

Get Trained Get Hired

Accenture – ΚΕΔΙΒΙΜ Πανεπιστημίου Μακεδονίας

Δεκέμβριος 2024

PROJECT DEVELOPMENT

Σύστημα διαχείρισης εκδηλώσεων

1. Γενικές πληροφορίες

Στο πλαίσιο συνεργασίας της ομάδας ανάπτυξης με τον Δήμο Θεσσαλονίκης, ζητήθηκε η ανάπτυξη ενός ολοκληρωμένου συστήματος διαχείρισης εκδηλώσεων. Το σύστημα αυτό θα παρέχει δυνατότητες σε διαφορετικούς τύπους χρηστών:

1. Υπάλληλοι Δήμου:

- Να διαχειρίζονται τη συνολική λίστα εκδηλώσεων του Δήμου (διαγραφή και έγκριση εκδηλώσεων).
- Να βλέπουν αναφορές με την κατάσταση των εκδηλώσεων (π.χ., **αριθμός εγγραφών, εκκρεμείς εγκρίσεις**).

2. Διοργανωτές Εκδηλώσεων:

- Να **καταχωρούν νέες εκδηλώσεις που θα πραγματοποιήσουν** (απαιτείται έγκριση από υπάλληλο του δήμου).
- Να έχουν την **δυνατότητα διαγραφής εκδήλωσής τους** (απαιτείται έγκριση από υπάλληλο του δήμου).
- Να βλέπουν τη λίστα συμμετεχόντων στις εκδηλώσεις τους.

3. Επισκέπτες:

- Να αναζητούν εκδηλώσεις βάσει κριτηρίων (ημερομηνία, τοποθεσία, θεματική ενότητα).
- Να κάνουν κράτηση για εκδηλώσεις.
- Να ακυρώνουν κρατήσεις.

2. Λειτουργικές απαιτήσεις

Γενική περιγραφή

Ζητείται η δημιουργία ενός συστήματος διαχείρισης εκδηλώσεων. Οι επισκέπτες θα μπορούν να παρακολουθήσουν τις διάφορες εκδηλώσεις που πραγματοποιούνται στον Δήμο Θεσσαλονίκης. Ο δήμος διαθέτει πολλές εκδηλώσεις, όπου η κάθε μία έχει και έναν διοργανωτή. Ένας διοργανωτής φυσικά μπορεί να πραγματοποιεί παραπάνω από μία εκδηλώσεις.

Κάθε εκδήλωση διαθέτει μια σειρά από σχετικές πληροφορίες, όπως τον τίτλο της, την ημερομηνία πραγματοποίησης, την τοποθεσία στην οποία θα διεξαχθεί, την θεματική ενότητα

στην οποία ανήκει, την ώρα έναρξης, μέγιστη χωρητικότητα, την διάρκειά της, την περιγραφή της, κτλ.

Ένας διοργανωτής μπορεί να δημιουργεί μία νέα εκδήλωση ή να διαγράφει μία ήδη υπάρχουσα δικιά του εκδήλωση. Για την δημιουργία ή διαγραφή εκδήλωσης, απαιτείται η έγκριση ενός υπαλλήλου.

Ένας υπάλληλος μπορεί ενημερώνεται για αιτήσεις προσθήκης ή διαγραφής εκδηλώσεων και έπειτα να τις εγκρίνει ή να τις απορρίπτει. Επίσης, θα μπορεί να διαγράψει μία τρέχουσα εκδήλωση.

Ο πολίτης θα μπορεί να αναζητά εκδηλώσεις βάσει συγκεκριμένων κριτηρίων, όπως την ημερομηνία, την τοποθεσία ή την θεματική ενότητα της εκδήλωσης. Έπειτα, εφόσον θέλει να παρακολουθήσει κάποια εκδήλωση, θα μπορεί να κάνει κράτηση. Σε περίπτωση που αλλάξει γνώμη μπορεί να ακυρώσει την κράτηση που έχει κάνει.

Οντότητες που εμπλέκονται και οι ιδιότητές τους

Δίνονται για ευκολία οι οντότητες του προβλήματος και οι ιδιότητές τους.

- Visitor (name, surname, email)
- Event (title, theme, description, location, maxCapacity, day, month, year, hour, minutes, duration, Organizer, status)
- Reservation (**Visitor** , **Event**)
- Organizer (afm, name, surname, description, **Events**)
- ApprovalRequest (type, Event, submittedBy, createdAt, status, handledBy, closedAt, comments)
- Employee (name, surname, email)

2.1 Μοντελοποίηση των οντοτήτων (Domain Entities)

Για την επιβεβαίωση της σωστής μοντελοποίησης των οντοτήτων ζητείται να υλοποιήσετε στη γλώσσα προγραμματισμού Java τις ανωτέρω οντότητες καθώς και τις μεταξύ τους σχέσεις. Δημιουργείτε επίσης τις κατάλληλες μεθόδους όπου αυτές χρειάζονται για να καλυφθούν σωστά τα δοκιμαστικά σενάρια που ακολουθούν. **Είστε ελεύθεροι, αν το κρίνετε απαραίτητο, να προσθέσετε νέες οντότητες, περισσότερες ιδιότητες ή δομές δεδομένων στις οντότητες σας.**

Στη συνέχεια δημιουργείτε στη main τα εξής αντικείμενα με τυχαίες τιμές στις ιδιότητές τους:

10 επισκέπτες. Προσθέστε τους σε μία λίστα.

3 υπαλλήλους.

4 διοργανωτές.

2 εκδηλώσεις για κάθε διοργανωτή.

Δημιουργήστε μία νέα εκδήλωση για τους 2 από τους 4 διοργανωτές και διαγράψτε μία από τις εκδηλώσεις των άλλων 2 οργανωτών.

Επιβεβαιώστε (με διαφορετικό υπόλληλο) μία διαγραφή και μία προσθήκη εκδήλωσης.

Κάντε εγγραφή παρακολούθησης εκδήλωσης τουλάχιστον σε μία εκδήλωση για όλους τους επισκέπτες.

Δοκιμάστε τι γίνεται σε περίπτωση που ένας επισκέπτης προσπαθήσει να εγγραφεί σε μία εκδήλωση που έχει εξαντλημένες όλες τις διαθέσιμες θέσεις.

Δοκιμάστε ότι άλλη λειτουργικότητα έχετε υλοποιήσει.

Εκτυπώστε στην κονσόλα (System.out.println) και σε αρχείο (event-agenda.txt) τα παρακάτω:

- Τις επικείμενες εκδηλώσεις για κάθε οργανωτή.
- Τις λίστες με τους εγγεγραμμένους επισκέπτες κάθε εκδήλωσης.
- Τις διαγραφές και προσθήκες εκδηλώσεων που αναμένουν έγκριση.
- Τις εγκεκριμένες αιτήσεις διαγραφής και προσθήκης εκδηλώσεων.

2.2 Σενάρια χρήσης

Αφού επιβεβαιώσατε τη σωστή μοντελοποίηση και αλληλεπίδραση των αντικειμένων με δοκιμαστικές κλήσεις στη main, καλείστε τώρα να επεκτείνετε την υλοποίηση δημιουργώντας την Back-end υποδομή του συστήματος. Οι λειτουργικές απαιτήσεις των οθονών που ακολουθούν παρουσιάζονται από την πλευρά του τελικού χρήστη. Πρέπει να κατανοήσετε τις απαιτήσεις και να δημιουργήσετε το back-end τμήμα της εφαρμογής (REST Controllers) ώστε να είναι λειτουργικό το front-end.

Σελίδα εκδηλώσεων

Αναζήτηση εκδήλωσης: Ο επισκέπτης θα μπορεί να αναζητήσει βλέπει τις διαθέσιμες εκδηλώσεις.

Nice to have:

- Η αναζήτηση να μπορεί να γίνει βάση διάφορων φίλτρων, όπως θεματική, ημερομηνία και τοποθεσία εκδήλωσης.

Δέσμευση θέσης σε εκδήλωση: Ο επισκέπτης θα δημιουργεί μία κράτηση για κάποια εκδήλωση δίνοντας τον τίτλο της.

Σελίδα διαχείρισης κρατήσεων

Διαχείριση κρατήσεων επισκέπτη: Ο επισκέπτης θα μπορεί να δει τις κρατήσεις που έχει κάνει και να ακυρώνει κάποιες από αυτές.

Nice to have:

- QR code για την κράτηση του επισκέπτη
- Δημιουργία ενός PDF με τις κρατήσεις των εκδηλώσεων

Σελίδα διαχείρισης εκδηλώσεων

Προβολή εκδηλώσεων: Ο διοργανωτής θα μπορεί να βλέπει τις εκδηλώσεις που έχει, καθώς και σχετικές πληροφορίες με αυτές (κατάσταση στην οποία βρίσκονται, πληρότητα, κτλ).

Προσθήκη εκδήλωσης: Ο διοργανωτής θα μπορεί να προσθέτει μία νέα εκδήλωση, δίνοντας τις απαραίτητες πληροφορίες.

Nice to have:

- Η προσθήκη να γίνεται μέσω της χρήσης JSON αρχείου.

Διαγραφή εκδήλωσης: Ο διοργανωτής θα μπορεί να διαγράφει μία εκδήλωση.

Σελίδα διαχείρισης υπαλλήλου

Διαγραφή εκδήλωσης: Ο υπάλληλος θα έχει την δυνατότητα διαγραφής εκδήλωσης, χωρίς να απαιτείται έγκριση.

Nice to have:

- Να καταγράφετε η διαγραφή στο ιστορικό του υπαλλήλου.

Σελίδα διαχείρισης αιτήσεων

Προβολή αιτήσεων: Ο υπάλληλος θα μπορεί να βλέπει όλες τις ενεργές αιτήσεις που εκκρεμούν.

Nice to have:

- Ο υπάλληλος να μπορεί να δει τις αιτήσεις που έχει επεξεργαστεί στο παρελθόν.

Έγκριση αίτησης: Ο υπάλληλος θα μπορεί να εγκρίνει μία αίτηση προσθήκης η διαγραφής

Απόρριψη αίτησης: Ο υπάλληλος θα μπορεί να απορρίπτει μία αίτηση προσθήκης η διαγραφής

Μη-Λειτουργικές απαιτήσεις

1. Programming environment
 - a. Java 11
 - b. Java SE
 - c. Spring boot
2. Ο κώδικας θα πρέπει να αναπτυχθεί και να παραδοθεί χρησιμοποιώντας GitHub repository.