



## Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών - Τμήμα Μηχανικών  
Πληροφορικής

### Πτυχιακή εργασία

Ανάπτυξη και υλοποίηση εφαρμογών παγκόσμιου ιστού και κινητών τηλεφώνων  
για σύσταση πληροφορίας

Μιχαλάκης Νικόλαος 3765



Επιβλέπων καθηγητής: Φραγκοπούλου Παρασκευή

Συνεπιβλέπων καθηγητής: Παπαδάκης Χαράλαμπος

ΗΡΑΚΛΕΙΟ

2017



## **Abstract**

Data recommendation systems are now an integral part of several applications, making suggestions to users about objects or services that they are likely to be interested in. The purpose of this Bachelor's thesis is the development of a Web & Android application that makes movie suggestions to the users according to their preferences. Chapter 2 includes a brief explanation of the recommender systems, analyzing their basic methods with some known examples of their use. The algorithm used in this thesis is also presented. Chapter 3 gives the details of the technologies used for the development of the software. Chapter 4 describes the data collection procedure, which was necessary for the functionality of both the algorithm and the applications. To conclude, the Web and Android movie recommendation applications are presented in Chapters 5 and 6 respectively.

## Σύνοψη

Τα συστήματα σύστασης πληροφορίας είναι πλέον βασικό κομμάτι πολλών εφαρμογών με σκοπό να προτείνουν στους χρήστες αντικείμενα που πιθανόν θα τους ενδιαφέρουν. Το θέμα της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη μιας Web & Android εφαρμογής για σύσταση ταινιών στους χρήστες σύμφωνα με τις προτιμήσεις τους. Στο κεφάλαιο 2 γίνεται μια εισαγωγή στα συστήματα σύστασης πληροφορίας αναλύοντας τις βασικές μεθόδους τους, ενώ στην συνέχεια γίνεται μια μικρή περιγραφή κάποιων γνωστών εφαρμογών τους. Επιπλέον γίνεται παρουσίαση του αλγορίθμου που χρησιμοποιήθηκε στην παρούσα πτυχιακή. Στο κεφαλαίο 3 αναφέρονται οι τεχνολογίες που χρησιμοποιήθηκαν για την εκπόνηση της. Στο κεφαλαίο 4 περιγράφεται ο τρόπος με τον οποίο έγινε η συλλογή δεδομένων, τα οποία ήταν απαραίτητα τόσο για την λειτουργία του αλγορίθμου όσο και των εφαρμογών. Τέλος, στα κεφάλαια 5 & 6 παρουσιάζονται η Web και η Android εφαρμογές σύστασης ταινιών αντίστοιχα.

# Περιεχόμενα

Abstract.....	3
Σύνοψη.....	4
Περιεχόμενα.....	5
Λίστα εικόνων.....	7
Λίστα πινάκων.....	9
<b>Κεφάλαιο 1 Εισαγωγή</b>	
1.1 Περίληψη.....	10
1.2 Κίνητρο για την διεξαγωγή της εργασίας.....	10
1.3 Δομή εργασίας.....	10
<b>Κεφάλαιο 2 Recommender Systems</b>	
2.1 Τεχνικές Personalized Recommendations.....	12
2.1.1 Collaborative Filtering.....	12
2.1.2 Content-based Filtering.....	13
2.1.3 Knowledge-based.....	14
2.1.4 Context-based Recommender Systems.....	15
2.1.4.1 Time Sensitive Recommender Systems.....	15
2.1.4.2 Location-Based Recommender Systems.....	15
2.1.5 Demographic-Based.....	15
2.1.6 Hybrid Recommender Systems.....	16
2.2 Real world examples of RS.....	16
2.2.1 GroupLens Recommender System.....	16
2.2.2 Amazon.com Recommender System.....	17
2.2.3 Netflix Movie Recommender System.....	18
2.2.4 Facebook Friend Recommendations.....	18
2.3 Impact of Recommender Systems.....	19
2.4 O SCoR.....	19
2.4.1 Ο αλγόριθμος Vivaldi.....	19
2.4.2 Περιγραφή του ScoR αλγορίθμου.....	20
<b>Κεφάλαιο 3 Οι τεχνολογίες που χρησιμοποιήθηκαν</b>	
3.1 Java.....	22
3.2 JavaServer Pages (JSP).....	22
3.3 Android.....	23
3.4 SQL.....	24
3.4.1 MySQL.....	24
3.5 Jsoup.....	24
<b>Κεφάλαιο 4 Συλλογή δεδομένων</b>	
4.1 Περιγραφή των φάσεων συλλογής δεδομένων.....	25
4.2 Τα δεδομένα που συλλέχθηκαν.....	27
<b>Κεφάλαιο 5 Web Εφαρμογή</b>	
5.1 Περιγραφή.....	28
5.2 Τα μέρη της εφαρμογής.....	28
5.2.1 Home Page.....	28
5.2.2 Response Page.....	30
5.3 Λειτουργικότητα της εφαρμογής.....	30

<b>Κεφάλαιο 6 Android Εφαρμογή</b>	
6.1 Περιγραφή.....	34
6.2 User Interface.....	34
6.2.1 Οθόνες πλοήγησης.....	34
6.2.2 Οθόνες παρουσίασης πληροφορίας.....	35
6.3 Επικοινωνία με την βάση.....	38
6.4 Βασικές λειτουργίες της εφαρμογής.....	40
6.4.1 Βαθμολόγηση ταινιών.....	40
6.4.2 Διαχείριση βαθμολογιών.....	42
6.4.3 Σύσταση πληροφορίας.....	42
6.4.4 Watchlist.....	44
6.4.5 Settings.....	44
<b>Κεφάλαιο 7 Αποτελέσματα</b>	
7.1 Μελλοντική εργασία και επεκτάσεις.....	46
<b>Βιβλιογραφία.....</b>	<b>47</b>

## Λίστα εικόνων

### Κεφάλαιο 2 Recommender Systems

Εικόνα 1 - Η χρησιμότητα των RS.....	11
Εικόνα 2 - Collaborative Filtering Example.....	12
Εικόνα 3 – Content-based Filtering Example.....	13
Εικόνα 4 – Knowledge-based.....	14
Εικόνα 5 – Παράδειγμα δημογραφικών στοιχείων.....	15
Εικόνα 6 – Hybrid Recommendation.....	16
Εικόνα 7 – Τρόποι σύστασης πληροφορίας.....	17
Εικόνα 8 – Amazon Recommendation.....	17
Εικόνα 9 – Netflix Recommendations.....	18
Εικόνα 10 – Ψευδοκώδικας του αλγορίθμου ScoR.....	21

### Κεφάλαιο 3 Οι τεχνολογίες που χρησιμοποιήθηκαν

Εικόνα	11	–
Java.....	22	
Εικόνα 12 – JSP Αρχιτεκτονική.....	22	
Εικόνα 13 – Android logo.....	23	

### Κεφάλαιο 4 Συλλογή δεδομένων

Εικόνα 14 – Σχεδιάγραμμα πρώτης φάσης.....	26
--	----

### Κεφάλαιο 5 Web Εφαρμογή

Εικόνα 15 – Home Page form.....	29
Εικόνα 16 – Home Page.....	29
Εικόνα 17 – Response Page.....	30
Εικόνα 18 – Filter.....	30
Εικόνα 19 – Home Page Form.....	30
Εικόνα 20 – Recommendation για συγκεκριμένη ταινία.....	31
Εικόνα 21 – Γενικό Recommendation.....	31
Εικόνα 22 – Filtered movies.....	32
Εικόνα 23 – Cleared Filter.....	33

### Κεφάλαιο 6 Android Εφαρμογή

Εικόνα 24 – Ιεραρχία Layout οθονών πλοήγησης.....	34
Εικόνα 25 – Οθόνες πλοήγησης.....	35
Εικόνα 26 – Κάθετη παρουσίαση πληροφορίας.....	35
Εικόνα 27 – Πλήρης οθόνη παρουσίασης πληροφορίας.....	35
Εικόνα 28 - Ιεραρχία Layout κάθετης παρουσίασης πληροφορίας.....	36
Εικόνα 29 - Movie layout.....	36
Εικόνα 30 - Fragment Layout.....	37
Εικόνα 31 - Flowchart εφαρμογής.....	38
Εικόνα 32 – βαθμολόγηση ταινιών.....	38
Εικόνα 33 - βαθμολόγηση συγκεκριμένης ταινίας.....	39
Εικόνα 34 - Recommendation.....	39
Εικόνα 35 - Recommendation για συγκεκριμένη ταινία.....	39
Εικόνα 36 – Import.....	40
Εικόνα 37 - Rate.....	41
Εικόνα 38 – Search .....	41
Εικόνα 39 – RateSpecific.....	41
Εικόνα 40 - Διαχείριση βαθμολογιών.....	42

Εικόνα 41 - οθόνη Recommendation.....	43
Εικόνα 42 - Οθόνη Filter.....	43
Εικόνα 43 - SpecificRecommendation (βήμα 1).....	44
Εικόνα 44 - SpecificRecommendation (βήμα 2).....	44
Εικόνα 45 – Watchlist.....	45
Εικόνα 46 – Settings.....	45
Εικόνα 47 – About.....	45



## **Λίστα πινάκων**

Πίνακας 1 – Σύγκριση RS.....	14
Πίνακας 2 – Android Versions.....	23
Πίνακας 3 - Συντομεύσεις ονομάτων οθονών.....	38

# 1 Εισαγωγή

## 1.1 Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας ήταν η δημιουργία web & android εφαρμογής για σύσταση ταινιών στους χρήστες σύμφωνα με τις προτιμήσεις τους (Personalized Recommendation). Λόγω του ότι ο αλγόριθμος σύστασης πληροφορίας χρειαζόταν βαθμολογίες χρηστών σε ταινίες, ενώ οι δύο εφαρμογές χρειάζονταν ταινίες και πληροφορίες γι' αυτές, αρχικά έγινε η συλλογή της απαιτούμενης πληροφορίας ενώ στην συνέχεια αναπτύχθηκαν οι δύο εφαρμογές. Η web εφαρμογή παρέχει την δυνατότητα στον χρήστη δίνοντας το ID του προφίλ του το οποίο αντιστοιχεί σε ένα url με βαθμολογίες του σε ταινίες, να λάβει recommendation. Ενώ η Android εφαρμογή παρέχει ένα μεγαλύτερο αριθμό «δραστηριοτήτων» στον χρήστη. Είναι τελείως ανεξάρτητη από εξωτερικές πηγές αφού δίνεται η δυνατότητα στο χρήστη να βαθμολογήσει ταινίες αλλά και να διαχειριστεί τις βαθμολογίες του εντός αυτής. Εντός της εφαρμογής ο χρήστης μπορεί:

- Να βαθμολογήσει ταινίες (είτε τυχαίες ταινίες είτε της επιλογής του)
- Να κάνει Import βαθμολογίες του από εξωτερική πηγή
- Να διαχειριστεί τις υπάρχουσες βαθμολογίες του (Διαγραφή ή αλλαγή μίας βαθμολογίας)
- Να λάβει recommendation για ταινίες (Είτε ταινίες που του προτείνει ο αλγόριθμος είτε τις επιλογής του)

Επίσης υπάρχουν και οι λειτουργίες:

- Watchlist (όπου αποθηκεύονται οι ταινίες που ο χρήστης ενδιαφέρεται να παρακολουθήσει στο μέλλον)
- Settings (Data Saving ώστε να μην κατεβαίνουν τα poster των εικόνων για εξοικονόμηση δεδομένων και δυνατότητα επιλογής της πηγής των βαθμολογιών για την σύσταση πληροφορίας)

## 1.2 Κίνητρο για την Διεξαγωγή της Εργασίας

Πολλοί άνθρωποι αναλώνουν αρκετό χρόνο έως ότου καταλήξουν ποια ταινία θα παρακολουθήσουν. Λόγω του ότι πλέον όλα κινούνται γύρω μας με γρήγορους ρυθμούς, αρκετοί άνθρωποι δεν έχουν την πολυτέλεια να σπαταλήσουν μισή ή μια ώρα στο να αναζητήσουν την ταινία που θα παρακολουθήσουν και επιπλέον 1-2 ώρες για να την δουν. Προτείνοντας τους όμως ταινίες που πιθανών θα τους αρέσουν μειώνεται ο χρόνος της αναζήτησης αφήνοντας τους περισσότερο ελεύθερο χρόνο για τις υπόλοιπες δραστηριότητες τους. Αυτοί οι παράγοντες σε συνδυασμό με ότι το θέμα της πτυχιακής είναι επίκαιρο άλλα και ενδιαφέρον ταυτόχρονα με οδήγησαν στην επιλογή της.

## 1.3 Δομή Εργασίας

Στο **κεφάλαιο 2** γίνεται μια εισαγωγή στα συστήματα σύστασης πληροφορίας αναλύοντας τις βασικές μεθόδους τους, ενώ στην συνέχεια γίνεται μια μικρή περιγραφή κάποιων γνωστών εφαρμογών τους. Επιπλέον γίνεται παρουσίαση του αλγορίθμου που χρησιμοποιήθηκε στην παρούσα πτυχιακή. Στο **κεφάλαιο 3** γίνεται μια μικρή αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν για την εκπόνηση της. Στο **κεφάλαιο 4** περιγράφεται ο τρόπος συλλογής της πληροφορίας με την χρήση ψευδοκώδικα και κειμένου. Στο **κεφάλαιο 5** γίνεται παρουσίαση της Web εφαρμογής. Στο **κεφάλαιο 6** γίνεται παρουσίαση της Android εφαρμογής με χρήση διαγραμμάτων αλλά και Screenshots.

## 2 Recommender Systems

Τα Recommender Systems (RS) ή συστήματα σύστασης πληροφορίας χρησιμοποιούνται για να προβλέψουν την βαθμολογία ή την προτίμηση ενός χρήστη σε συγκεκριμένα αντικείμενα ώστε να του προτείνουν αυτά που πιθανόν θα τον ενδιαφέρουν. Τα τελευταία χρόνια τα συστήματα αυτά είναι αρκετά δημοφιλή και χρησιμοποιούνται κυρίως σε εφαρμογές:

- Για διασκέδαση, όπου το περιεχόμενο που προτείνεται στους χρήστες μπορεί να είναι ταινίες, μουσική, βιβλία κλπ
- Για σύσταση online περιεχόμενου, όπου οι χρήστες δέχονται προτάσεις για ιστοσελίδες, posts κοινωνικών δικτύων, επιλογή ενός μαθήματος e-learning κλπ.
- Ηλεκτρονικού εμπορίου, όπου οι εταιρίες χρησιμοποιώντας τα RS μειώνουν τον χρόνο αναζήτησης για τους χρήστες, με αποτέλεσμα η εμπειρία χρήσης να βελτιώνεται οδηγώντας έτσι σε αύξηση των πωλήσεων.
- Σύσταση υπηρεσιών, όπου οι χρήστες δέχονται προτάσεις για προγράμματα ταξιδίων, σπιτιών προς ενοικίαση, υπηρεσίες υγείας κλπ.

### Why a Recommender System?



Εικόνα 1 - Η χρησιμότητα των RS

Υπάρχουν 2 τρόποι για σύσταση πληροφορίας, personalized & non-personalized recommendations. Στα personalized recommendations οι προτάσεις γίνονται στον χρήστη λαμβάνοντας υπόψιν δικά του actions (πχ αξιολογήσεις, επισκέψεις, τι του αρέσει και τι όχι) πολλές φορές σε συνδυασμό με βάση αντίστοιχα actions άλλων χρηστών. Ενώ στα non-personalized recommendations δεν λαμβάνονται υπόψιν οι προτιμήσεις του χρήστη αλλά προτείνονται κάποια δημοφιλή αντικείμενα ή επιλέγονται με τυχαίο τρόπο.

Ας δούμε μέσω ενός παραδείγματος τι είναι personalized και τι non-personalized. Έστω ότι έχουμε μια εφαρμογή για σύσταση πληροφορίας σε βιβλία. Στην περίπτωση των personalized ο αλγόριθμος θα μπορούσε να χρησιμοποιήσει πληροφορία όπως, αξιολογήσεις του χρήστη σε βιβλία, τι αναζητήσεις κάνει συνήθως (πχ για κλασικά, επιστημονικής φαντασίας κλπ), αν έχει προτίμηση σε κάποιον συγκεκριμένο συγγραφέα κλπ, ώστε να καταλήξει στα βιβλία που θα του προτείνει. Σε non-personalized recommendation μία προσέγγιση στο πρόβλημα θα ήταν απλά να του προτείνει τα δημοφιλέστερα βιβλία όλων των εποχών ή βιβλία που έχουν καλές κριτικές, με το σκεπτικό ότι ένα βιβλίο που αρέσει σε πολλούς έχει μεγαλύτερη πιθανότητα να αρέσει στον χρήστη από ένα άλλο τυχαία επιλεγμένο.

## 2.1 Τεχνικές Personalized Recommendation

Σε αυτήν την ενότητα γίνεται μια μικρή περιγραφή των ακόλουθων τεχνικών σύστασης πληροφορίας, του τρόπου με τον οποίο προτείνουν περιεχόμενο στους χρήστες αλλά και τα πιθανά πλεονεκτήματα/μειονεκτήματα μερικών εξ αυτών.

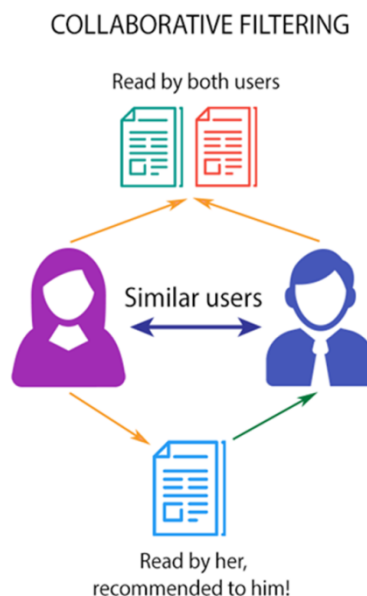
- Collaborative Filtering
- Content-based
- Knowledge-based
- Context-based (Time Sensitive & Location-Based)
- Demographic-based
- Hybrid (Συνδυασμός 2 ή περισσότερων από τις προαναφερθείσες τεχνικές)

### 2.1.1 Collaborative Filtering (CF)

Το Collaborative filtering είναι η πιο ευρέως χρησιμοποιούμενη μέθοδος για recommendations. Σε τέτοιους αλγόριθμους ο χρήστης αξιολογεί κάποια αντικείμενα και στην συνέχεια ο αλγόριθμος του προτείνει αντικείμενα που αρέσουν σε άλλους χρήστες με παρόμοιες βαθμολογίες. Ο όρος “Collaborative filtering” αναφέρεται στην χρήση των βαθμολογιών από πολλούς χρήστες για την πρόβλεψη των βαθμολογιών ενός άλλου χρήστη.

Οι Collaborative filtering αλγόριθμοι αντιμετωπίζουν τα εξής μειονεκτήματα:

- cold-start problem: το πώς θα προτείνεις περιεχόμενο σε νέους χρήστες έχοντας ελάχιστη ή καθόλου γνώση για τις προτιμήσεις τους.
- first-rater problem: το πώς θα προτείνεις ένα νέο αντικείμενο μη έχοντας αρκετές βαθμολογήσεις για αυτό. Ο αλγόριθμος για να μπορέσει να προτείνει ένα νέο αντικείμενο θα πρέπει πρώτα αυτό να βαθμολογηθεί από ένα μεγάλο αριθμό χρηστών.
- The “Gray Sheep” problem: αναφέρεται στους χρήστες των οποίων οι προτιμήσεις δεν εντάσσονται σε κάποια από της υπόλοιπες ομάδες χρηστών οπότε είναι δύσκολο ή ακατόρθωτο το σύστημα να πάρει απόφαση για το τι του αρέσει ή όχι.
- Synonyms: μερικά όμοια αντικείμενα έχουν διαφορετικό όνομα, με αποτέλεσμα τα περισσότερα RS να μην μπορούν να το αναγνωρίσουν θεωρώντας τα διαφορετικά. Για παράδειγμα “Adventure film” & “Adventure monie” αναφέρονται στο ίδιο αντικείμενο αλλά με διαφορετικό όνομα. Αυτό έχει ως αποτέλεσμα την μείωση της απόδοσης των CF συστημάτων
- Sparsity: Για να μπορέσουν να λειτουργήσουν χρειάζονται ένα μεγάλο αριθμό από χρήστες & αξιολογήσεις.



Εικόνα 2- Collaborative Filtering example (πηγή <http://i65.tinypic.com/>)

## 2.1.2 Content-based Filtering

Στις content-based προσεγγίσεις ο χρήστης αξιολογεί ένα σύνολο από αντικείμενα και στην συνέχεια δημιουργείται ένα profile των ενδιαφερόντων του σύμφωνα με τα χαρακτηριστικά των αντικειμένων που αξιολόγησε. Ο αλγόριθμος προτείνει στον χρήστη αντικείμενα τα οποία έχουν παρόμοια χαρακτηριστικά με αυτά του προφίλ του. Για παράδειγμα σε μια εφαρμογή για recommendation σε ταινίες, αν ο χρήστης έχει μια τάση/του αρέσουν οι ταινίες επιστημονικής φαντασίας, τότε ο αλγόριθμος θα του προτείνει περισσότερες ταινίες από αυτήν την κατηγορία.

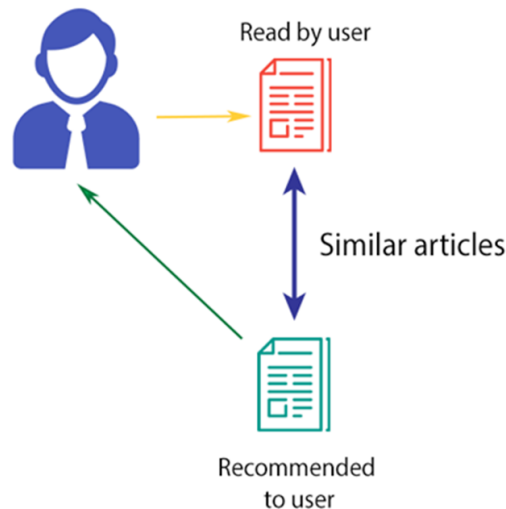
Πλεονεκτήματα του Content-based Filtering:

- user independence: Ενώ στο CF χρειάζονται τόσο οι βαθμολογίες του χρήστη όσο και των άλλων χρηστών ώστε να μπορέσει να γίνει σύσταση πληροφορίας, στο Content-Based χρειάζονται μόνο οι βαθμολογίες του χρήστη.
- Εξαλείφεται το first-rater problem: Λόγω του ότι οι Content-based αλγόριθμοι βασίζονται στα χαρακτηριστικά ενός αντικείμενου για να το προτείνουν στον χρήστη και όχι στις βαθμολογίες χρηστών πάνω σε αυτό μπορούν να προτείνουν χωρίς κανένα πρόβλημα νέα αντικείμενα.
- Διαφάνεια: Συμπεράσματα για το πως λειτουργεί το RS μπορούν να εξαχθούν παρατηρώντας τα χαρακτηριστικά του αντικείμενου που συστάθηκε. Βασιζόμενος σε αυτό ο χρήστης μπορεί να καταλήξει στο αν το RS είναι έμπιστο ή όχι. Ενώ στο CF ο χρήστης δεν μπορεί να ξέρει πως ακριβώς προέκυψαν τα recommendations αφού ο αλγόριθμος προτείνει αντικείμενα που αρέσουν σε χρήστες των οποίων οι βαθμολογίες μοιάζουν με τις δικές του.

Μειονεκτήματα Content-based Filtering:

- Ένα χρήστης πρέπει να βαθμολογήσει μεγάλο αριθμό από αντικείμενα προτού ο αλγόριθμος να “καταλάβει” τις προτιμήσεις του ώστε είναι σε θέση να του παρέχει αξιόπιστα recommendations.
- Filter-bubble: Λόγω της φύσης του το Content-based filtering προτείνει αντικείμενα τα οποία έχουν παρόμοια χαρακτηριστικά με αυτά που αρέσουν στον χρήστη, εάν αυτός έχει βαθμολογήσει θετικά για παράδειγμα μόνο ταινίες από μια συγκεκριμένη κατηγορία, τότε ο αλγόριθμος θα του προτείνει μόνο ταινίες από αυτήν.

### CONTENT-BASED FILTERING



Εικόνα 3 – Content-based filtering (Πηγή <http://i65.tinypic.com/>)

## 2.1.3 Knowledge-based

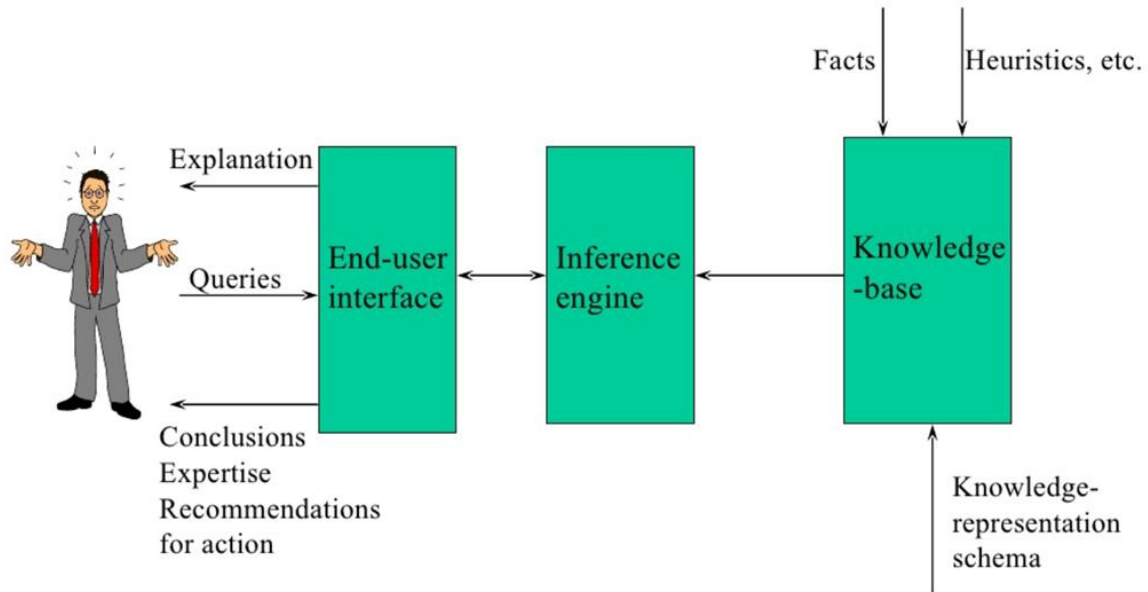
Τα Knowledge-based RS (KBRS) παρέχουν συμβουλές στον χρήστη για μια απόφαση που θέλει να λάβει ή μια ενέργεια που πρέπει να κάνει. Βασίζονται σε γνώση που παρέχεται από ειδικούς στον τομέα και σε συνδυασμό με τα δεδομένα εισόδου παράγουν τα recommendations. Εν αντιθέσει με τις περισσότερες προσεγγίσεις τα

KBRS δεν βασίζονται καθόλου στις αξιολογήσεις/συμπεριφορές των υπόλοιπων χρηστών. Ακόμα και στις ελάχιστες περιπτώσεις που θα το κάνουν δεν θα παίξει σημαντικό ρόλο στην εξαγωγή των recommendation.

Τα πλεονεκτήματα των KBRS είναι τα εξής:

- Τα cold-start, first-rater & Gray Sheep προβλήματα αποφεύγονται.
- Δεν απαιτείται dataset με ιστορικό χρήσης και βαθμολογιών.
- Τα recommendations είναι πιο αξιόπιστα αφού αφού τα δεδομένα δεν έχουν θόρυβο.

Με το μειονέκτημα τους να είναι ότι η συλλογή γνώσης (knowledge acquisition task) που απαιτείται για την



λειτουργία τους καθιστά την ανάπτυξη και την συντήρηση τους δαπανηρή.

Εικόνα 4 - Knowledge-based (πηγή <https://www.slideshare.net/>)

Στον ακόλουθο πίνακα γίνεται μια σύγκριση των 3 προαναφερθέντων προσεγγίσεων

Approach	Conceptual Goal	Input
Collaborative	Give me recommendations based on a collaborative approach that leverages the ratings and actions of my peers/myself.	User ratings + community ratings
Content-based	Give me recommendations based on the content (attributes) I have favored in my past ratings and actions.	User ratings + item attributes
Knowledge-based	Give me recommendations based on my explicit specification of the kind of content (attributes) I want.	User specification + item attributes + domain knowledge

Πίνακας 1 – σύγκριση RS (πηγή *Recommender Systems: The Textbook*)

## 2.1.4 Context-based Recommender Systems

Συστήματα σύστασης πληροφορίας με βάση το περιεχόμενο λαμβάνουν υπόψη τους τη συνάφεια των δεδομένων στις συστάσεις που κάνουν. Για παράδειγμα ένα σύστημα μπορεί να προτείνει στον χρήστη ρούχα, βασισμένο στην εποχή άλλα και στην τοποθεσία του.

### 2.1.4.1 Time Sensitive Recommender Systems

Σε πολλές περιπτώσεις η σύσταση πληροφορίας για ένα αντικείμενο μπορεί να αλλάξει με την πάροδο του χρόνου. Για παράδειγμα οι προτάσεις για ένα τραγούδι το οποίο την τρέχουσα περίοδο είναι αρκετά δημοφιλές μπορεί να αλλάξουν με την πάροδο του χρόνου λόγω του ότι το τραγούδι έπαψε να είναι δημοφιλές. Ενώ στην περίπτωση της σύστασης ρούχων ανάλογα με την εποχή, το σύστημα θα πρότεινε διαφορετικά ρούχα στους χρήστες. Για παράδειγμα το να προτείνει χειμωνιάτικα ρούχα ενώ είναι καλοκαίρι δεν θα ήταν και η καλύτερη επιλογή.

### 2.1.4.2 Location-Based Recommender Systems

Λόγω της ευρείας χρήσης των κινητών τηλεφώνων που παρέχουν υπηρεσία GPS, οι χρήστες ενδιαφέρονται για location-based recommendation. Για παράδειγμα ένας χρήστης που επισκέπτεται μια νέα πόλη θέλει να δειπνήσει σε ένα εστιατόριο, έτσι ο αλγόριθμος θα βασιστεί τόσο στις προηγούμενες βαθμολογίες του χρήστη σε εστιατόρια αλλά και στο ότι είναι σε μια συγκεκριμένη περιοχή προτείνοντας του εστιατόρια που πιθανόν θα του αρέσουν και είναι σε κοντινή απόσταση. Υπάρχουν δύο τύποι εγγύτητας στο χώρο που είναι κοινά σε αυτά τα συστήματα:

1. User-specific locality: Η γεωγραφική θέση του χρήστη ίσως παίζει σημαντικό ρόλο στις προτιμήσεις του. Για παράδειγμα ένας χρήστης από την Ελλάδα πιθανόν να ακούει διαφορετικού είδους μουσική από έναν χρήστη στην Ολλανδία. Αυτό αναφέρεται σαν “preference locality”
2. Item-specific locality: Η γεωγραφική θέση ενός αντικειμένου (πχ ένα εστιατόριο) ίσως έχει μεγάλη επίδραση στην σύσταση του, ανάλογα με την τρέχουσα γεωγραφική θέση του χρήστη. Για παράδειγμα ένας χρήστης πιθανόν να μην ενδιαφερόταν για ένα εστιατόριο που είναι 50 ή 100 χιλιόμετρα μακριά του. Αυτό αναφέρεται σαν “travel locality”.

## 2.1.5 Demographic-Based

Σε αυτήν την προσέγγιση η σύσταση της πληροφορίας γίνεται με βάση τα δημογραφικά χαρακτηριστικά του χρήστη. Για παράδειγμα σε μια εφαρμογή για αναζήτηση τουριστικού προορισμού, ανάλογα με την ηλικία του χρήστη οι συστάσεις θα ήταν διαφορετικές (πχ για ηλικίες από 18-30 προορισμοί για πιο περιπετειώδεις διακοπές, ενώ για ηλικίες από 70-90 πιο ήρεμους και ξεκούραστους προορισμούς). Συνήθως χρησιμοποιούνται μαζί με κάποια άλλη μέθοδο RS δημιουργώντας ένα Hybrid-RS.

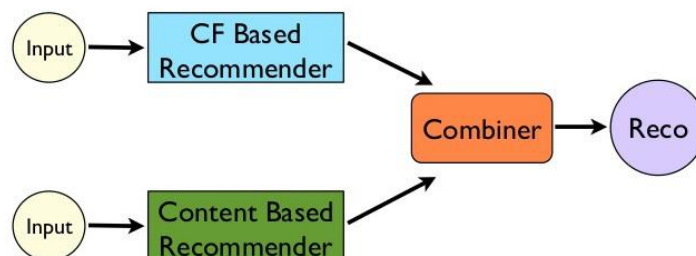
Attribute	Values
Gender	male, female
Age	1-11, 12-17, 18-30, 31-45, 46-59, 60+
Marital Status	single, engaged, loving secretly, married, relationship seeking, bereft of one's spouse, separated, divorced, ambiguous, loving
Education	literature, natural science, engineering, social sciences, medical science, art, others
Career	internet technology, designing, media, service industry, manufacturing, medicine, scientific research, management, others
Interests (Weibo tags)	travel, photographing, music and movie, computer games, Internet surfing, other

Εικόνα 5 – Παράδειγμα δημογραφικών στοιχείων (πηγή <https://www.semanticscholar.org/>)

## 2.1.6 Hybrid Recommender Systems

Τα Hybrid recommender systems συνδυάζουν δύο ή περισσότερες από τις προαναφερθείσες τεχνικές με σκοπό η μία να διορθώνει τα αρνητικά στοιχεία της άλλης. Για παράδειγμα στο Collaborative filtering υπάρχει το πρόβλημα των νέων αντικειμένων, γιατί ο αλγόριθμος δεν μπορεί να προτείνει αντικείμενα για τα οποία δεν υπάρχουν αξιολογήσεις. Ενώ στις Content-based προσεγγίσεις αυτό το πρόβλημα δεν υπάρχει διότι ο αλγόριθμος χρησιμοποιεί τα χαρακτηριστικά ενός αντικειμένου ώστε να καταλήξει στο συμπέρασμα αν πρέπει να το προτείνει στο χρήστη ή όχι.

### Hybrid Recommendations



Εικόνα 6 – Hybrid Recommendation (πηγή <http://dataconomy.com/>)

## 2.2 Real world examples of RS

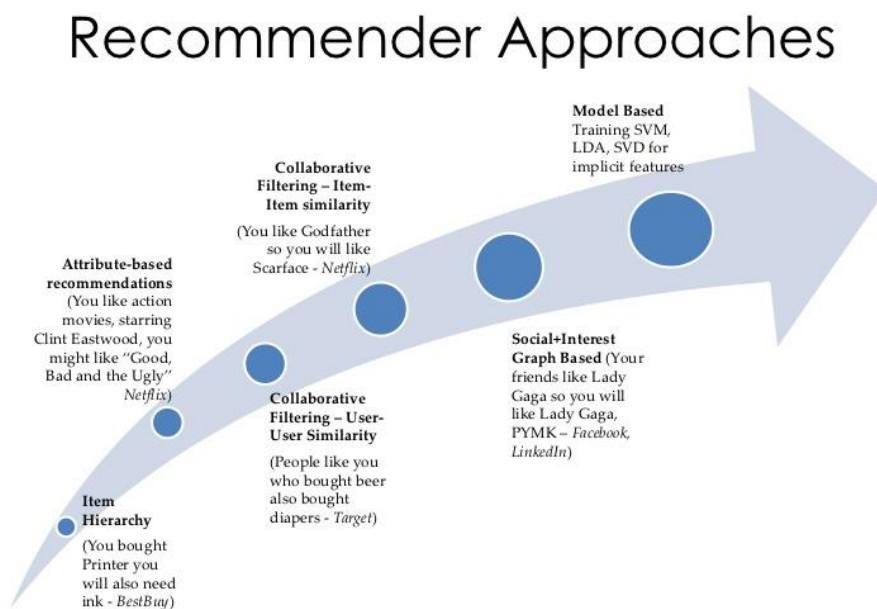
### 2.2.1 GroupLens Recommender System

Το GroupLens ήταν ένα πρωτοποριακό RS για σύσταση πληροφορίας σε άρθρα του Usenet και αναπτύχθηκε από το ομώνυμο ερευνητικό εργαστήριο του Τμήματος Επιστήμης Υπολογιστών & Μηχανικής του Πανεπιστημίου της Μινεσότα (Department of Computer Science and Engineering at the University of Minnesota). Το σύστημα συνέλεγε αξιολογήσεις χρηστών και τις χρησιμοποιούσε για να προβλέψει αν το συγκεκριμένο άρθρο θα άρεσε ή όχι σε άλλους χρήστες που δεν το είχαν διαβάσει ακόμα. Βασισόμενη στο GroupLens η ομάδα δημιούργησε και άλλους αλγορίθμους για σύσταση πληροφορίας σε βιβλία και ταινίες (BookLens & MovieLens). Πέρα από την συνεισφορά τους πάνω στην έρευνα του Collaborative filtering, κυκλοφόρησαν και πολλά μεγάλα datasets όταν ακόμα η έρευνα στο πεδίο ήταν σε αρχικό στάδιο βοηθώντας



έτσι στην εξέλιξη του. Πλέον στην ιστοσελίδα τους (<https://grouplens.org/datasets>) υπάρχουν διαθέσιμα αρκετά datasets όπως:

- **MovieLens 20M Dataset**, το οποίο περιλαμβάνει 20 εκατομμύρια βαθμολογίες από 138 χιλιάδες χρήστες σε 27 χιλιάδες ταινίες
- **HetRec 2011**, που περιλαμβάνει
  1. 105,000 bookmarks από 1867 χρήστες από το Delicious.com.
  2. 92,800 αρχεία με στοιχεία ακρόασης καλλιτεχνών από 1892 χρήστες στο Last.FM
  3. 86,000 βαθμολογίες από 2113 χρήστες του από διάφορα site
- **BookCrossing (BX) dataset**, το οποίο περιέχει 278,858 χρήστες και 1,149,780 βαθμολογίες τους (explicit / implicit) σε 271,379 βιβλία.



Εικόνα 7 – Τρόποι σύστασης πληροφορίας (<http://www.slideshare.net>)

### 2.2.2 Amazon.com Recommender System

Όπως και το GroupLens, η Amazon ήταν από τους πρωτοπόρους στα RS ειδικά στον εμπορικό τομέα. Ξεκίνησε πουλώντας βιβλία καταλήγοντας πλέον να εμπορεύεται σχεδόν τα πάντα ταινίες, μουσική, ρούχα, ηλεκτρονικές συσκευές κλπ. Για την σύσταση πληροφορίας χρησιμοποιεί της αξιολογήσεις (explicit rating/feedback) & την online δραστηριότητα (Buying behavior και την browsing behavior) του χρήστη (implicit rating/feedback). Σε πολλές περιπτώσεις παρέχεται επεξήγηση για τον λόγο που προτείνει στον χρήστη ένα συγκεκριμένο αντικείμενο.

### Frequently Bought Together



Price For All Three: \$258.02

[Add all three to Cart](#)

- This item:** [The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition \(Springer Series in Statistics\)](#) by Trevor Hastie
- [Pattern Recognition and Machine Learning \(Information Science and Statistics\)](#) by Christopher M. Bishop
- [Pattern Classification \(2nd Edition\)](#) by Richard O. Duda

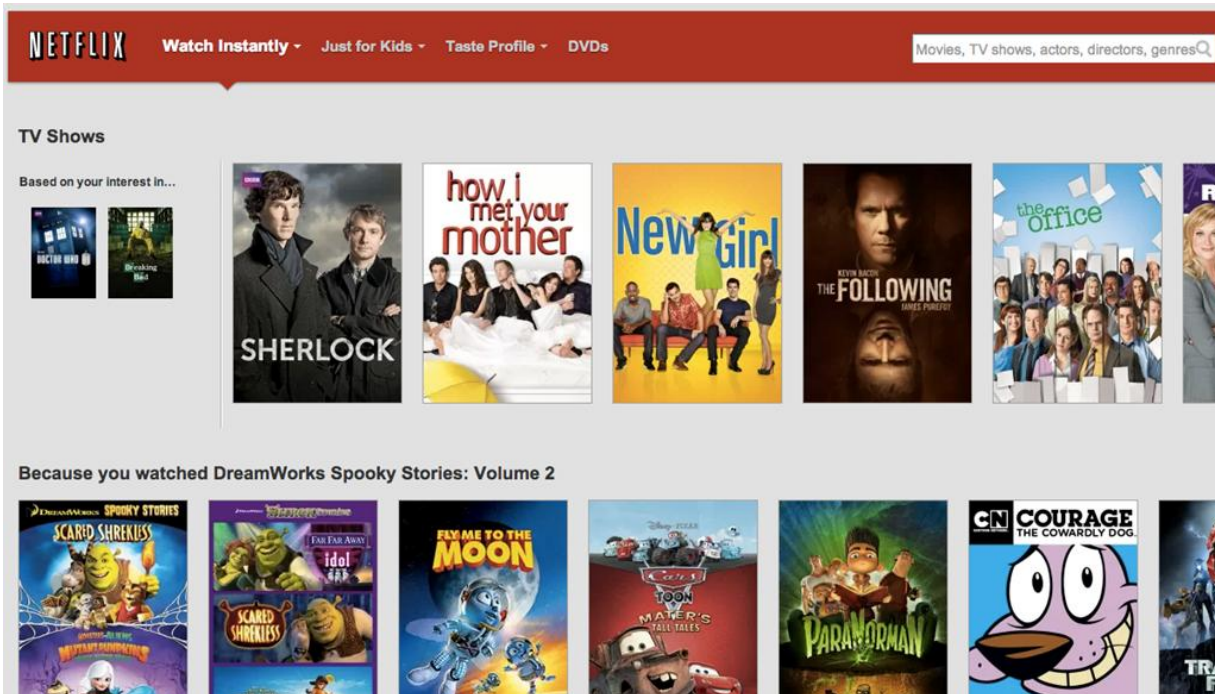
### Customers Who Bought This Item Also Bought

<a href="#">All of Statistics: A Concise Course in Statist...</a> by Larry Wasserman ★★★★☆ (9) \$60.00	<a href="#">Pattern Classification (2nd Edition)</a> by Richard O. Duda ★★★★☆ (27) \$117.25	<a href="#">Data Mining: Practical Machine Learning Tools an...</a> by Ian H. Witten ★★★★☆ (29) \$41.55	<a href="#">Bayesian Data Analysis, Second Edition (Texts in...</a> by Andrew Gelman ★★★★☆ (10) \$56.20	<a href="#">Data Analysis Using Regression and Multilevel /...</a> by Andrew Gelman ★★★★☆ (13) \$39.59

Εικόνα 8 – Amazon Recommendations (πηγη <https://images.martechadvisor.com/>)

## 2.2.3 Netflix Movie Recommender System

Το Netflix αρχικά όταν ιδρύθηκε ήταν εταιρία ενοικίασης DVD ταινιών και τηλεοπτικών σειρών μέσω ταχυδρομείου για να γίνει τελικά η εταιρία μετάδοσης πολυμεσικού περιεχομένου που όλοι πλέον γνωρίζουμε. Το Netflix δίνει την δυνατότητα στους χρήστες να βαθμολογήσουν ταινίες σε κλίμακα 5 σημείων/βαθμών. Επιπλέον συλλέγει τις ενέργειες του χρήστη όπως αν είδε μια ταινία ή σειρά και σε συνδυασμό με τις βαθμολογίες του παράγει τα ανάλογα recommendations. Επίσης παρέχει στον χρήστη μια περιγραφή για τον λόγο που προτείνει ένα αντικείμενο το οποίο τον βοηθάει να αποφασίσει αν θα δει μια ταινία ή όχι. Με αυτόν τον τρόπο αυξάνεται η πιθανότητα ένας χρήστης να ανταποκριθεί στο recommendation ενώ ταυτόχρονα βελτιώνει την εμπειρία χρήσης (user experience). Το Netflix θέλοντας να βελτιώσει τα recommendation του ξεκίνησε έναν ανοιχτό διαγωνισμό γνωστό και ως “Netflix Prize” αναζητώντας τον καλύτερο Collaborative-filtering αλγόριθμο, με το έπαθλο \$1.000.000 να δίνεται στην ομάδα που θα έβγαζε τον καλύτερο αλγόριθμο. Συνολικά έγιναν 44,014 υποβολές από τις 5,169 ομάδες που έλαβαν μέρος. Ο διαγωνισμός αυτός όπως ήταν αναμενόμενο είχε σημαντική συνεισφορά στην ερευνητική κοινότητα των RS.



Εικόνα 9 – Netflix Recommendations (πηγη <https://gigaom.com/>)

## 2.2.4 Facebook Friend Recommendations

Οι χρήστες χρησιμοποιούν τα κοινωνικά δίκτυα για να επικοινωνήσουν με φίλους, να γνωρίσουν νέους, να μοιραστούν τις απόψεις τους κλπ. Το Facebook προτείνει στους χρήστες, φίλους μέσω της “καρτέλας” People you may know. Για να εμφανιστεί ένας χρήστης στους προτεινόμενους φίλους κάποιου άλλου χρήστη συνήθως ισχύουν κάποιοι από τους ακόλουθους παράγοντες:

- Έχουν κοινούς φίλους
- Έχουν κοινά χαρακτηριστικά στο προφίλ τους (πόλη κατοικίας, πόλη καταγωγής, εργασία, εκπαίδευση κλπ)
- Έχουν κοινά ενδιαφέροντα πχ είναι μέλη στις ίδιες ομάδες ή έχουν κάνει like στις ίδιες σελίδες
- Τουλάχιστον ο ένας έχει αναζητήσει τον άλλον στο Facebook
- Συνδέονται σε αυτό μέσω του ίδιου τοπικού δικτύου
- Ο ένας χρήσης έκανε συγχρονισμό των email επαφών του και εκεί υπήρχε το email του άλλου χρήστη.

## 2.3 Impact of Recommender Systems

Σήμερα οι χρήστες του διαδικτύου καθημερινά έρχονται σε επαφή με κάποιο RS είτε απλώς σερφάροντας στο διαδίκτυο (πχ Facebook) το οποίο τους προτείνει φίλους, σελίδες, ομάδες κλπ, είτε αναζητώντας προϊόντα σε κάποιο e-shop (πχ Amazon) όπου οι συστάσεις είναι κυρίως για προϊόντα όπως βιβλία, κινητά, ρούχα κλπ. Ειδικότερα στον τομέα του ηλεκτρονικού εμπορίου όπου δεν υφίσταται ο περιορισμός χώρου όπως σε ένα πραγματικό κατάστημα, η ιστοσελίδα μπορεί να παρέχει έναν τεράστιο αριθμό προϊόντων. Λόγω του τεράστιου όγκου της πληροφορίας χωρίς την χρήση RS θα ήταν τρομερά δύσκολο για τον χρήστη να βρει αυτό που πραγματικά χρειάζεται. Επιπλέον αγοράζοντας ή βλέποντας κάποιο συγκεκριμένο προϊόν μπορούν να του προταθούν προϊόντα αντίστοιχα ή συνδυαζόμενα με αυτό. Για παράδειγμα εάν ο χρήστης αγόρασε ένα κινητό θα μπορούσε να του προτείνει και τις αντίστοιχες θήκες για αυτό, ενώ εάν βρίσκεται σε αναζήτηση για αγορά κάποιας συσκευής θα μπορούσε να του προτείνει κινητά με παρόμοια χαρακτηριστικά. Λόγω της ευρείας χρήσης των RS σε μεγάλο αριθμό ιστοσελίδων οι χρήστες προτιμούν αυτές που χρησιμοποιούν κάποιο είδος RS.

Τα τελευταία χρόνια έχουν γίνει αρκετές προσπάθειες προσδιορισμού της επίδρασης των RS κυρίως σε εμπορικές εφαρμογές και το πώς αυτά επηρεάζουν τις πωλήσεις. Η επικρατούσα άποψη είναι ότι υπάρχει ουσιαστική αύξηση στις πωλήσεις προϊόντων. Αναλυτικότερα:

- Οι πωλήσεις ενός προτεινόμενου προϊόντος επηρεάζονται από την πιθανότητα που έχει το συγκεκριμένο προϊόν στο να προταθεί στους χρήστες. Οι Dokyun Lee & Kartik Hosanagar[2] οι οποίοι μελέτησαν κάποια δεδομένα που τους παρείχε μια από τις μεγαλύτερες εταιρίες ηλεκτρονικού εμπορίου στην βόρεια Αμερική, παρατήρησαν ότι τα RS αύξησαν κατά 25% τις επισκέψεις σε προϊόντα ενώ ταυτόχρονα υπήρξε και 35% αύξηση στις πωλήσεις. Επιπλέον παρατήρησαν ότι οι “Recently Viewed” συστάσεις οι οποίες χρησιμοποιούνται από πολλά e-shop δεν είναι ιδιαίτερα αποτελεσματικές.
- Σύμφωνα με τους Daniel Fleder and Kartik Hosanagar[4] τα συστήματα σύστασης πληροφορίας τα οποία δεν λαμβάνουν υπ’ όψιν το πόσο δημοφιλής είναι τα αντικείμενα μπορεί να οδηγήσουν σε αυξήσεις των πωλήσεων. Επίσης παρατηρήθηκε ότι η ποικιλία των πωλήσεων μπορεί να αυξηθεί σε ατομικό επίπεδο όταν χρησιμοποιείται ένα RS ενώ ταυτόχρονα μειώνεται σε συλλογικό επίπεδο.

## 2.4 Ο SCoR

Ο SCoR είναι ένας αλγόριθμος για σύσταση πληροφορίας ο οποίος αναπτύχθηκε στο εργαστήριο Δικτύων και Κατανεμημένων Συστημάτων (NETDIL) του TEI Κρήτης. Βασίστηκε στον αλγόριθμο Vivaldi ο οποίος δημιουργήθηκε για την μέτρηση των καθυστερήσεων στο διαδίκτυο.

### 2.4.1 Ο αλγόριθμος Vivaldi

Ο αλγόριθμος χρησιμοποιεί τον ευκλείδειο χώρο συντεταγμένων.

- Στην αρχή κάθε κόμβος λαμβάνει μία τυχαία θέση στον χώρο
- Στην συνέχεια κάθε κόμβος παίρνει τυχαία ένα μικρό αριθμό  $X$  από άλλους κόμβους και μετράει την καθυστέρηση που έχει με αυτούς.
- Σε κάθε γύρο κατά την εκτέλεση του αλγορίθμου κάθε κόμβος επιλέγει τυχαία έναν από τους  $X$  αυτούς κόμβους και προσαρμόζει την θέση του στον χώρο ώστε αυτή να αντικατοπτρίζει την καθυστέρηση που υπάρχει μεταξύ τους.
- Μετά το πέρας της εκτέλεσης του αλγορίθμου η ευκλείδεια απόσταση μεταξύ 2 οποιονδήποτε κόμβων μας δίνει την συνολική καθυστέρηση που υπάρχει ανάμεσα τους.

### 2.4.2 Περιγραφή του SCoR αλγορίθμου

Αρχικά δημιουργείται ένας διμερής γράφος από χρήστες και αντικείμενα (αντί μόνο host που χρησιμοποιεί ο Vivaldi). Για κάθε βαθμολογία ενός χρήστη σε ένα αντικείμενο προστίθεται μια ακμή μεταξύ του χρήστη και του αντικειμένου. Σε κάθε ακμή δίνεται ένα βάρος ανάλογο με την βαθμολογία του χρήστη στο συγκεκριμένο αντικείμενο (όσο υψηλότερη η βαθμολογία του χρήστη τόσο μικρότερο το βάρος που της ανατίθεται). Το εύρος τιμών των βαρών κυμαίνεται από 0 έως 100 ανάλογα με την βαθμολογία του χρήστη (η τιμή 100 δίνεται όταν ο χρήστης έχει βαθμολογήσει το αντικείμενο με την μικρότερη δυνατή βαθμολογία, ενώ 0 όταν το έχει βαθμολογήσει με την υψηλότερη δυνατή βαθμολογία). Έτσι η επιθυμητή απόσταση μεταξύ ενός χρήστη και ενός αντικειμένου που έχει βαθμολογήσει δίνεται από τον ακόλουθο τύπο

$$dd(u, i) = 100 \cdot \left( \frac{\max R - r(u, i)}{\max R - \min R} \right)$$

όπου  $\max R$  &  $\min R$  είναι η μεγαλύτερη και η μικρότερη δυνατή τιμή του recommendation αντίστοιχα. Ο αλγόριθμος εκτελείται με τα ακόλουθα βήματα:

- όλοι οι κόμβοι (και χρήστες και αντικείμενα) παίρνουν μια τυχαία θέση στον χώρο (γραμμές 1-6).
- περιοδικά, κάθε κόμβος  $X$  επιλέγει τυχαία ένα άλλο κόμβο  $Y$  από τον μικρό αριθμό των κόμβων που συνδέεται. Στην συνέχεια ο  $X$  κόμβος υπολογίζει την τρέχουσα απόσταση του από τον  $Y$  σύμφωνα με την παρακάτω εξίσωση και την συγκρίνει με την επιθυμητή του απόσταση (γραμμή 8).

$$d(u, i) = \|p(u) - p(i)\|_2$$

- Έπειτα ο κόμβος  $X$  μετακινείται προς την ανάλογη κατεύθυνση, ώστε η ευκλείδεια απόσταση μεταξύ των κόμβων να είναι πιο κοντά στην επιθυμητή.
- Η διαδικασία αυτή επαναλαμβάνεται μέχρι η θέση κάθε κόμβου να σταθεροποιηθεί (γραμμή 9).

Μετά το πέρας της εκτέλεσης του αλγορίθμου η ευκλείδεια απόσταση μεταξύ ενός χρήστη και ενός αντικείμενου που δεν έχει βαθμολογήσει, μας δείχνει την πιθανή βαθμολογία που ο χρήστης θα έβαζε σε αυτό το αντικείμενο, χρησιμοποιώντας την ακόλουθη εξίσωση.

$$\hat{r}(u, i) = \max R - (\max R - \min R) \cdot \frac{\|p(u) - p(i)\|_2}{100}$$

```

input :  $i \in I, u \in U, (u, i, r(u, i)) \in TS, \min R, \max R.$ 
output:  $\hat{r}(u, i), (u, i) \in VS$ 

1 foreach  $u \in U$  do
2    $p(u) =$  random position in  $\mathbb{R}^n$ 
3 end
4 foreach  $i \in I$  do
5    $p(i) =$  random position in  $\mathbb{R}^n$ 
6 end
7 repeat
8    $(u, i) = \text{getRandomSample}(TS)$   $[p(u), p(i)] = \text{Vivaldi}(p(u), p(i), r(u, i))$ 
9 until  $\forall x \in I \cup U$   $p(x)$  is stable
10 foreach  $(u, i) \in TS$  do
11    $W(u, i) = e^{-0.2 \cdot MSE(u)} \cdot (d(u, i) - d(u, i))^2$ 
12 end
13 repeat
14    $(u, i) = \text{getWeightedRandomSample}(TS, W)$ 
    $[p(u), p(i)] = \text{Vivaldi}(p(u), p(i), r(u, i))$ 
15 until  $\forall x \in I \cup U$   $p(x)$  is stable
16 foreach  $(u, i) \in VS$  do
17    $\hat{r}(u, i) = \max R - (\max R - \min R) \cdot \frac{\|p(u) - p(i)\|_2}{100}$ 
18   if  $\hat{r}(u, i) < \min R$  then
19      $\hat{r}(u, i) = \min R$ 
20   else if  $\hat{r}(u, i) > \max R$  then
21      $\hat{r}(u, i) = \max R$ 
22   end
23 end

```

Εικόνα 10 - Ψευδοκώδικας του αλγορίθμου SCoR

### 3 Οι τεχνολογίες που χρησιμοποιήθηκαν

Σε αυτό το κεφάλαιο γίνεται περιγραφή των βασικών τεχνολογιών που χρησιμοποιήθηκαν κατά την υλοποίηση της παρούσας πτυχιακής.

#### 3.1 Java



Εικόνα 11 –

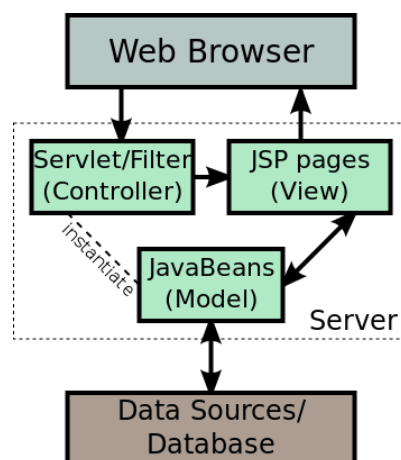
Java (Πηγή

<https://www.udemy.com/java-tutorial/>)

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού, σχεδιάστηκε από τον James Gosling στην Sun Microsystems. Η πρώτη εκδόσή της κυκλοφόρησε στις 23 Μαΐου του 1995. Τα Java προγράμματα μπορούν να τρέξουν σε οποιοδήποτε υπολογιστή ανεξαρτήτως λειτουργικού συστήματος ή αρχιτεκτονικής (write once, run anywhere) χωρίς να χρειάζονται αλλαγές στον κώδικα του.

#### 3.2 JavaServer Pages (JSP)

Το JSP είναι μια server-side τεχνολογία για την δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο γράφοντας Java κώδικα μέσα στο HTML αρχείο. Κάνοντας build to web app, παράγεται ένα war αρχείο το οποίο γίνεται deploy σε ένα web server όπως ο Apache Tomcat ή ο GlassFish.



Εικόνα 12 - JSP αρχιτεκτονική (πηγή <https://en.wikipedia.org>)

#### 3.3 Android

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα για κινητά τηλέφωνα και βασίζεται στον πυρήνα του Linux. Η ανάπτυξη του ξεκίνησε από την Android, Inc η οποία εξαγοράστηκε από την Google το 2005.



Εικόνα 13 – Android logo (πηγή storage.googleapis.com)

Η πρώτη (Alpha) έκδοση του λειτουργικού συστήματος κυκλοφόρησε στις 23 Σεπτεμβρίου του 2008 ενώ έχουν βγει ακόμα 25 εκδόσεις από τότε, με την τελευταία να είναι η Nougat (7.2.2) η οποία κυκλοφόρησε στις 22 Αυγούστου του 2016.

Code name	Version number	Initial release date	API level	Support status <sup>[2]</sup>
Alpha	1.0	September 23, 2008	1	Unsupported
Beta	1.1	February 9, 2009	2	Unsupported
Cupcake	1.5	April 27, 2009	3	Unsupported
Donut	1.6	September 15, 2009	4	Unsupported
Eclair	2.0 - 2.1	October 26, 2009	5 - 7	Unsupported
Froyo	2.2 - 2.2.3	May 20, 2010	8	Unsupported
Gingerbread	2.3 - 2.3.7	December 6, 2010	9 - 10	Unsupported
Honeycomb	3.0 - 3.2.6	February 22, 2011	11 - 13	Unsupported
Ice Cream Sandwich	4.0 - 4.0.4	October 18, 2011	14 - 15	Unsupported
Jelly Bean	4.1 - 4.3.1	July 9, 2012	16 - 18	Unsupported
KitKat	4.4 - 4.4.4	October 31, 2013	19	Unsupported
Lollipop	5.0 - 5.1.1	November 12, 2014	21 - 22	Unsupported
Marshmallow	6.0 - 6.0.1	October 5, 2015	23	Supported
<b>Nougat</b>	<b>7.0 - 7.1.2</b>	<b>August 22, 2016</b>	<b>24 - 26</b>	<b>Supported</b>
O	8.0	TBA	27	Supported

Πίνακας 2 – Android Versions (Πηγή <https://en.wikipedia.org/>)

### 3.4 SQL (Structured Query Language)

Η SQL είναι μια γλώσσα προγραμματισμού σχεδιασμένη για την διαχείριση σχεσιακών βάσεων δεδομένων.



Οι βασικές ενέργειες που εκτελούνται στην SQL είναι:

- Δημιουργία βάσεων & πινάκων (Create)
- Εισαγωγή δεδομένων σε ένα πίνακα (Insert)
- Ανάκτηση δεδομένων από την βάση και εκτέλεση ερωτημάτων (Select, Query)
- Ανανέωση τιμών σε υπάρχοντα πεδία ενός πίνακα (Update)
- Διαγραφή γραμμών από ένα πίνακα (Delete)

### 3.4.1 MySQL

Η MySQL είναι ένα ανοιχτού κώδικα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS).

### 3.5 Jsoup

Το Jsoup μια ανοιχτού κώδικα Java βιβλιοθήκη για την ανάλυση των περιεχομένων ενός HTML αρχείου (HTML parsing). Παρέχει τις εξής δυνατότητες:

- Σάρωση HTML από ένα URL, αρχείο, ή string
- Μπορεί να εντοπίσει και να εξάγει δεδομένα με τη χρήση διάρχισης DOM (DOM traversal) ή CSS selectors
- Να διαχειριστεί στοιχεία, χαρακτηριστικά και κείμενο HTML
- Να "καθαρίσει" το περιεχόμενο που δίνει ο χρήστης βάση μιας white-list για την αποφυγή επιθέσεων XSS
- Να εξάγει σωστά μορφοποιημένη HTML

## 4 Συλλογή δεδομένων

Σε αυτό το κεφάλαιο γίνεται περιγραφή του τρόπου με τον οποίο έγινε η συλλογή των δεδομένων τα οποία είναι απαραίτητα τόσο για την εκτέλεση του αλγορίθμου σύστασης πληροφορίας όσο και για τις δύο εφαρμογές.

Τα δεδομένα που χρειάζονται για την εκτέλεση του αλγορίθμου είναι βαθμολογίες χρηστών σε ταινίες. Ενώ για την λειτουργία των εφαρμογών τα δεδομένα (metadata) που χρειάζονται είναι τα εξής:

- Όνομα ταινίας
- Age Rating
- Είδος/Genre
- Σκηνοθέτης
- Διάρκεια
- Έτος κυκλοφορίας
- Το poster της ταινίας

### 4.1 Περιγραφή των φάσεων συλλογής δεδομένων

Η συλλογή των δεδομένων έγινε σε 2 φάσεις. Την φάση 1 όπου πάρθηκαν τα URL ταινιών και την φάση 2 όπου πάρθηκαν οι βαθμολογίες των χρηστών & τα metadata για κάθε ταινία που συλλέχτηκε στην φάση 1.

#### Φάση 1:

Για την συλλογή των URL των ταινιών εκτελέστηκε ένας Breadth First Search μηχανισμός, ο οποίος επισκέφθηκε το σύνολο των ταινιών μέσω των χρηστών που τις έχουν βαθμολογήσει. Χρησιμοποιήθηκαν οι 4 ακόλουθοι πίνακες:

- movies2visit όπου αποθηκεύονται τα URL των ταινιών από τις οποίες δεν έχουν ακόμα παρθεί οι χρήστες που τις βαθμολόγησαν
- moviesvisited όπου αποθηκεύονται τα URL των ταινιών από τις οποίες έχουν παρθεί οι χρήστες που τις βαθμολόγησαν
- users2visit όπου αποθηκεύονται τα URL των χρηστών από τους οποίους δεν έχουν ακόμα παρθεί τα url των ταινιών που έχουν βαθμολογήσει
- usersvisited όπου αποθηκεύονται τα URL των χρηστών από τους οποίους έχουν παρθεί τα url των ταινιών που έχουν βαθμολογήσει

Βήματα πρώτης φάσης:

- ⓐ Αρχικά συλλέγονται τα URLs κάποιων δημοφιλών ταινιών (Γραμμή 1)
- ⓑ Στην συνέχεια επισκέπτεται κάθε ταινία από αυτές και παίρνονται τα URL των χρηστών που τις έχουν βαθμολογήσει τα οποία τοποθετούνται στον πίνακα users2visit (Γραμμή 2 έως 7)
- ⓒ Στο επόμενο βήμα, για όσο ο πίνακας users2visit ή ο πίνακας movies2visit δεν είναι άδειος τότε (Γραμμή 9 έως 21):
  - Επισκέπτονται όλοι οι χρήστες από τον πίνακα users2visit από όπου συλλέγονται τα URL των ταινιών που έχουν βαθμολογήσει και τοποθετούνται στον πίνακα movies2visit. Στην συνέχεια ο χρήστης διαγράφεται από τον πίνακα users2visit και τοποθετείται στον πίνακα usersvisited (Γραμμή 10 έως 14).
  - Επισκέπτεται κάθε ταινία που υπάρχει στο movies2visit από την οποία συλλέγονται τα URL των προφίλ των χρηστών που την έχουν βαθμολογήσει τα οποία και αποθηκεύονται στον πίνακα users2visit. Τέλος η ταινία διαγράφεται από τον πίνακα movies2visit και τοποθετείται στον πίνακα moviesvisited (Γραμμή 16-20).

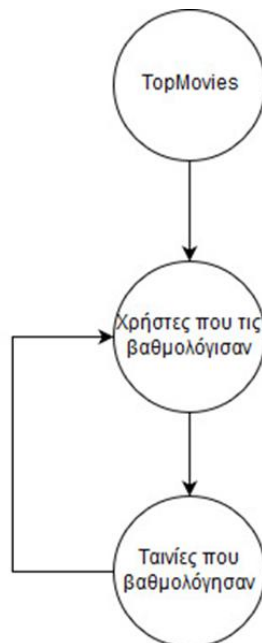
#### Ψευδοκώδικας:

1 *List moviesList = getTopMovies();*

```

2   FOR each movie in moviesList
3       List userProfile = getUserProfiles(movie);
4       FOR each userProfile in userProfiles
5           add userProfile to users2visit
6       ENDFOR
7   ENDFOR
8
9   WHILE movies2visit not empty OR users2visit not empty
10      FOR each user in users2visit
11          get movie urls from user profile and add them to movies2visit
12          remove user from users2visit
13          add user to usersvisited
14      ENDFOR
15
16      FOR each movie in movies2visit
17          get user profiles from movie and add them to users2visit
18          remove movie form movies2visit
19          add movie to moviesvisited
20      ENDFOR
21  ENDWHILE

```



Εικόνα 14 - Σχεδιάγραμμα πρώτης φάσης

## Φάση 2:

Επισκέπτεται κάθε ταινία που συλλέχθηκε στην φάση 1, από την οποία παίρνονται οι βαθμολογίες των χρηστών σε αυτήν την ταινία και τα metadata αυτής (Γραμμή 1 έως 4).

## Ψευδοκώδικας:

```

1   List moviesList = unionTables(movies2visit,moviesvisited)
2   FOR each movie in moviesList

```

```
3      get the metadata of the movie and add them to movieinfo table
4      get movie ratings and add them to ratings table
5      ENDFOR
```

## 4.2 Τα δεδομένα που συλλέχθηκαν

Κατά την εκτέλεση του προγράμματος συλλέχθηκαν:

- περίπου 6.000.000 βαθμολογίες χρηστών σε ταινίες
- 54025 ταινίες και τα metadata αυτών.

## 5 Web Εφαρμογή

Σε αυτό το κεφαλαίο γίνεται περιγραφή και παρουσίαση της web εφαρμογής σύστασης ταινιών.

### 5.1 Περιγραφή

Η εφαρμογή δίνει την δυνατότητα στον χρήστη δίνοντας το προφίλ ID του το οποίο αντιστοιχεί σε ένα URL με βαθμολογίες του σε ταινίες να λάβει recommendation. Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν οι εξής τεχνολογίες:

- HTML
- CSS
- JSP
- JavaScript
- GlassFish Web Server
- MySQL
- JSoup

Αποτελείται από 2 μέρη:

- το home page όπου ο χρήστης έχει την επιλογή να λάβει recommendation για πολλές ταινίες ή για μία ταινία της επιλογής του.
- το response page όπου εμφανίζονται τα αποτελέσματα και παρέχεται η δυνατότητα φιλτραρίσματος τους με διάφορους παράγοντες.

Στο home page υπάρχει μια φόρμα με δύο πεδία, ένα για να δώσει ο χρήστης το profile του ώστε να μπορεί ο αλγόριθμος να του προτείνει ταινίες και ένα δεύτερο (optional) στο οποίο ο χρήστης, γράφοντας το url μίας ταινίας λαμβάνει recommendation μόνο για αυτήν. Στο response page υπάρχει ένα table του οποίου κάθε γραμμή αντιστοιχεί σε μια προτεινόμενη ταινία, όπου παρουσιάζονται και τα ανάλογα metadata της:

- Movie Name
- Recommendation Rating
- Age Rating
- Genre
- Director
- Duration
- Year

Επίσης υπάρχει ένας δεύτερος πίνακας ο οποίος παρέχει την δυνατότητα φιλτραρίσματος των αποτελεσμάτων βάση διαφόρων κριτηρίων.

### 5.2 Τα μέρη της εφαρμογής

#### 5.2.1 Home Page

Όπως προαναφέρθηκε στο home page υπάρχει μια φόρμα με δύο πεδία όπου ο χρήστης δίνοντας το ID του προφίλ του έχει την δυνατότητα να λάβει recommendation

- Είτε για πολλές ταινίες
- Είτε για μία ταινία της επιλογής του (δίνοντας το url αυτής)

Επίσης υπάρχουν και οι ανάλογες οδηγίες για την χρήση την εφαρμογής.

Paste your rottentomatoes ID	<input type="text"/>	<input type="button" value="submit"/>
Paste the movie url (optional)	<input type="text"/>	

**Instructions:**

To Use this web app you need to:

- A) Login on [rotten tomatoes](#) using your facebook account
  - B) Make sure your rotten tomatoes account is public (check settings upper right, drop down menu->Account->Edit Privacy Settings->Show to all)
  - C) Rate a few movies on the site (both movies you liked and disliked)
- Note:** The quality of the results depends on the number of your ratings, more ratings equals better results
- D) Simply paste your rottentomatoes profile ID in the form above (the number at the end of the address of your profile)
  - E) (Optional) You can also paste the url of a specific movie on the rotten tomatoes site to get a recommendation for that movie only

Εικόνα 15 – Home Page form




Εικόνα 16 – Home page

## 5.2.2 Response Page

**Recommendations for John Kernel**  
total movies count: 100

Filter by: Year:  Director:  Rating:  Age Rating:  Genre:

Movie Name	Recommendation Rating	Age Rating	Genre	Director	Duration	Year
 <a href="#">X-Men: Days of Future Past</a>	5.0	PG-13	Action & Adventure, Science Fiction & Fantasy	Bryan Singer	2 hr. 11 min.	2014
 <a href="#">Rain Man</a>	5.0	R	Drama	Barry Levinson	2 hr. 13 min.	1988
	5.0	PG-13	Drama, Action &	Paul	1 hr. 55	2007

Εικόνα 17 - Response Page

Στο Response page εμφανίζονται οι ταινίες που προτείνονται στον χρήστη από τον αλγόριθμο. Όπως φαίνεται στην παραπάνω εικόνα κάθε γραμμή περιέχει τα metadata μίας ταινίας. Επίσης ο χρήστης έχει την δυνατότητα φιλτραρίσματος των ταινιών βάση:

- έτους
- σκηνοθέτη
- recommendation rating
- age rating
- και Genre.

Το φιλτράρισμα των ταινιών γίνεται με την χρήση Javascript.

Filter by: Year:  Director:  Rating:  Age Rating:  Genre:

Εικόνα 18 - Filter

### 5.3 Λειτουργικότητα της εφαρμογής

Στην περίπτωση του απλού recommendation ο χρήστης πληκτρολογεί το profile id του στο input πεδίο και πατάει submit

Paste your rottentomatoes ID   
 Paste the movie url (optional)


Εικόνα 19 – home page form

Αφού ο χρήστης πατήσει Submit τότε

1. Τρέχει JSoup κώδικας ο οποίος παίρνει τις βαθμολογίες από το προφίλ που έδωσε.
2. Τρέχει ο αλγόριθμος όπου

- στην περίπτωση του γενικού recommendation επιστρέφει ταινίες που θεωρεί ότι θα αρέσουν στον χρήστη
  - Ενώ στην περίπτωση του recommendation για συγκεκριμένη ταινία επιστρέφει μόνο αυτήν
- Και στις δύο περιπτώσεις ο χρήστης μεταβαίνει στο response page. Η μόνη διαφορά είναι ότι στην περίπτωση του γενικού recommendation υπάρχει η δυνατότητα φιλτραρίσματος των αποτελεσμάτων ενώ στην περίπτωση του recommendation για συγκεκριμένη ταινία όχι.




**Recommendations for John Kernel**  
total movies count: 1

Movie Name	Recommendation Rating	Age Rating	Genre	Director	Duration	Year
 <a href="#">Avatar</a>	3.93	PG-13	Action & Adventure, Mystery & Suspense, Science Fiction & Fantasy, Comedy	James Cameron	2 hr. 42 min.	2009

Εικόνα 20 – recommendation για συγκεκριμένη ταινία

**Recommendations for John Kernel**  
total movies count: 100

Filter by: Year:  Director:  Rating:  Age Rating:  Genre:

Movie Name	Recommendation Rating	Age Rating	Genre	Director	Duration	Year
 <a href="#">Rain Man</a>	4.86	R	Drama	Barry Levinson	2 hr. 13 min.	1988
 <a href="#">Lethal Weapon 2</a>	4.82	R	Drama, Action & Adventure	Richard Donner	1 hr. 52 min.	1989
 <a href="#">Speed</a>	4.81	R	Mystery & Suspense, Action & Adventure	Jan de Bont	1 hr. 55 min.	1994

Εικόνα 21 – γενικό recommendation

Το response page αποτελείται από 2 πίνακες

- Τον πίνακα για φιλτράρισμα
- Τον πίνακα παρουσίασης των ταινιών

καθώς και δύο <h1> tags όπου στο πρώτο αναγράφεται το username του προφίλ που έδωσε ο χρήστης, ενώ στο δεύτερο (“total movies count”) ο αριθμός των ταινιών που εμφανίζονται στον πίνακα παρουσίασης ταινιών.



Κάθε γραμμή του πίνακα παρουσίασης των ταινιών αντιστοιχεί σε μία ταινία. Ενώ υπάρχουν και 7 στήλες για τα metadata της, τα οποία είναι

1. Poster & όνομα της ταινίας
2. Recommendation Rating
3. Age Rating
4. Genre
5. Σκηνοθέτης
6. Διάρκεια
7. Έτος κυκλοφορίας

Ο πίνακας φιλτραρίσματος δίνει την δυνατότητα στον χρήστη να φιλτράρει τα αποτελέσματα βάση των ακόλουθων κριτηρίων

1. Έτος κυκλοφορίας
2. Σκηνοθέτη
3. Recommendation Rating
4. Age Rating
5. Genre



Στην περίπτωση που ο χρήστης επιλέξει να φιλτράρει τα αποτελέσματα τότε η τιμή του “total movies count” αλλάζει δείχνοντας έτσι στον χρήστη πόσες ταινίες εμφανίζονται μετά την εφαρμογή του φίλτρου.

### Recommendations for John Kernel

total movies count: 24

Filter by:

Any Science Fiction & Fantasy

Movie Name	Recommendation Rating	Age Rating	Genre	Director	Duration	Year
 <a href="#">Terminator 2: Judgment Day</a>	4.81	R	Action & Adventure, Science Fiction & Fantasy	James Cameron	2 hr. 33 min.	1991
 <a href="#">X-Men: Days of Future Past</a>	4.8	PG-13	Action & Adventure, Science Fiction & Fantasy	Bryan Singer	2 hr. 11 min.	2014
						

Εικόνα 22 – filtered movies

Κάνοντας Clear το φίλτρο όλα τα πεδία του φίλτρου επανέρχονται στην αρχική τιμή τους και εμφανίζονται ξανά όλες οι ταινίες, ενώ αλλάζει και η τιμή του “total movies count”.

## Recommendations for John Kernel

total movies count: 100

Filter by:

Year:

Director:

Rating:

Age Rating:

Any ▾

Genre:

Filter

Clear

Εικόνα 23 – cleared filter

## 6 Android Εφαρμογή

Σε αυτό το κεφάλαιο γίνεται περιγραφή και παρουσίαση της Android εφαρμογή για σύστασης ταινιών

### 6.1 Περιγραφή

Η Android εφαρμογή παρέχει ένα μεγαλύτερο αριθμό «δραστηριοτήτων» στον χρήστη. Είναι τελείως ανεξάρτητη από εξωτερικές πηγές αφού δίνεται η δυνατότητα στο χρήστη να βαθμολογήσει ταινίες αλλά και να διαχειριστεί τις βαθμολογίες του εντός αυτής. Η εφαρμογή δίνει την δυνατότητα στον χρήστη:

- Να βαθμολογήσει ταινίες (είτε τυχαίες ταινίες είτε της επιλογής του)
- Να κάνει Import βαθμολογίες του από εξωτερική πηγή
- Να διαχειριστεί τις υπάρχουσες βαθμολογίες του (Διαγραφή ή αλλαγή μίας βαθμολογίας)
- Να λάβει recommendation για ταινίες (Είτε ταινίες που του προτείνει ο αλγόριθμος είτε τις επιλογής του)

Επίσης υπάρχουν και οι λειτουργίες:

- Watchlist (όπου αποθηκεύονται οι ταινίες που ο χρήστης ενδιαφέρεται να παρακολουθήσει στο μέλλον)
- Settings (Data Saving ώστε να μην κατεβαίνουν τα poster των εικόνων για εξοικονόμηση δεδομένων και δυνατότητα επιλογής της πηγής των βαθμολογιών για την σύσταση πληροφορίας)

### 6.2 User Interface

Η διεπαφή της εφαρμογής αποτελείται από 2 κύρια μέρη

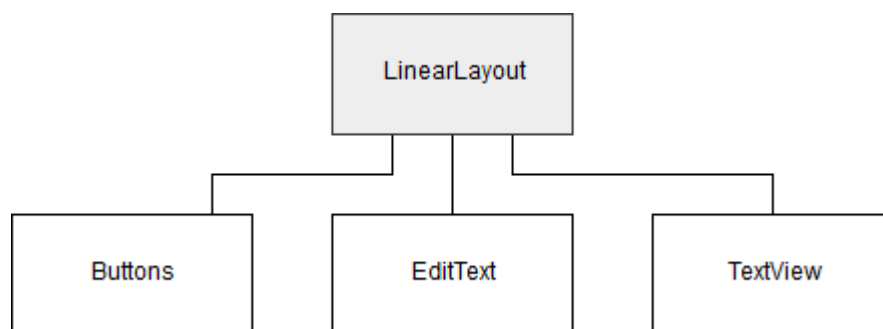
- Τις οθόνες πλοήγησης (Navigation)
- και τις οθόνες παρουσίασης πληροφορίας.

#### 6.2.1 Οθόνες πλοήγησης

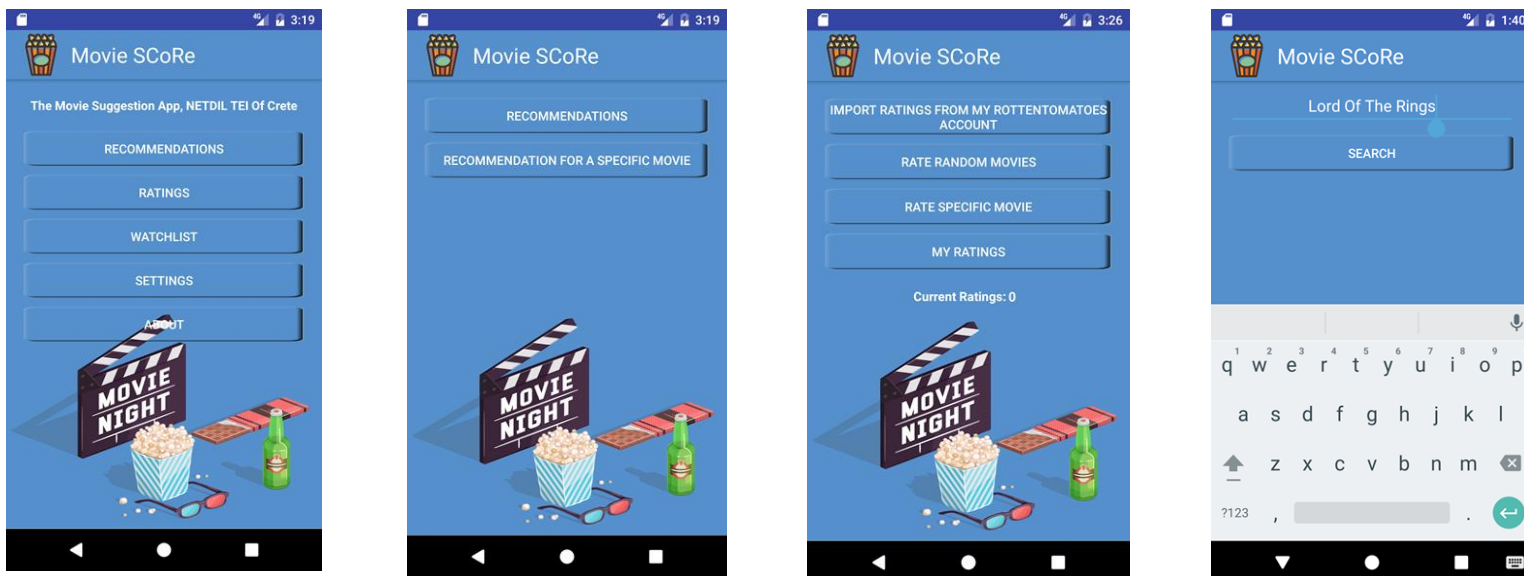
Υπάρχουν 4 οθόνες πλοήγησης

- Η αρχική οθόνη της εφαρμογής
- Η οθόνη για πλοήγηση σε recommendation
- Η οθόνη πλοήγησης για βαθμολόγηση ταινιών
- Οι οθόνες αναζήτησης.

Οι οποίες αποτελούνται από ένα LinearLayout για κάθετη τοποθέτηση των Views/Components. Τα views αυτά μπορεί να είναι Buttons, TextViews και EditTexts.



Εικόνα 24 – Ιεραρχία Layout οθονών πλοήγησης



Εικόνα 25 - Οθόνες Πλοήγησης

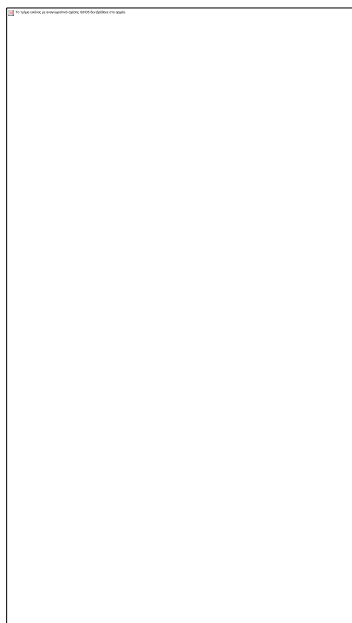
### 6.2.2 Οθόνες παρουσίασης πληροφορίας

Υπάρχουν 2 κατηγορίες οθονών παρουσίασης πληροφορίας. Στην πρώτη η ταινίες εμφανίζονται η μια κάτω από την άλλη και την συναντάμε στις δραστηριότητες:

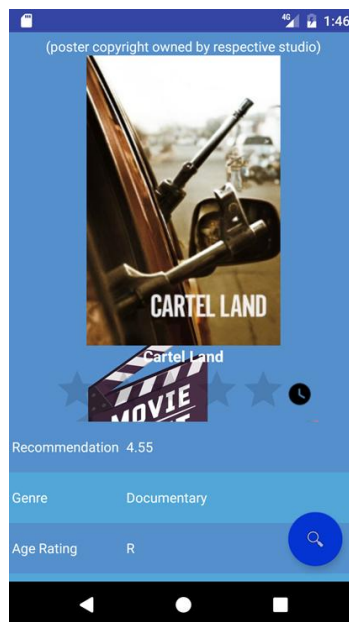
- Βαθμολόγησης ταινιών
- Διαχείρισης βαθμολογιών
- Watchlist
- και στο πρώτο μέρος της σύστασης συγκεκριμένων ταινιών

Ενώ στην δεύτερη η κάθε ταινία παρουσιάζεται σε μία (πλήρης) οθόνη και την συναντάμε στις δραστηριότητες:

- του απλού recommendation όπου η πλοήγηση από την μία ταινία στην άλλη γίνεται μέσω swipe
- του specific recommendation

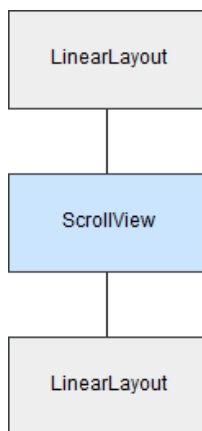


Εικόνα 26 - Οθόνη κάθετης παρουσίασης



Εικόνα 27- Πλήρης οθόνη παρουσίασης

Η ιεραρχία των περιεχομένων στις οθόνες κάθετης παρουσίασης πληροφορίας απεικονίζεται στην παρακάτω εικόνα:

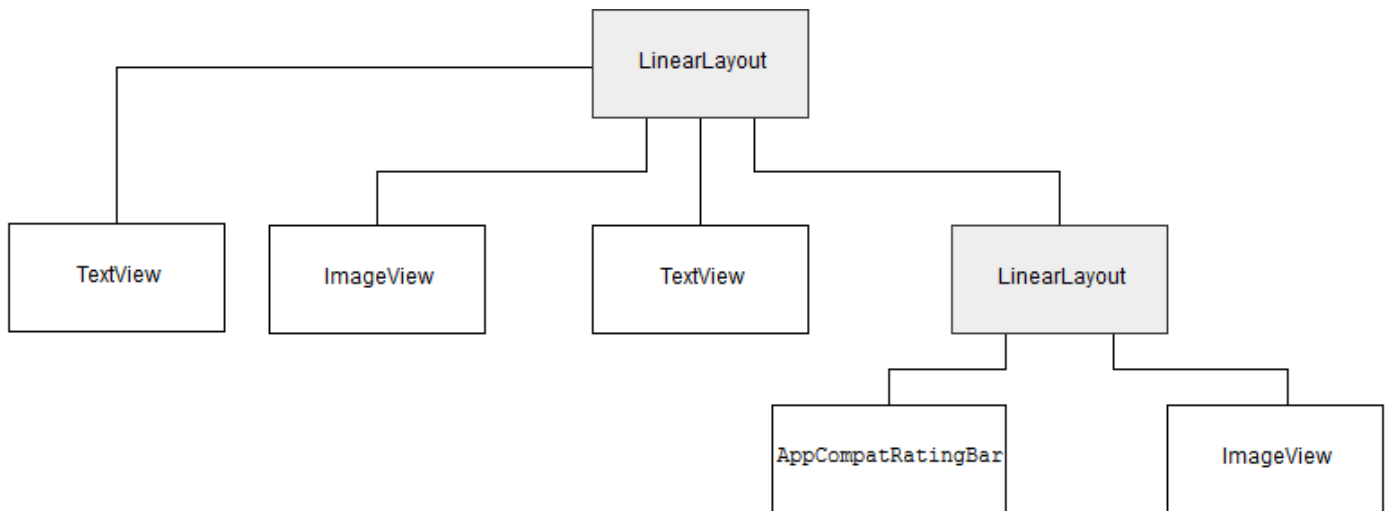


Εικόνα 28 – Ιεραρχία Layout κάθετης παρουσίασης πληροφορίας

Σε αυτό το layout εμφανίζονται ταινίες (εικόνα, όνομα, rating bar, εικόνα) η μία κάτω από την άλλη. Επειδή το σύνολο των αντικειμένων που απαρτίζουν μία ταινία

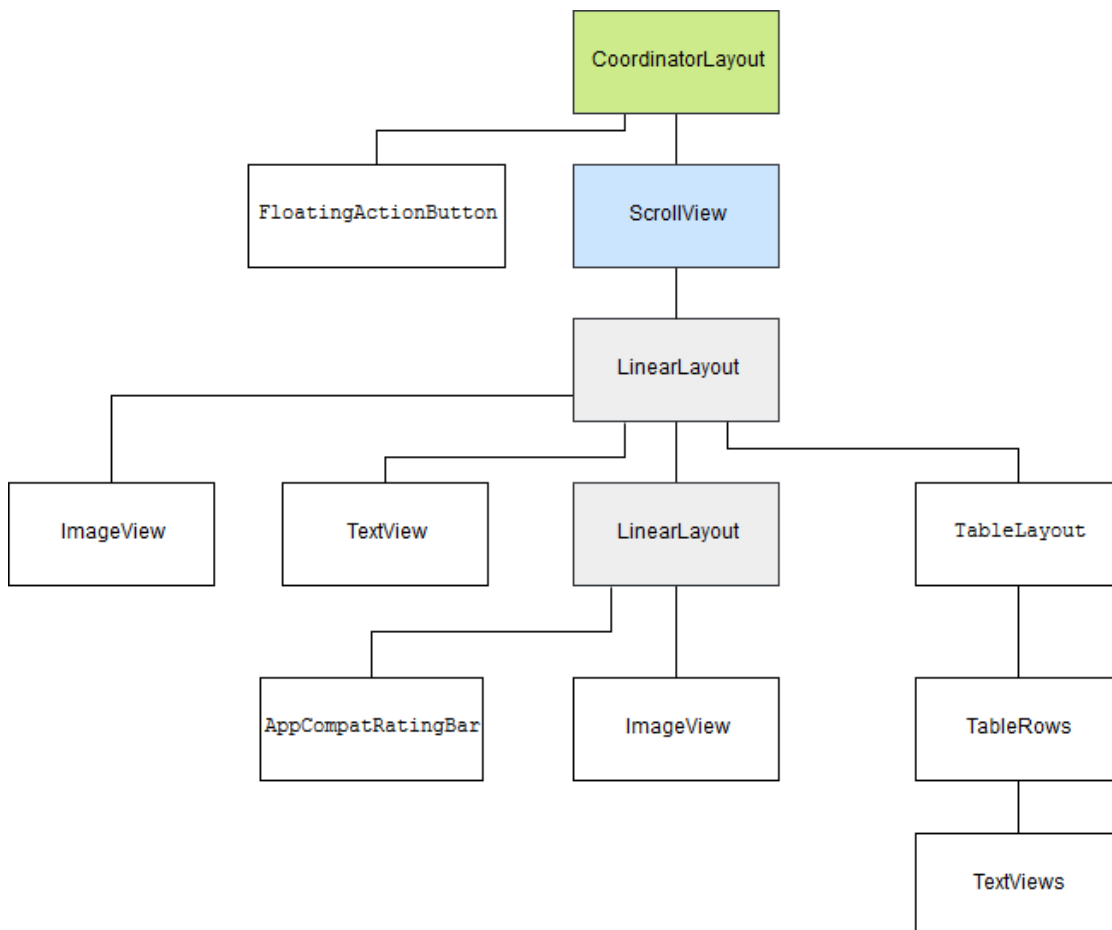
- Credits
- Poster
- Όνομα ταινίας
- Rating Bar
- Delete rating icon

χρησιμοποιούνται κατά κόρον, δημιουργήθηκε ένα layout που περιλαμβάνει όλα αυτά τα αντικείμενα, το οποίο στην συνέχεια παίρνουμε με την χρήση LayoutInflater και δίνουμε τις κατάλληλες τιμές στα views και το τοποθετούμε στο δεύτερο LinearLayout προγραμματιστικά.



Εικόνα 29 – Movie layout

Όσον αφορά την δεύτερη κατηγορία (μία ταινία ανά οθόνη), η υλοποίηση έγινε με την χρήση `FragmentManager` και `FragmentManagerAdapter` όπου κάθε προαναφερθείσα οθόνη αποτελείται ουσιαστικά από ένα `Fragment`. Το Layout κάθε `Fragment` περιγράφεται στο παρακάτω διάγραμμα:



Εικόνα 30 – Fragment Layout

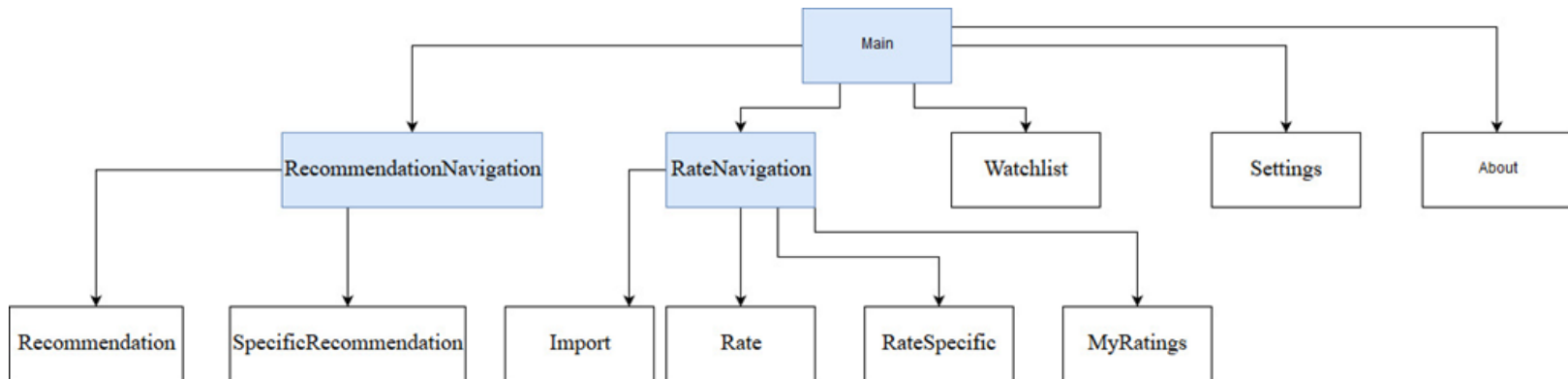
Στην περίπτωση της σύστασης πληροφορίας για συγκεκριμένη ταινία χρησιμοποιείται παρόμοιο layout με την μόνη διαφορά ότι δεν υπάρχει το CoordinatorLayout και το FloatingActionButton

Στον παρακάτω πίνακα γίνεται αντιστοιχία των οθονών της εφαρμογής με τα ονόματα που θα χρησιμοποιηθούν αντί αυτών για ευκολία στο flowchart και τις περιγραφές λειτουργιών της εφαρμογής.

Οθόνη	Όνομα
Αρχική οθόνη της εφαρμογής	Main
Οθόνη για πλοήγηση σε recommendation	RecommendationNavigation
Οθόνη για πλοήγηση σε βαθμολόγηση ταινιών	RateNavigation
Οθόνη βαθμολόγησης ταινιών	Rate
Οθόνη βαθμολόγησης συγκεκριμένης ταινίας	RateSpecific
Οθόνη προβολής και επεξεργασίας βαθμολογιών	MyRatings
Οθόνη εισαγωγής (import) ταινιών	Import
Οθόνη προβολής σύστασης πληροφορίας	Recommendation
Οθόνη φιλτραρίσματος των ταινιών	Filter
Οθόνη προβολής σύστασης πληροφορίας για συγκεκριμένη ταινία	SpecificRecommendation
Οθόνη Settings	Settings
Οθόνη Watchlist	Watchlist
Οθόνη αναζήτησης ταινίας	Search

Πίνακας 3 – Συνοτμεύσεις ονομάτων οθονών

Ακολουθεί το flowchart της εφαρμογής. Τα elements με μπλε χρώμα είναι οι οθόνες πλοήγησης ενώ με άσπρο χρώμα είναι οι οθόνες παρουσίασης πληροφορίας.



Εικόνα 31- Flowchart εφαρμογής

### 6.3 Επικοινωνία με την βάση

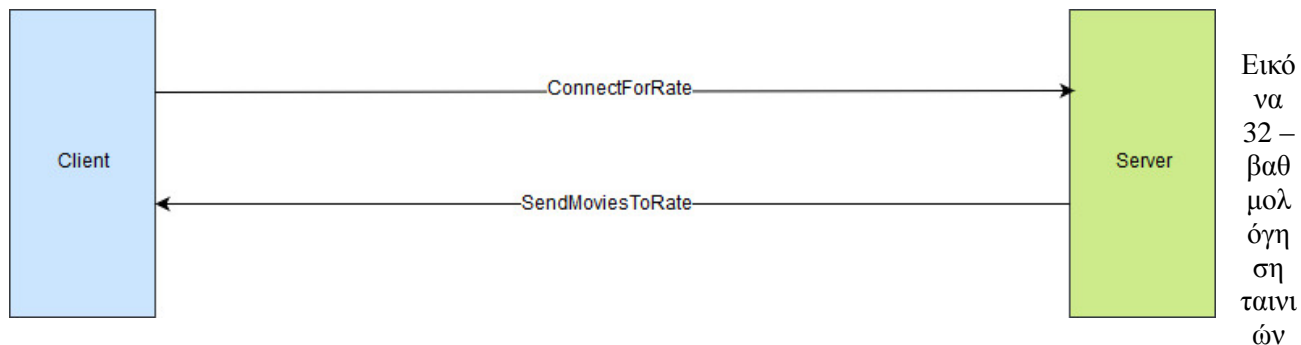
Η εφαρμογή είναι κατανεμημένη, οι μόνες πληροφορίες που αποθηκεύονται στο κινητό είναι:

- οι βαθμολογίες που χρήστη
- οι ταινίες που έχει τοποθετήσει στο Watchlist

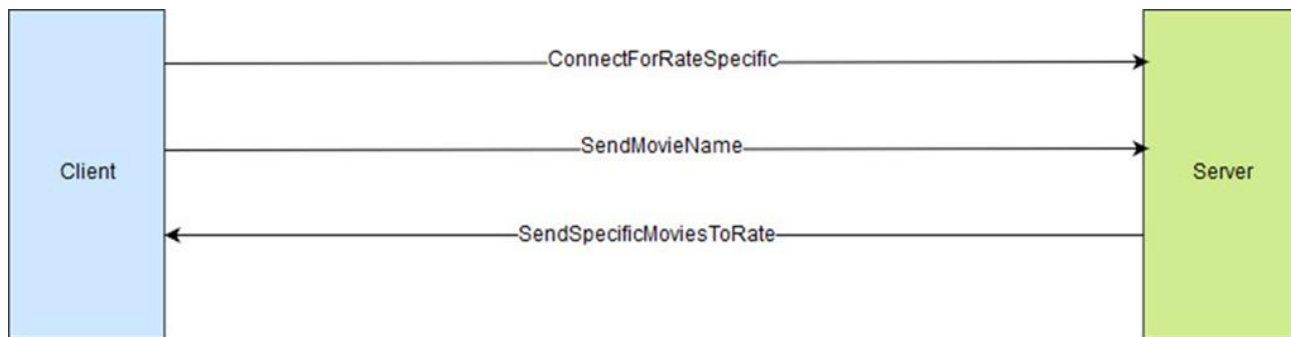
- το ιστορικό αναζήτησης
- τα προφίλ των χρηστών που έχουν χρησιμοποιηθεί για import.

Για τον λόγο αυτό δημιουργήθηκε ένα Java πρόγραμμα από την πλευρά του Server, ενώ στο Client κομμάτι χρησιμοποιήθηκε η κλάση AsyncTask του Android. Ακολουθούν διαγράμματα και επεξήγηση τις Client – Server επικοινωνίας.

Στο κομμάτι της **βαθμολόγησης** ταινιών ο Client στέλνει ένα μήνυμα στον Server δηλώνοντας το λόγο που άνοιξε το connection, στην συνέχεια ο Server επιλέγει τυχαία 60 δημοφιλείς ταινίες και τις επιστρέφει στον Client.

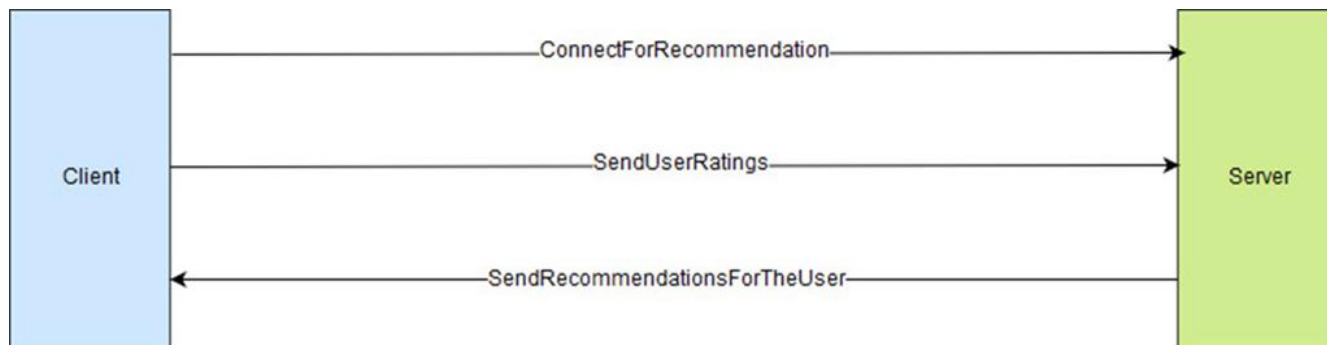


Στην περίπτωση της **βαθμολόγησης μίας συγκεκριμένης ταινίας**, αρχικά ο Client δηλώνει και πάλι το λόγο που άνοιξε το connection, στην συνέχεια στέλνει το όνομα της ταινίας στον Server και αυτός επιστρέφει το πολύ 60 ταινίες των οποίων το όνομα τους μοιάζει με αυτό που έλαβε.



Εικόνα 33 – βαθμολόγηση συγκεκριμένης ταινίας

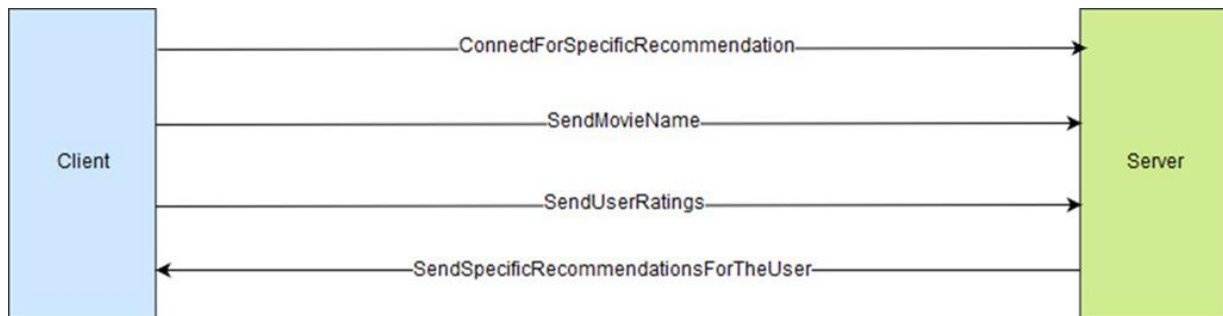
Στη περίπτωση της **σύστασης πληροφορίας** και πάλι αρχικά ο Client στέλνει ένα μήνυμα στον Server δηλώνοντας το λόγο που άνοιξε το connection, στην συνέχεια στέλνει τις βαθμολογίες του χρήστη, ο Server τρέχει τον αλγόριθμο επιλέγοντας τις ταινίες που θα προτείνει στον χρήστη, τις οποίες και στέλνει στον Client.



Εικόνα 34 – Recommendation



Τέλος, για **σύσταση πληροφορίας σε συγκεκριμένη ταινία** αρχικά ο Client στέλνει στον Server το όνομα αυτής και στην συνέχεια τις βαθμολογίες του χρήστη, αυτός επιλέγει τις ταινίες των οποίων το όνομα μοιάζει με αυτό που έδωσε ο χρήστης και επιστρέφει στον Client την πρόβλεψη μόνο για αυτές



Εικόνα 35 – Recommendation για συγκεκριμένη ταινία

## 6.4 Βασικές λειτουργίες της εφαρμογής

Οι Βασικές λειτουργίες της εφαρμογής εφαρμογής είναι

1. Η βαθμολόγηση ταινιών
2. Η διαχείριση βαθμολογιών
3. Η σύσταση ταινιών
4. Watchlist
5. Ρυθμίσεις

### 6.4.1 Βαθμολόγηση ταινιών

Για την βαθμολόγηση ταινιών ο χρήστης έχει 3 επιλογές

1. Να κάνει import τις βαθμολογίες του από εξωτερική πηγή
2. Να βαθμολογήσει τυχαίες ταινίες
3. Να βαθμολογήσει ταινίες της επιλογής του

Στην περίπτωση του import:

- Ο χρήστης δίνει το id του προφίλ του και με την χρήση Jsoup παίρνονται οι βαθμολογίες του οι οποίες και αποθηκεύονται στο imported\_ratings αρχείο.

- Οι βαθμολογίες που έχουν γίνει import δεν μπορούν να διαγραφούν ή να αλλάξουν εντός της εφαρμογής. Εάν ο χρήστης επιθυμεί να τις import ώστε να γίνει override το τροποποιήσει θα πρέπει να ξανακάνει αρχείο.

-



### Εικόνα 36 – Import

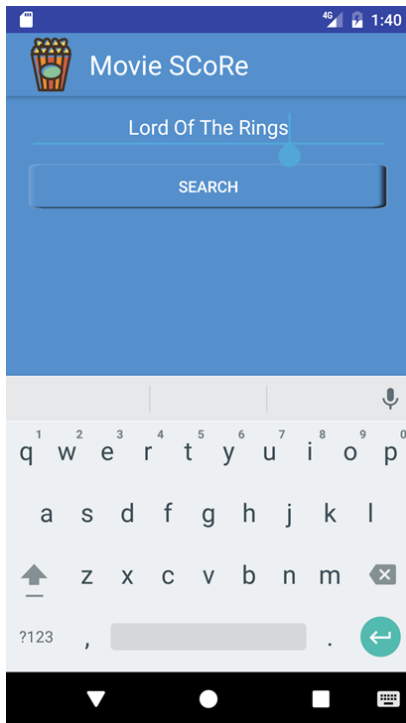
Στην περίπτωση της βαθμολόγησης τυχαίων ταινιών

- Αρχικά γίνεται μια σύνδεση με τον Server ο οποίος επιλέγει τυχαία 60 δημοφιλείς ταινίες τις οποίες και επιστρέφει στην εφαρμογή.
- Στην συνέχεια ο χρήστης μπορεί να βαθμολογήσει όποια από αυτές επιθυμεί (Οι ταινίες που βαθμολογούνται εντός της εφαρμογής αποθηκεύονται στο αρχείο user\_ratings).
- Επίσης ο χρήστης μπορεί να διαγράψει μια βαθμολογία πατώντας το delete icon που βρίσκεται δίπλα από το Rating Bar



Εικόνα 37 – Rate

Όσον αφορά την βαθμολόγηση συγκεκριμένης ταινίας η λειτουργικότητα είναι η ίδια με αυτήν της βαθμολόγησης τυχαίων ταινιών. Με την μόνη διαφορά είναι ότι ο χρήστης αρχικά πληκτρολογεί το όνομα της ταινίας που θέλει να βαθμολογήσει. Αυτό στέλνεται στον Server ο οποίος επιστρέφει μόνο ταινίες των οποίων το όνομα μοιάζει με αυτό που έδωσε ο χρήστης. Στις ακόλουθες εικόνες παρουσιάζονται οι οθόνες Search & RateSpecific στην περίπτωση που ο χρήστης θέλει να βαθμολογήσει κάποια από τις ταινίες ‘Lord Of The Rings’



Εικόνα 38 – Search



Εικόνα 39 – RateSpecific

### 6.4.2 Διαχείριση βαθμολογιών

Σε αυτή την οθόνη δίνεται η δυνατότητα στον χρήστη να διαχειριστεί (διαγράψει ή αλλάξει) τις βαθμολογίες του.

- Η διαγραφή μιας ταινίας γίνεται κάνοντας swipe δεξιά ή αριστερά πάνω της
- Ενώ η αλλαγή βαθμολογίας γίνεται με την χρήση του Rating bar



Εικόνα 40- Διαχείριση βαθμολογιών

### 6.4.3 Σύσταση πληροφορίας

Για λήψη σύστασης πληροφορίας ο χρήστης έχει δύο επιλογές

1. Recommendation

## 2. Specific Recommendation

Στο απλό **recommendation** κάθε προτεινόμενη ταινία προβάλλεται σε πλήρη οθόνη και ο χρήστης μεταβαίνει από την μία στην άλλη κάνοντας swiipe δεξιά ή αριστερά. Εκτός από την βαθμολόγηση μίας ταινίας που έχουμε δει έως τώρα, από εδώ δίνεται η δυνατότητα στον χρήστη

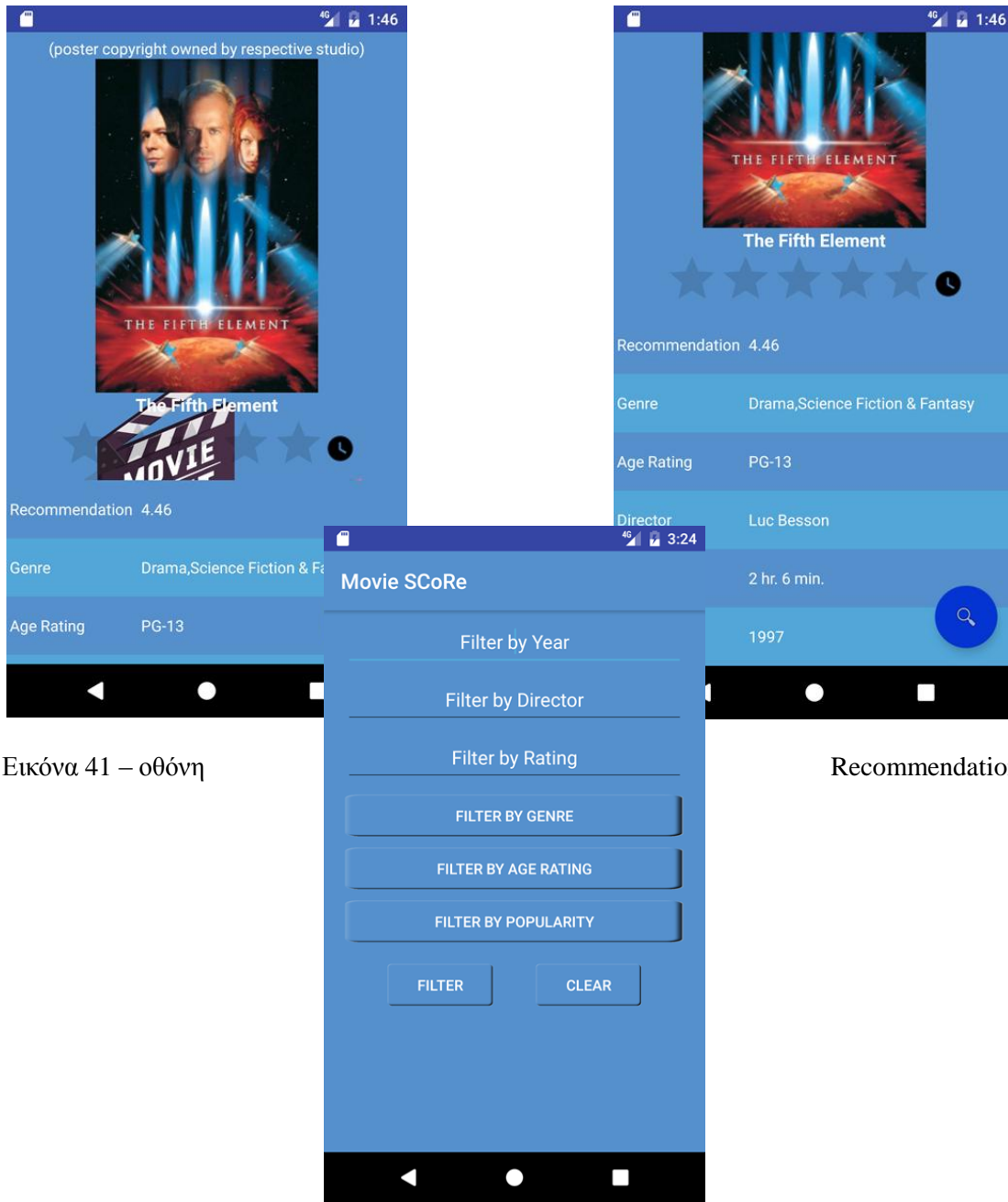
- Να τοποθετήσει μία ταινία στο Watchlist πατώντας στο εικονίδιο με το μαύρο ρολόι
- Να δει πληροφορίες για την ταινία
- Να φιλτράρει τα αποτελέσματα πατώντας το μπλε κουμπί με τον μεγεθυντικό φακό που βρίσκεται στο κάτω δεξί μέρος της οθόνης.

Η οθόνη Recommendation παρουσιάζεται στην εικόνα 41.

Όπως προαναφέρθηκε ο χρήστης έχει την δυνατότητα φιλτραρίσματος των ταινιών χρησιμοποιώντας ένα ή συνδυασμό των ακόλουθων κριτηρίων:

- Έτος
- Σκηνοθέτης
- Recommendation Rating
- Genre
- Age Rating
- Popularity

Ενώ υπάρχει και η δυνατότητα καθαρισμού του φίλτρου. Η οθόνη Filter παρουσιάζεται στην εικόνα 42.



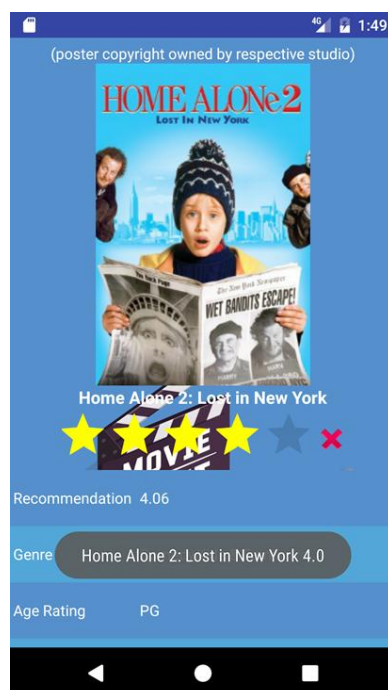
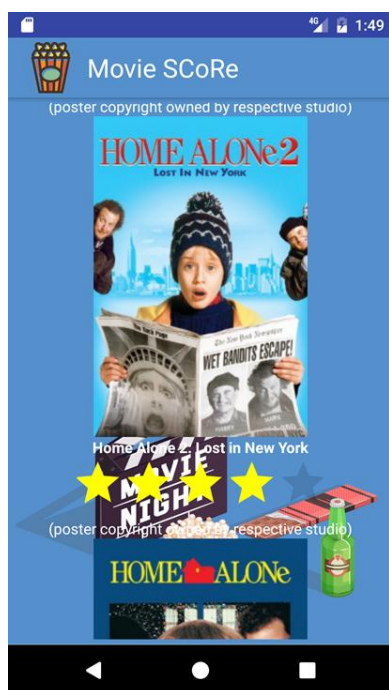
Εικόνα 41 – οθόνη

Recommendation

Εικόνα 42 - Οθόνη Filter

Στην περίπτωση του **Specific Recommendation** ο χρήστης λαμβάνει σύσταση για μία ταινία της επιλογής του. Η διαδικασία είναι παρόμοια με την βαθμολόγηση συγκεκριμένης ταινίας. Οι διαφορές που υπάρχουν είναι ότι:

1. Υπάρχει μια ενδιάμεση οθόνη στην οποία ο χρήστης επιλέγει την ταινία που επιθυμεί όπου τα αστεράκια πλέον αντιπροσωπεύουν το Recommendation Rating και δεν μπορεί να γίνει βαθμολόγηση μέσω αυτών
  2. Κάνοντας κλικ πάνω σε μία ταινία μεταβαίνει στην ίδια οθόνη με το απλό Recommendation με τη μόνη διαφορά ότι δεν υπάρχει η επιλογή φιλτραρίσματος.
- Ακολουθούν Screenshots της οθόνης SpecificRecommendation στην περίπτωση που ο χρήστης θέλει να λάβει σύσταση πληροφορίας για την ταινία “Home Alone”.



Εικόνα 43 – SpecificRecommendation (βήμα 1) Εικόνα 44 – SpecificRecommendation (βήμα 2)

## 6.4.4 Watchlist

Στο Watchlist ο χρήστης αποθηκεύει τις ταινίες που του προτάθηκαν και θέλει να παρακολουθήσει στο μέλλον. Η λειτουργικότητα του είναι παρόμοια με το MyRatings, με τις μόνες διαφορές να είναι ότι:

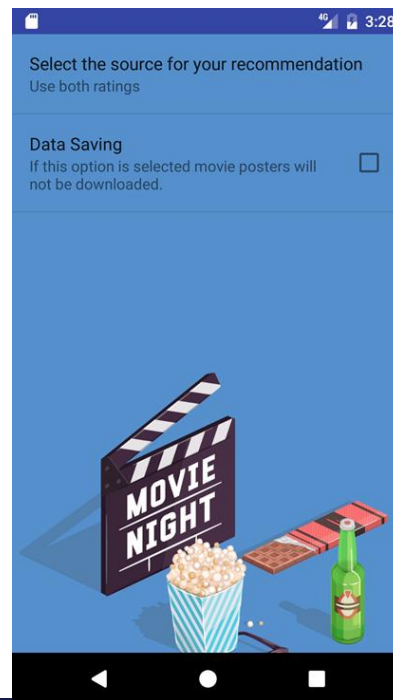
1. όταν ο χρήστης βαθμολογήσει μία ταινία που υπάρχει στο Watchlist, αυτή διαγράφεται από αυτό και τοποθετείται στο αρχείο με τις βαθμολογίες του χρήστη
  2. τα αστεράκια δεν περιέχουν βαθμολογία επειδή ο χρήστης δεν έχει βαθμολογήσει τις ταινίες αυτές.
- Η οθόνη WatchList παρουσιάζεται στην Εικόνα 45.

## 6.4.5 Settings

Στις ρυθμίσεις ο χρήστης έχει τις δυνατότητες:

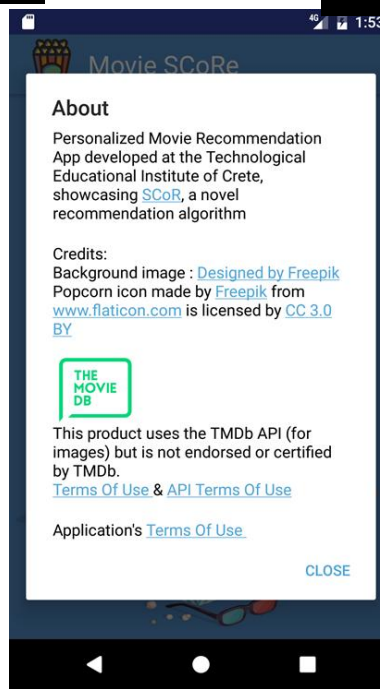
- εξοικονόμησης δεδομένων (Data Save)
- επιλογής της πηγής των βαθμολογιών του χρήστη. Οι 3 επιλογές για την πηγή των βαθμολογιών είναι:
  1. Use both ratings (όπου επιλέγονται και οι βαθμολογίες που έχει κάνει import αλλά και οι τοπικές)
  2. Use local Ratings (όπου επιλέγονται μόνο οι βαθμολογίες που έκανε εντός της εφαρμογής)
  3. Use imported Ratings (όπου επιλέγονται μόνο οι βαθμολογίες που έχουν γίνει import)

Η οθόνη Settings παρουσιάζεται στην εικόνα 47.



Εικόνα 45 –  
Εικόνα 46 – Settings

Τέλος, στο About υπάρχει μια μικρή  
τα Credits για τις εικόνες των ταινιών



Watchlist

περιγραφή της εφαρμογής αλλά και  
& της εφαρμογής.

Εικόνα 47 - About

## 7 Αποτελέσματα

Τα αποτελέσματα της πτυχιακής εργασίας ήταν:

1. Η δημιουργία ενός αρκετά μεγάλου dataset με βαθμολογίες χρηστών σε ταινίες αλλά και πληροφορίες ταινιών
2. Η δημιουργία δύο εφαρμογών σύστασης ταινιών στους χρήστες σύμφωνα με τις προτιμήσεις τους.  
Η Android εφαρμογή έχει ανέβει στο Play Store  
(<https://play.google.com/store/apps/details?id=com.netdil.recapp>).

### 7.1 Μελλοντική εργασία και επεκτάσεις

Προς το παρόν οι δύο εφαρμογές είναι ανεξάρτητες μεταξύ τους, στην Android εφαρμογή όλες οι βαθμολογίες του χρήστη αποθηκεύονται τοπικά στο κινητό του ενώ στην Web εφαρμογή ο χρήστης δεν έχει την δυνατότητα βαθμολόγησης ταινιών. Στο μέλλον ίσως θα ήταν καλό να δημιουργηθεί ένα προφίλ για κάθε χρήστη όπου πλέον οι βαθμολογίες του & οι ταινίες που θέλει να παρακολουθήσει αργότερα (Watchlist) να μην αποθηκεύονται πλέον τοπικά αλλά σε κάποιον Server έτσι ώστε να έχει πρόσβαση σε αυτές είτε από άλλη Android συσκευή είτε μέσω της Web εφαρμογής. Έτσι θα μπορούσαμε να δημιουργήσουμε ένα νέο μεγαλύτερο dataset με βαθμολογίες χρηστών το οποίο θα εμπλουτίζεται συνεχώς. Επιπλέον θα μπορούσε να γίνει συλλογή των δεδομένων χρήσης της εφαρμογής, για παράδειγμα ποιές κατηγορίες ταινιών επιλέγει κυρίως ο χρήστης όταν κάνει αναζήτηση, αν προτιμάει δημοφιλείς ή άγνωστες ταινίες κλπ. Έτσι μπορεί να γίνει και μία τροποποίηση του αλγορίθμου σύστασης πληροφορίας ώστε να λαμβάνει υπ' όψιν του αυτά τα δεδομένα με σκοπό να βελτιωθούν οι συστάσεις του. Μια άλλη επέκταση που θα μπορούσε να υλοποιηθεί είναι ο εμπλουτισμός της Web εφαρμογής ώστε να προσφέρει ακριβώς τις ίδιες δραστηριότητες στους χρήστες με την Android εφαρμογή. Τέλος, θα μπορούσε να γίνει κάποια βελτίωση στο User Interface της Android εφαρμογής βάση των αρχών του Material Design.



## Βιβλιογραφία

- [1] <http://www.inf.unibz.it/~ricci/papers/intro-rec-sys-handbook.pdf>
- [2] <http://facweb.cs.depaul.edu/mobasher/classes/ect584/Papers/ContentBasedRS.pdf>
- [3] C.C. Aggarwal, *Recommender Systems: The Textbook*,

## Πηγές από web

- [1] [https://en.wikipedia.org/wiki/Recommender\\_system](https://en.wikipedia.org/wiki/Recommender_system)
- [2] [https://www.w3schools.com/sql/sql\\_intro.asp](https://www.w3schools.com/sql/sql_intro.asp)
- [3] <http://www.javatpoint.com/jsoup-tutorial>
- [4] [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
- [5] <https://el.wikipedia.org/wiki/Java>
- [6] [https://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- [7] [https://en.wikipedia.org/wiki/Collaborative\\_filtering](https://en.wikipedia.org/wiki/Collaborative_filtering)
- [8] <https://grouplens.org/about/projects/>
- [9] [https://en.wikipedia.org/wiki/Netflix\\_Prize](https://en.wikipedia.org/wiki/Netflix_Prize)
- [10] <https://www.quora.com/How-does-Facebooks-People-You-May-Know-work>
- [11] <https://www.facebook.com/help/www/163810437015615>

## Papers

- [1] SCoR: A Synthetic Coordinate based Recommender System. Harris Papadakis, Costas Panagiotakis, Paraskevi Fragopoulou
- [2] Impact of Recommender Systems on Sales Volume and Diversity. Dokyun Lee, Kartik Hosanagar (Thirty Fifth International Conference on Information Systems, Auckland 2014)
- [3] Knowledge-based recommendation systems: A survey. Sarah Bouraga, Ivan Jureta, Stephane Faulkner, Caroline Herssens
- [4] Daniel Fleder and Kartik Hosanagar, Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity, *Management Science* Vol. 55, No. 5 (May, 2009), pp. 697-712