

[**ΤΕΧΝΟΛΟΓΙΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ**](https://exams-iee.the.ihu.gr/course/view.php?id=26)

Πληροφοριακό σύστημα μηχανογράφησης εταιρείας παραγωγής κινηματογραφικών ταινιών

**Στραπλάκης Κωνσταντίνος - 185280**

Εξάμηνο: 9 ο

Ακαδημαϊκό Έτος 2022 – 2023

**Βασίλειος Σιαμάτρας - 185273**

Εξάμηνο: 9 ο

Ακαδημαϊκό Έτος 2022 – 2023

Περιεχόμενα

[Εισαγωγή 3](#_Toc120567311)

[Εφαρμογή 3](#_Toc120567312)

[ER Diagram – Paramount Pictures 14](#_Toc120567313)

[Βάση Δεδομένων – PostgreSQL 15](#_Toc120567314)

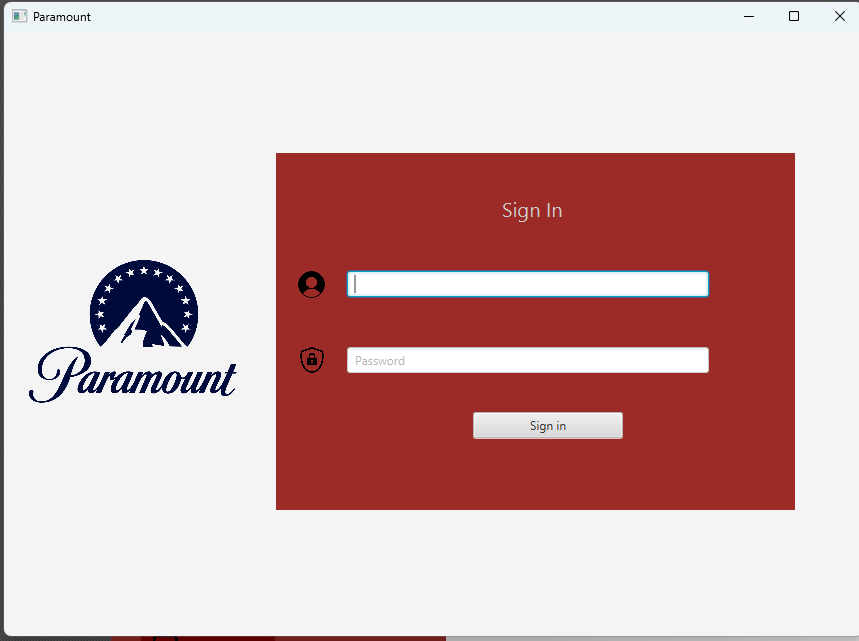
[Java – JavaFX – JDBC - IntelliJ 19](#_Toc120567315)

Εισαγωγή

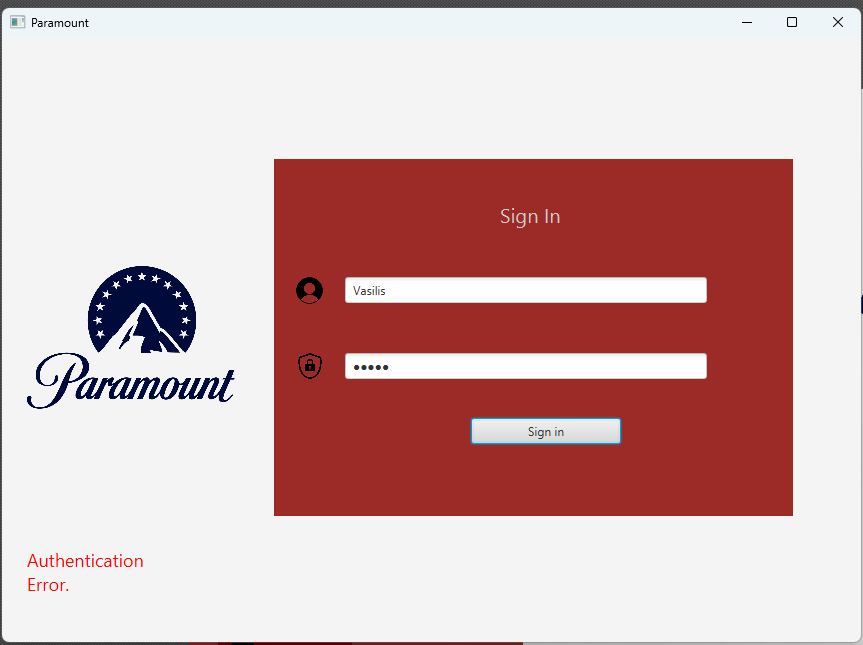
Το θέμα που διαλέξαμε να υλοποιήσουμε είναι μία τυχαία εταιρεία παραγωγής κινηματογραφικών ταινιών και ποιό συγκεκριμένα η δικιά μας επιλογή ήταν η Paramount Pictures. Στην συγκεκριμένη εφαρμογή υπάρχουν οι ταινίες, οι ηθοποιοί που πρωταγωνιστούν σε αυτές, οι σκηνοθέτες των ταινιών, η μουσική των ταινιών και οι επερχόμενες ταινίες που δεν έχουν κυκλοφορήσει ακόμα. Για την εφαρμογή χρησιμοποιείται μια βάση δεδομένων μέσω της PostgreSQL και οι ενέργειες που μπορεί να υλοποιήσει ο χρήστης είναι η διαγραφή και η εισαγωγή δεδομένων στις ταινίες, στους ηθοποιούς κοκ.

Εφαρμογή

Όπως παρουσιάζεται στις Εικόνες 1 και 2, η διεπιφάνεια εμφανίζει το log-in menu στο οποίο ο χρήστης καλείται να εισάγει τα στοιχεία του έτσι ώστε να κάνει σύνδεση στην εφαρμογή. Στην περίπτωση που εισάγει λάθος στοιχεία εμφανίζεται το μήνυμα “Authentication Error”.

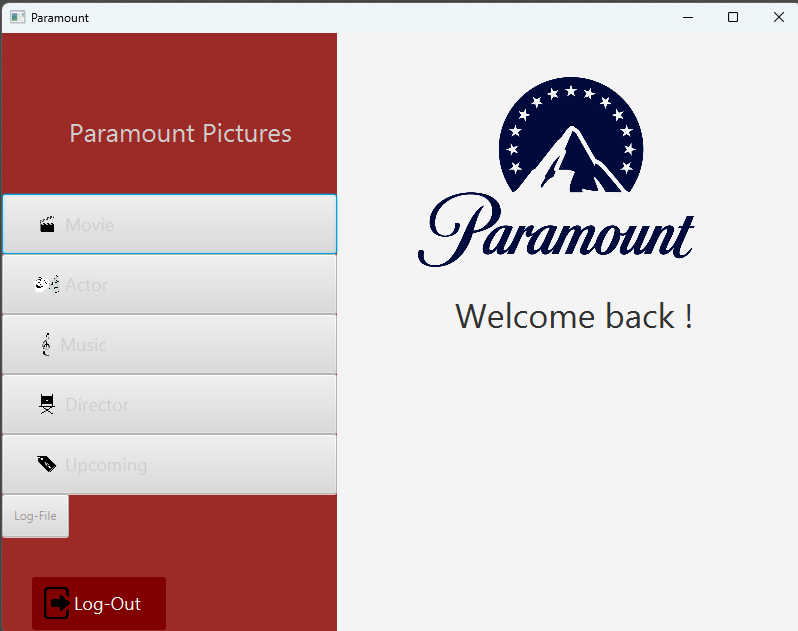


Εικόνα 1



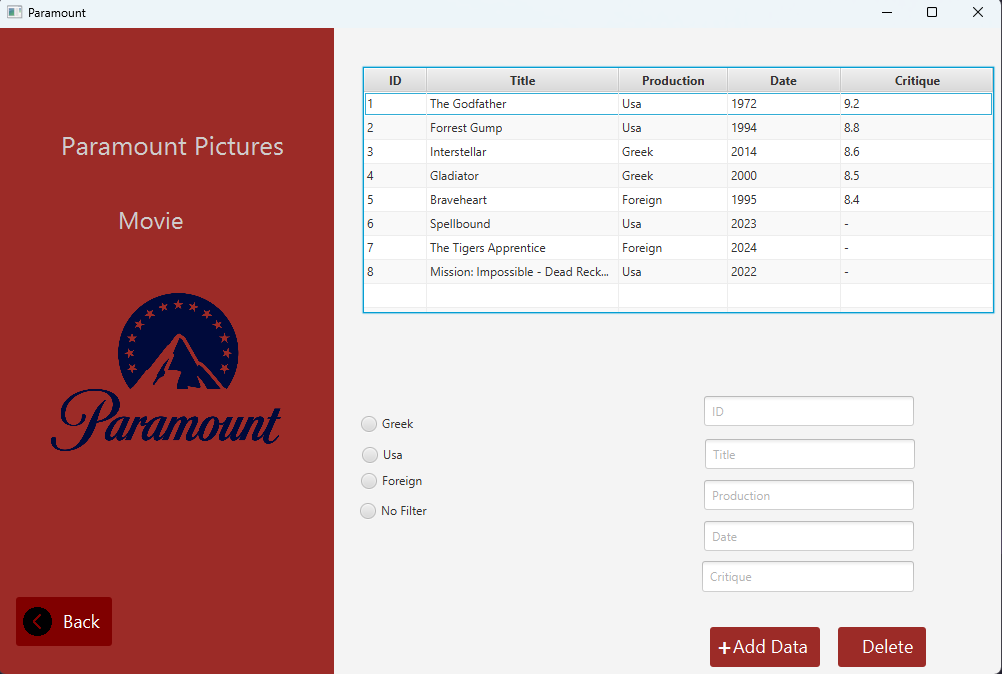
Εικόνα 2

Αφού εισάγει τα σωστά στοιχεία και γίνει η σύνδεση, εμφανίζεται το πρώτο παράθυρο (Εικόνα 3) που εμφανίζει ένα μήνυμα καλωσορίσματος και τις επιλογές Movie, Actor, Music, Director και Upcoming.

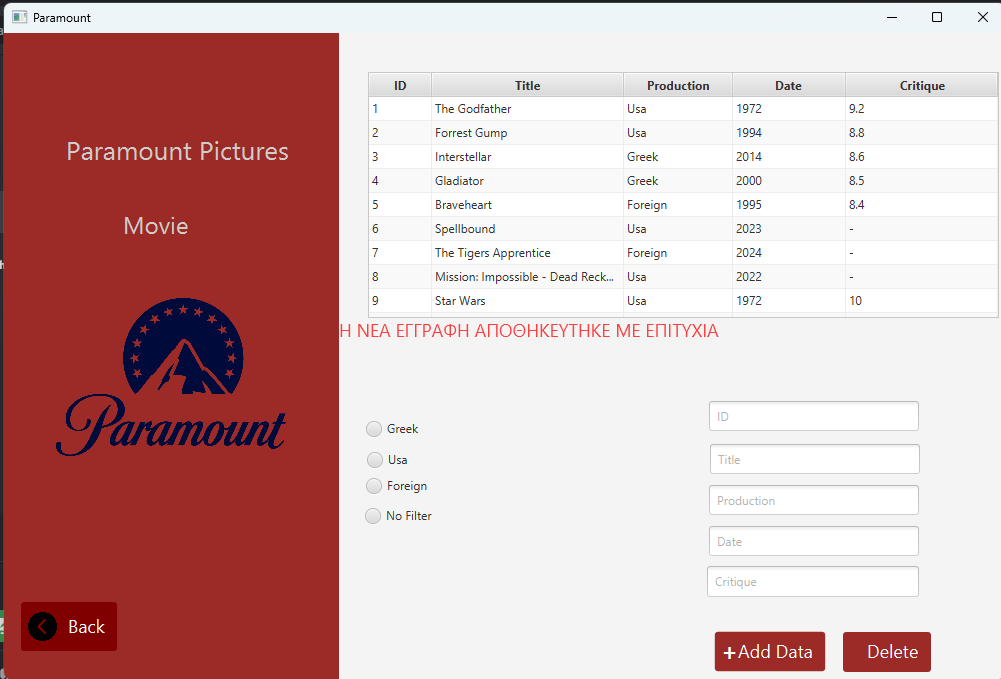


Εικόνα 3

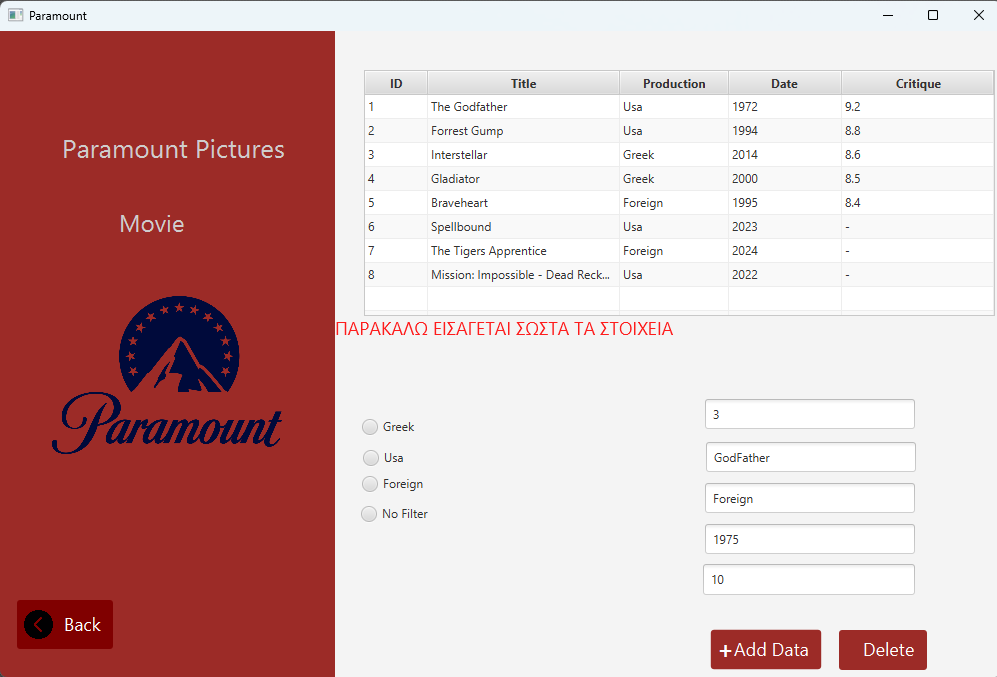
Ο χρήστης αφού κάνει κλικ στην επιλογή Movie, εμφανίζεται ένα καινούργιο παράθυρο (Εικόνα 4). Στο συγκεκριμένο παράθυρο εμφανίζονται τα στοιχεία των ταινιών, καθώς και οι ενέργειες που μπορεί να κάνει ο χρήστης. Στην περίπτωση που θέλει να εκχωρήσει μία καινούργια ταινία, ο χρήστης πρέπει να γράψει το ID όπου πρέπει να είναι ξεχωριστό, το όνομα της ταινίας, την χώρα παραγωγής, το έτος κυκλοφορίας και την κριτική (Εικόνα 4). Σε περίπτωση που η εισαγωγή γίνεται για ταινία που δεν έχει κυκλοφορήσει ακόμα στην κριτική καλό είναι να μπει παύλα. Πατώντας το κουμπί “Add data”, αυτομάτως προστίθενται τα στοιχεία της ταινίας στον πίνακα και εμφανίζεται ένα μήνυμα όπου μας ενημερώνει ότι η εγγραφή αποθηκεύτηκε με επιτυχία (Εικόνα 5). Σε περίπτωση που η εισαγωγή στοιχείων είναι λάθος (το Id υπάρχει ή κάποιο πεδίο είναι null) θα εμφανιστεί κατάλληλο μήνυμα που θα ενημερώνει για τα λάθος δεδομένα (Εικόνα 6). Στην περίπτωση που θέλουμε να διαγράψουμε μία εγγραφή, επιλέγουμε κάποιο στοιχείο του πίνακα και κάνουμε κλικ στο κουμπί “Delete”. Αφού γίνει η διαγραφή με επιτυχία, εμφανίζεται το κατάλληλο μήνυμα (Εικόνα 7).



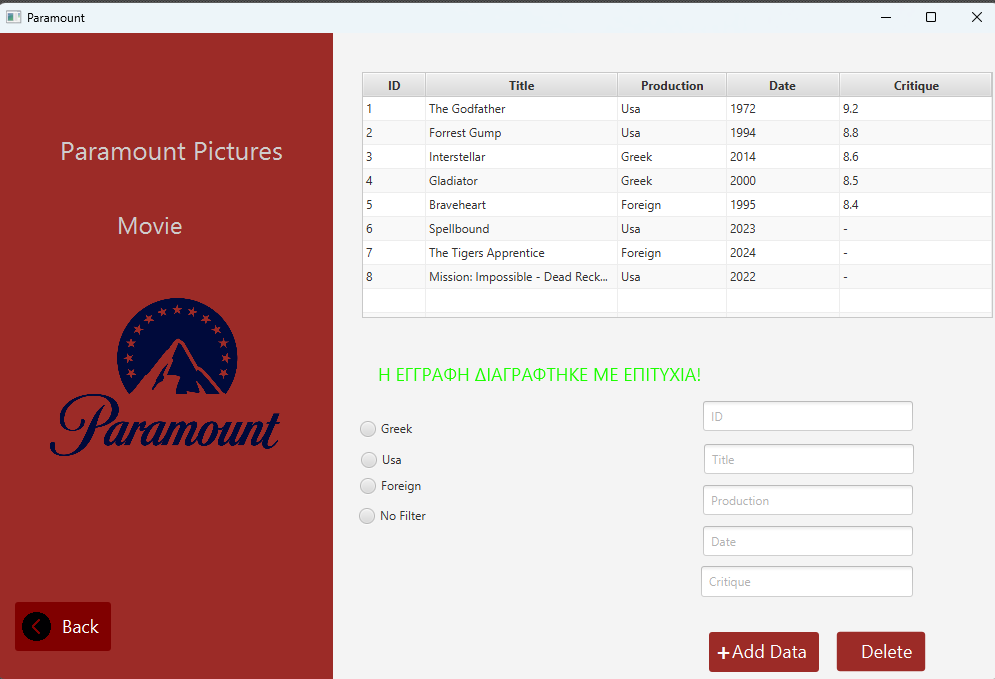
Εικόνα 4



Εικόνα 5

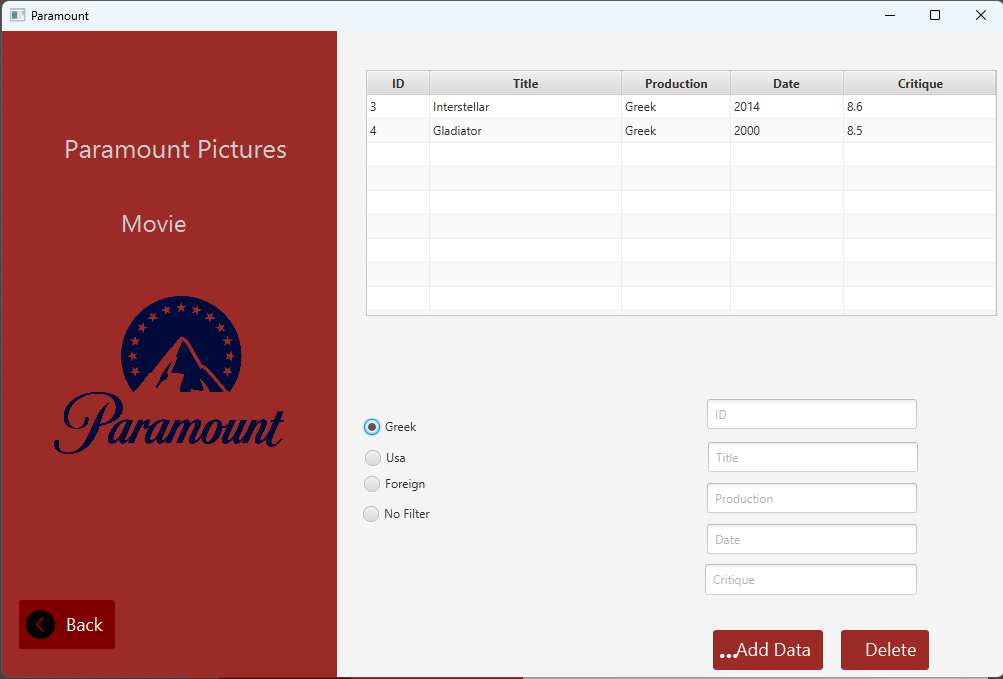


Εικόνα 6

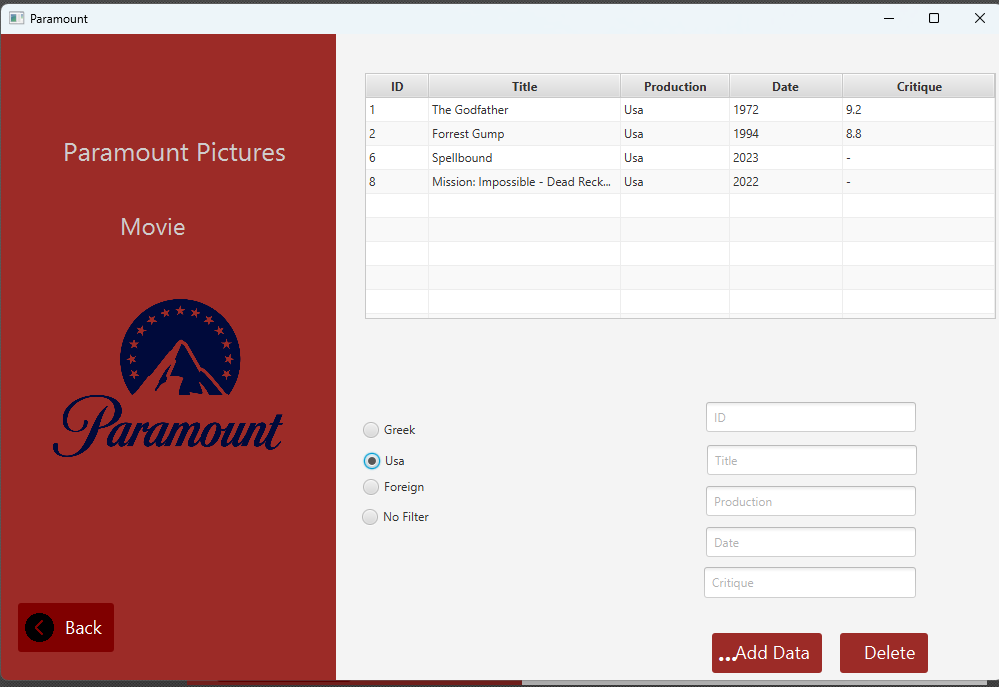


Εικόνα 7

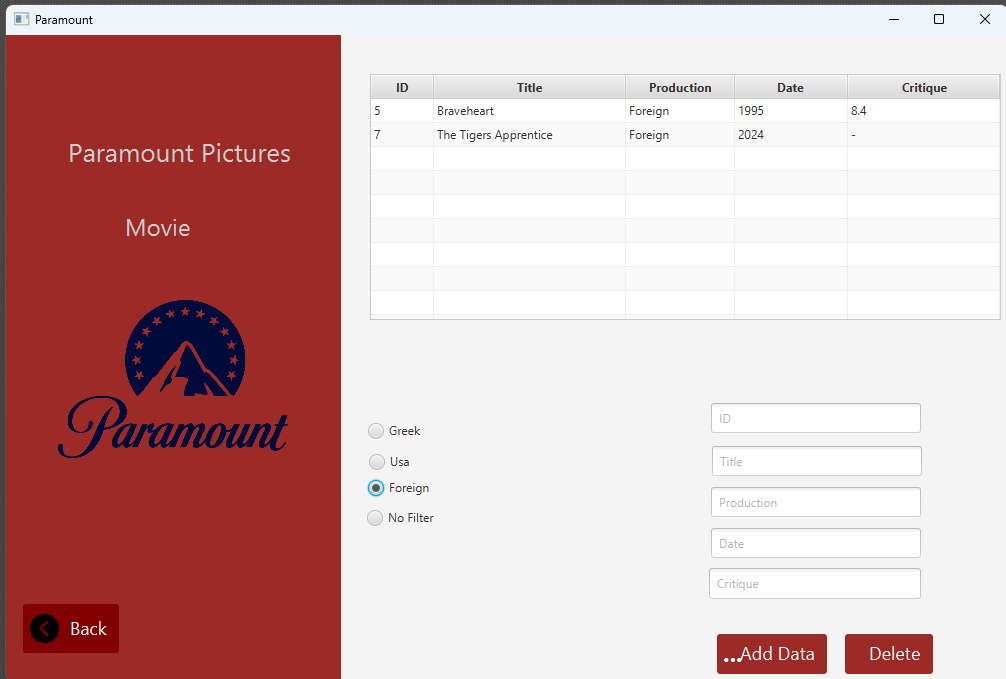
Επίσης μία δυνατότητα που δίνεται στο συγκεκριμένο παράθυρο είναι ότι υπάρχουν επιλογές για το φιλτράρισμα των ταινιών ανάλογα με την παραγωγή. Ο χρήστης θα μπορεί να επιλέγει ανάμεσα στις 4 επιλογές (Usa, Greek, Foreign, No Filter) και να εμφανίζονται οι ταινίες που έχουν παραγωγή την ίδια με την επιλογή (Εικόνα 8-9-10-11).



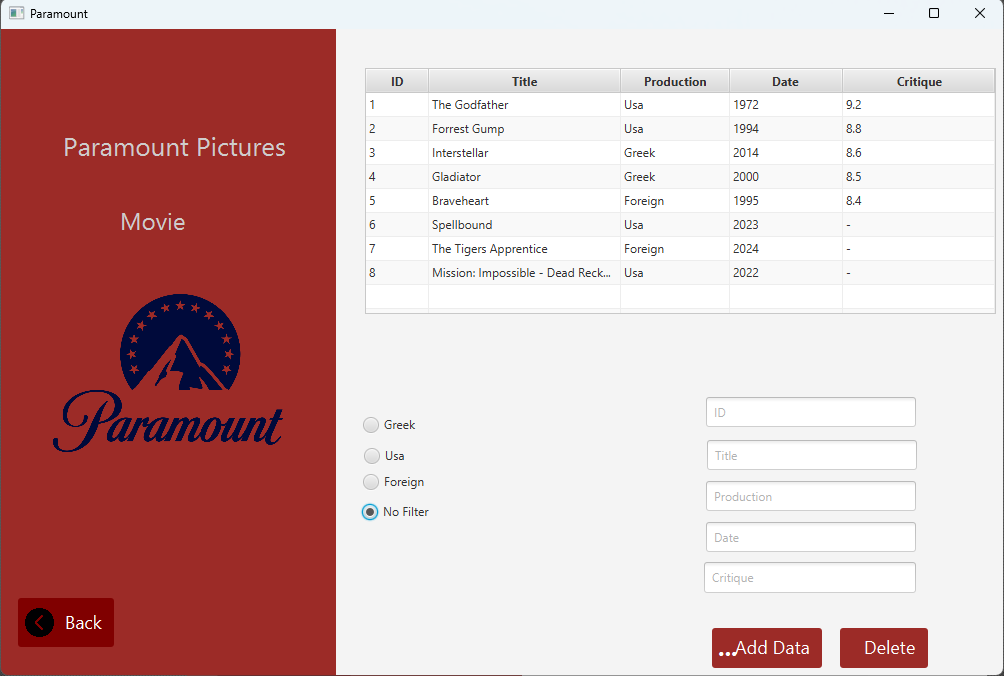
Εικόνα 8



Εικόνα 9

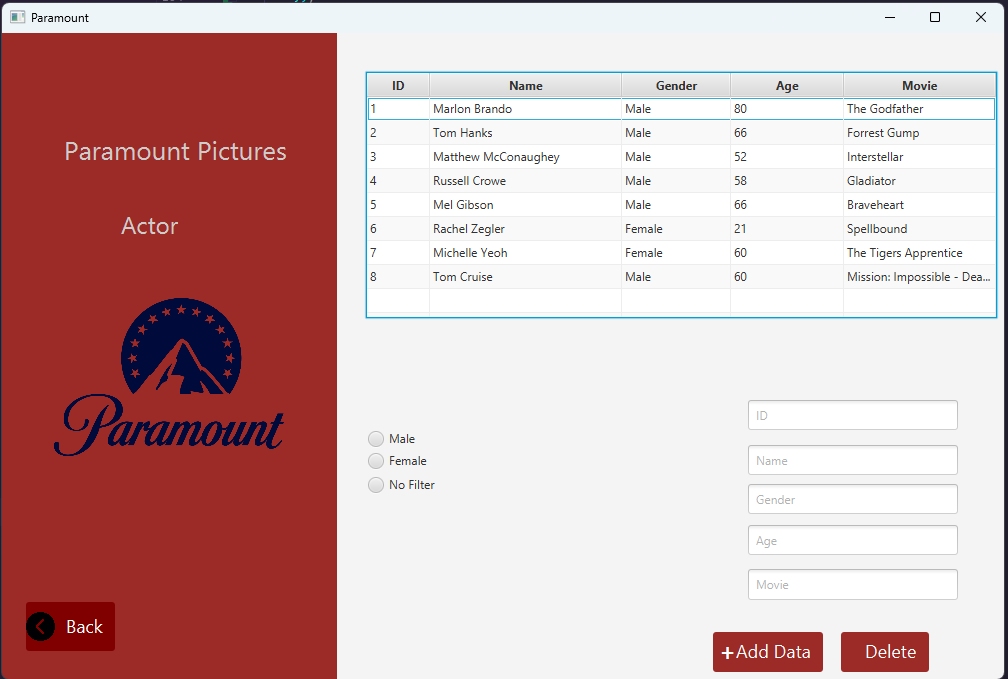


Εικόνα 10



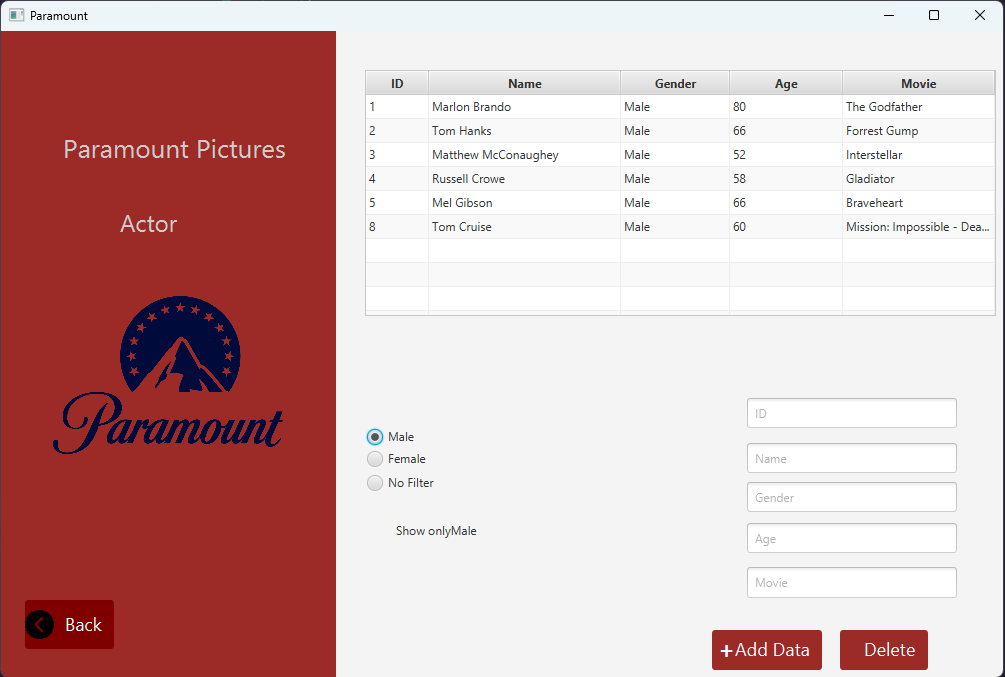
Εικόνα 11

Στην συνέχεια περνάμε στην επιλογή Actor που αφορά τους ηθοποιούς. Σε αυτή την επιλογή μπορούμε να πάρουμε τις πληροφορίες τους όπως το Id τους, το όνομα, το φίλο τους, την ηλικία τους και το όνομα της ταινίας που παίζουν (Εικόνα 12). Η πρόσθεση/αφαίρεση γίνεται με τον ίδιο τρόπο όπως και στον πίνακα Movie (Εικόνα 5-6-7).

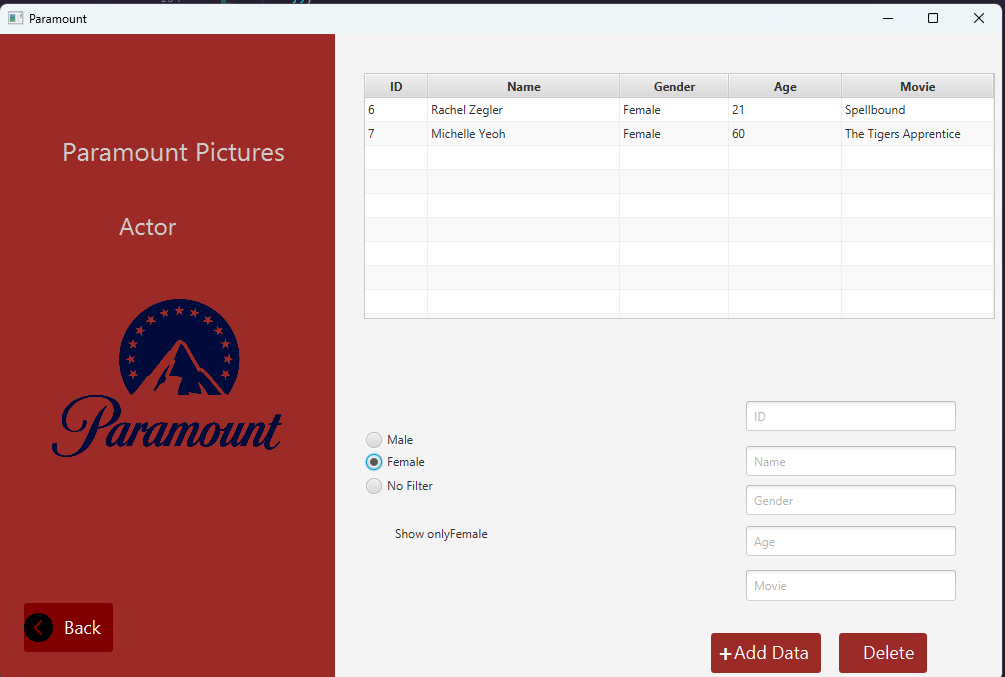


Εικόνα 12

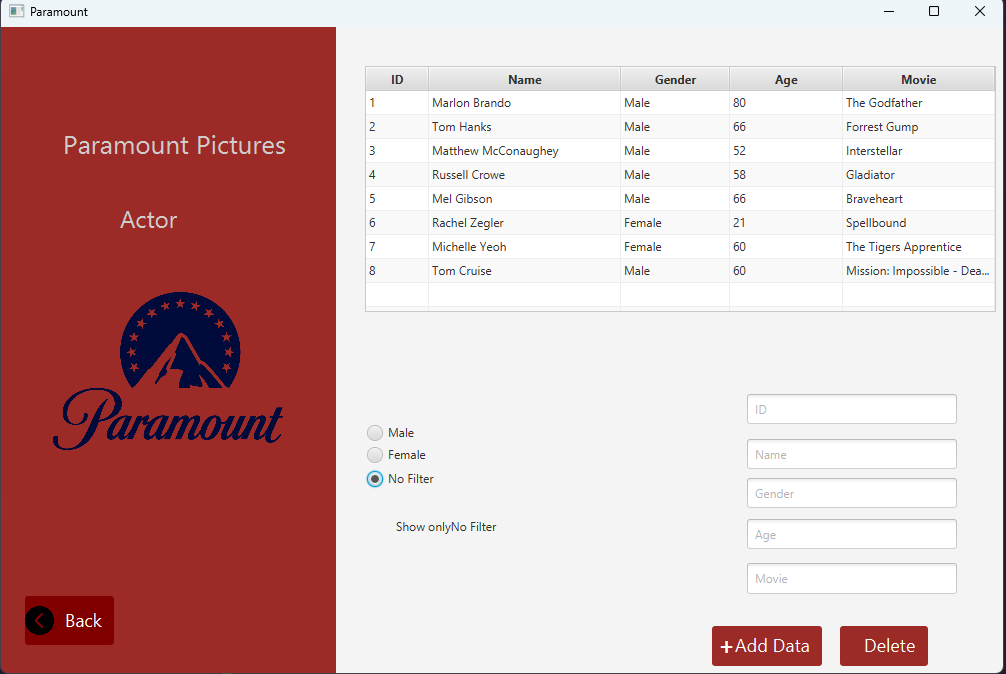
Κάτι αντίστοιχο με το παράθυρο του πίνακα Movie, θα υπάρχει φίλτρο για τους ηθοποιούς ανάλογα με το φύλο τους και θα υπάρχουν 3 επιλογές (Male, Female, No Filter) και με την επιλογή του κάθε φίλτρου θα εμφανίζονται τα στοιχεία των ηθοποιών (Εικόνα 13-14-15).



Εικόνα 13

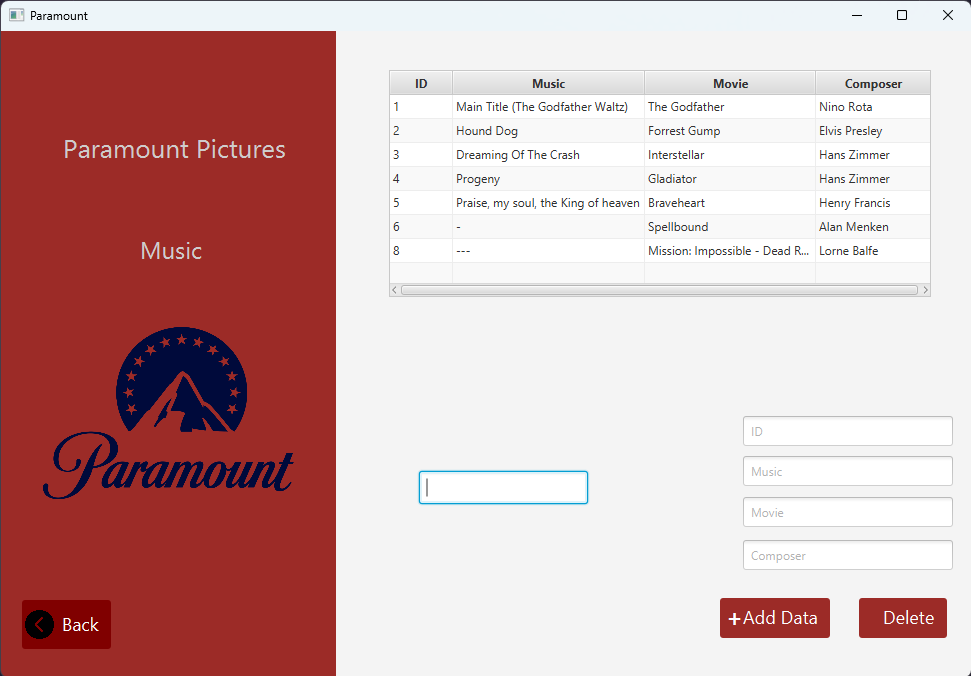


Εικόνα 14



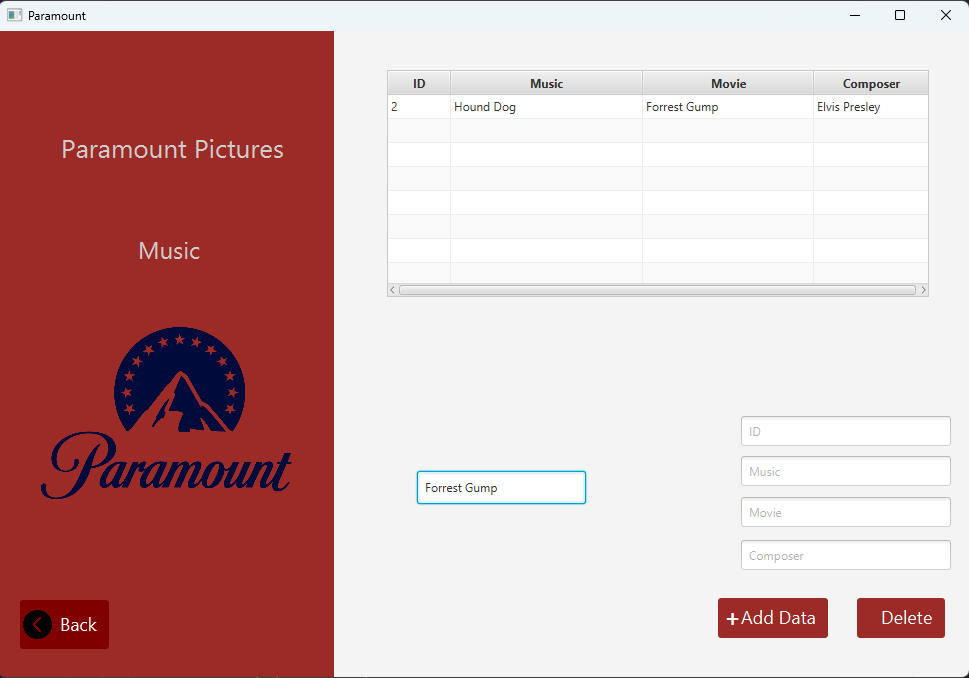
Εικόνα 15

Στην συνέχεια έχουμε την επιλογή Music όπου με την επιλογή της εμφανίζεται ο πίνακας που περιέχει όλα τα κόμματα και τα στοιχεία των τραγουδιών (Εικόνα 16) και η εισαγωγή και διαγραφή δεδομένων λειτουργεί όπως ακριβός και στον πίνακα Movie (Εικόνα 5-6-7).



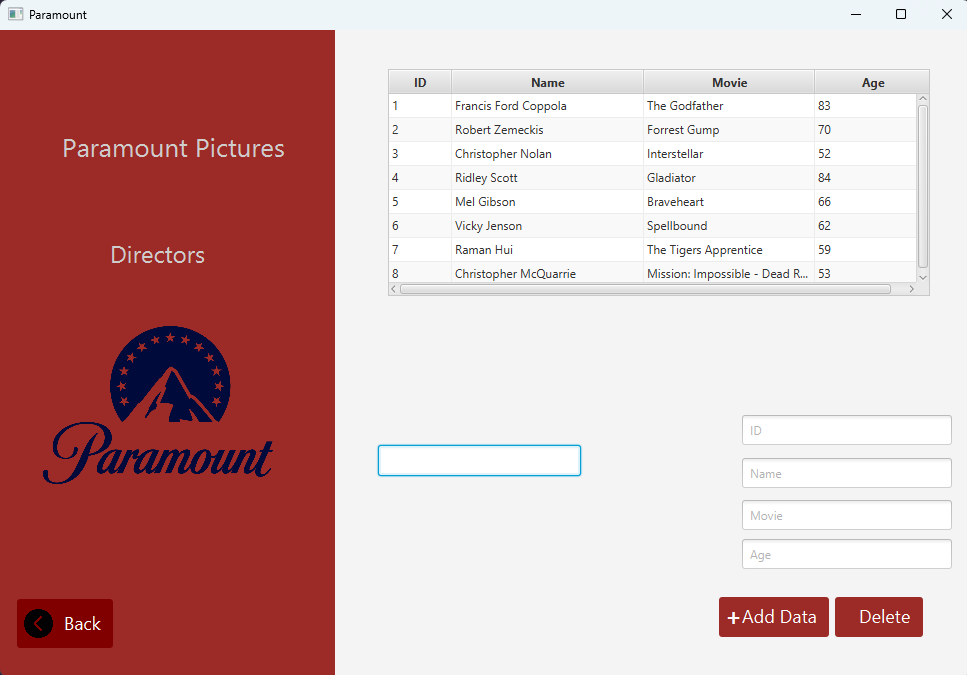
Εικόνα 16

Στον πίνακα Music υπάρχει και επιλογή για φίλτρο όπου στο ειδικό πεδίο ο χρήστης μπορεί να γράψει τον τίτλο μίας ταινίας και αν η ταινία υπάρχει θα εμφανιστούν όλα τα στοιχεία του πίνακα music που αφορούν την συγκεκριμένη ταινία (Εικόνα 17). Αν η ταινία δεν υπάρχει δεν θα εμφανιστεί τίποτα.



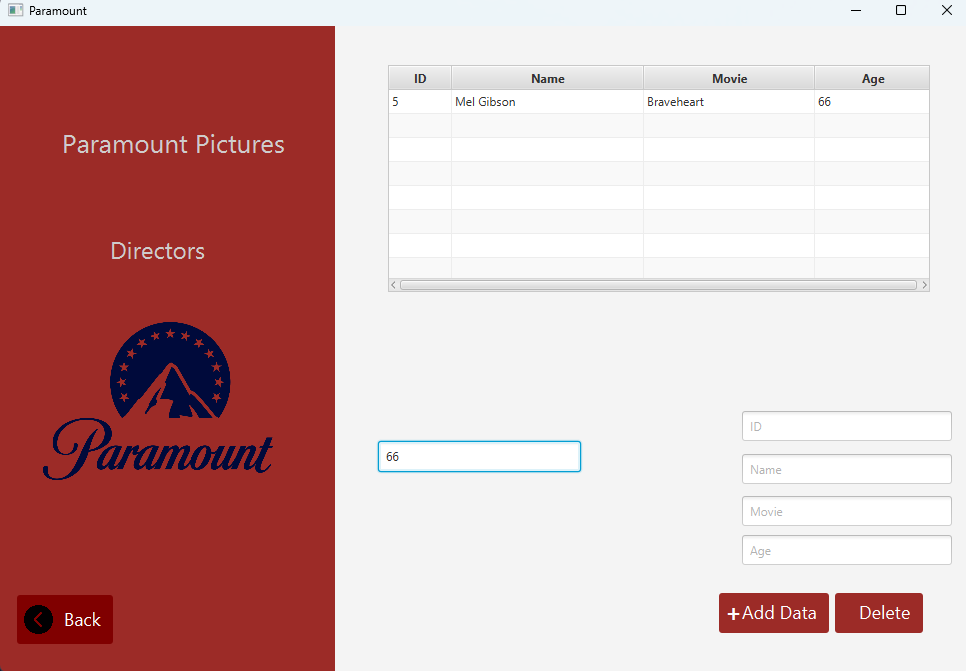
Εικόνα 17

Η επόμενη επιλογή είναι η επιλογή του Director όπου κατευθείαν βλέπουμε τα περιεχόμενα του πίνακα (Εικόνα 18) και όπως και στον πίνακα Movie υπάρχουν οι επιλογές insert και delete που λειτουργούν με τον ίδιο ακριβώς τρόπο (Εικόνα 5-6-7).



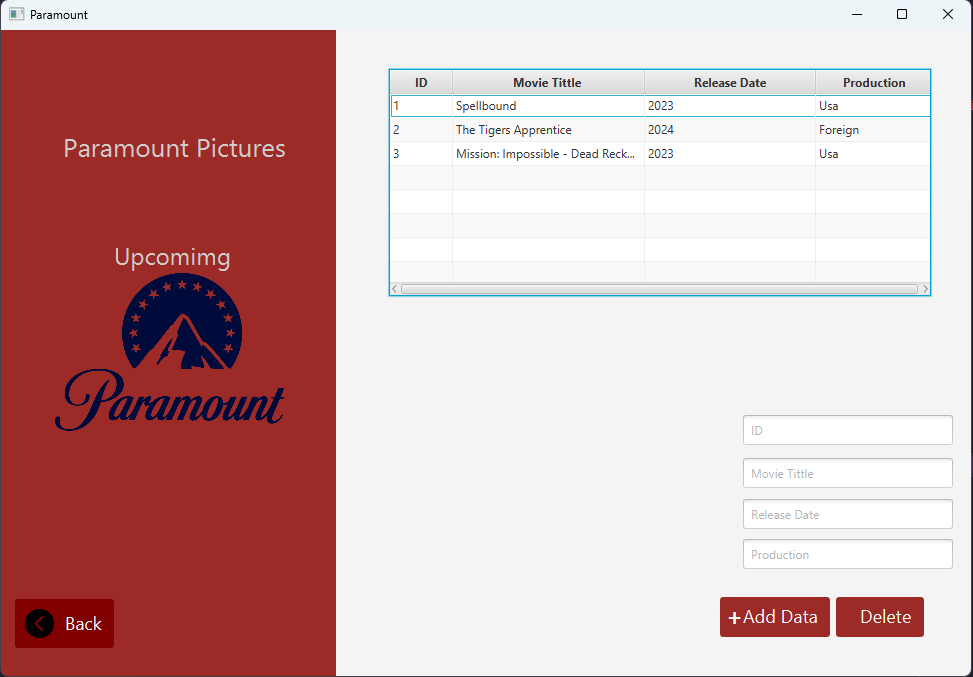
Εικόνα 18

Επίσης στον πίνακα director υπάρχει και επιλογή για φίλτρο ανάλογα με την ηλικία του κάθε director. Έτσι ο χρήστης θα μπορεί να γράψει μια ηλικία και θα του εμφανίζονται οι εγγραφές των directors που έχουν αυτή την ηλικία (Εικόνα 19).



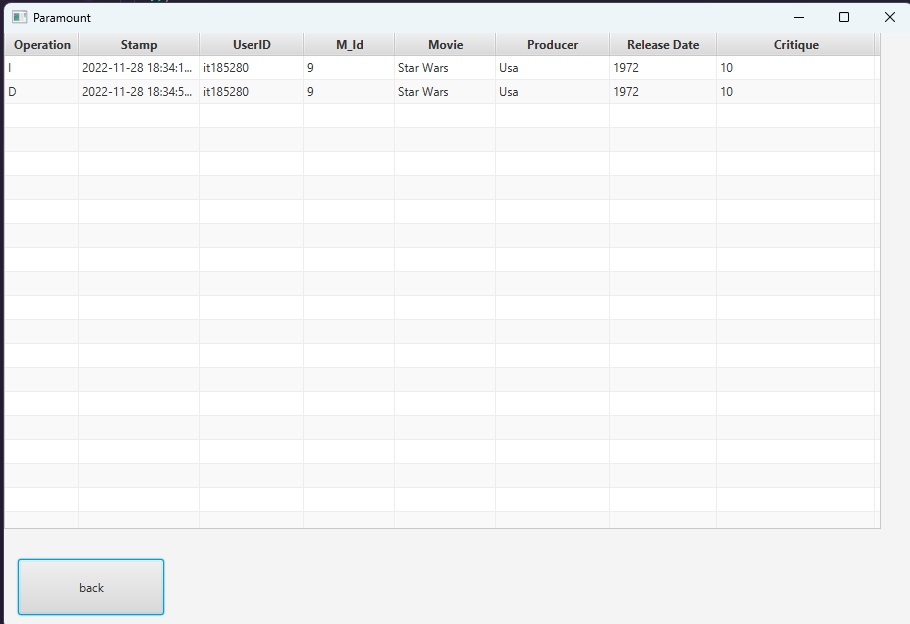
Εικόνα 19

Τέλος υπάρχει και η επιλογή Upcoming όπου εμφανίζονται οι ταινίες του πίνακα Movie που δεν έχουν κυκλοφορήσει ακόμα (Εικόνα 20). Οι λειτουργίες των Insert και Delete είναι ίδιες με του πίνακα Movie (Εικόνα 5-6-7).



Εικόνα 20

Τελευταία επιλογή είναι αυτή του Log File του πίνακα Movie όπου εκεί σε μορφή πίνακα θα εμφανίζονται το id του user, ένα timestamp, τα κεφαλαία I ή D που θα υποδηλώνουν την ενέργεια που πραγματοποιήθηκε (Insert or Delete) και τα στοιχεία που μπήκαν ή διαγράφτηκαν από τον πίνακα Movie (Εικόνα 21).



Εικόνα 21

# ER Diagram – Paramount Pictures

Μία βάση δεδομένων δεν μπορεί να μην αναπαραστάται και με ένα διάγραμμα ER, έτσι και στην συγκεκριμένη εφαρμογή φαίνεται το διάγραμμα ER που δημιουργήθηκε. Όπως φαίνεται στην εικόνα, βλέπουμε τις πέντε κλάσεις που θεωρούνται και οντότητες. Έχουμε τις κλάσεις Movie, Actor, Music, Director, Upcoming (Εικόνα 22).

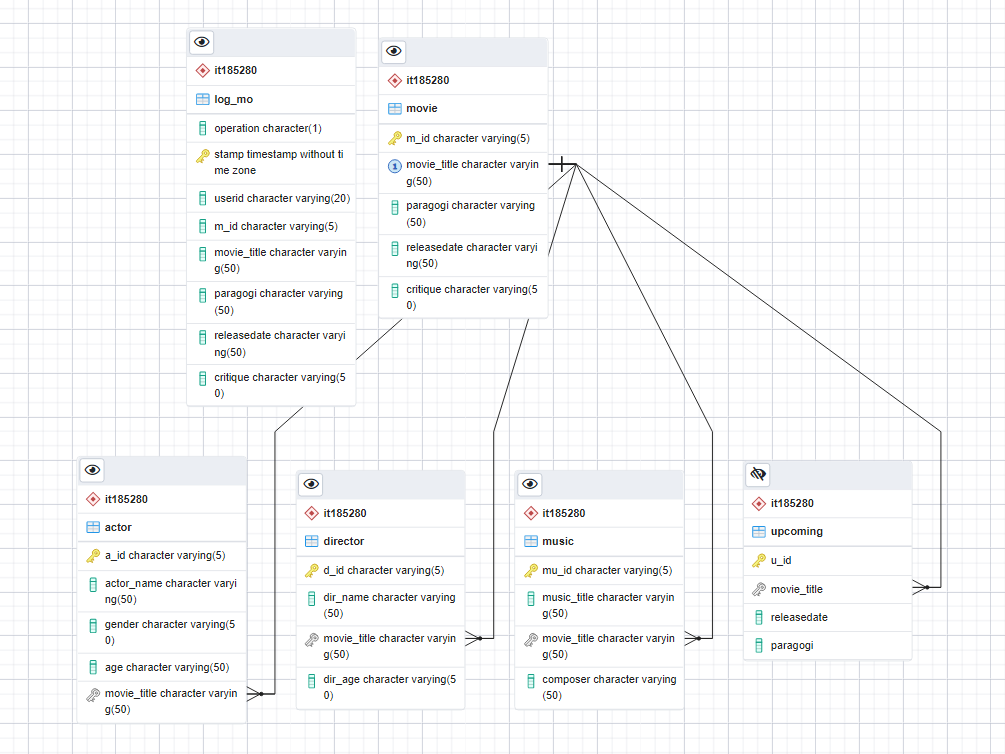
Η κλάση movie έχει ως πεδία το m\_id τύπου Character(5), το movie\_title Character(50), το paragogi Character(50), το releasedate Character(50) και το critique Character(50). Με κύριο κλειδί να είναι το m\_id.

Η κλάση actor έχει ως πεδία το a\_id τύπου Character(5), το actor\_name Character(50), το gender Character(50), το age Character(50) και το movie\_title Character(50). Με κύριο κλειδί να είναι το a\_id και δευτερεύον κλειδί το movie\_title σε σύνδεση με τον πίνακα movie.

Η κλάση music έχει ως πεδία το mu\_id τύπου Character(5), το music\_title Character(50), το movie\_title Character(50) και το composer Character(50). Με κύριο κλειδί να είναι το mu\_id και δευτερεύον κλειδί το movie\_title σε σύνδεση με τον πίνακα movie.

Η κλάση director έχει ως πεδία το d\_id τύπου Character(5), το dir\_name Character(50), το movie\_title Character(50) και το dir\_age Character(50). Με κύριο κλειδί να είναι το d\_id και δευτερεύον κλειδί το movie\_title σε σύνδεση με τον πίνακα movie.

Η κλάση upcoming έχει ως πεδία το u\_id τύπου Character(5), το movie\_title Character(50), το releasedate Character(50) και το paragogi Character(50). Με κύριο κλειδί να είναι το u\_id και δευτερεύον κλειδί το movie\_title σε σύνδεση με τον πίνακα movie.

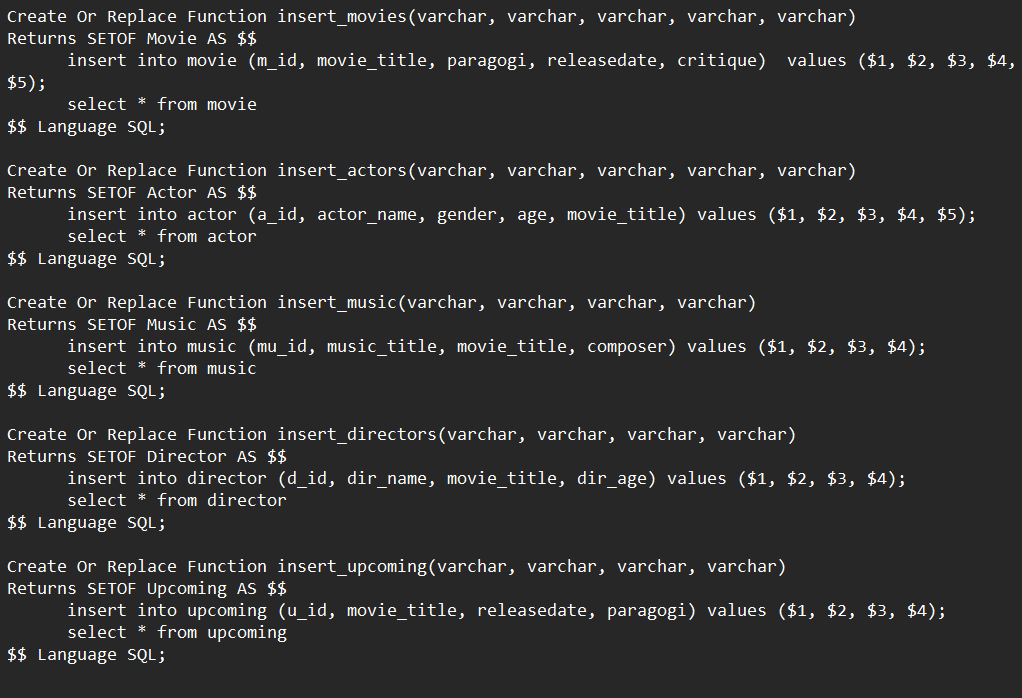


Εικόνα 22

# Βάση Δεδομένων – PostgreSQL

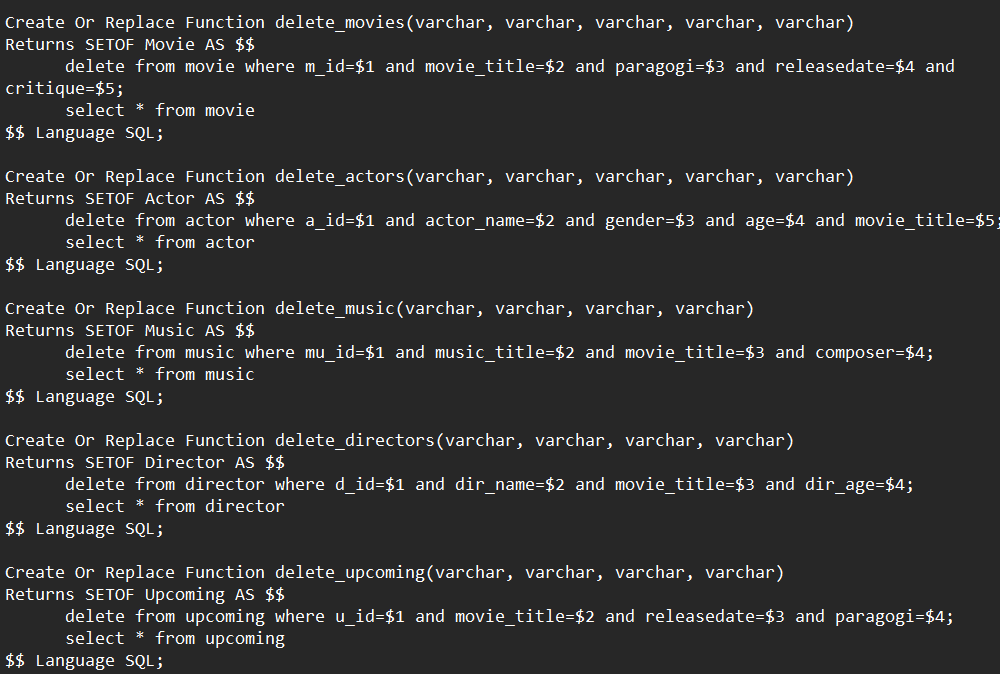
Πίσω από την εφαρμογή τρέχει μία βάση δεδομένων δημιουργημένη σε PostgreSQL. Για την υλοποίηση και επεξεργασία των δεδομένων, χρησιμοποιήθηκαν views, functions, triggers και dynamic queries. Επίσης έχει δημιουργηθεί και ένα log-file για τον πίνακα Movie, το οποίο καταγράφει όλες τις αλλαγές που έγιναν και θα γίνουν στον συγκεκριμένο πίνακα (Εικόνα 21).

* Functions για την εισαγωγή των δεδομένων



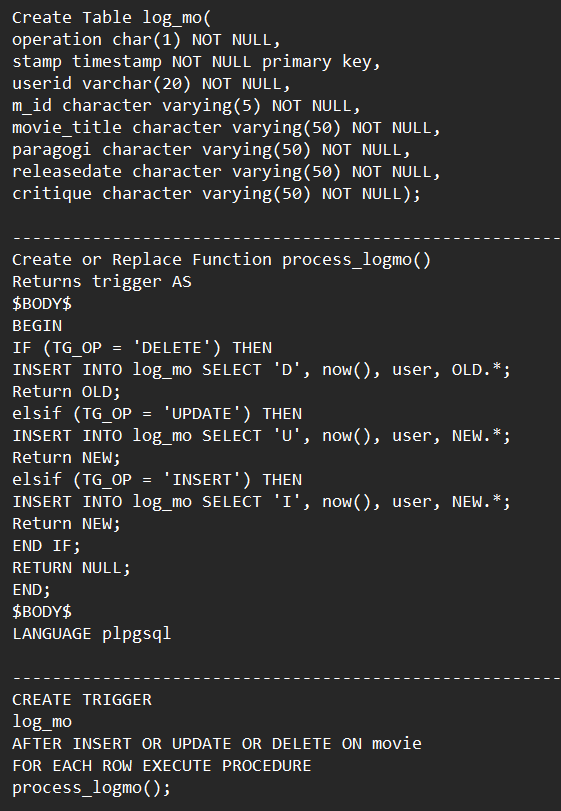
**Εικόνα 23**

* Functions για την διαγραφή των δεδομένων



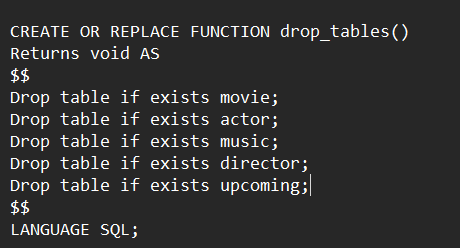
**Εικόνα 24**

* Log File
  + Για το Log-file δημιουργήθηκε ένας πίνακας Log για την καταγραφή των δεδομένων σε στήλες. Επίσης δημιουργήθηκε μία function που διαχειρίζεται τα συμβάντα Delete, Update και Insert. Τέλος δημιουργήθηκε το trigger το οποίο είναι υπεύθυνο να ενημερώνει τον πίνακα Log, μετά την εκτέλεση των Delete, Update και Insert.



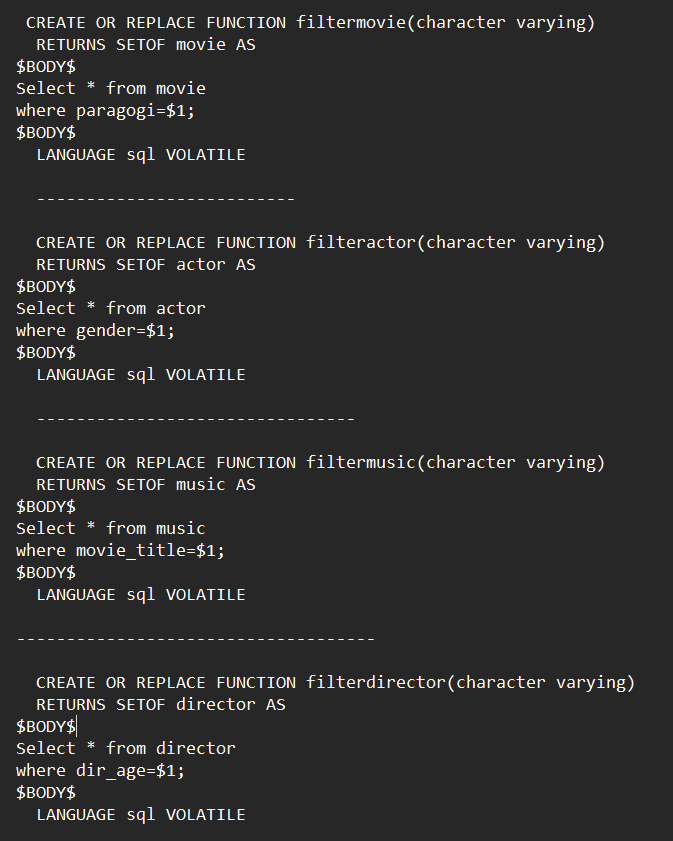
**Εικόνα 25**

* Function για την διαγραφή όλων των πινάκων

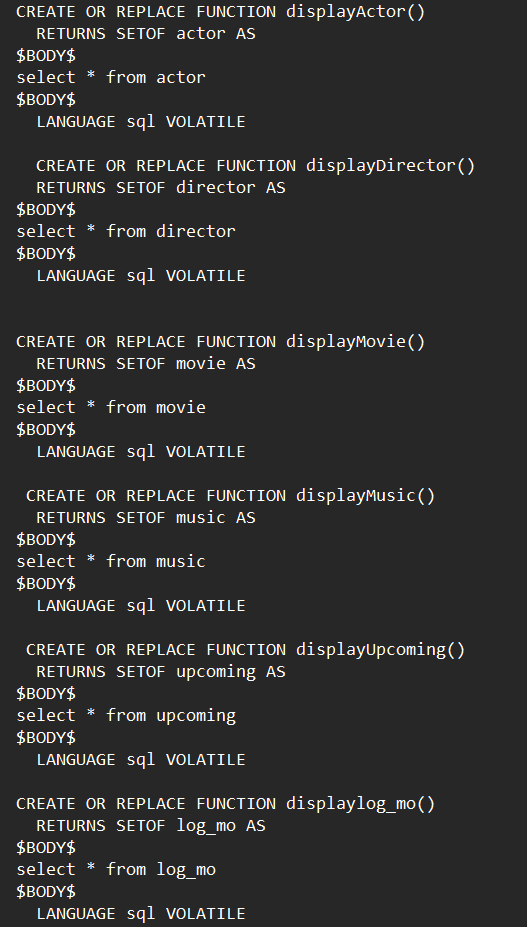


**Εικόνα 26**

* Dynamic Queries
  + Στην εφαρμογή υπάρχουν τα φίλτρα και για να εμφανιστούν τα σωστά αποτελέσματα χρησιμοποιήθηκαν τα κατάλληλα δυναμικά ερωτήματα.



**Εικόνα 27**



Εικόνα 28

# Java – JavaFX – JDBC – IntelliJ

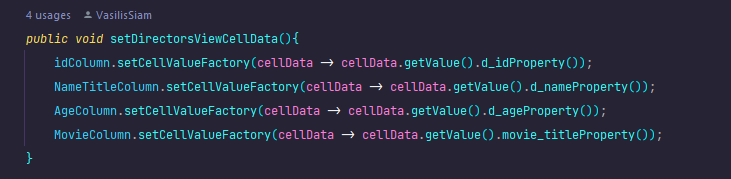
Μέσα στην εφαρμογή δημιουργήθηκαν οι ανάλογες κλάσεις για κάθε ένα πίνακα, ένα αρχείο .css το οποίο αφορά το design-styling και εφαρμόζεται στο UI. Προφανώς υπάρχουν και τα fxml αρχεία αλλά και οι controllers για κάθε μία κλάση. Όσον αφορά την αρχιτεκτονική της εφαρμογής, έχουμε χρησιμοποιήσει το μοντέλο MVC όπου για κάθε οντότητα της εφαρμογής(Actor,Director,etc) υπάρχουν 3 κλάσεις. Μία κλάση που περιγράφει την οντότητα και χρησιμοποιείται για να την περιγράψουμε, μία κλαση όπου είναι αρχείο .fxml η οποία περιγράφει το UI για κάθε οντότητα και μία κλάση Controller για την διαχείριση/τροποποίηση του UI. Επιπλέον για κάθε οντότητα έχουμε δημιουργήσει μία κλάση στην οποία γίνεται η επικοινωνία με την Βάση Δεδομένων(DAO πχ ActorDAO). Στις συγκεκριμένες κλάσεις γίνονται όλες οι διαδικασίες που τροποποιούν την βάση όπως είναι η εισαγωγή, η διαγραφή και τα δυναμικά ερωτήματα.

* Κλάση σύνδεσης με την Βάση

Σε κάθε κλάση DAO, χρησιμοποιούμε τον απαραίτητο κώδικα για να γίνει η σύνδεση με την βάση δεδομένων.



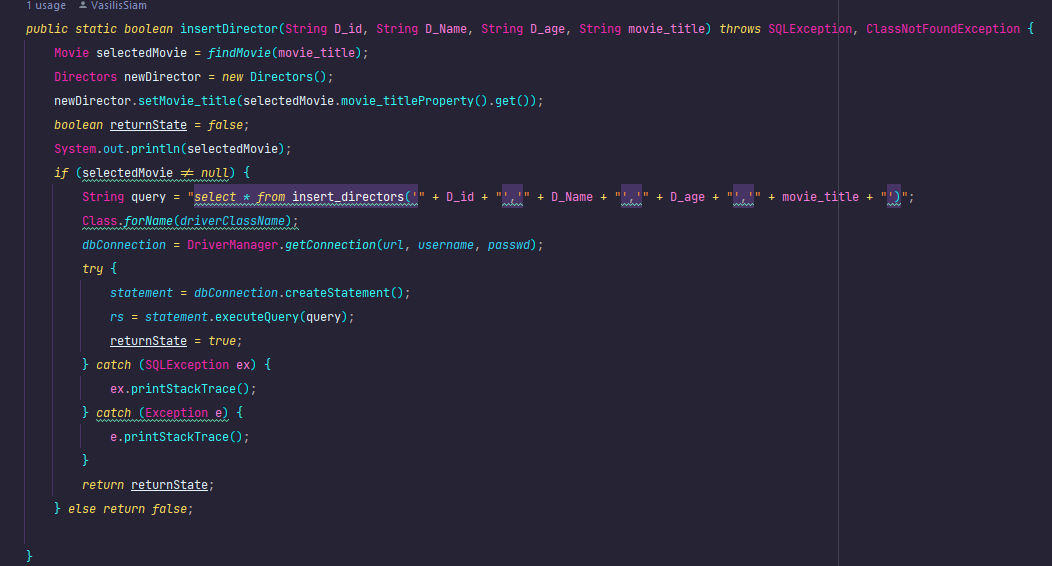
* Υλοποίηση function για τα dynamic queries
  + Οι παρακάτω μέθοδοι υλοποιούν την αρχική μορφή που έχουν το TableView<Directors> στην συγκεκριμένη περίπτωση επιστρέφοντας μια ObservableList στην κλάση του Controller όπου εκεί και έπειτα μέσων των μεθόδων setDirectorsViewCellData() και populateTable(ObservableList) εισάγονται τα δεδομένα στο TableView του UI).

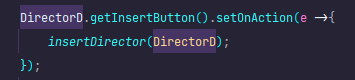


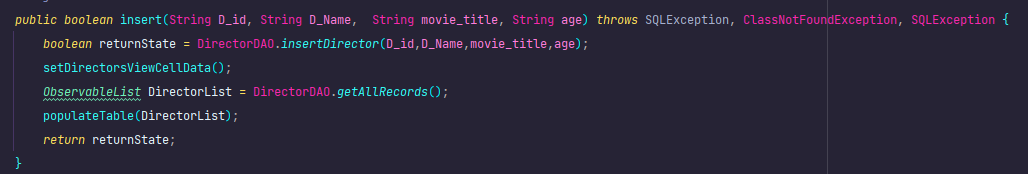
Επίσης οι κλάσεις Controller υλοποιούν την initialize() η οποία καλεί τις παραπάνω μεθόδους και διατυπώνει την αρχική μορφή του αντίστοιχου Scene (DirectorScene στην συγκεκριμένη περίπτωση).



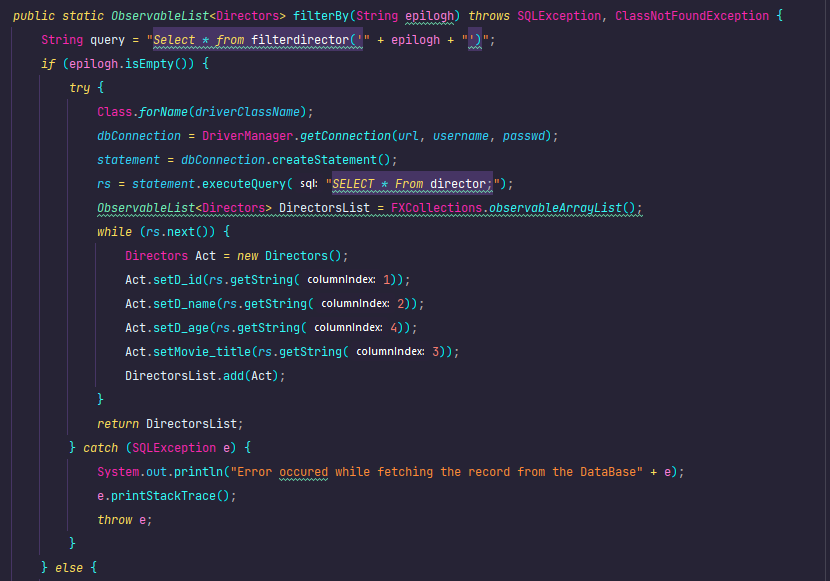
Όσον αφορά την διάδραση του χρήστη, για κάθε δράση που πρέπει να γίνει υπάρχει μία στατική μέθοδος event Handler (lambda expression e->{}) η οποία καθορίζει το τι θα γίνει όταν πραγματοποιηθεί κάποια δράση από τον χρήστη. Για παράδειγμα:

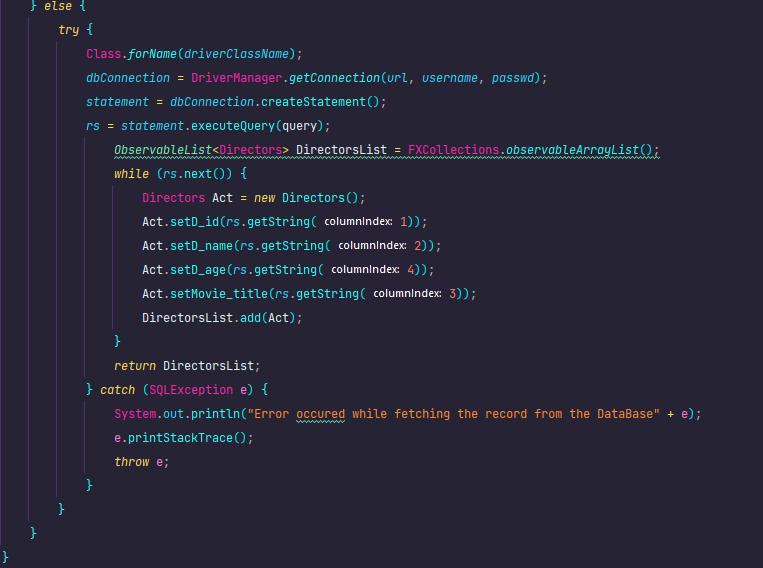
* + Όταν ο χρήστης έχει εισάγει τα στοιχεία στα αντίστοιχα TextFields μέσω τις event handler μεθόδου setOnAction() καλείται η μέθοδος insertDirectors(). Εφόσον ο χρήστης έχει εισάγει τα στοιχεία, στην insertDirectors() γίνεται έλεγχος τους για εισαγωγή στον πίνακα Director και εφόσον είναι σωστά καλείται η μέθοδος insert() της κλάσης DirectorsController και μέσα από αυτή καλείται η μέθοδος insertDirectors() της κλάσης DAO που επικοινωνεί με την Βάση όπου και εκτελείται το statement για εισαγωγή στον πίνακα SQL Director. Τέλος εφόσον εκτελεστεί σωστά το SQL ερώτημα η μέθοδος επιστρέφει μια boolean τιμή στην μέθοδο του Controller και εφόσον είναι true ανανεώνει τον πίνακα TableView<Director> στο UI.

****

****

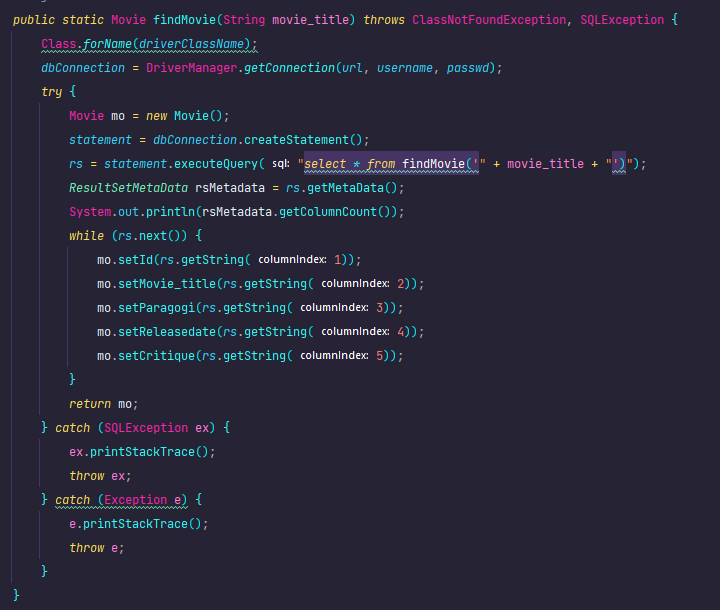
Ο controller της κλάσης Directors διαχειρίζεται όλες τις αλληλεπιδράσεις που μπορεί να κάνει ο χρήστης στην κατηγορία που αφορά τους Σκηνοθέτες. Στις επόμενες 2 εικόνες υλοποιείτε το dynamic query φίλτρο director στην μέθοδο filterBY.



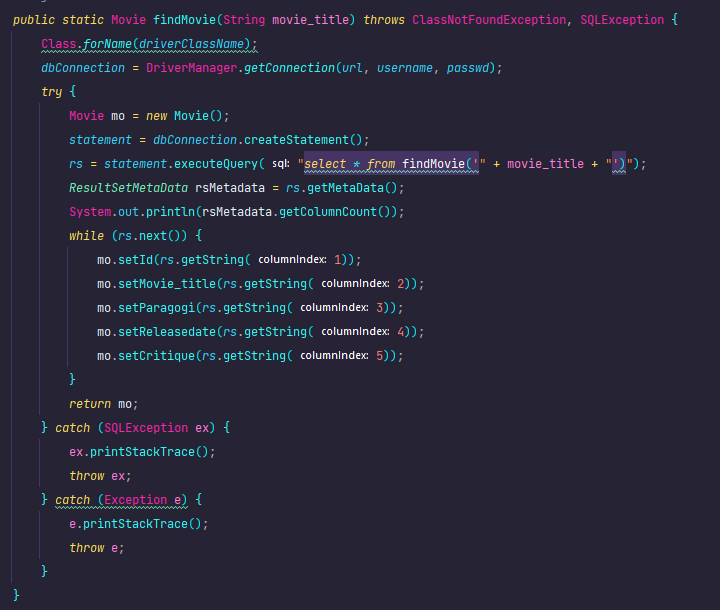


Οι λειτουργία των παραπάνω stored functions είναι ότι φιλτράρει τα δεδομένα που εμφανίζονται στο UI στο TableView<Directors> με βάση την ηλικία που εισάγει ο χρήστης στο ειδικό text field.

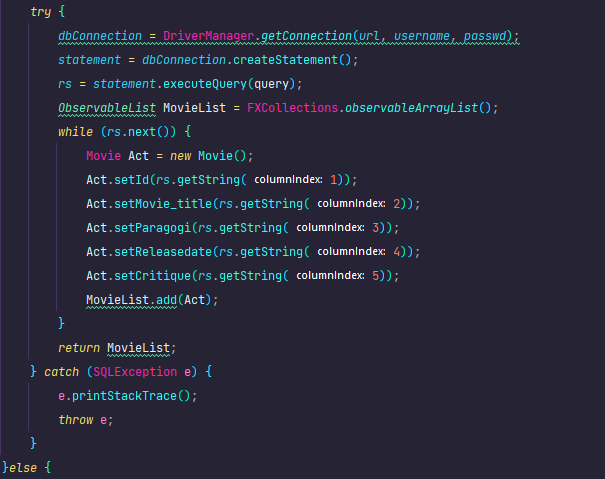
Η μέθοδος findMovie υλοποιεί το stored function findMovie η οποία ελέγχει αν υπάρχει η συγκεκριμένη ταινία στον πίνακα και λόγο της βάσης χρησιμοποιείται σε όλες τις κλάσεις DAO του κώδικα, καθώς όλοι οι πίνακες (εκτός του πίνακα Movie) έχουν σαν δευτερεύων κλειδί το movie\_title.

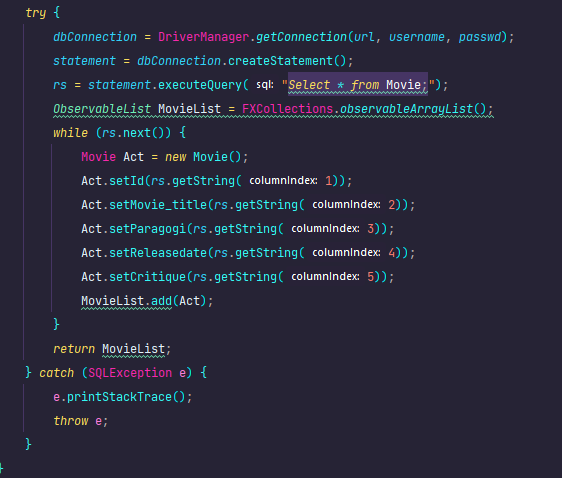


Η μέθοδος filterBy της κλάσης MovieDAO υλοποιεί το dynamic query filterMovie το οποίο φιλτράρει την χώρα παραγωγής της ταινίας. Για παράδειγμα η επιλογή Usa φιλτράρει και εμφανίζει μέσα από τον πίνακα Movie τις χώρες που έχουν αμερικάνικη παραγωγή.







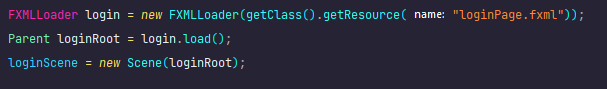


Στην κλάση Main αρχικά δημιουργούμε αντικείμενα Scene για κάθε σκηνή της εφαρμογή και παίρνουμε το αντικείμενο primaryStage που επιστρέφει η start().

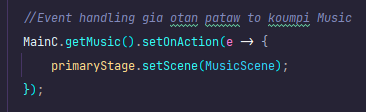
https://cdn.discordapp.com/attachments/765517337913851914/1047874636278411354/Screenshot_39.png

https://cdn.discordapp.com/attachments/765517337913851914/1047874717811495023/Screenshot_40.png

Στην παρακάτω εικόνα βλέπουμε τον τρόπο με τον οποίο φορτώνονται τα fxml αρχεία.

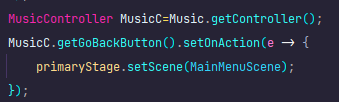


Μέσο των αντικειμένων των κλάσεων Controller διαχειριζόμαστε το ποια σκηνή θέλει να επισκεφθεί ο χρήστης.

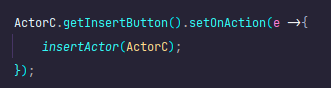


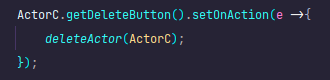
Τέλος, υπάρχουν event handlers μέθοδοι για κάθε δράση που επιθυμεί να κάνει ο χρήστης. Συγκεκριμένα :

* Για κάθε κουμπί που επιστρέφει στην αρχική σκηνή.



* Για κάθε κουμπί που εισάγει/διαγράφει δεδομένα.

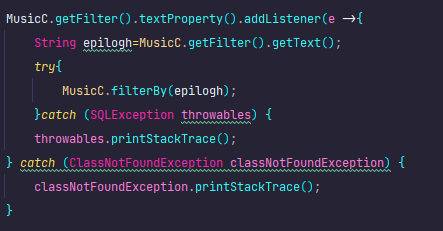




* Για κάθε RadioButton που φιλτράρει τα δεδομένα που εμφανίζονται στους πίνακες.



* Τέλος για κάθε TextField που φιλτράρει τα δεδομένα που εμφανίζονται στους πίνακες.



Η κλάση Log είναι η μοναδική που συνδυάζει την σύνδεση της βάσης και την επεξεργασία των αντικειμένων. Η συγκεκριμένη κλάση αποτελείται από 2 μεθόδους, την getAllRecords όπου υλοποιεί το dynamic query displaylog\_mo() και την getLogObjects η οποία βάζει τα στοιχεία της βάσης στον πίνακα του UI.

