

Ατομική Διπλωματική Εργασία

**ΜΕΛΕΤΗ ΤΩΝ CONTEXT-AWARE RECOMMENDER  
ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΔΙΑΔΥΚΤΙΑΚΗΣ  
ΕΦΑΡΜΟΓΗΣ MOVIE RECOMMENDER**

Ανδρέας Βιολάντης

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**



**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Μάιος 2017**

# **ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**

## **ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Μελέτη των context-aware recommender συστημάτων και υλοποίηση  
διαδικτυακής εφαρμογής movie recommender**

**Ανδρέας Βιολάντης**

Επιβλέπων Καθηγητής

Γιώργος Παπαδόπουλος

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των  
απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του  
Πανεπιστημίου Κύπρου

Μάιος 2017

## Ευχαριστίες

Καταρχάς θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Δρ. Γιώργο Παπαδόπουλο, καθηγητή του τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου, που μου ανάθεσε αυτό το θέμα και μελετώντας το κατάφερα να εμπλουτίσω τις γνώσεις μου όσο αφορά στον τομέα των Context-Aware Recommendation συστημάτων, κάτι που αναμφίβολα θα μου φανεί χρήσιμο στη μελλοντική μου σταδιοδρομία.

Ακολούθως, θα ήθελα να ευχαριστήσω το κ. Χρίστο Μεττούρη, ο οποίος ήταν πάντα πρόθυμος να με βοηθά και να με συμβουλεύει με οποιαδήποτε προβλήματα και αν αντιμετώπιζα κατά την έρευνα και εκπόνηση της διπλωματικής μου εργασίας.

Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου και ιδιαιτέρως τους γονείς μου, που με στήριζαν τα τελευταία τέσσερα χρόνια με κάθε τρόπο έτσι ώστε να καταφέρω να ολοκληρώσω τις σπουδές μου και να φτάσω στο σημείο που βρίσκομαι.

## Περίληψη

Η ατομική διπλωματική μου εργασία αποτελείται από δύο μέρη.

Το πρώτο μέρος είναι έρευνα που έγινε όσο αφορά τα Συστήματα Συστάσεων με επίγνωση συγκεκριμένων (Context-Aware Recommender Systems). Συγκεκριμένα μελετήθηκαν πολλά είδη των Συστημάτων Συστάσεων (Recommender Systems) που υπάρχουν σήμερα και το τρόπο που λειτουργούν, βγάζοντας κάποια συμπεράσματα για το καθένα. Συμπεράσματα τα οποία βοηθούν στο να καθορίσουμε ποιο Σύστημα Συστάσεων πρέπει να εφαρμοστεί σε κάθε περίπτωση που έχουμε στη διάθεσή μας ένα σετ δεδομένων. Ποιο θα μας δώσει πιο ακριβή αποτελέσματα σε κάθε περίπτωση και ποιο θα είναι το πιο αποδοτικό. Επίσης μέρος της έρευνας ήταν να δούμε εφαρμογές Συστημάτων Συστάσεων σε πραγματικές εφαρμογές και πως η συγκεκριμένη πληροφορία παρουσιάζεται στο χρήστη. Κάνοντας αυτή τη μελέτη καταλήξαμε στο συμπέρασμα ότι τα Συστήματα Συστάσεων με επίγνωση συγκεκριμένων είναι ζωτικής σημασίας για εφαρμογές που σκοπός τους είναι να πουλήσουν στο χρήστη προϊόντα (π.χ. Amazon, E-bay) αλλά όχι μόνο αφού χρησιμοποιούνται από εφαρμογές που θέλουν να διαφημίσουν ταινίες (π.χ. IMDB (Internet Movies Data Base)).

Το δεύτερο μέρος ήταν η εφαρμογή ενός Συστήματος Συστάσεων με επίγνωση συγκεκριμένων σε πραγματικά δεδομένα. Αφού έγινε η επιλογή των δεδομένων, όπου συνδύαζαν βαθμολογίες χρηστών σε ταινίες οι οποίες έγιναν συλλέγοντας παράλληλα και πληροφορίες για το συγκεκριμένο, υλοποιήσαμε μία διαδικτυακή εφαρμογή όπου ο κάθε χρήστης να μπορεί να βλέπει συστάσεις που έχουν προταθεί σε αυτόν ανά πάσα στιγμή.

Η επιλογή του συγκεκριμένου σετ δεδομένων έγινε γιατί είχε ικανοποιητικό αριθμό από βαθμολογίες μεταξύ χρηστών-ταινιών και περιείχε αρκετές πληροφορίες τόσο για τους χρήστες όσο και για το αντικείμενο, που στη περίπτωση μας είναι οι ταινίες. Επίσης περιείχαν συγκεκριμένο όπως για παράδειγμα το πότε κάθε χρήστης βαθμολόγησε μια ταινία. Άρα τα συγκεκριμένα δεδομένα μας παρείχαν ευελιξία ως προς το είδος του συστήματος συστάσεων θα εφαρμοζόταν.

Σκοπός της υλοποίησης ήταν να γίνει μια εφαρμογή που να μπορεί να εφαρμοστεί και σε πραγματικά σενάρια και να είναι χρήσιμη σε πραγματικούς χρήστες. Για το λόγο αυτό έγινε η επιλογή της εφαρμογής να είναι διαδικτυακή έτσι ώστε να είναι προσβάσιμη από όλους και από κάθε συσκευή. Κύριο μέλημα ήταν η απλότητα ως προς τη διεπαφή έτσι ώστε να καλύπτει βασικούς κανόνες αλληλεπίδρασης ανθρώπου-υπολογιστή όπως ευχρηστία και ευκολία εκμάθησης.

## Περιεχόμενα

|                   |                                                                        |           |
|-------------------|------------------------------------------------------------------------|-----------|
| <b>Κεφάλαιο 1</b> | <b>Εισαγωγή.....</b>                                                   | <b>1</b>  |
|                   | 1.1 Γενική Εισαγωγή στα Συστήματα Συστάσεων                            | 1         |
|                   | 1.2 Γενική Εισαγωγή στα Συστήματα Συστάσεων<br>με επίγνωση συγκειμένου | 2         |
|                   | 1.3 Στόχος Διπλωματικής Εργασίας                                       | 2         |
|                   | 1.4 Δομή Διπλωματικής Εργασίας                                         | 3         |
| <br>              |                                                                        |           |
| <b>Κεφάλαιο 2</b> | <b>Περιγραφή προβλήματος και ανασκόπηση Βιβλιογραφίας.....</b>         | <b>5</b>  |
|                   | 2.1 Περιγραφή του προβλήματος                                          | 5         |
|                   | 2.1.1 Υποπροβλήματα και λύσεις υποπροβλημάτων                          | 7         |
|                   | 2.2 Άλλα παρόμοια συστήματα                                            | 8         |
|                   | 2.3 Ανασκόπηση Βιβλιογραφίας                                           | 10        |
| <br>              |                                                                        |           |
| <b>Κεφάλαιο 3</b> | <b>Απαιτούμενη Γνώση Και Τεχνολογίες.....</b>                          | <b>13</b> |
|                   | 3.1 Γενική εισαγωγή για το CarsKit Framework                           | 13        |
|                   | 3.1.1 Αλγόριθμοι του CarsKit Framework                                 | 14        |
|                   | 3.1.2 Προϋποθέσεις και παράμετροι του<br>CarsKit Framework             | 15        |
|                   | 3.1.3 Μορφή αποτελεσμάτων του CarsKit Framework                        | 16        |
|                   | 3.1.4 Γιατί CarsKit Framework                                          | 17        |
|                   | 3.1.5 Γιατί CAMF_CU αλγόριθμος                                         | 17        |
|                   | 3.2 Γενική εισαγωγή στα Web Applications                               | 18        |
|                   | 3.2.1 Γιατί Web Application                                            | 18        |
|                   | 3.3 HTML 5 – CSS 3                                                     | 19        |
|                   | 3.4 JavaScript – JQuery                                                | 20        |
|                   | 3.5 Bootstrap                                                          | 21        |
|                   | 3.6 PHP                                                                | 21        |
|                   | 3.7 MySQL                                                              | 22        |

|                           |                                                                            |           |
|---------------------------|----------------------------------------------------------------------------|-----------|
| <b>Κεφάλαιο 4</b>         | <b>Ανάλυση Απαιτήσεων, Προδιαγραφές .....</b>                              | <b>25</b> |
| 4.1                       | Εισαγωγή στις έννοιες Ανάλυση Απαιτήσεων<br>και Εύρεση Προδιαγραφών        | 25        |
| 4.2                       | Σκοπός Ανάλυσης Απαιτήσεων στη παρούσα εργασία                             | 27        |
| 4.3                       | Απαιτήσεις που εξάχθηκαν                                                   | 27        |
| <b>Κεφάλαιο 5</b>         | <b>Σχεδιασμός συστήματος, υλοποίηση .....</b>                              | <b>30</b> |
| 5.1                       | Actors                                                                     | 30        |
| 5.2                       | Περιγραφή Use Case                                                         | 31        |
| 5.2.1                     | Γενική περίπτωση χρήσης του συστήματος<br>από το χρήστη                    | 31        |
| 5.2.2                     | Γενική περίπτωση χρήσης του συστήματος<br>από τον διχειριστή               | 32        |
| 5.2.3                     | Περίπτωση Χρήσης του See<br>Overall Recommendations                        | 33        |
| 5.2.4                     | Περίπτωση Χρήσης όταν εμφανίζονται τα<br>watchlisted movies                | 33        |
| 5.2.5                     | Περίπτωση Χρήσης όταν θέλει ο χρήστης να βαθμολογήσει<br>μια ταινία        | 34        |
| 5.3                       | Υλοποίηση των λειτουργιών της διαδικτυακής εφαρμογής<br>για το χρήστη      | 35        |
| 5.4                       | Υλοποίηση των λειτουργιών της διαδικτυακής εφαρμογής<br>για το διαχειριστή | 41        |
| <b>ΚΕΦΑΛΑΙΟ 6</b>         | <b>Αξιολόγηση, Συμπεράσματα και Μελλοντική Εργασία .....</b>               | <b>44</b> |
| 6.1                       | Αξιολόγηση Συστήματος                                                      | 44        |
| 6.2                       | Συμπεράσματα                                                               | 49        |
| 6.3                       | Μελλοντική Εργασία                                                         | 51        |
| <b>Βιβλιογραφία .....</b> |                                                                            | <b>52</b> |

|                                 |            |
|---------------------------------|------------|
| <b>Π α ρ ά ρ τ η μ α Α.....</b> | <b>A-1</b> |
|---------------------------------|------------|





# Κεφάλαιο 1

## Εισαγωγή

---

|                                                                       |   |
|-----------------------------------------------------------------------|---|
| 1.1 Γενική Εισαγωγή στα Συστήματα Συστάσεων                           | 1 |
| 1.2 Γενική Εισαγωγή στα Συστήματα Συστάσεων με επίγνωση συγκεκριμένου | 2 |
| 1.3 Στόχος Διπλωματικής Εργασίας                                      | 2 |
| 1.4 Δομή Διπλωματικής Εργασίας                                        | 3 |

---

### 1.1 Γενική Εισαγωγή στα Συστήματα Συστάσεων (Recommender Systems)

Ο όρος Συστήματα Συστάσεων αναφέρεται συνήθως σε μία πλατφόρμα η λογισμικό που κάνει προβλέψεις για τη βαθμολογία (rating) που πρόκειται να δώσει ένας χρήστης σε ένα αντικείμενο[1].

Έχουν γίνει πολύ δημοφιλείς τα τελευταία χρόνια, αφού αυξάνεται αισθητά η εμφάνιση τους. Χρησιμοποιούνται κυρίως σε ιστοσελίδες ή εφαρμογές που πουλούν προϊόντα ή που θέλουν να διαφημίσουν προϊόντα. Κάποια παραδείγματα που χρησιμοποιούνται είναι ταινίες, μουσική, βιβλία, άρθρα κ.α.

Για οποιοδήποτε χρήστη, θα ήταν επιθυμητό να χρησιμοποιεί μια εφαρμογή που κατά την αγορά κάποιου προϊόντος να του προτείνει και άλλα αντίστοιχα προϊόντα σχετικά με αυτό που έχει αγοράσει. Έτσι, σιγά σιγά όλες οι εταιρίες που παρέχουν προϊόντα (items) μέσω κάποιας διαδικτυακής εφαρμογής ενσωματώνουν τέτοια συστήματα σε αυτές για να πουλούν περισσότερο και να μην μειονεκτούν σε σύγκριση με τους ανταγωνιστές τους.

## **1.2 Γενική Εισαγωγή στα Συστήματα Συστάσεων με επίγνωση συγκειμένου**

Ο όρος Επίγνωση Συγκειμένου (Context-Awareness) αναφέρεται στη χρήση συγκειμένου, δηλαδή πληροφοριών που σχετίζονται με μια οντότητα. Με τον όρο οντότητα μπορούμε να χαρακτηρίσουμε ανθρώπους, αντικείμενα (προϊόντα), τοποθεσίες, το χρόνο και οτιδήποτε άλλο μπορεί να πάρει σαν είσοδο μια συσκευή. Οι πληροφορίες χαρακτηρίζουν τις οντότητες έτσι ώστε να μπορέσει ακολούθως ένας υπολογιστής να κάνει συσχετίσεις μεταξύ των οντοτήτων και να βγάλει κάποια γενικά συμπεράσματα[2].

Ο τομέας των Συστήματα Συστάσεων με επίγνωση συγκειμένου (Context-Aware Recommender Systems) είναι ένας τομέας ο οποίος απασχολεί πολύ τους επιστήμονες της πληροφορικής τις τελευταίες δεκαετίες.

Ο ευρύτερος όρος Συστήματα Συστάσεων με επίγνωση συγκειμένου, είναι ένας πολύ δημοφιλής τομέας της πληροφορικής τις τελευταίες δεκαετίες. Η διαφορά με τα παραδοσιακά Συστήματα Συστάσεων είναι ότι έρχεται να προστεθεί το συγκείμενο στη προσπάθεια να γίνει η πρόβλεψη του αντικειμένου που θα προτιμήσει ο χρήστης. Είναι η μέθοδος με την οποία ένα σύστημα αξιοποιεί την πληροφορία έτσι ώστε να προτίνει στο χρήστη items (προϊόντα).

Άρα καταλαβαίνουμε πως τα Συστήματα Συστάσεων με επίγνωση συγκειμένου είναι μια βελτιωμένη έκδοση των απλών Συστημάτων Συστάσεων γιατί αξιοποιούν περισσότερη πληροφορία για να κάνουν τις προβλέψεις τους και αυτό συνεπάγεται ότι έχουν μεγαλύτερα ποσοστά επιτυχίας.

## **1.3 Στόχος Διπλωματικής Εργασίας**

Στόχος της διπλωματικής εργασίας είναι αρχικά να μελετηθούν τα πιο σημαντικά είδη των Συστημάτων Συστάσεων και Συστήματα Συστάσεων με επίγνωση συγκειμένου και ακολούθως να υλοποιηθεί μια διαδικτυακή εφαρμογή σε ένα πραγματικό σετ

δεδομένων ταινιών, που να παρέχει συστάσεις σε χρήστες. Η διαδικτυακή εφαρμογή έχει ως σκοπό να εξυπηρετεί κάθε χρήστη που είναι εγγεγραμμένος στην ιστοσελίδα, παρουσιάζοντας ανά πάσα στιγμή προτεινόμενες ταινίες γι' αυτόν οι οποίες να είναι και προσωποποιημένες (personalized). Επίσης ο ίδιος ο χρήστης βαθμολογώντας ταινίες μέσα από την εφαρμογή θα διαμορφώνει συνεχώς το profile του έτσι ώστε να βελτιώνονται οι συστάσεις προς αυτόν σε βάθος χρόνου.

Ο λόγος που επιλέξαμε αυτό το σετ ταινιών μέσα από πολλά πιθανά ήταν γιατί περιείχε μεγάλο αριθμό από δεδομένα. Συγκεκριμένα περιείχε γύρω στους 1000 χρήστες, γύρω στις 1700 ταινίες και 100000 ratings. Άρα θα είχα ένα πολύ αξιόλογο δείγμα για να κάνω σε μεταγενέστερο στάδιο τις συστάσεις. Κάτι άλλο που είχε το συγκεκριμένο σετ δεδομένων που θα μου ήταν χρήσιμο, είναι ότι στα χαρακτηριστικά κάθε item (ταινίας) συμπεριλαμβανόταν και ένας εξωτερικός σύνδεσμος σχετικός με την ταινία. Ο συγκεκριμένος σύνδεσμος μας βοήθησε κατά την δημιουργία της εφαρμογής να κατεβάσω την εικόνα κάθε ταινίας και να μπορέσω να κάνω την εφαρμογή μου πιο φιλική προς το χρήστη χρησιμοποιώντας τις συγκεκριμένες εικόνες.

#### **1.4 Δομή Διπλωματικής Εργασίας**

Πιο κάτω θα γίνει αναφορά στη δομή της διπλωματικής εργασίας.

#### **Κεφάλαιο 1. Εισαγωγή**

Στο πρώτο κεφάλαιο γίνεται αναφορά στα Συστήματα Συστάσεων και στα Συστήματα Συστάσεων με επίγνωση σεικειμένου, έτσι ώστε να καταλάβει ο χρήστης περί τίνος πρόκειται και που εφαρμόζονται. Επίσης γίνεται μια συνοπτική αναφορά στους στόχους της διπλωματικής εργασίας.

#### **Κεφάλαιο 2. Περιγραφή προβλήματος και ανασκόπηση Βιβλιογραφίας**

Στο δεύτερο κεφάλαιο περιγράφεται εκτενώς το γενικό πρόβλημα και υπό προβλήματα που προέκυψαν κατά την υλοποίηση. Γίνεται αναφορά στη μελέτη που έγινε όσο αφορά

τα παρόμοια συστήματα που υπάρχουν στην αγορά και τη βιβλιογραφία που μας οδήγησε στην εκμάθηση των πιο σημαντικών αλγορίθμων τόσο για απλά Συστήματα Συστάσεων όσο και για τα Συστήματα Συστάσεων με επίγνωση συγκεκριμένων.

### **Κεφάλαιο 3. Απαιτούμενη Γνώση Και Τεχνολογίες**

Στο τρίτο κεφάλαιο θα γίνει αναφορά στις τεχνολογίες που επιλέχθηκαν να χρησιμοποιηθούν για την ανάπτυξη της διαδικτυακής εφαρμογής. Διευκρινίζεται ο λόγος που επιλέχθηκαν οι συγκεκριμένες τεχνολογίες και που χρησιμοποιήθηκαν.

### **Κεφάλαιο 4. Ανάλυση Απαιτήσεων, Προδιαγραφές**

Στο τέταρτο κεφάλαιο θα γίνει αναφορά στο τι είναι η φάση ανάλυσης απαιτήσεων και προδιαγραφών και πόσο σημαντική ήταν αυτή η φάση στη διπλωματική μου εργασία. Θα αναφερθούν επίσης οι απαιτήσεις που θα εξαχθούν μέσα από τη συγκεκριμένη φάση.

### **Κεφάλαιο 5. Σχεδιασμός συστήματος, υλοποίηση**

Στο πέμπτο κεφάλαιο θα γίνει αναφορά στη σχεδίαση του συστήματος και στην υλοποίηση. Πιο συγκεκριμένα θα παρουσιαστούν κάποια use cases όσο αφορά τους χρήστες του συστήματος και ακολούθως θα παρουσιαστεί το αποτέλεσμα με screen shots από την εφαρμογή.

### **Κεφάλαιο 6. Συμπεράσματα και Μελλοντική Εργασία**

Στο έκτο και τελευταίο κεφάλαιο θα γίνει αναφορά στα τελικά συμπεράσματα που θα καταλήξουμε και θα αναφερθούν κάποιες ιδέες για μελλοντική εργασία έτσι ώστε να επεκταθεί το σύστημα και να γίνει ακόμα καλύτερο.

## Κεφάλαιο 2

### Περιγραφή προβλήματος και ανασκόπηση Βιβλιογραφίας

---

|                                               |    |
|-----------------------------------------------|----|
| 2.1 Περιγραφή του προβλήματος                 | 5  |
| 2.1.1 Υποπροβλήματα και λύσεις υποπροβλημάτων | 7  |
| 2.2 Άλλα παρόμοια συστήματα                   | 8  |
| 2.3 Ανασκόπηση Βιβλιογραφίας                  | 10 |

---

#### 2.1 Περιγραφή Προβλήματος

Το πρόβλημα που έπρεπε να επιλυθεί είναι ότι έπρεπε να χτιστεί μια διαδικτυακή εφαρμογή την οποία θα τη χρησιμοποιούν διάφοροι χρήστες διαφορετικών προτιμήσεων που θέλουν να δουν προτεινόμενες ταινίες σε αυτούς και ως επίσης να μπορούν να βαθμολογήσουν ταινίες από το 1 μέχρι το 5 έτσι ώστε να υπολογίζονται από το σύστημα οι νέες τους προτιμήσεις και να διαμορφώνονται οι μελλοντικές προτεινόμενες σε αυτούς ταινίες. Για το λόγο αυτό έπρεπε να χτιστεί μια ευχάριστη εφαρμογή που να είναι παράλληλα και απλή έτσι ώστε ο χρήστης να μπορεί να τη χρησιμοποιεί με ευκολία χωρίς μεγάλο χρόνο προσαρμογής.

Όσον αφορά τις συστάσεις θέλαμε να αξιοποιήσουμε όση περισσότερη πληροφορία μας παρείχε το τα δεδομένα μας (data set). Για το χρήστη οι πληροφορίες που μπορούσαν να αξιοποιηθούν ως συγκείμενο είναι η ηλικία του και το επάγγελμα του. Για τις ταινίες οι πληροφορίες που αξιοποιήθηκαν σαν συγκείμενο είναι το είδος της ταινίας (μια ταινία μπορεί να ανήκει σε περισσότερα από ένα είδος). Τέλος, κάτι που ήταν χρήσιμο να αξιοποιηθεί είναι το η μέρα και ώρα (timestamp) που έγινε η βαθμολόγηση. Το timestamp είναι ένας ακέραιος αριθμός που αντιστοιχεί σε ένα δευτερόλεπτο μετά την 01/01/1970. Αξιοποιήθηκε αυτή η πληροφορία για να μπορούμε να ξέρουμε εάν μία βαθμολόγηση έγινε καθημερινή μέρα ή μέρα μέσα στο Σαββατοκύριακο.

Αρχικά χώρισα σε δύο διαφορετικά σημεία της εφαρμογής το «See Overall Recommendations» και το «Rate A Movie». Στο σημείο «See Overall Recommendations» ο χρήστης μπορεί να δει σε ένα carousel τις προτεινόμενες ταινίες ως προς το προφίλ του ταξινομημένες με βάση την πιο κοντινή σε αυτόν. Ο χρήστης μπορεί να δει για κάθε ταινία τη φωτογραφία της, το τίτλο της και την ημερομηνία που κυκλοφόρησε. Επιπλέον μπορεί να δει περισσότερες πληροφορίες για τη ταινία πατώντας ένα κουμπί που θα του ανοίξει την σελίδα της ταινίας στην IMDB[16]. Στο σημείο «Rate A Movie» ο χρήστης μπορεί να αναζητήσει την ταινία που θέλει να βαθμολογήσει σε ένα πεδίο αναζήτησης, ακολούθως να του εμφανιστεί η φωτογραφία της ταινίας για να είναι σίγουρος για την επιλογή του, να επιλέξει τη βαθμολογία που θέλει (1-5) και τέλος να πατήσει το κουμπί “submit”.

Στην συνέχεια προσθέσαμε ακόμα μια λειτουργικότητα για να κάνω την εφαρμογή πιο φιλική προς το χρήστη. Συγκεκριμένα πρόσθεσα στο carousel που εμφανίζονται οι προτεινόμενες ταινίες ένα κουμπί «Add To Watchlist» για να δηλώνει ο χρήστης το ενδιαφέρον του για να παρακολουθήσει στο μέλλον τη συγκεκριμένη ταινία. Έτσι όταν κάνει login στην εφαρμογή του εμφανίζονται οι ταινίες που έχει «watchlisted» για να τις βαθμολογήσει απ’ ευθείας εάν τις έχει δει.

Τέλος, για τους νέους χρήστες, για το λόγο ότι οι αλγόριθμοι που χρησιμοποιεί το σύστημα είναι δομημένοι με τρόπο που να πρέπει να κάνεις κάποια αρχικά rates για να σου παρέχει συστάσεις, προστέθηκαν ακόμα δύο λίστες με ποιο απλές μεθόδους συστάσεων έτσι ώστε να έχουν κάποιες αρχικές συστάσεις και οι συγκεκριμένοι χρήστες. Πρόκειται για δύο κατηγορίες από συστάσεις που η πρώτη ανάλογα με την ηλικία και η δεύτερη ανάλογα με το επάγγελμα του χρήστη, προτείνουν ταινίες προς αυτόν που βαθμολογήθηκαν με τη μέγιστη βαθμολογία (5) από άλλους χρήστες συνομήλικους ή στο ίδιο επάγγελμα αντίστοιχα.

Με τους πιο πάνω τρόπους κάναμε την εφαρμογή πιο λειτουργική και χρηστική.

### 2.1.1 Υποπρόβλημα και λύσεις υποπροβλημάτων

Ένα υποπρόβλημα που αντιμετωπίσαμε είναι ότι η δημιουργία των συστάσεων είναι μια χρονοβόρα διαδικασία που δεν θα μπορούσε να γίνεται κάθε φορά που ένα χρήστης προσθέτει μία βαθμολογία σε μία ταινία. Θα ήταν το ιδανικό σενάριο αυτό, όμως αποκλείστηκε λόγω του χρόνου απόκρισης που χρειάζεται. Η εναλλακτική μας λύση ήταν να γίνεται αυτή η διαδικασία ανά τακτά χρονικά διαστήματα. Εδώ πρέπει να αναφερθεί ότι η δημιουργία των συστάσεων γίνεται από ένα ξεχωριστό σύστημα, το CarsKit[6]. Πρόκειται για ένα Java Based λογισμικό το οποίο ενσωματώσαμε με το δικό μας έτσι ώστε να μας παράγει τις συστάσεις. Η ενσωμάτωση έγινε με απόλυτη επιτυχία όμως έπρεπε να γίνει κάποιος αυτοματοποιημένος τρόπος να γίνεται μόνη της αυτή η λειτουργία που όπως είπαμε είναι χρονοβόρα. Για να λύσουμε αυτό το πρόβλημα δημιουργήθηκε ένα crone job όπου κάθε μία ώρα ξεκινά τη διαδικασία που κάνει ενημέρωση τις συστάσεις. Crone job είναι κάποιες διαδικασίες υλοποιημένες στο επίπεδο του λειτουργικού συστήματος που τρέχουν περιοδικά ανάλογα με ένα χρονικό διάστημα που παίρνουν σαν παράμετρο. Άρα όταν ο χρήστης βαθμολογεί μια ταινία δεν αλλάζουν οι προσωπικές του συστάσεις άμεσα αλλά στη χειρότερη περίπτωση θα αλλάζουν σε μία ώρα. Η επιλογή του crone job έγινε έτσι ώστε το σύστημα να είναι αυτοματοποιημένο και να μην χρειάζεται κάποιος διαχειριστής σε κάποια φάση της ημέρας να ενεργοποιεί αυτή τη διαδικασία. Παρόλα αυτά υπάρχει και η επιλογή να ξεκινήσει ο διαχειριστής τη διαδικασία όποτε θέλει από τη σελίδα διαχειριστή.

Ένα άλλο υποπρόβλημα που αντιμετωπίσαμε είναι ότι η δομή των δεδομένων έπρεπε να περάσει από μεγάλη επεξεργασία για να μπορέσουμε να τα εισαγάγουμε στη βάση δεδομένων. Επίσης δεν είχαμε διατεθειμένη την εικόνα κάθε ταινίας που ήταν απαραίτητη για τη διαμόρφωση της δομής της ιστοσελίδας. Το μόνο που είχαμε διατεθειμένο ήταν ένα εξωτερικό σύνδεσμο για κάθε ταινία. Χρησιμοποιώντας αυτό τον σύνδεσμο γράφτηκε ένα script που να διαπερνά τους εξωτερικούς συνδέσμους όλων των ταινιών και να αποθηκεύει τοπικά στον υπολογιστή την εικόνα κάθε ταινίας. Έτσι καταφέραμε να έχουμε στη διάθεση μας τις εικόνες όλων των ταινιών.



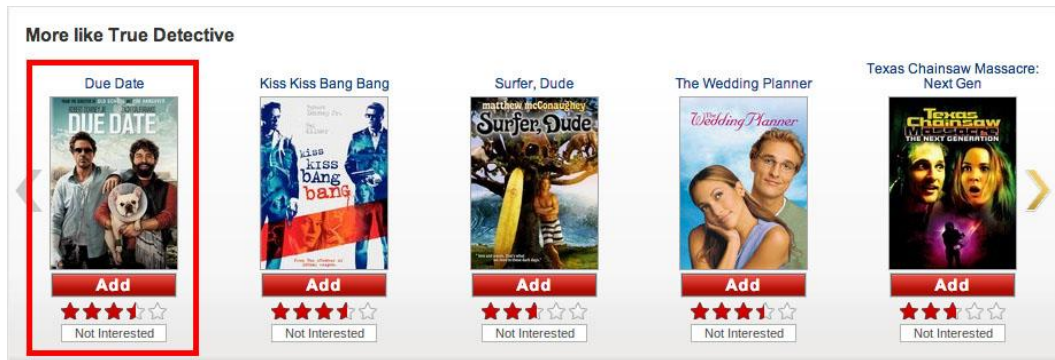
## 2.2 Άλλα Παρόμοια Συστήματα

Όπως αναφέρθηκε και στην εισαγωγή πάρα πολλά συστήματα σήμερα χρησιμοποιούν συστάσεις για τους χρήστες τους. Κάνοντας μία έρευνα βρέθηκαν πάρα πολλά παραδείγματα αλλά θα γίνει αναφορά στα πιο φημισμένα. Αρχικά θα αναφερθεί η IMDB γιατί είναι η πιο κοντινή στο θέμα που επιλέξαμε. Η IMDB είναι ίσως η μεγαλύτερη πλατφόρμα ανεύρεσης ταινιών, τηλεοπτικών σειρών και οτιδήποτε σχετίζεται με το χώρο. Η συγκεκριμένη πλατφόρμα παρέχει συστάσεις για τους χρήστες της στην αρχική της σελίδα.



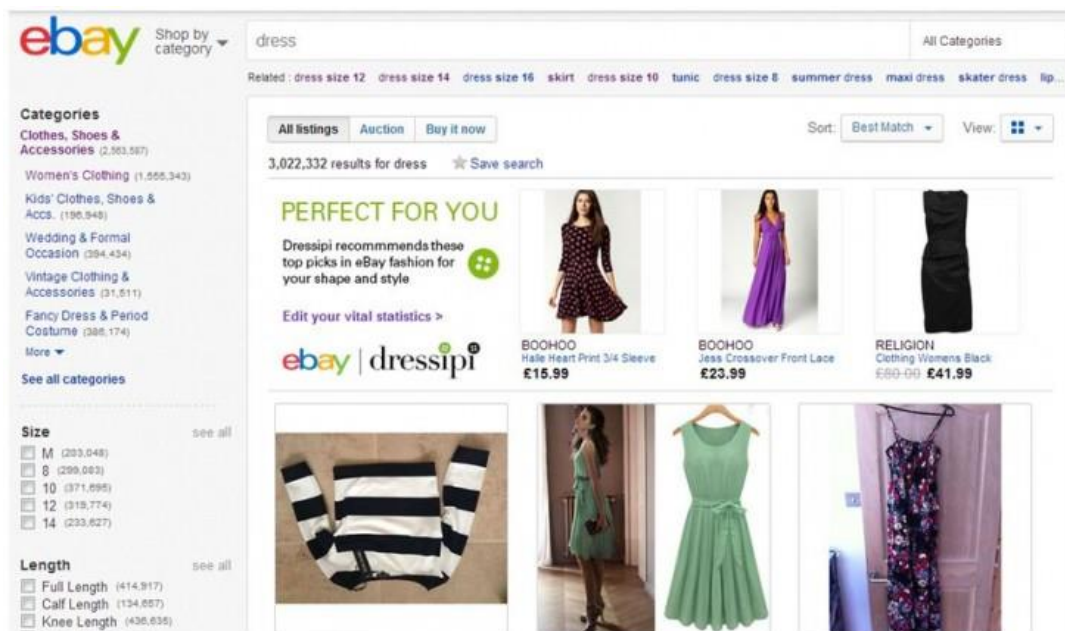
Σχήμα 2.1 – Recommendation System of IMDB

Παραμένοντας στη κατηγορία ταινιών, άλλη μια ιστοσελίδα ταινιών που παρέχει συστάσεις με μεγάλη επιτυχία είναι η Netflix[17].



Σχήμα 2.2 – Recommendation System of Netflix

Άλλα Συστήματα Συστάσεων είναι αυτά της Amazon[18] και του E-bay[19]. Αν και δεν έχουν να κάνουν αποκλειστικά με ταινίες αλλά πληθώρα προϊόντων αξίζουν να αναφερθούν γιατί είναι από τα πιο διαδεδομένα.



Σχήμα 2.3 – Recommendation System of E-bay



Σχήμα 2.4 - Recommendation System of Amazon

Βλέπουμε πώς η διάφορες ιστοσελίδες που προαναφερθήκαν επιλέγουν να παρουσιάσουν τις συστάσεις συνήθως σε carousels ή με κάποιο paging. Παρουσιάζουν αρχικά τις βασικές πληροφορίες του προϊόντος και όταν επιλέξεις ένα προϊόν σου εμφανίζουν περισσότερες. Επίσης ένα χαρακτηριστικό στοιχείο είναι τα αστέρια. Έχει διαδοθεί πολύ το να παρουσιάζουν τη βαθμολογία ενός προϊόντος με αστέρια συνήθως από το 1 μέχρι το 5 και πιο σπάνια από το 1 μέχρι το 10.

## 2.3 Ανασκόπηση Βιβλιογραφίας

Η ανασκόπηση της βιβλιογραφία έγινε με σκοπό να μάθουμε τους αλγορίθμους που υπάρχουν τόσο για τα απλά Συστήματα Συστάσεων όσο και για τα Συστήματα Συστάσεων με επίγνωση συγκεκριμένου. Η μελέτη των αλγορίθμων έπρεπε να γίνει έτσι ώστε να κατανοηθεί τι κάνει κάθε αλγόριθμος και ποιος είναι κατάλληλος να εφαρμοστεί σε κάθε είδους δεδομένα.

Μελετώντας κάποια papers και συνδυάζοντας τις πληροφορίες από αυτά έχω καταλήξει στους πιο δημοφιλείς αλγόριθμους που προσφέρουν τα συγκεκριμένα συστήματα.

Όσο αφορά τα απλά Συστήματα Συστάσεων χωρίζονται σε τρεις βασικές υποκατηγορίες αλγορίθμων.

1. **Content-Based Recommender Systems:** Αυτή η υποκατηγορία αντιπροσωπεύει τα Συστήματα Συστάσεων τα οποία χρησιμοποιούν την ίδια την πληροφορία των προϊόντων, δηλαδή πληροφορία από τα χαρακτηριστικά τους.

Συγκεκριμένα χωρίζει τα αντικείμενα σε κατηγορίες με βάση τα διάφορα χαρακτηριστικά τους και προτείνει στους χρήστες αντικείμενα της ίδιας κατηγορίας με αντικείμενα που έχουν ήδη προτιμήσει στο παρελθόν. Για παράδειγμα, σε ένα χρήστη που προτιμά ταινίες που εντάσσονται στη κατηγορία ‘κωμωδίες’ θα του προταθούν άλλες ταινίες που εντάσσονται στη κατηγορία ‘κωμωδίες’.[1]

**2. Collaborative Recommender Systems:** Αυτή η υποκατηγορία αντιπροσωπεύει Συστήματα Συστάσεων τα οποία επωφελούνται την πληροφορία των χρηστών. Η φιλοσοφία αυτών των συστημάτων είναι να προτείνουν σε χρήστες αντικείμενα που τα έχουν ήδη προτιμήσει ‘παρόμοιοι’ χρήστες. Χαρακτηρίζουμε ‘παρόμοιους’ δύο χρήστες εάν έχουν πολλές κοινές προτιμήσεις. Για παράδειγμα εάν δύο χρήστες έχουν βαθμολογήσει πέντε διαφορετικές ταινίες με κοινές βαθμολογίες είναι πολύ πιθανόν να συμφωνήσουν και σε μια έκτη ταινία. [1,3]

**3. Hybrid Recommender Systems:** Ουσιαστικά αυτή η υποκατηγορία είναι ένας συνδυασμός των δύο πιο πάνω κατηγοριών όταν το επιτρέπουν τα δεδομένα. Θεωρητικά μπορούμε να έχουμε πολύ πιο ακριβή αποτελέσματα από το να εφαρμόσουμε μια από τις πιο πάνω μεθοδολογίες ξεχωριστά αφού αξιοποιείς περισσότερες πληροφορίες.[1,4]

Τα Συστήματα Συστάσεων δεν σταματούν εδώ. Μέχρι το σημείο αυτό αναφέραμε συστήματα τα οποία επεξεργάζονται δισδιάστατα δεδομένα User - Item. Υπάρχουν και συστήματα τα οποία μπορούν να επεξεργαστούν πολυδιάστατα δεδομένα συγκεκριμένου για να κάνουν τα αποτελέσματά τους πιο ακριβή, τα Συστήματα Συστάσεων με επίγνωση συγκεκριμένου. Ένα συνηθισμένο παράδειγμα είναι τα δεδομένα που περιέχουν στοιχεία User - Item - Location. Για παράδειγμα ένας χρήστης στο σπίτι του μπορεί να προτιμά να παρακολουθεί κωμωδίες ενώ στο κινηματογράφο να προτιμά να παρακολουθεί ταινίες φαντασίας. Πρόκειται για μεγαλύτερα σετ δεδομένων και πιο ακριβή σε κόστος επεξεργασία για να παράγουμε αυτά τα αποτελέσματα.[1]

Τα Συστήματα Συστάσεων με επίγνωση συγκεκριμένου χωρίζονται και αυτά σε τρεις μεγάλες κατηγορίες αλγορίθμων.

**1. Contextual Pre-Filtering:** Η προσέγγιση του συγκεκριμένου αλγορίθμου, είναι να επωφελείται της πληροφορίας του συγκεκριμένου έτσι ώστε να φιλτράρει το σετ δεδομένων και να κρατά μόνο τα σχετικά δισδιάστατα δεδομένα (User -

Item) και ακολούθως να υπολογίζει τις συστάσεις. Για παράδειγμα εάν ένας χρήστης θέλει να δει μία ταινία το Σάββατο, θα φιλτραριστούν μόνο τα δεδομένα που αξιολογήθηκαν για τα Σάββατα ή τα Σαββατοκυρίακα για να του προταθούν ταινίες.[2]

**2. Contextual Post-Filtering:** Αυτή η προσέγγιση λειτουργεί κατά κάποιο τρόπο αντίθετα από την προηγούμενη. Αρχικά θα γίνει μια ταξινομημένη λίστα με συστάσεις αγνοώντας το συγκεκριμένο έχοντας πιο ψηλά τις πιο κορυφαίες συστάσεις για ένα χρήστη. Ακολούθως ο αλγόριθμος με χρήση του συγκεκριμένου θα έρθει να αλλάξει αυτή τη λίστα για το συγκεκριμένο χρήστη. Η αλλαγή μπορεί να γίνει με δύο τρόπους:

- a. Να αφαιρεθούν τα στοιχεία που είναι άσχετα με ένα συγκεκριμένο συγκεκριμένο.
- b. Να γίνει αναπροσαρμογή της κατάταξης της λίστας λαμβάνοντας υπόψη το συγκεκριμένο.[2]

**3. Contextual Modelling:** Η διαφορά αυτής της προσέγγισης σε σχέση με τις άλλες δύο είναι ότι χρησιμοποιεί απευθείας τις πληροφορίες του συγκεκριμένου μέσα στη διαδικασία υπολογισμού για να δώσει σαφείς προβλέψεις. Δηλαδή χρησιμοποιεί συναρτήσεις που επεξεργάζονται πολυδιάστατα δεδομένα(User – Item - Context) σε αντίθεση με τους δύο προηγούμενους αλγορίθμους που χρησιμοποιούσαν συναρτήσεις που επεξεργάζονταν δυσδιάστατα δεδομένα(User - Item).[2]

Βλέποντας όλους αυτούς τους αλγορίθμους καταλήξαμε στο συμπέρασμα πως χρειάζεται αρκετή μελέτη για να καταλήξει κάποιος στον αλγόριθμο που θα εφαρμόσει σε κάθε περίπτωση. Πολύ σημαντικός παράγοντας για να κάνει κάποιος την επιλογή είναι η μορφή το δεδομένων που έχει στη διάθεσή του και οι πληροφορίες που περιέχουν.

## Κεφάλαιο 3

### Απαιτούμενη Γνώση Και Τεχνολογίες

---

|                                                         |    |
|---------------------------------------------------------|----|
| 3.1 Γενική εισαγωγή για το CarsKit Framework            | 13 |
| 3.1.1 Αλγόριθμοι του CarsKit Framework                  | 14 |
| 3.1.2 Προϋποθέσεις και παράμετροι του CarsKit Framework | 15 |
| 3.1.3 Μορφή αποτελεσμάτων του CarsKit Framework         | 16 |
| 3.1.4 Γιατί CarsKit Framework                           | 17 |
| 3.1.5 Γιατί CAMF_CU αλγόριθμος                          | 17 |
| 3.2 Γενική εισαγωγή στα Web Applications                | 18 |
| 3.2.1 Γιατί Web Application                             | 18 |
| 3.3 HTML 5 – CSS 3                                      | 19 |
| 3.4 JavaScript – JQuery                                 | 20 |
| 3.5 Bootstrap                                           | 21 |
| 3.6 PHP                                                 | 21 |
| 3.7 MySQL                                               | 22 |

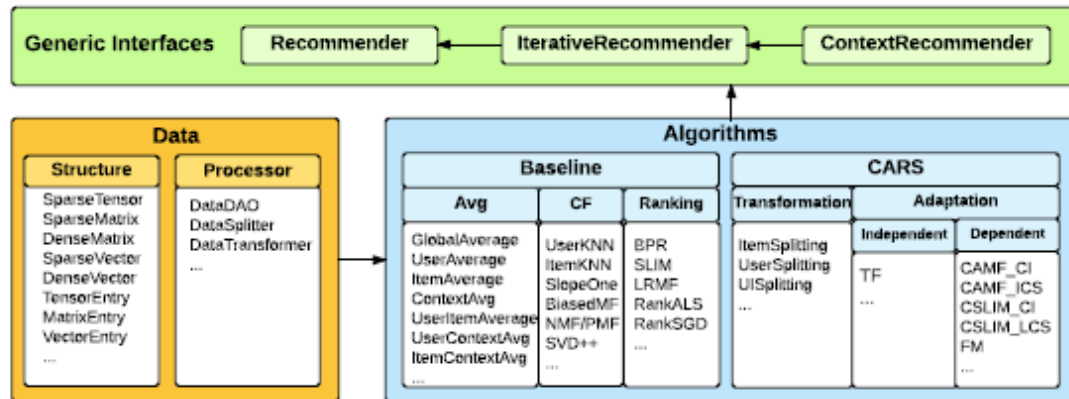
---

#### 3.1 Γενική εισαγωγή για το CarsKit Framework

Το CarsKit Framework είναι το Framework που χρησιμοποιήθηκε για την παραγωγή των συστάσεων.

Το CarsKit Framework είναι ένα λογισμικό ανοικτού κώδικα γραμμένο σε Java με άδεια GNU General Public License (GPL). Ουσιαστικά είναι ένα λογισμικού που παράγει συστάσεις με επίγνωση συγκεκριμένου όταν του δοθούν κάποια δεδομένα στην είσοδο.

Μπορεί να υποστηρίξει πάρα πολλούς αλγορίθμους, τους οποίους τους έχει ομαδοποιήσει κάτω από μία απλή αρχιτεκτονική που βοηθά τους χρήστες να κατανοήσουν τους αλγορίθμους καλύτερα και να μπορούν να προσθέσουν αλγορίθμους όταν το επιθυμήσουν έτσι ώστε να επεκτείνουν το σύστημα. [6]



Σχήμα 3.1 – Algorithms Architecture [6]

### 3.1.1 Αλγόριθμοι του CarsKit Framework

Όπως έχει προαναφερθεί το CarsKit Framework υποστηρίζει πολλούς αλγόριθμους. Αρχικά οι αλγόριθμοι χωρίζονται σε δύο μεγάλες υποκατηγορίες. Η πρώτη κατηγορία είναι οι baseline algorithms όπου είναι κάποιοι γνωστοί αλγόριθμοι συστάσεων με επίγνωση συγκεκριμένου που εφαρμόζονται στα πλείστα παρόμοια συστήματα, όπως για παράδειγμα ο UserKNN και ο ItemKNN. Η δεύτερη κατηγορία είναι οι CARS αλγόριθμοι όπου ανήκουν στο συγκεκριμένο Framework.

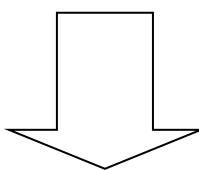
Η υποκατηγορία CARS χωρίζεται σε δύο άλλες υποκατηγορίες, transformation αλγόριθμοι και adaptation αλγόριθμοι. Οι transformation αλγόριθμοι επεξεργάζονται το σετ δεδομένων που παίρνουν ως είσοδο και το μετατρέπουν σε ένα δυσδιάστατο πίνακα από βαθμολογίες μεταξύ χρηστών και αντικειμένων έτσι ώστε να μπορέσουν να εφαρμοστούν πάνω τους παραδοσιακοί αλγόριθμοι που παράγουν συστάσεις. Όσο αφορά τους adaptation αλγόριθμους διαχωρίζονται και αυτοί σε δύο υποκατηγορίες, τους independent και τους dependent αλγόριθμους όπου αυτές οι δύο κατηγορίες

αποτελούν το τελευταίο στρώμα τις αρχιτεκτονικής αυτής. Και οι δύο κατηγορίες ενσωματώνουν την πληροφορία σε μια συνάρτηση πρόβλεψης. Οι διαφορές τους είναι ότι στους independent οι πληροφορίες (context) είναι ανεξάρτητη από τους χρήστες, ενώ από την άλλη στους dependent χρήστες, αντικείμενα και πληροφορίες συσχετίζονται.[6]

### 3.1.2 Προϋποθέσεις και παράμετροι του CarsKit Framework

Για να μπορέσει να λειτουργήσει σωστά το CarsKit πρέπει να του δοθούν τα δεδομένα εισόδου με μια συγκεκριμένη μορφή που να μπορεί να τα αναγνωρίσει. Συγκεκριμένα δέχεται τα δεδομένα με δύο διαφορετικές μορφές όπου θα δουλέψουν εξίσου. Στο εγχειρίδιο του αναφέρονται ως Loose Format και Compact Format. Οι δύο αυτές μορφές δεδομένων θα μετατραπούν σε ένα binary format και τότε θα περαστούν στο σύστημα.

| (a) Loose Format |        |        |          |           | (b) Compact Format |        |        |         |          |
|------------------|--------|--------|----------|-----------|--------------------|--------|--------|---------|----------|
| UserID           | ItemID | Rating | Context  | Condition | UserID             | ItemID | Rating | Time    | Location |
| U1               | T1     | 3      | Time     | Weekend   | U1                 | T1     | 3      | Weekend | Work     |
| U1               | T1     | 3      | Location | Work      | U2                 | T2     | 4      | Weekday | Home     |
| U2               | T2     | 4      | Time     | Weekday   | U1                 | T1     | 4      | Weekend | Home     |
| U2               | T2     | 4      | Location | Home      | U2                 | T2     | 2      | Weekday | Work     |

| UserID | ItemID | Rating | Time:Weekend | Time:Weekday | Location:Home | Location:Work |
|--------|--------|--------|--------------|--------------|---------------|---------------|
| U1     | T1     | 3      | 1            | 0            | 0             | 1             |
| U2     | T2     | 4      | 0            | 1            | 1             | 0             |
| U1     | T1     | 4      | 1            | 0            | 1             | 0             |
| U2     | T2     | 2      | 0            | 1            | 0             | 1             |

Σχήμα 3.2 – Data format[6]

Το συγκεκριμένο Framework μπορούν να τεθούν ως παράμετροι πολλοί παράγοντες, μέσω ενός configuration file. Το configuration file πρέπει να



βρίσκεται στο ίδιο path με το εκτελέσιμο του CarsKit. Οι πιο σημαντικοί παράμετροι που αξίζουν να αναφερθούν γιατί χρησιμοποιήθηκαν στη συγκεκριμένη εφαρμογή είναι το path του σετ δεδομένων, το όριο των συστάσεων που θα εμφανίζει για κάθε χρήστη και ο αλγόριθμος που θα χρησιμοποιηθεί.[6]

### 3.1.3 Μορφή αποτελεσμάτων του CarsKit Framework

Κατά την εκτέλεση του CarsKit παίρνουμε στην έξοδο ένα αρχείο (results.txt) με κάποια στατιστικά αποτελέσματα για την εκτέλεση μας. Πιο κάτω παρουσιάζεται ένα δείγμα του συγκεκριμένου αρχείου όπου προέκυψε από δική μας δοκιμή.

```
WorkingPath: C:\wamp64\www\ade\CARSKit-master\context-aware_data_sets\MyMovies\CARSKit.Workspace\
Your original rating data path: C:\wamp64\www\ade\CARSKit-master\context-aware_data_sets\MyMovies\ratings.txt
Current working path: C:\wamp64\www\ade\CARSKit-master\context-aware_data_sets\MyMovies\CARSKit.Workspace\
Dataset: ...ARSKit.Workspace\ratings_binary.txt
DataPath: C:\wamp64\www\ade\CARSKit-master\context-aware_data_sets\MyMovies\CARSKit.Workspace\ratings_binary.txt
Rating data set has been successfully loaded.

/*****
*
* Dataset: C:\wamp64\www\ade\CARSKit-master\context-aware_data_sets\MyMovies\CARSKit.Workspace\ratings_binary.txt
*
* Statistics of U-I-C Matrix:
* User amount: 944
* Item amount: 1682
* Rate amount: 100008
* Context dimensions: 3 (Day, AgeRange, Type)
* Context conditions: 28 (Day: 2, AgeRange: 7, Type: 19)
* Context situations: 2210
* Data density: 0.0237%
* Scale distribution: [4.0 x 34178, 1.0 x 6110, 2.0 x 11371, 5.0 x 21203, 3.0 x 27146]
* Average value of all ratings: 3.529874
* Standard deviation of all ratings: 1.125645
* Mode of all rating values: 4.000000
* Median of all rating values: 4.000000
*
* Distribution of rate counts per user: mean = 105.94067796610169, median = 64.5, sd = 100.89220983961062
* Distribution of rate counts per item: mean = 59.45778834720571, median = 27.0, sd = 80.36619333708555
* Distribution of rate counts per context condition: mean = 15041.25, median = 8235.5, sd = 16337.948200486673
*
* Average rating in each context condition: (Average, Counts)
```

Σχήμα 3.3 – Sample of results.txt

Εννοείται ότι στην έξοδο εκτός από το συγκεκριμένο αρχείο παίρνουμε και τα τελικά αποτελέσματα, δηλαδή τα συστάσεις σε ένα άλλο αρχείο. Κάθε γραμμή του αρχείου που προκύπτει αντιστοιχεί σε ένα χρήστη και στη γραμμή του παρουσιάζονται οι ταινίες που του προτείνονται με σειρά προτεραιότητας το υψηλότερο score.[6]

### 3.1.4 Γιατί CarsKit Framework

Η επιλογή του CarsKit Framework έγινε για πολλούς λόγους.

Ο κυριότερος λόγος είναι η ευελιξία που παρέχει, αφού μπορεί να υποστηρίξει πάρα πολλούς αλγορίθμους και η αναπροσαρμογή του αλγορίθμου είναι πολύ εύκολη αφού το μόνο που πρέπει να κάνεις είναι να αλλάξεις μία γραμμή στο configuration file. Αυτή η ευελιξία που παρέχει κάνει αμέσως και το σύστημα μας ευέλικτο αφού δεν θα είναι δεσμευμένο να λειτουργεί με ένα μόνο αλγόριθμο. Ανά πάσα στιγμή που θα θεωρήσουμε ότι τα αποτελέσματα μας δεν είναι αυτά που θα ήθελαν οι χρήστες μπορούμε χωρίς ιδιαίτερο κόπο να τους προσφέρουμε κάτι νέο.

Άλλοι σημαντικοί λόγοι που λήφθηκαν σοβαρά υπόψη πριν την επιλογή του, είναι ότι μπορεί να τρέξει σε οποιοδήποτε λειτουργικά σύστημα, είναι λογισμικό ανοικτού κώδικα (open source) και η άδεια λογισμικού του μας επιτρέπει να το επεκτείνουμε εάν χρειαστεί.

### 3.1.5 Γιατί CAMF\_CU αλγόριθμος

Η επιλογή του αλγορίθμου CAMF\_CU (Context-Aware Matrix Factorization) έγινε με κύριο μέλημα την απόδοση του. Θέλαμε ένα αλγόριθμο ο οποίος δεν θα ήθελε ώρες να δώσει αποτελέσματα μιας και η εκτέλεση του θα είναι μια επαναλαμβανόμενη διαδικασία. Ο συγκεκριμένος αλγόριθμος είναι ένας πολύ γρήγορος αλγόριθμος ακόμα και σε μεγάλο αριθμό δεδομένων, όπου στην περίπτωση μας είχαμε. Επίσης οι αλγόριθμοι κατηγορίας CAMF είναι σύγχρονοι αλγόριθμοι οι οποίοι, όπως φαίνεται και από το όνομά τους, χρησιμοποιούν το συγκεκριμένο. Εκτός από το συγκεκριμένο χρησιμοποιούν και τη μάθηση έτσι ώστε να προβλέπουν πιο ακριβή αποτελέσματα, κάτι που δεν το προσφέρουν οι πιο παλιοί αλγόριθμοι. Χρησιμοποιούνται πολύ τα τελευταία χρόνια από την επιστημονική κοινότητα και η τεχνική Matrix Factorization θεωρείται η καλύτερη τεχνική

για παροχή συστάσεων με αξιοποίηση δεδομένων συγκειμένου μέχρι σήμερα. [5]

## **3.2 Γενική εισαγωγή στα Web Applications**

Για την υλοποίηση της εφαρμογής επιλέχτηκε να γίνει Web Application.

Στο τομέα της Πληροφορικής Web Application (διαδικτυακή εφαρμογή) ορίζουμε κάθε εφαρμογή η οποία μπορεί να διατεθεί στους χρήστες μέσω του διαδικτύου και χρειάζεται browser για να την χρησιμοποιήσει. Συνήθως οι διαδικτυακές εφαρμογές είναι υλοποιημένες σε ισχυρές υπολογιστικές μηχανές οι οποίες είναι οι εξυπηρετητές και παρέχουν υπηρεσίες σε περισσότερους από ένα χρήστες.[7]

### **3.2.1 Γιατί Web Application**

Ο λόγος που επιλέχτηκε να γίνει η εφαρμογή διαδικτυακή είναι γιατί παρέχει αρκετά πλεονεκτήματα σε σχέση με άλλου τύπου εφαρμογές όσο αφορά την αλληλεπίδραση με το χρήστη.

Αρχικά οι χρήστες θα μπορούν να έχουν άμεση πρόσβαση από οποιαδήποτε συσκευή έχει σύνδεση στο διαδίκτυο και ανεξάρτητα της τοποθεσίας στην οποία βρίσκονται. Η μόνη προϋπόθεση είναι να έχει η συσκευή εγκατεστημένο κάποιο browser. Δηλαδή μπορεί να λειτουργήσει χωρίς να χρειάζεται η εγκατάσταση κάποιου επιπλέον λογισμικού. Επίσης οι διαδικτυακές εφαρμογές είναι συμβατές με όλα τα λειτουργικά συστήματα επειδή εκτελούνται μέσω του browser. Τέλος ένα άλλο πολύ σημαντικό πλεονέκτημα είναι ότι δεν καταναλώνουν πόρους αφού όπως προαναφέραμε η εφαρμογή δεν εκτελείτε στη συσκευή αλλά στο browser.[7]

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν οι πιο κάτω τεχνολογίες οι οποίες θα αναλυθούν στη συνέχεια:

- HTML
- CSS
- JavaScript
- JQuery

- Bootstrap
- PHP
- MySQL

### 3.3 HTML 5 – CSS 3



Σχήμα 3. 4

Η γλώσσα HTML(Hypertext Mark-Up Language, Γλώσσα Σήμανσης Υπερκειμένου) σε συνδυασμό με τη γλώσσα CSS(Cascading Style Sheets), είναι οι βασικές γλώσσες για δημιουργία ιστοσελίδων. Η τελευταίες εκδόσεις τους είναι η HTML5 και CSS3 αντίστοιχα, οι οποίες χρησιμοποιήθηκαν για την ανάπτυξη της δικής μου ιστοσελίδας.

Η HTML συντάσσεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες. Ο browser διαβάζει τα έγγραφα HTML και τα παρουσιάζει στο χρήστη σε σελίδες που μπορεί να τις διαβάσει ή να τις ακούσει. Ο browser δηλαδή, δεν εμφανίζει τις ετικέτες αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας και να το εμφανίσει.[8]

Η CSS είναι η γλώσσα η οποία είναι υπεύθυνη για να ορίζει την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου κειμένου που θα εμφανιστεί. Η CSS συντάσσεται με κανόνες μορφοποίησης οι οποίοι εφαρμόζονται σε συγκεκριμένες ετικέτες που καθορίζονται με τη χρήση επιλογών. Οι CSS κανόνες μπορούν να

ενσωματώνονται στα HTML αρχεία ή να γράφονται σε ξεχωριστό αρχείο που να περιλαμβάνεται στο HTML αρχείο.[9]

### 3.4 JavaScript - JQuery



Σχήμα 3. 5

Η JavaScript είναι μία γλώσσα προγραμματισμού η οποία εκτελείται στο επίπεδο του πελάτη (client-side) και έχει σαν σκοπό την παραγωγή δυναμικού περιεχομένου. Προσφέρεται για να μπορείς να δημιουργήσεις μια διαδραστική ιστοσελίδα. Όπως και το CSS έτσι και ο κώδικας JavaScript μπορεί να συμπεριληφθεί στον HTML κώδικα ή να συνταχθεί σε ξεχωριστό αρχείο και να περιλαμβάνεται στο HTML αρχείο.[10]

Η JQuery είναι μια βιβλιοθήκη JavaScript, συμβατή με όλους τους browser. Το πλεονέκτημα της JQuery είναι ότι μπορείς να συντάξεις κώδικα JavaScript με πολύ πιο εύκολη και κατανοητή σύνταξη. Το κυριότερο πρόβλημα που επιλύει είναι το πρόβλημα συμβατότητας της JavaScript με τους διάφορους browser που υπάρχουν.[11]

Οι συγκεκριμένες τεχνολογίες χρησιμοποιήθηκαν σε σημεία της ιστοσελίδας που θέλαμε να γίνουν διαδραστικά, όπως για παράδειγμα για άνοιγμα και κλείσιμο ενός modal window. Χρησιμοποιήθηκαν επίσης για ασύγχρονα καλέσματα (ajax calls), σε σημεία που θέλαμε να παρουσιάσουμε δυναμικό περιεχόμενο.

### 3.5 Bootstrap



Σχήμα 3. 6

Το bootstrap είναι μια συλλογή εργαλείων ανοιχτού κώδικα που χρησιμεύει στη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει HTML και CSS για τη παρουσίαση των δεδομένων καθώς και προαιρετικές επεκτάσεις JavaScript. Πρόκειται για το πιο δημοφιλές πρόγραμμα στο GitHub.

Το κυριότερο πλεονέκτημα αυτής της τεχνολογίας είναι ότι το περιεχόμενο της σελίδας προσαρμόζεται δυναμικά ανάλογα με τα χαρακτηριστικά της συσκευής. Άρα χρησιμοποιώντας το bootstrap καταφέραμε να δημιουργήσουμε μια ιστοσελίδα responsive σε όλες τις συσκευές που ήταν απαραίτητη προϋπόθεση. [12]

### 3.5 PHP



Σχήμα 3. 7

Η PHP είναι μια scripting server-side γλώσσα προγραμματισμού που σχεδιάστηκε για τη δημιουργία δυναμικών ιστοσελίδων. Μπορεί να ενσωματωθούν PHP Scripts σε μια HTML σελίδα που θα εκτελείται κάθε φορά που θα φορτώνεται η σελίδα. Ο PHP κώδικας μεταφράζεται στο Web διακομιστή και προκύπτει HTML κώδικας που θα δει ο χρήστης. Είναι προϊόν ανοιχτού κώδικα με άδεια λογισμικού GNU.[13]

Η PHP χρησιμοποιείται σε σημεία της ιστοσελίδας που θέλουμε να φέρνουμε δυναμικά στοιχεία από τη βάση δεδομένων και να τα εμφανίζουμε. Για παράδειγμα οι ταινίες που θέλουμε να εμφανίζονται σε κάθε χρήστη δεν είναι στατικό περιεχόμενο μέσα στην ιστοσελίδα μας αλλά δυναμικό το οποίο με τη βοήθεια της PHP μπορούμε να το παρουσιάσουμε σε κάθε χρήστη.

### 3.6 MySQL



Σχήμα 3. 9

Η MySQL είναι ίσως το πιο διαδεδομένο σύστημα διαχείρισης βάσεων δεδομένων. Μια βάση δεδομένων επιτρέπει την αποθήκευση, την αναζήτηση, την ταξινόμηση και την ανάκληση των δεδομένων αποτελεσματικά. Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα για να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα. Επίσης παρέχει ασφάλεια στα δεδομένα αφού επιτρέπει μόνο σε πιστοποιημένους χρήστες να έχουν πρόσβαση σε αυτά.

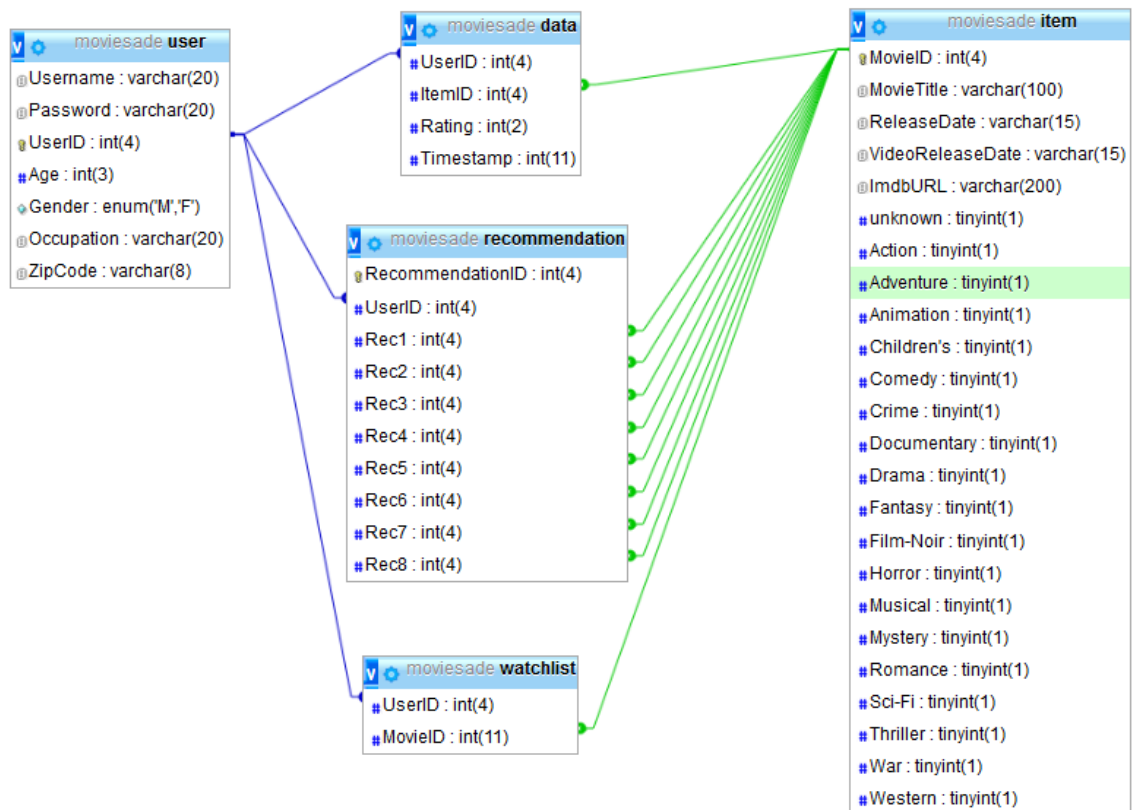
Χρησιμοποιεί την SQL (Structured Query Language), την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων, παγκόσμια.[14]

Για τους παραπάνω λόγους επιλέχτηκε η βάση δεδομένων να γίνει MySQL. Οι πίνακες που χρησιμοποιήθηκαν είναι οι εξής:

1. **Πίνακας User:** Ο πίνακας αυτός δημιουργήθηκε για να αποθηκεύονται οι χρήστες. Τα γνωρίσματα του καθορίστηκαν με βάση το συγκεκριμένο που περιείχε το σετ δεδομένων όσο αφορά τους χρήστες επειδή θέλαμε να έχουμε αποθηκευμένο όλο το συγκεκριμένο σε περίπτωση που θέλουμε να το περάσουμε ως παράμετρο σε ένα αλγόριθμο.
2. **Πίνακας item:** Ο πίνακας αυτός δημιουργήθηκε για να αποθηκεύονται οι ταινίες. Τα γνωρίσματα του ομοίως με τους χρήστες καθορίστηκαν με βάση το συγκεκριμένο που περιέχει το σετ δεδομένων για τις ταινίες. Θέλαμε και εδώ να κρατήσουμε όλη τη πληροφορία από το σετ δεδομένων γιατί κάποιες πληροφορίες θέλαμε να τις εμφανίζουμε κατά τη παρουσίαση μιας ταινίας στην ιστοσελίδα (π.χ. το όνομα) και κάποιες άλλες για να περνιούνται ως συγκεκριμένο στους αλγορίθμους(π.χ. το είδος της ταινίας) .
3. **Πίνακας Data:** Ο πίνακας αυτός αποτελεί όλα τα ratings από κάποιο χρήστη σε ταινία. Δημιουργήθηκε για να απεικονίσει τη many to many σχέση που έχουν οι χρήστες με τις ταινίες.
4. **Πίνακας Recommendation:** Πρόκειται για το πίνακα που κρατά τα top 8 recommendations του χρήστη. Σχεδιάστηκε έτσι ώστε οι συστάσεις κάθε χρήστη να είναι γνωρίσματα του πίνακα γιατί έτσι κάθε χρήστης θα έχει ένα record καταχωρημένο και όχι 8. Με αυτό το τρόπο επιταχύνουμε τη διαδικασία ενημέρωσης του συγκεκριμένου πίνακα που είναι μια επαναλαμβανόμενη διαδικασία.
5. **Πίνακας Watchlist:** Αυτός ο πίνακας είναι επίσης μια many to many σχέση μεταξύ χρηστών και ταινιών. Σε αυτόν αποθηκεύουμε ένα χρήστη που έχει watchlisted μια ταινία.

Το σχεδιάγραμμα οντοτήτων που προέκυψε είναι το πιο κάτω:





### Περιγραφή του σχήματος:

Από το σχήμα βλέπουμε ότι οι πίνακες User και Item είναι οι πιο βασικοί και οι πίνακες Data, Recommendation και Watchlist είναι ενδιάμεσες σχέσεις τους.

Βλέπουμε ότι ο πίνακας Data έχει δύο γνωρίσματα UserID και ItemID που είναι Foreign Keys στους πίνακες User και Item αντίστοιχα. Έχει και ως επίσης δύο άλλα γνωρίσματα, το rating ου απευθύνεται στη βαθμολογία του χρήστη προς τη ταινία και το timestamp που είναι το timestamp που καταχωρήθηκε το record.

Ο πίνακας Recommendation έχει ως γνωρίσματα ένα RecommendationID που είναι το Primary Key του. Ακολούθως έχει ένα γνώρισμα foreign key σε user όπου διευκρινίζει σε ποιο χρήστη ανήκουν οι συστάσεις και 8 πανομοιότυπες πλειάδες που αναφέρονται στις ταινίες που είναι ουσιαστικά οι συστάσεις του χρήστη.

Τέλος ο πίνακας watchlist είναι απλά μια many to many σχέση μεταξύ User και Item με πλειάδες UserID και MovieID αντίστοιχα.

## Κεφάλαιο 4

### Ανάλυση Απαιτήσεων, Προδιαγραφές

---

|                                                                      |    |
|----------------------------------------------------------------------|----|
| 4.1 Εισαγωγή στις έννοιες Ανάλυση Απαιτήσεων και Εύρεση Προδιαγραφών | 25 |
| 4.2 Σκοπός Ανάλυσης Απαιτήσεων στη παρούσα εργασία                   | 27 |
| 4.3 Απαιτήσεις που εξάχθηκαν                                         | 27 |

---

#### 4.1 Εισαγωγή στις έννοιες Ανάλυση Απαιτήσεων και Εύρεση Προδιαγραφών

Είναι καλύτερο να προλαμβάνεις, παρά να θεραπεύεις. Είναι αποτελεσματικότερο να σχεδιάζεις κάτι προσεγμένα από την αρχή, παρά να το διορθώνεις εκ των υστέρων. Καλύτερα να ξέρεις τι ζητάς από την αρχή παρά να τα ανακαλύπτεις στη πορεία. Αυτές είναι οι κυριότερες αρχές που διέπουν την ανάλυση απαιτήσεων (requirements analysis), όσο αφορά την πληροφορική και τα υπό ανάπτυξη έργα της.

Δυστυχώς σε πολλές περιπτώσεις προγραμματιστές, αναλυτές πελάτες και χρήστες υποτιμούν την ανάλυση απαιτήσεων και δεν τις δίνουν την απαραίτητη σημασία με αποτέλεσμα τα έργα που δημιουργούνται να μην καλύπτουν τους αρχικούς όρους και προδιαγραφές που είχαν τεθεί αρχικά. Αυτό κοστίζει αρκετά γιατί οι βελτιώσεις και αλλαγές στα έργα πληροφορικής είναι δαπανηρές, δύσκολες και χρονοβόρες. Άρα λαμβάνοντας υπόψη τα παραπάνω αντιλαμβανόμαστε πως η ανάλυση απαιτήσεων είναι και πρέπει να θεωρείται μια διαδικασία ζωτικής σημασίας στην ανάπτυξη έργων πληροφορικής.

Η ανάλυση απαιτήσεων είναι η διαδικασία καθορισμού των απαιτούμενων υπηρεσιών και περιορισμών λειτουργίας και ανάπτυξης του συστήματος. Η διαδικασία καθορισμού απαιτήσεων αποτελείται από τα πιο κάτω στάδια:

1. Μελέτη σκοπιμότητας: Σε αυτό το στάδιο γίνεται εκτίμηση αν οι ανάγκες των χρηστών μπορούν να ικανοποιηθούν με τρέχουσες τεχνολογίες, αν είναι συμφέρον από επιχειρηματικής άποψης, αν το απαιτούμενο κόστος είναι εντός προϋπολογισμού.
2. Εξαγωγή και ανάλυση απαιτήσεων: Σε αυτό το στάδιο εξετάζονται τα υπάρχοντα συστήματα, γίνεται επικοινωνία με τους τελικούς χρήστες και κατασκευάζονται μοντέλα.
3. Προδιαγραφή απαιτήσεων: Σε αυτό το στάδιο κατασκευάζεται το έγγραφο απαιτήσεων.
4. Επικύρωση απαιτήσεων: Σε αυτό το στάδιο ελέγχονται οι απαιτήσεις από άποψη ρεαλιστικότητας, συνέπειας, πληρότητας και μπορεί επίσης να τροποποιηθεί το έγγραφο.

Μια απαίτηση μπορεί να είναι κάτι πολύ μεγάλο ή κάτι πολύ μικρό όπως μια πολύ εξειδικευμένη και μικρή λειτουργία που εμπίπτει σε μία μεγαλύτερη.

Η κάθε απαίτηση εμπίπτει σε μια από τις πιο κάτω κατηγορίες:

1. Λειτουργικές απαιτήσεις: Πρόκειται για δηλώσεις που ορίζουν τις λειτουργίες που πρέπει να παρέχει το σύστημα. Δηλαδή πως θα πρέπει να αντιδρά σε συγκεκριμένες εισόδους και πως θα πρέπει να συμπεριφέρεται σε συγκεκριμένες καταστάσεις.
2. Μη λειτουργικές απαιτήσεις: Πρόκειται για περιορισμούς στις λειτουργίες του συστήματος όπως χρονικούς περιορισμούς.
3. Απαιτήσεις πεδίου: Απαιτήσεις που προέρχονται από το πεδίο εφαρμογής του συστήματος και αντανakλούν χαρακτηριστικά και περιορισμούς αυτού του πεδίου.

Η λίστα με τις απαιτήσεις που προκύπτει από την παραπάνω διαδικασία χρησιμεύει τόσο σε αυτούς που θα αναπτύξουν το σύστημα όσο και στους τελικούς χρήστες. Εφαρμόζεται τόσο σε μικρά όσο και σε μεγάλα έργα πληροφορικής. [15]

#### **4.2 Σκοπός Ανάλυσης Απαιτήσεων στη παρούσα εργασία**

Σκοπός της δομημένης ανάλυσης είναι η πλήρης κατανόηση του προβλήματος. Μόνο όταν το πρόβλημα καθοριστεί ακριβώς είναι δυνατή η εγγύηση της αποτελεσματικότητας του συστήματος. Αν οι απαιτήσεις δεν είναι ξεκάθαρες είναι πολύ πιθανόν το σύστημα να μην εξυπηρετεί το αρχικό πρόβλημα αλλά κάποιο άλλο.

Σκοπός της συγκεκριμένης εργασίας είναι να δημιουργήσουμε μία εφαρμογή που να βοηθά τους χρήστες που θα την χρησιμοποιούν να βρίσκουν προτεινόμενες ταινίες γι' αυτούς, να μπορούν να βαθμολογούν ταινίες έτσι ώστε να διαμορφώνουν το προφίλ τους και να αναπροσαρμόζεται η λίστα με τις προτεινόμενες ταινίες γι' αυτούς, να μπορούν να δουν πληροφορίες για διάφορες ταινίες και να κρατούν ένα αρχείο με τις ταινίες που τους ενδιαφέρουν να τις παρακολουθήσουν στο μέλλον.

Έχοντας γνώση από ανάλυση συστημάτων είναι γνωστό ότι οι απαιτήσεις του συστήματος μπορούν να τροποποιούνται συνεχώς ή να αναπροσαρμόζονται. Το συγκεκριμένο έργο δεν απευθύνεται σε συγκεκριμένους χρήστες αλλά σε όλο το κόσμο και αυτό είναι κάτι που λήφθηκε υπόψη, παρόλα αυτά στη συγκεκριμένη εργασία αν γίνει κάποιο λάθος κατά την υλοποίηση δεν θα υπάρξει μεγάλη ζημιά σε υλικά και χρόνο.

#### **4.3 Απαιτήσεις που εξάχθηκαν**

Κατά τη διαδικασία ανάλυσης απαιτήσεων προέκυψαν οι απαιτήσεις για το συγκεκριμένο έργο. Οι απαιτήσεις απαριθμούνται πιο κάτω:

1. Είχαμε στη διάθεση μας ένα μεγάλο σετ δεδομένων από χρήστες, ταινίες και βαθμολογίες ταινιών από χρήστες στις ταινίες. Αυτό το σετ δεδομένων έπρεπε με κάποιο τρόπο να αποθηκευτεί σε ένα χώρο που να έχουμε πρόσβαση πάνω του. Μετά από διάφορες επιλογές καταλήξαμε να δημιουργήσουμε μία βάση δεδομένων όπου θα ήταν σε θέση να αποθηκεύσει τα συγκεκριμένα δεδομένα αρχικά. Ακολούθως έπρεπε να χτιστούν μια σειρά από scripts που να είναι υπεύθυνα να αποθηκεύσουν όλα τα δεδομένα στη βάση. Τα δεδομένα δεν ήταν σε μια μορφή που να ταιριάζει με τους πίνακες της βάσης που δημιουργήσαμε άρα έπρεπε να περάσουν από επεξεργασία για να αποθηκευτούν με τον

επιθυμητό τρόπο. Αυτή η διαδικασία δεν είναι μια επαναλαμβανόμενη διαδικασία όμως έπρεπε να γίνει στα αρχικά για να υπάρχει ένα σεβαστό δείγμα δεδομένων στη βάση μας.

2. Στα δεδομένα μας δεν είχαμε εικόνες των ταινιών, κάτι που θα θέλαμε να έχουμε για να τις χρησιμοποιούμε στη διεπαφή μας έτσι ώστε να είναι πιο φιλική ως προς το χρήστη. Το μόνο που είχαμε στη διάθεσή μας ήταν ένας εξωτερικός σύνδεσμος για κάθε ταινία. Η εικόνα της ταινίας υπήρχε σε αυτό τον εξωτερικό σύνδεσμο. Άρα μια άλλη απαίτηση που προέκυψε ήταν να χτιστεί ένα script που να διατρέχει ένα-ένα τους συνδέσμους των ταινιών και να αποθηκεύει τις εικόνες των ταινιών τοπικά για να μπορούμε να τις χρησιμοποιούμε όποτε θέλουμε.
3. Στην απαίτηση 1 κάναμε αναφορά στα αρχικά μας δεδομένα και πως αυτά θα αποθηκευτούν στο σύστημα μας. Τα δεδομένα μας όμως θέλαμε να μπορούν να επεκτείνονται συνεχώς, άρα μια απαίτηση ήταν να μπορεί κάποιος μέσω του συστήματος να εισάγει καινούρια αντικείμενα(ταινίες) στη βάση δεδομένων. Επίσης όσο αφορά τους χρήστες, να μπορούν να εγγράφονται νέοι χρήστες στη βάση.
4. Η επόμενη απαίτηση είναι μια από τις πιο σημαντικές. Συγκεκριμένα είναι το χτίσιμο μια λειτουργίας που να μπορεί ο χρήστης να βλέπει τις προτεινόμενες του ταινίες από το σύστημα. Η λειτουργία αυτή παρουσιάζει στο χρήστη το τίτλο της ταινίας, τη ημερομηνία κυκλοφορίας της, την εικόνα της και ένα κουμπί για να ανοίξει ένας εξωτερικός σύνδεσμος της ταινίας που να δείχνει περισσότερες πληροφορίες όπως την υπόθεση της.
5. Άλλη απαίτηση ήταν να μπορεί ο χρήστης να βαθμολογεί ταινίες και να καταχωρούνται στο σύστημα σαν καινούρια βαθμολογία από το συγκεκριμένο χρήστη σε μια συγκεκριμένη ταινία.
6. Επίσης μία άλλη απαίτηση ήταν ότι ο χρήστης έπρεπε να έχει τη δυνατότητα να μπορεί να καταχωρίσει μια ταινία σε μια λίστα με ταινίες που πρόκειται να δει. Ακόμα αυτές οι ταινίες («watchlisted movies») να του εμφανίζονται περιοδικά από το σύστημα για να μην ξεχάσει να τις βαθμολογήσει.
7. Μια από τις πιο σημαντικές απαιτήσεις ήταν ότι έπρεπε με κάποιο τρόπο να γίνει ένα integration του συστήματος με το εξωτερικό σύστημα που παράγει τις συστάσεις. Αυτή είναι μια επαναλαμβανόμενη διαδικασία που θα έπρεπε να

ξεκινά από το δικό μας σύστημα τη πλατφόρμα που παράγει τις συστάσεις και αφού ολοκληρωθεί αυτή η διαδικασία να ενημερώνεται η βάση με τα νέα δεδομένα. Είναι μια σειρά διαδικασιών που έπρεπε να εκτελούνται σειριακά και να μην προκύπτουν σφάλματα στη πορεία γιατί θα ήταν καταστροφικό για το σύστημα.

8. Από την προηγούμενη απαίτηση και την απαίτηση 3 προκύπτει μια νέα απαίτηση. Η απαίτηση του να υπάρχουν δύο ειδών χρήστες. Εκτός από τον απλό χρήστη να υπάρχει και μία σελίδα διαχειριστή για να εκτελεί κάποιες λειτουργίες που δεν θα θέλαμε να έχουν πρόσβαση πάνω τους οι απλοί χρήστες. Η πιο σημαντική είναι να μπορεί να ξεκινά τη διαδικασία δημιουργίας των συστάσεων.
9. Όπως προαναφέραμε η διαδικασία δημιουργίας συστάσεων θα είναι μια επαναλαμβανόμενη διαδικασία που πρέπει να εκτελείται πολλές φορές τη μέρα. Αν και όπως αναφέραμε στη προηγούμενη απαίτηση αυτή η διαδικασία θα μπορεί να ξεκινά με την εντολή ενός διαχειριστή δεν είναι αρκετό. Δεν θα θέλαμε να αφήσουμε αυτή την ευθύνη μόνο σε ανθρώπινο παράγοντα. Αυτή η διαδικασία έπρεπε να αυτοματοποιηθεί να εκτελείται χωρίς την ανάγκη ανθρώπινου παράγοντα.

## Κεφάλαιο 5

### Σχεδιασμός συστήματος, υλοποίηση

---

|                                                                             |    |
|-----------------------------------------------------------------------------|----|
| 5.1 Actors                                                                  | 30 |
| 5.2 Περιγραφή Use Case                                                      | 31 |
| 5.2.1 Γενική περίπτωση χρήσης του συστήματος από το χρήστη                  | 31 |
| 5.2.2 Γενική περίπτωση χρήσης του συστήματος από το διαχειριστή             | 32 |
| 5.2.3 Περίπτωση Χρήσης του See Overall Recommendations                      | 33 |
| 5.2.4 Περίπτωση Χρήσης όταν εμφανίζονται τα watchlisted movie               | 33 |
| 5.2.5 Περίπτωση Χρήσης όταν θέλει ο χρήστης να βαθμολογήσει μια ταινία      | 34 |
| 5.3 Υλοποίηση των λειτουργιών της διαδικτυακής εφαρμογής για το χρήστη      | 35 |
| 5.4 Υλοποίηση των λειτουργιών της διαδικτυακής εφαρμογής για το διαχειριστή | 41 |

---

#### 5.1 Actors

Στην εφαρμογή εμφανίζονται σαν actors ο απλός χρήστης και ο διαχειριστής (administrator). Και οι δύο είναι χρήστες του συστήματος που αλληλοεπιδρούν με τις διάφορες λειτουργίες του συστήματος.

#### Actor Description:

Απλός Χρήστης:

|                 |                                                                                        |
|-----------------|----------------------------------------------------------------------------------------|
| Περιγραφή       | Πρόκειται για ένα χρήστη που θα χρησιμοποιεί τις λειτουργίες που προσφέρει το σύστημα. |
| Ψευδώνυμο       | User                                                                                   |
| Κληρονομικότητα | -                                                                                      |
| Είδος Χρήστη    | Είναι ενεργός και αποτελεί άνθρωπο.                                                    |

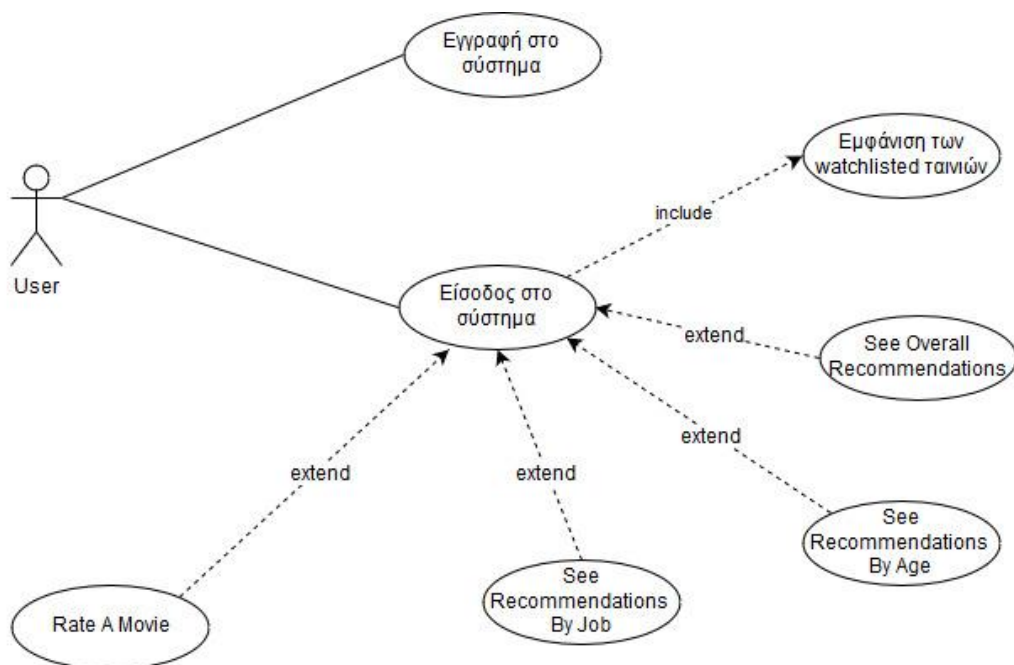
Διαχειριστής:

|                 |                                                           |
|-----------------|-----------------------------------------------------------|
| Περιγραφή       | Πρόκειται για ένα χρήστη που θα διαχειρίζεται το σύστημα. |
| Ψευδώνυμο       | Admin                                                     |
| Κληρονομικότητα | -                                                         |
| Είδος Χρήστη    | Είναι ενεργός και αποτελεί άνθρωπο.                       |

## 5.2 Περιγραφή Use Case

Παρουσιάζονται όλες οι περιπτώσεις χρήσης δείχνοντας τις μεταξύ τους διασυνδέσεις, εξαρτήσεις και επικοινωνίες.

### 5.2.1 Γενική περίπτωση χρήσης του συστήματος από το χρήστη

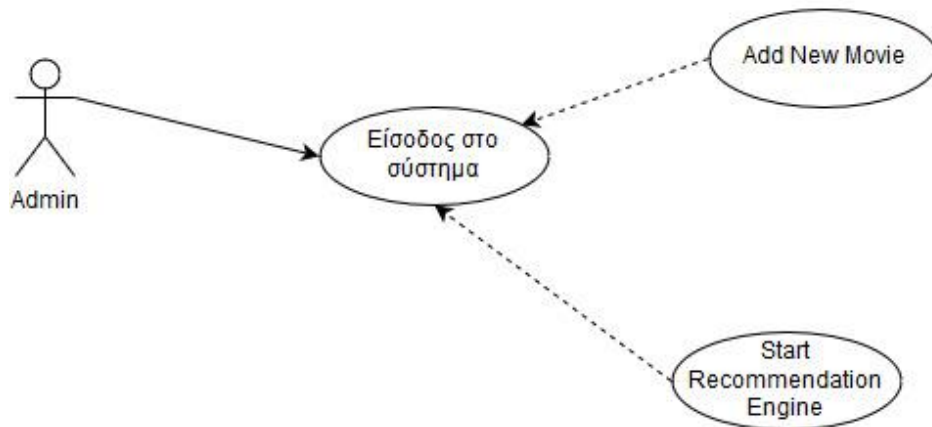


Σχήμα 5. 1



**Περιγραφή:**

Ο χρήστης κάνει εγγραφή στο σύστημα ή είσοδο στο σύστημα και έχει στη διάθεση του όλες τις λειτουργίες του συστήματος.

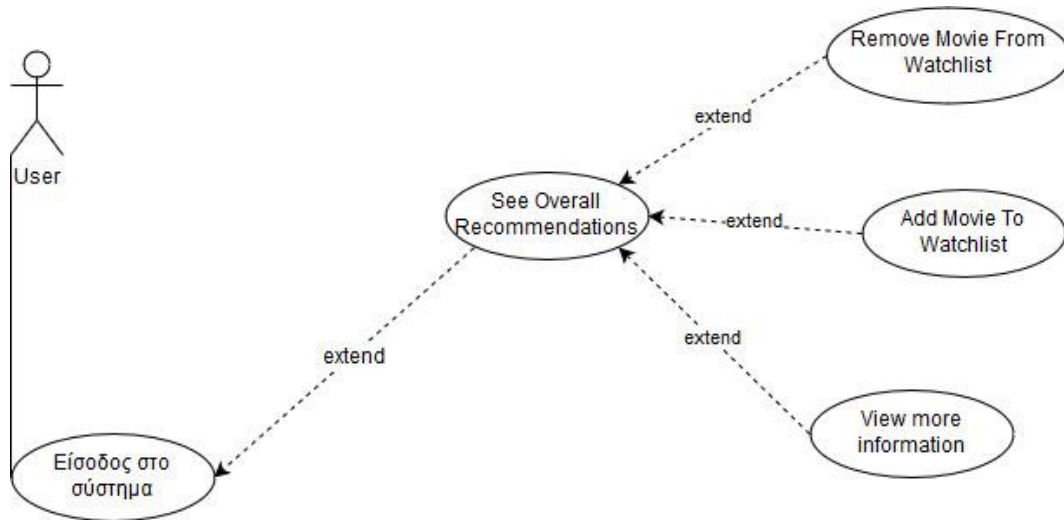
**5.2.2 Γενική περίπτωση χρήσης του συστήματος από τον διαχειριστή**

Σχήμα 5. 2

**Περιγραφή:**

Ο διαχειριστής μπαίνει στο σύστημα και έχει στη διάθεση του όλες τις λειτουργίες της αρμοδιότητάς του.

### 5.2.3 Περίπτωση Χρήσης του See Overall Recommendations

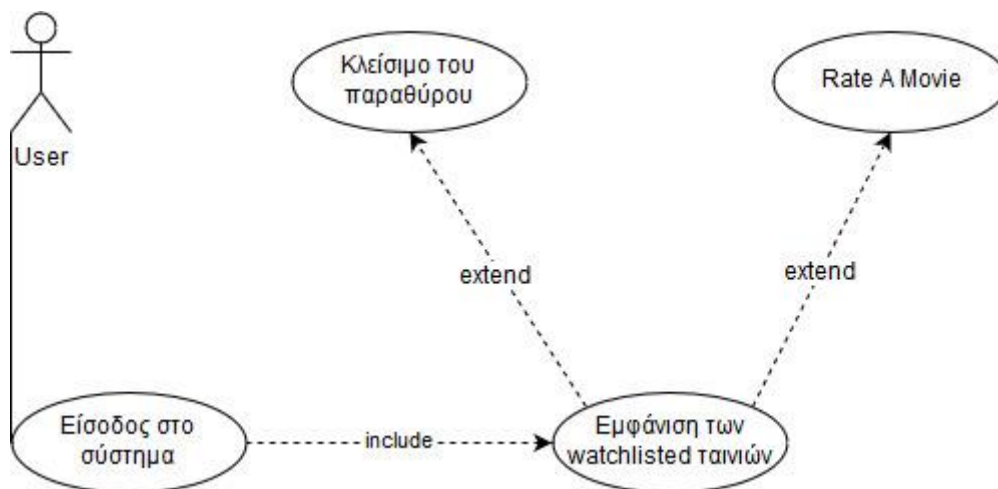


Σχήμα 5. 3

#### Περιγραφή:

Ο χρήστης κάνει είσοδο στο σύστημα και θέλει να δει τις συστάσεις του. Βλέποντας τα μπορεί μια συγκεκριμένη ταινία να την προσθέσει στη λίστα με τις ταινίες που θέλει να δει και αντίστοιχα να την αφαιρέσει εάν την έχει προσθέσει πριν. Επίσης μπορεί να δει περισσότερες πληροφορίες για καθεμία από τις ταινίες.

### 5.2.4 Περίπτωση Χρήσης όταν εμφανίζονται τα watchlisted movies

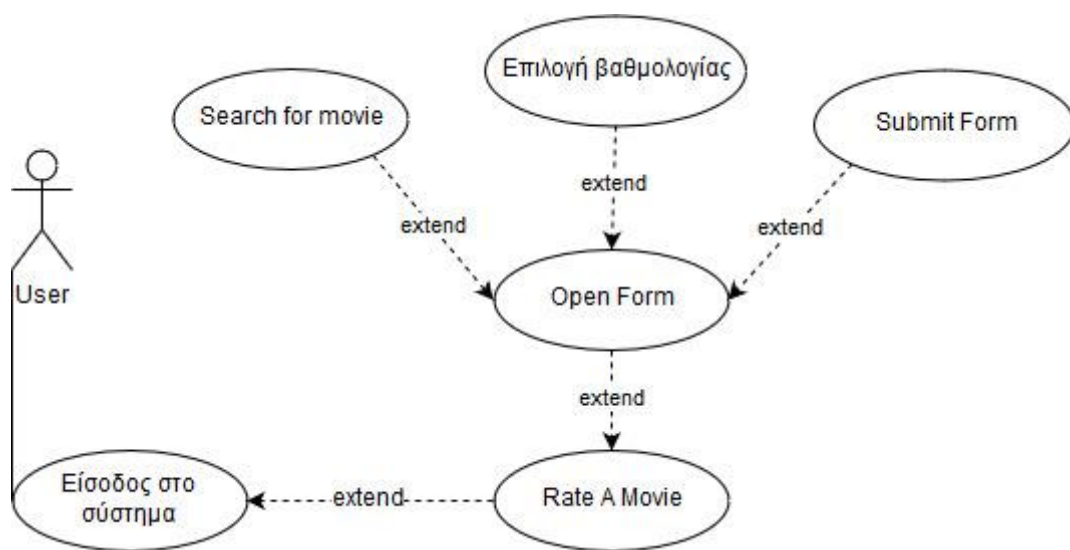


Σχήμα 5. 4

### Περιγραφή:

Ο χρήστης κάνει είσοδο στο σύστημα και αυτόματα του εμφανίζονται οι ταινίες που έχει ο ίδιος προσθέσει στο watchlist. Ο χρήστης έχει δύο επιλογές, να κάνει βαθμολογήσει μία ή περισσότερες από τις ταινίες εάν τις έχει δει ή να κλείσει το παράθυρο.

#### 5.2.5 Περίπτωση Χρήσης όταν θέλει ο χρήστης να βαθμολογήσει μια ταινία



Σχήμα 5. 5

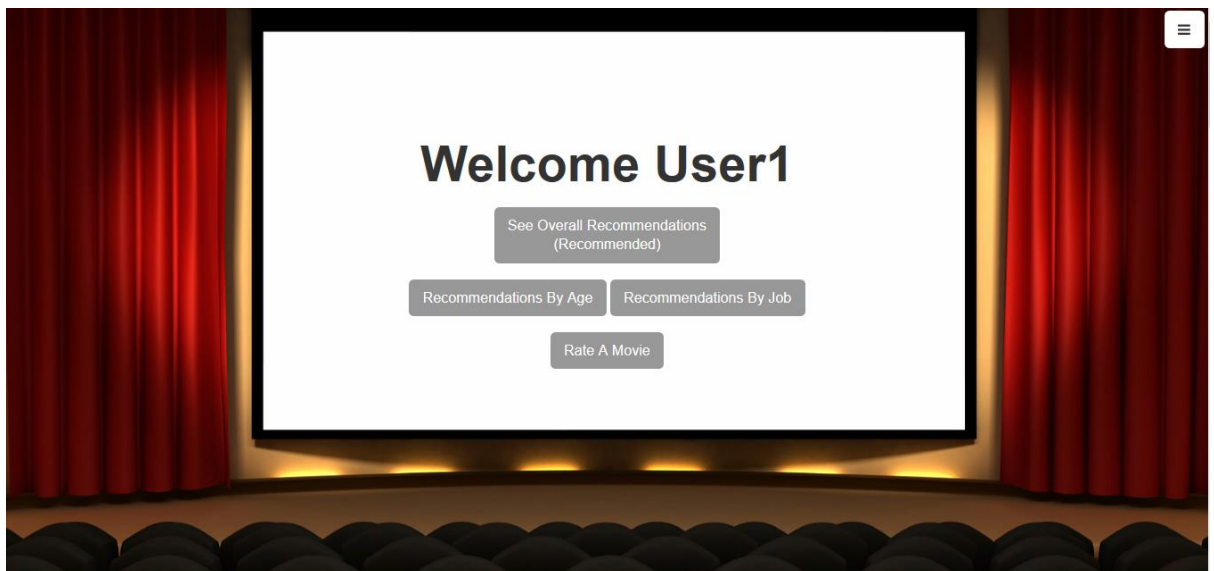
### Περιγραφή:

Ο χρήστης κάνει είσοδο στο σύστημα και επιλέγει τη λειτουργία να βαθμολογήσει μία ταινία. Ακολούθως ανοίγει μία φόρμα για να αναζητήσει ο χρήστης την ταινία να επιλέξει βαθμολογία (1-5) και να πατήσει το κουμπί submit.

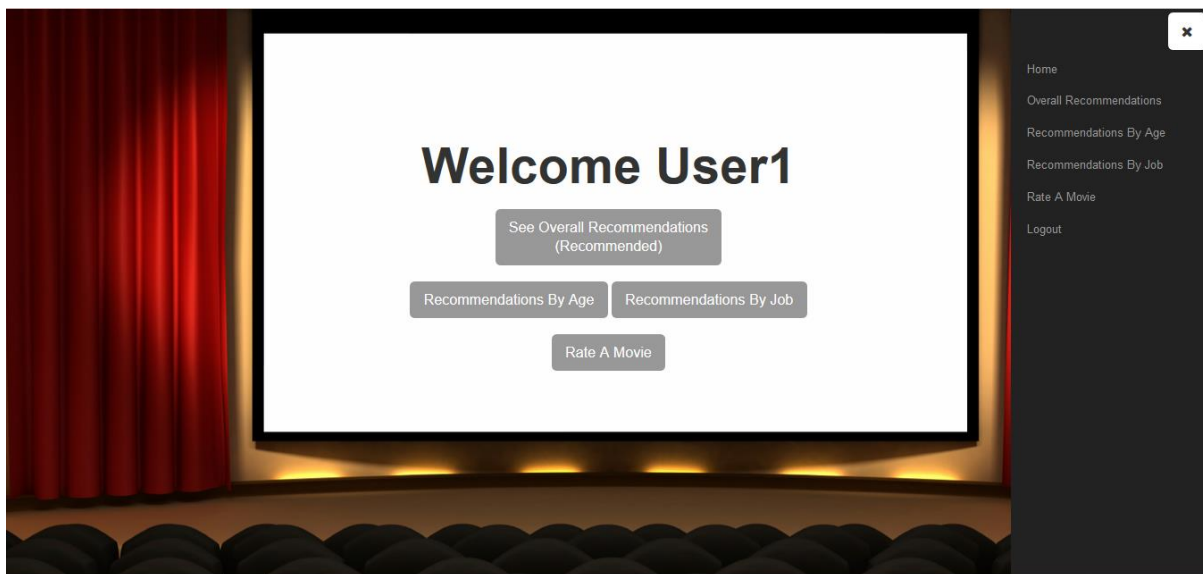
### 5.3 Υλοποίηση των λειτουργιών της διαδικτυακής εφαρμογής για το χρήστη

Η διαδικτυακή εφαρμογή προορίζεται για όλων των ειδών χρήστες. Άρα η εφαρμογή έπρεπε να χαρακτηρίζεται από απλότητα για να μπορούν να τη χρησιμοποιήσουν τόσο έμπειροι χρήστες όσο και άπειροι, μεγάλης ηλικίας αλλά και νεαροί.

Οι λειτουργίες που παρέχει η εφαρμογή εμφανίζονται σε αρχικά κουμπιά μόλις γίνει είσοδος στη σελίδα αλλά και σε ένα μενού που ανοίγει και κλίνει από τα δεξιά. Επιλέγοντας μία λειτουργία από το μενού ή από τα διαθέσιμα κουμπιά ο χρήστης παραπέμπεται στο σημείο που βρίσκεται η λειτουργία που επέλεξε.



Σχήμα 5.6 – Page Header



**Σχήμα 5.7 – Expanded Menu On The Right**

Επιλέγοντας ο χρήστης την επιλογή «See Overall Recommendations» παραπέμπεται στο σημείο της σελίδας που βρίσκονται οι δικές του συστάσεις.

Παρουσιάζονται 8 ταινίες σε ένα carousel. Το carousel όσο αφορά τη ταινία περιέχει το τίτλο της, την εικόνα της και την ημερομηνία κυκλοφορίας της. Εκτός από αυτά υπάρχει ένα κουμπί «See More» όπου πατώντας το ανοίγει ένα εξωτερικό σύνδεσμο της ταινίας για να μπορεί να δει ο χρήστης περισσότερες πληροφορίες γι' αυτή.

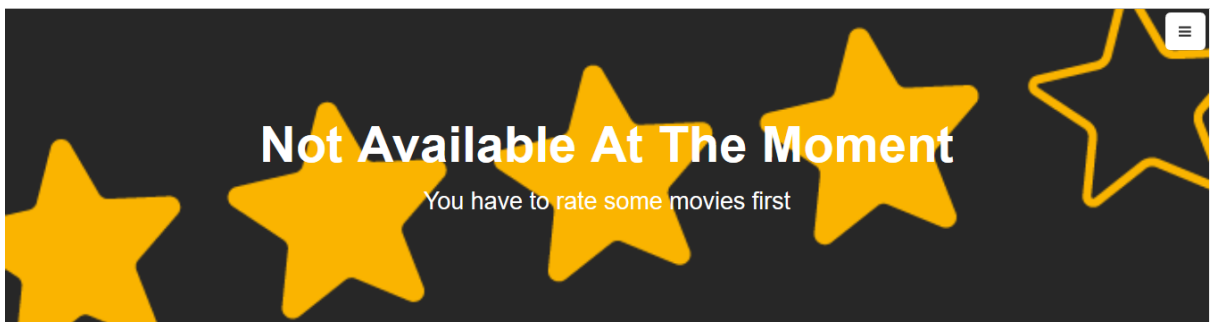
Οι ταινίες είναι ταξινομημένες βάσει του ποια ταινία ταιριάζει καλύτερα στο χρήστη. Για το λόγο αυτό στο πάνω μέρος του carousel υπάρχει τίτλος «Recommendation 1» για τη πρώτη ταινία «Recommendation 2» για τη δεύτερη ταινία κ.ο.κ. Επίσης υπάρχει μία μπάρα που γεμίζει ανάλογα με το πόσο ταιριάζει η ταινία στο χρήστη. Δηλαδή στη πρώτη ταινία η μπάρα θα είναι γεμάτη και στην όγδοη θα είναι πού λίγο γεμάτη.

Τέλος υπάρχει ένα άσπρο κουμπί «Add To Watchlist» όπου πατώντας το ο χρήστης θα προσθέσει τη ταινία στο «Watchlist» του και το κουμπί θα γίνει πράσινο. Επιτρέπεται και η αντίθετη διαδικασία.



**Σχήμα 5.8 – Carousel with recommendations**

Αξίζει να σημειωθεί πώς για νέους χρήστες που δεν έχουν προλάβει ακόμα να κάνουν κάποιες βαθμολογίες η συγκεκριμένη λειτουργία δεν μπορεί να τους εμφανιστεί. Για το λόγο αυτό στους συγκεκριμένους χρήστες, στη θέση αυτής της λειτουργίας τους εμφανίζεται ένα μήνυμα για να κάνουν τα δικά τους rates έτσι ώστε στο μέλλον να μπορούν να χρησιμοποιήσουν τη συγκεκριμένη λειτουργία.



**Σχήμα 5.9 – Message For New Users**

Για αυτή τη κατηγορία χρηστών, τους νέους χρήστες, επειδή δεν θέλαμε να μην μπορούν να δουν καθόλου συστάσεις μέχρι να κάνουν τις δικές τους βαθμολογήσεις προστέθηκαν κάποια άλλα πιο απλά είδη συστάσεων. Πιο συγκεκριμένα του παρουσιάζουμε στο χρήστη τις τρεις πιο δημοφιλείς ταινίες από άτομα της ίδιας ηλικίας με αυτόν και τις τρεις πιο δημοφιλείς ταινίες από άτομα του ιδίου επαγγέλματος με αυτόν.

People of the same age (22) rated with 5 stars the below movies



The Silence of the Lambs (1991)

See More



Star Wars (1977)

See More

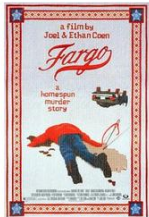


Titanic (1997)

See More

Σχήμα 1.10 – Recommendations By Age

People with the same job (executive) rated with 5 stars the below movies



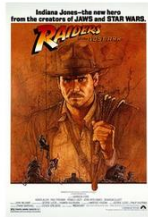
Fargo (1996)

See More



Star Wars (1977)

See More



Raiders of the Lost Ark (1981)

See More

Σχήμα 5.11 – Recommendations By Job

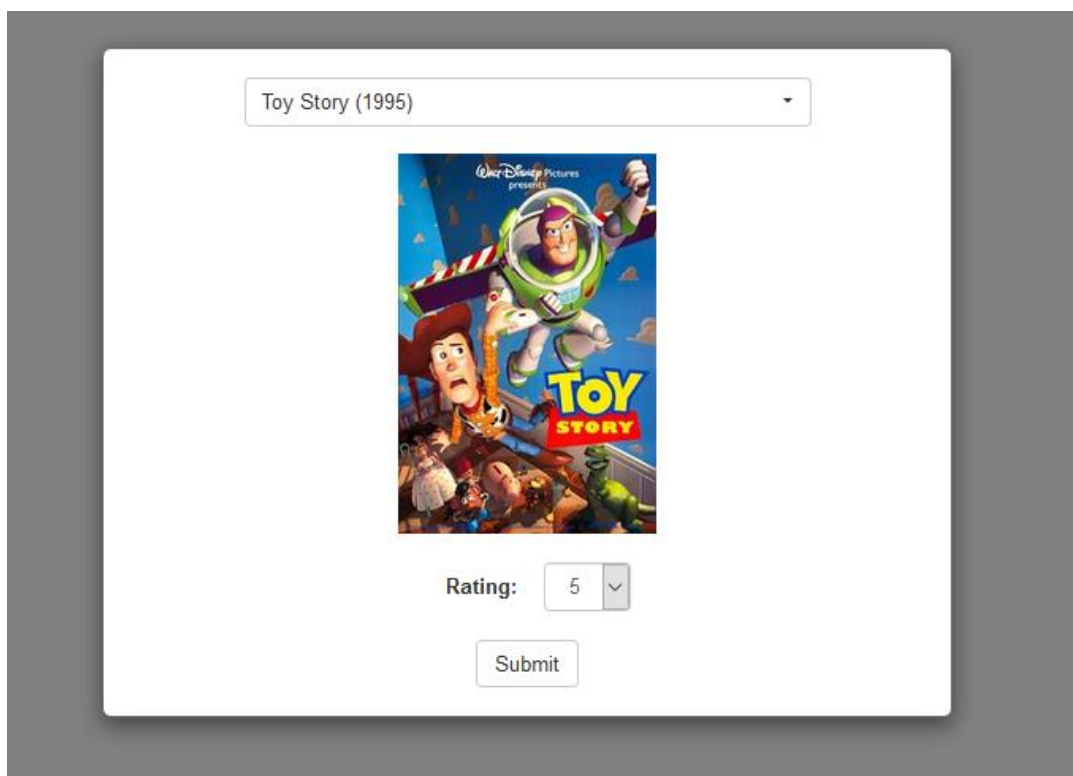
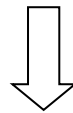
Μια άλλη πολύ σημαντική λειτουργία είναι η λειτουργία της βαθμολόγησης ταινίας, «Rate A Movie». Ο χρήστης το μόνο που έχει να κάνει είναι να μεταφερθεί στο σημείο που βρίσκεται η λειτουργία, να πατήσει πάνω σε μια clickable εικόνα και να του ανοίξει μια φόρμα σε ένα modal window όπου περιέχει ένα search box για να αναζητήσει τη

ταινία που θέλει, ένα dropdown για να επιλέξει τη βαθμολογία που θέλει και κουμπί submit για να καταχωρίσει την επιλογή του.

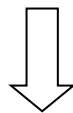
## Rate A Movie



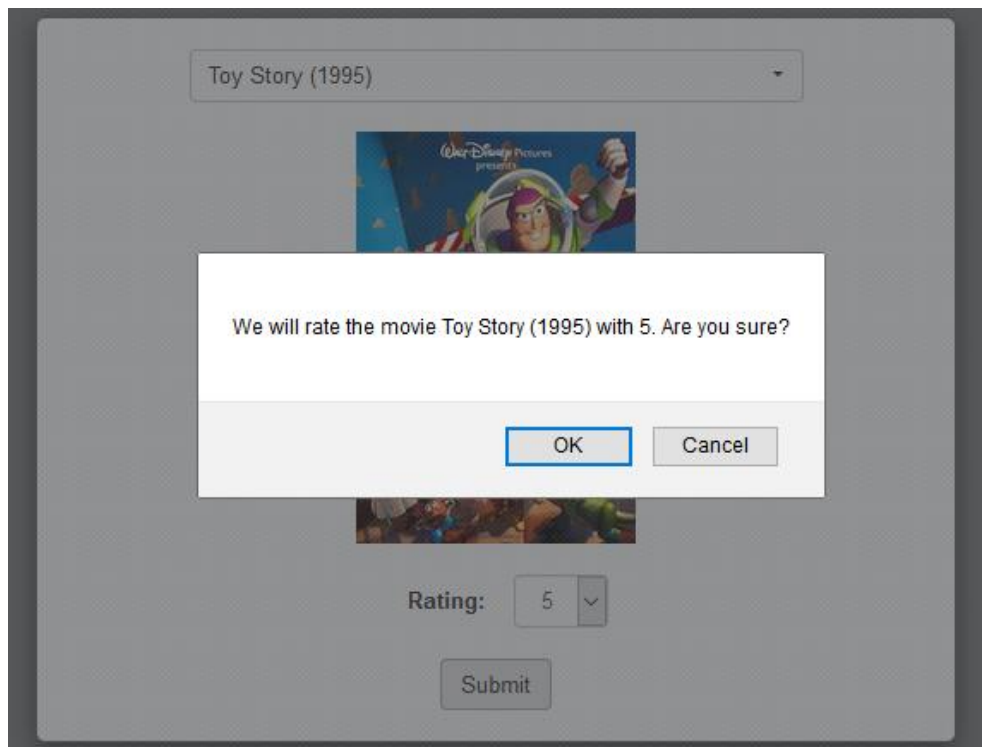
Σχήμα 5.12 – Clickable Image

A modal window for rating the movie Toy Story (1995). It includes a dropdown menu for the movie title, a movie poster, a rating selector set to 5, and a submit button.

Σχήμα 5.13 – Form In A Modal Window

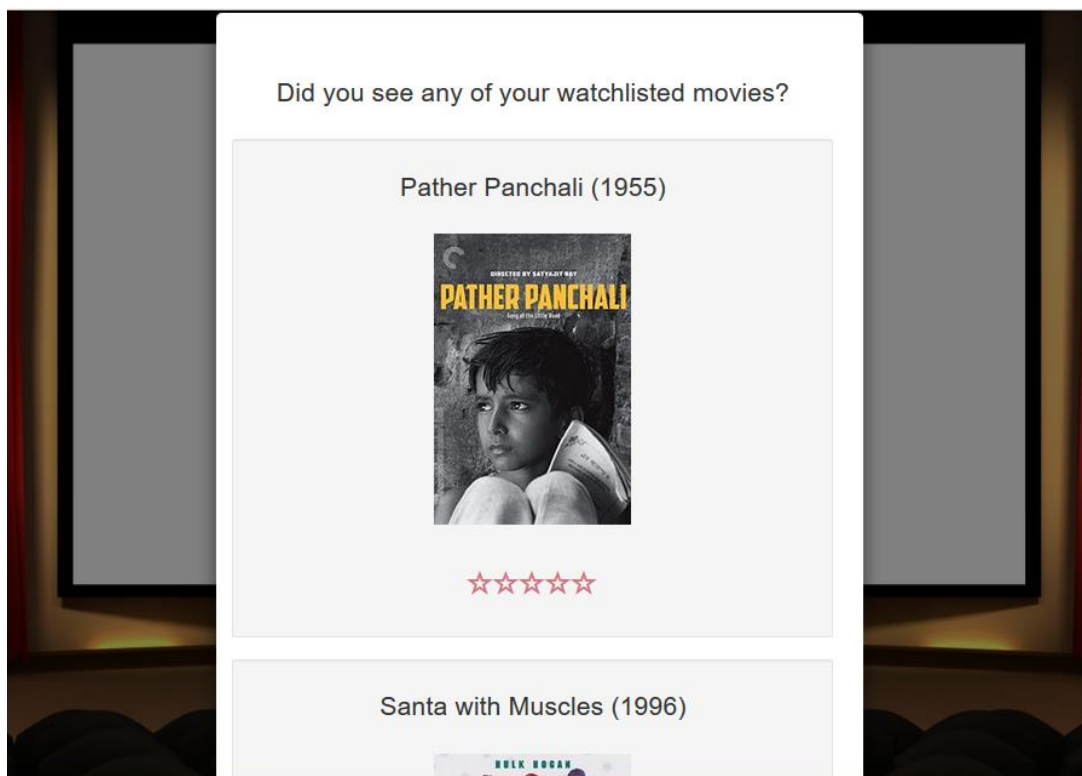






**Σχήμα 5.14 – Alert Window To Complete The Action**

Ο χρήστης μπορεί να βαθμολογήσει μια ταινία και από μία άλλη λειτουργία που υλοποιήθηκε στο σύστημα. Μόλις κάνει είσοδο στο σύστημα του εμφανίζονται ταινίες που έχει στο watchlist του σε ένα modal window για να καταχωρίσει σε αυτές τις ταινίες βαθμολογία απ' ευθείας. Ο λόγος που έγινε αυτή η λειτουργία είναι γιατί αυτές τις ταινίες έχει αυξημένες πιθανότητες να τις έχει δει και να θέλει να της βαθμολογήσει. Έτσι με αυτή τη λειτουργία γλιτώνει πολύτιμο χρόνο επειδή θα αποφύγει τη προηγούμενη διαδικασία βαθμολόγησης.

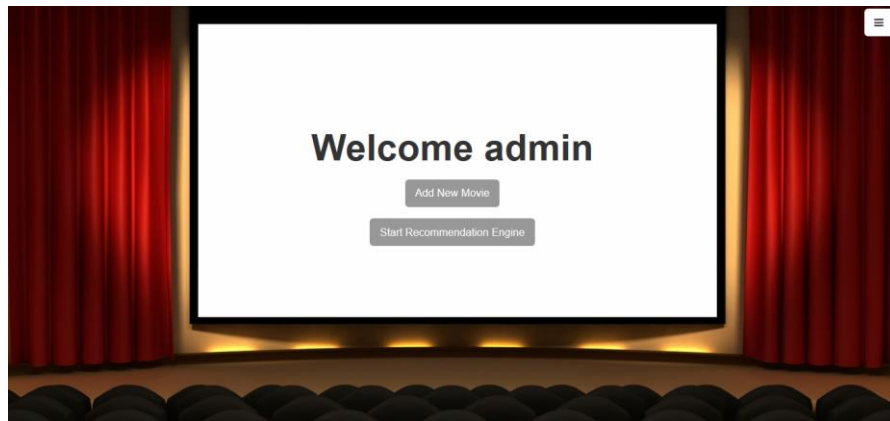


Σχήμα 5.15 – Modal With User's Watchlisted Movies

#### 5.4 Υλοποίηση των λειτουργιών της διαδικτυακής εφαρμογής για το διαχειριστή

Όπως έχουμε προαναφέρει η εφαρμογή παρέχει ξεχωριστή υλοποίηση για το διαχειριστή. Ο διαχειριστής μπορεί να εκτελέσει λειτουργίες που δεν μπορούν οι απλοί χρήστες.

Η διεπαφή ακολουθεί πανομοιότυπο σχεδιασμό με αυτή του απλού χρήστη αλλά περιέχει άλλες λειτουργίες. Οι λειτουργίες που έχει στη διάθεση του ο διαχειριστής είναι να κάνει καταχώριση στο σύστημα καινούρια ταινία, και να ξεκινήσει τη διαδικασία που παράγει τις συστάσεις.



Σχήμα 5.16 – Admin's Header

Για να καταχωρίσει μια καινούρια ταινία αρκεί να επιλέξει την επιλογή «Add New Movie» για να μεταφερθεί στο χώρο που βρίσκεται η φόρμα που πρέπει να τη συμπληρώσει. Όταν την συμπληρώσει πρέπει να πατήσει το κουμπί submit για να ολοκληρωθεί η διαδικασία.

### Add New Movie

Movie Title

Year      Month      Day

Imdb URL

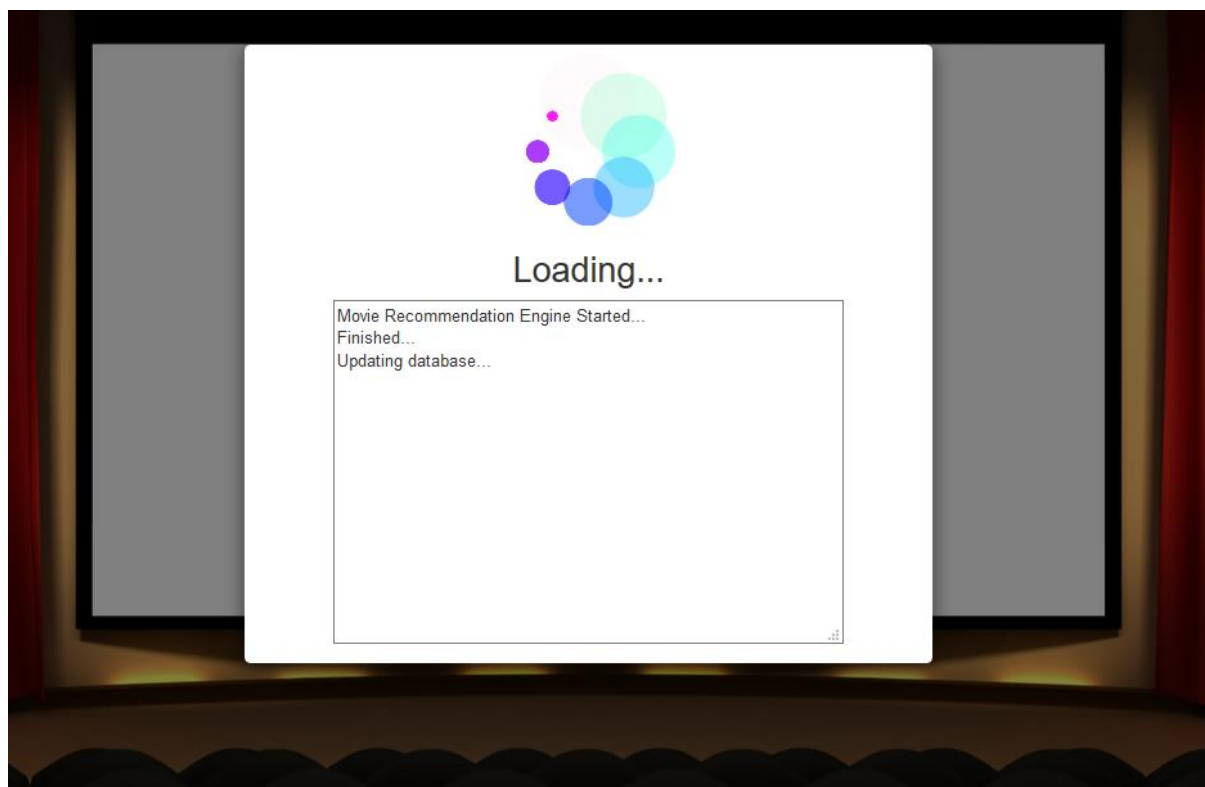
|                                      |                                    |                                    |                                     |                                 |                                  |
|--------------------------------------|------------------------------------|------------------------------------|-------------------------------------|---------------------------------|----------------------------------|
| <input type="checkbox"/> Action      | <input type="checkbox"/> Adventure | <input type="checkbox"/> Animation | <input type="checkbox"/> Children's | <input type="checkbox"/> Comedy | <input type="checkbox"/> Crime   |
| <input type="checkbox"/> Documentary | <input type="checkbox"/> Drama     | <input type="checkbox"/> Fantasy   | <input type="checkbox"/> Film-Noir  | <input type="checkbox"/> Horror | <input type="checkbox"/> Musical |
| <input type="checkbox"/> Mystery     | <input type="checkbox"/> Romance   | <input type="checkbox"/> Sci-Fi    | <input type="checkbox"/> Thriller   | <input type="checkbox"/> War    | <input type="checkbox"/> Western |

Submit

Σχήμα 5.17 – Add New Movie Form

Για να ξεκινήσει τη διαδικασία παραγωγής των συστάσεων πατώντας το κουμπί ανοίγει ένα modal window που παραμένει ανοικτό μέχρι να ολοκληρωθεί η διαδικασία. Είναι μια διαδικασία που θέλει κάποια δευτερόλεπτα να ολοκληρωθεί, για το λόγο αυτό υπάρχει και ένας χώρος που ενημερώνει το χρήστη σε πιο σημείο βρίσκεται η διαδικασία.



**Σχήμα 5.18 – Recommendations Generation Procedure**

## Κεφάλαιο 6

### Αξιολόγηση, Συμπεράσματα και Μελλοντική Εργασία

---

|                           |    |
|---------------------------|----|
| 6.1 Αξιολόγηση Συστήματος | 44 |
| 6.2 Συμπεράσματα          | 49 |
| 6.3 Μελλοντική Εργασία    | 51 |

---

#### 6.1 Αξιολόγηση Συστήματος

Στην ανάπτυξη πληροφοριακών συστημάτων είναι πολύ σημαντικό να αξιολογούνται τα αποτελέσματα. Είναι μια φάση πολύ σημαντική για να αποφασιστεί κατά πόσο επιτεύχθηκε το ζητούμενο και αν οι λειτουργίες του συστήματος ανταποκρίνονται στις αρχικές απαιτήσεις που τέθηκαν. Σε αυτή τη φάση θα εντοπιστούν τα πιθανά λάθη που μπορεί να προκύψουν κατά τη φάση της υλοποίησης. Εντοπίζονται ακόμα και οι λειτουργίες οι οποίες είναι σύνθετες και όχι απόλυτα κατανοητές από το χρήστη. Κάποιες τεχνικές για να αξιολογήσεις ένα πληροφοριακό σύστημα είναι μέσω συνεντεύξεων, ερωτηματολογίων, παρουσιάσεων κ.α.

Για αξιολόγηση του δικού μας πληροφοριακού συστήματος, της διαδικτυακής εφαρμογής, δημιουργήθηκε ένα ερωτηματολόγιο για να αξιολογηθεί η ευχρηστία του και η απλότητά του ως προς τους χρήστες και ως επίσης κατά πόσο επιτεύχθηκε το ζητούμενο, το να γίνονται σωστές συστάσεις ταινιών από το σύστημα στους χρήστες.

Το ερωτηματολόγιο αποτελείται από 9 ερωτήσεις και απαντήθηκε από 15 άτομα. Οι ερωτήσεις που περιείχε το ερωτηματολόγιο είναι οι ακόλουθες:

1. Ήταν εύκολη η αλληλεπίδραση σας με το σύστημα;
2. Ήταν εύκολη η αναζήτηση προτεινόμενων ταινιών από το σύστημα;
3. Ήταν εύκολη η γενική αναζήτηση ταινιών;
4. Η ταινίες μου σας προτάθηκαν ήταν της αρεσκείας σας;

5. Η σειρά προτεραιότητας ήταν η κατάλληλη για σας;
6. Πόσες από τις προτεινόμενες ταινίες σας ήταν αδιάφορες;
7. Η πληροφορίες που μπορούσατε να βρείτε για κάθε ταινία ήταν αρκετές;
8. Ήταν εύκολη η λειτουργία βαθμολόγησης ταινιών;
9. Σας διευκόλυνε η λειτουργία “Watchlist”;

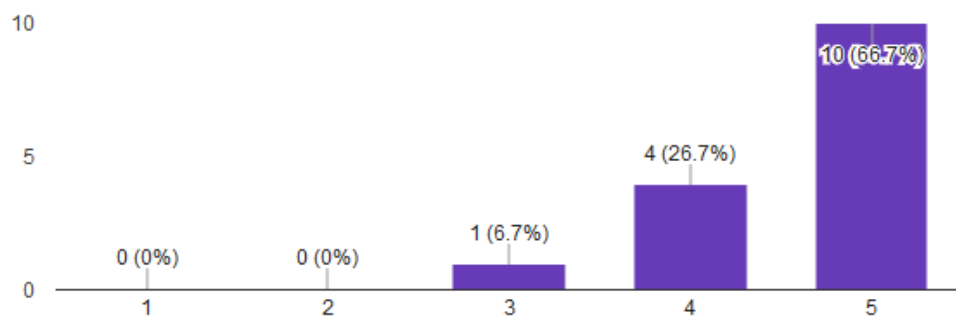
Τα αποτελέσματα που εξάχθηκαν ήταν αρκετά ενθαρρυντικά για όλες τις ερωτήσεις κάτι που μας οδήγησε στο συμπέρασμα ότι το σύστημά μας είναι απλό ως προς τη χρήση του, οι λειτουργίες είναι αρκετά ξεκάθαρες και οι συστάσεις ταινιών είναι σε μεγάλο βαθμό επιτυχείς.

### **Αποτελέσματα ερωτηματολογίου:**

Οι 8 από τις 9 ερωτήσεις ήταν απαντήσεις από το 1 μέχρι το 5. Το 5 αντιστοιχεί με «Συμφωνώ απόλυτα» και το 1 αντιστοιχεί με «Διαφωνώ απόλυτα».

**Ήταν εύκολη η αλληλεπίδραση σας με το σύστημα;**

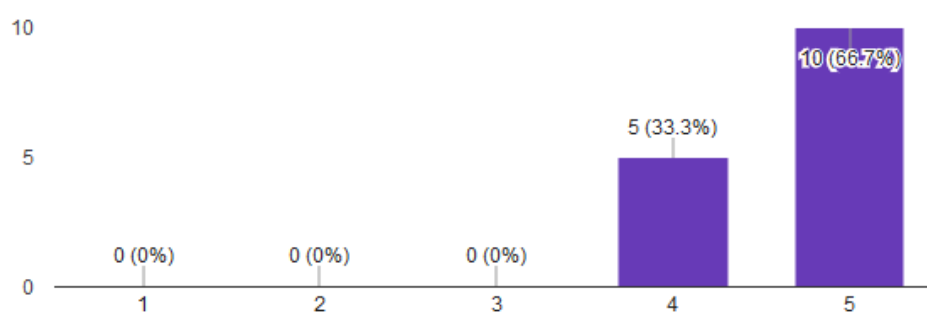
15 responses



**Σχήμα 6.1 – Ερώτηση 1**

Ήταν εύκολη η αναζήτηση προτεινόμενων ταινιών από το σύστημα;

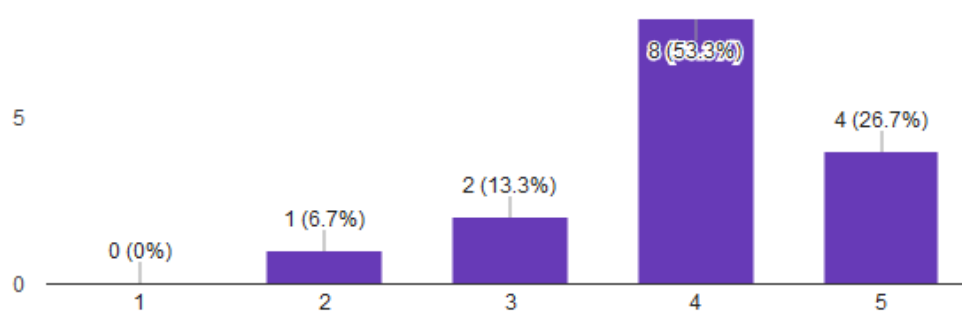
15 responses



Σχήμα 6.2 – Ερώτηση 2

Ήταν εύκολη η γενική αναζήτηση ταινιών;

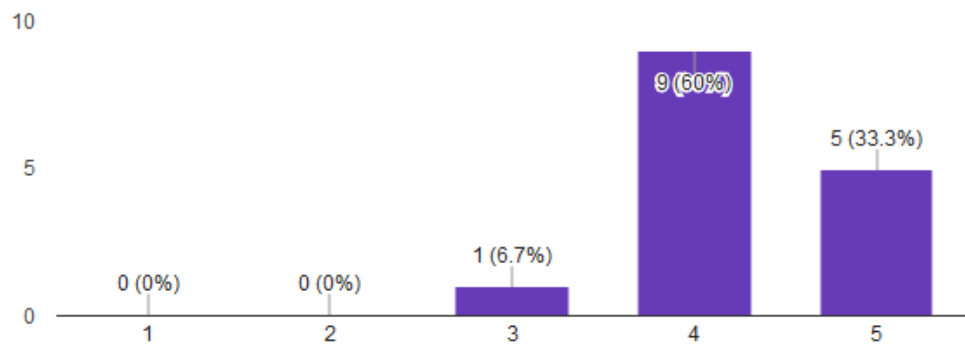
15 responses



Σχήμα 6.3 – Ερώτηση 3

Η ταινίες μου σας προτάθηκαν ήταν της αρεσκείας σας;

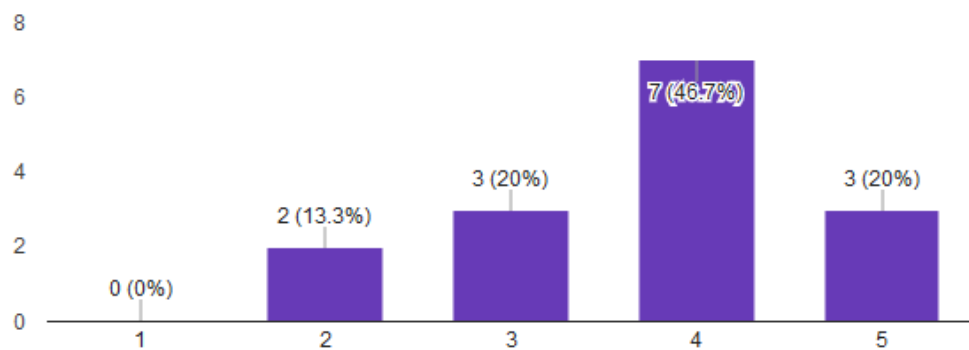
15 responses



Σχήμα 6.4 – Ερώτηση 4

Η σειρά προτεραιότητας ήταν η κατάλληλη για σας;

15 responses

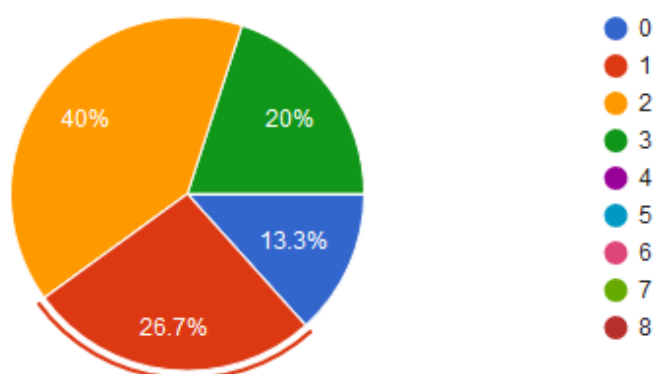


Σχήμα 6.5 – Ερώτηση 5



Πόσες από τις προτεινόμενες ταινίες σας ήταν αδιάφορες;

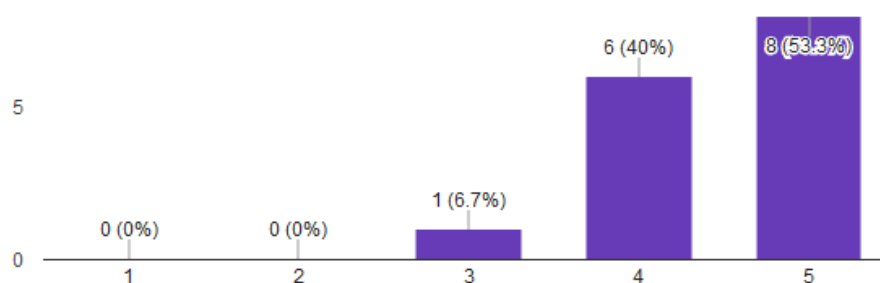
15 responses



Σχήμα 6.6 – Ερώτηση 6

Η πληροφορίες που μπορούσατε να βρείτε για κάθε ταινία ήταν αρκετές;

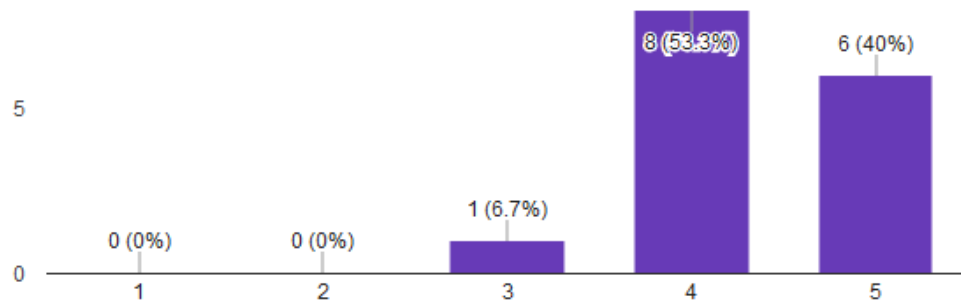
15 responses



Σχήμα 6.7 – Ερώτηση 7

### Ήταν εύκολη η λειτουργία βαθμολόγησης ταινιών;

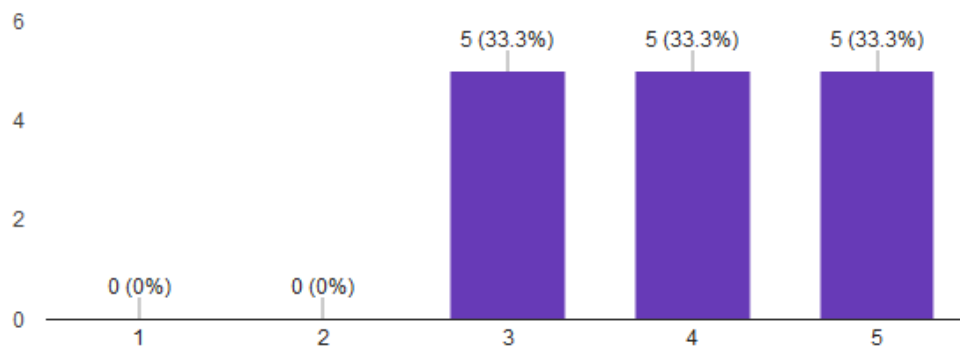
15 responses



Σχήμα 6.8 - Ερώτηση 8

### Σας διευκόλυνε η λειτουργία "Watchlist";

15 responses



Σχήμα 6.9 – Ερώτηση 9

## 6.2 Συμπεράσματα

Ο στόχος της παρούσας διπλωματικής εργασίας είναι η μελέτη των Συστημάτων Συστάσεων με επίγνωση συγκεκριμένου και η υλοποίηση μιας διαδικτυακής εφαρμογής movie recommender.

Κατά το στάδιο της μελέτης των Συστημάτων Συστάσεων με επίγνωση συγκεκριμένου, κατάλαβα πόσο ενεργός και αναπτυσσόμενος τομέας της Πληροφορικής είναι αφού

απασχολεί πολύ την επιστημονική κοινότητα. Έχουν δημιουργηθεί πολύ ικανοί αλγόριθμοι που αξιοποιούν το συγκεκριμένο και να κάνει αυτούς τους αλγορίθμους να είναι πολύ πιο εύστοχοι από τους απλούς αλγορίθμους συστάσεων.

Δεν είναι τυχαίο που τεράστιες εταιρίες παρέχουν Συστάσεις με επίγνωση συγκριμένου στους χρήστες των ιστοσελίδων τους και προσπαθούν να αυξάνουν το συγκεκριμένο όλο και περισσότερο. Ο λόγος είναι ότι επιτυγχάνουν το ζητούμενο που είναι να παρέχουν στους χρήστες τους επιτυχημένες συστάσεις και με αυτό το τρόπο υπερυψώνουν το επίπεδο της ιστοσελίδας τους.

Κατά την υλοποίηση της διαδικτυακής εφαρμογής είχαμε στη διάθεση μας το CarsKit Framework που είναι μία πλατφόρμα ανοιχτού κώδικα γραμμένη σε Java που έχει υλοποιημένους πολλούς αλγορίθμους συστάσεων. Η πρόκληση για μας ήταν να καταφέρουμε να ενώσουμε την διαδικτυακή εφαρμογή με το CarsKit Framework για να μην περιοριζόμαστε ως προς το πιο αλγόριθμο να χρησιμοποιήσουμε. Αν δεν ήταν εφικτή η ένωση των δύο συστημάτων πολύ πιθανόν να οδηγούμασταν σε υλοποίηση ενός αλγορίθμου που δεν ήταν το επιθυμητό. Μετά από μελέτη που έγινε ως προς τα δύο συστήματα και τους διάφορους τρόπους καταλήξαμε στο συμπέρασμα πως η ένωση των δύο συστημάτων είναι εφικτή. Έτσι λύνονται τα χέρια μας σε περίπτωση που ένας αλγόριθμος δεν ικανοποιεί τις απαιτήσεις των χρηστών αφού μπορούμε κάλλιστα να των αλλάξουμε.

Πολύ σημαντικό παράγοντα σε τέτοιου είδους εφαρμογές παίζει και ο χρόνος εκτέλεσης ενός αλγορίθμου για να βγάλει αποτελέσματα. Τέτοιες εφαρμογές έχουν τεράστιο όγκο δεδομένων και αξιοποιούν μεγάλη ποσότητα συγκεκριμένου. Πρέπει κατά την υλοποίηση τέτοιων εφαρμογών να γίνει σωστή μελέτη έτσι ώστε να επιλεγεί ο κατάλληλος αλγόριθμος που να δίνει το ζητούμενο μέσα σε λογικά όρια χρόνου. Επίσης είναι λογικό πως όσο περισσότερο συγκεκριμένο δίνεται σε ένα αλγόριθμο για να το επεξεργαστεί τόσο πιο αργός θα είναι. Άρα θα πρέπει οι δημιουργοί τέτοιων εφαρμογών να είναι προσεκτικοί με το συγκεκριμένο που περνούν στους αλγορίθμους έτσι ώστε να μην δίνουν περιττές πληροφορίες συγκεκριμένου που θα καθυστερούν χωρίς λόγο τη παραγωγή των αποτελεσμάτων.

### 6.3 Μελλοντική Εργασία

Ο πρωταρχικός στόχος αυτής της διπλωματικής εργασίας ήταν η ανάπτυξη μιας διαδικτυακής εφαρμογής movie recommender. Παρόλα αυτά, αρκετές μελλοντικές εργασίες για αναβάθμιση της θα μπορούσαν να πραγματοποιηθούν.

Καταρχάς όσον αφορά λειτουργίες που θα μπορούσαν να προστεθούν στη διαδικτυακή εφαρμογή είναι πολλές. Θα μπορούσε κατά τη γνώμη μου αυτή η εφαρμογή να εξελιχθεί σαν ένα κοινωνικό δίκτυο. Ένα κοινωνικό δίκτυο που να μπορεί ένας χρήστης να σχηματίζει ένα profile με τα χαρακτηριστικά του και να μπορεί να αναζητήσει φίλους μέσω του δικτύου αυτού.

Μέσω αυτού του κοινωνικού δικτύου να μπορεί να μοιράζεται τις βαθμολογίες του με τους φίλους του. Δηλαδή να προτείνει ή και να αποθαρρύνει ο ίδιος στους φίλους του ταινίες για να παρακολουθήσουν ή να μην παρακολουθήσουν αντίστοιχα. Επίσης να έχει ο κάθε χρήστης λίστα με τις αγαπημένες του ταινίες που να μπορούν να τις δουν οι φίλοι του. Ακόμα μια ενδιαφέρουσα λειτουργία που θα μπορούσε να υλοποιηθεί είναι να ενημερώνει το χρήστη που δείχνει ενδιαφέρον σε μια ταινία να τη παρακολουθήσει, ποιοι άλλοι φίλοι του έχουν δείξει και αυτοί ενδιαφέρον στην ίδια ταινία για να μπορούν να οργανώνουν συναντήσεις και να παρακολουθούν ταινίες ομαδικώς.

Γενικά η υλοποίηση ενός τέτοιου κοινωνικού δικτύου θα είναι η αρχή για υλοποίηση πολλών ακόμα λειτουργιών.

Όσο αφορά την υποδομή αυτής της ιστοσελίδας για το λόγο ότι τα δεδομένα της θα αυξάνονται ολοένα και περισσότερο θα ήταν καλά να αγοραστεί εξωτερικός server με υψηλές υπολογιστικές ικανότητες για να εκτελούνται ακόμα πιο γρήγορα οι αλγόριθμοι. Μια εναλλακτική επιλογή θα ήταν και το νέφος (cloud) αντί ένας εξυπηρετητής μιας και στην εποχή μας όλα τα δεδομένα τείνουν να αποθηκεύονται στο cloud.

## Βιβλιογραφία

- [1] Incorporating contextual information in recommender systems using a multidimensional approach, Available from:  
<http://dl.acm.org/citation.cfm?id=1055714> [Date Accessed: May, 2017]
- [2] Context-Aware Recommender Systems, Available from:  
<http://ids.csom.umn.edu/faculty/gedas/nsfcareer/CARS-chapter-2010.pdf>  
[Date Accessed: May, 2017]
- [3] Evaluating collaborative filtering recommender systems, Available from:  
<http://dl.acm.org/citation.cfm?id=963772> [Date Accessed: May, 2017]
- [4] Hybrid Recommender Systems: Survey and Experiments, Available from:  
<https://pdfs.semanticscholar.org/5880/b9bc3f75f4649b8ec819c3f983a14fca9927.pdf> [Date Accessed: May, 2017]
- [5] Matrix Factorization Techniques for Context Aware Recommendation, Available from:  
<https://pdfs.semanticscholar.org/140a/c1b3c48e97bdf5a566ae6ecee42961f0c8.pdf> [Date Accessed: May, 2017]
- [6] A User's Guide to CARSKit, Available from:  
<https://arxiv.org/pdf/1511.03780.pdf> [Date Accessed: May, 2017]
- [7] Wikipedia, Available from:  
[https://en.wikipedia.org/wiki/Web\\_application](https://en.wikipedia.org/wiki/Web_application) [Date Accessed: May, 2017]
- [8] Wikipedia, Available from:  
<https://en.wikipedia.org/wiki/HTML5> [Date Accessed: May, 2017]

- [9] Wikipedia, Available from:  
[https://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets#CSS\\_3](https://en.wikipedia.org/wiki/Cascading_Style_Sheets#CSS_3)  
[Date Accessed: May, 2017]
- [10] Wikipedia, Available from:  
<https://en.wikipedia.org/wiki/JavaScript> [Date Accessed: May, 2017]
- [11] Wikipedia, Available from:  
<https://en.wikipedia.org/wiki/JQuery> [Date Accessed: May, 2017]
- [12] Wikipedia, Available from:  
[https://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))  
[Date Accessed: May, 2017]
- [13] Wikipedia, Available from:  
<https://en.wikipedia.org/wiki/PHP> [Date Accessed: May, 2017]
- [14] Wikipedia, Available from:  
<https://en.wikipedia.org/wiki/MySQL> [Date Accessed: May, 2017]
- [15] Ανάλυση Απαιτήσεων και Καθορισμός Προδιαγραφών, Available from:  
<http://apsisd.wikispaces.com/file/view/kefalaio3.pdf>  
[Date Accessed: May, 2017]
- [16] Ιστοσελίδα IMDB, Available from:  
<http://www.imdb.com>  
[Date Accessed: May, 2017]
- [17] Ιστοσελίδα Netflix, Available from: <https://www.netflix.com/cy/>  
[Date Accessed: May, 2017]

- [18] Ιστοσελίδα Amazon, Available from: <https://www.amazon.com>  
[Date Accessed: May, 2017]
- [19] Ιστοσελίδα E-bay, Available from: <http://www.ebay.com/>  
[Date Accessed: May, 2017]

## Παράρτημα Α

Σε αυτό το παράρτημα θα παρουσιαστούν κομμάτια από το κώδικα της διπλωματικής εργασίας.

Παραδείγματα κώδικα που γράφτηκε για εισαγωγή των δεδομένων στη βάση δεδομένων:

```
|<?php
$GLOBALS["link"]="localhost";
$GLOBALS["DB"]="root";
$GLOBALS["DBpass"]="";
$GLOBALS["DBName"]="moviesade";

$GLOBALS["dbh"]= mysqli_connect($GLOBALS["link"],$GLOBALS["DB"],$GLOBALS["DBpass"],$GLOBALS["DBName"])
or die("Couldn't connect to database.");

$myfile = fopen("ml-100k/u.user", "r") or die("Unable to open file!");
// Output one line until end-of-file
while(!feof($myfile)) {
    $line = fgets($myfile);
    $user = explode('|',$line);
    $sql = "INSERT INTO user ( `UserID`, `Age`, `Gender`, `Occupation`, `ZipCode` )
        VALUES
        ( ".$user[0].", ".$user[1].", ' ".$user[2]."', ' ".$user[3]."', ' ".$user[4]."' )";
    mysqli_query($GLOBALS["dbh"],$sql);
}
fclose($myfile);
echo('finished');

?>
```

Σχήμα Α1 – Insert users to database using php



```

<?php
$GLOBALS["link"]="localhost";
$GLOBALS["DB"]="root";
$GLOBALS["DBpass"]="";
$GLOBALS["DBName"]="moviesade";

$GLOBALS["dbh"]= mysqli_connect($GLOBALS["link"],$GLOBALS["DB"],$GLOBALS["DBpass"],$GLOBALS["DBName"])
or die("Couldn't connect to database.");

$myfile = fopen("ml-100k/u.item", "r") or die("Unable to open file!");
// Output one line until end-of-file
while(!feof($myfile)) {
    $line = fgets($myfile);
    $user = explode('|',$line);
    $sql = "INSERT INTO item ( `MovieID`, `MovieTitle`, `ReleaseDate`,`VideoReleaseDate`,
        `ImdbURL`, `unknown`, `Action`, `Adventure`, `Animation`,
        `Children's`, `Comedy`, `Crime`, `Documentary`, `Drama`,
        `Fantasy`, `Film-Noir`, `Horror`, `Musical`, `Mystery`,
        `Romance`, `Sci-Fi`, `Thriller`, `War`, `Western` )
        VALUES
        (". $user[0].",\". $user[1].\", \".\". $user[2].\", \".\". $user[3].\", \".\". $user[4].\", \".\". $user[5]
        .\", \".\". $user[6].\", \".\". $user[7].\", \".\". $user[8].\", \".\". $user[9].\", \".\". $user[10].\", \".\". $user[11].\", \".\". $user[12].\", \".\". $user[13].\", \".\". $user[14]
        .\", \".\". $user[15].\", \".\". $user[16].\", \".\". $user[17].\", \".\". $user[18].\", \".\". $user[19].\", \".\". $user[20].\", \".\". $user[21].\", \".\". $user[22].\", \".\". $user[23]
        .\")";
    //print_r($sql);exit;
    mysqli_query($GLOBALS["dbh"],$sql);
}
fclose($myfile);
echo('finished');
?>

```

Σχήμα Α2 – Insert movies to database using php

Κώδικας που γράφτηκε για κατέβασμα των εικόνων των ταινιών από εξωτερικό σύνδεσμο:

```

function jsonParse(result){
    obj = JSON.parse(result);
    for(var i in obj){
        $.get("../test.php?imdbURL="+obj[i].ImdbURL+"&Title="+obj[i].MovieTitle, function(response) {
            document.getElementById("btn").innerHTML = response;
        });
    }
}

function getMovieData(){
    $.ajax({url: "../GenerateArrays/getImages.php", success: function(result){
        jsonParse(result);
    }});
}

```

Σχήμα Α3 – JavaScript functions for downloading movie images

```

<?php
include '../WebApp/Content/dbConnect.php';
$sql = 'SELECT MovieTitle, ImdbURL From item';
$result = mysqli_query($GLOBALS["dbh"], $sql);

$rows = array();
while($r = mysqli_fetch_assoc($result)) {
    $r["MovieTitle"] = rawurlencode($r["MovieTitle"]);
    array_push($rows, $r);
}

print json_encode($rows);

?>

```

Σχήμα A4 – Backend functionality to get all movies hyperlinks

```

<?php

include "simple_html_dom.php";

$imdbURL = $_GET["imdbURL"];
$title = $_GET["Title"];

$imdbURL = str_replace(" ", "%20", $imdbURL);
$content = file_get_html($imdbURL);

$i=0;
$s= "";
foreach($content->find('img') as $element){
    if($i==3){
        //make directory
        mkdir("C:/wamp64/www/ade/MovieImages/" . $title);

        $content = file_get_contents($element->src);

        //Store in the filesystem.
        $fp = fopen("C:/wamp64/www/ade/MovieImages/" . $title . "/MovieImage.jpg", "wb");
        fwrite($fp, $content);
        fclose($fp);
        echo 'Image for movie ' . $title . ' saved.';
        break;
    }
    $i++;
}

?>

```

Σχήμα A 5 – Save Images locally using PHP

Κώδικας που γράφτηκε για το χτίσιμο του Home Page με χρήση PHP components:

```

<body onload="setWatchlistedMovies(<?php echo $_SESSION["UserID"] ?>);onLoadModal(<?php echo $_SESSION["UserID"] ?>);" >
  <!-- Navigation -->
  <?php
  |   include 'Content/loginModal.php';
  ?>

  <!-- Navigation -->
  <?php
  |   include 'Content/navigation.php';
  ?>

  <!-- Header -->
  <?php
  |   include 'Content/header.php';
  ?>

  <!-- Recommendations -->
  <?php
  |   include 'Content/recommendations.php';
  ?>

  <!-- Recommendations By Age -->
  <?php
  |   include 'Content/recsAge.php';
  ?>

  <!-- Recommendations By Job -->
  <?php
  |   include 'Content/recsJob.php';
  ?>

  <!-- Modal -->
  <?php
  |   include 'Content/modal.php';
  ?>

  <!-- Image For Modal -->
  <?php
  |   include 'Content/imageForModal.php';
  ?>

  <!-- Footer -->
  <?php
  |   include 'Content/footer.php';
  ?>

```

**Σχήμα A6 – Home Page using included PHP components**

Πιο κάτω ακολουθεί ο κώδικας που παρουσιάζει δυναμικά τις συστάσεις με τη χρήση της PHP.

```
1 <?php
2 include 'dbConnect.php';
3 $sql = 'SELECT Rec1,Rec2,Rec3,Rec4,Rec5,Rec6,Rec7,Rec8 From recommendation where UserID = '.$_SESSION["UserID"];
4 $result = mysqli_query($GLOBALS["dbh"],$sql);
5 while ($row = mysqli_fetch_array($result)) {
6     if(is_null($row["Rec1"])){
7         echo '<aside class="callout">
8             <div id="recommendations" class="text-vertical-center">
9                 <h1>Not Available At The Moment</h1>
10                <h2>You have to rate some movies first</h2>
11            </div>
12        </aside>';
13        return;
14    }
15 }
16 $result = mysqli_query($GLOBALS["dbh"],$sql);
17 echo '
18 <!-- Recommendations -->
19 <div id="recommendations" class="carousel slide" data-ride="carousel" style="background: linear-gradient(to right,
20 #2b0202,#4e0806,#df2a1e,#ffb66b,white,white,white,white,white,white,#ffb66b,#df2a1e,#4e0806,#2b0202);">
21 <!-- Indicators -->
22 <ol class="carousel-indicators">';
23
24 $i=0;
25 while ($i<8) {
26     if ($i==0)
27         echo '<li data-target="#recommendations" data-slide-to="'.$i.'" class="active"></li>';
28     else
29         echo '<li data-target="#recommendations" data-slide-to="'.$i.'"></li>';
30     $i++;
31 }
32 echo '</ol>';
33
34 echo '
35 <div class="carousel-inner" role="listbox">';
36
37 $i=0;
38 while ($row = mysqli_fetch_array($result)) {
39     for($i=0;$i<8;$i++){
40         $sql = 'SELECT * From item I Where I.MovieID = '.$row[$i];
41         $result2 = mysqli_query($GLOBALS["dbh"],$sql);
42         while ($row2 = mysqli_fetch_array($result2)) {
43             if($i==0){
44                 echo '
45                     <div class="item active">
```

```

45         <ul id="skill">
46             <li><span style="width:100%" class="bar jquery"></span><center><h3>Recommendation 1</h3></center></li>
47         </ul>
48         <center>
49         <div class="" data-toggle="buttons">
50             <label class="btn btn-lg btn-block btn-success text-success active" style="width:15%">
51                 <input type="radio" name="options" id="option1" onchange="watchlistButton(0,'.$row2["MovieID"]','.$_SESSION["UserID"].')'" autocomplete="off" checked>
52                 <i class="fa fa-check-circle-o fa-3x animated fadeIn"></i>
53                 <br />
54                 Watchlisted
55             </label>
56             <label class="btn btn-lg btn-block btn-default text-default" style="width:15%">
57                 <input type="radio" name="options" id="option2" onchange="watchlistButton(1,'.$row2["MovieID"]','.$_SESSION["UserID"].')'" autocomplete="off">
58                 <i class="fa fa-circle-o fa-3x"></i>
59                 <br />
60                 Add To Watchlist
61             </label>
62         </div>
63         </center><br>
64         
65         <div class="carousel-caption">
66             <h1>'.$row2["MovieTitle"].'</h1>
67             <p>Release Date: '.$row2["ReleaseDate"].'</p>
68             <div class="col-lg-12 text-center">
69                 <a target="_blank" href=".$row2["ImdbURL"].'" class="btn btn-lg btn-dark">See More</a>
70             </div>
71         </div>
72         </div>';
73     </div>';
74
75     }
76     else{
77         echo '
78         <div class="item">
79             <ul id="skill">
80                 <li><span style="width:'.$((100*(8-$i))/8).'%" class="bar jquery"></span><center><h3>Recommendation '.$(i+1).'</h3></center></li>
81             </ul>
82             <center>
83                 <div class="" data-toggle="buttons">
84                     <label class="btn btn-lg btn-block btn-success text-success active" style="width:15%">
85                         <input type="radio" name="options" id="option1" onchange="watchlistButton(0,'.$row2["MovieID"]','.$_SESSION["UserID"].')'" autocomplete="off" checked>
86                         <i class="fa fa-check-circle-o fa-3x animated fadeIn"></i>
87                         <br />
88                         Watchlisted
89                     </label>
90                     <label class="btn btn-lg btn-block btn-default text-default" style="width:15%">
91                         <input type="radio" name="options" id="option2" onchange="watchlistButton(1,'.$row2["MovieID"]','.$_SESSION["UserID"].')'" autocomplete="off">
92                         <i class="fa fa-circle-o fa-3x"></i>
93                         <br />
94                         Add To Watchlist
95                     </label>
96                 </div>
97             </center><br>
98             
99             <div class="carousel-caption">
100                 <h1>'.$row2["MovieTitle"].'</h1>
101                 <p>Release Date: '.$row2["ReleaseDate"].'</p>
102                 <div class="col-lg-12 text-center">
103                     <a target="_blank" href=".$row2["ImdbURL"].'" class="btn btn-lg btn-dark">See More</a>
104                 </div>
105             </div>
106             </div>';
107         </div>';
108     }
109 }
110
111 }
112
113
114 echo '</div>';
115
116 echo '
117 <a class="left carousel-control" href="#recommendations" role="button" data-slide="prev">
118     <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
119     <span class="sr-only">Previous</span>
120 </a>
121 <a class="right carousel-control" href="#recommendations" role="button" data-slide="next">
122     <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
123     <span class="sr-only">Next</span>
124 </a>
125 </div>';

```

Σχήμα A7 – Recommendations Component

Ποιο κάτω ακολουθεί ο κώδικας που γίνονται εισάγονται στη βάση δεδομένων βαθμολογίες.

```
1
2 ▼ function onFormSubmit(userID){
3     var selectPicker = document.getElementsByClassName("selectpicker")[0].value;
4
5 ▼     if(selectPicker == ""){
6         alert("You have to select a movie first!");
7         return;
8     }
9 ▼     else{
10        var rate = document.getElementById("rating").value;
11        var movieID = document.getElementsByClassName("selectpicker")[0].selectedOptions[0].attributes[0].value.split("_")[1]
12        var res = window.confirm("We will rate the movie " + selectPicker + " with " + rate + ". Are you sure?");
13 ▼        if(res){
14            //close the modal
15            $("#myModal").modal("toggle");
16            $('.modal-backdrop').remove();
17
18            $.post("Content/addRating.php",
19 ▼            {
20                MovieID: movieID,
21                Rate: rate,
22                UserID: userID
23
24            },function(result){
25                location.reload();
26            });
27
28        }
29        else
30            return;
31    }
32 }
33
34
```

Σχήμα A8 – Post request using JavaScript and JQuery

```
1 |<?php
2 include 'dbConnect.php';
3
4 $MovieID = $_POST["MovieID"];
5 $Rate = $_POST["Rate"];
6 $UserID = $_POST["UserID"];
7
8
9 $sql = 'insert INTO data
10        (UserID, ItemID, Rating, data.Timestamp)
11        VALUES('.$UserID.','.$MovieID.','.$Rate.',UNIX_TIMESTAMP(CURRENT_TIMESTAMP))';
12
13 mysqli_query($GLOBALS["dbh"],$sql);
14
15 ?>
```

Σχήμα A9 – Insert ratings to database using PHP

Κώδικας της υλοποίησης του crone job:

```
1 |<?php
2
3 |include 'C:\wamp64\www\ade\GenerateArrays\generateRatings.php';
4 |include 'C:\wamp64\www\ade\GenerateArrays\clearFile.php';
5 ▼|include 'C:\wamp64\www\ade\GenerateArrays\updateRecommendations.php';
6
7 |?>
```

**Σχήμα A10 – PHP Functionalities to executed**

```
1 |"C:\wamp64\bin\php\php5.6.19\php.exe" -f "C:\wamp64\www\ade\GenerateArrays\CroneJob\crone_job.php"
```

**Σχήμα A11 – Bat file**

```
1 |Set WinScriptHost = CreateObject("WScript.Shell")
2 |WinScriptHost.Run Chr(34) & "C:\wamp64\www\ade\GenerateArrays\CroneJob\script.bat" & Chr(34), 0
3 |Set WinScriptHost = Nothing
```

**Σχήμα A12 – VBS File**

```

1 <?php
2 $GLOBALS["link"]="localhost";
3 $GLOBALS["DB"]="root";
4 $GLOBALS["DBpass"]="";
5 $GLOBALS["DBName"]="moviesade";
6
7 $GLOBALS["dbh"]= mysqli_connect($GLOBALS["link"],$GLOBALS["DB"],$GLOBALS["DBpass"],$GLOBALS["DBName"])
8 or die("couldn't connect to database.");
9
10 $myfile = fopen("C:/wamp64/www/ade/CARSKit-master/context-aware_data_sets/MyMovies/CarsKit.Workspace/ratings_binary.txt", "w")
11 or die("Unable to open file!");
12 $txt = "UserID,ItemID,Rating,Day:Weekday,Day:Weekend,AgeRange:<20,AgeRange:20-30,AgeRange:30-40,AgeRange:40-50,AgeRange:50-60,AgeRange:60-70,AgeRange:
13 :>70,Type:unknown,Type:Action,Type:Adventure,Type:Animation,Type:Children's,Type:Comedy,Type:Crime,Type:Documentar
14 y,Type:Drama,Type:Fantasy,Type:Film-Noir,Type:Horror,Type:Musical,Type:Mystery,Type:Romance,Type:Sci-Fi,Type:Thriller,Type:War,Type:Western\n";
15 fwrite($myfile, $txt);
16
17 $sql = "SELECT D.UserID, D.ItemID, D.Rating,
18 CASE WHEN (WEEKDAY(from_unixtime(D.Timestamp, '%Y-%m-%d')) BETWEEN '0' AND '4' THEN 1 ELSE 0 END AS 'Weekday',
19 CASE WHEN (WEEKDAY(from_unixtime(D.Timestamp, '%Y-%m-%d')) BETWEEN '5' AND '6' THEN 1 ELSE 0 END AS 'Weekend',
20 CASE WHEN U.Age BETWEEN '1' AND '19' THEN 1 ELSE 0 END AS '<20',
21 CASE WHEN U.Age BETWEEN '20' AND '30' THEN 1 ELSE 0 END AS '20-30',
22 CASE WHEN U.Age BETWEEN '31' AND '40' THEN 1 ELSE 0 END AS '30-40',
23 CASE WHEN U.Age BETWEEN '41' AND '50' THEN 1 ELSE 0 END AS '40-50',
24 CASE WHEN U.Age BETWEEN '51' AND '60' THEN 1 ELSE 0 END AS '50-60',
25 CASE WHEN U.Age BETWEEN '61' AND '70' THEN 1 ELSE 0 END AS '60-70',
26 CASE WHEN U.Age BETWEEN '71' AND '120' THEN 1 ELSE 0 END AS '>70',
27 I.unknown, I.Action, I.Adventure, I.Animation, I.'Children's', I.Comedy, I.Crime, I.Documentary, I.Drama, I.Fantasy, I.'Film-Noir', I.Horror
, I.Musical, I.Mystery, I.Romance, I.'Sci-Fi', I.Thriller, I.War, I.Western
28
29 FROM `item` I,`data` D, `user` U
30 WHERE I.`MovieID` = D.ItemID AND D.UserID = U.UserID
31 ORDER BY D.UserID, D.ItemID";
32
33 $results = mysqli_query($GLOBALS["dbh"],$sql);
34 ini_set('memory_limit', '256M');
35 while ($row = mysqli_fetch_array($results)) {
36
37     $s = $row["UserID"].",".$row["ItemID"].",".$row["Rating"].",".$row["Weekday"].",".$row["Weekend"].",".$row["<20"].",".
38     $row["20-30"].",".$row["30-40"].",".$row["40-50"].",".$row["50-60"].",".$row["60-70"].",".$row[">70"].",".
39     $row["unknown"].",".$row["Action"].",".$row["Adventure"].",".$row["Animation"].",".$row["Children's"].",".
40     $row["Comedy"].",".$row["Crime"].",".$row["Documentary"].",".$row["Drama"].",".$row["Fantasy"].",".
41     $row["Film-Noir"].",".$row["Horror"].",".$row["Musical"].",".$row["Mystery"].",".$row["Romance"].",".
42     $row["Sci-Fi"].",".$row["Thriller"].",".$row["War"].",".$row["Western"]."\n";
43
44     fwrite($myfile, $s);
45 }
46
47 fclose($myfile);
48 chdir('C:/wamp64/www/ade/CARSKit-master');
49 exec("java -jar CARSKit-v0.3.0.jar");
50 ?>

```

Σχήμα A13 – Generate Recommendations Using PHP