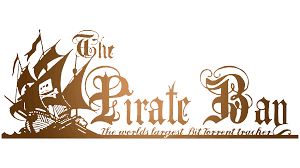
**ΟΜΑΔΙΚΗ ΕΡΓΑΣΙΑ**

**ΤΕΧΝΟΛΟΓΙΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ**



Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, ΔΙ.ΠΑ.Ε.

**Γρηγοριάδης Βαγγέλης**

AM: 174921

vagri1999@yahoo.gr

**Πολυζωίδης Γιώργος**

AM: 134050

georgepol01@yahoo.com

**ΕΙΣΑΓΩΓΉ**

Pirate Bay, είναι μία εφαρμογή διαχείρισης δεδομένων για ένα Video Club. Τα δεδομένα αποθηκεύονται σε μία βάση δεδομένων SQL. Υπάρχει δυνατότητα προβολής, προσθήκης, αφαίρεσης και επεξεργασίας δεδομένων. Επιπλέον, έχουν χτιστεί μερικά δυναμικά ερωτήματα προβολής δεδομένων.

*Εικόνα 1: Logo της εφαρμογής*

**ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ**

**Πίνακες:**

Η βάση αποτελείται από 12 πίνακες. Οι πρώτοι 6 αποθηκεύουν τα δεδομένα χρήσιμα στον χρήστη της εφαρμογης:

1. **Video\_Clubs:** Αποθηκεύει όλα τα Video Club που βρίσκονται στην ίδια πόλη με χαρακτηριστικά (vid,address, phone) κωδικός καταστήματος, διεύθυνση και τηλέφωνο αντίστοιχα. Κύριο κλειδί του πίνακα είναι το vid.
2. 2)**Customers:** Αποθηκεύει όλους τους πελάτες που έχουν νοικιάσει τουλάχιστον μία φορά από κάποιο κατάστημα με χαρακτηριστικά (cid,cname,address,phone) κωδικός πελάτη, όνομα, διεύθυνση και τηλέφωνο αντίστοιχα. Κύριο κλειδί του πίνακα είναι το cid.
3. **Products:** Αποθηκεύει όλα τα προιόντα που διαθέτουν για ενοικίαση τα Video Club με χαρακτηριστικά (pid, ptype) κωδικός προιόντος και είδος αντίστοιχα. Κύριο κλειδί του πίνακα είναι το pid. Επίσης το πεδίο ptype μπορεί να πάρει μόνο 2 τιμές (game, movie).
4. **Movies:** Αποθηκεύει όλες τις ταινίες που διαθέτουν για ενοικίαση τα Video Club με χαρακτηριστικά (mid, title, director, genre, year, rating) κωδικός ταινίας, τίτλος, σκηνοθέτης, είδος, έτος κυκλοφορίας και βαθμολογία αντίστοιχα. Κύριο κλειδί του πίνακα είναι το mid το οποίο είναι αναφορά σε ξένο κλειδί (pid) του πίνακα products και απαγορεύεται να γίνει διαγραφή μίας ταινίας από τον πίνακα products (on delete restrict). Επίσης το πεδίο rating μπορεί να πάρει μόνο τιμές μεταξύ 1 και 10.
5. **Games**: Αποθηκεύει όλα τα παιχνίδια που διαθέτουν για ενοικίαση τα Video Club με χαρακτηριστικά (gid, title, console, genre, year, rating) κωδικός παιχνιδιού, τίτλος, κονσόλα, είδος, έτος κυκλοφορίας και βαθμολογία αντίστοιχα. Κύριο κλειδί του πίνακα είναι το gid το οποίο είναι αναφορά σε ξένο κλειδί (pid) του πίνακα products και απαγορεύεται να γίνει διαγραφή ενός παιχνιδιού από τον πίνακα products (on delete restrict). Επίσης το πεδίο rating μπορεί να πάρει μόνο τιμές μεταξύ 1 και 10 και το πεδίο console τις τιμές (PC, XBOX, PS4, PS5).
6. **Rentals**: Αποθηκεύει όλες τις ενοικίασεις που έχουν γίνει από τους πελάτες στα διάφορα Video Club με χαρακτηριστικά(cid, vid, pid, rdate) κωδικός πελάτη, κωδικός καταστήματος ,κωδικός προιόντος και ημερομηνία ενοικίασης αντίστοιχα. Κύριο κλειδί του πίνακα είναι ο συνδυασμός των ξένων κλειδιών (cid, vid, pid) με την ημερομηνία (rdate). Επίσης τα πεδία (cid, vid, pid) είναι αναφόρες σε ξένα κλειδιά των πινάκων customers, video\_clubs και products. Σε περίπτωση διαγραφής ενός καταστήματος από τον πίνακα video\_clubs αυτόματα θα γίνει και διαγράφη του στον πινάκα rentals, περιορισμός (on delete cascade).Τέλος απαγορεύεται να γίνει διαγραφή ενός πελάτη από τον πίνακα customers, όπως και ενός προιόντος από τον πίνακα products (on delete restrict).

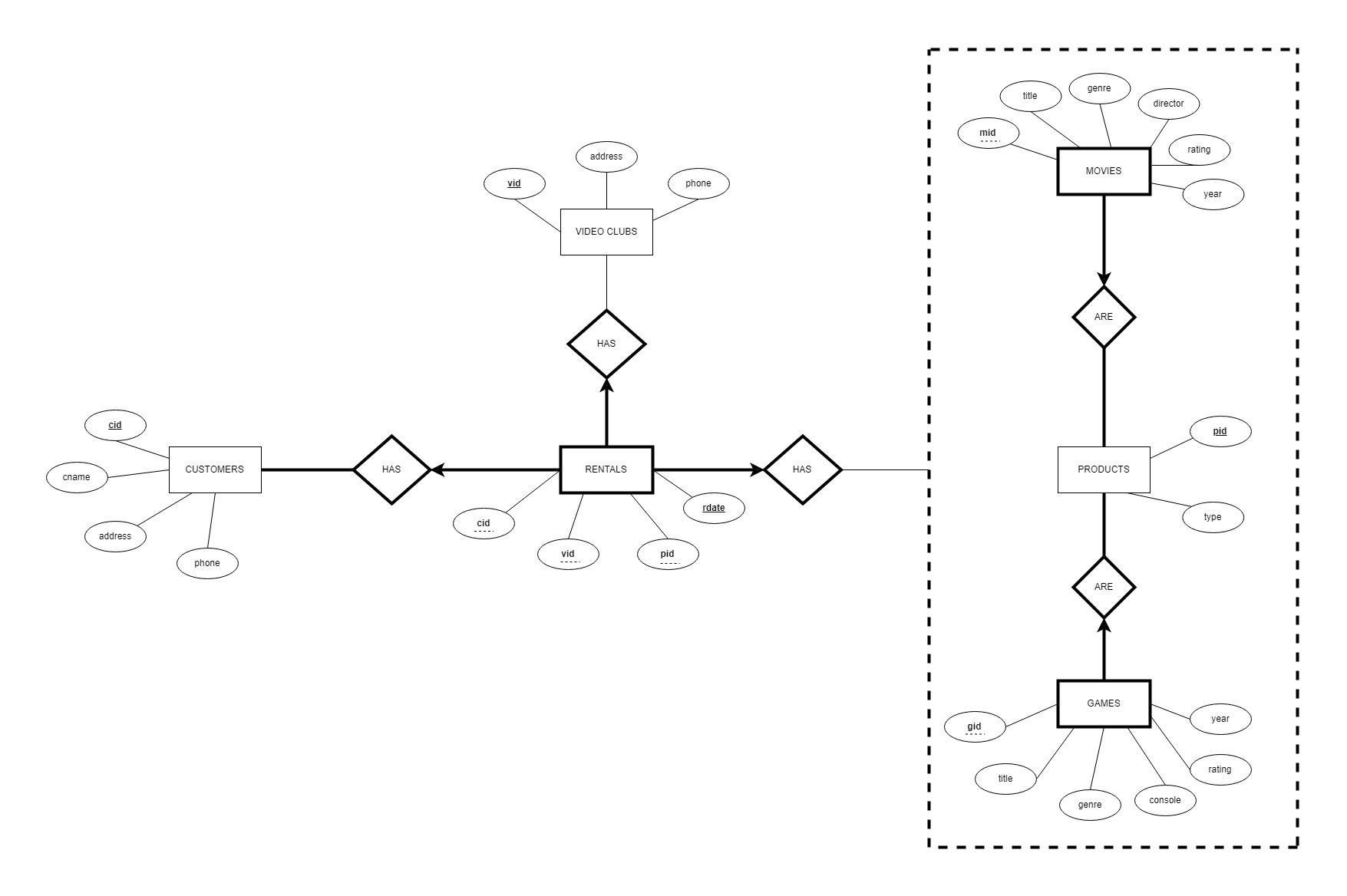
Οι επόμενοι 5 πίνακες υπάρχουν για την διατήρηση τον logs. Αποθηκεύουν προσθήκες, αφαιρέσεις και τροποποιήσεις δεδομένων μαζί με την ημερομηνία που έγιναν.

1. **Movies\_audit**: logs για αλλαγες στον πίνακα **Movies.**
2. **Games\_audit**: logs για αλλαγες στον πίνακα **Games.**
3. **Video\_clubs\_audit**: logs για αλλαγες στον πίνακα **Video\_Clubs.**
4. **Customers\_audit**:logs για αλλαγες στον πίνακα **Customers.**
5. **Rentals\_audit**: logs για αλλαγες στον πίνακα **Rentals.**

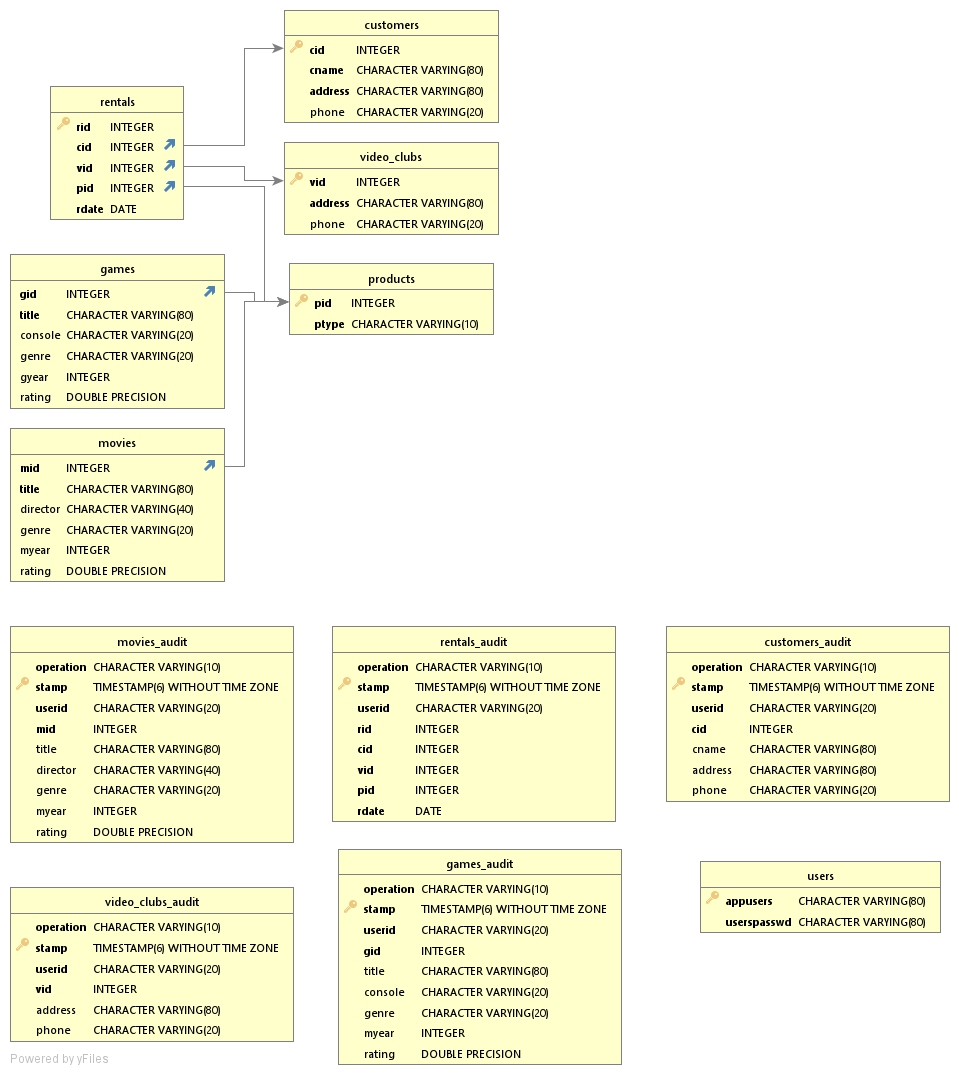
Οι τελευταίος πίνακας **Users** έχει τα δεδομένα των χρηστών της εφαρμογής.

**Restrictions:**

1. Ένας πελάτης μπορεί να νοικιάσει ένα ή περισσότερα προιόντα από ένα ή περισσότερα Video Club ταυτόχρονα ή και όχι.
2. Ένας πελάτης μπορεί να νοικιάσει το ίδιο προιόν από το ίδιο Video Club σε διαφορέτικές ημερομηνίες.
3. Ένα Video Club μπορεί να έχει κανέναν, έναν ή περισσότερους πελάτες.
4. Τα προιόντα που διαθέτει κάθε Video Club αποτελούνται από τουλάχιστον ένα παιχνίδι και τουλάχιστον μια ταινία.
5. Όλα τα Video Club έχουν μία κοινή αποθήκη όπου διαθέτουν τα προιόντα τους.
6. Όλα τα Video Club αποτελούν καταστήματα της ίδιας αλυσίδας (Pirate Bay) και βρισκόνται στην ίδια πόλη.

**ERs:**

*Εικόνα 2.1: Αρχική σχεδίαση διαγράμματος ER.*

****

*Εικόνα 2.1: Τελική σχεδίαση διαγράμματος ER.*

**WELCOME FRAME**

Η οθόνη log in, είναι η πρώτη οθόνη που εμφανίζεται. Χρησιμοποιείται για την σύνδεση του χρήστη με τον λογαριασμό του.

**ELEMENTS**

1) welcomeImageLabel: Εικόνα με το ονομα της εφαρμογής.

2) jlabel: Κείμενο “Καλωσορίσατε στο Pirate Bay!”.

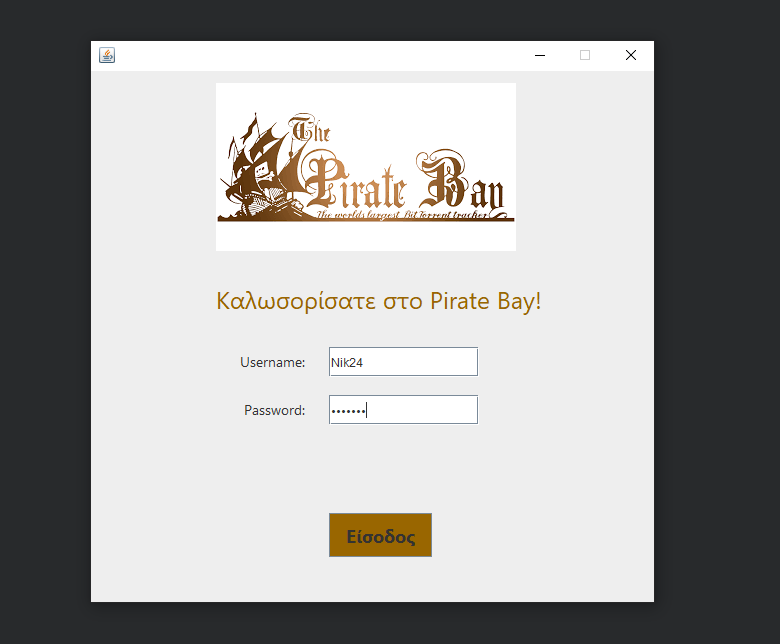
3) usernamelabel: Κείμενο “Username:”.

4) passwordlabel: Κείμενο “Password:”.

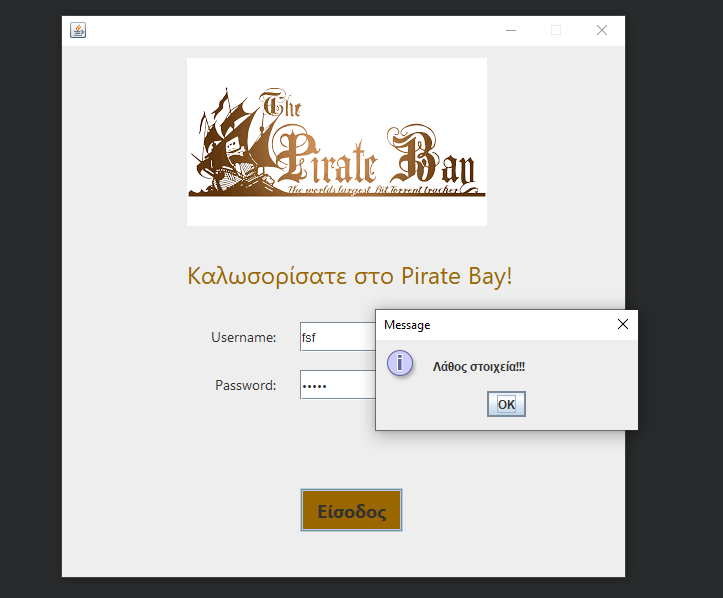
5) usernamefield: Textfield, οπου ο χρηστης γραφει το username του.

6) passwordfield: Passwordfield, οπου ο χρηστης γραφει το password του.

7) loginButton: Κουμπί, το οποίο ελέγχει αν τα στοιχεία ταιριάζουν σε κάποιον χρήστη. Αν ταιριάζουν, κλείνει το Login Frame, και ανοίγει το Main Frame. Αν δεν ταιριάζουν τα στοιχεία εμφανίζει Error Window, με κείμενο “Λάθος στοιχεία!!!”.



*Εικόνα 3.1: Welcome Frame της εφαρμογής.*

**

*Εικόνα 3.2: Περίπτωση λανθασμένων στοιχείων στο Welcome Frame της εφαρμογής.*

**MAIN FRAME**

Το Main παράθυρο εμφανίζεται με την χρήση του 7ου element(loginButton) του Login frame. Αυτή η οθόνη χρησιμοποιείται για την μετακίνηση του χρήστη στις διάφορες λειτουργίες της εφαρμογής. Η μετακίνηση από οθόνη σε οθόνη γίνεται με την χρήση των Panel.

**MAIN PANEL ELEMENTS**

1) logoImageLabel: Εικόνα με το logo της εφαρμογής.

2) welcomeImageLabel: Εικόνα με το ονομα της εφαρμογής.

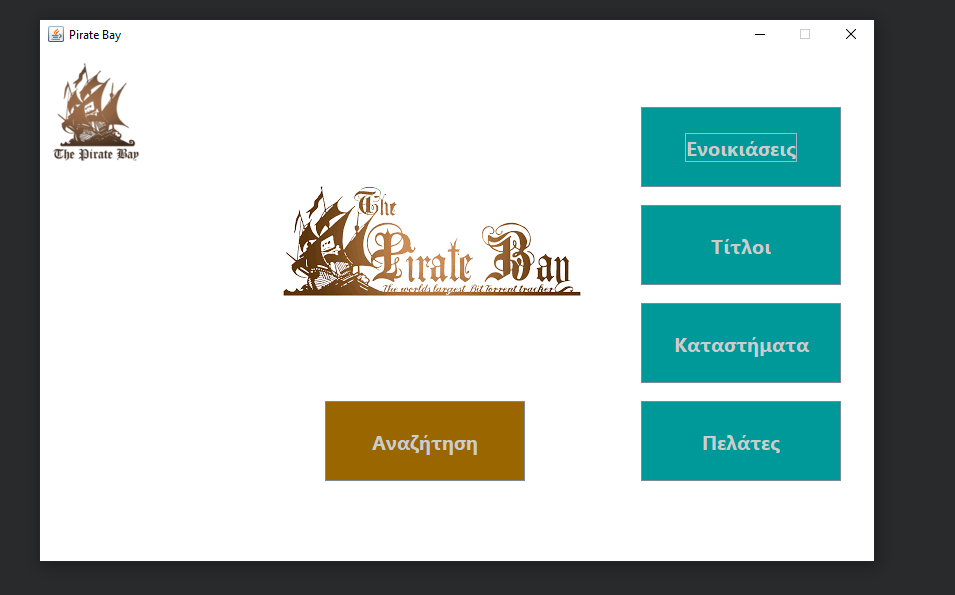
3) rentButton: Κουμπί με κείμενο “Ενοικίασεις” το οποίο μας μετακινεί στην οθόνη Rent.

4) storesButton: Κουμπί με κείμενο “Καταστήματα” το οποίο μας μετακινεί στην οθόνη Stores.

5) titlesButton: Κουμπί με κείμενο “Τίτλοι” το οποίο μας μετακινεί στην οθόνη Titles.

6) customersButton: Κουμπί με κείμενο “Πελάτες” το οποίο μας μετακινεί στην οθόνη Customers.

7) queryButton: Κουμπί με κείμενο “Αναζήτηση” το οποίο μας μετακινεί στην οθόνη Query.



*Εικόνα 4: Main Frame της εφαρμογής.*

**CUSTOMER, RENT AND STORES PANEL ELEMENTS**

Τα 3 αυτά Panel έχουν τα εξής παρόμοια elements:

1) logoImageLabel: Εικόνα με το Icon της εφαρμογής.

2) jLabel: Κείμενο που υποδεικνύει το Panel στο οποίο βρισκόμαστε.

3) jTable: Εμφανίζει όλα τα δεδομένα από την βάση δεδομένων. Τα δεδομένα θα εμφανιστούν όταν πρώτο ανοίξει η εφαρμογή και όποτε γίνεται κάποια αλλαγή στα δεδομένα.

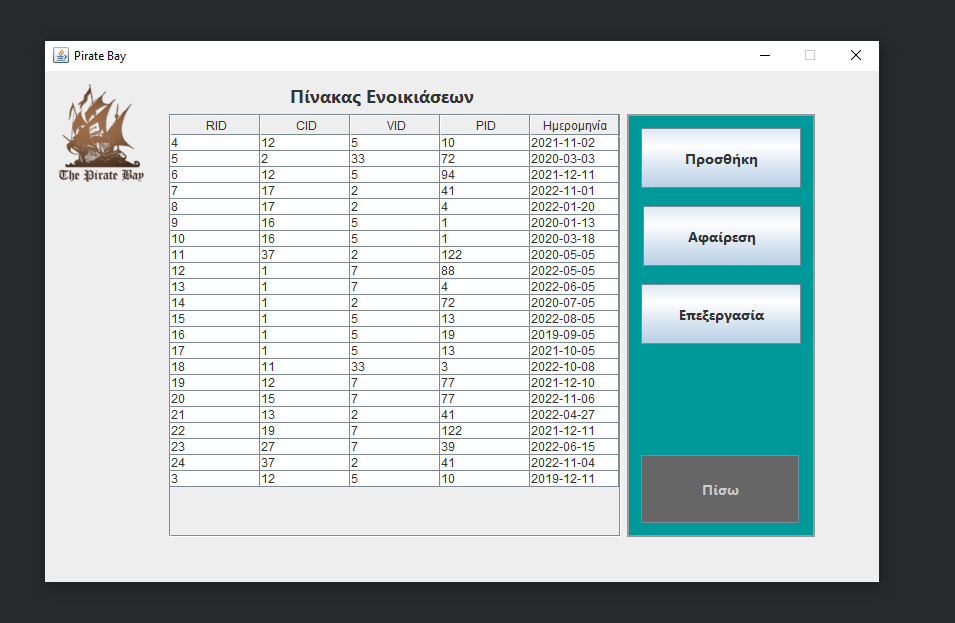
4) buttonPanel: Περιέχει όλα τα Buttons.

5) addButton: Κουμπί με κείμενο “Προσθήκη” το οποίο ανοίγει το αντίστοιχο παράθυρο AddFrame, ανάλογα με το Panel που βρισκόμαστε.

6) removeButton: Κουμπί με κείμενο “Αφαίρεση” το οποίο ανοίγει το αντίστοιχο παράθυρο RemoveFrame, ανάλογα με το Panel που βρισκόμαστε.

7) editButton: Κουμπί με κείμενο “Επεξεργασία” το οποίο ανοίγει το αντίστοιχο παράθυρο SearchFrame, ανάλογα με το Panel που βρισκόμαστε.

8) backButton: Κουμπί με κείμενο “Πίσω στην αρχική” το οποίο μας μετακινεί πίσω στην οθόνη Main.

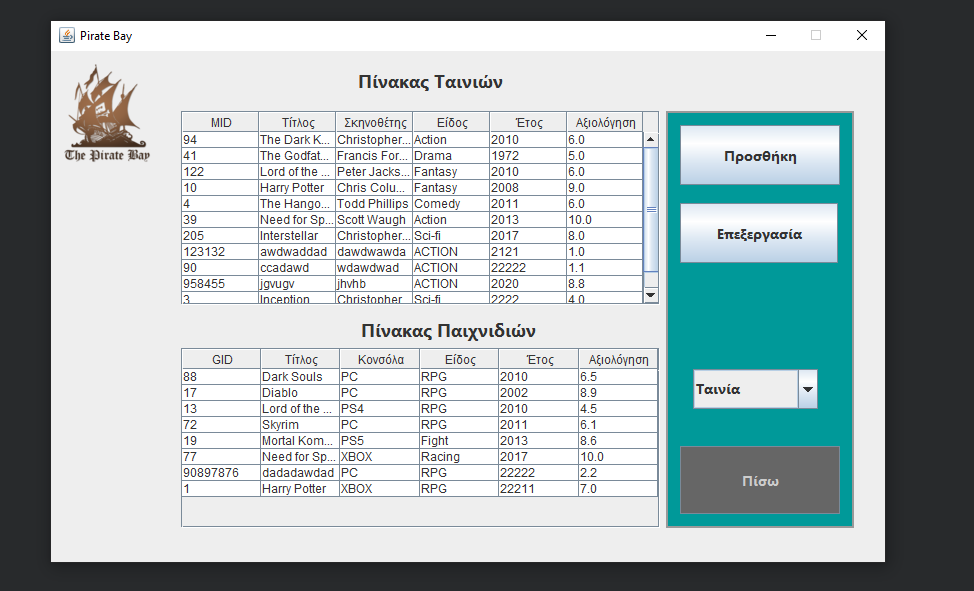


*Εικόνα 5: Rent Panel της εφαρμογής.*

**TITLES PANEL ELEMENTS**

Τo Panel αυτό είναι πανομοιότυπο με τα προηγούμενα τρία. Η διαφορά βρίσκεται στο γεγονός ότι έχουμε δύο είδη τίτλων, movies και games. Για αυτό τον λόγο, υπάρχουν 2 διαφορετικοί πίνακες και ένα καινούργιο element:

titleTypeComboBox: Επιλογές: Ταινίες, Παιχνίδια. Αλλάζοντας την επιλογή αυτή, όταν πατηθεί ένα κουμπί, θα ανοίξει το αντίστοιχο παράθυρο. Για παράδειγμα, αν επιλεχθεί στο comboBox η επιλογη “Ταινίες” και πατηθεί το κουμπί προσθήκη, θα ανοίξει το παράθυρο προσθήκης ταινίας και όχι παιχνιδιού.



*Εικόνα 6: Titles Panel της εφαρμογής.*

**QUERY PANEL ELEMENTS**

1) logoImageLabel: Εικόνα με το logo της εφαρμογής.

2) jLabel: Κείμενο “Αναζήτηση στην Βάση”.

3) jLabel: Κείμενο “Επιλέξτε ένα ερώτημα: ”.

4) questionComboBox: Μέσο αυτού του element, γίνεται η επιλογή ανάμεσα σε 10 διαφορετικές ερωτήσεις, με δυναμικές μεταβλητές.

5) labelans: Label, στο οποίο εμφανίζεται η επιλεγμένη ερώτηση από τό questionComboBox.

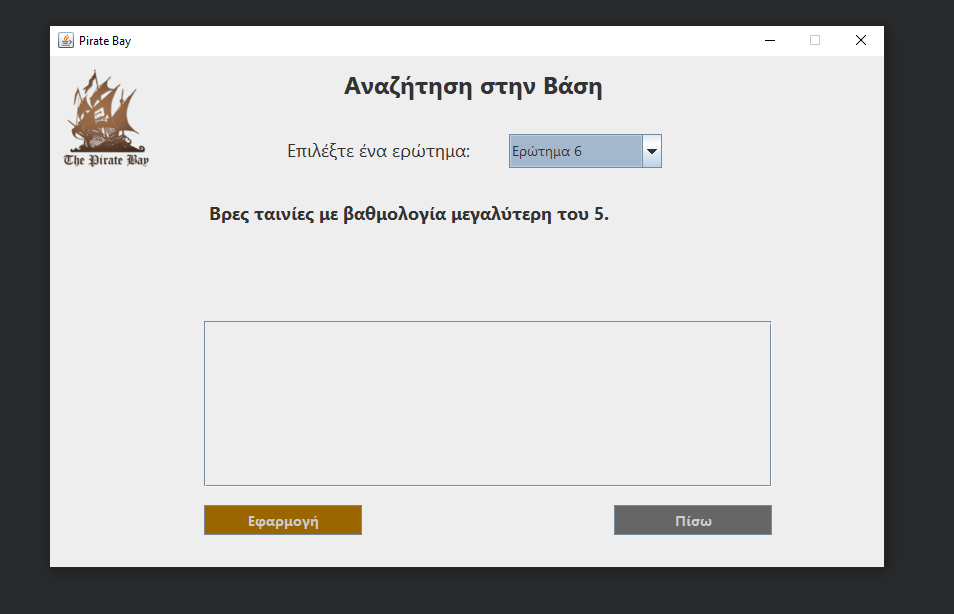
6) variableField: Field το οποίο ο χρήστης θα χρησιμοποιήσει για να αλλάξει την μεταβλητή της ερώτησης.

7) answerComboBox: Εναλλακτικό στοιχείο του variableField, για μεταβλητες οι οποιες ειναι πιο συγκεκριμένες και όχι τόσο γενικές.

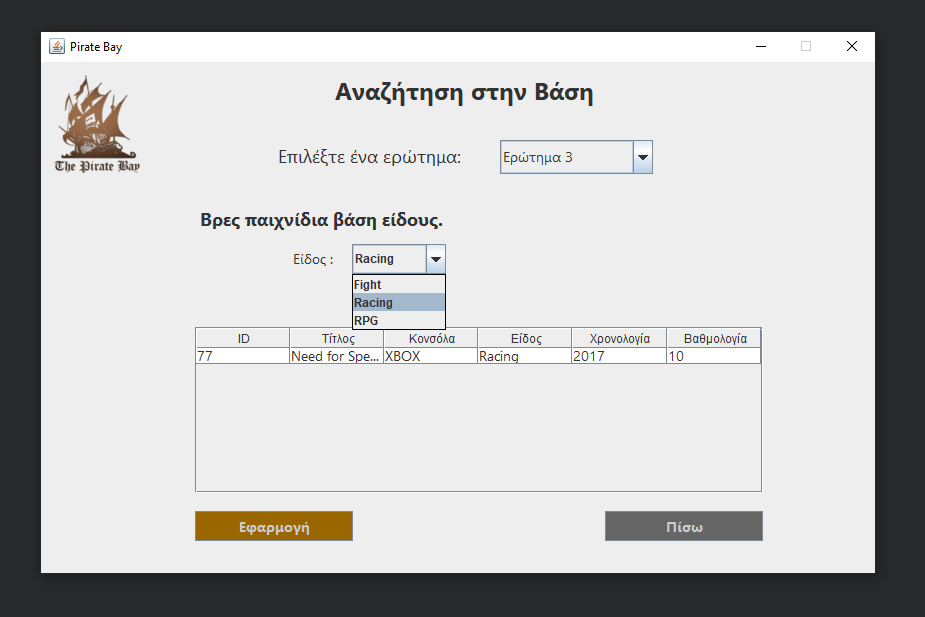
8) answerTable: Εδώ εμφανίζεται η απάντηση της ερώτησης

9) answerButton: Κουμπί με κείμενο “Εφαρμογή”, το οποίο αφού πατηθεί, παίρνει τα δεδομένα από τα προηγούμενα elements και εμφανίζει το αποτέλεσμα στο answerTable.

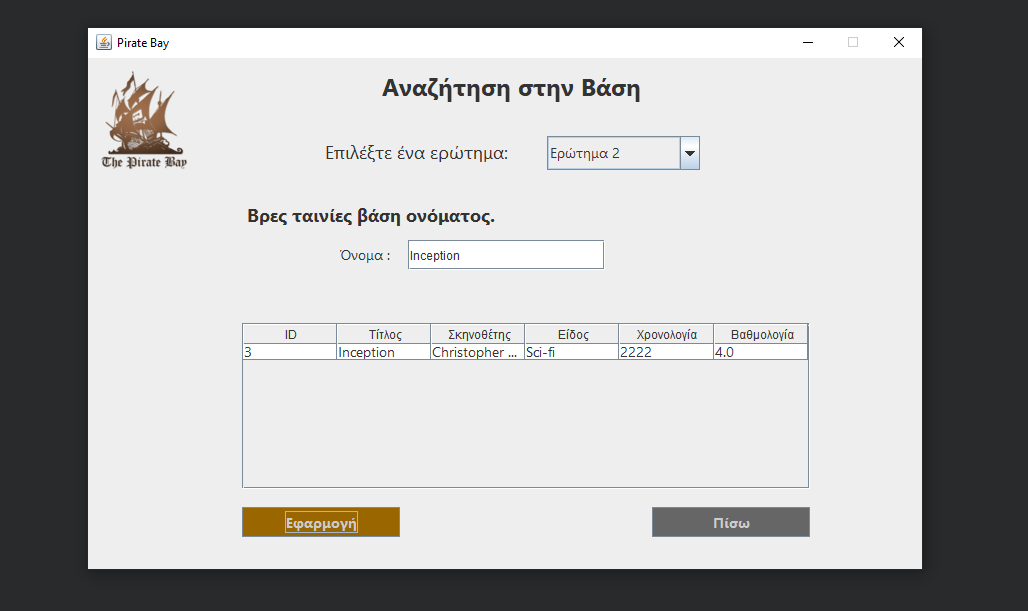
10) backButton: Κουμπί με κείμενο “Πίσω στην αρχική” το οποίο μας μετακινεί πίσω στην οθόνη Main.

****

*Εικόνα 7.1: Query Panel της εφαρμογής.*

**

*Εικόνα 7.2: Query Panel της εφαρμογής, με δυναμική επιλογή combobox.*

****

*Εικόνα 7.3: Query Panel της εφαρμογής, με δυναμική επιλογή textfield.*

**ADD FRAMES**

Το Add παράθυρο εμφανίζεται με την χρήση του addButton και χρησιμοποιείται για την προσθήκη δεδομένων, από την βάση δεδομένων.

**ELEMENTS**

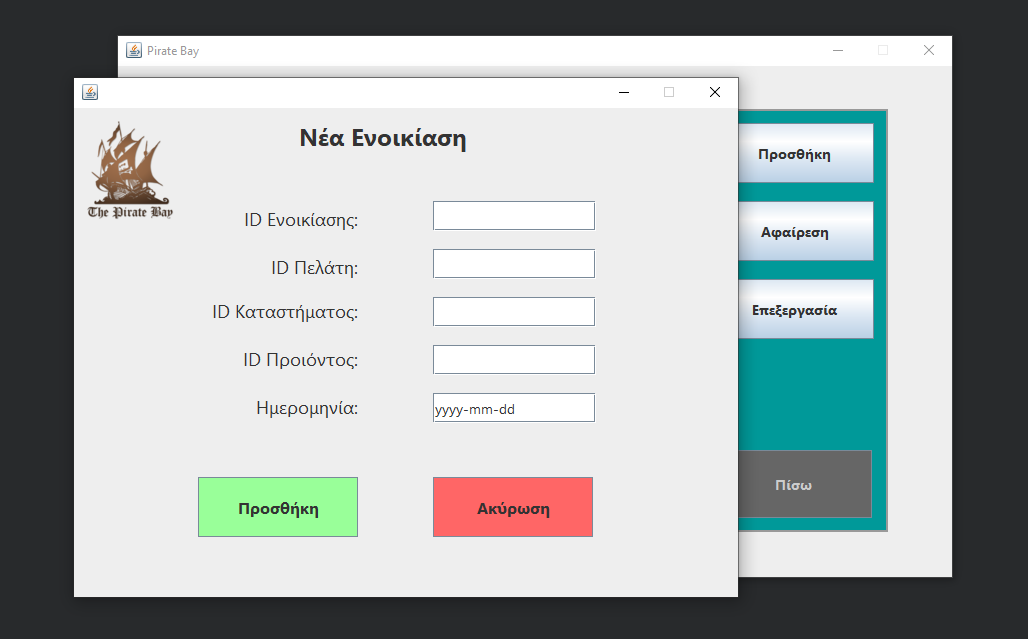
1) logoImageLabel: Εικόνα με το logo της εφαρμογής.

2) jLabel: Κείμενο που επιδεικνύει την οθόνη που βρίσκεται ο χρήστης.

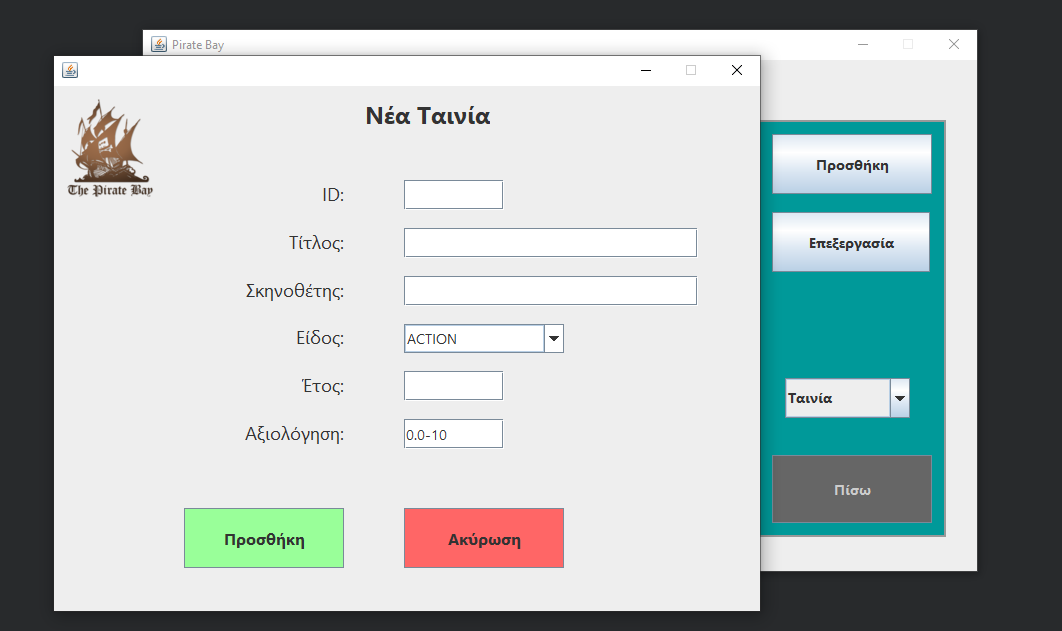
Στην συνεχεία ακολουθούν μια σειρά από labels, text fields και combo boxes για την εισαγωγή των δεδομένων.

3) confirmButton: Κουμπί με κείμενο “Προσθήκη” το οποίο προσθέτει τα δεδομένα που είχαν γραφεί στα προηγούμενα labels. Στην συνέχεια, εμφανίζει μήνυμα για την επιτυχία της πράξης αυτής, κλείνει το παράθυρο και ενημερώνει τον πίνακα του προηγούμενου παραθύρου.

4)cancelButton: Κλείνει το παράθυρο, χωρίς να αποθηκεύει δεδομένα.



*Εικόνα 8.1: Add Frame, από το Rent Panel.*



*Εικόνα 8.2: Add Frame, από το Titles Panel.*

**REMOVE FRAMES**

Το Remove παράθυρο εμφανίζεται με την χρήση του removeButton και χρησιμοποιείται για την αφαίρεση δεδομένων, από την βάση δεδομένων.

**ELEMENTS**

1) logoImageLabel: Εικόνα με το logo της εφαρμογής.

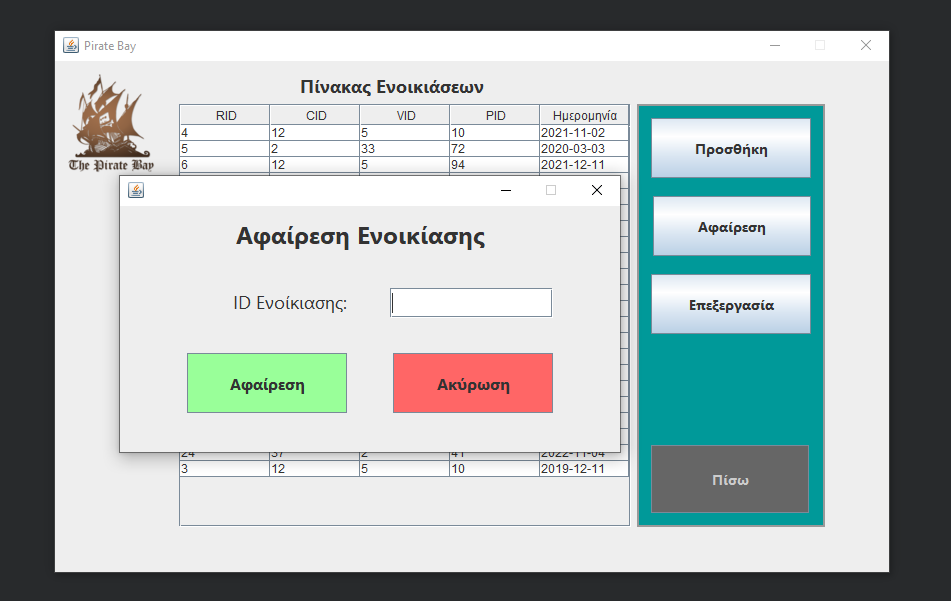
2) jLabel: Κείμενο που επιδεικνύει την οθόνη που βρίσκεται ο χρήστης.

3) jLabel: Κείμενο που ζητάει το ID του δεδομένου που θα αφαιρεθεί.

4) IDField: Text field στο οποίο ο χρήστης συμπληρώνει το ID του δεδομένου που επιθυμεί να διαγράψει.

5) confirmButton: Κουμπί με κείμενο “Αφαίρεση” το οποίο αφαιρεί τα δεδομένα με το ID που έγραψε ο χρήστης. Στην συνέχεια, εμφανίζει μήνυμα για την επιτυχία της πράξης αυτής, κλείνει το παράθυρο και ενημερώνει τον πίνακα του προηγούμενου παραθύρου.

6)cancelButton: Κλείνει το παράθυρο, χωρίς να αφαιρέσει δεδομένα.



*Εικόνα 9: Remove Frame, από το Rent Panel.*

**SEARCH FRAMES**

Το Search παράθυρο εμφανίζεται με την χρήση του editButton και χρησιμοποιείται για την αναζήτηση δεδομένων, έτσι ώστε να μπορέσουν να επεξεργαστούν από τον χρήστη, στην οθόνη Edit.

**ELEMENTS**

1) logoImageLabel: Εικόνα με το logo της εφαρμογής.

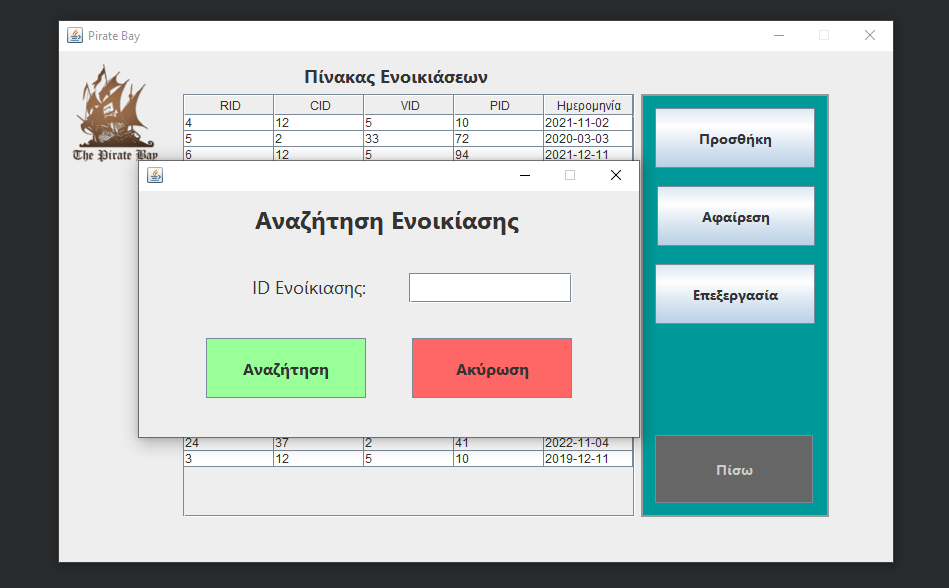
2) jLabel: Κείμενο που επιδεικνύει την οθόνη που βρίσκεται ο χρήστης.

3) jLabel: Κείμενο που ζητάει το ID του δεδομένου που θα γίνει επεξεργασία.

4) IDField: Text field στο οποίο ο χρήστης συμπληρώνει το ID του δεδομένου που επιθυμεί να επεξεργαστεί.

5) confirmButton: Κουμπί με κείμενο “Αναζήτηση” το οποίο βρίσκει τα δεδομένα με το ID αυτό . Στην συνέχεια, εμφανίζει μήνυμα για την επιτυχία της πράξης αυτής, κλείνει το παράθυρο αυτό και ανοίγει το παράθυρο Edit, στο οποίο βάζει όλα τα δεδομένα.

6)cancelButton: Κλείνει το παράθυρο.



*Εικόνα 10: Search Frame, από το Rent Panel.*

**EDIT FRAMES**

Το Edit παράθυρο εμφανίζεται με την χρήση του confirmButton στην οθόνη Search και χρησιμοποιείται για την επεξεργασία των δεδομένων, που έχουν βρεθεί με την χρήση του ID, στην προηγούμενη οθόνη.

**ELEMENTS**

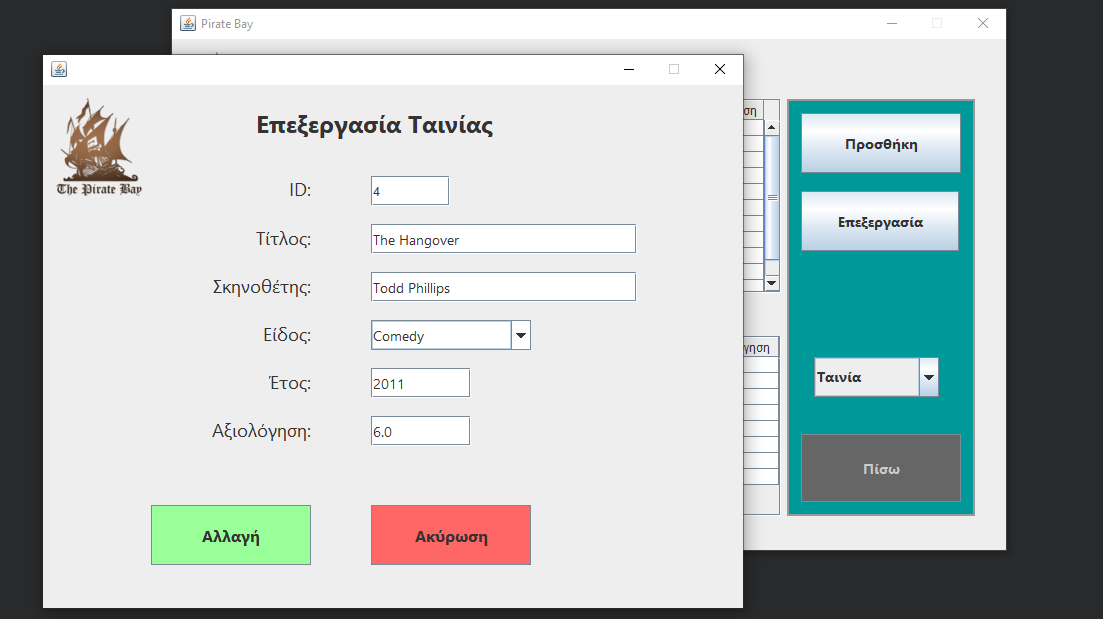
1) logoImageLabel: Εικόνα με το logo της εφαρμογής.

2) jLabel: Κείμενο που επιδεικνύει την οθόνη που βρίσκεται ο χρήστης.

Στην συνεχεία ακολουθούν μια σειρά από labels, text fields και combo boxes γεμισμένα με τα δεδομένα που βρέθηκαν. Τα δεδομένα αυτά μπορεί ο χρήστης να μεταβάλει.

3) confirmButton: Κουμπί με κείμενο “Αλλαγή” το οποίο μεταβάλλει τα δεδομένα που είχαν γραφεί στα προηγούμενα labels. Στην συνέχεια, εμφανίζει μήνυμα για την επιτυχία της πράξης αυτής, κλείνει το παράθυρο και ενημερώνει τον πίνακα του προηγούμενου παραθύρου.

4)cancelButton: Κλείνει το παράθυρο, χωρίς να αποθηκεύει δεδομένα

**

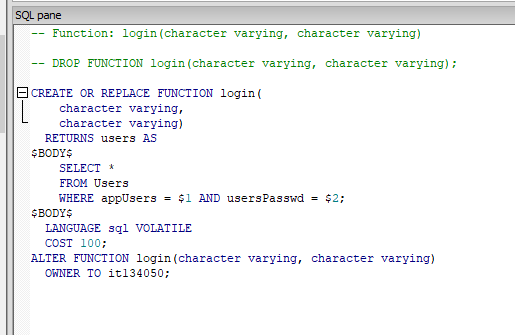
*Εικόνα 11: Edit Frame, μετά από το Search Frame, του Titles Panel.*

**TRIGGERS - PROCEDURES - DB FUNCTIONS**

Στην βάση δεδομένων, υπάρχουν πάνω από 40 Functions, που εκτελούν διάφορες εργασίες εντός της εφαρμογής, όπως προσθήκη, αναζήτηση, αφαίρεση και επεξεργασία δεδομένων και 5 Triggers που χρησιμοποιούνται για την αποθηκευση των log files. Τα Functions εχουν χρησιμοποιηθεί με την χρήση των Prepared Statements, για να γίνεται απόκρυψη των ευαίσθητων πληροφοριών της βάσης δεδομένων μέσα από τον κώδικα της java.

**ΠΑΡΑΔΕΙΓΜΑ 1 (LOGIN FUNCTION)**

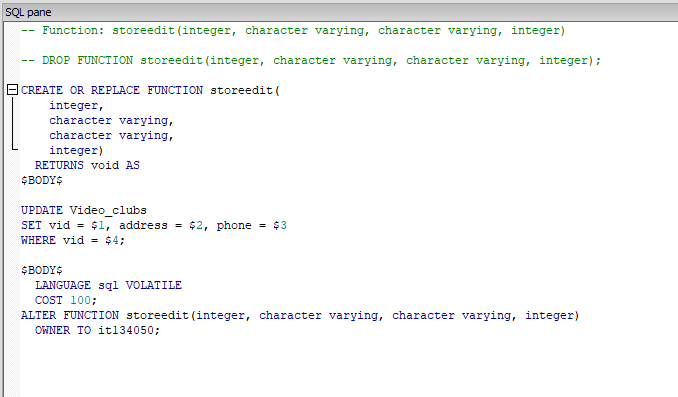
Για την διαδικασία Log in, γίνεται η χρήση του function login, το οποίο ελέγχει αν τα στοιχεία που δόθηκαν, ταιριάζουν σε κάποιο χρήστη από τον πίνακα **users**. Αν ταιρίαζουν, ο χρήστης μπορεί να προχωρήσει, ενώ αν δεν ταιρίαζουν, βγαίνει κατάλληλο μύνημα (εικόνα 3.2).



*Εικόνα 12: Function login.*

**ΠΑΡΑΔΕΙΓΜΑ 2 (STORE EDIT FUNCTION)**

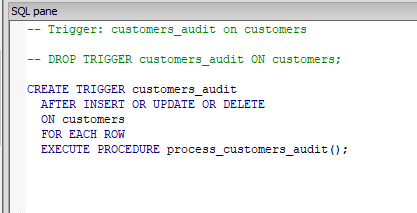
Σε όλη την εφαρμογή, βλέπουμε παραδείγματα χρήσης function σαν αυτό. Στην προκειμένη περίπτωση, βρισκόμασταν στο **Store Panel**, και πατήσαμε το κουμπί με κείμενο “Επεξεργασία”. Μετά από την αναζήτηση, καλείται το function storeedit, το οποίο διαχειρίζεται τα στοιχεία του πίνακα **Video\_clubs**.



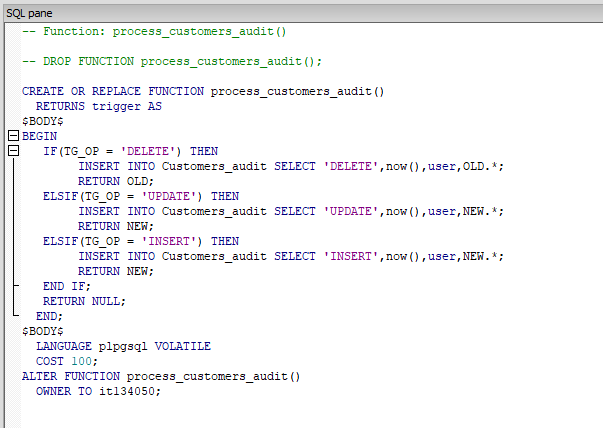
*Εικόνα 13: Function storeedit.*

**ΠΑΡΑΔΕΙΓΜΑ 3 (LOG FILE TRIGGER AND PROCEDURE)**

Κάνοντας, οποιαδήποτε αλλαγή στην βάση δεδομένων, τα log files ενημερώνονται αναλόγως. Στο παράδειγμα αυτό, όποτε γίνεται αλλαγή στον πίνακα **Customers**, ενεργοποιείται το trigger customers\_audit, το οποίο με την σειρά του, ενεργοποιεί το procedure process\_customer\_audit, το οποίο προσθέτει στον πίνακα **Customers\_audit**, τις αλλαγές που έγιναν στον πίνακα **Customers**.



*Εικόνα 14.1: Trigger Customers\_audit.*

**

*Εικόνα 14.2: Procedure Customers\_audit.*

**ΕΡΓΑΛΕΙΑ**

Παρακάτω εμφανίζονται όλα τα εργαλεία που χρησιμοποιήθηκαν για την κατασκευή της εργασίας.

Στα αρχικά στάδια χρησιμοποιήθηκε το Diagram Editor για την των ER, σε συνδυασμο με το DB Visualizer. <https://www.diagrameditor.com/>, https://www.dbvis.com/.

Στο περιβάλλον της javaswing, με την χρήση του Apache NetBeans, κατασκευάστηκε η εφαρμογή. https://netbeans.apache.org

Το κομμάτι της SQL, αναπτύχθηκε με το pgAdmin. https://www.pgadmin.org/download/