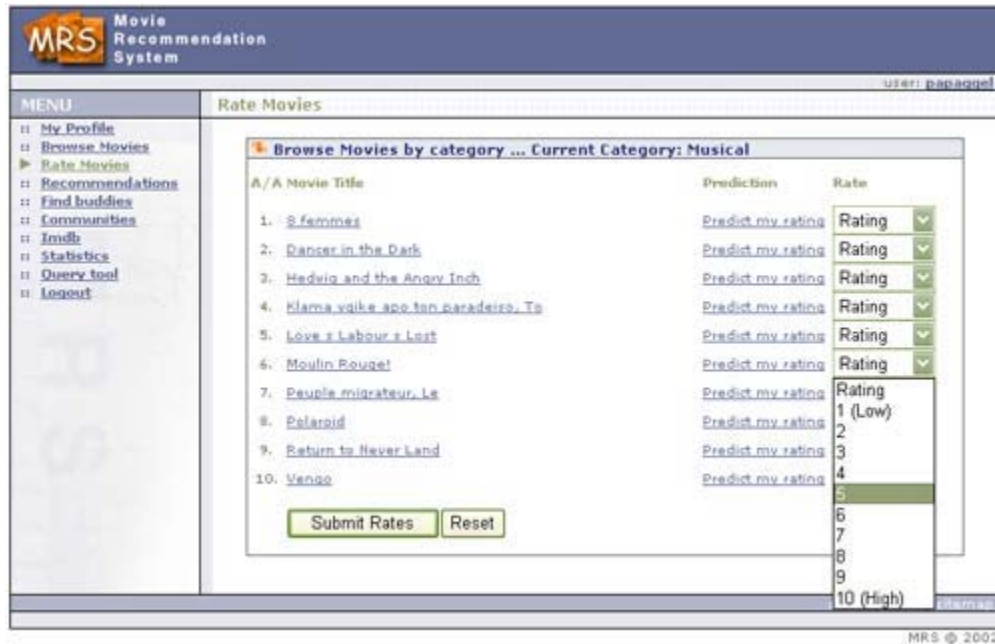
## 7.4 Rate Movies

Η λειτουργία της βαθμολόγησης των ταινιών είναι μία από τις σημαντικότερες διαδικασίες του συστήματος. Η διαδικασία αυτή έχει την ίδια ακριβώς δομή με τη διαδικασία περιήγησης για να είναι πιο λειτουργική προς τον χρήστη. Έτσι λοιπόν καθώς κάποιος επιλέγει να αρχίσει να βαθμολογεί ταινίες μπορεί να περιηγηθεί στις ταινίες με τους εξής τρόπους:

- Με βάση την κατηγορία
- Με βάση το έτος έκδοσης
- Με βάση το όνομα – Αλφαβητικά
- Με βάση το όνομα με βοήθεια μηχανής αναζήτησης

Αυτό που διαφοροποιεί την λειτουργία αυτή από τη λειτουργία περιήγησης είναι ότι ο χρήστης τώρα σε κάθε βήμα έχει τη δυνατότητα βαθμολόγησης των ταινιών που βλέπει.

Η κλίμακα της βαθμολογίας ξεκινάει από το 1 το οποίο είναι η μικρότερη βαθμολογία την οποία μπορεί να βάλει σε μία ταινία ο χρήστης και φτάνει μέχρι το 10 το οποίο είναι η μεγαλύτερη βαθμολογία που μπορεί να πάρει μία ταινία (Σχεδιάγραμμα 6).



(Σχεδιάγραμμα 6)

Για να διατηρηθεί η αξιοπιστία του συστήματος κάθε χρήστης μπορεί μόνο μία φορά να ψηφίσει μία ταινία. Εάν ψηφίσει μία ταινία την οποία είχε ψηφίσει στο παρελθόν η ψήφος του ανανεώνεται, ισχύει δηλαδή η καινούρια ψήφος.

Δίνεται στον χρήστη η δυνατότητα να βαθμολογήσει μία-μία τις ταινίες ή ακόμη και πολλές μαζί. Έτσι απλοποιείται κατά πολύ η διαδικασία της βαθμολόγησης η οποία είναι απαραίτητη για να εμπλουτιστεί το προφίλ του χρήστη.

Μία πολύ σημαντική διαδικασία η οποία εμφανίζεται κατά τη διάρκεια της βαθμολόγησης είναι η διαδικασία της πρόβλεψης ψήφου.

## 7.5 Rating Prediction

Η διαδικασία πρόβλεψης ψήφου είναι ενσωματωμένη στη διαδικασία βαθμολόγησης και είναι μία από τις σημαντικότερες του συστήματος. Σύμφωνα με αυτήν το σύστημα είναι σε θέση να προβλέψει τη ψήφο του χρήστη για μία ταινία εάν αυτός δεν την έχει ήδη ψηφίσει και εάν ένας ικανοποιητικός αριθμός άλλων χρηστών την έχουν ψηφίσει

(Σχεδιάγραμμα 7). Η διαδικασία πρόβλεψης ψήφου χρησιμοποιεί τον αλγόριθμο πρόβλεψης ψήφου ο οποίος αναλύεται παρακάτω.



(Σχεδιάγραμμα 7)

## 7.6 Recommendations

Η διαδικασία των συστάσεων είναι η πιο σημαντική διαδικασία του συστήματος. Ο χρήστης έχει τη δυνατότητα να πάρει συστάσεις από το σύστημα για το ποιες ταινίες θα του αρέσουν. Υπάρχουν τρεις μέθοδοι δημιουργίας συστάσεων (Σχεδιαγράμματα 8,9).

- Η πρώτη μπορεί να χρησιμοποιηθεί και από μη εγγεγραμμένους χρήστες και στηρίζεται στην λογική να προτείνει ταινίες οι οποίες έχουν το μεγαλύτερο μέσο όρο, ο οποίος προκύπτει από επεξεργασία των ψήφων που έχουν πάρει οι ταινίες από εγγεγραμμένους και μη εγγεγραμμένους χρήστες.
- Η δεύτερη προϋποθέτει πως ο χρήστης είναι εγγεγραμμένος. Η μέθοδος αυτή γνωστή με το όνομα Collaborative Filtering (CF) στηρίζεται στη λογική να προτείνει στον χρήστη ταινίες τις οποίες χρήστες με παρόμοιο προφίλ έχουν βαθμολογήσει καλά. Η μέθοδος αυτή για να συγκρίνει δύο προφίλ χρησιμοποιεί ως πληροφορία προς σύγκριση τις βαθμολογίες τις οποίες έχουν βάλει οι διάφοροι χρήστες σε κοινές ταινίες.
- Η τρίτη μέθοδος στηρίζεται στη λογική να προτείνει στον χρήστη ταινίες οι οποίες ανήκουν σε κατηγορίες τις οποίες ο χρήστης προτιμάει, δηλαδή κατηγορίες στις οποίες ανήκαν ταινίες που στο παρελθόν ο χρήστης βαθμολόγησε θετικά.

Η δεύτερη και η τρίτη μέθοδος προϋποθέτουν ο χρήστης να είναι εγγεγραμμένος αφού στηρίζουν τα αποτελέσματά τους στα προφίλ τους.



(Σχεδιάγραμμα 8)



(Σχεδιάγραμμα 9)

## 7.7 Find Buddies

Η διαδικασία εύρεσης φίλων επιτρέπει σε ένα χρήστη να γνωρίζει με ποιους άλλους χρήστες τους συστήματος ταιριάζει, δηλαδή με ποιων χρηστών το προφίλ ταιριάζει το δικό του προφίλ. Επειδή για τον υπολογισμό της συσχέτισης δύο προφίλ υπάρχουν

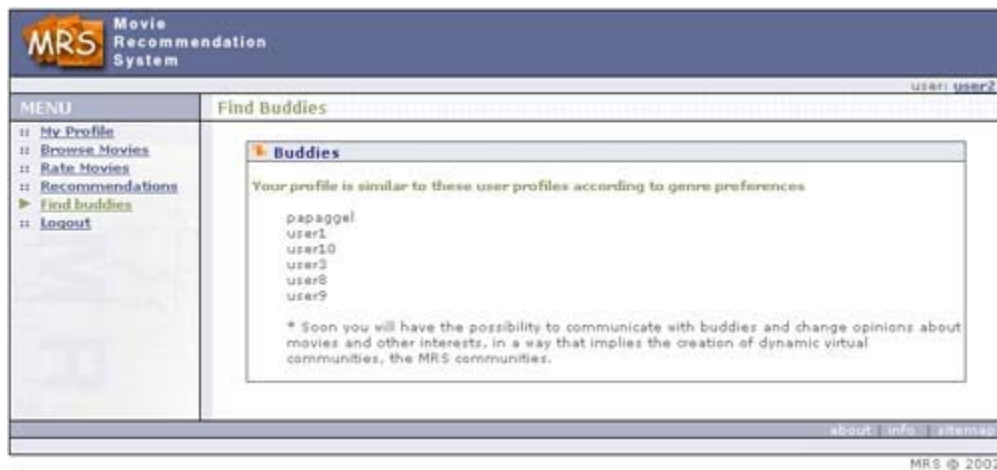
δύο τρόποι οι οποίοι δίνουν διαφορετικά αποτελέσματα ο χρήστης μέσω αυτής της διαδικασίας μπορεί να επιλέξει να βρει φίλους με βάση τις συσχετίσεις των προφίλ που χρησιμοποιούν ως πληροφορία (Σχεδιάγραμμα 10):

- τις βαθμολογίες χρηστών σε κοινές ταινίες
- τις προτιμήσεις σε συγκεκριμένες κατηγορίες ταινιών



(Σχεδιάγραμμα 10)

Το σύστημα αποκρίνεται στον χρήστη με μία σειρά χρηστών οι οποίοι είναι σχετικοί με αυτόν και του δίνει τη δυνατότητα να επικοινωνήσει μαζί τους δημιουργώντας έτσι φιλίες (Σχεδιάγραμμα 11).



(Σχεδιάγραμμα 11)

Με τον τρόπο αυτό μέσα από την υπηρεσία αυτή γίνεται αντιληπτό ότι γεννιέται η έννοια των κοινοτήτων (communities), χρηστών δηλαδή οι οποίοι έχουν κοινές προτιμήσεις όσον αφορά τις ταινίες κινηματογράφου και οι οποίοι έχουν πιθανότητα κάποιο λόγο να έρθουν σε επικοινωνία μεταξύ τους.

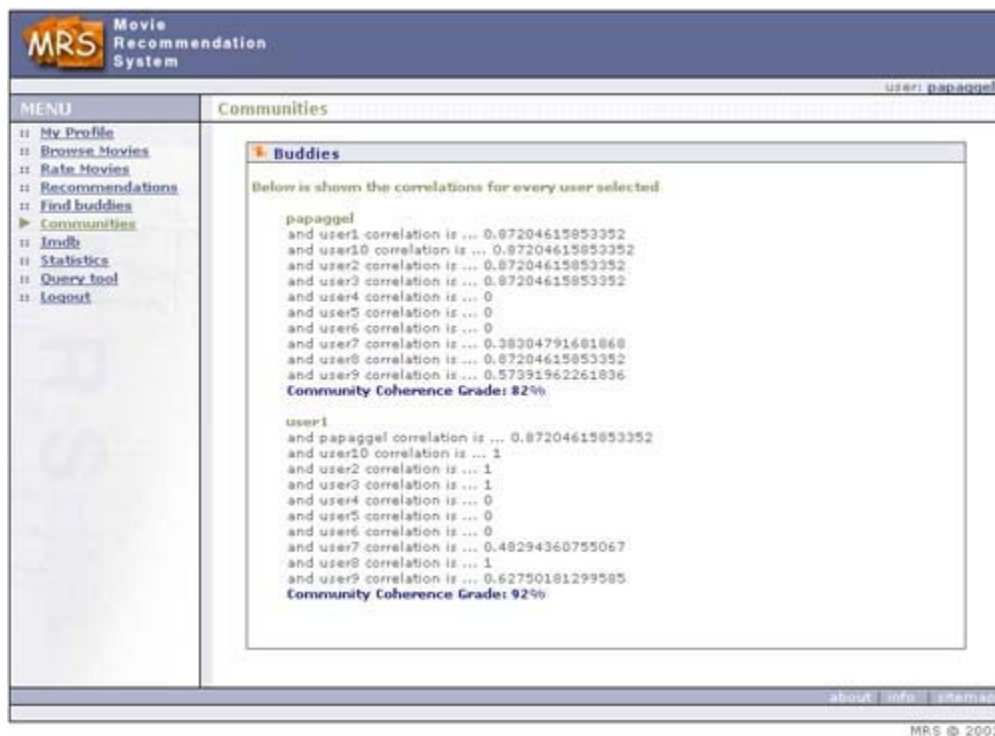


## 7.8 Communities

Η διαδικασία αυτή είναι προσιτή μόνο στον διαχειριστή (administrator) του συστήματος. Αποτελεί ουσιαστικά ένα εργαλείο με το οποίο ο χρήστης μπορεί να εντοπίζει το βαθμό συσχέτισης χρηστών και να εφαρμόζει διάφορα κριτήρια με σκοπό να δει ποιοι χρήστες τα ικανοποιούν ή όχι.

- Μπορεί να εφαρμόσει διαφόρων ειδών φιλτραρίσματα ώστε να εντοπίσει τις προτιμήσεις κάποιου χρήστη σε συγκεκριμένη κατηγορία ταινιών
- Μπορεί να εξετάσει το βαθμό συσχέτισης κάποιου χρήστη με τους άλλους χρήστες με βάση τις βαθμολογίες που έχουν δώσει.
- Μπορεί να εξετάσει το βαθμό συσχέτισης κάποιου χρήστη με τους άλλους χρήστες με βάση τις προτιμήσεις σε συγκεκριμένες κατηγορίες ταινιών.

Ένα σημαντικό στοιχείο είναι ότι στην σελίδα η οποία δείχνει το βαθμό συσχέτισης κάποιου χρήστη με άλλους χρήστες, υπολογίζεται και ένας άλλος συντελεστής ο βαθμός συνοχής της κοινότητας (Community Coherence Grade) (Σχεδιάγραμμα 12).



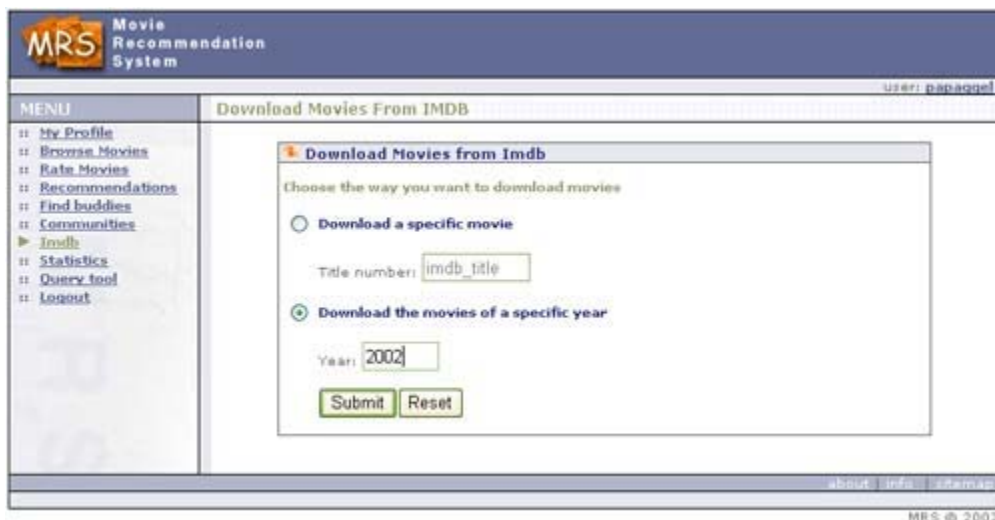
(Σχεδιάγραμμα 12)

Ο συντελεστής αυτός υπολογίζει όπως ορίζει το όνομά του τον βαθμό συνοχής μίας κοινότητας, δηλαδή πόσο σχετικά είναι τα άτομα τα οποία την αποτελούν. Η κοινότητα αυτή είναι δυναμική και μπορεί να αλλάζει από στιγμή σε στιγμή καθώς στηρίζεται στα προφίλ των χρηστών τα οποία επίσης είναι δυναμικά και αλλάζουν με τις βαθμολογίες των χρηστών.

Μέσω αυτής της διαδικασίας λοιπόν ο διαχειριστής είναι σε θέση να παρακολουθεί ποιες από τις κοινότητες είναι πιο δεμένες, πιο δραστήριες, πιο πραγματικές.

## 7.9 Imdb

Η διαδικασία αυτή είναι προσιτή μόνο στον administrator του συστήματος. Αποτελείται από τρεις επιλογές οι οποίες επιτρέπουν την αυτόματη ενημέρωση της βάσης δεδομένων με όλες τις καινούριες ταινίες. Η ενημέρωση γίνεται από τη βάση του IMDB(Internet Movie Data Base) η οποία είναι η μεγαλύτερη και γνωστότερη βάση δεδομένων για ταινίες στο Διαδίκτυο. Με μηχανισμούς parsing είναι δυνατή η ενημέρωση της βάσης δεδομένων του συστήματος. Ο administrator μπορεί να επιλέξει εάν θέλει να κατεβάσει μία συγκεκριμένη ταινία οπότε θα πρέπει να γνωρίζει το συγκεκριμένο «Title» το οποίο έχει η ταινία στο IMDB, ή μπορεί να κατεβάσει τις ταινίες μίας συγκεκριμένης χρονιάς οπότε δίνει ως όρισμα το έτος που επιθυμεί. Επίσης με ξεχωριστό parser μπορεί να κατεβάσει τα εξώφυλλα των ταινιών τις οποίες έχει στη βάση του (Σχεδιάγραμμα 13) ή να ανανεώσει τις βαθμολογίες των ταινιών που ήδη υπάρχουν στη βάση με τις καινούριες βαθμολογίες του IMDB.



(Σχεδιάγραμμα 13)

Στη διαδικασία αυτή οφείλεται η γρήγορη, εύκολη και έγκυρη ενημέρωση της βάσης δεδομένων του συστήματος MRS.

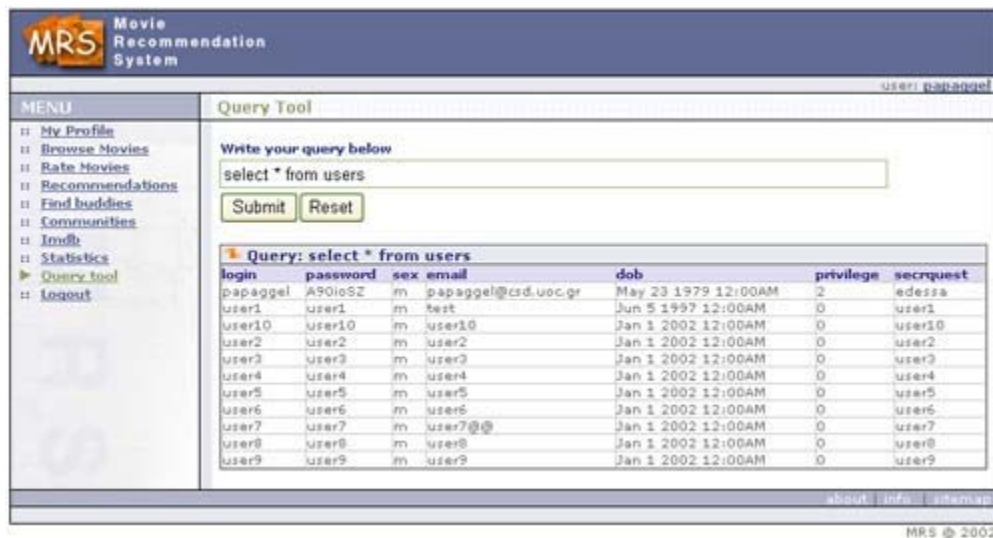
### **7.10 Statistics**

Η διαδικασία αυτή είναι μία από τις σημαντικότερες τις οποίες έχει στη διάθεσή του ο administrator γιατί του επιτρέπει να παρακολουθεί τη δραστηριότητα των χρηστών του συστήματος ανά πάσα στιγμή. Με τη βοήθεια της διαδικασίας αυτής ο administrator μπορεί:

- Να δει τον συνολικό αριθμό των εγγεγραμμένων χρηστών
- Να δει το συνολικό αριθμό των ψήφων από μη-εγγεγραμμένους χρήστες
- Να δει τη κατανομή των ψήφων των μη-εγγεγραμμένων χρηστών σε απόλυτες και ποσοστιαίες επί τις 100 τιμές
- Να δει το μέσο όρο των ψήφων των μη-εγγεγραμμένων χρηστών
- Να δει το συνολικό αριθμό των ψήφων από εγγεγραμμένους χρήστες
- Να δει τη κατανομή των ψήφων των εγγεγραμμένων χρηστών σε απόλυτες και ποσοστιαίες επί τις 100 τιμές
- Να δει το μέσο όρο των ψήφων των εγγεγραμμένων χρηστών
- Να δει γραφήματα για τα στατιστικά αυτά σε διάφορες μορφές (pie chart, bar chart, horizontal bar chart, line chart, area chart)
- Να σώσει τις γραφήματα αυτά καθώς είναι εικόνες

### **7.11 Query Tool**

Το εργαλείο επερωτήσεων είναι ένα πολύ χρήσιμο εργαλείο στη διάθεση του administrator το οποίο του δίνει τη δυνατότητα να εκτελεί SQL επερωτήσεις μέσω του Browser της εφαρμογής. Με τον τρόπο αυτό ο administrator μπορεί ανά πάσα στιγμή να γνωρίζει τι γίνεται στη βάση και να επιλύει άμεσα διάφορα προβλήματα τα οποία προκύπτουν (Σχεδιάγραμμα 14).



(Σχεδιάγραμμα 14)

## 7.12 Logout

Η διαδικασία εξόδου επιτρέπει στον χρήστη να βγει από τις προσωπικές του σελίδες και να μην είναι πλέον αναγνωρίσιμος από το σύστημα. Η αναγκαιότητα του έγκειται κυρίως σε λόγους ασφαλείας. Το σύστημα τη στιγμή της εξόδου σβήνει όλα τα στοιχεία που κρατούσε πρόσκαιρα για τον χρήστη στο session και αποχαιρετά τον χρήστη, ευχαριστώντας τον για τη χρήση του συστήματος.

## 8. Αλγόριθμοι Υλοποίησης Συστάσεων

---

### 8.1 Αλγόριθμος Υλοποίησης Συστάσεων 1

Ο πρώτος αλγόριθμος ο οποίος χρησιμοποιήθηκε στηρίζεται στην απλή λογική του να προτείνει ταινίες οι οποίες έχουν την μεγαλύτερη MRS βαθμολογία. Η MRS βαθμολογία προκύπτει από τη μέση βαθμολογία των χρηστών του MRS συστήματος για αυτή την ταινία και από τη βαθμολογία την οποία έχει δώσει το IMDB για αυτή την ταινία [A4]. Ο αλγόριθμος δέχεται ως ορίσματα τα βάρη με τα οποία κάθε μέση βαθμολογία (MRS και IMDB) συμμετέχει στον υπολογισμό της τελικής βαθμολογίας. Τυπικά βάρη είναι 80% να επηρεάζει το IMDB και 20% το MRS. Επειδή το MRS δέχεται ψήφους τόσο από εγγεγραμμένους χρήστες όσο και από μη εγγεγραμμένους χρήστες κρίθηκε απαραίτητο για την μεγαλύτερη αξιοπιστία του αποτελέσματος να συμμετέχουν οι ψήφοι εγγεγραμμένων και μη εγγεγραμμένων χρηστών με διαφορετικό βάρος στον υπολογισμό της μέσης MRS βαθμολογίας. Τυπικές τιμές των βαρών είναι να επηρεάζουν κατά 80% οι ψήφοι των εγγεγραμμένων και 20% οι ψήφοι των μη-εγγεγραμμένων.

Τα βήματα του αλγορίθμου [A5] φαίνονται παρακάτω αναλυτικά:

1. Βρες ποιες ταινίες δεν έχει ψηφίσει ο χρήστης (αυτό μόνο εάν είναι εγγεγραμμένος χρήστης αλλιώς για όλες τις ταινίες)
2. Για κάθε ταινία κάνε τα εξής:
3. Υπολόγισε τον μέσο όρο των ψήφων που έχει πάρει η ταινία από μη εγγεγραμμένους χρήστες. (Έστω  $Unreg$ )
4. Υπολόγισε τον μέσο όρο των ψήφων που έχει πάρει η ταινία από εγγεγραμμένους χρήστες. (Έστω  $Reg$ )
5. Υπολόγισε το μέσο όρο των δύο τιμών που έχουν προκύψει από τα βήματα 3 και 4 δεδομένου ότι αυτοί συμμετέχουν με διαφορετικά βάρη. Ο  $UnregMean$  με βάρος  $W_{unreg}$  και ο  $Reg$  με βάρος  $W_{reg}$  (Έστω  $Mrs\_rating$ )
6. Πάρε τον βαθμό του IMDB για την ταινία αυτή. (Έστω  $IMDB\_rating$ )
7. Υπολόγισε το μέσο όρο των δύο τιμών που έχουν προκύψει από τα βήματα 5 και 6 δεδομένου ότι αυτοί συμμετέχουν με διαφορετικά βάρη. Ο  $MRS\_rating$  με βάρος  $W_{MRS}$  και ο  $IMDB\_rating$  με βάρος  $W_{IMDB}$  (Έστω  $Mrs$ )
8. Εάν η τιμή του βήματος 7 (MRS) είναι μεγαλύτερο από το όριο ποιότητας (quality bound) τότε πρότεινέ την. Τυπική τιμή του ορίου ποιότητας είναι το 8

## 9. Πήγαινε στο βήμα 2

Υπολογισμός του MRS

$$MRS = W_{IMDB} * IMDB_{rating} + W_{MRS} * (W_{reg} * Reg_{avg} + W_{unreg} * UnReg_{avg})$$

## 8.2 Αλγόριθμος Υλοποίησης Συστάσεων 2 (Collaborative Filtering)

Ο δεύτερος αλγόριθμος υλοποίησης συστάσεων χρησιμοποιεί μία γνωστή τεχνική στα συστήματα συστάσεων η οποία ονομάζεται Collaborative Filtering (CF). Σύμφωνα με την τεχνική αυτή σε έναν χρήστη θα πρέπει να αρέσουν ταινίες οι οποίες αρέσουν σε χρήστες με παρόμοιο προφίλ.

Αντικείμενο ερευνητικής μελέτης αποτελεί ο τρόπος με τον οποίο δημιουργείται το προφίλ του χρήστη αλλά και ο τρόπος με τον οποίο προκύπτει ο βαθμός συσχέτισης μεταξύ δύο προφίλ χρηστών.

Ο αλγόριθμος αυτός στηρίζει τα αποτελέσματά του στα προφίλ των χρηστών τα οποία προκύπτουν με βάση τις αξιολογήσεις (ratings) που αυτοί έχουν κάνει στις ταινίες.

Έτσι λοιπόν το πρόβλημα επικεντρώνεται στο πώς μπορούμε να συγκρίνουμε τις ψήφους δύο χρηστών.

Ένας περιορισμός που προκύπτει είναι ότι για να γίνει η σύγκριση θα πρέπει και οι δύο χρήστες να έχουν ψηφίσει μερικές ίδιες ταινίες.

Έχοντας τις βαθμολογίες των δύο χρηστών για τις κοινές ταινίες που έχουν βαθμολογήσει μπορούμε να υπολογίσουμε το βαθμό συσχέτισής τους με τη βοήθεια του συντελεστή συσχέτισης του Pearson (Pearson Correlation Coefficient) [A6]. Αυτός ο συντελεστής συσχέτισης παίρνει τιμές από -1 που υποδηλώνει ότι είναι άσχετα τα προφίλ μέχρι 1 το οποίο υποδηλώνει πλήρη ταύτιση.

Αν και ο συντελεστής συσχέτισης Pearson παίρνει τιμές από -1 έως 1 μπορούμε να θέσουμε ένα όριο με το οποίο διακρίνουμε το βαθμό συσχέτισης των δύο προφίλ σε ικανοποιητικό ή μη ικανοποιητικό. Με τον τρόπο αυτό μπορούμε εύκολα να οδηγηθούμε σε τεχνικές που παράγουν συστάσεις σε κάποιο χρήστη.

Για κάποιο χρήστη συγκρίνουμε το προφίλ του με το προφίλ όλων των άλλων χρηστών προσπαθώντας να βρούμε αυτούς που ικανοποιούν ένα όριο βαθμού συσχέτισης το οποίο ορίζουμε. Το σύνολο των χρηστών που προκύπτει θα το ονομάζουμε «συμβούλους» (consultants), καθώς από αυτούς θα προκύψει αν πρέπει ή όχι να προταθεί μία ταινία [A7].

Για κάθε ταινία που δεν έχει δει ο χρήστης επιλέγονται εκείνοι από την ομάδα των συμβούλων οι οποίοι έχουν βαθμολογήσει την ταινία. Υπολογίζεται ο μέσος όρος των βαθμολογιών αυτών. Αν ο μέσος όρος είναι μεγαλύτερος από ένα όριο ποιότητας (quality bound) που ορίζεται αρχικά στο σύστημα τότε η ταινία αυτή θα προταθεί στον χρήστη, αλλιώς δε θα προταθεί.

Όταν η διαδικασία αυτή τελειώσει θα έχει επιλεγεί ένας αριθμός ταινιών οι οποίες θα προταθούν στον χρήστη με προτεραιότητα σε αυτές που είχαν τον μεγαλύτερο μέσο όρο μεταξύ των συμβούλων.

Τα βήματα του αλγορίθμου [A10] φαίνονται παρακάτω αναλυτικά:

1. Βρες ποιοι είναι οι σύμβουλοι αυτού του χρήστη.
  - a. Βρες όλους τους χρήστες οι οποίοι έχουν ψηφίσει τουλάχιστον μία φορά
  - b. Για τον καθένα από αυτούς σύγκρινε το προφίλ του με του αρχικού χρήστη
    - i. Βρες όλες τις ταινίες τις οποίες τις έχουν ψηφίσει και οι δύο
    - ii. Χρησιμοποίησε το συντελεστή συσχέτισης του Pearson (Pearson Correlation Coefficient) για να υπολογίσεις το βαθμό συσχέτισης μεταξύ των δύο προφίλ [A6].
  - c. Αν ικανοποιείται ο βαθμός συσχέτισης τότε αυτός είναι σύμβουλός αλλιώς δεν είναι
2. Βρες ποιες ταινίες δεν έχει βαθμολογήσει ο χρήστης.
3. Για κάθε τέτοια ταινία βρες ποιοι από τους συμβούλους την έχουν ψηφίσει.
4. Από τις βαθμολογίες αυτών υπολόγισε το μέσο όρο βαθμολογίας.
5. Αν ο μέσος όρος είναι μεγαλύτερος από το όριο ποιότητας τότε πρότεινέ την αλλιώς μην την προτείνεις.
6. Συνέχισε με το βήμα 3 μέχρι να εξαντληθούν όλες οι ταινίες

Pearson Correlation Coefficient (PCC) between user X and Y [A6]

$$PCC(X, Y) = \frac{\sum (U_x - U_{avg\_x})(U_y - U_{avg\_y})}{\sqrt{\sum (U_x - U_{avg\_x})^2 * \sum (U_y - U_{avg\_y})^2}}$$

Όπου  $U_x$  είναι η βαθμολογία του χρήστη X στην συγκεκριμένη ταινία και  $U_{avg\_x}$  είναι ο μέσος όρος των βαθμολογιών που έχει δώσει ο χρήστης X στις κοινές ταινίες. Αντίστοιχα ισχύουν για τον χρήστη Y.

Αν ο PCC είναι μεγαλύτερος από το όριο ποιότητας (quality bound), τότε ο Y είναι σύμβουλος (consultant), αλλιώς δεν είναι.

### 8.3 Αλγόριθμος Υλοποίησης Συστάσεων 3

Ο αλγόριθμος αυτός σε αντίθεση με τους προηγούμενους δεν στηρίζεται στα explicit ratings τα οποία δίνει ο χρήστης αλλά στο προφίλ που έχει δημιουργηθεί από το σύστημα για τον χρήστη αυτόν, σε συνδυασμό με τις βαθμολογίες που βάζει, αλλά και με το είδος των ταινιών που ψηφίζει. Για να γίνει πιο κατανοητό θα δούμε πρώτα τον τρόπο με τον οποίο δημιουργείται το προφίλ του χρήστη.

Όλες οι ταινίες που υπάρχουν στη βάση δεδομένων είναι κατηγοριοποιημένες. Μία ταινία μπορεί να ανήκει σε περισσότερες από μία κατηγορίες. Σκοπός του προφίλ του χρήστη είναι να συγκεντρώνει πληροφορία σχετικά με το ποια είδη ταινιών προτιμάει να βλέπει ο χρήστης και ποια όχι.

Για το λόγο αυτό κάθε φορά που ο χρήστης ψηφίζει μία ταινία που ανήκει στις κατηγορίες π.χ. Action, Drama, Thriller, αν η ψήφος του είναι θετική τότε στο προφίλ του η κατηγορία ActionPos, DramaPos, ThrillerPos, αυξάνεται κατά ένα. Αν η ψήφος του ήταν αρνητική τότε στο προφίλ του η κατηγορία ActionNeg, DramaNeg, ThrillerNeg θα αυξανόταν κατά ένα. Έτσι λοιπόν το προφίλ του κάθε χρήστη αποτελείται από το πόσες φορές έχει ψηφίσει θετικά και πόσες φορές έχει ψηφίσει αρνητικά για κάποιες κατηγορίες. Ο τρόπος αυτός για δημιουργία προφίλ είναι ουσιαστικά ένας έμμεσος τρόπος βαθμολόγησης (implicit rating), καθώς μπορεί ο



χρήστης να μη γνωρίζει σε ποιες ακριβώς κατηγορίες ανήκει η ταινία που ψήφισε. Θεωρούμε ότι θετική ψήφος είναι ψήφος από 6-10 και αρνητική ψήφος από 1-5.

Έτσι λοιπόν καθώς ο χρήστης ψηφίζει όλο και περισσότερες ταινίες τόσο το προφίλ του εμπλουτίζεται και τόσο πιο αξιόπιστες γίνονται οι συστάσεις του συστήματος. Παρακάτω αναλύεται ο τρόπος με τον οποίο το σύστημα επιλέγει ποιες ταινίες θα προτείνει στον χρήστη με βάση το ήδη δημιουργημένο προφίλ του.

Για κάθε ταινία που δεν έχει ψηφίσει ο χρήστης το σύστημα θα βρει σε ποιες κατηγορίες ανήκει η ταινία αυτή. Έπειτα θα υπολογίσει το ποσοστό των θετικών ψήφων του χρήστη σε ταινίες που ανήκουν σε τέτοια κατηγορία έναντι των αρνητικών ψήφων του χρήστη σε ταινίες που ανήκουν σε τέτοια κατηγορία [A12]. Μετά θα υπολογιστεί ο μέσος όρος των θετικών ποσοστών και αν αυτό είναι μεγαλύτερο από αυτό που ορίζεται από το όριο ποιότητας (quality bound) τότε η ταινία θα προταθεί.

Τα βήματα του αλγορίθμου [A11] φαίνονται παρακάτω αναλυτικά:

1. Βρες όλες τις ταινίες τις οποίες δεν έχει ψηφίσει ο χρήστης.
2. Για κάθε μία από αυτές τις ταινίες βρες σε ποιες κατηγορίες ανήκει.
3. Για κάθε μία από αυτές τις κατηγορίες υπολόγισε το ποσοστό των θετικών ψήφων που έχει δώσει ο χρήστης [A12].
4. Υπολόγισε τον μέσο όρο των ποσοστών αυτών.
5. Αν ο μέσος όρος είναι μεγαλύτερος από το όριο ποιότητας τότε πρότεινε την ταινία αυτή.
6. Συνέχισε με το βήμα 2 μέχρι να τελειώσουν όλες οι ταινίες.

Μέσος όρος προτίμησης σε ταινία με βάση τις κατηγορίες που αυτή ανήκει

$$Avg_{Movie} = \frac{\sum_{i=1}^k \frac{Pos_i}{Pos_i + Neg_i}}{k}, i = 1, 2, \dots, k$$

Όπου  $k$  είναι ο αριθμός των κατηγοριών στις οποίες ανήκει η ταινία.  $Pos_i$  είναι οι θετικές ψήφοι που έχει δώσει ο χρήστης στην κατηγορία  $i$  και  $Neg_i$  είναι οι αρνητικές ψήφοι που έχει δώσει ο χρήστης στην κατηγορία  $i$ .

#### **8.4 Ανάλυση και συμπεράσματα από την εφαρμογή των αλγορίθμων συστάσεων**

Στο σημείο αυτό θα επιχειρήσουμε μία σύγκριση των παραπάνω αλγορίθμων με σκοπό να εντοπίσουμε τα πλεονεκτήματα και τα μειονεκτήματα του καθενός. Η φύση των τριών αλγορίθμων είναι διαφορετική καθώς ο καθένας στηρίζει τα αποτελέσματά του σε διαφορετικά δεδομένα.

Αναφορικά για κάθε αλγόριθμο ισχύουν τα εξής:

- Ο πρώτος αλγόριθμος στηρίζεται αποκλειστικά στις βαθμολογίες τις οποίες έχει συγκεντρώσει μία ταινία και οι οποίες προκύπτουν από ψήφους χρηστών.
- Ο δεύτερος αλγόριθμος στηρίζεται στον εντοπισμό χρηστών με παρόμοιο προφίλ και στις βαθμολογίες που αυτοί δίνουν σε ταινίες που έχουν δει.
- Ο τρίτος αλγόριθμος στηρίζεται στις προτιμήσεις του χρήστη σε συγκεκριμένες κατηγορίες ταινιών.

Ο πρώτος αλγόριθμος έχει το πλεονέκτημα ότι στηρίζει τα αποτελέσματά του στο σύνολο των βαθμολογιών που έχει πάρει μία ταινία, ανεξάρτητα από τον χρήστη που έδωσε την κάθε ψήφο. Ο μέσος όρος των ψήφων που υπολογίζεται για κάθε ταινία τις περισσότερες φορές είναι ικανοποιητικός για να αποτελέσει κριτήριο σχετικό με το εάν η ταινία είναι καλή, μέτρια ή κακή. Αναμφισβήτητα μία ταινία με μέσο όρο ψήφων 8, από το σύνολο των χρηστών, πιθανότατα είναι υποψήφια για σύσταση. Ένας περιορισμός που προκύπτει είναι η ταινία να έχει ψηφιστεί από έναν ικανοποιητικό αριθμό χρηστών, ούτως ώστε ο μέσος όρος να είναι αξιόπιστος.

Παρόλα αυτά ο αλγόριθμος αυτός θα προτείνει ταινίες οι οποίες είναι αρεστές στην πλειοψηφία των χρηστών. Στην ουσία θα υποτιμήσει την άποψη των λίγων και θα υπερεκτιμήσει την άποψη των πολλών. Θα μπορούσαμε στον αλγόριθμο αυτό να προσδώσουμε τον χαρακτηρισμό «Δημοκρατικός».

Ο δεύτερος αλγόριθμος έχει το πλεονέκτημα ότι στηρίζει τα αποτελέσματά του στην άποψη των «συμβούλων». Οι σύμβουλοι είναι οι χρήστες οι οποίοι έχουν παρόμοιο προφίλ με το προφίλ του χρήστη που ζητάει συστάσεις. Από τις βαθμολογίες που έχουν δώσει αυτοί υπολογίζεται ο μέσος όρος της ταινίας και αν αυτός είναι μεγάλος τότε η ταινία θα προταθεί στον χρήστη.

Παρόλα αυτά ο αλγόριθμος αυτός προϋποθέτει την ύπαρξη πολλών χρηστών ούτως ώστε να είναι δυνατός ο εντοπισμός μία ομάδας συμβούλων για οποιοδήποτε χρήστη. Ακόμη όμως και να υπάρχουν πολλοί χρήστες στο σύστημα αν ένας χρήστης έχει πολύ ιδιόρρυθμο τρόπο να βαθμολογεί ταινίες τότε για αυτόν θα είναι πολύ δύσκολο να βρει συμβούλους οδηγώντας έτσι τον αλγόριθμο σε αποτυχία.

Ο αλγόριθμος αυτός προσομοιώνει σε μεγάλο βαθμό τον τρόπο με τον οποίο κάποιος αποφασίζει να δει μία ταινία κινηματογράφου στην πραγματική ζωή. Συνήθως κάποιος πριν πάει να δει μία ταινία συμβουλευεται τους γνωστούς του ποια ταινία θεωρούν ότι είναι καλή, και κυρίως παίρνει υπ' όψιν του τη γνώμη των ανθρώπων με τους οποίους στο παρελθόν έχουν συμφωνήσει για το ποια ταινία ήταν καλή και ποια δεν ήταν. Ωστόσο υπάρχουν άνθρωποι που ποτέ η γνώμη τους δεν συμφωνεί με τη γνώμη των άλλων.

Ο τρίτος αλγόριθμος έχει το πλεονέκτημα ότι στηρίζει τα αποτελέσματά του στις προτιμήσεις του ίδιου του χρήστη σε συγκεκριμένες κατηγορίες ταινιών. Καθώς αυτός ψηφίζει θετικά ή αρνητικά ταινίες το σύστημα διαμορφώνει το προφίλ του με βάση σε ποιες κατηγορίες ταινιών δίνει καλή ψήφο και σε ποιες κατηγορίες δίνει αρνητική ψήφο. Το αποτέλεσμα είναι το σύστημα να είναι σε θέση να κρίνει αν μία ταινία η οποία ανήκει σε συγκεκριμένες κατηγορίες θα αρέσει ή δε θα αρέσει στον χρήστη. Το πλεονέκτημα του αλγορίθμου αυτού είναι ότι δεν προϋποθέτει την ύπαρξη άλλων χρηστών.

Παρόλα αυτά ο αλγόριθμος αυτός θα αποτύχει αν ο χρήστης έχει πολύ φτωχό προφίλ, δηλαδή δεν έχει δώσει πολλές πληροφορίες για το ποιες κατηγορίες ταινιών του αρέσουν και ποιες όχι. Ένας άλλος κίνδυνος είναι ο κίνδυνος της γενίκευσης. Δηλαδή αν στον χρήστη αρέσουν οι ταινίες με δράση είναι μεγάλος ο κίνδυνος ο αλγόριθμος να συστήνει όλες τις ταινίες δράσης.

### **8.5 Θέματα απόδοσης και αξιοπιστίας αλγορίθμων**

Οι παραπάνω αλγόριθμοι είναι διαφορετικοί σε φύση όπως αναφέρθηκε και προηγουμένως με αποτέλεσμα η απόδοσή τους να επηρεάζεται με ιδιόρρυθμο τρόπο. Μπορεί ωστόσο να θεωρηθεί ακριβές ότι η απόδοσή τους και η αξιοπιστία τους επηρεάζεται από συγκεκριμένα χαρακτηριστικά του συστήματος.

Όσον αφορά την απόδοση αυτά είναι:

- Ο αριθμός των χρηστών
- Ο αριθμός των ταινιών
- Η δραστηριότητα του χρήστη

Όσον αφορά την αξιοπιστία αυτά είναι:

- Ο αριθμός των ψήφων της κάθε ταινίας
- Ο αριθμός των κατηγοριών στις οποίες ανήκει κάθε ταινία
- Η διαφοροποίηση του χρήστη από τους υπόλοιπους χρήστες

Η απόδοση του πρώτου αλγόριθμου επηρεάζεται αποκλειστικά από τον αριθμό των ταινιών που υπάρχουν στο σύστημα. Όσο λιγότερες είναι οι ταινίες που υπάρχουν τόσο πιο γρήγορα θα τελειώσει ο αλγόριθμος.

Η αξιοπιστία του αλγορίθμου επηρεάζεται αποκλειστικά από τον αριθμό των ψήφων που έχει πάρει μία ταινία. Όσο περισσότερες ψήφους έχει πάρει μία ταινία τόσο πιο αξιόπιστος είναι και ο μέσος όρος βαθμολογίας για αυτήν. Για λόγους αξιοπιστίας ο αλγόριθμος αυτός συνυπολογίζει στον καθορισμό του μέσου όρου και τη βαθμολογία που η ταινία έχει πάρει από το IMDB.

Τυπικός χρόνος απόκρισης πρώτου αλγορίθμου για 550 ταινίες: 3.5512279272079 sec

Η απόδοση του δεύτερου αλγορίθμου επηρεάζεται από τον αριθμό των χρηστών και από τον αριθμό των ταινιών που δεν έχει δει ο χρήστης. Όσο περισσότεροι είναι οι χρήστες τόσο πιο πολύ θα αργεί ο αλγόριθμος να βρει τους «συμβούλους». Όσο πιο μεγάλος είναι ο αριθμός των ταινιών τόσο πιο πολύ θα αργεί ο αλγόριθμος αφού σε κάθε ταινία αντιστοιχίζεται μία βαθμολογία (ο μέσος όρος της βαθμολογίας των «συμβούλων»).

Η αξιοπιστία του αλγορίθμου επηρεάζεται από τον αριθμό των συμβούλων που θα προκύψουν. Αν κάποιος χρήστης είναι ιδιόρρυθμος τότε η πιθανότητα είναι να προκύψουν λίγοι «σύμβουλοι» και αυτό να οδηγήσει σε μη πολύ αξιόπιστα αποτελέσματα. Αντίθετα αν βρεθούν πολλοί «σύμβουλοι» το αποτέλεσμα θα είναι πιο αξιόπιστο.

Τυπικός χρόνος απόκρισης δεύτερου αλγορίθμου για 550 ταινίες και 55 χρήστες:  
1.9614459037781 sec

Η απόδοση του τρίτου αλγορίθμου επηρεάζεται από τον αριθμό των ταινιών και από τον αριθμό των κατηγοριών στις οποίες ανήκει μία ταινία. Όσο περισσότερες ταινίες υπάρχουν τόσο πιο πολύ αργεί ο αλγόριθμος. Επίσης όσο πιο πολλές είναι οι κατηγορίες στις οποίες ανήκει μία ταινία τόσο πιο πολύ αργεί ο αλγόριθμος.

Η αξιοπιστία του αλγορίθμου επηρεάζεται από τη δραστηριότητα του χρήστη και από τον αριθμό των κατηγοριών στις οποίες ανήκει κάθε ταινία. Αν ο χρήστης έχει ψηφίσει πολλές ταινίες με αποτέλεσμα να έχει πλούσιο προφίλ τα αποτελέσματα του αλγορίθμου θα είναι καλύτερα. Αν οι κατηγορίες στις οποίες ανήκει μία ταινία είναι πολλές τότε επίσης η αξιοπιστία θα είναι μεγαλύτερη.

Τυπικός χρόνος απόκρισης τρίτου αλγορίθμου για 550 ταινίες με 2,3 κατηγορίες για κάθε ταινία: 4.1300460100174 sec

Από τα παραπάνω γίνεται σαφές ότι στις περισσότερες περιπτώσεις το θέμα της απόδοσης και το θέμα της αξιοπιστίας έρχονται σε σύγκρουση. Αυτό είναι λογικό μιας και θα περιμέναμε όσο περισσότερη υπολογιστική δύναμη διαθέτουμε τόσο πιο αξιόπιστα αποτελέσματα να παίρνουμε και το αντίστροφο.

Οι μετρήσεις που αφορούν την απόδοση των αλγορίθμων έγιναν σε υπολογιστή Pentium III στα 600MHz με μνήμη RAM 256MB, σε λειτουργικό σύστημα Windows 2000.

Στο σημείο αυτό θα πρέπει να γίνει σαφές ότι υπάρχει διαχωρισμός μεταξύ του χρόνου τον οποίο χρειάζεται ο αλγόριθμος για να υπολογίσει τις ταινίες που θα συσταθούν στον τελικό χρήστη και στον χρόνο που χρειάζεται για να γραφούν οι πληροφορίες για τις ταινίες αυτές στην HTML σελίδα. Ο χρόνος για να γραφούν οι πληροφορίες για την ταινία στην HTML σελίδα ποικίλει και εξαρτάται κυρίως από τον αριθμό των ταινιών που τελικά θα προταθούν στον χρήστη αλλά είναι πολύ μεγαλύτερος γενικά.

## 9. Άλλοι αλγόριθμοι

---

### 9.1 Αλγόριθμος Πρόβλεψης Ψήφου

Ο αλγόριθμος πρόβλεψης ψήφου εκμεταλλεύεται την πληροφορία σχετικά με τις ψήφους που έχει ήδη δώσει ο χρήστης στο σύστημα καθώς και τις ψήφους άλλων χρηστών για κάποια συγκεκριμένη ταινία για να προβλέψει την ψήφο που θα έδινε ο χρήστης αν ψήφιζε.

Ο αλγόριθμος συνοπτικά υπολογίζει το μέσο όρο των ψήφων που έχει δώσει ο χρήστης μέχρι τότε και σε αυτόν τον μέσο όρο προσθέτει μία απόκλιση που είναι ο βεβαρημένος μέσος όρος των ψήφων που έχει πάρει η συγκεκριμένη ταινία από άλλους χρήστες με βάρη τον βαθμό συσχέτισης του προφίλ του εκάστοτε χρήστη με το προφίλ του χρήστη για τον οποίο γίνεται η πρόβλεψη ψήφου.

Τα βήματα του αλγορίθμου [A14] φαίνονται παρακάτω αναλυτικά:

1. Βρες το μέσο όρο των ψήφων που έχει δώσει ο χρήστης.
2. Βρες όλους τους χρήστες που έχουν ψηφίσει την συγκεκριμένη ταινία.
3. Για κάθε χρήστη βρες το βαθμό συσχέτισης του προφίλ του με το προφίλ του αρχικού χρήστη (για τον οποίο γίνεται η πρόβλεψη) με τη βοήθεια του συντελεστή συσχέτισης του Pearson [A6] (Αυτός ο συντελεστής καθορίζει πόσο θα επηρεάζει ο κάθε χρήστης στο τελικό αποτέλεσμα - βάρος)
4. Για κάθε χρήστη βρες τον μέσο όρο των ψήφων που έχει δώσει
5. Για κάθε χρήστη βρες τον βαθμό που έχει δώσει στη συγκεκριμένη ταινία
6. Υπολόγισε το βεβαρημένο άθροισμα που αποτελεί την απόκλιση σύμφωνα με τον τύπο που φαίνεται παρακάτω.
7. Η πρόβλεψη προκύπτει από το άθροισμα του μέσου όρου των ψήφων που έχει δώσει ο χρήστης του βήματος 1 με την απόκλιση του βήματος 6.

Αλγόριθμος πρόβλεψης ψήφου [A14]

Προβλέπει πως θα ψηφίσει ο χρήστης  $i$  την ταινία  $j$ .

$$\hat{u}_{ij} = \bar{U}_i + \frac{\sum_1^k w_{kj} (u_{kj} - \bar{U}_k)}{\sum_1^k |w_{kj}|}$$

Όπου

K είναι το σύνολο των χρηστών που έχουν ψηφίσει την ταινία j.

U<sub>i</sub> είναι ο μέσος όρος των ψήφων που έχει δώσει ο χρήστης i και για τον οποίο γίνεται η πρόβλεψη

Το βάρος w<sub>kj</sub> είναι ο βαθμός συσχέτισης του χρήστη k και του χρήστη i.

## 9.2 Αλγόριθμος εύρεσης «φίλων» με βάση τις προτιμήσεις σε κατηγορίες ταινιών

Ο αλγόριθμος αυτός [A9] χρησιμοποιείται για την εύρεση «φίλων» με βάση τις προτιμήσεις σε κατηγορίες ταινιών. Δηλαδή, καθώς για τον κάθε χρήστη έχει αναπτυχθεί ένα προφίλ το οποίο στηρίζεται στις προτιμήσεις του σε κάποιες κατηγορίες ταινιών είναι χρήσιμο να μπορεί αυτό το προφίλ να συγκριθεί με το αντίστοιχο προφίλ άλλων χρηστών. Ο αλγόριθμος θα υπολογίσει τη σχέση των προφίλ αυτών [A8] και θα κρίνει αν δύο υποκείμενοι χρήστες είναι σχετικοί ή όχι, αν δηλαδή έχουν παρόμοιες προτιμήσεις.

Τα βήματα του αλγορίθμου [A9] φαίνονται παρακάτω αναλυτικά:

1. Βρες όλους τους χρήστες που έχουν προφίλ (έχουν ψηφίσει τουλάχιστον μία φορά)
2. Βρες το βαθμό συσχέτισης του προφίλ του κάθε χρήστη με αυτό του αρχικού χρήστη.
  - i. Πάρε το προφίλ του πρώτου χρήστη (από τον πίνακα userprofiler)
  - ii. Πάρε το προφίλ του δεύτερου χρήστη (από τον πίνακα userprofiler)
  - iii. Υπολόγισε το συντελεστή συσχέτισης του Pearson για τα δύο προφίλ [A8]
3. Αν ο βαθμός συσχέτισης ικανοποιεί το όριο ποιότητας (quality bound) τότε θεώρησέ τον φίλο, αλλιώς όχι
4. Συνέχισε με το βήμα 2 μέχρι να τελειώσουν όλοι οι χρήστες.

### **9.3 Αλγόριθμος εύρεσης χρηστών που ικανοποιούν συγκεκριμένα κριτήρια (φίλτρα) όσον αφορά τις προτιμήσεις τους σε συγκεκριμένες κατηγορίες ταινιών**

Με τη βοήθεια του αλγορίθμου αυτού [A13] μπορούμε να εφαρμόσουμε φίλτρα που αφορούν το βαθμό προτίμησης σε συγκεκριμένες κατηγορίες και να εντοπίσουμε ποιοι από τους χρήστες ικανοποιούν τα κριτήρια που επιλέγηκαν από τη διαδικασία φιλτραρίσματος. Δηλαδή ο αλγόριθμος αυτός επιτρέπει τον εντοπισμό χρηστών οι οποίοι πληρούν κάποιες προϋποθέσεις, όσον αφορά τις προτιμήσεις τους στις κατηγορίες ταινιών.

Για παράδειγμα επιτρέπει να εντοπίσουμε χρήστες οι οποίοι προτιμούν ταινίες με 'Action' σε ποσοστό 80-100% ΚΑΙ προτιμούν 'Thriller' σε ποσοστό 0-20%.

Τα φίλτρα μπορούν να είναι πολύ πιο πολύπλοκα συνδυάζοντας όλες τις κατηγορίες ταινιών και 5 βαθμίδες προτίμησης. (0-20, 20-40, 40-60, 60-80, 80-100).

Ο αλγόριθμος σε όλες τις περιπτώσεις επιστρέφει τους χρήστες οι οποίοι πληρούν όλα τα κριτήρια του φιλτραρίσματος.

Τα βήματα του αλγορίθμου [A13] είναι τα εξής:

1. Βρες ποιες κατηγορίες ταινιών αφορούν τα κριτήρια που ορίστηκαν
2. Βρες για κάθε κατηγορία τι ποσοστό προτίμησης εφαρμόστηκε
3. Βρες όλους τους χρήστες του συστήματος
4. Για κάθε χρήστη βρες αν ικανοποιεί τα κριτήρια
  - i. Για κάθε κατηγορία ταινίας που ορίστηκε βρες το ποσοστό που έχει ο χρήστης στο προφίλ του
  - ii. Αν το ποσοστό αυτό ικανοποιεί το ποσοστό προτίμησης που ορίστηκε από τα κριτήρια τότε συνέχισε με την επόμενη κατηγορία, αλλιώς ο χρήστης δεν ικανοποιεί όλα τα κριτήρια.
  - iii. Αν ο χρήστης ικανοποιεί όλα τα κριτήρια τότε επιστρέφεται από τον αλγόριθμο
5. Συνέχισε με το βήμα 4 μέχρι να τελειώσουν όλοι οι χρήστες



Ο αλγόριθμος αυτός [A13] επιτρέπει να εντοπίζονται ομάδες ανθρώπων οι οποίες ικανοποιούν κάποια συγκεκριμένα κριτήρια. Με τον τρόπο αυτό είναι δυνατόν να εντοπιστούν ομοιότητες σε συγκεκριμένα θέματα. Με τους υπόλοιπους αλγορίθμους γινόταν σύγκριση του συνολικού προφίλ των χρηστών. Αυτό οδηγεί πολλές φορές σε λανθασμένα συμπεράσματα. Δηλαδή, ενώ δύο χρήστες μπορεί να θεωρούνται «άσχετοι» αν συγκρίνουμε τα προφίλ τους, είναι δυνατόν αυτοί σε συγκεκριμένες κατηγορίες ταινιών να έχουν ακριβώς τις ίδιες προτιμήσεις. Ο αλγόριθμος αυτός λοιπόν μας επιτρέπει να βρίσκουμε ομάδες χρηστών που ικανοποιούν συγκεκριμένα κριτήρια χωρίς απαραίτητα αυτό να σημαίνει ότι το συνολικό τους προφίλ μοιάζει.

#### **9.4 Κοινότητες χρηστών και βαθμός συνοχής κοινότητας**

Μέσα από την ανάπτυξη και ανάλυση των αλγορίθμων του συστήματος υπονοείται έντονα η ανάπτυξη κοινοτήτων χρηστών, χρηστών δηλαδή με κοινά χαρακτηριστικά, με ομοιόμορφη δραστηριότητα, με παρόμοιες προτιμήσεις σε κατηγορίες ταινιών.

Χρήστες που ψηφίζουν τις ίδιες ταινίες με παρόμοιους βαθμούς, χρήστες που προτιμούν να ψηφίζουν θετικά ή αρνητικά ταινίες με συγκεκριμένες κατηγορίες δεν μπορεί παρά να έχουν λόγο να έρθουν σε επαφή και να ανταλλάξουν απόψεις και ιδέες.

Το MRS σύστημα δίνει τη δυνατότητα στους χρήστες να οργανωθούν σε κοινότητες, και να βρουν ανθρώπους με παρόμοιες προτιμήσεις. Αυτή είναι η επιπρόσθετη αξία την οποία προσθέτει το σύστημα αυτό σε μία υπηρεσία απλής ενημέρωσης σχετικά με το ποιες ταινίες του κινηματογράφου είναι καλές και ποιες όχι, μία πληροφορία που μπορεί από μόνη της να αποδειχτεί άχρηστη αν κανείς σκεφτεί την υποκειμενικότητα που διακρίνει τους ανθρώπους που αγαπούν τον κινηματογράφο.

Οι κοινότητες χρηστών στο Διαδίκτυο είναι μία πραγματικότητα σήμερα, όμως διακρίνονται από τη στατικότητα τους. Ο χρήστης επιλέγει εξ' αρχής σε ποια κοινότητα θέλει να ανήκει και στις περισσότερες περιπτώσεις είναι μέλος μόνο μίας κοινότητας. Στο MRS σύστημα οι κοινότητες δημιουργούνται δυναμικά και αλλάζουν δυναμικά καθώς οι χρήστες αλληλεπιδρούν με το σύστημα ψηφίζοντας ταινίες. Επίσης ένας χρήστης μπορεί να ανήκει σε πολλές κοινότητες ταυτόχρονα. Αυτό το στοιχείο διαφοροποιεί το MRS από τα άλλα συστήματα που υποστηρίζουν τη δημιουργία

κοινοτήτων και του δίνει δυναμική διαφορετική για περαιτέρω ανάλυση της δραστηριότητας των χρηστών και της οργάνωσής τους σε εικονικές κοινότητες.

Ένα από τα χαρακτηριστικά των κοινοτήτων του MRS συστήματος αποτελεί ο βαθμός συνοχής της κοινότητας. Για τον κάθε χρήστη υπάρχει ένας αριθμός άλλων χρηστών οι οποίοι θεωρούνται «φίλοι» του (buddies). Αυτός και οι φίλοι του σχηματίζουν μία κοινότητα. Ο μέσος όρος των βαθμών συσχέτισης του προφίλ του χρήστη με τα προφίλ των φίλων του αποτελεί ένα χαρακτηριστικό της κοινότητας που θα το ονομάζουμε βαθμό συνοχής της κοινότητας (Community Coherence Grade). Καθώς οι κοινότητες είναι δυναμικές ο βαθμός συνοχής συνεχώς αλλάζει.

## 10. Μελλοντικές Επεκτάσεις

---

Το MRS σύστημα αποτελεί μία πρώτη προσπάθεια συγκέντρωσης πληροφορίας σχετικά με ταινίες κινηματογράφου, συγκέντρωσης ανθρώπων που έχουν ως ενδιαφέρον τους το κινηματογράφο, οργάνωσή τους σε κοινότητες, δημιουργία συστάσεων για ταινίες.

Είναι λογικό ένα τέτοιο σύστημα με ευρύ τόσο εμπορικό όσο και ερευνητικό ενδιαφέρον να συγκεντρώνει πολλές ιδέες επέκτασης. Παρακάτω αναφέρονται μερικές από τις ιδέες αυτές.

- Ανάπτυξη υβριδικών μεθόδων δημιουργίας συστάσεων, οι οποίες να συγκεντρώνουν περισσότερα πλεονεκτήματα και λιγότερα μειονεκτήματα έναντι των άλλων αλγορίθμων.
- Βελτιστοποίηση των υπαρχόντων αλγορίθμων ούτως ώστε να πραγματοποιείται πιο γρήγορα η απόκριση του συστήματος.
- Ανάπτυξη εφαρμογών για επικοινωνία μεταξύ των μελών μίας κοινότητας
- Βελτίωση του τρόπου εμφάνισης των ταινιών, έτσι ώστε να μην δημιουργούνται μεγάλες HTML σελίδες
- Εμπλουτισμός της βάσης δεδομένων με όλα τα στοιχεία κάθε ταινίας

## Παράρτημα Α (Ενδεικτικός Κώδικας)

Παρακάτω γίνεται παράθεση κώδικα του συστήματος. Σκοπός είναι να γίνει περαιτέρω κατανόηση των λειτουργιών του συστήματος καθώς και του τρόπου ανάπτυξης της εφαρμογής.

### A1. Η κλάση που υλοποιεί τις διάφορες συναλλαγές με τη βάση

```
// This class implements MSSQL connection and transactions  
class database{  
    var $Host    = "XXXXXXXXX";  
    var $Database = "XXXXXXXXX";  
    var $User    = "XXXXXXXXX";  
    var $Password = "XXXXXXXXX";  
  
    var $Link_ID = 0;  
    var $Query_ID = 0;  
    var $Record  = array();  
    var $Row     = 0;  
  
    var $Errno  = 0;  
    var $Error  = "";  
  
    var $Auto_Free = 0;  ### set this to 1 to automatically free results  
  
    function connect() {  
        if ( 0 == $this->Link_ID ) {  
            $this->Link_ID=mssql_connect($this->Host, $this->User, $this->Password);  
            if (!$this->Link_ID)  
                $this->halt("Link-ID == false, mssql_pconnect failed");  
            else  
                mssql_select_db($this->Database, $this->Link_ID);  
        }  
    }  
  
    function free_result(){  
        mssql_free_result($this->Query_ID);  
        $this->Query_ID = 0;  
    }  
  
    function query($Query_String) {
```

```

        if (!$this->Link_ID)
            $this->connect();

        //printf("<br>Debug: query = %s<br>\n", $Query_String);

        $this->Query_ID = mssql_query($Query_String, $this->Link_ID);
        $this->Row = 0;
        if (!$this->Query_ID) {
            $this->Errno = 1;
            $this->Error = "General Error (The MSSQL interface cannot return detailed error messages).";
            $this->halt("Invalid SQL: ".$Query_String);
        }
        return $this->Query_ID;
    }

    function next_record() {
        if ($this->Record = mssql_fetch_row($this->Query_ID)) {
            // add to Record[<key>]
            $count = mssql_num_fields($this->Query_ID);
            for ($i=0; $i<$count; $i++){
                $fieldinfo = mssql_fetch_field($this->Query_ID,$i);
                $this->Record[strtolower($fieldinfo->name)] = $this->Record[$i];
            }
            $this->Row += 1;
            $stat = 1;
        } else {
            if ($this->Auto_Free) {
                $this->free_result();
            }
            $stat = 0;
        }
        return $stat;
    }

    function seek($pos) {
        mssql_data_seek($this->Query_ID,$pos);
        $this->Row = $pos;
    }

    function metadata($table) {
        $count = 0;
        $id = 0;
    }

```

```

$res = array();

$this->connect();
$id = mssql_query("select * from $table", $this->Link_ID);
if (!$id) {
    $this->Errno = 1;
    $this->Error = "General Error (The MSSQL interface cannot return detailed error messages).";
    $this->halt("Metadata query failed.");
}
$count = mssql_num_fields($id);

for ($i=0; $i<$count; $i++) {
    $info = mssql_fetch_field($id, $i);
    $res[$i]["table"] = $table;
    $res[$i]["name"] = $info["name"];
    $res[$i]["len"] = $info["max_length"];
    $res[$i]["flags"] = $info["numeric"];
}
$this->free_result();
return $res;
}

function affected_rows() {
    return mssql_rows_affected($this->Link_ID);
}

function num_rows() {
    return mssql_num_rows($this->Query_ID);
}

function num_fields() {
    return mssql_num_fields($this->Query_ID);
}

function nf() {
    return $this->num_rows();
}

function np() {
    print $this->num_rows();
}

```

```
function f($Field_Name) {
    return $this->Record[strtolower($Field_Name)];
}

function p($Field_Name) {
    print $this->f($Field_Name);
}

function halt($msg) {
    printf("<b>Database error:</b> %s<br>\n", $msg);
    printf("<b>MSSQL Error</b>: %s (%s)<br>\n",
        $this->Errno,
        $this->Error);
    die("Session halted.");
}
}
```

## A2. Το αρχείο διαμόρφωσης (configuration file)

```
// Layout
$layoutouterlinecolor = "#000000";
$boxouterlinecolor = "#696969";

//Method 1
$unreg_weight = 0.2;
$reg_weight = 0.8;
$imdb_weight = 0.8;
$mrs_weight = 0.2;
$method1_quality_bound = 8;

//Method 2
$method2_quality_bound = 8;
$correlation_bound = 0.5; // O suntelesths susxetishs gia na 8ewrountai sxetikoi 2 users

//Method 3
$method3_quality_bound = 8.5;

//Buddies
$buddies_bound = 0.5; // O suntelesths susxetishs gia na 8ewrountai buddies 2 users

//Communities
$ccg_bound = 0.5 // The bound for the Community Coherence Grade
```

### A3. Συνάρτηση συντακτικής ανάλυσης σελίδων του IMDB (IMDB parser)

```

// This function implements the imdb parser . It parses a movie page and
// gathers all information needed by MRS system
function imdbMovie($title_number){

    $imdb_regs = array(
        "single" => array(
            "title" => "<h1><strong class='title'\>(.*<small>",
            "year" => "<a href='\"/Sections/Years/(.*)'\>.*</a>",
            "image" => "<img alt='\"cover\" align='\"left\" src='\"([^\"]+)\"\"",
            "rating" => "<b>(.*)/10</b> (.* votes)",
            "plot" => "Plot Summary:<b>(.*",
            "plotold" => "</b>(.*<a href='\"/Plot"
        ),
        "multi" => array (
            "country" => "Sections/Countries/([^\"]+)",
            "genre" => "Sections/Genres/([^\"]+)"
        ),
        "block" => array (
            "director" => array (
                "start" => "Directed By",
                "end" => "Writing credits"
            ),
            "cast" => array (
                "start" => "Cast overview",
                "end" => "\ (more)",
                "seperator" => "\n"
            ),
            "alt_title1" => array (
                "start" => "Also Known As:",
                "end" => "HREF='\"/mpaa\"",
                "seperator" => "\n"
            )
        )
    );

    $fd = fsockopen ("www.imdb.com", 80, $errno, $errstr, 30);

    if (!$fd){
        echo "error getting web page ( errno=$errno: $errstr)<br>\n";
    }
}

```



```

// read in the web page and specify "Opera/5.0" as our browser ( my favourite :_)
fputs ($fd, "GET /Title?${title_number} HTTP/1.0\r\nUser-Agent: Opera/5.0\r\n\r\n");
//fputs ($fd, "GET /Recent/Greece HTTP/1.0\r\nUser-Agent: Opera/5.0\r\n\r\n");
$content = fread($fd, 100000);
fclose ($fd);

// write the web page to a temporary file in the cover directory
$fd = fopen("imdb8.tmp", "w");
if ( !$fd ){
    echo "Error: cant write temp file!<br>";
    return;
}
fwrite($fd, $content);
fclose($fd);

$content = file ("imdb8.tmp");
//print_r($content);

// process the single line matches
foreach ( $content as $line ){
    foreach ( $imdb_regs{"single"} as $key => $pattern ){
        if ( ereg( $pattern, $line, $regs ) ){
            $movie{$key} = htmlspecialchars($regs[1]);
        }
    }
}

// process the multi line matches
// ( multiple matches are converted to a comma separated list )
foreach ( $content as $line ){
    foreach ( $imdb_regs{"multi"} as $key => $pattern ){
        if ( preg_match_all( "\|$pattern|i", $line, $matches ) ){
            $movie{$key} = join(", ", $matches[1]);
        }
    }
}

// process the block matches
// ( the line inbetween are separated by the specified separator )
$matches = array();
$flags = array();

```

```

foreach ( $contents as $line ){
    foreach ( $imdb_regs{"block"} as $key => $description ){
        // already past the end tag ?
        if ( !isset($flags{$key}) || $flags{$key} == false){
            // search for the start of the block
            $pattern = "|" . $description{"start"} . "|i";
            if ( preg_match( $pattern, $line ) ){
                $flags{$key} = true;
                $matches{$key} = "";
            }
        }
        else{
            // search for the end of the block
            $pattern = "|" . $description{"end"} . "|i";

            if ( preg_match( $pattern, $line ) ){
                $flags{$key} = false;
            }

            else{
                // we are inside the block
                $clean_line = trim( strip_tags($line) );
                $matches{$key} .= $clean_line . $description{"separator"};
            }
        }
    }
}

// the "-" is not allowed inside a mysql set name, so remove it
$movie{"genre"} = str_replace( "-", "", $movie{"genre"} );

// now fill the movie hash with the result from the block search
foreach ( $imdb_regs{"block"} as $key => $description ){
    $movie{$key} = $matches{$key};
}

return $movie;
}

```

#### A4. Υπολογισμός της βαθμολογίας MRS

```

// Υπολογίζει την MRS βαθμολογία μιας ταινίας
function mrs_rating($mid,$type,$imdb_rating){

```

```

switch($type){
  case "method1":
    global $unreg_weight;
    global $reg_weight;
    global $imdb_weight;
    global $mrs_weight;

    $unreg_mean = calculateUnregMean($mid);
    $reg_mean = calculateRegMean($mid);
    $mrs_mean = ($unreg_weight*$unreg_mean) +
($reg_weight*$reg_mean);
    $mrs_rating = round(($imdb_weight*$imdb_rating) +
($mrs_weight*$mrs_mean), 1);
    break;
  default:
    $mrs_rating=5;
    break;
}
return $mrs_rating;
}

```

### A5. Αλγόριθμος υλοποίησης πρώτης μεθόδου συστάσεων

```

// This function calculates the recommended movies according to the MRS ratings
function method1($login){
  global $method1_quality_bound;
  $mids = getMids($login);
  //print_r($mids);
  $w=0;
  for($i=0;$i<count($mids[0]);$i++){
    $mrs_rating = mrs_rating($mids[0][$i], "method1", $mids[1][$i]);
    if($mrs_rating >= $method1_quality_bound){
      $recommended[0][$w] = $mids[0][$i];
      $recommended[1][$w] = $mrs_rating;
      $w++;
    }
  }
  if($w == 0){
    return false;
  }
  else{

```

```

        array_multisort($recommended[1], SORT_NUMERIC, SORT_DESC, $recommended[0]);
        return $recommended;
    }
}

```

## A6. Υπολογισμός συντελεστή συσχέτισης Pearson (Pearson Correlation Coefficient)

```

// Υπολογίζει το Pearson Correlation Coefficient που δίνεται από τον τύπο:
//  $S(Ux - Uavg_x)(Uy - Uavg_y) / \sqrt{(S(Ux - Uavg_x)^2) * (S(Uy - Uavg_y)^2)}$ 
function PearsonCorrelation($vectorX,$vectorY){
    $sizeX = count($vectorX);
    $sizeY = count($vectorY);

    $sumX = array_sum($vectorX);
    $sumY = array_sum($vectorY);

    $averageX = $sumX/$sizeX;
    $averageY = $sumY/$sizeY;

    $arithmitis = 0;
    for($i=0;$i<$sizeX;$i++){
        $diaforaX = $vectorX[$i] - $averageX;
        $diaforaY = $vectorY[$i] - $averageY;
        $ginomeno = $diaforaX * $diaforaY;
        $arithmitis = $arithmitis + $ginomeno;
    }

    $sum1 = 0;
    for($i=0;$i<$sizeX;$i++){
        $diaforaX = $vectorX[$i] - $averageX;
        $tetragwono = pow($diaforaX,2);
        $sum1 = $sum1 + $tetragwono;
    }

    $sum2 = 0;
    for($i=0;$i<$sizeY;$i++){
        $diaforaY = $vectorY[$i] - $averageY;
        $tetragwono2 = pow($diaforaY,2);
        $sum2 = $sum2 + $tetragwono2;
    }
}

```

```

if($sum1 == 0){
    $sum1 = 1;
}
if($sum2 == 0){
    $sum2 = 1;
}
$paronomastis = sqrt($sum1*$sum2);

$correlation = $arithmitis/$paronomastis;

return $correlation;
}

```

### A7. Αλγόριθμος Υπολογισμού Συσχέτισης δύο Προφίλ με βάση τις βαθμολογήσεις τους

```

// Sygkrinei duo profiles bash tvn yhfwn pou exoun dwsei se koines tainies
function compareProfiles($login1,$login2){
    $db = CreateObject();
    $query = "select * from votes where mid in (select mid from votes where login='$login1' and mid in
(select mid from votes where login='$login2'))";
    $db->query($query);

    $i=0;
    $x=0;
    $y=0;
    while($db->next_record()){
        if($db->Record['login'] == $login1){
            $vectorX[$x] = $db->Record['vote'];
            $x++;
        }
        if($db->Record['login'] == $login2){
            $vectorY[$y] = $db->Record['vote'];
            $y++;
        }
        $i++;
    }
    if($i==0){// Kamia koinh tainia
        return false;
    }
    else{
        $correlation = PearsonCorrelation($vectorX,$vectorY);
    }
}

```

```

        return $correlation;
    }
}

```

### A8. Αλγόριθμος Υπολογισμού Συσχέτισης δύο Προφίλ με βάση τις προτιμήσεις τους σε συγκεκριμένες κατηγορίες

*// This function compares two profiles according to their genre preferences*

```

function compareProfiles2($login1,$login2){
    $db = CreateObject();

    //Calculate VectorX
    $query = "select * from userprofiler where login='$login1'";
    $db->query($query);
    $db->next_record();
    $x = 0;
    for($w=0;$w<count($db->Record)/2;$w++){
        $VectorX[$w] = $db->Record[$w];
    }
    array_shift($VectorX);

    //Calculate VectorY for each user
    $query = "select * from userprofiler where login='$login2'";
    $db->query($query);
    $db->next_record();
    $y = 0;
    for($w=0;$w<count($db->Record)/2;$w++){
        $VectorY[$w] = $db->Record[$w];
    }
    array_shift($VectorY);

    //Calculate correlation of VectorX and VectorY
    $correlation = PearsonCorrelation($VectorX,$VectorY);

    return $correlation;
}

```

### A9. Αλγόριθμος εύρεσης φίλων με βάση τις προτιμήσεις τους σε κατηγορίες ταινιών

*// This function will calculate the buddies of the user s\_login according to genre preferences*

```

function findBuddies($s_login){

```

```

global $buddies_bound;
$db = CreateObject();
$query = "select login from userprofiler where login!='$s_login'";
$db->query($query);
$i = 0;
while($db->next_record()){
    $users[$i] = $db->Record['login'];
    $i++;
}
if($i == 0){
    return false;
}
else{
    $k=0;
    for($i=0;$i<count($users);$i++){
        $correlation = compareProfiles2($s_login,$users[$i]);
        if($correlation>$buddies_bound){
            $buddies[$k] = $users[$i];
            $k++;
        }
    }
    if($k==0){
        return false;
    }
    else{
        return $buddies;
    }
}
}

```

### A10. Αλγόριθμος υπολογισμού δεύτερης μεθόδου συστάσεων (Collaborative Filtering)

```

// This function implements the collaborative filtering algorithm fo user s_login
function method2($s_login,$consultants){
    global $method2_quality_bound;

    $db = CreateObject();
    $query = "select mid from movie where mid not in(select mid from votes where login='$s_login')";
    $db->query($query);

    $i = 0;

```

```

while($db->next_record()){
    $movies[$i] = $db->Record['mid'];
    $i++;
}

if($i == 0){
    //Exw yhfisei oles tis tainies sthn bash
}
else{
    $r = 0;
    for($i=0;$i<count($movies);$i++){
        $m = 0;
        $sum=0;
        $counter=0;
        for($w=0;$w<count($consultants);$w++){
            $query = "select vote from votes where mid=$movies[$i] and
login='$consultants[$w]'";

            $db->query($query);
            $ar = $db->affected_rows();
            if($ar == 0){
                //O consultant $consultants[$w] den exei yhfisei thn tainia
                $movies[$i]
            }
            else{
                $db->next_record();
                $vote = $db->Record['vote'];
                $votes[$m] = $vote;
                $m++;
            }
        }
        if($m == 0){
            // Kanenas apo tous consultants den exei yhfisei authn thn tainia.
        }
        else{
            $counter = count($votes);
            $sum = array_sum($votes);
            $average = $sum/$counter;

            if($average >= $method2_quality_bound){
                $recommended[0][$r] = $movies[$i];
                $recommended[1][$r] = $average;
                $r++;
            }
        }
    }
}

```



```

    }
    else{
        // O mesos oros tw n yhfwn tw n consultants gia authn thn tainia
        einai katw apo to quality bound
    }
}
unset($votes);
}

if($r==0){ // None movie satisfies the criteria
    return false;
}
else{ // Dwse ton pinaka recommended
    array_multisort($recommended[1], SORT_NUMERIC, SORT_DESC,
$recommended[0]);
    return $recommended;
}
}
}
}

```

### A11. Αλγόριθμος υλοποίησης της τρίτης μεθόδου συστάσεων

```

/* Implicit Rating me bash ta genres
1. Bres tis tainies pou den exei yhfisei
2. Gia ka8e tetoia tainia bres ta genres ths
3. Apo to userProfileer bres to suntelesth tou user gia auta ta genres
4. An o suntelesths einai megaluteros apo to quality_bound proteine thn alliws oxi
5. Zana sto bhma 2
*/
function method3($s_login){
    global $method3_quality_bound;

    $db = CreateObject();
    $query = "select mid from movie where mid not in(select mid from votes where login='$s_login')";
    $db->query($query);

    $i = 0;
    while($db->next_record()){
        $movies[$i] = $db->Record['mid'];
        $i++;
    }
}

```

```

if($i == 0){
    //Exw yhfisei oles tis tainies sthn bash
}
else{
    $r = 0;
    for($i=0;$i<count($movies);$i++){
        $query = "select gname from genre where mid=$movies[$i]";
        $db->query($query);
        $w=0;
        while($db->next_record()){
            $categories[$w] = $db->Record['gname'];
            $w++;
        }
        if($w == 0){
            // H tainia den anhkei se kamia category
        }
        else{
            // Pare tis Pos times twn genres ths tainias gia auto to user profile
            $string = "";
            for($k=0;$k<count($categories);$k++){
                $string = $string.$categories[$k]."Pos,";
            }
            $string = substr($string,0,-1);
            $query = "select $string from userprofiler where login='$s_login'";
            $db->query($query);
            $db->next_record();
            for($q=0;$q<count($db->Record)/2;$q++){
                $genre_Pos[$q] = $db->Record[$q];
            }

            // Pare tis Neg times twn genres ths tainias gia auto to user profile
            $string = "";
            for($k=0;$k<count($categories);$k++){
                $string = $string.$categories[$k]."Neg,";
            }
            $string = substr($string,0,-1);
            $query = "select $string from userprofiler where login='$s_login'";
            $db->query($query);
            $db->next_record();
            for($q=0;$q<count($db->Record)/2;$q++){
                $genre_Neg[$q] = $db->Record[$q];
            }
        }
    }
}

```

```

        $sum = 0;
        for($w=0;$w<count($genre_Pos);$w++){
            $sum = $genre_Pos[$w] + $genre_Neg[$w];
            if($sum == 0){
                $Pos_percentage = 0;
                $Neg_percentage = 0;
            }
            else{
                $Pos_percentage =
round($genre_Pos[$w]/$sum,2)*10;
                $Neg_percentage =
round($genre_Neg[$w]/$sum,2)*10;
            }
            // $likelihood[$w] = $Pos_percentage - $Neg_percentage;
            $likelihood[$w] = $Pos_percentage ;
        }
        $counter = count($likelihood);
        $sum = array_sum($likelihood);
        $coefficient = round($sum/$counter,2);
        //echo("<br>coefficient$movies[$i]: ".$coefficient);
        if($coefficient >= $method3_quality_bound){
            $recommended[0][$r] = $movies[$i];
            $recommended[1][$r] = $coefficient;
            $r++;
        }
        else{
            // Movie Not Recommended
        }
        unset($likelihood);
        unset($categories);
        unset($genre_Pos);
        unset($genre_Neg);
    }
}

if($r==0){
    // None movie satisfies the criteria
    return false;
}
else{
    // Dwse ton pinaka recommended

```

```

        array_multisort($recommended[1], SORT_NUMERIC, SORT_DESC,
    $recommended[0]);
        return $recommended;
    }
}
}

```

### A12. Αλγόριθμος που επιστρέφει το ποσοστό της θετικής προτίμησης ενός χρήστη σε μία κατηγορία ταινίας

```

// Input: ena apo ta pedia tou userprofiler
// Output: to pososto tou pediou autou
function getPercentage($genre,$s_login){

    $Positive = $genre."Pos";
    $Negative = $genre."Neg";
    $db = CreateObject();
    $query = "select $Positive,$Negative from userprofiler where login='$s_login'";
    $db->query($query);
    $db->next_record();
    $pos_ = $db->Record[0];
    $neg_ = $db->Record[1];
    $sum = $pos_ + $neg_ ;
    if($sum == 0){
        $pos_percentage = 0;
        $neg_percentage = 0;
    }
    else{
        $pos_percentage = round($pos_/$sum,2);
        $neg_percentage = round($neg_/$sum,2);
    }
    return $pos_percentage;
}
}

```

### A13. Αλγόριθμος που υπολογίζει εάν ένας χρήστης ικανοποιεί τα κριτήρια του φιλτραρίσματος ή όχι

```

// This function decides whether a user satisfies criteria
// specified by preferences
function satisfiesCriteria($s_login,$preferences){
    for($i=0;$i<count($preferences);$i++){
        $parts = explode(".", $preferences[$i]);
    }
}

```

```
$genre = $parts[0];
$criteria = $parts[1];
$percentage = getPercentage($genre,$s_login);

switch($criteria){
    case 5: //80%-100%
        if($percentage < 0.8){
            return false;
        }
        break;
    case 4: //60%-80%
        if($percentage < 0.6 || $percentage > 0.8){
            return false;
        }
        break;
    case 3: //40%-60%
        if($percentage < 0.4 || $percentage > 0.6){
            return false;
        }
        break;
    case 2: //20%-40%
        if($percentage < 0.2 || $percentage > 0.4){
            return false;
        }
        break;
    case 1: //0%-20%
        if($percentage > 0.2){
            return false;
        }
        break;
    default://other
        return false;
        break;
}
}
return true;
}
```

#### A14. Αλγόριθμος Υπολογισμού Πρόβλεψης ψήφου

*// O algorithmos autos problepei to βαθμο που θα ebaze o xrhsths sthn tainia*

```
function calculatePrediction($id,$s_login){
```

```

$db = CreateObject();
$query = "select count(vote),sum(vote) from votes where login='$s_login'";
$db->query($query);
$ar = $db->affected_rows();
if($ar == 0){
    return false;
}
else{
    $db->next_record();
    $total_votes = $db->Record[0];
    $sum = $db->Record[1];
    $user_average = round($sum/$total_votes,2);
    //echo("<br>user_average: ".$user_average);
    $query = "select login,vote from votes where mid=$id and login not in (select login from
votes where
        login='$s_login')";
    $db->query($query);
    $ar = $db->affected_rows();
    if($ar == 0){
        return false;
    }
    else{
        $i = 0;
        while($db->next_record()){
            $logins[$i] = $db->Record['login'];
            $votes[$i] = $db->Record['vote'];
            $weight[$i] = compareProfiles($s_login,$db->Record['login']);
            $i++;
        }

        // Calculate averages for each login
        for($i=0;$i<count($logins);$i++){
            $query = "select count(vote),sum(vote) from votes where
login='$logins[$i]'";

            $db->query($query);
            $ar = $db->affected_rows();
            if($ar == 0){
                return false;
            }
            else{
                $db->next_record();

```

```

        $total_votes = $db->Record[0];
        $sum = $db->Record[1];
        $average[$i] = round($sum/$total_votes,2);
    }
}
$arithmitis = 0;
for($i=0;$i<count($logins);$i++){
    $arithmitis = $arithmitis + $weight[$i]*($votes[$i] - $average[$i]);
    $sparonomastis = $sparonomastis + abs($weight[$i]);
}

if($sparonomastis==0){
    return false;
}

$variation = $arithmitis/$sparonomastis;
$prediction = round($user_average + $variation,0);
return $prediction;
}
}
}

```

### A15. Υπολογισμός της δραστηριότητας του χρήστη (κατανομή των ψήφων)

```

// This function will return information about the vote activation
// of a user in order to show its activity
function votesAllocation($s_login){
    $db = CreateObject();
    $query = "select vote from votes where login='$s_login'";
    $db->query($query);

    $i = 0;
    while($db->next_record()){
        $votes[$i] = $db->Record['vote'];
        $i++;
    }
    if($i==0){
        $error = "You haven't yet rate movies. Your activity cannot be represented graphically.<br>Please spend some time to rate some movies. It is interesting! <br><br><a href=\"/web/rs/rating/\" class=\"link\">I want to rate some now</a>";
        return false;
    }
}

```

```

else{
    $total_votes = count($votes);
    $sum = array_sum($votes);
    $average = round($sum/$total_votes,1);
    for($i=0;$i<10;$i++){
        $k=$i+1;
        $sanstext_table[$i] = "$k";
    }
    $graph = "bars";
    $votes_allocation = array_count_values($votes);
    for($i=0,$k=0;$i<10;$i++){
        $k = $k+1;
        if($votes_allocation[$k]){
            $votes_table[$i] = $votes_allocation[$k];
        }
        else{
            $votes_table[$i] = 0;
        }
    }
    for($i=0;$i<10;$i++){
        $k = $i+1;
    }
    $imgname = createImage($sanstext_table,$votes_table,$graph,"large");

    $info = array("imgname"=>$imgname, "average"=>$average, "total_votes"=>$total_votes,
"votes_table"=>$votes_table);
    return $info;
}
}

```



## Πίνακας Διαγραμμάτων

---

Χρήση της php μέχρι τον Ιούλιο του 2002	10
Η ανάπτυξη της php ως module του apache	10
Η δημοτικότητα και ο ρυθμός ανάπτυξης των γλωσσών ανάπτυξης διαδικτυακών εφαρμογών	12
Αρχιτεκτονική του Client/Server Μοντέλου	13
Ενσωμάτωση Βάσης Δεδομένων στο Client/Server Μοντέλο	15
Διάγραμμα Οντοτήτων-Σχέσεων του MRS	21
Το σχήμα της βάσης δεδομένων του MRS συστήματος	23
Το σχήμα της βάσης δεδομένων του MRS συστήματος και οι τύποι των γνωρισμάτων	23

---

## Βιβλιογραφία

---

### Βιβλία

- [1] *Introduction to Database Management Systems*, C.J. Date, 6<sup>th</sup> american edition.
- [2] *Database System Concepts*, Silberschatz, Korth & Sudarshan, 3rd edition, McGraw-Hill, 1997.
- [3] *Fundamentals of Database Systems*, Elmasri & Navathe, 2nd edition, Benjamin-Cummings, 1994.
- [4] *Machine Learning*. T.Mitchell. McGraw-Hill, New York, NY, 1997

### Δημοσιεύσεις

- [SKKR] *Item-Based Collaborative Filtering Recommendation Algorithms*. Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. GroupLens Research Group/Army HPC Research Center. Department of Computer Science and Engineering University of Minnesota, Minneapolis, MN 55455.
- [RACLMKR] *Getting to Know You: Learning New User Preferences in Recommender Systems*. Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K. Lam, Sean M. McNee, Joseph A. Konstan, John Riedl. GroupLens Research Project. Department of Computer Science and Engineering, University of Minnesota Minneapolis, MN 55455 USA.
- [SM] *Social Information Filtering: Algorithms for Automating "Word of Mouth"*. Upendra Shardanand, Pattie Maes. MIT Media-Lab, Cambridge, MA 02139.
- [GLGBG] *Recommending Web Documents Based on User Preferences*. Eric J. Glover, Steve Lawrence, Michael D. Gordon, William P. Birmingham, C. Lee Giles. NEC Research Institute, Princeton, Artificial Intelligence

Laboratory, University of Michigan, Business Administration, University of Michigan.

[PUPL] *Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments*. Alexandrin Popescul and Lyle H. Ungar, Department of Computer and Information, Science University of Pennsylvania Philadelphia, PA 19104, David M. Pennock and Steve Lawrence, NEC Research Institute, Princeton, NJ 08540. In Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI-2001), pp. 437-444, Morgan Kaufmann, San Francisco, 2001.

[FBH] *Mining Navigation History for Recommendation*. Xiaobin Fu, Jay Budzik, Kristian J. Hammond. Infolab, Northwestern University, Evanston, IL 60201.

[MMNa] *Content-Boosted Collaborative Filtering*. Prem Melville, Raymond J. Mooney and Ramadass Nagarajan. Department of Computer Sciences, University of Texas Austin, TX 78712.

[AAM] *Compositional Recommender Systems Using Case-Based Reasoning Approach*. Stefano Aguzzoli, Paolo Avesani and Paolo Massa, ITC-irst - 38050 Povo, Italy.

[PHLG] *Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach*. David M. Pennock, NEC Research Institute, Princeton, NJ 08540, Eric Horvitz, Microsoft Research, Redmond, WA 98052-6399, Steve Lawrence and C. Lee Giles, NEC Research Institute, Princeton, NJ 08540. In Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI-2000), pp. 473-480, Morgan Kaufmann, San Francisco, 2000.

[MMNb] *Content-Boosted Collaborative Filtering for Improved Recommendations*. Prem Melville and Raymond J. Mooney and Ramadass Nagarajan. Department of Computer Sciences, University of Texas, Austin, TX 78712. To Appear in Proceedings of the Eighteenth

National Conference on Artificial Intelligence (AAAI-Edmonton, Canada, July 2002.

[PP] *The Role of Semantic Relevance in Dynamic User Community Management and the Formulation of Recommendations.* Nick Papadopoulos, Institute of Computer Science, Foundation for Research and Technology - Hellas, Dimitris Plexousakis, Department of Computer Science, University of Crete, Heraklion, GREECE.

[SS] *Beyond Algorithms: An HCI Perspective on Recommender Systems.* Kirsten Swearingen & Rashmi Sinha. SIMS, UC Berkeley, 94720.

### **Διαδικτυακές Πηγές**

[1] *Internet Movie Database,*  
<http://www.imdb.com>

[2] *EachMovie,*  
<http://www.research.compaq.com/SRC/eachmovie/>

[3] *Tutorial on Recommender Systems,*  
<http://www.scs.carleton.ca/~weiss/courses/541/slides/RecommenderSystems.html>

[4] *GroupLens Research,*  
<http://www.cs.umn.edu/Research/GroupLens/research.html>

[5] *Rating Zone,*  
<http://www.ratingzone.com/>

[6] Computing Pearson's correlation coefficient,  
<http://www.ruf.rice.edu/~lane/hyperstat/A51911.html>

[7] Movie Collector,  
<http://www.olaf-stauffer.de/projects/MovieCollector/>