



1^η ΟΜΑΔΙΚΗ ΕΡΓΑΣΙΑ

Θέμα – Ανάπτυξη καταναεμημένου συστήματος με Java Sockets και RMI

ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ

Το καταναεμημένο σύστημα που θέλουμε να αναπτύξουμε ακολουθεί το μοντέλο πελάτη - εξυπηρετητή - εξυπηρετητή και αφορά μια απλοποιημένη εφαρμογή η οποία θα δίνει τη δυνατότητα αγοράς εισιτηρίων για διάφορα θεάματα όπως θεατρικές παραστάσεις, συναυλίες, αθλητικά γεγονότα και άλλες εκδηλώσεις μέσω του διαδικτύου.

Η εφαρμογή των κρατήσεων αποτελείται από δύο εξυπηρετητές (ο 1ος εξυπηρετητής διαχειρίζεται το **Κεντρικό Σύστημα Κρατήσεων** και ο 2ος εξυπηρετητής την **Κεντρική Βάση Δεδομένων**) και πολλούς χρήστες που μπορεί να είναι ή οι διαχειριστές του συστήματος ή οι πιθανοί αγοραστές των εισιτηρίων.

Ο 1^{ος} εξυπηρετητής αναλαμβάνει όλη την επικοινωνία με τους χρήστες (πρέπει να υποστηρίζει την ταυτόχρονη εξυπηρέτηση πολλών χρηστών) και είναι αυτός που είναι υπεύθυνος και για την εγγραφή αλλά και σύνδεση των χρηστών στο σύστημα, δηλαδή εκτελεί διαδικασίες αυθεντικοποίησης. Για να εξυπηρετήσει τους χρήστες θα πρέπει να επικοινωνεί και με τον 2^ο εξυπηρετητή, ο οποίος είναι αυτός που διαθέτει αρχείο με όλες τις πληροφορίες για τα διαθέσιμα θεάματα/εκδηλώσεις. Οι τεχνολογίες που θα πρέπει να χρησιμοποιηθούν για την δημιουργία του συστήματος είναι **Java RMI** για την σύνδεση των πελατών με τον 1^ο εξυπηρετητή και **sockets** για την σύνδεση του 1ου εξυπηρετητή με τον 2ο εξυπηρετητή.

Οι βασικές λειτουργίες που πρέπει να παρέχει μια τέτοια εφαρμογή είναι:

Η δυνατότητα εγγραφής αλλά και διαγραφής ενός χρήστη

Όλοι οι χρήστες που εγγράφονται στο σύστημα καταχωρούν τα προσωπικά τους στοιχεία (ονοματεπώνυμο, τηλέφωνο επικοινωνίας, email διεύθυνση) καθώς και το όνομα σύνδεσης (login name), το συνθηματικό τους και τον ρόλο τους. Στην εφαρμογή αυτή, οι χρήστες μπορεί να έχουν δύο ρόλους. Αυτόν του *διαχειριστή* (admin) και του *απλού χρήστη* (user).

Αν οι χρήστες αιτηθούν την διαγραφή τους από το σύστημα, θα πρέπει η εφαρμογή να αποδεχθεί το αίτημα τους και ακολούθως να τους διαγράψει από το σχετικό αρχείο.

Την διαδικασία της εγγραφής και διαγραφής των χρηστών την αναλαμβάνει αποκλειστικά ο 1^{ος} εξυπηρετητής ο οποίος διατηρεί αρχείο των χρηστών της εφαρμογής.

Η σύνδεση χρήστη με σκοπό την χρήση των παρεχόμενων υπηρεσιών της εφαρμογής αλλά και η αποσύνδεση του

Ο χρήστης συνδέεται στο σύστημα συμπληρώνοντας το όνομα του και τον κωδικό του. Για να ολοκληρωθεί η σύνδεση θα πρέπει να γίνει έλεγχος των στοιχείων που έδωσε ο χρήστης, από τον 1ο εξυπηρετητή του συστήματος και να δοθεί η σχετική έγκριση.

Όταν ο χρήστης ολοκληρώσει τις εργασίες του μπορεί να προχωρήσει στη διαδικασία της αποσύνδεσης.

Η καταχώριση σχετικού θεάματος/εκδήλωσης

Η καταχώριση νέου θεάματος επιτρέπεται αποκλειστικά από τους χρήστες που έχουν τον ρόλο του διαχειριστή. Ο διαχειριστής θα πρέπει να συμπληρώσει όλα τα απαραίτητα στοιχεία για το νέο θέαμα



που θέλει να καταχωρήσει (π.χ. Τίτλος, Είδος, Ημερομηνία έναρξης, διαθέσιμες θέσεις, κλπ.). Θα πρέπει να λάβετε υπόψη σας ότι ένα θέαμα μπορεί να έχει περισσότερες και από μία παραστάσεις. Συνεπώς σε αυτή την περίπτωση θα πρέπει να ορίσει για κάθε μία παράσταση ημερομηνία, ώρα αλλά και το κόστος (μπορεί να είναι διαφορετικό ανά παράσταση για κάποιες συγκεκριμένες ώρες και ημέρες).

Για την ολοκλήρωση αυτής της λειτουργίας ο χρήστης επικοινωνεί μέσω γραφικού περιβάλλοντος με τον 1^ο εξυπηρετητή του συστήματος. Δηλαδή το αίτημα του και τα δεδομένα που εισάγει αποστέλλονται στον 1^ο εξυπηρετητή και αυτός αναλαμβάνει να επικοινωνήσει με τον 2^ο εξυπηρετητή που διαχειρίζεται τα αρχεία με τις εκδηλώσεις. Ο 2^{ος} εξυπηρετητής επικοινωνεί αποκλειστικά μόνο με τον 1^ο. Τον ενημερώνει για το εάν είναι δυνατή ή όχι η ολοκλήρωση της ενέργειας αποστέλλοντας του κατάλληλα μηνύματα και ενημερώνει αντίστοιχα τα απαιτούμενα αρχεία. Αυτή η διαδικασία επαναλαμβάνεται και για όλες τις λειτουργίες που περιγράφονται ακολούθως.

Η απενεργοποίηση θεάματος

Όπως και η προηγούμενη λειτουργία η απενεργοποίηση ενός θεάματος επιτρέπεται να γίνει μόνο από τον διαχειριστή του συστήματος. Η απενεργοποίηση μπορεί να γίνει είτε επειδή για κάποιο λόγο ακυρώθηκε η εκδήλωση είτε επειδή έχουν ολοκληρωθεί όλες οι παραστάσεις του θεάματος.

Η παραγγελία εισιτηρίου και η εξόφληση του

Ο απλός χρήστης θα πρέπει να έχει τη δυνατότητα να αναζητήσει τα διαθέσιμα θεάματα. Επιλέξτε εσείς με ποια κριτήρια έχει νόημα να γίνουν οι αναζητήσεις. Ακολούθως επιλέγει το θέαμα και την παράσταση που τον ενδιαφέρει και παραγγέλλει εισιτήρια. Ο χρήστης μπορεί να παραγγείλει εισιτήρια για όσους ανθρώπους επιθυμεί αρκεί να υπάρχουν διαθέσιμες θέσεις. Η παραγγελία εισιτηρίων θα πρέπει να έχει ως αποτέλεσμα την δέσμευση των θέσεων στο αντίστοιχο θέαμα. Ο χρήστης άμεσα μετά τη δέσμευση των θέσεων οφείλει να πληρώσει το κόστος των εισιτηρίων.

Ο 1^{ος} εξυπηρετητής (μετά την επικοινωνία του με τον 2^ο) ενημερώνει τον χρήστη για το συνολικό ποσό πληρωμής και ο χρήστης στέλνει το ονοματεπώνυμο του και τον αριθμό της πιστωτικής κάρτας του. Ακολούθως ενημερώνεται για την αποδοχή της πληρωμής και ολοκληρώνεται η διαδικασία (δεν απαιτείται να εκτελέσετε κάποια πρόσθετη λειτουργία για τη διαδικασία της πληρωμής).

Αν ο χρήστης έκανε κράτηση σε ημέρα εκτός σαββατοκύριακου και μετά την ολοκλήρωση της πληρωμής του, απομένουν για την συγκεκριμένη εκδήλωση-παράσταση λιγότερο από 10 εισιτήρια, ο 1^{ος} εξυπηρετητής οφείλει να ενημερώσει όλους τους χρήστες που είναι συνδεδεμένοι την τρέχουσα χρονική στιγμή ότι για την συγκεκριμένη παράσταση κάνει έκπτωση 40% για τις εναπομείναντες θέσεις (χρήση κλήσεων επιστροφής(callback)).

Η ακύρωση παραγγελίας

Ο χρήστης θα πρέπει να έχει τη δυνατότητα να ακυρώσει την παραγγελία του. Σε αυτή την περίπτωση πρέπει το σύστημα να αποδεσμεύσει όποιες και όσες θέσεις είχε δεσμεύσει όταν έκανε την παραγγελία ο χρήστης.

Η ακύρωση δεν μπορεί να γίνει στην περίπτωση που ζητείται την ίδια μέρα που προβάλλεται η εκδήλωση.

ΖΗΤΟΥΜΕΝΑ

Α) Υλοποίηση [65 μονάδες]

Να υλοποιηθεί το παραπάνω σύστημα σε Java χρησιμοποιώντας RMI και Sockets για την διαδιεργασιακή επικοινωνία. Στην αναφορά σας θα πρέπει να περιγράψετε σύντομα:



- α) τις βασικές σχεδιαστικές αποφάσεις που έχετε λάβει για την υλοποίηση της εφαρμογής,
- β) τυχόν παραδοχές που έχετε κάνει, πέραν των προδιαγραφών που σας έχουν δοθεί,
- γ) οδηγίες για την εκτέλεση τόσο της εφαρμογής του πελάτη όσο και των εξυπηρετητών και
- δ) οθόνες εκτέλεσης της εφαρμογής με διαφορετικά σενάρια.

B) Συνθήκες ανταγωνισμού (race conditions) [20 μονάδες]

Στην περιγραφή του παραπάνω σεναρίου, μπορείτε να εντοπίσετε κάποιες καταστάσεις που μπορούν να οδηγήσουν σε συνθήκες ανταγωνισμού; Περιγράψτε σενάρια εκτέλεσης της εφαρμογής που να δείχνουν την εμφάνιση συνθηκών ανταγωνισμού και πως τις αντιμετωπίσατε; Φροντίστε για την αντιμετώπιση αυτών των καταστάσεων στην υλοποίηση της εφαρμογής σας, με μηχανισμούς που διαθέτει η Java.

Γ) Αρχιτεκτονική/Τεχνολογία Υλοποίησης [10 μονάδες]

Ποια θεωρείται ότι είναι τα πλεονεκτήματα της αρχιτεκτονικής (πελάτης - εξυπηρετητής - εξυπηρετητής) με την οποία υλοποιείται η συγκεκριμένη κατανεμημένη εφαρμογή έναντι του απλού μοντέλου πελάτη/εξυπηρετητή; Στην παρούσα εργασία χρησιμοποιείται RMI και Sockets για την διασύνδεση των οντοτήτων του συστήματος σας. Ποια θεωρείτε ότι είναι τα πλεονεκτήματα της υλοποίησης της εφαρμογής με RMI έναντι της υλοποίησης με Sockets;

Δ) Ασφάλεια [5 μονάδες]

Ένα ζήτημα σε μια ρεαλιστική υλοποίηση της συγκεκριμένης εφαρμογής είναι η διαχείριση της ασφάλειας του συστήματος κατά την αλληλεπίδραση πελάτη - εξυπηρετητή αλλά και στην επικοινωνία μεταξύ των δύο εξυπηρετητών. Περιγράψτε μηχανισμούς που μπορούν να χρησιμοποιηθούν για την προστασία του συστήματος. Το ζητούμενο αυτό δεν θα πρέπει να υλοποιηθεί στην τρέχουσα έκδοση της εφαρμογής.

Υποδείξεις για την υλοποίηση

- Η υλοποίηση θα περιλαμβάνει **τρία διαφορετικά projects**, το πρόγραμμα του πελάτη, το πρόγραμμα του Κεντρικού Συστήματος Κρατήσεων (1ος εξυπηρετητής) και της Κεντρικής Βάσης Δεδομένων (2ος εξυπηρετητής).
- Οι πελάτες επικοινωνούν κάνοντας χρήση Java RMI, ενώ οι εξυπηρετητές χρησιμοποιώντας Java sockets για την διαδιεργασιακή επικοινωνία.
- Ο 1^{ος} εξυπηρετητής μπορεί να εξυπηρετεί ταυτόχρονα πολλούς διαφορετικούς πελάτες.
- Θα χρειαστεί να ορίσετε ένα request/reply πρωτόκολλο επικοινωνίας μεταξύ του πελάτη, του 1^{ου} εξυπηρετητή και του 2^{ου} εξυπηρετητή και τη μορφή των μηνυμάτων που ανταλλάσσονται.
- Οι λειτουργίες που θα υλοποιήσετε για την επικοινωνία του 1^{ου} εξυπηρετητή με τον 2^ο θα πρέπει να υλοποιηθούν κατάλληλα τόσο στην πλευρά του 1^{ου} όσο και στην πλευρά του 2^{ου} εξυπηρετητή.
- Ο χρήστης επιλέγει την λειτουργία/ που επιθυμεί μέσα από ένα απλό μενού που παρουσιάζεται στην οθόνη. Μόνο για τους πελάτες θα πρέπει υλοποιήσετε γραφική διεπαφή.
- Το γραφικό περιβάλλον δεν θα πρέπει να είναι αυτοματοποιημένο από σχετικό εργαλείο (π.χ. Netbeans) αλλά υλοποιημένο από εσάς.
- Εφόσον δεν έχετε δίκτυο υπολογιστών ώστε να τρέξετε τις διεργασίες πελάτη και εξυπηρετητή σε διαφορετικούς κόμβους μπορείτε να χρησιμοποιήσετε την διεύθυνση localhost (IP address 127.0.0.1) η οποία δίνει την ψευδαισθηση ενός δικτύου με έναν κόμβο. Ανοίξτε διαφορετικά παράθυρα για τους πελάτες και τερματικά παράθυρα για τους δύο εξυπηρετητές.



Πανεπιστήμιο Αιγαίου
Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Κατανεμημένα Συστήματα - 1^η Ομαδική Εργασία
Ημερομηνία Παράδοσης : Τρίτη 12/05/2020

Εργαστηριακοί Διδάσκοντες: Δούμα Αναστασία

- Διασφαλίστε ότι κάθε φορά που μια σύνδεση με τον εξυπηρετητή τερματίζεται απελευθερώνονται οι πόροι του συστήματος (δυναμική μνήμη, socket descriptors, κ.α.).

Γενικές υποδείξεις

- 1) Η εργασία μπορεί να εκπονηθεί από ομάδα φοιτητών (1 ή 2 φοιτητές).
- 2) Η αποστολή της εργασίας θα πρέπει να γίνει μέχρι την προκαθορισμένη ημερομηνία με χρήση της ηλεκτρονικής πλατφόρμας e-class. Εκπρόθεσμη υποβολή μέχρι μια εβδομάδα χάνει το 25% της βαθμολογίας, ενώ πάνω από μια εβδομάδα δεν γίνεται δεκτή.
- 3) Να αναφέρετε τις βιβλιογραφικές πηγές που τυχόν έχετε χρησιμοποιήσει.
- 4) Η υποβολή κοινών απαντήσεων από διαφορετικούς φοιτητές ή διαφορετικές ομάδες φοιτητών δεν επιτρέπεται και θεωρείται ως **ΑΝΤΙΓΡΑΦΗ**. Η αντιγραφή έχει ως αποτέλεσμα το ΜΗΔΕΝΙΣΜΟ ΤΗΣ ΕΡΓΑΣΙΑΣ ΣΥΝΟΛΙΚΑ.

Υποδείξεις/κανόνες για τη συγγραφή και υποβολή της εργασίας

- 5) Μόνο το ένα μέλος της ομάδας θα πρέπει να στείλει την εργασία με μορφή συμπιεσμένου αρχείου zip ή rar. Το όνομα του αρχείου θα είναι: DSPROJECTicsdxxxxx.<rar|zip>. Το zip|rar αρχείο θα περιλαμβάνει το pdf αρχείο της αναφοράς σας και τα project της υλοποίησης σας.
- 6) Ο πηγαίος κώδικας θα αξιολογηθεί ως προς το αν υλοποιεί τα βασικά ζητούμενα της εκφώνησης, εκτελείται χωρίς να προκύπτουν σφάλματα λογισμικού (bugs), ακολουθεί «καλές αρχές» προγραμματισμού (π.χ. σχολιασμό, στοίχιση, εύγλωττη ονοματοδοσία μεταβλητών, επαναχρησιμοποίηση κώδικα, κλπ).

Η εφαρμογή των παραπάνω κανόνων είναι **ΥΠΟΧΡΕΩΤΙΚΗ** και βαθμολογείται σύμφωνα με το αντίστοιχο κριτήριο αξιολόγησης.