

# Κατανεμημένα Συστήματα: Ομαδική Εργασία

*Αβραάμ Κατσίγκας 321/2015087*

*Αντώνιος Πίττας 321/2017158*

Ζητούμενο 1<sup>ο</sup>:

α)

Ο βασικός σχεδιασμός της εφαρμογής έχει ως εξής: Ο χρήστης πριν τη σύνδεσή του στο σύστημα, αποκτά πρόσβαση στο απομακρυσμένο αντικείμενο του εξυπηρετητή, μέσω της διεπαφής που ακολουθεί. Έπειτα συνδέεται στο σύστημα φτιάχνοντας λογαριασμό (επιλέγει το ρόλο που θέλει διαχειριστής/χρήστης) και μέσω της γραφικής διεπαφής χειραγωγεί το απομακρυσμένο αντικείμενο. Παράλληλα, όσον αφορά τα θεάματα, αυτά τα διαχειρίζεται η βάση δεδομένων, με την οποία επικοινωνεί το αντικείμενο του χρήστη, από σχεδιασμό. Όποτε χρειάζεται να επικοινωνήσει ο εξυπηρετητής με την βάση το αντικείμενο παράγει το κατάλληλο μήνυμα, ακολουθώντας τους κανόνες του πρωτοκόλλου. Η χρήση νημάτων διευκόλυνε πολύ αυτή τη διαδικασία, μιας και τα μηνύματα που παράγει η βάση ή ο εξυπηρετητής συσχετίζονται και δεν περιπλέκονται οι διαδικασίες των δύο.

β)

Επειδή το απομακρυσμένο αντικείμενο που χρησιμοποιεί ο χρήστης διαχειρίζεται οτιδήποτε αφορά τον χρήστη, συμπεριλαμβανομένων των θεαμάτων, το αντικείμενο αναλαμβάνει να δημιουργεί τα αντικείμενα με τα οποία επικοινωνούν ο εξυπηρετητής με την βάση. Έπειτα το αντικείμενο τα στέλνει στην πλευρά του εξυπηρετητή και λαμβάνει την απάντηση. Ουσιαστικά, το αντικείμενο το οποίο διαχειρίζεται την επικοινωνία βάσης – εξυπηρετητή είναι μέρος του αντικειμένου.

γ)

Ο χρήστης ξεκινά με δημιουργία λογαριασμού. Συμπληρώνει τα στοιχεία και επιλέγει αν ο λογαριασμός είναι προορισμένος για διαχειριστή ή όχι. Ο διαχειριστής μπορεί να δημιουργεί θεάματα, να προσθέτει παραστάσεις και να διαγράφει τα θεάματα. Ο απλός χρήστης μπορεί να αναζητήσει θεάματα με βάση τον τίτλο τους ή το είδος τους και έπειτα να κάνει κράτηση εισιτηρίων.

Για τη δημιουργία θεάματος ο διαχειριστής απλώς επιλέγει το κουμπί “Add Event” και συμπληρώνει τον τίτλο και το είδος του θεάματος. Έπειτα προωθείται στην οθόνη που προσθέτει μία παράσταση σε αυτό το θέαμα. Συμπληρώνει τα στοιχεία και συνεχίζει. Ο διαχειριστής έχει τη δυνατότητα να προσθέσει όσες παραστάσεις θέλει στο ίδιο θέαμα, αρκεί να μην συμπίπτουν η μία με την άλλη. Μπορεί επίσης να διαγράψει όποτε θέλει ένα θέαμα πατώντας το αντίστοιχο κουμπί και συμπληρώνοντας το όνομα του θεάματος.

Για την αναζήτηση θεαμάτων ο χρήστης επιλέγει το κουμπί “Search Events”. Εκεί επιλέγει αν θέλει να αναζητήσει με βάση το όνομα ή το είδος. Αφού κάνει την επιλογή συμπληρώνει την λέξη-κλειδί και του εμφανίζονται τα θεάματα, αν υπάρχουν. Έπειτα μπορεί να συμπληρώσει τα στοιχεία στα δεξιά της οθόνης για να κάνει μία κράτηση. Αφού εισάγει τα στοιχεία επιβεβαιώνει την κράτηση εισάγοντας τον αριθμό της κάρτας του και το ονοματεπώνυμό του και γίνεται η κράτηση.

δ)

#### Σενάριο Διαχειριστή:

The screenshot shows a 'Sign Up' window with the following fields and elements:

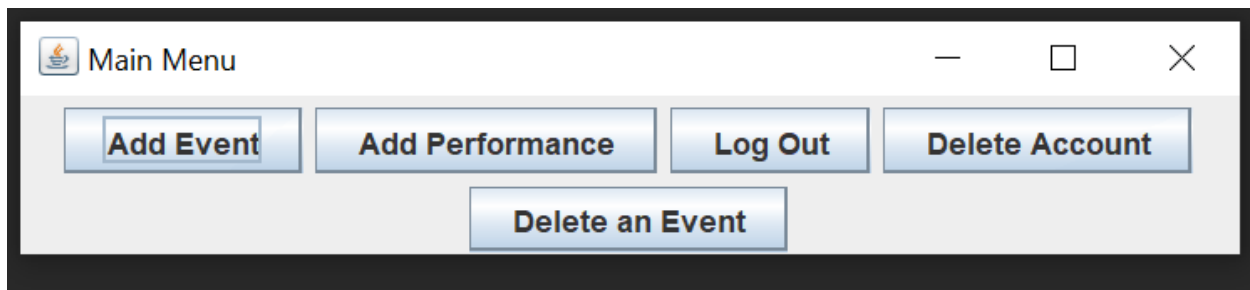
- User name:** Text box containing 'avraam'.
- Password:** Text box with masked characters (\*\*\*\*\*). A 'Show' button is to its right.
- Confirm password:** Text box containing 'katanemi'.
- Avraam Katsigras** (Name field).
- Phone:** Text box containing '2273003722'.
- ☒ **Admin:** checkbox.
- Sign Up** button.
- Success Message:** A modal dialog box with an information icon, the text 'User avraam signed up successfully!', and an 'OK' button.

Δημιουργία λογαριασμού διαχειριστή

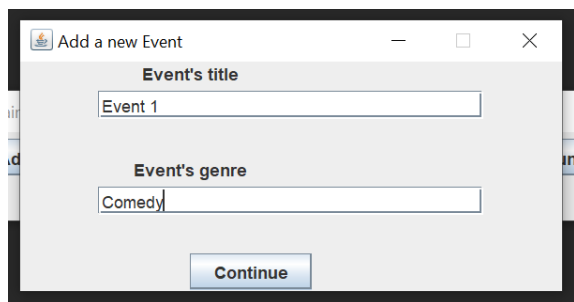
The screenshot shows a 'Log in' window with the following fields and elements:

- User name:** Text box containing 'avraam'.
- Password:** Text box with masked characters (\*\*\*\*\*). A 'Show' button is to its right.
- Log In** and **Sign Up** buttons.
- Success Message:** A modal dialog box with an information icon, the text 'Log in successful!', and an 'OK' button.

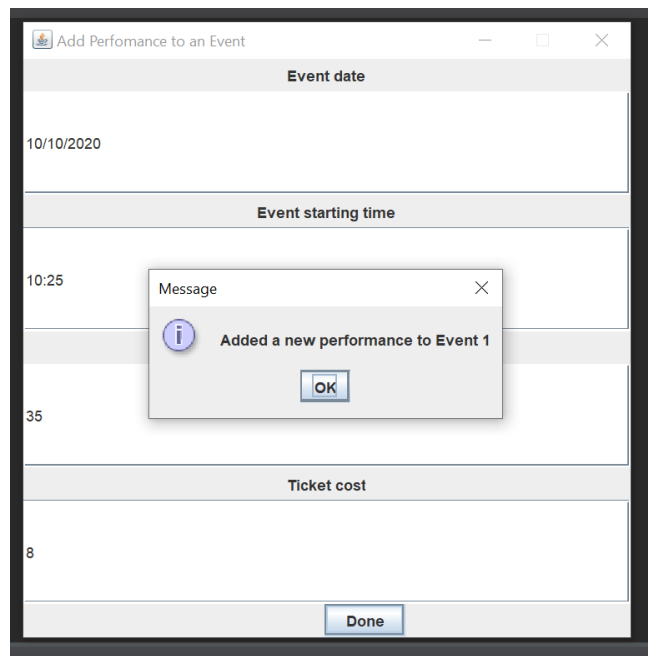
Σύνδεση διαχειριστή



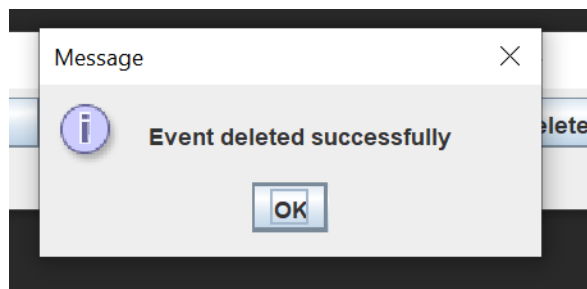
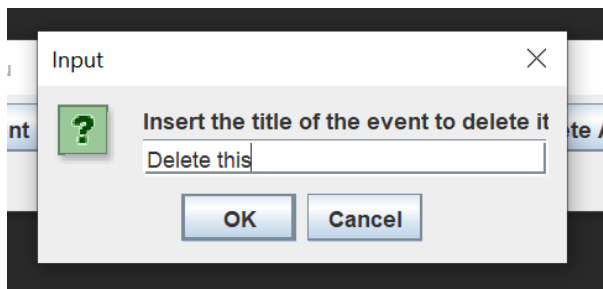
Βασική οθόνη διαχειριστή



Δημιουργία θεάματος

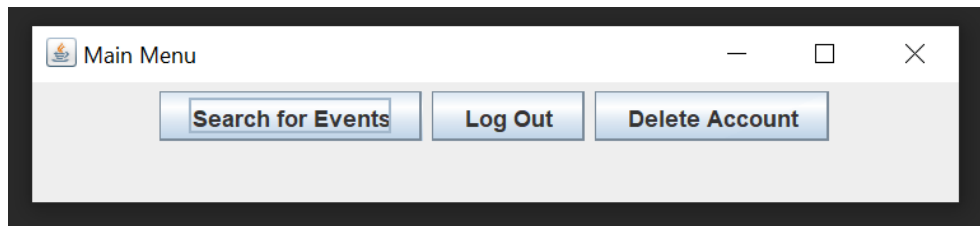


Δημιουργία παράστασης

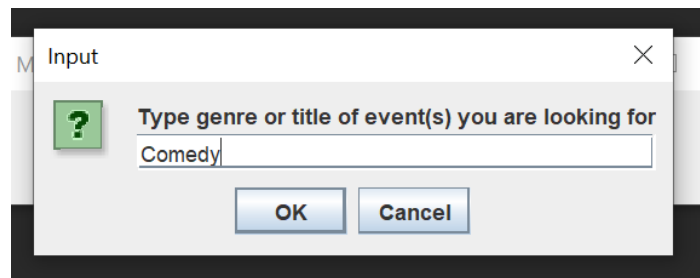
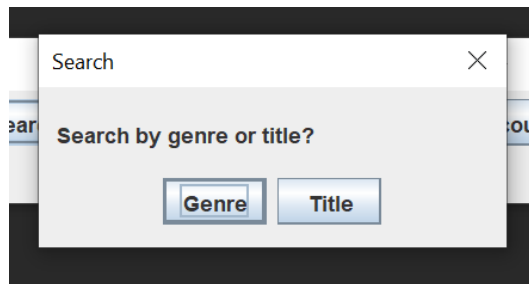


Διαγραφή θεάματος

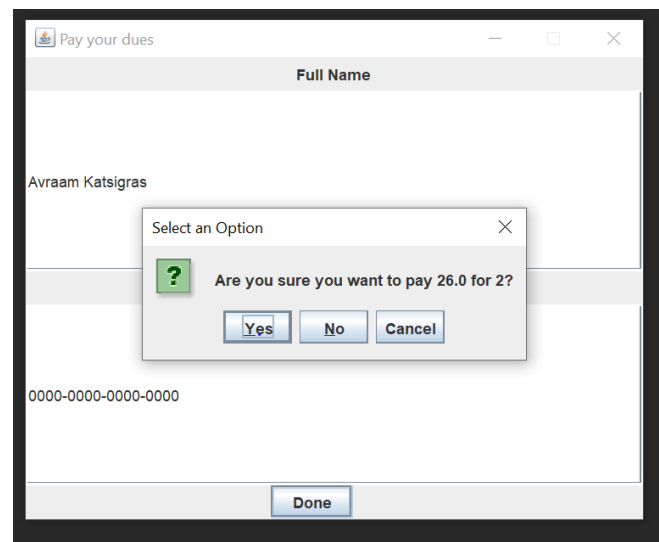
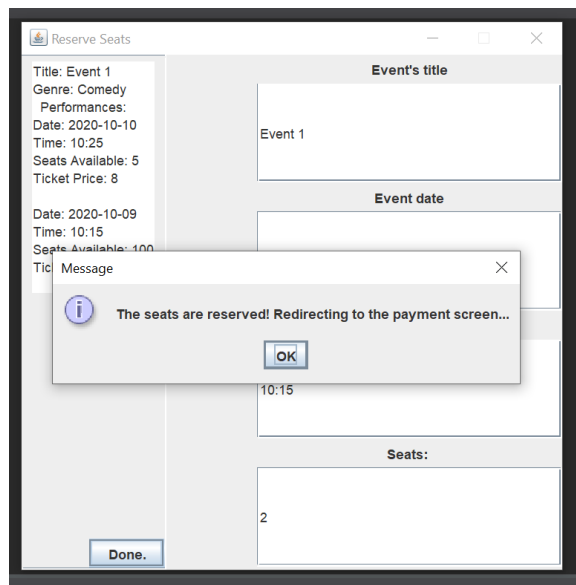
## Σενάριο Χρήστη:



## Κύρια οθόνη χρήστη



## Αναζήτηση θεαμάτων με βάση το είδος



## Κράτηση εισιτηρίων

## Ζητούμενο 2°:

Ένα παράδειγμα συνθήκης ανταγωνισμού είναι στην επικοινωνία μεταξύ εξυπηρετητή και βάσης, όταν οποιοσδήποτε από τους δύο στέλνει το μήνυμά του και προσπαθεί να πάρει απάντηση, χωρίς να υπάρχει κατευθείαν. Αυτό το αντιμετωπίσαμε συγχρονίζοντας τις μεθόδους που αφορούσαν την διάδοση και παραγωγή μηνυμάτων. Έτσι τρέχει μία μέθοδος τη φορά, αυτή που έχει τους κατάλληλους πόρους για να τρέξει, ενώ οι υπόλοιπες περιμένουν να αποδεσμευτούν ή να παραχθούν αυτοί οι πόροι.

Ένα ακόμη παράδειγμα συνθήκης ανταγωνισμού είναι η περίπτωση στην οποία δύο χρήστες προσπαθούν να κλείσουν τα τελευταία εισιτήρια παράλληλα, ή όταν ο διαχειριστής προσπαθεί να αποσύρει ένα θέαμα την ίδια στιγμή που ο χρήστης κλείνει τις θέσεις. Αυτά τα σενάρια δεν τα έχουμε καλύψει λόγω χρονική ανεπάρκειας.

## Ζητούμενο 3°:

Το μοντέλο πελάτης-εξυπηρετητής-βάση έχει ως πλεονέκτημα την αποσυμφόρηση του βασικού εξυπηρετητή όταν η βάση επεξεργάζεται αρχεία και αναγκάζεται να περιμένει. Στον χρόνο που περιμένει την βάση μπορούν να εξυπηρετηθούν και άλλοι πελάτες, πράγμα που βγάζει περισσότερο νόημα από το να κάνει ένας εξυπηρετητής και τις επεξεργασίες στα αρχεία και την επικοινωνία με τους χρήστες.

Βρήκαμε την RMI πιο εύχρηστη από τα Sockets στην επικοινωνία πελάτη-εξυπηρετητή επειδή μέσω της διεπαφής ο χρήστης μπορεί πολύ εύκολα να κάνει τις διαδικασίες που χρειάζεται από το να στέλνει μηνύματα στον εξυπηρετητή και να περιμένει την απάντηση. Η RMI είναι υψηλότερου επιπέδου σε σχέση με τα Sockets για αυτόν ακριβώς τον λόγο. Ο χρήστης ασχολείται με ένα αντικείμενο και δεν χρειάζεται να ασχοληθεί με πρωτόκολλα πράγμα που κάνει και την επικοινωνία πολύ ταχύτερη. Ένα ακόμη πλεονέκτημα είναι ότι με την RMI δύναται πολιτική ασφαλείας μέσω της οποίας ελέγχεται καλύτερα η πρόσβαση στον εξυπηρετητή κάτι που με Sockets θα ήταν σίγουρα δυσκολότερο.