

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Πληροφορικής



Εργασία Μαθήματος

«Τεχνολογία Λογισμικού (6^ο εξ.)»

Ομάδα 8	
Όνομα φοιτητή – Αρ. Μητρώου	Μίμογλου Γιώργος – Π17073 Παπαθεοχάρους Ιωάννης – Π17002 Παπαξενίδης Κωνσταντίνος – Π17100
Ημερομηνία παράδοσης	02/07/2020



Περιεχόμενα

1.	Εισαγωγή	3
1.1.	Στόχοι της εργασίας.....	3
1.2.	Υλοποίηση της εργασίας	3
1.3.	Προβλήματα – Αντιμετώπιση.....	3
1.4.	Εκτέλεση σε Visual Studio	3
1.5.	Demo	6
2.	Σύντομη παρουσίαση της RUP	7
3.	Φάση: Έναρξη (Inception)	8
3.1.	Σύλληψη απαιτήσεων	8
3.2.	Ανάλυση και Σχεδιασμός.....	9
4.	Φάση: Εκπόνηση Μελέτης (Elaboration)	12
4.1.	Ανάλυση και σχεδιασμός	12
4.2.	Υλοποίηση και έλεγχος.....	28
5.	Φάση: Κατασκευή (Construction)	31
5.1.	Ανάλυση και Σχεδιασμός.....	31
5.2.	Υλοποίηση και Έλεγχος.....	53
6.	Εγχειρίδιο χρήστη.....	56
6.1.	Σύντομη παρουσίαση του προγράμματος.....	56
6.2.	Παρουσίαση σεναρίων λειτουργίας	57



1. Εισαγωγή

Όπως η κατασκευή μιας μεγάλης γέφυρας απαιτεί σχεδιασμό, οργάνωση, επιτήρηση και έλεγχο, έτσι και τα μεγάλα έργα στην Πληροφορική έχουν τις ίδιες ανάγκες. Χρόνια μετα τα χρόνια, τα έργα αυτά γίνονται όλο και πιο πολύπλοκα, αφού οι απαιτήσεις της αγοράς και των τεχνολογιών είναι αυστηρές. Πόσο μάλλον, σε ένα σκληρό ανταγωνιστικό και μεταβαλλόμενο περιβάλλον, που το προϊόν τεχνολογίας θα πρέπει να παραδίδεται όσο το δυνατόν γρηγορότερα, με ελάχιστα σφάλματα και ικανά χαρακτηριστικά. Για να επιτευχθούν οι παραπάνω στόχοι, οι ομάδες πληροφορικής πρέπει να ακολουθούν μια πεπατημένη και να οδηγούνται από μεθοδολογίες. Μια από αυτές, αποτελεί η Rational Unified Process ή RUP. Πως όμως μια άπειρη ομάδα καταφέρνει σε περίοδο πανδημίας να εκπληρώσει τον στόχο της, την σχεδίαση και την υλοποίηση μιας εφαρμογής;

1.1. Στόχοι της εργασίας

Η εργασία που μας ανατέθηκε αποτελείται από τη σχεδίαση και τη δημιουργία μιας ιστοσελίδας κοινοποίησης και αξιολόγησης ταινιών. Ο κύκλος ζωής του λογισμικού με τον οποίο θα πρέπει να υλοποιηθεί το λογισμικό είναι η RUP. Κατά τη διάρκεια κάθε φάσης, πρέπει να γίνονται αντίστοιχα διαγράμματα με τη γλώσσα μοντελοποίησης Unified Modelling Language ή UML.

1.2. Υλοποίηση της εργασίας

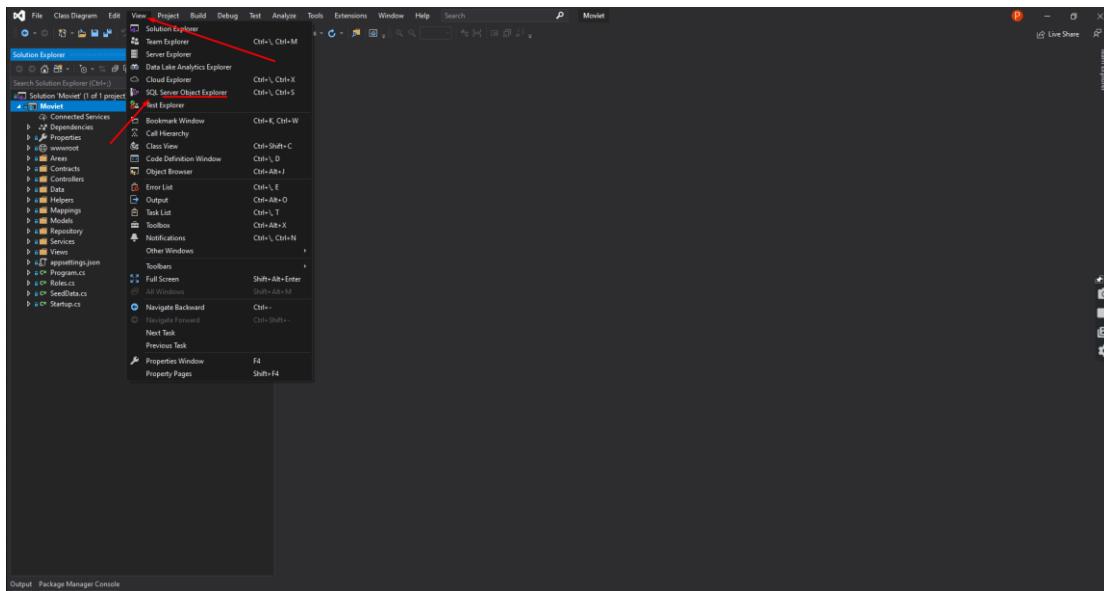
Η εργασία υλοποιήθηκε σε ASP.NET Core MVC και χρησιμοποιήθηκαν οι τεχνολογίες MSSQL Server για τη βάση δεδομένων, Entity Framework Core για ORM και Identity Core API για user management. Προσπαθήσαμε υλοποιήσουμε όσο περισσότερο γίνεται έννοιες, αρχές και μοτίβα του αντικειμενοστρεφή προγραμματισμού όπως SOLID, Dependency Injection, Repository Pattern.

1.3. Προβλήματα – Αντιμετώπιση

Αυτή η εργασία ήταν μια αφορμή ώστε να ασχοληθούμε με τεχνολογίες που χρησιμοποιούνται στην αγορά, αλλά ακόμα πιο σημαντικό, να αναπτυχθούν τα soft skills μας, όπως συνεργασία, επικοινωνία και διαρκής εκπαίδευση. Η έλλειψη γνώσης αποτέλεσε πρόβλημα, με την πανδημία να το διογκώνει. Έτσι, η εργασία, μελέτη και συνεργασία εξ 'αποστάσεως ήταν μονόδρομος. Αρπάξαμε την ευκαιρία και με επιμονή ξεπεράσαμε τα προβλήματα αυτά. Μετά την μελέτη της UML, ακολούθησαν μαθήματα μεταξύ της ομάδας που αφορούν principles και patterns όπως MVC, SOLID και Dependency Injection. Στη συνέχεια, έγιναν μαθήματα πάνω στο ASP.NET Core MVC και τέλος πάνω στο Version Control System git και την πλατφόρμα GitHub. Τα μαθήματα έγιναν υπό την καθοδήγηση του Μίμογλου Γ..

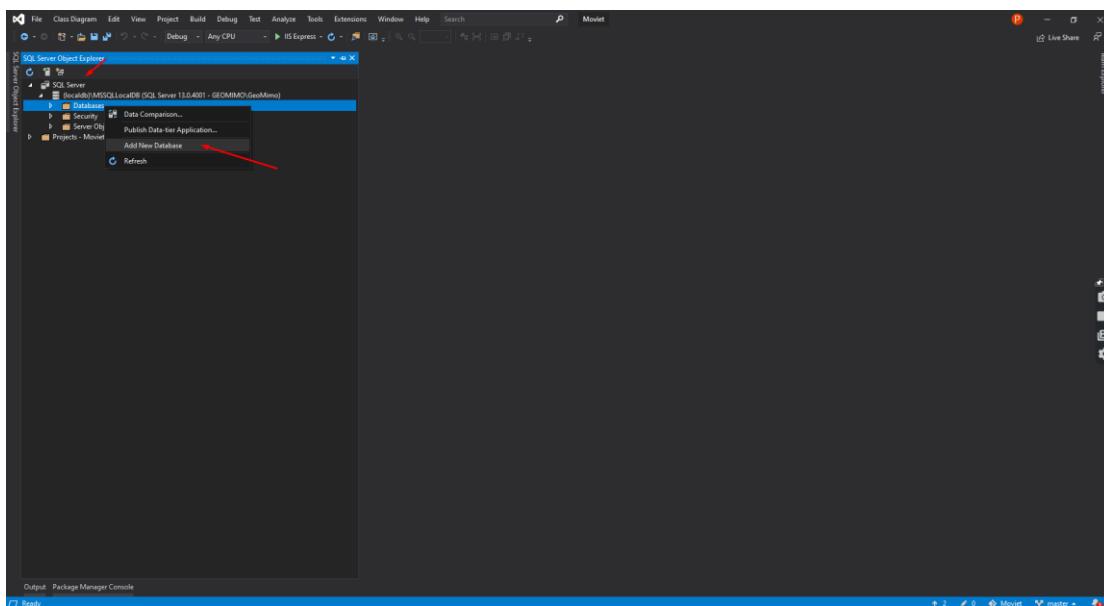
1.4. Εκτέλεση σε Visual Studio

Ανοίξτε το Visual Studio και φορτώστε το solution "Moviet.sln". Πατήστε View -> Sql Server Object Explorer.



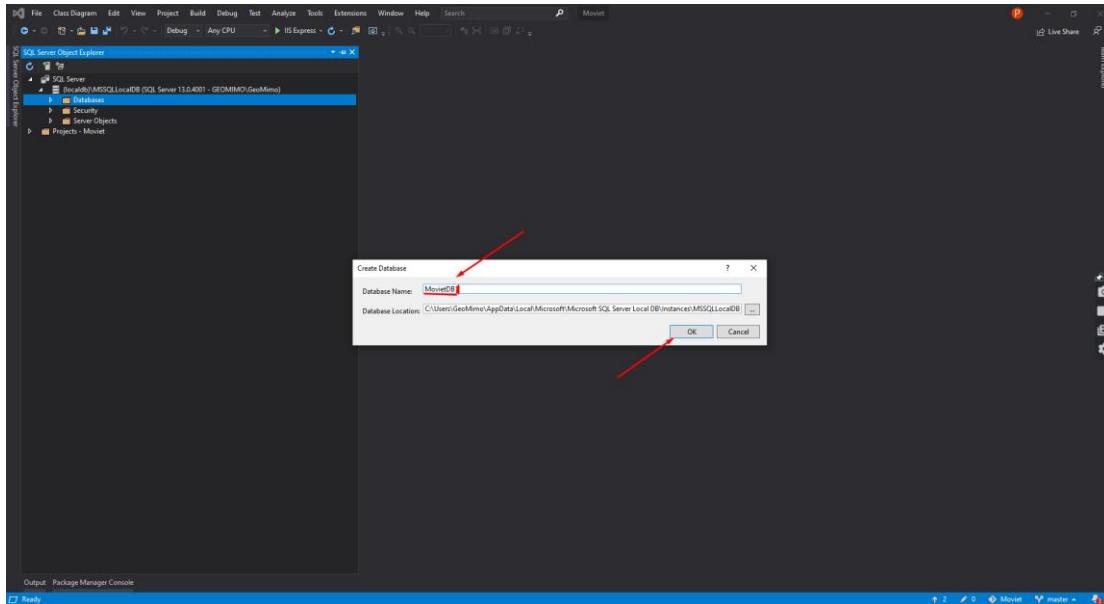
Εικόνα 1 Database (1)

Στον explorer που εμφανίζεται, ανοίγετε SQL Server και στη συνέχεια το (localdb)\MSSQLLocalDB. Στον φάκελο Databases, πατήστε δεξί κλικ και μετά “Add New Database”.



Εικόνα 2 Database (2)

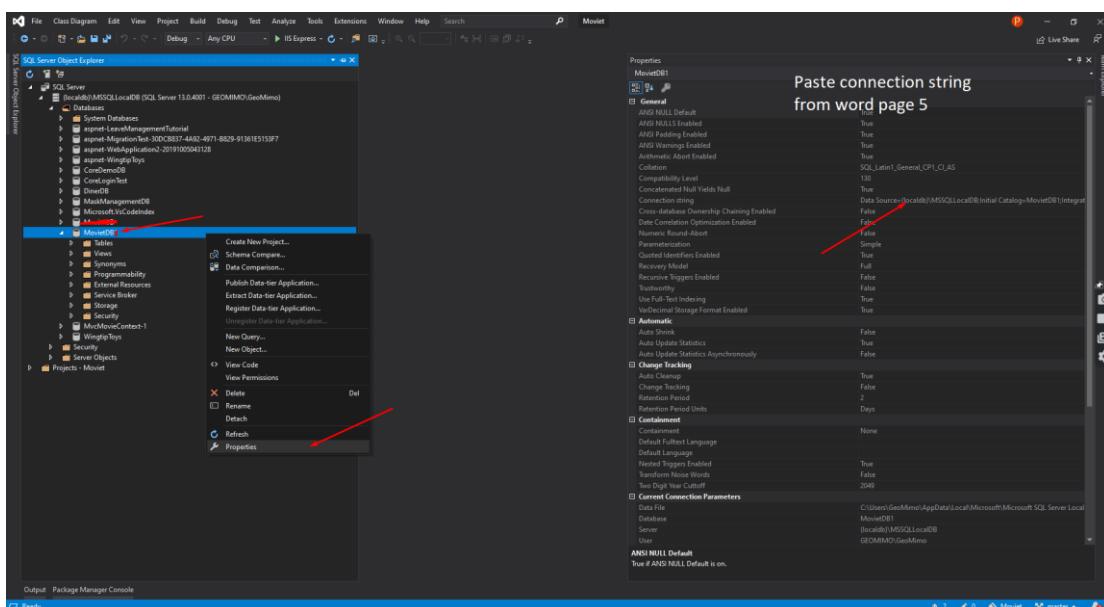
Εισάγετε το όνομα “MovietDB” στο πεδίο Database Name και πατήστε OK.



Εικόνα 3 Database (3)

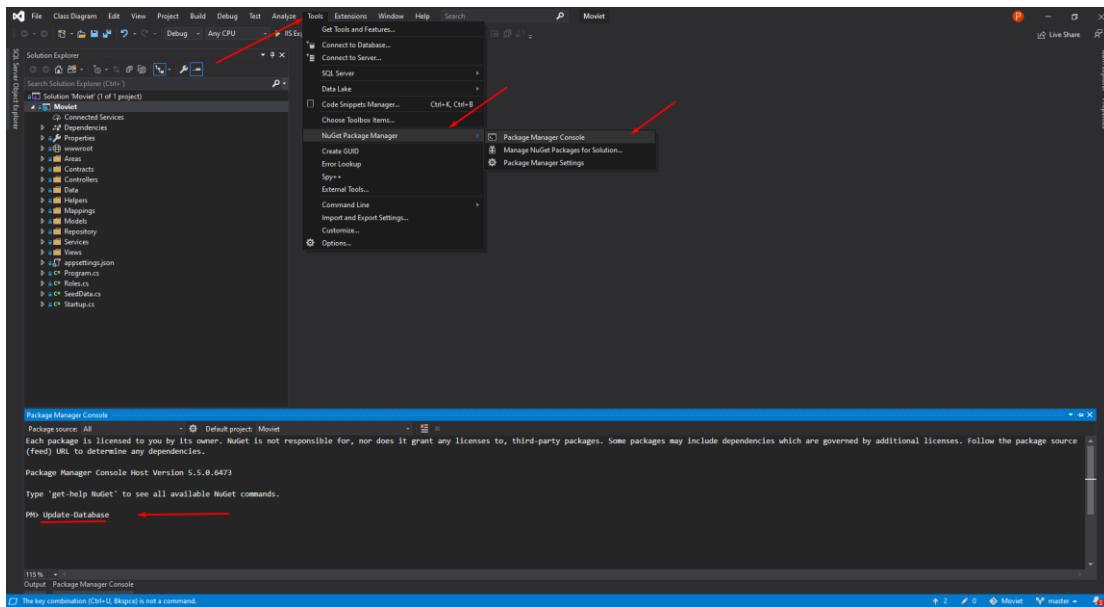
Στον φάκελο Databases θα δημιουργηθεί μια νέα βάση με το όνομα “MovietDB”. Ανοίξτε τη και στη συνέχεια δεξί κλικ και επιλέξτε Properties. Βρείτε το πεδίο Connection String και κάντε paste το παρακάτω:

```
Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=MovietDB;Integrated
Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
ubnetFailover=False
```



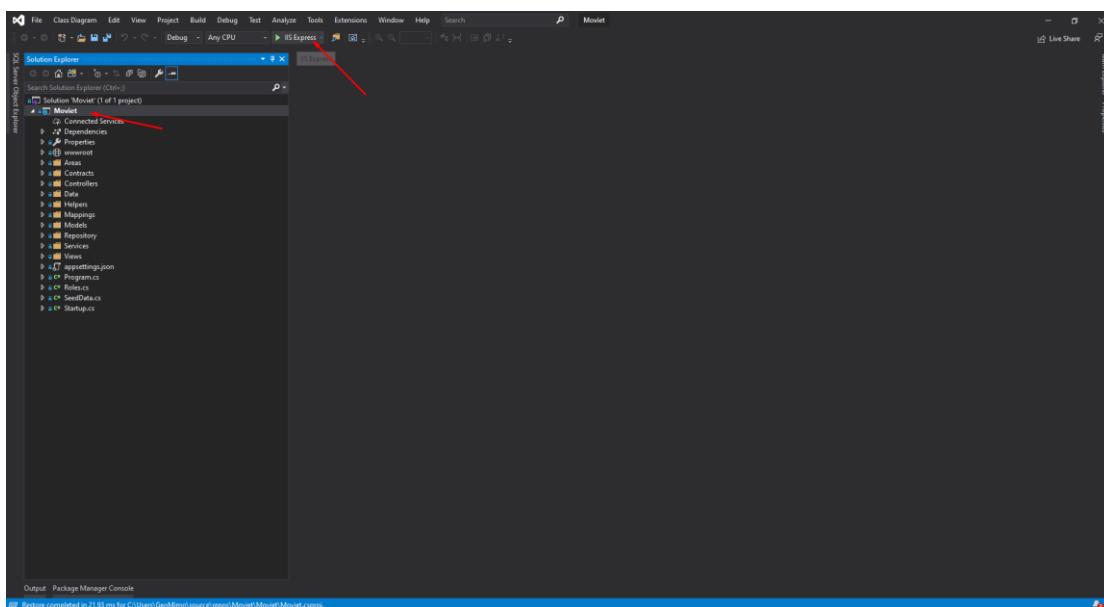
Εικόνα 4 Database (4)

Στη συνέχεια, πατήστε Tools -> NuGet Package Manager -> Package Manager Console. Στο παράθυρο που ανοίγει, γράψτε την εντολή *Update-Database* και πατήστε Enter.



Εικόνα 5 Database (5)

Στο Solution Explorer, κλικάρετε το project και στη συνέχεια πατήστε το κουμπί ISS Express για να τρέξει η εφαρμογή.



Εικόνα 6 Database (6)

1.5. Demo

Μπορείτε να παρακολουθήσετε ένα demo της εφαρμογής [εδώ](#). Παρουσιάζονται οι λειτουργίες της εφαρμογής, καθώς επίσης και η δομή του προγράμματος από τεχνική σκοπιά.

https://drive.google.com/file/d/151XYTxNtoTcUDuzCPkpi5Al0_c-GuSwu/view?usp=sharing



2. Σύντομη παρουσίαση της RUP

Η Rational Unified Process αποτελεί μια επαναληπτική διαδικασία ανάπτυξης λογισμικού, που προτάθηκε από την IBM. Χωρίζει την ανάπτυξη σε τέσσερεις φάσεις, που προσαρμόζονται κάθε φορά ανάλογα τις απαιτήσεις και το χρόνο. Οι φάσεις είναι:

- **Έναρξη (Inception):** Ορίζεται ο κύριος στόχος και το domain. Γίνεται πρόβλεψη του κόστους και του χρόνου της ανάπτυξης και αναλύονται τα ρίσκα. Συλλέγονται οι βασικοί πόροι, πολύ γενικές απαιτήσεις αλλά και οι stakeholders. Αποτελεί το πρώτο στάδιο, στο οποίο η ιδέα «γεννιέται», συζητείται και αναλύεται από την όψη του business plan. Αν η φάση αυτή δεν ολοκληρωθεί, το έργο εγκαταλείπεται.
- **Εκπόνηση Μελέτης (Elaboration):** Το έργο αναλύεται σε βάθος και σχεδιάζεται. Αρχίζει να παίρνει «σάρκα και οστά». Σε αυτή τη φάση, συλλέγονται ειδικότερες απαιτήσεις (λειτουργικές/ μη λειτουργικές), γίνεται επιλογή εργαλείων, αρχιτεκτονικής και μεθόδων. Το domain αναλύεται περαιτέρω και ορίζεται το αρχικό development plan.
- **Κατασκευή (Construction):** Σε αυτό το στάδιο πρωταγωνιστεί η υλοποίηση του έργου-λογισμικού. Αναπτύσσονται εξαρτήματα και χαρακτηριστικά του συστήματος. Περιλαμβάνει, κυρίως, το γράψιμο του κώδικα, εξσφαλμάτωση, μικρές δοκιμές και αξιολόγηση. Είναι πιθανό να προστεθούν, ακόμα, ορισμένα χαρακτηριστικά που δεν υπήρχαν στην αρχική σχεδίαση.
- **Μετάβαση (Transition):** Αποτελεί την τελική φάση, όπου το λογισμικό αξιολογείται και ελέγχεται πλήρως, διανέμεται και εγκαθίσταται. Οι τελικοί χρήστες εκπαιδεύονται πάνω στο λογισμικό.

Κάθε φάση, αναλύεται περαιτέρω σε έξι διαδικασίες ως εξής:

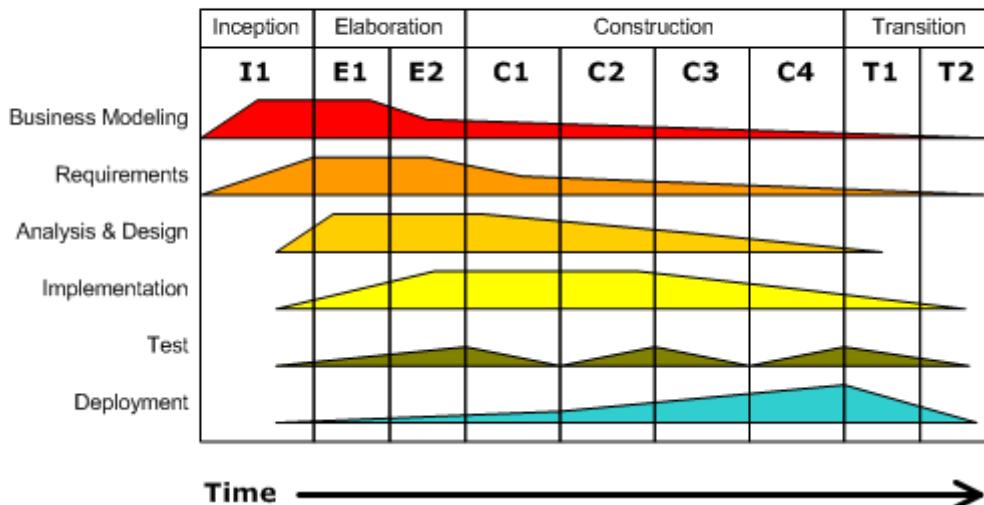
- **Business Modelling**
- **Requirements**
- **Analysis & Design**
- **Implementation**
- **Test**
- **Deployment**

Αυτές ορίζουν μια επανάληψη. Κάθε φάση μπορεί να έχει διαφορετικό αριθμό επαναλήψεων, καθώς και να δίδεται έμφαση σε διαφορετικές διαδικασίες. Ένας γενικός κανόνας είναι:



Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



Παρατηρούμε πως η ανάλυση και η σχεδίαση λαμβάνουν χώρα σχεδόν σε όλο το κύκλο ζωής του έργου. Για αυτό το λόγο, κάθε φάση της εργασίας μας αποτελείται από διάφορα σχεδιαστικά διαγράμματα μοντελοποίησης.

3. Φάση: Έναρξη (Inception)

Όταν μας ανατέθηκε εργασία, γρήγορα ορίσαμε και αναλύσαμε το domain. Τελικά, αποφασίσαμε πως το domain είναι το Film Rating.

3.1. Σύλληψη απαιτήσεων

Με γνωστό το domain του έργου μας, έπρεπε να συλλέξουμε τις λειτουργικές, αλλά και μη, απαιτήσεις, καθώς και άλλες έννοιες που θα μας είναι χρήσιμες. Ως μια μικρή ομάδα, χωρίς ειδικό στο domain, η μόνη λύση για την συλλογή των απαιτήσεων ήταν η απλή και γνωστή μέθοδος *Brainstorming*. Πριν την παρουσίαση των απαιτήσεων, ας ορίσουμε αρχικά τι είναι απαίτηση, ποιες οι λειτουργικές και ποιες οι μη λειτουργικές.

Τι είναι απαίτηση;

Μπορεί να είναι η ανάπτυξη μιας υπηρεσίας του συστήματος από τον πελάτη, ένας περιορισμός του συστήματος, ή ο τεχνικός προσδιορισμός ενός προβλήματος.

Λειτουργικές απαιτήσεις:

- Ορισμός των υπηρεσιών που θα πρέπει να υποστηρίζει το σύστημα.
- Πως πρέπει το σύστημα να αντιδρά όταν δέχεται πληροφορίες.
- Πως θα πρέπει να συμπεριφέρεται σε διάφορες καταστάσεις.

Μη λειτουργικές απαιτήσεις:

- Περιορισμοί στις υπηρεσίες ή τις λειτουργίες τις οποίες προσφέρει το σύστημα όπως χρονικοί περιορισμοί, standards κλπ.

Στη φάση αυτή, λοιπόν, συλλέξαμε τις παρακάτω λειτουργικές απαιτήσεις:



- Login, Logout, Register δυνατότητες για τους χρήστες
- Κοινοποίηση trailer
- Αξιολόγηση ταινιών από τους χρήστες
- Δημιουργία δημοσιεύσεων ταινιών με τα χαρακτηριστικά τους (Τίτλος, Είδος, Περιγραφή κ.τ.λ.)
- Προβολή όλων των δημοσιευμένων ταινιών
- Μπάρα αναζήτησης για της δημοσιευμένες ταινίες
- Επεξεργασία λογαριασμού
- Επεξεργασία δημοσίευσης
- Προβολή όλων των εγγεγραμμένων χρηστών (Administrator)
- Αποβολή χρήστη (Administrator)

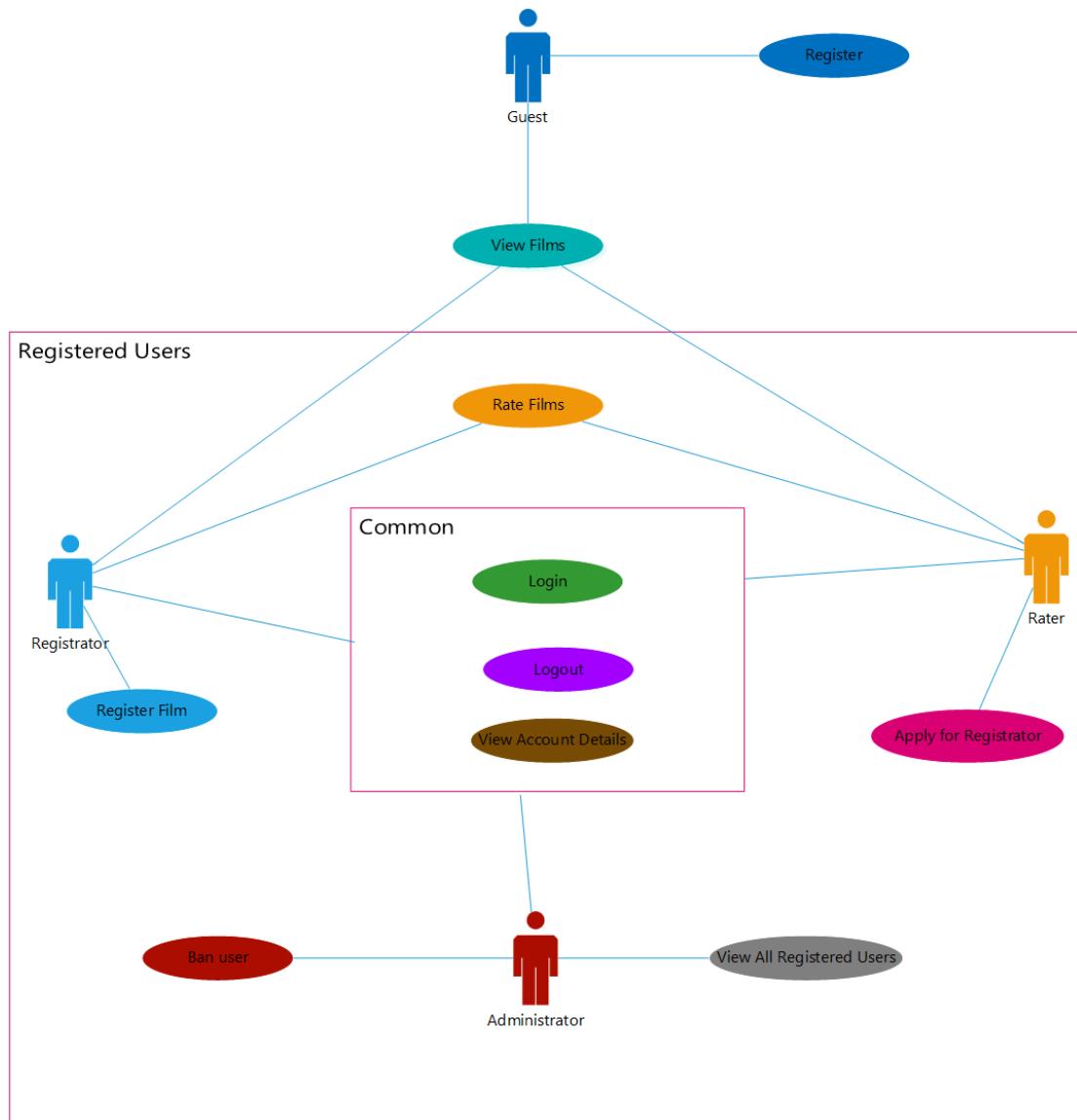
Επιπλέον, οι μη λειτουργικές απαιτήσεις που ορίσαμε είναι:

- Η κλίμακα της αξιολόγησης να είναι 5 αστέρων.
- Αυθεντικοποίηση
- Οι επισκέπτες δεν μπορούν να βαθμολογήσουν ή να αξιολογήσουν ταινίες

3.2. Ανάλυση και Σχεδιασμός

3.2.1. Διαγράμματα Περιπτώσεων Χρήσης (Πρώτη έκδοση)

Βάσει των παραπάνω απαιτήσεων, προέκυψε η πρώτη έκδοση του διαγράμματος Περιπτώσεων Χρήσης (Use Case Diagram)



Διάγραμμα 1 Use Case v1

*Σε επόμενες εκδόσεις, ο Registrator μετονομάστηκε σε Content Manager.

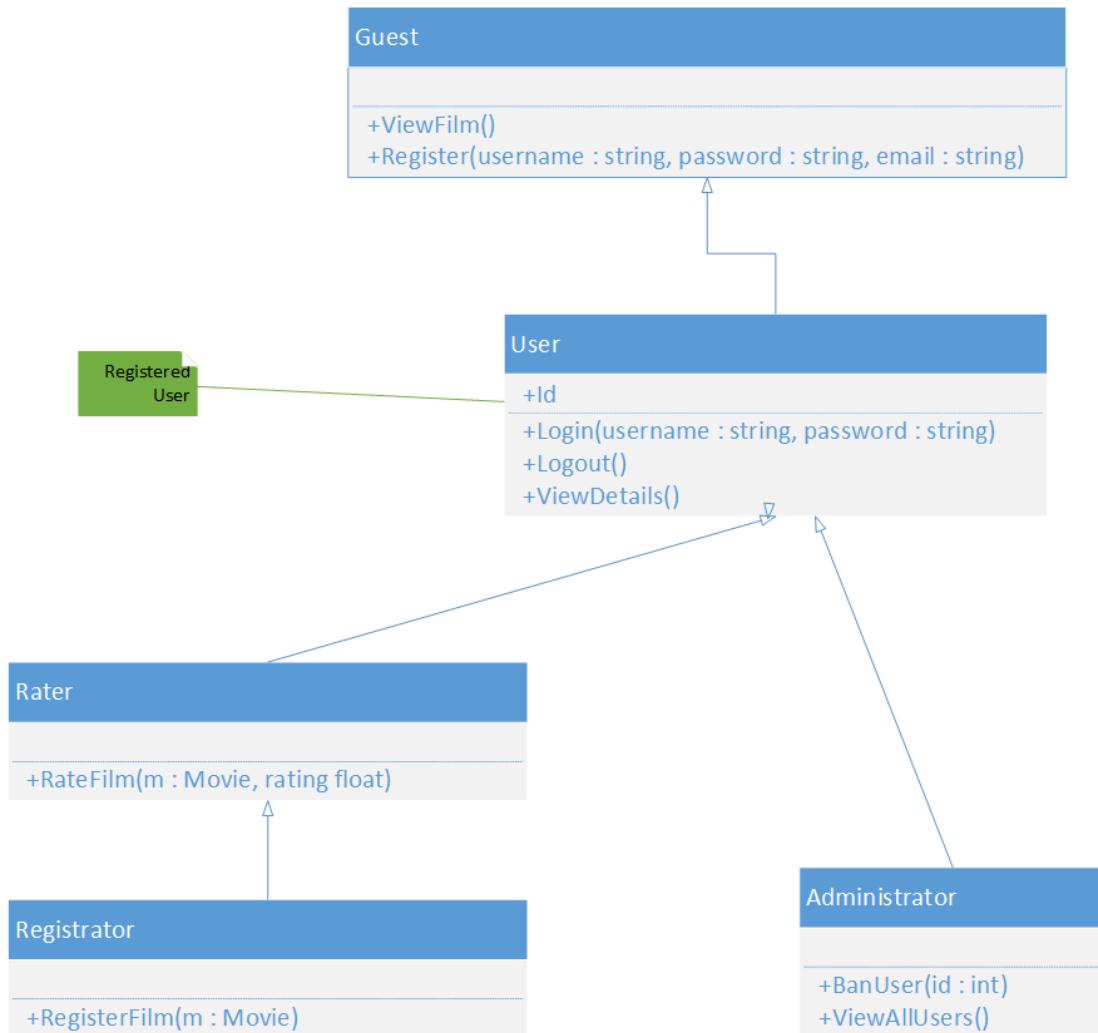
Σε αυτό το σημείο να αναφερθεί πως θεωρούμε 4 είδη χρηστών.

- **Επισκέπτης (Guest)** - Οι μη εγγεγραμμένοι χρήστες. Μπορούν μόνο να δουν τις δημοσιευμένες ταινίες.
- **Βαθμολογητής (Rater)** - Το πιο απλό στάδιο εγγεγραμμένων χρηστών. Μπορούν μόνο να δουν τις δημοσιευμένες ταινίες και να τις βαθμολογήσουν
- **Διαχειριστής Περιεχομένου (Content Manager)** - Η αναβαθμισμένη έκδοση του Rater. Μπορούν να δουν τις δημοσιευμένες ταινίες, να τις βαθμολογήσουν αλλά και να δημιουργήσουν νέες δημοσιεύσεις
- **Διαχειριστής (Administrator)** - Ο χρήστης που επιβλέπει την λειτουργία της εφαρμογής. Μπορεί να δει όλους τους εγγεγραμμένους χρήστες αλλά και να τους αποβάλει από την εφαρμογή.

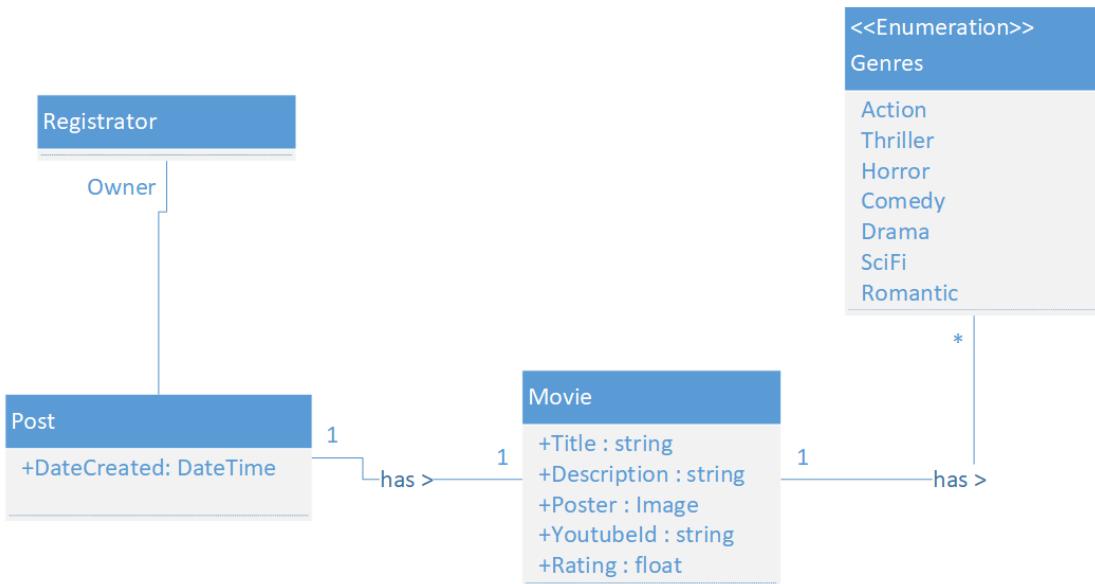


3.2.2. Διαγράμματα Τάξεων (Πρώτη έκδοση)

Έχοντας υπόψη διάγραμμα Περιπτώσεων χρήσης, έπρεπε να ορίσουμε τις βασικές κλάσεις που θα μας απασχολήσουν. Έτσι, παρήχθησαν τα εξής Διαγράμματα κλάσεων (Class Diagram).



Διάγραμμα 2 User Class Diagram



Διάγραμμα 3 Post Class Diagram

4. Φάση: Εκπόνηση Μελέτης (Elaboration)

Μετά από αρκετή συζήτηση, άρχισε η επόμενη φάση, η φάση της εκπόνησης μελέτης. Σε αυτό το στάδιο, συζητήθηκαν πιο τεχνικές λεπτομέρειες καθώς και η λογική των διαδικασιών που θα υλοποιούνταν. Παράλληλα, άρχισε δειλά-δειλά η υλοποίηση, δημιουργώντας τους βασικούς φακέλους και αρχεία, και γράφοντας τον βασικό κώδικα.

4.1. Ανάλυση και σχεδιασμός

Η φάση αυτή είχε επαναληπτικό χαρακτήρα. Διήρκησε 2 επαναλήψεις και σε κάθε επανάληψη υπήρχαν πρόσθετα χαρακτηριστικά και διάφορες διορθώσεις. Αναλυτικότερα, οι περιπτώσεις χρήσης που προστέθηκαν είναι:

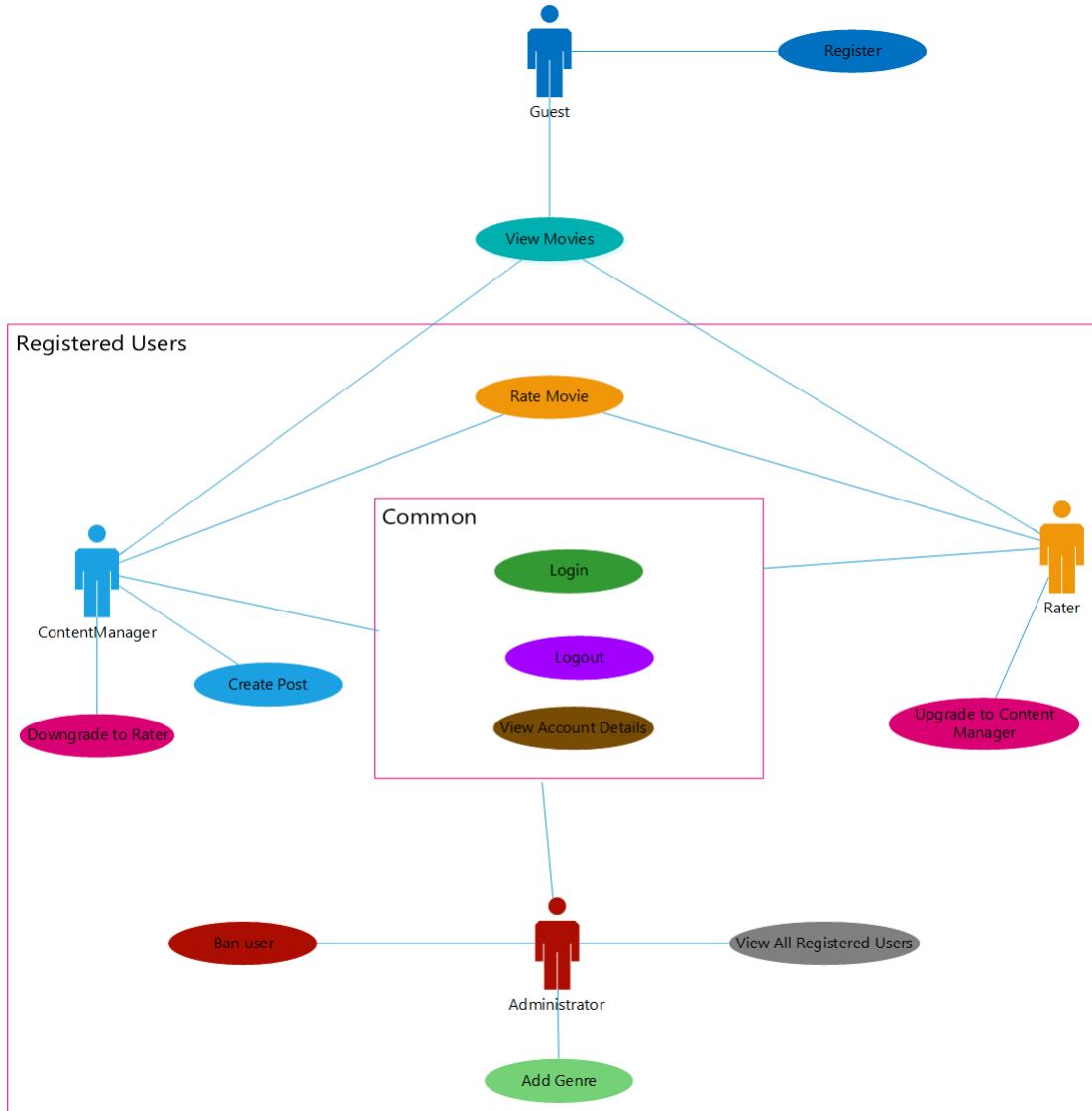
- 1^η επανάληψη
 - O administrator να μπορεί να δημιουργεί νέα genres
 - O Rater να έχει τη δυνατότητα να αναβαθμίζεται σε Content Manager, αλλά και το αντίστροφο.
- 2^η επανάληψη
 - Κάθε εγγεγραμμένος χρήστης να μπορεί να δει σε λίστα όλες τις ταινίες τις οποίες έχει βαθμολογήσει.
 - Κάθε Content Manager να μπορεί να δει σε λίστα όλες τις δημοσιεύσεις που έχει δημιουργήσει.

Να σημειωθεί πως τα διαγράμματα που παρουσιάζονται παρακάτω περιέχουν μεγάλο βαθμό αφαίρεσης (κυρίως τα συμπεριφορικά).



4.1.1. Διάγραμμα Περιπτώσεων Χρήσης (Δεύτερη έκδοση)

Λαμβάνοντας υπόψη τα νέα χαρακτηριστικά της εφαρμογής, το νέο διάγραμμα περιπτώσεων χρήσης είναι:

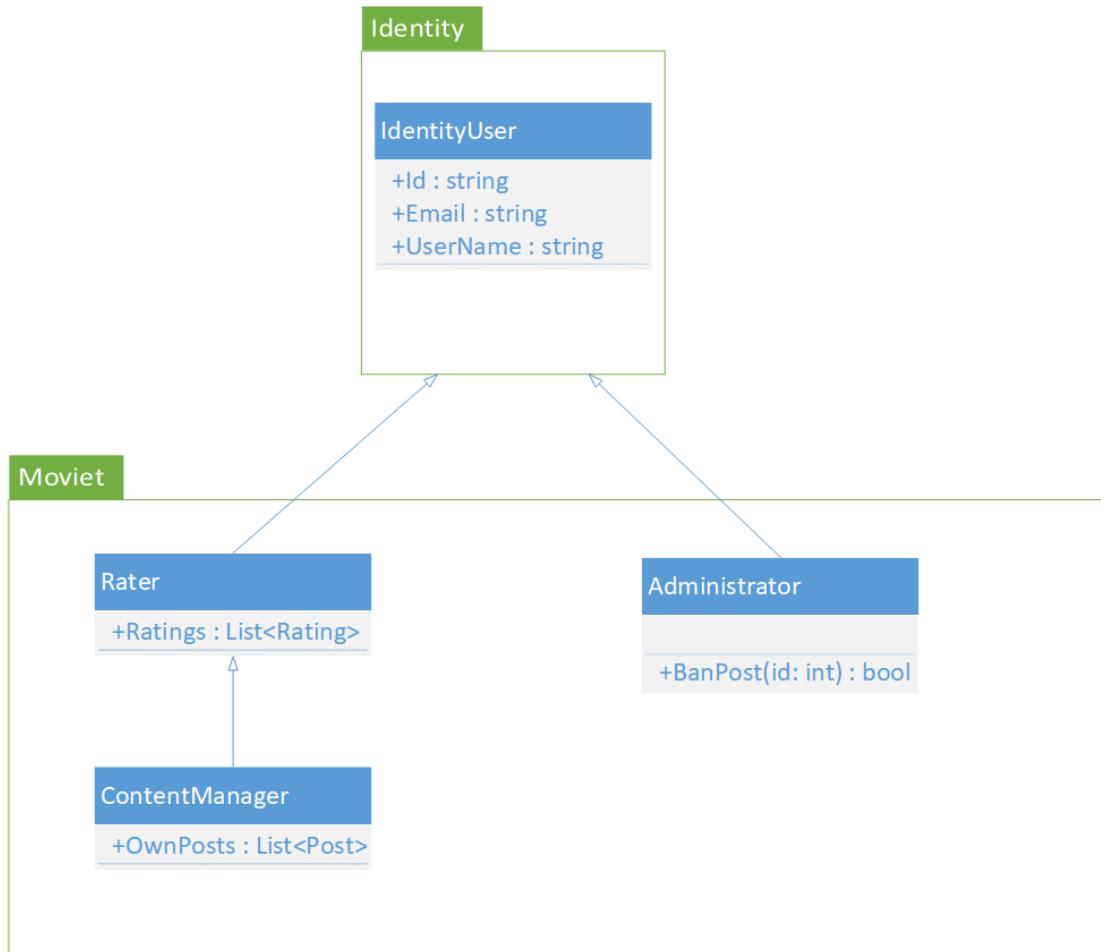


Διάγραμμα 4 Use Case Diagram v2



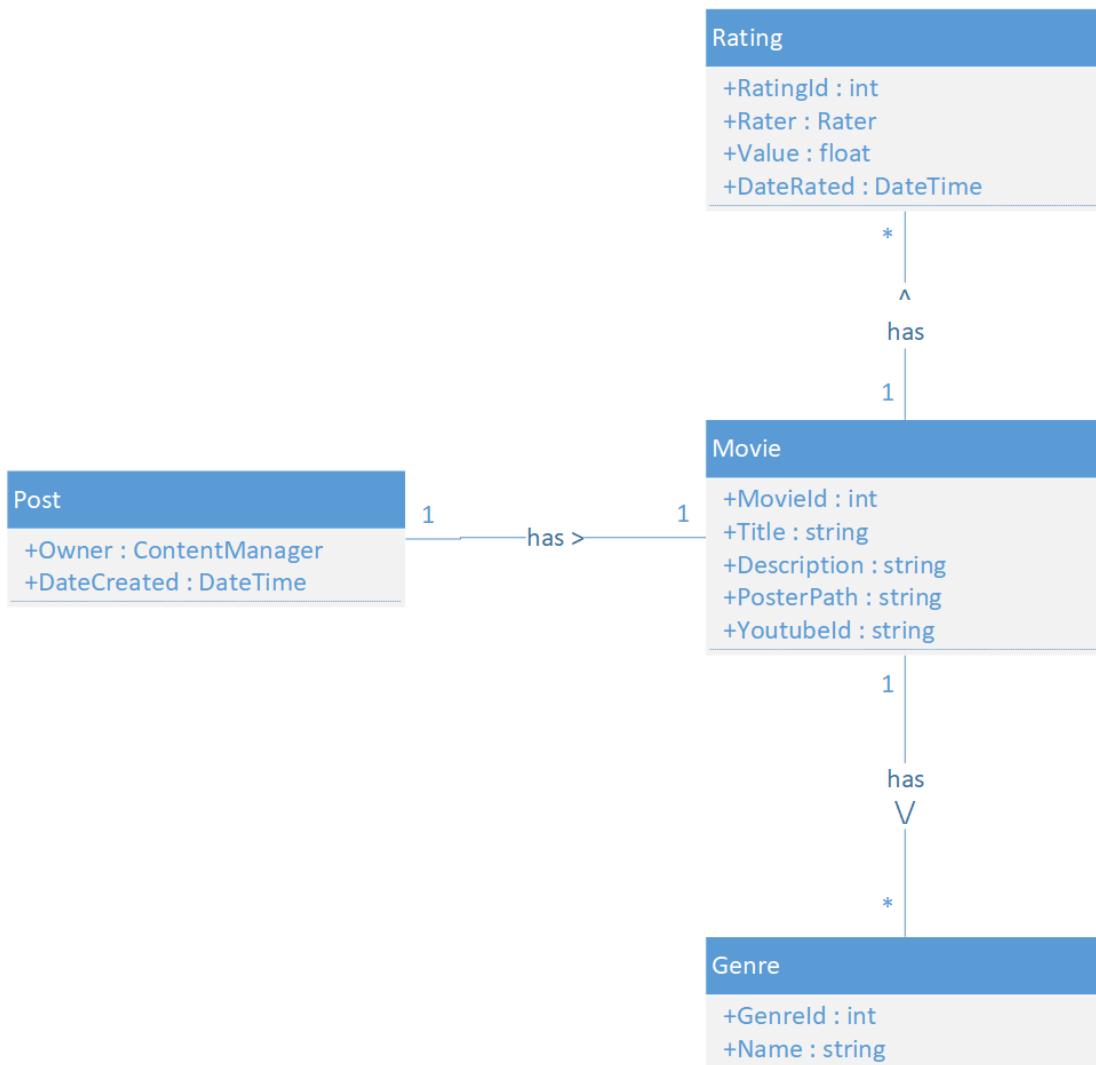
4.1.2. Διαγράμματα Τάξεων (Δεύτερη έκδοση)

Με την έναρξη της υλοποίησης, παρουσιάστηκαν διάφορα προβλήματα στη δομή των κλάσεων. Έτσι, τα ανανεωμένα, αλλά και νέα, διαγράμματα κλάσεων είναι:

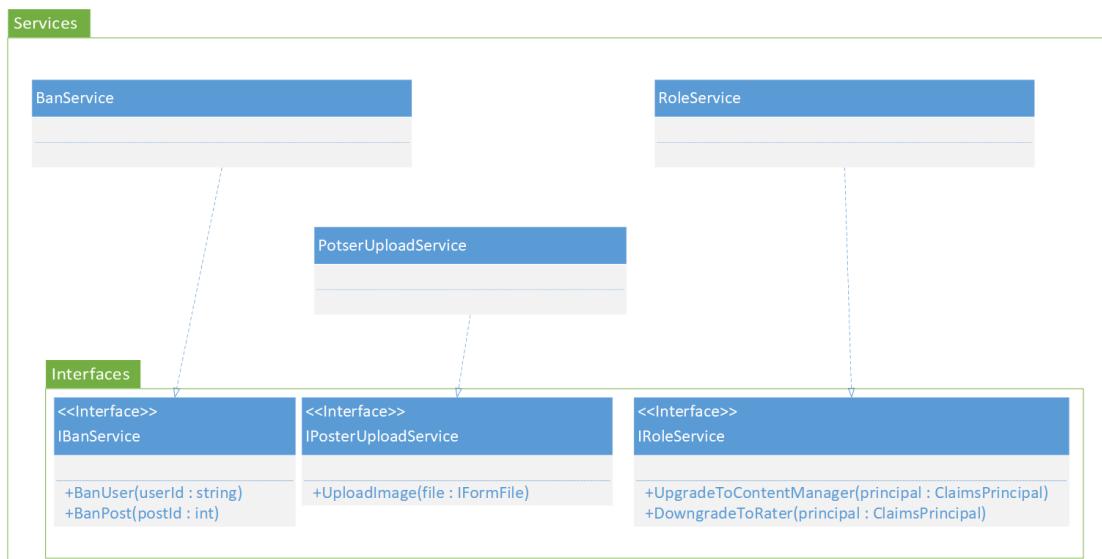


Διάγραμμα 5 User Class Diagram v2

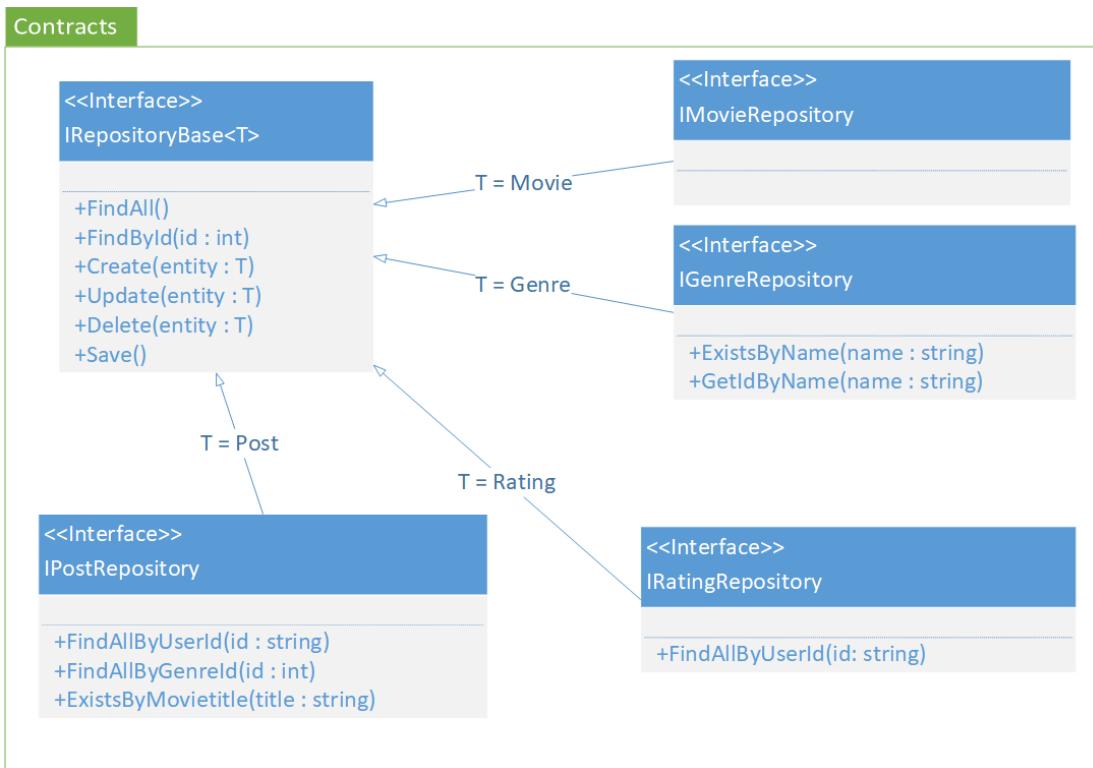
Το πακέτο Identity ανήκει στο ASP.NET Core. Διευκολύνει τη διαχείριση των χρηστών και προσδίδει δυνατότητες login, logout και registration.



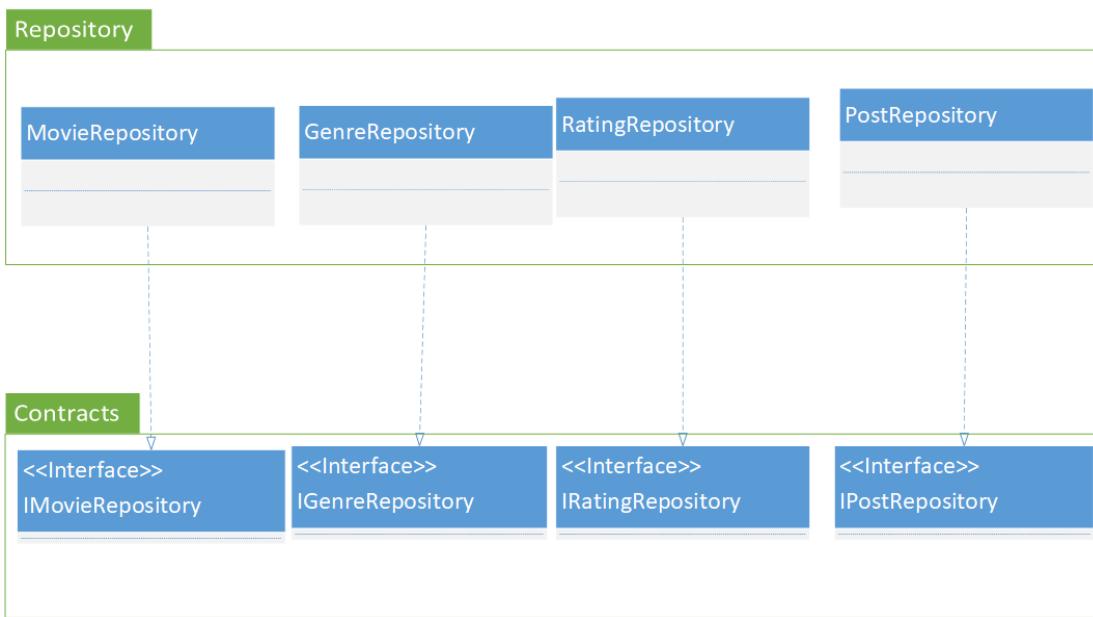
Διάγραμμα 6 Post Class Diagram v2



Διάγραμμα 7 Services Class Diagram v1

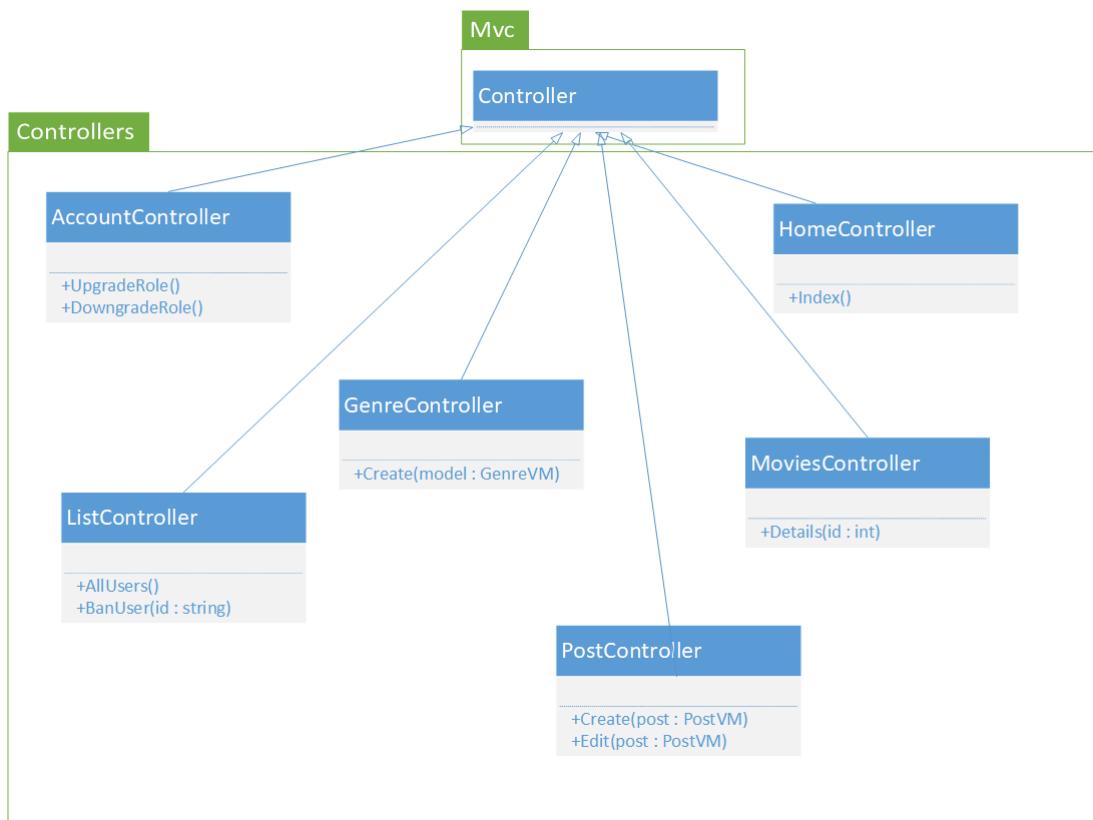


Διάγραμμα 8 Contracts Class Diagram v1



Διάγραμμα 9 Repository Class Diagram v1

Τα δύο παραπάνω διαγράμματα αποτελούν το Repository Pattern.

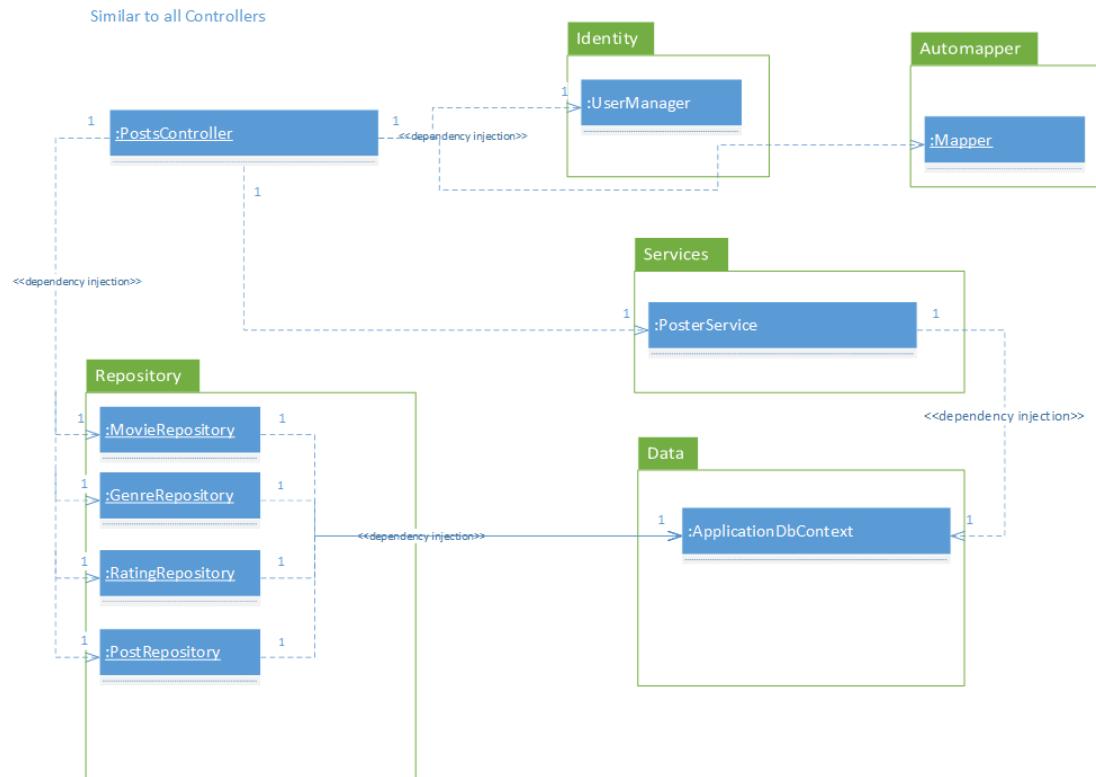


Διάγραμμα 10 Controller Class Diagram v1



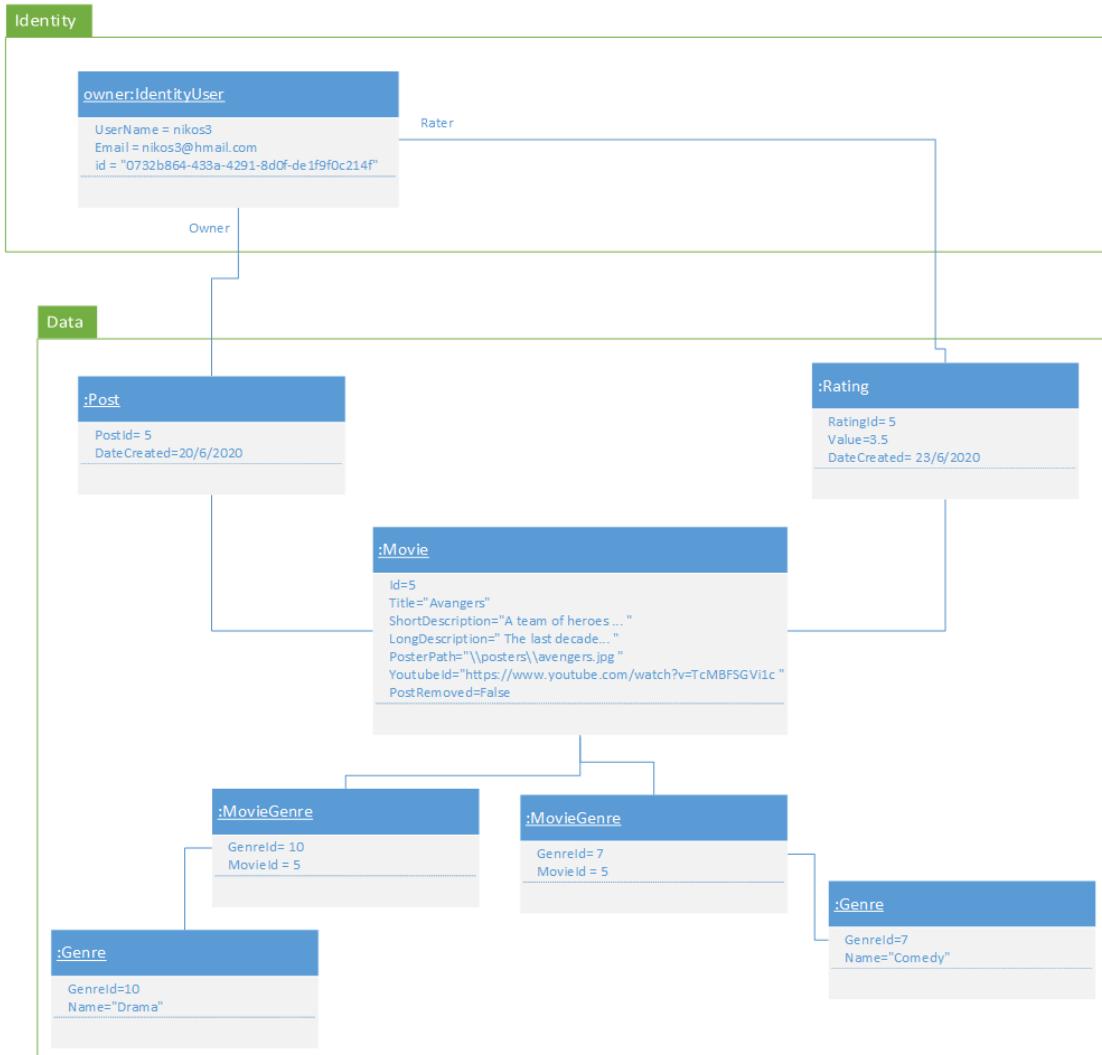
4.1.3. Διαγράμματα Αντικειμένων (Πρώτη έκδοση)

Μια ενδεικτική κατάσταση των αντικειμένων κατά την εκτέλεση της εφαρμογής, θα μπορούσε να είναι η παρακάτω:



Διάγραμμα 11 Controller Object Diagram v1

Στο διάγραμμα 11, φαίνεται ενδεικτικά η κατάσταση ενός controller καθώς και των εξαρτήσεων του. Σημειώνεται το dependency injection που γίνεται, αλλά δεν εμφανίζεται ο injector. Όλα τα dependency injections τα αναλαμβάνει το framework. Όμοια συμβαίνει και με τα άλλα controllers.



Διάγραμμα 12 Data Object Diagram v1

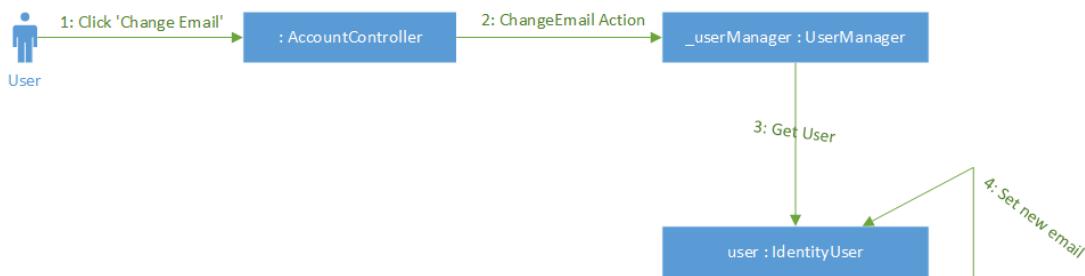
Στο παραπάνω διάγραμμα φαίνεται η κατάσταση των αντικειμένων που αφορούν μια δημοσιευμένη ταινία. Όπως αναφέρεται και στην αναφορά ελέγχου αργότερα, μόνο ο δημιουργός της δημοσίευσης μπορεί να βαθμολογήσει την ταινία στην παρούσα εκτελέσιμη έκδοση.



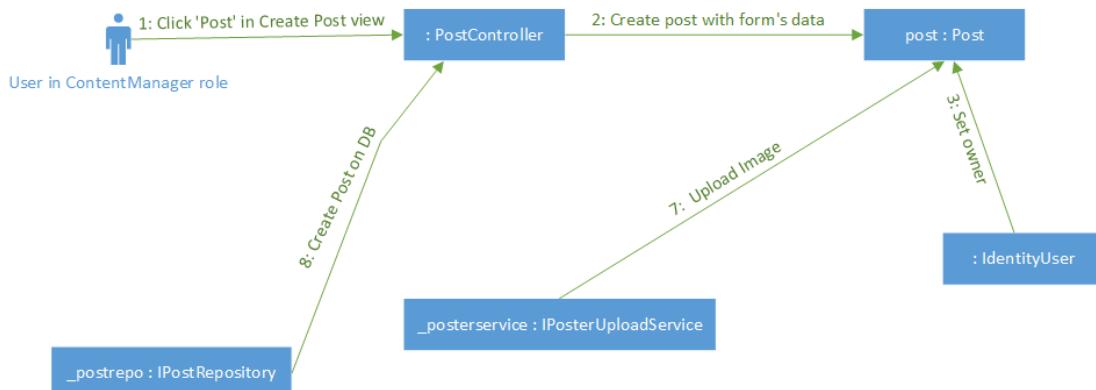
4.1.4. Διαγράμματα Συνεργασίας (Πρώτη έκδοση)

Παρακάτω παρουσιάζονται ορισμένα διαγράμματα συνεργασίας. Διαγράμματα που μοιάζουν σε μεγάλο βαθμό με άλλα, έχουν παραλειφθεί. Γίνεται, κάθε φορά, αναφορά σε αυτά τα διαγράμματα στο πάνω αριστερό μέρος των εικόνων.

[Similar to Change Username](#)



Διάγραμμα 13 Change Email Collaboration Diagram v1



Διάγραμμα 14 Create Post Collaboration Diagram v1

[Similar to Ban User](#)



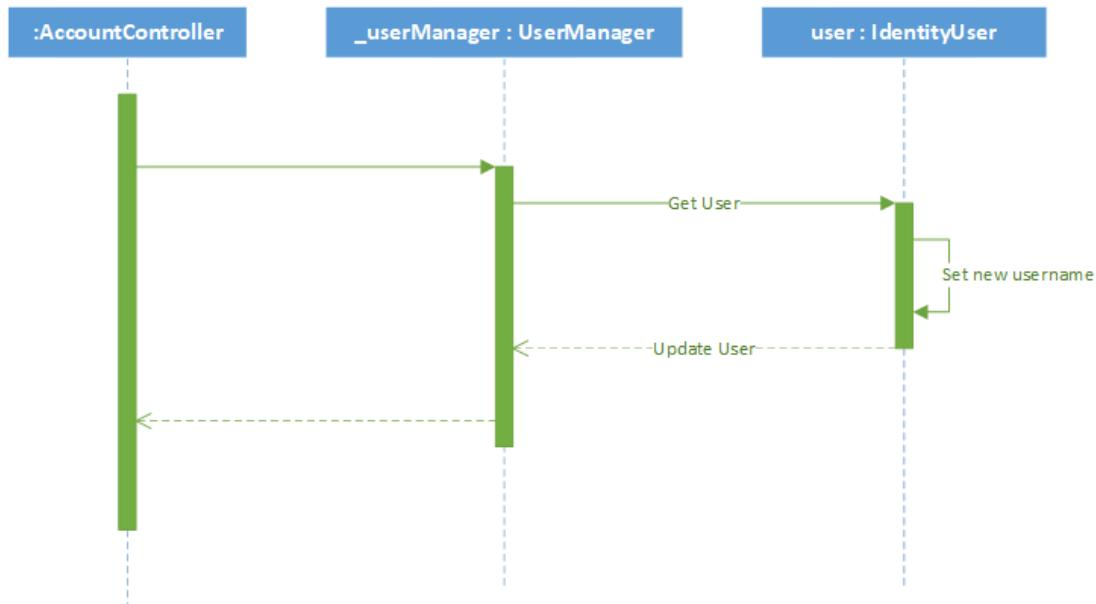
Διάγραμμα 15 Ban Post Collaboration Diagram v1



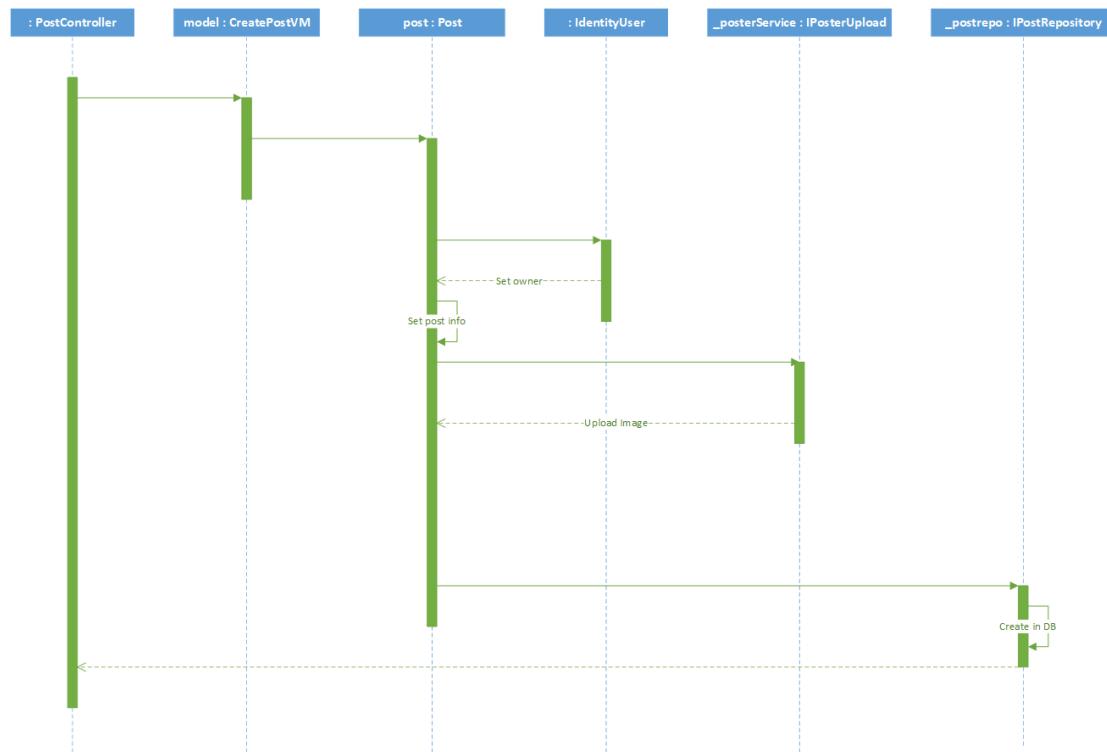
4.1.5. Διαγράμματα Σειράς (Πρώτη έκδοση)

Παρακάτω παρουσιάζονται ορισμένα διαγράμματα σειράς. Διαγράμματα που μοιάζουν σε μεγάλο βαθμό με άλλα, έχουν παραλειφθεί. Γίνεται, κάθε φορά, αναφορά σε αυτά τα διαγράμματα στο πάνω αριστερό μέρος των εικόνων.

[Similar to Change Email](#)



Διάγραμμα 16 Change Username Sequence Diagram v1

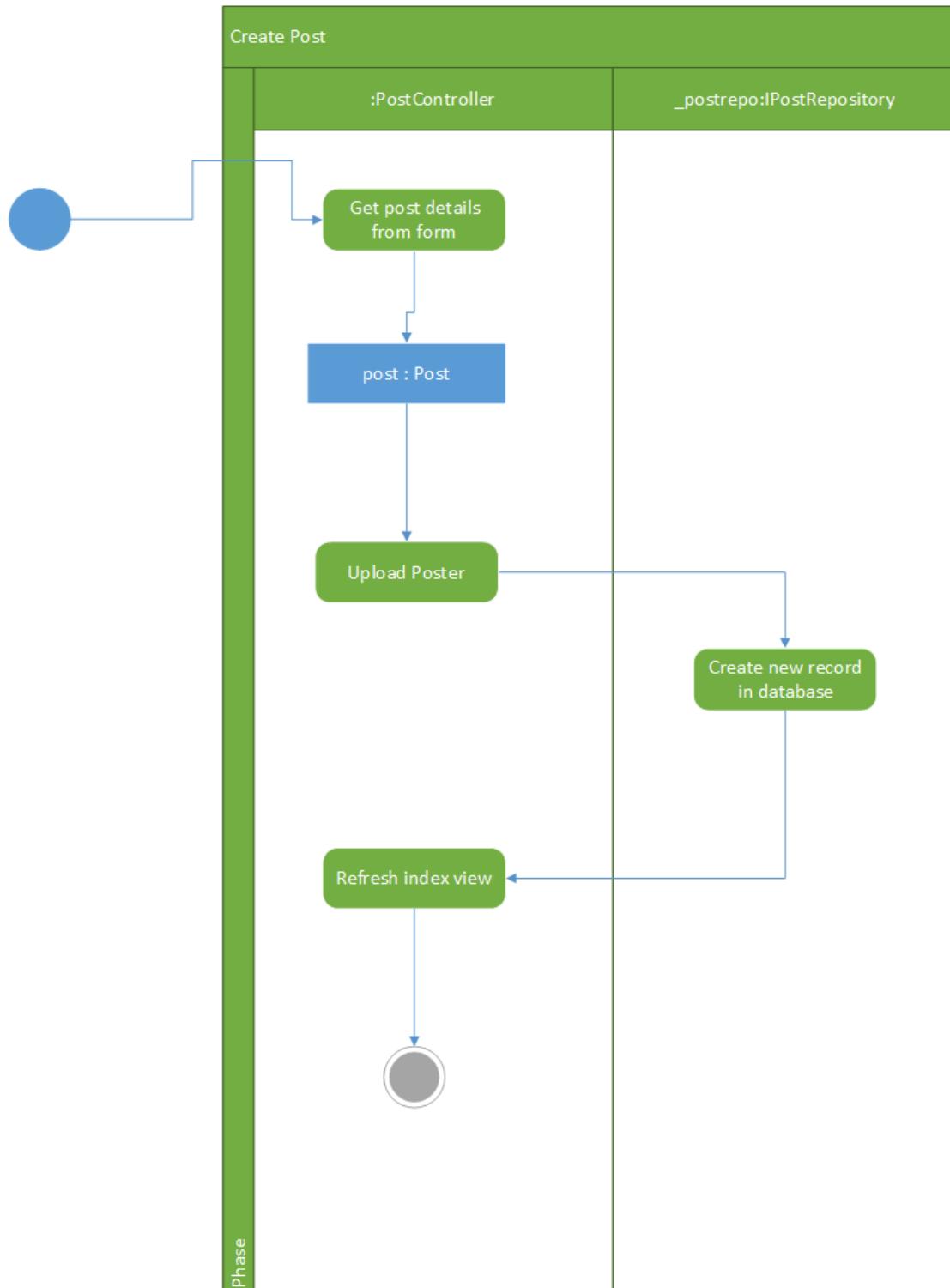


Διάγραμμα 17 Create Post Sequence Diagram v1

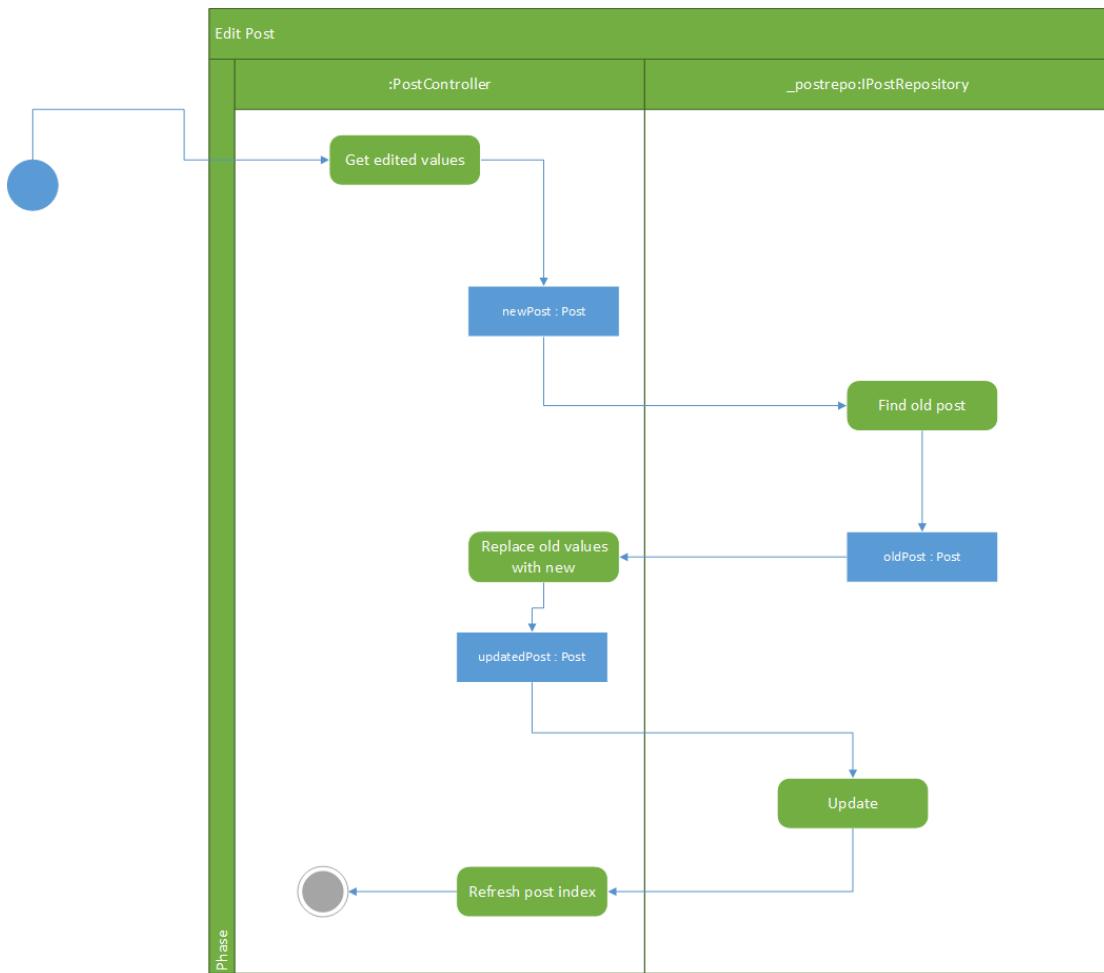


4.1.6. Διαγράμματα Δραστηριοτήτων (Πρώτη έκδοση)

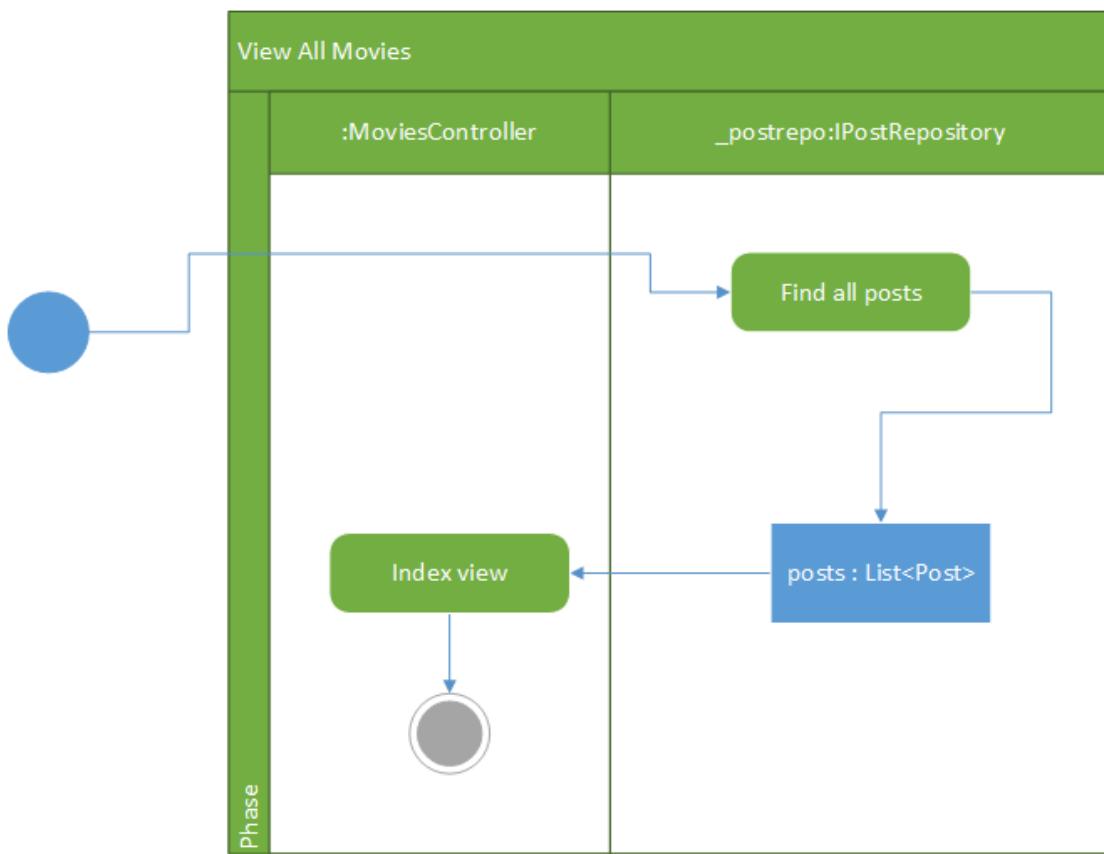
Δεδομένων των περιπτώσεων χρήσης, έχουμε τα ακόλουθα διαγράμματα δραστηριοτήτων.



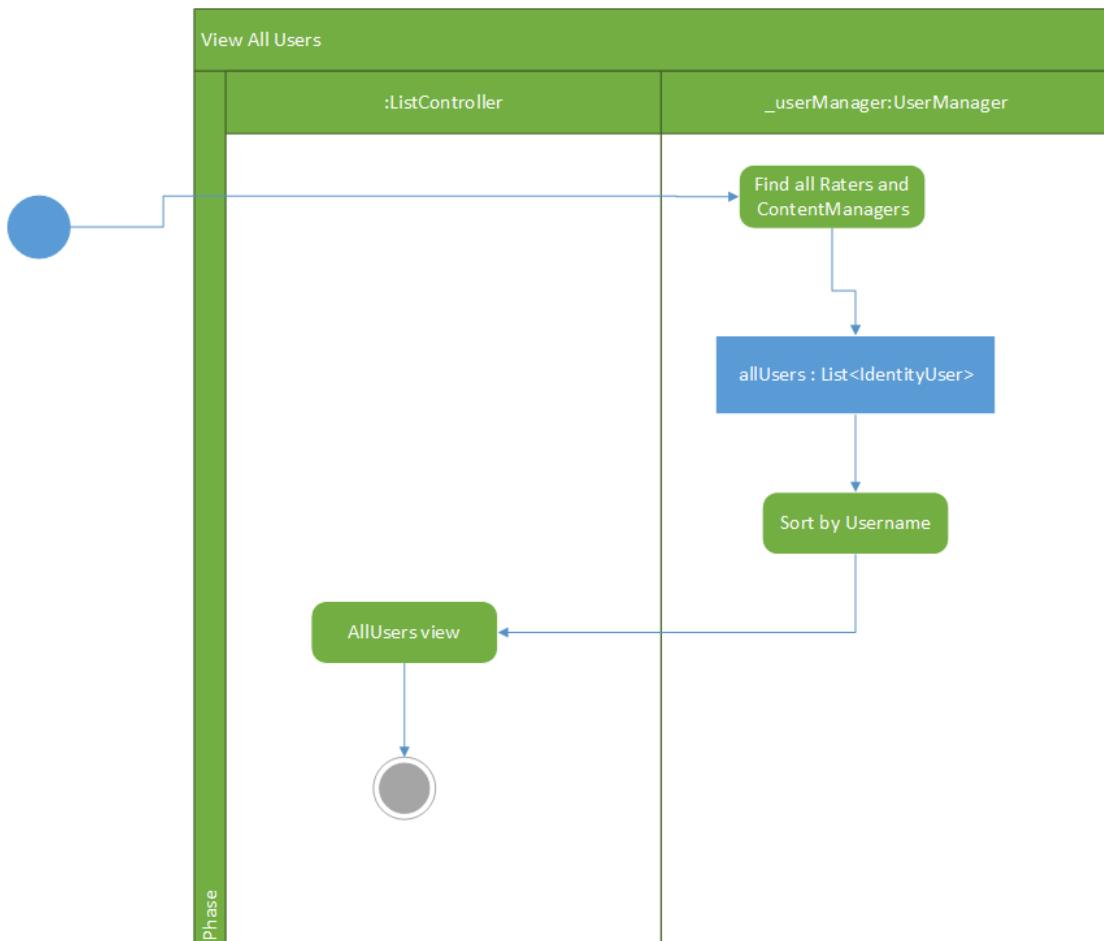
Διάγραμμα 18 Create Post Activity Diagram v1



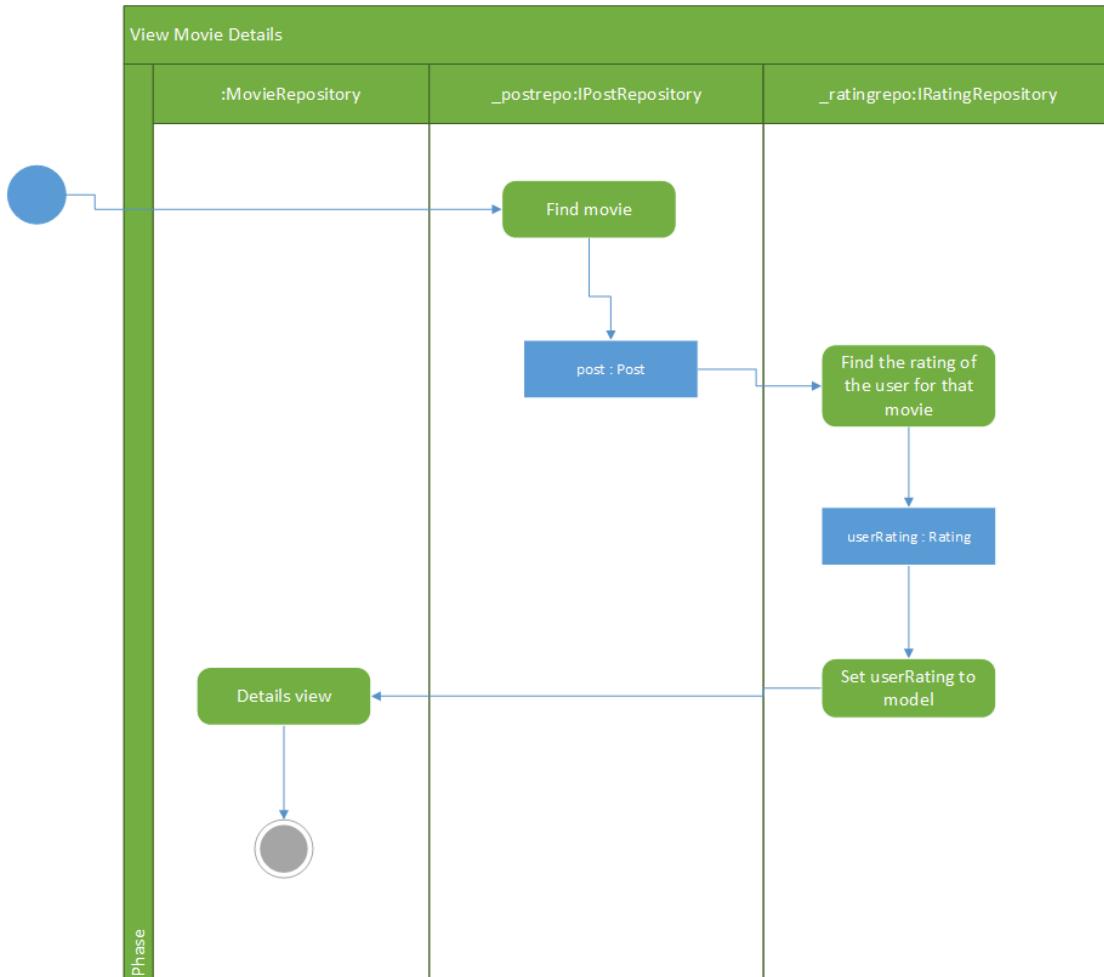
Διάγραμμα 19 Edit Post Activity Diagram v1



Διάγραμμα 20 View All Movies Activity Diagram v1



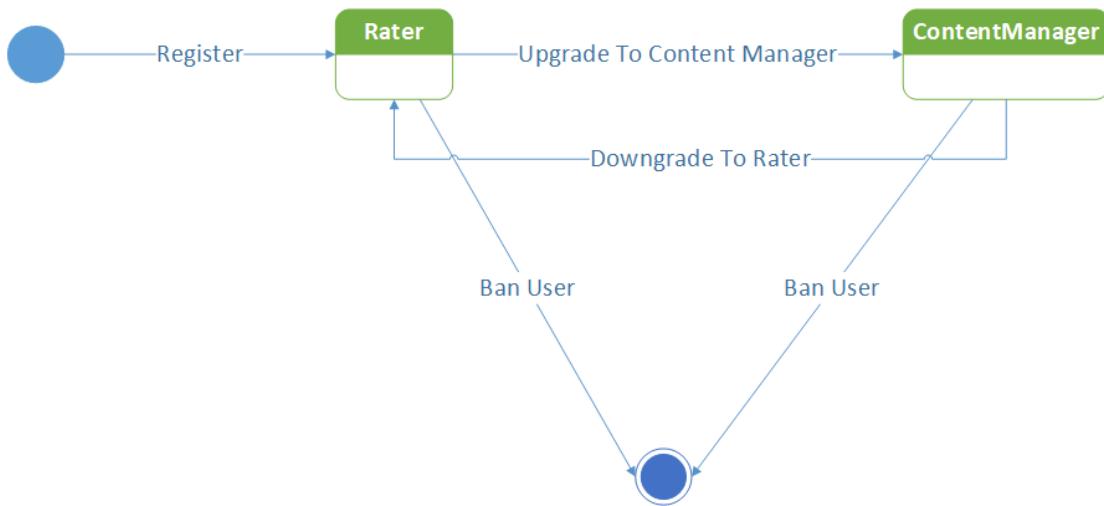
Διάγραμμα 21 View All Users Activity Diagram v1



Διάγραμμα 22 View Movie Details Activity Diagram v1

4.1.7. Διαγράμματα Καταστάσεων (Πρώτη έκδοση)

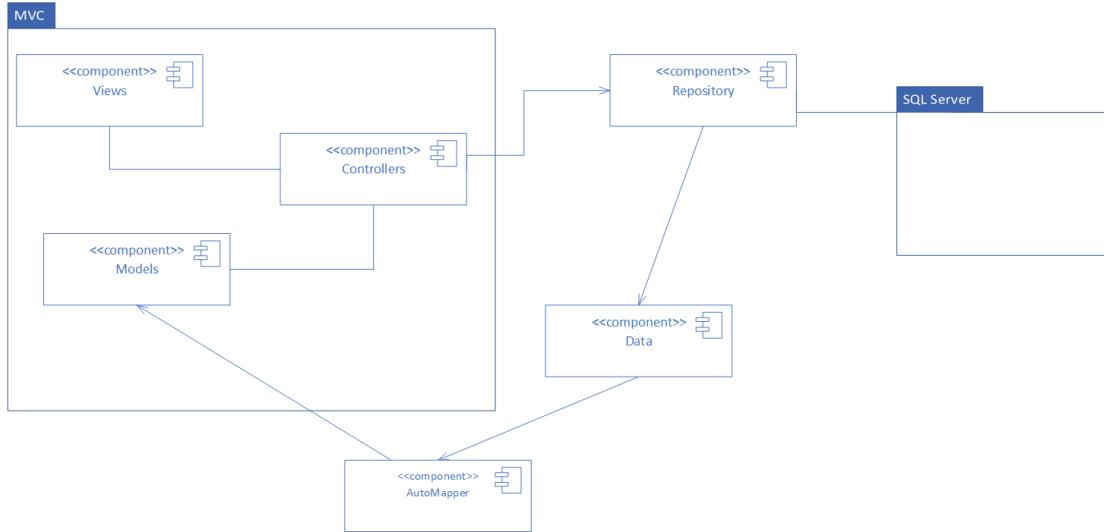
Παρατηρήθηκε μόνο μια μηχανή καταστάσεων, αυτή που αφορά τις καταστάσεις του χρήστη.



Διάγραμμα 23 User State Machine Diagram

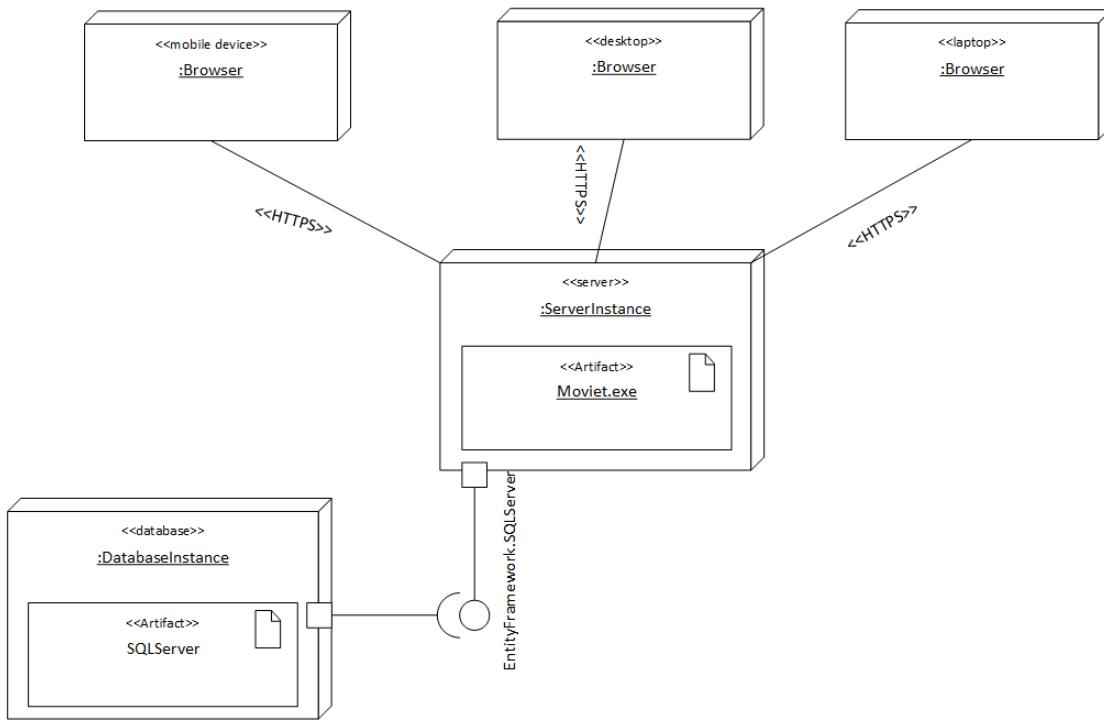


4.1.8. Διαγράμματα Εξαρτημάτων (Πρώτη έκδοση)



Διάγραμμα 24 Component Diagram v1

4.1.9. Διαγράμματα Διανομής (Πρώτη έκδοση)



Διάγραμμα 25 Deployment Diagram v1



4.2. Υλοποίηση και έλεγχος

4.2.1. Υλοποίηση: 1^η εκτελέσιμη έκδοση

Όπως αναφέρθηκε προηγουμένως, σε αυτή τη φάση άρχισε και η κατασκευή του λογισμικού. Στην πρώτη επανάληψη των διαδικασιών της RUP, στήθηκε η δομή των βασικών αρχείων και δημιουργήθηκαν οι πρώτες κλάσεις. Στη συνέχεια, προστέθηκαν βασικές λειτουργίες.

Να σημειωθεί πως παρόλο που η αρχική σχεδίαση περιείχε 3 κλάσεις χρηστών για την διαχείριση των διαφορετικών τύπων χρηστών, τελικά δεν υλοποιήθηκε η συγκεκριμένη ιδέα. Αντιθέτως, για τεχνικούς λόγους, χρησιμοποιήθηκε μια κλάση που παρείχε το Identity και ο διαχωρισμός έγινε με τη βοήθεια ρόλων (Μια δυνατότητα που προσφέρει το Identity).

Αναλυτικότερα η δομή:

- **Areas:** Ο φάκελος Areas περιέχει ομάδες αρχείων κάτω από μια συγκεκριμένη λειτουργική έννοια. Στην προκειμένη περίπτωση, περιέχει τον φάκελο Identity, στον οποίο βρίσκονται όλα τα απαραίτητα razor pages για τις λειτουργίες login, logout και register, καθώς ακόμα άλλες λειτουργίες που προσφέρει το ASP.NET Core Identity, οι οποίες όμως δεν χρησιμοποιήθηκαν.
- **Contracts:** Ο φάκελος Contracts περιέχει interfaces απαραίτητα για το Repository Pattern.
- **Controllers:** Ο φάκελος Controllers περιέχει όλα τα controllers που χρησιμοποιούνται.
 - **AccountController:** Υπεύθυνο για τις αιτήσεις που αφορούν το λογαριασμό του χρήστη. Σε αυτή την υλοποίηση, περιέχει actions για την αναβάθμιση του χρήστη από Rater σε Content Manager και το αντίστροφο.
 - **GenresController:** Υπεύθυνο για τις αιτήσεις που αφορούν την διαχείριση των διαθέσιμων ειδών που υπάρχουν στην εφαρμογή. Παρέχει actions για την δημιουργία, την επεξεργασία και τη διαγραφή ενός είδους. Μπορεί να χρησιμοποιηθεί μόνο από χρήστες που βρίσκονται σε ρόλο Administrator.
 - **HomeController:** Υπεύθυνο για τις αιτήσεις που αφορούν τη προβολή της αρχικής σελίδας.
 - **ListController:** Υπεύθυνο για τις αιτήσεις που αφορούν την προβολή των εγγεγραμμένων χρηστών και την αποβολή τους. Μπορεί να χρησιμοποιηθεί μόνο από χρήστες που βρίσκονται σε ρόλο Administrator.
 - **MoviesController:** Υπεύθυνο για τις αιτήσεις που αφορούν την προβολή των λεπτομερειών μιας ταινίας-δημοσίευσης.
 - **PostsController:** Υπεύθυνο για τις αιτήσεις που αφορούν τη δημιουργία και την επεξεργασία δημοσιεύσεων. Μπορεί να χρησιμοποιηθεί μόνο από χρήστες που βρίσκονται σε ρόλο Content Manager.
- **Data:** Ο φάκελος Data περιέχει POCO κλάσεις, καθώς και το context απαραίτητο για την επικοινωνία μετα βάση.
 - **ApplicationDbContext:** Προσφέρει πρόσβαση στα δεδομένα της βάσης. Είναι ο ενδιάμεσος κρίκος μεταξύ της εφαρμογής και του Entity Framework Core.
 - **Admin:** Περιγράφει τον διαχειριστή. Τελικά δεν υλοποιήθηκε διότι έγινε χρήση ρόλων.



- **ContentManager:** Περιγράφει τον διαχειριστή περιεχομένου. Τελικά δεν υλοποιήθηκε διότι έγινε χρήση ρόλων.
- **Genres:** Περιγράφει ένας είδος ταινίας.
- **Movie:** Περιγράφει μια ταινία.
- **MovieGenre:** Περιγράφει τη σύνδεση μιας ταινίας με ένα είδος ταινίας.
- **Post:** Περιγράφει μια δημοσίευση ενός χρήστη.
- **Rater:** Περιγράφει τον βαθμολογητή. Τελικά δεν υλοποιήθηκε διότι έγινε χρήση ρόλων.
- **Rating:** Περιγράφει μια αξιολόγηση μια ταινίας από ένα χρήστη.
- **Helpers:** Ο φάκελος Helpers περιέχει βιοθητικές κλάσεις που λειτουργούν ως Tag Helpers στα views.
 - **CustomHelper:** Προσφέρει ανάδραση στην αλλαγή των σελίδων της εφαρμογής.
- **Models:** Ο φάκελος Models περιέχει τα απαραίτητα μοντέλα που βοηθούν στην παρουσίαση των POCO στον χρήστη. Τα αρχεία τελειώνουν σε VM που σημαίνει ViewModel. Αυτό, ώστε να είναι πιο κατανοητή η χρήση τους.
 - **ContentManagerVM:** Κατάλληλο για παρουσίαση στοιχείων των Content Managers. Τελικά δεν υλοποιήθηκε διότι έγινε χρήση ρόλων.
 - **GenreVM:** Κατάλληλο για την παρουσίαση των ειδών των ταινιών.
 - **IdentityUserVM:** Κατάλληλο για την παρουσίαση των στοιχείων των εγγεγραμμένων χρηστών.
 - **MovieGenreVM:** Κατάλληλο για τη σύνδεση ενός μοντέλου ταινίας με ένα μοντέλο είδους.
 - **MovieVM:** Κατάλληλο για την παρουσίαση μιας ταινίας.
 - **PostVM:** Κατάλληλο για την παρουσίαση μιας δημοσίευσης.
 - **RatingVM:** Κατάλληλο για την παρουσίαση των βαθμολογιών.
- **Repository:** Ο φάκελος Repository περιέχει απαραίτητες κλάσεις για την πρόσβαση των δεδομένων της βάσης. (Repository Pattern)
 - **GenreRepository:** Παρέχει πρόσβαση σε δεδομένα των διαθέσιμων ειδών.
 - **MovieRepository:** Παρέχει πρόσβαση σε δεδομένα των καταχωρημένων ταινιών.
 - **PostRepository:** Παρέχει πρόσβαση σε δεδομένα των δημοσιεύσεων.
- **Services:** Ο φάκελος Services περιέχει διεπαφές και κλάσεις υλοποίησης που αφορούν ορισμένες λειτουργίες business logic. Η δημιουργία των διεπαφών δεν ήταν απαραίτητη, όμως ήταν ένα καλό best practice πάνω σε έννοιες και αρχές του αντικειμενοστρεφής προγραμματισμού.
 - **Interfaces:** Ο φάκελος Interfaces περιέχει τις εν λόγω διεπαφές.
 - **IBanService:** Υπεύθυνο για την αποβολή ενός χρήστη από την εφαρμογή.
 - **IPosterUploadService:** Υπεύθυνο για την αποθήκευση των αφισών στην εφαρμογή.
 - **IRoleService:** Υπεύθυνο για την αναβάθμιση και υποβάθμιση ενός χρήστη.
 - **BanService:** Η υλοποίηση του interface IBanService.
 - **PosterUploadService:** Η υλοποίηση του interface IPosterUploadService.
 - **RoleService:** Η υλοποίηση του interface IRoleService.



- **Views:** Ο φάκελος Views περιέχει όλα τα views της εφαρμογής, χρήσιμα για την παραγωγή των τελικά εμφανιζόμενων ιστοσελίδων.
 - **Account:** Περιέχει όλα τα views που αφορούν το λογαριασμό του χρήστη.
 - **Index:** Προβάλει τις πληροφορίες του χρήστη και του προσφέρει δυνατότητες αλλαγής email/username και αναβάθμισης/υποβάθμισης.
 - **Genres:** Περιέχει όλα τα views που αφορούν τα διαθέσιμα είδη ταινιών. Είναι διαθέσιμα μόνο σε διαχειριστές.
 - **Create:** Προβάλει μια φόρμα για την δημιουργία ενός διαθέσιμου είδους.
 - **Index:** Προβάλει μια λίστα με όλα τα διαθέσιμα είδη ταινιών.
 - **Rename:** Προβάλει μια φόρμα επεξεργασίας του ονόματος ενός είδους.
 - **Home:** Περιέχει όλα τα views που αφορούν την αρχική σελίδα της εφαρμογής.
 - **Index:** Προβάλει πληροφορίες για την εφαρμογή, καθώς και τέσσερεις από τις πιο πρόσφατες αναρτήσεις και πιο υψηλά βαθμολογημένες.
 - **List:** Περιέχει όλα τα views που αφορούν την προβολή διαφόρων πληροφοριών στον διαχειριστή.
 - **AllUsers:** Προβάλει σε λίστα όλους τους εγγεγραμμένους χρήστες.
 - **Movies:** Περιέχει όλα τα views που αφορούν την προβολή ταινιών-δημοσιεύσεων.
 - **Details:** Προβάλει πληροφορίες μιας ταινίας-δημοσίευσης.
 - **Index:** Προβάλει όλες τις διαθέσιμες ταινίες-δημοσιεύσεις.
 - **Posts:** Περιέχει όλα τα views που αφορούν την διαχείριση δημοσιεύσεων. Είναι διαθέσιμα μόνο στους διαχειριστές περιεχομένου.
 - **Create:** Προβάλει μια φόρμα δημιουργίας δημοσίευσης.
 - **Edit:** Προβάλει μια φόρμα επεξεργασίας δημοσίευσης.
 - **Index:** Προβάλει σε λίστα όλες τις αναρτημένες δημοσιεύσεις του χρήστη.
 - **Shared:** Περιέχει όλα τα απαραίτητα Partial Views. Ενδεικτικά,
 - **_Layout:** Ορίζει τη βασική δομή των σελίδων.
 - **_LoginPartial:** Εμφανίζει κουμπιά με δυνατότητες Login, Logout και Register.
 - **Roles:** Βοηθητική κλάση που παρέχει τα ονόματα των ρόλων, για την αποφυγή τυπογραφικού λάθους.
 - **SeedData:** Δημιουργεί απαραίτητα δεδομένα στη βάση.

Το ιστορικό των αλλαγών μπορείτε να το δείτε μέσω του repository στο Github (<https://github.com/geomimo/Moviet/>).



4.2.2. Αναφορά ελέγχου για την 1^η εκτελέσιμη έκδοση

Η εν λόγω εκτελέσιμη έκδοση περιέχει τον βασικό σκελετό της εφαρμογής. Μέχρι αυτό το σημείο έχουν υλοποιηθεί όλες οι λειτουργίες που αναφέρονται στο διάγραμμα περιπτώσεων χρήσης (Διάγραμμα 4). Λείπουν, όμως, ακόμα μερικά χαρακτηριστικά που υλοποιούνται αργότερα. Για παράδειγμα, στην έκδοση αυτή, η αξιολόγηση μιας ταινίας γίνεται μόνο από τον Content Manager, κατά τη δημοσίευσή της και οι Raters μπορούν μόνο να δουν τις αναρτημένες ταινίες, αλλά όχι να τις βαθμολογήσουν. Οι βελτιώσεις που πρέπει να γίνουν και τα χαρακτηριστικά που πρέπει να προστεθούν είναι αρκετά μέχρι την τελική εφαρμογή.

5. Φάση: Κατασκευή (Construction)

Στην παρούσα φάση, το implementation είναι η βασική διαδικασία των επαναλήψεων. Θα μπορούσε να πει κανείς πως υπήρξε αρκετό implementation και στην προηγούμενη φάση, αλλά προσπαθήσαμε να χτίσουμε μια γερή βάση, ώστε να είμαστε πιο ευέλικτη στην φάση της κατασκευής.

5.1. Ανάλυση και Σχεδιασμός

Τα διαγράμματα που παρουσιάζονται παρακάτω περιέχουν αρκετές λεπτομέρειες σε βάθος. Θέλαμε κάποιος που θα δει τα διαγράμματα, να καταλάβει πως είναι η υλοποίηση. Σε αυτόν τον σχεδιασμό, προστίθενται διάφορα χαρακτηριστικά που λάβαμε υπόψη στην πρώτη διαδικασία σύλληψης απαιτήσεων αλλά δεν υλοποιήθηκαν στη φάση Μελέτης, αφού δεν αποτελούσαν το σκελετό της εφαρμογής. Ορισμένες λειτουργίες είναι:

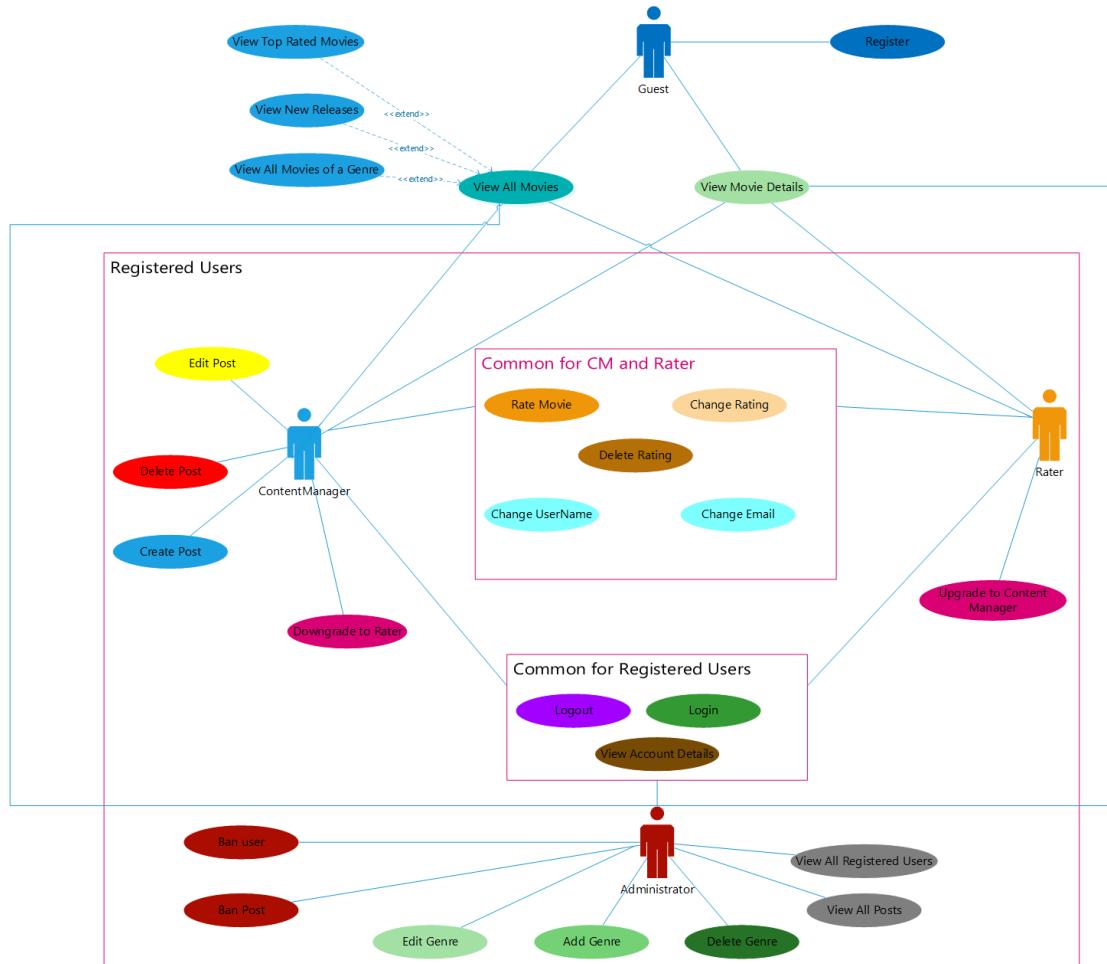
- Αξιολόγηση ταινίας από τους εγγεγραμμένους χρήστες.
- Διαγραφή Αξιολόγησης.
- Άλλαγή Αξιολόγησης.
- Προβολή όλων των αξιολογήσεων ενός χρήστη.
- Προβολή ταινιών βάσει βαθμολογίας
- Προβολή ταινιών βάσει χρόνου δημιουργίας της ανάρτησης.
- Προβολή ταινιών βάσει είδους
- Αποβολή δημοσίευσης
- Διαγραφή δημοσίευσης
- Προβολή λεπτομερειών δημοσίευσης

Επιπλέον, υπάρχουσες λειτουργίες διορθώθηκαν αλλά και βελτιώθηκαν.



5.1.1. Διαγράμματα Περιπτώσεων χρήσης (Τρίτη Έκδοση)

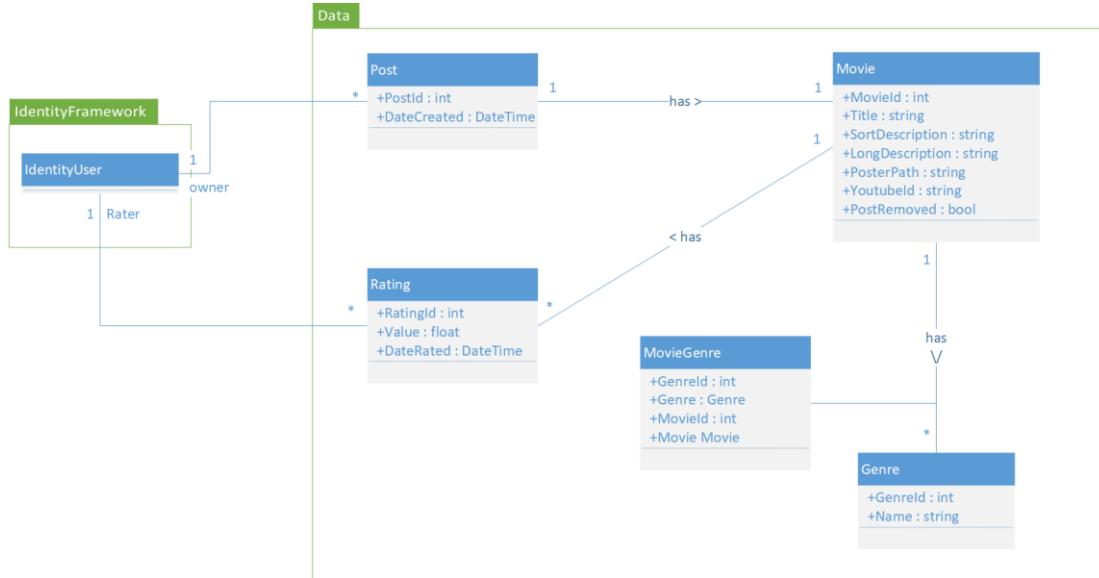
Το βελτιωμένο διάγραμμα περιπτώσεων χρήσης που προκύπτει είναι το εξής



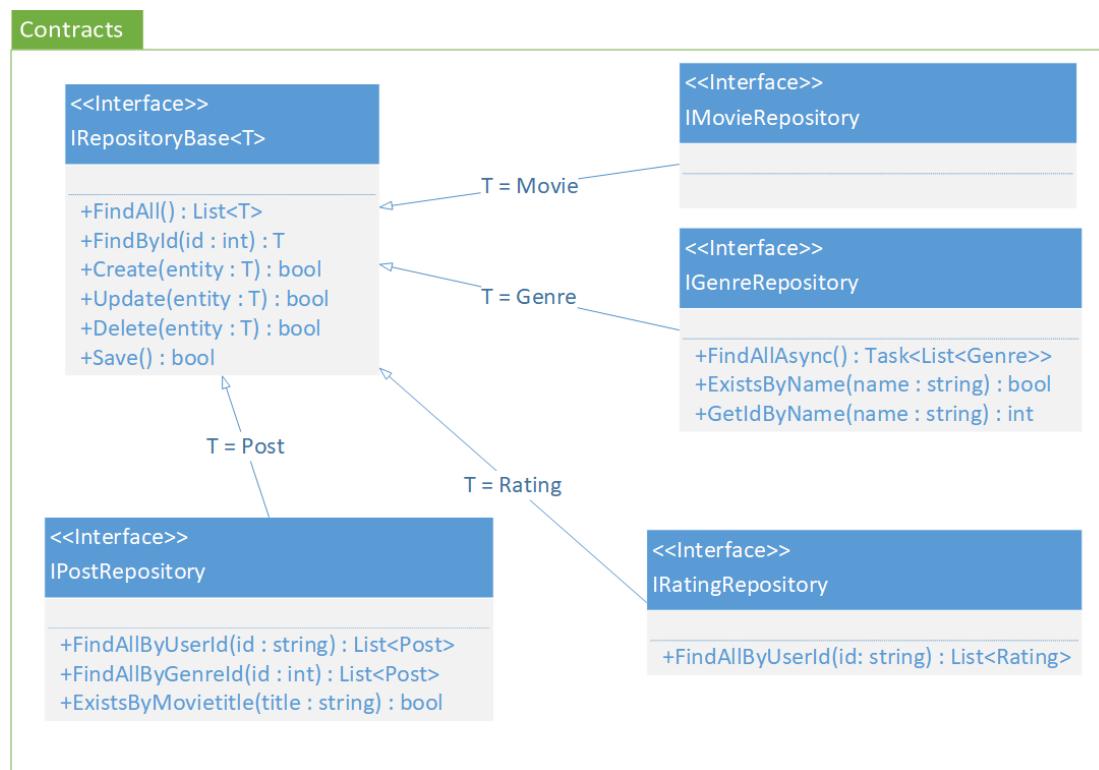
Διάγραμμα 26 Use Case Diagram v3



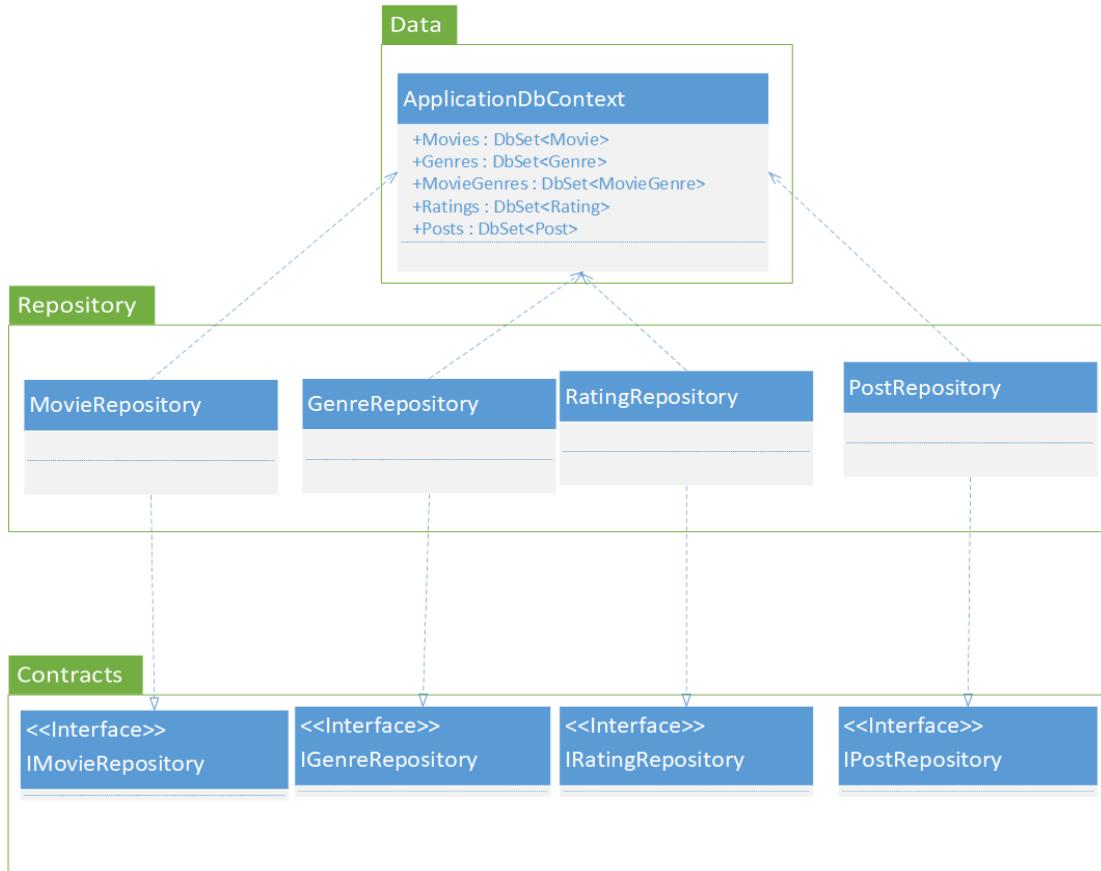
5.1.2. Διαγράμματα Τάξεων (Τρίτη Έκδοση)



Διάγραμμα 27 Data Class Diagram v3

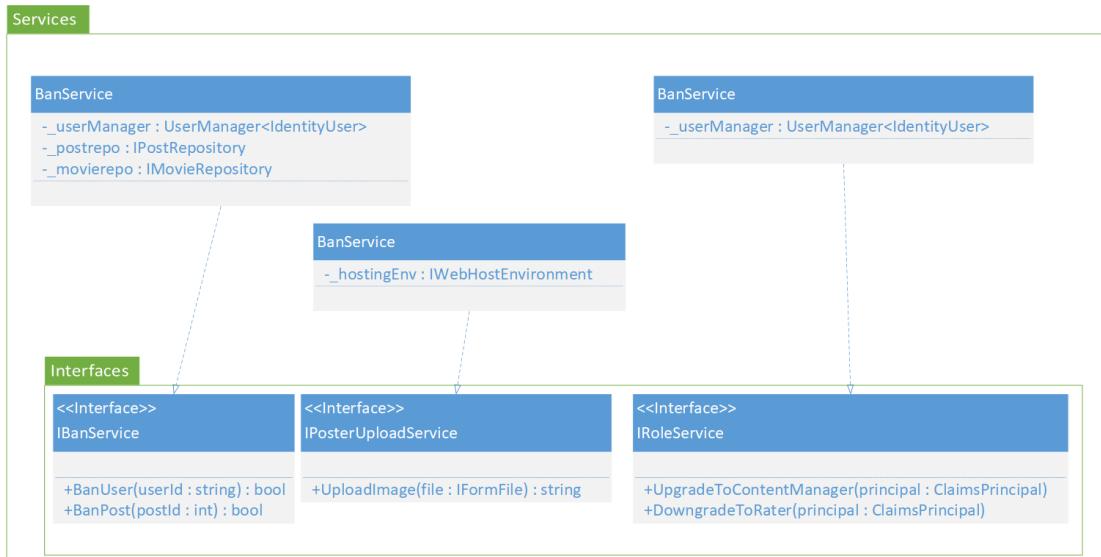


Διάγραμμα 28 Contracts Class Diagram v3

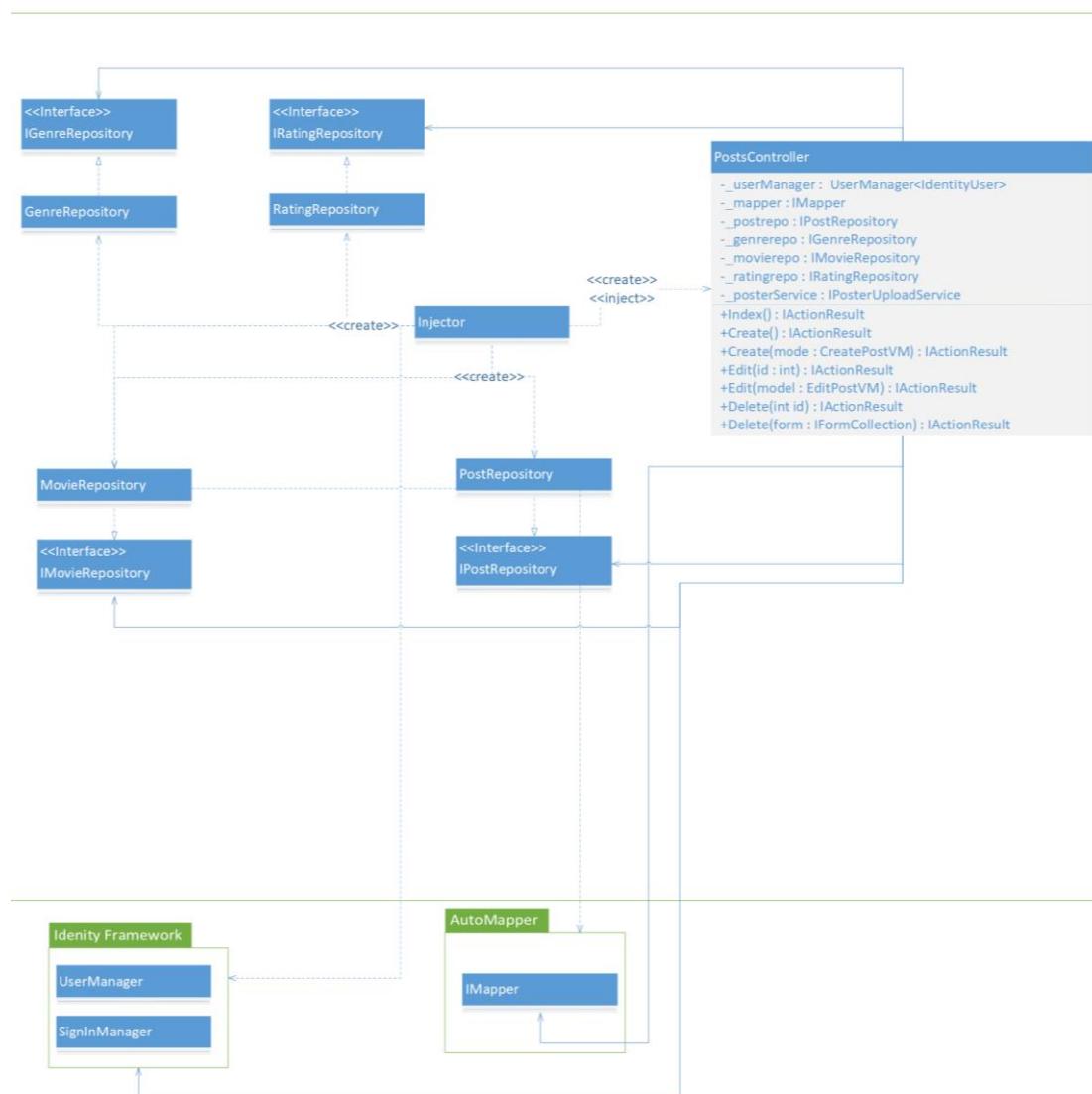


Διάγραμμα 29 Repository Class Diagram

Τα 2 παραπάνω διαγράμματα αποτελούν το Repository Pattern.



Διάγραμμα 30 Services Class Diagram v3



Διάγραμμα 31 Controllers Class Diagram v3 (1)

Το παραπάνω διάγραμμα εμφανίζει τις εξαρτήσεις ενός controller. Τον injector διαχειρίζεται το framework, οπότε είναι άγνωστες οι πραγματικές διαδικασίες. Όμοια συμβαίνει και στα υπόλοιπα controllers, αλλά και σε οποιαδήποτε κλάση γίνεται χρήση Dependency Injection.



Controllers

AccountController

```
-_userManager : UserManager<IdentityUser>
-_signInManager : SignInManager<IdentityUser>
-_roleService : IRoleService
-_mapper : IMapper
+Index() : IActionResult
+ChangeUserName(model : IdentityUser) : IActionResult
+ChangeEmail(model : IdentityUser) : IActionResult
+UpgradeRole() : Task<ActionResult>
+DowngradeRole() : Task<ActionResult>
```

RatingsController

```
-_userManager : UserManager<IdentityUser>
-_mapper : IMapper
-_postrepo : IPostRepository
-_ratingrepo : IRatingRepository
+Index() : IActionResult
+Rate(form : IFormCollection) : IActionResult
+Edit(id : int) : IActionResult
+Edit(model : RatingVM) : IActionResult
+Delete(int id) : IActionResult
```

GenresController

```
-_genrerepo : IGenreRepository
-_mapper : IMapper
+Index() : IActionResult
+Create() : IActionResult
+Create(model : GenreVM) : IActionResult
+Edit(id : int) : IActionResult
+Edit(model : GenreVM) : IActionResult
+Delete(id : int) : IActionResult
```

HomeController

```
-_postrepo : IPostRepository
-_mapper : IMapper
+Index() : IActionResult
```

MoviesController

```
-_userManager : UserManager<IdentityUser>
-_mapper : IMapper
-_postrepo : IPostRepository
-_genrerepo : IGenreRepository
-_movierepo : IMovieRepository
-_ratingrepo : IRatingRepository
+Index() : IActionResult
+TopRated() : IActionResult
+NewReleases() : IActionResult
+ByGenre(id : int) : IActionResult
+Details(id : int) : IActionResult
+DetailsByMovie(int id) : IActionResult
```

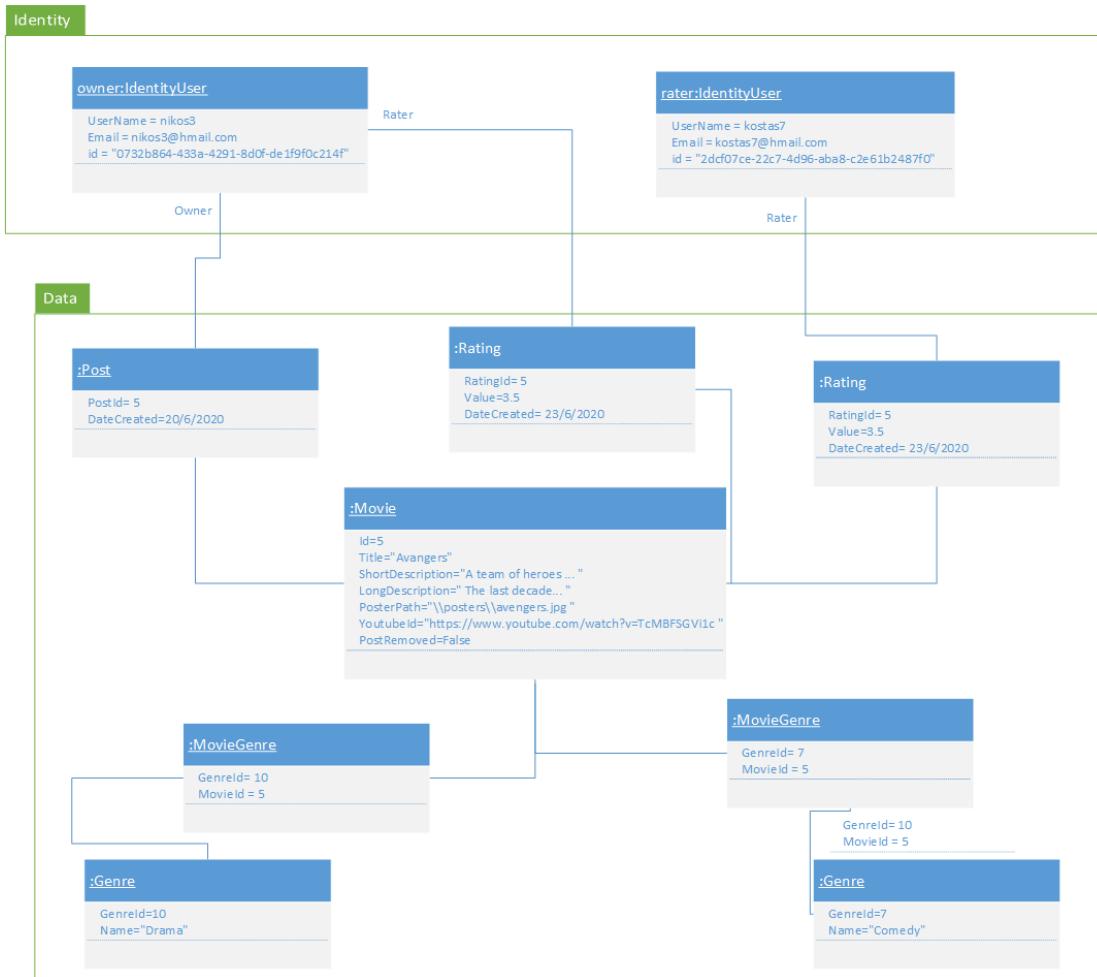
ListController

```
-_userManager : UserManager<IdentityUser>
-_mapper : IMapper
-_banService : IBanService
-_postrepo : IPostRepository
+AllUsers() : IActionResult
+AllPosts() : IActionResult
+BanUser(id : string) : IActionResult
+BanPost(id : int) : IActionResult
```

Διάγραμμα 32 Controllers Class Diagram v3 (2)

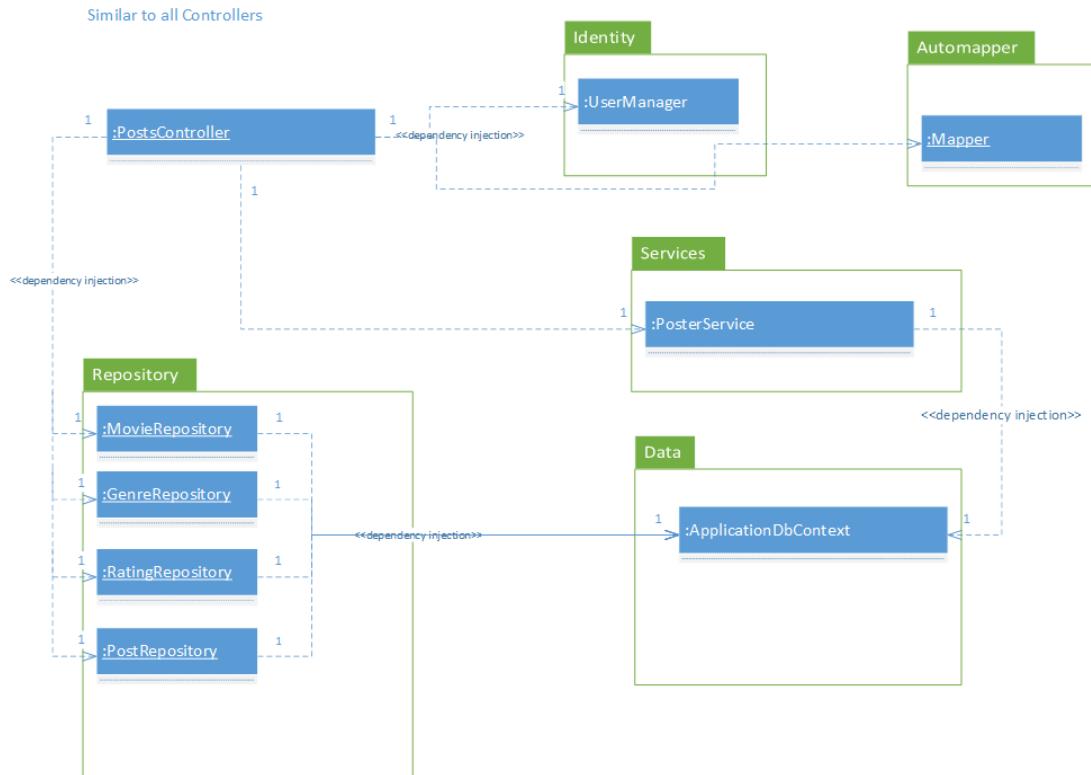


5.1.3. Διαγράμματα Αντικειμένων (Δεύτερη Έκδοση)



Διάγραμμα 33 Data Object Diagram v2

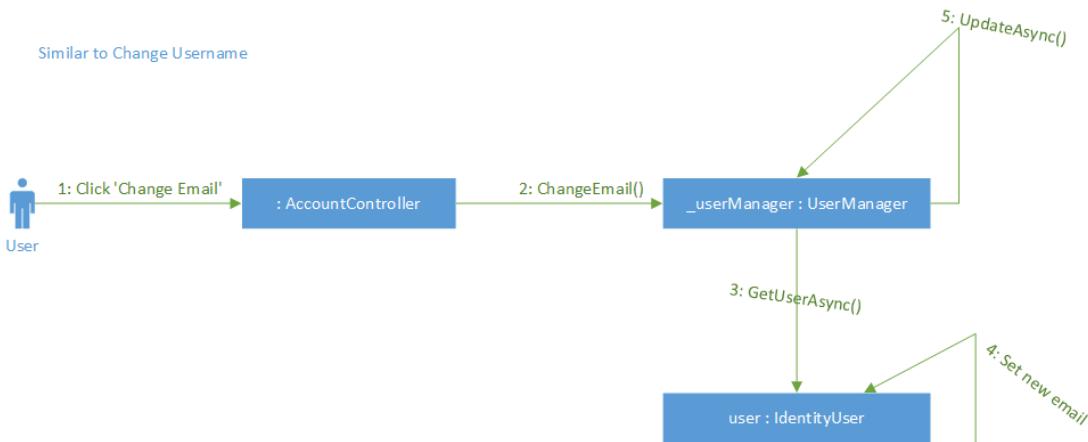
Στο διάγραμμα 33 φαίνεται πως πλέον ένας χρήστης διάφορος του post owner μπορεί να αξιολογήσει την ταινία. Υπενθυμίζουμε πως αυτή η λειτουργία δεν ήταν δυνατή στην προηγούμενη εκτελέσιμη έκδοση (βλ. Διάγραμμα 12). Να σημειωθεί πως στο συγκεκριμένο διάγραμμα δεν απεικονίζεται το Dependency Injection που συμβαίνει (βλ. Διάγραμμα 34).



Διάγραμμα 34 Controllers Object Diagram v2

Στο παραπάνω διάγραμμα φαίνεται το Dependency Injection που συμβαίνει κατά την εκτέλεση του προγράμματος. Όμοια συμβαίνει σε κάθε controller αλλά και σε οποιοδήποτε άλλο αντικείμενο στο οποίο γίνεται Dependency Injection.

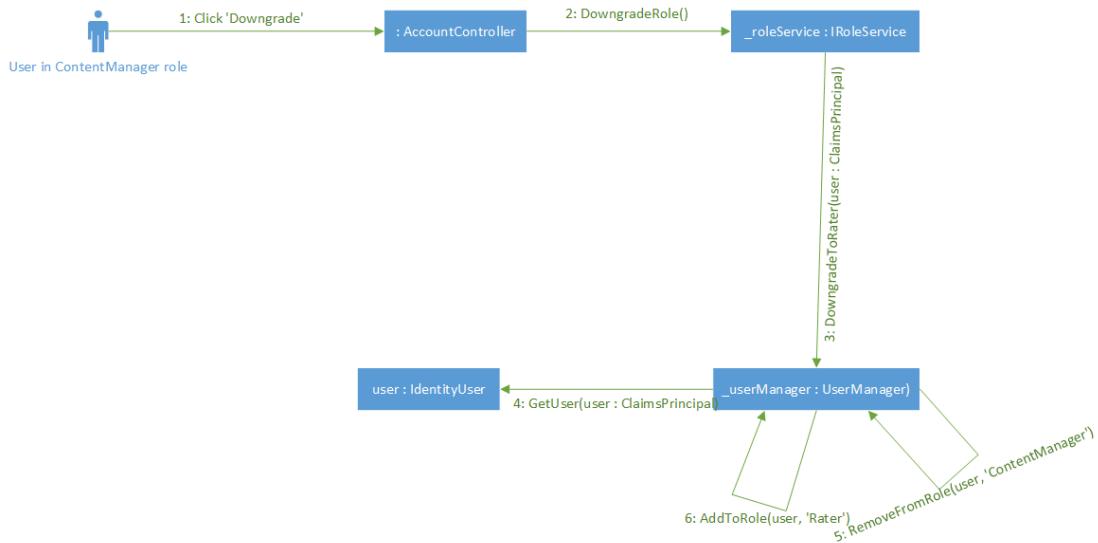
5.1.4. Διαγράμματα Συνεργασίας (Δεύτερη Έκδοση)



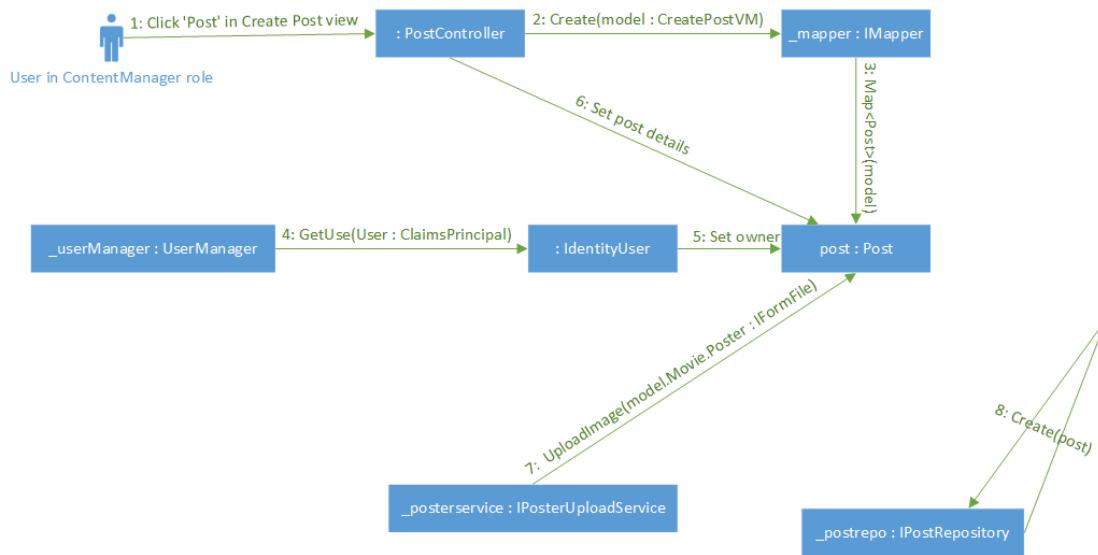
Διάγραμμα 35 Change Email Collaboration Diagram v2



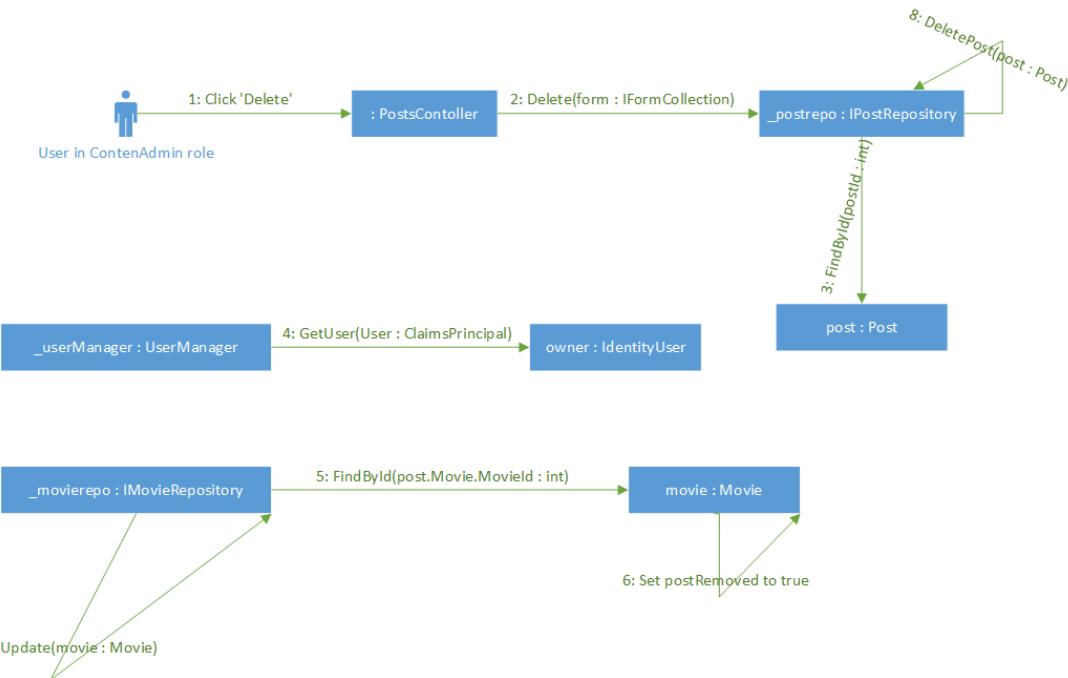
Similar to upgrade role



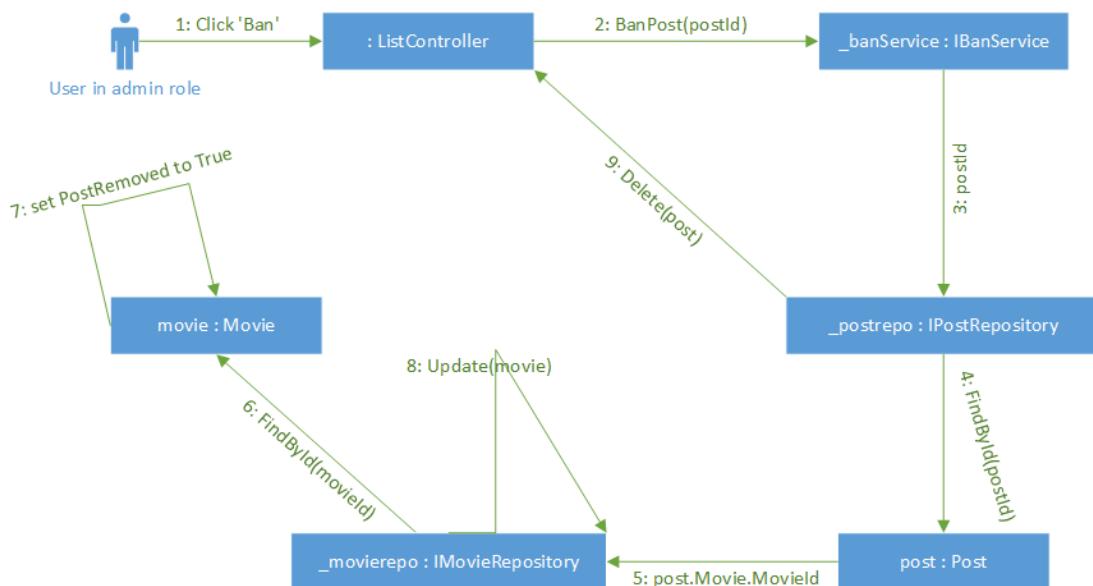
Διάγραμμα 36 Downgrade to Rater Collaboration Diagram v2



Διάγραμμα 37 Create Post Collaboration Diagram v2



Διάγραμμα 38 Delete Post Collaboration Diagram v2

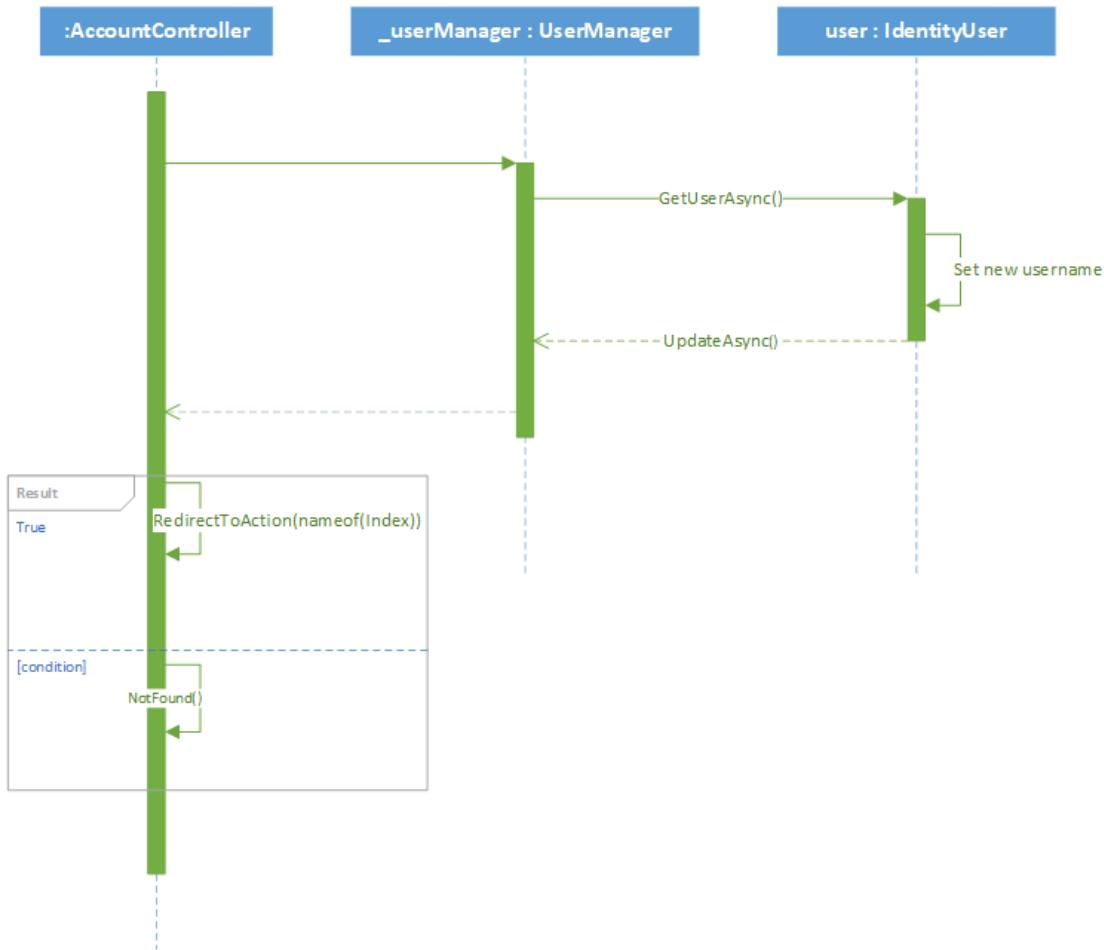


Διάγραμμα 39 Ban Post Collaboration Diagram v2

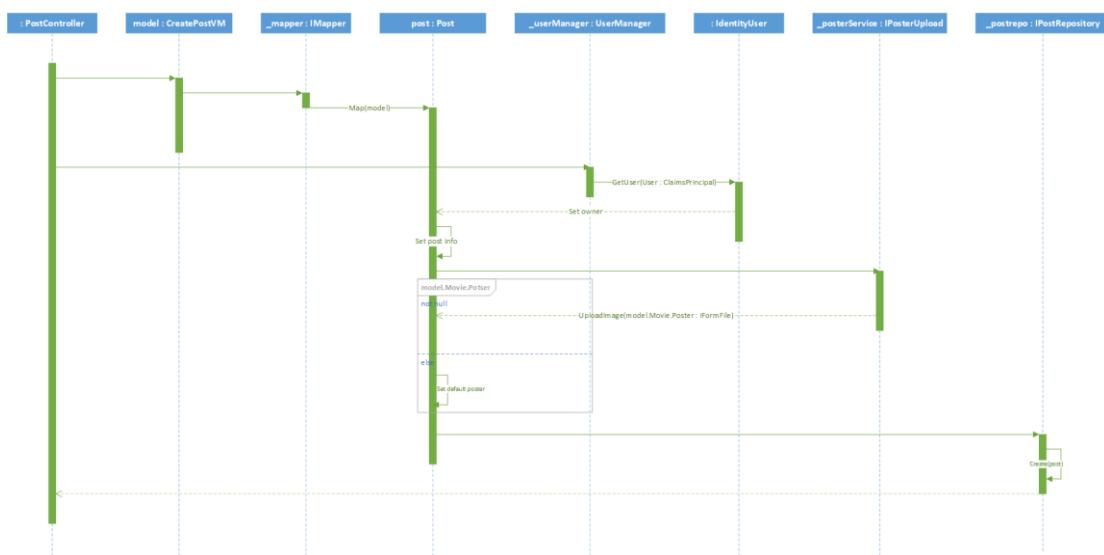


5.1.5. Διαγράμματα Σειράς (Δεύτερη Έκδοση)

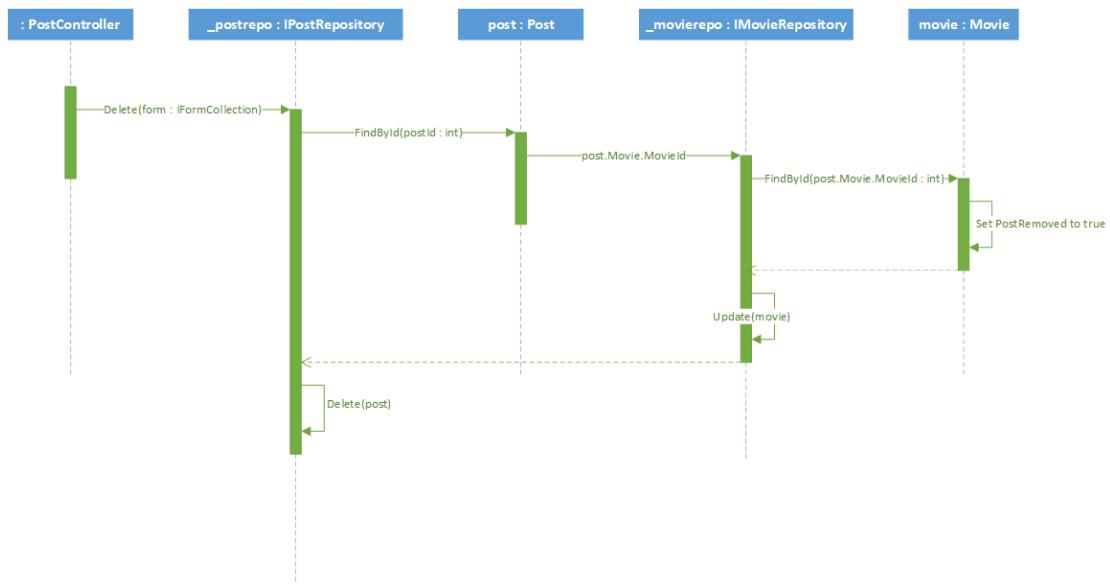
Similar to Change Email



Διάγραμμα 40 Change Username Sequence Diagram v2



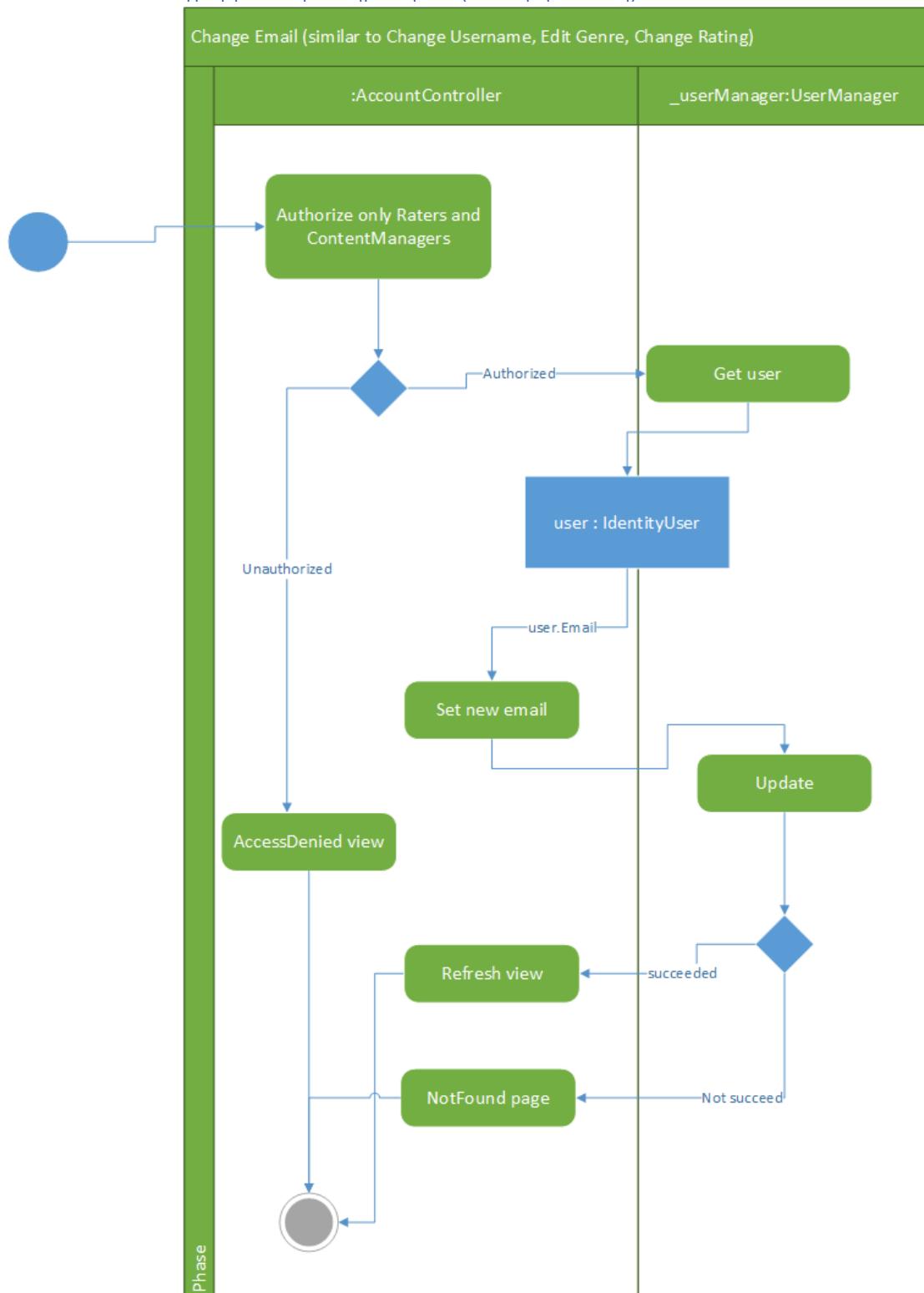
Διάγραμμα 41 Create Post Sequence Diagram v2



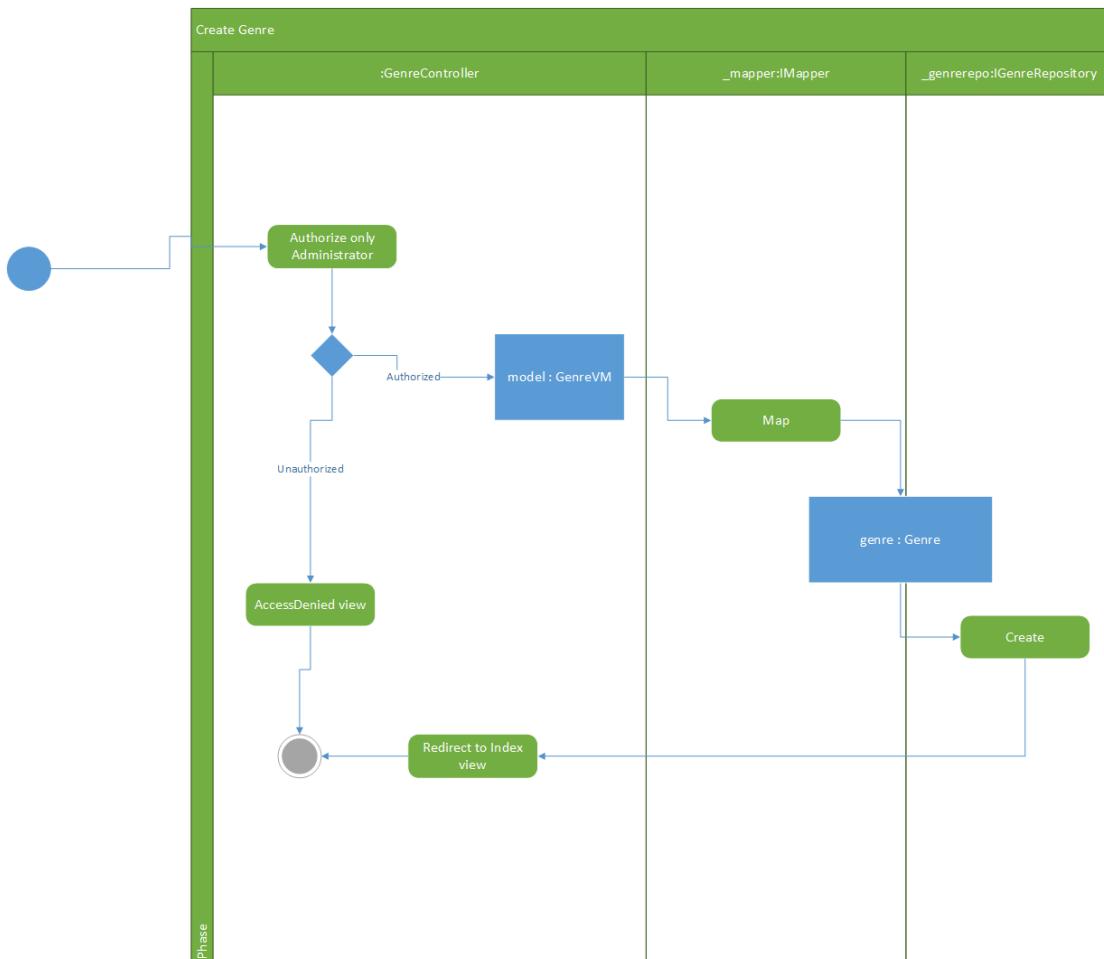
Διάγραμμα 42 Delete Post Sequence Diagram v2



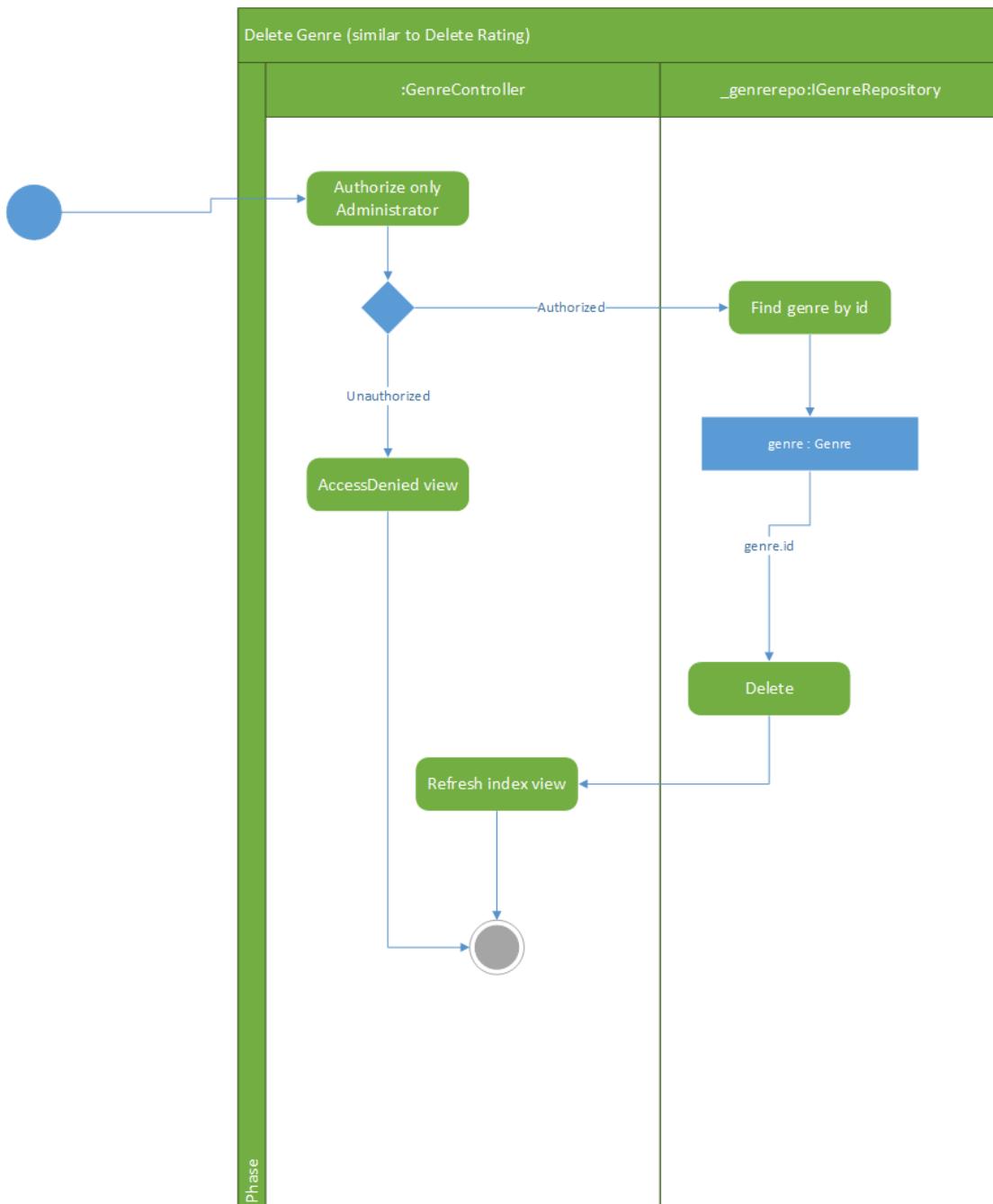
5.1.6. Διαγράμματα Δραστηριοτήτων (Δεύτερη Έκδοση)



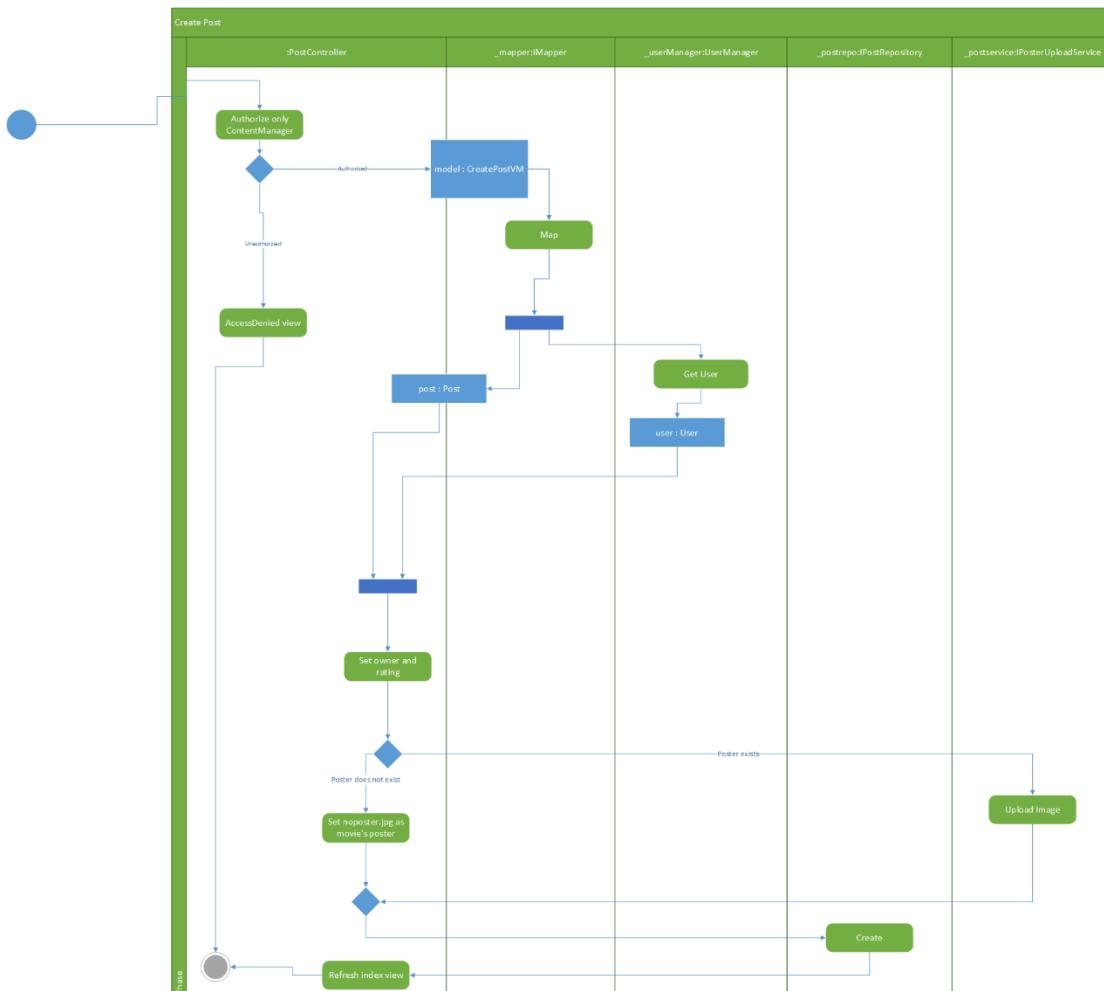
Διάγραμμα 43 Change Email Activity Diagram v2



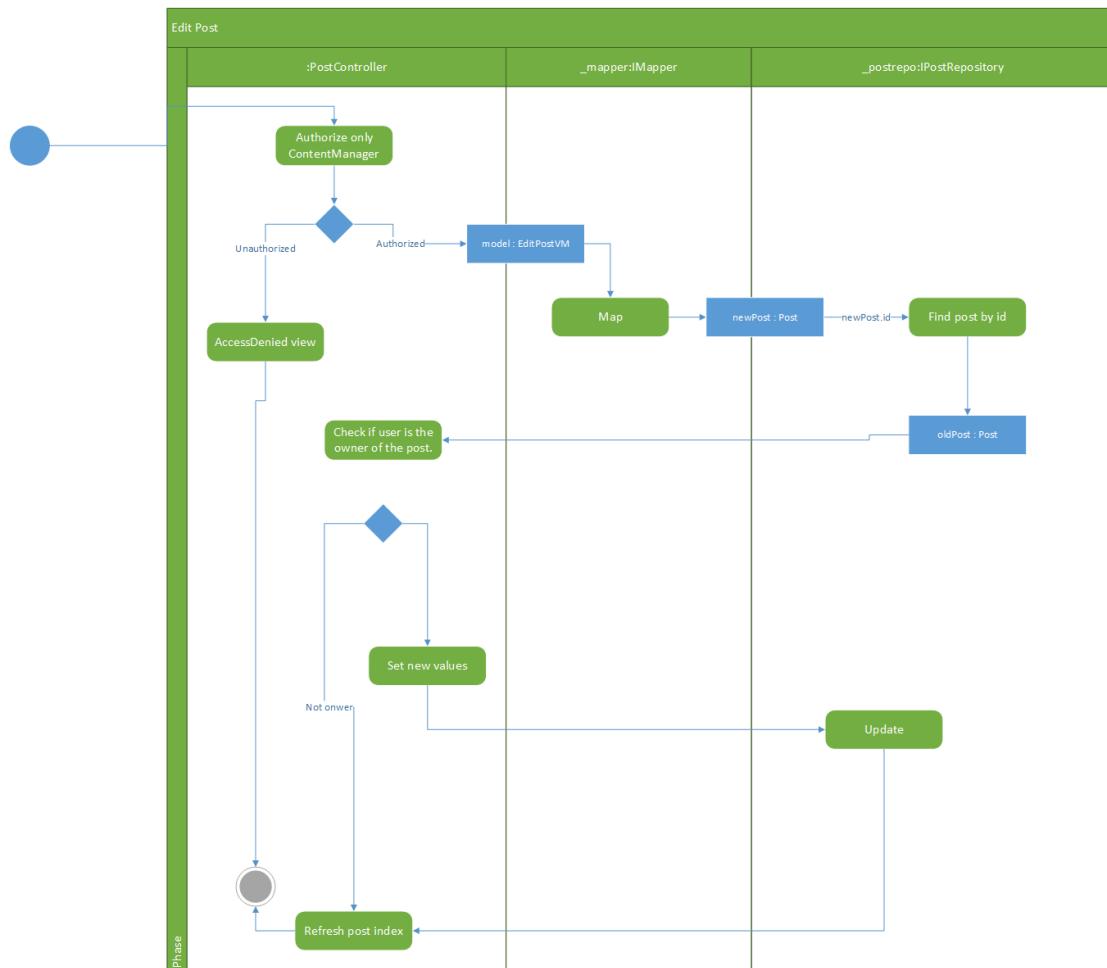
Διάγραμμα 44 Create Genre Activity Diagram v2



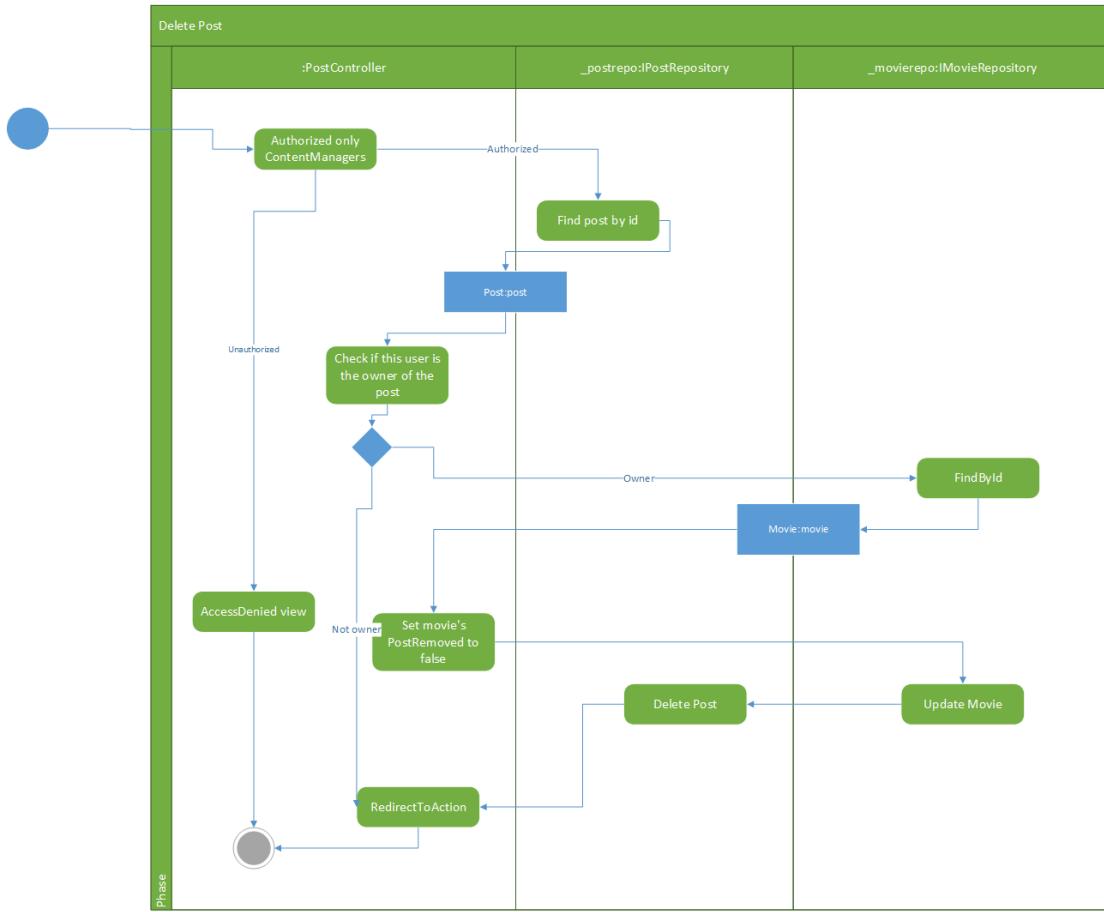
Διάγραμμα 45 Delete Genre Activity Diagram v2



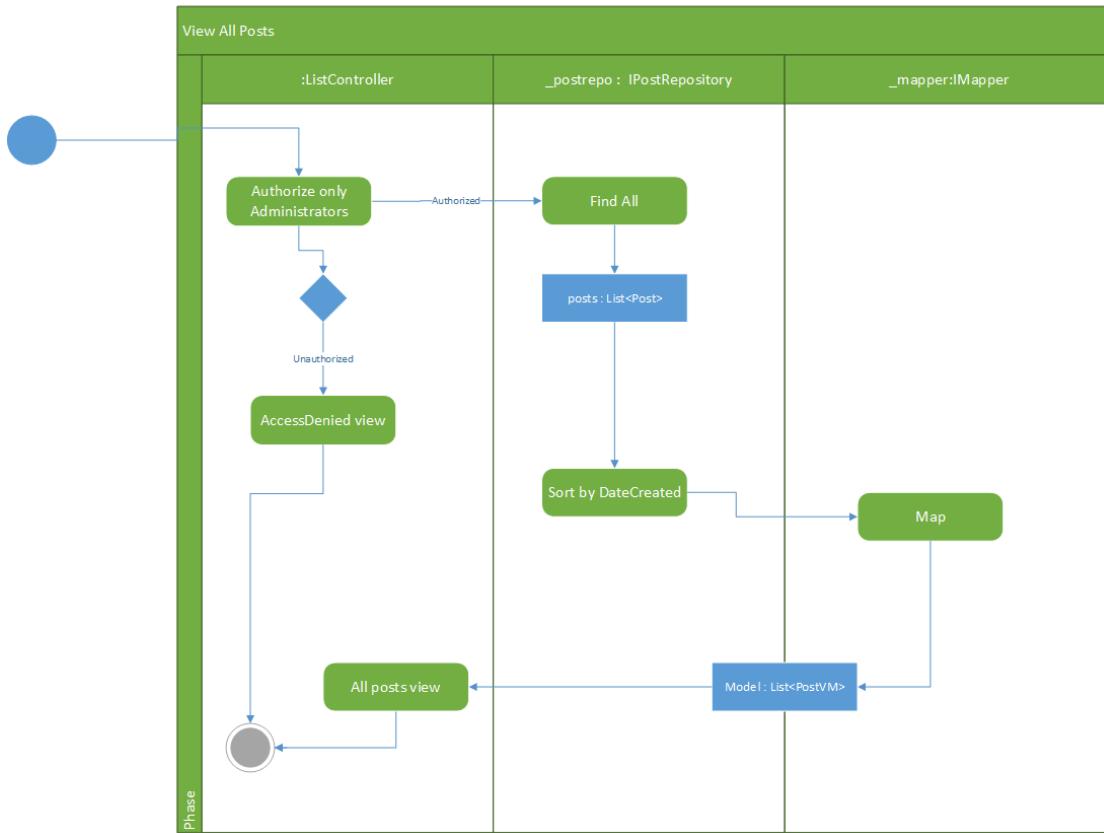
Διάγραμμα 46 Create Post Activity Diagram v2



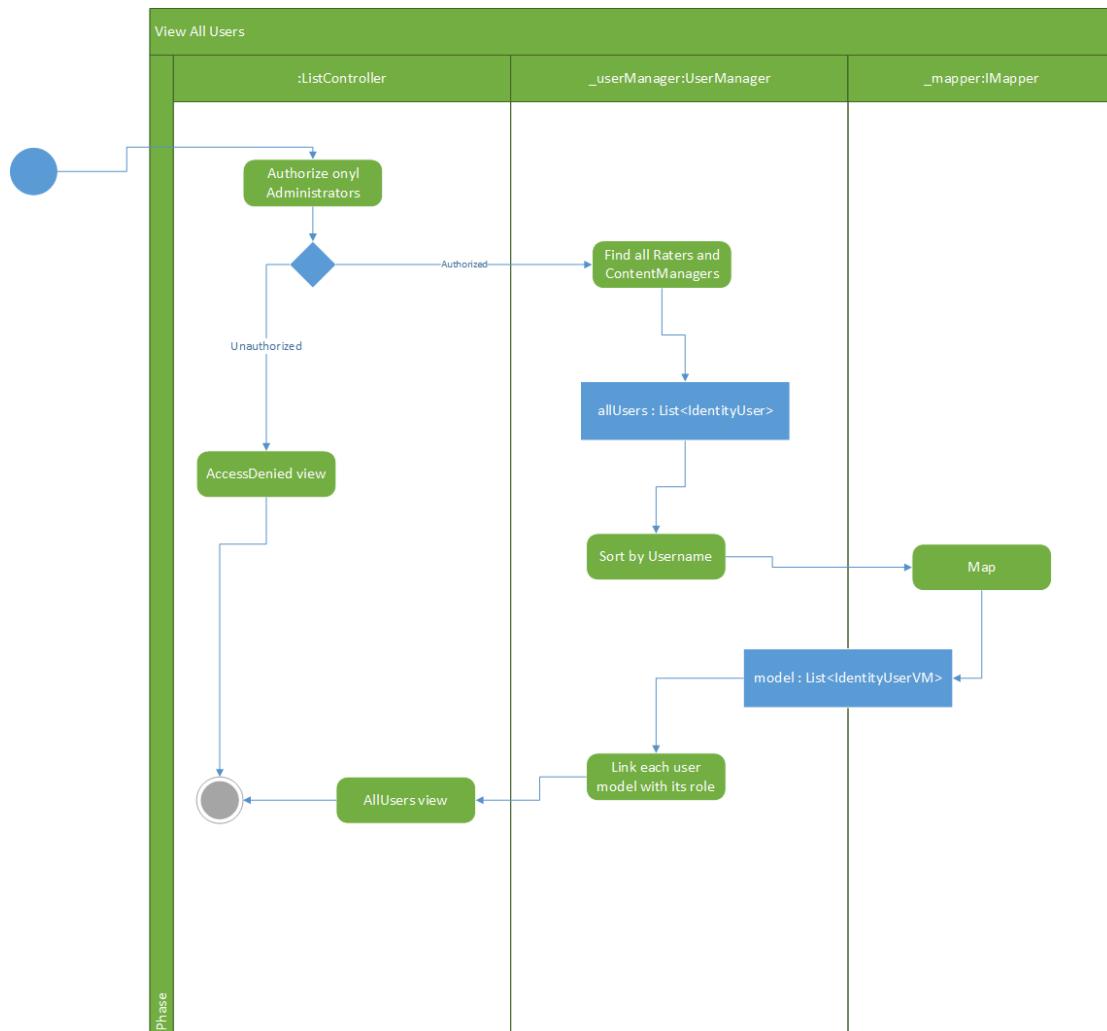
Διάγραμμα 47 Edit Post Activity Diagram v2



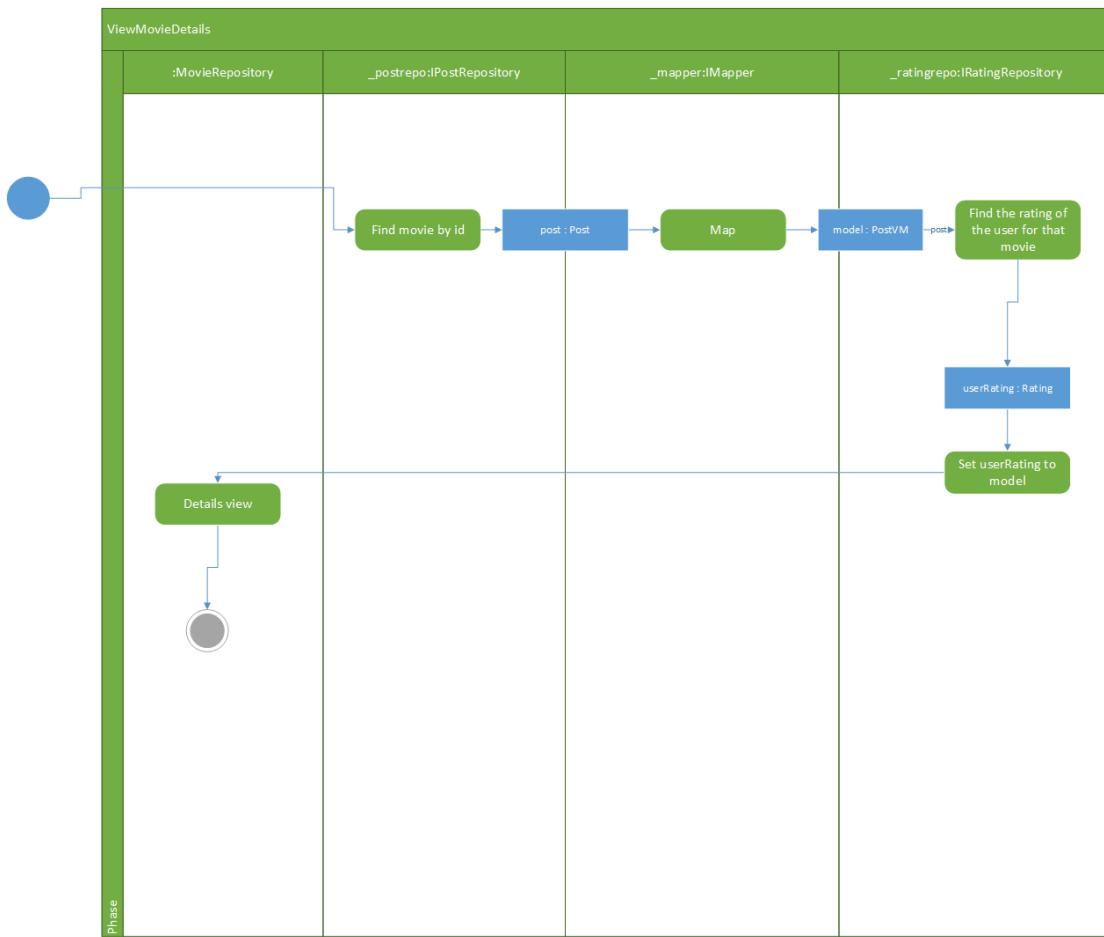
Διάγραμμα 48 Delete Post Activity Diagram v2



Διάγραμμα 49 View All Posts Activity Diagram v2



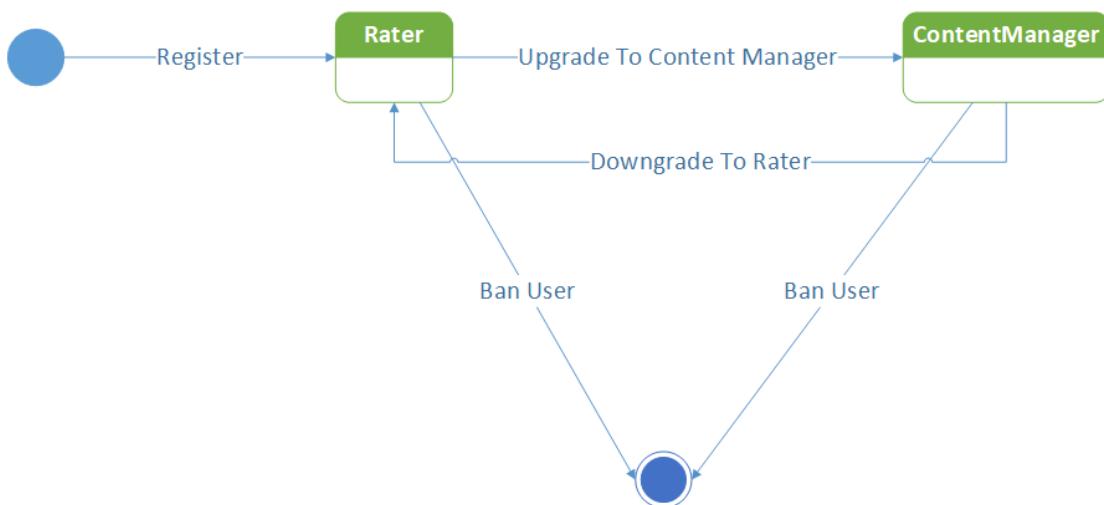
Διάγραμμα 50 View All Users Activity Diagram v2



Διάγραμμα 51 View Movie Details Activity Diagram v2

5.1.7. Διαγράμματα Καταστάσεων (Δεύτερη Έκδοση)

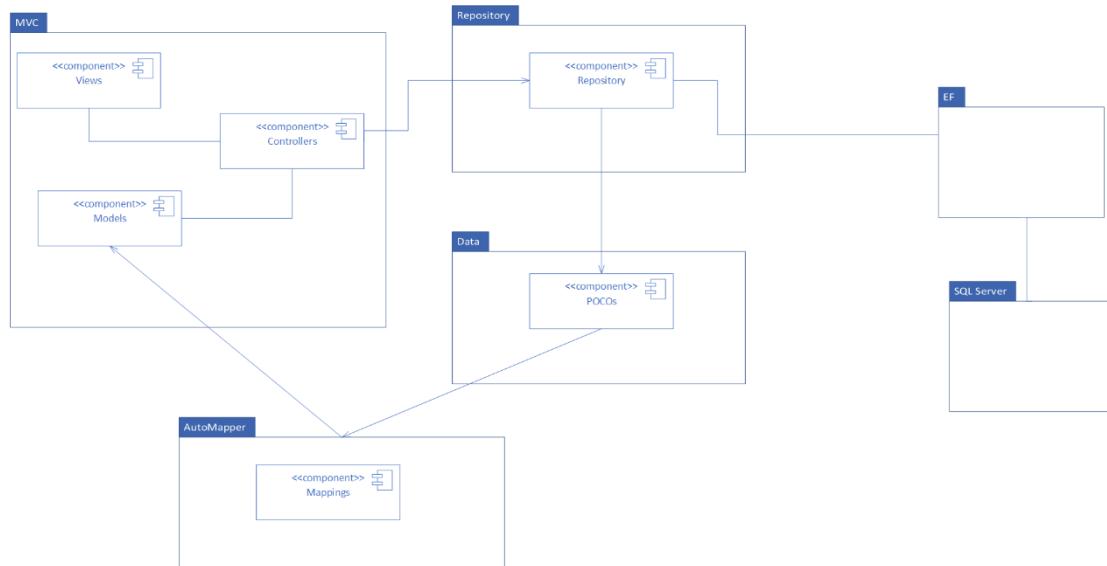
Το διάγραμμα καταστάσεων του χρήστη παραμένει ίδιο. Επιπλέον, δεν εντοπίστηκε άλλη μηχανή καταστάσεων.



Διάγραμμα 52 User State Machine Diagram v2

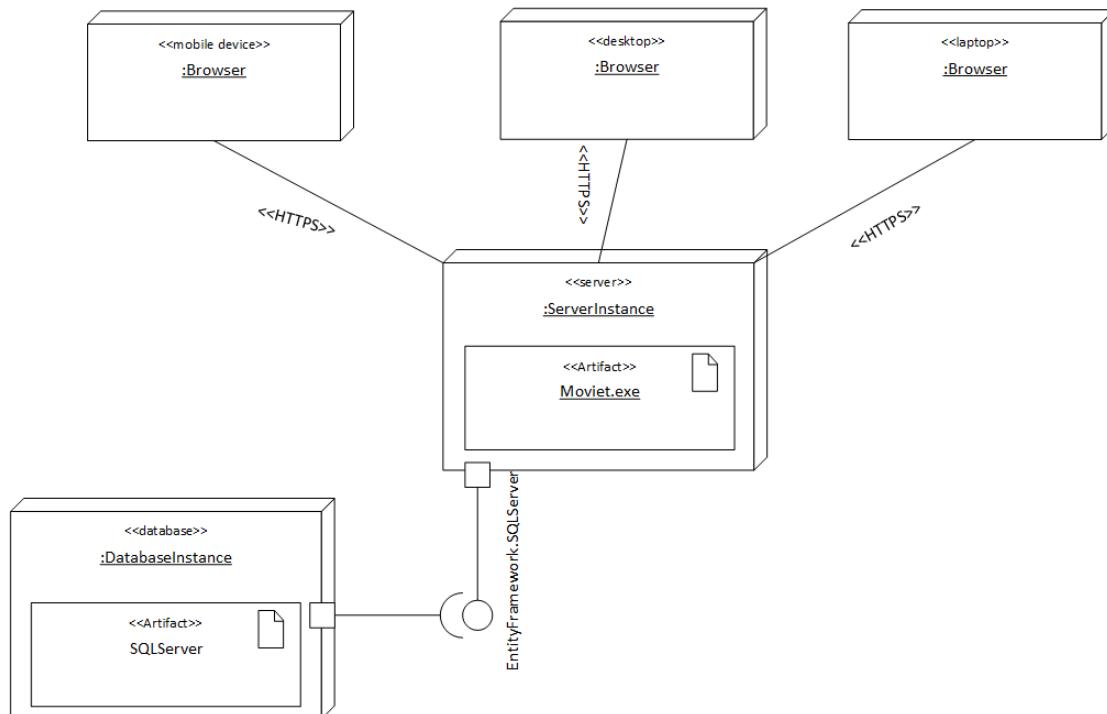


5.1.8. Διαγράμματα Εξαρτημάτων (Δεύτερη Έκδοση) Το διάγραμμα εξαρτημάτων παραμένει ίδιο.



Διάγραμμα 53 Component Diagram v2

5.1.9. Διαγράμματα Διανομής (Δεύτερη Έκδοση) Το διάγραμμα διανομής παραμένει ίδιο.





5.2. Υλοποίηση και Έλεγχος

5.2.1. Υλοποίηση: Τελική εκτελέσιμη έκδοση

Σε αυτό το στάδιο, έχουν υλοποιηθεί σχεδόν όλες οι αρχικές περιπτώσεις χρήσεις. Σε σχέση με την πρώτη έκδοση, προστέθηκαν λειτουργίες αξιολόγησης ταινίας, αποβολής ανάρτησης και άλλα όπως αναφέρθηκαν στην εισαγωγή του κεφαλαίου 5. Επίσης, διορθώθηκαν διάφορα bugs και βελτιώθηκαν οι διαδικασίες.

Αναλυτικότερα η δομή και οι λειτουργίες:

- **Areas:** Ο φάκελος Areas περιέχει ομάδες αρχείων κάτω από μια συγκεκριμένη λειτουργική έννοια. Στην προκειμένη περίπτωση, περιέχει τον φάκελο Identity, στον οποίο βρίσκονται όλα τα απαραίτητα razor pages για τις λειτουργίες login, logout και register, καθώς ακόμα άλλες λειτουργίες που προσφέρει το ASP.NET Core Identity, οι οποίες όμως δεν χρησιμοποιήθηκαν.
- **Contracts:** Ο φάκελος Contracts περιέχει interfaces απαραίτητα για το Repository Pattern.
- **Controllers:** Ο φάκελος Controllers περιέχει όλα τα controllers που χρησιμοποιούνται.
 - **NavBarGenreViewComponent:** Ένα view component, υπεύθυνο για την προβολή των διαθέσιμων ειδών στην μπάρα πλοήγησης.
 - **AccountController:** Υπεύθυνο για τις αιτήσεις που αφορούν το λογαριασμό του χρήστη. Σε αυτή την υλοποίηση, περιέχει actions για την αναβάθμιση του χρήστη από Rater σε Content Manager και το αντίστροφο.
 - **GenresController:** Υπεύθυνο για τις αιτήσεις που αφορούν την διαχείριση των διαθέσιμων ειδών που υπάρχουν στην εφαρμογή. Παρέχει actions για την δημιουργία, την επεξεργασία και τη διαγραφή ενός είδους. Μπορεί να χρησιμοποιηθεί μόνο από χρήστες που βρίσκονται σε ρόλο Administrator.
 - **HomeController:** Υπεύθυνο για τις αιτήσεις που αφορούν τη προβολή της αρχικής σελίδας.
 - **ListController:** Υπεύθυνο για τις αιτήσεις που αφορούν την προβολή των εγγεγραμμένων χρηστών και την αποβολή τους. Μπορεί να χρησιμοποιηθεί μόνο από χρήστες που βρίσκονται σε ρόλο Administrator.
 - **MoviesController:** Υπεύθυνο για τις αιτήσεις που αφορούν την προβολή των λεπτομερειών μιας ταινίας-δημοσίευσης.
 - **PostsController:** Υπεύθυνο για τις αιτήσεις που αφορούν τη δημιουργία και την επεξεργασία δημοσιεύσεων. Μπορεί να χρησιμοποιηθεί μόνο από χρήστες που βρίσκονται σε ρόλο Content Manager.
 - **RatingsController:** Υπεύθυνο για τις αιτήσεις που αφορούν την προβολή, την δημιουργία, την επεξεργασία και τη διαγραφή αξιολογήσεων από τον χρήστη. Μπορεί να χρησιμοποιηθεί μόνο από χρήστες που βρίσκονται σε ρόλο Content Manager ή Rater.
- **Data:** Ο φάκελος Data περιέχει POCO κλάσεις, καθώς και το context απαραίτητο για την επικοινωνία μετα βάση.
 - **ApplicationDbContext:** Προσφέρει πρόσβαση στα δεδομένα της βάσης. Είναι ο ενδιάμεσος κρίκος μεταξύ της εφαρμογής και του Entity Framework Core.



- **Admin:** Περιγράφει τον διαχειριστή. Τελικά δεν υλοποιήθηκε διότι έγινε χρήση ρόλων.
- **ContentManager:** Περιγράφει τον διαχειριστή περιεχομένου. Τελικά δεν υλοποιήθηκε διότι έγινε χρήση ρόλων.
- **Genres:** Περιγράφει ένας είδος ταινίας.
- **Movie:** Περιγράφει μια ταινία.
- **MovieGenre:** Περιγράφει τη σύνδεση μιας ταινίας με ένα είδος ταινίας.
- **Post:** Περιγράφει μια δημοσίευση ενός χρήστη.
- **Rater:** Περιγράφει τον βαθμολογητή. Τελικά δεν υλοποιήθηκε διότι έγινε χρήση ρόλων.
- **Rating:** Περιγράφει μια αξιολόγηση μια ταινίας από ένα χρήστη.
- **Helpers:** Ο φάκελος Helpers περιέχει βιοθητικές κλάσεις που λειτουργούν ως Tag Helpers στα views.
 - **CustomHelper:** Προσφέρει ανάδραση στην αλλαγή των σελίδων της εφαρμογής.
- **Mappings:** Ο φάκελος Mappings περιέχει κλάσεις για την μετατροπή των POCO σε View Models με τη βοήθεια του πακέτου AutoMapper.
 - **Maps:** Εκεί βρίσκονται οι δηλώσεις μετατροπής.
 - **MovieGenreResolver:** Περιέχει κλάσεις για την κατάλληλη μετατροπή των MovieGenres.
 - **RatingResolver:** Περιέχει κλάσεις για την κατάλληλη μετατροπή των Ratings.
 - **TotalRatingResolver:** Περιέχει κλάση για την κατάλληλη μετατροπή και τον υπολογισμό της μέσης βαθμολογίας μιας ταινίας.
 - **YoutubeIdResolver:** Περιέχει κλάσεις για την κατάλληλη μετατροπή των Youtube urls σε embedded μορφή.
- **Models:** Ο φάκελος Models περιέχει τα απαραίτητα μοντέλα που βοηθούν στην παρουσίαση των POCO στον χρήστη. Τα αρχεία τελειώνουν σε VM που σημαίνει ViewModel. Αυτό, ώστε να είναι πιο κατανοητή η χρήση τους.
 - **GenreVM:** Κατάλληλο για την παρουσίαση των ειδών των ταινιών.
 - **IdentityUserVM:** Κατάλληλο για την παρουσίαση των στοιχείων των εγγεγραμμένων χρηστών.
 - **MovieGenreVM:** Κατάλληλο για τη σύνδεση ενός μοντέλου ταινίας με ένα μοντέλο είδους.
 - **MovieVM:** Κατάλληλο για την παρουσίαση μιας ταινίας.
 - **PostVM:** Κατάλληλο για την παρουσίαση μιας δημοσίευσης.
 - **RatingVM:** Κατάλληλο για την παρουσίαση των βαθμολογιών.
- **Repository:** Ο φάκελος Repository περιέχει απαραίτητες κλάσεις για την πρόσβαση των δεδομένων της βάσης. (Repository Pattern)
 - **GenreRepository:** Παρέχει πρόσβαση σε δεδομένα των διαθέσιμων ειδών.
 - **MovieRepository:** Παρέχει πρόσβαση σε δεδομένα των καταχωρημένων ταινιών.
 - **PostRepository:** Παρέχει πρόσβαση σε δεδομένα των δημοσιεύσεων.
 - **RatingRepository:** Παρέχει πρόσβαση σε δεδομένα των αξιολογήσεων.
- **Services:** Ο φάκελος Services περιέχει διεπαφές και κλάσεις υλοποίησης που αφορούν ορισμένες λειτουργίες business logic. Η δημιουργία των διεπαφών δεν



ήταν απαραίτητη, όμως ήταν ένα καλό best practice πάνω σε έννοιες και αρχές του αντικειμενοστρεφής προγραμματισμού.

- **Interfaces:** Ο φάκελος Interfaces περιέχει τις εν λόγω διεπαφές.
 - **IBanService:** Υπεύθυνο για την αποβολή ενός χρήστη από την εφαρμογή.
 - **IPosterUploadService:** Υπεύθυνο για την αποθήκευση των αφισών στην εφαρμογή.
 - **IRoleService:** Υπεύθυνο για την αναβάθμιση και υποβάθμιση ενός χρήστη.
- **BanService:** Η υλοποίηση του interface IBanService.
- **PosterUploadService:** Η υλοποίηση του interface IPosterUploadService.
- **RoleService:** Η υλοποίηση του interface IRoleService.
- **Views:** Ο φάκελος Views περιέχει όλα τα views της εφαρμογής, χρήσιμα για την παραγωγή των τελικά εμφανιζόμενων ιστοσελίδων.
 - **Account:** Περιέχει όλα τα views που αφορούν το λογαριασμό του χρήστη.
 - **Index:** Προβάλει τις πληροφορίες του χρήστη και του προσφέρει δυνατότητες αλλαγής email/username και αναβάθμισης/υποβάθμισης.
 - **Genres:** Περιέχει όλα τα views που αφορούν τα διαθέσιμα είδη ταινιών. Είναι διαθέσιμα μόνο σε διαχειριστές.
 - **Create:** Προβάλει μια φόρμα για την δημιουργία ενός διαθέσιμου είδους.
 - **Index:** Προβάλει μια λίστα με όλα τα διαθέσιμα είδη ταινιών.
 - **Rename:** Προβάλει μια φόρμα επεξεργασίας του ονόματος ενός είδους.
 - **Home:** Περιέχει όλα τα views που αφορούν την αρχική σελίδα της εφαρμογής.
 - **Index:** Προβάλει πληροφορίες για την εφαρμογή, καθώς και τέσσερεις από τις πιο πρόσφατες αναρτήσεις και πιο υψηλά βαθμολογημένες.
 - **List:** Περιέχει όλα τα views που αφορούν την προβολή διαφόρων πληροφοριών στον διαχειριστή.
 - **AllUsers:** Προβάλει σε λίστα όλους τους εγγεγραμμένους χρήστες.
 - **AllPosts:** Προβάλει σε λίστα όλες τις δημοσιεύσεις.
 - **Movies:** Περιέχει όλα τα views που αφορούν την προβολή ταινιών-δημοσιεύσεων.
 - **Details:** Προβάλει πληροφορίες μιας ταινίας-δημοσίευσης.
 - **Index:** Προβάλει όλες τις διαθέσιμες ταινίες-δημοσιεύσεις.
 - **Posts:** Περιέχει όλα τα views που αφορούν την διαχείριση δημοσιεύσεων. Είναι διαθέσιμα μόνο στους διαχειριστές περιεχομένου.
 - **Create:** Προβάλει μια φόρμα δημιουργίας δημοσίευσης.
 - **Delete:** Προβάλει μια σελίδα επιβεβαίωσης για την διαγραφή μιας δημοσίευσης.
 - **Edit:** Προβάλει μια φόρμα επεξεργασίας δημοσίευσης.
 - **Index:** Προβάλει σε λίστα όλες τις αναρτημένες δημοσιεύσεις του χρήστη.



- **Ratings:** Περιέχει όλα τα views που αφορούν την διαχείριση των αξιολογήσεων ενός χρήστη. Είναι διαθέσιμα μόνο σε διαχειριστές περιεχομένου και βαθμολογητές.
 - **Edit:** Προβάλει μια φόρμα επεξεργασίας της αξιολόγησης.
 - **Index:** Προβάλει σε λίστα όλες τις αξιολογήσεις του χρήστη.
- **Shared:** Περιέχει όλα τα απαραίτητα Partial Views. Ενδεικτικά,
 - **_Layout:** Ορίζει τη βασική δομή των σελίδων.
 - **_LoginPartial:** Εμφανίζει κουμπιά με δυνατότητες Login, Logout και Register.
- **Roles:** Βοηθητική κλάση που παρέχει τα ονόματα των ρόλων, για την αποφυγή τυπογραφικού λάθους.
- **SeedData:** Δημιουργεί απαραίτητα δεδομένα στη βάση.

Το ιστορικό των αλλαγών μπορείτε να το δείτε μέσω του repository στο Github (<https://github.com/geomimo/Moviet/>).

5.2.2. Αναφορά ελέγχου για την τελική έκδοση

Η εφαρμογή είναι επεκτάσιμη και πολλά ακόμα χαρακτηριστικά μπορούν να προστεθούν. Έχουν υλοποιηθεί τα περισσότερα use cases που είχαν δηλωθεί αρχικά. Ορισμένα όμως δεν έφτασαν μέχρι την τελική εκτελέσιμη έκδοση λόγω περιορισμένης προγραμματιστικής γνώσης. Για παράδειγμα δεν υλοποιήθηκε η μπάρα αναζήτησης με auto complete. Προτιμήσαμε να μην προσθέσουμε κάποιο χαρακτηριστικό το οποίο δεν θα ικανοποιήσουμε κάποιες απαιτήσεις μας ή επρόκειτο να ήταν προβληματικό. Επιπρόσθετα, σε γενικές γραμμές, καλύπτονται και θέματα ασφαλείας και εξουσιοδότησης. Λόγου χάριν, αν ένας επισκέπτης εισάγει το url: <https://localhost:<port>/Posts/Delete/35>, θα εμφανιστεί σελίδας αποτροπής, αφού, πρώτον, μόνο οι Content Managers έχουν εξουσιοδότηση στο controller Posts και δεύτερον, δεν ανήκει το post με id = 35 στον χρήστη αυτόν. Σίγουρα υπάρχουν κενά ασφαλείας, αλλά έχουμε καλύψει όσα περισσότερα μπορούμε. Επιπλέον, να αναφερθεί πως η εφαρμογή δεν είναι responsive, δηλαδή οι αλλαγές στο μέγεθος της οθόνης ή χρήση σε μικρές συσκευές, έχουν αρνητικό αντίκτυπο στην εμφάνιση των σελίδων.

6. Εγχειρίδιο χρήστη

6.1. Σύντομη παρουσίαση του προγράμματος

To Moviet δίνει τη δυνατότητα στους χρήστες να κοινοποιούν αγαπημένες τους ταινίες, προβάλλοντας πληροφορίες για αυτές, καθώς επίσης και να τις αξιολογούν. Χωρίζει τους χρήστες σε 4 κατηγορίες, με διαφορετικές δυνατότητες η κάθε μια.

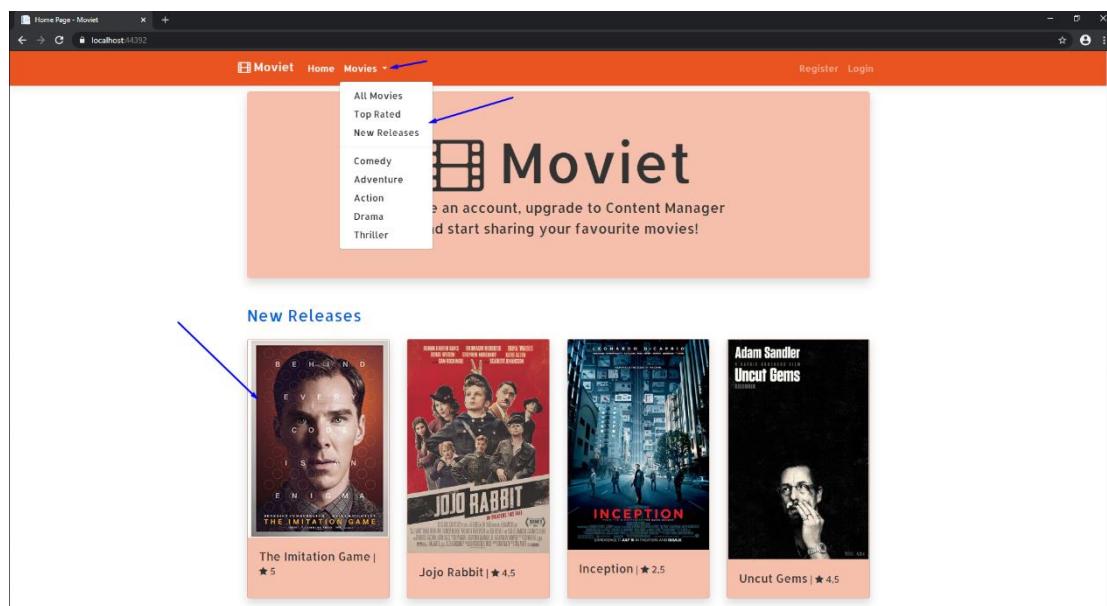
- **Guests:** Οι Guests είναι οι χρήστες που επισκέπτονται την ιστοσελίδα του Moviet και δεν έχουν κάποιο ενεργό λογαριασμό. Μπορούν να δουν τις αναρτημένες ταινίες, με τις πληροφορίες τους και τη μέση βαθμολογία τους.
- **Raters:** Οι Raters είναι χρήστες που έχουν ενεργό λογαριασμό στην πλατφόρμα. Έχουν όλες τι δυνατότητες των Guests και επιπλέον μπορούν να βαθμολογήσουν κάποια ταινία.



- **Content Manager:** Οι Content Managers είναι χρήστες που έχουν ενεργό λογαριασμό στην πλατφόρμα και έχουν επεκτείνει τη «συνδρομή» τους. Ουσιαστικά, κάθε εγγεγραμμένος χρήστης ξεκινά ως Rater και στη συνέχεια μπορεί να κάνει αίτηση ώστε να αναβαθμιστεί ο λογαριασμός του σε Content Manager. Έχει όλες τις δυνατότητες του Rater και επιπλέον μπορεί να αναρτά νέες δημοσιεύσεις-ταινίες.
- **Administrator:** Ο Administrator είναι αποτελεί ένα είδος λογαριασμού δημιουργημένο από την ομάδα του Moviet και έχει ανατεθεί σε άνθρωπο υπεύθυνο για την διαχείριση της κοινότητας. Έχει τις δυνατότητες ενός Guest αλλά με την διαφορά ότι μπορεί να διαγράψει δημοσιεύσεις και χρήστες αν θεωρεί πως δεν ακολουθούν τους όρους κοινότητας. Ο λογαριασμός αυτός δεν μπορεί να διαγραφτεί ή να επεξαργαστεί.

6.2. Παρουσίαση σεναρίων λειτουργίας

Κατά την επίσκεψη στην ιστοσελίδα, ο επισκέπτης βλέπει την αρχική σελίδα όπου προβάλλονται χρήσιμες πληροφορίες, καθώς επίσης και μερικές προτάσεις ταινιών βάσει βαθμολογίας και ημερομηνία ανάρτησης. Ο Επισκέπτης μπορεί να κλικάρει την αφίσα μιας ταινίας ή να περιηγηθεί σε όλες τις ταινίες επιλέγοντας το είδος της εμφάνισης από τη μπάρα περιήγησης.



Εικόνα 7 Home page for guests



Top Rated

Parasite ★ 5 Greed and class discrimination threaten the newly formed symbiotic relationship between the wealthy Park family and the destitute Kim clan.	The Imitation Game ★ 5 During World War II, the English mathematical genius Alan Turing tries to crack the German Enigma code with help from fellow mathematicians.	Joker ★ 4.8 A mentally troubled comedian Arthur Fleck is disregarded and mistreated by society. He embarks on a downward spiral of revolution and bloody crime.	Avengers: Endgame ★ 4.7 After the devastating events of Avengers: Infinity War, the universe is in ruins. The Avengers assemble once more in order to reverse Thanos' actions.
• Comedy • Drama • Thriller	• Drama • Thriller	• Drama	• Adventure • Action • Thriller
Author: content_manager3, 16/06/2020	Author: content_manager3, 16/06/2020	Author: content_manager2, 15/06/2020	Author: content_manager1, 15/06/2020

Εικόνα 8 All movies sorted by rating

Πατώντας πάνω σε μια αφίσα, εμφανίζεται η σελίδα με τις πληροφορίες της ταινίας. Επίσης, εμφανίζεται κατάλληλο μήνυμα ώστε να είναι δυνατή η βαθμολόγηση της ταινίας.

Details

The Imitation Game

★ 5 | Author: content_manager3 | Date: 16/06/2020 | Category: Drama , Thriller

In 1951, two policemen, Nock and Staehl, investigate mathematician Alan Turing following an apparent break-in at his home. During an interrogation by Nock, Turing tells of his time working at Bletchley Park during the Second World War. In 1927, young Turing, unhappy bullied at boarding school, develops a friendship with Christopher Morcom, who sparks his interest in cryptography. Turing develops romantic feelings for him, but Christopher soon dies from tuberculosis. When Britain declares war on Germany in 1939, Turing travels to Bletchley Park. Under the direction of Commander Alastair Denniston, he joins the cryptography team of Hugh Alexander, John Cairncross, Peter Hilton, Keith Furman and Charles Richards, who try to decrypt the Enigma machine that the Nazis use to send coded messages.

Login to rate any movie.

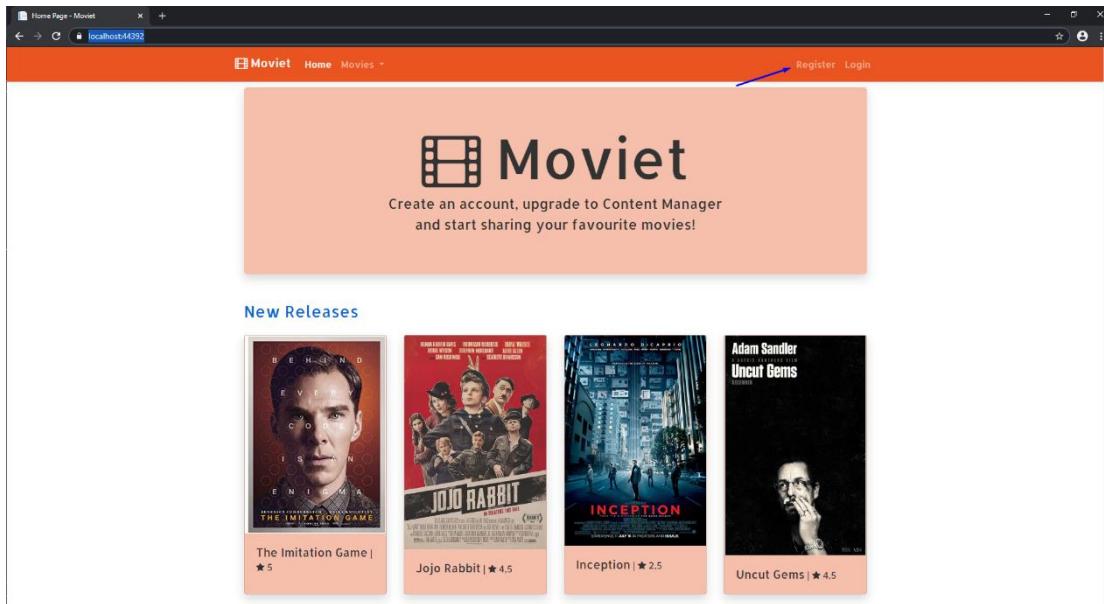
Trailer

The Imitation Game Official Trailer #1 (2014) - Benedict Cumberbatch Movie HD

Εικόνα 9 Movie details for guests

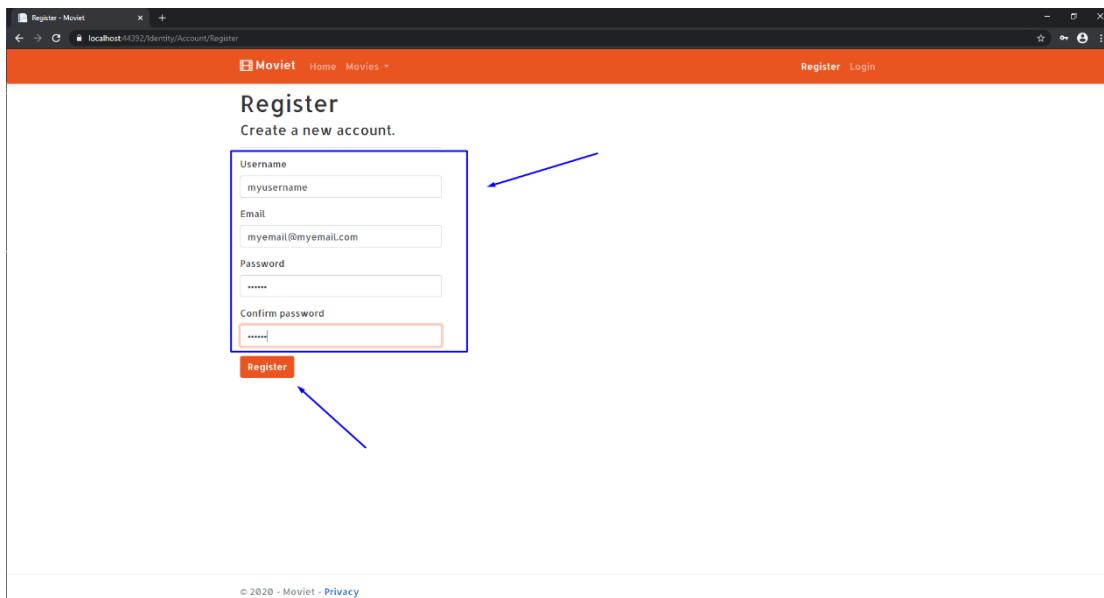


Αν ο χρήστης δεν έχει λογαριασμό, μπορεί να πατήσει το κουμπί “Register” πάνω δεξιά.



Εικόνα 10 Registration (1)

Στη συνέχεια θα εμφανιστεί μια φόρμα εισαγωγής στοιχείων, την οποία συμπληρώνει ο χρήστης και στο τέλος πατάει το κουμπί “Register”.



Εικόνα 11 Registration (2)

Αν ο επισκέπτης έχει ήδη λογαριασμό, τότε μπορεί να συνδεθεί πατώντας το κουμπί “Login” πάνω δεξιά.



The screenshot shows the Moviet website's home page. At the top, there is a navigation bar with links for 'Home' and 'Movies'. On the right side of the bar, there are 'Register' and 'Login' buttons. A blue arrow points from the text 'Εικόνα 12 Login (1)' to the 'Login' button. Below the navigation bar, there is a large orange header area with the Moviet logo and the text 'Create an account, upgrade to Content Manager and start sharing your favourite movies!'. Underneath this, there is a section titled 'New Releases' featuring four movie posters: 'The Imitation Game' (Benedict Cumberbatch), 'Jojo Rabbit' (Taika Waititi), 'Inception' (Leonardo DiCaprio), and 'Uncut Gems' (Adam Sandler). Each poster includes a rating: 'The Imitation Game | ★ 5', 'Jojo Rabbit | ★ 4.5', 'Inception | ★ 2.5', and 'Uncut Gems | ★ 4.5'.

Εικόνα 12 Login (1)

Έπειτα, εισάγει τα στοιχεία του και πατάει "Log in"

The screenshot shows the Moviet website's login page. At the top, there is a navigation bar with links for 'Home' and 'Movies'. On the right side of the bar, there are 'Register' and 'Login' buttons. A blue arrow points from the text 'Εικόνα 13 Login (2)' to the 'Log in' button. The main form has fields for 'Username' (containing 'admin@admin.com') and 'Password' (containing '*****'). There is also a 'Remember me?' checkbox and a 'Log in' button. Below the form, there are links for 'Forgot your password?' and 'Register as a new user'. To the right of the form, there is a table titled 'Users' listing various accounts:

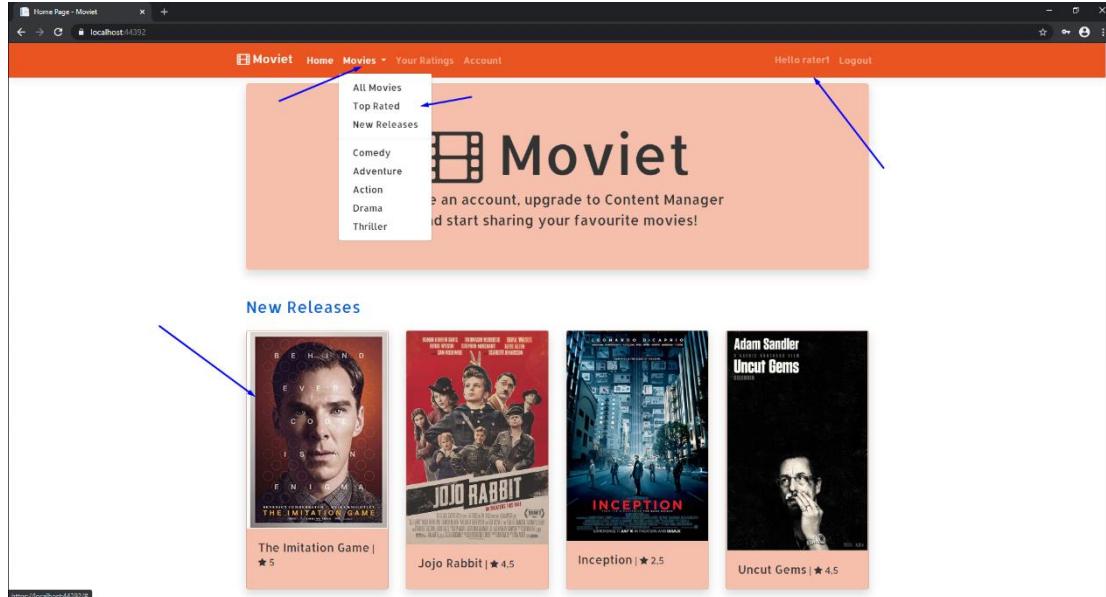
Username	Password
rater1	rater1
rater2	rater2
rater3	rater3
content_manager1	content_manager1
content_manager2	content_manager2
content_manager3	content_manager3
admin@admin.com	admin

At the bottom of the page, there is a copyright notice: '© 2020 - Moviet - Privacy'.

Εικόνα 13 Login (2)

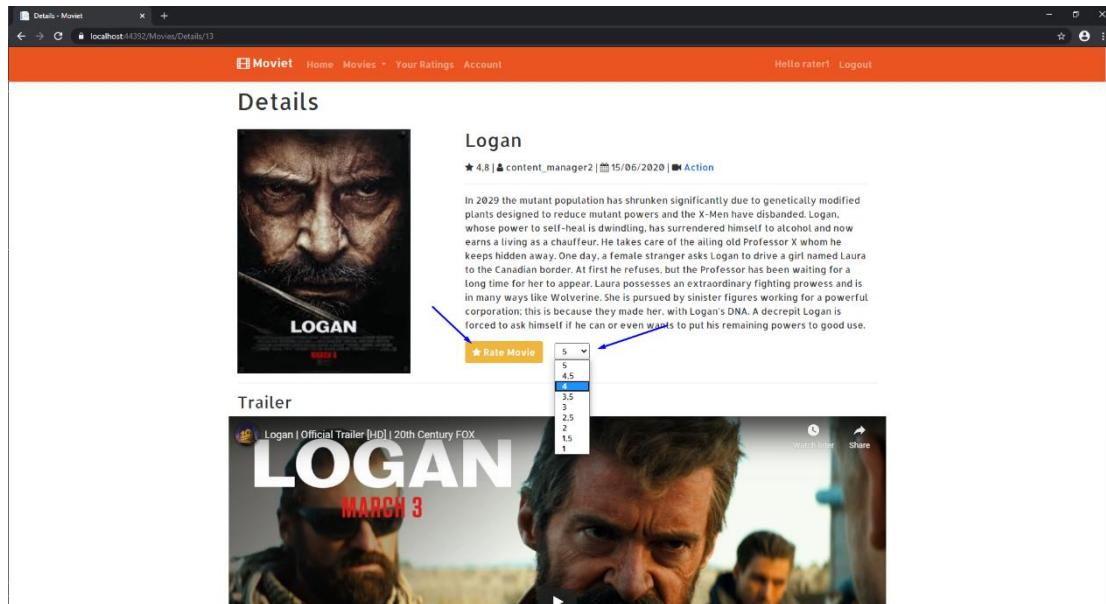


Όταν, πλέον, ο χρήστης συνδεθεί, εμφανίζεται ξανά η αρχική σελίδα. Μπορεί να δει τις πληροφορίες μιας ταινίας με τον τρόπο που παρουσιάστηκε παραπάνω.



Εικόνα 14 Home page for rater

Πλέον, αν επιλέξουμε να δούμε τις πληροφορίες μιας ταινίας, δίνεται η δυνατότητα καταχώρησης βαθμολογίας.



Εικόνα 15 Rate movie

Ο χρήστης μπορεί να επιλέξει την επιθυμητή βαθμολογία και στη συνέχεια να πατήσει το κουμπί "Rate Movie". Η μέση βαθμολογία της ταινίας θα ανανεωθεί. Επιπλέον, αν ο χρήστης έχει ήδη βαθμολογήσει την ταινία, εμφανίζεται διαφορετικό μήνυμα στο κουμπί και η βαθμολογία που έχει ήδη καταχωρήσει. Αν επιθυμεί να αλλάξει την βαθμολογία, μπορεί να το κάνει επιλέγοντας μια νέα βαθμολογία και στη συνέχεια πατήσει το κουμπί "Edit Rating".



Logan

★ 4,5 | content_manager2 | 15/06/2020 | Action

In 2029 the mutant population has shrunk significantly due to genetically modified plants designed to reduce their powers and the X-Men have disbanded. Logan, whose power to see heat is dwindling, has surrendered himself to alcohol and now earns a living as a chauffeur. He takes care of the ailing old Professor X whom he keeps hidden away. One day, a female stranger asks Logan to drive a girl named Laura to the Canadian border. At first he refuses, but the Professor has been waiting for a long time for her to appear. Laura possesses an extraordinary fighting prowess and is in many ways like Wolverine. She is pursued by sinister figures working for a powerful corporation; this is because they made her with Logan's DNA. A deceptile Logan is forced to ask himself if he can or even wants to put his remaining powers to good use.

Trailer

Logan | Official Trailer [HD] | 20th Century FOX

LOGAN
MARCH 3

Εικόνα 16 Edit rating

Επιπλέον, ο χρήστης τύπου Rater έχει τη δυνατότητα να δει σε λίστα όλες τις καταχωρημένες αξιολογήσεις πατώντας το κουμπί "Your Ratings" στην μπάρα περιήγησης. Με κόκκινο background και την ένδειξη "Removed" φαίνονται οι ταινίες των οποίων η δημοσίευση έχει διαγραφεί. Πατώντας το κουμπί "Edit", εμφανίζεται μια φόρμα επεξεργασίας της αξιολόγησης. Πατώντας το κουμπί "Details", εμφανίζεται η σελίδα με τις λεπτομέρειες της ταινίας. Πατώντας το κουμπί "Delete", διαγράφεται η αξιολόγηση

Title	Poster	Value	Date Rated	Action
Baby Driver		3	15/06/2020 00:14:02	Removed
Joker		5	15/06/2020 00:14:52	Edit
1917		4,5	16/06/2020 02:17:54	Edit
The Godfather		4,5	16/06/2020 02:17:54	Edit

Εικόνα 17 Your Ratings list



Εικόνα 18 Edit rating form

Πατώντας το κουμπί “Account” στην μπάρα περιήγησης, εμφανίζεται η σελίδα με τις πληροφορίες του λογαριασμού, όπως ο τύπος (Rater / Content Manager / Administrator), το username και το email. Επιπλέον, υπάρχει δυνατότητα επεξεργασίας των στοιχείων καθώς και κουμπί αναβάθμισης του λογαριασμού σε Content Manager.

Εικόνα 19 Account page (1)

Πατώντας το κουμπί, ο τύπος του λογαριασμού αλλάζει σε Content Manager. Όταν γίνει αυτό, αλλάζει το badge του τύπου του λογαριασμού δίπλα από την επικεφαλίδα της σελίδας και εμφανίζεται κατάλληλο κουμπί ώστε ο χρήστης να ξαναγίνει Rater.



© 2020 - Moviet - Privacy

Εικόνα 20 Account page (2)

Αν ο χρήστης είναι Content Manager, τότε «ξεκλειδώνουν» νέες λειτουργίες για αυτόν. Αν βλέπει τις πληροφορίες μιας δημοσίευσής του, τότε εμφανίζονται δύο κουμπιά δίπλα από τον τίτλο της ταινίας. “Edit” για επεξεργασία και “Delete” για διαγραφή.

Aladdin [Edit](#) [Delete](#)

★ 3.5 | [content_manager1](#) | 15/06/2020 | [Comedy](#) , [Adventure](#)

Aladdin who regularly steals to get by with the aid of his pet monkey, Abu. One day while roaming the streets, Aladdin spots a beautiful girl who gets in trouble after giving away bread to children without paying. Aladdin comes to her rescue, and together they get chased by the Royal Guards. After a while they elude their pursuers, and Aladdin takes the girl to his place for some tea. The girl calls herself Dala, and is the handmaiden to the Princess of Agrabah. She suddenly has to leave as another suitor for the princess, Prince Anders, arrives. Dala happens to be Princess Jasmine and Dala is the name of her handmaid and best friend. Meanwhile, The Sultan's trusted councilor, Jafar, is plotting to overthrow the Sultan by getting his hands on the Magic Lamp.

[Update Rating](#) [3.5](#)

Trailer

Aladdin (2019) Prince Ali - Will Smith Scene

Εικόνα 21 Movie details for Content Manager

Πατώντας το κουμπί “Your Posts” στην μπάρα περιήγησης, ο χρήστης μπορεί να δει όλες τις αναρτήσεις του. Επιπλέον, υπάρχουν κουμπιά με λειτουργίες δημιουργίας, επεξεργασίας, εμφάνισης πληροφοριών και διαγραφής μιας δημοσίευσης.



Date Created	Poster	Title	Genres	Action Buttons
15/06/2020		Aladdin	• Comedy • Adventure	Edit Details Delete
15/06/2020		Avengers: Endgame	• Adventure • Action • Thriller	Edit Details Delete
15/06/2020		Dunkirk	• Adventure	Edit Details Delete
16/06/2020		Inception	• Adventure	Edit Details Delete

Εικόνα 22 Your Posts page

Πατώντας το κουμπί “New Post”, εμφανίζεται μια φόρμα δημιουργίας μιας νέας ανάρτησης. Ο χρήστης συμπληρώνει τα πεδία και πατάει το κουμπί “Post”.

Post
New Post

Title
A Beautiful Mind

Sort Description
in cryptography, his life takes a turn for the nightmarish.

Long Description
eventually triumphed over his tragedy, and finally - late in life - received the Nobel Prize.

Poster
 [Browse](#)

Trailer
<https://www.youtube.com/watch?v=WF>

Genres Drama •

Rating 4 • ★

[Post](#) [Back](#)

Εικόνα 23 New Post form

Έτσι, εμφανίζεται ξανά η λίστα με τις αναρτήσεις του, η οποία περιέχει και την νέα, πλέον, ανάρτηση.



The screenshot shows a list of movie posts on the Moviet platform. The third post, titled 'A Beautiful Mind', is highlighted with a blue arrow pointing to its 'Edit' button. The post details are as follows:

- Title:** A Beautiful Mind
- Date:** 29/06/2020
- Genre:** Drama
- Posters:** One poster is shown.
- Actions:** Edit, Details, Delete

Εικόνα 24 Updated post list

Όταν ο χρήστης κλικάρει το κουμπί “Edit”, εμφανίζεται μια φόρμα επεξεργασίας της δημοσίευσης. Όταν αλλάξει τα επιθυμητά στοιχεία, πατάει το κουμπί “Update”, ώστε να ανανεώσει τις αλλαγές.

The screenshot shows the 'Edit Post' form for the movie 'A Beautiful Mind'. The form fields include:

- Title:** A Beautiful Mind
- Sort Description:** After John Nash, a brilliant but asocial m...
- Long Description:** From the heights of notoriety to the dep...
- Poster:** Choose image... (button), Browse (button), Current Poster (image of the movie poster)
- Trailer:** https://www.youtube.com/watch?v=WF...
- Genres:** Drama
- Rating:** 3 ★

A blue arrow points to the 'Update' button at the bottom left of the form. Other buttons include 'Back' and 'Cancel'.

Εικόνα 25 Edit post form

Πατώντας το κουμπί “Delete”, εμφανίζεται μια φόρμα επιβεβαίωσης διαγραφής της δημοσίευσης. Όταν επιβεβαιώσει ο χρήστης τη διαγραφή, η δημοσίευση δεν είναι πλέον διαθέσιμη για προβολή. Η ταινία παραμένει ως Removed (βλ. Εικόνα 11).



The screenshot shows a browser window for the 'Moviet' application. The URL is 'localhost:44392/Posts/Delete/27'. The page has a header with 'Moviet', 'Home', 'Movies', 'Your Ratings', 'Your Posts', and 'Account'. It also shows 'Hello content_manager! Logout'. Below the header, there's a 'Delete' section with the question 'Are you sure you want to delete this?'. It lists the movie details: Title 'A Beautiful Mind' and Genres 'Drama'. To the right is a movie poster for 'A Beautiful Mind' featuring Russell Crowe. At the bottom left is a 'Delete' button and a 'Back' link. A blue arrow points from the text above to the 'Delete' button.

Εικόνα 26 Delete post confirmation

Στην περίπτωση που ο χρήστης είναι Administrator, έχει διαφορετικές δυνατότητες. Μπορεί να αφαιρέσει μια δημοσίευση, ακόμα και κάποιον χρήστη, αν κρίνει πως δεν ακολουθούνται οι όροι της κοινότητας. Στην σελίδα με της πληροφορίες μιας ταινίας, εμφανίζεται ένα κουμπί "Ban" δίπλα από τον τίτλο. Επιπλέον, υπάρχει κατάλληλο μήνυμα, το οποίο ενημερώνει τον χρήστη πως δεν μπορεί να αξιολογήσει μια ταινία με την λογαριασμό τύπου Administrator.

The screenshot shows a browser window for the 'Moviet' application. The URL is 'localhost:44392/Movies/Details/25'. The page has a header with 'Moviet', 'Home', 'Movies', 'List', 'Genres', and 'Account'. It also shows 'Hello admin@admin.com Logout'. Below the header, there's a 'Details' section for the movie 'The Imitation Game'. It features a movie poster for 'The Imitation Game' with Benedict Cumberbatch. To the right of the poster is a summary: 'In 1951, two policemen, Nock and Staehl, investigate mathematician Alan Turing following an apparent break-in at his home. During an interrogation by Nock, Turing tells of his time working at Bletchley Park during the Second World War. In 1927, young Turing, unhappy bullied at boarding school, develops a friendship with Christopher Morcom, who sparks his interest in cryptography. Turing develops romantic feelings for him, but Christopher soon dies from tuberculosis. When Britain declares war on Germany in 1939, Turing travels to Bletchley Park. Under the direction of Commander Alastair Denniston, he joins the cryptography team of Hugh Alexander, John Cairncross, Peter Hilton, Keith Furman and Charles Richards, who try to decrypt the Enigma machine that the Nazis use to send coded messages.' Below the summary is a note: 'You cannot rate movies as Administrator'. A red arrow points from the text above to the 'Ban' button. Another red arrow points from the text above to the note about rating restrictions. At the bottom, there's a 'Trailer' section with a video player for 'The Imitation Game Official Trailer #1 (2014) - Benedict Cumberbatch Movie HD'. The video player has controls for play, volume, and sharing.

Εικόνα 27 Movie details for Administrator

Πατώντας το κουμπί List στην μπάρα περιήγησης, εμφανίζονται δύο επιλογές. Μια για την προβολή λίστας με όλες τις δημοσιεύσεις και μια για όλους τους χρήστες. Στην πρώτη περίπτωση, υπάρχουν κουμπιά για την προβολή της σελίδας με τις λεπτομέρειες κάθε δημοσίευσης και για την αφαίρεσή τους. Στην δεύτερη, υπάρχει μόνο κουμπί για αποβολή του χρήστη.



The screenshot shows a table titled "All Posts" with 14 rows of movie data. The columns are: PostID, Date Created, Owner, Title, Details, and Ban. A blue arrow points to the "List" tab in the navigation bar.

PostID	Date Created	Owner	Title	Details	Ban
12	15/06/2020	content_manager1	Aladdin	Details	Ban
13	15/06/2020	content_manager2	Logan	Details	Ban
15	15/06/2020	content_manager1	Avengers: Endgame	Details	Ban
17	15/06/2020	content_manager2	Joker	Details	Ban
18	15/06/2020	content_manager1	Dunkirk	Details	Ban
19	16/06/2020	content_manager3	Parasite	Details	Ban
20	16/06/2020	content_manager3	1917	Details	Ban
21	16/06/2020	content_manager2	The Godfather	Details	Ban
22	16/06/2020	content_manager2	Uncut Gems	Details	Ban
23	16/06/2020	content_manager1	Inception	Details	Ban
24	16/06/2020	content_manager2	Jojo Rabbit	Details	Ban
25	16/06/2020	content_manager3	The Imitation Game	Details	Ban

Εικόνα 28 All Posts list

The screenshot shows a table titled "All Users" with 6 rows of user data. The columns are: UserName, Email, and Role. A blue arrow points to the "All Users" button in the navigation bar.

UserName	Email	Role
content_manager1	content_manager1@content_manager.com	ContentManager
content_manager2	content_manager2@content_manager.com	ContentManager
content_manager3	content_manager3@content_manager.com	ContentManager
rater1	rater1@rater.com	Rater
rater2	rater2@rater.com	Rater
rater3	rater3@rater.com	Rater

Εικόνα 29 All Users list

Επιπλέον, ο Administrator έχει τη δυνατότητα να διαχειριστεί τα διαθέσιμα είδη ταινιών. Πατώντας το κουμπί “Genres” στην μπάρα περιήγησης, εμφανίζεται μία λίστα με όλα τα καταχωρημένα ήδη. Επίσης, υπάρχουν κουμπιά για την δημιουργία ενός νέου είδους, για την επεξεργασία και διαγραφή ενός καταχωρημένου.



The screenshot shows the 'Genres' index page of the Moviet application. At the top, there is a navigation bar with links for Home, Movies, List, Genres, and Account. The 'Genres' link is highlighted. On the left, there is a sidebar with a 'Genres' section and a '+ New Genre' button. The main area displays a table of genres with columns for Name, Edit, and Delete. The genres listed are Comedy, Adventure, Action, Drama, and Thriller. Each row has an 'Edit' button in blue and a 'Delete' button in red.

Εικόνα 30 Genres list

The screenshot shows the 'Add Genre' page of the Moviet application. At the top, there is a navigation bar with links for Home, Movies, List, Genres, and Account. The 'Genres' link is highlighted. The main area has a form titled 'Genres' with a 'Add Genre' section. It includes a 'Name' input field containing 'Horror', a 'Create' button in green, and a 'Back' button in grey. The 'Name' input field is highlighted with a red border.

Εικόνα 31 Create new genre

Μπορείτε να παρακολουθήσετε και το βίντεο παρουσίασης της εφαρμογής στο
https://drive.google.com/file/d/151XYTxNtoTcUDuzCPkpi5AI0_c-GuSwu/view?usp=sharing

Ευχαριστούμε πολύ!